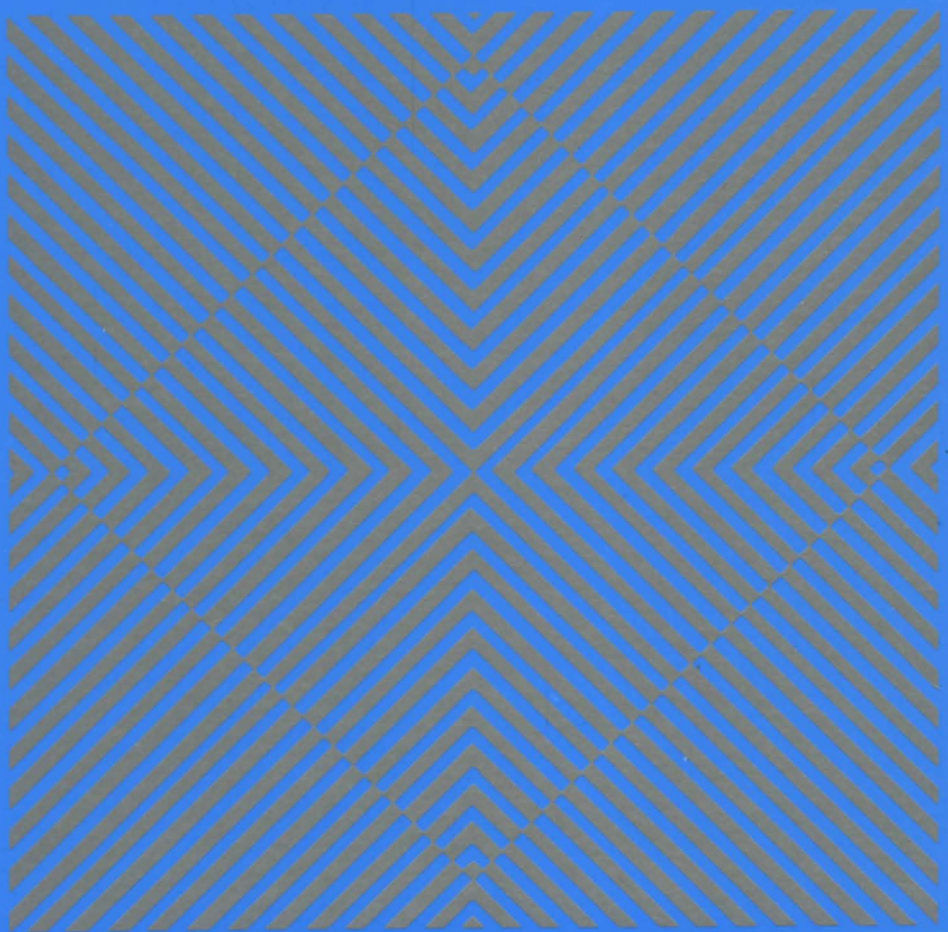
MITSUBISHI SEMICONDUCTORS

M37702 Group
M37703 Group

USER'S MANUAL



MITSUBISHI ELECTRIC

Preface

This manual describes the hardware of the Mitsubishi CNOS 16-bit microcomputer M37702 group. After reading this manual, the user should be able to fully utilize the functions of the microcomputers of M37702 group and M37703 group.

For details concerning the softwares for the M37702 group and M37703 group, refer to the MELPS 7700 SOFTWARE MANUAL. For details concerning the development support tools (assembler, option boards), refer to the respective operation manuals.

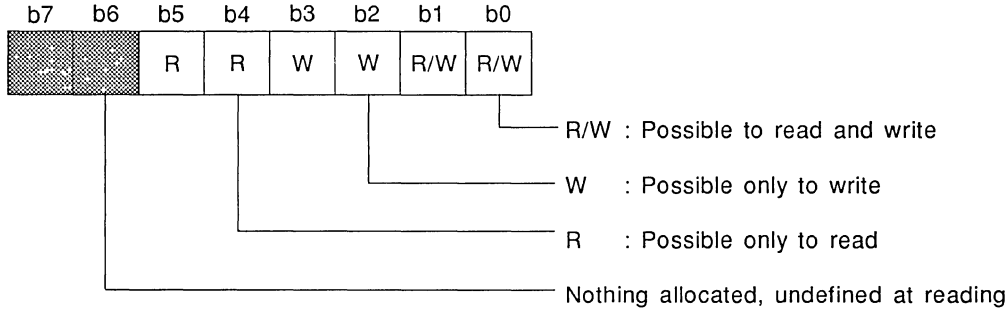
Table of contents

APPENDIX

For using this manual

This manual defines following items.

•f(XIN)
f(XIN) means oscillating clock frequency.
•Internal clock ϕ
Internal clock ϕ means operating clock of this microcomputer. It is obtained by dividing the input clock to XIN pin by 2 (= f(XIN)/2).
•Clock ϕ_1
Clock ϕ_1 means the internal clock ϕ output from P42 pin.
•Bit attribute
Bit attributes are described in the figure of register structure.
The following abbreviations are used to indicate the attributes.



•Overflow and Underflow of timers
Overflow of the timer means that the counter content reaches FFFF16 → reload value "n".
Underflow of the timer means that the counter content reaches 000016 → reload value "n".

"n" : Value set in reload register

CHAPTER 1
DESCRIPTION

DESCRIPTION

The M37702M2-XXXFP is a 16-bit single-chip microcomputer designed with high-performance CMOS silicon gate technology. It is housed in an 80-pin plastic molded flat package.

This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.

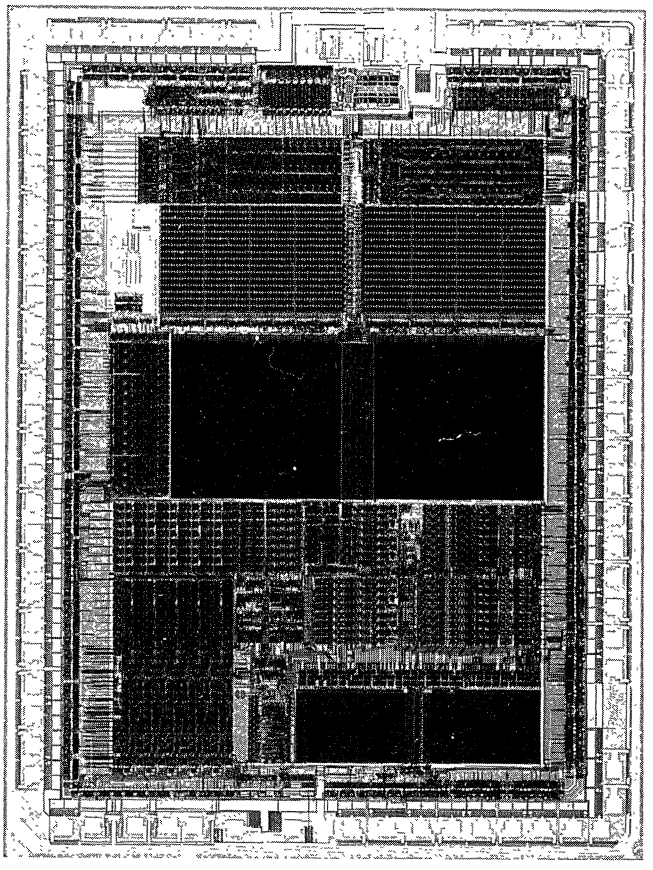


Photo of M37702M2-XXXFP Chip

1.1 M37702 group

The M37702 group consists of chips shown in Table 1.1.1 with the M37702M2-XXXFP as the base chip. These chips are all pin compatible and provide a variety of memory characteristics, memory size, and operating clock frequencies to enable the user to select the chip best suited for his system. Hereafter, the M37702 group microcomputers will be referred to simply as the M37702 unless there is a specific difference by version.

Table 1.1.1 M37702 group

Type name	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)	Remarks
M37702M2-XXXFP	16K (Mask ROM)	512	8	
M37702M2AXXXFP			16	High-speed version of M37702M2-XXXFP
M37702M2BXXXFP			25	Super high-speed version of M37702M2-XXXFP
M37702S1FP	—		8	External ROM version of M37702M2-XXXFP
M37702S1AFP			16	External ROM version of M37702M2AXXXFP
M37702S1BFP			25	External ROM version of M37702M2BXXXFP
M37702E2-XXXFP	16K (One time PROM)		8	One time PROM version of M37702M2-XXXFP
M37702E2AXXXFP			16	One time PROM version of M37702M2AXXXFP
M37702E2BXXXFP			25	One time PROM version of M37702M2BXXXFP
M37702E2FS	16K (EPROM)		8	EPROM version of M37702M2-XXXFP
M37702E2AFS			16	EPROM version of M37702M2AXXXFP
M37702E2BFS			25	EPROM version of M37702M2BXXXFP
M37702M4-XXXFP	32K (Mask ROM)	2048	8	Memory expansion version of M37702M2-XXXFP
M37702M4AXXXFP			16	Memory expansion version of M37702M2AXXXFP
M37702M4BXXXFP			25	Memory expansion version of M37702M2BXXXFP
M37702S4FP	—		8	External ROM version of M37702M4-XXXFP
M37702S4AFP			16	External ROM version of M37702M4AXXXFP
M37702S4BFP			25	External ROM version of M37702M4BXXXFP
M37702E4-XXXFP	32K (One time PROM)		8	One time PROM version of M37702M4-XXXFP
M37702E4AXXXFP			16	One time PROM version of M37702M4AXXXFP
M37702E4BXXXFP			25	One time PROM version of M37702M4BXXXFP
M37702E4FS	32K (EPROM)		8	EPROM version of M37702M4-XXXFP
M37702E4AFS			16	EPROM version of M37702M4AXXXFP
M37702E4BFS			25	EPROM version of M37702M4BXXXFP

1.2 Performance overview

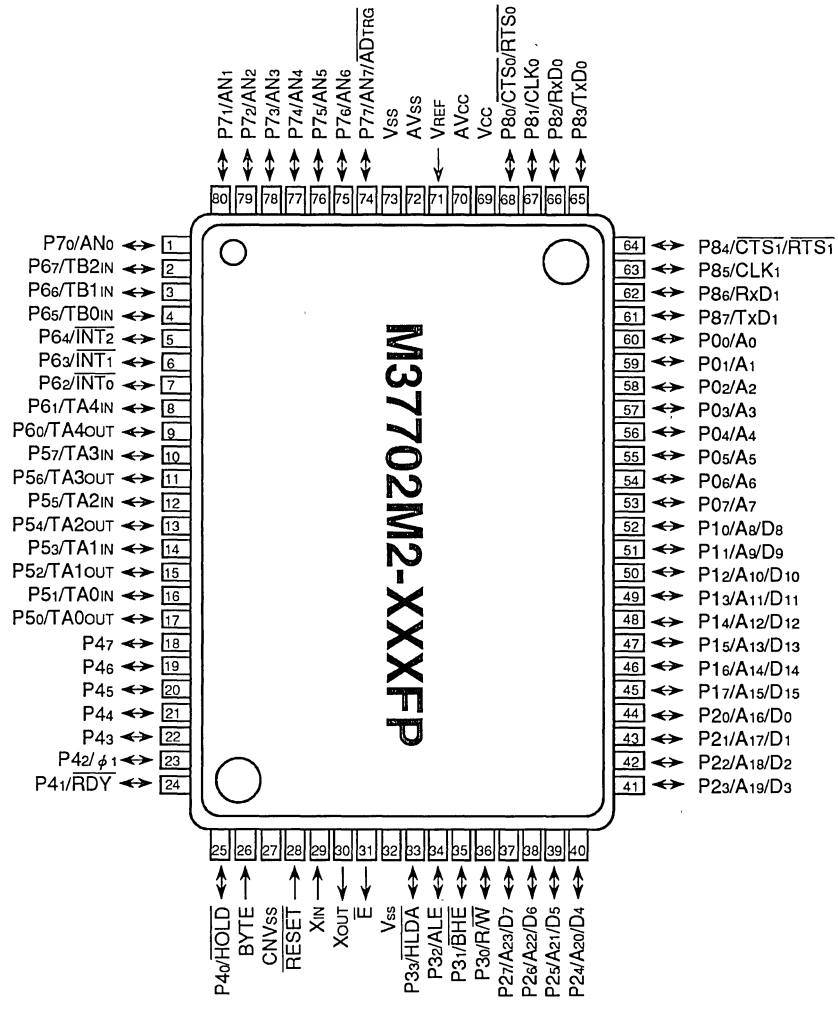
Table 1.2.1 shows the performance overview of the M37702M2-XXXFP/ M37702M2AXXXFP/ M37702M2BXXXFP.

Table 1.2.1 M37702M2-XXXFP / M37702M2AXXXFP / M37702M2BXXXFP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37702M2-XXXFP	500ns (the fastest instruction at 8MHz frequency)
	M37702M2AXXXFP	250ns (the fastest instruction at 16MHz frequency)
	M37702M2BXXXFP	160ns (the fastest instruction at 25MHz frequency)
Clock frequency	M37702M2-XXXFP	8MHz (maximum)
	M37702M2AXXXFP	16MHz (maximum)
	M37702M2BXXXFP	25MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 bytes
Input/Output ports	Ports P0–P2, P4–P8	8 bits × 8
	Port P3	4 bits × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits × 5
	TB0, TB1, TB2	16 bits × 3
Serial I/O		(UART or clock synchronous serial I/O) × 2
A-D converter		8 bits × 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 quartz crystal oscillator)
Supply voltage		5V±10%
Power dissipation		30mW (at external 8MHz frequency)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		–20 to 85°C
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

1.3 Pin configuration

Figure 1.3.1 shows the M37702M2-XXXFP pin configuration.



Outline 80P6N

Fig. 1.3.1 M37702M2-XXXFP pin configuration

1.4 Pin description

Table 1.4.1 shows the pin description.

Table 1.4.1 Pin description (1)

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5V±10% to V_{CC} and 0V to V_{SS}.
CNVss	CNVss input	Input	This pin controls the processor mode. Connect to Vss for single-chip mode. It must be connected to V_{CC} for external ROM types.
$\overline{RESET}$	Reset input	Input	The microcomputer is reset when this pin is set to "L" level.
X_{IN}	Clock input	Input	These are the I/O pins of the internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L" level.
BYTE	Bus width selection input	Input	When in memory expansion mode or microprocessor mode, this pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when the signal level is "L" and 8 bits when the signal level is "H".
AVcc, AVss	Analog supply input		Power supply for the A-D converter. Externally connect AVcc to Vcc and AVss to Vss.
V_{REF}	Reference voltage input	Input	This is a reference voltage input pin for the A-D converter.
$P0_0$–$P0_7$	I/O port P0	I/O	This port is a CMOS I/O port. An I/O direction register is available so that each pin can be programmed for input or output. Address (A_0–A_7) is output in memory expansion mode or microprocessor mode.
$P1_0$–$P1_7$	I/O port P1	I/O	This port is an 8-bit I/O port with the same function as P0. When the BYTE pin is set to "H" level in memory expansion mode or microprocessor mode, address (A_8–A_{15}) is output. In case the BYTE pin is set to "L" level, an address (A_8–A_{15}) is output when $\overline{E}$ pin level is "H", and high-order data (D_8–D_{15}) is input or output when $\overline{E}$ pin level is "L".
$P2_0$–$P2_7$	I/O port P2	I/O	This port is an 8-bit I/O port with the same function as P0. In memory expansion mode or microprocessor mode, an address (A_{16}–A_{23}) is output when $\overline{E}$ pin level is "H", and low-order data (D_0–D_7) is input or output when $\overline{E}$ pin level is "L".
$P3_0$–$P3_3$	I/O port P3	I/O	This port is a 4-bit I/O port with the same function as P0. In memory expansion mode or microprocessor mode, $P3_0$–$P3_3$ output R/$\overline{W}$, $\overline{BHE}$, ALE, and $\overline{HLDA}$ signals respectively.
$P4_0$–$P4_7$	I/O port P4	I/O	This port is an 8-bit I/O port with the same function as P0. In memory expansion mode or microprocessor mode, $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin respectively. $P4_2$ can be programmed for a ϕ_1 output pin. In microprocessor mode, $P4_2$ always outputs the clock ϕ_1.

Table 1.4.1 Pin description (2)

Pin	Name	Input/Output	Functions
$P5_0$–$P5_7$	I/O port P5	I/O	This port 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	This port 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, external interrupt input pins for $\overline{INT_0}$–$\overline{INT_2}$, and input pins for timers B0–B2.
$P7_0$–$P7_7$	I/O port P7	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can be programmed as analog input pins AN_0–AN_7. $P7_7$ also functions as the $\overline{AD_{TRG}}$ pin for an A-D conversion trigger.
$P8_0$–$P8_7$	I/O port P8	I/O	This port is an 8-bit I/O port with the same function as P0. These pins can be programmed as $\overline{CTS}$/$\overline{RTS}$, CLK, RxD, TxD pins for UART0 and UART1.

1.5 Block diagram

Figure 1.5.1 shows the M37702M2-XXXFP block diagram

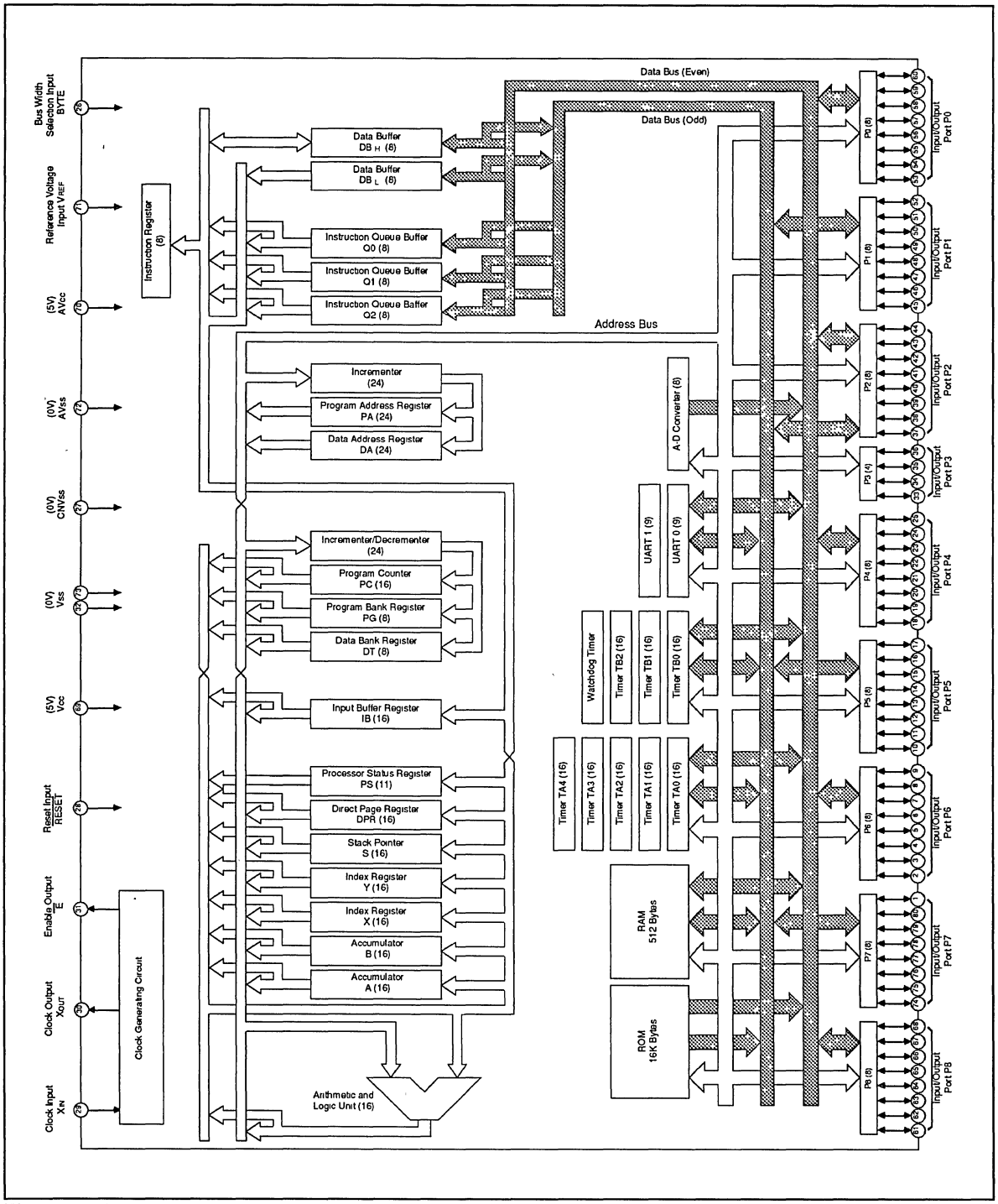


Fig. 1.5.1 M37702M2-XXXFP block diagram

CHAPTER 2
FUNCTIONAL DESCRIPTION

2.1 Central processing unit (CPU)

The MELPS 7700 CPU has ten registers as shown in Figure 2.1.1. Each of these registers is described below.

2.1.1 Accumulator (Acc)

Accumulators A and B are available and each can be used as 8-bit or 16-bit register as necessary.

(1) Accumulator A (A)

Accumulator A is the main register of the microcomputer. Data operations such as calculations, data transfer, and input/output are executed mainly through accumulator A. It consists of 16 bits and the low-order 8 bits can be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag m is "0" and as an 8-bit register when flag m is "1". Flag m is a part of the processor status register (PS) which is described later. When an 8-bit register is selected, the low-order 8 bits of the accumulator A are used and the contents of the high-order 8 bits are unchanged.

(2) Accumulator B (B)

Accumulator B has the same functions as accumulator A. The MELPS 7700 instructions can use accumulator B instead of accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A. Accumulator B is also controlled by the data length flag m.

2.1.2 Index register X (X)

Index register X consists of 16 bits and the lower 8 bits can 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. It is used as a 16-bit register when flag x is "0" and as an 8-bit register when flag x is "1". Flag x is a part of the processor status register (PS) which is described later. When an 8-bit register is selected, the low-order 8 bits of the index register X are used and the contents of the high-order 8 bits are unchanged.
In index addressing mode, register X is used as the index register and the contents of this address is added to obtain the real address.
Also, when executing a block transfer instruction **MVP** or **MVN**, the contents of the index register X indicate the low-order 16 bits of the source data address. The third byte of the **MVP** or **MVN** is the high-order 8 bits of the source data address.

2.1.3 Index register Y

Index register Y is a 16-bit register with the same function as index register X. As with 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. Also, when executing a block transfer instruction **MVP** or **MVN**, the content of index register Y indicates the low-order 16 bits of the destination data address. The second byte of the **MVP** or **MVN** is the high-order 8 bits of the destination data address.

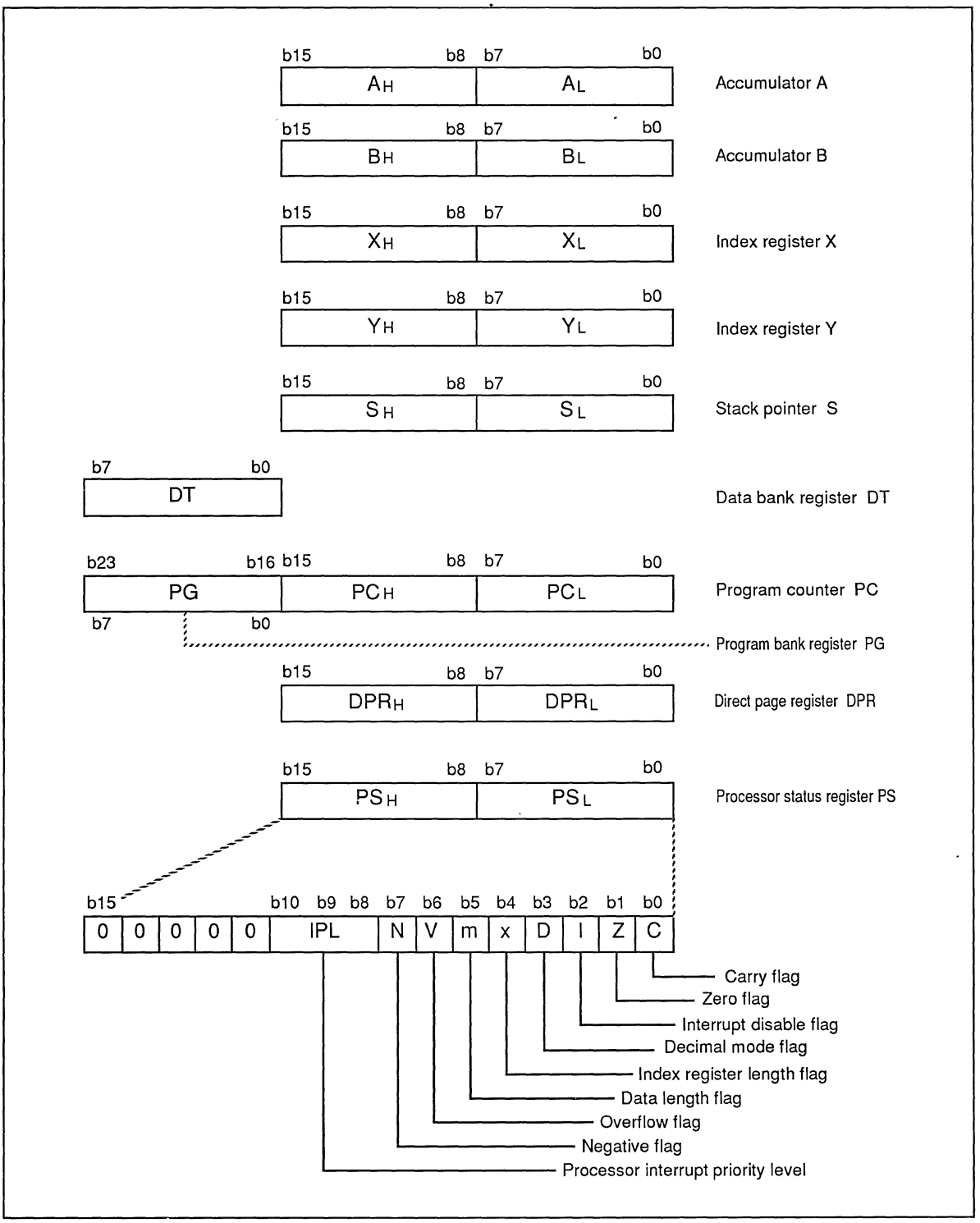


Fig. 2.1.1 CPU registers structure

2.1.4 Stack pointer (S)

Stack pointer S is a 16-bit register. It is used during a subroutine call or interrupt. It is also used during addressing modes using the stack. The contents of the stack pointer S indicates the address (stack area) for storing registers during subroutine calls and interrupts. The bank 0 must be designated for the stack area (reffer to section " 2.3 Addressable memory space"). Normally, the stack area is reserved in internal RAM.

When an interrupt is accepted, the contents of the program bank register PG is stored at the address indicated by the content of the stack pointer S, and the content of the stack pointer S is decremented by 1. Then the contents of the program counter PC and the processor status register PS are stored with the high-order bytes followed by the low-order bytes (PC_H, PC_L, PS_H, PS_L). The contents of the stack pointer S after accepting an interrupt is equal to the content before the interrupt decremented by 5. Figure 2.1.2 shows the stored registers when an interrupt is accepted.

When returning to the original routine after processing the interrupt, the registers stored in the stack area are restored to the original registers in the reverse sequence and the content of the stack pointer is returned to the status before the interrupt. The same operation is performed during a subroutine call, but the content of the processor status register PS is not stored (the content of the program bank register PG may not be stored either depending on the addressing mode).

The user is responsible for storing registers other than those described above during interrupts or subroutine calls. In addition, the stack pointer S must be initialized at the beginning of the program because its content is undefined at reset. Normally, the stack pointer is initialized with the highest address of the internal RAM. The contents of the stack area changes when subroutines are nested or when multiple interrupts are accepted. Therefore, make sure necessary data in the internal RAM are not destroyed when nesting subroutines.

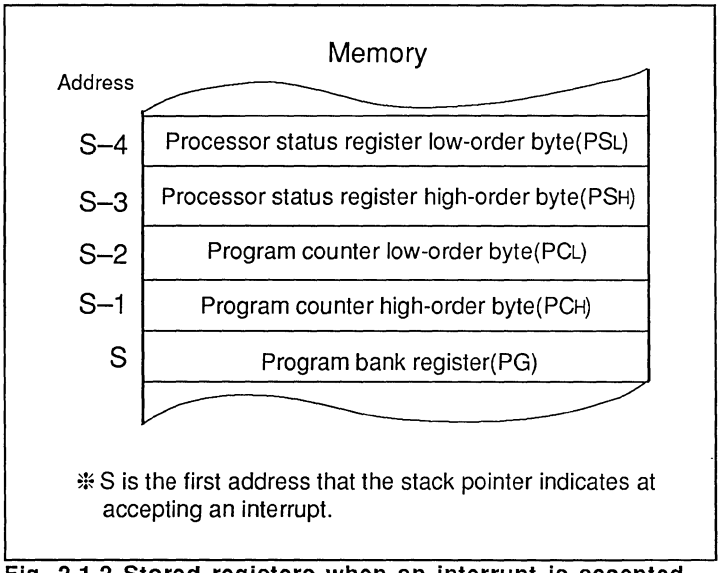


Fig. 2.1.2 Stored registers when an interrupt is accepted

2.1.5 Program counter (PC)

Program counter PC is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. The content of the high-order program counter(PC_H) becomes "FF_{16}", and the low-order program counter(PC_L) becomes "FE_{16}" at reset. The content of the program counter PC becomes the content of the reset vector address(address $FFFE_{16}$, $FFFF_{16}$) after removing reset state.
Figure 2.1.3 shows the program counter PC and the program bank register.

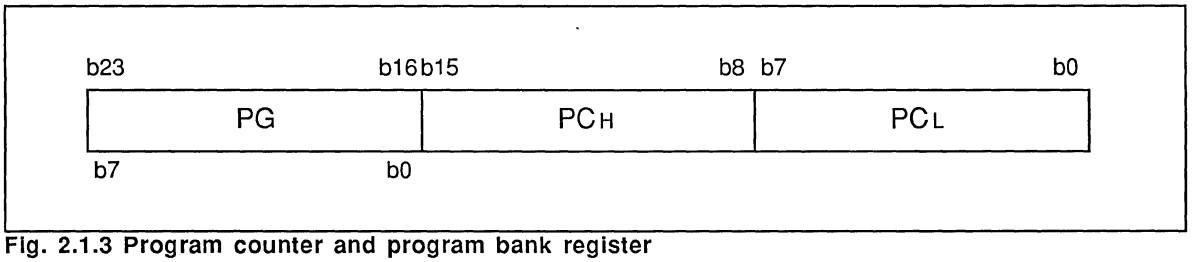


Fig. 2.1.3 Program counter and program bank register

2.1.6 Program bank register (PG)

Program bank register PG is an 8-bit register that indicates the high-order 8 bits (bank) of the next program memory address to be executed. When a carry occurs after incrementing the content of the program counter PC, the content of the program bank register PG is incremented by 1. Also, when a carry or borrow occurs after adding or subtracting the content of the program counter PC, the content of the program bank register PG is incremented or decremented by 1 so that programs can be written without considering bank boundaries, usually.
In single-chip mode, set the value "00_{16}" because only address between 0000_{16} and $FFFF_{16}$ can be accessed.
This register is cleared to "00_{16}" at reset.

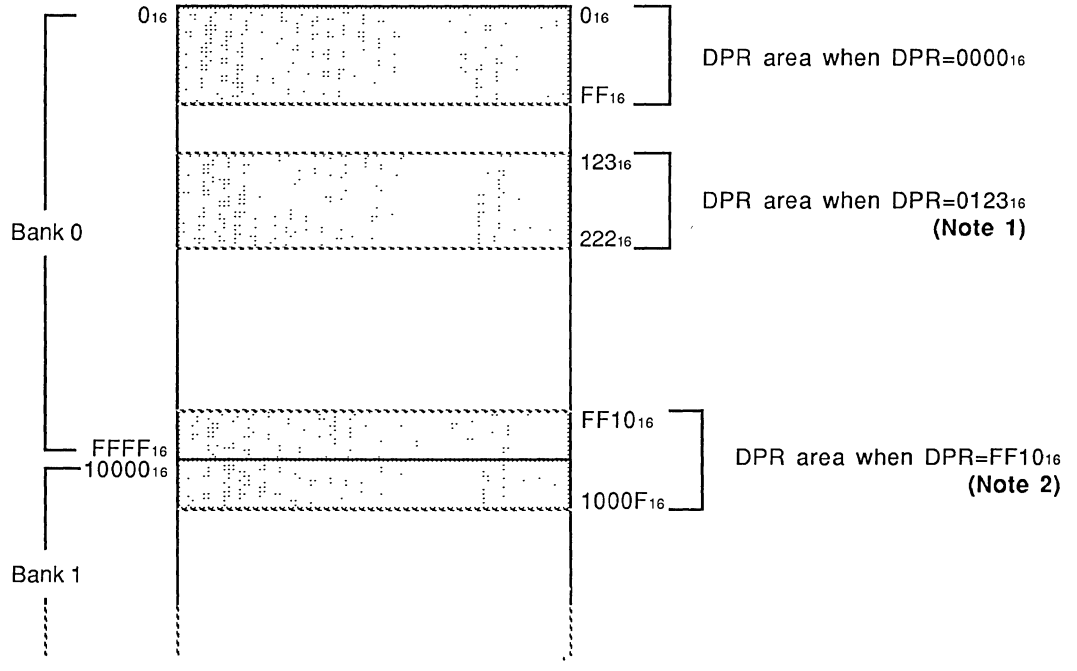
2.1.7 Data bank register (DT)

Data bank register DT is an 8-bit register. With some addressing modes using the data bank registerDT, the content of this register is used as the high-order 8 bits of a 24-bit address. In single-chip mode, set the value "00_{16}" because only address between 0000_{16} and $FFFF_{16}$ can be accessed.
Use the **LDT** instruction to set the value in this register. This register is cleared to "00_{16}" at reset.

※Refer to "MEPS 7700 SOFTWARE MANUAL" for the addressing modes.

2.1.8 Direct page register (DPR)

Direct page register DPR is a 16-bit register. The content of this register indicates whether the direct page area is allocated in bank 0 or spans across bank 0 and 1. This area can be accessed with two bytes by using the direct page addressing mode.
The content of the DPR is the base address (lowermost address) of the direct page area which extends 256 bytes above this address. The DPR can contain a value from 0000_{16} to $FFFF_{16}$. If it contains a value equal to or greater than "$FF01_{16}$", the direct page area spans across banks 0 and 1. If the low-order 8 bits of the DPR is "00_{16}", the number of cycles required to generate an address is minimized. Therefore, the low-order 8 bits of the DPR should normally be set to "00_{16}".
This register is cleared to "0000_{16}" at reset. Figure 2.1.4 shows the setting example of the direct page with the direct page register(DPR).



Note 1 : The number of execution cycles is incremented by 1 when the low-order 8 bits of the DPR are not "00_{16}".
Note 2 : The direct page spans across banks 0 and 1 when the DPR is "$FF01_{16}$" or greater.

Fig. 2.1.4 Setting example of direct page with direct page register (DPR)

2.1.9 Processor status register (PS)

Processor status register is an 11-bit register. It consists of flags to indicate the result of operation and CPU interrupt levels. The flags C, Z, V, and N are tested by branch instructions.
Figure 2.1.5 shows the register structure of the processor status register.
The details of the processor status register bits are described below.

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

Note : Bits 11 to 15 always are "0" when the contents of the processor status register are read.

Fig. 2.1.5 Processor status register structure

(1) Carry flag (C)
The carry flag is assigned to bit 0 of the processor status register. It contains the carry or borrow bit from the arithmetic and logic unit (ALU) after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set with the **SEC** or **SEP** instruction and cleared with the **CLC** or **CLP** instruction.

(2) Zero flag (Z)
The zero flag is assigned to bit 1 of the processor status register. It is set to "1" if the result of an arithmetic operation or data transfer is zero, and cleared to "0" if otherwise. This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction directly.
Note : The content of this flag has no meaning during decimal mode addition (**ADC** instruction).

(3) Interrupt disable flag (I)
The interrupt disable flag is assigned to bit 2 of the processor status register. It disables all maskable interrupts (interrupts other than watchdog timer, **BRK** instruction, and zero divide). Interrupts are disabled when this flag is "1". When an interrupt is accepted, it is set to "1" automatically to prevent multiple interrupts. This flag can be set with the **SEI** or **SEP** instruction and cleared with the **CLI** or **CLP** instruction. This flag is set to "1" at reset.

(4) Decimal mode flag (D)
The decimal mode flag is assigned to bit 3 of the processor status register. It determines whether addition and subtraction are performed in binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal (determined by the data length flag m). Decimal adjust is performed automatically. Decimal operation is possible only with the **ADC** and **SBC** instructions. This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction. This flag is cleared to "0" at reset.

(5) Index register length flag (x)
The index register length flag is assigned to bit 4 of the processor status register. It determines whether the index register X or index register Y is used as a 16-bit register or an 8-bit register. The register is used as a 16-bit register when flag x is "0" and as an 8-bit register when it is "1". This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction. This flag is cleared to "0" at reset.

(6) Data length flag (m)
The data length flag is assigned to bit 5 of the program status register. It determines whether to treat data as 16-bit or as 8-bit. A data is treated as 16-bit when flag m is "0" and as 8-bit when it is "1". This flag can be set with the **SEM** or **SEP** instruction and cleared with the **CLM** or **CLP** instruction. This flag is cleared to "0" at reset.

(7) Overflow flag (V)

The overflow flag is assigned to bit 6 of the processor status register. It is used when adding or subtracting a word as signed binary. In case the data length flag m is "0", the overflow flag is set to "1" when the result of addition or subtraction is outside the range between –32768 and +32767, and cleared to "0" in all other cases. In case the data length flag m is "1", the overflow flag is set to "1" when the result of addition or subtraction is outside the range between –128 and +127, and cleared to "0" in all other cases. The overflow flag can also be set and cleared directly with the **SEP**, **CLV**, and **CLP** instructions.

Note : This flag has no meaning in decimal mode.

(8) Negative flag (N)

The negative flag is assigned to bit 7 of the processor status register. It is set when the result of arithmetic operation or data transfer is negative (Data bit 15 is 1 when data length flag m is "0", or data bit 7 is 1 when data length flag m is "1".). It is cleared in all other cases. It can also be set with the **SEP** instruction and cleared with the **CLP** instruction.

Note : This flag has no meaning in decimal mode.

(9) Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) is assigned to bits 8, 9, and 10 of the processor status register. These three bits determine the priority level of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority level of the requested interrupt (set with the interrupt control register) is higher than the processor interrupt priority. When an interrupt is accepted, the IPL is stored in the stack and the processor interrupt priority is replaced by the interrupt priority of the accepted interrupt. This simplifies control of multiple interrupts.

There are no instructions to directly set or clear the IPL. It can be changed by placing the new IPL on the stack and updating the processor status register with the **PUL** or **PLP** instruction.

The content of the IPL is cleared to "000" at reset.

2.2 Internal bus interface

2.2.1 Internal bus interface overview

A bus interface unit (BIU) is provided between the CPU and the internal bus. Transfer of data between the CPU and memory or I/O device is always performed through the BIU. When the CPU reads data from memory or I/O device, it sends the address to be read to the BIU. The BIU reads the data from the specified address and the CPU receives the data from the BIU. Similarly, the CPU sends the address to be written to the BIU when writing data. Thus the BIU controls the transfer of data between the CPU and bus.

Figure 2.2.1 shows the block diagram of the bus interface unit.

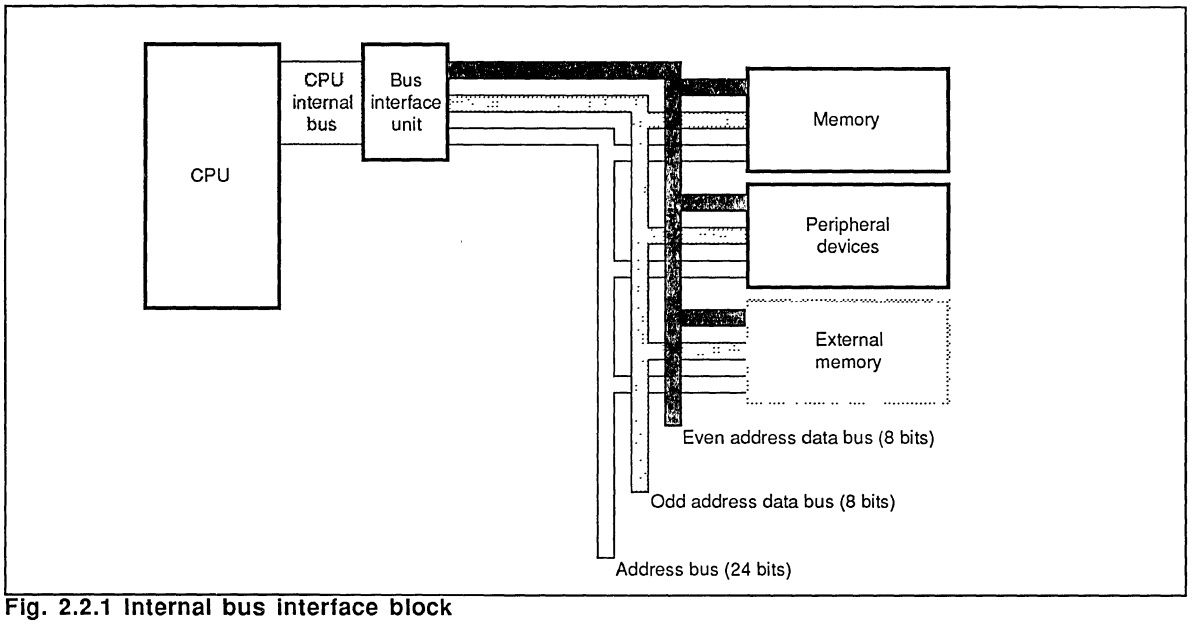


Fig. 2.2.1 Internal bus interface block

2.2.2 Bus interface unit functions

The M37702 group uses the clock ϕ (=f(XIN)/2) as the clock. The CPU also uses clock ϕ as the clock. However, since the CPU clock may be extended due to CPU wait under certain conditions, it is referred to as ϕ_{CPU} to distinguish it from clock ϕ.

The M37702 group internal bus (address bus and data bus) operates at timing $\overline{E}$ which is slower than clock ϕ. The operating clock of the CPU is different from the bus cycle because timing $\overline{E}$ is normally f(XIN)/4. Therefore, the BIU is provided between the CPU and bus to synchronize the transfer of data to and from memory and I/O device. The BIU enables the CPU to transfer data to and from memory through the bus without decreasing the instruction execution speed.

The BIU consists of four registers as shown in Figure 2.2.2. Table 2.2.1 shows the functions of each register and buffer.

Table 2.2.1 Functions of BIU registers and buffers

Name	Function
Program address register	Indicates the address of the program.
Instruction queue buffer	A three-byte buffer for temporarily holding instruction prefetched from memory.
Data address register	Indicates the address to be read from or to be written to memory or I/O.
Data buffer	A two-byte buffer for temporarily holding data read from memory or I/O device by the BIU or data written to memory or I/O device by the CPU.

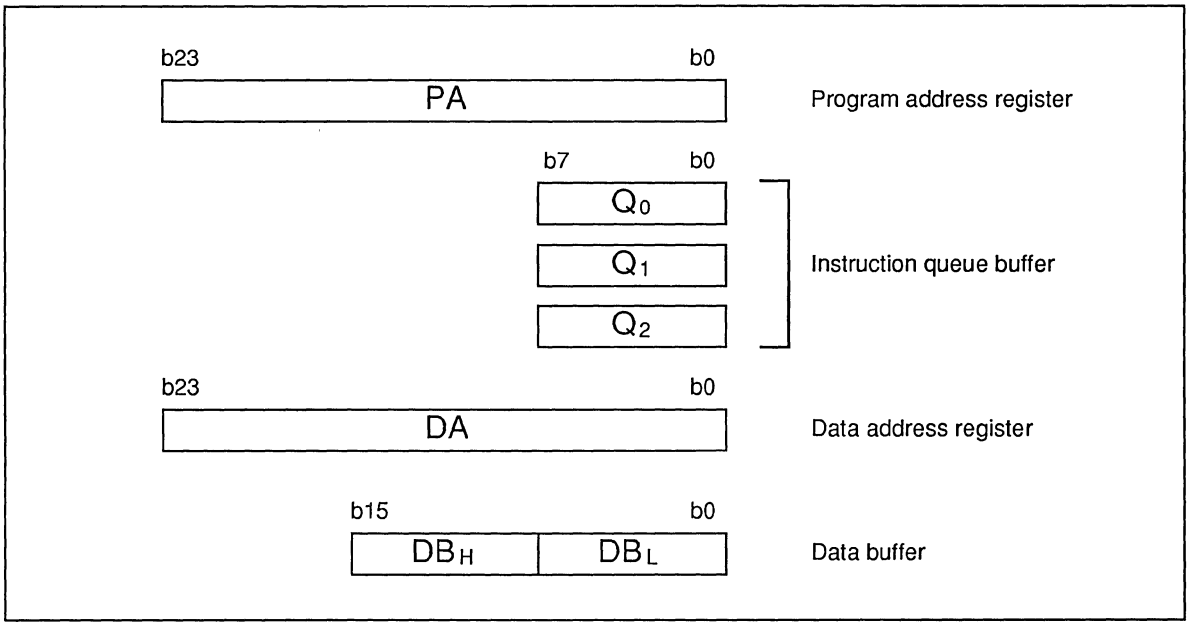


Fig. 2.2.2 Bus interface unit registers

The BIU performs the following operations.

1. **Prefetches an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer.**
   Normally, a program is executed sequentially in ascending order of address. Therefore, if the next instruction code is prefetched in the instruction queue buffer, the CPU can execute instructions simply by obtaining the instruction code from the instruction queue buffer. This will eliminate the time needed by the CPU to access the memory.
   When the CPU is not using the bus (for example when performing register to register operation), the BIU reads an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer. Data up to three bytes can be prefetched because the instruction queue buffer is three bytes long. Refer to Section "2.2.4 Data read/write operations" for more information concerning instruction code prefetch.

2. **Reads data at the specified address into the BIU when the CPU requests data in memory and transfers it to the CPU.**
   When executing instructions that processes data in memory or I/O device, the CPU must access the address assigned to the memory or I/O device and read the data. Because the operating clock of the CPU and bus are different, the CPU reads the data through the data buffer of the BIU.

3. **Writes the data obtained from the CPU to the specified address in memory.**
   When writing data to a specific address, the CPU sends the address and data to the BIU. And after that, the CPU continues to execute the next instruction extracting from the instruction queue buffer, because actual writing to memory or I/O device is performed by the BIU.

4. **Controls read of word data from odd number address and outputs the control signals required to access external memory in byte unit.**
   The transfer of data between the CPU and BIU is always performed through a 24-bit address bus and 16-bit data bus. This is also true between the BIU and internal memory or I/O device. The wait bit and BYTE pin (external bus width selection input pin) determine the data width only when an external memory is accessed.

2.2.3 Bus interface unit operations

Figure 2.2.3 shows the operating waveforms of the bus interface unit in memory expansion mode or microprocessor mode. The M37702 group BIU always operates at one of the waveforms shown in Figure 2.2.3.

The meaning of signals ALE and $\overline{E}$ in Figure 2.2.3 are as follows:

- ALE (Address Latch Enable)
  Signal used to latch only address signals from multiplexed signals containing data and address.

- $\overline{E}$
  Signal set to "L" level when the bus interface unit reads instruction code or data from memory or when it writes data to memory. Table 2.2.2 shows the bus status according to $\overline{E}$ and R/$\overline{W}$ signals.

Table 2.2.2 Bus status according to $\overline{E}$ and R/$\overline{W}$

$\overline{E}$	R/$\overline{W}$	Bus Status
H	H	Not used
H	L	Not used
L	H	Read
L	L	Write

(1) Basic operation

Waveform (a) is the bus interface operating waveform under the following conditions:

- When a one byte internal/external memory is accessed.
- When two bytes in internal memory are accessed together (starting on an even address).
- When two bytes in external memory are accessed together (starting on an even address when the BYTE pin is at "L" level).
- When the instruction code is obtained from memory into the instruction queue buffer.

Waveform (b) is the bus interface operating waveform when accessing in byte unit under the following conditions:

- When two bytes in internal/external memory are accessed together (starting on an odd address).
- When two bytes in external memory are accessed together with the BYTE pin at "H" level.

Waveforms (a) and (b) are the basic operating waveforms of the BIU. Waveform (a) or (b) is always used when accessing the internal memory. However, signals other than $\overline{E}$ cannot be observed in single-chip mode because the port P3 is used as a programmable I/O port.

(2) Effect of the wait bit

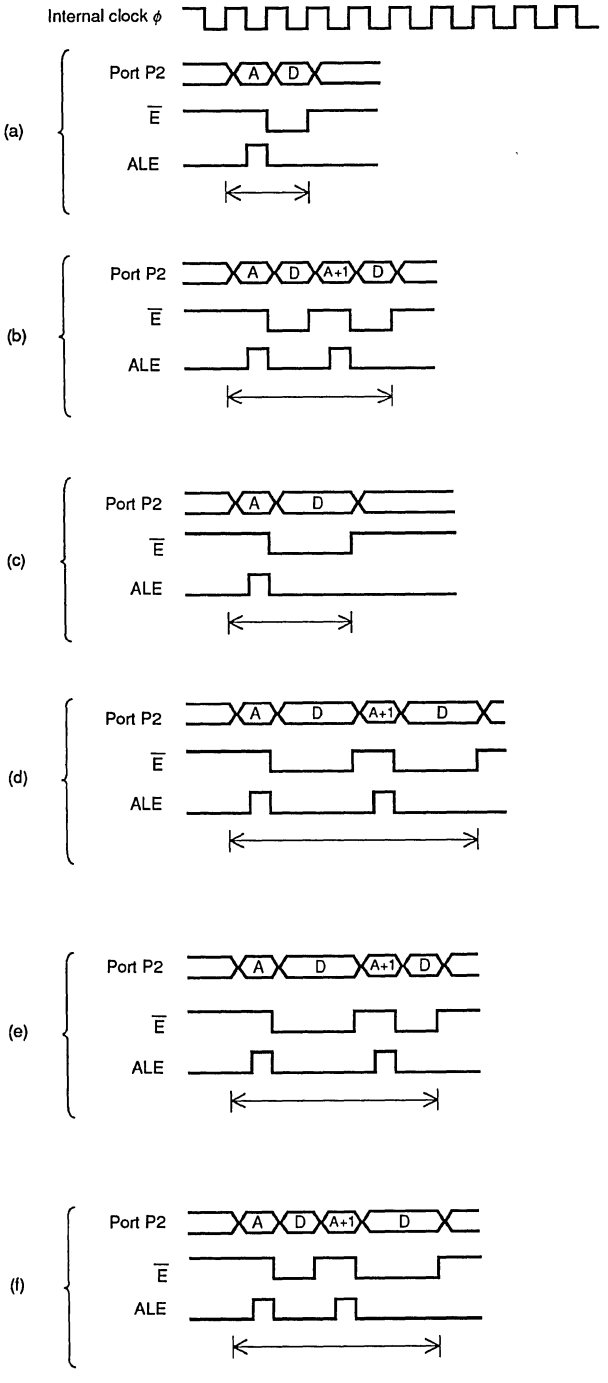
When accessing the external memory area, the BIU operating waveform changes according to the wait bit.

With the M37702 group, the external memory access time can be 1.5 times as long (the "L" level width of $\overline{E}$ signal becomes twice) by clearing the wait bit (bit 2) to "0" in the processor mode register (address $005E_{16}$). This enables external expansion of slow memories and peripheral LSIs.

Note : Internal memory access is not affected by the wait bit.

Figure 2.2.3 (c) to (f) show the effect of the wait bit on waveforms (a) and (b). Waveform (c) is the waveform when an external memory area is accessed under the conditions for waveform (a) with the wait bit cleared to "0".

Waveforms (d) to (f) are the waveforms when an external memory area is accessed under the conditions for waveform (b) with the wait bit cleared to "0". The entire waveform is affected by the wait bit for waveform (d) and the first half or the last half is affected respectively for waveforms (e) and (f).



- A: Address
- D: Data
- The waveform is for memory expansion mode or microprocessor mode.

Fig. 2.2.3 Bus interface device operating waveform

2.2.4 Data read/write operations

(1) Instruction code read

The CPU reads instructions codes from the instruction queue buffer of the BIU and executes them. The CPU notifies the BIU that an instruction code is needed during the instruction code fetch cycle. At this point, the operation depends on whether the instruction queue buffer contains an instruction code or not. If there is an instruction code in the instruction queue buffer, it is passed to the CPU. If there is no instruction code in the instruction queue buffer, or if the amount of data in the instruction queue buffer is less than the necessary instruction code, the BIU halts the CPU until a sufficient amount of instruction codes is stored in the instruction queue buffer.

Even when there is no request for instruction code from the CPU, if the instruction queue buffer is empty or if there is only one instruction code and the bus is available at the next cycle (the CPU does not use the bus at the next cycle), the BIU reads instruction codes from memory and stores them in the instruction queue buffer (instruction prefetch). During instruction prefetch, if the first address accessed when reading an instruction code from memory is even, then the data at the next odd number address is also read and stored in the instruction queue buffer. If the first accessed address is odd, only one byte is read and stored in the instruction queue buffer. However, if the instruction code is read from external memory with the BYTE pin at "H" level (external bus width 8-bit) in memory expansion or microprocessor mode, only one byte is read regardless of the accessed address.

Instruction code read is performed with operation (a) or (c) shown in Figure 2.2.3. When a branch or a jump or subroutine call instruction or an interrupt is executed, the content of the instruction queue buffer is cleared and a new instruction code is read from the new address.

(2) Data read/write

The CPU reads and writes data from/to the BIU data buffer. The CPU issues a request to BIU when it attempts to read or write data. At this point, if the BIU is using the bus or if there is a higher priority request, the CPU is made to wait until the BIU becomes ready. When the bus is available for data read or write, the BIU operates at one of the waveforms (a) to (f) shown in Figure 2.2.3.

- Data read

  When the CPU requests data from the BIU, it waits until the data becomes complete data in the data buffer. The BIU sends the address received from the CPU on the address bus, reads the content of memory when $\overline{E}$ signal is "L" level, and stores it in the data buffer.

- Data write

  The CPU sends address (address at which the data is written) and data to BIU.

  The address is written in the BIU data address register and the data is written in the data buffer. The actual writing in memory is performed by BIU and the CPU can proceed to the next step without waiting for the BIU to complete writing data in memory. The BIU sends the address received from the CPU to the address bus, sends the contents of the data buffer to the data bus, and writes it in memory when $\overline{E}$ signal is "L" level.

2.3 Addressable memory space

The M37702 group allocates all ROM, RAM, I/O, and various control registers in the same memory space. Therefore, data transfer and operation can be performed with the same instruction without distinguishing memory and I/O area.

The M37702 group program counter (PC) consists of 16 bits. It is used together with an 8-bit program bank register (PG) to directly access a 16M-byte address space from 0_{16} to $FFFFFF_{16}$.

2.3.1 Banks

The M37702 group address space is divided into 64K-byte blocks called banks. The M37702 group can access 256 banks from bank 0 to bank 255 (FF_{16}) in memory expansion or microprocessor mode.

The high-order 8 bits of the 24-bit address indicate the bank and the content of the program bank register (PG) or the data bank register (DT) indicates the bank to be used.

If the program counter overflows at a bank boundary, the content of the program bank register is incremented by 1. If a borrow occurs in the program counter register, the content of the program bank register is decremented by 1. Therefore, programs can be written without considering the bank boundaries, usually.

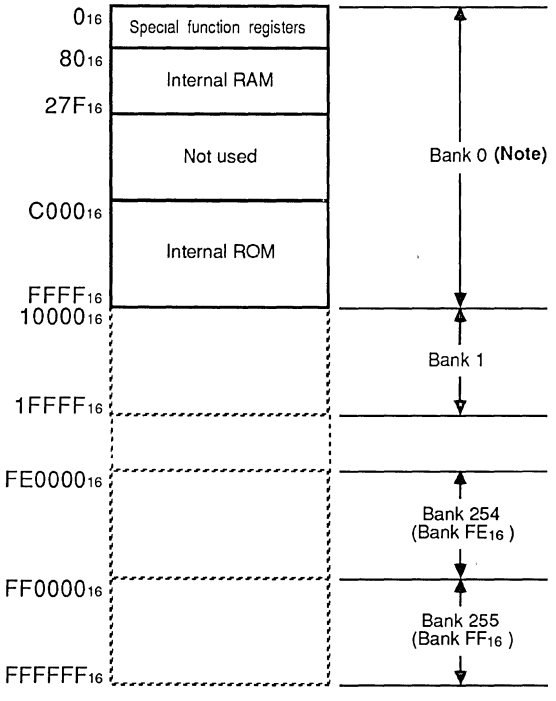
The banks can be accessed efficiently by using an addressing mode that uses the data bank register.

Bank 0 (address 0_{16} to $FFFF_{16}$) contains the internal ROM, internal RAM, and internal I/O control registers.

Note : In single-chip mode, only bank 0 can be accessed.

2.3.2 Direct page

By using the direct page register (DPR), bank 0 or a 256-byte space spanning across bank 0 and bank 1 can be accessed with fewer instruction cycles by using direct page addressing mode. This area is referred to as the direct page and is normally used for frequently accessed information. The direct page area can be specified by setting the lowermost address of the required area in the direct page register (see section "2.1.8 Direct page register").



Note : The memory allocation of bank 0 depends on the microprocessor type. This diagram shows the allocation for M37702M2-XXXFP. Refer to the M37702 group memory map in Appendix 1 for other types. Also note that only bank 0 can be accessed in single-chip mode.

Fig. 2.3.1 Addressable memory space

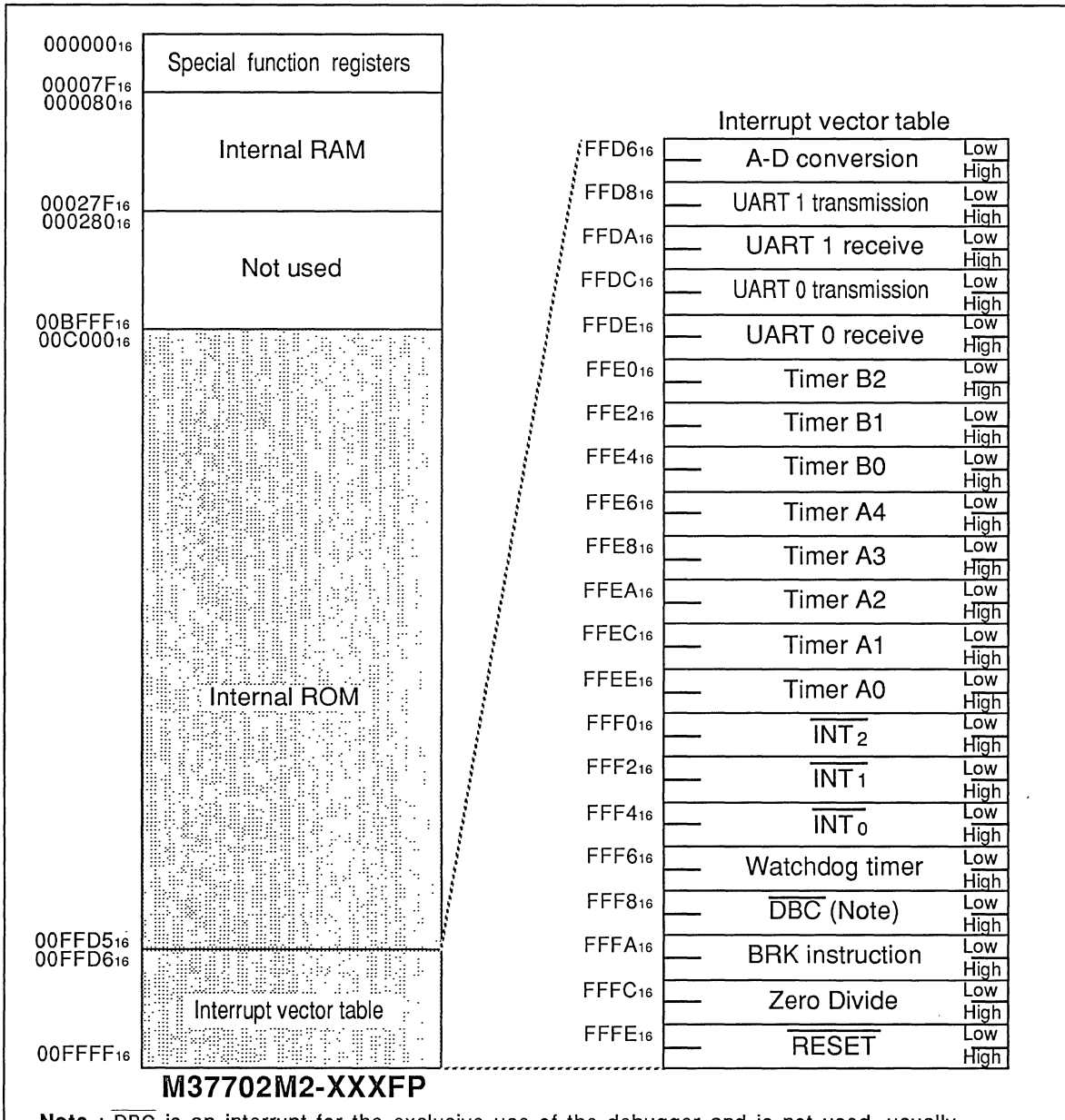
2.4 Memory allocation

Figure 2.4.1 shows the memory map in single-chip mode. The allocated memory and I/O are described below.

2.4.1 Internal memory and peripheral device memory allocation

(1) SFR (Special Function Register) area

Address 0000_{16} to $007F_{16}$ of bank 0 are the SFR (Special Function Register) area. This area contains the control registers of internal peripheral devices, I/O ports, timers, and so on. Internal peripheral devices can be accessed through these registers. Figure 2.4.2 shows the memory map of the SFR area.



Note : $\overline{DBC}$ is an interrupt for the exclusive use of the debugger and is not used, usually.

Fig. 2.4.1 Memory map (in single-chip mode)

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

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One-shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0 register
000047	
000048	Timer A1 register
000049	
00004A	Timer A2 register
00004B	
00004C	Timer A3 register
00004D	
00004E	Timer A4 register
00004F	
000050	Timer B0 register
000051	
000052	Timer B1 register
000053	
000054	Timer B2 register
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000018	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register
00005F	
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	
000063	
000064	
000065	
000066	
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	A-D conversion interrupt control register
000071	UART 0 transmission interrupt control register
000072	UART 0 receive interrupt control register
000073	UART 1 transmission interrupt control register
000074	UART 1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control register

Fig. 2.4.2 SFR area memory map

Each bit in the register can be either read only, write only, or read/write bit. Refer to each block description for the register function in the SFR area, and to section "3.1.2 Internal status at reset" for the status of the SFR at reset.

(2) RAM

The M37702M2-XXXFP, M37702M2AXXXFP and M37702M2BXXXFP have a 512-byte static RAM at address 0080_{16} to $027F_{16}$ in bank 0 (Note). In addition to storing data, the internal RAM area is used as stack area during subroutine calls and interrupts. Therefore, be careful of subroutine nesting levels and multiple interrupt levels so that important data is not destroyed.

Note 1 : Refer to "Appendix 1. M37702 group memory map" for other types.

(3) ROM

The M37702M2-XXXFP, M37702M2AXXXFP and M37702M2BXXXFP have a 16K-byte mask ROM at address $C000_{16}$ to $FFFF_{16}$ in bank 0 (Note). Address $FFD6_{16}$ to $FFFF_{16}$ are allocated to the interrupt vector table containing branch destinations (address of interrupt handling routines) when reset or interrupt occurs. This area must be allocated to ROM in microprocessor mode and external ROM version which prohibit internal ROM.

Note 2 : Refer to "Appendix 1. M37702 group memory map" for other types.

2.4.2 Processor modes

The M37702 group can operate in single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, memory allocation, and address space depend on the processor mode. The processor mode can be selected internally or externally as described below.

- Externally changing the processor mode

  The processor mode after reset start can be selected with the input level to the CNV_{SS} pin during reset start. Table 2.4.1 shows the relationship between the processor mode and the input level to the CNV_{SS} pin.

Table 2.4.1 Relationship between the processor mode and CNV_{SS} pin input level

CNV_{SS} Pin	Processor mode
V_{SS} level (0V)	Starts in single-chip mode after reset. One of the three modes can be selected by changing the processor mode bits.
V_{CC} level (5V)	Starts in microprocessor mode after reset. The other mode must not be selected by changing the processor mode bits.

- Internally changing the processor mode

  After reset start with the CNVss pin set to Vss level, the processor mode can be changed internally from program by changing the processor mode bits in the processor mode register (bits 1 and 0 at address $5E_{16}$). Figure 2.4.4 shows the structure of the processor mode register. In case of changing the processor mode internally, the actual function of each pin changes when the bus cycle $\overline{E}$ used to write to the processor mode register returns to "H" level.

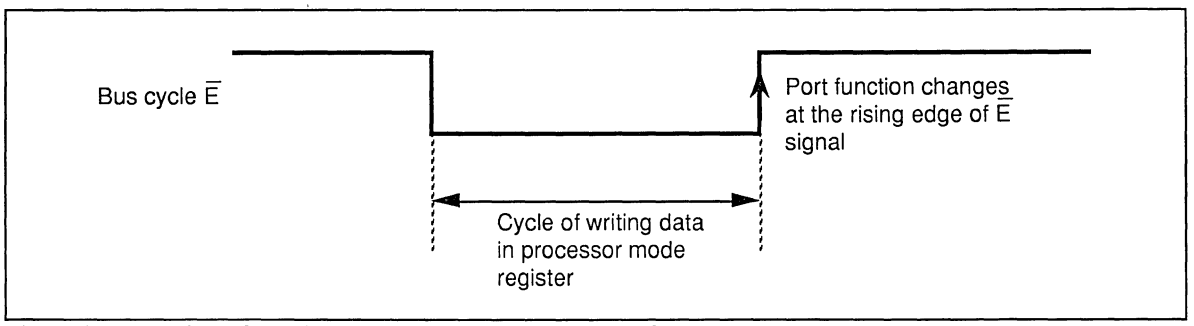
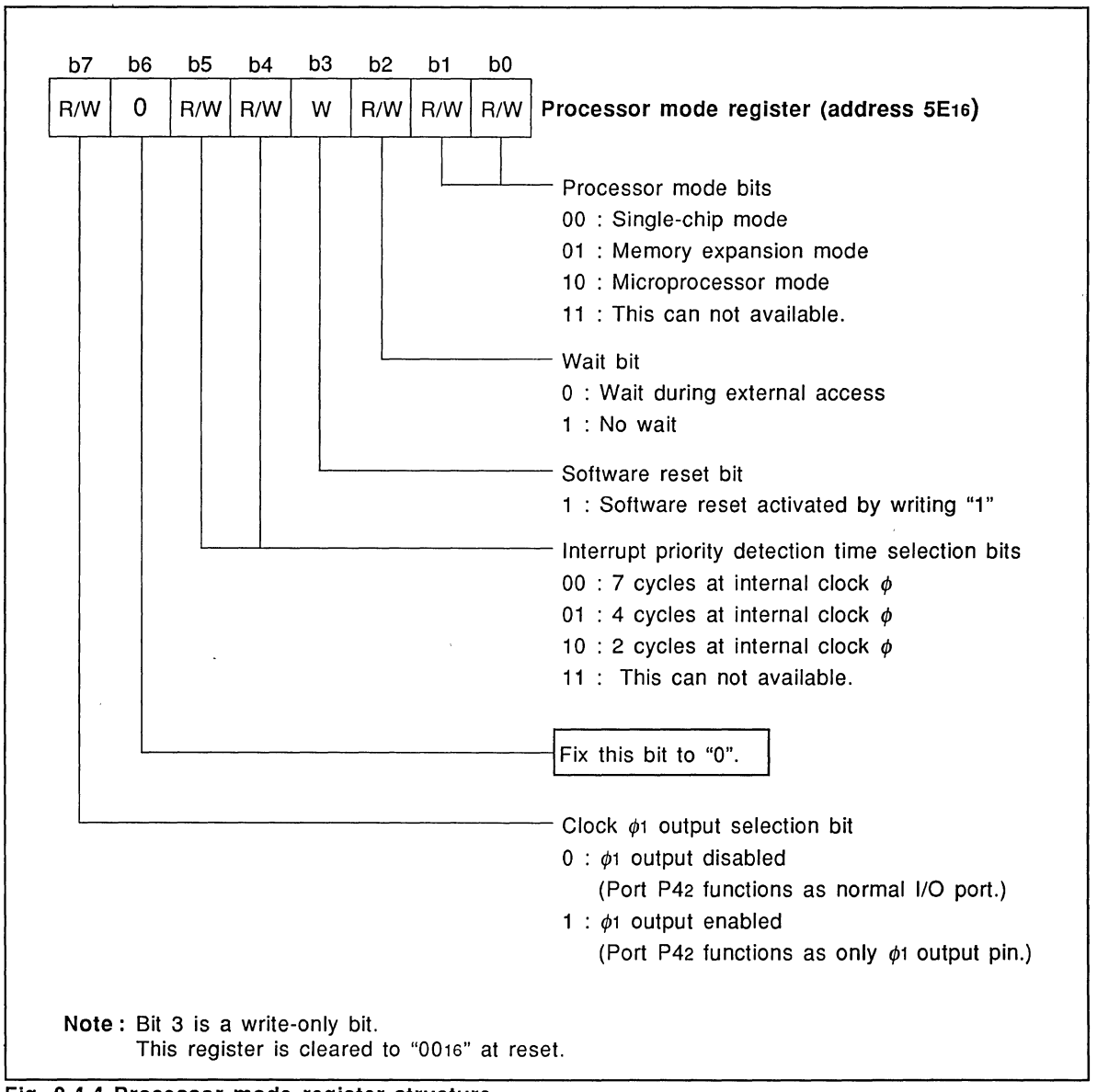


Fig. 2.4.3 Port function change timing due to change in processor mode

(1) Single-chip mode

This mode is selected when starting after reset with the CNV_{SS} pin set to Vss level. In this mode, the address bus and data bus are not output externally and all ports function as programmable I/O pins (internal peripheral device I/O pins when internal peripheral devices are used). Also note that in single-chip mode, a zero value must be stored in the data bank register and program bank register because only bank 0 can be accessed. Furthermore, in this mode, the wait bit, which is described later, is ignored and internal memory and I/O are always accessed at no wait.



Note : Bit 3 is a write-only bit.
This register is cleared to "0016" at reset.

Fig. 2.4.4 Processor mode register structure

(2) Memory expansion mode

This mode is selected when the processor mode bits are set to "01" after reset start with the CNV_{SS} pin set to V_{SS} level.

This mode is used when just using internal memory and I/O is not sufficient. In this mode, the memory and peripherals can be expanded to any area within a 16M-byte addressable memory space.

When the memory expansion mode is selected, ports P0 to P2 become the address bus and data bus and port P3 and part of P4 become the control signal I/O pins. In this case, the port register area associated with ports P0 to P3 and part of P4 become unusable and lose their normal I/O pin functions, but other memory and peripherals can be used. Refer to section "2.5 Input/Output pins" for more details concerning the functions of ports P0 to P4 when the memory expansion mode is selected.

If an area overlapping the internal memory area is read when the external memory is extended, only the data in the internal memory is read into the CPU and the data in the external memory is not read into the CPU. However, if data is written in this area, it is written both in the internal memory and external memory.

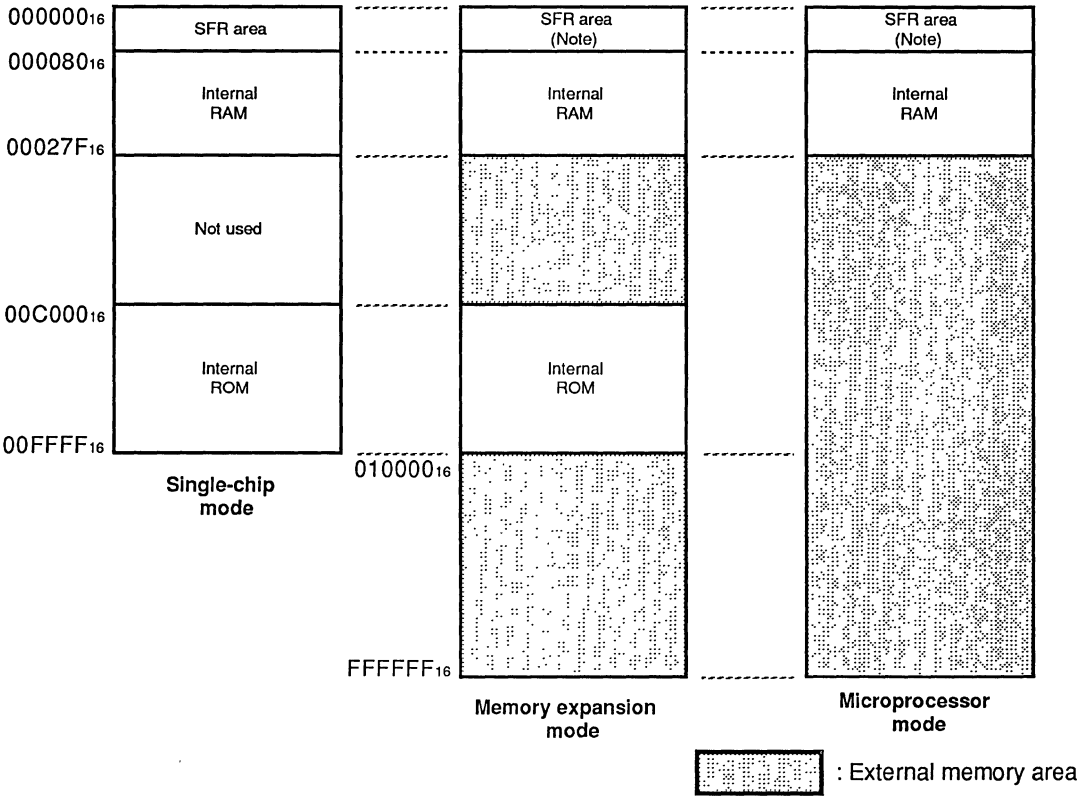
Furthermore, the accessing of external memory in this mode is affected by the level of the BYTE pin and wait bit described in the next section.

(3) Microprocessor mode

This mode is selected when the processor mode bits are set to "10" after reset start with the CNV_{SS} pin set to V_{SS} level. Also, this mode is selected when starting from reset with the CNV_{SS} pin set to V_{CC} level.

The function of this mode is the same as the memory expansion mode except that access to internal ROM is disabled and port $P4_2$ always outputs the clock ϕ_1 regardless of the content of the clock ϕ_1 output selection bit. This mode is suitable for small volume production or prototype models before full scale production because external ROM can be installed easily.

Figure 2.4.5 shows the memory map in each processor mode. Refer to section "2.5 Input/Output pins" for the change in port functions.



Note : Address 2_{16} to 9_{16} can also be used as external memory area in memory expansion mode and microprocessor mode.

Fig. 2.4.5 Memory map in each processor mode (for M37702M2-XXXFP)

2.4.3 External memory area bus control

The BYTE pin and the wait bit are provided to simplify access to external memory area in memory expansion mode and microprocessor mode. The BYTE pin and the wait bit are valid only when accessing external memory area and have no effect when accessing internal memory or internal peripherals. Therefore, the BYTE pin and the wait bit are ignored in single-chip mode.

(1) BYTE pin (external bus width selection pin)

When accessing the external memory in memory expansion mode or microprocessor mode, the input level to the BYTE pin is used to select whether 8-bit data bus or 16-bit data bus (refer to section "2.5.4.(2) Data bus").

The external bus width becomes 8-bit when the BYTE pin is at "H" level. In this case, data read/write to the external area is always performed in 8-bit (1-byte) unit and the port P2 pins become the data (D_0 to D_7) I/O pins. The use of 8-bit peripheral ICs is simplified by setting the bus width for external area to 8-bit.

The external bus width becomes 16-bit when the BYTE pin is at "L" level. In this case, data read/write to the external area is always performed in 16-bit (1 word) unit and the port P2 pins become the data I/O pins for the low-order byte (even address data: D_0 to D_7) of a 16-bit data and the port P1 pins become the data I/O pins for the high-order byte (odd address data: D_8 to D_{15}) of a 16-bit data.

The data width is always 16-bit when accessing the internal memory area regardless of the BYTE pin level.

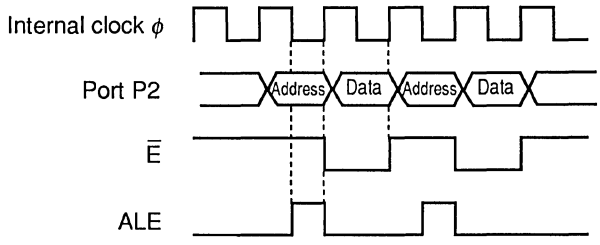
(2) Wait bit

The wait bit (bit 2 at address $5E_{16}$) is used to attach slower memory when expanding external memory or I/O in memory expansion mode or microprocessor mode. When the wait bit is "0", a wait for external area access is enabled (one-wait mode) and bus operation is performed at the speed of 2/3 of the bus cycle (bus cycle=$f(X_{IN})/4$) during no wait. When the wait bit is "1", bus operation becomes no wait mode and bus cycle is $f(X_{IN})/4$.

The wait bit is cleared to "0" at reset and the system starts in one-wait mode. Internal memory access is always performed at no wait because this bit is ignored.

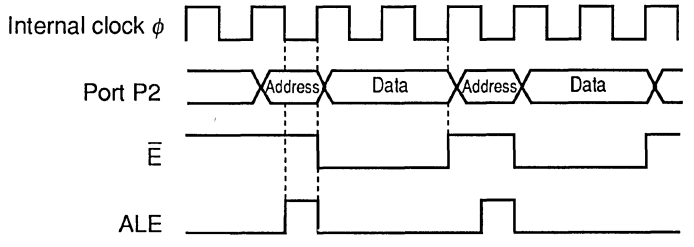
Figure 2.4.6 shows the effect of the wait bit for external access.

(a) Waveform when the external memory area is accessed with the wait bit set to "1".



※ This waveform is always used when accessing the internal memory.

(b) Waveform when the external memory area is accessed with the wait bit cleared to "0".



※ Waveform (a) is used for internal memory area access even when the wait bit is "0".

Fig. 2.4.6 Effect of wait bit for external access

2.5 Input/Output pins

The M37702 group has 68 programmable I/O pins (ports P0 to P8). These ports also function as I/O ports for internal peripheral devices. There are pins that function depend on the processor mode or not.

2.5.1 Programmable I/O ports

Each of the programmable I/O ports (ports P0 to P8) has a direction register and a port register. The direction register is used to select the input/output mode by one bit. The direction registers and the data registers are allocated in the SFR area of bank 0. Input level is read from the pin selecting input mode by reading the port register. The data written to the port register is output from the pin selecting output mode. Figure 2.5.1 shows the memory allocation of the direction registers and the port registers.

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

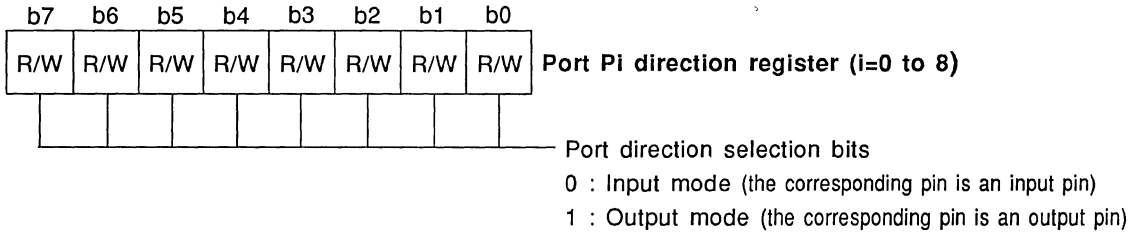
Fig. 2.5.1 Memory allocation of direction registers and port registers

(1) Direction register

Each bit of the direction register corresponds to a pin. Figure 2.5.2 shows the structure of the direction register. The port is set to input mode (input pin) when the corresponding bit is "0", and to output mode (output pin) when the corresponding bit is "1".

The direction registers are cleared to "00_{16}" at reset. Therefore, I/O ports are set to input mode.

If I/O port are not used as output pins, set the corresponding direction register bit to "0" for input mode.



Bit	b7	b6	b5	b4	b3	b2	b1	b0
Corresponding pin	Pi7	Pi6	Pi5	Pi4	Pi3	Pi2	Pi1	Pi0

Note: Nothing is allocated in the high-order 4 bits of port P3 direction register. The high-order 4 bits of port P3 direction register is fixed to "0" at reading.
These registers are cleared to "00_{16}" at reset.

Fig. 2.5.2 Relationship between the port direction register and pins

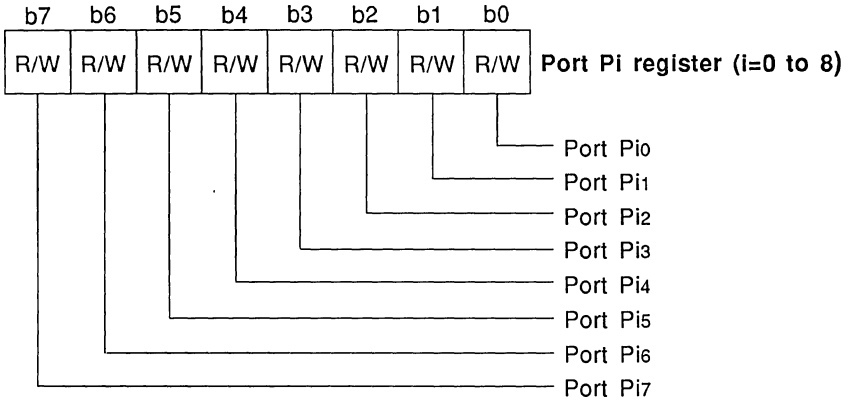
(2) Port register

The port register is used to transfer data with external devices through the I/O ports. Figure 2.5.3 shows the relationship between the port register and the pins.

To output data from a port set to output mode, the data must be written to the corresponding bits of the port register. This data is written in the port latch and is output from the port set to output mode.

If a port programmed for output is read, the status of the output pin is not read but the content of the port latch is read. Therefore, the previously output value can be read correctly even if the output "H" voltage drops or "L" voltage rises due to external load.

A pin programmed for input is floated and the value input to the pin can be read by reading the corresponding bit of the port register. If a value is written to a pin programmed for input, it is written in the port latch and the pin remains floating.



Data I/O is performed by reading the corresponding bit to the pin or writing to the bit.

Note: Nothing is allocated in the high-order 4 bits of port P3 direction register. The high-order 4 bits of port P3 direction register is fixed to "0" at reading.

Fig. 2.5.3 Relationship between the port register and pins

2.5.2 Pin functions

Figure 2.5.4 shows the port peripheral circuits. The functions of some pins depend on the processor mode while others are not affected. This section describes those pins that are not affected by the processor mode. The next section describes the pin functions according to the processor mode.

(1) Effect of processor mode on pin functions

The function of some pins depends on the processor mode. Table 2.5.1 shows the pin functions according to processor mode. Table 2.5.2 shows the pin functions of ports P0 to P4 according to processor mode. The function of port P1 also depends on the input level of the BYTE pin (external bus width selection pin). The details of the following pins are described in the next section.

Table 2.5.1 Pin functions according to processor mode

Mode / Pin	Single-chip mode	Memory expansion mode and microprocessor mode	
		External 16-bit bus (BYTE="L")	External 8-bit bus (BYTE="H")
Port P0	Programmable I/O port	Address bus (A_0 to A_7)	
Port P1	Programmable I/O port	Address bus (A_8 to A_{15}) /Data bus (D_8 to D_{15})	Address bus (A_8 to A_{15})
Port P2	Programmable I/O port	Address bus (A_{16} to A_{23})/Data bus (D_0 to D_7)	
Port P3	Programmable I/O port	$P3_0$....$\overline{R/W}$ output pin $P3_1$....$\overline{BHE}$ output pin $P3_2$....$\overline{ALE}$ output pin $P3_3$....$\overline{HLDA}$ output pin	
Port P4	Programmable I/O port **Note** : $P4_2$ pin can be programmed for ϕ_1 output pin.	$P4_0$....$\overline{HOLD}$ input pin $P4_1$....$\overline{RDY}$ input pin $P4_3$ to $P4_7$....Same as single-chip mode $P4_2$....Programmable I/O port/clock ϕ_1 output pin (in memory expansion mode) Clock ϕ_1 output pin (in microprocessor mode)	
BYTE	Ignored	External bus width selection pin	

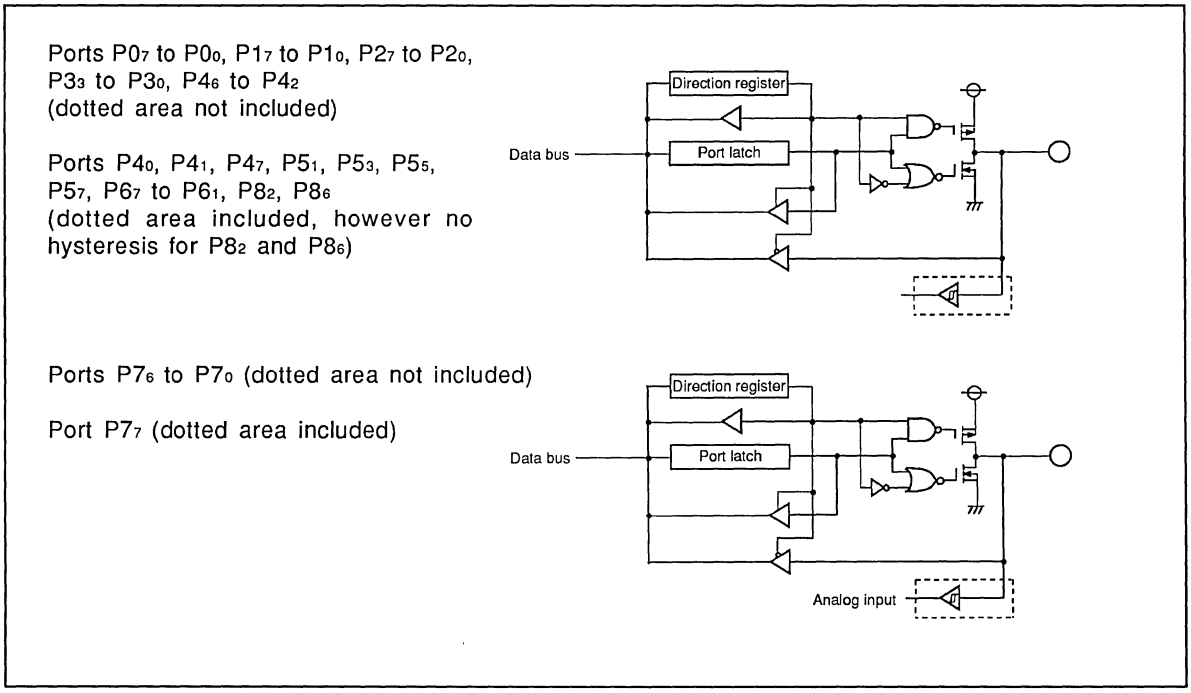
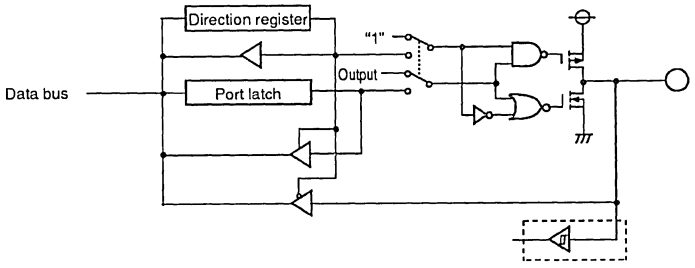


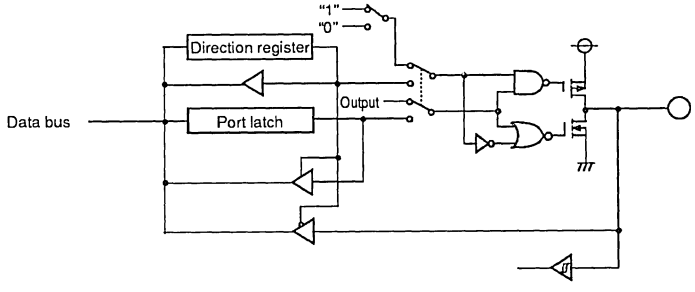
Fig. 2.5.4 Port peripheral circuits (1)

Ports $P8_3$, $P8_7$
(dotted area not included)

Ports $P5_0$, $P5_2$, $P5_4$, $P5_6$, $P6_0$
(dotted area included)



Ports $P8_0$, $P8_1$, $P8_4$, $P8_5$



$\overline{E}$ output pin

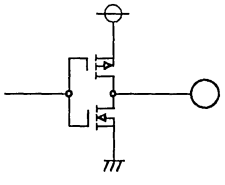


Fig. 2.5.4 Port peripheral circuit (2)

Table 2.5.2 Functions of ports P0 to P4 according to processor mode

Port \ Processor mode	Processor mode register b1 b0 = 0 0 (Single-chip mode)	b1 b0 = 0 1 (Memory expansion mode)	b1 b0 = 1 0 (Microprocessor mode)
Port P0	$\overline{E}$ / P0₇ to P0₀: I/O port	$\overline{E}$ / P0₇ to P0₀: A₇ to A₀	
Port P1 — BYTE="L"	$\overline{E}$ / P1₇ to P1₀: I/O port	$\overline{E}$ / P1₇ to P1₀: A₁₅ to A₈ / DATA(odd)	
Port P1 — BYTE="H"		$\overline{E}$ / P1₇ to P1₀: A₁₅ to A₈	
Port P2 — BYTE="L"	$\overline{E}$ / P2₇ to P2₀: I/O port	$\overline{E}$ / P2₇ to P2₀: A₂₃ to A₁₆ / DATA(even)	
Port P2 — BYTE="H"		$\overline{E}$ / P2₇ to P2₀: A₂₃ to A₁₆ / DATA(odd,even)	
Port P3	$\overline{E}$ / P3₃ to P3₀: I/O port	$\overline{E}$ / P3₃: $\overline{HLDA}$ / P3₂: ALE / P3₁: $\overline{BHE}$ / P3₀: R/$\overline{W}$	
Port 4 — Processor mode register bit 7 = "0"	$\overline{E}$ / P4₇ to P4₀: I/O port	$\overline{E}$ / P4₇ to P4₂: I/O port / P4₁: $\overline{RDY}$ / P4₀: $\overline{HOLD}$	
Port 4 — Processor mode register bit 7 = "1"	P4₂: ϕ_1 clock **Note :** Same as above except for P4₂.	P4₂: ϕ_1 clock **Note :** Same as above except for P4₂. But port P4₂ always outputs the clock ϕ_1 in microprocessor mode regardless of the processor mode register bit 7.	

(2) Functions of pins unaffected by processor mode

Table 2.5.3 shows the functions of pins unaffected by processor mode. The functions of these pins are the same in all modes.

Table 2.5.3 Functions of pins unaffected by processor mode

Pin	Function
Port P5	8-bit programmable I/O pin. (Also used as timer I/O pin.)
Port P6	8-bit programmable I/O pin. (Also used as timer I/O and external interrupt input pin.)
Port P7	8-bit programmable I/O pin. (Also used as analog input pin.)
Port P8	8-bit programmable I/O pin. (Also used as serial I/O pin.)
V_{CC}, V_{SS}	Supply voltage pins. 5V±10% is applied to V_{CC} and V_{SS} is connected to GND.
CNVss	This pin controls the processor mode. The processor mode is selected by changing the input voltage level to this pin (except change after reset start). Refer to section "2.4.2 Processor modes" for detail information concerning the processor mode. In single-chip mode, this pin must be set to the same level as V_{SS}.
AVcc, AVss	A-D conversion circuit supply voltage pins. Connect AVcc to Vcc and AVss to Vss.
V_{REF}	Reference voltage input pin for the A-D converter. Analog input voltage from Vss level to the level of this pin can be converted. Apply any voltage up to Vcc level to this pin.
X_{IN}, X_{OUT}	Clock I/O pin for the internal oscillator circuit. The M37702 group is equipped with an internal clock generator and the oscillating frequency is set by connecting a ceramic resonator or quartz crystal oscillator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open. The maximum clock input frequency is 8MHz for M37702M2-XXXFP, 16MHz for M37702M2AXXXFP, and 25MHz for M37702M2BXXXFP.
$\overline{RESET}$	Reset input pin. Set this pin to "L" level to enter the reset state. Then when this pin is returned to "H" level, the reset state is removed and program loading starts from the address set in the reset vector. Refer to "Chapter 3. Reset" for the contents of registers immediately after returning from reset state.
$\overline{E}$	Internal bus cycle $\overline{E}$ is output.

Ports P5 to P8 have the programmable I/O port function as well as special functions such as I/O pins for external interrupt, timer, A-D converter, and serial I/O. When these multiple function ports are used as special function output pins, they are automatically set to output mode, but when they are used as special function input pins, the port direction register must be set to input mode. The methods for selecting special functions are described under each function.

All ports function as programmable I/O port immediately after returning from reset state.

2.5.3 Single-chip mode pin functions

In single-chip mode, 68 ports can be used as programmable I/O pins (using multiple function pins as I/O ports).

Figure 2.5.5 shows the pin connection diagram in single-chip mode.

Table 2.5.4 shows the pin functions depending on processor mode (ports P0 to P4, BYTE pin) in single-chip mode. Refer to section "2.5.1 Programmable I/O ports" for the programmable I/O port functions.

Refer to table 2.5.3 for the functions of other pins.

Table 2.5.4 Functions of ports P0 to P4 and BYTE pin in single-chip mode

Pin	Functions	Pin	Functions
Port P0	8-bit programmable I/O port	Port P3	4-bit programmable I/O port
Port P1	8-bit programmable I/O port	Port P4	8-bit programmable I/O port (Note)
Port P2	8-bit programmable I/O port	BYTE	Ignored in single-chip mode

Note : Port $P4_2$ also functions as the clock ϕ_1 output pin by program.

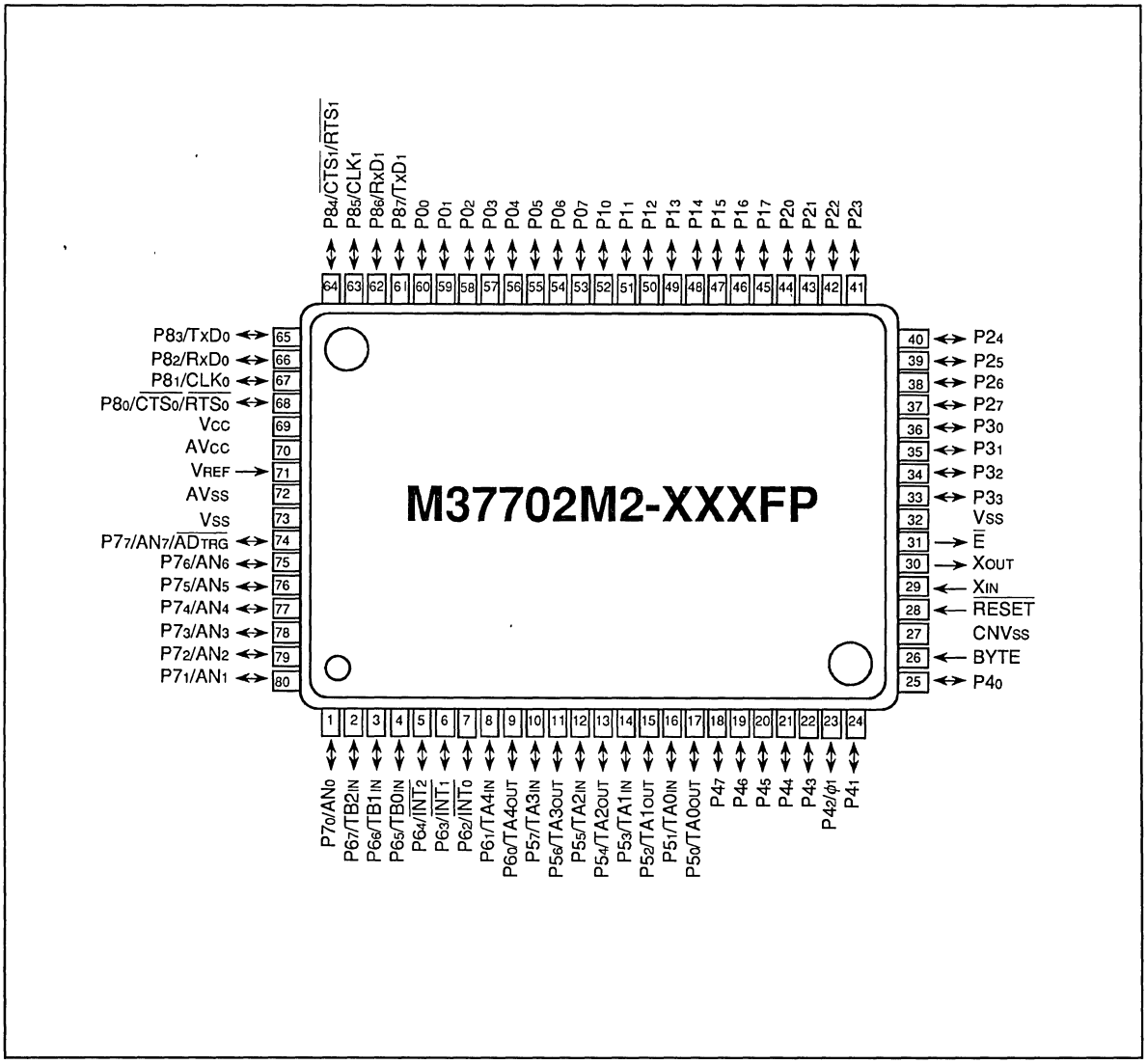


Fig. 2.5.5 Single-chip mode pin connection diagram

Port $P4_2$ can be programmed for the clock ϕ_1 output pin by setting the processor mode register. The clock ϕ_1 output starts at the rising edge of bus cycle $\overline{E}$ that was changed to "L" level to write "1" to the clock ϕ_1 output selection bit in the processor mode register (bit 7 of address $5E_{16}$).

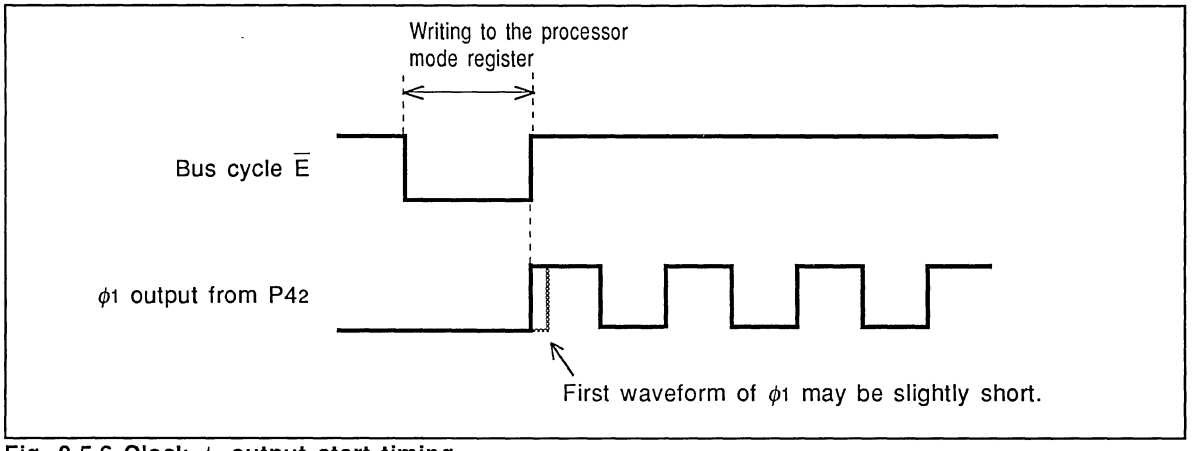


Fig. 2.5.6 Clock ϕ_1 output start timing

2.5.4 Memory expansion and microprocessor mode pin functions

The only difference between the memory expansion mode and microprocessor mode is the port $P4_2$ function.

- In memory expansion mode.....$P4_2$ function is selected by the clock ϕ_1 output selection bit.
- In microprocessor mode.....$P4_2$ always functions as the clock ϕ_1 output pin (the clock ϕ_1 output pin selection bit is ignored).

The function of each pin except for port $P4_2$ is identical in memory expansion mode and microprocessor mode.
In memory expansion mode, there are 38 I/O ports (ports $P4_2$ to $P4_7$ and P5 to P8). In microprocessor mode, there are 37 I/O ports (ports $P4_3$ to $P4_7$, and P5 to P8). The internal address bus and data bus can be used externally in microprocessor mode.
Figure 2.5.7 shows the pin connection diagram in memory expansion and microprocessor mode.
Table 2.5.5 shows the pin functions depending on processor mode (ports P0 to P4 and BYTE pin) in memory expansion mode and microprocessor mode.
Refer to Table 2.5.3 for the functions of other pins.

Table 2.5.5 Pin functions in memory expansion and microprocessor mode

Pin	Functions	Pin	Functions
Port P0	Address bus	$P4_0$	$\overline{HOLD}$ signal input pin (Note 2)
Port P1	Address bus/data bus (Note1)	$P4_1$	$\overline{RDY}$ signal input pin (Note 2)
Port P2	Address bus/data bus	$P4_2$	Clock ϕ_1 output pin (Note 3)
Port P3	External memory control signal output	BYTE	External bus width selection signal input

Note 1 : This may be address bus only depending on the input level of the BYTE pin.
Note 2 : In memory expansion mode and microprocessor mode, the direction register bits of ports $P4_0$ and $P4_1$ must be set to input mode.
Note 3 : In memory expansion mode, port $P4_2$ functions as the clock ϕ_1 output pin by program.

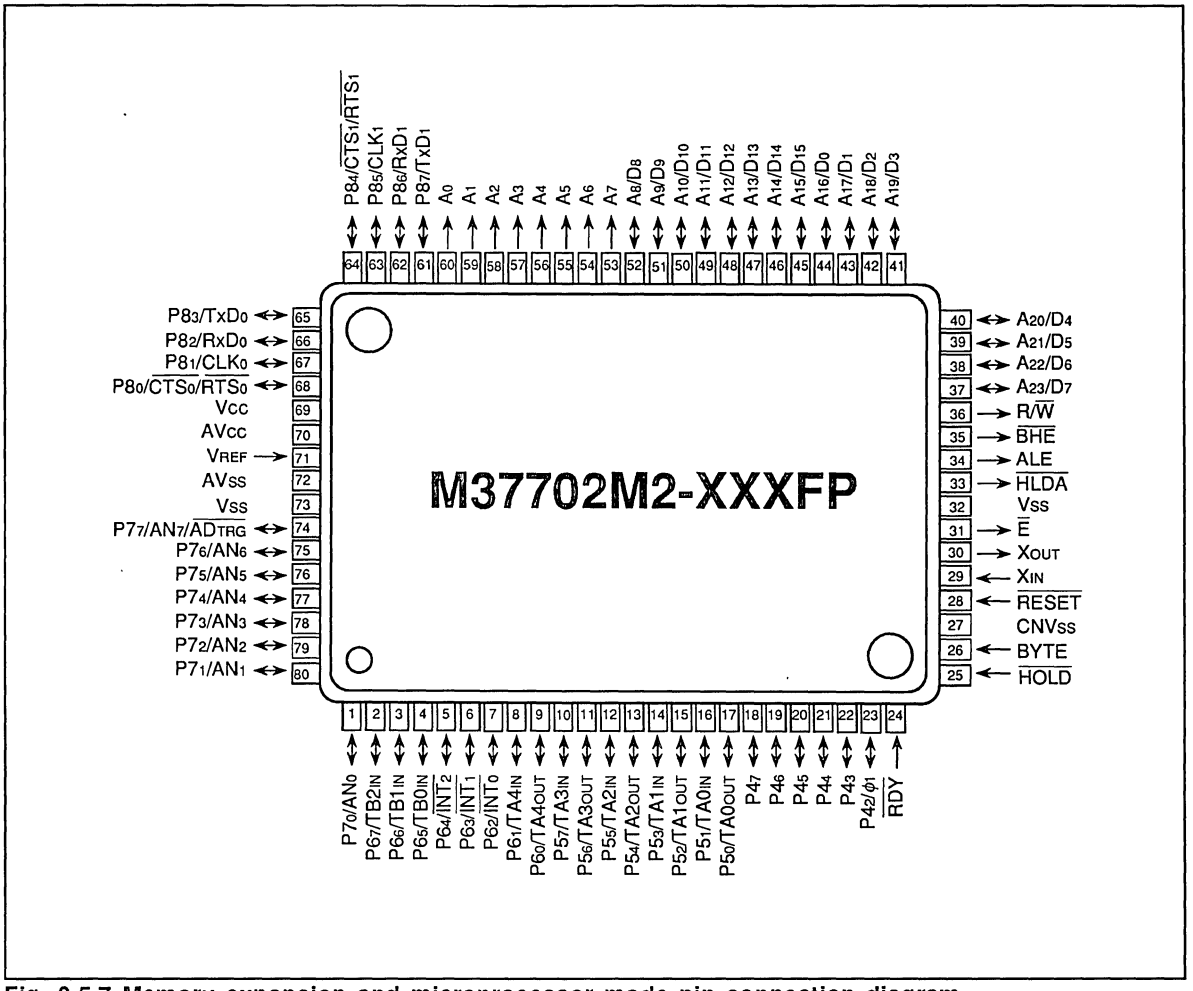


Fig. 2.5.7 Memory expansion and microprocessor mode pin connection diagram

The functions of each pin in memory expansion mode and microprocessor mode are described below.

(1) Address bus (ports P0, P1, and P2)

Ports P0, P1, and P2 become address signal output pins and lose their programmable I/O port functions.

The M37702 group allows direct access to 16M-byte memory space from address 000000_{16} to $FFFFFF_{16}$. Therefore, 24 address signals are output externally in memory expansion mode and microprocessor mode which allow memory and I/O to be expanded externally.

Port P1 and P2 also function as data I/O pins as the same time.

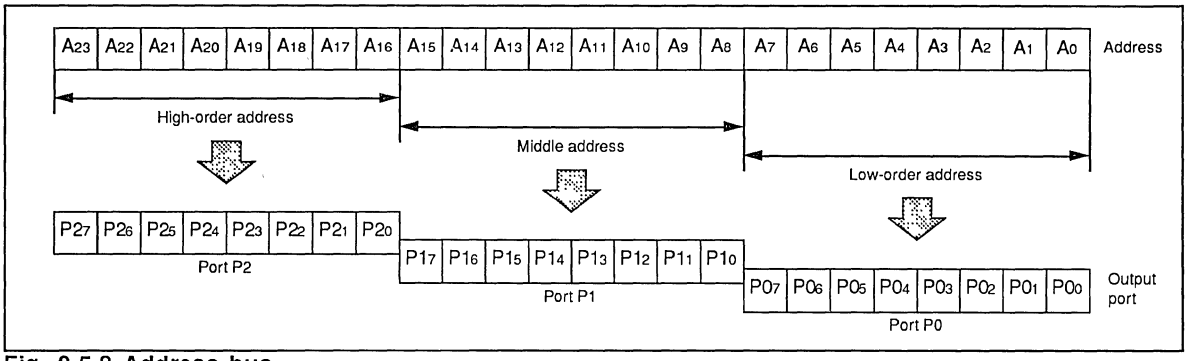


Fig. 2.5.8 Address bus

(2) Data bus

In addition to address signal (high-order and middle address bus) output function, ports P1 and P2 also function as data I/O pins. The level of the BYTE pin can be used to select between 8-bit or 16-bit data bus width.

● When the BYTE pin is at "L" level (16-bit external bus width)

When the BYTE pin is at "L" level, the external bus width is 16 bits and even address data and odd address data are output simultaneously. Ports P1 and P2 are used as address bus and data bus, and multiplexed (address signal and data signal) signals are output from these ports.

Port P1 performs time division multiplexing of address (A_{15} to A_8) output and data input/output in odd address (high-order byte of 16-bit data). Middle address is output while $\overline{E}$ signal is at "H" level, and data input/output in odd address is performed while $\overline{E}$ signal is at "L" level.

Similarly, port P2 performs time division multiplexing of address (A_{23} to A_{16}) output and data input/output in even address (low-order byte of 16-bit data). High-order address is output while $\overline{E}$ signal is at "H" level, and data input/output in even address is performed while $\overline{E}$ signal is at "L" level.

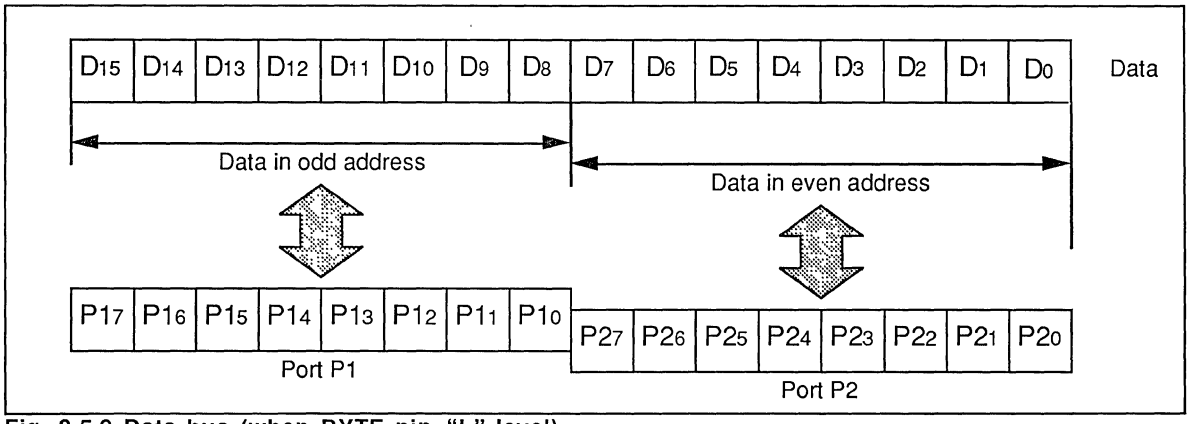


Fig. 2.5.9 Data bus (when BYTE pin="L" level)

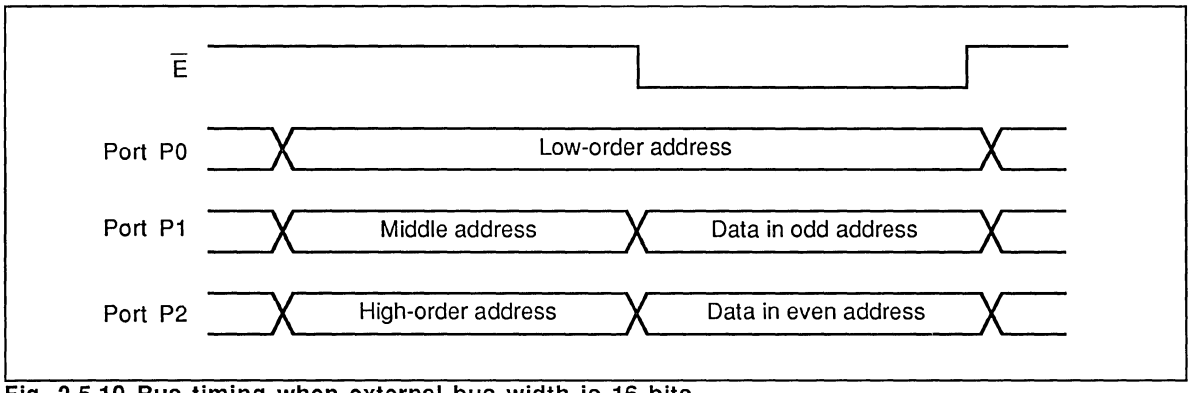


Fig. 2.5.10 Bus timing when external bus width is 16 bits

- **When the BYTE pin is at "H" level (8-bit external bus width)**
  When the BYTE pin is at "H" level, the external bus width becomes 8 bits, and port P2 performs time division multiplexing of address (A_{23} to A_{16}) output and data I/O.
  Address is output while $\overline{E}$ signal is at "H" level, and 8-bit data is input/output when $\overline{E}$ signal is at "L" level.

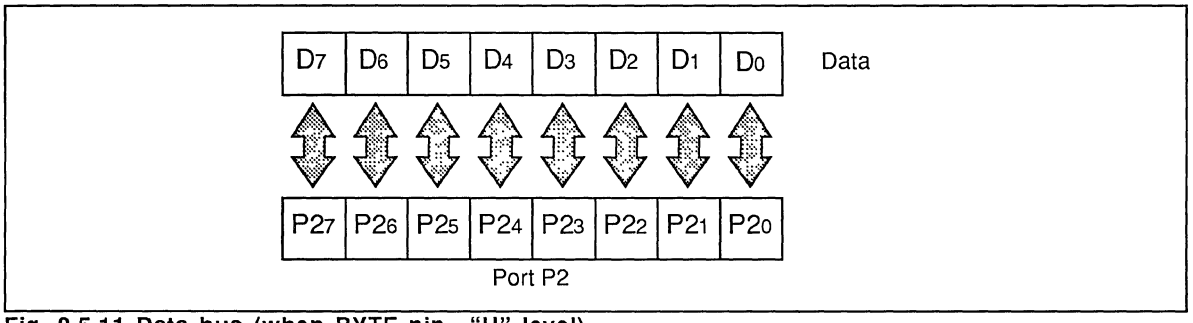


Fig. 2.5.11 Data bus (when BYTE pin= "H" level)

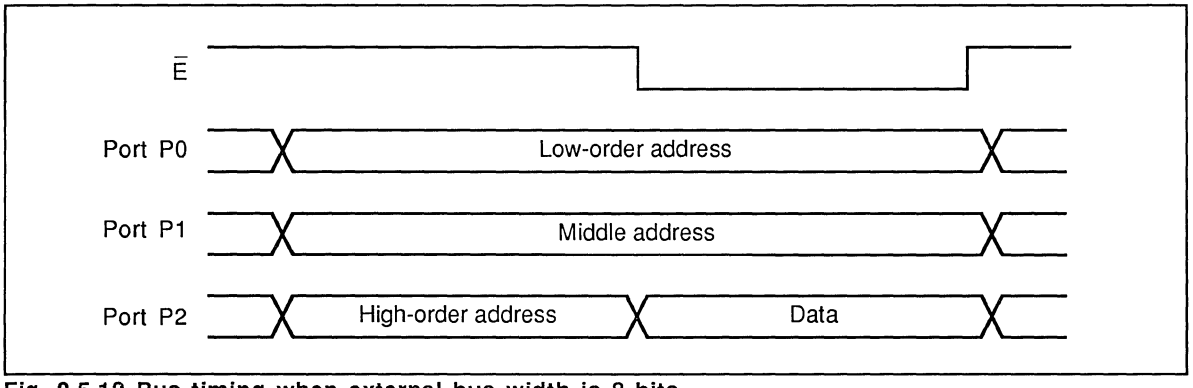


Fig. 2.5.12 Bus timing when external bus width is 8 bits

(3) Port $P4_2$

Port $P4_2$ has a different function between memory expansion and microprocessor mode.
In memory expantion mode, port $P4_2$ functions as the clock ϕ_1 output pin by setting the clock ϕ_1 output selection bit in the processor mode register to "1", and as programmable I/O port by setting the clock ϕ_1 output selection bit to "0". In microprocessor mode, port $P4_2$ always functions as the clock ϕ_1 output pin regardless of the clock ϕ_1 output selection bit.

(4) R/$\overline{W}$ output pin

A read/write signal indicating the data bus direction is output. The data bus is read when the level of this pin is at "H", and data is written to data bus when it is at "L". This signal is used for external memory input/output requests.

(5) $\overline{BHE}$ output pin

A byte high enable signal is output. This pin is at "L" level when an odd number address is accessed. This signal is used to expand the 8-bit memory and I/O when the external bus is used at 16-bit width. Refer to section "Chapter 7. Application" for memory and I/O expantion method.

(6) ALE signal output pin

This signal is used to obtain only address signal from the multiplexed signals of ports P1 and P2. A latch is opened externally when the ALE signal is at "H" level to obtain the address data and the latched content is held while the ALE signal is at "L" level.

(7) $\overline{HOLD}$ input pin

This pin is used to input the hold request signal. The microcomputer becomes Hold state while this pin is at "L" level.
Refer to section "2.12 Hold function" for details.

(8) $\overline{HLDA}$ signal output pin

This pin is used to externally output the hold acknowledge signal. The hold acknowledge signal indicates that "L" level is input to the $\overline{HOLD}$ pin and the microcomputer is in Hold state. An "L" level is output from this pin while the microcomputer is in Hold state.

(9) $\overline{RDY}$ signal input pin

This pin is used to input the ready signal. The bus cycle $\overline{E}$ can be stopped (Ready state) when "L" level is input to this pin. The port and bus status at inputing "L" level to the $\overline{RDY}$ pin is maintained while in Ready state. The $\overline{RDY}$ signal is used when slow memory is externally connected.
Refer to section "2.13 Ready function" for details.

(10) $\overline{E}$ output pin

This pin is used to output the enable signal. Data input/output is performed when the output of this pin is at "L" level. This signal controls the time division multiplexing of address information and data.

(11) BYTE pin

This pin is used to input the byte enable signal. The input level to this pin determines whether the external memory is used with 16-bit data width or 8-bit. When the BYTE pin input level is at "L", the data width is 16 bits, and ports P1 and P2 become the data I/O pins (data bus). When the BYTE pin input level is at "H", the data width is 8 bits and port P2 becomes the data I/O pin (data bus). However, the data width is always 16 bits regardless of the BYTE pin level when accessing an internal memory.

(12) CNV_{SS} pin

This pin controls the microprocessor operating mode. Memory expansion or microprocessor mode is selected by changing the processor mode bit in the processor mode register after reset start with setting this pin to the same level as the V_{SS} pin.
The microprocessor mode can also be selected by reset start with setting this pin to the same level as the V_{CC} pin. This pin must be set to Vcc level for external ROM version models such as the M37702S1FP (refer to section "2.4.2 Processor modes").

2.6 Interrupts

The suspension of the current operation in order to perform another operation due to a certain event is referred to as an "interrupt". Interrupt is used when there is a request to execute a higher priority routine or when an operation must be performed at a certain timing.

2.6.1 Interrupt functions

The M37702 group has 19 different sources of interrupts. When an interrupt is generated, a branch is made to the address (branch address) corresponding to the source. The branch address must be stored in the interrupt vector table. The interrupt vector table is allocated at address $FFD6_{16}$ to $FFFF_{16}$ in bank 0. When writing programs, branch must be made to the address in the interrupt vector table corresponding to each interrupt (interrupt vector address). The branch address is the start address of the interrupt handling routines (interrupt service routine). Figure 2.6.1 shows the interrupt mechanism.

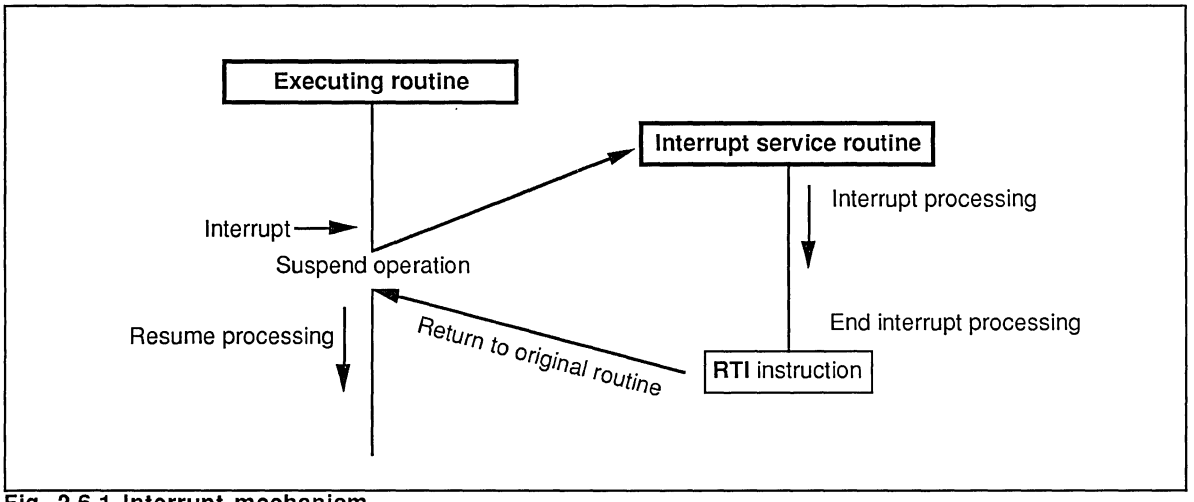


Fig. 2.6.1 Interrupt mechanism

When an interrupt is accepted, the contents of the following registers before the interrupt are stored to the stack area in the order ①→②→③.

① Program bank register (PG)

② Program counter (PC_L, PC_H)

③ Processor status register (PS_L, PS_H)

The procedure for storing these registers depends on whether the content of the stack pointer S is even or odd. When the content of the stack pointer S is even, the content of the program counter PC and processor status register PS are stored in 16-bit unit. When the content of the stack pointer S is odd, they are stored in 8-bit unit. Figure 2.6.2 shows the status of the stack area when an interrupt is accepted.

When interrupt processing completes, the control must be returned to the original routine to resume processing. Therefore, the **RTI** instruction is used to return to the original routine and the above registers stored to the stack area are restored in the order of ③→②→① to resume operation.
Only the above registers ① to ③ are stored automatically when an interrupt is accepted. The user is responsible for storing other necessary registers.

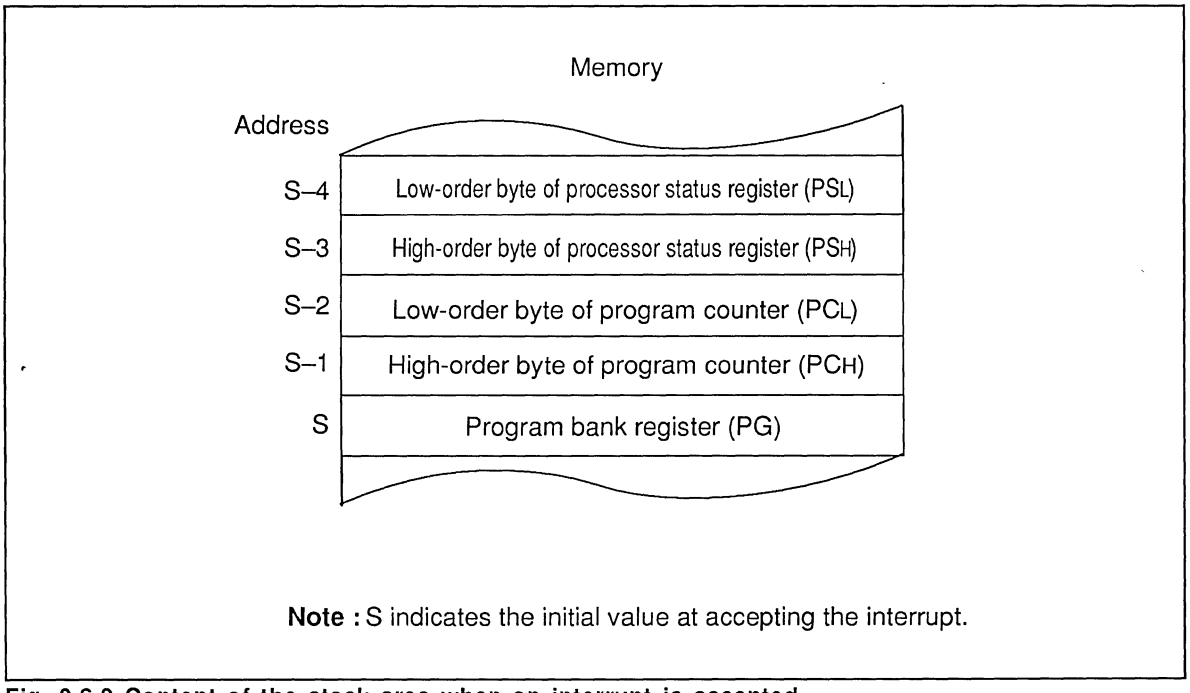


Fig. 2.6.2 Content of the stack area when an interrupt is accepted

2.6.2 Sources of interrupts

Table 2.6.1 shows the sources of interrupts and the corresponding vector address. Store the start address of the interrupt service routine at the vector address shown in this table.

Table 2.6.1 Interrupt sources and vector address

Interrupt source	Vector address		Remarks
	High-order address	Low-order address	
Reset (Note 1)	$00FFFF_{16}$	$00FFFE_{16}$	Non-maskable
Zero divide	$00FFFD_{16}$	$00FFFC_{16}$	Non-maskable software interrupt
BRK instruction	$00FFFB_{16}$	$00FFFA_{16}$	Non-maskable software interrupt
$\overline{DBC}$ (Note 2)	$00FFF9_{16}$	$00FFF8_{16}$	Not used normally
Watchdog timer	$00FFF7_{16}$	$00FFF6_{16}$	Non-maskable interrupt
$\overline{INT_0}$	$00FFF5_{16}$	$00FFF4_{16}$	External interrupt due to $\overline{INT_0}$ pin input signal
$\overline{INT_1}$	$00FFF3_{16}$	$00FFF2_{16}$	External interrupt due to $\overline{INT_1}$ pin input signal
$\overline{INT_2}$	$00FFF1_{16}$	$00FFF0_{16}$	External interrupt due to $\overline{INT_2}$ pin input signal
Timer A0	$00FFEF_{16}$	$00FFEE_{16}$	Timer A0 internal interrupt
Timer A1	$00FFED_{16}$	$00FFEC_{16}$	Timer A1 internal interrupt
Timer A2	$00FFEB_{16}$	$00FFEA_{16}$	Timer A2 internal interrupt
Timer A3	$00FFE9_{16}$	$00FFE8_{16}$	Timer A3 internal interrupt
Timer A4	$00FFE7_{16}$	$00FFE6_{16}$	Timer A4 internal interrupt
Timer B0	$00FFE5_{16}$	$00FFE4_{16}$	Timer B0 internal interrupt
Timer B1	$00FFE3_{16}$	$00FFE2_{16}$	Timer B1 internal interrupt
Timer B2	$00FFE1_{16}$	$00FFE0_{16}$	Timer B2 internal interrupt
UART0 receive	$00FFDF_{16}$	$00FFDE_{16}$	Valid only when the UART0 function is selected.
UART0 transmission	$00FFDD_{16}$	$00FFDC_{16}$	
UART1 receive	$00FFDB_{16}$	$00FFDA_{16}$	Valid only when the UART1 function is selected.
UART1 transmission	$00FFD9_{16}$	$00FFD8_{16}$	
A-D conversion	$00FFD7_{16}$	$00FFD6_{16}$	Internal interrupt that occurs when A-D conversion completes.

Note 1 : Reset is included in this table.
Note 2 : The $\overline{DBC}$ interrupt is a debug control interrupt and is not normally used.

Each interrupt source is described below.

(1) Internal interrupt

Table 2.6.2 shows the sources of internal interrupt.

Table 2.6.2 Internal interrupt sources

Interrupt	Interrupt source
Zero divide	Occurs when 0 is specified as the divisor for a **DIV** instruction. (See "MELPS 7700 Software Manual")
BRK instruction	Occurs when a **BRK** instruction is executed. (See "MELPS 7700 Software Manual")
Watchdog timer	Occurs when the topmost bit of the 12-bit watchdog timer becomes "0". (See section "2.12 Watchdog timer")
Timer Ai	Occurs when timer Ai (i=0 to 4) underflows or overflows. (See section "2.7 Timer A")
Timer Bi	Occurs when timer Bi (i=0 to 2) underflows or overflows. (See section "2.8 Timer B")
UARTi receive	Occurs during UARTi (i=0,1) receive. (See section "2.9 Serial I/O")
UARTi transmission	Occurs during UARTi (i=0,1) transmit. (See section "2.9 Serial I/O")
A-D conversion	Occurs when A-D conversion completes. (See section "2.11 A-D Converter")

(2) External interrupt ($\overline{INT_0}$ to $\overline{INT_2}$ interrupts)

There are three external interrupts ($\overline{INT_0}$ to $\overline{INT_2}$). These interrupts are generated by input level or input edge to pins $\overline{INT_0}$ to $\overline{INT_2}$. The interrupt sources can be selected by using bits 4 and 5 of the $\overline{INT_i}$ (i=0 to 2) interrupt control register shown in Figure 2.6.4. Table 2.6.3 shows the sources of $\overline{INT_i}$ interrupts. Pins $\overline{INT_0}$ to $\overline{INT_2}$ are shared with ports $P6_2$ to $P6_4$. Therefore, the corresponding bit in the port P6 direction register must be cleared to "0" in order to use these pins as external interrupt input pins. If the $\overline{INT_i}$ interrupts are not used, the $\overline{INT_i}$ interrupt priority (see next section) should be set to 0 because the $\overline{INT_i}$ interrupts always monitor the status of ports $P6_2$ to $P6_4$ to raise interrupt requests.

The input signal to the $\overline{INT_i}$ pins must have pulse width greater than 250ns at "H" level or "L" level regardless of the source oscillating frequency f(X_{IN}).

Table 2.6.3 $\overline{INT_i}$ Interrupt sources

b5	b4	Interrupt source
0	0	Falling edge of the signal input to the $\overline{INT_i}$ pin
0	1	Rising edge of the signal input to the $\overline{INT_i}$ pin
1	0	When the $\overline{INT_i}$ pin level becomes "H"
1	1	When the $\overline{INT_i}$ pin level becomes "L"

2.6.3 Interrupt control

The enabling and disabling of interrupts are controlled by the interrupt request bit, interrupt priority level, processor interrupt priority (IPL), and interrupt disable flag (I) (excluding some software interrupts). The interrupt disable flag and the processor interrupt priority level are assigned to the processor status register (PS). The interrupt request bit and the interrupt priority level are assigned to the interrupt control register of the respective interrupt. Figure 2.6.3 shows the memory map of the interrupt control register and Figure 2.6.4 shows the structure. However, there is no interrupt control register for non-maskable interrupts such as zero divide interrupt, **BRK** instruction interrupt, and watchdog timer interrupt.

- Non-maskable interrupt: An interrupt that causes branch to the interrupt service routine regardless of the interrupt control flags.
- Maskable interrupt: An interrupt that can be disabled with the interrupt control flags.

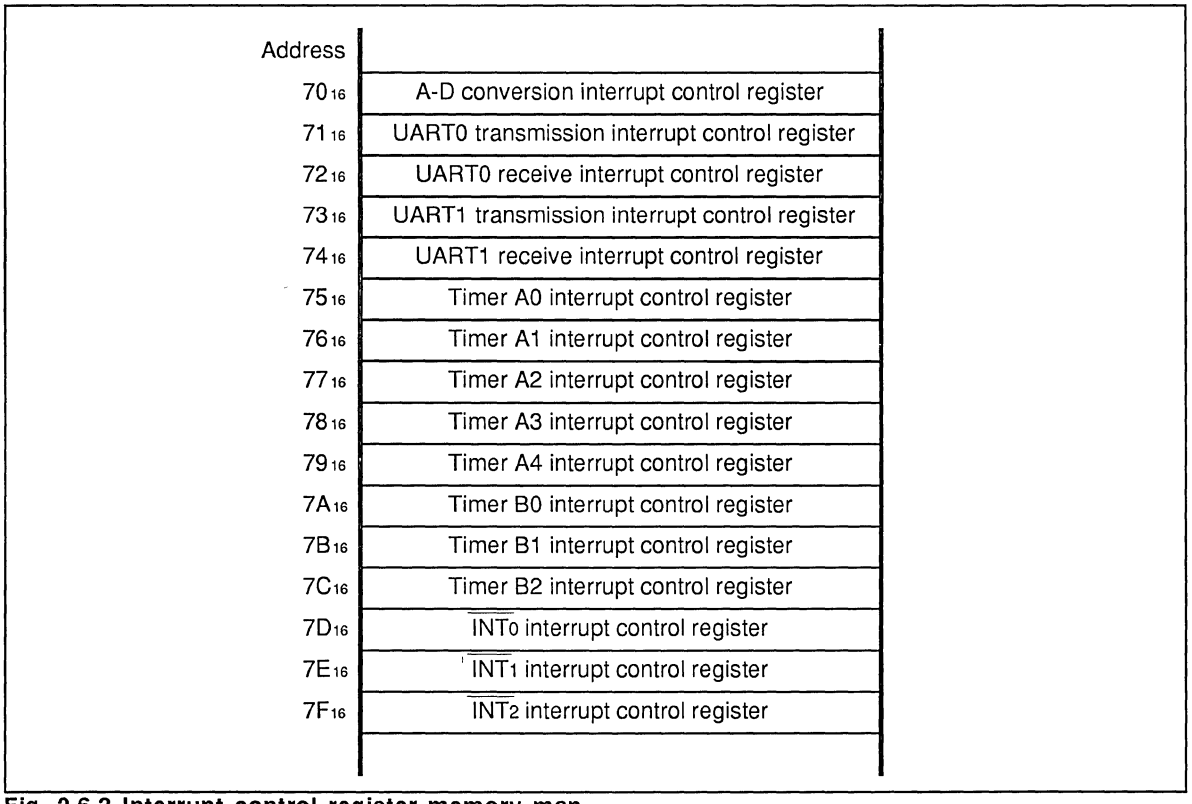
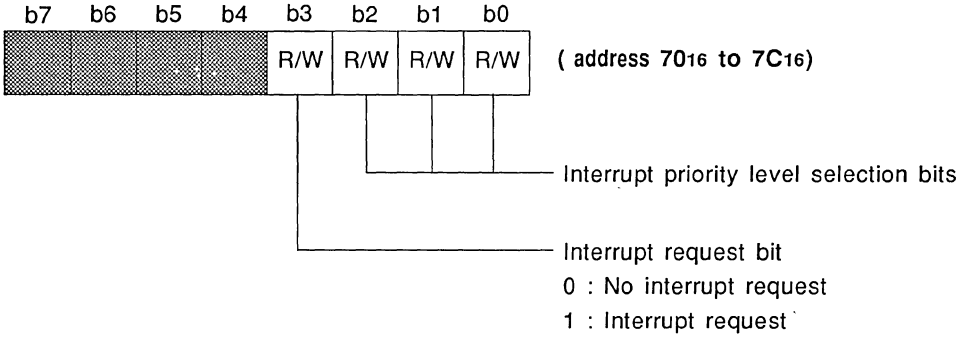


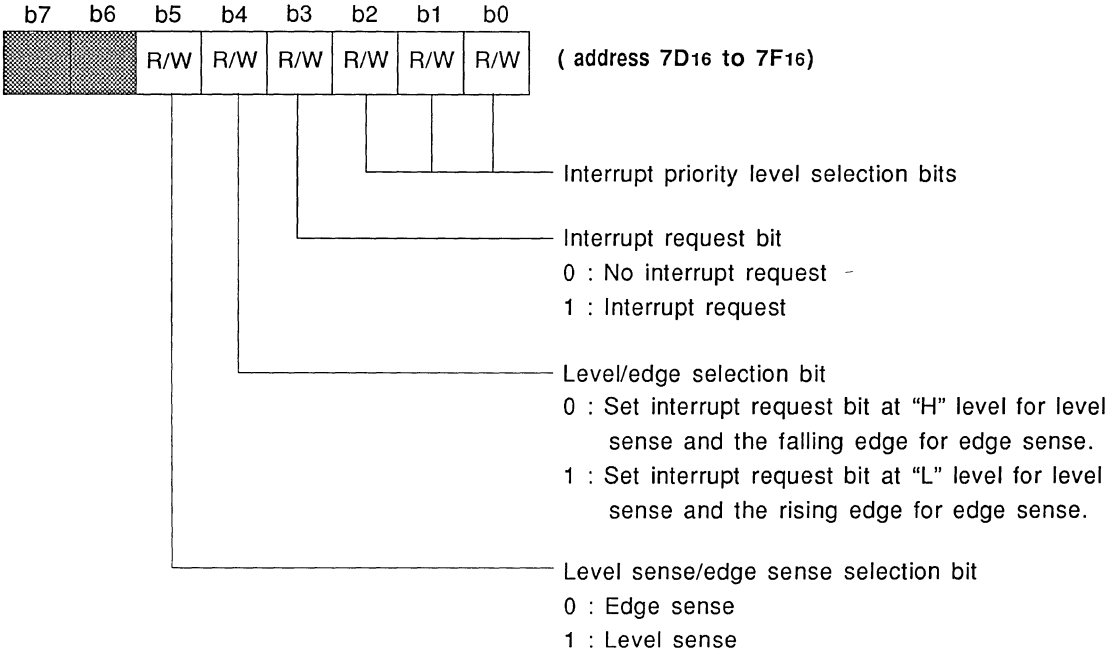
Fig. 2.6.3 Interrupt control register memory map

●A-D conversion, UART0 and 1 transmission, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers



Note: Bits 4 to 7 are undefined at reading.
Bits 0 to 3 are cleared to "0" at reset.

●$\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers



Note: Bits 6 and 7 are undefined at reading.
Bits 0 to 5 are cleared to "0" at reset.

Fig. 2.6.4 Interrupt control register structure

The interrupt control bits are described below.

(1) Interrupt disable flag (I flag)

The interrupt disable flag (I flag) is assigned to the processor status register bit 2. This flag can be used to disable all maskable interrupts. All maskable interrupts are disabled when the I flag is set to "1" and enabled when it is cleared to "0". This flag is set to "1" at reset and must be cleared to "0" if interrupts are to be enabled.

(2) Interrupt request bit

When an interrupt request occurs, the interrupt request bit which is bit 3 of the corresponding interrupt control register is set to "1". The interrupt request bit remains set until the interrupt is accepted, and is cleared to "0" when the interrupt is accepted. This flag is used to indicate that an interrupt request has occurred. This bit can also be set or cleared by program.

The $\overline{INT_i}$ interrupt request bit has a function different from the above description when used in level sense mode.

●$\overline{INT_i}$ interrupt request bit

When the $\overline{INT_i}$ interrupt is used in level sense mode (level sense/edge sense selection bit set to "1"), the $\overline{INT_i}$ interrupt request bit becomes a status bit indicating the status of the $\overline{INT_i}$ input pin (ports $P6_2$ to $P6_4$).

Therefore, the $\overline{INT_i}$ interrupt request bit is set to "1" when the $\overline{INT_i}$ input pin level is "H", and to "0" when it is "L" regardless of whether an interrupt request occurs or is accepted.

(3) Interrupt priority level and processor interrupt priority level (IPL)

An interrupt priority level between 0 and 7 can be assigned to each interrupt by using the interrupt priority level selection bits which are assigned to bits 0 to 2 of each interrupt control register. When an interrupt request occurs, this priority level is compared with the processor interrupt priority level in the processor status register.

Interrupt priority level > Processor interrupt priority level (IPL)

An interrupt is enabled when the above condition is satisfied. Therefore, an interrupt can be disabled by setting its priority level to 0.
The interrupt disable flag, interrupt request bit, interrupt priority level, and IPL are independent of each other and do not affect each other. An interrupt is generated only when all of these bits are properly set. These bits can be used to control interrupt priorities in a variety of ways by program.
Table 2.6.4 shows the setting of interrupt priority levels and Table 2.6.5 shows the interrupt enable levels corresponding to IPL setting.

Table 2.6.4 Setting of interrupt priority level

Interrupt control register			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 2.6.5 Interrupt enable levels corresponding to IPL setting

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 above interrupts.
1	1	0	Enable level 7 interrupts.
1	1	1	Disable all maskable interrupts.

IPL_0: Processor status register bit 8
IPL_1: Processor status register bit 9
IPL_2: Processor status register bit 10

2.6.4 Interrupt priority level

All interrupts have an assigned priority level. When more than one interrupt request occurs during the same sampling interval (interval in which interrupt requests are checked) while all interrupts are enabled, the one with the highest priority is accepted.

The priority level of all of the 19 sources except for software interrupts (zero divide and **BRK** instruction interrupt) and watchdog timer interrupt can be set by program using the interrupt priority level selection bits in the interrupt control register. Reset (highest priority) and watchdog timer interrupt priorities are set by the hardware. Figure 2.6.5 shows the hardware controlled interrupt priorities.

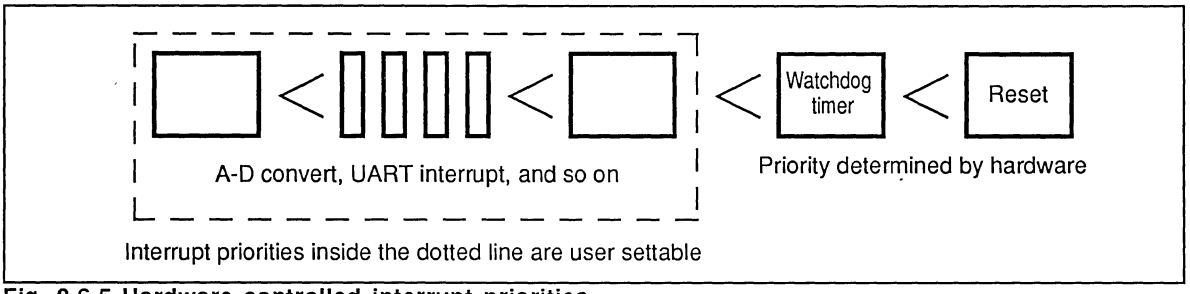


Fig. 2.6.5 Hardware controlled interrupt priorities

2.6.5 Interrupt priority level detection circuit

The M37702 group is equipped with an interrupt priority level detection circuit to select the highest priority when more than one interrupt request occurs during the same sampling interval. Figure 2.6.6 shows the interrupt priority level detection circuit.

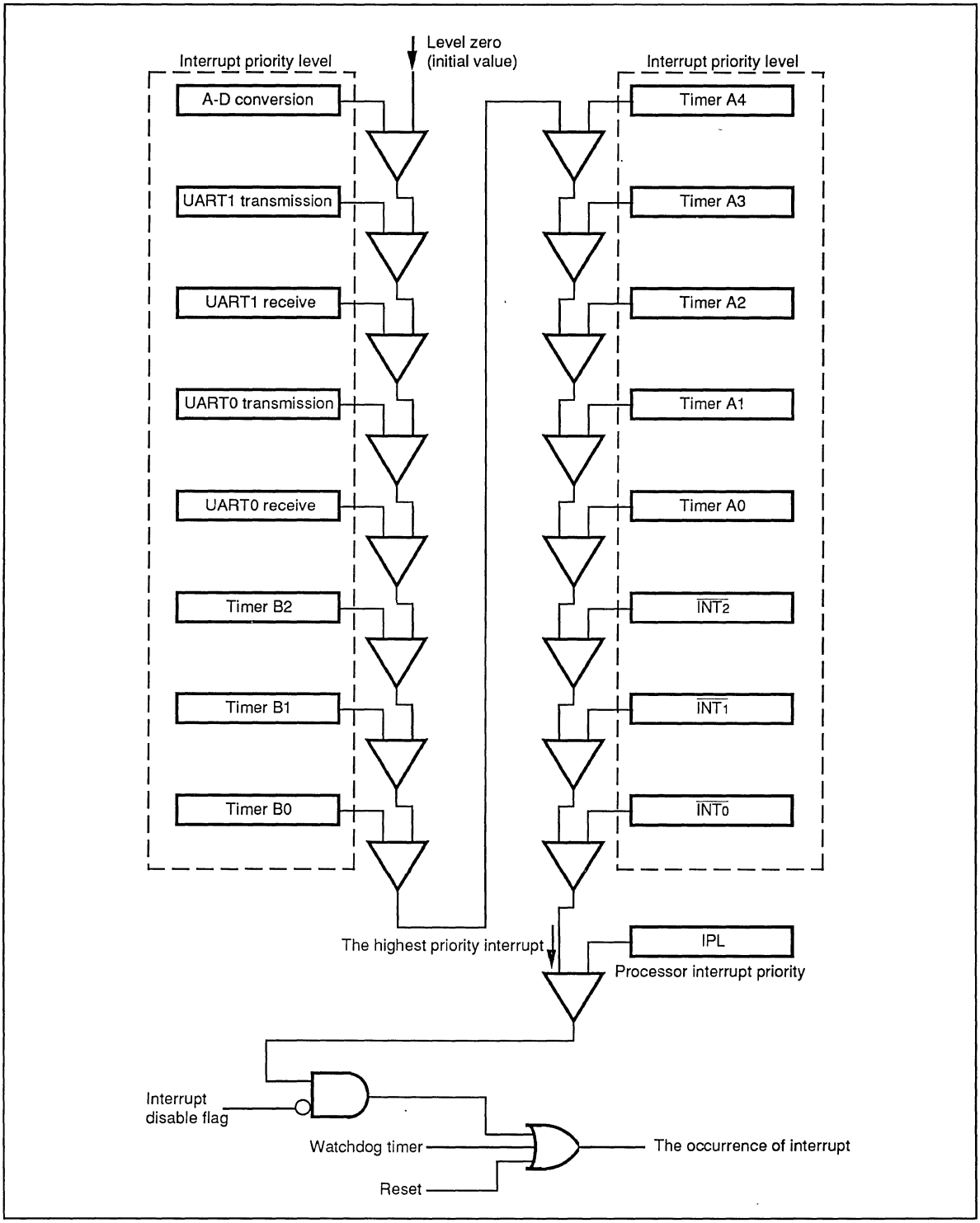
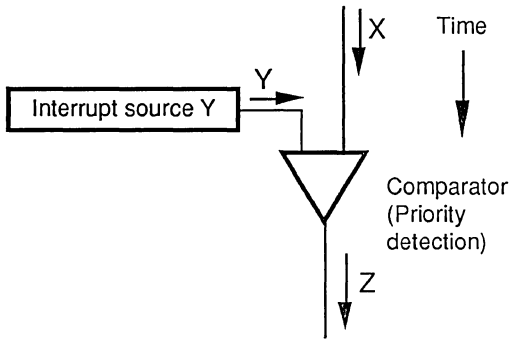


Fig. 2.6.6 Interrupt priority level detection circuit



X:The highest priority level at this point
Y:Priority level of interrupt source Y
Z:The highest priority level at this point

- If X≥Y then Z=X
- If X<Y then Z=Y

[Relation when the software set priority levels are same]

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

Fig. 2.6.7 Interrupt priority level detection model

The priority level of the requested interrupt (Y in Figure 2.6.7) (initial priority level is 0) is compared with the priority of the upstream interrupt (X in Figure 2.6.7) in the order shown in Figure 2.6.6 and the higher level interrupt is sent downstream for comparison with next interrupt (Z in Figure 2.6.7). Unrequested interrupts are not compared and upstream interrupt is passed unchanged to downstream. If the priority levels are equal, the upstream interrupt is selected. Therefore, the following relation exists if the software set priority levels are the same.

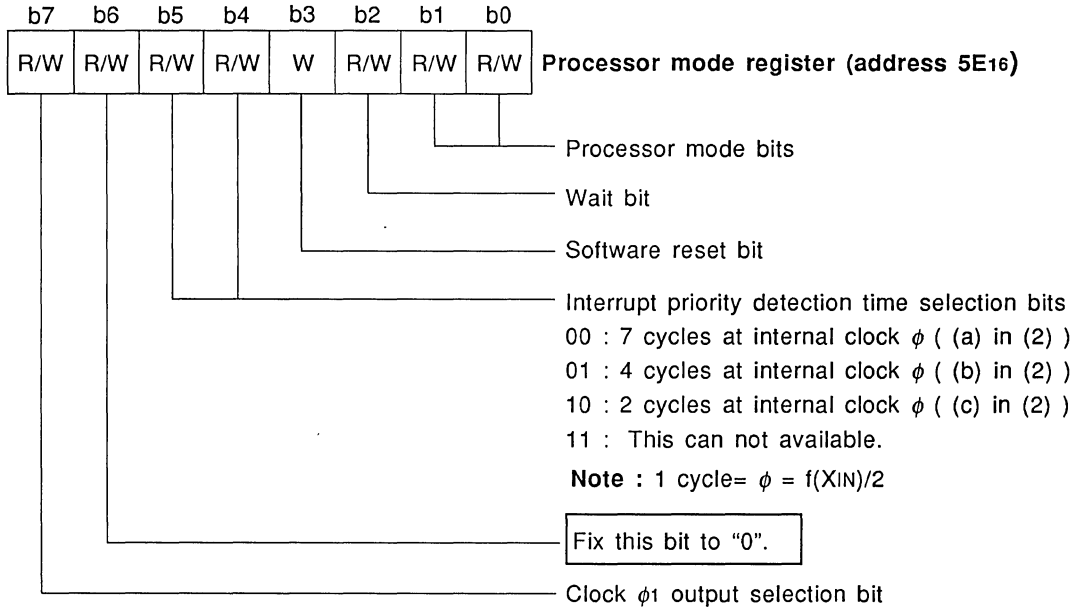
$\overline{INT_0}$ < $\overline{INT_1}$ < $\overline{INT_2}$ < Timer A0 < Timer A1 < Timer A2 < Timer A3 < Timer A4 < Timer B0 < Timer B1 < Timer B2 < UART0 receive < UART0 transmission < UART1 receive < UART1 transmission < A-D conversion

When there are multiple interrupts during the same sampling interval, the interrupt with the highest priority is selected as the result of this comparison. Then, that interrupt is enabled and its interrupt service routine is executed if its interrupt priority level is higher than the processor interrupt priority level (IPL) and the interrupt disable flag is "0".
The detection of interrupt priority level is synchronized with the sampling pulse generated during the operation code fetch cycle. While the interrupt level is being checked, the interrupt request bit and the interrupt priority level are latched so that they do not change. They are sampled at the first half of the operation code fetch cycle and latched from the second half to the end of the level detection. Note that while the priority is being checked, no sampling pulse is generated even when it is the operation code fetch cycle. (See Figure 2.6.8.)

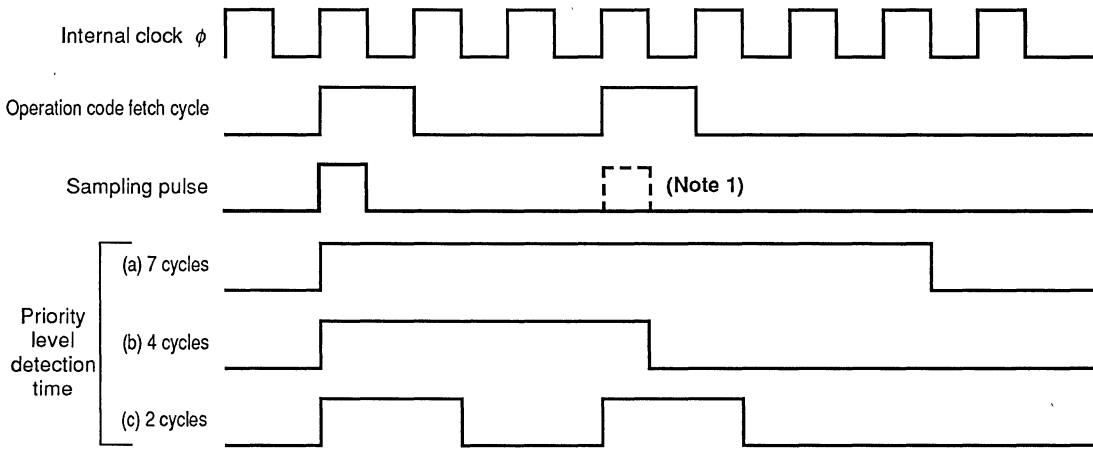
2.6.6 Interrupt priority level detection time

With the M37702 group. the time which it takes for the interrupt priority level detection circuit to determine the level of an interrupt can be set by program. Figure 2.6.8 shows the interrupt priority level detection time. The detection time can be set to 7, 4, or 2 cycles according to the content of the interrupt priority level detection time selection bits (bits 4, and 5 at address $5E_{16}$) in the processor mode register. The interrupt priority level detection time selection bits are cleared to "00" at reset.

(1) Interrupt priority detection time selection bits



(2) Priority level detection time



Note 1 : Pulse exists if 2 cycles is selected for priority level detection time.

Fig. 2.6.8 Interrupt priority level detection time

2.6.7 Interrupt processing sequence

The sequence of events from the receiving of interrupt by the main routine to the start of the interrupt routine is referred to as the interrupt processing sequence.
When an interrupt is accepted, interrupt processing starts from the next cycle of the interrupted instruction. Figure 2.6.9 shows the interrupt processing sequence.
After execution of an interrupted instruction completes, an INTACK (Interrupt Acknowledge) sequence is executed and control is passed to the beginning of the interrupt service routine. The INTACK sequence operates as follows.

① The content of the program bank register (PG) just before the INTACK sequence is stored to stack.
② The content of the program counter (PC) just before the INTACK sequence is stored to stack.
③ The content of the processor status register (PS) just before the INTACK sequence is stored to stack.
④ The interrupt disable flag is set to "1".
⑤ The interrupt request bit of the accepted interrupt is cleared to "0".
⑥ The priority level of the accepted interrupt is set in IPL.
⑦ The content of the program bank register (PG) is set to "00_{16}" and the interrupt vector address is loaded in the program counter (PC).

The INTACK sequence requires a minimum 13 cycles (1 cycle=ϕ=f(X_{IN})/2). Figure 2.6.10 shows the INTACK sequence timing.

The interrupt service routine is started after completing the INTACK sequence.

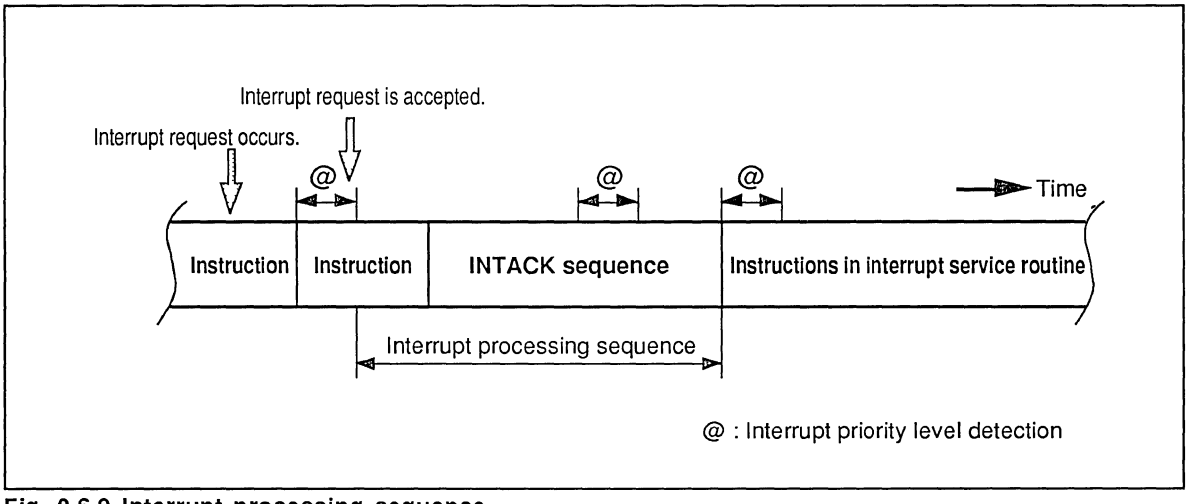


Fig. 2.6.9 Interrupt processing sequence

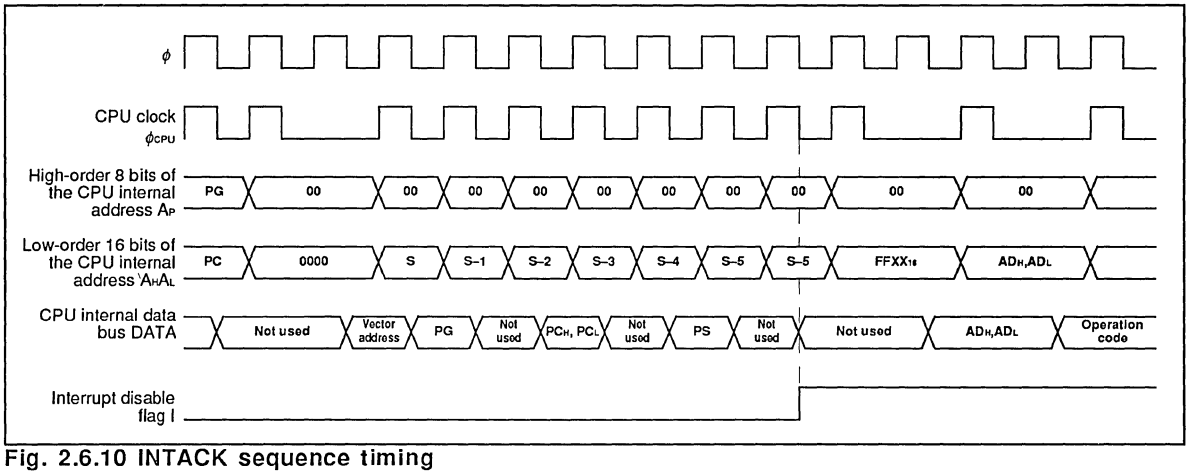


Fig. 2.6.10 INTACK sequence timing

(1) Change in IPL when an interrupt is generated

When an interrupt is accepted, the IPL in the processor status register is replaced by the priority level of the accepted interrupt.

If the interrupt is a reset, watchdog timer, or software interrupt, the value shown in Table 2.6.6 is set in the IPL. This operation is meaningful when multiple interrupts are used (refer to section "2.6.9 Multiple interrupts").

Table 2.6.6 Change in IPL when an interrupt is generated

Interrupt source	Change in processor interrupt level
Reset	0 (000_2)
Watchdog timer	7 (111_2)
Zero divide	No change
BRK instruction	No change
Other interrupts	Priority level of the accepted interrupt

(2) Storing registers

The register storing operation performed during INTACK sequence depends on whether the content of the stack pointer S at the time of accepting interrupt is even or odd.

When the content of the stack pointer S is even, the 16-bit contents of the program counter PC and processor status register PS are stored simultaneously at each other. When the content of the stack pointer S is odd, they are stored in 8-bit unit. Figure 2.6.11 shows the register storing operations.

In the INTACK sequence, only the contents of the program bank register PG, program counter PC, and the stack pointer S are stored to stack area. Other registers must be stored at the beginning of the interrupt service routine as necessary.

The M37702 group provides a **PSH** instruction which stores all registers except for the stack pointer with a single instruction.

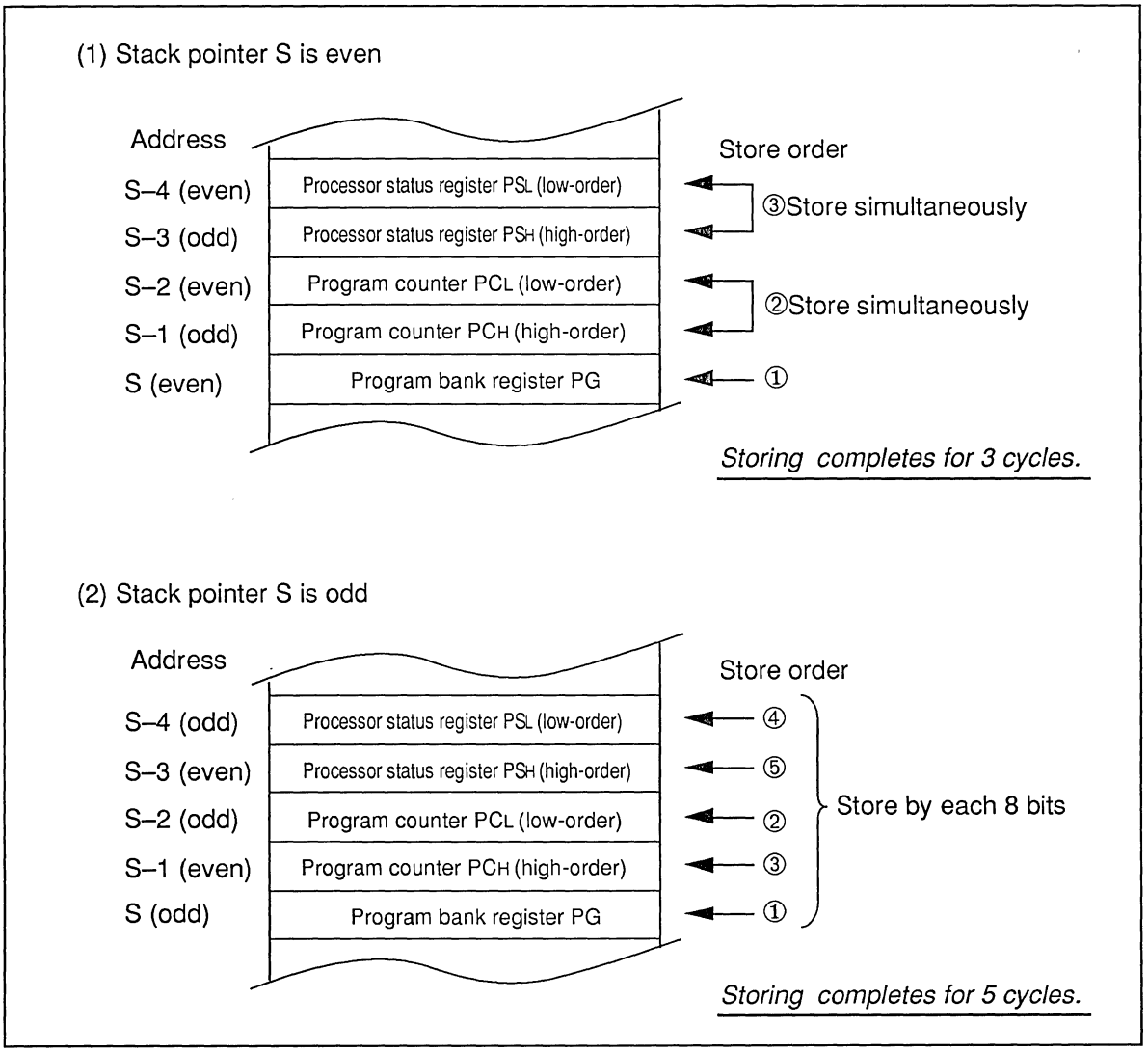


Fig. 2.6.11 Register storing operation

2.6.8 Returning from an interrupt service routine

An **RTI** instruction is used at the end of the interrupt service routine to return to the interrupted routine and continue processing. The **RTI** instruction restores the contents of the program bank register PG, the program counter PC, and the processor status register PS stored before entering the interrupt service routine to their original registers.

The registers stored within the interrupt service routine must be restored with the **PUL** instruction before executing the **RTI** instruction. The status of other interrupt request bits are retained after branching to the interrupt service routine. Therefore, if these interrupts are to be disabled after returning, these request bits must be cleared to "0" before executing the **RTI** instruction.

2.6.9 Multiple interrupts

The interrupt disable flag I is set to "1" in order to disable further interrupts and the interrupt request bit of the accepted interrupt is cleared to "0" when a branch is made to an interrupt service routine.

However, if there are multiple interrupts, the interrupt request bit of the interrupt that was rejected by the priority detection circuit remains set to "1".

The IPL (processor interrupt priority level) in the processor status register changes to the priority of the accepted interrupt. Therefore, if the interrupt disable flag I is cleared to "0" in the interrupt handling routine for the accepted interrupt, interrupts with priority higher than the current interrupt can be accepted as long as IPL is unchanged. This is referred to as multiple interrupts.

Figure 2.6.12 shows the multiple interrupt mechanism.

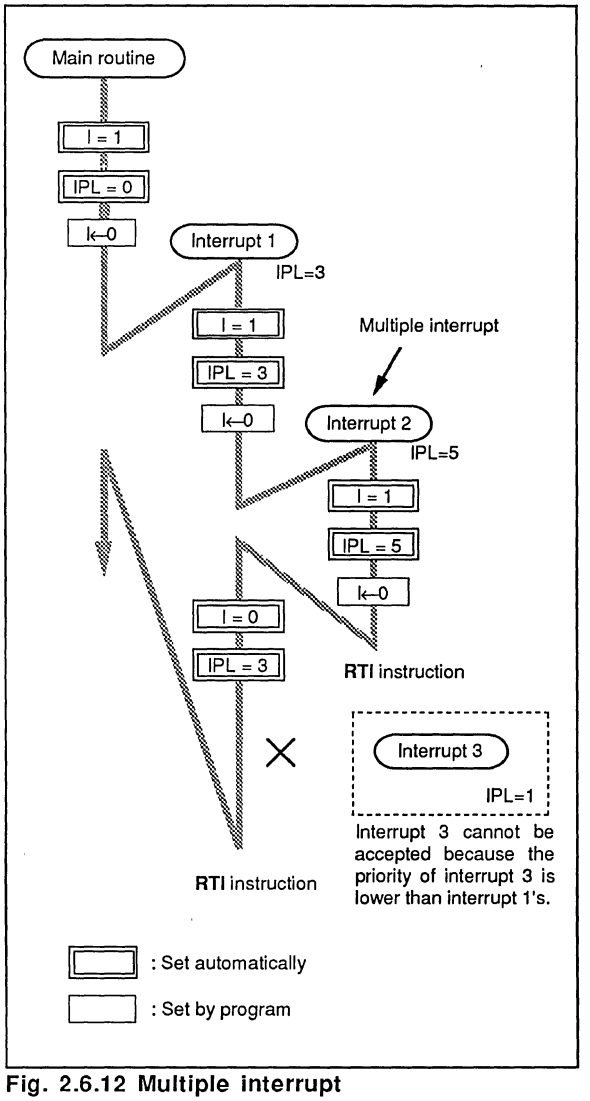


Fig. 2.6.12 Multiple interrupt

2.6.10 Interrupt response time and interrupt delay time

When an interrupt request occurs, the priority must be checked and the INTACK sequence must be executed before interrupt service routine can start. The interval between the interrupt request and the start of the interrupt service routine is referred to as the interrupt response time. This is shown in Figure 2.6.13. Interrupt priority detection is performed at the beginning of each instruction and during the INTACK sequence. This is performed in parallel with the instruction execution. Therefore, the interrupt response time is the sum of ① through ③ as follows (in Figure 2.6.13):

① The interval from the time of the interrupt request until the instruction being executed completes.

② The interval from the start of the next instruction (start of interrupt priority detection) until the end of the instruction being executed at the end of priority detection.

③ Time required to execute the INTACK sequence (13 cycles minimum, 15 cycles maximum).

If registers are stored at the beginning of the interrupt service routine, the time required for this operation (context switching time) must also be added. The sum of the interrupt response time and the context switching time is the interrupt delay time (see Figure 2.6.13). This is the time required for the interrupt processing program to start after an interrupt request occurs. The interrupt delay time is expressed by the following equation.

Interrupt delay time=(time required to execute instruction 1) + (time required to execute instruction 2) − 0.5ϕ + (INTACK sequence interval) + (context switching time)

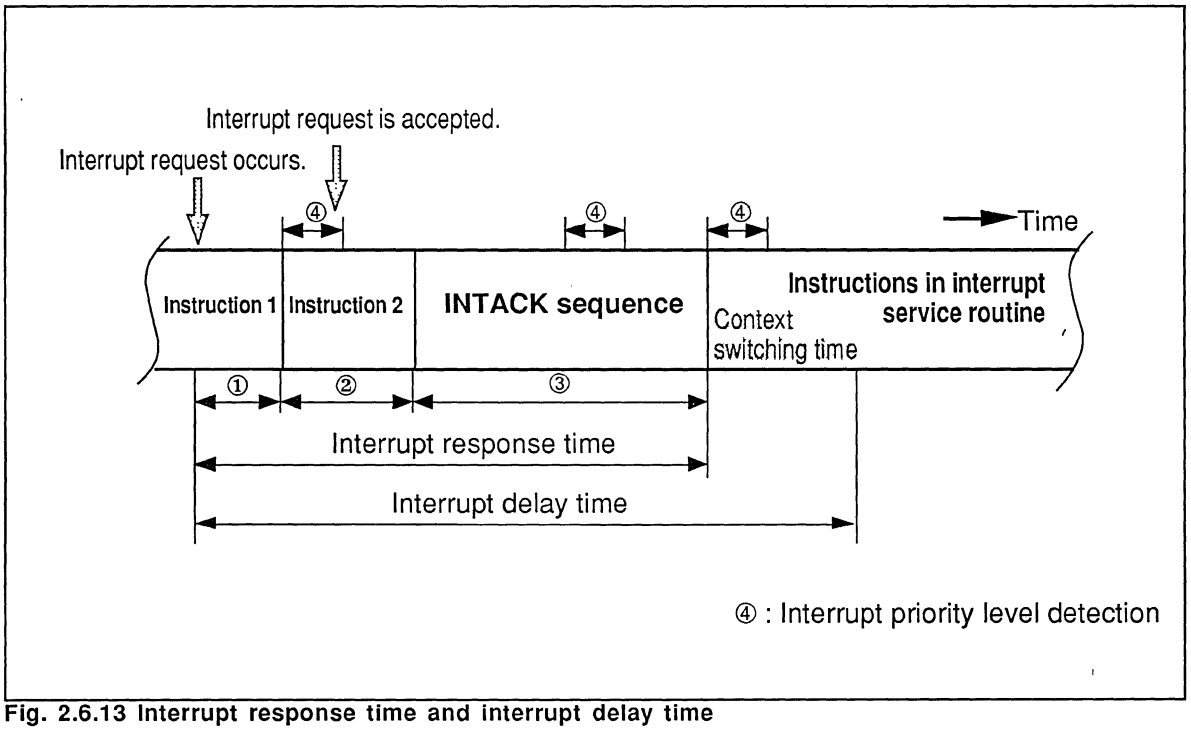


Fig. 2.6.13 Interrupt response time and interrupt delay time

[Precautions when using interrupts]

1. When the $\overline{INT_i}$ interrupt is used in level sense mode, the $\overline{INT_i}$ interrupt request bit functions as a status bit which indicates whether the $\overline{INT_i}$ pin level is valid or invalid. It is set to "1" while valid and to "0" while invalid. After an interrupt is accepted, the interrupt request bit remains set to "1" while valid level is supplied to the $\overline{INT_i}$ pin.
   The interrupt request is not retained when changing from valid level to invalid level if no interrupt is generated during valid level. Figure 2.6.14 shows the $\overline{INT_i}$ interrupt during level sense mode.
   If the level of the $\overline{INT_i}$ pin is valid (did not change from valid level to invalid level) when returning to the original routine after processing an interrupt, another interrupt is generated as shown in Figure 2.6.15.

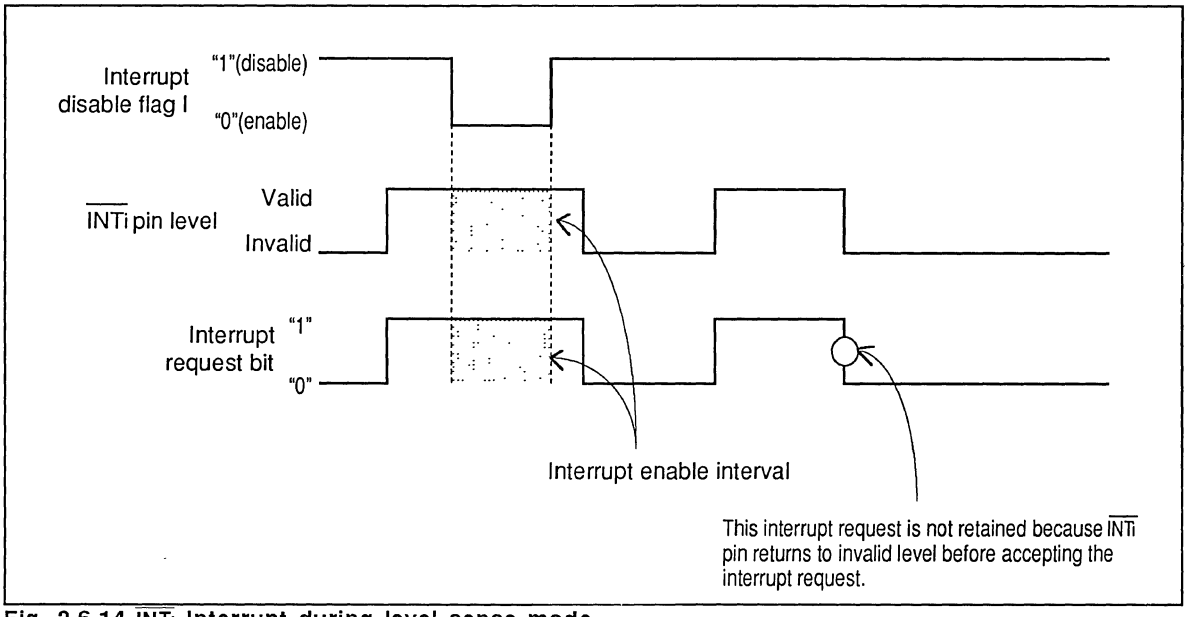


Fig. 2.6.14 $\overline{INT_i}$ Interrupt during level sense mode

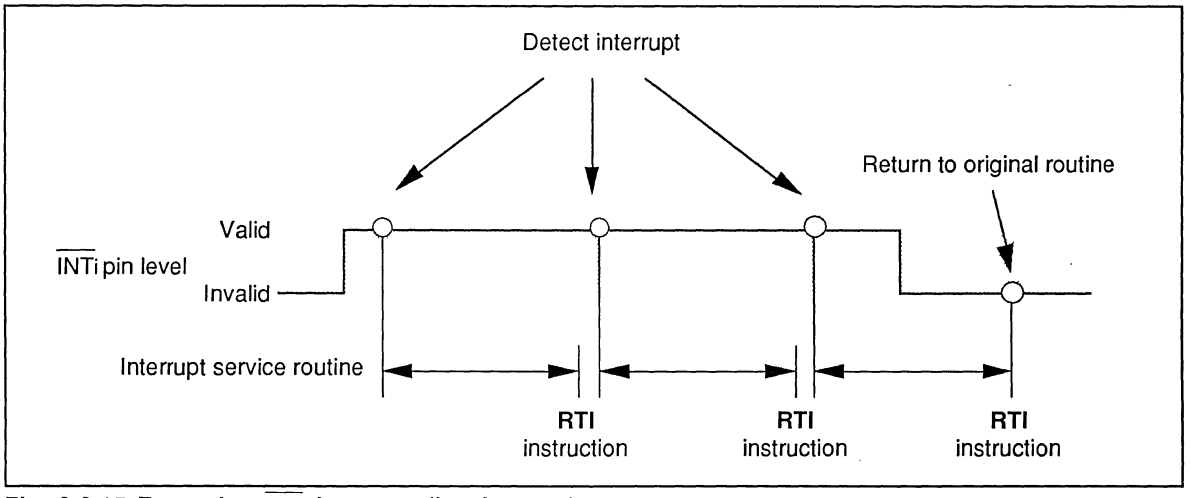


Fig. 2.6.15 Repeating $\overline{INT_i}$ interrupt (level sense)

2. To change the $\overline{INTi}$ interrupt from level sense to edge sense, set the $\overline{INTi}$ interrupt control register in the sequence shown in Figure 2.6.16.

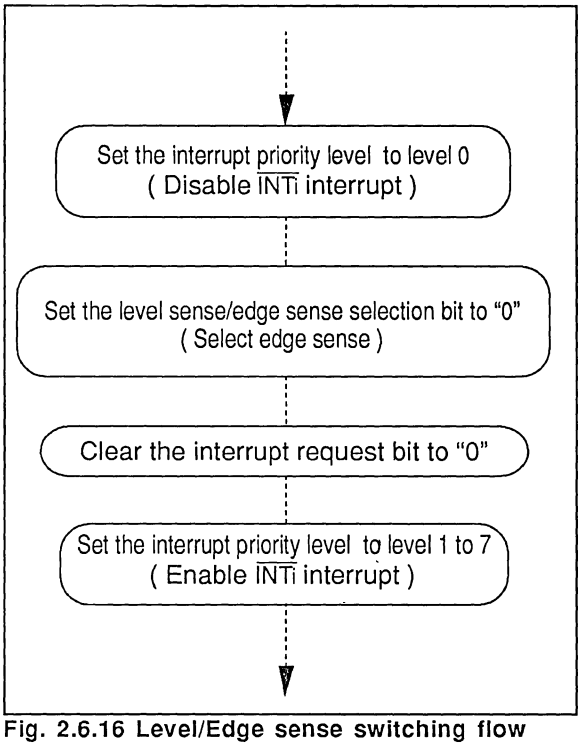


Fig. 2.6.16 Level/Edge sense switching flow

3. To change the $\overline{INTi}$ interrupt phase, set the $\overline{INTi}$ interrupt control register in the sequence shown in Fig. 2.6.17.

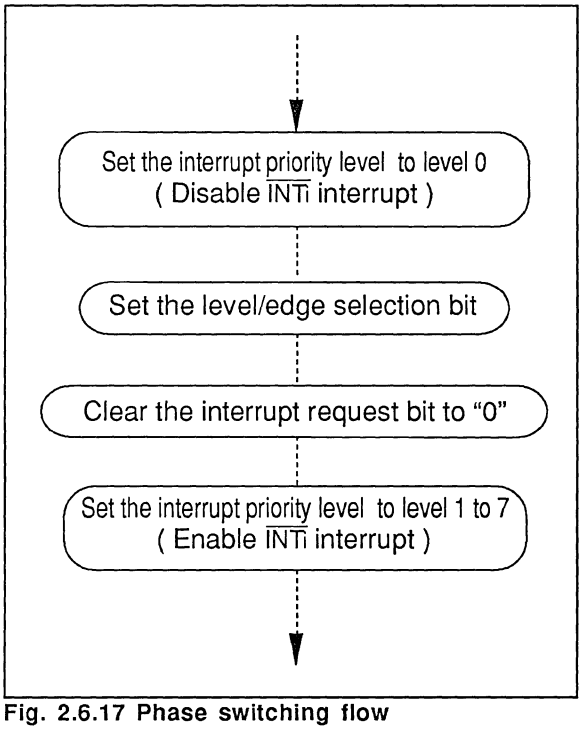


Fig. 2.6.17 Phase switching flow

2.7 Timer A

Timer A consists of five 16-bit timers (timers A0 to A4). External output is the main function of these timers. Timers A0 to A4 operate independently and each can operate in one of four different modes.

2.7.1 Timer A description

Timer Ai (i=0 to 4) has four operating modes as described below. These timers have identical functions except the two-phase pulse signal processing function in event counter mode.
The timer I/O pins are shared with ports $P5_0$ to $P5_7$, $P6_0$, and $P6_1$.

●Timer mode

This mode counts the selected internal clock. The counter is decremented by 1 each time a count source (selected clock) is input and a timer Ai interrupt request occurs when the content of the counter reaches 0000_{16} → reload value "n" (hereafter referred to as underflow). Gate function (control count operation with the input level to the TAi_{IN} pin) and pulse output function (output from the TAi_{OUT} pin, a signal that changes phase each time the counter underflows) are available and can be selected by program.

●Event counter mode

This mode counts the external clock input to the TAi_{IN} pin. The counter can be incremented (add 1 to the content of the counter with each clock input) or decremented (subtract 1 from the content of the counter with each clock input) by program or with an external signal. In case used as an increment counter, a timer Ai interrupt request occurs when the counter reaches $FFFF_{16}$ → reload value "n" (hereafter referred to as overflow). In case used as a decrement counter, a timer Ai interrupt request occurs when the counter underflows. In addition, the pulse output function (output from the TAi_{OUT} pin, a signal that changes phase each time the counter underflows or overflows) can be selected by program. Timers A2, A3, and A4 also have two-phase pulse signal processing function which controls the counter increment/decrement by a two-phase signal with the phases shifted by 90 degrees.

●One-shot pulse mode

In this mode, the timer is driven by an internal or external trigger and "H" level is output from the TAi_{OUT} pin for an arbitrary interval.

●Pulse width modulation (PWM) mode

In this mode, an arbitrary pulse width signal is output repeatedly from the TAi_{OUT} pin. PWM output is started by an internal or external trigger.

2.7.2 Block description

Figure 2.7.1 shows the block diagram of timer Ai. It is followed by the description of timer Ai related registers.

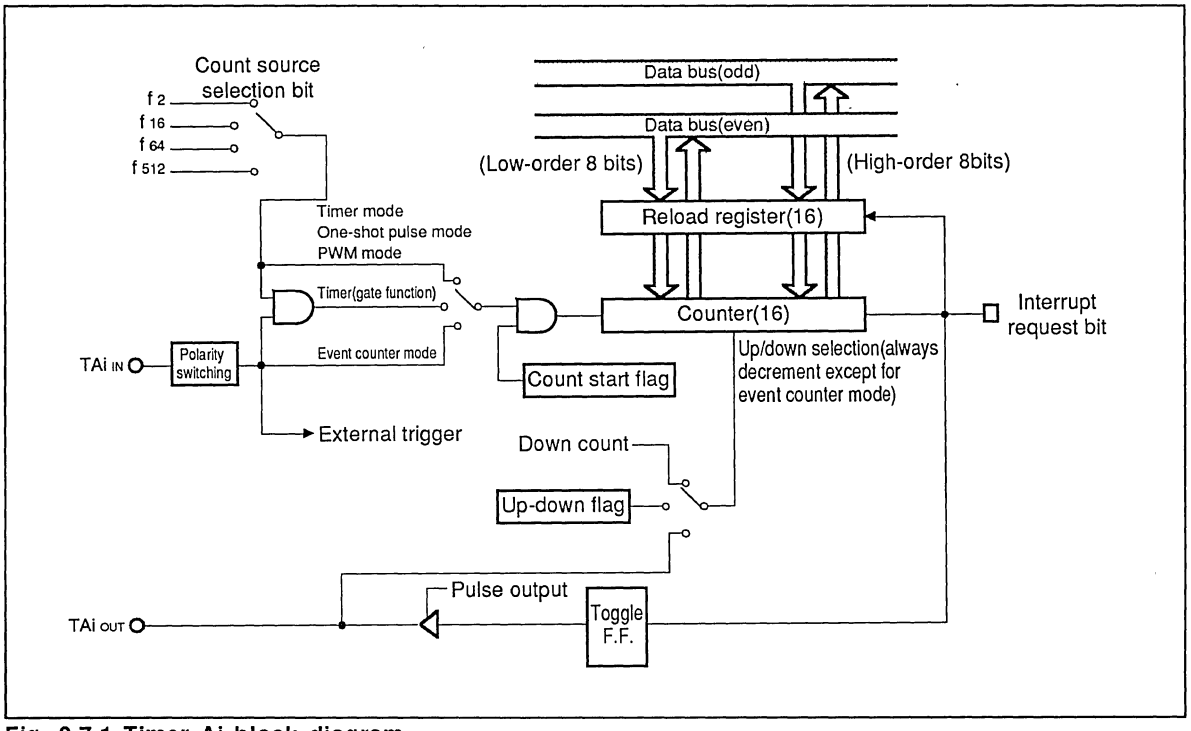


Fig. 2.7.1 Timer Ai block diagram

(1) Counter and reload register (timer Ai register)

Timer Ai counter and reload register consist of 16 bits. The counter counts the selected count source and its content is decremented by 1 each time a count source is input. In event counter mode, it can also function as an increment counter and the counter is incremented by 1 each time a count source is input. The reload register is used to store the initial value of the counter. The content of the reload register is reloaded into the counter when the counter underflows (or when it overflows in case used as an increment counter in event counter mode).

The content of the counter changes each time a count source is input, but the content of the reload register remains unchanged.

Values are stored in the counter and reload registers by writing to the timer Ai register.

Table 2.7.1 shows the memory allocation of timer Ai register.

Table 2.7.1 Timer Ai register memory allocation

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 $4E_{16}$

The value written in timer Ai register when the timer is not operating is stored in the counter and reload register. The value written in timer Ai register when the timer is operating is stored only in the reload register. In this case, the updated value is transferred to the counter during next reload.

If the timer Ai register is read, the result depends on the operating mode. Table 2.7.2 shows results of timer Ai register read and write.

The value of the timer Ai register is undefined at reset. Therefore, the counter and the reload register must be initialized when first using timer Ai.

Table 2.7.2 Timer Ai register read and write

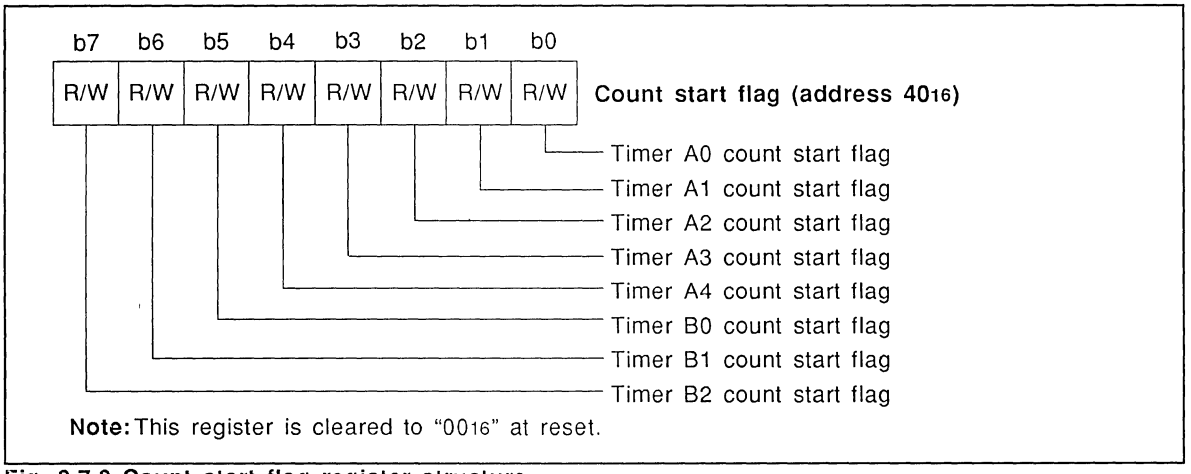
Mode	Read	Write
Timer mode	Read timer counting value	<Timer operating> Write to reload register <Timer halted> Write to reload register and counter
Event counter mode		
One-shot pulse mode	Read undefined value	
Pulse width modulation mode		

(2) Count start flag

The count start flag (address 40_{16}) separately controls starting/stopping of each timer. Each bit corresponds to one of the timers.

A count source is input to the counter when this flag is set to "1" and disabled when it is set to "0".

Figure 2.7.2 shows the structure of the count start flag.



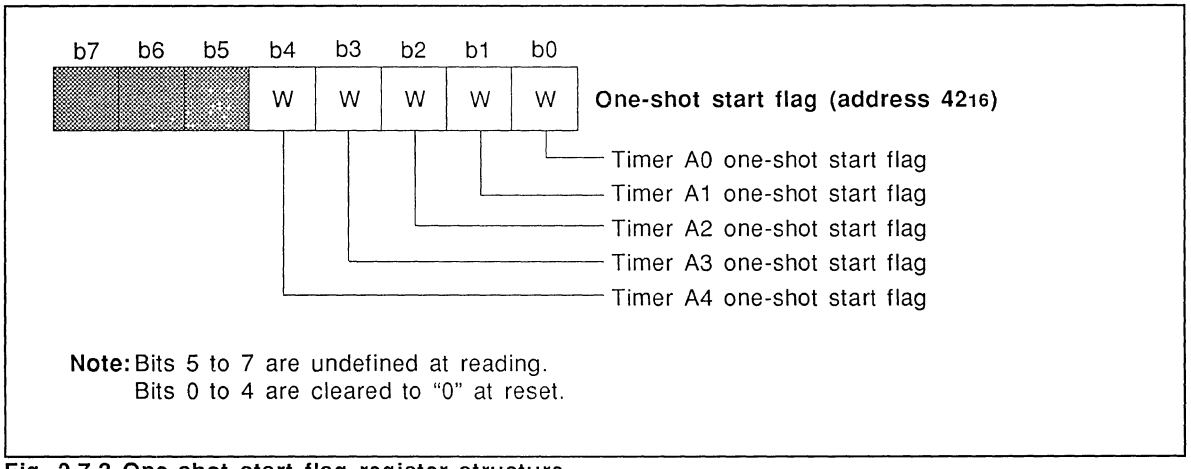
Note: This register is cleared to "00₁₆" at reset.

Fig. 2.7.2 Count start flag register structure

(3) One-shot start flag

The one-shot start flag (address 42_{16}) generates a software trigger used in one-shot pulse mode. When a corresponding one-shot start flag to each timer is set to "1" in case software trigger is selected, a software trigger starting one-shot pulse output is generated. Refer to section "2.7.5 One-shot pulse mode" for more information.

Figure 2.7.3 shows the structure of the one-shot start flag.

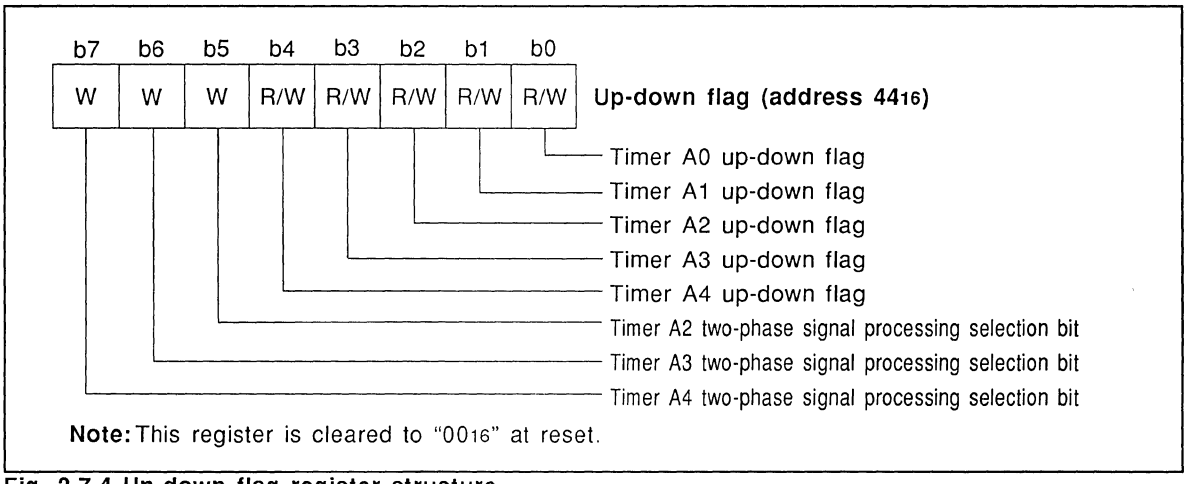


Note: Bits 5 to 7 are undefined at reading.
Bits 0 to 4 are cleared to "0" at reset.

Fig. 2.7.3 One-shot start flag register structure

(4) Up-down flag

The up-down flag (address 44_{16}) is a register consisting of up-down flags and two-phase pulse signal processing selection bits used in event counter mode. Figure 2.7.4 shows the structure of the up-down flag register followed by a description of each bit.



Note: This register is cleared to "00_{16}" at reset.

Fig. 2.7.4 Up-down flag register structure

●Timer Ai up-down flags (bits 0 to 4)

These flags are valid in event counter mode when the count up-down flag is selected as the increment/decrement trigger. The counter of the corresponding timer is decremented when this flag is "0" and incremented when it is "1".

●Two-phase signal processing selection bits (bits 5 to 7)

In event counter mode, these bits select the two-phase pulse signal processing function which controls the counter using two-phase pulse with their phases shifted by 90 degrees. This bit can be an only written to. The corresponding timer operates with the two-phase pulse signal processing when this bit is set to "1". This bit must be set to "0" when the two-phase pulse signal processing function is not used and in modes other than event counter mode. It is cleared to "0" at reset.

Use **LDM** and **STA** instructions to write to this bit. Do not use **SEB** and **CLB** instructions.

(5) Timer Ai mode register

The timer Ai mode register (address 56_{16} to $5A_{16}$) consists of operating mode selection bits, count source selection bits, and timer function selection bits.

Figure 2.7.5 shows the structure of the timer Ai mode register. The operating mode selection bits and count source selection bits are described below. The functions of bits 2 to 5 depend on the operating mode and are described under the description of each operating mode.

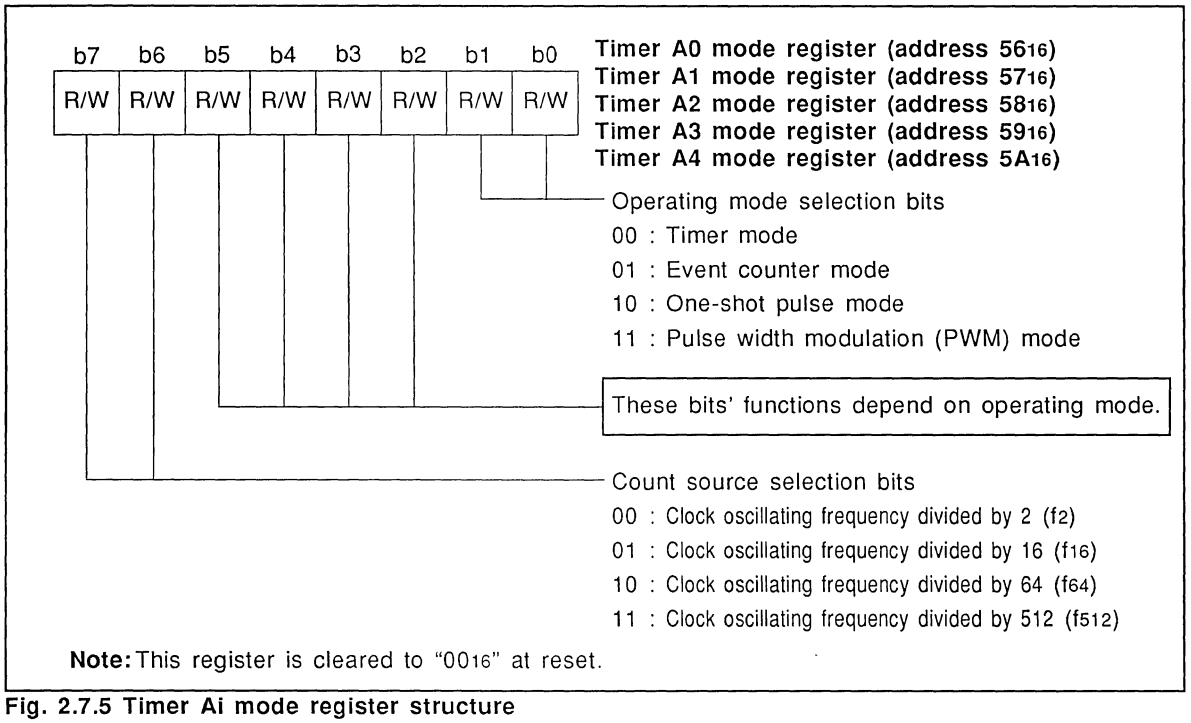


Fig. 2.7.5 Timer Ai mode register structure

●Operating mode selection bits (bits 0 and 1)

The operating mode selection bits are used to select the timer operating mode. Table 2.7.3 shows the relationship between the operating mode selection bits and the timer operating modes.

Table 2.7.3 Relationship between operating mode selection bits and operating mode

b1	b0	Operating mode
0	0	Timer mode
0	1	Event counter mode
1	0	One-shot pulse mode
1	1	Pulse width modulation (PWM) mode

●Count source selection bits (bits 6 and 7)

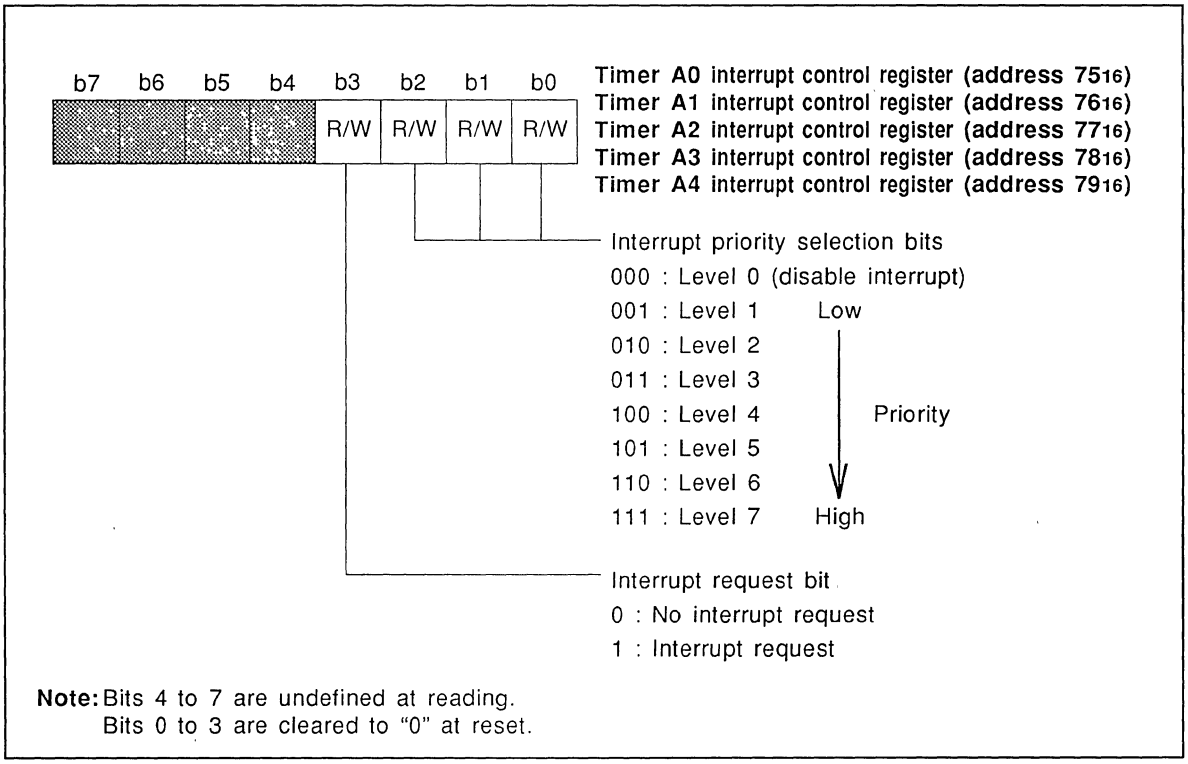
The count source selection bits are used to select the count source. Table 2.7.4 shows the relationship between the count source selection bits and the timer count sources.
These bits are ignored in event counter mode.

Table 2.7.4 Relationship between count source selection bits and count sources

b7	b6	Timer count source	Input clock to the counter		
			f(XIN)=8MHz	f(XIN)=16MHz	f(XIN)=25MHz
0	0	Clock oscillating frequency divided by 2 (f2)	4MHz	8MHz	12.5MHz
0	1	Clock oscillating frequency divided by 16 (f16)	500kHz	1MHz	1.5625MHz
1	0	Clock oscillating frequency divided by 64 (f64)	125kHz	250kHz	390.625kHz
1	1	Clock oscillating frequency divided by 512 (f512)	15625Hz	31250Hz	48.8281kHz

(6) Timer Ai interrupt control register

The timer Ai interrupt control register (address 75_{16} to 79_{16}) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.7.6 shows the structure of the timer Ai interrupt control register. The function of each bit is described below. Refer to section "2.6 Interrupts" for more information.



Note: Bits 4 to 7 are undefined at reading.
Bits 0 to 3 are cleared to "0" at reset.

Fig. 2.7.6 Timer Ai interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using a timer Ai interrupt. When there is an interrupt request, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is greater than IPL(interrupt disable flag I must be "0"). Set these bits to "000" to disable only timer Ai interrupt.

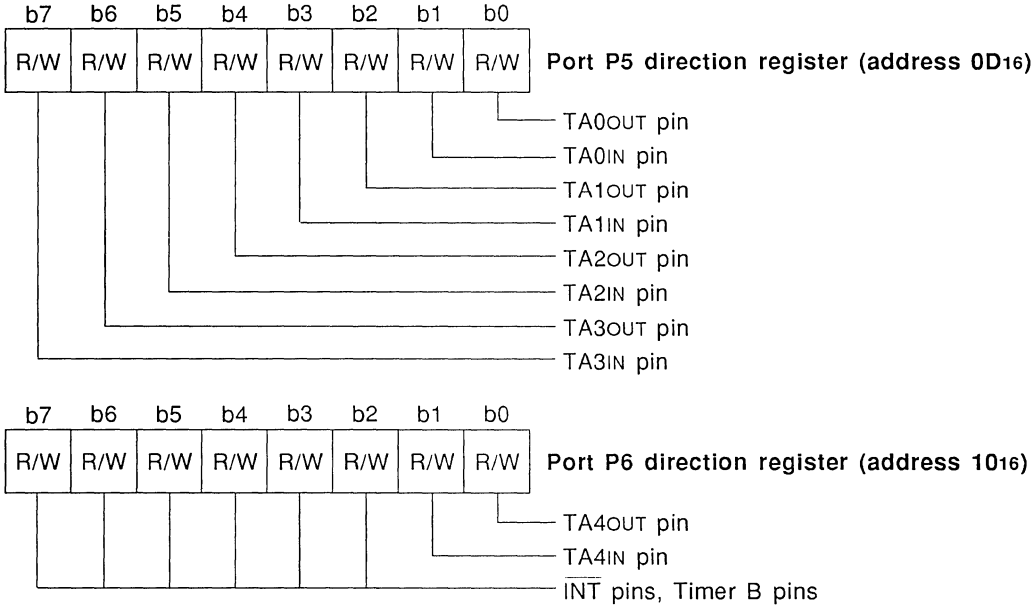
●Interrupt request bit (bit 3)

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

(7) Ports P5 and P6 direction registers

The input/output pins of timers A0 to A3 are shared with port P5 and the input/output pins of timer A4 are shared with port P6. When using these ports as timer input pins, the corresponding bit in the direction register must be set to "0" (input mode). When using these ports as timer output pins, these ports function as timer output pins regardless of the content of the direction register.

Figure 2.7.7 shows the relationship between port P5 direction register (address $0D_{16}$) and port P6 direction register (address 10_{16}) with timer pins.



Note: These registers are cleared to "00_{16}" at reset

Fig. 2.7.7 Relationship between port P5 and P6 direction registers with timer pins

2.7.3 Timer mode [timer Ai mode register bits 1, 0 = "00"]

The timer mode is selected by setting the timer Ai mode register bits 1 and 0 to "00". When this mode is selected, bit 5 of the timer Ai mode register must be set to "0".
Figure 2.7.8 shows the structure of the timer Ai mode register in timer mode.
In timer mode, the selected internal clock is decremented and an interrupt request occurs each time the counter underflows (the content of the counter reaches 0000_{16} → reload value "n"). The timer dividing ratio is expressed as follows:

Timer dividing ratio=1/(n+1)

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

The following functions can be selected with the timer Ai mode register.

●Gate function
Controls count with the input signal to the TAi_{IN} pin.

●Pulse output function
Outputs from the TAi_{OUT} pin, a signal that changes phase each time the counter underflows.

(1) Timer mode operation

First, select the operating mode, pulse output function, gate function, and count source with the timer Ai mode register. Next, write a value "n" ("n" = 0000_{16} to $FFFF_{16}$) in the timer Ai register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register.
When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and starts the count operation.
Figure 2.7.9 shows the setting example of the timer mode related registers.
The content of the counter is decremented by 1 each time the count source is input. When the counter underflows, the content of the reload register is loaded into the counter and the interrupt request bit is set to "1".
Count operation continues in this manner. The interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, an interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.

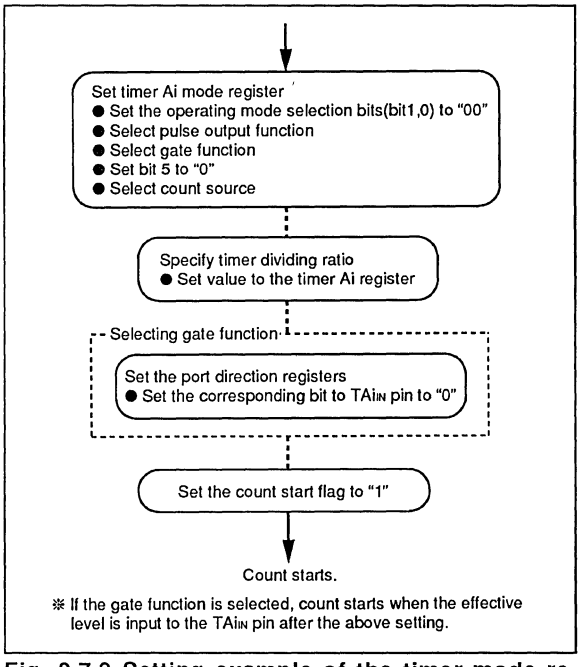
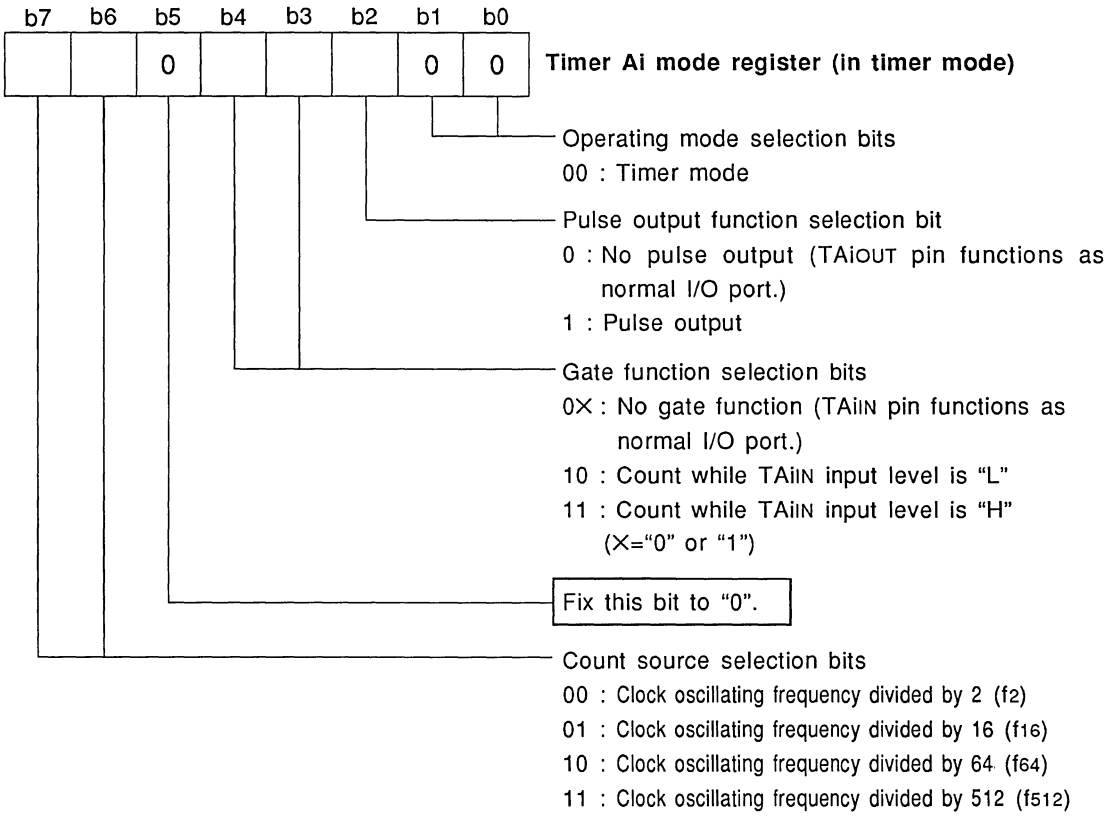


Fig. 2.7.9 Setting example of the timer mode related registers

The content of the counter can be read at any time by reading the content of the timer Ai register, but the content of the reload register can not be read. In order to change the timer dividing ratio while the timer is operating, a 16-bit update value must be written simultaneously in the timer Ai register. This value is stored in the reload register, and loaded into the counter next time the counter underflows after writing to timer Ai register.
Figure 2.7.10 shows the timer mode operation diagram (without pulse output or gate function).



Note: This register is cleared to "00$_{16}$" at reset.

Fig. 2.7.8 Timer Ai mode register structure in timer mode

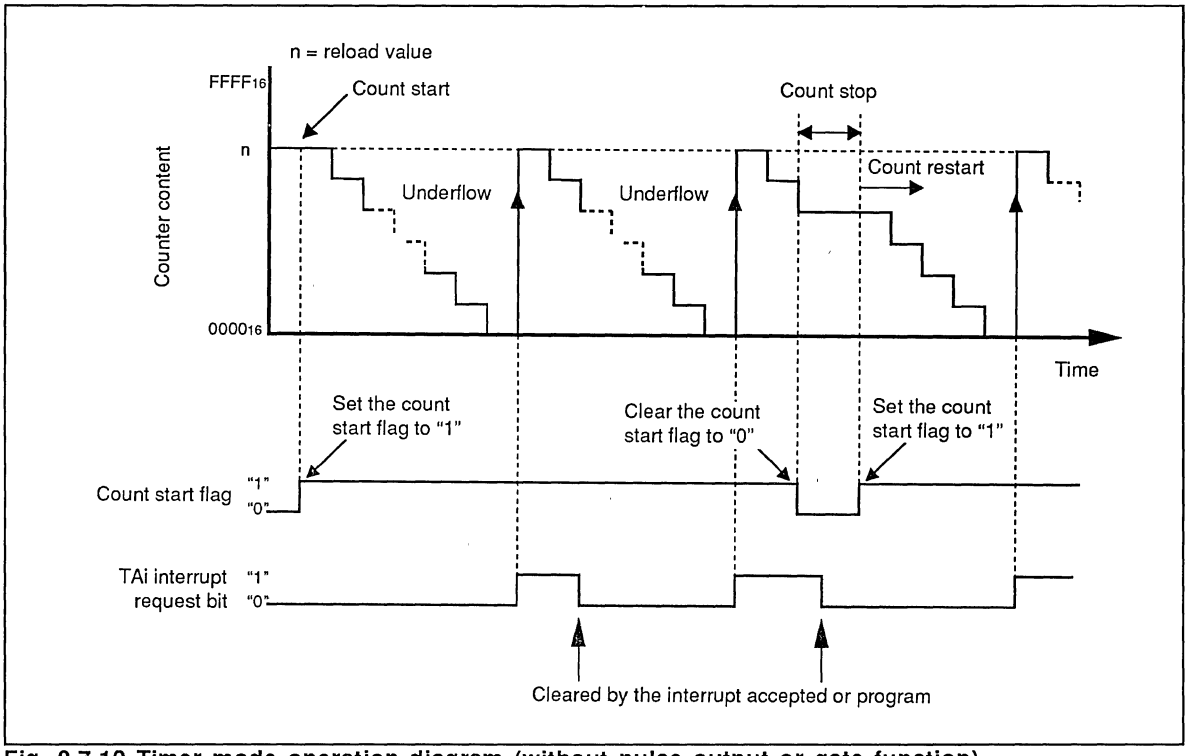


Fig. 2.7.10 Timer mode operation diagram (without pulse output or gate function)

[Selection Function]
Program selectable gate function and pulse output function are described below. These functions can be used together.

●Gate function
The gate function is selected by setting the gate function selection bits of the timer Ai mode register to "10" or "11". The gate function controls the starting and stopping of the timer count by the level of the signal input to the TAiIN pin. Table 2.7.5 shows the relationship between the gate function selection bits and the count effective level.
When using the gate function, the corresponding port direction register to the TAiIN pin must be set to "0" for input mode.

Table 2.7.5 Relationship between gate function selection bit and effective level

Gate function selection bits		Effective level
b4	b3	
1	0	Count while the input level to the TAiIN pin is "L"
1	1	Count while the input level to the TAiIN pin is "H"

When the gate function is selected, counting is performed when count source is input to the counter while the TAiIN pin is at effective level and the count start flag is "1". No count source is input and the count stops while the input level is not effective. The content of the counter is preserved when the TAiIN pin input level changes from effective to non-effective. Therefore, counting can resume when the input level returns to an effective level.

The pulse width of the TAiIN pin input signal during count interval and count halt interval must be at least 2 cycles of the selected count source.

Figure 2.7.11 shows the timer operation example when the gate function is selected.

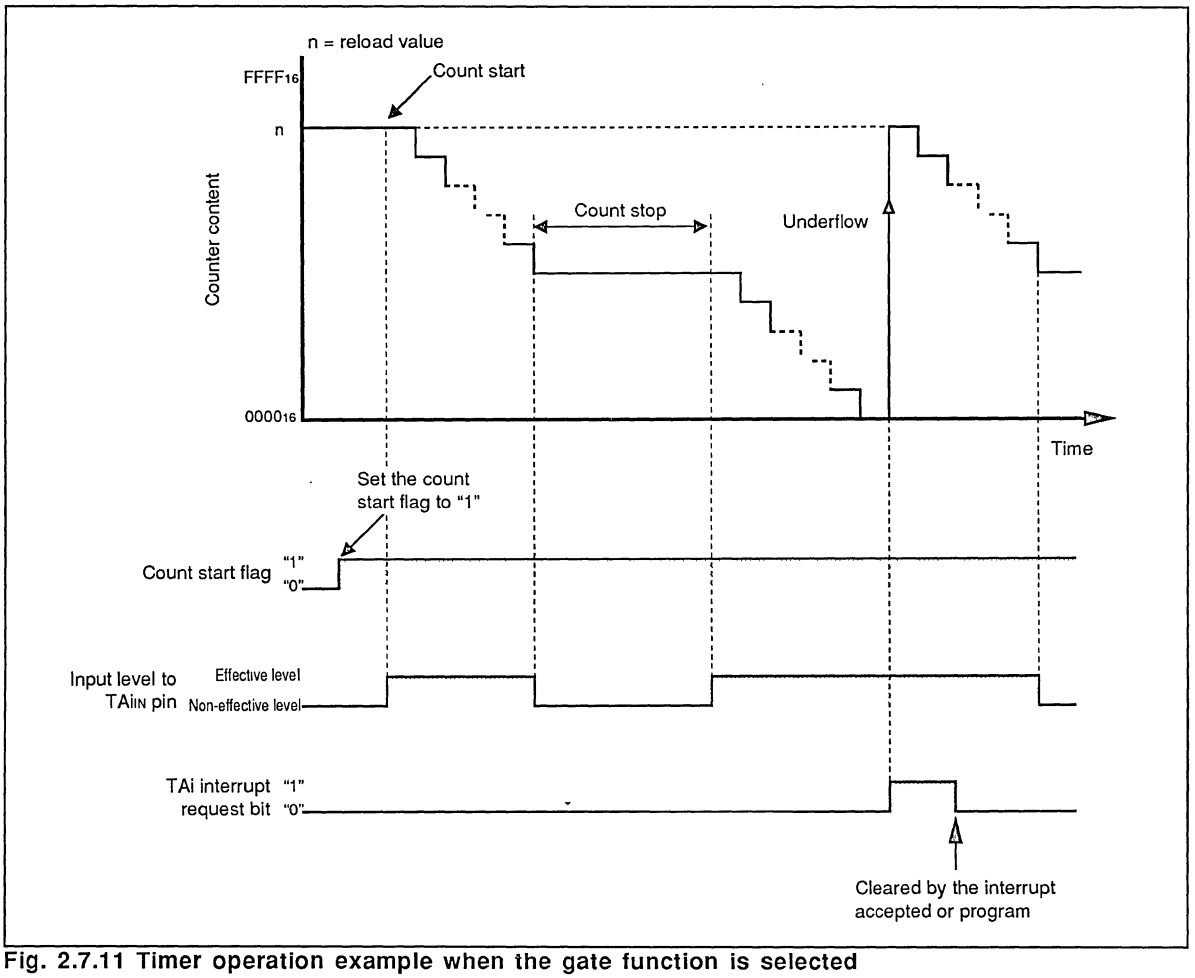


Fig. 2.7.11 Timer operation example when the gate function is selected

●**Pulse output function**

The pulse output function is selected by setting the pulse output function selection bit to "1". When the pulse output function is selected, a signal that changes phase each time the counter underflows is output from the TAiOUT pin. The TAiOUT pin is shared with ports P5 and P6. When the pulse output function is selected, the corresponding port is forced to output mode and functions as the TAiOUT pin regardless of the content of the port direction register. It can be used as a programmable I/O port once the pulse output function selection bit is set to "0".

"L" level is output from the TAiOUT pin while the count start flag is "0" and count disabled. Therefore, the initial level of the output pulse is always "L".

Figure 2.7.12 shows the timer operation example when the pulse output function is selected.

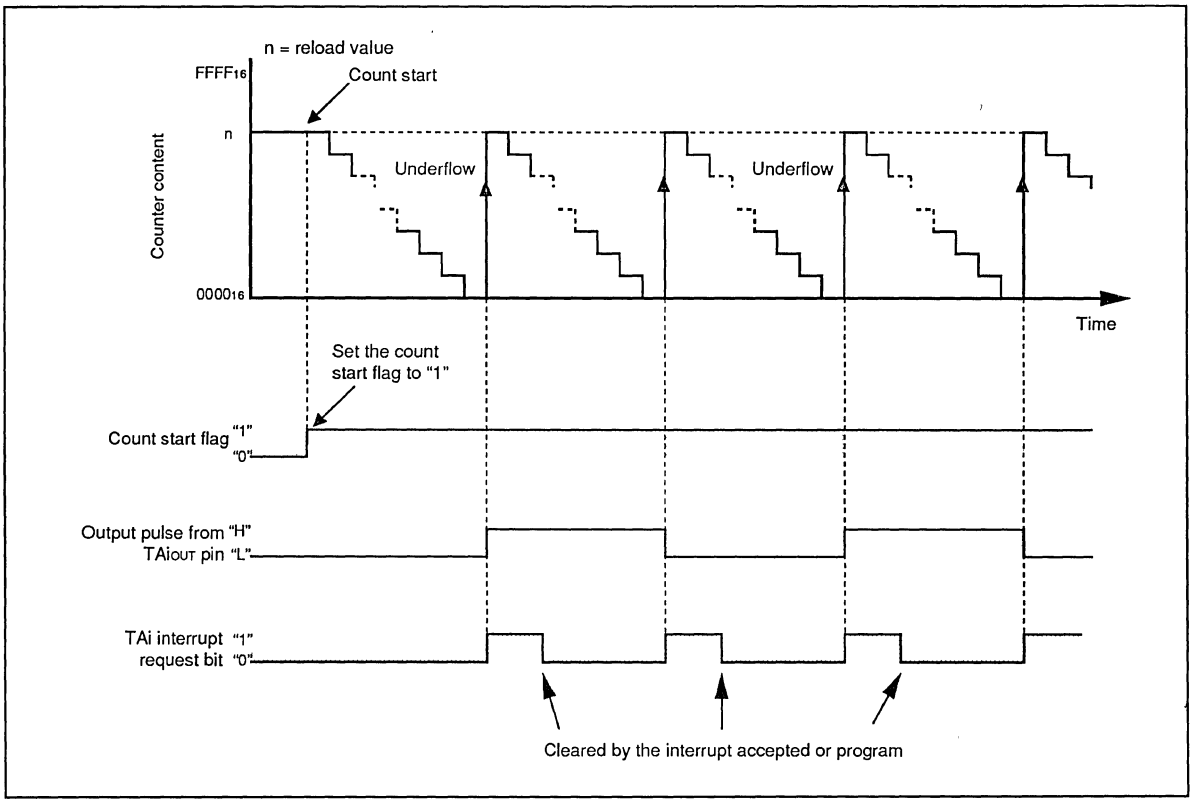


Fig. 2.7.12 Timer operation example when the pulse output function is selected

[Precautions when using the timer mode]

1. The value of the counter while operating can be read at any timing by reading the timer Ai register. However, if it is read at the reload timing shown in Figure 2.7.13, "FFFF$_{16}$" is returned instead of the reload value. The reload value is returned if it is read after the timer Ai register is set and before the count source is input and count started.

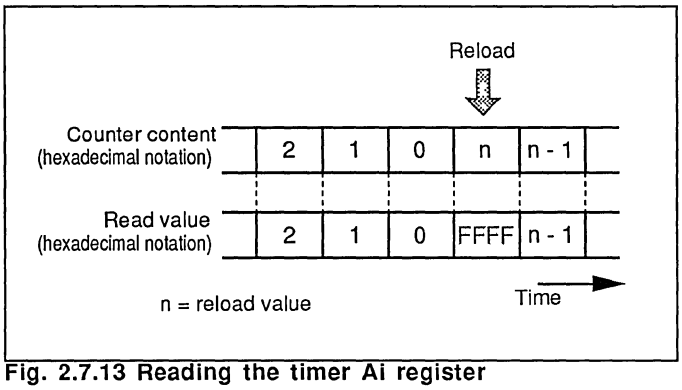


Fig. 2.7.13 Reading the timer Ai register

2.7.4 Event counter mode [timer Ai mode register bits 1, 0 = "01"]

The event counter mode is selected by setting the timer Ai mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, bit 5 of the timer Ai mode register must be set to "0". Figure 2.7.14 shows the structure of the timer Ai mode register in event counter mode.

In event counter mode, the external clock input to the TAi_{IN} pin is counted. The counter can be either an increment counter or a decrement counter according to the up-down flag or the input level to the TAi_{OUT} pin. Figure 2.7.15 shows the structure of the up-down flag register.

In increment counter, an interrupt request occurs when the counter overflows (the content of the counter reaches $FFFF_{16} \rightarrow$ reload value "n"). In decrement counter, an interrupt request occurs when the counter underflows (the content of the counter reaches $0000_{16} \rightarrow$ reload value "n"). The timer dividing ratio is expressed as follows:

●Increment counter **Timer dividing ratio=1/(n+1)**

●Decrement counter **Timer dividing ratio=1/($FFFF_{16}$–n+1)**

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

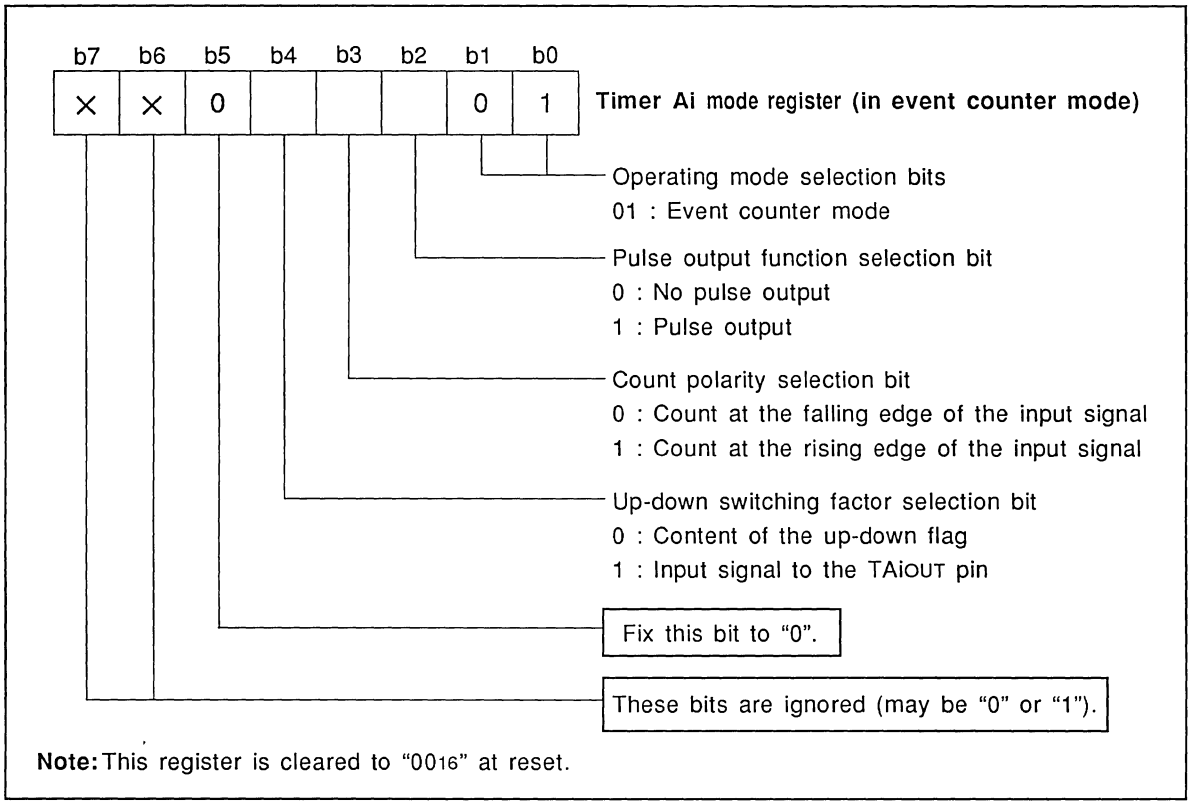
In addition, in this mode, the pulse output function and two-phase pulse signal processing function can be selected by program. Two-phase pulse signal output processing function is available only with timers A2, A3, and A4.

●Pulse output function

A pulse that changes phase is output from the TAi_{OUT} pin each time the counter underflows (decrement counter) or overflows (increment counter).

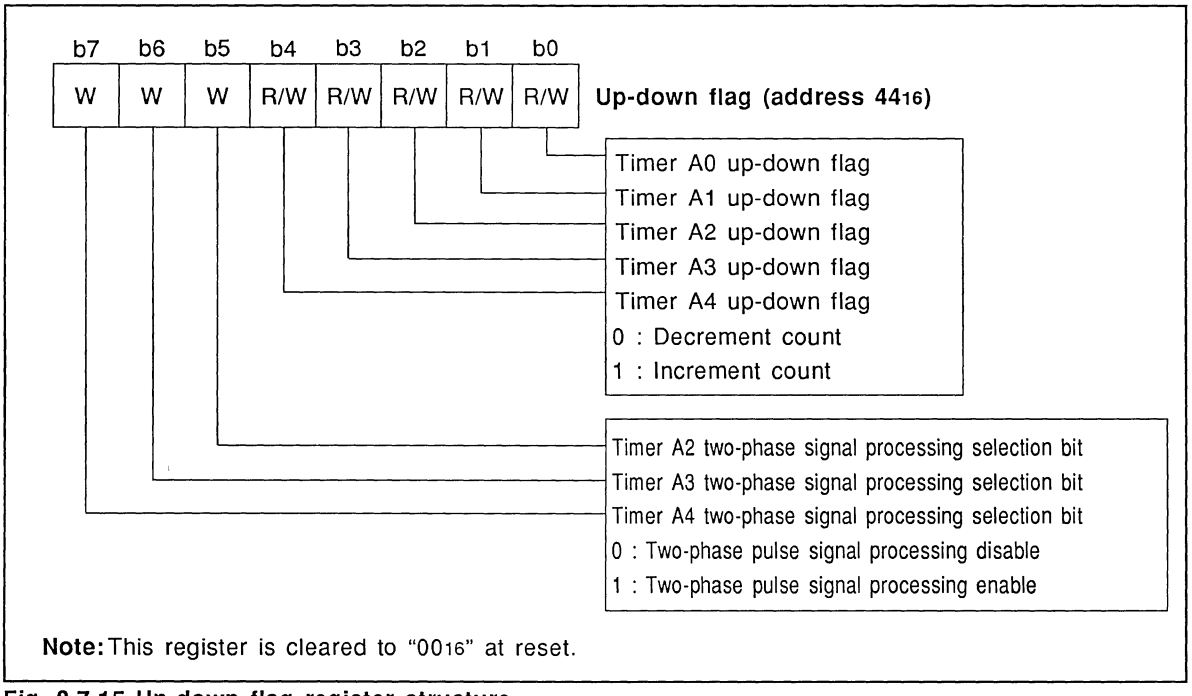
●Two-phase pulse signal processing function (timers A2 to A4)

Whether to increment or decrement the counter is selected by using two signals with phase shifted 90 degrees.



Note: This register is cleared to "00$_{16}$" at reset.

Fig. 2.7.14 Timer Ai mode register structure in event counter mode



Note: This register is cleared to "00$_{16}$" at reset.

Fig. 2.7.15 Up-down flag register structure

(1) Event counter mode operation

First, select the operating mode, count polarity, pulse output function, and up-down switching factor with the timer Ai mode register. Next, write a value "n" ("n" = 0000_{16} to $FFFF_{16}$) in the timer Ai register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register. Also set the port direction register bit corresponding to the TAi_{IN} pin to "0" (input mode). When the count start flag is set to "1" (count enabled), the external clock input to the TAi_{IN} pin is input to the counter. The counter operates at the falling edge (when the count polarity selection bit is "0") or rising edge (when the count polarity selection bit is "1") of the input clock.

Whether to increment or decrement the counter can be selected with the up-down flag or the level of the input signal to the TAi_{OUT} pin. The content of the up-down flag is used if the up-down switching factor selection bit is "0", and the level of the input signal to the TAi_{OUT} pin is used if it is "1".

Figure 2.7.16 shows the setting example of the event counter mode related registers.

●When using the content of the up-down flag

The counter is decremented when the corresponding bit to the up-down flag is set to "0", and incremented when it is set to "1".

●When using the input signal to TAi_{OUT} pin

The counter is decremented when the input level to the TAi_{OUT} pin is "L", and incremented when it is "H". The TAi_{OUT} pin is shared with port pins. Therefore, the direction register of the corresponding port pin must be set to "0" (input mode) when the input level of the TAi_{OUT} pin is used to control increment/decrement.

The pulse output function described later cannot be used when the input level to the TAi_{OUT} pin is used to control increment/decrement.

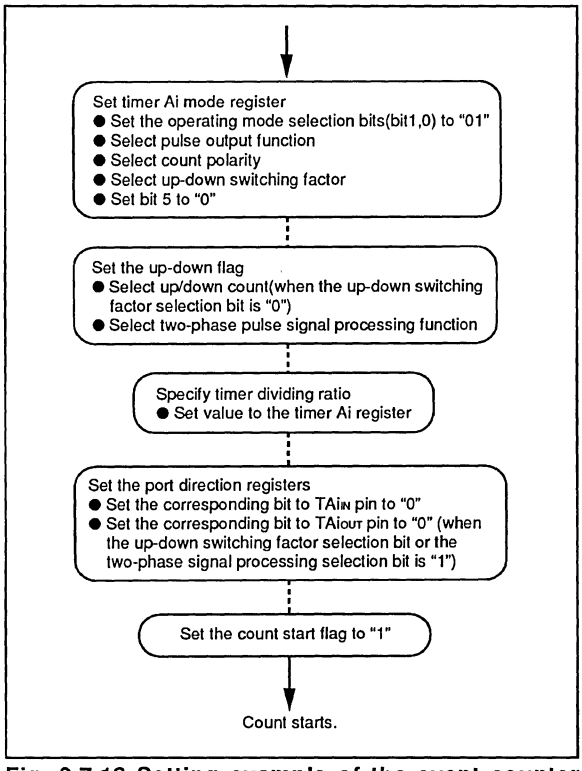


Fig. 2.7.16 Setting example of the event counter mode related registers

The count direction can be changed while counting. The count direction changes when the next effective edge of the count source is input after the content of the up-down flag changes if the up-down flag is used for up-down switching factor, and after the input level to the TAiOUT pin changes if the input signal to the TAiOUT pin is used for up-down switching factor.
Count operation continues and the counter underflows (decrement count) or overflows (increment count) at which time an interrupt request occurs and an interrupt request bit is set to "1". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.
The content of the counter can be read at any time by reading the content of timer Ai register, but the content of the reload register cannot be read. In order to change the timer dividing ratio while the timer is operating, a 16-bit update value must be written simultaneously in the timer Ai register. This value is stored in the reload register, and loaded into the counter next time the counter underflows after writing to timer Ai register.
Figure 2.7.17 shows the event counter mode operation diagram (without pulse output or two-phase pulse signal processing function).

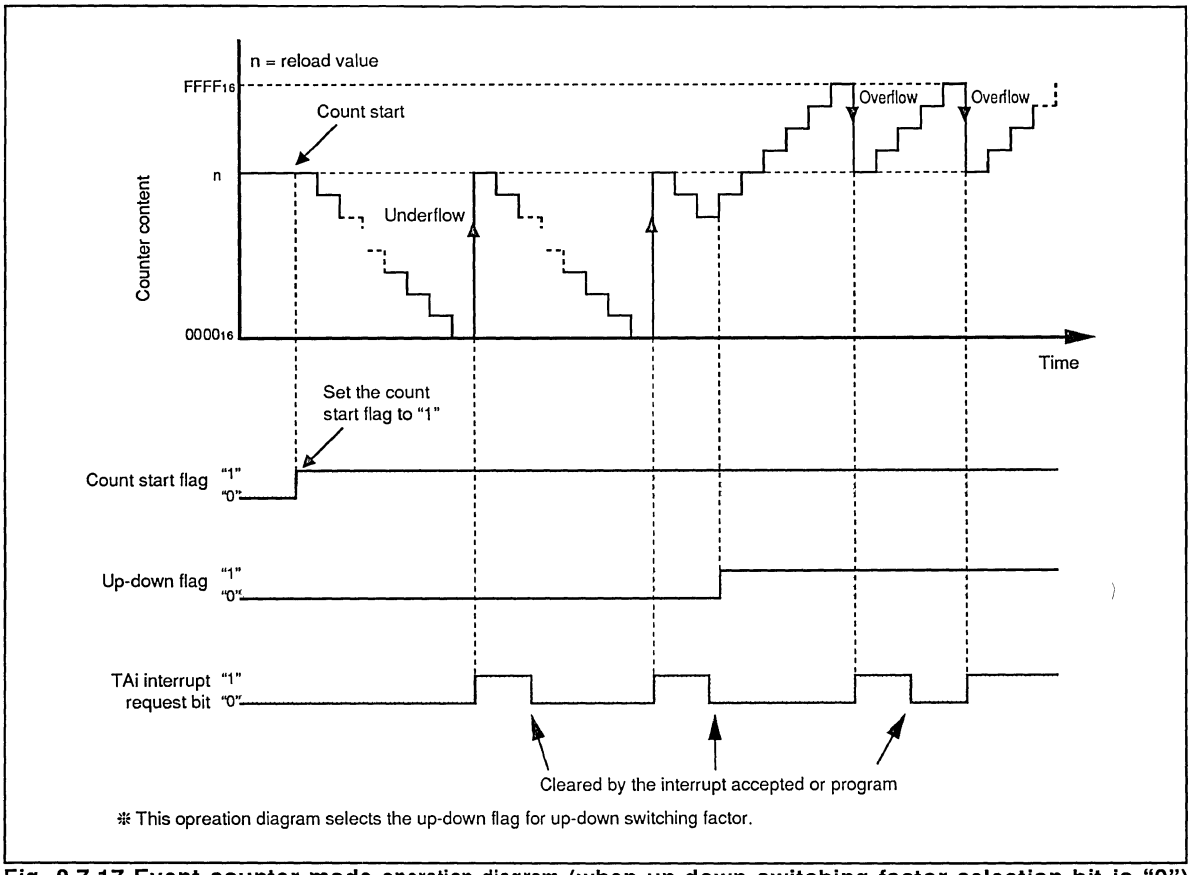


Fig. 2.7.17 Event counter mode operation diagram (when up-down switching factor selection bit is "0")

[Selection Function]
Pulse output function and two-phase pulse signal processing function can be selected by program. However, only timers A2, A3, and A4 can use the two-phase pulse signal processing function. The pulse output function and the two-phase pulse signal processing function both use the TAiOUT pin. Therefore, only one of these functions can be used at any one time.

●Pulse output function
Pulse output function is selected when the pulse output function selection bit is set to "1". When this function is selected, a signal that changes phase each time the content of the counter underflows (decrement count) or overflows (increment count) is output from the TAiOUT pin.
The TAiOUT pin is shared with ports P5 and P6. When the pulse output function is selected, the corresponding port is forced to output mode and functions as the TAiOUT pin regardless of the content of the port direction register. It can be used as a programmable I/O port once the pulse output function selection bit is set to "0".
"L" level is output from the TAiOUT pin while the count start flag is "0" and count disabled. Therefore, the initial level of the output pulse is always "L".

●Two-phase pulse signal processing function
Two-phase pulse signal processing function is selected for timers A2 to A4 when the two-phase signal processing function selection bit is set to "1". When this function is selected, set the pulse output function selection bit to "0", the count polarity selection bit to "0", and the up-down selection bit to "1". Figure 2.7.18 shows the timer Aj mode register (j=2 to 4) when two-phase pulse signal processing function is selected.
The TAjOUT pin is used in input mode and the corresponding port direction register bit must be set to "0".

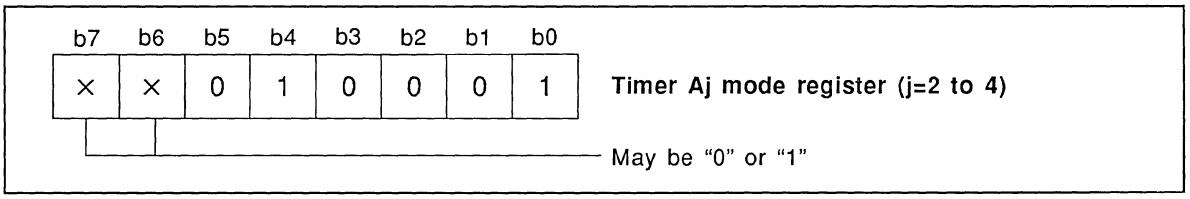


Fig. 2.7.18 Timer Aj mode register (j=2 to 4) when two-phase pulse signal processing function is selected

A timer with two-phase pulse signal processing function selected controls the counter by a two-phase pulse with phase shifted by 90 degrees. There are two types of two-phase pulse signal processing operation; one for timers A2 and A3 and another for timer A4 (quadruple processing).

<Timers A2 and A3>

The counter is incremented when the rising edge is input to the TAkIN (k=2, 3) pin and decremented when the falling edge is input to the TAkIN pin after the level of the TAkOUT pin changes from "L" to "H". (See Figure 2.7.19)

<Timer A4>

The counter is incremented at every rising and falling edge of the TA4OUT and the TA4IN pins when a phase related pulse with the rising edge input to the TA4IN pin is input after the level of the TA4OUT pin changes from "L" to "H".

The counter is decremented at every rising and falling edge of the TA4OUT and the TA4IN pins when a phase related pulse with the falling edge input to the TA4OUT pin is input after the level of the TA4IN pin changes from "H" to "L". (See Fig. 2.7.20)

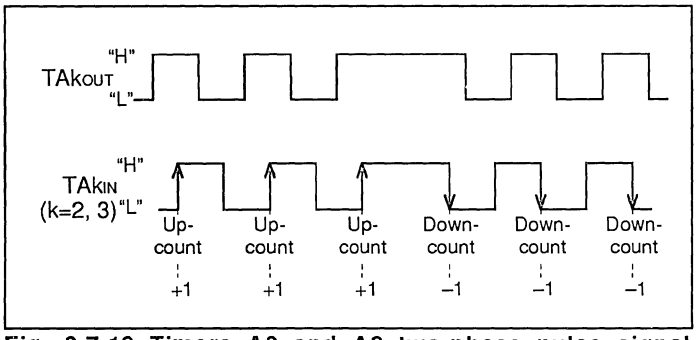


Fig. 2.7.19 Timers A2 and A3 two-phase pulse signal processing operation

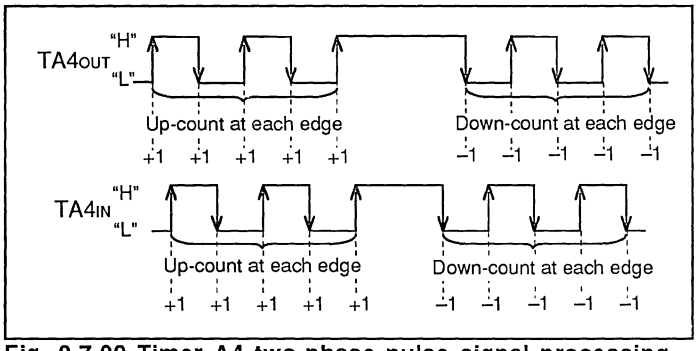


Fig. 2.7.20 Timer A4 two-phase pulse signal processing operation (quadruple processing)

[Precautions when using event counter mode]

1. The value of the counter while operating can be read at any timing by reading the timer Ai register. However, if it is read at the reload timing shown in Figure 2.7.21, "FFFF$_{16}$" is returned at underflow and "0000$_{16}$" is returned at overflow instead of the reload value. The reload value is returned if it is read after the timer Ai register is set and before the count source is input and count started.

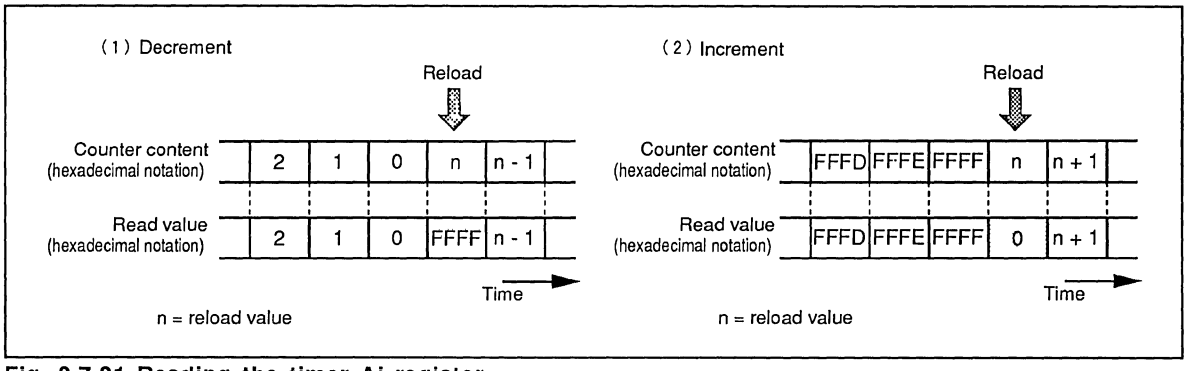


Fig. 2.7.21 Reading the timer Ai register

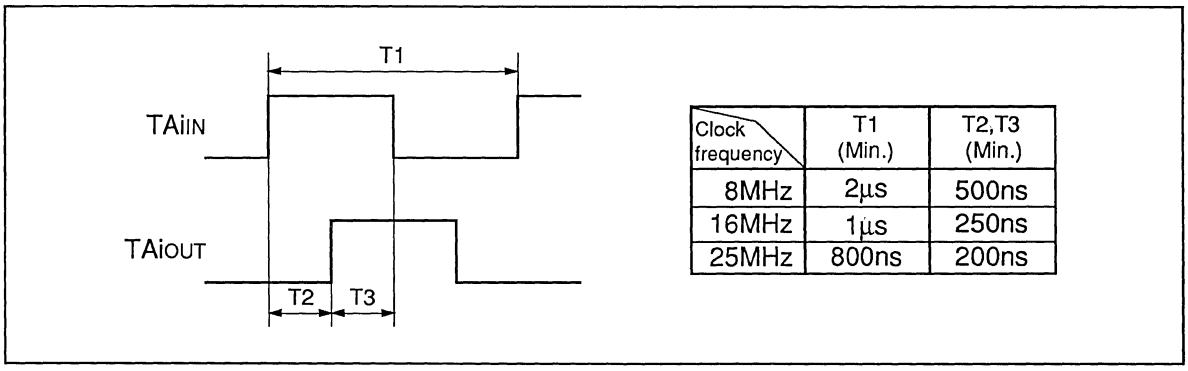
2. All of the following functions uses the TAi$_{OUT}$ pin. Only one of these functions can be selected for each timer at any one time.

   ●Count control using input signal to TAi$_{OUT}$ pin

   ●Pulse output function

   ●Two-phase pulse signal processing function (timers A2 to A4)

3. The phase difference of the two-phase pulse used for two-phase pulse processing function (input clocks to TAi$_{OUT}$ and TAi$_{IN}$ pins) must be the characteristics shown in Figure 2.7.22.



Clock frequency	T1 (Min.)	T2,T3 (Min.)
8MHz	2μs	500ns
16MHz	1μs	250ns
25MHz	800ns	200ns

Fig. 2.7.22 The characteristics of the two-phase pulse signal

2.7.5 One-shot pulse mode [timer Ai mode register bits 1, 0 = "10"]

The one-shot pulse mode is selected by setting the timer Ai mode register bit 1 to "1" and bit 0 to "0". When this mode is selected, the timer Ai mode register bit 5 must be set to "0" and bit 2 must be set to "1". Figure 2.7.23 shows the structure of the timer Ai mode register in one-shot pulse mode.

In one-shot pulse mode, "H" level is output for an arbitrary interval from the TAi_{OUT} pin after a trigger. The trigger can be either an internal trigger generated by writing "1" into the one-shot start flag shown in Figure 2.7.24 or an external trigger generated by the effective edge of the input signal to the TAi_{IN} pin. Counting starts with a trigger and at the same time, "H" level is output from the TAi_{OUT} pin. The content of the counter is decremented by 1 each time a count source is input. When the content of the counter reaches "0001_{16}" → reload value "n", the output level from the TAi_{OUT} pin changes to "L" and counting stops. At this point, an interrupt request is occurs.

The width of the output pulse ("H" level width) can be expressed as follows:

Output pulse width = n/f_i [s]

f_i: Selected count source frequency

n: Value set in counter

(0000_{16} to $FFFF_{16}$ during count halted)

(0001_{16} to $FFFF_{16}$ during count operating)

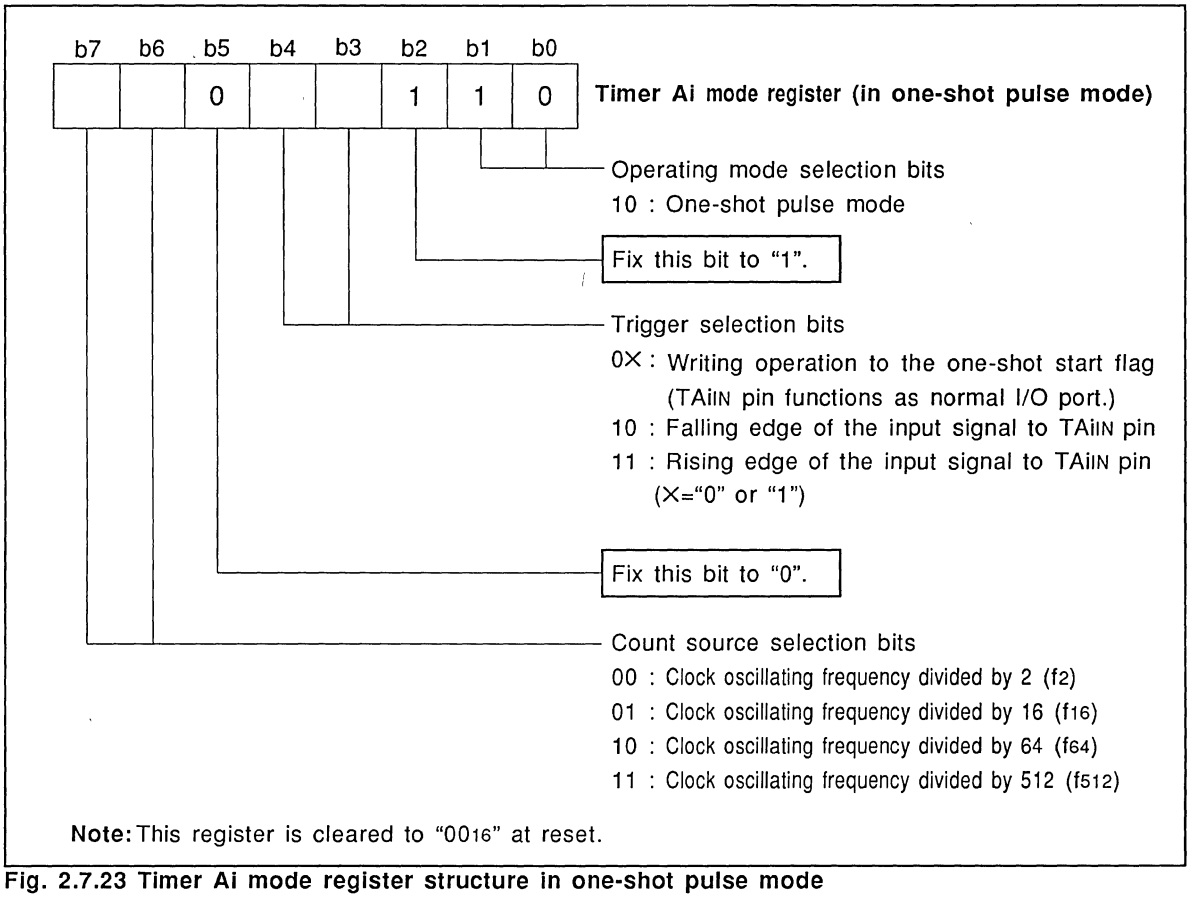


Fig. 2.7.23 Timer Ai mode register structure in one-shot pulse mode

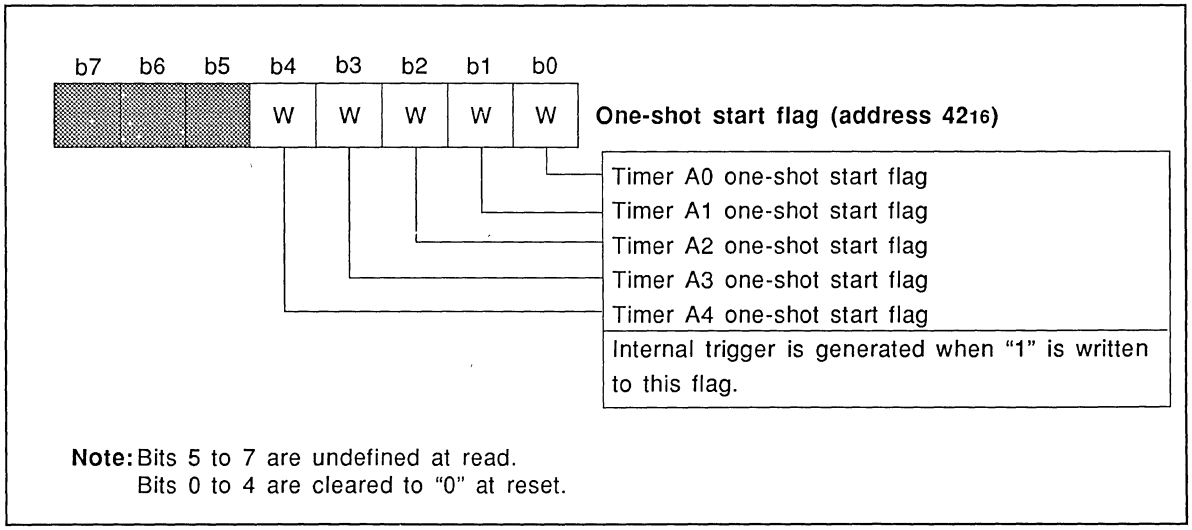


Fig. 2.7.24 One-shot start flag register structure

(1) One-shot pulse mode operation

First select the operating mode, trigger occurrence factor, and count source with the timer Ai mode register. The TAiOUT pin starts to output "L" level when one-shot pulse mode is selected with the operating mode selection bits. Next, write a value "n" ("n"= 0000_{16} to $FFFF_{16}$) in the timer Ai register to set the output pulse width. At this point, "n" is stored in the counter and the reload register. Count is enabled when the count start flag is set to "1", and then count starts when a trigger occurs.

If bit 4 of the timer Ai mode register is "0", an internal trigger is generated by setting the corresponding bit to each timer in the one-shot start flag to "1". If bit 4 is "1" and bit 3 is "0", a trigger is generated at the falling edge of the TAiIN pin input signal and if bit 4 is "1" and bit 3 is "1", a trigger is generated at the rising edge of the TAiIN pin input signal. When using an external trigger, the corresponding port direction register bit to the TAiIN pin must be set to "0" (input mode).

Figure 2.7.25 shows the setting example of the one-shot pulse mode related registers.

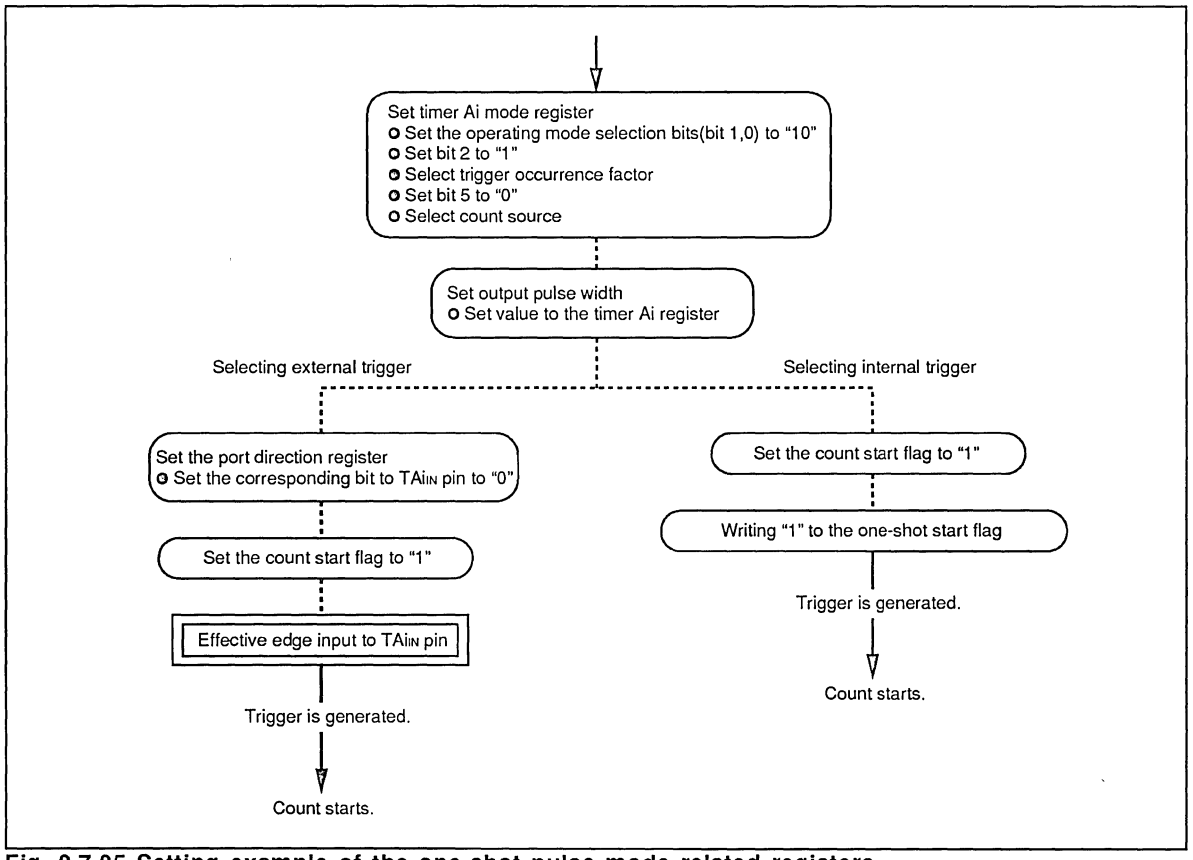


Fig. 2.7.25 Setting example of the one-shot pulse mode related registers

When triggered, counting starts and "H" level is output from the TAiOUT pin. (However, if the timer Ai register contains "0000$_{16}$", the TAiOUT pin level remains at "L" and counting does not start.) The counter is decremented by 1 each time a count source is input. When the content of the counter reaches 0001$_{16}$ → "n" , the TAiOUT pin level changes to "L" and counting stops. The "H" level width of the output pulse from the TAiOUT pin is (count source cycle) x n.

An interrupt request occurs and the interrupt request bit is set to "1" when the TAiOUT pin level changes from "H" to "L". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.

If a value is written in the timer Ai register while the counter is operating ("H" level is output from the TAiOUT pin), this value is only set in the reload register. It is loaded into the counter at the next reload timing. Values other than 0000$_{16}$ can be written while the counter is operating.

If the value of timer Ai register is read, the result is undefined.

Figure 2.7.26 shows the one-shot pulse mode operation diagram (when external trigger is selected).

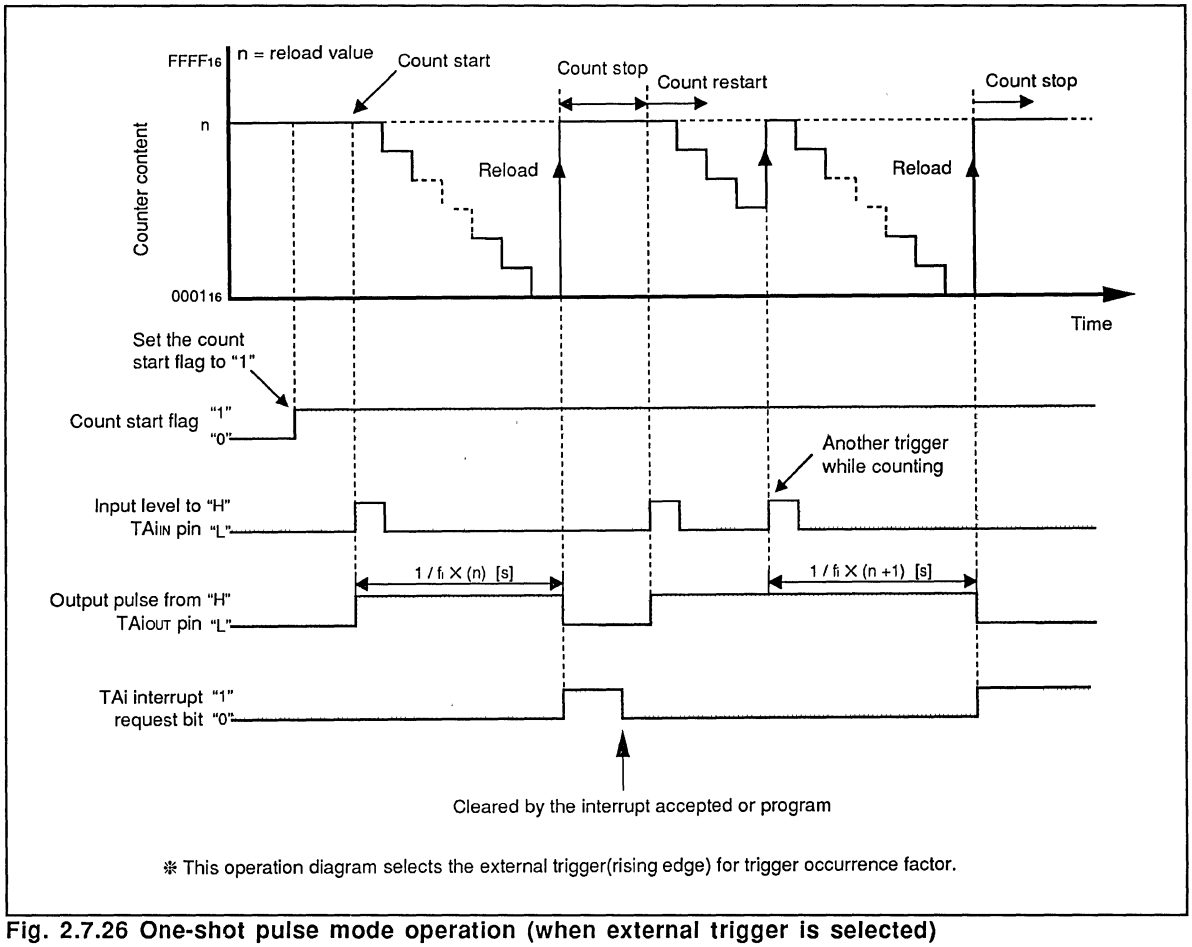


Fig. 2.7.26 One-shot pulse mode operation (when external trigger is selected)

The counter stops after completing one cycle of output, and repeats the same operation when the next trigger occurs. When the count start flag is "0" (count disabled), "L" level is output from the TAi_{OUT} pin. Therefore, an arbitrary pulse width can be generated by setting a value in the timer Ai register before setting the count start flag to "1".

If another trigger is generated while counting, the content of the reload register is transferred to the counter and decrement continues from that value. In this case, the TAi_{OUT} pin level becomes "L" at n+1 count after the trigger. The content of the reload register is transferred to the counter only when a trigger occurs while counting. At least one cycle of the timer count source should elapse before retriggering.

[Precautions when using one-shot pulse mode]

1. If the count start flag is set to "0" while counting, counting stops and the output level of the TAi_{OUT} pin becomes "L". At the same time an interrupt request occurs and the interrupt request bit is set to "1".

2. Values between 0001_{16} and $FFFF_{16}$ can be written in the timer Ai register while the counter is operating ("H" level is output from the TAi_{OUT} pin).

3. When external trigger is selected, there may be a time lag between the time the effective edge is input to TAi_{IN} pin and the time the trigger is generated. This is because the input signal to the TAi_{IN} pin is not synchronized with the internal operating clock.

4. When the operating mode is switched from timer mode or event counter mode to one-shot pulse mode, the interrupt request bit in timer Ai interrupt control register is set to "1". Clear the interrupt request bit to "0" after switching modes when using timer Ai interrupt or the interrupt request bit in one-shot pulse mode.

2.7.6 Pulse width modulation (PWM) mode [timer Ai mode register bits 1, 0 = "11"]

Pulse width modulation mode (hereafter referred to as PWM) is selected and timer Ai functions as a pulse width modulator when timer Ai mode register bits 1, 0 are set to "11". When this mode is selected, bit 2 of the timer Ai mode register must be set to "1". Figure 2.7.27 shows the structure of the timer Ai mode register in PWM mode.

In PWM mode, an arbitrary pulse width signal is output continuously from the TAi$_{OUT}$ pin. A 16-bit PWM mode or an 8-bit PWM mode can be selected by program.

●16-bit PWM mode

The counter functions as a 16-bit pulse width modulator.

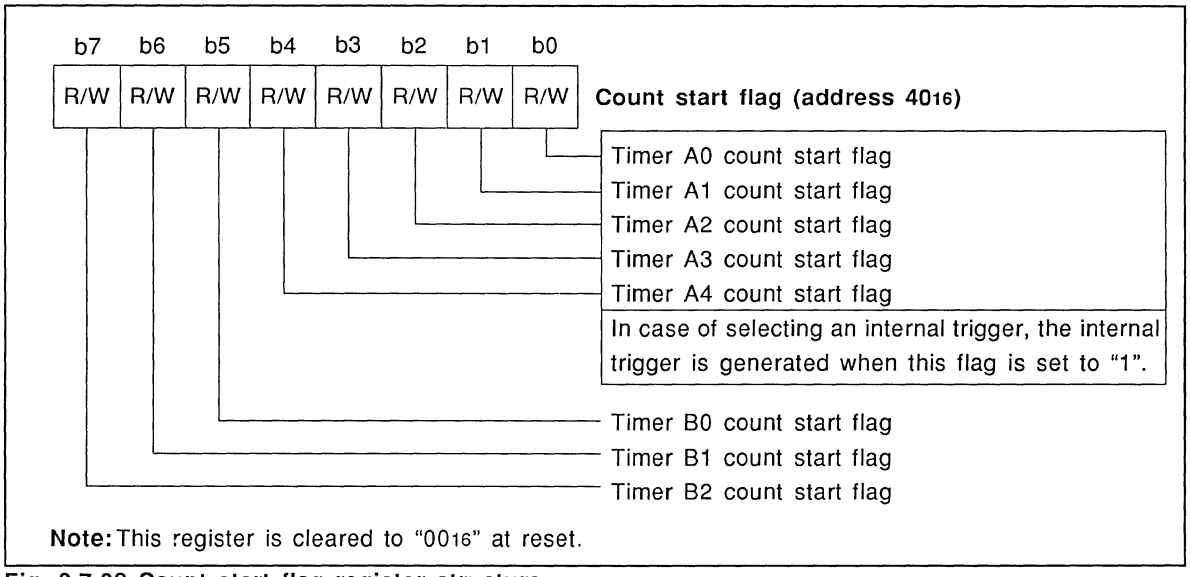
●8-bit PWM mode

The reload register and the counter are both divided into 8-bit halves. The high-order 8 bits of the counter function as a pulse width modulator and the low-order 8 bits function as a prescaler.

The trigger is either an internal trigger generated when the count start flag shown in Figure 2.7.28 is set to "1" or an external trigger generated when the effective edge signal is input to the TAi$_{IN}$ pin with the count start flag set to "1". When a trigger occurs, the pulse width modulator starts and pulses are output from the TAi$_{OUT}$ pin.

An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the output pulse changes from "H" to "L",.

When the pulse width modulator starts operation with a trigger, the next trigger is not accepted (pulses are output continuously).



Note: This register is cleared to "00₁₆" at reset.

Fig. 2.7.28 Count start flag register structure

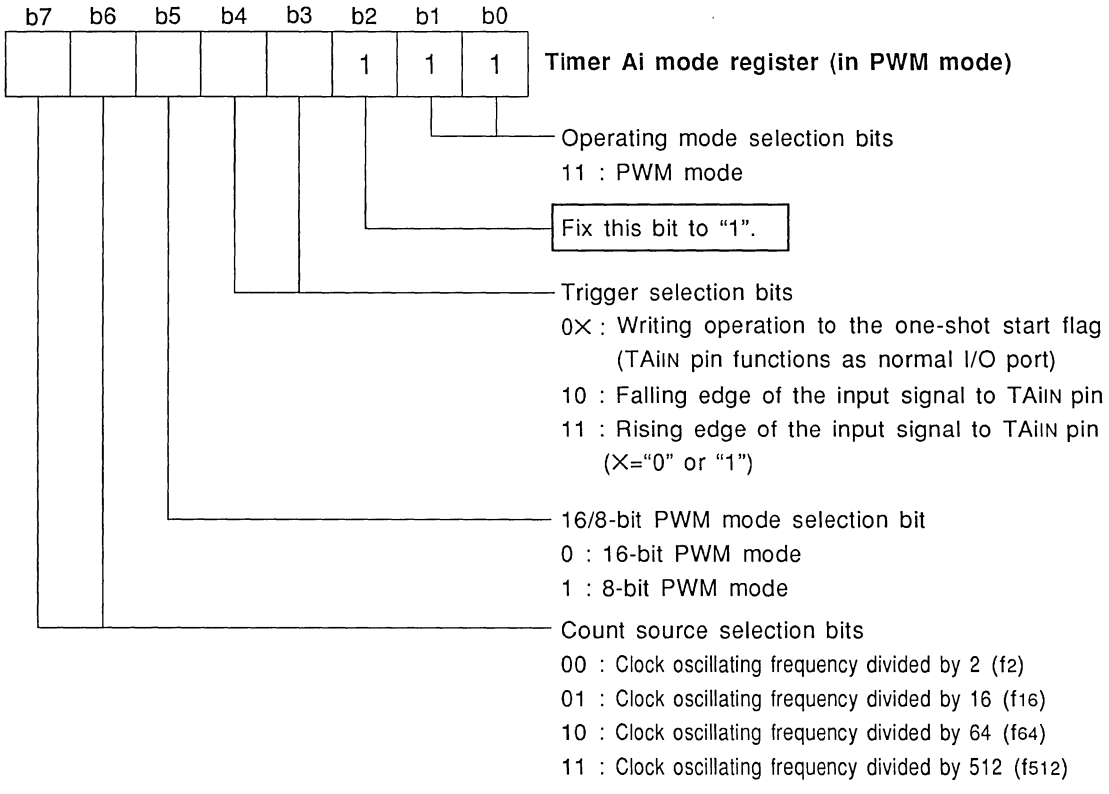


Fig. 2.7.27 Timer Ai mode register structure in PWM mode

(1) PWM mode operation

First, select the operating mode, trigger occurrence factor, 16/8-bit PWM mode, and count source with the timer Ai mode register. The TAiOUT pin outputs "L" level when PWM mode is selected with the operating mode selection bits. Next, write an arbitrary value "n" in the timer Ai register to set the output pulse width. At this point, "n" is set in the counter and the reload register. Then when a trigger is generated, the pulse width modulator starts and pulses are output from the TAiOUT pin. When the timer Ai mode register bit 4 is "0", an internal trigger occurs each time the corresponding bit to each timer in the count start flag is set to "1". If bit 4 is "1" and bit 3 is "0", the trigger occurs at the falling edge of the input signal to the TAiIN pin. If bit 4 is "1" and bit 3 is "1", the trigger occurs at the rising edge of the input signal. When using an external trigger, set the corresponding port direction register to the TAiIN pin to "0" (input mode).

A pulse width modulator continuously outputs pulses when it starts operation. An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the output signal changes from "H" to "L". Interrupts must be enabled before they can be use. See "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until an interrupt request is accepted or it is cleared by program.

If a value is written in the timer Ai register while the counter is operating, the value is set only in the reload register. It is loaded in the counter next time the level of the output pulse changes from "L" to "H".

If the value of the timer Ai register is read, the result is undefined.

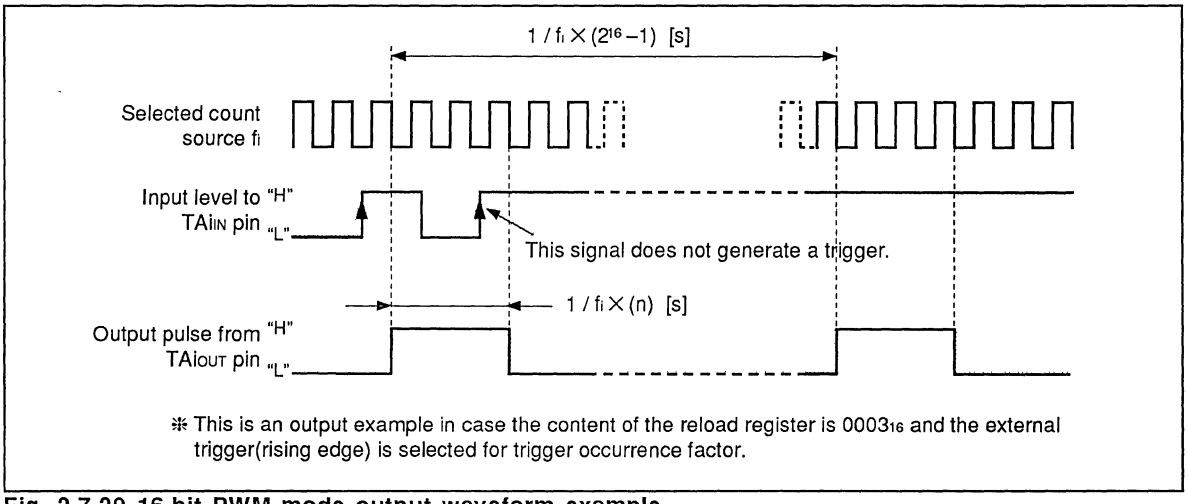
The 16-bit PWM mode and 8-bit PWM mode operations are described below.

[16-bit PWM mode]
16-bit PWM mode is selected when the 16/8-bit PWM mode selection bit is set to "0". In this mode, the cycle and width of the output pulse from the TAiOUT pin can be expressed as follows:

Output pulse cycle = (1/fi) × (2^{16}–1) [s]
Output pulse "H" width = (1/fi) × n [s]

fi: Frequency of selected count source [Hz]
n: Value set in counter
(value between 0000_{16} and $FFFE_{16}$)

Figure 2.7.29 shows an example of an output waveform in 16-bit PWM mode. An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the output pulse signal changes from "H" to "L".
The "H" width of the output pulse can be changed while the pulse width modulator is operating (outputting the pulse signal) by writing a value in the timer Ai register. This value is written in the reload register and loaded into the counter next time the level of the output pulse changes from "L" to "H". Therefore, the pulse width changes from the pulse following the pulse being output when the value was written.



※ This is an output example in case the content of the reload register is 0003_{16} and the external trigger(rising edge) is selected for trigger occurrence factor.

Fig. 2.7.29 16-bit PWM mode output waveform example

[8-bit PWM mode]
8-bit PWM mode is selected when the 16/8-bit PWM mode selection bit is set to "1". In this mode, the reload register and the counter are divided into 8-bit halves. The high-order 8 bits of the counter function as an 8-bit pulse width modulator and the low-order 8 bits function as a prescaler.
The prescalar counts the clock selected with the count source selection bit. The prescalar is decremented and an underflow signal is generated when its content reaches 00_{16} → "m" ("m" = value set in low-order 8 bits of the reload register). The content of the reload register is loaded into the prescalar and counting continues. The pulse width modulator (high-order 8 bits of the counter) counts the underflow signal generated by the prescalar. The cycle and width of the pulse output from the TAiOUT pin can be expressed as follows:

Output pulse cycle = (1/fi) × (m+1) × (2^8–1) [s]
Output pulse "H" width = (1/fi) × (m+1) × n [s]

fi : Frequency of selected count source [Hz]
n: Value in the high-order 8 bits of the counter
(Value between 00_{16} and FE_{16})
m: Value in the low-order 8 bits of the counter
(Value between 00_{16} and FF_{16})

Figure 2.7.30 shows an output waveform example in 8-bit PWM mode. In 8-bit PWM mode, pulse output starts at the set pulse cycle after "L" level with the width equal to the "H" level of the set pulse is output. The "H" width of the output pulse can be changed while the pulse width modulator is operating (outputting the pulse signal) by writing a value in the timer Ai register. This value is written in the reload register and loaded into the counter next time the level of the output pulse changes from "L" to "H". Therefore, the pulse width changes from the pulse following the pulse being output when the value was written.

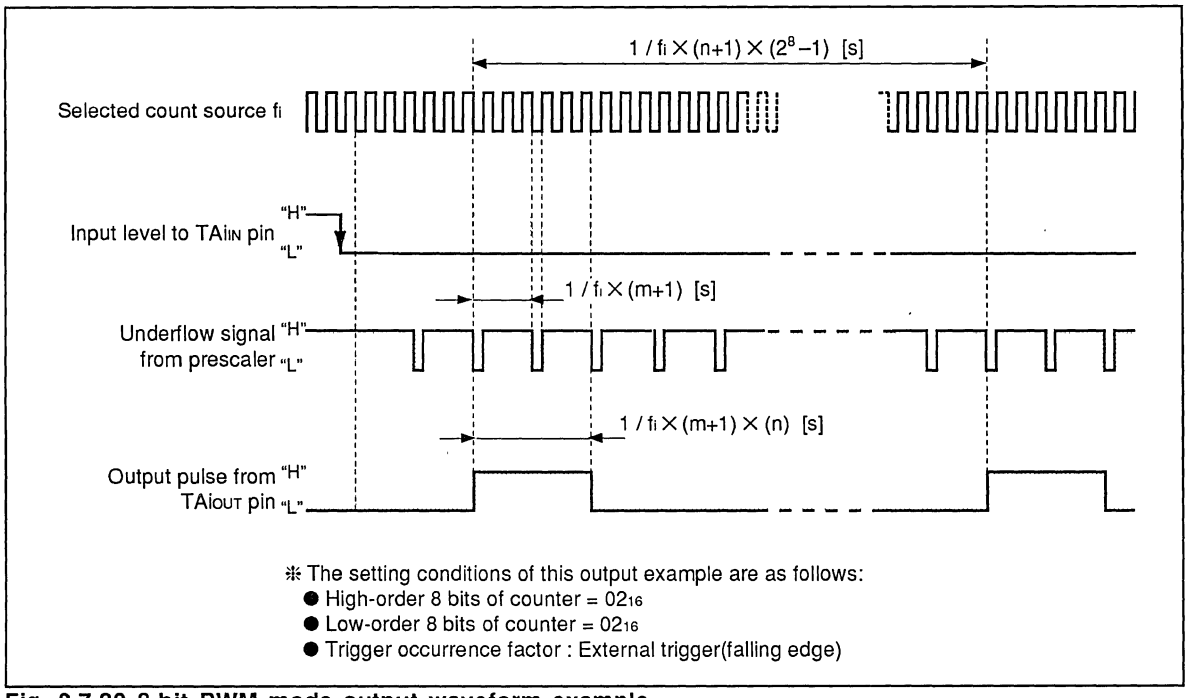


Fig. 2.7.30 8-bit PWM mode output waveform example

[Precautions when using PWM mode]

1. When the operating mode is switched from timer mode or event counter mode to PWM mode, the interrupt request bit in timer Ai interrupt control register is set to "1". Clear the interrupt request bit to "0" after switching modes when using timer Ai interrupt or the interrupt request bit in PWM mode.

2.8 Timer B

Timer B consists of three 16-bit timers (timers B0 to B2). Timers B0 to B2 operate independently and each can operate in one of three different modes.

2.8.1 Timer B description

Timer Bi (i= 0 to 2) has three operating modes as described below. These timers have identical functions. The input pins (TBi_{IN}) of these timers are shared with ports $P6_5$ to 6_7.

●Timer mode

This mode counts the selected internal clock. The counter is decremented by 1 each time a count source (selected clock) is input and a timer Bi interrupt request occurs when the content of the counter reaches 0000_{16} → reload value "n" (hereafter referred to as underflow).

●Event counter mode

This mode counts the external clock input to the TBi_{IN} pin. The counter is decremented by 1 each time a clock is input and a Timer Bi interrupt request occurs when the counter underflows.

●Pulse period/pulse width measurement mode

In this mode, the period or the pulse width of the input signal to the TBi_{IN} pin is measured.

2.8.2 Block description

Figure 2.8.1 shows the block diagram of timer Bi. It is followed by the description of the timer Bi related registers.

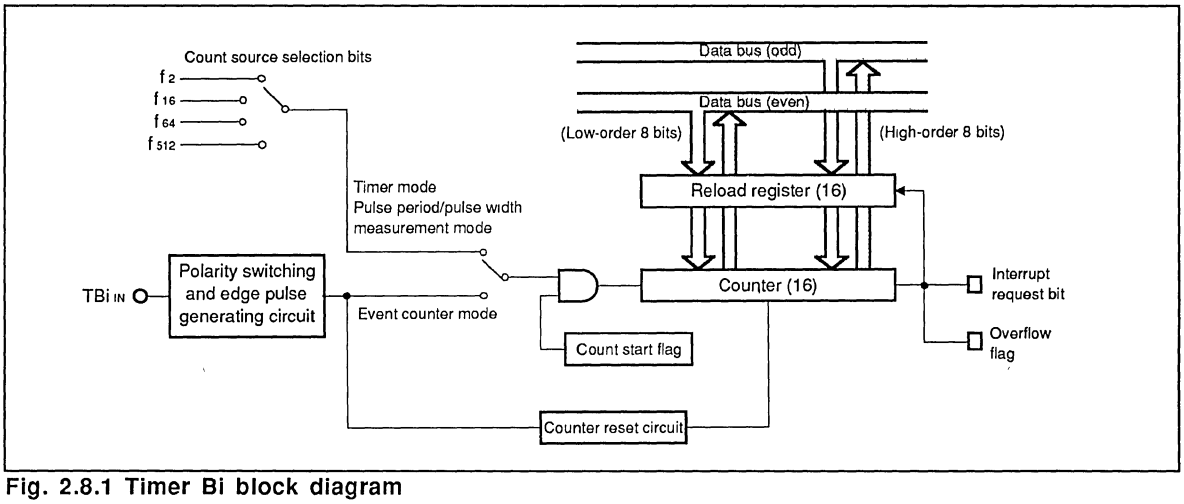


Fig. 2.8.1 Timer Bi block diagram

(1) Counter and reload register (timer Bi register)

Timer Bi counter and reload register consist of 16 bits. The counter counts the selected count source. In timer mode and event counter mode, the content of the counter is decremented by 1 each time a count source is input. The reload register is used to store the initial value of the counter. The content of the counter changes each time a count source is input to the counter, but the content of the reload register remains unchanged. The content of the reload register is reloaded into the counter when the counter underflows. A value is set in the counter and the reload register by writing a value in the timer Bi register.

In pulse period/pulse width measurement mode, the content of the counter is incremented by 1 each time a count source is input to the counter. The result of measuring the pulse signal is transferred to the reload register. Table 2.8.1 shows the memory allocation of the timer Bi register.

Table 2.8.1 Timer Bi register memory allocation

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}

In timer mode and event counter mode, the value written in the timer Bi register while the timer is not operating is stored in the counter and the reload register. The value written in the timer Bi register while the timer is operating is stored only in the reload register. In this case, the updated value is transferred to the counter during the next reload. When the timer Bi register is read, the counting value is read. When the timer Bi register is read in pulse period/pulse width measurement mode, the pulse width or the pulse period measurement result is read.

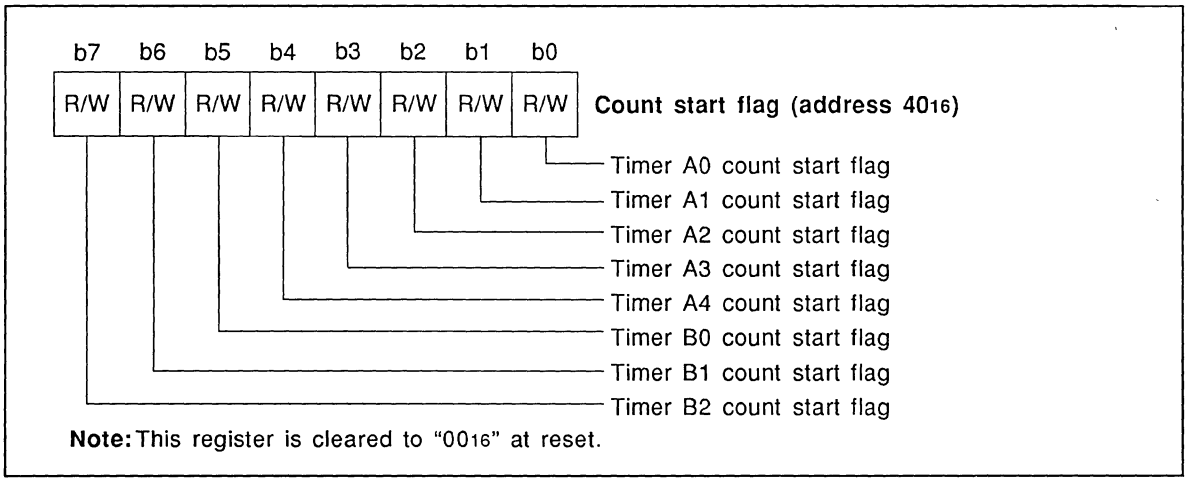
The value of the timer Bi register is undefined at reset. Therefore, the counter and the reload register must first be initialized when using timer Bi in timer or event counter mode.

(2) Count start flag

The count start flag (address 40_{16}) separately controls starting/stopping of each timer. Each bit corresponds to one of the timers.

A count source is input to the counter when this flag is "1", and disabled when it is set to "0".

Figure 2.8.2 shows the structure of the count start flag.



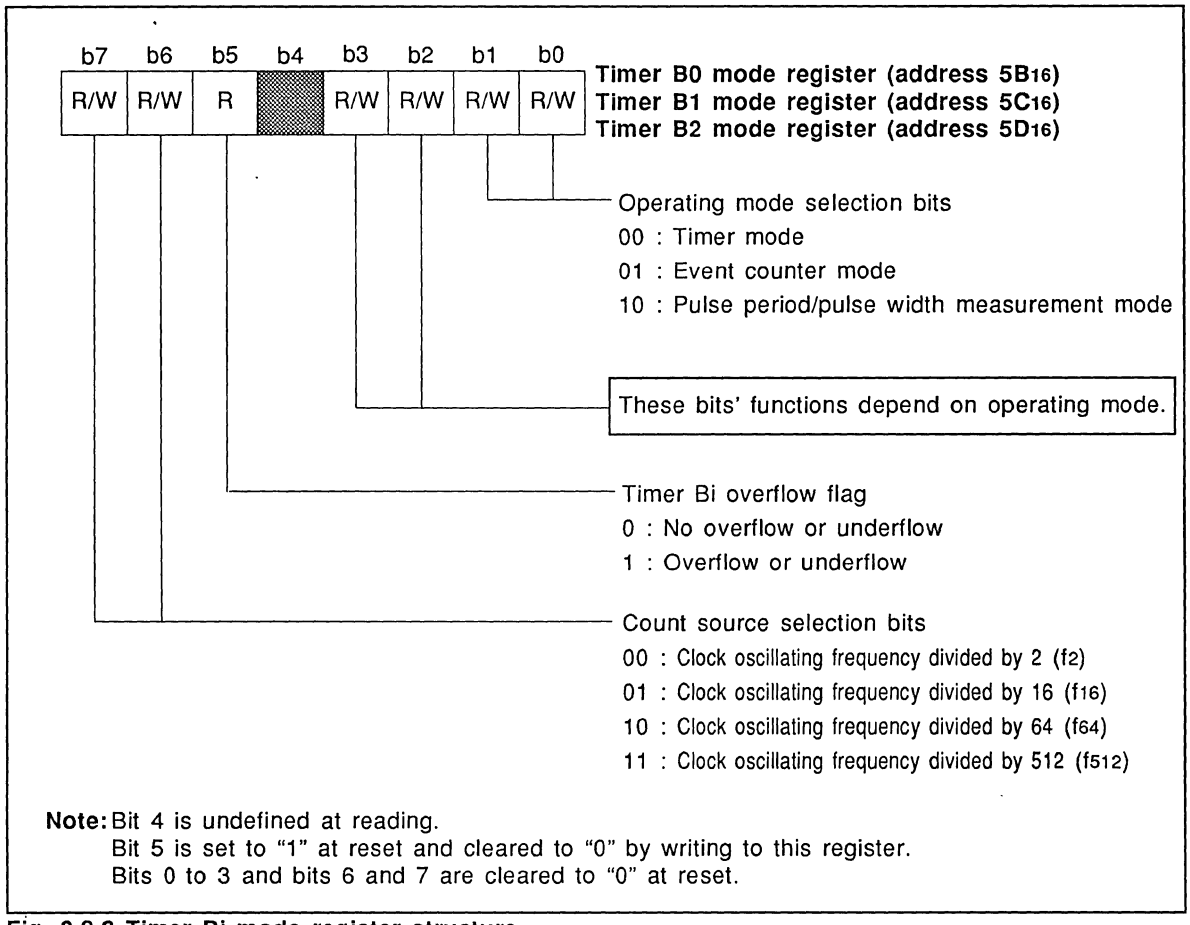
Note: This register is cleared to "00_{16}" at reset.

Fig. 2.8.2 Count start flag register structure

(3) Timer Bi mode register

The timer Bi mode register (address $5B_{16}$ to $5D_{16}$) consists of operating mode selection bits, overflow flag, and count source selection bits.

Figure 2.8.3 shows the structure of the timer Bi mode register. The operating mode selection bits, overflow flag, and count source selection bits are described below. The function of bits 2 and 3 depend on the operating mode and are described under the description of each operating mode.



Note: Bit 4 is undefined at reading.
Bit 5 is set to "1" at reset and cleared to "0" by writing to this register.
Bits 0 to 3 and bits 6 and 7 are cleared to "0" at reset.

Fig. 2.8.3 Timer Bi mode register structure

●Operating mode selection bits (bits 0 and 1)
The operating mode selection bits are used to select the timer operating mode. Table 2.8.2 shows the relationship between the operating mode selection bits and the timer operating modes.

Table 2.8.2 Relationship between operating mode selection bits and operating modes

b1	b0	Operating mode
0	0	Timer mode
0	1	Event counter mode
1	0	Pulse period/pulse width measurement mode

●Timer Bi overflow flag (bit 5)
The timer Bi overflow flag is set to "1" when the counter underflows in timer mode or event counter mode. In pulse period/pulse width measurement mode, it is set to "1" when the content of the counter reaches $FFFF_{16} \rightarrow 0000_{16}$.
This bit is set to "1" at reset and cleared to "0" when a value is written in the timer Bi mode register.

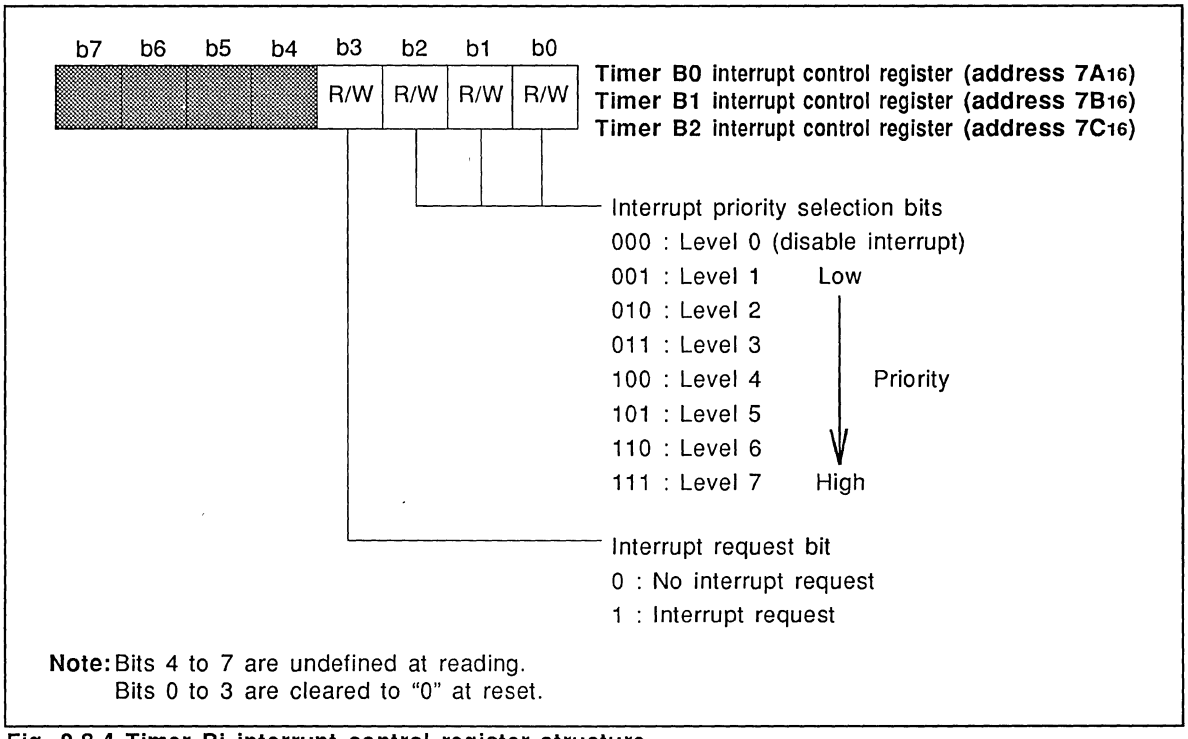
●Count source selection bits (bits 6 and 7)
The count source selection bits are used to select the count source. Table 2.8.3 shows the relationship between the count source selection bits and the timer count source.
These bits are ignored in event counter mode.

Table 2.8.3 Relationship between count source selection bits and count sources

b7	b6	Timer count source	Input clock to the counter		
			$f(X_{IN})$=8MHz	$f(X_{IN})$=16MHz	$f(X_{IN})$=25MHz
0	0	Clock oscillating frequency divided by 2 (f_2)	4MHz	8MHz	12.5MHz
0	1	Clock oscillating frequency divided by 16 (f_{16})	500kHz	1MHz	1.5625MHz
1	0	Clock oscillating frequency divided by 64 (f_{64})	125kHz	250kHz	390.625kHz
1	1	Clock oscillating frequency divided by 512 (f_{512})	15625Hz	31250Hz	48.8281kHz

(4) Timer Bi interrupt control register

The timer Bi interrupt control register (address $7A_{16}$ to $7C_{16}$) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.8.4 shows the structure of the timer Bi interrupt control register. The function of each bit is described below. Refer to section "2.6 Interrupts" for more information.



Note: Bits 4 to 7 are undefined at reading.
Bits 0 to 3 are cleared to "0" at reset.

Fig. 2.8.4 Timer Bi interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using timer Bi interrupt. When there is an interrupt request, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is greater than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable only timer Bi interrupt.

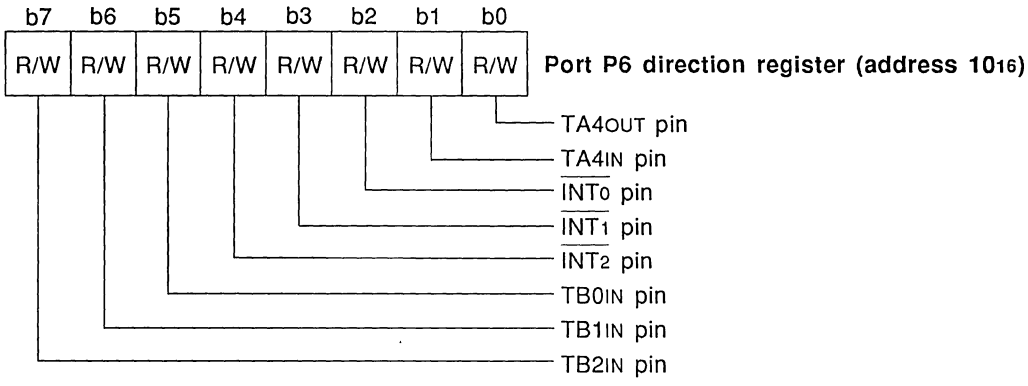
●Interrupt request bit (bit 3)

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

(5) Port P6 direction register

Timers B0 to B2 input pins are shared with port P6. When using these ports as timer input pins, the corresponding bit in the direction register must be set to "0" (input mode).

Figure 2.8.5 shows the relationship between the port P6 direction register (address 10_{16}) and the timer pins.



Note: This register is cleared to "00_{16}" at reset.

Fig. 2.8.5 Relationship between port P6 direction register and timer pins

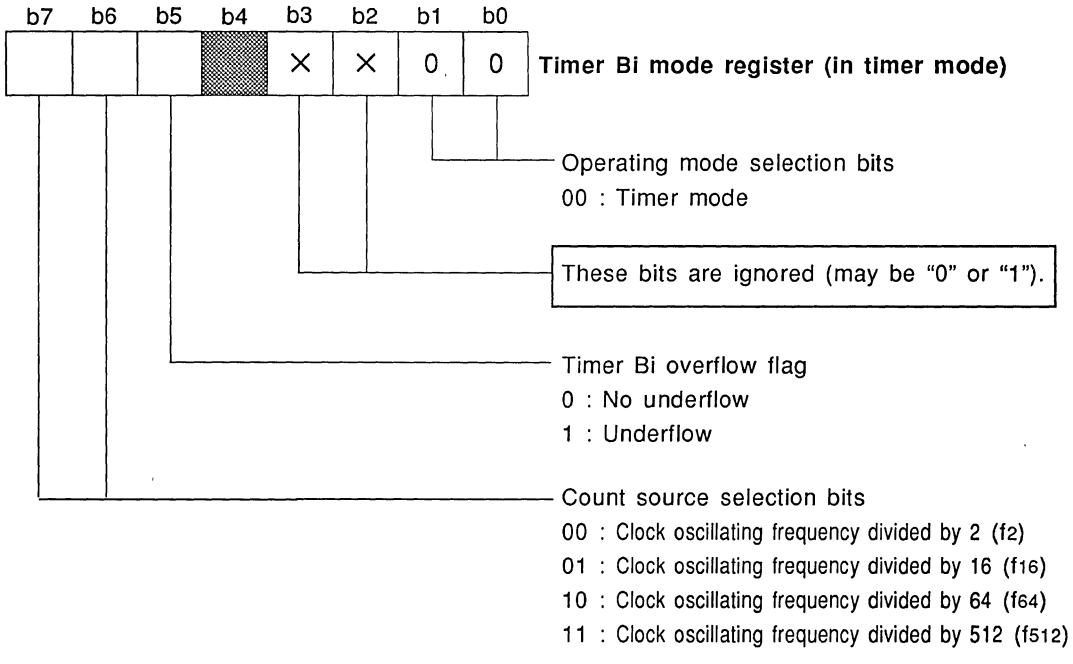
2.8.3 Timer mode [timer Bi mode register bits 1, 0 = "00"]

Timer mode is selected by setting the timer Bi mode register bits 1 and 0 to "00". Figure 2.8.6 shows the structure of the timer Bi mode register in timer mode.

In timer mode, the selected internal clock is decremented and an interrupt request occurs each time the counter underflows (the content of the counter reaches 0000_{16} → reload value "n"). The timer dividing ratio is expressed as follows:

Timer dividing ratio = 1/(n+1)

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)



Note: Bit 4 is undefined at reading.
Bit 5 is set to "1" at reset and cleared to "0" by writing to this register.
Bits 0 to 3 and bits 6 and 7 are cleared to "0" at reset.

Fig. 2.8.6 Timer Bi mode register structure in timer mode

(1) Timer mode operation

First, select the operating mode and count source with the timer Bi mode register. Next, write an value "n" ("n"=0000$_{16}$ to FFFF$_{16}$) in the timer Bi register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register.

When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and starts the count operation.

Figure 2.8.7 shows the setting example of the timer mode related registers.

The content of the counter is decremented by 1 each time the count source is input. When the counter underflows, the content of the reload register is loaded in the counter and the interrupt request bit is set to "1". At the same time, the timer Bi overflow flag is set to "1". Count operation continues in this manner. The interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, an interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.

The content of the counter can be read at any time by reading the content of the timer Bi register, but the content of the reload register can not be read. In order to change the timer dividing ratio while the timer is operating, a 16-bit update value must be written simultaneously in the timer Bi register. This value is stored in the reload register and loaded in the counter next time the counter underflows after writing to timer Bi register. Figure 2.8.8 shows the timer mode operation diagram.

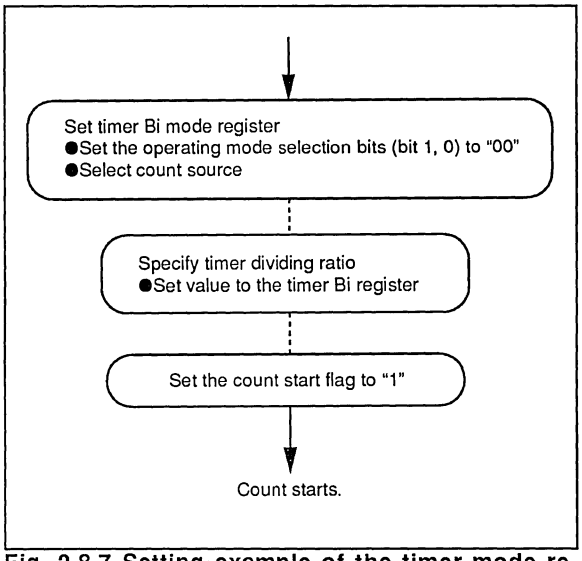


Fig. 2.8.7 Setting example of the timer mode related registers

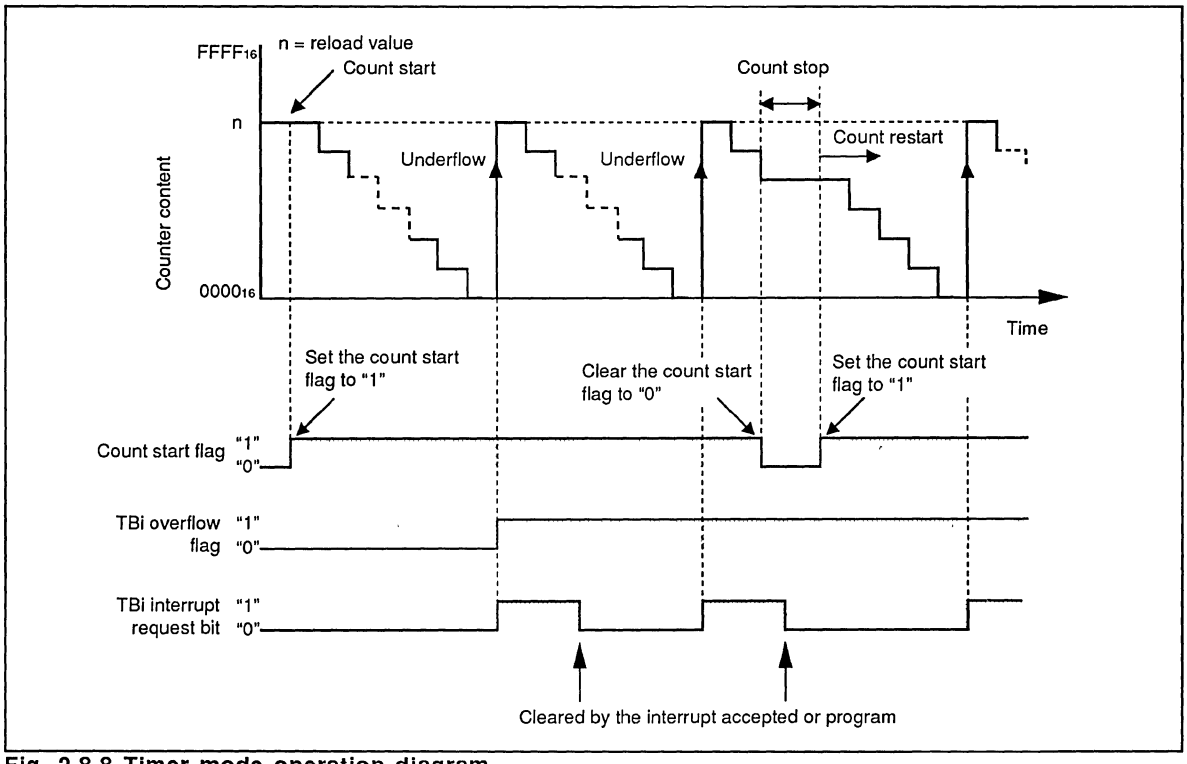


Fig. 2.8.8 Timer mode operation diagram

2.8.4 Event counter mode [timer Bi mode register bits 1, 0 = "01"]

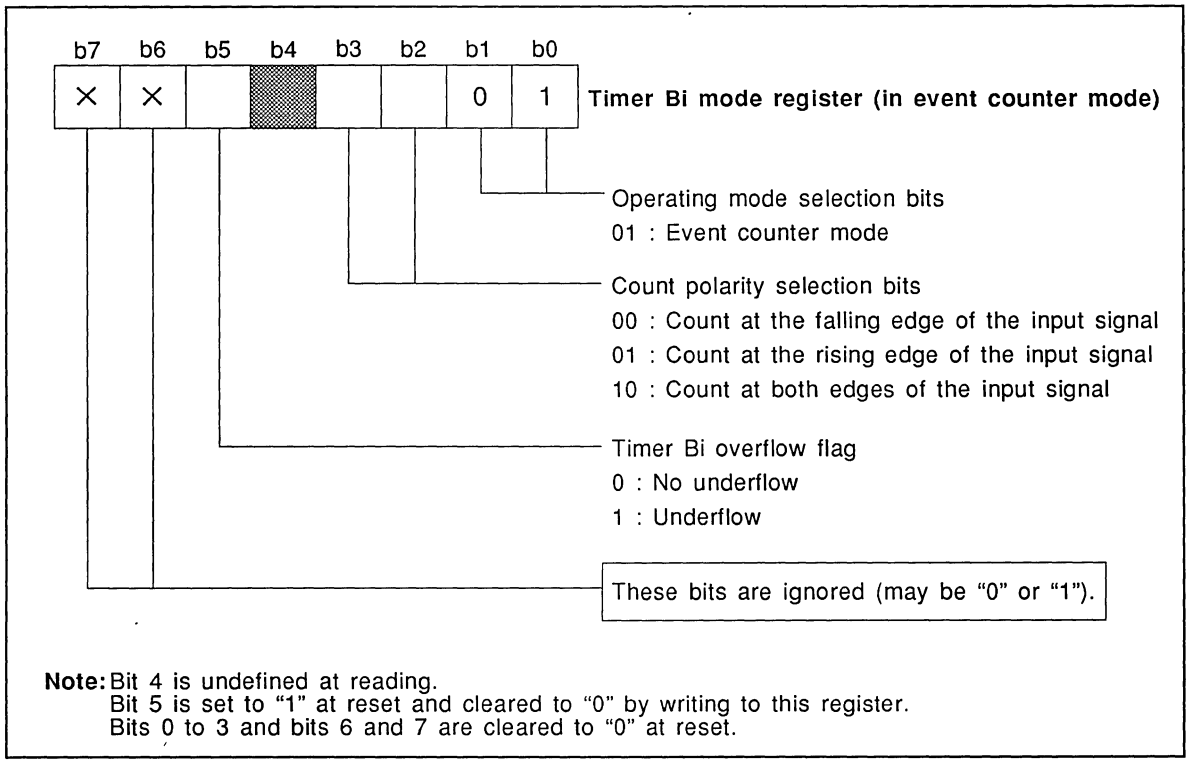
The event counter mode is selected by setting the timer Bi mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, the timer Bi mode register bits 6 and 7 are ignored. Figure 2.8.9 shows the structure of the timer Bi mode register in event counter mode.

In event counter mode, the input external clock to the TBi_{IN} pin is counted. The counter is decremented each time a effective edge is input and an interrupt request occurs when the counter underflows (the content of the counter reaches 0000_{16} → reload value "n").

The timer dividing ratio is expressed as follows:

Timer dividing ratio = 1/(n+1)

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)



Note: Bit 4 is undefined at reading.
Bit 5 is set to "1" at reset and cleared to "0" by writing to this register.
Bits 0 to 3 and bits 6 and 7 are cleared to "0" at reset.

Fig. 2.8.9 Timer Bi mode register structure in event counter mode

(1) Event counter mode operation

First, select the operating mode and the effective edge of the count source with the timer Bi mode register. Next, write a value "n" ("n"=0000$_{16}$ to FFFF$_{16}$) in the timer Bi register specity the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register. Also, set the corresponding port P6 direction register bit to TBi$_{IN}$ pin to "0" (input mode). When the count start flag is set to "1" (count enabled), the effective edge of the input signal to TBi$_{IN}$ pin is detected and count operation starts.

Table 2.8.4 shows the relationship between the count polarity selection bits and the effective edge of the count. Figure 2.8.10 shows the setting example of the event counter mode related registers.

Table 2.8.4 Relationship between count polarity selection bits and effective edge

b3	b2	Count effective edge
0	0	Falling edge of input signal
0	1	Rising edge of input signal
1	0	Both edges of input signal

The content of the counter is decremented by 1 each time an effective edge is detected. When the counter underflows, the content of the reload register is loaded in the counter and the interrupt request bit is set to "1". At the same time, the timer Bi overflow flag is set to "1". The count operation continues in this manner and an interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, a timer Bi interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.

The content of the counter can be read at any time by reading the content of timer Bi register, but the content of the reload register can not be read.

In order to change the timer dividing ratio while the timer is operating, a 16-bit update value must be written simultaneously in the timer Bi register. This value is stored in the reload register and loaded in the counter next time the counter underflows after writing to timer Bi register.

The operation in event counter mode is identical to that of the timer mode except that an external clock input to the TBi$_{IN}$ pin is counted. Refer to "Fig. 2.8.8 Timer mode operation diagram" about an operation diagram.

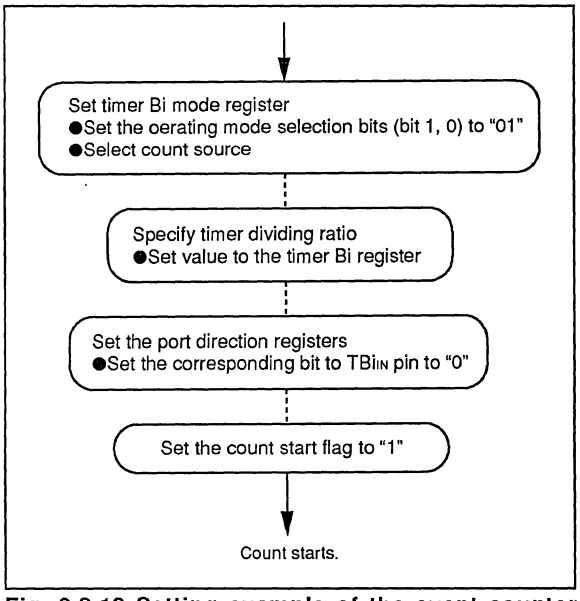


Fig. 2.8.10 Setting example of the event counter mode related registers

2.8.5 Pulse period/pulse width measurement mode [timer Bi mode register bits 1, 0="10"]

The pulse period/pulse width measurement mode is selected by setting the timer Bi mode register bit 1 to "1" and bit 0 to "0". The difference between pulse period measurement and pulse width measurement is the effective edge of the input signal determining the count term.

●Pulse period measurement

Pulse period is measured by counting during the period between the falling edge and the next falling edge or between the rising edge and the next rising edge of the input signal to the TBi_{IN} pin.

●Pulse width measurement

Pulse width is measured by counting during the period between the falling edge and the next rising edge or between the rising edge and the next falling edge of the input signal to the TBi_{IN} pin.

Figure 2.8.11 shows the structure of the timer Bi mode register in pulse period/pulse width measurement mode.

(1) Pulse period /pulse width measurement mode description

First, select the operating mode, whether to measure the pulse period or width, and count source with the timer Bi mode register. Set the corresponding port P6 direction register bit to TBi_{IN} pin to "0" (input mode). When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and count operation starts. The counter is incremented by 1 each time a count source is input.

The content of the counter is transferred to the reload register when an effective edge is input to the TBi_{IN} pin. Then the counter is cleared to 0000_{16}. The count operation continues and the content of the counter is transferred to the reload register again when the next effective edge is input to the TBi_{IN} pin. Then the counter is cleared to 0000_{16}. At the same time, an interrupt request occurs and the interrupt request bit is set to "1". The measured result can be obtained by reading the timer Bi register because the content of the reload register is read. An interrupt request does not occur at the first effective edge after setting the count start flag to "1".

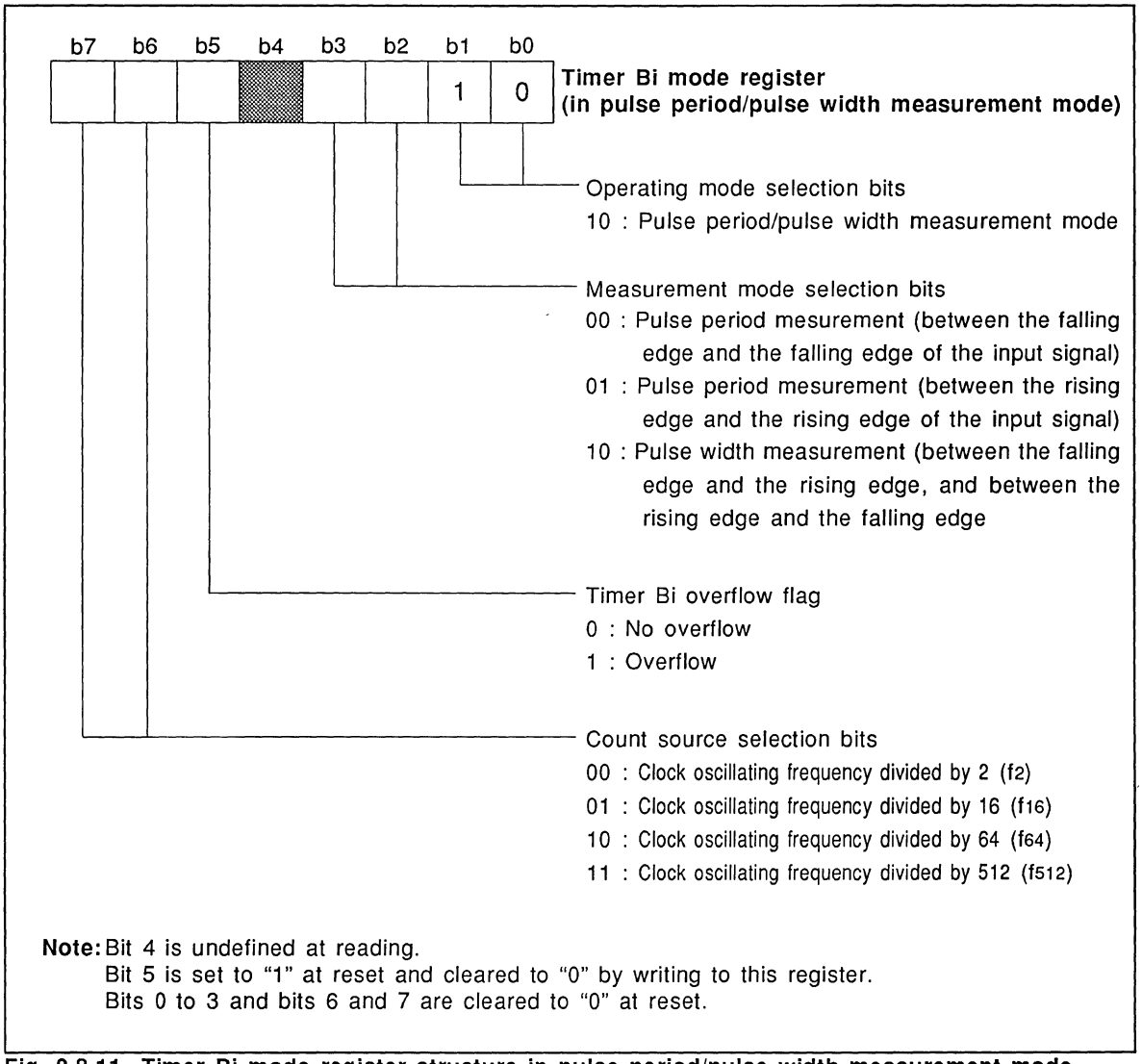
The interval between the falling edge and the rising edge or between the rising edge and the falling edge of the input signal to the TBi_{IN} pin must be at least 2 cycles of the count source. When measuring the pulse width of a signal other than 50% duty, whether the result is "H" level width or "L" level width must be determined with program.

Table 2.8.5 shows the relationship between measurement mode selection bits and measured intervals.

Table 2.8.5 Relationship between measurement mode selection bits and measured intervals

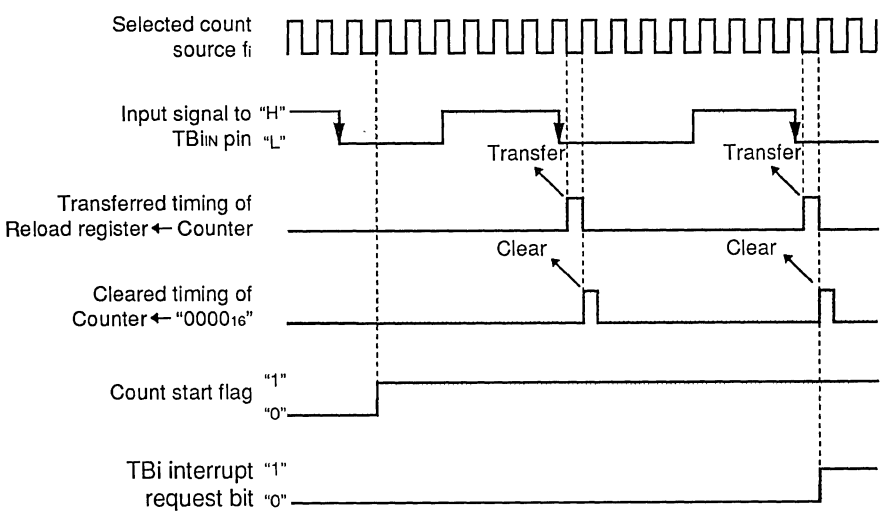
b3	b2	Measurement mode	Measured intervals
0	0	Pulse period	Between the falling edge and the falling edge of the input signal
0	1		Between the rising edge and the rising edge of the input signal
1	0	Pulse width	Between the falling edge and the rising edge, and between the rising edge and the falling edge

In this mode, an interrupt request also occurs when the content of the counter reaches $FFFF_{16} \rightarrow 0000_{16}$ in addition to the effective edge. When an overflow occurs, the timer Bi overflow flag is set to "1". Therefore, the cause of interrupt must be determined in the interrupt service routine by checking the timer Bi overflow flag.
Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt is accepted or it is cleared by program.
Figure 2.8.12 shows the pulse period/pulse width measurement mode operation diagram in pulse period measurement mode. Figure 2.8.13 shows the pulse period/pulse width measurement mode operation diagram in pulse width measurement mode.



Note: Bit 4 is undefined at reading.
Bit 5 is set to "1" at reset and cleared to "0" by writing to this register.
Bits 0 to 3 and bits 6 and 7 are cleared to "0" at reset.

Fig. 2.8.11 Timer Bi mode register structure in pulse period/pulse width measurement mode



This operation diagram measures between the falling edge and the falling edge.

Fig. 2.8.12 Pulse period/pulse width measurement mode operation diagram in pulse period measurement mode

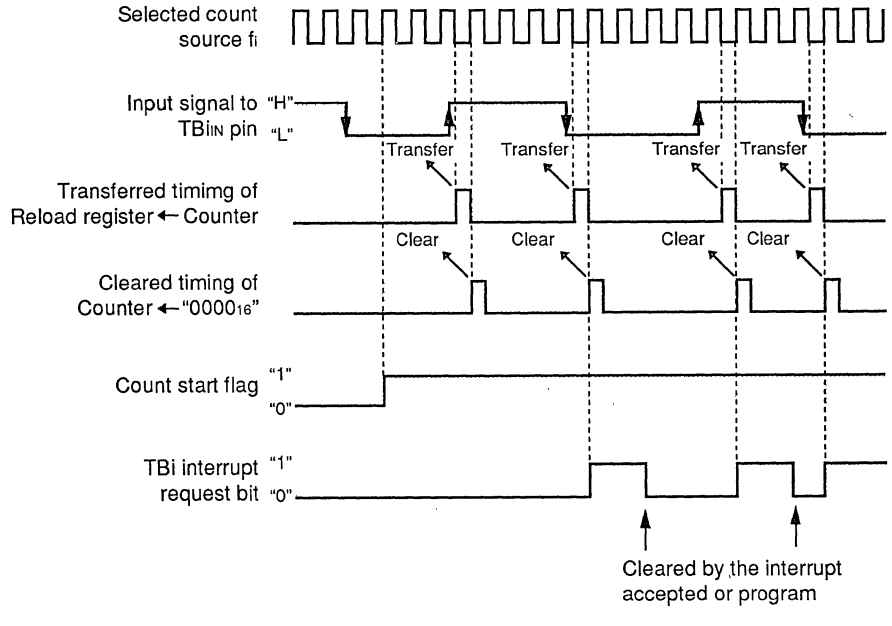


Fig. 2.8.13 Pulse period/pulse width measurement mode operation diagram in pulse width measurement mode

[Precaution when using the timer B]

1. The value of the counter while operating in timer mode or event counter mode can be read at any time by reading the timer Bi register. However, if it is read at the reload timing shown in Figure 2.8.14, "$FFFF_{16}$" is returned instead of the reload value. The reload value is returned if it is read after the timer Bi register is set and before the count source is input and count started.

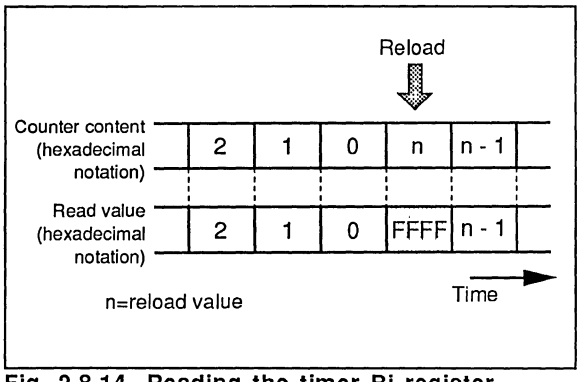


Fig. 2.8.14 Reading the timer Bi register

2. Interrupt request occurs by two factors in pulse period/pulse width measurement mode. These interrupts use the same interrupt vector address of timer Bi.
   Two factors are follows.

   ●Completing the measurement of the pulse period/pulse width
   ●Overflow of the counter

2.9 Serial I/O

Serial I/O consists of UART0 and UART1 that have same functions. These serial I/O can operate either as clock synchronous serial I/O port or asynchronous serial I/O (UART) port.

2.9.1 Serial I/O description

UART0 and UART1 can operate either as clock synchronous serial I/O port or asynchronous serial I/O (UART) port. These two serial I/O ports are independent, but have identical functions. Each serial I/O port has a transfer clock generation timer (baud rate generator referred to BRG) and can be set a variety of data transfer rate.

Figure 2.9.1 shows the serial I/O operating modes.

Each serial I/O has four operating modes. The following modes are available:

●Clock synchronous serial I/O

In this mode, both the transmission side and receiving side use the same clock to transfer data. The data (character) length is 8 bits.

●7-bit UART

In this mode, the data is transferred at an arbitrary rate and data format. The data (character) length is 7 bits.

●8-bit UART

This mode is identical to 7-bit UART except that the data length is 8 bits.

●9-bit UART

This mode is identical to 7-bit UART except that the data length is 9 bits.

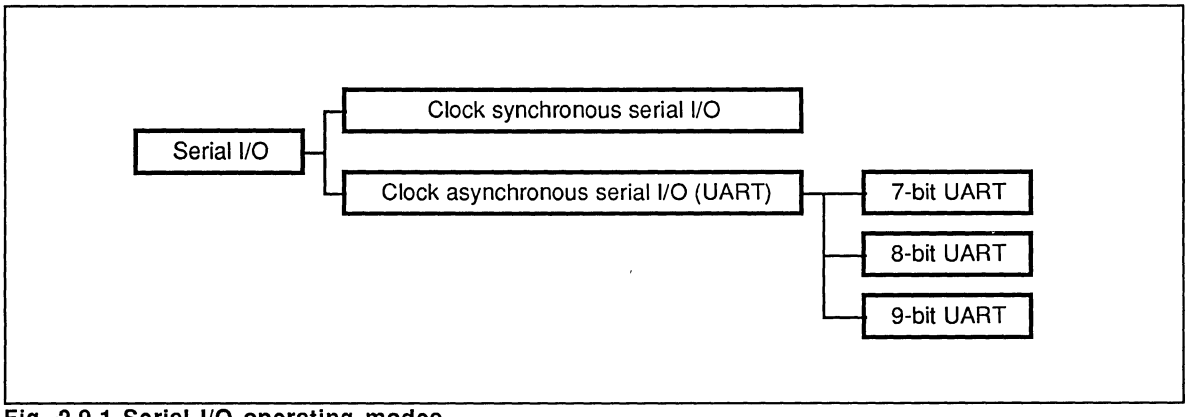


Fig. 2.9.1 Serial I/O operating modes

2.9.2 Block description

Figure 2.9.2 shows the block diagram of serial I/O. The function of each related registers are described below.

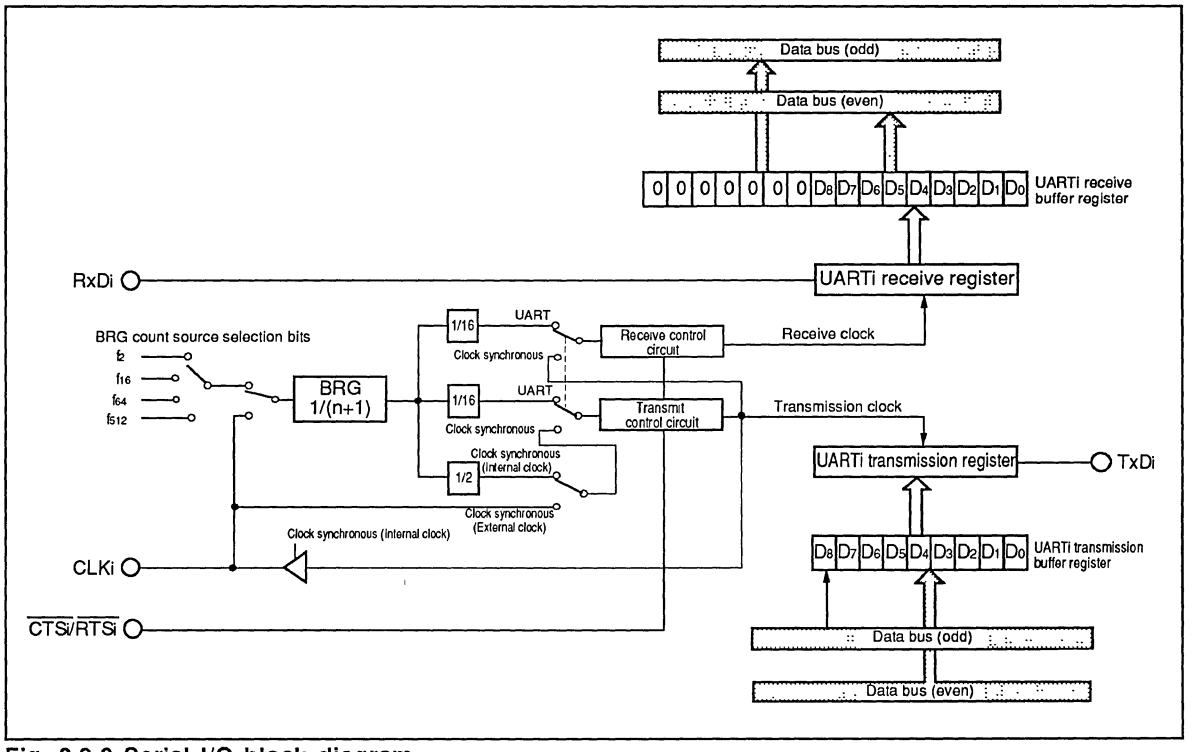
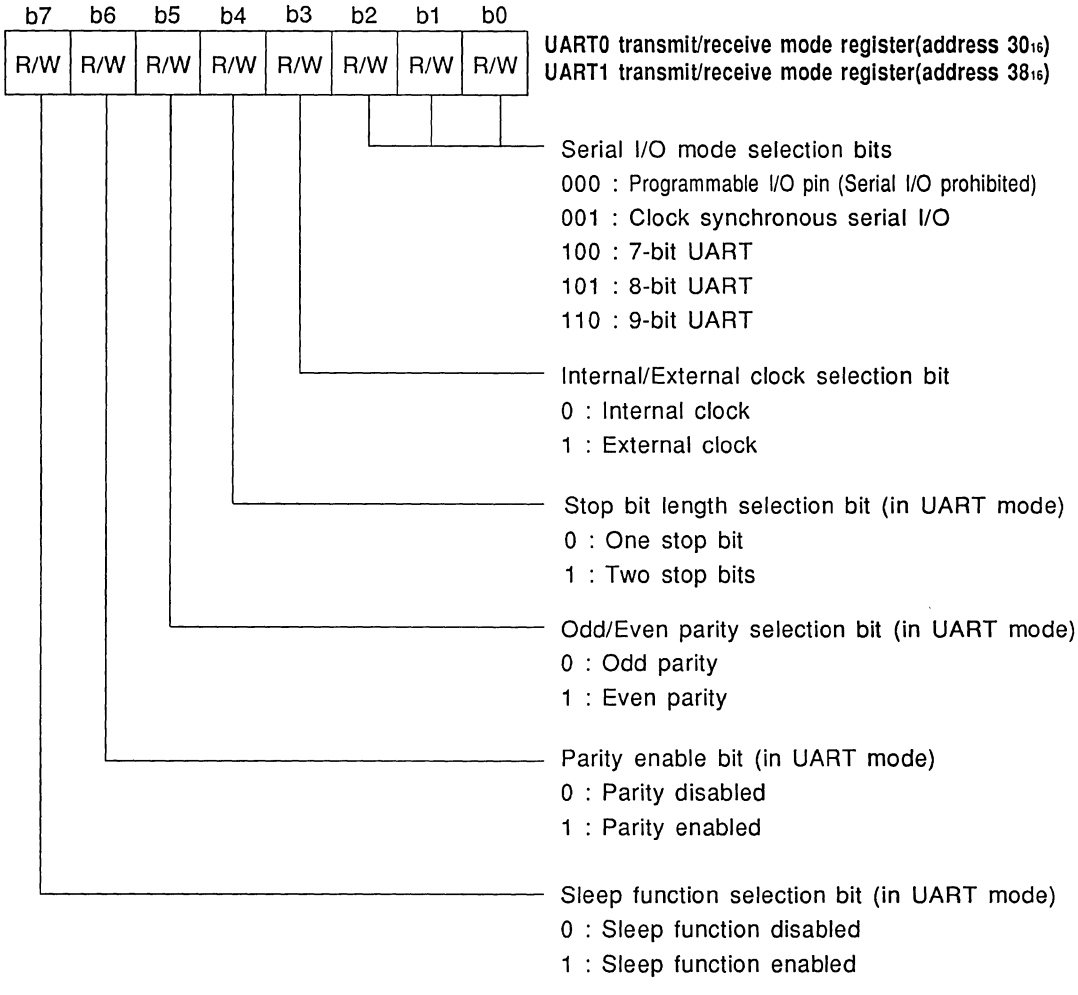


Fig. 2.9.2 Serial I/O block diagram

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

The UART0 transmit/receive mode register(address 30_{16}) and the UART1 transmit/receive mode register(address 38_{16}) consist of bits to set serial I/O modes, transfer format.
Figure 2.9.3 shows the structure of the UARTi transmit/receive mode register.



Note : Bits 4 to 6 are ignored in clock synchronous mode.
Bit 7 must be set to "0" when using clock synchronous mode.
This register is cleared to "00_{16}" at reset.

Fig. 2.9.3 UARTi transmit/receive mode register structure

●**Serial I/O mode selection bits (bits 0 to 2)**

These bits are used to select serial I/O modes.

Table 2.9.1 shows the relationship between the serial I/O mode selection bits and serial I/O modes. When bits 2 to 0 are set to "000", serial I/O is disabled and ports $P8_0$ to $P8_3$, and $P8_4$ to $P8_7$ function as programmable I/O ports. When one of the serial I/O modes is selected, port P8 has the function shown in Table 2.9.2 and loses its programmable I/O port function (except for some pins in UART mode).

Table 2.9.1 Relationship between serial I/O mode selection bits and serial I/O modes

b2	b1	b0	Serial I/O mode selection bits
0	0	0	Programmable I/O pin (Serial I/O prohibited)
0	0	1	Clock synchronous serial I/O
0	1	0	Not available
0	1	1	Not available
1	0	0	7-bit UART
1	0	1	8-bit UART
1	1	0	9-bit UART
1	1	1	Not available

Table 2.9.2 Function of port P8 when serial I/O is selected

Using UART0	Using UART1	Function
$P8_0$	$P8_4$	$\overline{CTS}/\overline{RTS}$ (transmission control signal I/O pin)
$P8_1$	$P8_5$	CLK (transfer clock I/O pin) (Note 1)
$P8_2$	$P8_6$	RxD (serial data input pin)
$P8_3$	$P8_7$	TxD (serial data output pin) (Note 2)

Note 1 : This depends on the internal/external clock selection bit as follows:
When external clock is selected : Clock input pin
When internal clock is selected : •Clock output pin in clock synchronous mode
•Normal I/O port in UART mode

Note 2 : TxD pin starts to output "H" level when one of serial I/O modes is selected.

●**Internal/external clock selection bit (bit 3)**

[Clock synchronous mode]

This bit is used to select either an internal clock or an external clock as the synchronous clock (shift clock) for data transfer.

When this bit is set to "0" to select an internal clock, the divided clock by 2 which the later described baud rate generator (BRG) generates is used as the shift clock. In addition, the CLKi pin becomes the output pin and the shift clock is output from this pin.

When this bit is set to "1" to select an external clock, the CLKi pin becomes the input pin and data transfer is synchronized with the clock input to this pin.

[UART mode]

This bit is used to select either an internal clock or an external clock as the input to the BRG which is described later.

When this bit is set to "0" to select an internal clock, the clock selected with the BRG count source selection bit in the UARTi transmit/receive control register 0 becomes the BRG input clock. In this case, the CLKi pin(ports $P8_1$, $P8_5$) can be used as a programmable I/O pin.

When this bit is set to "1" to select an external clock, the CLKi pin becomes the clock input pin and the clock input to this pin becomes the BRG input clock.

●**Stop bit length selection bit (bit 4)**

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to select between 1 and 2 bits as the stop bit to indicate the end of data.

1 stop bit is selected when this bit is "0". 2 stop bits are selected when this bit is "1".

●**Odd/even parity selection bit (bit 5)**

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to select between even parity and odd parity. Odd parity is selected when this bit is "0", and even parity is selected when this bit is "1".

This bit is valid if the parity enable bit is set to "1" (enabled).

●**Parity enable bit (bit 6)**

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to specify whether to add a parity bit at the end of transmitted data or to perform parity check of received data. Whether to use odd parity or even parity is specified with bit 5.

When this bit is "1", a parity is added at transmitting, or parity check is performed at receiving.

●**Sleep function selection bit (bit 7)**

[Clock synchronous mode]

This bit must be set to "0".

[UART mode]

This bit is used to enable or disable the sleep function. If this bit is set to "1" to enable the sleep function, the data is ignored when the most significant bit (MSB) of the received data is "0".

This function is used when multiple microcomputers are connected through the serial I/O port. Refer to section "2.9.5 Sleep mode".

(2) UARTi transmit/receive control register 0

The UART0 transmit/receive control register 0(address 34_{16}) and the UART1 transmit/receive control register 0(address $3C_{16}$) consist of bits to select the BRG count source and $\overline{CTS}/\overline{RTS}$ function, and a flag that indicates the UARTi transmission register status. Figure 2.9.4 shows the structure of the UARTi transmit/receive control register 0.

●BRG count source selection bits (bits 0 and 1)

This bit is used to select the count source of the baud rate generator (BRG) when an internal clock is selected. Table 2.9.3 shows the relationship between the BRG count source selection bits and the count source.

Table 2.9.3 Relationship between BRG count source selection bits and count source

b1	b0	BRG count source
0	0	f_2 selected which is the oscillating clock f(XIN) divided by 2
0	1	f_{16} selected which is the oscillating clock f(XIN) divided by 16
1	0	f_{64} selected which is the oscillating clock f(XIN) divided by 64
1	1	f_{512} selected which is the oscillating clock f(XIN) divided by 512

●$\overline{CTS}/\overline{RTS}$ function selection bit (bit 2)

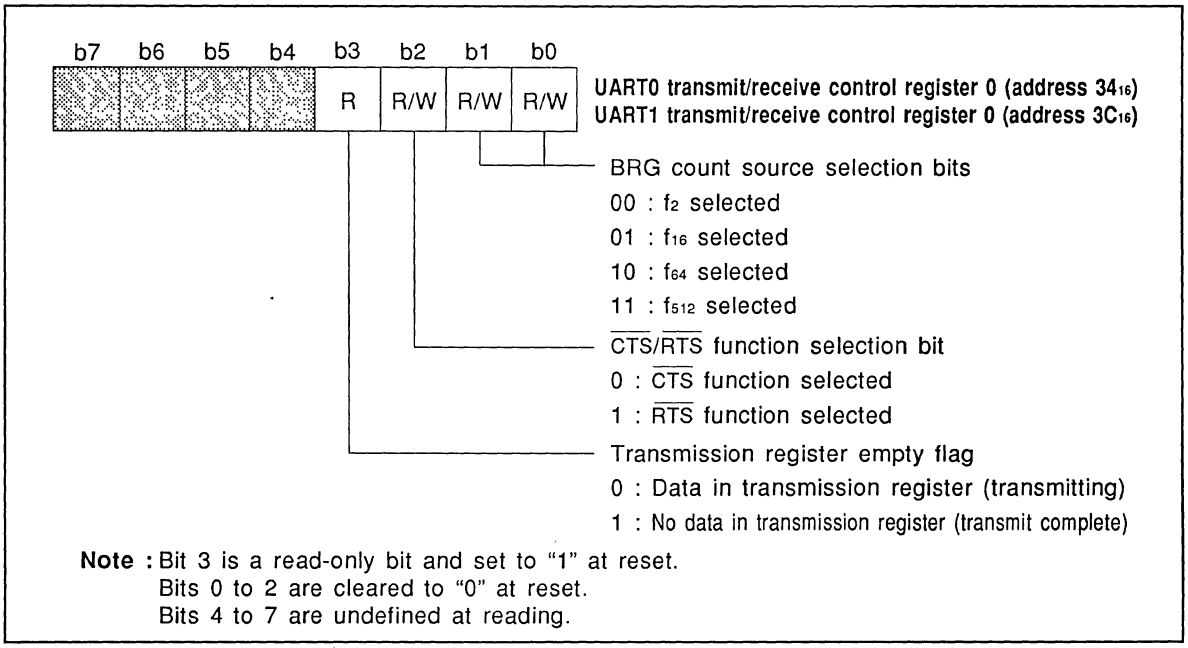
This bit is used to select $\overline{CTS}$ and $\overline{RTS}$ function. Port $P8_0$ functions as $\overline{CTS}/\overline{RTS}$ pin for UART0, $P8_4$ as $\overline{CTS}/\overline{RTS}$ pin for UART1.

When this bit is "0", $\overline{CTS}$ function is selected. Port $P8_0$ or $P8_4$ becomes the $\overline{CTS}$ input pin. This pin must be at "L" level in order for transmission to start.

When this bit is "1", $\overline{RTS}$ function is selected. port $P8_0$ or $P8_4$ becomes the $\overline{RTS}$ output pin. "H" level is output when receive is disabled (the receive enable bit in UARTi transmit/receive control register 1 is "0"). "L" level is output when receive is enabled (the receive enable bit is "1"). It returns to "H" level when receive starts.

●Ttransmission register empty flag (bit 3)

This flag is set to "0" when the content of UARTi transmission buffer register is transferred to the UARTi transmission register. It is set to "1" when transmission completes and the UARTi transmission register becomes empty.

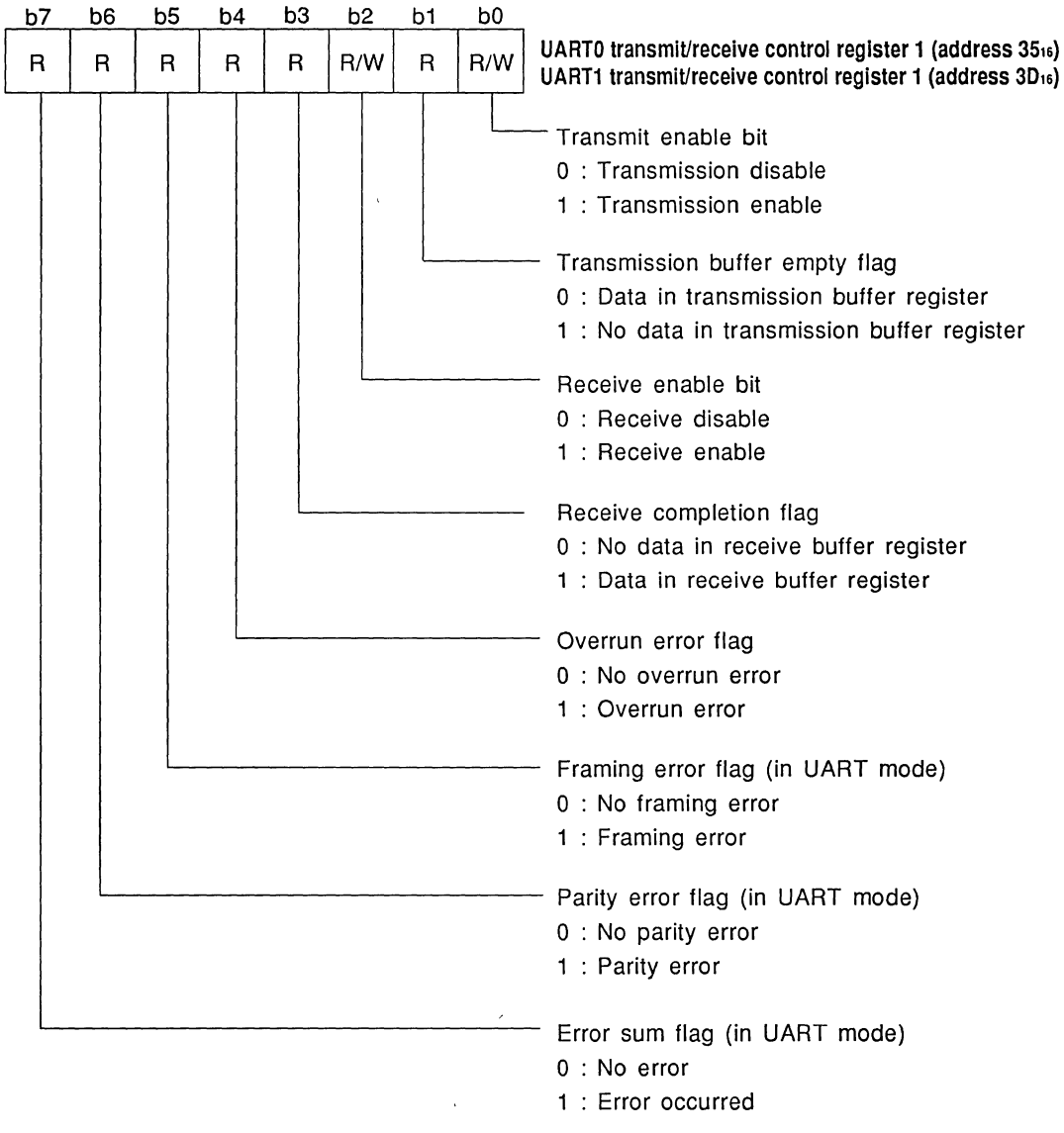


Note : Bit 3 is a read-only bit and set to "1" at reset.
Bits 0 to 2 are cleared to "0" at reset.
Bits 4 to 7 are undefined at reading.

Fig. 2.9.4 UARTi transmit/receive control register 0 structure

(3) UARTi transmit/receive control register 1

The UART0 transmit/receive control register 1(address 35_{16}) and the UART1 transmit/receive control register 1(address $3D_{16}$) consist of serial I/O enable bits, serial I/O status flags, and serial I/O error flags. Figure 2.9.5 shows the structure of the UARTi transmit/receive control register 1.



Note : Bits 5 to 7 are ignored in clock synchronous mode.
Bit 0 and bits 2 to 7 are cleared to "0" at reset.
Bit 1 is set to "1" at reset.
The receive enable bit and each error flag are cleared to "0" when the receive buffer register is read.

Fig. 2.9.5 UARTi transmit/receive control register 1 structure

●**Ttransmit enable bit (bit 0)**
Serial I/O transmission is enabled when this bit is set to "1". If this bit is set to "0" during transmitting, serial I/O transmission is disabled after the current transmission completes.

●**Transmission buffer empty flag (bit 1)**
This flag indicates the status of the transmission buffer register. This bit is set to "1" when the content of the transmission buffer register is transferred to the transmission register. This flag is automatically cleared to "0" when data is written in the transmission buffer register.

●**Receive enable bit (bit 2)**
Serial I/O reception is enabled when this bit is set to "1". If the $\overline{RTS}$ function is selected, the $\overline{RTS}$ pin becomes "L" level when this bit is set to "1".

●**Receive completion flag (bit 3)**
This flag is set to "1" when the data in the receive register is transferred to the receive buffer register (receive completion). This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

●**Overrun error flag (bit 4)**
This flag is set to "1" when receiving of the next data completes and the content of the receive buffer register is updated while there is data remaining in the receive buffer register (before the content of the receive buffer register is read).
This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

●**Framing error flag (bit 5)**
[Clock synchronous mode]
This flag is ignored.

[UART mode]
This flag is set to "1" when the number of stop bits is not the number specified with bit 4 of the UARTi transmit/receive mode register. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

●**Parity error flag (bit 6)**
[Clock synchronous mode]
This flag is ignored.

[UART mode]
This flag is set to "1" when the parity odd/even is not the one specified. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

●**Error sum flag (bit 7)**
[Clock synchronous mode]
This flag is ignored.

[UART mode]
This flag is set to "1" when either an overrun error, a framing error, or a parity error occurs. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

(4) UARTi transmission register and UARTi transmission buffer register

The UART0 transmission buffer register(address 32_{16}, 33_{16}) and the UART1 transmission buffer register(address $3A_{16}$, $3B_{16}$) are registers to set data output from TxDi pin.

When transmit conditions are satisfied, the transmit data written in the UARTi transmission buffer register is transferred to the UARTi transmission register, and is synchronously transmitted from the TxDi pin with the specified clock.

In clock synchronous mode and 7 or 8-bit UART mode, only the low-order byte of the UARTi transmission buffer register is used. In 9-bit UART mode, bit 8 of the transmit data is written in bit 0 of the high-order byte, and the remaining 0 to 7 bits are written in the low-order byte.

The UARTi transmission buffer register becomes empty after the data is transferred to the UARTi transmission register. Therefore, the next transmit data can be written during transmission.

The content of UARTi transmission buffer register can not be read because it is a write-only register.

Figure 2.9.6 shows the block diagram of serial I/O transmission.

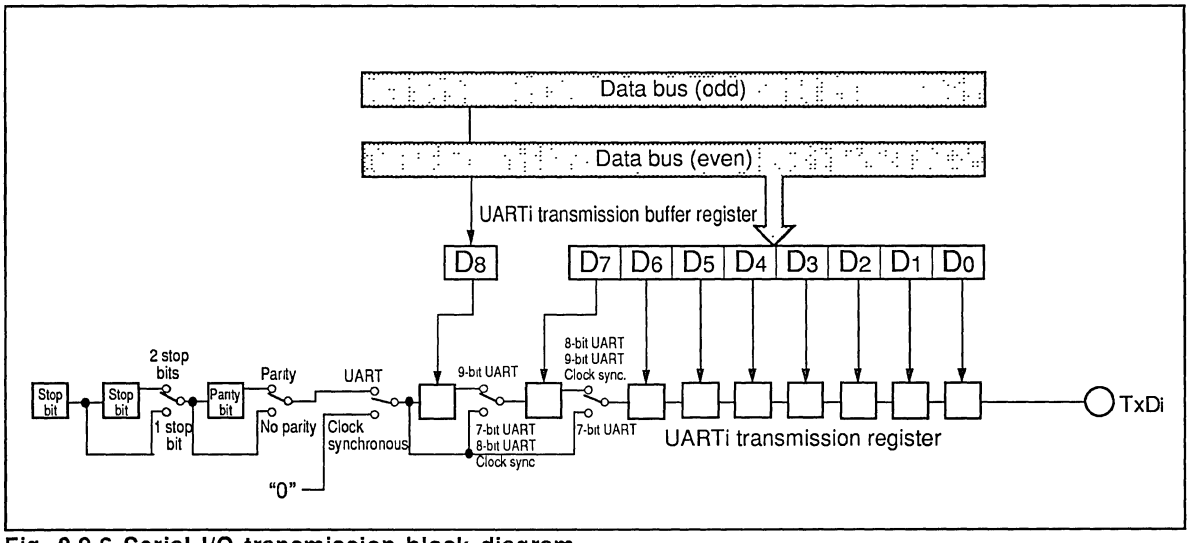


Fig. 2.9.6 Serial I/O transmission block diagram

(5) UARTi receive register and UARTi receive buffer register

The UARTi receive register converts serial data input to the RxDi pin to parallel data. The RxDi pin level is moved bit by bit to the UARTi receive register synchronized with the rising edge of the synchronous clock.

The UART0 receive buffer register(address 36_{16}; 37_{16}) and the UART1 receive buffer register(address $3E_{16}$, $3F_{16}$) are registers to read the received data. The content of UARTi receive register is automatically transferred to the UARTi receive buffer register when data receive completes.

The contents of the high-order 7 bits of the UARTi receive buffer register are always "0" at reading. The same data as the MSB (most significant bit) of effective receive data can be read from the unused bits of the low-order 9 bits as follows:

- D_7 and D_8 in 7-bit UART mode.
- D_8 in 8-bit UART mode and clock synchronous mode.

Note that the content of UARTi receive buffer register will be updated if the next receive data becomes available before the UARTi receive buffer register is read(overrun error occurs).

Figure 2.9.7 shows the block diagram of serial I/O receive.

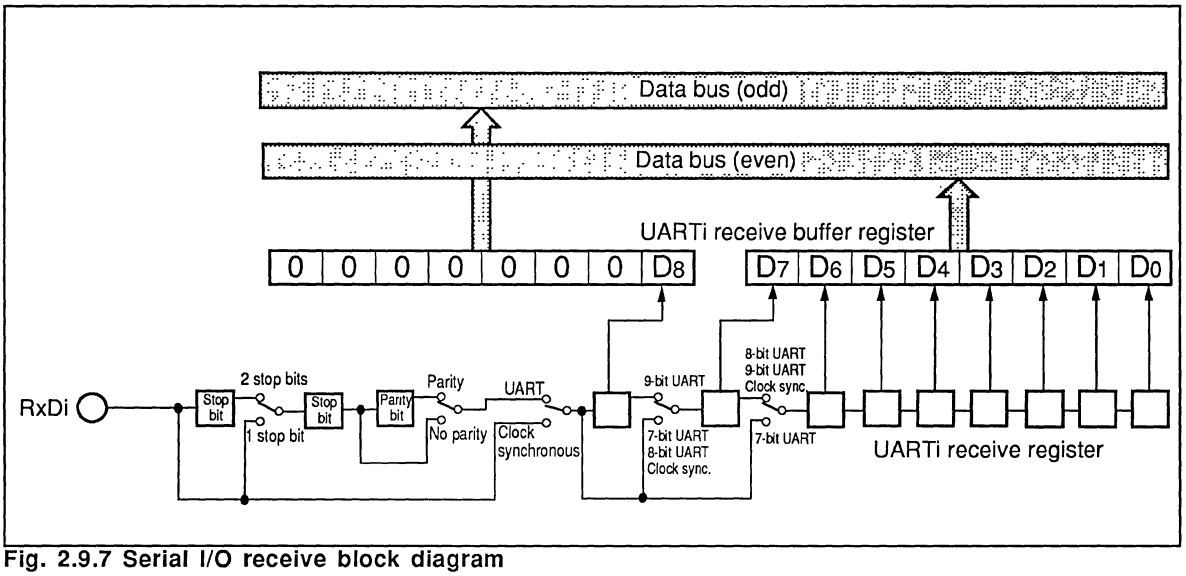


Fig. 2.9.7 Serial I/O receive block diagram

(6) UARTi baud rate generator (BRG)

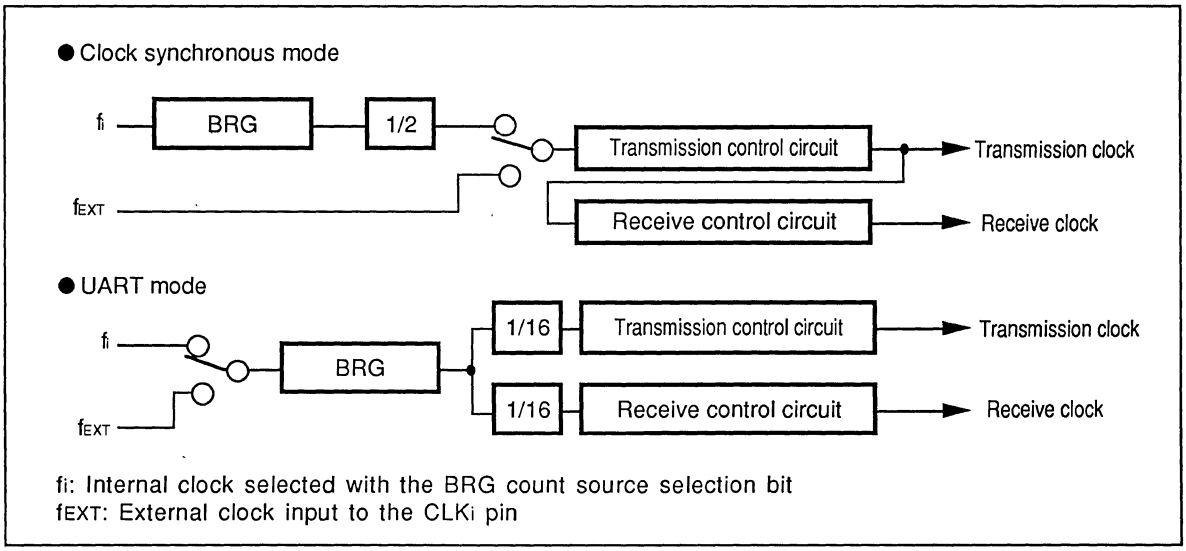
The UART0 baud rate generator (address 31_{16}) and the UART1 baud rate generator (address 39_{16}) are timers used exclusively for serial I/O. It is equipped with a reload register and has a 8-bit structure. The BRG divides the input clock by (n+1), where "n" is the value set in the BRG register. This register can contain a value between 00_{16} and FF_{16}.

In clock synchronous serial I/O mode, the BRG becomes effective when an internal clock is selected and the BRG output divided by 2 becomes the transmit/receive clock.

In UART mode, the BRG is effective regardless of the clock type and the BRG output divided by 16 becomes the transmit/receive clock.

The content of the BRG register can not be read because it is a write-only register.

Figure 2.9.8 shows the block diagram of shift clock generation.



fi: Internal clock selected with the BRG count source selection bit
fEXT: External clock input to the CLKi pin

Fig. 2.9.8 Shift clock generation block diagram

(7) UARTi transmission interrupt control register and UARTi receive interrupt control register

Transmit interrupt and receive interrupt can be used when the serial I/O function is selected. Each interrupt has an interrupt control register. Interrupt control register consists of interrupt priority level selection bits and interrupt request bit. Figure 2.9.9 shows the structure of UARTi transmission interrupt control register and UARTi receive interrupt control register.

Refer to section "2.6 Interrupts" for more information.

●Interrupt priority level selection bits (bits 0 to 2)

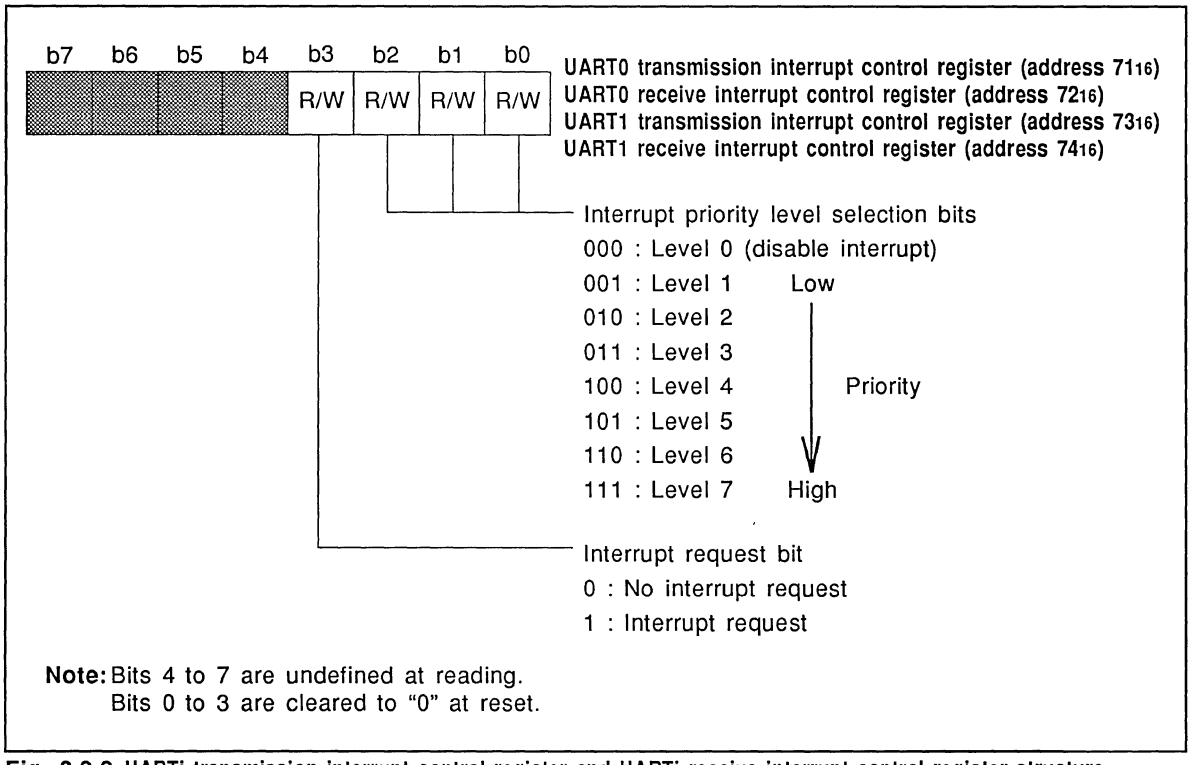
These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using serial I/O interrupts. When there is an interrupt request, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS). The interrupt is allowed only when this level is greater than IPL (interrupt disable flag I must be "0"). Set the corresponding bits to "000" (level 0) to disable an interrupt.

●Interrupt request bit (bit 3)

The transmit interrupt request bit is set to "1" when data is transferred from the UARTi transmission buffer register to the UARTi transmission register for data transmission.

The receive interrupt request bit is set to "1" when data receive completes and data is transferred from the UARTi receive register to the UARTi receive buffer register.

The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted. This bit can be set or cleared by program.



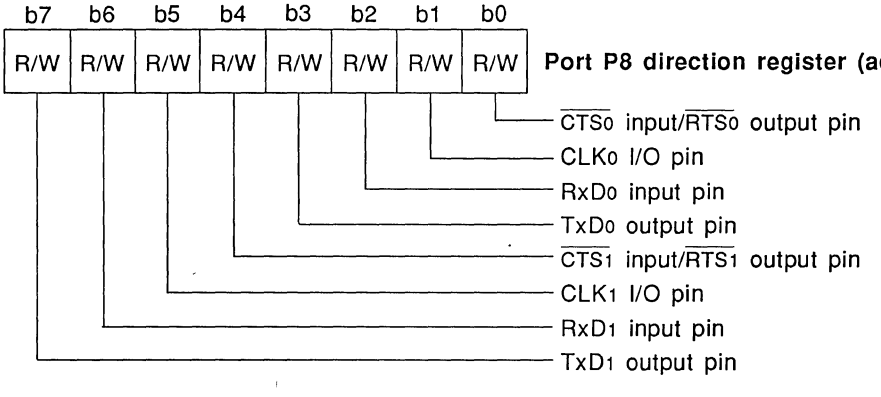
Note: Bits 4 to 7 are undefined at reading.
Bits 0 to 3 are cleared to "0" at reset.

Fig. 2.9.9 UARTi transmission interrupt control register and UARTi receive interrupt control register structure

(8) Port P8 direction register

Serial I/O input/output pins are shared with port P8. Port P8 function is selected by the serial I/O mode selection bits of the UARTi transmit/receive mode register. When using port P8 as serial I/O input pins, the corresponding bit in the direction register must be set to "0"(input mode). When using port P8 as serial I/O output pins, it functions as serial I/O output pins regardless of the direction register.

Figure 2.9.10 shows the relationship between the port P8 direction register(address 14_{16}) and the serial I/O pins.



Note: This register is cleared to "00_{16}" at reset.

Fig. 2.9.10 Relationship between port P8 direction register and serial I/O pins

2.9.3 Clock synchronous serial I/O

Table 2.9.4 shows the performance of clock synchronous serial I/O mode.

Table 2.9.4 Clock synchronous serial I/O description

Parameter		Function
Data format		8 bit fixed, LSB first
Transmission speed	Internal clock	BRG output divided by 2
	External clock	2Mbps maximum (at f(X_{IN})=8MHz)
		4Mbps maximum (at f(X_{IN})=16MHz)
		5Mbps maximum (at f(X_{IN})=25MHz)
Transmit/receive control		$\overline{CTS}$ input or $\overline{RTS}$ output can be selected by program.

(1) Synchronous clock (shift clock)

The serial I/O data transfer rate is determined by the synchronous clock (shift clock). The M37702 group can select whether to generate this clock internally or to use an external clock. The synchronous clock is generated internally when the UARTi transmit/receive mode register bit 3 is set to "0", and externally when it is set to "1".

In clock synchronous mode, the synchronous clock used for data transfer is generated by activating the transmitter. Therefore, the transmitter must be activated even when performing receive only.

●Using internal generation clock as synchronous clock

When the internal/external clock selection bit is set to "0", the BRG output divided by 2 is used as the synchronous clock. In this case, the CLKi pin becomes output mode and the transmit/receive synchronous clock is output from the CLKi pin.

The BRG is a serial I/O timer which has a 8-bit structure and is used as a frequency divider to generate the desired frequency. The BRG divides the clock selected with bits 0 and 1 in the UARTi transmit/receive control register 0 by (n+1). The synchronous clock is the divided clock by 2 which has been divided by (n+1) with the BRG. "n" is the value set in the BRG register. It can be set a value between 00_{16} and FF_{16}.

Synchronous clock frequency ..... f_i / (2(n+1))

f_i : BRG input frequency (i=2, 16, 64, 512)

8 synchronous clocks are generated by activating the transmitter.

●Using external input clock as synchronous clock

When the internal/external clock selection bit is set to "1", the external clock is used as the synchronous clock.

When an external clock is selected, the clock input to the CLKi pin becomes the synchronous clock. Set the port P8 direction register bit 1(CLK0) and bit 5(CLK1) to "0" to select input mode.

(2) Serial data transmission

The data transmission method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

Set each serial I/O control register for transmission.

●UARTi transmit/receive mode register

•Serial I/O mode selection bits

Set the bits 2 to 0 to "001".

•Internal/external clock selection bit

Select either an internal clock ("0") or an external clock ("1").

•Setting the bit 7 to "0" (disable sleep mode)

●UARTi transmit/receive control register 0

•BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for synchronous clock.

•$\overline{CTS}/\overline{RTS}$ function selection bit

Set the bit 2 to "0" when using $\overline{CTS}$ function, and to "1" when not use.

●UARTi baud rate generator(BRG)

•Dividing ratio

Set the BRG value between 00_{16} and FF_{16} when an internal clock is selected for synchronous clock.

●Port P8 direction register

•Port direction selection bits

Set the corresponding bit to "0" when the $\overline{CTS}$ function is selected and an external clock is selected.

●UARTi transmission interrupt control register

•Interrupt priority level selection bits

When using UARTi transmission interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTi transmission buffer register

•Transfer data

Set a transfer data to the low-order byte of UARTi transmission buffer register. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTi transmit/receive control register 1

•Transmit enable bit

Set the bit 0 to "1" to enable transmitting.

Figure 2.9.11 shows the setting example of clock synchronous serial I/O related registers at transmitting.

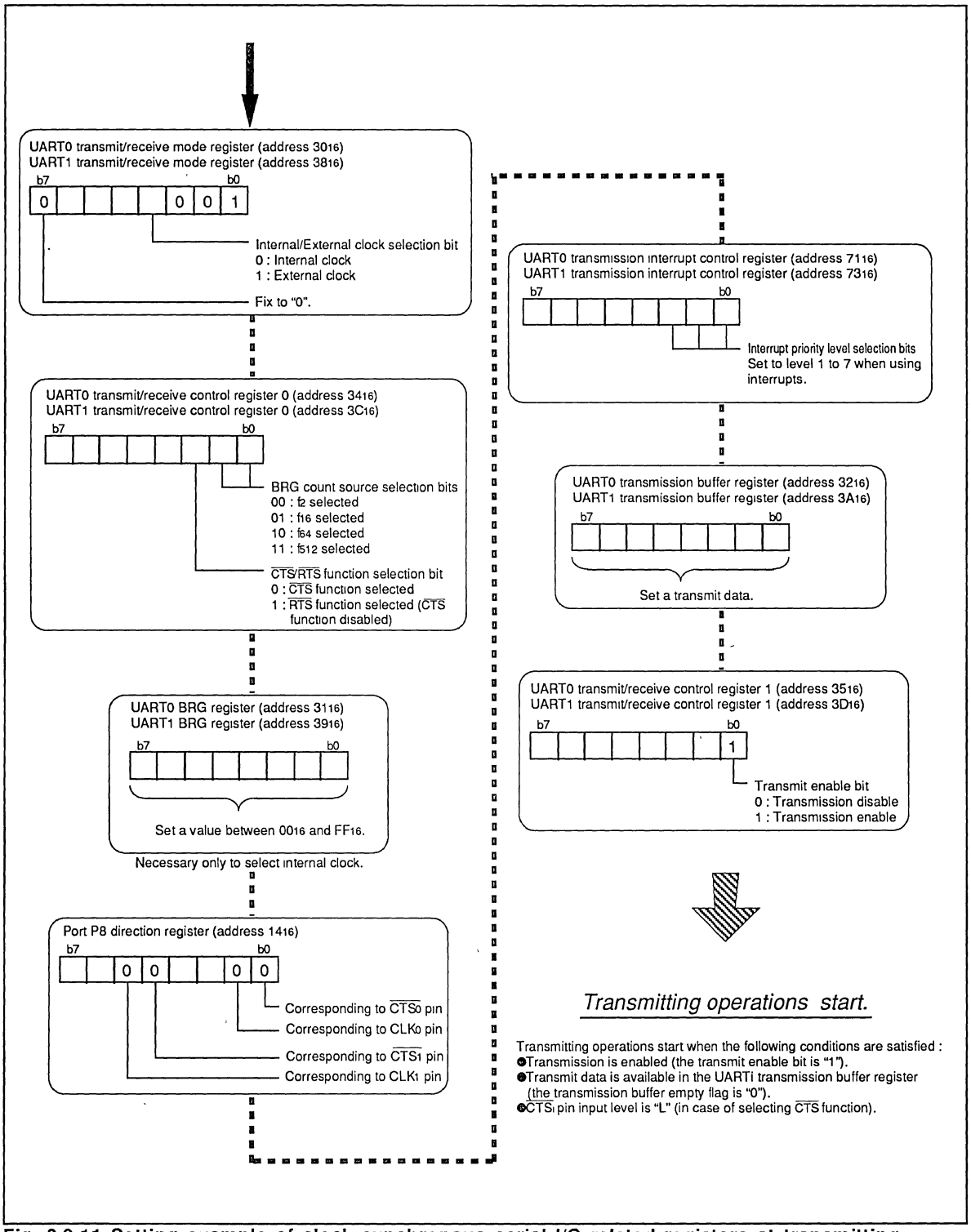


Fig. 2.9.11 Setting example of clock synchronous serial I/O related registers at transmitting

[Transmit operation]

The transmission of serial data starts when the following conditions are satisfied :

① Transmission is enabled (the transmit enable bit is "1").
② Transmit data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").
③ $\overline{CTSi}$ pin input level is "L".
(**Note :** This condition is ignored if the $\overline{CTS}$ function is not selected.)

When the above three (① to ③) conditions are satisfied (two (① and ②) if $\overline{CTS}$ function is not selected), the following operations are performed automatically at the same time :

- Transfer the content of UARTi transmission buffer register to the UARTi transmission register.
- Generate 8 shift clocks.
- Set the transmission buffer empty flag to "1".
- Clear the transmission register empty flag to "0".
- UARTi transmission interrupt request occurs and set the interrupt request bit to "1" .

Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information.
The shift clock is input to the transmit control circuit through the CLKi pin. The data in the UARTi transmission register is transmitted bit by bit from the TxDi pin (starting at the low-order bit) at each falling edge of shift clock. When the 1-byte data transmission is completed by the 8 shift clocks, the transmission register empty flag is set to "1". Figure 2.9.12 shows the clock synchronous serial I/O transmit operation.
The synchronous clock is generated continuously if the conditions for the next data are satisfied when the transmission completes. Therefore, to transmit data continuously, the next data must be written in the UARTi transmission buffer register while data is being transmitted (when the transmission register empty flag is "0"). If the conditions to transmit the next data are not satisfied, the synchronous clock halts at "H" level.
Figure 2.9.13 shows the timing diagram of clock synchronous serial I/O at transmitting (internal clock is selected as synchronous clock, $\overline{CTS}$ function is selected).

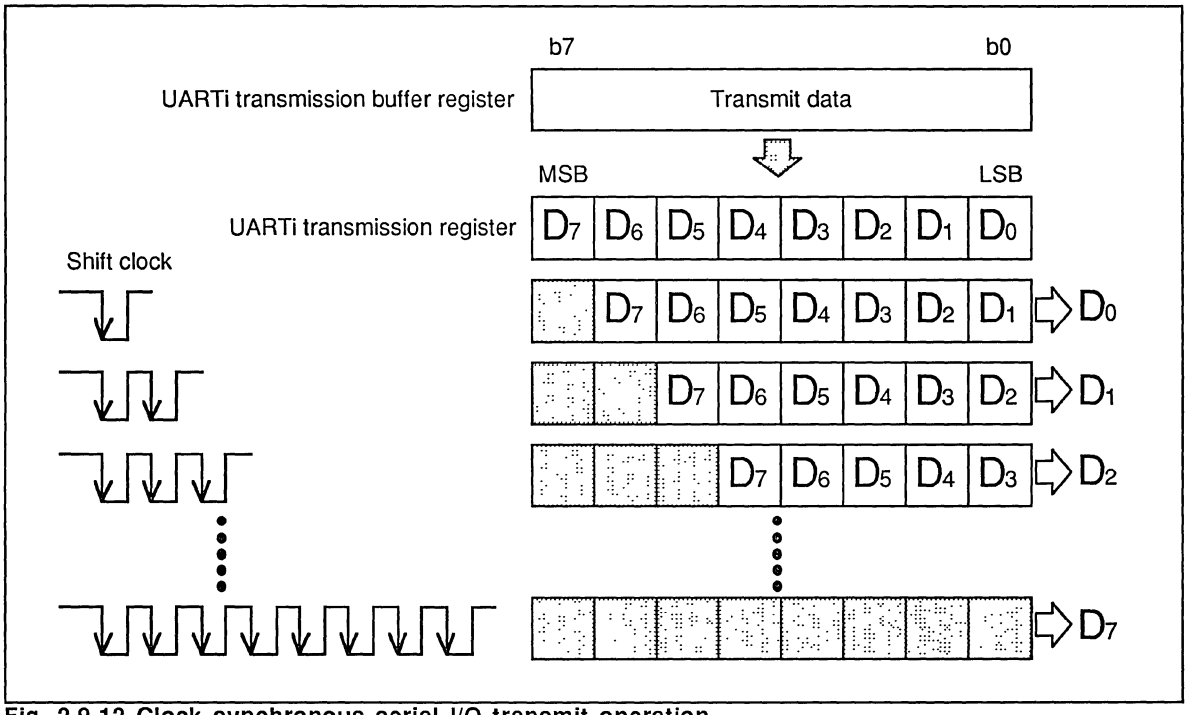
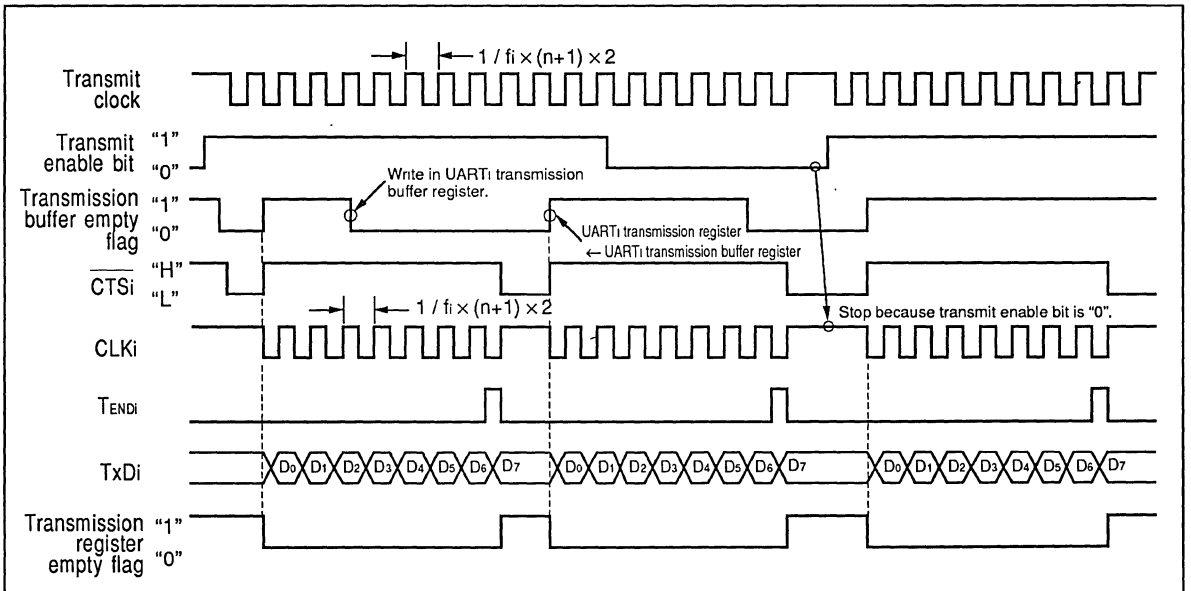


Fig. 2.9.12 Clock synchronous serial I/O transmit operation



fi : Selected internal clock frequency
TENDi : Check next transmit condition when this signal becomes "H" level.
(TENDi is an internal signal and can not be observed externally.)
n : Value set in the BRG

Fig. 2.9.13 Clock synchronous serial I/O timing diagram at transmitting (selecting internal clock)

(3) Serial data receive

The data receive method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

Set each serial I/O control register for receive.

●UARTi transmit/receive mode register

•Serial I/O mode selection bits
Set the bits 2 to 0 to "001".

•Internal/external clock selection bit
Select either an internal clock ("0") or an external clock ("1").

•Setting the bit 7 to "0" (disable sleep mode)

●UARTi transmit/receive control register 0

•BRG count source selection bits
Select the BRG count source with bits 0 and 1 when an internal clock is selected for synchronous clock.

•$\overline{CTS}/\overline{RTS}$ function selection bit
Set the bit 2 to "1" when using $\overline{RTS}$ function, and to "0" when not use.

●UARTi baud rate generator (BRG)

•Dividing ratio
Set the BRG value between 00_{16} and FF_{16} when an internal clock is selected for synchronous clock.

●Port P8 direction register

•Port direction selection bits
Set the corresponding bit to "0" to the CLKi pin in case that an external clock is selected, and RxDi pin.

●UARTi receive interrupt control register

•Interrupt priority level selection bits
When using UARTi receive interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTi transmission buffer register

•Dummy data
Set a dummy data to the low-order byte of UARTi transmission buffer register in order to activate a transmitter. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTi transmit/receive control register 1

•Transmit enable bit
Set the bit 0 to "1" in order to enable transmitting.

•Receive enable bit
Set the bit 2 to "1" in order to enable receiving.

Figure 2.9.14 shows the setting example of clock synchronous serial I/O related registers at receiving.

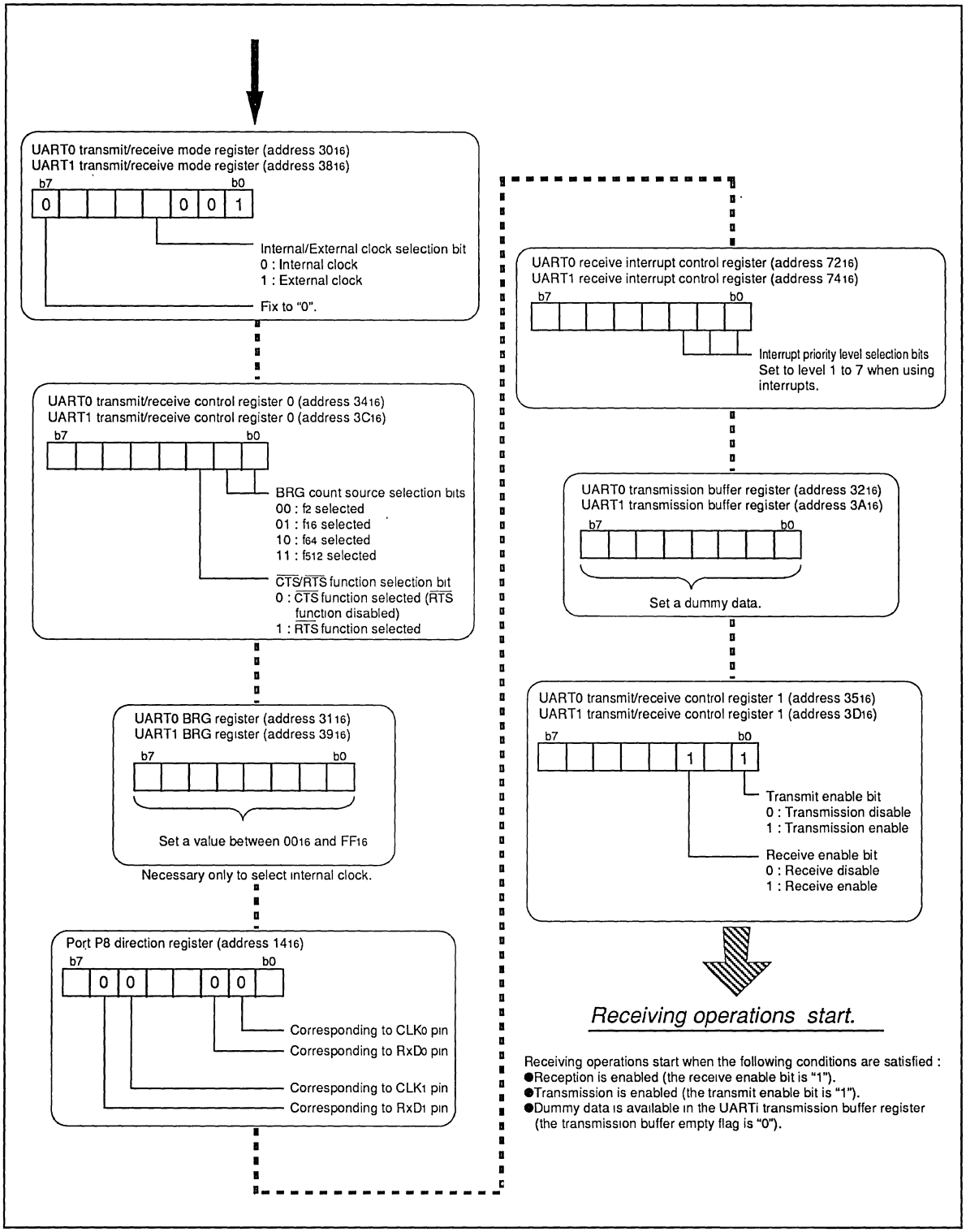


Fig. 2.9.14 Setting example of clock synchronous serial I/O related registers at receiving

[Receive operation]

The reception of serial data starts when the following conditions are satisfied :

① Reception is enabled (the receive enable bit is "1").
② Transmission is enabled (the transmit enable bit is "1").
③ Dummy data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").

Serial data receive is enabled by enabling transmission and setting the receive enable bit to "1". When the receive enable bit is set to "1", the $\overline{RTSi}$ pin becomes "L" level to indicate externally that the microcomputer is ready to receive serial data (in case that $\overline{RTS}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{RTSi}$ output pin to the $\overline{CTSi}$ pin on the transmit side. Figure 2.9.15 shows the connecting example of clock synchronous serial I/O.
The RxDi pin level is used to establish the most significant bit of the UARTi receive register at the rising edge of the shift clock (the clock input to the CLKi pin when an external clock is selected), and the content of the UARTi receive register is shifted by 1 bit to the right. This operation is repeated each time a rising edge is input. When 1-byte data is accumulated in the UARTi receive register after 8 shift clocks, the content of UARTi receive register is transferred to the UARTi receive buffer register. At the same time, the receive completion flag is set to "1". When the receive completion flag is set to "1", UARTi receive interrupt request occurs and the interrupt request bit is set to "1". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The receive completion flag is cleared to "0" when the UARTi receive buffer register is read.
Figure 2.9.16 shows the clock synchronous serial I/O receive operation and Figure 2.9.17 shows the timing diagram at receiving (external clock is selected, $\overline{RTS}$ function is selected).

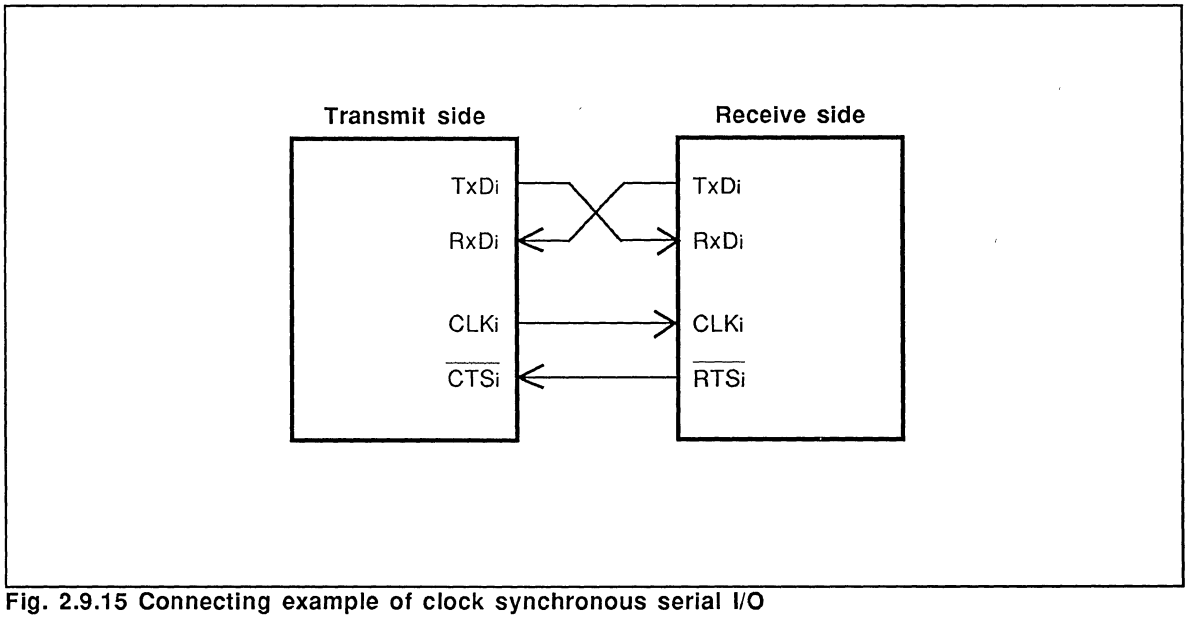


Fig. 2.9.15 Connecting example of clock synchronous serial I/O

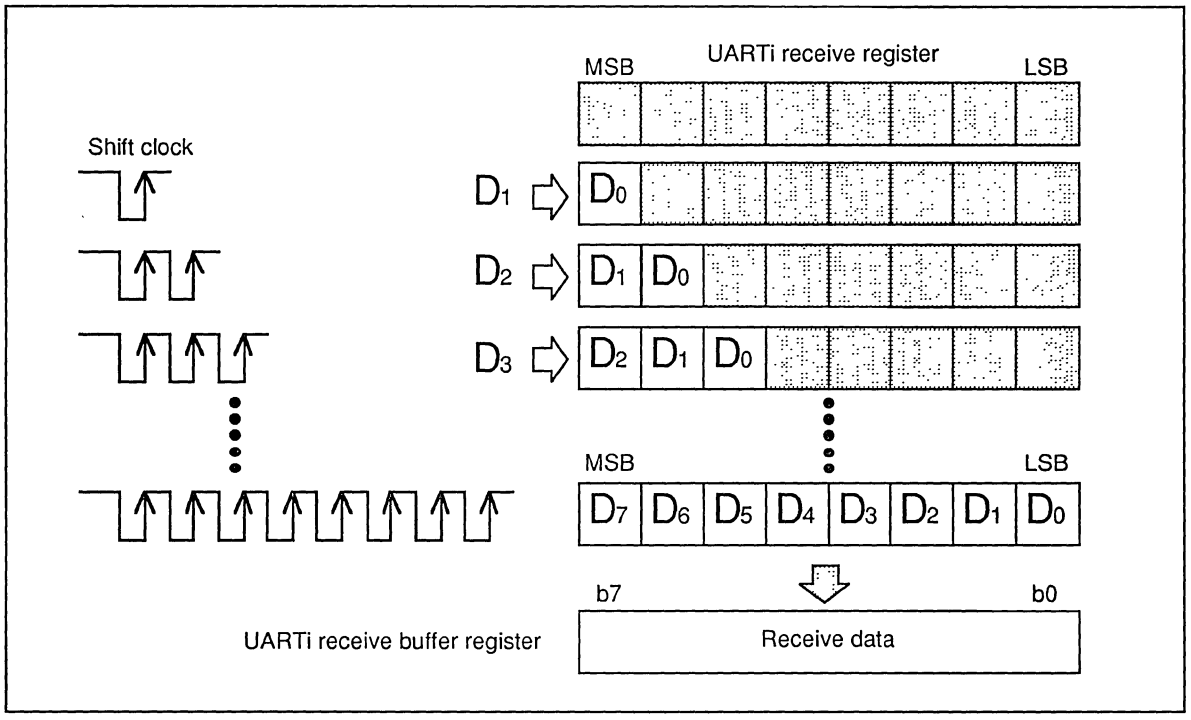


Fig. 2.9.16 Clock synchronous serial I/O receive operation

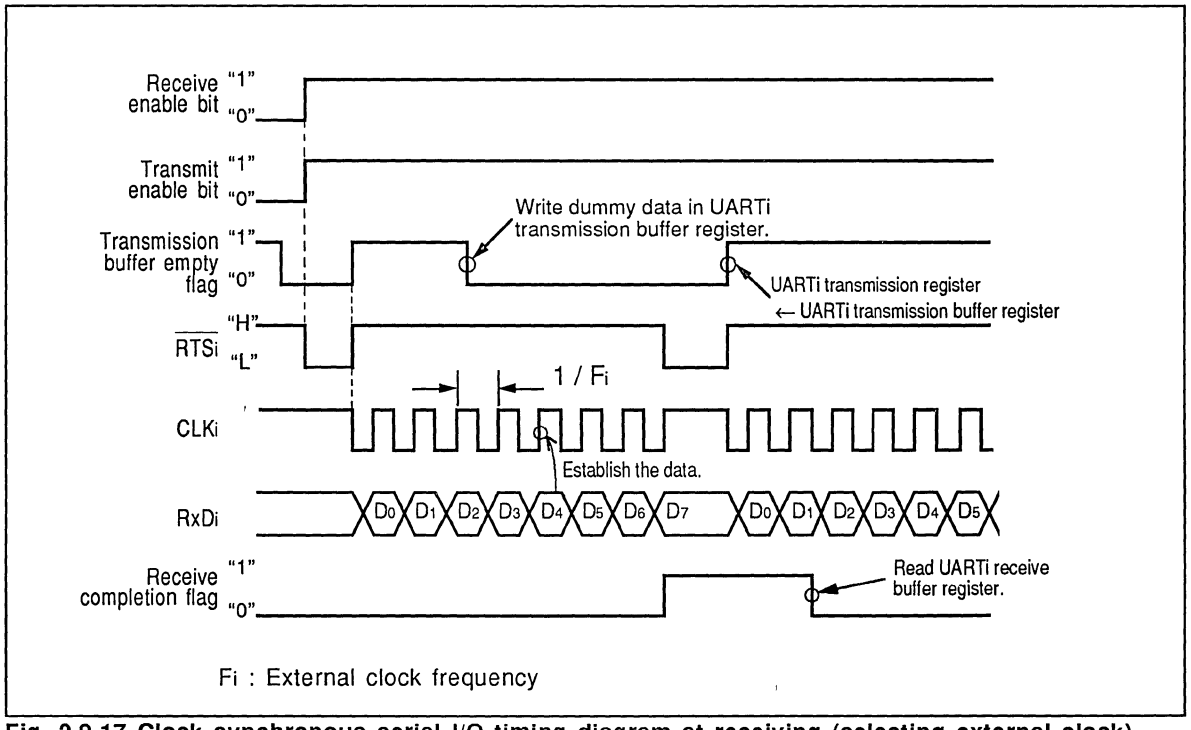


Fig. 2.9.17 Clock synchronous serial I/O timing diagram at receiving (selecting external clock)

[Precaution during clock synchronous serial I/O receive]

1. With clock synchronous serial I/O, shift clocks are generated by operating a transmitter. Therefore, transmission operations must be performed even if only receive is necessary. Also note that dummy data is output from the TxDi pin (transmit pin) during receive.

2. In case that an internal clock is selected, a shift clock is generated when the transmit enable bit is set to "1" (transmit enabled) and a dummy data is set in the UARTi transmission buffer register.
   In case that an external clock is selected, a shift clock is generated when the transmit enable bit is set to "1", a dummy data is set in the UARTi transmission buffer register, and external clock is input to the CLKi pin.

3. When receiving data continuously, an overrun error occurs and bit 4 (overrun error flag) in the UARTi transmit/receive control register 1 is set to "1" if the next receive data becomes available in the UARTi receive register while the receive completion flag is "1" (before reading the content of the UARTi receive buffer register). In this case, the UARTi receive buffer register contains the next data. Therefore, the transmit and receive programs must make arrangements to re-transmit the previous data when an overrun error occurs.
   The interrupt request bit is not set to "1" when an overrun error occurs.

4. When continuously receiving data, a dummy data must be set in the low-order byte of the UARTi transmission buffer register as each receive.

2.9.4 Clock asynchronous serial I/O (UART)

Table 2.9.5 shows the serial I/O performance in UART mode.

Table 2.9.5 UART description

Parameter		Function
Data format	Start bit	1 bit
	Data bit (character length)	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
Baud rate	Internal clock	BRG output divided by 16
	External clock	125Kbps maximum ($f(X_{IN})$=8MHz)
		250Kbps maximum ($f(X_{IN})$=16MHz)
		312.5Kbps maximum ($f(X_{IN})$=25MHz)
Error detection		4 types (overrun, parity, framing, error sum) (Error sum can be used to check existence of error.)

In UART mode, the baud rate*1 and the data format must be set beforehand. The setting of the baud rate and the transfer format are described below.

Baud rate*1 : Frequency of the clock used for transmission and receive.

(1) Transmission rate

The serial data transfer rate is determined by the baud rate. The baud rate is set by the BRG. The BRG is a frequency divider that has 8-bit structure. The BRG input clock can be either an internal clock or an external clock input to the CLKi pin with the internal/external clock selection bit.

When an internal clock is selected, 1/2, 1/16, 1/64, or 1/512 of the $f(X_{IN})$ is selected with the BRG count source selection bits. When an external clock is selected, the clock input to the CLKi pin is input to the BRG.

The clock input to BRG is divided by (n+1) and then by 16 to obtain the baud rate.

Table 2.9.6 shows the baud rate selection table.

If the required baud rate is B (bps), use the following equation to determine the value "n" set in the BRG.

"n" = F_i /(16×B) − 1, where "B"=the required baud rate, "F_i"=the clock frequency input to the BRG

Table 2.9.6 Baud rate selection table (1)

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

Table 2.9.6 Baud rate selection table (2)

Baud rate	f(XIN)=20MHz			f(XIN)=25MHz		
(bps)	fi	BRG value	Actual (bps)	fi	BRG value	Actual (bps)
150	f_{64}	129 (81_{16})	150.24	f_{64}	162 ($A2_{16}$)	149.78
300	f_{64}	64 (40_{16})	300.48	f_{64}	80 (55_{16})	301.41
600	f_{16}	129 (81_{16})	600.96	f_{16}	162 ($A2_{16}$)	599.12
1200	f_{16}	64 (40_{16})	1201.92	f_{16}	80 (55_{16})	1205.63
2400	f_{16}	32 (20_{16})	2367.42	f_{16}	40 (28_{16})	2381.86
4800	f_2	129 (81_{16})	4807.69	f_2	162 ($A2_{16}$)	4792.94
9600	f_2	64 (40_{16})	9615.38	f_2	80 (55_{16})	9645.06
19200	f_2	32 (20_{16})	18939.39	f_2	40 (28_{16})	19054.88
31250	f_2	19 (13_{16})	31250.00	f_2	24 (18_{16})	31250.00

(2) Transfer format

The format of the transfer data is set with the UARTi transmit/receive mode register. Data can be transferred in the following modes shown by Figure 2.9.18.

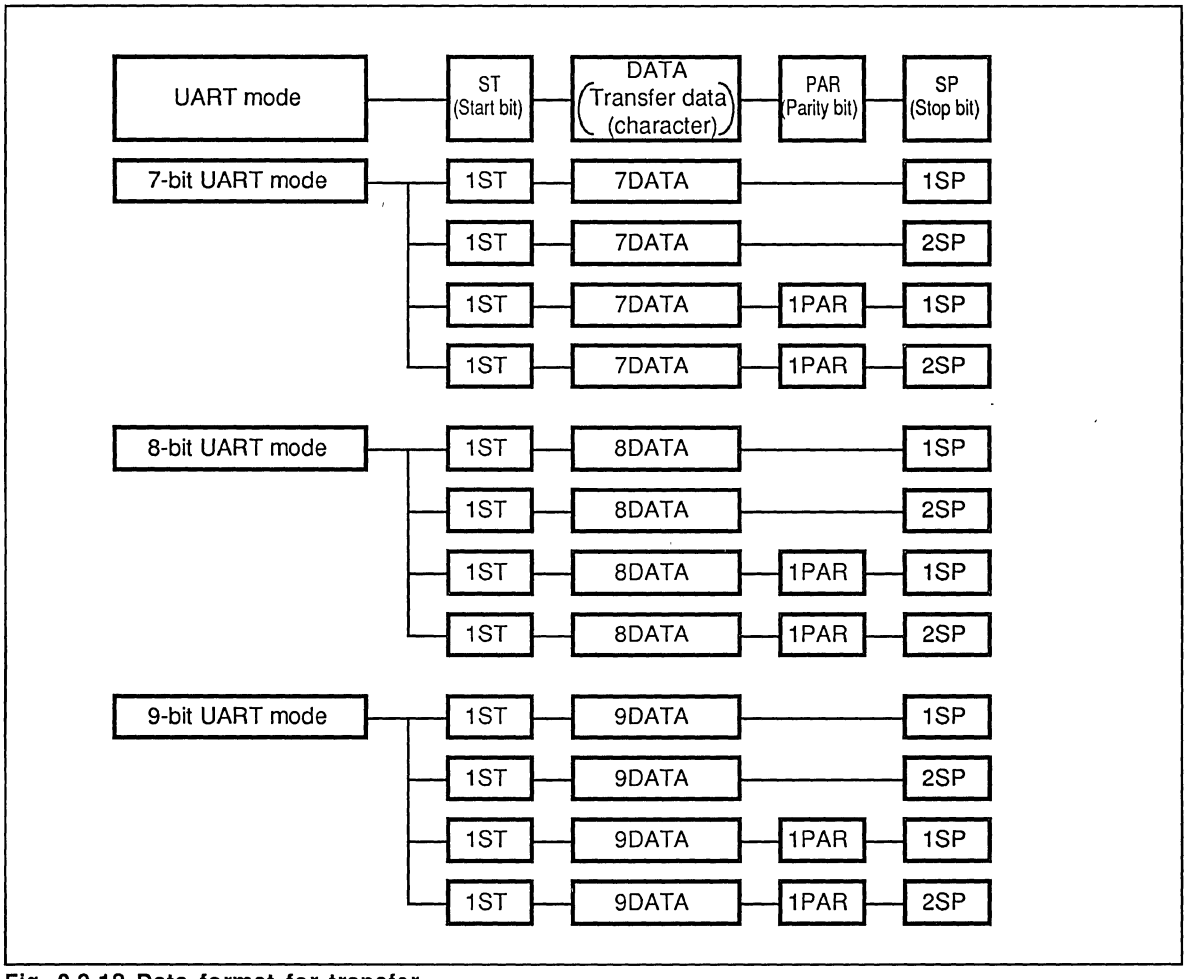


Fig. 2.9.18 Data format for transfer

Figure 2.9.19 shows the data format for example and Table 2.9.7 shows the transfer data in UART mode.

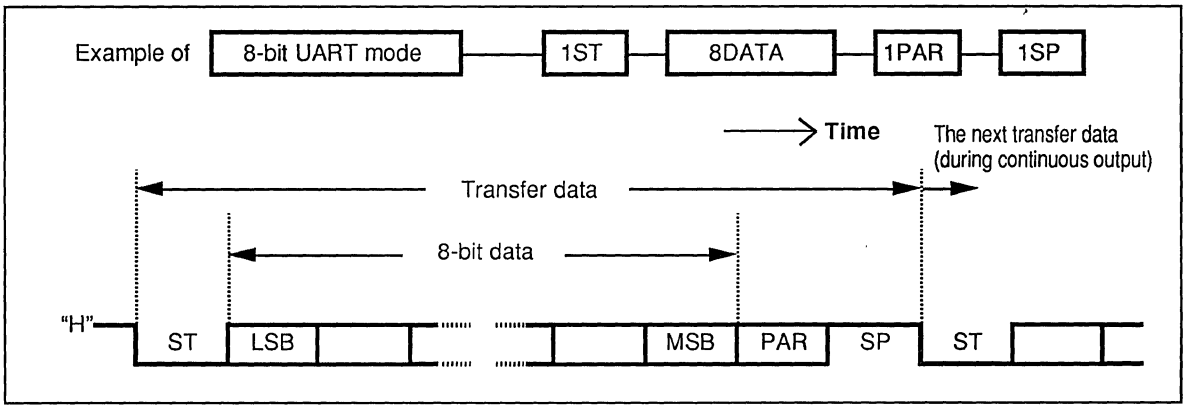


Fig. 2.9.19 Data format example

Table 2.9.7 Transfer data in UART mode

Item	Function
ST (Start bit)	This bit indicates the start of data transmission. A 1-bit "L" signal is appended in front of the transmission data.
DATA (Character)	This is the transmission data written in the UARTi transmission buffer register.
SP (Stop bit)	This bit appends after the data (or after the parity bit if it is included) to indicate the end of transmission. A 1 or 2-bit "H" signal is output as a stop bit.
PAR (Parity bit)	This bit appends to the end of data to improve the reliability of data. This bit is appended so that the number of 1s in the data including the parity bit is always even or odd.

(3) Serial data transmission

The data transmission method in UART mode is described below.

[Setting the control registers]

Set each serial I/O control register for transmission.

●UARTi transmit/receive mode register

•Serial I/O mode selection bits

Select the data length with the bits 0 to 2.

Table 2.9.8 Setting of UART mode

b2	b1	b0	UART operation mode
1	0	0	7-bit UART mode
1	0	1	8-bit UART mode
1	1	0	9-bit UART mode

•Transfer format

Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.

•Internal/external clock selection bit

Select either an internal clock ("0") or an external clock ("1") as the BRG count source.

•Sleep function selection bit

Set the bit 7 to "1" when enable the sleep function, and to "0" when disable it.
(See "2.9.5 Sleep mode" for details of sleep mode.)

●UARTi transmit/receive control register 0

•BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for BRG input clock.

•$\overline{CTS}/\overline{RTS}$ function selection bit

Set the bit 2 to "0" when using $\overline{CTS}$ function, and to "1" when not use.

●UARTi baud rate generator(BRG)

•Dividing ratio

Set the BRG value between 00_{16} and FF_{16}.

●Port P8 direction register

•Port direction selection bits

Set the corresponding bit to "0" when the $\overline{CTS}$ function is selected and an external clock is selected.

●UARTi transmission interrupt control register

•Interrupt priority level selection bits

When using UARTi transmission interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTi transmission buffer register

•Transfer data

Set a transfer data to the UARTi transmission buffer register. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTi transmit/receive control register 1

•Transmit enable bit

Set the bit 0 to "1" to enable transmitting.

Figure 2.9.20 shows the setting example of UART related registers at transmitting.

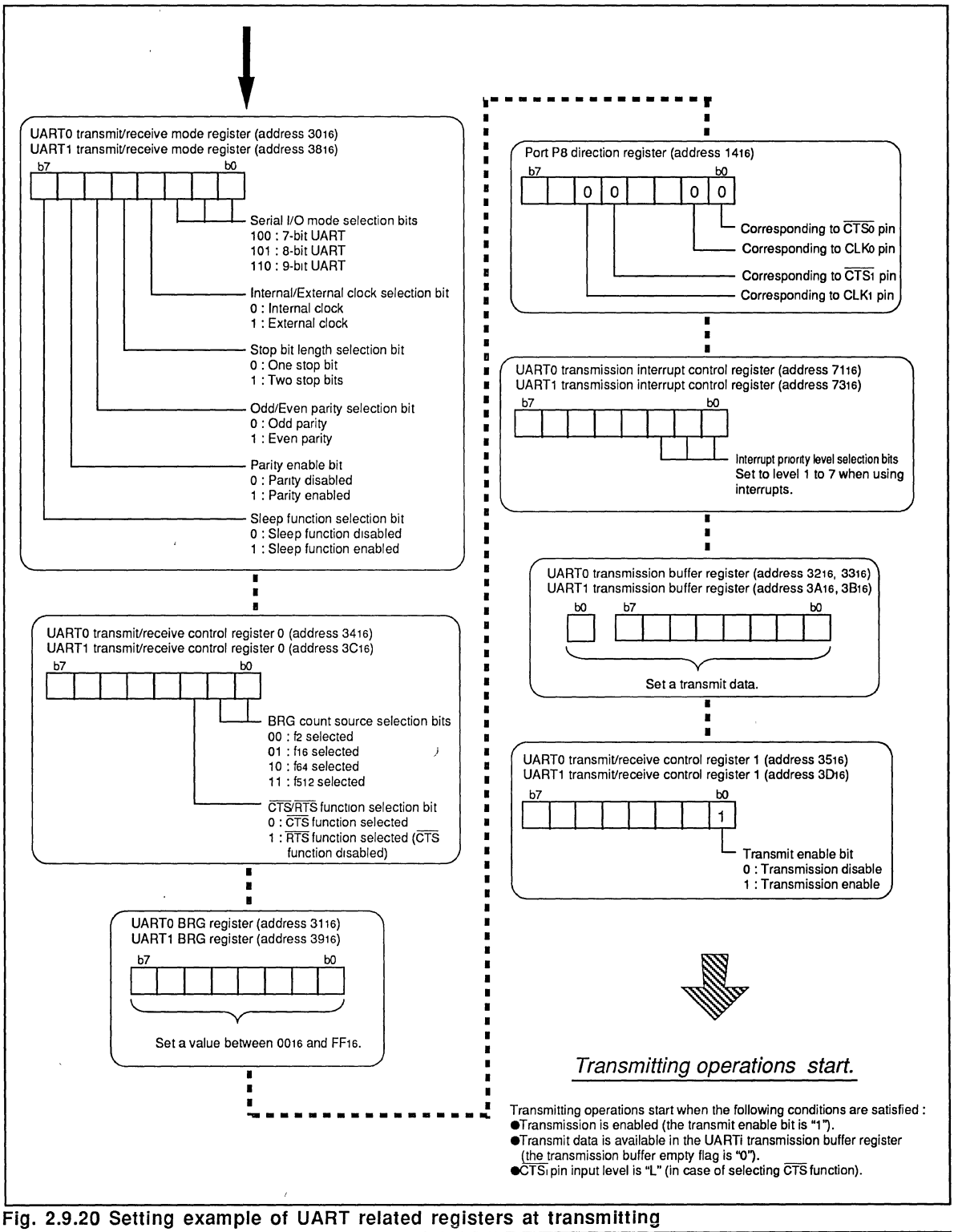


Fig. 2.9.20 Setting example of UART related registers at transmitting

[Transmit operation]

The only difference between 7-bit UART, 8-bit UART, and 9-bit UART is the length of the transmitted data. The low-order byte of the UARTi transmission buffer register is used for 7-bit and 8-bit UART. The low-order byte and bit 0 of the high-order byte is used for 9-bit UART.

The transmission of serial data starts when the following conditions are satisfied :

① Transmission is enabled (the transmit enable bit is "1").
② Transmit data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").
③ $\overline{CTS_i}$ pin input level is "L".
(**Note**: This condition is ignored if the $\overline{CTS}$ function is not selected.)

When the above three (① to ③) conditions are satisfied (two (① and ②) if $\overline{CTS}$ function is not selected), the following operations are performed automatically at the same time :

- Transfer the content of UARTi transmission buffer register to the UARTi transmission register.
- Set the transmission buffer empty flag to "1".
- Clear the transmission register empty flag to "0".
- UARTi transmission interrupt request occurs and set the interrupt request bit to "1".

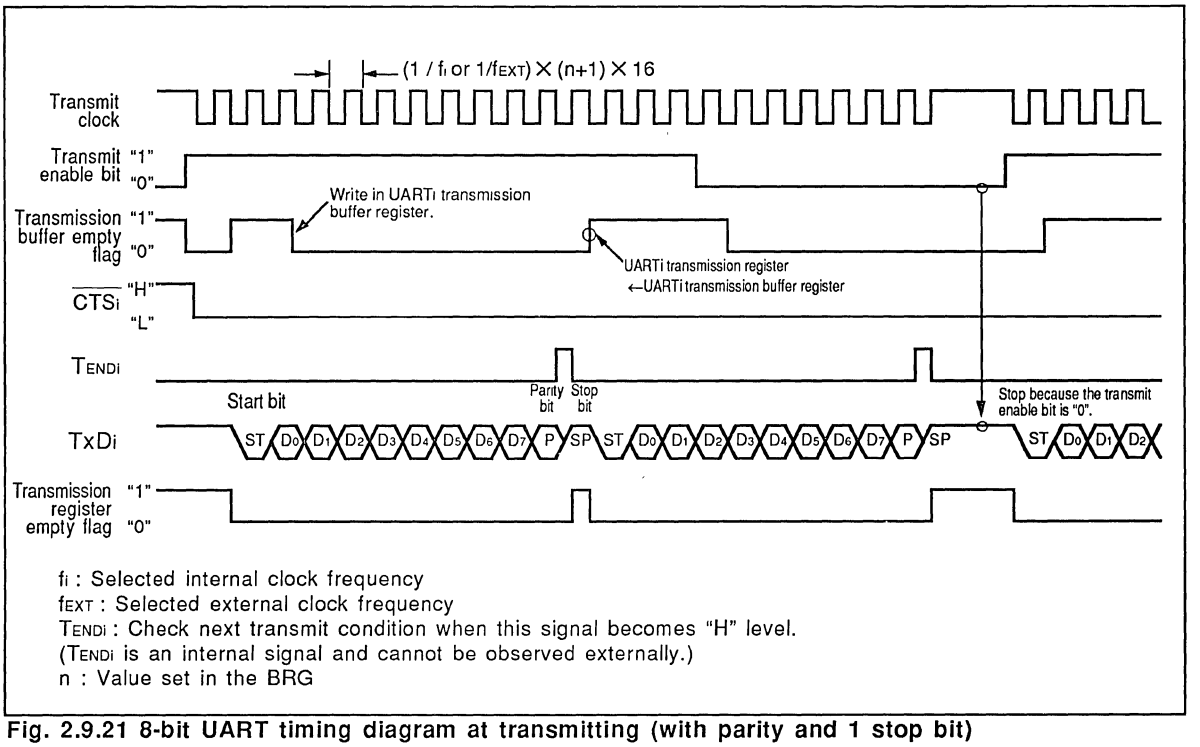
Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information.

Data transmission starts from the TxDi pin when the data is transferred to the UARTi transmission register. When transmission starts, data is output from the TxDi pin in the format specified by the UARTi transmit/receive mode register. The data is output bit by bit in the order;

ST→DATA(LSB)→...→DATA(MSB)→PAR→SP.

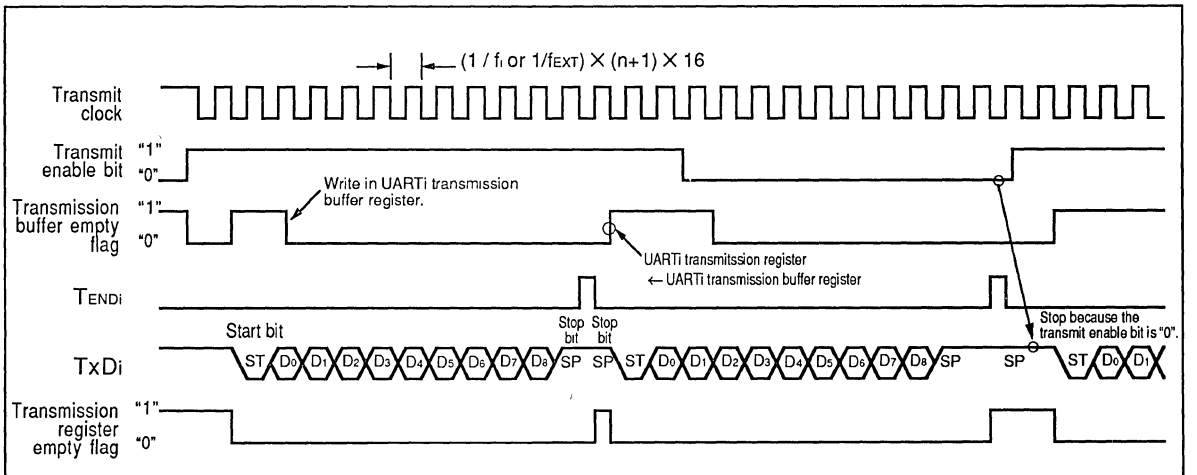
After the stop bit has been output, the transmission register empty flag is set to "1" to indicate that the transmission has completed. If the next data is available when transmission completes, a start bit is generated following the stop bit and the next data is transmitted. In order to continuously transfer data, the next transmission data must be set in the UARTi transmission buffer register during transmitting operations (when the transmission register empty flag is "0"). If the transmit conditions for the next data is not satisfied, "H" level is output from the TxDi pin.

Figure 2.9.21 shows the timing diagram of 8-bit UART at transmitting (with parity, 1 stop bit and $\overline{CTS}$ function). Figure 2.9.22 shows the timing diagram of 9-bit UART (with 2 stop bits, no parity and no $\overline{CTS}$ function).



fi : Selected internal clock frequency
fEXT : Selected external clock frequency
TENDi : Check next transmit condition when this signal becomes "H" level.
(TENDi is an internal signal and cannot be observed externally.)
n : Value set in the BRG

Fig. 2.9.21 8-bit UART timing diagram at transmitting (with parity and 1 stop bit)



fi : Selected internal clock frequency
fEXT : Selected external clock frequency
TENDi : Check next transmit condition when this signal becomes "H" level.
(TENDi is an internal signal and cannot be observed externally.)
n : Value set in the BRG

Fig. 2.9.22 9-bit UART timing diagram at transmitting (no parity and 2 stop bits)

(4) Serial data receive

The data receive method in UART mode is described below.

[Setting the control registers]

Set each serial I/O control register for receive.

❶UARTi transmit/receive mode register

Match the format with the transmitting side.

•Serial I/O mode selection bits

Select the data length with the bits 0 to 2.

Table 2.9.9 Setting of UART mode

b2	b1	b0	UART operation mode
1	0	0	7-bit UART mode
1	0	1	8-bit UART mode
1	1	0	9-bit UART mode

•Transfer format

Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.

•Internal/external clock selection bit

Select either an internal clock ("0") or an external clock ("1") as the BRG count source.

•Sleep function selection bit

Set the bit 7 to "1" when enable the sleep function, and to "0" when disable it.
(See "2.9.5 Sleep mode" for details of sleep mode.)

❷UARTi transmit/receive control register 0

•BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for BRG input clock.

•$\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit

Set the bit 2 to "1" when using $\overline{\text{RTS}}$ function, and to "0" when not use.

❸UARTi baud rate generator(BRG)

•Dividing ratio

Set the BRG value between 00_{16} and FF_{16}.

❹Port P8 direction register

•Port direction selection bits

Set the corresponding bit to "0" to the TxDi pin and the CLKi pin when an external clock is selected.

❺UARTi receive interrupt control register

•Interrupt priority level selection bits

When using UARTi receive interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

❻UARTi transmit/receive control register 1

•Receive enable bit

Set the bit 2 to "1" to enable receiving.

Figure 2.9.23 shows the setting example of UART related registers at receiving.

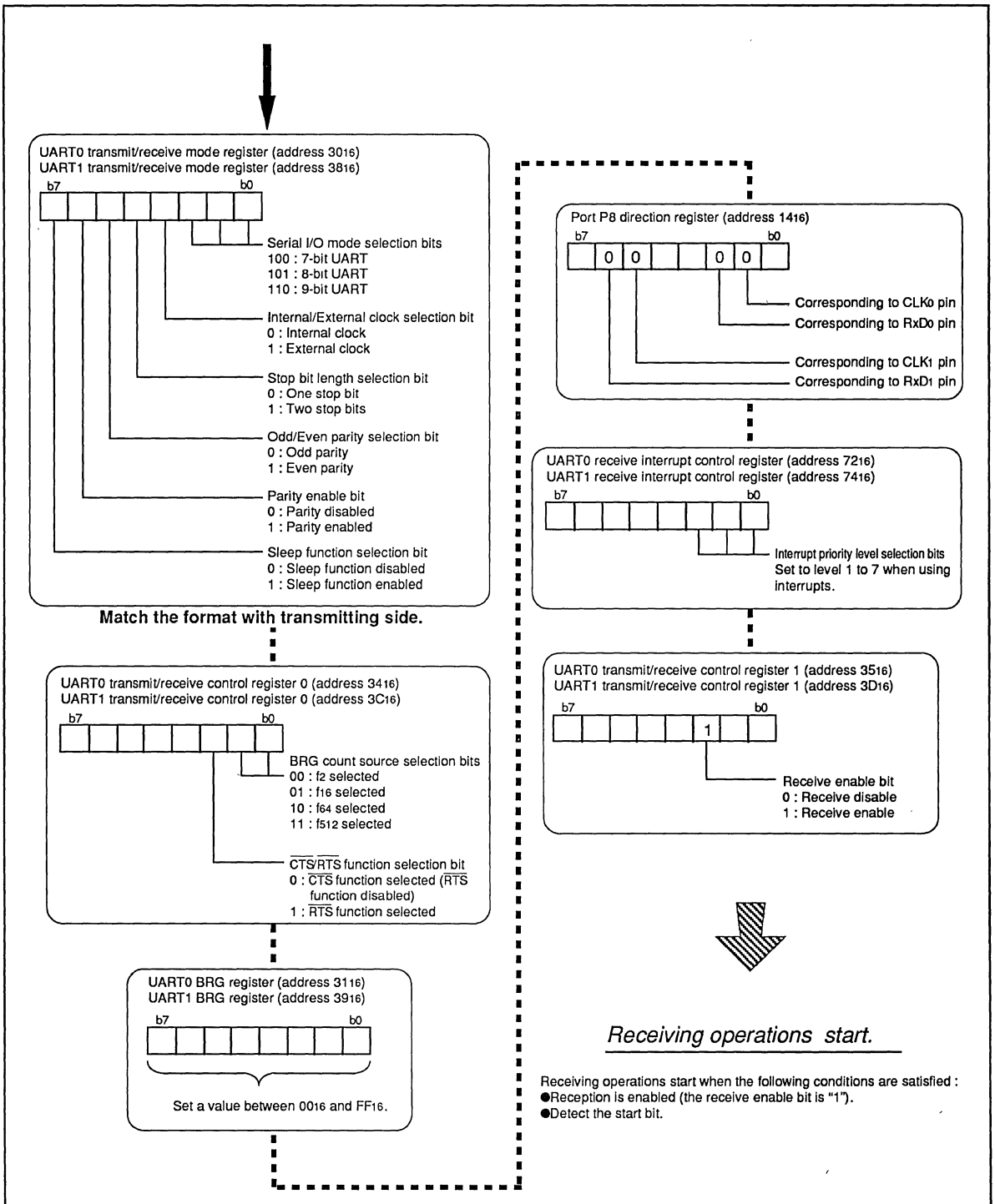


Fig. 2.9.23 Setting example of UART related registers at receiving

[Receive operation] (when using an external clock)

The reception of serial data starts when the following conditions are satisfied :

① Reception is enabled (the receive enable bit is "1").
② Detect the start bit.

Serial data receive is enabled by setting the receive enable bit to "1". When the receive enable bit is set to "1", the $\overline{RTSi}$ pin becomes "L" level to indicate externally that the microcomputer is ready to receive serial data (in case that $\overline{RTS}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{RTSi}$ output pin to the $\overline{CTSi}$ pin on the transmit side. Figure 2.9.24 shows the connecting example of UART.

When the RxDi pin detects a start bit, a receive clock is generated and data receive starts. At the same time, the $\overline{RTSi}$ pin returns to "H" level if $\overline{RTS}$ function is selected. The RxDi pin level is used to establish the most significant bit of the UARTi receive register at the rising edge of the receive clock and the content of UARTi receive register is shifted by 1 bit to the right. This operation is repeated to receive the entire data from ST to SP. Then the content of UARTi receive register is transferred to the UARTi receive buffer register. At the same time, the receive completion flag is set to "1". When the receive completion flag is set to "1" by detecting SP, the UARTi receive interrupt request occurs and the interrupt request bit is set to "1". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The receive completion flag is cleared to "0" when the UARTi receive buffer register is read.

Figure 2.9.25 shows the timing diagram of 8-bit UART at receive (no parity, with 1 stop bit and $\overline{RTS}$ function).

When receiving data continuously, an overrun error occurs and the bit 4 (overrun error flag) in the UARTi transmit/receive control register 1 is set to "1" if the next receive data becomes available in the UARTi receive register while the receive completion flag is "1" (before reading the content of the UARTi receive buffer register). In this case, the next data is written in the UARTi receive buffer register. Therefore, if an overrun occurs, the transmit and receive programs must make arrangements to re-transmit the data. The interrupt request bit is not set to "1" when an overrun error occurs.

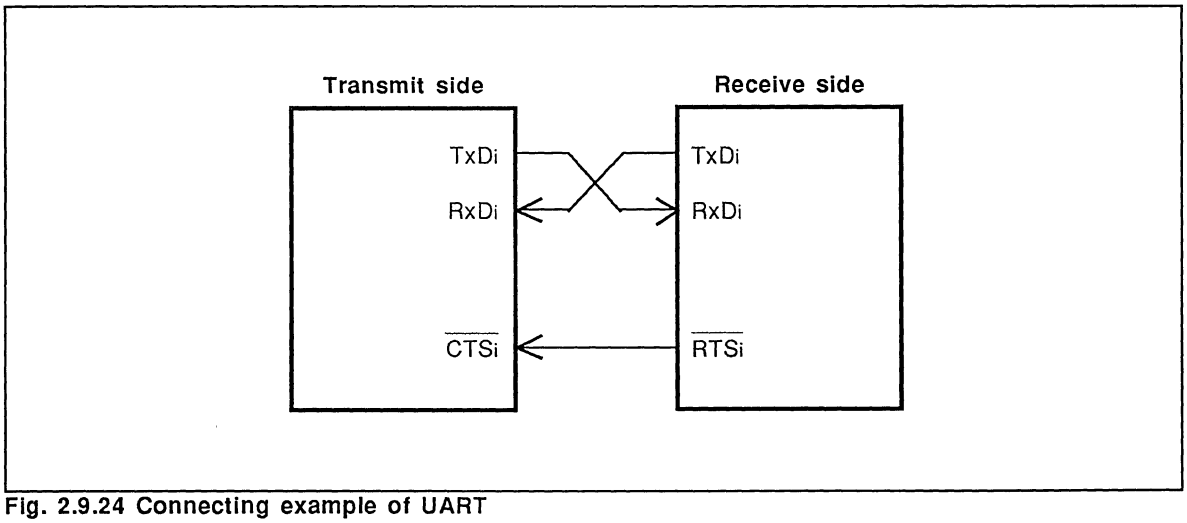


Fig. 2.9.24 Connecting example of UART

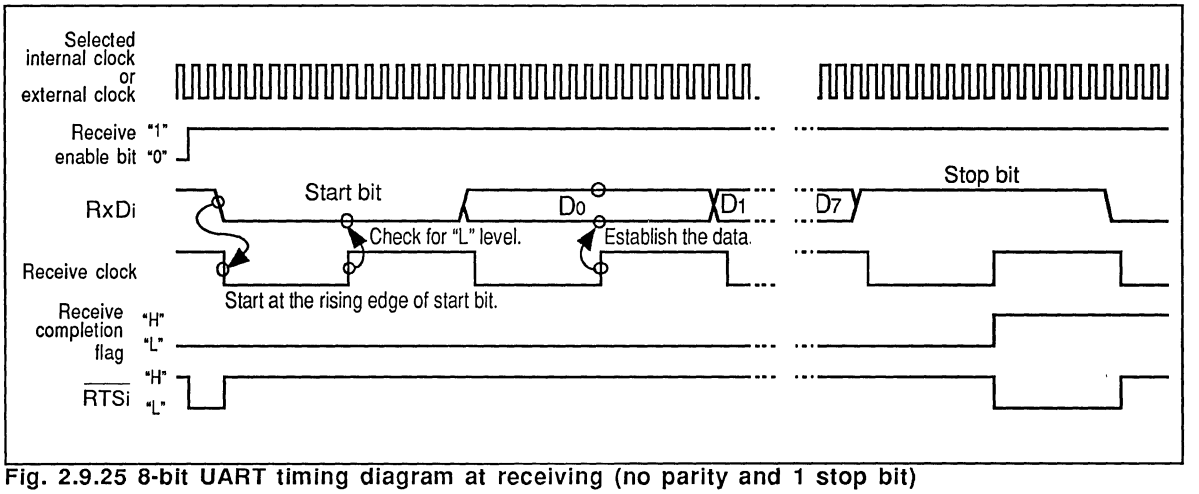


Fig. 2.9.25 8-bit UART timing diagram at receiving (no parity and 1 stop bit)

[Error flag]

During UART mode operation, transfer data errors can be detected using four error flags. These errors are detected when transferring data from the UARTi receive register to the UARTi receive buffer register. The error flags are cleared to "0" when the low-order byte of the UARTi receive buffer register is read or when the receive enable bit is set to "0".

●Overrun error

An overrun error occurs and the overrun error flag is set to "1" when the next receive data becomes available in the UARTi receive register and transferred to the UARTi receive buffer register while the receive completion flag is "1" (data exists in UARTi receive buffer register), or the next receive data becomes available before the content of the UARTi receive buffer register is read.

●Framing error

A framing error occurs and the framing error flag is set to "1" when there is insufficient number of stop bits.

●Parity error

A parity error occurs and the parity error flag is set to "1" when parity checking is enabled and the number of 1s in the data including the parity bit conflicts with the parity specified by the odd/even parity selection bit.

●Sum error

The error sum flag is set to "1" when either an overrun error, a framing error, or a parity error occurs. The existence of errors can be determined by checking the error sum flag.

2.9.5 Sleep mode

Sleep mode is used for communication between certain microcomputers when multiple microcomputers are connected through serial I/O.

Sleep mode is entered by setting the UARTi transmit/receive mode register bit 7 to "1". In sleep mode, the content of UARTi receive register is not transferred to the UARTi receive buffer register when the most significant bit (MSB: bit 8 if 9-bit UART mode, bit 7 if 8-bit UART mode, and bit 6 if 7-bit UART mode) of the received data is "0". In this case, the receive completion flag and the error flags remain unchanged and no receive interrupt request occurs. Normal receive operation is performed only when the most significant bit of the received data is "1".

The following is a description of sleep mode usage in 8-bit UART mode. The main microcomputer first transmits a data with bit 7 set to "1" and the remaining bits 0 to 6 forming the address of the destination microcomputer. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data and sets the sleep function selection bit to "0" if the address matches its own address and to "1" if otherwise. Next the main microcomputer starts transmitting data with bit 7 set to "0". Then only the microcomputer with the sleep function selection bit set to "0" will receive this data. This enables communication between the main microcomputer and a specific subordinate microcomputer. Figure 2.9.26 shows the sleep mode.

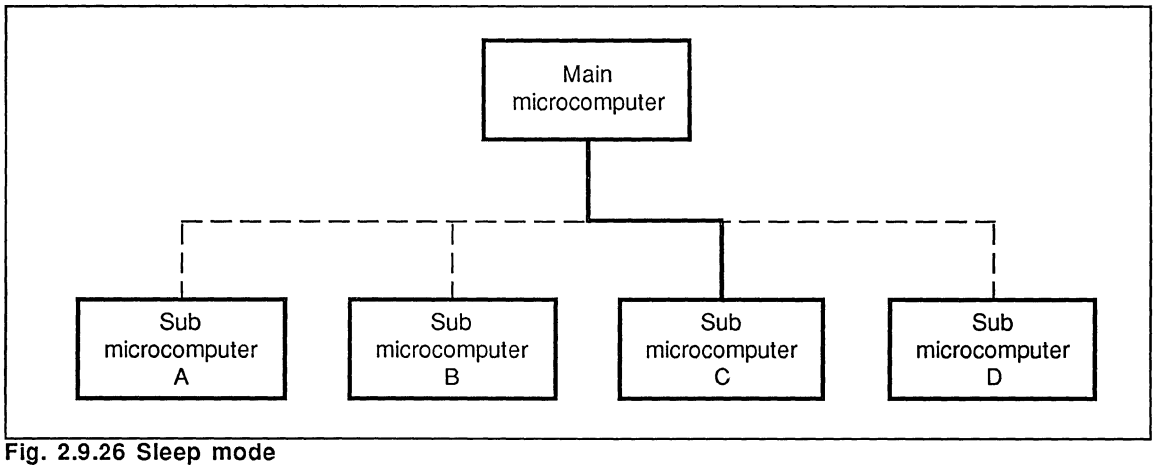


Fig. 2.9.26 Sleep mode

2.10 A-D converter

The M37702 has a built-in 8-bit A-D converter that performs successive approximation to convert analog values input from pins AN_0 – AN_7 to digital values.
The A-D converter provides four selectable conversion modes.

2.10.1 A-D converter overview

Table 2.10.1 shows a performance overview of the A-D converter.

Table 2.10.1 A-D converter performance overview

Parameter	Description
Analog input pin	8 pins (AN_0 to AN_7)
A-D conversion mode	One-shot mode
	Repeat mode
	Single sweep mode
	Repeat sweep mode
A-D conversion method	Successive approximation
Resolution	8 bits
Absolute accuracy	±3LSB
Conversion speed	57ϕ_{AD} cycles ϕ_{AD} : A-D converter operating clock (for 1 analog input pin)

The A-D converter provides the following four conversion modes.

- ●One-shot mode ..... The input voltage to the selected analog input pin is converted. After conversion, the result is stored in the corresponding A-D register and an A-D conversion interrupt request occurs.
- ●Repeat mode ..... The input voltage to the selected analog input pin is repeatedly converted. The results are stored in the corresponding A-D register, but no A-D conversion interrupt request occurs.
- ●Single sweep mode ..... The analog input pins to be converted can be selected with the A-D sweep pin selection register. The selected pins are converted in the order AN_0, AN_1,... and an A-D conversion interrupt request occurs when the last pin is converted. The result is stored in the corresponding A-D register when each pin is converted.
- ●Repeat sweep mode ..... This is similar to single sweep mode except that conversion is repeated in order from the AN_0 pin without an interrupt request after converting the last pin.

2.10.2 Block description

Figure 2.10.1 shows the block diagram of the A-D converter. The A-D converter related registers are described below.

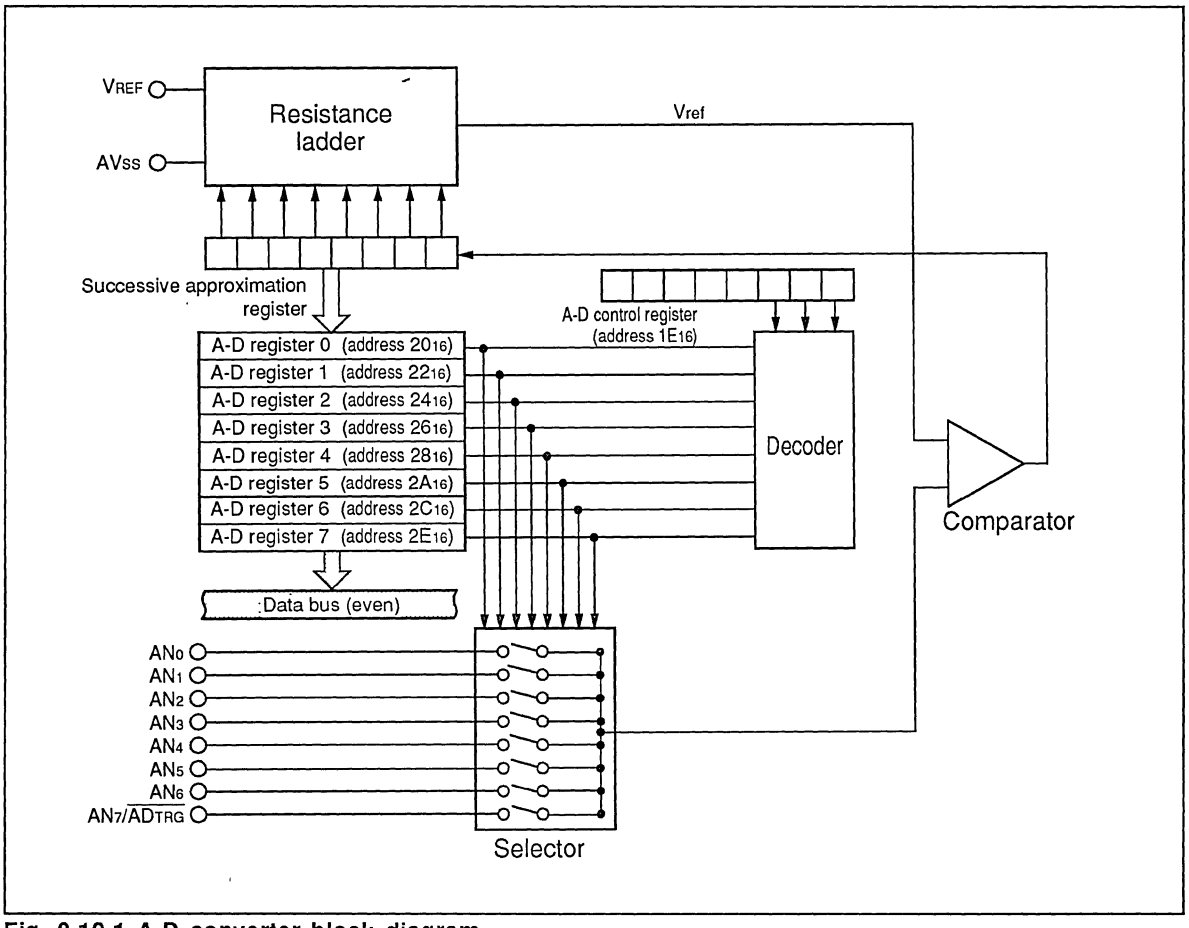
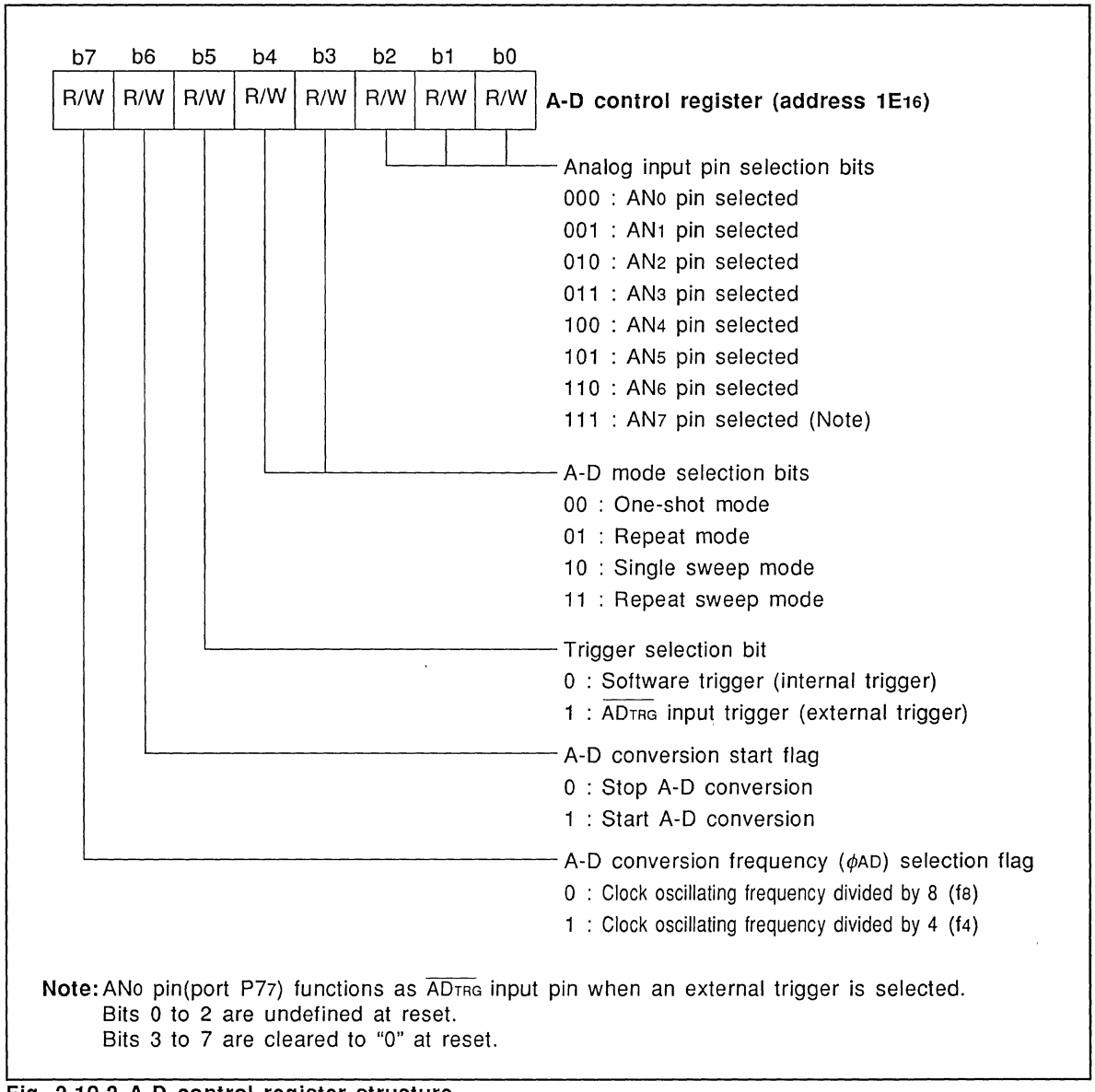


Fig. 2.10.1 A-D converter block diagram

(1) A-D control register

The A-D control register ($1E_{16}$) consists of bits that control the A-D converter. Figure 2.10.2 shows the structure of the A-D control register followed by description of each bit.



Note: AN0 pin(port P77) functions as $\overline{ADTRG}$ input pin when an external trigger is selected.
Bits 0 to 2 are undefined at reset.
Bits 3 to 7 are cleared to "0" at reset.

Fig. 2.10.2 A-D control register structure

●Analog input selection bits (bits 0 to 2)

The analog input selection bits are used to select the analog input pin in one-shot mode and repeat mode. The selected analog input pin remains unchanged when the mode is switched between one-shot mode and repeat mode. These bits are ignored in single sweep mode and repeat sweep mode. If an external trigger is selected with the trigger selection bit (described later), port $P7_7$ functions as $\overline{AD_{TRG}}$ pin and cannot be used as AN_7 pin.

Table 2.10.2 shows the relationship between the analog input selection bits and analog input pins.

Table 2.10.2 Relationship between analog input pin selection bits and analog input pins

b2	b1	b0	Analog input pin
0	0	0	AN_0 pin selected
0	0	1	AN_1 pin selected
0	1	0	AN_2 pin selected
0	1	1	AN_3 pin selected
1	0	0	AN_4 pin selected
1	0	1	AN_5 pin selected
1	1	0	AN_6 pin selected
1	1	1	AN_7 pin selected

●A-D mode selection bits (bits 3, 4)

The A-D mode selection bits are used to select one of the four available conversion modes. Table 2.10.3 shows the relationship between the A-D mode selection bits and the conversion modes.

Table 2.10.3 Relationship between A-D mode selection bits and conversion modes

b4	b3	A-D conversion mode
0	0	One-shot mode
0	1	Repeat mode
1	0	Single sweep mode
1	1	Repeat sweep mode

OTrigger selection bit (bit 5)

The trigger selection bit is used to select the trigger occurrence factor which start an A-D conversion operation. An internal trigger or an external trigger is available for the trigger. An internal trigger (software trigger) is selected when this bit is "0" and an external trigger (input signal to $\overline{AD_{TRG}}$ pin) is selected when this bit is "1".

<Internal trigger>

A trigger is generated and A-D conversion starts when the A-D conversion start flag (described later) is set to "1".

<External trigger>

A trigger is generated when the level of the signal input to the $\overline{AD_{TRG}}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is "1". When an external trigger is selected, a retrigger is available during A-D conversion. In this case, the conversion is repeated from the beginning. The $\overline{AD_{TRG}}$ pin is shared with the AN_7 (port $P7_7$) pin. Therefore, the AN_7 pin cannot be used as analog input pin when an external trigger is selected.

When an external trigger is selected in each A-D conversion mode, the port P7 direction register bit 7 must be set to "0" (input mode).

●**A-D conversion start flag (bit 6)**

The A-D conversion start flag is used to start or stop A-D conversion.

<Internal trigger>

An internal trigger is generated and A-D conversion starts when the A-D conversion start flag is set to "1". A-D conversion stops when it is cleared to "0". This bit is automatically cleared after A-D conversion in one-shot mode and single sweep mode. It is not cleared in other modes and conversion continues until it is cleared to "0".

<External trigger>

The A-D conversion start flag must be set to "1" before the falling edge is input to the $\overline{AD_{TRG}}$ pin. If external trigger is selected, this flag is not cleared to "0" after conversion.

●**A-D conversion frequency selection flag (bit 7)**

This flag is used to select the A-D converter operating frequency (ϕ_{AD}). When this flag is "0", the clock frequency f(X_{IN}) divided by 8 is selected. When this flag is "1", the clock frequency f(X_{IN}) divided by 4 is selected.

In one-shot mode and repeat mode, A-D conversion completes after 57 x ϕ_{AD} cycles from the beginning of A-D conversion. In single sweep and repeat sweep mode, A-D conversion completes after 57 x number of selected pins x ϕ_{AD} cycles from the beginning of A-D conversion.

The A-D converter operating clock ϕ_{AD} during A-D conversion must be no less than 250kHz because the comparator in the A-D conversion circuit consists of capacity coupling amplifiers.

Table 2.10.4 shows the relationship between the A-D conversion frequency selection flag and the A-D converter operating clock and conversion time.

Table 2.10.4 Relationship between A-D conversion frequency selection flag and the A-D converter operating clock and conversion time

A-D frequency selection flag		"0"	"1"
A-D converter operating clock		$\phi_{AD}=\frac{f(X_{IN})}{8}$	$\phi_{AD}=\frac{f(X_{IN})}{4}$
Conversion time (Note)	f(X_{IN})=8MHz	57.0 µs	28.5 µs
	f(X_{IN})=16MHz	28.5 µs	14.25 µs
	f(X_{IN})=25MHz	18.24 µs	9.12 µs

Note. Conversion time per one analog input pin

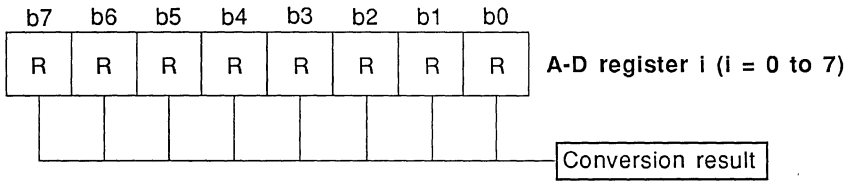
(2) A-D register i (i=0 to 7)

The A-D registers (address 20_{16} to $2E_{16}$) are 8-bit read only registers. The conversion results (content of successive approximation register) are stored in these registers. Each A-D register corresponds to an analog input pin. The content of the A-D register can be read during A-D conversion. However, if the A-D register corresponding to the analog input being converted is read, the previous conversion result is obtained.

Table 2.10.5 shows the relationship between the analog input pin and A-D register and Figure 2.10.3 shows the structure of A-D register.

Table 2.10.5 Relationship between analog input pin and A-D register

Analog input pin	Register containing the result	Address
AN_0 pin	A-D register 0	20_{16}
AN_1 pin	A-D register 1	22_{16}
AN_2 pin	A-D register 2	24_{16}
AN_3 pin	A-D register 3	26_{16}
AN_4 pin	A-D register 4	28_{16}
AN_5 pin	A-D register 5	$2A_{16}$
AN_6 pin	A-D register 6	$2C_{16}$
AN_7 pin	A-D register 7	$2E_{16}$



Note: This register is undefined at reset.

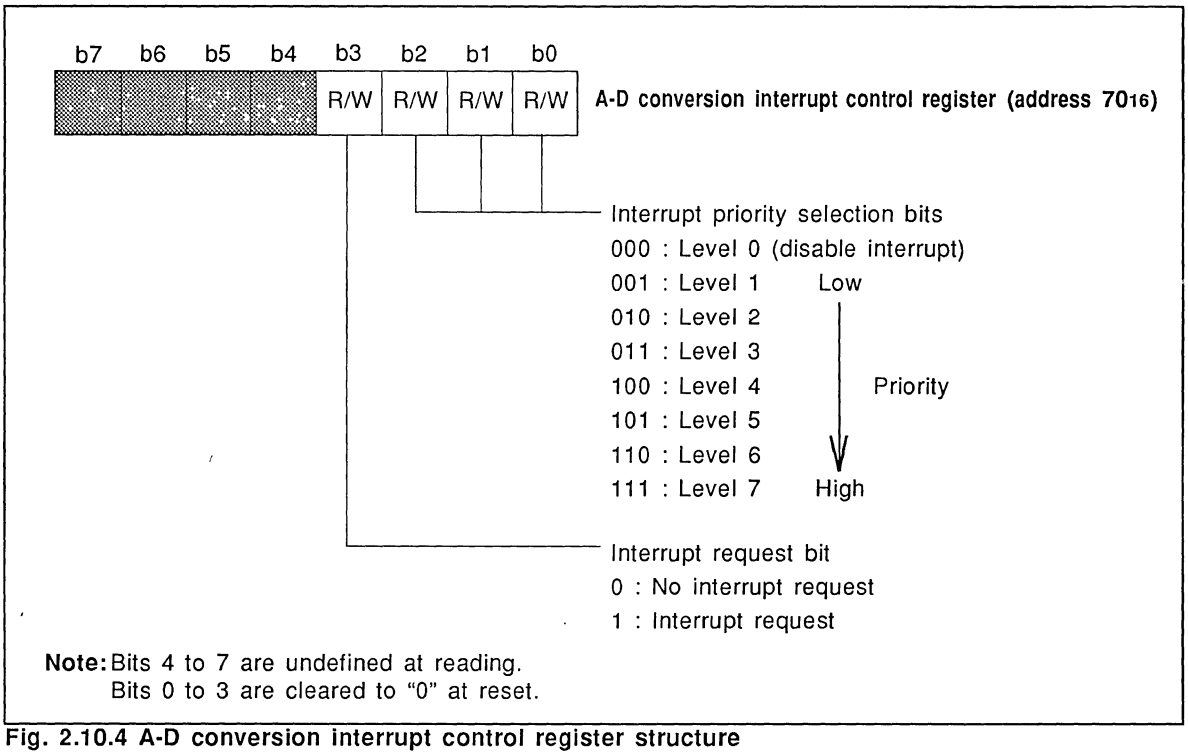
Fig. 2.10.3 A-D register i structure

(3) Comparator and successive approximation register

The compare reference voltage V_{ref} and analog input voltage V_{IN} are compared for bits 7 to 0 of successive approximation register and the result is set in each bit. Comparison starts from bit 7 and the contents of this register (conversion result) is transferred to A-D register after comparing bit 0. The content of the successive approximation register changes according to the comparison result of each bit. Therefore, the compare reference voltage V_{ref} also changes according to the content of the successive approximation register. The analog input voltage V_{IN} is selected by the decoder (refer to section "2.10.3 Successive approximation conversion").

(4) **A-D conversion interrupt control register**

The A-D conversion interrupt control register (address 70_{16}) consists of interrupt priority selection bits and interrupt request bit. Figure 2.10.4 shows the structure of the A-D conversion interrupt control register. Refer to section "2.6 Interrupts" for more information concerning interrupts.



Note: Bits 4 to 7 are undefined at reading.
Bits 0 to 3 are cleared to "0" at reset.

Fig. 2.10.4 A-D conversion interrupt control register structure

●**Interrupt priority selection bits (bits 0 to 2)**

These bits are used to select the interrupt priority level. They should be set to a level between 1and 7 when using an A-D conversion interrupt. When there is an interrupt request, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is greater than IPL (interrupt disable flag I must be "0"). Set these bits to "000" to disable only A-D conversion interrupt.

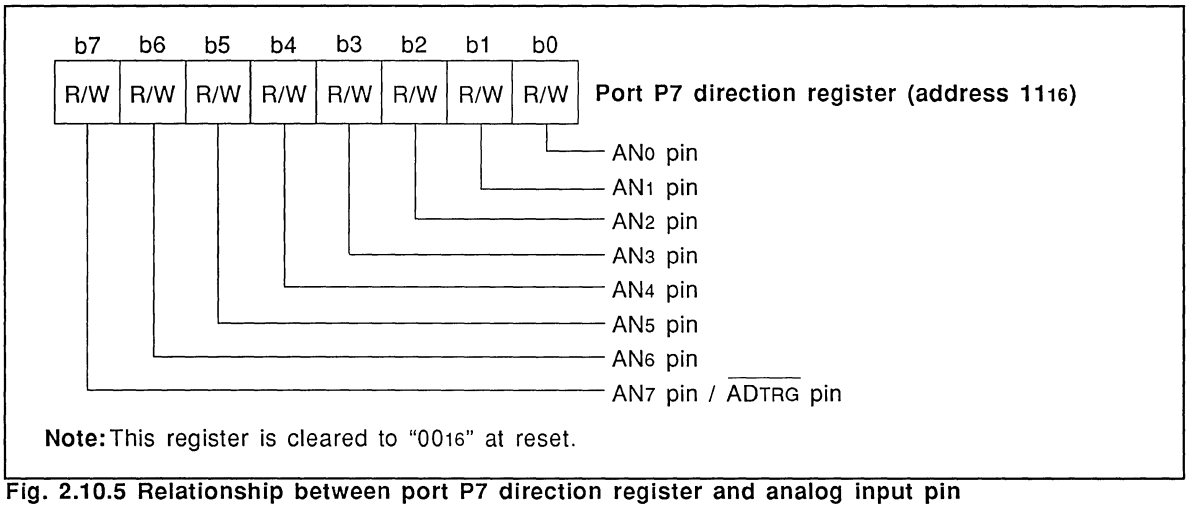
●**Interrupt request bit (bit 3)**

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

(5) Port P7 direction register

The analog input pin is shared with port P7. When using these ports as analog input pins or using port $P7_7$ as external trigger input pin, the corresponding bit in the port P7 direction register must be set to "0" (input mode).

Figure 2.10.5 shows the relationship between the port P7 direction register (11_{16}) and analog input pins.



Note: This register is cleared to "00$_{16}$" at reset.

Fig. 2.10.5 Relationship between port P7 direction register and analog input pin

2.10.3 Successive approximation conversion

A-D conversion starts when an internal or external trigger is generated.
A-D conversion is performed by successive approximation. When A-D conversion starts, the following operations are performed to convert analog values to digital values.

①Initialization of successive approximation register
The successive approximation register is cleared to "00_{16}".

②Setting the most significant bit (bit 7)
The successive approximation register bit 7 is set to "1". Then the reference voltage V_{ref} is compared with the input voltage V_{IN} and bit 7 changes as follows:

Unchanged if $V_{ref} < V_{IN}$
Cleared to "0" if $V_{ref} > V_{IN}$

The reference voltage input to V_{REF} pin must be set between AV_{SS} and AV_{CC}.
The compare reference voltage V_{ref} depends on the value in the successive approximation register. Table 2.10.6 shows the relationship between V_{ref} and the value in the successive approximation register.

Table 2.10.6 Relationship between successive approximation register and V_{ref}

Content of successive approximation register	0	1 to 255
Comparison reference voltage V_{ref} (V)	0	$V_{REF}/256 \times (n-0.5)$ n : The content of the successive approximation register

Step ② above is repeated for all bits from bit 7 to bit 0 and the value in the successive approximation register (digital equivalent of the analog input voltage) is stored in the A-D register when comparison of bit 0 completes.
Table 2.10.7 shows the change in the successive approximation register and compare voltage during A-D conversion.

Table 2.10.7 Change in successive approximation register and compare voltage during A-D conversion

	Successive approximation register	Compare voltage Vref
Conversion start	b7 b0 : 0 0 0 0 0 0 0 0	0 [V]
First comparison	1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$ [V]
Second comparison	n_7 1 0 0 0 0 0 0 (1st comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$ [V]
Third comparison	n_7 n_6 1 0 0 0 0 0 (2nd comparison result)	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$ [V]
Eighth comparison	n_7 n_6 n_5 n_4 n_3 n_2 n_1 1	
Conversion completed	n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0	

2.10.4 A-D conversion mode

Four different A-D conversion modes can be selected with the A-D mode selection bits. In each mode, the trigger selection bit is used to determine whether to use a software trigger (internal trigger) or an external input signal (external trigger).

Error is magnified if the input voltage to the analog input pin changes during A-D conversion. Make sure the input voltage to the analog input pin does not change during the A-D conversion interval which is ϕ_{AD} x 57 cycles (ϕ_{AD} is $f(X_{IN})/8$ or $f(X_{IN})/4$) per pin.

Each conversion mode is described below for case selecting an internal trigger and an external trigger.

(1) One-shot mode [A-D control register bits 4, 3="00"]

In one-shot mode, the input voltage to the one analog input pin selected with the analog input selection bits is converted and an A-D interrupt request occurs when conversion completes. The analog input pin must be selected before generating the trigger. Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is generated and A-D conversion starts. After 57 cycles of ϕ_{AD}, A-D conversion completes, the content of the successive approximation register (converted result) is transferred to the A-D register. At the same time, the A-D interrupt request occurs and the A-D interrupt request bit is set to "1". Then the A-D conversion start flag is cleared to "0" and A-D converter operation stops.

●When an external trigger is selected to start A-D conversion

A-D conversion starts when the input level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1" . When A-D conversion completes after 57 cycles of ϕ_{AD}, the content of the successive approximation register (converted result) is transferred to the A-D register. At the same time, the A-D interrupt request occurs and the A-D interrupt request bit is set to "1". At this point, the A-D conversion start flag remains "1". Therefore A-D conversion can be repeated by generating another trigger. A trigger can also be generated during A-D conversion.

(2) Repeat mode [A-D control register bits 4, 3="01"]

In repeat mode, the input voltage to the one analog input pin selected with the analog input selection bits is repeatedly converted. No interrupt request occur and the A-D conversion start flag is not cleared to "0" automatically. The conversion of the selected pin is repeated while the A-D conversion start flag is "1".

The analog input pin must be selected before generating the trigger. Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is generated and A-D conversion starts. Each time an A-D conversion completes, the content of the successive approximation register (converted result) is transferred to the A-D register. The A-D converter does not stop at this point and conversion is repeated.

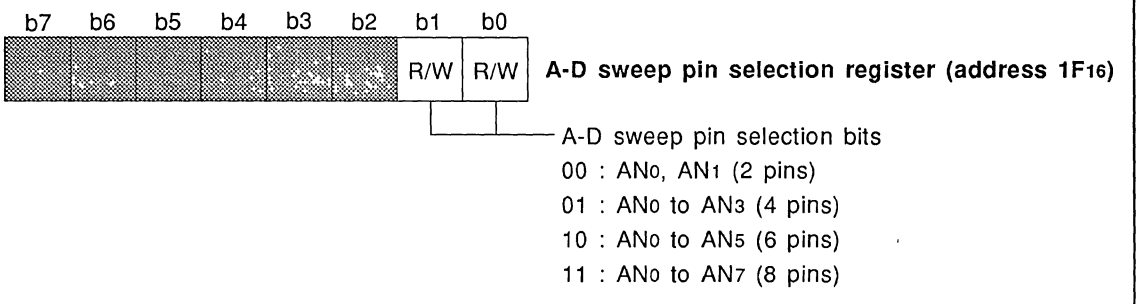
●When an external trigger is selected to start A-D conversion

A-D conversion starts when the input level of the $\overline{\text{AD}_{\text{TRG}}}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1". Each time an A-D conversion completes, the content of the successive approximation register (conversion result) is transferred to the A-D register. The A-D converter does not stop at this point and conversion is repeated.

(3) Single sweep mode [A-D control register bits 4, 3="10"]

In single sweep mode, multiple analog input pins can be converted. The analog input pins are selected by bits 0 and 1 of the A-D sweep pin selection register (address $1F_{16}$) shown in Figure 2.10.6. The number of analog input pins can be selected among either 2, 4, 6, or 8 pins. The analog input pins must be selected before generating the trigger.

A-D conversion is performed only for the input voltage of the selected input pins. An A-D interrupt request occurs and the A-D interrupt request bit is set to "1" when all selected pins are converted. The analog input selection bits in the A-D control register are ignored in this mode.



Note: Bits 0 and 1 are set to "1" at reset and AN_0 to AN_7 pins are selected for analog input pins. Bits 2 to 7 are undefined at reading.

Fig. 2.10.6 A-D sweep pin selection register structure

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is generated and A-D conversion of the AN_0 pin starts. After the AN_0 pin is converted, the selected analog input pins are converted in sequence. The converted result is transferred from the successive approximation register to the corresponding A-D register each time a pin is converted. When all of the selected pins are converted, an A-D interrupt request is generated and the interrupt request bit is set to "1". At this point, the A-D conversion start flag is cleared and the A-D converter stops.

●When an external trigger is selected to start A-D conversion

The selected pins are converted in order starting from the AN_0 pin similar to selecting an internal trigger when the input level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1" . The converted result is transferred from the successive approximation register to the corresponding A-D register each time a pin is converted. When all of the selected pins are converted, an A-D interrupt request occurs and the interrupt request bit is set to "1". At this point, the A-D conversion start flag maintains "1". Therefore, A-D conversion can be repeated from the AN_0 pin by generating another trigger. A trigger can also be generated during A-D conversion.

(4) Repeat sweep mode [A-D control register bits 4, 3="11"]

In repeat sweep mode, the A-D sweep pin selection register can be used to select multiple analog input pins to be converted as with single sweep mode. Conversion is performed in order from the AN_0 pin. After converting all selected pins, A-D converter does not stop, but repeats conversion from the AN_0 pin. No A-D interrupt request occur when all selected pins are converted.
The analog input selection bits in the A-D control register are ignored in this mode.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is generated and A-D conversion starts from the AN_0 pin. After the AN_0 pin is converted, the selected analog input pins are converted in sequence. The converted result is transferred from the successive approximation register to the corresponding A-D register each time a pin is converted.
When all of the selected pins are converted, conversion is repeated from the AN_0 pin. Conversion is repeated until the A-D conversion start flag is cleared to "0".

●When an external trigger is selected to start A-D conversion

The selected pins are converted in order starting from the AN_0 pin similar to selecting an internal trigger when the input level of the $\overline{ADTRG}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1" . The converted result is transferred from the successive approximation register to the corresponding A-D register each time a pin is converted. Conversion is repeated until the A-D conversion start flag is cleared to "0".

[Precautions when using an A-D converter]

1. Analog input pins must be selected (with analog input selection bits of A-D control register and A-D sweep pin selection register) before internal or external trigger is generated.

2. The port P7 direction register bit corresponding to the pin selected as analog input pin and external trigger input pin (port $P7_7$) must be set to "0" (input mode).

3. When an external trigger is selected, port $P7_7$ functions as $\overline{ADTRG}$ pin. If at the same time, AN_7 pin is selected as the analog input, the external trigger input signal is converted and the converted result is transferred to A-D register 7.

2.11 Watchdog timer

The watchdog timer is a 12-bit timer that is used to detect unexpected execution sequence caused by software run-away. It is also used to stabilize the oscillator when returning from the **STP** instruction. Figure 2.11.1 shows the block diagram of the watchdog timer.

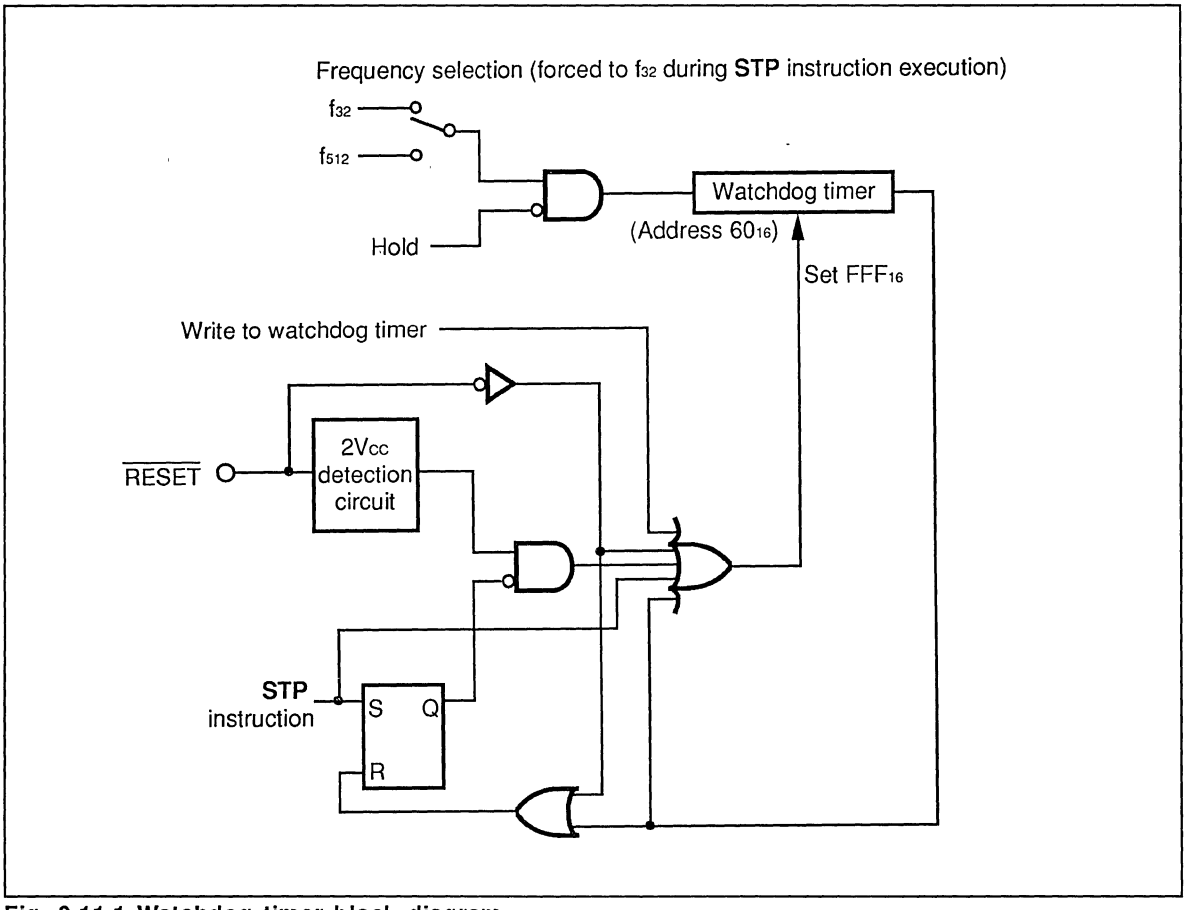


Fig. 2.11.1 Watchdog timer block diagram

2.11.1 Operation description

The watchdog timer (address 60_{16}) consists of 12 bits and its content is decremented by 1 each time the clock selected with the watchdog timer frequency selection flag (bit 0 at address 61_{16}) is input to the watchdog timer.

The watchdog timer frequency selection flag is set to 0 at reset, and f_{512} (source oscillating frequency $f(X_{IN})$ divided by 512) is selected as the watchdog timer count source after reset. Thereafter, it can also be set to f_{32} (source oscillating frequency $f(X_{IN})$ divided by 32) by changing the watchdog timer frequency selection flag by program. Figure 2.11.2 shows the structure of the watchdog timer frequency selection flag.

When there is a reset, "FFF_{16}" is set in the watchdog timer. Then the count source f_{512} is counted after reset. The content of the watchdog timer is decremented by 1 each time a clock is input. An interrupt request occurs when the most significant bit of the watchdog timer becomes "0" after 2048 counts. The watchdog timer interrupt is a non-maskable interrupt with the highest priority. Processor interrupt priority level (IPL) is set to level 7 when accepting the interrupt.

An arbitrary value cannot be set in the watchdog timer. A value "FFF_{16}" is automatically set in the watchdog timer when there is a reset, when the **STP** instruction is executed, or when a writing operation is performed in the watchdog timer. The watchdog timer is a write-only register and its content is undefined at reading. The watchdog timer is in Hold state and the clock input of the watchdog timer is disabled while "L" level is applied to the $\overline{HOLD}$ pin (in Hold state).

When using the watchdog timer to detect program run-away, the program must write to the watchdog timer before its most significant bit becomes "0". Then if this code is not executed due to program run-away, the most significant bit of the watchdog timer becomes "0" and an interrupt is generated. Thereafter, the control should be passed to the interrupt service routine.

To restart from reset after detecting a program run-away, bit 3 of the processor mode register (software reset bit) must be set to "1" in the watchdog timer interrupt service routine. In this way, a run-away program can be automatically reset and returned to normal routine.

In addition to detecting program run-away, the watchdog timer is also used as a return timer from a stop mode (halting of oscillating circuit with the **STP** instruction). When the **STP** instruction is executed, the watchdog timer count source is forced to f_{32} and "FFF_{16}" is set in the watchdog timer. Then when the watchdog timer is started with an external interrupt and a watchdog timer interrupt request occurs, a supply of internal clock ϕ starts. This is because some time is required for the oscillator to stabilize. See section "4.2 Clock generator" for more detail concerning the stop mode.

In order to stop the watchdog timer (disable its function), twice of the V_{CC} voltage must be applied to the $\overline{RESET}$ pin. During this time, the watchdog timer stops with "FFF_{16}" set.

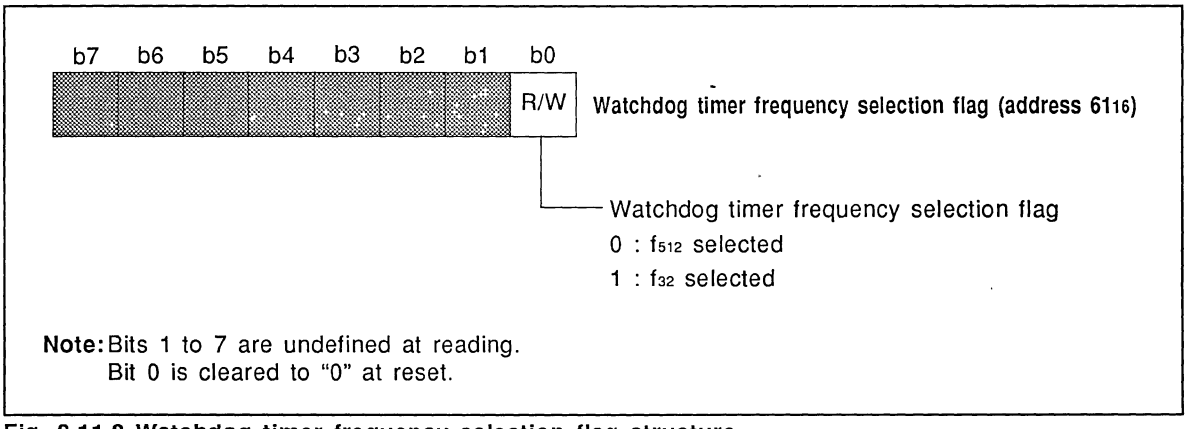


Fig. 2.11.2 Watchdog timer frequency selection flag structure

2.12 Hold function

The microcomputer is in Hold state while "L" level is input to the $\overline{HOLD}$ pin ($P4_0$) in memory expansion mode and microprocessor mode. Table 2.12.1 shows the status of the microcomputer during Hold. Input mode must be selected by setting the port P4 direction register bit 0 to "0".

The $\overline{HOLD}$ input ("L" level input) to the $\overline{HOLD}$ pin is accepted at the falling edge of the internal clock ϕ from "H" to "L" while the bus is unused.

When the $\overline{HOLD}$ input is accepted, ϕ_{CPU} (CPU operating clock: $f(X_{IN})/2$) stops at "L" level, and $\overline{E}$ output stops at "H" level at the completion of the executing bus cycle. In addition, "L" level is output from the $\overline{HLDA}$ pin ($P3_3$) to indicate that the microcomputer is in Hold state. Ports P0 to P2, $P3_0$, and $P3_1$ are floated during Hold. These ports are floated after one cycle of the internal clock ϕ later than the $\overline{HLDA}$ pin changes from "H" level to "L" level.

Only ϕ_{CPU} is stopped during Hold. The oscillator is operating and other internal peripherals can be operating. However, the watchdog timer is stopped.

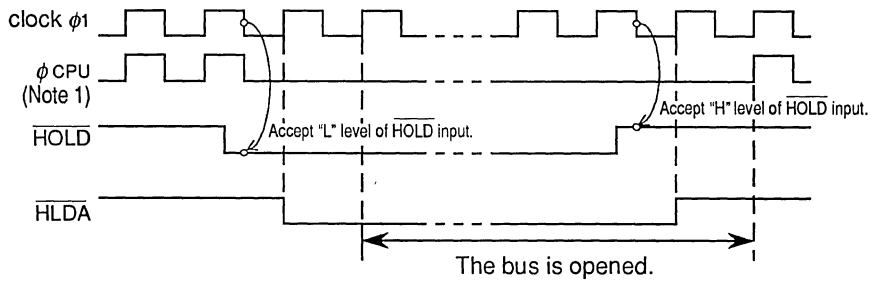
Hold state can be removed by returning the $\overline{HOLD}$ pin to "H" level. In this case, $\overline{HOLD}$ input ("H" level input) is also accepted at the falling edge of the internal clock ϕ from "H" to "L" while the bus is unused. At the removing of Hold state, these ports are removed from Hold state after one cycle of the internal clock ϕ later than the $\overline{HLDA}$ pin changes from "L" level to "H" level.

Figure 2.12.1 shows the timing example when the Hold.

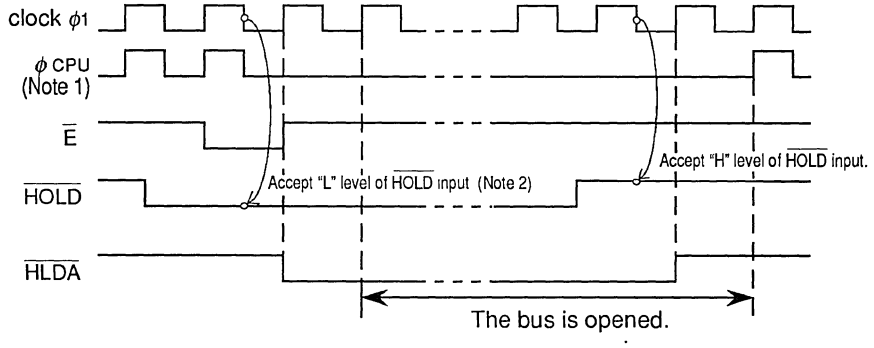
Table 2.12.1 Microcomputer status during Hold (in memory expansion and microprocessor mode)

Parameter	Status during Hold
Oscillation	Operating
Internal clock ϕ	Operating
Clock ϕ_1	In memory expansion mode·····Output when the processor mode register bit 7 is "1" In microprocessor mode·····Always output
ϕ_{CPU}	Stopped at "L" level
$\overline{E}$ output	Stopped at "H" level
Ports P0–P2, $P3_0$, $P3_1$	Floating
Port $P3_2$, $\overline{HLDA}$ pin ($P3_3$)	Stopped at "L" level
Ports $P4_3$–$P4_7$, P5, P6, P7, P8	Retain the status at inputting "L" level to the $\overline{HOLD}$ pin
Watchdog timer	Stopped

(a) $\overline{HOLD}$ input is performed when the bus is unused



(b) $\overline{HOLD}$ input is performed when the bus is used



Note 1 : φ CPU is an internal signal and cannot be observed externally.
Note 2 : $\overline{E}$ output is "L" level at this point, however, $\overline{HOLD}$ input is accepted because the bus is unused internally.

Fig. 2.12.1 Timing example when the Hold

2.13 Ready function

In memory expansion mode and microprocessor mode, the microcomputer is in Ready state while "L" level is input to the $\overline{RDY}$ pin ($P4_1$). Table 2.13.1 shows the status of the microcomputer during Ready. Input mode must be selected by setting the port P4 direction register bit 1 to "0".
The Ready function is used for example when externally connecting slow memory.
When the $\overline{RDY}$ pin becomes "L" level, the internal clock ϕ and ϕ_{CPU} (CPU operating clock: $f(X_{IN})/2$) stop at "L" level. The bus and port retain the status when "L" level is input to the $\overline{RDY}$ pin.
During Ready state, the internal clock ϕ and ϕ_{CPU} are stopped, but the oscillator is operating so that other internal peripherals can be operating.
Ready state can be removed by returning the $\overline{RDY}$ pin to "H" level.
Figure 2.13.1 shows the timing example when the Ready.

Table 2.13.1 Microcomputer status during Ready state (in memory expansion and microprocessor mode)

Parameter	Status during Ready
Oscillation	Operating
Internal clock ϕ	Stopped at "L" level
Clock ϕ_1	In memory expansion mode·····Output when the processor mode register bit 7 is "1" In microprocessor mode ·····Always output
ϕ_{CPU}	Stopped at "L" level
$\overline{E}$ output	Stopped at either "H" level or "L" level
Ports P0–P2, $P3_0$–$P3_2$, $P4_3$–$P4_7$, P5–P8	Retain the status at inputting "L" level to the $\overline{RDY}$ pin
Watchdog timer	Operating

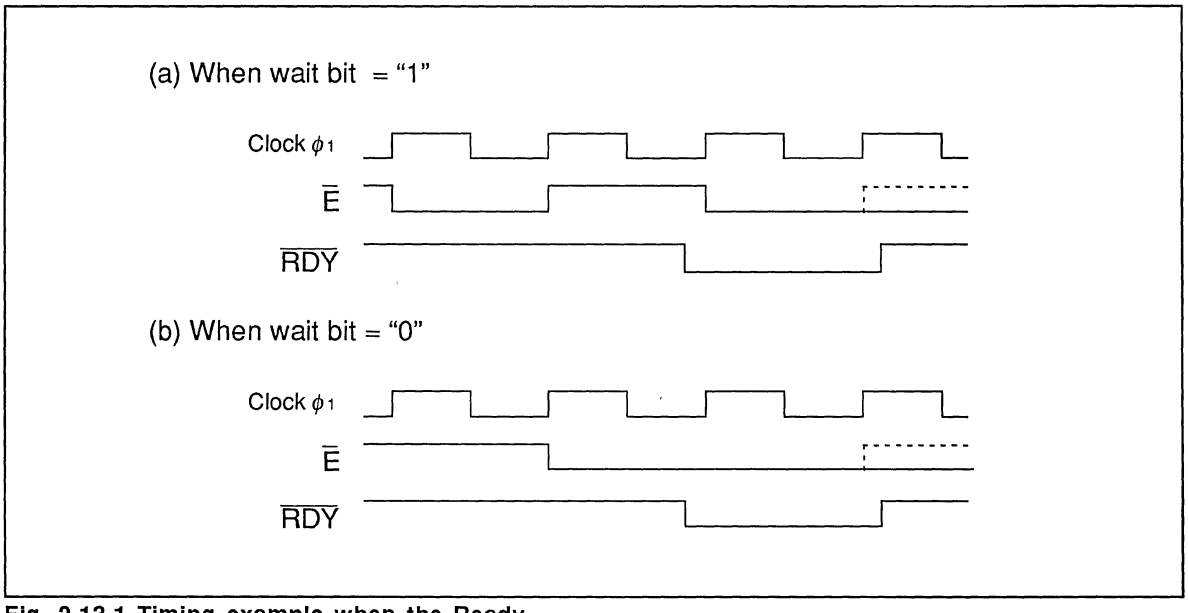


Fig. 2.13.1 Timing example when the Ready

CHAPTER 3
RESET

3.1 Reset

The CPU becomes reset state when "L" level is applied to the $\overline{\text{RESET}}$ pin. Reset state is removed and program execution starts from address set in the reset vector table when "H" level is then applied to the $\overline{\text{RESET}}$ pin.

3.1.1 Reset operation

The CPU becomes reset state when "L" level is applied to the $\overline{\text{RESET}}$ pin in case the supply voltage is 5V±10%. When oscillation is stable, the "L" level must be applied for at least 2μs.

Apply "L" level to the $\overline{\text{RESET}}$ pin for sufficient interval (approximately 10ms) before returning to "H" level if sufficient time is required for the oscillator to stabilize such as reset during stop mode entered with the **STP** instruction. Reset state is removed if "H" level is applied to the $\overline{\text{RESET}}$ pin while in reset state.

When reset state is removed, program execution starts from the address formed by using the content of address $FFFF_{16}$ at bank 0 as high-order and address $FFFE_{16}$ as low-order. Figure 3.1.1 shows the internal processing sequence after removing reset state.

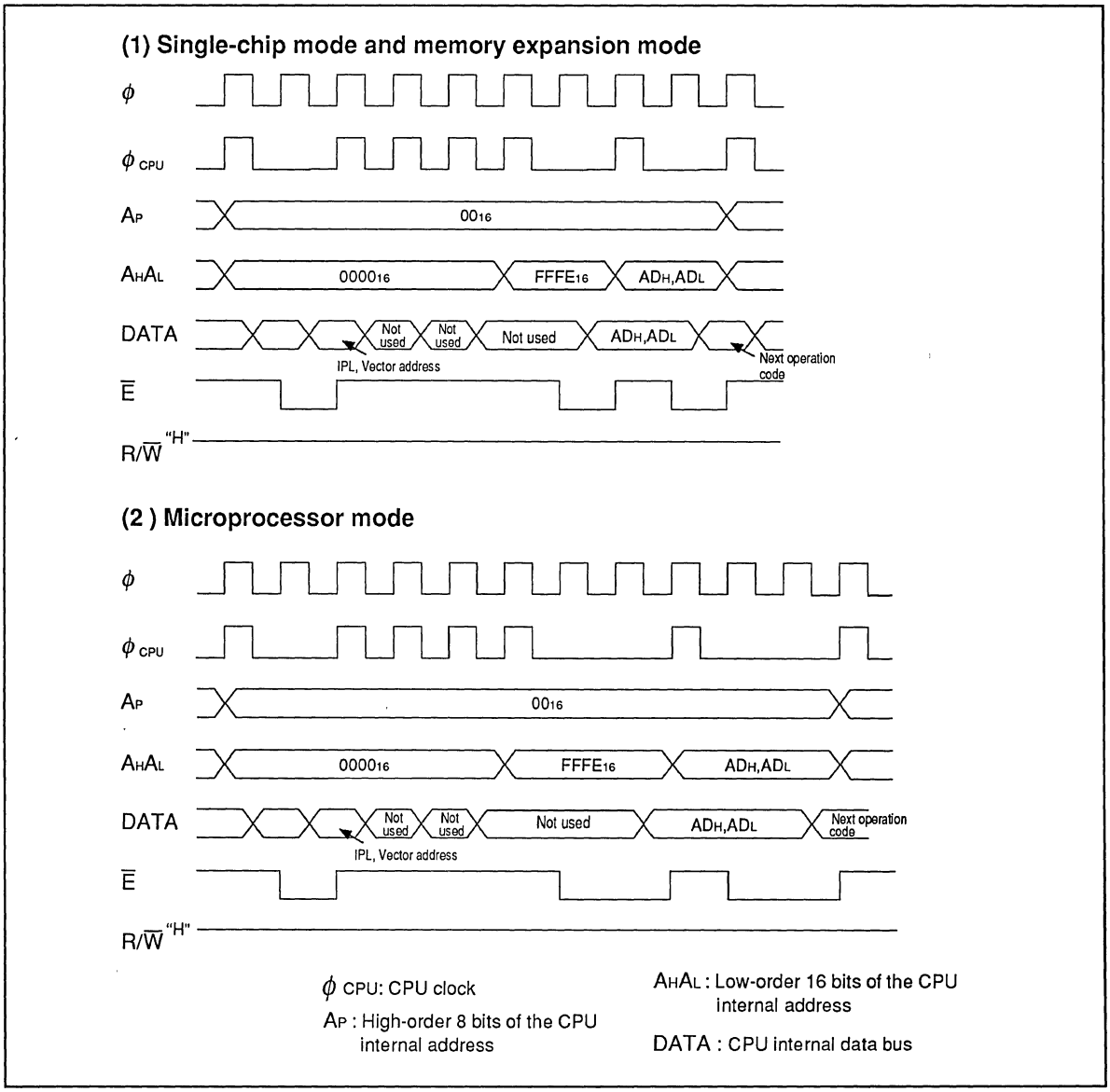


Fig. 3.1.1 Internal processing sequence after removing reset state

3.1.2 Internal status at reset

Figure 3.1.2 shows the contents of internal registers immediately at reset.

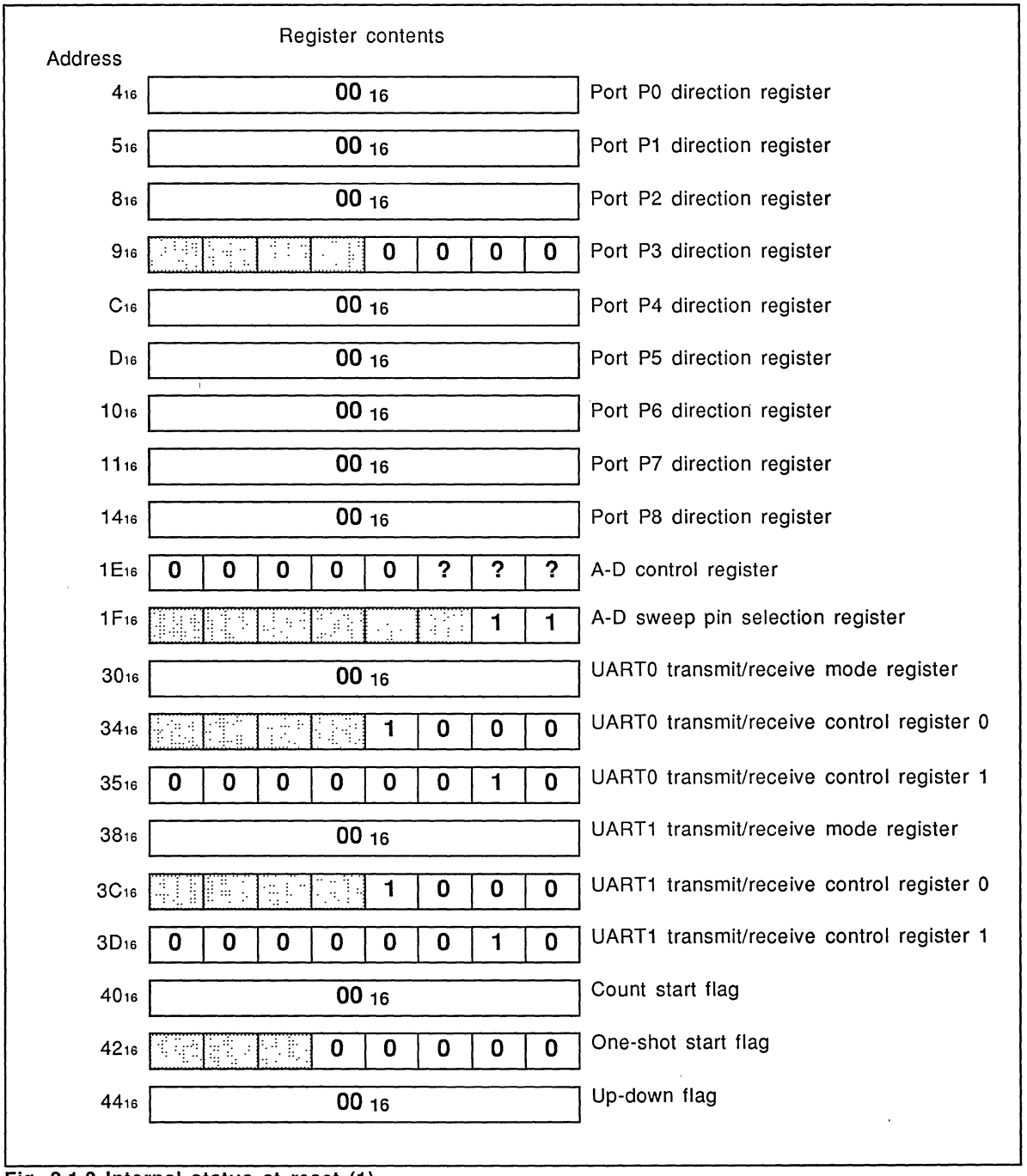
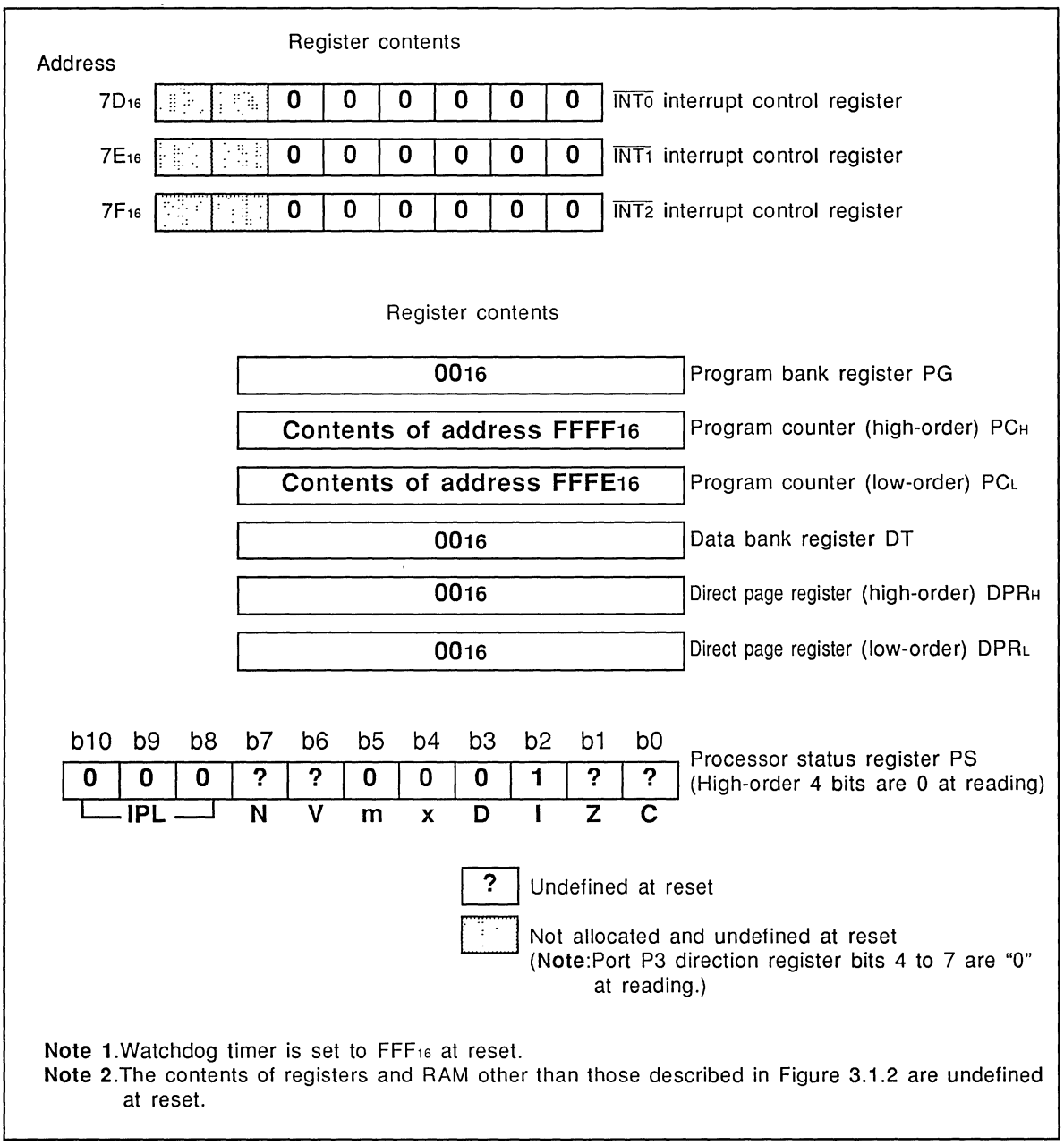


Fig. 3.1.2 Internal status at reset (1)

Address	Register contents	
56_{16}	00 16	Timer A0 mode register
57_{16}	00 16	Timer A1 mode register
58_{16}	00 16	Timer A2 mode register
59_{16}	00 16	Timer A3 mode register
$5A_{16}$	00 16	Timer A4 mode register
$5B_{16}$	0 0 1 0 0 0 0 0	Timer B0 mode register
$5C_{16}$	0 0 1 0 0 0 0 0	Timer B1 mode register
$5D_{16}$	0 0 1 0 0 0 0 0	Timer B2 mode register
$5E_{16}$	00 16	Processor mode register
60_{16}	FFF 16	Watchdog timer (**Note 1**)
61_{16}	0	Watchdog timer frequency selection flag
70_{16}	0 0 0 0	A-D conversion interrupt control register
71_{16}	0 0 0 0	UART0 transmit interrupt control register
72_{16}	0 0 0 0	UART0 receive interrupt control register
73_{16}	0 0 0 0	UART1 transmit interrupt control register
74_{16}	0 0 0 0	UART1 receive interrupt control register
75_{16}	0 0 0 0	Timer A0 interrupt control register
76_{16}	0 0 0 0	Timer A1 interrupt control register
77_{16}	0 0 0 0	Timer A2 interrupt control register
78_{16}	0 0 0 0	Timer A3 interrupt control register
79_{16}	0 0 0 0	Timer A4 interrupt control register
$7A_{16}$	0 0 0 0	Timer B0 interrupt control register
$7B_{16}$	0 0 0 0	Timer B1 interrupt control register
$7C_{16}$	0 0 0 0	Timer B2 interrupt control register

Fig. 3.1.2 Internal status at reset (2)



Note 1. Watchdog timer is set to FFF16 at reset.
Note 2. The contents of registers and RAM other than those described in Figure 3.1.2 are undefined at reset.

Fig. 3.1.2 Internal status at reset (3)

3.1.3 Bus state during reset in microprocessor mode

Table 3.1.1 shows the state of the address bus, data bus, and control bus during reset ("L" level is applied to $\overline{\text{RESET}}$ pin) for Mask ROM version and PROM version models in microprocessor mode, and for external ROM version models.

Table 3.1.1 Bus state during reset for Mask ROM version and PROM version in microprocessor mode and for external ROM version

Model	Address bus, data bus, control bus state	
Mask ROM version External ROM version	Ports P0, P1, P2	Address output ("H" or "L" level)
	E, Port $P3_0$ (R/$\overline{W}$)	"H" level output
	Port $P3_1$($\overline{BHE}$)	"H" or "L" level output (depends on address output)
	Port $P3_2$ (ALE)	"L" level output
	Port $P4_2$(ϕ_1)	Clock ϕ_1 output
PROM version	Ports P0, P1, P2	When "H" level is applied to either or both pins $P5_1$ and $P5_2$; ●Ports P0, P1, P2.....floating When "L" level is applied to pins $P5_1$ and $P5_2$; ●Ports P0, P1 .....floating ●Port P2 .....data output
	E, Port $P3_0$ (R/$\overline{W}$)	"H" level output
	Port $P3_1$ ($\overline{BHE}$)	"H" level or "L" level output
	Port $P3_2$ (ALE)	"L" level output
	Port $P4_2$ (ϕ_1)	Clock ϕ_1 output

3.2 Reset circuit

The reset circuit must be designed so that the reset input voltage drops below 0.9V when the source voltage reaches 4.5V as shown in Figure 3.2.1.
Figure 3.2.2 shows the example of power-on reset circuit using a system reset IC M51957AL.

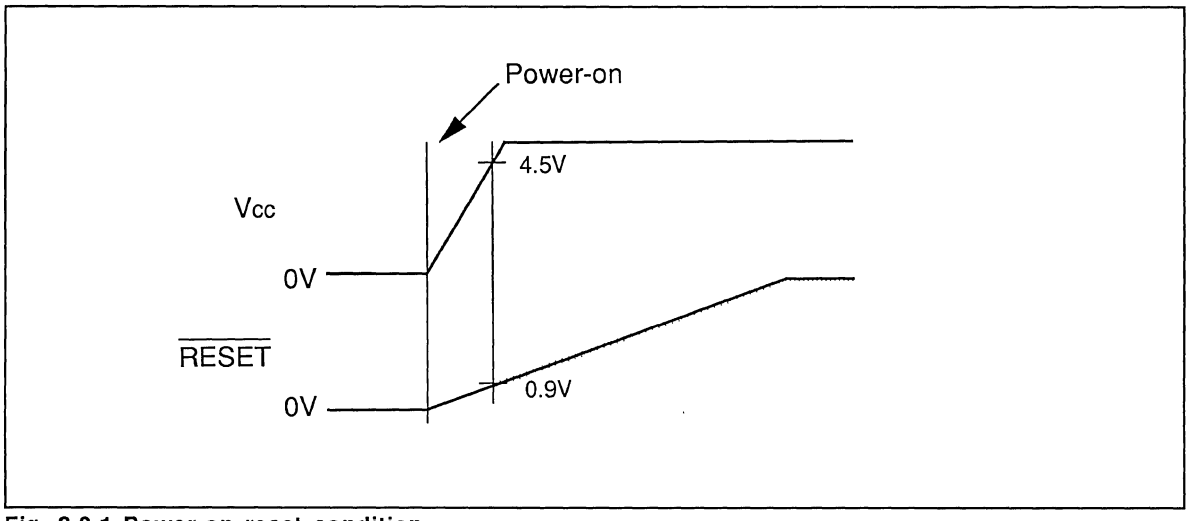


Fig. 3.2.1 Power-on reset condition

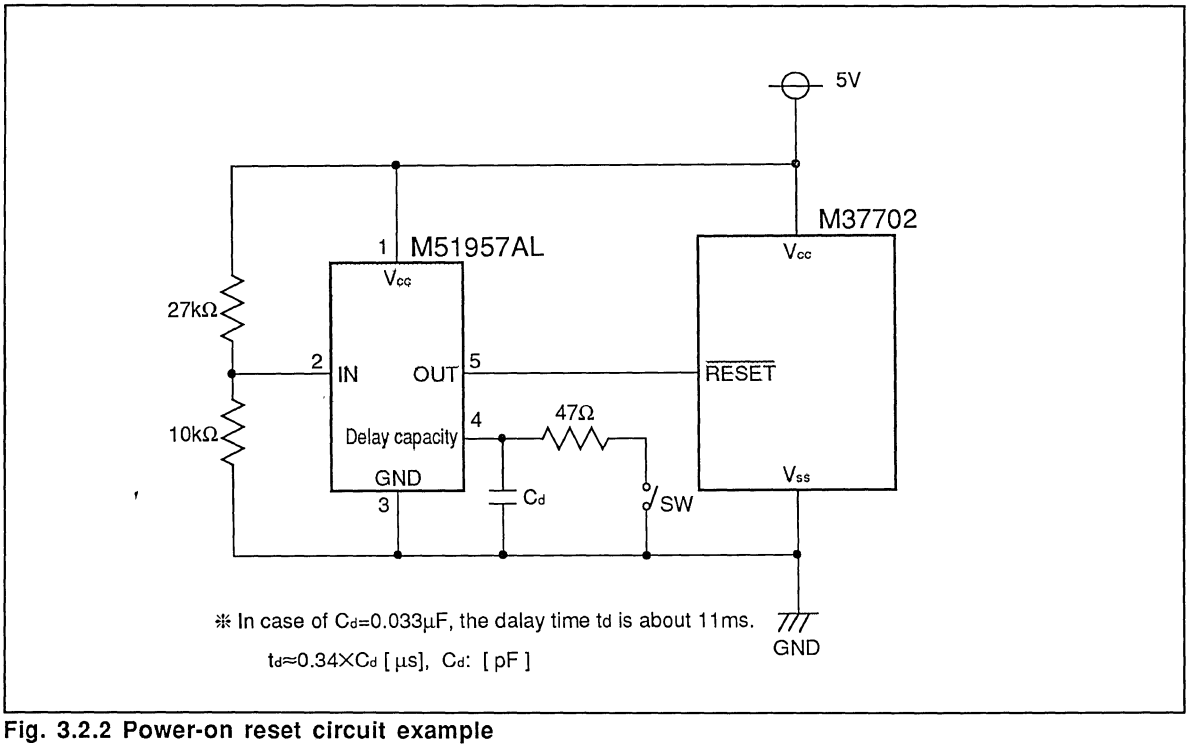
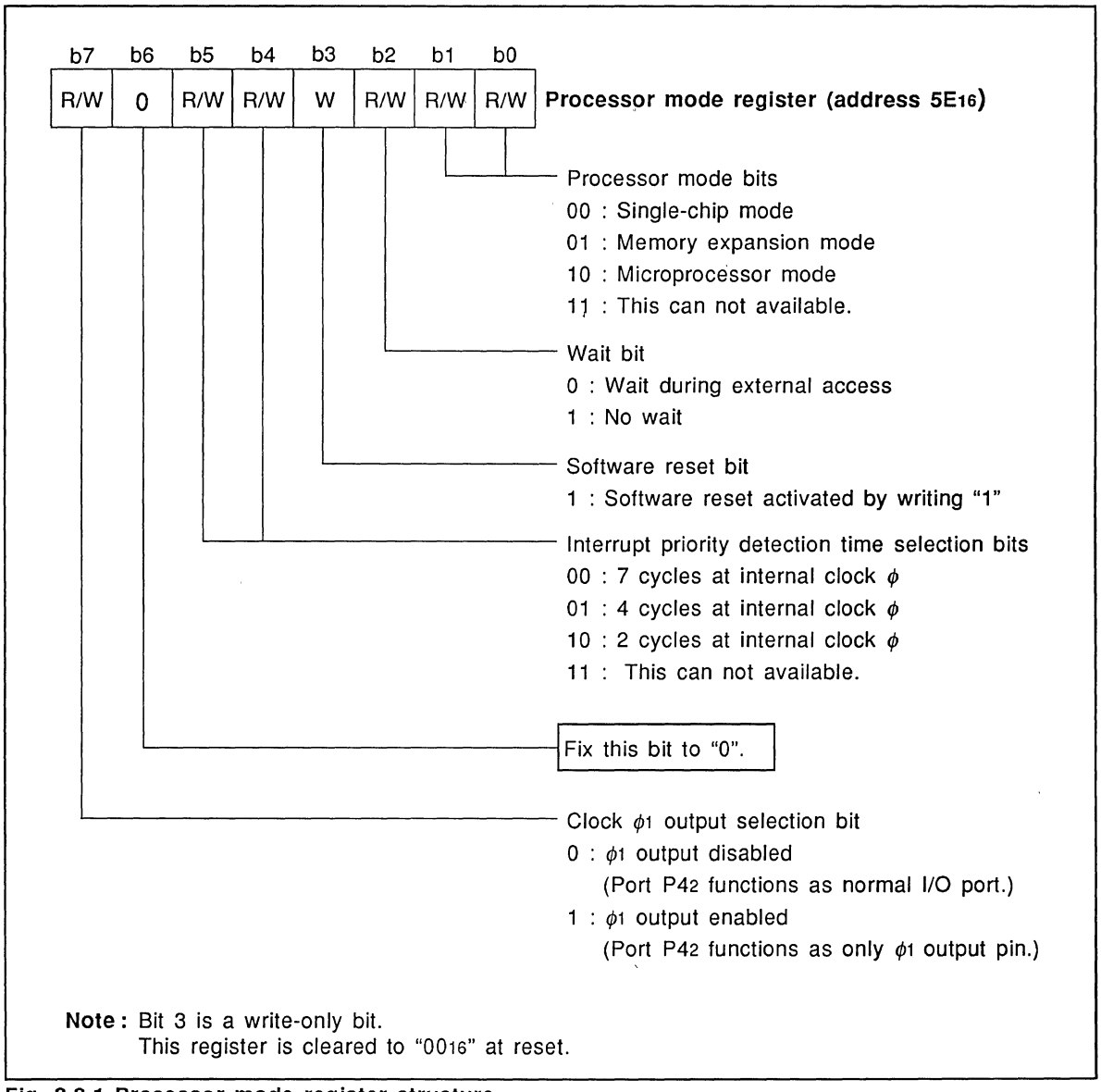


Fig. 3.2.2 Power-on reset circuit example

3.3 Software reset

The M37702 group can be reset internally by program. Software reset is generated by writing "1" to the processor mode register bit 3. Figure 3.3.1 shows the structure of the processor mode register.
Software reset is the same as hardware reset (when the $\overline{\text{RESET}}$ pin level is returned to "H" after applying "L" level) except that the contents of the internal RAM are preserved. Therefore, the contents of each register after software reset is initialized to values shown in Figure 3.1.2.



Note : Bit 3 is a write-only bit.
This register is cleared to "0016" at reset.

Fig. 3.3.1 Processor mode register structure

CHAPTER 4
OSCILLATING CIRCUIT

4.1 Oscillating circuit

The M37702 group is equipped with an oscillating circuit to generate the necessary clock. The frequency input to the clock input pin X_{IN} is divided in half to obtain the internal clock ϕ. This ϕ is further divided in half to obtain the bus cycle. Either a ceramic resonator or a crystal resonator can be connected externally to the internal oscillating circuit.

4.1.1 Circuit using a ceramic resonator or a crystal resonator

Figure 4.1.1 shows the circuit example using a ceramic resonator and Figure 4.1.2 shows the circuit example using a crystal resonator. An oscillating circuit is formed by connecting the resonator between X_{IN} pin and X_{OUT} pin as shown in the figures. The circuit constants such as R_f, R_d, C_{IN}, and C_{OUT} must be set to the resonator manufacturer's recommended values.

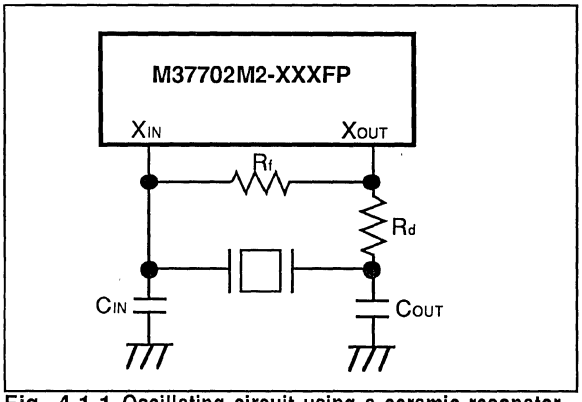


Fig. 4.1.1 Oscillating circuit using a ceramic resonator

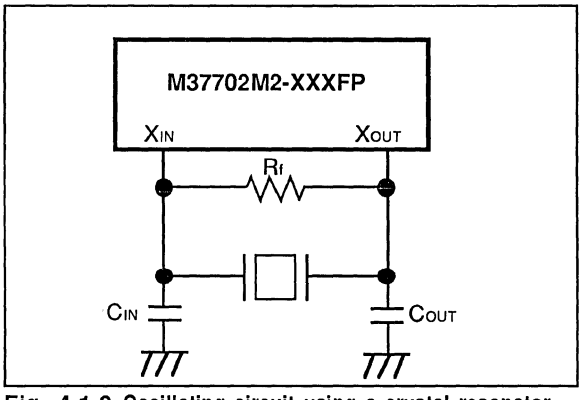


Fig. 4.1.2 Oscillating circuit using a crystal resonator

4.1.2 External clock input circuit

An external clock signal can be supplied to the internal oscillating circuit. Figure 4.1.3 shows the circuit example of external clock input. Note that the external clock must be input from X_{IN} pin, and X_{OUT} pin must be left open.

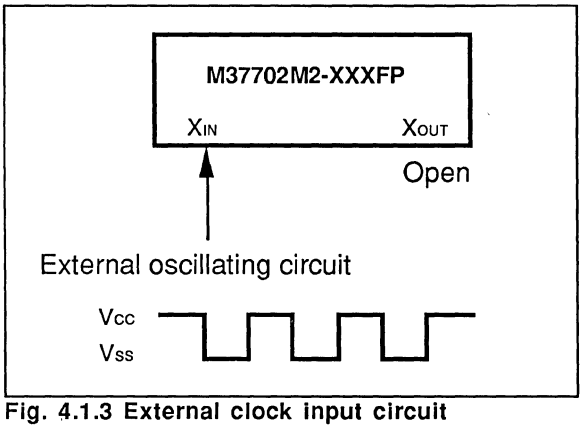


Fig. 4.1.3 External clock input circuit

4.2 Clock generator

The oscillating circuit consists of oscillating gate which functions as an amplifier to obtain the necessary gain and oscillation control flip-flop to control this amplifier. Therefore, oscillation can be stopped and restarted as necessary. The clock generation circuit shown in Figure 4.2.1 is built in the M37702 group.

When the **STP** instruction is executed, the internal clock ϕ stops oscillation at the "L" state. At the same time, FFF_{16} is set in the watchdog timer and the input to the watchdog timer is connected to f_{32}. This connection is cancelled and connected to the input determined by the content of the watchdog timer frequency selection flag when the most significant bit of the watchdog timer becomes "0" or when there is a reset. Oscillation is resumed when an interrupt is accepted, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer becomes "0". This is done to avoid the initial unstable interval when using a ceramic resonator.

When the **WIT** instruction is executed, the internal clock ϕ stops at the "L" state, but the oscillator does not stop. Therefore, the timer, serial I/O, and A-D converter can be used. The stopping of internal clock ϕ is cancelled when an interrupt is accepted. An instruction can be executed immediately because the oscillator does not stop.

Refer to "Appendix 4. Stop, wait, one-wait, Ready, and Hold state" and "MELPS 7700 SOFTWARE MANUAL" for information concerning the **STP** and **WIT** instructions.

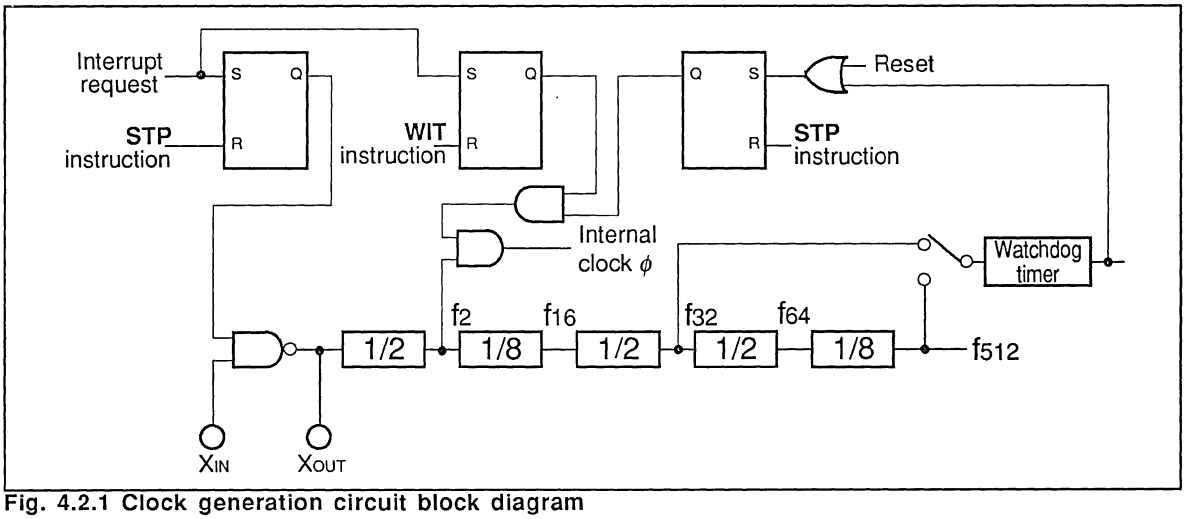


Fig. 4.2.1 Clock generation circuit block diagram

MEMO

CHAPTER 5
ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1 Electrical characteristics

5.1.1 Absolute maximum ratings

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3 to 7	V
AV_{CC}	Analog supply 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
Pd	Power dissipation	Ta=25°C	300	mW
Topr	Operating temperature		−20 to 85	°C
Tstg	Storage temperature		−40 to 150	°C

5.1.2 Recommended operating conditions

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

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AVcc	Analog supply voltage			Vcc		V
Vss	Supply voltage			0		V
AVss	Analog supply 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}, $\overline{RESET}$, CNVss, BYTE	0.8Vcc		Vcc	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	0.8Vcc		Vcc	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	0.5Vcc		Vcc	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}, $\overline{RESET}$, CNVss, BYTE	0		0.2Vcc	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in single-chip mode)	0		0.2Vcc	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (in memory expansion mode and microprocessor mode)	0		0.16Vcc	V
$I_{OH\ (peak)}$	High-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			−10	mA
$I_{OH\ (avg)}$	High-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			−5	mA
$I_{OL\ (peak)}$	Low-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			10	mA
$I_{OL\ (avg)}$	Low-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$			5	mA
$f(X_{IN})$	External clock frequency input	M37702M2-XXXFP, M37702S1FP			8	MHz
		M37702M2AXXXFP, M37702S1AFP			16	
		M37702M2BXXXFP, M37702S1BFP			25	

Note 1. Average output current is the average value of a 100ms interval.

2. The sum of I_{OL}(peak) for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of I_{OH}(peak) for ports P0, P1, P2, P3, and P8 must be 80mA or less, the sum of I_{OL}(peak) for ports P4, P5, P6, and P7 must be 80mA or less, and the sum of I_{OH}(peak) for ports P4, P5, P6, and P7 must be 80mA or less.

5.1.3 Electrical characteristics and A-D converter characteristics

M37702M2-XXXFP

Electrical characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=8MHz, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OH}=–10mA	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}=–400µA	4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH}=–10mA	3.1			V
			I_{OH}=–400µA	4.8			
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH}=–10mA	3.4			V
			I_{OH}=–400µA	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}=10mA			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}=2mA			0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL}=10mA			1.9	V
			I_{OL}=2mA			0.43	
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL}=10mA			1.6	V
			I_{OL}=2mA			0.4	
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$–$TA4_{IN}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT0}$–$\overline{INT2}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, CLK0, CLK1		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=5V			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=0V			–5	µA
V_{RAM}	RAM hold voltage		When clock is stopped.	2			V
I_{CC}	Power supply current		In single-chip mode output only pin is open and other pins are Vss during reset. f(X_{IN})=8MHz, square waveform		6	12	mA
			Ta=25°C when clock is stopped.			1	µA
			Ta=85°C when clock is stopped.			20	

A-D converter characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=8MHz, 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		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

M37702M2AXXXFP

Electrical characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=16MHz, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
V_{OH}	High-level output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$	I_{OH}=–10mA		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}=–400µA		4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH}=–10mA		3.1			V
			I_{OH}=–400µA		4.8			
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH}=–10mA		3.4			V
			I_{OH}=–400µA		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}=10mA				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}=2mA				0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL}=10mA				1.9	V
			I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL}=10mA				1.6	V
			I_{OL}=2mA				0.4	
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$–$TA4_{IN}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT0}$–$\overline{INT2}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, $CLK0$, $CLK1$			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=5V				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=0V				–5	µA
V_{RAM}	RAM hold voltage		When clock is stopped.		2			V
I_{CC}	Power supply current		In single-chip mode output only pin is open and other pins are Vss during reset.	f(X_{IN})=16MHz, square waveform		12	24	mA
				Ta=25°C when clock is stopped.			1	µA
				Ta=85°C when clock is stopped.			20	

A-D converter characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=16MHz, 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		14.25			µs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

M37702M2BXXXFP

Electrical characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=25MHz, 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}=–10mA		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}=–400µA		4.7			V
V_{OH}	High-level output voltage	$P3_2$	I_{OH}=–10mA		3.1			V
			I_{OH}=–400µA		4.8			
V_{OH}	High-level output voltage	$\overline{E}$	I_{OH}=–10mA		3.4			V
			I_{OH}=–400µA		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}=10mA				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}=2mA				0.45	V
V_{OL}	Low-level output voltage	$P3_2$	I_{OL}=10mA				1.9	V
			I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage	$\overline{E}$	I_{OL}=10mA				1.6	V
			I_{OL}=2mA				0.4	
V_{T+}–V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$–$TA4_{IN}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT0}$–$\overline{INT2}$, AD_{TRG}, $\overline{CTS0}$, $\overline{CTS1}$, $CLK0$, $CLK1$			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=5V				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=0V				–5	µA
V_{RAM}	RAM hold voltage		When clock is stopped.		2			V
I_{CC}	Power supply current		In single-chip mode output only pin is open and other pins are Vss during reset.	f(X_{IN})=25MHz, square waveform		19	38	mA
				Ta=25°C when clock is stopped.			1	µA
				Ta=85°C when clock is stopped.			20	

A-D converter characteristics (V_{CC}=5V, V_{SS}=0V, Ta=25°C, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
—	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		9.12			µs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

5.1.4 Timing requirements

Timing requiements (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_C	External clock input cycle time	125		62		40		ns
$t_{W(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{W(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P0D-E)}$	Port P0 input setup time	200		100		60		ns
$t_{SU(P1D-E)}$	Port P1 input setup time	200		100		60		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	200		100		60		ns
$t_{SU(P3D-E)}$	Port P3 input setup time	200		100		60		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{h(E-P0D)}$	Port P0 input hold time	0		0		0		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(E-P3D)}$	Port P3 input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{SU(P1D-E)}$	Port P1 input setup time	60		45		30		ns
$t_{SU(P2D-E)}$	Port P2 input setup time	60		45		30		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	70		60		55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	70		60		55		ns
$t_{h(E-P1D)}$	Port P1 input hold time	0		0		0		ns
$t_{h(E-P2D)}$	Port P2 input hold time	0		0		0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		0		0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		0		0		ns

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA_{IN} input cycle time	250		125		80		ns
$t_{w(TAH)}$	TA_{IN} input high-level pulse width	125		62		40		ns
$t_{w(TAL)}$	TA_{IN} input low-level pulse width	125		62		40		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA_{IN} input cycle time	1000		500		320		ns
$t_{w(TAH)}$	TA_{IN} input high-level pulse width	500		250		160		ns
$t_{w(TAL)}$	TA_{IN} input low-level pulse width	500		250		160		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA_{IN} input cycle time	500		250		160		ns
$t_{w(TAH)}$	TA_{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(TAH)}$	TA_{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA_{IN} input low-level pulse width	250		125		80		ns

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

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(UP)}$	TA_{iOUT} input cycle time	5000		2500		2000		ns
$t_{w(UPH)}$	TA_{iOUT} input high-level pulse width	2500		1250		1000		ns
$t_{w(UPL)}$	TA_{iOUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-TIN)}$	TA_{iOUT} input setup time	1000		500		400		ns
$t_{h(TIN-UP)}$	TA_{iOUT} input hold time	1000		500		400		ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB_{IIN} input cycle time (one edge count)	250		125		80		ns
$t_{w(TBH)}$	TB_{IIN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{w(TBL)}$	TB_{IIN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{c(TB)}$	TB_{IIN} input cycle time (both edges count)	500		250		160		ns
$t_{w(TBH)}$	TB_{IIN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{w(TBL)}$	TB_{IIN} input low-level pulse width (both edges count)	250		125		80		ns

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$\overline{TBi_{IN}}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$\overline{TBi_{IN}}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$\overline{TBi_{IN}}$ input low-level pulse width	500		250		160		ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(TB)}$	$\overline{TBi_{IN}}$ input cycle time	1000		500		320		ns
$t_{W(TBH)}$	$\overline{TBi_{IN}}$ input high-level pulse width	500		250		160		ns
$t_{W(TBL)}$	$\overline{TBi_{IN}}$ input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{W(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	500		250		200		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	250		125		100		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD_i output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD_i hold time	30		30		30		ns
$t_{SU(D-C)}$	RxD_i hold time	60		30		20		ns
$t_{h(C-D)}$	RxD_i input hold time	90		90		90		ns

External interrupt $\overline{INT_i}$ input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{W(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		250		250		ns
$t_{W(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		250		250		ns

5.1.5 Switching characteristics

Switching characteristics (V_{CC}=5V±10%, V_{SS}=0V, Ta=25°C, unless otherwise noted)

Single-chip mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time		200		100		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time		200		100		80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		200		100		80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time		200		100		80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100		80	ns

Note : Test conditions are shown in Figure 5.1.1.

Memory expansion mode and microprocessor mode (when wait bit is "1")

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")		110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time		5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	80		24		5		ns
$t_{d(\phi 1-HLDA)}$	$\overline{HLDA}$ output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_{W(ALE)}$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	100		30		20		ns
$t_{d(E-\phi 1)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time	50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")	9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")	50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")	50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")	50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time	50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width	220		95		50		ns

Note : Test conditions are shown in Figure 5.1.1.

Memory expansion mode and microprocessor mode (when wait bit = "0", and external memory area accessed)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(P0A-E)}$	Port P0 address output delay time	100		30		12		ns
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")		110		70		45	ns
$t_{PXZ(E-P1Z)}$	Port P1 floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(P1A-E)}$	Port P1 address output delay time	100		30		12		ns
$t_{d(P1A-ALE)}$	Port P1 address output delay time	80		24		5		ns
$t_{d(E-P2Q)}$	Port P2 data output delay time		110		70		45	ns
$t_{PXZ(E-P2Z)}$	Port P2 floating start delay time		5		5		5	ns
$t_{d(P2A-E)}$	Port P2 address output delay time	100		30		12		ns
$t_{d(P2A-ALE)}$	Port P2 address output delay time	80		24		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_{W(ALE)}$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time	100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_{h(E-P0A)}$	Port P0 address hold time	50		25		18		ns
$t_{h(ALE-P1A)}$	Port P1 address hold time (BYTE="L")	9		9		9		ns
$t_{h(E-P1Q)}$	Port P1 data hold time (BYTE="L")	50		25		18		ns
$t_{PZX(E-P1Z)}$	Port P1 floating release delay time (BYTE="L")	50		25		18		ns
$t_{h(E-P1A)}$	Port P1 address hold time (BYTE="H")	50		25		18		ns
$t_{h(ALE-P2A)}$	Port P2 address hold time	9		9		9		ns
$t_{h(E-P2Q)}$	Port P2 data hold time	50		25		18		ns
$t_{PZX(E-P2Z)}$	Port P2 floating release delay time	50		25		18		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time	18		18		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time	18		18		18		ns
$t_{W(EL)}$	$\overline{E}$ pulse width	470		220		130		ns

Note : Test conditions are shown in Figure 5.1.1.

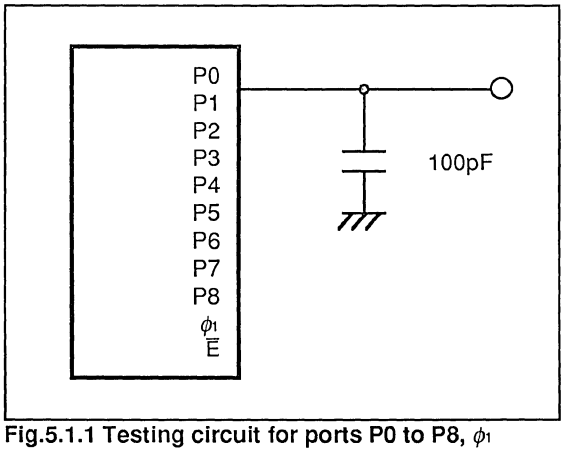


Fig.5.1.1 Testing circuit for ports P0 to P8, ϕ_1

5.1.6 Equations for calculating the parameters

Table 5.1.1 shows the equations for calculating the following parameters :

$t_{w(ALE)}$ ..... ALE pulse width

$t_{w(EL)}$ ..... $\overline{E}$ pulse width

$t_{d(PiA-E)}$ ..... Port Pi address output delay time (i=0 to 2)

$t_{d(BHE-E)}$ ..... $\overline{BHE}$ output delay time

$t_{d(R/W-E)}$ ..... R/$\overline{W}$ output delay time

$t_{pzx(E-PiZ)}$ ..... Port Pi floating release delay time

$t_{h(E-PiQ)}$ ..... Port Pi data hold time

$t_{d(PiA-ALE)}$ ..... Port Pi address output delay time

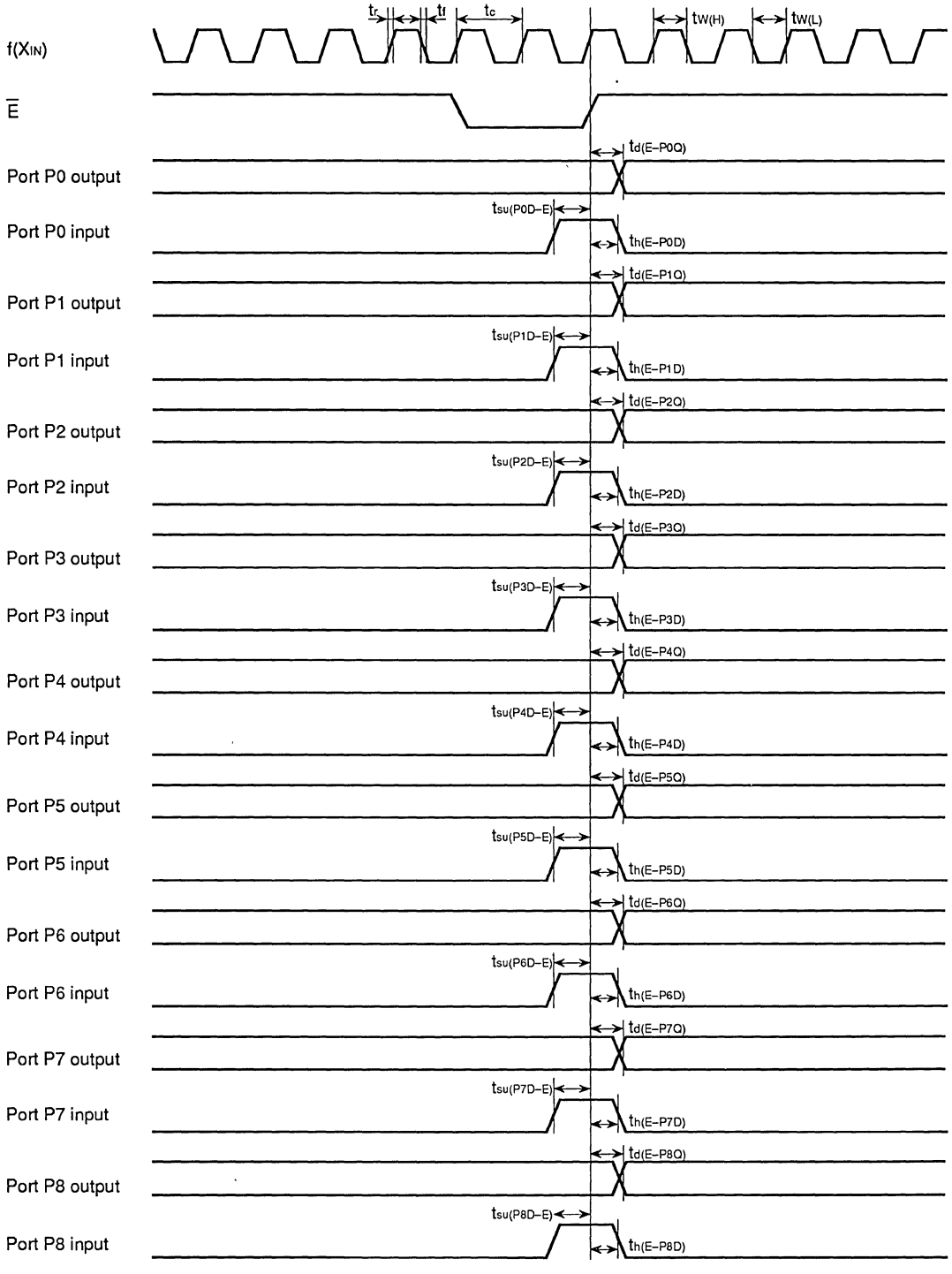
Table 5.1.1 Equations for calculating parameters

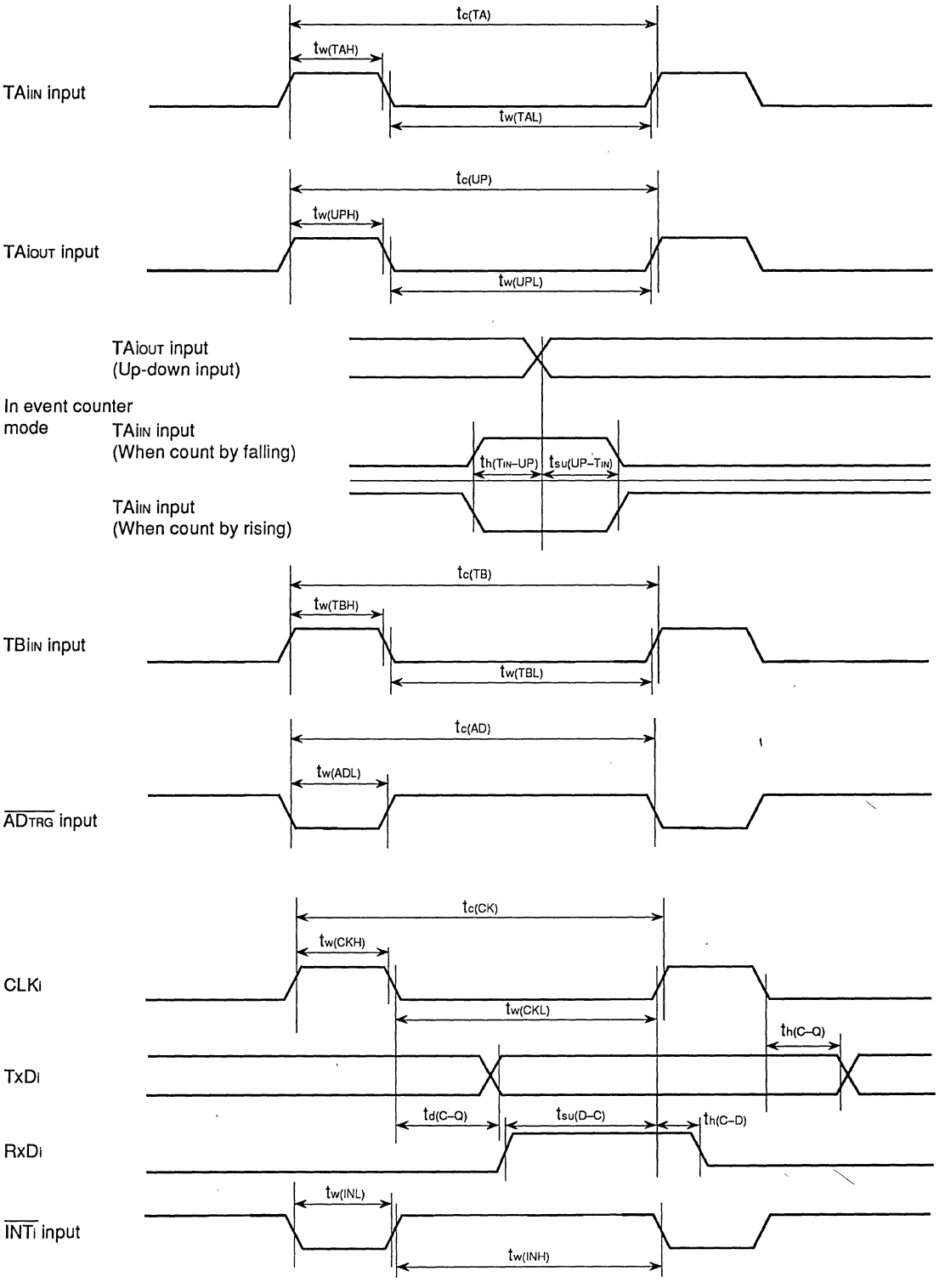
Parameter \ Frequency	8MHz version	16MHz version	25MHz version
$t_{w(ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$	$\frac{1 \times 10^9}{f(X_{IN})} - 27.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 18$
$t_{w(EL)}$ Wait bit = "1"	$\frac{2 \times 10^9}{f(X_{IN})} - 30$	←	←
$t_{w(EL)}$ Wait bit = "0"	$\frac{4 \times 10^9}{f(X_{IN})} - 30$	←	←
$t_{d(PiA-E)}$	$100 + \frac{2 \times 10^9}{f(X_{IN})} - 250$	$30 + \frac{2 \times 10^9}{f(X_{IN})} - 125$	$12 + \frac{2 \times 10^9}{f(X_{IN})} - 80$
$t_{d(BHE-E)}$ $t_{d(R/W-E)}$			$20 + \frac{2 \times 10^9}{f(X_{IN})} - 80$
$t_{pzx(E-PiZ)}$ $t_{h(E-PiQ)}$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 12.5$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 6.25$	$\frac{1 \times 10^9}{2 \times f(X_{IN})} - 2$
$t_{d(PiA-ALE)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 45$	$\frac{1 \times 10^9}{f(X_{IN})} - 38.5$	$\frac{1 \times 10^9}{f(X_{IN})} - 35$

(Dimension in ns)

5.1.7 Timing diagram

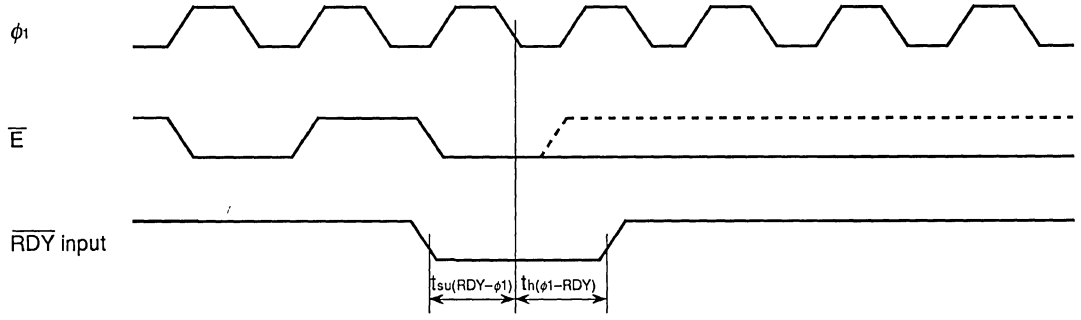
Single-chip mode



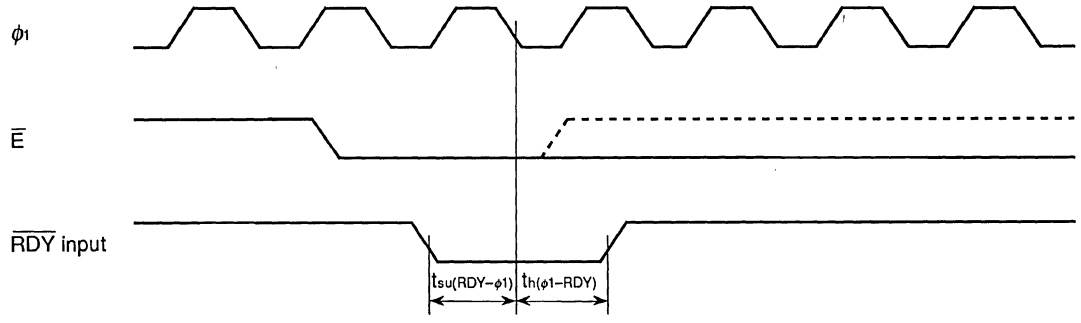

TAiIN input
tc(TA)
tw(TAH)
tw(TAL)
TAiOUT input
tc(UP)
tw(UPH)
tw(UPL)
TAiOUT input
(Up-down input)
In event counter
mode
TAiIN input
(When count by falling)
TAiIN input
(When count by rising)
th(TIN–UP)
tsu(UP–TIN)
TBiIN input
tc(TB)
tw(TBH)
tw(TBL)
ADTRG input
tc(AD)
tw(ADL)
CLKi
tc(CK)
tw(CKH)
tw(CKL)
th(C–Q)
TxDi
td(C–Q)
tsu(D–C)
th(C–D)
RxDi
INTi input
tw(INL)
tw(INH)

Memory expansion mode and microprocessor mode

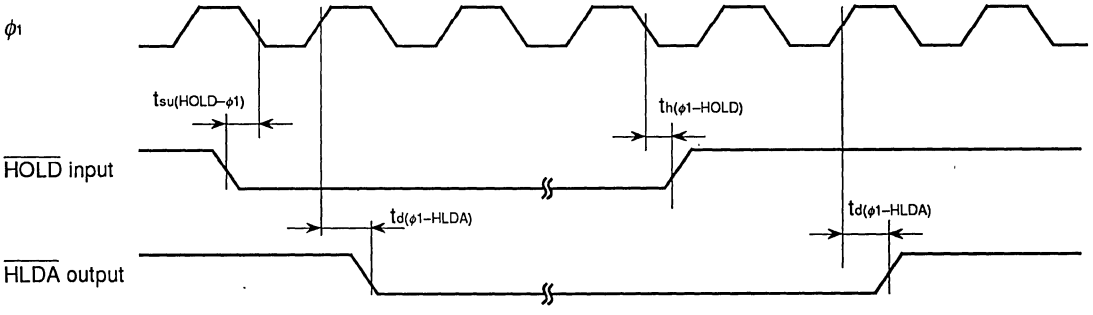
(When wait bit="1")



(When wait bit="0")

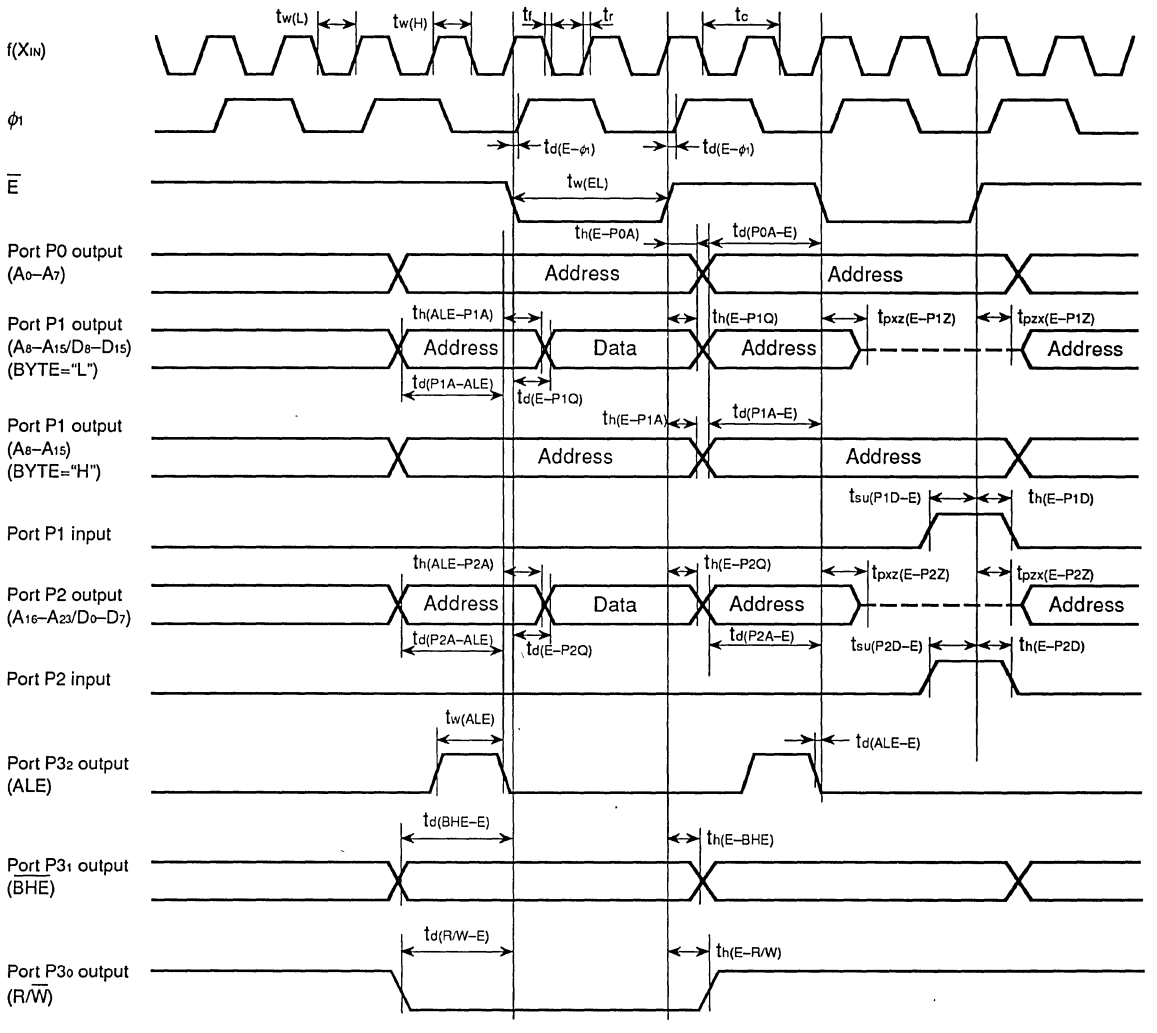


(When wait bit="1" or "0" in common)



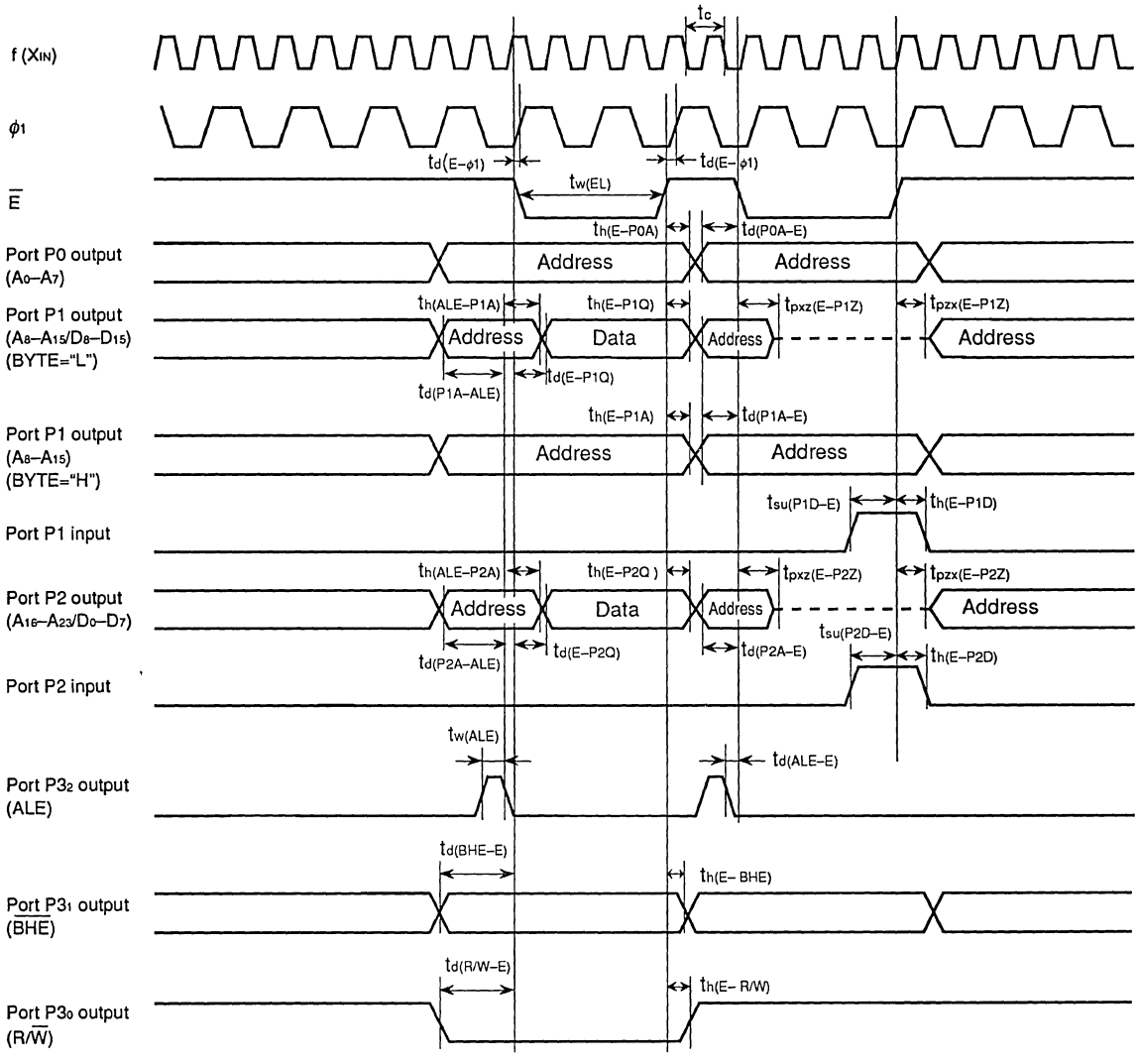
Test conditions
- Vcc=5V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

Memory expansion mode and microprocessor mode (When wait bit="1")



Test conditions
- Vcc=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- Ports P1, P2 input : V_{IL}=0.8V, V_{IH}=2.5V
- Port $P4_1$ input : V_{IL}=1.0V, V_{IH}=4.0V

Memory expansion mode and microprocessor mode (When wait bit="0", and external memory area is accessed)



Test conditions
- Vcc=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- Ports P1, P2 input : V_{IL}=0.8V, V_{IH}=2.5V
- Port P41 input : V_{IL}=1.0V, V_{IH}=4.0V

MEMO

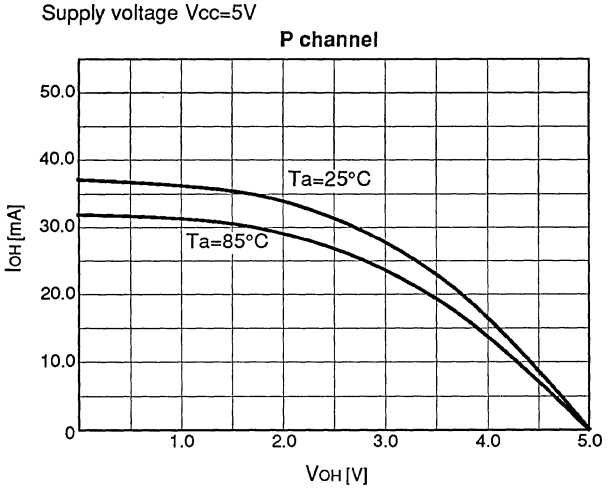
CHAPTER 6
STANDARD CHARACTERISTICS

6.1 Standard characteristics

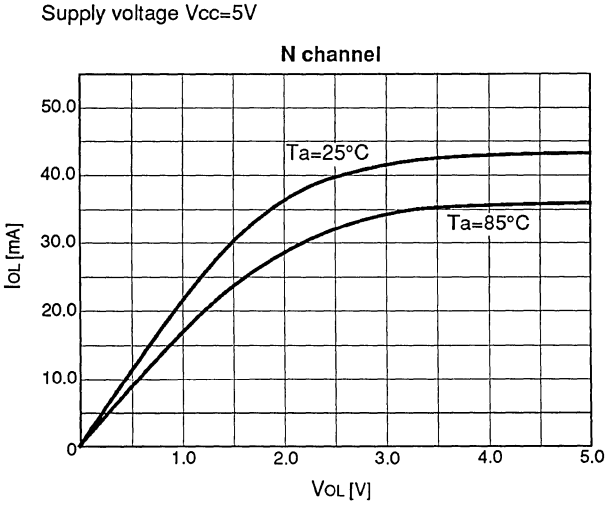
The data described in this chapter are characteristic examples for M37702M2BXXXFP. The data is not guaranteed value. Refer to "Chapter 5. Electrical characteristics" for rated values.

6.1.1 Standard port characteristics

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



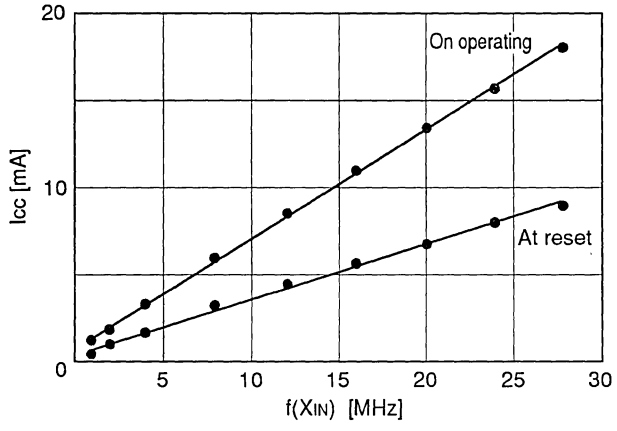
(2) Programmable I/O port (CMOS output) N channel I_{OL}–V_{OL} characteristics



6.1.2 Icc–f(XIN) standard characteristics

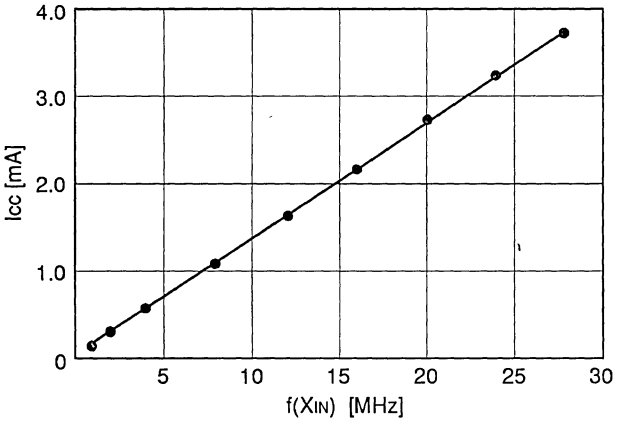
(1) Icc–f(XIN) characteristics on operating and at reset

Measurement condition (Vcc=5V, Ta=25°C, f(XIN): square wave, single-chip mode)



(2) Icc–f(XIN) characteristics during wait

Measurement condition (Vcc=5V, Ta=25°C, f(XIN): square wave, single-chip mode)



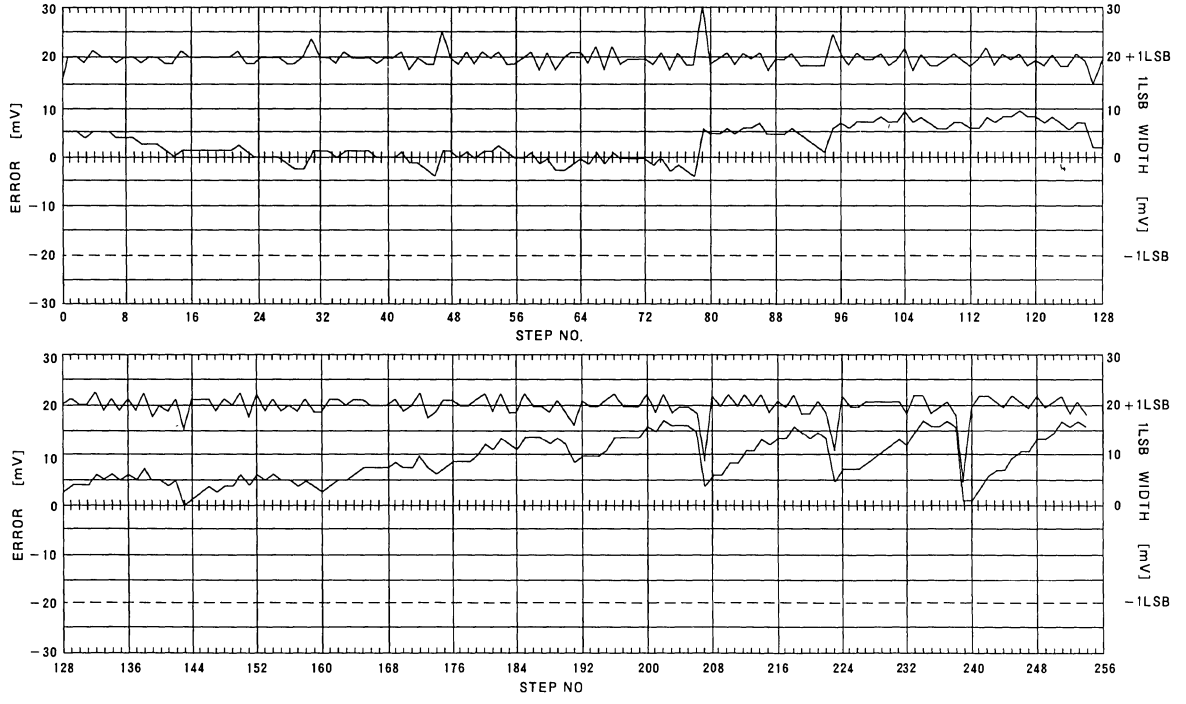
6.1.3 A-D Converter standard characteristics

The lines at the bottom 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 ANi=10mV, but the measured value is 5.0mV. Therefore, the measured point of change is 10+5.0=15.0mV.

The lines at the top 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 22.0mV. Therefore, the differential non-linear error is 22.0–20=2.0mV (0.1LSB).

[Measurement condition]

- V_{CC} = 5.12V
- V_{REF} = 5.12V
- X_{IN} = 25MHz
- Temp. = 25°C



CHAPTER 7
APPLICATION

7.1 Memory expansion

This section describes the external memory expansion of the M37702 group.

7.1.1 Memory expansion model

The memory can be expanded in memory expansion mode and microprocessor mode by using the external bus width selection pin (BYTE pin). Four memory expansion models shown in Table 7.1.1 are available.

(1) Minimum model

External memory area within the 64K bytes memory space is accessed using an 8-bit data bus. No external address latch is necessary. This is the most cost effective expansion model for externally adding an 8-bit element.

(2) Medium model A

Memory space beyond 64K bytes is accessed using an 8-bit data bus as same as the minimum model. An n bit (n≤8) address latch is required to latch the address data from the high-order 8 bits (A_{23} to A_{16}) of the address bus multiplexed with the data bus, but the accessible memory space is expanded up to 16M bytes.

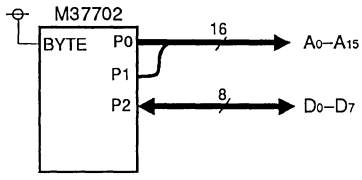
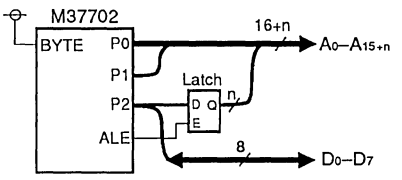
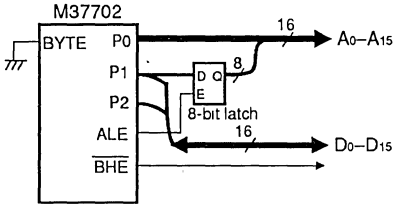
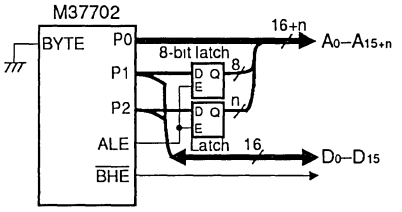
(3) Medium model B

This expansion model limits memory space to within 64K bytes, but provides optimum speed. The external data bus width is 16 bits and expanded memory area can be accessed as fast as internal area if no wait (software wait nor hardware wait due to the $\overline{RDY}$ input) is used. This expansion model requires an 8 bit address latch because the middle 8 bits (A_{15} to A_8) of the address bus are multiplexed with the data bus.

(4) Maximum model

This expansion model uses 16-bit width external data bus to access up to 16M bytes. An 8-bit latch to latch the middle 8 bits (A_{15} to A_8) of the address bus and an n bit (n≤8) latch to latch n bits of the high-order 8 bits (A_{23} to A_{16}) are required.

Table 7.1.1 Memory expansion model

Access space / External bus width	Less than 64K-byte	64K-byte or more
8-bit BYTE = "H"	 Memory expansion model: Minimum model	 Memory expansion model: Medium model A
16-bit BYTE = "L"	 Memory expansion model: Medium model B	 Memory expansion model: Maximum model

The M37702 group must operate in memory expansion mode or microprocessor mode in order to perform memory expansion. When either of these modes is selected, ports P0, P1, P2, P3, and part of P4 function as address/data bus pins or memory expansion related control pins and lose their I/O port functions. Therefore, port expansion must be performed together with memory expansion for applications that use many ports because the number of I/O ports is reduced by 30 compared with single chip mode. Port expansion is described in section "7.2 I/O expansion".
The Vss line of the M37702 group should be reenforced during memory expansion and I/O expansion because the address bus is used at 24 bits.

7.1.2 Memory access time calculation

This section describes how to calculate the memory access time necessary to satisfy the memory expansion timing requirements. The memory access time calculation methods are described for minimum model which does not use the ALE signal and other three models which use the ALE signal.

(1) Minimum model memory access time

Figure 7.1.1 shows the bus timing diagram for the minimum model. The memory access time $t_{a(AD)}$ for the minimum model is obtained by the following equation.

$$t_{a(AD)}=t_{d(P0A/P1A-E)}+t_{w(EL)}-t_{su(P2D-E)}-\{\text{address decode time}\}\cdots\cdots ①$$

Note: $t_{d(P0A/P1A-E)}$ in equation ① represents either $t_{d(P0A-E)}$ or $t_{d(P1A-E)}$.

$t_{d(P0A/P1A-E)}$ and $t_{w(EL)}$ in equation ① are parameters that depend on the operating clock frequency and are calculated by the equations shown in Table 7.1.2. $t_{su(P2D-E)}$ is a constant that depends on the operating clock frequency (8, 16, or 25MHz). Address decode time is the time required to decode the address and make the chip select signal valid.
The data setup time $t_{su(D)}$ for write is obtained from the following equation.

$$t_{su(D)}=t_{w(EL)}-t_{d(E-P2Q)} \cdots\cdots ②$$

If the setup time requested by the device do not satisfy these values, waits must be inserted in the bus cycle with the wait bit or $\overline{RDY}$ input.
Figure 7.1.2 shows the relationship between the memory access time $t_{a(AD)}$ and operating clock frequency $f(X_{IN})$ for a minimum model. The graph in Figure 7.1.2 shows the memory access time ignoring the address decode time in equation ①. Therefore, the actual memory access time is the value of the graph minus the address decode time.

Figure 7.1.3 shows the relationship between the data setup time $t_{su(D)}$ for write and the operating clock frequency.

Table 7.1.2 Operating clock frequency dependent parameter calculation equations and constants

Parameter \ Type	8MHz version M37702M2	16MHz version M37702M2A	25MHz version M3702M2B
$t_{d(P0A-E)}$ $t_{d(P1A-E)}$	$100 + \frac{2\times10^9}{f(X_{IN})} - 250$	$30 + \frac{2\times10^9}{f(X_{IN})} - 125$	$12 + \frac{2\times10^9}{f(X_{IN})} - 80$
$t_{w(EL)}$	$\frac{2\times10^9}{f(X_{IN})} - 30$	←	←
Under software one-wait (the wait bit = "0")	$\frac{4\times10^9}{f(X_{IN})} - 30$	←	←
$t_{su(P2D-E)}$	60	45	30
$t_{d(E-P2Q)}$	110	70	45

(Dimension in ns)

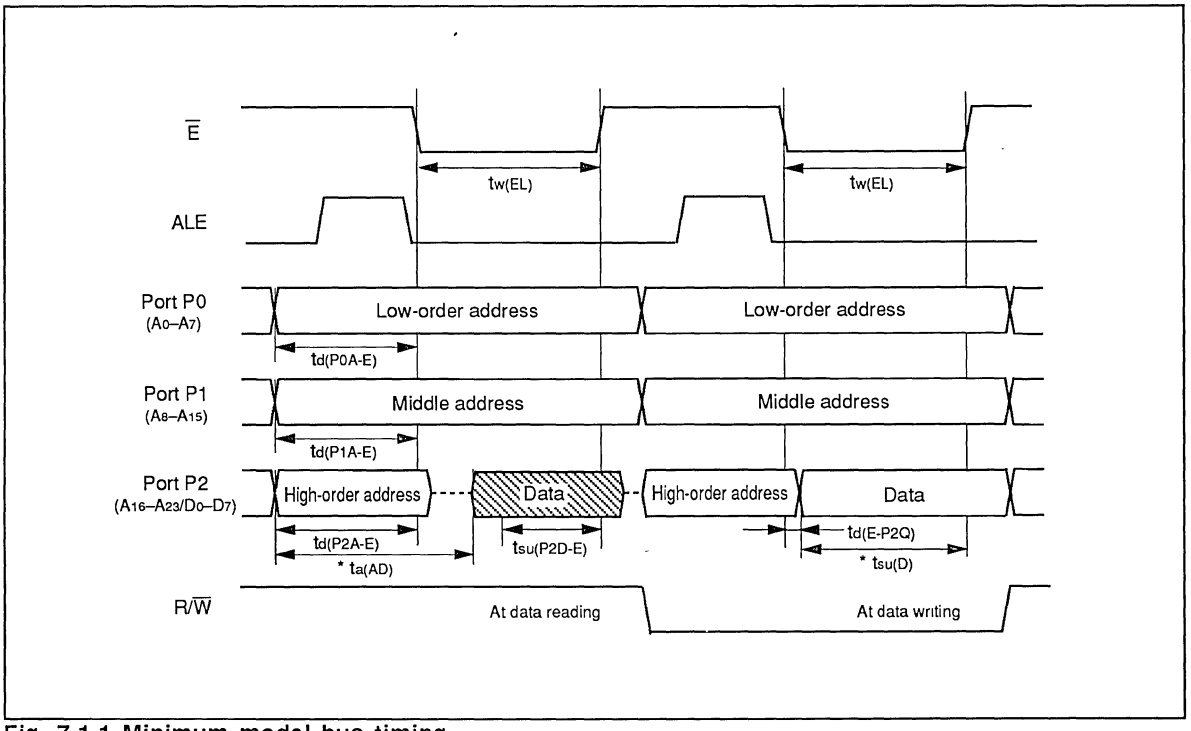


Fig. 7.1.1 Minimum model bus timing

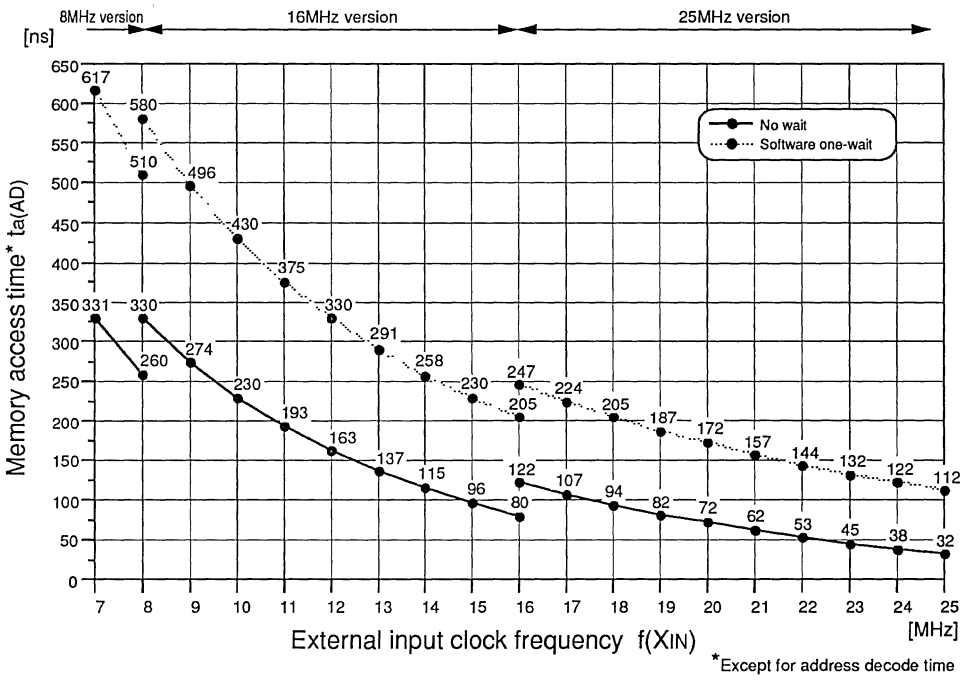


Fig. 7.1.2 Relationship between memory access time and operating clock frequency for minimum model

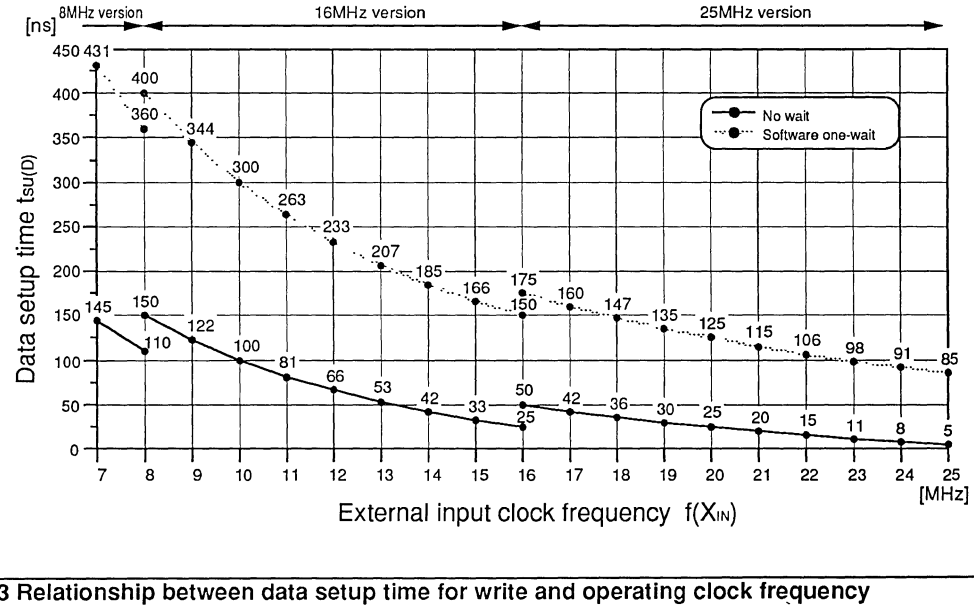


Fig. 7.1.3 Relationship between data setup time for write and operating clock frequency

(2) Medium/maximum model memory access time

Figure 7.1.4 shows the bus timing diagram for medium and maximum models.
ALE signal must be considered in addition to address output delay time when calculating the memory access time for this model.
The memory access time for the medium and maximum model depends on the following two conditions:

●When address is determined before ALE is enabled.
(when $t_{w(ALE)}+t_{d(ALE-E)} \leq t_{d(P1A/P2A-E)}$)

$$t_{a(AD)}=t_{w(ALE)}+t_{d(ALE-E)}+t_{w(EL)}-t_{su(P2D/P1D-E)}$$
$$-\{\text{address latch delay}+\text{address decode time}\} \cdots\cdots ③$$

●When address is determined after ALE is enabled.
(when $t_{w(ALE)}+t_{d(ALE-E)} > t_{d(P1A/P2A-E)}$)

$$t_{a(AD)}=t_{d(P1A/P2A-E)}+t_{w(EL)}-t_{su(P2D/P1D-E)}$$
$$-\{\text{address latch delay}+\text{address decode time}\} \cdots\cdots ④$$

Note: $t_{d(P1A/P2A-E)}$ in equations ③ and ④ represents either $t_{d(P1A-E)}$ or $t_{d(P2A-E)}$ and $t_{su(P2D/P1D-E)}$ represents either $t_{su(P1D-E)}$ or $t_{su(P2D-E)}$.

$t_{d(P1A/P2A-E)}$, $t_{w(ALE)}$, and $t_{w(EL)}$ in equations ③ and ④ are parameters that depend on the operating clock frequency and are calculated by the equations shown in Table 7.1.3. $t_{su(P2D/P1D-E)}$ is a constant that depends on the operating clock frequency (8, 16, or 25MHz). The address latch delay is the delay caused when latching the address data. Note that its value is different under the conditions for equation ③ and equation ④. The address decode time is the time required to decode the address and make the chip select signal valid.
The data setup time for write can be obtained by equation ⑤ similar to the minimum model.

$$t_{su(D)}=t_{w(EL)}-t_{d(E-P2Q/P1Q)} \cdots\cdots ⑤$$

Note: $t_{d(E-P2Q/P1Q)}$ in equation ⑤ represents either $t_{d(E-P1Q)}$ or $t_{d(E-P2Q)}$.

If the setup time requested by the device does not satisfy these values, waits must be inserted in the bus cycle using wait bits or the $\overline{RDY}$ input.
Figure 7.1.5 shows the relationship between memory access time $t_{a(AD)}$ and operating clock frequency $f(X_{IN})$ for medium and maximum models. The graph in Figure 7.1.5 does not take into consideration the address decode time in equations ③ and ④ and shows the memory access time when M74F573 is used for address latch. The address decode time must be subtracted and the actual address latch delay must be considered in order to obtain the actual memory access time.
The relationship between the data setup time $t_{su(D)}$ for write and the operating clock frequency is similar to the minimum model as shown in Figure 7.1.3.

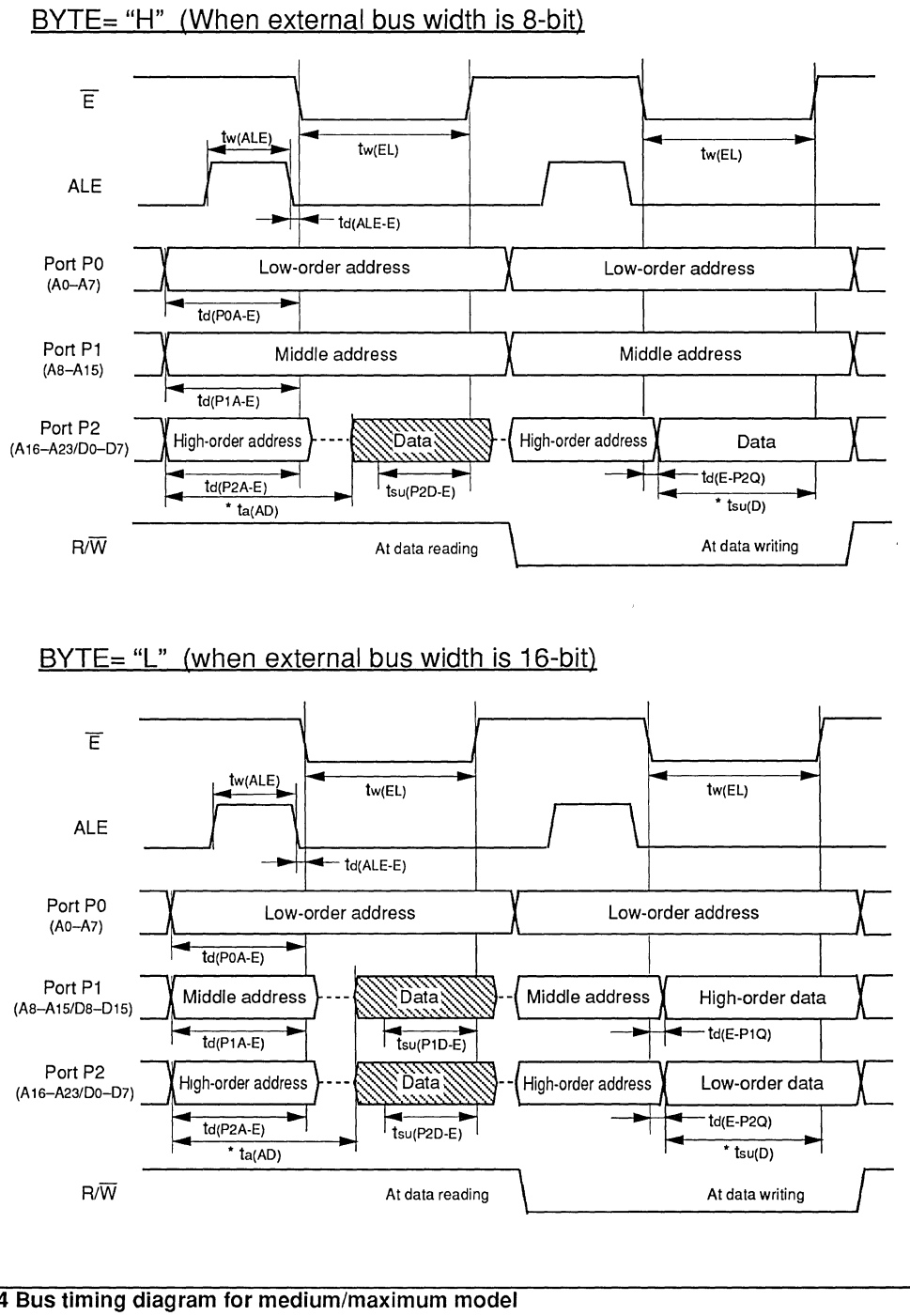


Fig. 7.1.4 Bus timing diagram for medium/maximum model

Table 7.1.3 Parameter equations and constant depending on the operating clock frequency

Parameters \ Products	8MHz version M37702M2	16MHz versionn M37702M2A	25MHz version M37702M2B
$t_{d(P1A-E)}$ $t_{d(P2A-E)}$	$100 + \frac{2\times10^9}{f(X_{IN})} - 250$	$30 + \frac{2\times10^9}{f(X_{IN})} - 125$	$12 + \frac{2\times10^9}{f(X_{IN})} - 80$
$t_{W(ALE)}$	$\frac{1\times10^9}{f(X_{IN})} - 35$	$\frac{1\times10^9}{f(X_{IN})} - 27.5$	$\frac{1\times10^9}{f(X_{IN})} - 18$
$t_{W(EL)}$	$\frac{2\times10^9}{f(X_{IN})} - 30$	←	←
Under software one-wait (wait bit = "0")	$\frac{4\times10^9}{f(X_{IN})} - 30$	←	←
$t_{SU(P2D-E)}$ $t_{SU(P1D-E)}$	60	45	30
$t_{d(E-P2Q)}$ $t_{d(E-P1Q)}$	110	70	45
$t_{d(ALE-E)}$	4	←	←

(Unit : ns)

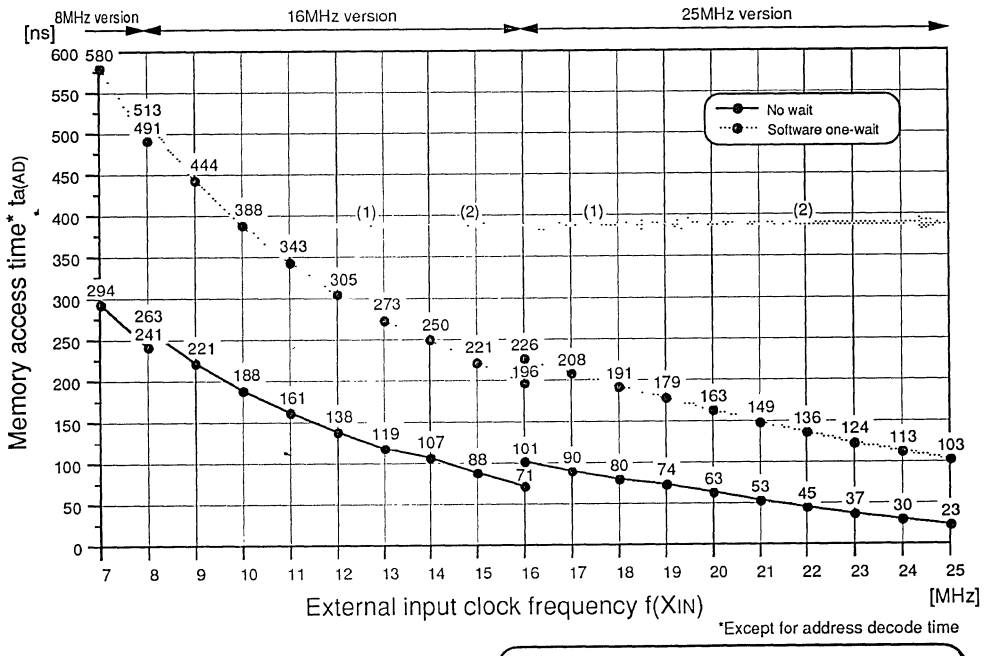


Fig. 7.1.5 Relationship between memory access time and operating clock frequency for midium/maximum model

7.1.3 Memory expansion precautions

When setting $\overline{E}$ signal to "L" level and reading the data on the bus, the M37702 group has an address hold time ($t_{pxz(E-P1Z/P2Z)}$) of up to 5ns. When the $\overline{E}$ signal becomes "H" level and data on the bus is read, at minimum the floating time ($t_{pzx(E-P1Z/P2Z)}$) shown in Table 7.1.4 is reserved before the next address data is output. Therefore, when using devices that output data on the data bus within 5ns from the fall of the $\overline{E}$ signal ($t_{en(OE)} \leq 5ns$) or output data on the data bus for more than $t_{pzx(E-P1Z/P2Z)}$ at the rise of the $\overline{E}$ signal, considerations must be made to prevent bus contention between the address data output by the M37702 group and the data output by the device.

When using devices with $t_{en(0E)}$ that does not satisfy $t_{pxz(E-P1Z/P2Z)}$, generate the device read signal $\overline{OE}$ with only the leading edge of the fall of the $\overline{E}$ signal delayed for few nanoseconds.

When using devices with t_{DF} and $t_{dis(0E)}$ that do not satisfy $t_{pzx(E-P1Z/P2Z)}$, delete the data output by the device using a bus buffer for example. Figures 7.1.7 and 7.1.8 show examples of using a bus buffer. Table 7.1.5 shows Mitsubishi memories that can be connected to the M37702 group without bus buffer. When requesting memory with specifications shown in Table 7.1.5, specify "t_{DF} 15ns microcomputer and kit."

Table 7.1.4 Address hold time and data floating time

Parameter \ Type	8MHz version M37702M2	16MHz versionn M37702M2A	25MHz version M37702M2B
$t_{pxz(E-P1Z)}$ $t_{pxz(E-P2Z)}$	5	5	5
$t_{pxz(E-P1Z)}$ $t_{pxz(E-P2Z)}$	$\frac{1\times10^9}{2\times f(X_{IN})} - 12.5$	$\frac{1\times10^9}{2\times f(X_{IN})} - 6.25$	$\frac{1\times10^9}{2\times f(X_{IN})} - 2.0$

(Dimension in ns)

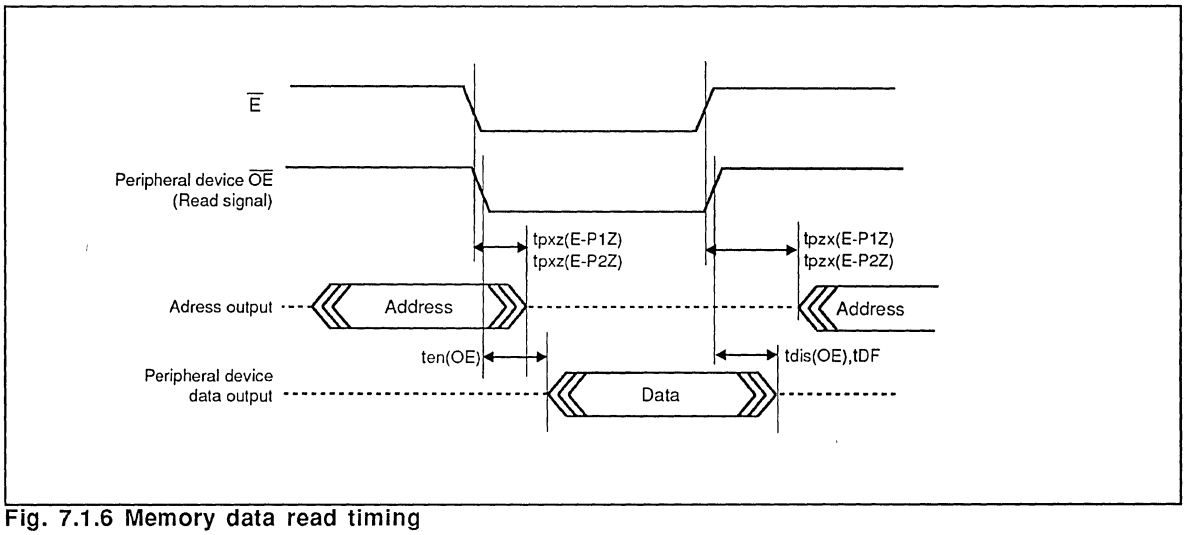


Fig. 7.1.6 Memory data read timing

Table 7.1.5 Memory usable without bus buffer

Memory type	Type name	$t_{DF}/t_{dis\ (OE)}$	Conditions
EPROM	M5M27C256AK-85/-10/-12/-15 or equivalent OTP component M5M27C512AK-10/-12/-15 or equivalent OTP component M5M27C100K-12/-15 M5M27C101K-12/-15 M5M27C102K-12/-15	15ns	$f(X_{IN})\leq 20MHz$ (Note)
	M5L27512K-17/-2	30ns	$f(X_{IN})\leq 8MHz$
SRAM	M5M5256BP-70/-85/-10/-12/-15 or equivalent L,LL types M5M5178P-35/-45/-55	15ns	$f(X_{IN})\leq 20MHz$ (Note)

Note: M74F32 or equivalent component is required for read signal generation when using at 16MHz frequency or greater.

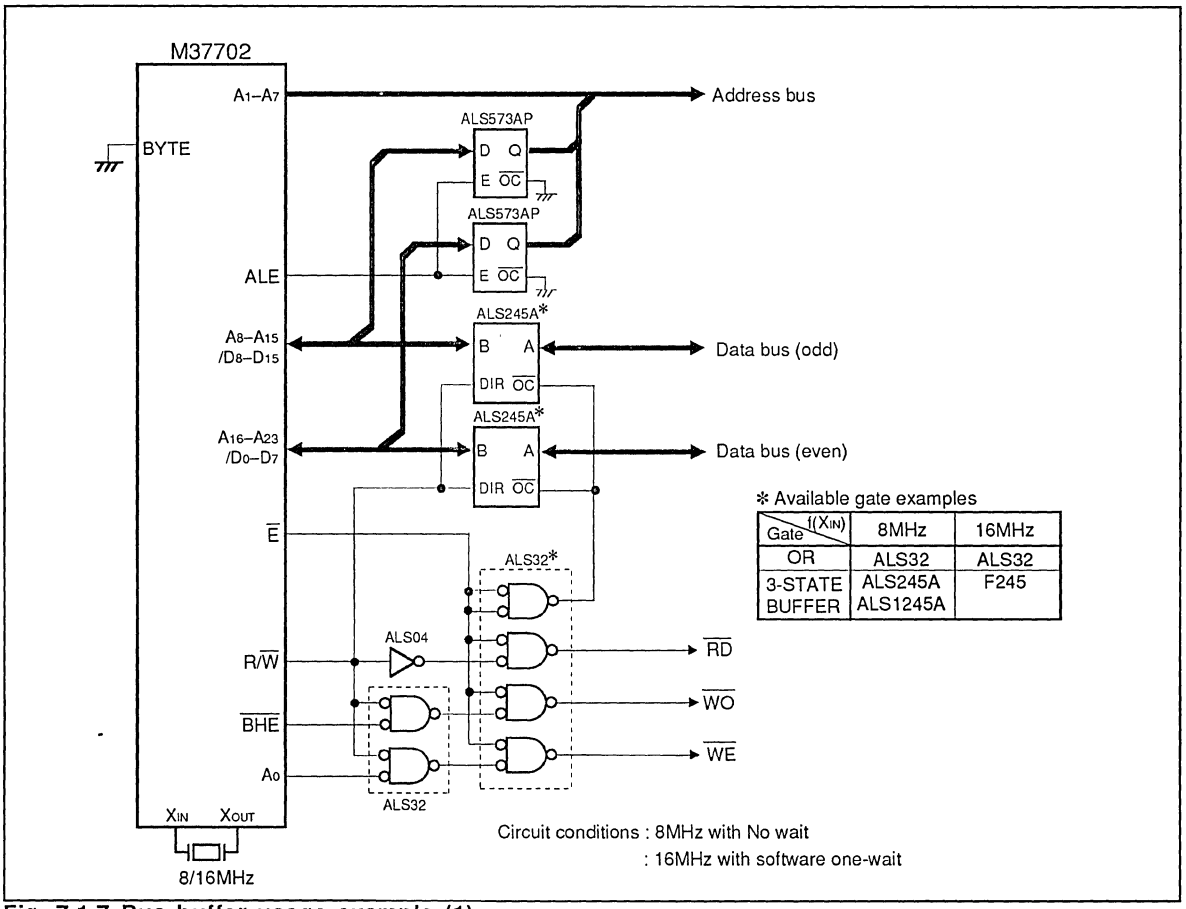


Fig. 7.1.7 Bus buffer usage example (1)

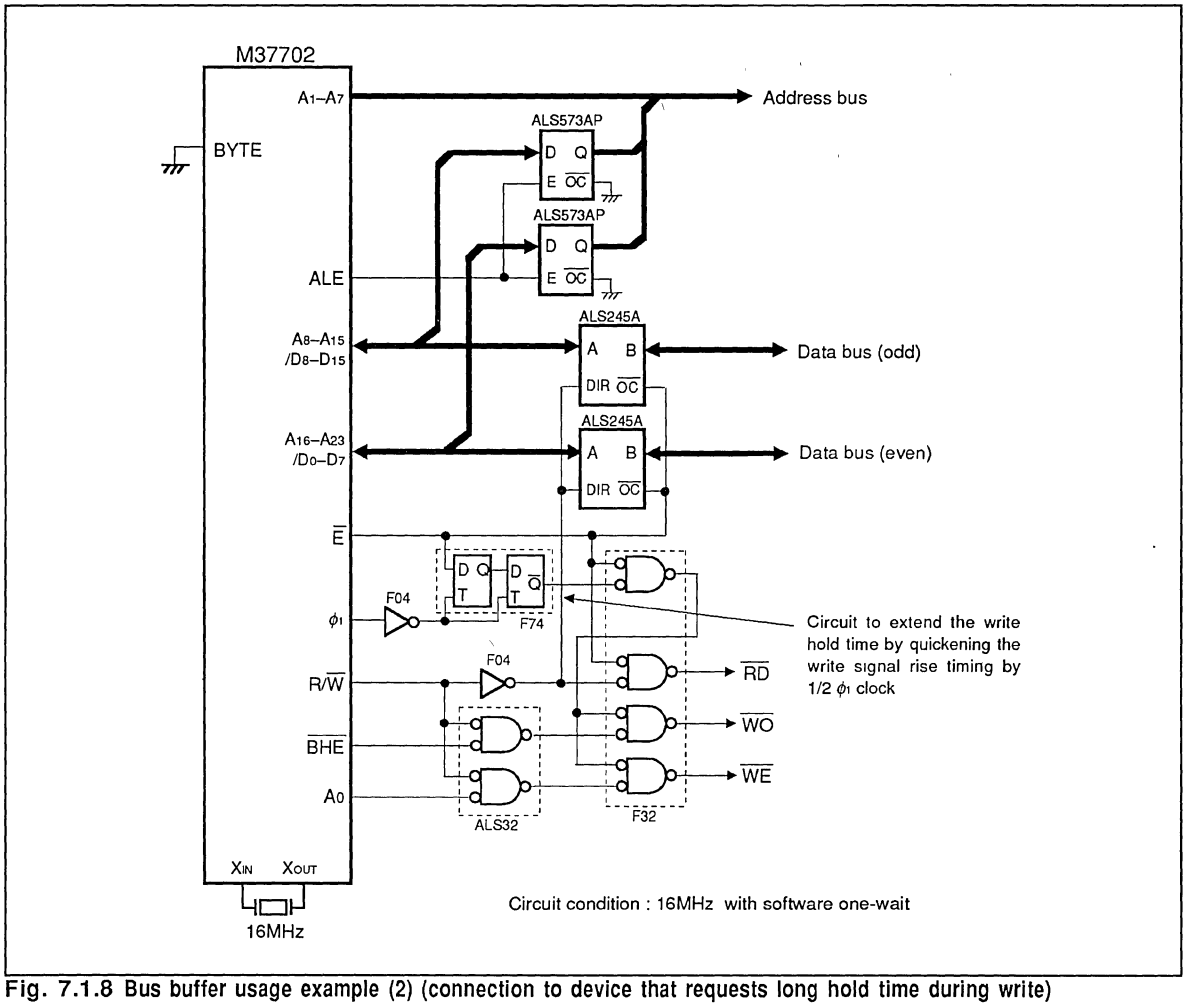


Fig. 7.1.8 Bus buffer usage example (2) (connection to device that requests long hold time during write)

7.1.4 Connection to devices that request long access time

Waits can be inserted in the bus to extend the access time when connecting to devices that request long access time. Wait can be inserted by software one-wait method which sets the wait bit of the processor mode register bit 2 to "0" and inserts waits for 1 cycle of the clock ϕ_1 during the "L" level cycle of the $\overline{E}$ signal, or by hardware using $\overline{RDY}$ to insert any number of wait cycles. Figure 7.1.9 shows an $\overline{RDY}$ generation circuit example for one-wait insertion by hardware. Hardware wait by $\overline{RDY}$ input can generate wait for internal area access as well. Therefore, this circuit example uses a chip select signal to specify the area for inserting a wait. If the clock frequency is 14.9MHz or greater, this circuit cannot be used because the setup time of the $\overline{RDY}$ input at the fall of the clock ϕ_1 is insufficient. Refer to section "7.1.5 (5) Memory expansion example at 25MHz using software one-wait+$\overline{RDY}$" for the use of $\overline{RDY}$ when $\overline{RDY}$ input setup time is insufficient.

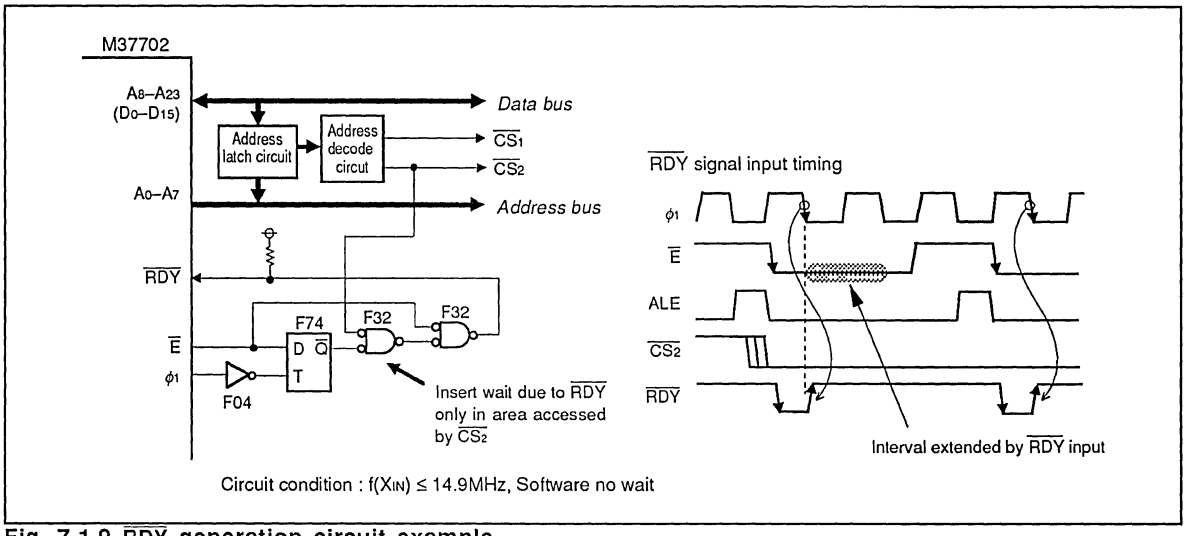


Fig. 7.1.9 RDY generation circuit example

7.1.5 Memory expansion example

Memory expansion examples are described below.

(1) Minimum model memory expansion example

Figure 7.1.10 shows a minimum model memory expansion example. In a minimum model, the R/W signal and E signal are used to generate the memory read signal RD and write signal WR. The BHE signal is unused and must be kept open.

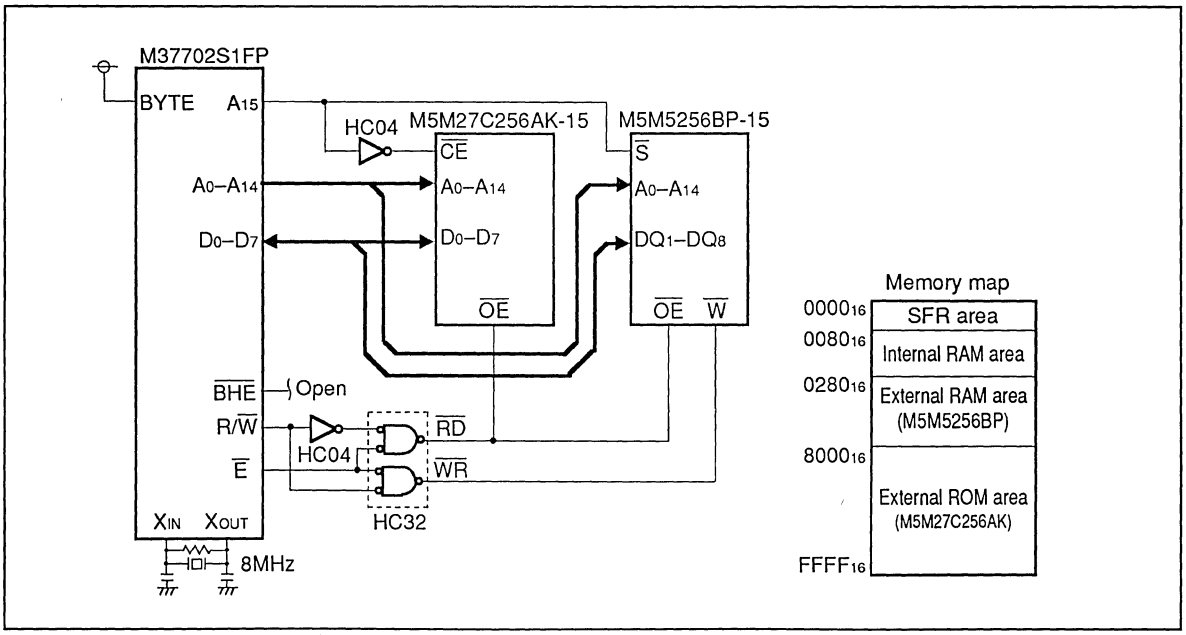


Fig. 7.1.10 Minimum model memory expansion example

(2) Maximum model memory expansion example

Figure 7.1.11 shows a maximum model memory expansion example. In maximum mode, the memory write signal must be separated into even address write signal $\overline{WE}$ and odd address write signal $\overline{WO}$ because the external data bus is used as a 16-bit bus. The even address write signal and odd address write signal are separated by using A_0 of the address and the $\overline{BHE}$ signal. The read signal need not be separated because the microcomputer selectively inputs the data.

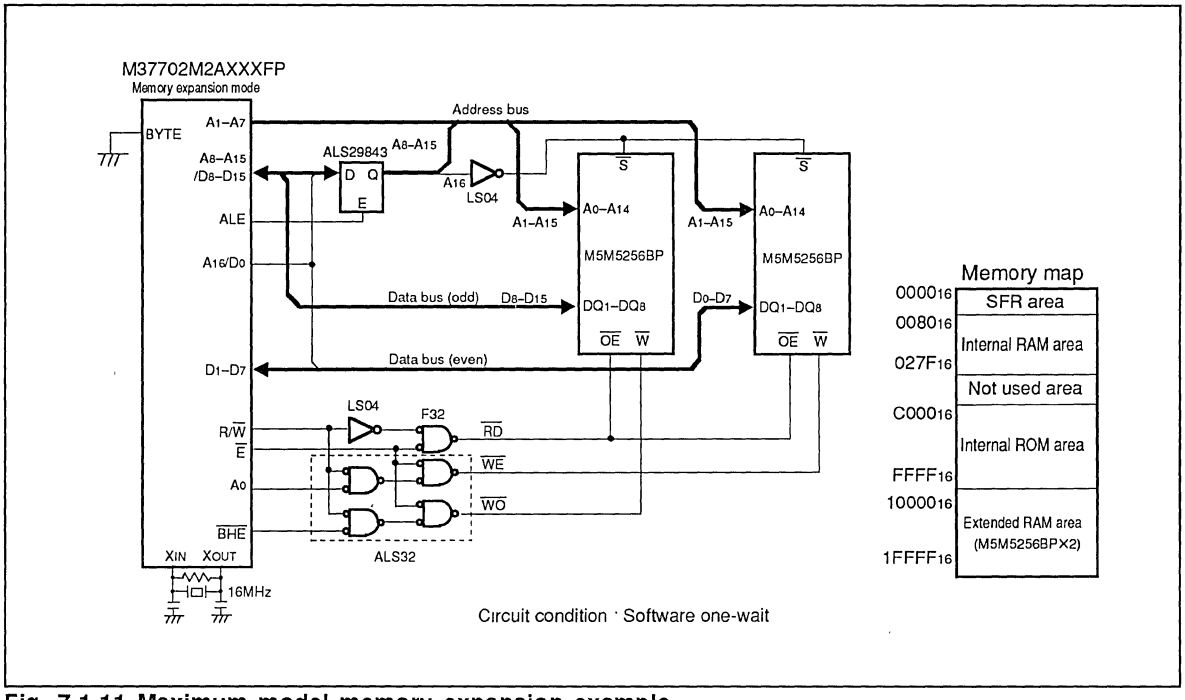


Fig. 7.1.11 Maximum model memory expansion example

(3) Memory expansion example at 20MHz using software one-wait

Figure 7.1.12 shows a memory expansion example at 20MHz using software one-wait (maximum model). No bus buffer is required in this example because M5M27C102K-15 is used as an external ROM (see Table 7.1.5).

(4) Memory expansion example at 25MHz using software one-wait

Figure 7.1.13 shows a memory expansion example at 25MHz using software one-wait (medium model B). This example requires a bus buffer to prevent bus contention.

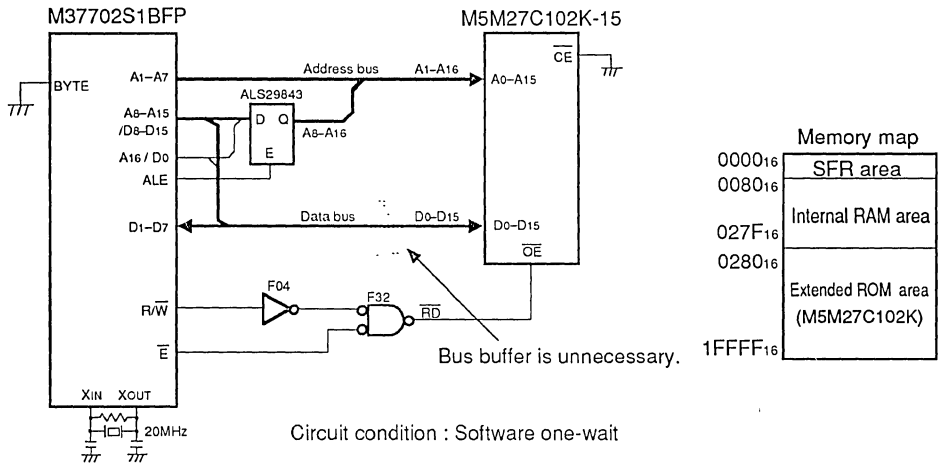


Fig. 7.1.12 Memory expansion example at 20MHz using software one-wait

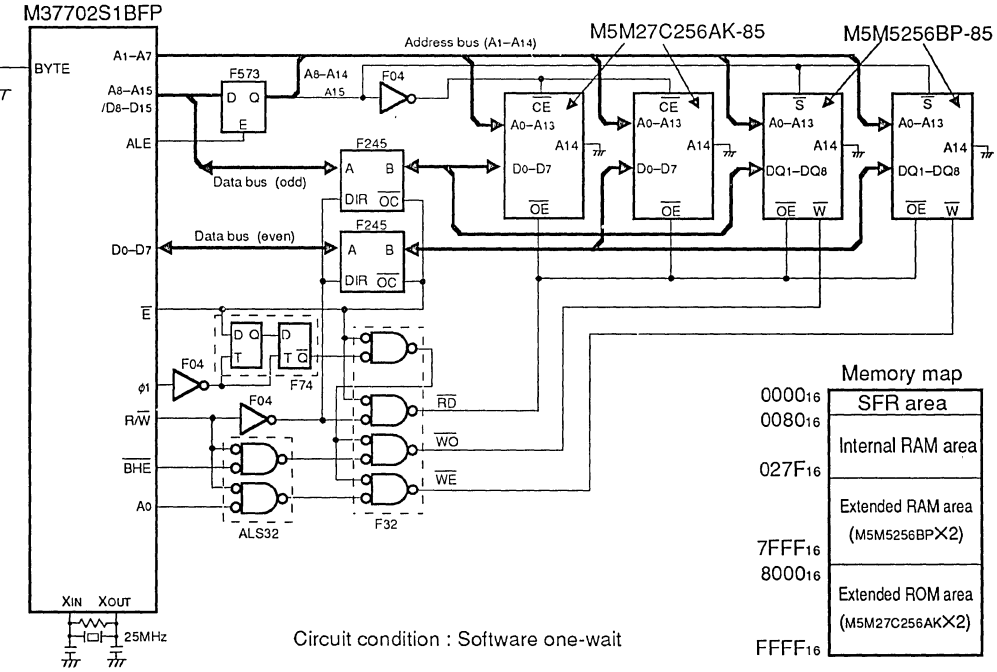
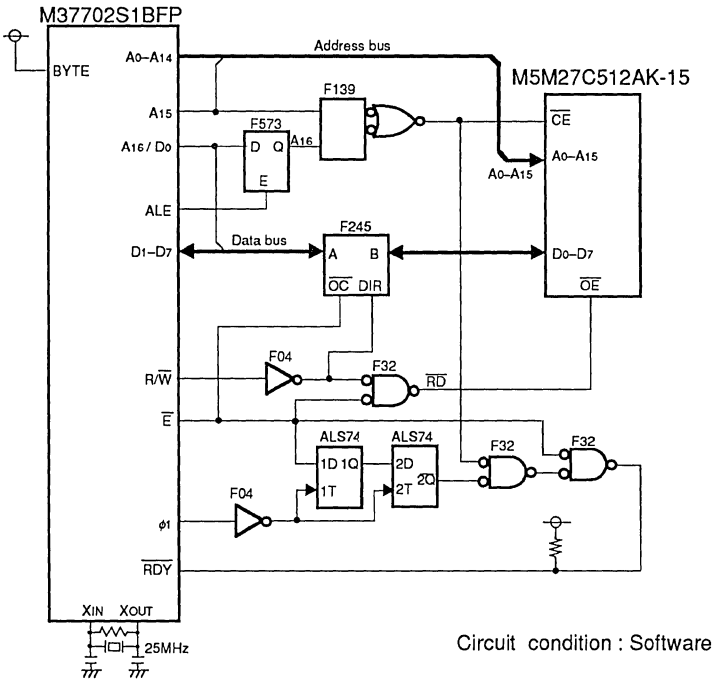


Fig. 7.1.13 Memory expansion example at 25MHz using software one-wait

(5) Memory expansion example at 25MHz using software one-wait+$\overline{RDY}$

Figure 7.1.14 shows a memory expansion example at 25MHz using software one-wait and $\overline{RDY}$ (medium model A). In this example, a software wait and a hardware wait by $\overline{RDY}$ are inserted during ROM access.



Circuit condition : Software one-wait

$\overline{RDY}$ signal input timing

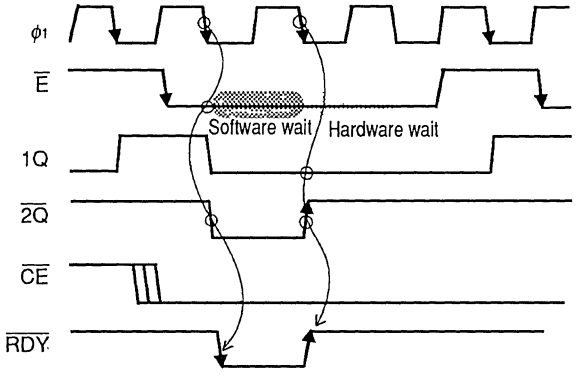


Fig. 7.1.14 Memory expansion example at 25 MHz using software one-wait+$\overline{RDY}$

(6) Memory expansion example using M66800SP/FP

Figures 7.1.15 to 7.1.17 show memory expansion examples using the M66800SP/FP (hereafter referred to as M66800). The M66800 is a memory control IC that can be connected to either word bus or byte bus of the M37702 group. It is equipped with an internal address decoder, read/write signal generator, and an $\overline{RDY}$ signal generator.

{Precautions when expanding memory using the M66800}

Reenforce the GND lines of the M37702 group and M66800 in order to prevent errors due to noise. Also add an 80pF capacitor between the ALE line and GND line for safety.

Figure 7.1.15 shows an example of adding a 64KB EPROM (M5M27C512AK-10) and a 32KB SRAM (M5M5256BP-10) with no wait to the M37702 group that has an 8-bit external bus and operates at 12.288MHz. The $\overline{RDY}$ output pin of the M66800 is open because it operates at no wait.

Figure 7.1.16 shows an example of adding a 64KB EPROM (M5M27C512AK-15) and a 8KB high-speed SRAM (M5M5178P-45) to the M37702 group that has an 8-bit external bus and operates at 15MHz. The 8KB high-speed SRAM is no wait and one-wait is inserted with $\overline{RDY}$ for the EPROM. $\overline{CS5}$ (EPROM chip select signal) is input as the wait request input ($\overline{WRQ}$) to the M66800 in order to limit the area in which the $\overline{RDY}$ is valid.

Figure 7.1.17 shows an example of adding a 128KB EPROM (M5M27C102K-15) and two 32KB SRAMs (M5M5256BP-10) to the M37702 group that has a 16-bit external bus and operates at 12.288MHz. The SRAM is no wait and one-wait is inserted with $\overline{RDY}$ for the EPROM. Similar to Figure 7.1.16, an EPROM chip select signal is used to limit the area in which the $\overline{RDY}$ is valid.

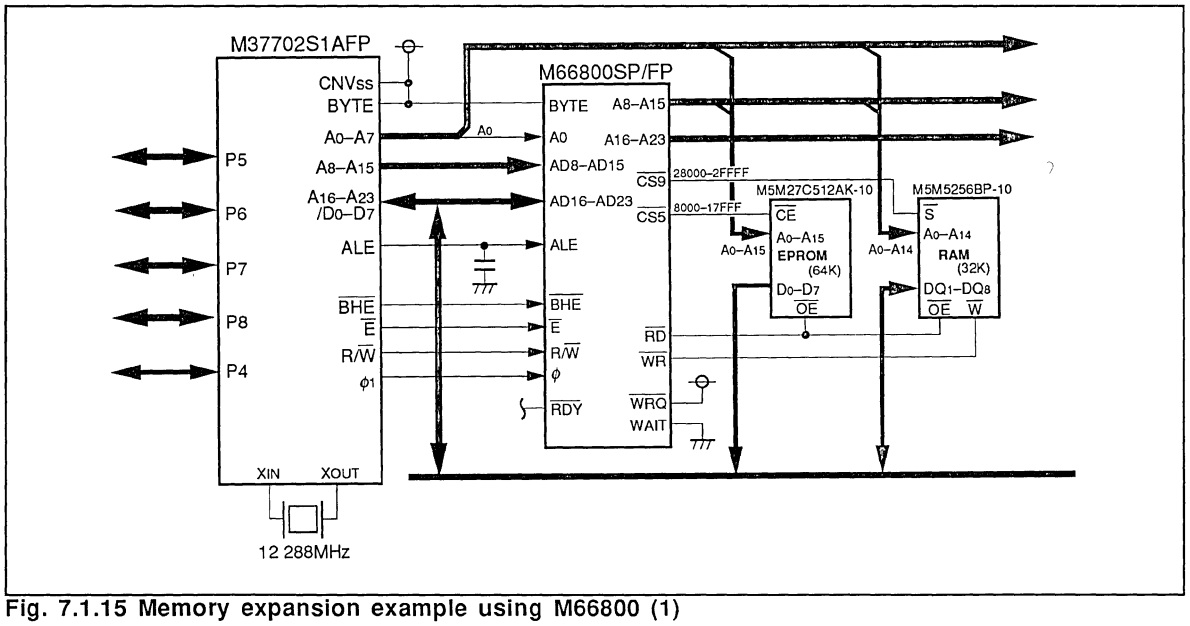


Fig. 7.1.15 Memory expansion example using M66800 (1)

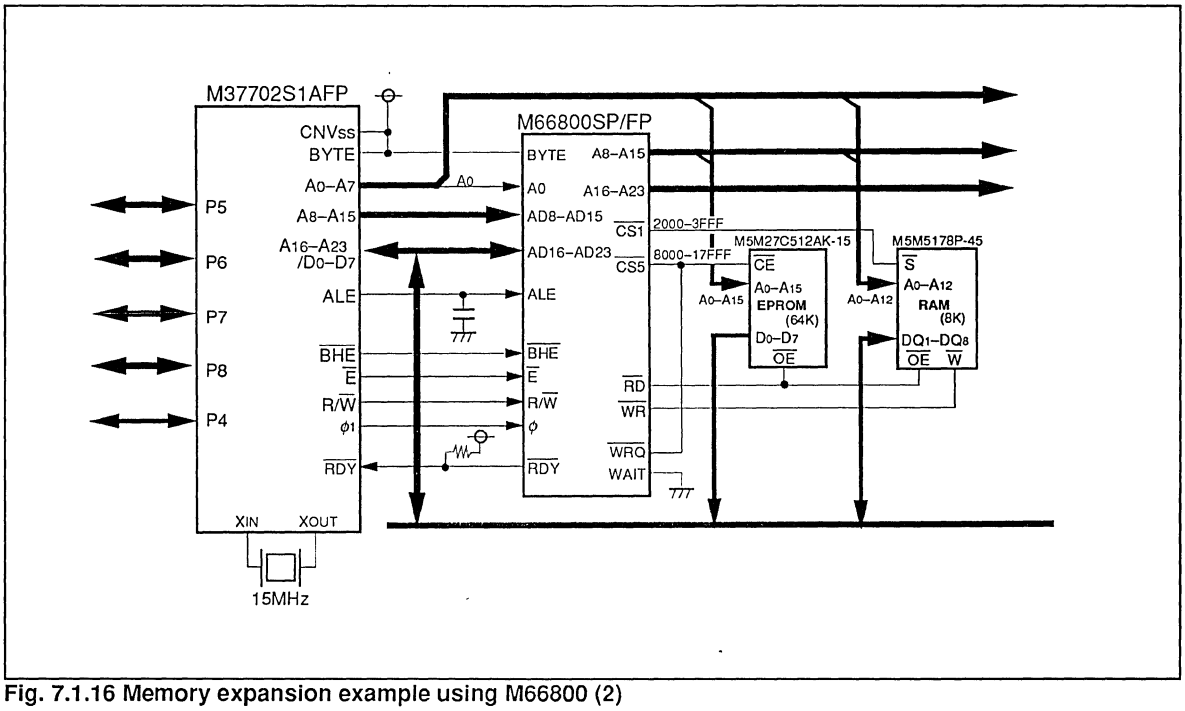


Fig. 7.1.16 Memory expansion example using M66800 (2)

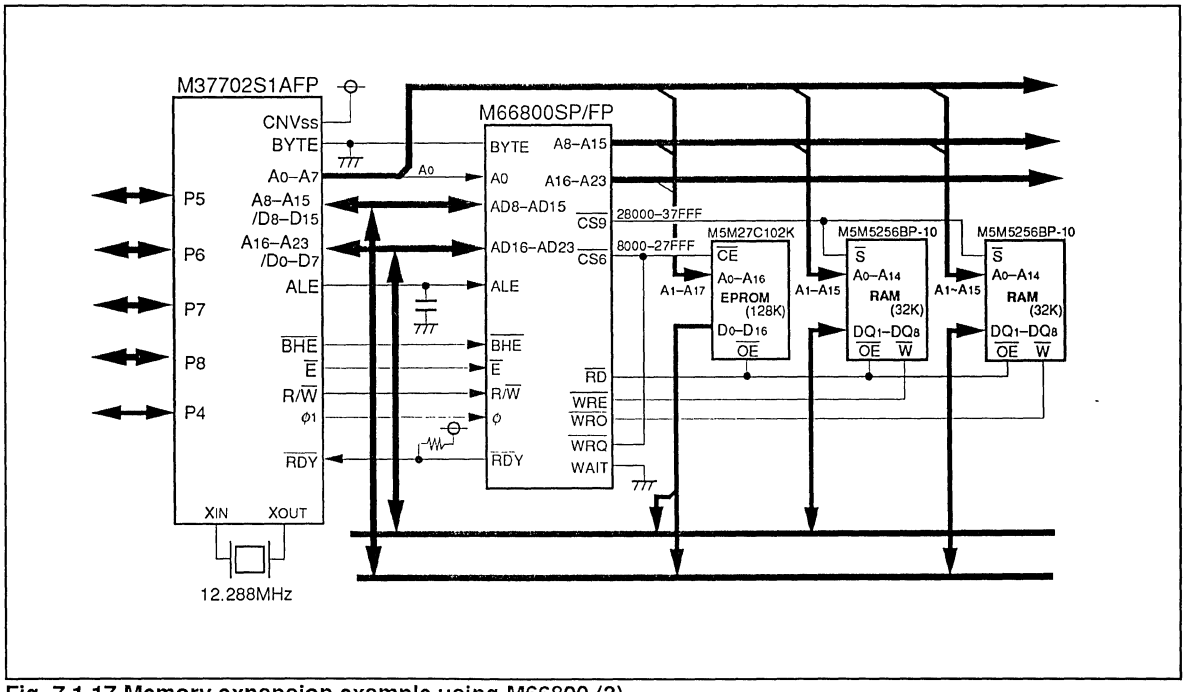


Fig. 7.1.17 Memory expansion example using M66800 (3)

7.2 I/O expansion

I/O expansion is described below.

7.2.1 I/O expansion model

Similar to memory expansion, I/O expansion for the M37702 group in memory expansion mode and microprocessor mode can be performed for the four models shown in Table 7.1.1. Memory mapped I/O is used for I/O expansion. The expansion methods and precautions are the same as for memory expansion.

7.2.2 I/O expansion examples

(1) Port expansion example using M5M82C55AP-2/FP-2

Figure 7.2.1 shows a port expansion example using M5M82C55AP-2/FP-2. This is an expansion example for a medium model B,and M5M82C55AP-2/FP-2 is connected to the even number port side and odd number port side of the M37702 group data bus to expand the I/O port to 48. The device reset signal is supplied from port $P4_3$.

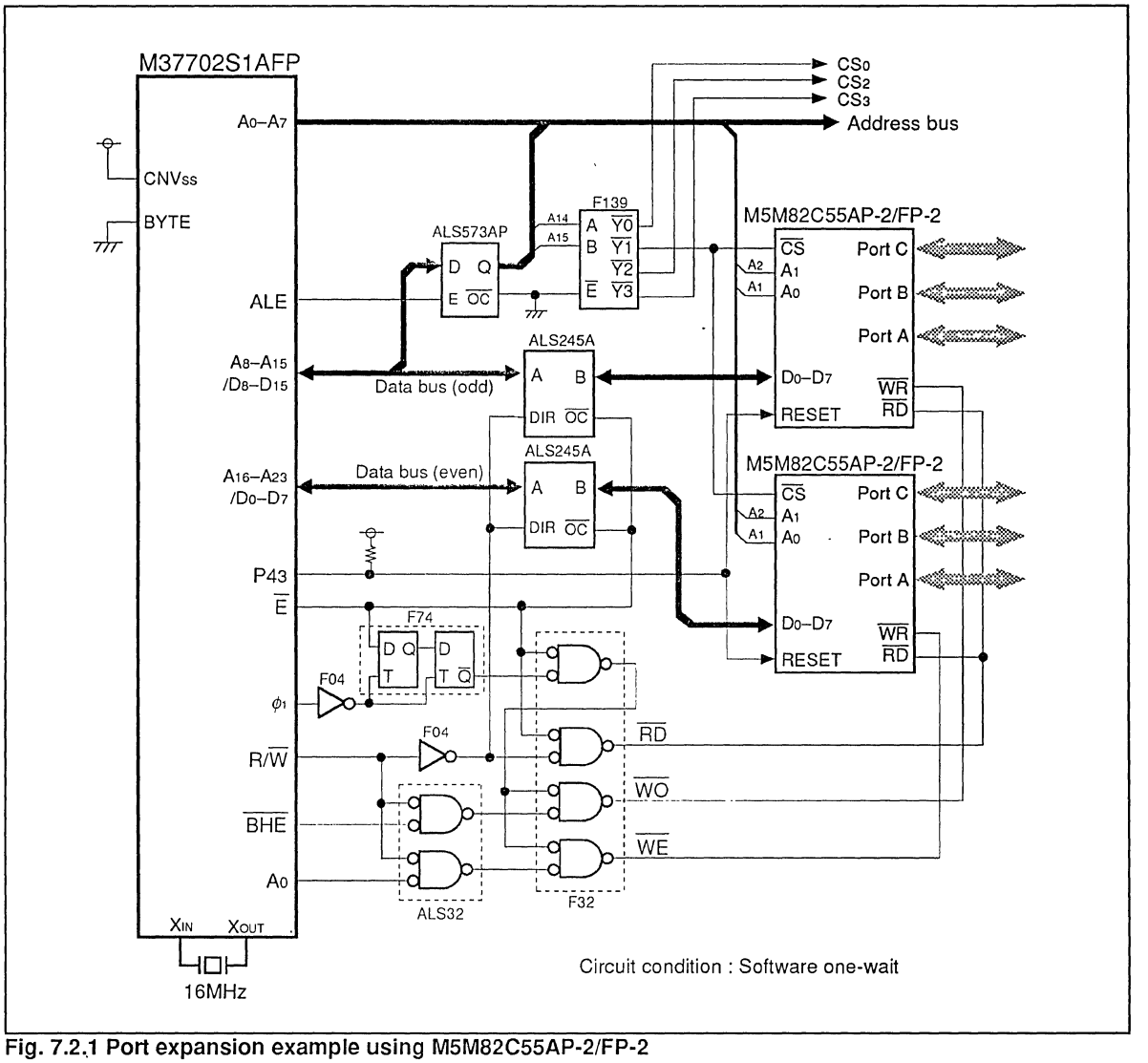


Fig. 7.2.1 Port expansion example using M5M82C55AP-2/FP-2

(2) Port expansion example using M66500SP/FP

Figure 7.2.2 shows a port expansion example using M66500SP/FP. This is an expansion example for a minimum model which adds 24 I/O ports, 16 high-breakdown-voltage output ports, and 4 bits I/O ports.

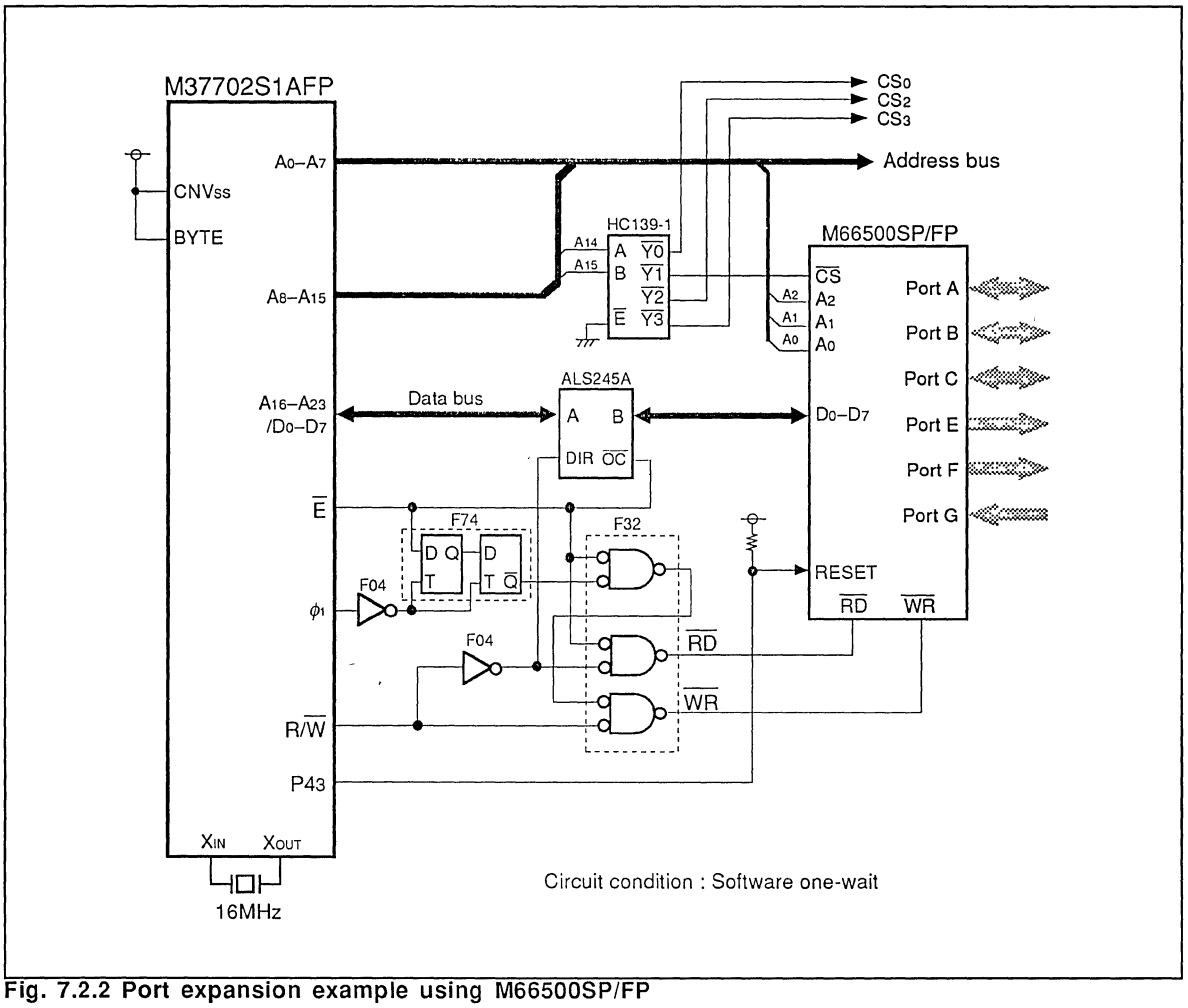


Fig. 7.2.2 Port expansion example using M66500SP/FP

(3) Port expansion example using M37451M4-XXXFP

Figure 7.2.3 shows an expansion example using an 8-bit single-chip microcomputer M37451M4-XXXFP. This is an expansion example for a minimum model and the host bus interface function of the M37451M4-XXXFP is used.

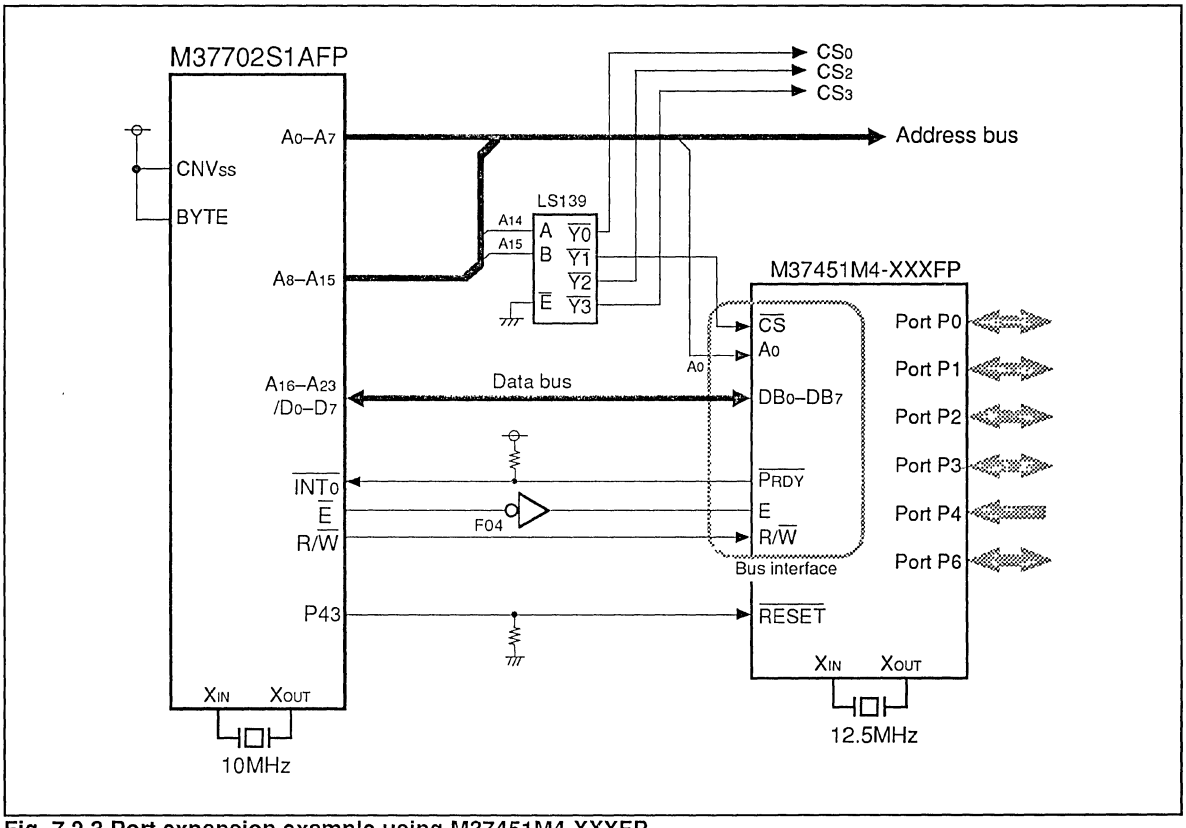


Fig. 7.2.3 Port expansion example using M37451M4-XXXFP

7.3 Program examples

7.3.1 Hardware definition

7.3.2 Initialization examples

7.3.3 Timer modes setting examples
(1) Timer A .....Timer mode
(2) Timer A .....Event counter mode
(3) Timer A .....Two phase pulse processing function
(4) Timer A .....One-shot pulse mode
(5) Timer A .....PWM mode
(6) Timer B .....Pulse period measurement mode
(7) Timer B .....Pulse width measurement mode

7.3.4 Serial I/O modes setting examples
(1) 8-bit UART .....1 byte receive
(2) 8-bit UART .....1 byte transmit
(3) 8-bit UART .....n-byte transmit
(4) Clock synchronous ....1 byte receive
(5) Clock synchronous ....1 byte transmit
(6) Clock synchronous ....n-byte transmit
(7) Error processing

7.3.5 A-D conversion modes setting examples
(1) One-shot mode
(2) Repeat mode
(3) Single sweep mode
(4) Repeat sweep mode

7.3.6 Interrupt processing examples
(1) Interrupt setting example
(2) Interrupt routine processing example.....When memory space is 64K bytes or less.
(3) Interrupt routine processing example.....When memory space exceeds 64K bytes.

7.3.7 Watchdog timer setting examples

7.3.8 Software timer setting examples

7.3.9 Interrupt vector table setting example

7.3.1 Hardware definition

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

  4                     1         .PAGE   'M37702 HARDWARE DEFINE'
  5                     1 ;
  6                     1 ;=================================================
  7                     1 ;             M37702 Hardware define              =
  8                     1 ;=================================================
  9                     1 ;
 10                     1         .SECTION        SFR_AREA
 11                     1 ;
 12                     1 ;=========================================
 13                     1 ;           Port registers                =
 14                     1 ;=========================================
 15                     1 ;
 16                     1         .ORG    000002H
 17  000002             1 P01:                                  ; Port P0, P1 registers
 18 (000002)     1H BYTE 1 P0:     .BLKB   1                     ; Port P0 register
 19 (000003)     1H BYTE 1 P1:     .BLKB   1                     ; Port P1 register
 20                     1 ;-----------------------------------------
 21  000004             1 P01D:                                 ; Port P0, P1 direction registers
 22 (000004)     1H BYTE 1 P0D:    .BLKB   1                     ; Port P0 direction register
 23 (000005)     1H BYTE 1 P1D:    .BLKB   1                     ; Port P1 direction register
 24                     1 ;-----------------------------------------
 25  000006             1 P23:                                  ; Port P2, P3 registers
 26 (000006)     1H BYTE 1 P2:     .BLKB   1                     ; Port P2 register
 27 (000007)     1H BYTE 1 P3:     .BLKB   1                     ; Port P3 register
 28                     1 ;-----------------------------------------
 29  000008             1 P23D:                                 ; Port P2, P3 direction registers
 30 (000008)     1H BYTE 1 P2D:    .BLKB   1                     ; Port P2 direction register
 31 (000009)     1H BYTE 1 P3D:    .BLKB   1                     ; Port P3 direction register
 32                     1 ;-----------------------------------------
 33  00000A             1 P45:                                  ; Port P4, P5 registers
 34 (00000A)     1H BYTE 1 P4:     .BLKB   1                     ; Port P4 register
 35 (00000B)     1H BYTE 1 P5:     .BLKB   1                     ; Port P5 register
 36                     1 ;-----------------------------------------
 37  00000C             1 P45D:                                 ; Port P4, P5 direction registers
 38 (00000C)     1H BYTE 1 P4D:    .BLKB   1                     ; Port P4 direction register
 39 (00000D)     1H BYTE 1 P5D:    .BLKB   1                     ; Port P5 direction register
 40                     1 ;-----------------------------------------
 41  00000E             1 P67:                                  ; Port P6, P7 registers
 42 (00000E)     1H BYTE 1 P6:     .BLKB   1                     ; Port P6 register
 43 (00000F)     1H BYTE 1 P7:     .BLKB   1                     ; Port P7 register
 44                     1 ;-----------------------------------------
 45  000010             1 P67D:                                 ; Port P6, P7 direction registers
 46 (000010)     1H BYTE 1 P6D:    .BLKB   1                     ; Port P6 direction register
 47 (000011)     1H BYTE 1 P7D:    .BLKB   1                     ; Port P7 direction register
 48                     1 ;-----------------------------------------
 49 (000012)     2H BYTE 1 P8:     .BLKB   2                     ; Port P8 register
 50                     1 ;-----------------------------------------
 51 (000014)     2H BYTE 1 P8D:    .BLKB   2                     ; Port P8 direction register
 52                     1 ;
```

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

  53                    1         .PAGE
  54                    1 ;
  55                    1 ;=============================================
  56                    1 ;       A-D conversion registers              =
  57                    1 ;=============================================
  58                    1 ;
  59                    1         .ORG    00001EH
  60 (00001E)     1H BYTE   1 ADCON:  .BLKB   1                   ; A-D control register
  61 (00001F)     1H BYTE   1 ADSPS:  .BLKB   1                   ; A-D sweep pin selection register
  62                    1 ;---------------------------------------
  63 (000020)     2H BYTE   1 AD0:    .BLKB   2                   ; A-D register 0
  64                    1 ;---------------------------------------
  65 (000022)     2H BYTE   1 AD1:    .BLKB   2                   ; A-D register 1
  66                    1 ;---------------------------------------
  67 (000024)     2H BYTE   1 AD2:    .BLKB   2                   ; A-D register 2
  68                    1 ;---------------------------------------
  69 (000026)     2H BYTE   1 AD3:    .BLKB   2                   ; A-D register 3
  70                    1 ;---------------------------------------
  71 (000028)     2H BYTE   1 AD4:    .BLKB   2                   ; A-D register 4
  72                    1 ;---------------------------------------
  73 (00002A)     2H BYTE   1 AD5:    .BLKB   2                   ; A-D register 5
  74                    1 ;---------------------------------------
  75 (00002C)     2H BYTE   1 AD6:    .BLKB   2                   ; A-D register 6
  76                    1 ;---------------------------------------
  77 (00002E)     2H BYTE   1 AD7:    .BLKB   2                   ; A-D register 7
  78                    1 ;
  79                    1 ;=============================================
  80                    1 ;       Serial I/O 0 registers                =
  81                    1 ;=============================================
  82                    1 ;
  83  000030            1 SOMxB:                                    ;
  84 (000030)     1H BYTE   1 SOMR:   .BLKB   1                   ; UART0 transmit/receive mode register
  85 (000031)     1H BYTE   1 SOBRG:  .BLKB   1                   ; UART0 baud rate generator (BRG)
  86                    1 ;---------------------------------------
  87  000032            1 SOTB:                                     ; UART0 transmission buffer registers
  88 (000032)     1H BYTE   1 SOTBL:  .BLKB   1                   ; UART0 transmission buffer register (low-order)
  89 (000033)     1H BYTE   1 SOTBH:  .BLKB   1                   ; UART0 transmission buffer register (high-order)
  90                    1 ;---------------------------------------
  91  000034            1 SOC:                                      ; UART0 transmit/receive control registers
  92 (000034)     1H BYTE   1 SOCL:   .BLKB   1                   ; UART0 transmit/receive control register 0
  93 (000035)     1H BYTE   1 SOCH:   .BLKB   1                   ; UART0 transmit/receive control register 1
  94                    1 ;---------------------------------------
  95  000036            1 SORB:                                     ; UART0 receive buffer registers
  96 (000036)     1H BYTE   1 SORBL:  .BLKB   1                   ; UART0 receive buffer register (low-order)
  97 (000037)     1H BYTE   1 SORBH:  .BLKB   1                   ; UART0 receive buffer register (high-order)
  98                    1 ;
  99                    1 ;=============================================
 100                    1 ;       Serial I/O 1 registers                =
 101                    1 ;=============================================
 102                    1 ;
 103  000038            1 S1MxB:                                    ;
 104 (000038)     1H BYTE   1 S1MR:   .BLKB   1                   ; UART1 transmit/receive mode register
 105 (000039)     1H BYTE   1 S1BRG:  .BLKB   1                   ; UART1 baud rate generator (BRG)
 106                    1 ;---------------------------------------
 107  00003A            1 S1TB:                                     ; UART1 transmission buffer registers
 108 (00003A)     1H BYTE   1 S1TBL:  .BLKB   1                   ; UART1 transmission buffer register (low-order)
 109 (00003B)     1H BYTE   1 S1TBH:  .BLKB   1                   ; UART1 transmission buffer register (high-order)
 110                    1 ;---------------------------------------
 111  00003C            1 S1C:                                      ; UART1 transmit/receive control registers
 112 (00003C)     1H BYTE   1 S1CL:   .BLKB   1                   ; UART1 transmit/receive control register 0
 113 (00003D)     1H BYTE   1 S1CH:   .BLKB   1                   ; UART1 transmit/receive control register 1
 114                    1 ;---------------------------------------
 115  00003E            1 S1RB:                                     ; UART1 receive buffer registers
 116 (00003E)     1H BYTE   1 S1RBL:  .BLKB   1                   ; UART1 receive buffer register (low-order)
 117 (00003F)     1H BYTE   1 S1RBH:  .BLKB   1                   ; UART1 receive buffer register (high-order)
 118                    1 ;
```

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 119                    1        .PAGE
 120                    1 ;
 121                    1 ;=============================================
 122                    1 ;       Timer registers                     =
 123                    1 ;=============================================
 124                    1 ;
 125 (000040)     2H BYTE   1 TABSR:  .BLKB  2                    ; Count start flag
 126                    1 ;-----------------------------------------
 127 (000042)     2H BYTE   1 ONSF:   .BLKB  2                    ; One-shot start flag
 128                    1 ;-----------------------------------------
 129 (000044)     2H BYTE   1 UDF:    .BLKB  2                    ; Up/down flag
 130                    1 ;-----------------------------------------
 131  000046            1 TA0:                                    ; Timer A0 register
 132 (000046)     1H BYTE   1 TA0L:   .BLKB  1                    ; Timer A0 register (low-order)
 133 (000047)     1H BYTE   1 TA0H:   .BLKB  1                    ; Timer A0 register (high-order)
 134                    1 ;-----------------------------------------
 135  000048            1 TA1:                                    ; Timer A1 register
 136 (000048)     1H BYTE   1 TA1L:   .BLKB  1                    ; Timer A1 register (low-order)
 137 (000049)     1H BYTE   1 TA1H:   .BLKB  1                    ; Timer A1 register (high-order)
 138                    1 ;-----------------------------------------
 139  00004A            1 TA2:                                    ; Timer A2 register
 140 (00004A)     1H BYTE   1 TA2L:   .BLKB  1                    ; Timer A2 register (low-order)
 141 (00004B)     1H BYTE   1 TA2H:   .BLKB  1                    ; Timer A2 register (high-order)
 142                    1 ;-----------------------------------------
 143  00004C            1 TA3:                                    ; Timer A3 register
 144 (00004C)     1H BYTE   1 TA3L:   .BLKB  1                    ; Timer A3 register (low-order)
 145 (00004D)     1H BYTE   1 TA3H:   .BLKB  1                    ; Timer A3 register (high-order)
 146                    1 ;-----------------------------------------
 147  00004E            1 TA4:                                    ; Timer A4 register
 148 (00004E)     1H BYTE   1 TA4L:   .BLKB  1                    ; Timer A4 register (low-order)
 149 (00004F)     1H BYTE   1 TA4H:   .BLKB  1                    ; Timer A4 register (high-order)
 150                    1 ;-----------------------------------------
 151  000050            1 TB0:                                    ; Timer B0 register
 152 (000050)     1H BYTE   1 TB0L:   .BLKB  1                    ; Timer B0 register (low-order)
 153 (000051)     1H BYTE   1 TB0H:   .BLKB  1                    ; Timer B0 register (high-order)
 154                    1 ;-----------------------------------------
 155  000052            1 TB1:                                    ; Timer B1 register
 156 (000052)     1H BYTE   1 TB1L:   .BLKB  1                    ; Timer B1 register (low-order)
 157 (000053)     1H BYTE   1 TB1H:   .BLKB  1                    ; Timer B1 register (high-order)
 158                    1 ;-----------------------------------------
 159  000054            1 TB2:                                    ; Timer B2 register
 160 (000054)     1H BYTE   1 TB2L:   .BLKB  1                    ; Timer B2 register (low-order)
 161 (000055)     1H BYTE   1 TB2H:   .BLKB  1                    ; Timer B2 register (high-order)
 162                    1 ;-----------------------------------------
 163  000056            1 TA01MR:                                 ; Timer A0, A1 mode registers
 164 (000056)     1H BYTE   1 TA0MR:  .BLKB  1                    ; Timer A0 mode register
 165 (000057)     1H BYTE   1 TA1MR:  .BLKB  1                    ; Timer A1 mode register
 166                    1 ;-----------------------------------------
 167  000058            1 TA23MR:                                 ; Timer A2, A3 mode registers
 168 (000058)     1H BYTE   1 TA2MR:  .BLKB  1                    ; Timer A2 mode register
 169 (000059)     1H BYTE   1 TA3MR:  .BLKB  1                    ; Timer A3 mode register
 170                    1 ;-----------------------------------------
 171  00005A            1 TA4B0MR:                                ; Timer A4, B0 mode registers
 172 (00005A)     1H BYTE   1 TA4MR:  .BLKB  1                    ; Timer A4 mode register
 173 (00005B)     1H BYTE   1 TB0MR:  .BLKB  1                    ; Timer B0 mode register
 174                    1 ;-----------------------------------------
 175  00005C            1 TB12MR:                                 ; Timer B1, B2 mode registers
 176 (00005C)     1H BYTE   1 TB1MR:  .BLKB  1                    ; Timer B1 mode register
 177 (00005D)     1H BYTE   1 TB2MR:  .BLKB  1                    ; Timer B2 mode register
 178                    1 ;
 179                    1 ;=============================================
 180                    1 ;       Processor mode register             =
 181                    1 ;=============================================
 182                    1 ;
 183 (00005E)     2H BYTE   1 PMR:    .BLKB  2                    ; Processor mode register
 184                    1 ;
```

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 185                    1        .PAGE
 186                    1 ;
 187                    1 ;=========================================
 188                    1 ;       Watchdog timer registers              =
 189                    1 ;=========================================
 190                    1 ;
 191 (000060)     1H BYTE   1 WDT:    .BLKB   1                   ; Watchdog timer
 192 (000061)     1H BYTE   1 WDC:    .BLKB   1                   ; Watchdog timer frequency selection flag
 193                    1 ;
 194                    1 ;=========================================
 195                    1 ;       Interrupt control registers           =
 196                    1 ;=========================================
 197                    1 ;
 198                    1             .ORG      000070H
 199  000070            1 ADxSOTIC:
 200 (000070)     1H BYTE   1 ADIC:   .BLKB   1                   ; A-D conversion interrupt control register
 201 (000071)     1H BYTE   1 SOTIC:  .BLKB   1                   ; UART0 transmission interrupt control register
 202                    1 ;-----------------------------------------
 203  000072            1 SORxS1TIC:
 204 (000072)     1H BYTE   1 SORIC:  .BLKB   1                   ; UART0 receive interrupt control register
 205 (000073)     1H BYTE   1 S1TIC:  .BLKB   1                   ; UART1 transmission interrupt control register
 206                    1 ;-----------------------------------------
 207  000074            1 S1RxTA0IC:
 208 (000074)     1H BYTE   1 S1RIC:  .BLKB   1                   ; UART1 receive interrupt control register
 209 (000075)     1H BYTE   1 TA0IC:  .BLKB   1                   ; Timer A0 interrupt control register
 210                    1 ;-----------------------------------------
 211  000076            1 TA1xTA2IC:
 212 (000076)     1H BYTE   1 TA1IC:  .BLKB   1                   ; Timer A1 interrupt control register
 213 (000077)     1H BYTE   1 TA2IC:  .BLKB   1                   ; Timer A2 interrupt control register
 214                    1 ;-----------------------------------------
 215  000078            1 TA3xTA4IC:
 216 (000078)     1H BYTE   1 TA3IC:  .BLKB   1                   ; Timer A3 interrupt control register
 217 (000079)     1H BYTE   1 TA4IC:  .BLKB   1                   ; Timer A4 interrupt control register
 218                    1 ;-----------------------------------------
 219  00007A            1 TB0xTB1IC:
 220 (00007A)     1H BYTE   1 TB0IC:  .BLKB   1                   ; Timer B0 interrupt control register
 221 (00007B)     1H BYTE   1 TB1IC:  .BLKB   1                   ; Timer B1 interrupt control register
 222                    1 ;-----------------------------------------
 223  00007C            1 TB2xINT0IC:
 224 (00007C)     1H BYTE   1 TB2IC:  .BLKB   1                   ; Timer B2 interrupt control register
 225 (00007D)     1H BYTE   1 INT0IC: .BLKB   1                   ; INT0 interrupt control register
 226                    1 ;-----------------------------------------
 227  00007E            1 INT1xINT2IC:
 228 (00007E)     1H BYTE   1 INT1IC: .BLKB   1                   ; INT1 interrupt control register
 229 (00007F)     1H BYTE   1 INT2IC: .BLKB   1                   ; INT2 interrupt control register
 230                    1 ;-----------------------------------------
 231                    1 ;
```

7.3.2 Initialization examples

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 242                                .PAGE   'MAIN ROUTINE'
 243                         ;
 244                         ;================================================
 245                         ;       Main routine                             =
 246                         ;================================================
 247                         ;
 248                         ;=============================================
 249                         ;       Initialization routine                   =
 250                         ;=============================================
 251                         ;
 252                         ;-------------------------------------
 253                         ;       Assembler declarations          =
 254                         ;-------------------------------------
 255                         ;
 256                         ;  These are assembler declarations.
 257                         ;  Flags and registers used
 258                         ;  in the program must match
 259                         ;  these declarations.
 260                         ;
 261                                .SECTION        PROGRAM         ; Section name
 262                                .ORG    0C000H                  ; Start address
 263                                .DATA   16                      ; Data length
 264                                .INDEX  16                      ; Index register length
 265                                .DP     0000H                   ; Direct page
 266                                .DT     00H                     ; Data bank
 267                         ;
 268                         ;-------------------------------------
 269                         ;       Initialize registers and flags  =
 270                         ;-------------------------------------
 271                         ;
 272  00C000                 INITIAL:
 273  00C000  78                     SEI                             ; Disable interrupt
 274  00C001  C238                   CLP     m,x,D                   ; Set data and index register length 16 bits
 275                                                                 ; Binary operation mode
 276  00C003  A27F02                 LDX     #027FH                  ;
 277  00C006  9A                     TXS                             ; Stack pointer=Highest address of RAM
 278  00C007  A90000                 LDA     A,#0                    ;
 279  00C00A  5B                     TAD                             ; Direct page=0H to FFH
 280  00C00B  89C200                 LDT     #0                      ; Data bank=Bank 0
 281  00C00E *645E2000       L       LDM     #0020H,PMR              ; Interrupt priority detection time=2φ cycles
 282                                                                 ; Single-chip mode
 283                         ;
 284                         ;-------------------------------------
 285                         ;       Clear RAM                       =
 286                         ;-------------------------------------
 287                         ;
 288  00C012  A27E02                 LDX     #027EH                  ;
 289  00C015  A90000                 LDA     A,#0                    ; Set initial value (0H) in accumulator A
 290  00C018                 RAM_CLR:                                ;
 291  00C018 *9500                   STA     A,0,X                   ; Write content of accumulator A to specified
 292  00C01A  CA                     DEX                             ; RAM in 16-bit unit
 293  00C01B  CA                     DEX                             ;
 294  00C01C  E07E00                 CPX     #07EH                   ;
 295  00C01F  D0F7           L       BNE     RAM_CLR                 ;
 296                         ;
 297                         ;-------------------------------------
 298                         ;       Change register model           =
 299                         ;       (if necessary)                  =
 300                         ;-------------------------------------
 301                         ;
 302                                .DATA   8                       ; Data length
 303  00C021  F8                     SEM                             ; Data length 8 bits
 304                         ;
 305                         ;
 306                         ;
 307  00C022  2000D0         L       JSR     SUB_T1_1
 308  00C025  80FE           L  E_LOOP: BRA     E_LOOP
 309
```

7.3.3 Timer modes setting examples

(1) Timer A .....Timer mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 327                    1        .PAGE
 328                    1 ;
 329                    1 ;=========================================
 330                    1 ;      Timer setting example 1            =
 331                    1 ;      [Timer A: Timer mode]              =
 332                    1 ;=========================================
 333                    1 ;
 334                    1 ;-------------------------------------
 335                    1 ;      When there is no pulse output   =
 336                    1 ;-------------------------------------
 337                    1 ;      Timer          :Timer A0
 338                    1 ;      Mode           :Timer mode
 339                    1 ;      Pulse output   :Disabled
 340                    1 ;      Gate function  :Disabled
 341                    1 ;      Count source   :f(XIN)/2
 342                    1 ;
 343  00D000            1 SUB_T1_1:
 344                    1        .DATA  8
 345  00D000  E224      1        SEP    m,I                  ; Data length 8 bits, disable interrupt
 346  00D002 *144001    L1       CLB    #00000001B,TABSR     ; Stop TA0 count
 347  00D005 *645600    L1       LDM    #00000000B,TA0MR     ; Timer mode, disable pulse output and gate function
 348                    1                                    ; Count source=f(XIN)/2
 349                    1        .DATA  16
 350  00D008  D8        1        CLM                         ; Data length 16 bits
 351  00D009 *6446CF07  L1       LDM    #2000-1,TA0          ; Set counter value
 352                    1        .DATA  8
 353  00D00D  F8        1        SEM                         ; Data length 8 bits
 354  00D00E *647507    L1       LDM    #00000111B,TA0IC     ; Clear TA0 interrupt request bit
 355                    1                                    ; Enable TA0 interrupt (level 7)
 356  00D011 *044001    L1       SEB    #00000001B,TABSR     ; Start TA0 count
 357  00D014  58        1        CLI                         ; Enable interrupt
 358                    1 ;
 359                    1 ;      ※ Hereafter, interrupt request occurs after each timer A0 overflows.
 360                    1 ;
 361  00D015  60        1        RTS
 362                    1 ;
 363                    1 ;-------------------------------------
 364                    1 ;      When there is pulse output      =
 365                    1 ;-------------------------------------
 366                    1 ;      Timer          :Timer A0
 367                    1 ;      Mode           :Timer mode
 368                    1 ;      Pulse output   :Enabled
 369                    1 ;      Gate function  :Disabled
 370                    1 ;      Count source   :f(XIN)/2
 371                    1 ;
 372  00D016            1 SUB_T1_2:
 373                    1        .DATA  8
 374  00D016  F8        1        SEM                         ; Data length 8 bits
 375  00D017 *144001    L1       CLB    #00000001B,TABSR     ; Stop  TA0 count
 376  00D01A *645604    L1       LDM    #00000100B,TA0MR     ; Timer mode, enable pulse output, disable gate function
 377                    1                                    ; Count source=f(XIN)/2
 378                    1        .DATA  16
 379  00D01D  D8        1        CLM                         ; Data length 16 bits
 380  00D01E *6446CF07  L1       LDM    #2000-1,TA0          ; Set counter value
 381                    1        .DATA  8
 382  00D022  F8        1        SEM                         ; Data length 8 bits
 383  00D023 *647500    L1       LDM    #00000000B,TA0IC     ; Disable TA0 interrupt
 384  00D026 *044001    L1       SEB    #00000001B,TABSR     ; Start TA0 count
 385                    1 ;
 386                    1 ;      ※ Hereafter, the phase of output pulse inverts every time timer A0 overflows.
 387                    1 ;
 388  00D029  60        1        RTS
 389                    1 ;
 390                    1 ;
```

(2) Timer A .....Event counter mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 391                      1        .PAGE
 392                      1 ;
 393                      1 ;===========================================
 394                      1 ;       Timer setting example 2             =
 395                      1 ;       [Timer A: Event counter mode]       =
 396                      1 ;===========================================
 397                      1 ;
 398                      1 ;       Timer          :Timer A1
 399                      1 ;       Mode           :Event counter mode
 400                      1 ;       Pulse output   :Disabled
 401                      1 ;       Count phase    :Fall
 402                      1 ;       Up/down switch :Content of up/down flag
 403                      1 ;
 404  00D02A              1 SUB_T2:
 405                      1        .DATA   8
 406  00D02A  E224        1        SEP     m,I                    ; Data length 8 bits, disable interrupt
 407  00D02C *140D08      L1       CLB     #00001000B,P5D         ; Set P53/TA1IN pin to input mode
 408  00D02F *144002      L1       CLB     #00000010B,TABSR       ; Stop TA1 count
 409  00D032 *645701      L1       LDM     #00000001B,TA1MR       ; Event counter mode, disable pulse output
 410                      1                                       ; Select falling edge count and up/down flag
 411  00D035 *144402      L1       CLB     #00000010B,UDF         ; Set to decrement count
 412                      1        .DATA   16
 413  00D038  D8          1        CLM                            ; Data length 16 bits
 414  00D039 *64486300    L1       LDM     #100-1,TA1             ; Set counter value
 415                      1        .DATA   8
 416  00D03D  F8          1        SEM                            ; Data length 8 bits
 417  00D03E *647603      L1       LDM     #00000011B,TA1IC       ; Clear TA1 interrupt request bit
 418                      1                                       ; Enable TA1 interrupt (level 3)
 419  00D041 *044002      L1       SEB     #00000010B,TABSR       ; Start TA1 count
 420  00D044  58          1        CLI                            ; Enable interrupt
 421                      1 ;
 422                      1 ;       ※ Hereafter, an interrupt request occurs at every 100 count.
 423                      1 ;
 424  00D045  60          1        RTS
 425                      1 ;
 426                      1 ;
```

(3) Timer A .....Two phase pulse processing function

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 427                      1        .PAGE
 428                      1 ;
 429                      1 ;================================================
 430                      1 ;      Timer setting example 3                   =
 431                      1 ;      [Timer A: Two phase pulse processing function]  =
 432                      1 ;================================================
 433                      1 ;
 434                      1 ;      Timer          :Timer A2
 435                      1 ;      Mode           :Event counter mode
 436                      1 ;                      (two phase pulse signal processing function)
 437                      1 ;
 438  00D046              1 SUB_T3:
 439                      1        .DATA   8
 440  00D046  F8          1        SEM                          ; Data length 8 bits
 441  00D047 *140D30      L1        CLB     #00110000B,P5D      ; Set P54/TA2OUT, P55/TA2IN pin to input mode
 442  00D04A *144004      L1        CLB     #00000100B,TABSR    ; Stop TA2 count
 443  00D04D *645811      L1        LDM     #00010001B,TA2MR    ; Event counter mode
 444  00D050 *644420      L1        LDM     #00100000B,UDF      ; Select two phase pulse signal processing function
 445                      1        .DATA   16
 446  00D053  D8          1        CLM                          ; Data length 16 bits
 447  00D054 *644A0080    L1        LDM     #8000H,TA2          ; Set counter value
 448                      1        .DATA   8
 449  00D058  F8          1        SEM                          ; Data length 8 bits
 450  00D059 *647700      L1        LDM     #00000000B,TA2IC    ; Disable TA2 interrupt
 451  00D05C *044004      L1        SEB     #00000100B,TABSR    ; Start TA2 count
 452                      1 ;
 453                      1 ;      ※ Hereafter, count is performed with input signals to TA2OUT and TA2IN pins.
 454                      1 ;
 455  00D05F  60          1        RTS
 456                      1 ;
```

(4) Timer A .....One-shot pulse mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 457                    1        .PAGE
 458                    1 ;
 459                    1 ;=============================================
 460                    1 ;      Timer setting example 4                =
 461                    1 ;      [Timer A: One-shot pulse mode]         =
 462                    1 ;=============================================
 463                    1 ;
 464                    1 ;-----------------------------------------
 465                    1 ;      When internal trigger is selected =
 466                    1 ;-----------------------------------------
 467                    1 ;      Timer          :Timer A3
 468                    1 ;      Mode           :One-shot pulse mode
 469                    1 ;      Start trigger  :One-shot start flag
 470                    1 ;                      (Software trigger)
 471                    1 ;      Count source   :f(XIN)/2
 472                    1 ;
 473  00D060            1 SUB_T4_1:
 474                    1        .DATA  8
 475  00D060  F8        1        SEM                          ; Data length 8 bits
 476  00D061 *144008    L1       CLB    #00001000B,TABSR      ; Stop TA3 count
 477  00D064 *645906    L1       LDM    #00000110B,TA3MR      ; One-shot pulse mode, software trigger
 478                    1                                     ; Count source=f(XIN)/2
 479                    1        .DATA  16
 480  00D067  D8        1        CLM                          ; Data length 16 bits
 481  00D068 *644CD007  L1       LDM    #2000,TA3             ; Set counter value
 482                    1        .DATA  8
 483  00D06C  F8        1        SEM                          ; Data length 8 bits
 484  00D06D *647800    L1       LDM    #00000000B,TA3IC      ; Disable TA3 interrupt
 485  00D070 *044008    L1       SEB    #00001000B,TABSR      ; Enable TA3 count
 486  00D073 *644208    L1       LDM    #00001000B,ONSF       ; Generate TA3 one-shot trigger
 487                    1 ;
 488                    1 ;      ※ Generate one-shot pulse
 489                    1 ;
 490  00D076  60        1        RTS
 491                    1 ;
 492                    1 ;
 493                    1 ;-----------------------------------------
 494                    1 ;      When external trigger is selected =
 495                    1 ;-----------------------------------------
 496                    1 ;      Timer          :Timer A3
 497                    1 ;      Mode           :One-shot pulse mode
 498                    1 ;      Start trigger  :Falling edge of TA3IN
 499                    1 ;                      (Hardware trigger)
 500                    1 ;      Count source   :f(XIN)/2
 501                    1 ;
 502  00D077            1 SUB_T4_2:
 503                    1        .DATA  8
 504  00D077  F8        1        SEM                          ; Data length 8 bits
 505  00D078 *140D80    L1       CLB    #10000000B,P5D        ; Set P57/TA3IN pin to input mode
 506  00D07B *144008    L1       CLB    #00001000B,TABSR      ; Stop TA3 count
 507  00D07E *645916    L1       LDM    #00010110B,TA3MR      ; One-shot pulse mode, external trigger
 508                    1                                     ; Count source=f(XIN/2)
 509                    1        .DATA  16
 510  00D081  D8        1        CLM                          ; Data length 16 bits
 511  00D082 *644CD007  L1       LDM    #2000,TA3             ; Set counter value
 512                    1        .DATA  8
 513  00D086  F8        1        SEM                          ; Data length 8 bits
 514  00D087 *647800    L1       LDM    #00000000B,TA3IC      ; Disable TA3 interrupt
 515  00D08A *044008    L1       SEB    #00001000B,TABSR      ; Enable TA3 count
 516                    1 ;
 517                    1 ;      ※ Generate one-shot pulse at falling edge input to TA3IN pin
 518                    1 ;
 519  00D08D  60        1        RTS
 520                    1 ;
 521                    1 ;
```

(5) Timer A .....PWM mode

```
SEQ.  LOC.   OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 522                    1        .PAGE
 523                    1 ;
 524                    1 ;=========================================
 525                    1 ;      Timer setting example 5            =
 526                    1 ;      [Timer A: PWM mode]                =
 527                    1 ;=========================================
 528                    1 ;-----------------------------------
 529                    1 ;      16-bit PWM                  =
 530                    1 ;-----------------------------------
 531                    1 ;      Timer          :Timer A4
 532                    1 ;      Mode           :PWM mode
 533                    1 ;      Start trigger  :Count start flag
 534                    1 ;                      (Software trigger)
 535                    1 ;      PWM mode       :16-bit PWM mode
 536                    1 ;      Count source   :f(XIN)/2
 537                    1 ;
 538  00D08E            1 SUB_T5_1:
 539                    1        .DATA  8
 540  00D08E  F8        1        SEM                          ; Data length 8 bits
 541  00D08F *144010    L1       CLB    #00010000B,TABSR      ; Stop TA4 count
 542  00D092 *645A07    L1       LDM    #00000111B,TA4MR      ; 16-bit PWM mode, software trigger
 543                    1                                     ; Count source=f(XIN)/2
 544                    1        .DATA  16
 545  00D095  D8        1        CLM                          ; Data length 16 bits
 546  00D096 *644ED007  L1       LDM    #2000,TA4             ; Set counter value (output "H" pulse width)
 547                    1        .DATA  8
 548  00D09A  F8        1        SEM                          ; Data length 8 bits
 549  00D09B *647900    L1       LDM    #00000000B,TA4IC      ; Disable TA4 interrupt
 550  00D09E *044010    L1       SEB    #00010000B,TABSR      ; Start PWM output
 551                    1 ;
 552                    1 ;      ※ Generate pulse
 553                    1 ;
 554  00D0A1  60        1        RTS
 555                    1 ;
 556                    1 ;
 557                    1 ;-----------------------------------
 558                    1 ;      8-bit PWM                   =
 559                    1 ;-----------------------------------
 560                    1 ;      Timer          :Timer A4
 561                    1 ;      Mode           :PWM mode
 562                    1 ;      Start trigger  :Count start flag
 563                    1 ;                      (Software trigger)
 564                    1 ;      PWM mode       :8-bit PWM mode
 565                    1 ;      Count source   :f(XIN)/2
 566                    1 ;
 567  00D0A2            1 SUB_T5_2:
 568                    1        .DATA  8
 569  00D0A2  F8        1        SEM                          ; Data length 8 bits
 570  00D0A3 *144010    L1       CLB    #00010000B,TABSR      ; Stop TA4 count
 571  00D0A6 *645A27    L1       LDM    #00100111B,TA4MR      ; 8-bit PWM mode, software trigger
 572                    1                                     ; Count source=f(XIN)/2
 573  00D0A9 *644EC7    L1       LDM    #200-1,TA4L           ; Set prescaler value
 574  00D0AC *644F0A    L1       LDM    #10,TA4H              ; Set counter value (output "H" pulse width)
 575  00D0AF *647900    L1       LDM    #00000000B,TA4IC      ; Disable TA4 interrupt
 576  00D0B2 *044010    L1       SEB    #00010000B,TABSR      ; Start PWM output
 577                    1 ;
 578                    1 ;      ※ Generate pulse
 579                    1 ;
 580  00D0B5  60        1        RTS
 581                    1 ;
 582                    1 ;
```

(6) Timer B .....Pulse period measurement mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 583                    1        .PAGE
 584                    1 ;
 585                    1 ;=============================================
 586                    1 ;       Timer setting example 6               =
 587                    1 ;       [Timer B: Pulse period measurement mode] =
 588                    1 ;=============================================
 589                    1 ;       Timer          :Timer B1
 590                    1 ;       Mode           :Pulse period measurement mode
 591                    1 ;                       (between falling edge and falling edge)
 592                    1 ;       Count source   :f(XIN)/64
 593                    1 ;
 594  00D0B6            1 SUB_T6:
 595                    1        .DATA   8
 596  00D0B6  F8        1        SEM                             ; Data length 8 bits
 597  00D0B7 *141040    L1       CLB     #01000000B,P6D          ; Set P66/TB1IN pin to input mode
 598  00D0BA *144040    L1       CLB     #01000000B,TABSR        ; Stop TB1 count
 599  00D0BD *645C82    L1       LDM     #10000010B,TB1MR        ; Pulse period measurement mode
 600                    1                                        ; Count source=f(XIN)/64
 601  00D0C0 *647B00    L1       LDM     #00000000B,TB1IC        ; Clear TB1 interrupt request bit
 602                    1                                        ; Disable TB1 interrupt
 603  00D0C3 *044040    L1       SEB     #01000000B,TABSR        ; Start TB1 count
 604  00D0C6            1 L1_T6:
 605  00D0C6 *347B08FC  L1       BBC     #00001000B,TB1IC,L1_T6  ; TB1 interrupt request bit= "1" ?
 606                    1        .DATA   16
 607  00D0CA  D8        1        CLM                             ; Data length 16 bits
 608  00D0CB *A552      L1       LDA     A,TB1                   ; Read out measurement result in accumulator A
 609                    1        .DATA   8
 610  00D0CD  F8        1        SEM                             ; Data length 8 bits
 611                    1 ;
 612                    1 ;       Note: TB1 overflow period is assumed to be sufficiently longer
 613                    1 ;             than measured pulse period.
 614                    1 ;
 615  00D0CE  60        1        RTS
 616                    1 ;
```

(7) Timer B .....Pulse width measurement mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 617                    1        .PAGE
 618                    1 ;
 619                    1 ;=============================================
 620                    1 ;      Timer setting example 7                =
 621                    1 ;      [Timer B: Pulse width measurement mode] =
 622                    1 ;=============================================
 623                    1 ;      Timer          :Timer B2
 624                    1 ;      Mode           :Pulse with measurement mode
 625                    1 ;      Count source   :f(XIN)/64
 626                    1 ;
 627                    1 ;      ※ This program is a TB2IN input signal "H" level measurement example.
 628                    1 ;         The TB2 overflow period is assumed to be sufficiently longer than
 629                    1 ;         measured pulse width.
 630                    1 ;
 631  00D0CF            1 SUB_T7:
 632                    1        .DATA  8
 633  00D0CF  F8        1        SEM                          ; Data length 8 bits
 634  00D0D0 *141080    L1       CLB    #10000000B,P6D        ; Set P67/TB2IN pin to input mode
 635  00D0D3 *144080    L1       CLB    #10000000B,TABSR      ; Stop TB2 count
 636  00D0D6 *645D8A    L1       LDM    #10001010B,TB2MR      ; Pulse width measurement mode
 637                    1                                     ; Count source=f(XIN)/64
 638  00D0D9 *647C00    L1       LDM    #00000000B,TB2IC      ; Clear TB2 interrupt request bit
 639                    1                                     ; Disable TB2 interrupt
 640  00D0DC *044080    L1       SEB    #10000000B,TABSR      ; Enable TB2 count
 641  00D0DF            1 L1_T7:
 642  00D0DF *240E80FC  L1       BBS    #10000000B,P6,L1_T7   ; TB2IN= "L" ?
 643  00D0E3            1 L2_T7:
 644  00D0E3 *340E80FC  L1       BBC    #10000000B,P6,L2_T7   ; TB2IN= "H" ?
 645  00D0E7 *147C08    L1       CLB    #00001000B,TB2IC      ; Clear interrupt request bit
 646  00D0EA            1 L3_T7:
 647  00D0EA *347C08FC  L1       BBC    #00001000B,TB2IC,L3_T7 ; Interrupt request bit= "1" ?
 648  00D0EE *147C08    L1       CLB    #00001000B,TB2IC      ; Clear interrupt request bit
 649                    1        .DATA  16
 650  00D0F1  D8        1        CLM                          ; Data length 16 bits
 651  00D0F2 *A554      L1       LDA    A,TB2                 ; Read measurement result
 652                    1        .DATA  8
 653  00D0F4  F8        1        SEM                          ; Data length 8 bits
 654                    1 ;
 655  00D0F5  60        1        RTS
 656                    1 ;
```

7.3.4 Serial I/O modes setting examples

(1) 8-bit UART .....1 byte receive

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 673                    1        .PAGE
 674                    1 ;
 675                    1 ;=============================================
 676                    1 ;       Serial I/O setting example 1          =
 677                    1 ;       [8-bit UART (1 byte receive)]         =
 678                    1 ;=============================================
 679                    1 ;       Channel         :UART1
 680                    1 ;       Mode            :8-bit UART
 681                    1 ;       Stop bit length :1 bit
 682                    1 ;       Parity          :None
 683                    1 ;       Clock           :Internal (9600bps)
 684                    1 ;       Sleep function  :Disabled
 685                    1 ;
 686                    1 ;       8-bit UART (1 byte receive) programming model
 687                    1 ;                                M37702
 688                    1 ;                               +-------+
 689                    1 ;                               |       |
 690                    1 ;             DATA ---------->| RxD1  |
 691                    1 ;                               | ____  |
 692                    1 ;                  <----------| RTS1  |
 693                    1 ;                               |       |
 694                    1 ;                               +-------+
 695                    1 ;                                9600bps
 696                    1 ;
 697  00D800            1 SUB_S1:
 698                    1        .DATA   8
 699  00D800  F8        1        SEM                              ; Data length 8 bits
 700  00D801 *141440    L1       CLB     #01000000B,P8D           ; Set P86/RXD1 pin to input mode
 701  00D804 *643805    L1       LDM     #00000101B,S1MR          ; 8-bit UART, internal clock, 1 stop bit
 702                    1                                         ; No parity, disable sleep mode
 703  00D807 *643C04    L1       LDM     #00000100B,S1CL          ; BRG count source=f(XIN)/2
 704                    1                                         ; Select RTS function
 705  00D80A *643919    L1       LDM     #25,S1BRG                ; Set value in BRG (at 9615bps: 8MHz)
 706  00D80D *643D04    L1       LDM     #00000100B,S1CH          ; Enable receive
 707                    1 ;
 708                    1 ;       ※ Receive starts when RTS output becomes "L" level to enable receive
 709                    1 ;          and a start bit is detected.
 710                    1 ;
 711  00D810            1 L1_S1:
 712  00D810 *343D08FC  L1       BBC     #00001000B,S1CH,L1_S1    ; Receive complete (receive completion flag= "1" )?
 713  00D814 *343D8003  L1       BBC     #10000000B,S1CH,L2_S1    ; Check error (error sum flag)
 714  00D818  20BAD8    L1       JSR     SUB_ERR0                 ; Jump to UART receive error processing routine
 715  00D81B            1 L2_S1:
 716  00D81B *A53E      L1       LDA     A,S1RB                   ; Read receive buffer
 717  00D81D *143D04    L1       CLB     #00000100B,S1CH          ; Disable receive
 718  00D820  60        1        RTS
 719                    1 ;
```

(2) 8-bit UART .....1 byte transmit

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 720                      1        .PAGE
 721                      1 ;
 722                      1 ;=============================================
 723                      1 ;       Serial I/O setting example 2          =
 724                      1 ;       [8-bit UART (1 byte transmit)]        =
 725                      1 ;=============================================
 726                      1 ;       Channel         :UART0
 727                      1 ;       Mode            :8-bit UART
 728                      1 ;       Stop bit length :1 bit
 729                      1 ;       Parity          :None
 730                      1 ;       Clock           :Internal (9600bps)
 731                      1 ;       Sleep function  :Disabled
 732                      1 ;
 733                      1 ;       8-bit UART (1 byte transmit) programming model
 734                      1 ;        M37702
 735                      1 ;       +-------+
 736                      1 ;       I       I
 737                      1 ;       I  TxD0 I----------> DATA
 738                      1 ;       I  ____ I
 739                      1 ;       I  CTS0 I<------+
 740                      1 ;       I       I       I
 741                      1 ;       I       I     -+- GND
 742                      1 ;       +-------+
 743                      1 ;        9600bps
 744                      1 ;
 745  00D821              1 SUB_S2:
 746                      1        .DATA   8
 747  00D821  F8          1        SEM                             ; Data length 8 bits
 748  00D822 *141401     L1        CLB     #00000001B,P8D          ; Set P80/CTS0 pin to input mode
 749  00D825 *643005     L1        LDM     #00000101B,S0MR         ; 8-bit UART, internal clock, 1 stop bit
 750                      1                                        ; No parity, disable sleep mode
 751  00D828 *643400     L1        LDM     #00000000B,S0CL         ; BRG count source=f(XIN)/2
 752                      1                                        ; Select CTS function
 753  00D82B *643119     L1        LDM     #25,S0BRG               ; Set value in BRG (at 9615bps; 8MHz)
 754  00D82E *643501     L1        LDM     #00000001B,S0CH         ; Enable transmit
 755  00D831 *A582        1        LDA     A,T_DATA                ; Read transmit data (T_DATA)
 756  00D833 *8532       L1        STA     A,S0TB                  ; Transmit data -> transmission buffer
 757  00D835 *143501     L1        CLB     #00000001B,S0CH         ; Disable transmit
 758  00D838  60          1        RTS
 759                      1 ;
```

(3) 8-bit UART .....n-byte transmit

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 760                    1        .PAGE
 761                    1 ;
 762                    1 ;=========================================
 763                    1 ;       Serial I/O setting example 3           =
 764                    1 ;       [8-bit UART (n-byte transmit)]         =
 765                    1 ;=========================================
 766                    1 ;       Channel         :UART0
 767                    1 ;       Mode            :8-bit UART
 768                    1 ;       Stop bit length :1 bit
 769                    1 ;       Parity          :None
 770                    1 ;       Clock           :Internal (9600bps)
 771                    1 ;       Sleep function  :Disabled
 772                    1 ;
 773                    1 ;       8-bit UART (n-byte transmit) programming model
 774                    1 ;        M37702
 775                    1 ;       +-------+
 776                    1 ;       I       I
 777                    1 ;       I  TxD0 I-------> DATA
 778                    1 ;       I       I
 779                    1 ;       I  ____ I         ___
 780                    1 ;       I  CTS0 I<------- RTS
 781                    1 ;       I       I
 782                    1 ;       +-------+
 783                    1 ;        9600bps
 784                    1 ;
 785                    1 ;---------------------------------------
 786                    1 ;       Prepare transmit               =
 787                    1 ;---------------------------------------
 788                    1 ;
 789  00D839            1 SUB_S3:
 790                    1        .DATA   8
 791                    1        .INDEX  8
 792  00D839  E230      1        SEP     m,x                    ; Data length, index register length 8 bits
 793  00D83B *141401    L1       CLB     #00000001B,P8D         ; Set P80/CTS0 pin to input mode
 794  00D83E *643005    L1       LDM     #00000101B,S0MR        ; 8-bit UART, internal clock, 1 stop bit
 795                    1                                       ; No parity, disable sleep mode
 796  00D841 *643400    L1       LDM     #00000000B,S0CL        ; BRG count source=f(XIN)/2
 797                    1                                       ; Select RTS function
 798  00D844 *643119    L1       LDM     #25,S0BRG              ; Set value in BRG (at 9615bps; 8MHz)
 799  00D847 *643501    L1       LDM     #00000001B,S0CH        ; Enable transmit
 800                    1 ;
 801                    1 ;---------------------------------------
 802                    1 ;       Operate transmit               =
 803                    1 ;---------------------------------------
 804                    1 ;
 805  00D84A  A200      1        LDX     #0                     ;
 806  00D84C            1 L1_S3:
 807  00D84C *B582      1        LDA     A,T_DATA,X             ; Read transmit data
 808  00D84E *8532      L1       STA     A,S0TB                 ; Transmit data → transmission buffer
 809  00D850            1 L2_S3:
 810  00D850 *343502FC  L1       BBC     #00000010B,S0CH,L2_S3  ; Transmission buffer empty flag= "1" ?
 811  00D854  E8        1        INX
 812  00D855 *E480      1        CPX     DAT_CNT                ; Compare with transfer data count (DAT_CNT)
 813  00D857  D0F3      L1       BNE     L1_S3                  ; Continue data transmit?
 814                    1 ;
 815                    1 ;---------------------------------------
 816                    1 ;       Complete transmit              =
 817                    1 ;---------------------------------------
 818                    1 ;
 819  00D859 *143501    L1       CLB     #00000001B,S0CH        ; Disable transmit
 820                    1 ;
 821  00D85C  60        1        RTS
 822                    1 ;
 823                    1 ;
```

(4) Clock synchronous ....1 byte receive

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 824                    1        .PAGE
 825                    1 ;
 826                    1 ;=========================================
 827                    1 ;       Serial I/O setting example 4           =
 828                    1 ;       [Clock synchronous (1 byte receive)]   =
 829                    1 ;=========================================
 830                    1 ;       Channel         :UART1
 831                    1 ;       Mode            :Clock synchronous
 832                    1 ;       Clock           :External
 833                    1 ;
 834                    1 ;       Clock synchronous (1 byte receive) programming model
 835                    1 ;        M37702
 836                    1 ;       +-------+
 837                    1 ;       I  RxD1 I<-------
 838                    1 ;       I       I
 839                    1 ;       I  CLK1 I<------- CLOCK
 840                    1 ;       I  ____ I         ___
 841                    1 ;       I  RTS1 I-------> CTS
 842                    1 ;       I       I
 843                    1 ;       +-------+
 844                    1 ;
 845  00D85D            1 SUB_S4:
 846                    1        .DATA   8
 847  00D85D  F8        1        SEM                             ; Data length 8 bits
 848  00D85E *141460    L1       CLB     #01100000B,P8D          ; Set P86/RXD1, P85/CLK1 input to input mode
 849  00D861 *643809    L1       LDM     #00001001B,S1MR         ; Select clock synchronous, external clock
 850  00D864 *643C04    L1       LDM     #00000100B,S1CL         ; Select RTS function
 851  00D867 *643A55    L1       LDM     #55H,S1TB               ; Set transmit dummy set
 852                    1 ;
 853                    1 ;       ※ With clock synchronous serial I/O, the transmitter must be operating
 854                    1 ;          even if only receive is performed.
 855                    1 ;
 856  00D86A *643D05    L1       LDM     #00000101B,S1CH         ; Enable receive and transmit
 857                    1 ;
 858                    1 ;       ※ Hereafter, receive operation starts when synchronous clock
 859                    1 ;          is input to CLK1 pin.
 860                    1 ;
 861  00D86D            1 L1_S4:
 862  00D86D *343D08FC  L1       BBC     #00001000B,S1CH,L1_S4   ; Receive completion flag= "1" ?
 863  00D871 *343D1003  L1       BBC     #00010000B,S1CH,L2_S4   ; Overrun error?
 864  00D875  20C1D8    L1       JSR     SUB_ERR1                ; Jump to clock synchronous error processing routine
 865  00D878            1 L2_S4:
 866  00D878 *A53E      L1       LDA     A,S1RB                  ; Read received result
 867  00D87A *143D05    L1       CLB     #00000101B,S1CH         ; Disable receive and transmit
 868                    1 ;
 869  00D87D  60        1        RTS
 870                    1 ;
 871                    1 ;
```

(5) Clock synchronous ....1 byte transmit

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 872                    1        .PAGE
 873                    1 ;
 874                    1 ;=============================================
 875                    1 ;       Serial I/O setting example 5          =
 876                    1 ;       [Clock synchronous (1 byte transmit)] =
 877                    1 ;=============================================
 878                    1 ;       Channel         :UART0
 879                    1 ;       Mode            :Clock synchronous
 880                    1 ;       Clock           :Internal (2Mbps)
 881                    1 ;
 882                    1 ;       Clock synchronous (1 byte transmit) programming model
 883                    1 ;        M37702
 884                    1 ;       +-------+
 885                    1 ;       |       |
 886                    1 ;       |  TxD0 |-------> DATA
 887                    1 ;       |       |
 888                    1 ;       |  CLK0 |-------> CLOCK(2Mbps)
 889                    1 ;       |  ____ |
 890                    1 ;       |  CTS0 |<------+
 891                    1 ;       |       |       |
 892                    1 ;       |       |      -+- GND
 893                    1 ;       +-------+
 894                    1 ;
 895  00D87E            1 SUB_S5:
 896                    1        .DATA   8
 897  00D87E  F8        1        SEM                             ; Data length 8 bits
 898  00D87F *141401    L1       CLB     #00000001B,P8D          ; Set P80/CTS0 pin to input port
 899  00D882 *643001    L1       LDM     #00000001B,S0MR         ; Clock synchronous, internal clock
 900  00D885 *643400    L1       LDM     #00000000B,S0CL         ; BRG count source=f(XIN)/2
 901                    1                                        ; Select CTS function
 902  00D888 *643100    L1       LDM     #0,S0BRG                ; Set value in BRG (at 2Mbps; 8MHz)
 903  00D88B *643501    L1       LDM     #00000001B,S0CH         ; Enable transmit
 904  00D88E *A582       1       LDA     A,T_DATA                ; Read transmit data (T_DATA)
 905  00D890 *8532      L1       STA     A,S0TB                  ; Transmit data → transmission buffer
 906  00D892 *143501    L1       CLB     #00000001B,S0CH         ; Disable transmit
 907                    1 ;
 908  00D895  60        1        RTS
 909                    1 ;
```

(6) Clock synchronous ....n-byte transmit

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 910                      1        .PAGE
 911                      1 ;
 912                      1 ;=========================================
 913                      1 ;      Serial I/O setting example 6          =
 914                      1 ;      [Clock synchronous (n-byte transmit)]  =
 915                      1 ;=========================================
 916                      1 ;      Channel        :UART0
 917                      1 ;      Mode           :Clock synchronous
 918                      1 ;      Clock          :Internal (2Mbps)
 919                      1 ;
 920                      1 ;      Clock synchronous (n-byte transmit) programming model
 921                      1 ;       M37702
 922                      1 ;      +-------+
 923                      1 ;      |  TxD0 |-------> DATA
 924                      1 ;      |       |
 925                      1 ;      |  CLK0 |-------> CLOCK(2Mbps)
 926                      1 ;      |  ____ |         ___
 927                      1 ;      |  CTS0 |<------- RTS
 928                      1 ;      |       |
 929                      1 ;      +-------+
 930                      1 ;
 931                      1 ;---------------------------------------
 932                      1 ;      Prepare transmit              =
 933                      1 ;---------------------------------------
 934                      1 ;
 935  00D896              1 SUB_S6:
 936                      1        .DATA  8
 937                      1        .INDEX 8
 938  00D896  E230        1        SEP    x,m                  ; Data length, index register length 8 bits
 939  00D898 *141401      L1       CLB    #00000001B,P8D       ; Set P80/CTS0 pin to input mode
 940  00D89B *643001      L1       LDM    #00000001B,S0MR      ; Clock synchronous, internal clock
 941  00D89E *643400      L1       LDM    #00000000B,S0CL      ; BRG count source=f(XIN)/2
 942                      1                                    ; Select CTS function
 943  00D8A1 *643100      L1       LDM    #0,S0BRG             ; Set value in BRG (at 2Mbps; 8MHz)
 944  00D8A4 *643501      L1       LDM    #00000001B,S0CH      ; Enable transmit
 945                      1 ;
 946                      1 ;---------------------------------------
 947                      1 ;      Transmit data                 =
 948                      1 ;---------------------------------------
 949                      1 ;
 950  00D8A7  A200        1        LDX    #0
 951  00D8A9              1 L1_S6:
 952  00D8A9 *B582        1        LDA    A,T_DATA,X           ; Read transmit data
 953  00D8AB *8532        L1       STA    A,S0TB               ; Transmit data → transmission buffer
 954  00D8AD              1 L2_S6:
 955  00D8AD *343502FC    L1       BBC    #00000010B,S0CH,L2_S6 ; Transmission buffer empty?
 956  00D8B1  E8          1        INX
 957  00D8B2 *E480        1        CPX    DAT_CNT              ; Compare with trasfer data count (DAT_CNT)
 958  00D8B4  D0F3        L1       BNE    L1_S6                ; Continue data transmit?
 959                      1 ;
 960                      1 ;---------------------------------------
 961                      1 ;      Complete transmit             =
 962                      1 ;---------------------------------------
 963                      1 ;
 964  00D8B6 *143501      L1       CLB    #00000001B,S0CH      ; Disable transmit
 965                      1 ;
 966  00D8B9  60          1        RTS
 967                      1 ;
```

(7) Error processing

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

 968                      1        .PAGE
 969                      1 ;
 970                      1 ;=============================================
 971                      1 ;       Serial I/O setting example 7          =
 972                      1 ;       [Error processing]                    =
 973                      1 ;=============================================
 974                      1 ;
 975                      1 ;---------------------------------------
 976                      1 ;       UART receive error processing   =
 977                      1 ;---------------------------------------
 978                      1 ;
 979  00D8BA              1 SUB_ERR0:
 980                      1 ;               :
 981                      1 ;               :       Check error type (overrun, framing, parity)
 982                      1 ;               :
 983                      1 ;               :       Each error processing
 984                      1 ;               :
 985                      1 ;
 986  00D8BA *143D04     L1         CLB     #00000100B,S1CH         ; Disable receive
 987                      1                                         ; (Clear error flag)
 988  00D8BD *043D04     L1         SEB     #00000100B,S1CH         ; Enable receive
 989  00D8C0  60          1         RTS
 990                      1 ;
 991                      1 ;-------------------------------------------------
 992                      1 ;       Clock synchronous receive error processing =
 993                      1 ;-------------------------------------------------
 994                      1 ;
 995  00D8C1              1 SUB_ERR1:
 996                      1 ;               :
 997                      1 ;               :       Overrun error processing
 998                      1 ;               :
 999                      1 ;               :
1000                      1 ;
1001  00D8C1 *143D04     L1         CLB     #00000100B,S1CH         ; Disable receive
1002                      1                                         ; (Clear error flag)
1003  00D8C4 *043D04     L1         SEB     #00000100B,S1CH         ; Enable receive
1004                      1 ;
1005  00D8C7  60          1         RTS
1006                      1 ;
```

7.3.5 A-D conversion modes setting examples

(1) One-shot mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1018                      1         .PAGE
1019                      1 ;
1020                      1 ;=============================================
1021                      1 ;       A-D conversion setiing example 1        =
1022                      1 ;       [A-D conversion (one-shot mode)]        =
1023                      1 ;=============================================
1024                      1 ;       A-D conversion mode     :One-shot mode
1025                      1 ;       Analog input pin        :AN0
1026                      1 ;       Start trigger           :Software trigger
1027                      1 ;       A-D conversion frequency:f(XIN)/4
1028                      1 ;
1029  00E000              1 SUB_AD1:
1030                      1         .DATA   8
1031  00E000  F8          1         SEM                             ; Data length 8 bits
1032  00E001 *141101      L1        CLB     #00000001B,P7D          ; Set P70/AN0 pin to input mode
1033  00E004 *641E80      L1        LDM     #10000000B,ADCON        ; One-shot mode, software trigger,
1034                                                                  A-D conversion frequency =f(XIN)/4
1035  00E007 *647000      L1        LDM     #00000000B,ADIC         ; Clear A-D interrupt request bit, disable A-D interrupt
1036  00E00A *041E40      L1        SEB     #01000000B,ADCON        ; Start A-D conversion
1037  00E00D               1 L_AD1:
1038  00E00D *347008FC    L1        BBC     #00001000B,ADIC,L_AD1   ; A-D conversion complete (interrupt request bit= "1" )?
1039  00E011 *147008      L1        CLB     #00001000B,ADIC         ; Clear A-D interrupt request bit
1040  00E014 *A520        L1        LDA     A,AD0                   ; Read conversion result (A-D register 0)
1041  00E016  60          1         RTS
1042                      1 ;
```

(2) Repeat mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1043                      1         .PAGE
1044                      1 ;
1045                      1 ;=========================================
1046                      1 ;       A-D conversion setting example 2        =
1047                      1 ;       [A-D conversion (repeat mode)]          =
1048                      1 ;=========================================
1049                      1 ;       A-D conversion mode      :Repeat mode
1050                      1 ;       Analog input pin         :AN1
1051                      1 ;       Start trigger            :Software trigger
1052                      1 ;       A-D conversion frequency :f(XIN)/4
1053                      1 ;
1054  00E017              1 SUB_AD2:
1055                      1         .DATA   8
1056  00E017  F8          1         SEM                              ; Data length 8 bits
1057  00E018 *141102     L1         CLB     #00000010B,P7D           ; Set P71/AN1 pin to input mode
1058  00E01B *641E89     L1         LDM     #10001001B,ADCON         ; Repeat mode, software trigger,
1059                                                                    A-D conversion frequency =f(XIN)/4
1060  00E01E *041E40     L1         SEB     #01000000B,ADCON         ; Start A-D conversion
1061                      1 ;
1062                      1 ;       ※ After time equal to A-D conversion interval (28.5 μs; at 8MHz), the latest
1063                      1 ;          conversion result can be obtained by reading A-D register 1 at any timing.
1064                      1 ;
1065  00E021  60          1         RTS
1066                      1 ;
```

(3) Single sweep mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1067                      1         .PAGE
1068                      1 ;
1069                      1 ;=========================================
1070                      1 ;       A-D conversion setting example 3        =
1071                      1 ;       [A-D conversion (single sweep mode)]    =
1072                      1 ;=========================================
1073                      1 ;       A-D conversion mode      :Single sweep mode
1074                      1 ;       Analog input pin         :AN0-AN5
1075                      1 ;       Start trigger            :External trigger
1076                      1 ;       A-D conversion frequency :f(XIN)/4
1077                      1 ;
1078  00E022              1 SUB_AD3:
1079                      1         .DATA   8
1080  00E022  F8          1         SEM                              ; Data length 8 bits
1081  00E023 *1411BF     L1         CLB     #10111111B,P7D           ; Set AN0-AN5, ADTRG pins to input mode
1082  00E026 *641F02     L1         LDM     #00000010B,ADSPS         ; Select AN0-AN5 pin for sweep
1083  00E029 *641EB0     L1         LDM     #10110000B,ADCON         ; Single sweep mode, external trigger,
1084                                                                   A-D conversion frequency =f(XIN)/4
1085  00E02C *647000     L1         LDM     #00000000B,ADIC          ; Clear A-D conversion request bit, disable A-D interrupt
1086  00E02F *041E40     L1         SEB     #01000000B,ADCON         ; Enable A-D conversion
1087                      1 ;
1088                      1 ;       ※ A-D conversion starts at falling edge input to ADTRG pin.
1089                      1 ;
1090  00E032              1 L_AD3:
1091  00E032 *347008FC   L1         BBC     #00001000B,ADIC,L_AD3    ; Sweep complete (interrupt request bit= "1" )?
1092  00E036 *147008     L1         CLB     #00001000B,ADIC          ; Clear A-D interrupt request bit
1093                      1 ;
1094                      1 ;       ※ Conversion result can be obtained by reading A-D registers 0-5.
1095                      1 ;          Then, A-D conversion is resumed when falling edge is input to
1096                      1 ;          ADTRG pin.
1097                      1 ;          Clear A-D conversion start flag to "1" if reconversion is no necessary.
1098                      1 ;
1099  00E039  60          1         RTS
1100                      1 ;
```

(4) Repeat sweep mode

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1101                    1         .PAGE
1102                    1 ;
1103                    1 ;=========================================
1104                    1 ;       A-D conversion setting example 4        =
1105                    1 ;       [A-D conversion (repeat sweep mode)]    =
1106                    1 ;=========================================
1107                    1 ;       A-D conversion mode      :Repeat sweep mode
1108                    1 ;       Analog input pin         :AN0-AN7
1109                    1 ;       Start trigger            :Software trigger
1110                    1 ;       A-D conversion frequency :f(XIN)/4
1111                    1 ;
1112  00E03A            1 SUB_AD4:
1113                    1         .DATA   8
1114  00E03A  F8        1         SEM                             ; Data length 8 bits
1115  00E03B *641100    L1        LDM     #00H,P7D                ; Set AN0-AN7 pins to input mode
1116  00E03E *641F03    L1        LDM     #00000011B,ADSPS        ; Select AN0-AN7 pins for sweep
1117  00E041 *641E98    L1        LDM     #10011000B,ADCON        ; Repeat sweep mode, internal trigger,
                                                                    A-D conversion frequency =f(XIN)/4
1118  00E044 *041E40    L1        SEB     #01000000B,ADCON        ; Start A-D conversion
1119                    1 ;
1120                    1 ;       ※ After the first A-D sweep (228 μs; at 8MHz), the
1121                    1 ;          latest conversion result can be obtained by reading
1122                    1 ;          A-D registers 0-7 at any timing.
1123                    1 ;
1124  00E047  60        1         RTS
1125                    1 ;
```

7.3.6 Interrupt processing examples

(1) Interrupt setting example

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1133                    1         .PAGE
1134                    1 ;=============================================
1135                    1 ;       Interrupt setting example             =
1136                    1 ;=============================================
1137                    1 ;
1138                    1 ;       In order to execute an interrupt, the I flag
1139                    1 ;       must be cleared within the main routine and
1140                    1 ;       interrupt priority level must be set to level 1
1141                    1 ;       or greater in each interrupt control register.
1142                    1 ;
1143  00E800  78        1         SEI                           ; The I flag is initialized to "1" after reset.
1144                    1 ;
1145                    1 ;
1146                    1         .DATA   8
1147  00E801  F8        1         SEM
1148  00E802 *141008    L1        CLB     #00001000B,P6D        ; Set P63/INT1 pin to input mode
1149  00E805 *647E02    L1        LDM     #00000010B,INT1IC     ; Set INT1 interrupt priority level to level 2
1150  00E808  58        1         CLI                           ; Enable interrupt
1151                    1 ;
1152                    1 ;       ※ Hereafter, INT1 interrupt request occurs with input signal to INT1 pin.
1153                    1 ;
```

(2) Interrupt routine processing example.....When memory space is 64K bytes or less.

```
SEQ.  LOC.   OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1154                  1        .PAGE
1155                  1 ;
1156                  1 ;=============================================
1157                  1 ;       Interrupt routine processing example 1  =
1158                  1 ;       When memory space is 64K bytes or less  =
1159                  1 ;       (When data bank need not be changed)    =
1160                  1 ;=============================================
1161                  1 ;
1162  00E809          1 INTERRUPT_1:
1163                  1 ;---------------------------------------
1164                  1 ;       Store registers                 =
1165                  1 ;---------------------------------------
1166                  1 ;
1167                  1        .DATA          16              ; Register model declaration
1168                  1        .INDEX         16
1169  00E809  C230    1        CLP            m,x             ; Data length, index register length 16 bits
1170  00E80B  EB0D    1        PSH            X,Y,A           ; Store registers
1171                  1 ;
1172                  1        .DATA          8               ; Change register model
1173                  1        .INDEX         8
1174  00E80D  E230    1        SEP            m,x             ; Data length, index register length 8 bits
1175                  1 ;
1176                  1 ;              :
1177                  1 ;       Interrupt processing
1178                  1 ;              :
1179                  1 ;
1180                  1 ;---------------------------------------
1181                  1 ;       Restore registers               =
1182                  1 ;---------------------------------------
1183                  1 ;
1184                  1        .DATA          16              ; Change register mode
1185                  1        .INDEX         16
1186  00E80F  C230    1        CLP            m,x             ; Data length, index register length 16 bits
1187  00E811  FB0D    1        PUL            X,Y,A           ; Restore registers X, Y, A
1188                  1 ;
1189                  1 ;-----------------------------------------------
1190                  1 ;       Return from interrupt processing routine =
1191                  1 ;-----------------------------------------------
1192                  1 ;
1193  00E813  40      1        RTI                            ; Return to processing before interrupt
1194                  1 ;
1195                  1 ;       ※ The PS, PG, and PC are restored automatically to their values
1196                               before the interrupt with the RTI instruction.
1197                  1 ;
```

(3) Interrupt routine processing example.....When memory space exceeds 64K bytes.

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1198                    1               .PAGE
1199                    1 ;
1200                    1 ;=============================================
1201                    1 ;       Interrupt routine processing example 2  =
1202                    1 ;       When memory space exceeds 64K bytes     =
1203                    1 ;       (When data bank is changed)             =
1204                    1 ;=============================================
1205                    1 ;
1206  00E814            1 INTERRUPT_2:                                    ;
1207                    1 ;---------------------------------------
1208                    1 ;       Store registers                 =
1209                    1 ;---------------------------------------
1210                    1 ;
1211                    1         .DATA           16              ; Register model declaration
1212                    1         .INDEX          16
1213  00E814  C230      1         CLP             m,x             ; Data length, index register length 16 bits
1214  00E816  EB0D      1         PSH             X,Y,A           ; Store registers
1215  00E818  8B        1         PHT                             ; Store data bank register
1216                    1         .DATA           8               ; Change register model
1217                    1         .INDEX          8
1218  00E819  E230      1         SEP             m,x             ; Data length, index register length 8 bits
1219                    1         .DT             012H            ; Declare data banks used for interrupt processing
1220  00E81B  89C212    1         LDT             #012H           ; Set data bank registers used for interrupt processing
1221                    1 ;
1222                    1 ;               :
1223                    1 ;               :       Interrupt processing
1224                    1 ;               :
1225                    1 ;
1226                    1 ;---------------------------------------
1227                    1 ;       Restore registers               =
1228                    1 ;---------------------------------------
1229                    1 ;
1230  00E81E  AB        1         PLT                             ; Restore data bank registers
1231                    1         .DATA           16              ; Change register model
1232                    1         .INDEX          16
1233  00E81F  C230      1         CLP             m,x             ; Data length, index register length 16 bits
1234  00E821  FB0D      1         PUL             X,Y,A           ; Restore registers X, Y, A
1235                    1 ;
1236                    1 ;-----------------------------------------------
1237                    1 ;       Return from interrupt processing routine =
1238                    1 ;-----------------------------------------------
1239                    1 ;
1240  00E823  40        1         RTI                             ; Return to processing before interrupt
1241                    1 ;
1242                    1 ;       ※ The PS, PG, and PC are restored automatically to their values
1243                                 before the interrupt with the RTI instruction
1244                    1 ;
```

7.3.7 Watchdog timer setting examples

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1254                      1         .PAGE
1255                      1 ;
1256                      1 ;===========================================
1257                      1 ;       Watchdog timer setting example 1        =
1258                      1 ;       [Watchdog timer write routine]          =
1259                      1 ;===========================================
1260                      1 ;   "FFFH" is set in watchdog timer by writing
1261                      1 ;   to watchdog timer (60H)
1262                      1 ;
1263  00F000              1 WDT_SET:
1264  00F000  08          1         PHP                           ; Store PS
1265  00F001  F8          1         SEM                           ; Data length 8 bits (note)
1266  00F002 *8560       L1         STA     A,WDT                 ; Write to watchdog timer
1267  00F004  28          1         PLP                           ; Restore PS
1268  00F005  60          1         RTS
1269                      1 ;
1270                      1 ; Note: The address following the watchdog timer
1271                      1 ;       contains the watchdog timer frequency
1272                      1 ;       selection flag. Therefore, be careful not
1273                      1 ;       to change this value when accessing the
1274                      1 ;       watchdog timer in 16-bit unit.
1275                      1 ;
1276                      1 ;
1277                      1 ;=================================================
1278                      1 ;       Watchdog timer setting example 2            =
1279                      1 ;       [Watchdog timer interrupt processing example] =
1280                      1 ;=================================================
1281                      1 ;  In this example, a software reset is performed
1282                      1 ;  when a watchdog timer interrupt is generated.
1283                      1 ;
1284  00F006              1 WATCH_DOG:
1285                      1         .DATA   8
1286  00F006  F8          1         SEM                           ;
1287  00F007 *045E08     L1         SEB     #00001000B,PMR        ; Software reset
1288  00F00A  40          1         RTI
1289                      1 ;
```

7.3.8 Software timer setting examples

```
SEQ.  LOC.    OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1296                      1         .PAGE
1297                      1 ;
1298                      1 ;================================================
1299                      1 ;       Software timer examples                  =
1300                      1 ;================================================
1301                      1 ; Note: The wait time in the following subroutine depends
1302                      1 ;       on the operating frequency and external bus
1303                      1 ;       width of the microcomputer.
1304                      1 ;       The following routine assumes single-chip mode
1305                      1 ;       with external input clock frequency at 8MHz.
1306                      1 ;
1307                      1 ;============================================
1308                      1 ;       10 μs wait routine                 =
1309                      1 ;============================================
1310                      1 ;       1 φcyc=250ns at f(XIN)=8MHz
1311                      1 ;
1312  00F800              1 WIT10:
1313                      1         .DATA   8
1314  00F800  F8          1         SEM                          ;         2 φcyc
1315  00F801  894920      1         RLA     #32                  ;      6+32 φcyc
1316                      1                                      ; ----------------
1317                      1                                      ; Total  40 φcyc
1318                      1 ;
1319                      1 ;============================================
1320                      1 ;       50 μs wait routine                 =
1321                      1 ;============================================
1322                      1 ;       1 φcyc=250ns at f(XIN)=8MHz
1323                      1 ;
1324  00F804              1 WIT50:
1325                      1         .DATA   8
1326  00F804  F8          1         SEM                          ;         2 φcyc
1327  00F805  8949C0      1         RLA     #192                 ;      6+192 φcyc
1328                      1                                      ; --------------
1329                      1                                      ; Total  200 φcyc
1330                      1 ;
1331                      1 ;
1332
```

7.3.9 Interrupt vector table setting example

```
SEQ.  LOC.   OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1358                              .PAGE   'INTERRUPT VECTOR TABLE'
1359                        ;
1360                        ;=================================================
1361                        ;       Interrupt vector table                   =
1362                        ;=================================================
1363                        ;
1364                              .SECTION        VECTOR_AREA
1365                              .ORG    OFFD6H
1366  00FFD6 0000       L         .WORD   INT_AD              ; A-D conversion interrupt vector
1367  00FFD8 0000       L         .WORD   INT_S1T             ; UART1 transmission interrupt vector
1368  00FFDA 0000       L         .WORD   INT_S1R             ; UART1 receive interrupt vector
1369  00FFDC 0000       L         .WORD   INT_SOT             ; UART0 transmission interrupt vector
1370  00FFDE 0000       L         .WORD   INT_SOR             ; UART0 receive interrupt vector
1371  00FFE0 0000       L         .WORD   INT_TB2             ; Timer B2 interrupt vector
1372  00FFE2 0000       L         .WORD   INT_TB1             ; Timer B1 interrupt vector
1373  00FFE4 0000       L         .WORD   INT_TB0             ; Timer B0 interrupt vector
1374  00FFE6 0000       L         .WORD   INT_TA4             ; Timer A4 interrupt vector
1375  00FFE8 0000       L         .WORD   INT_TA3             ; Timer A3 interrupt vector
1376  00FFEA 0000       L         .WORD   INT_TA2             ; Timer A2 interrupt vector
1377  00FFEC 0000       L         .WORD   INT_TA1             ; Timer A1 interrupt vector
1378  00FFEE 0000       L         .WORD   INT_TA0             ; Timer A0 interrupt vector
1379  00FFF0 0000       L         .WORD   INT2                ; INT2 interrupt vector
1380  00FFF2 0000       L         .WORD   INT1                ; INT1 interrupt vector
1381  00FFF4 0000       L         .WORD   INT0                ; INT0 interrupt vector
1382  00FFF6 0000       L         .WORD   INT_WDT             ; Watchdog timer interrupt vector
1383  00FFF8 0000       L         .WORD   RESERVED            ; (Reserved area)
1384  00FFFA 0000       L         .WORD   INT_BRK             ; BRK instruction interrupt vector
1385  00FFFC 0000       L         .WORD   INT_DIV0            ; Zero divide interrupt vector
1386  00FFFE C000       L         .WORD   INITIAL             ; Reset vector
1387
```

7.4 M37702 group execution performance

The execution performance of the M37702 group is described below.

7.4.1 Comparing the execution speed of M37702 and M37700

One of the differences between the M37702 group and the M37700 group is the difference in the external area access operation while a software one-wait caused by the wait bit is valid (see Figure 7.4.1). The difference in program execution speed due to this difference in access operation is described below.

Figure 7.4.3 compares the execution time of the M37702 group and M37700 group when executing two sample programs shown in Figure 7.4.2 under the conditions shown in Table 7.4.1 and the same clock frequency. Figure 7.4.3 shows that the M37702 group is much faster when accessing an external area while software one-wait is valid.

This result depends on the frequency of the access and the difference increases as the access frequency increases. The effect for microprocessor mode using an 8-bit external bus appears in Figure 7.4.3.

Table 7.4.1 M37702 group and M37700 group execution speed comparison conditions

Parameter	Conditions
Operating mode	Microprocessor mode
External bus width	16 bits or 8 bits
Software one-wait	Valid
Program area	External EPROM
Work area	External or internal SRAM

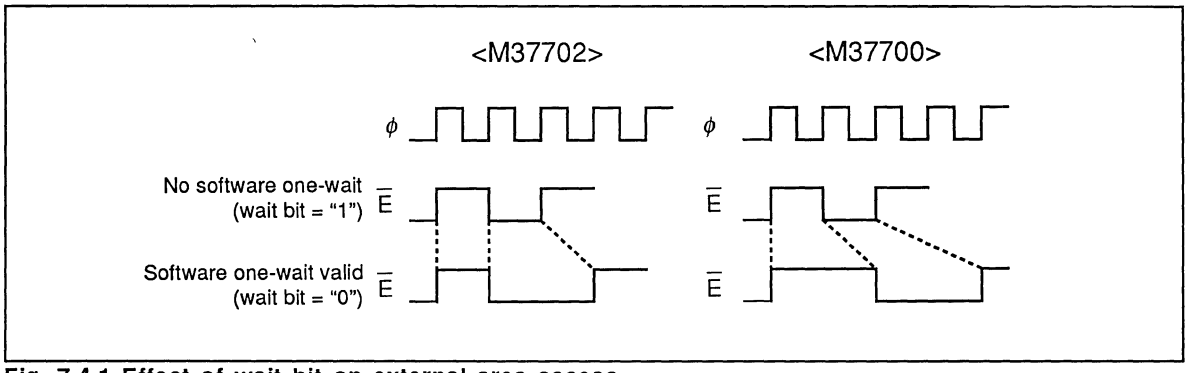


Fig. 7.4.1 Effect of wait bit on external area access

Sample program 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 program B

```
        SEP     X
        CLM
        . DATA  16
        . INDEX 8
        LDY     #69
L00P0:  LDX     #69
L00P1:  ASL     SOUR, X
        SEM
        . DATA  8
        ROL     SOUR+2, X
        ROL     B
        CLM
        . DATA  16
        ROR     A
        DEX
        DEX
        DEX
        BNE     L00P1
        STA     A, DEST, Y
        SEM
        . DATA  8
        STA     B, DEST+2, Y
        CLM
        . DATA  16
        DEY
        DEY
        DEY
        BNE     L00P0
```

* SOUR, DEST : Work area

Fig. 7.4.2 Sample program list

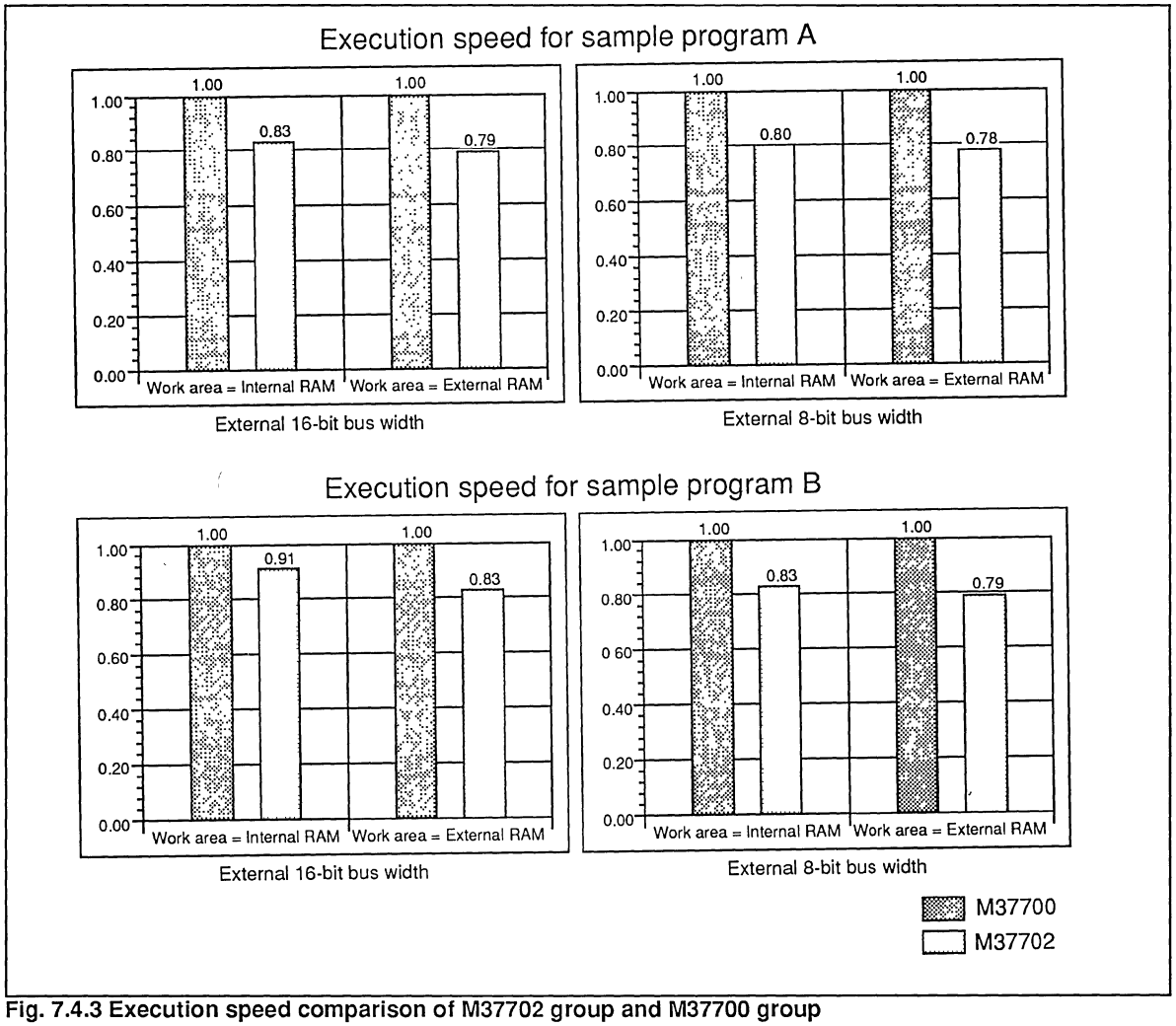


Fig. 7.4.3 Execution speed comparison of M37702 group and M37700 group

7.4.2 Software one-wait (20MHz) and software+hardware wait (25MHz) execution speed comparison

The program execution time is for M37702 group operating at 20MHz with software one-wait and M37702 group operating at 25MHz with software one-wait and $\overline{RDY}$ (hardware wait).

Figure 7.4.4 shows the execution time when two sample programs (see Figure 7.4.2) are executed under the conditions shown in table 7.4.2.

Table 7.4.2 Software one-wait (20MHz) and software+hardware wait (25MHz) execution speed comparison

Parameter	Conditions of software one-wait side	Conditions of software + hardware side
Operating mode	Microprocessor mode	←
Clock frequency	20MHz	25MHz
External bus width	16 bits	←
Software one-wait	Valid	←
$\overline{RDY}$ (hardware wait)	Invalid	Valid only in external EPROM area
Program area	External EPROM	←
Work area	External or internal SRAM	←

In both cases, the M37702 group is used in microprocessor mode with 16-bit external bus and program stored in external EPROM. One wait is inserted for external memory access because software one-wait is used. In addition, $\overline{RDY}$ is used for external EPROM access during 25MHz operation for a total of two waits ($\overline{RDY}$ is invalidated for access of external RAM used for work area).

Figure 7.4.5 shows the memory map during execution speed comparison.

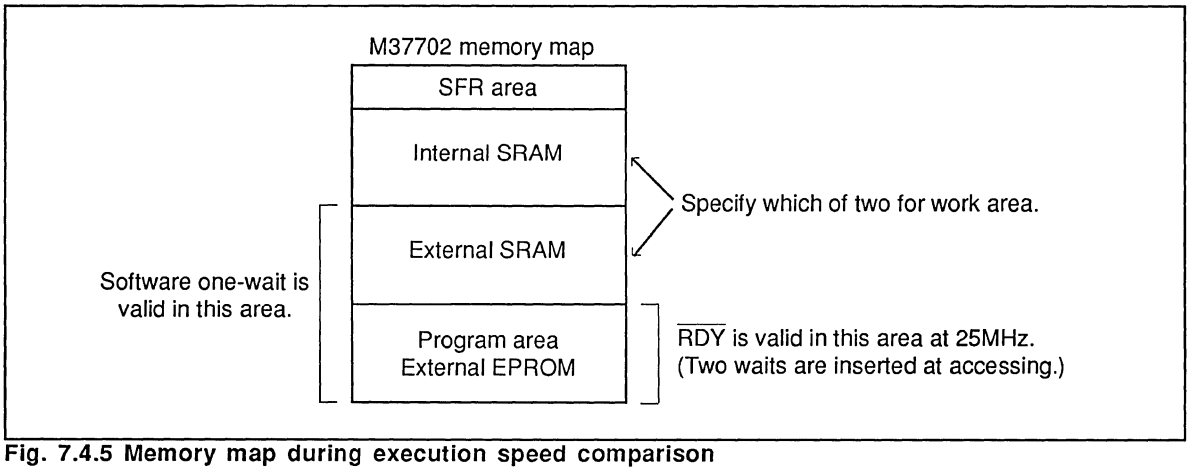


Fig. 7.4.5 Memory map during execution speed comparison

Figure 7.4.4 shows that the difference in execution speed is small between 20MHz operation with software one-wait and 25MHz operation with software one-wait and $\overline{RDY}$. 20MHz operation with software one-wait provides better cost performance because the use of specified memory eliminates the need for a bus buffer.

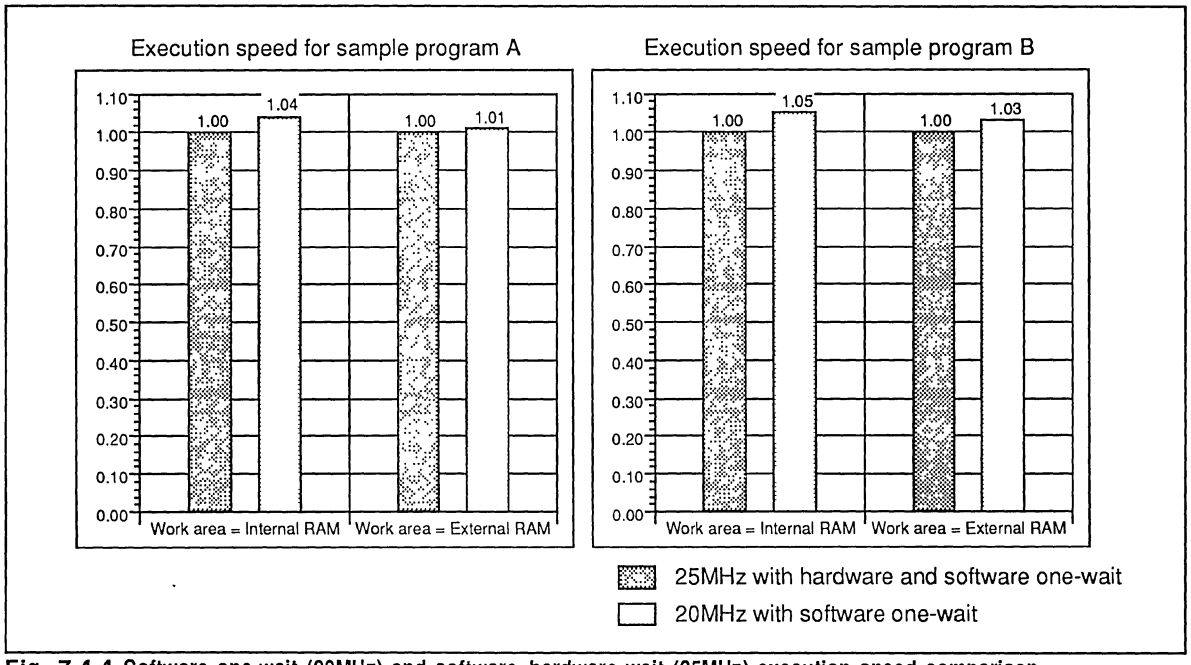


Fig. 7.4.4 Software one-wait (20MHz) and software+hardware wait (25MHz) execution speed comparison

MEMO

CHAPTER 8
PROM VERSION

8.1 Product expansion

Internal PROM (programmable ROM) version has the following 2 types :

- ●One time PROM version ..... Possible to write program in ROM once.
- ●EPROM version ..... Possible to rewrite program in ROM because a written program is erased by exposing the erase window on top of the package to an ultraviolet light source.

Table 8.1.1 shows the product expansion of internal PROM version

Table 8.1.1 Product expansion of internal PROM version

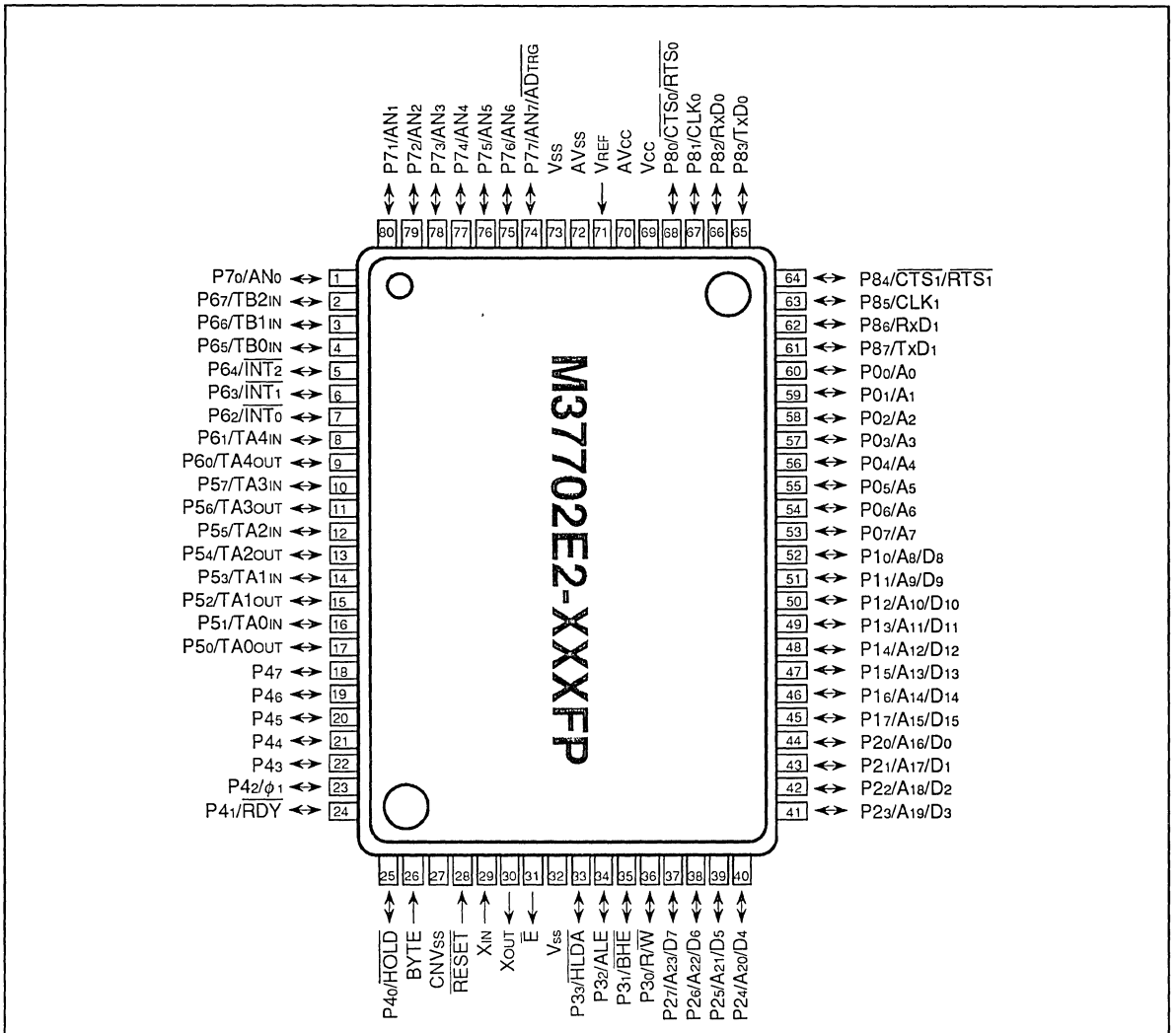
Type name	ROM	RAM	Clock frequency	Writing adapter
M37702E2-XXXFP	One time PROM 16K bytes	512 bytes	8MHz	PCA4774
M37702E2AXXXFP			16MHz	
M37702E2BXXXFP			25MHz	
M37702E2FS	EPROM 16K bytes		8MHz	PCA4708
M37702E2AFS			16MHz	
M37702E2BFS			25MHz	
M37702E4-XXXFP	One time PROM 32K bytes	2048 bytes	8MHz	PCA4774
M37702E4AXXXFP			16MHz	
M37702E4BXXXFP			25MHz	
M37702E4FS	EPROM 32K bytes		8MHz	PCA4708
M37702E4AFS			16MHz	
M37702E4BFS			25MHz	

8.2 M37702E2-XXXFP

The following descriptions will be for M37702E2-XXXFP. Internal PROM version has the same functions as M37702E2-XXXFP unless otherwise noted.

8.2.1 Description

The M37702E2-XXXFP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology, and has the same functions as M37702M2-XXXFP except that PROM is built in.
Since general purpose PROM writers can be used for the built-in PROM, this microcomputer is suitable for small quantity production runs.
Figure 8.2.1 shows the M37702E2-XXXFP pin configuration.



Outline 80P6N

Fig. 8.2.1 M37702E2-XXXFP pin configuration

8.2.2 Functional description

The pin arrangement of M37702E2-XXXFP is identical to the mask ROM version M37702M2-XXXFP. Internal PROM version has a normal operating mode which provides the same functions as the mask ROM version and an EPROM mode used to write to built-in PROM.

In normal operating mode, the pin functions are equivalent to the corresponding mask ROM version. In EPROM mode, the pin functions are shown in Table 8.2.1.

Table 8.2.1 Pin functions in EPROM mode

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5V±10% to V_{CC}, and 0V to V_{SS}.
CNVss	V_{PP} input	Input	Connect to V_{PP} when programming or verifying.
BYTE	V_{PP} input	Input	Connect to V_{PP} when programming or verifying.
$\overline{RESET}$	Reset input	Input	Connect to V_{SS}.
X_{IN}	Clock input	Input	Connect a ceramic resonator between X_{IN} pin and X_{OUT} pin. When an external clock is used, the clock source should be connected to the X_{IN} pin and X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Open.
AVcc, AVss	Analog power supply input		Externally connect AVcc to Vcc and AVss to Vss.
V_{REF}	Reference voltage input	Input	Connect to V_{SS}.
$P0_0$–$P0_7$	Address input (A_0–A_7)	Input	The low-order 8-bit (A_0–A_7) address input pins.
$P1_0$–$P1_7$	Address input (A_8–A_{14})	Input	$P1_0$–$P1_6$ are high-order 7-bit address input pins. Connect $P1_7$ to V_{CC}.
$P2_0$–$P2_7$	Data input/output	I/O	8-bit data (D_0–D_7) input/output pins.
$P3_0$–$P3_3$	Input port P3	Input	Connect to V_{SS}.
$P4_0$–$P4_7$	Input port P4	Input	Connect to V_{SS}.
$P5_0$–$P5_7$	Control input	Input	$P5_1$ and $P5_2$ function as $\overline{OE}$ and $\overline{CE}$ input. Connect $P5_0$, $P5_3$, $P5_4$, and $P5_5$ to V_{CC}, and $P5_6$ and $P5_7$ to V_{SS}.
$P6_0$–$P6_7$	Input port P6	Input	Connect to V_{SS}.
$P7_0$–$P7_7$	Input port P7	Input	Connect to V_{SS}.
$P8_0$–$P8_7$	Input port P8	Input	Connect to V_{SS}.

(1) EPROM mode

The EPROM mode is entered by setting the $\overline{RESET}$ pin to "L" level. In EPROM mode, ports P0, P1, P2, $P5_1$, $P5_2$ and pins CNV_{SS} and BYTE become EPROM pins (M5M27C256K equivalent) and read/write to built-in PROM can be performed in the same manner as for M5M27C256K. However, there is no device identification code. Therefore, program conditions must be set carefully. X_{IN} and X_{OUT} pins must be connected to a clock (ceramic resonator or an external input).

Table 8.2.2 shows the pin assignments in EPROM mode and Figure 8.2.2 shows the pin connections in EPROM mode.

The program area should specify the following:

Addresses 4000_{16}–$7FFF_{16}$ for the models that have internal 16K bytes PROM, and 512 bytes RAM.
Addresses 0000_{16}–$7FFF_{16}$ for the models that have internal 32K bytes PROM, and 2048 bytes RAM.

Caution : Describing in this section, the built-in PROM can be written to or read in the same way as with the M5M27C256K (256K mode).
But in the future, for M37702E2BXXXFP, M37702E2BFS, M37702E4BXXXFP and M37702E4BFS, 1M mode may become standard.

Table 8.2.2 Pin assignments in EPROM mode

	M37702E2-XXXFP	M5M27C256K
V_{CC}	V_{CC}	V_{CC}
V_{PP}	CNV_{SS}, BYTE	V_{PP}
V_{SS}	V_{SS}	V_{SS}
Address input	Ports P0, $P1_0$–$P1_6$	A_0–A_{14}
Data I/O	Port P2	D_0–D_7
$\overline{CE}$	$P5_2$	$\overline{CE}$
$\overline{OE}$	$P5_1$	$\overline{OE}$

●Read

To read the EPROM, set the $\overline{CE}$ and $\overline{OE}$ pins to "L" level and input the address of the data (A_0–A_{14}) to be read. The data will be output to the data I/O pins D_0–D_7. The data I/O pins will be floating when either the $\overline{CE}$ or $\overline{OE}$ pin is at "H" level.

●Write

To write to the EPROM, set the $\overline{OE}$ pin to "H" level. The CPU enters the program mode when V_{PP} is applied to the V_{PP} pin. Set the address to be written to with pins A_0–A_{14} and input the data to be written through the data input pins D_0–D_7. The data is written when the $\overline{CE}$ pin is set to "L" level.

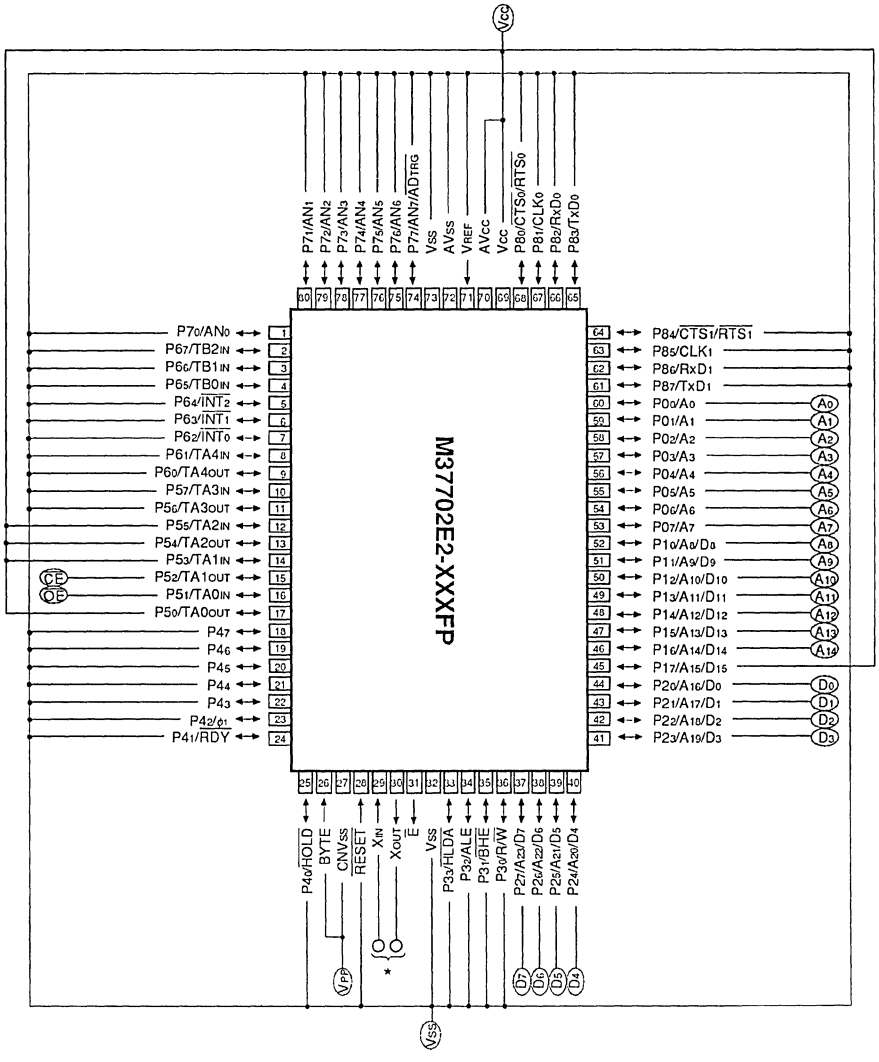
●Erase (EPROM version only)

The program is erased by exposing the glass window on top of the package to an ultraviolet light having a wave length of 2537 Angstrom. The light must be at least 15W·s/cm².

Table 8.2.3 Input/Output signals in each mode

Mode \ Pin name	$\overline{CE}$	$\overline{OE}$	V_{PP}	V_{CC}	Data I/O
Read-out	V_{IL}	V_{IL}	5V	5V	Output
Output	V_{IL}	V_{IH}	5V	5V	Floating
disable	V_{IH}	×	5V	5V	Floating
Programming	V_{IL}	V_{IH}	12.5V	6V	Input
Programming verify	V_{IH}	V_{IL}	12.5V	6V	Output
Program disable	V_{IH}	V_{IH}	12.5V	6V	Floating

Note: "X" indicates either V_{IL} or V_{IH}.



Outline 80P6N

✱ : Connect to ceramic oscillating circuit.

◯ : Same function as EPROM (M5M27C256K).

Fig. 8.2.2 Pin connections in EPROM mode

8.2.3 Fast programming algorithm

To program the M37702E2-XXXFP using a fast programming algorithm, first set V_{CC}=6V, V_{PP}=12.5V, and address to 0_{16}. Then apply a 1ms write pulse, check that the data can be read. If it cannot be read, repeat the procedure until the data can be read. Record the number of pulses applied (N) before the data was read and then write the data again, further applying three times the number of pulses (3 x N ms).
When this series of write operation is complete, increment the address and repeat the above procedure until the last address is reached.
Finally, after writing to all addresses, read with V_{CC}=V_{PP}=5V (or V_{CC}=V_{PP}=5.25V).
Figure 8.2.3 shows the fast programming algorithm flow chart.

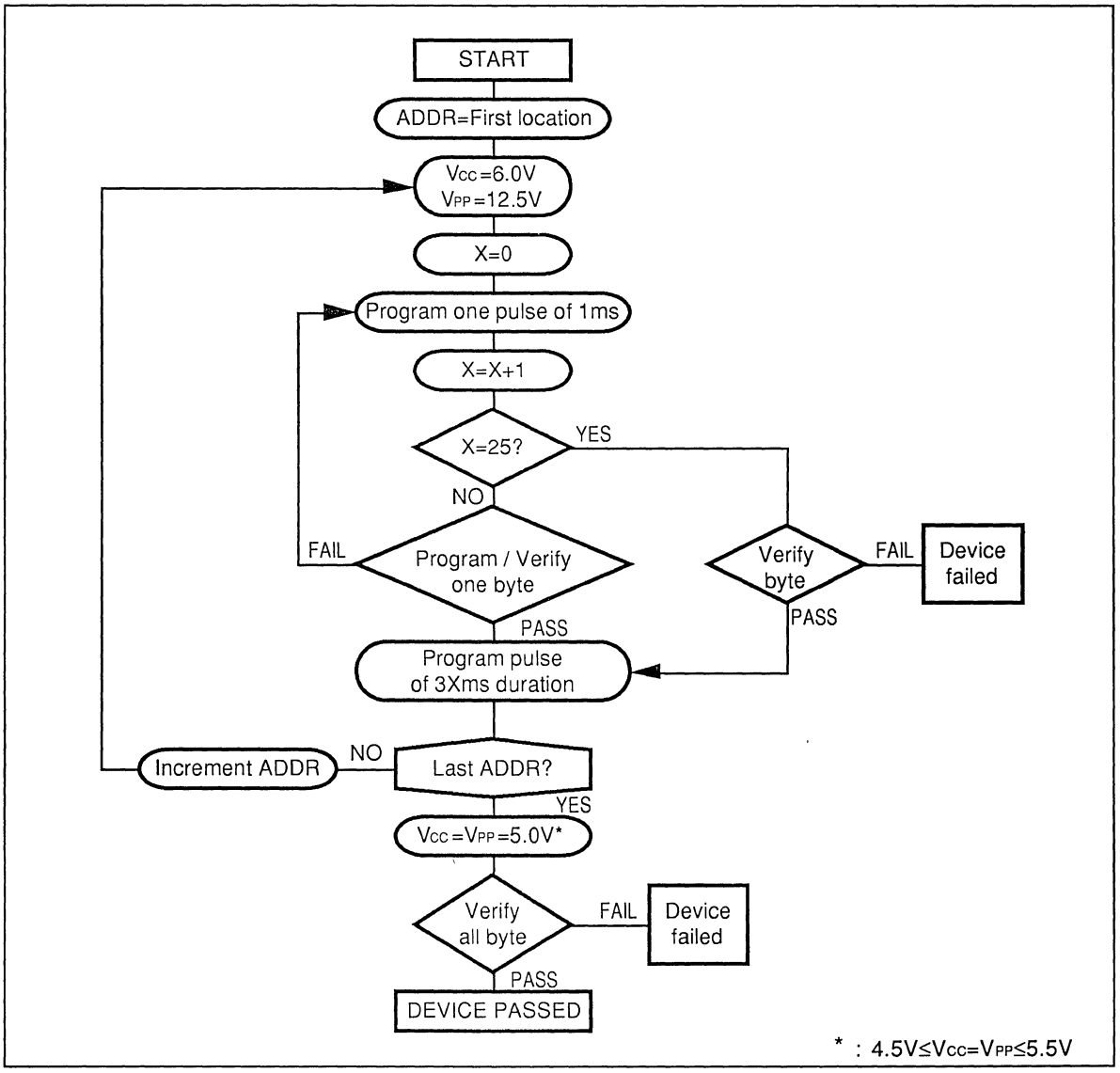


Fig. 8.2.3 Fast programming algorithm flow chart

(1) Electrical characteristics of the fast programming algorithm

Table 8.2.4 AC electrical characteristics (T_a=25±5°C, V_{CC}=6V±0.25V, V_{PP}=12.5±0.3V, 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 enable to output float delay	0		130	ns
t_{VCS}	V_{CC} 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}	$\overline{CE}$ over program pulse width	2.85		78.75	ms
t_{OE}	Data valid from $\overline{OE}$			150	ns

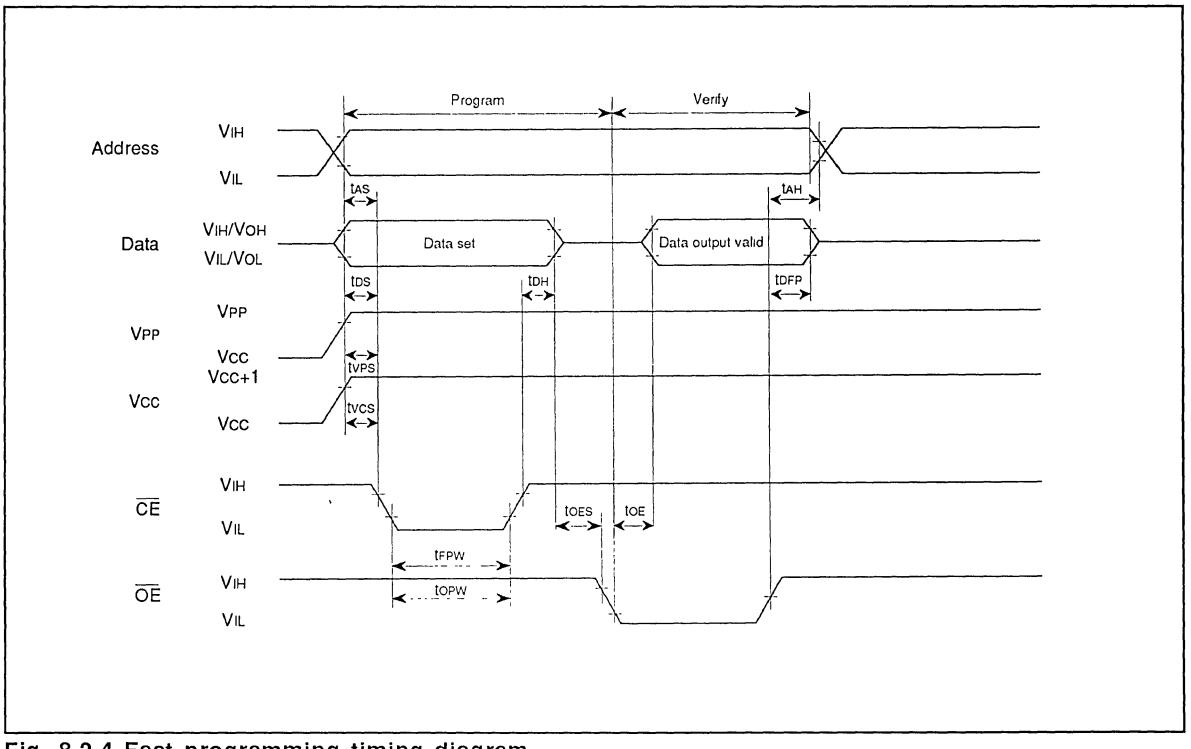
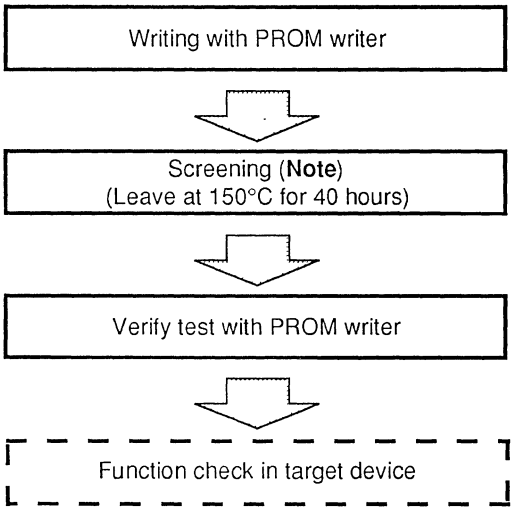


Fig. 8.2.4 Fast programming timing diagram

8.3 Usage precaution

[Precaution on all internal PROM versions]
High voltage is required to write to the built-in PROM. Be careful not to apply excessive voltage. Be especially careful during power-on.

[Precaution on one time PROM version]
User programmable one time PROM versions (M37702E2FP, M37702E2AFP, M37702E2BFP, M37702E4FP, M37702E4AFP, M37702E4BFP) that are shipped in blank are also provided. A write test and screening after assembly process are not performed for these models.
To improve their reliability after writing, we recommend that they are written and tested as flow shown in Figure 8.3.1.



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

Fig. 8.3.1 Writing and test flow for one time PROM version

[Precaution on EPROM version]
- Cover the transparent glass window during read mode because exposing to sun light or fluorescent lamp can cause the information to be erased.
- A shield to cover the transparent window is available from Mitsubishi Electric corp.. Be careful that the shield does not touch the microcomputer lead pins.
- Clean the transparent glass before erasing. Fingers' flat and paste disturb the passage of ultraviolet rays and may affect badly the erasure capability.
- Use a fit IC socket to mount the EPROM version models except for evaluation. Settle the ceramic package in an IC socket with silicon resin and the like, surely.

MEMO

CHAPTER 9
M37703 GROUP

9.1 Product expansion

The M37703M2-XXXSP is equal to the M37702M2-XXXFP enclosed in a 64-pin shrink plastic molded DIP. The M37703 group consists of chips shown in Table 9.1.1 with the M37703M2-XXXSP as the base chip. These chips are all pin compatible with each other. Only the memory type and size, and operating clock are different.

Table 9.1.1 Product expansion of M37703 group

Type name	ROM	RAM	Clock frequency
M37703M2-XXXSP	Mask ROM 16K bytes	512 bytes	8MHz
M37703M2AXXXSP			16MHz
M37703M2BXXXSP			25MHz
M37703S1SP	—		8MHz
M37703S1ASP			16MHz
M37703S1BSP			25MHz
M37703E2-XXXSP	One time PROM 16K bytes		8MHz
M37703E2AXXXSP			16MHz
M37703E2BXXXSP			25MHz
M37703M4-XXXSP	Mask ROM 32K bytes	2048 bytes	8MHz
M37703M4AXXXSP			16MHz
M37703M4BXXXSP			25MHz
M37703S4SP	—		8MHz
M37703S4ASP			16MHz
M37703S4BSP			25MHz
M37703E4-XXXSP	One time PROM 32K bytes		8MHz
M37703E4AXXXSP			16MHz
M37703E4BXXXSP			25MHz

9.1.1 M37703M2-XXXSP characteristics

- Number of basic instructions 103
- Memory size ROM ..... 16K bytes
  RAM ..... 512 bytes
- Instruction execution time (the fastest instruction at 8MHz) 500ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz) ..... 30mW (Typ.)
- Interrupts ..... 19 sources, 7 levels
- Multi-function 16-bit timers ..... 5+3
- UART 2
- 8-bit A-D converter ..... 4-channel input
- Watchdog timer
- Programmable I/O (ports P0, P1, P2, P3, P4, P5, P6, P7, and P8) 53

9.2 M37703M2-XXXSP

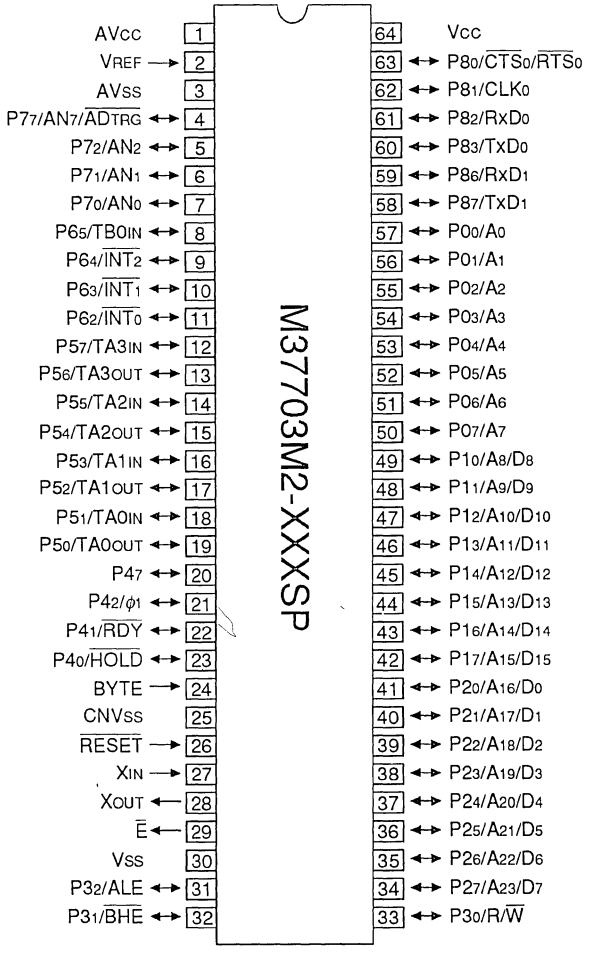
The following descriptions will be for M37703M2-XXXSP. The products of M37703 group have the same functions as the M37703M2-XXXSP unless otherwise noted.

9.2.1 Description

The M37703M2-XXXSP is a 16-bit single-chip microcomputer designed with high-performance CMOS silicon gate technology. It is housed in a 64-pin shrink plastic molded DIP.

This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts data.

Figure 9.2.1 shows the M37703M2-XXXSP pin configuration.



Outline 64P4B

Fig. 9.2.1 M37703M2-XXXSP pin configuration

9.2.2 Performance overview

Table 9.2.1 shows the performance overview of the M37703M2-XXXSP.

Table 9.2.1 M37703M2-XXXSP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37703M2-XXXSP	500ns (the fastest instruction at 8MHz frequency)
	M37703M2AXXXSP	250ns (the fastest instruction at 16MHz frequency)
	M37703M2BXXXSP	160ns (the fastest instruction at 25MHz frequency)
Clock frequency	M37703M2-XXXSP	8MHz (maximum)
	M37703M2AXXXSP	16MHz (maximum)
	M37703M2BXXXSP	25MHz (maximum)
Memory size	ROM	16384 bytes
	RAM	512 bytes
Input/Output ports	Ports P0, P1, P2, P5	8 bits × 4
	Port P8	6 bits × 1
	Ports P4, P6, P7	4 bits × 3
	Port P3	3 bits × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits × 5 (4 with I/O functions)
	TB0, TB1, TB2	16 bits × 3 (1 with I/O functions)
Serial I/O		Clock asynchronous serial I/O × 2 (UART0 can also be used as clock synchronous)
A-D converter		8 bits × 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 quartz crystal resonator)
Supply voltage		5V±10%
Power dissipation		30mW (at external 8MHz frequency)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		–20 to 85°C
Device structure		CMOS high-performance silicon gate process
Package		64-pin shrink plastic molded DIP

9.2.3 Differences between M37703M2-XXXSP and M37702M2-XXXFP

Table 9.2.2 shows the differences between M37703M2-XXXSP and M37702M2-XXXFP.

Table 9.2.2 Differences between M37703M2-XXXSP and M37702M2-XXXFP

Functions			M37703M2-XXXSP	M37702M2-XXXFP
I/O ports			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 ($P3_3$/$\overline{HLDA}$ unavailable)	4 bits
	Port P4		4 bits ($P4_3$–$P4_6$ unavailable)	8 bits
	Port P5		8 bits	8 bits
	Port P6		4 bits ($P6_0$, $P6_1$, $P6_6$, and $P6_7$ unavailable)	8 bits
	Port P7		4 bits ($P7_3$–$P7_6$ unavailable)	8 bits
	Port P8		6 bits ($P8_4$ and $P8_5$ unavailable)	8 bits
Timers			16 bits × 8	16 bits × 8
	Timer A	TA0	Timer I/O pins available	Timer I/O pins available
		TA1	Input=TAj_{IN}, output=TAj_{OUT} (j=0 to 3)	Input=TAi_{IN}, output=TAi_{OUT} (i=0 to 4)
		TA2		
		TA3		
		TA4	Internal timer ($TA4_{IN}$ and $TA4_{OUT}$ unavailable)	
	Timer B	TB0	Timer input pin ($TB0_{IN}$) available	Timer input pin (TBk_{IN}) available
		TB1	Internal timer ($TB1_{IN}$ and $TB2_{IN}$	(k=0 to 2)
		TB2	unavailable)	
Serial I/O			2	2
	UART0		Clock asynchronous/synchronous serial I/O	Clock asynchronous/synchronous serial I/O
	UART1		Clock asynchronous serial I/O	Clock asynchronous/synchronous serial I/O
A-D converter			One 8-bit resolution 4-channel analog input pin AN_0, AN_1, AN_2, AN_7 (AN_3–AN_6 unavailable) **Note** : AN_7 pin is in common with external trigger pin.	One 8-bit resolution 8-channel analog input pin AN_0, AN_1, AN_2, AN_3, AN_4, AN_5, AN_6, AN_7 **Note** : AN_7 pin is in common with external trigger pin.
Package			64-pin shrink plastic molded DIP (64P4B)	80-pin plastic molded QFP (80P6N)

9.2.4 Functional description

The internal circuit of the M37703M2-XXXSP is identical to that of the M37702M2-XXXFP including the control registers and memory allocation in SFR area. However, since the M37703M2-XXXSP has only 64 pins, some functions are different from the M37702M2-XXXFP.
The functional differences are described below.

(1) A-D converter

Analog input pins are 4 channels of AN_0–AN_2, and AN_7 pins.

[One-shot mode and repeat mode]

The analog input pin selection bits in the A-D control register must be set to "000", "001", "010", or "111".
Bits 3 to 6 in the port P7 direction register must be set to "1" to select output mode because AN_3–AN_6 pins are not available.

[Single sweep mode and repeat sweep mode]

Bits 0 to 2 and bit 7 in the port P7 direction register must be set to "0" to select input mode. Bits 3 to 6 in the port P7 direction register must be set to "1" to select output mode because AN_3–AN_6 pins are not available.
The contents of the corresponding A-D registers to analog input AN_3–AN_6 which have no input pins are undefined.

(2) Timers

I/O functions of timer A4, and input functions of timers B1 and B2 are not available. Therefore, these timers operate only in timer mode. Only count source can be selected for timers A4, B1 and B2, and bits 0 to 5 in each timer mode register must be fixed to "0".
Other timers (timers A0 to A3, and timer B0) have the same functions as the M37702M2-XXXFP.

(3) Serial I/O

UART1 can be used only in UART mode. It cannot be used in clock synchronous serial I/O mode. Therefore, the serial I/O mode selection bits in the UART1 transmit/receive mode register must be set to the value except for "001".
The $\overline{CTS}/\overline{RTS}$ function selection bit in the UART1 transmit/receive control register 0 must be fixed to "1" because $\overline{CTS}/\overline{RTS}$ function is not available (this bit is set to "0" at reset).
UART0 has the same functions as the M37702M2-XXXFP.

(4) Ports

The port direction registers for ports P4, P6, P7, and P8 contain 8 bits. However, the bits in the each direction register with no corresponding pins must be set to "1" to select output mode. The port P3 direction register bit 3 which is corresponding to port $P3_3$ must be set to "1" to select output mode.

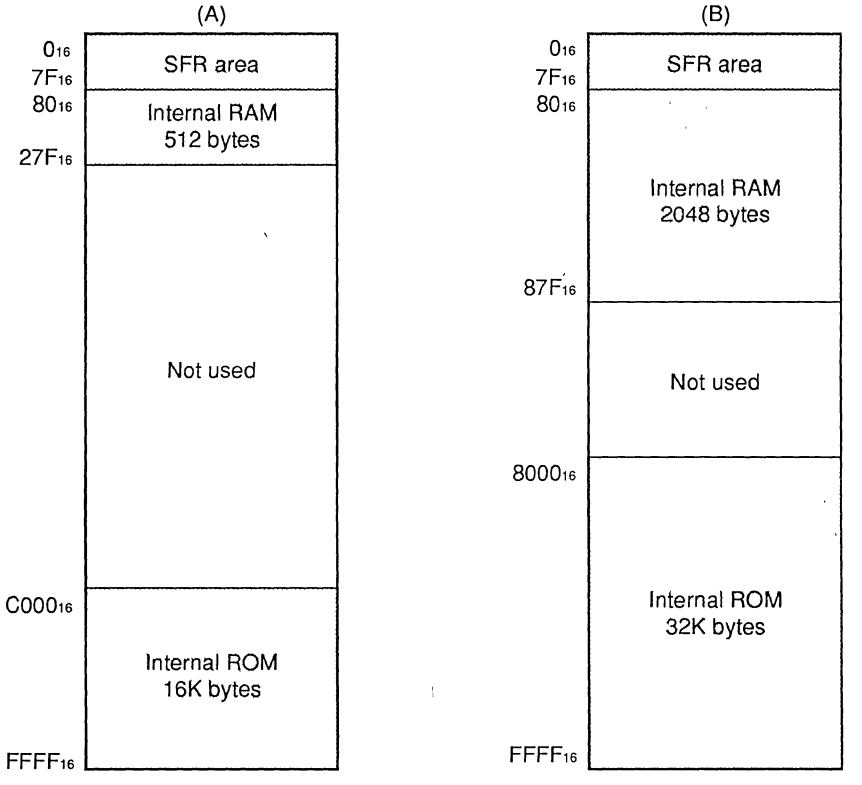
APPENDIX

APPENDIX

Appendix 1. M37702 group memory map

1. Memory map in single-chip mode

Figure 1 shows the memory map in single-chip mode.



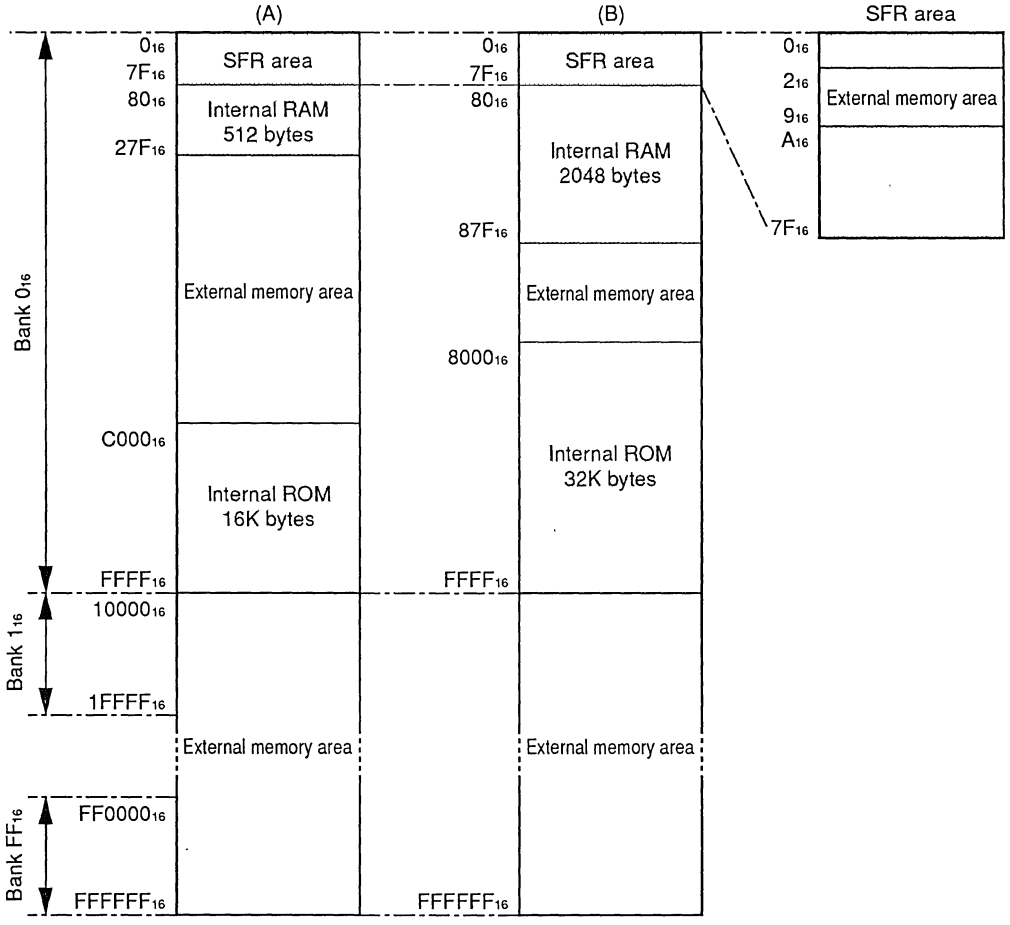
No.	Type name	Internal ROM
(A)	M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP	Mask ROM
	M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP	One time PROM
	M37702E2FS, M37702E2AFS, M37702E2BFS	EPROM
(B)	M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP	Mask ROM
	M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP	One time PROM
	M37702E4FS, M37702E4AFS, M37702E4BFS	EPROM

Fig. 1 Memory map in single-chip mode

APPENDIX

2. Memory map in memory expansion mode

Figure 2 shows the memory map in memory expansion mode.



No.	Type name	Internal ROM
(A)	M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP	Mask ROM
	M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP	One time PROM
	M37702E2FS, M37702E2AFS, M37702E2BFS	EPROM
(B)	M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP	Mask ROM
	M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP	One time PROM
	M37702E4FS, M37702E4AFS, M37702E4BFS	EPROM

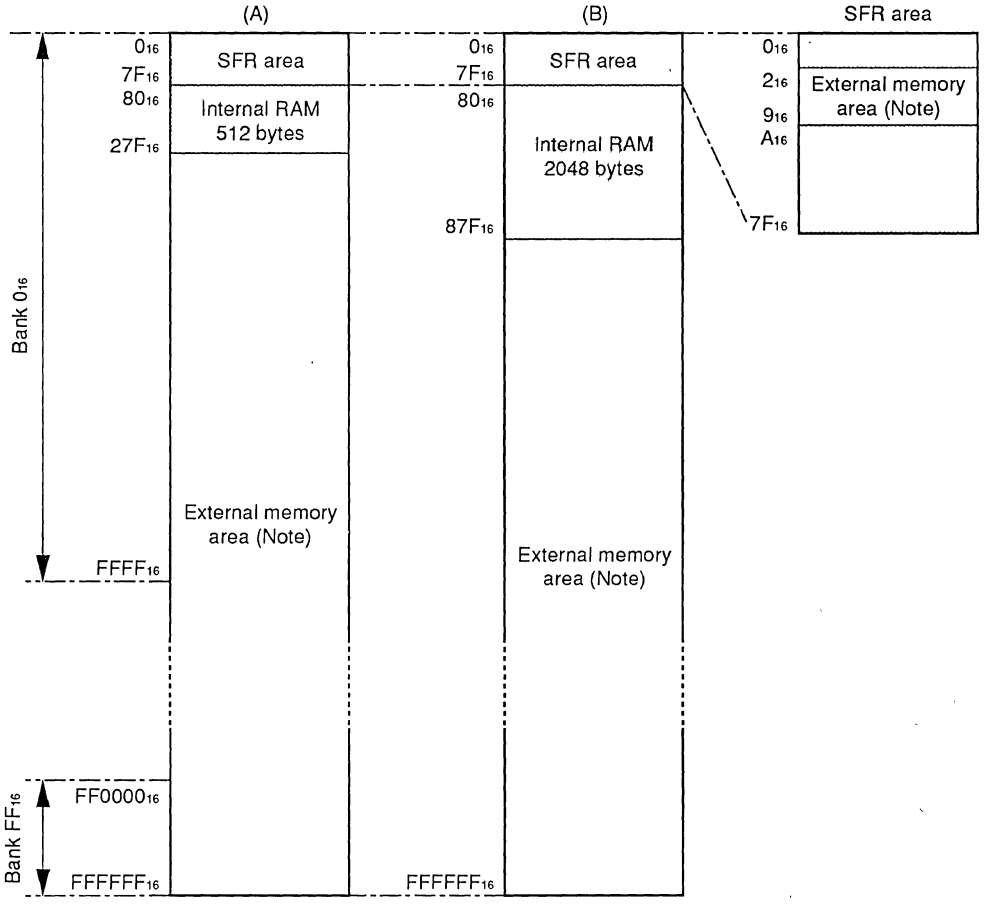
Fig. 2 Memory map in memory expansion mode

APPENDIX

Appendix 1. M37702 group memory map

3. Memory map in microprocessor mode and of external ROM version

Figure 3 shows the memory map in microprocessor mode and of external ROM version.



No.	Type name	Internal ROM
(A)	M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP	Mask ROM
	M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP	One time PROM
	M37702E2FS, M37702E2AFS, M37702E2BFS	EPROM
	M37702S1FP, M37702S1AFP, M37702S1BFP	External ROM
(B)	M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP	Mask ROM
	M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP	One time PROM
	M37702E4FS, M37702E4AFS, M37702E4BFS	EPROM
	M37702S4FP, M37702S4AFP, M37702S4BFP	External ROM

Note: Address FFD6$_{16}$ to FFFF$_{16}$ in bank 0$_{16}$ is the interrupt vector table and must be assigned to ROM.

Fig. 3 Memory map in microprocessor mode and of external ROM version

Appendix 2. SFR area memory map

Address (Hexadecimal notation)	Registers	Access
000000		
000001		
000002	Port P0 register	RW
000003	Port P1 register	RW
000004	Port P0 direction register	RW
000005	Port P1 direction register	RW
000006	Port P2 register	RW
000007	Port P3 register	→
000008	Port P2 direction register	RW
000009	Port P3 direction register	→
00000A	Port P4 register	RW
00000B	Port P5 register	RW
00000C	Port P4 direction register	RW
00000D	Port P5 direction register	RW
00000E	Port P6 register	RW
00000F	Port P7 register	RW
000010	Port P6 direction register	RW
000011	Port P7 direction register	RW
000012	Port P8 register	RW
000013		
000014	Port P8 direction register	RW
000015		
00001D		
00001E	A-D control register	RW
00001F	A-D sweep pin selection register	RW
000020	A-D register 0	RO
000021		
000022	A-D register 1	RO
000023		
000024	A-D register 2	RO
000025		

RO ··· Read only
WO ··· Write only
RW ··· Read/Write
? ····· Impossible to write, and undefine at reading
0 ····· Impossible to write, and fixed to "0" at reading

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

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

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	?	?	RW	RW

Address (Hexadecimal notation)	Registers	Access
000026	A-D register 3	RO
000027		
000028	A-D register 4	RO
000029		
00002A	A-D register 5	RO
00002B		
00002C	A-D register 6	RO
00002D		
00002E	A-D register 7	RO
00002F		
000030	UART0 transmit/receive mode register	RW
000031	BRG0 register	WO
000032	UART0 transmission buffer register (L)	WO
000033	UART0 transmission buffer register (H)	
000034	UART0 transmit/receive control register 0	→
000035	UART0 transmit/receive control register 1	→
000036	UART0 receive buffer register (L)	RO
000037	UART0 receive buffer register (H)	→
000038	UART1 transmit/receive mode register	RW
000039	BRG1 register	WO
00003A	UART1 transmission buffer register (L)	WO
00003B	UART1 transmission buffer register (H)	
00003C	UART1 transmit/receive control register 0	→
00003D	UART1 transmit/receive control register 1	→
00003E	UART1 receive buffer register (L)	RO
00003F	UART1 receive buffer register (H)	→
000040	Count start flag	RW
000041		
000042	One-shot start flag	WO
000043		
000044	Up-down flag	→
000045		
000046	Timer A0 register (L)	RW
000047	Timer A0 register (H)	
000048	Timer A1 register (L)	RW
000049	Timer A1 register (H)	

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

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

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

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

b7	b6	b5	b4	b3	b2	b1	b0
WO	WO	WO	RW	RW	RW	RW	RW

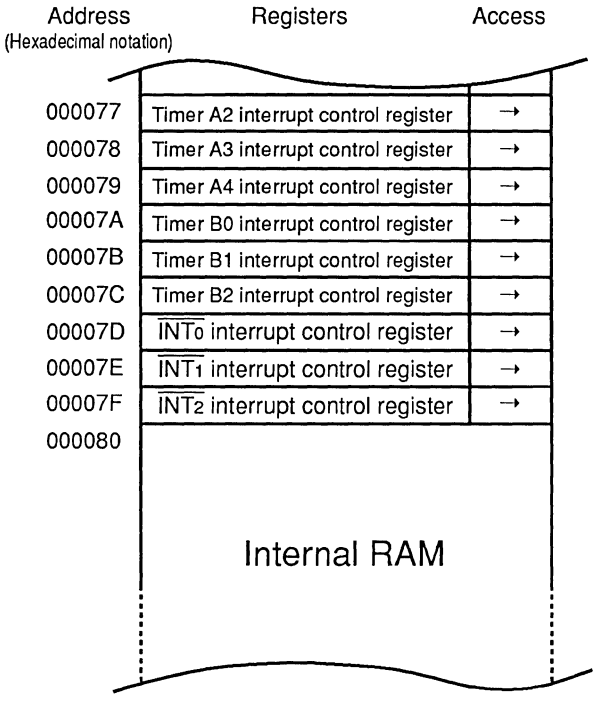
Appendix 2. SFR area memory map

Address (Hexadecimal notation)	Registers	Access
00004A 00004B	Timer A2 register (L / H)	RW
00004C 00004D	Timer A3 register (L / H)	RW
00004E 00004F	Timer A4 register (L / H)	RW
000050 000051	Timer B0 register (L / H)	RW
000052 000053	Timer B1 register (L / H)	RW
000054 000055	Timer B2 register (L / H)	RW
000056	Timer A0 mode register	RW
000057	Timer A1 mode register	RW
000058	Timer A2 mode register	RW
000059	Timer A3 mode register	RW
00005A	Timer A4 mode register	RW
00005B	Timer B0 mode register	RW
00005C	Timer B1 mode register	RW
00005D	Timer B2 mode register	RW
00005E	Processor mode register	→
00005F		
000060	Watchdog timer	WO
000061	Watchdog timer frequency selection flag	RW
000062		
000063		
000064		
000065		
000070	A-D conversion interrupt control regiser	→
000071	UART0 transmission interrupt control regiter	→
000072	UART0 receive interrupt control register	→
000073	UART1 transmission interrupt control regiter	→
000074	UART1 receive interrupt control register	→
000075	Timer A0 interrupt control register	→
000076	Timer A1 interrupt control register	→

b7	b6	b5	b4	b3	b2	b1	b0
RW	RW	RW	RW	WO	RW	RW	RW

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	?	?	?	RW

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW



Address (Hexadecimal notation)	Registers	Access
000077	Timer A2 interrupt control register	→
000078	Timer A3 interrupt control register	→
000079	Timer A4 interrupt control register	→
00007A	Timer B0 interrupt control register	→
00007B	Timer B1 interrupt control register	→
00007C	Timer B2 interrupt control register	→
00007D	$\overline{INT_0}$ interrupt control register	→
00007E	$\overline{INT_1}$ interrupt control register	→
00007F	$\overline{INT_2}$ interrupt control register	→
000080	Internal RAM	

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	?	?	RW	RW	RW	RW
?	?	RW	RW	RW	RW	RW	RW
?	?	RW	RW	RW	RW	RW	RW
?	?	RW	RW	RW	RW	RW	RW

APPENDIX

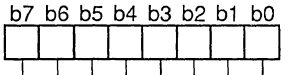
Appendix 3. Control registers

The register structure of each control register allocated in the SFR area are shown on the following pages. Each table shows the bit names, functions, content when reset is removed, and bit attributes.

* Bit attributes: Each bit in the control register is either read only, write only, or read/write. The following abbreviations are used to indicate the attribute.

  R : Read
  W : Write
  ○ : Possible to read or write
  × : Impossible to read or write

1.Port Pi direction registers (i=0–8)

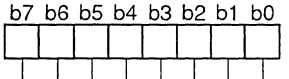


Port Pi direction register
(Address 04_{16},05_{16},08_{16},09_{16},$0C_{16}$,$0D_{16}$,10_{16},11_{16},14_{16})

Bit	Bit name	Functions	At reset	R	W
0	Port P_{i0} direction selection bit	0 : Input mode 1 : Output mode	0	○	○
1	Port P_{i1} direction selection bit		0	○	○
2	Port P_{i2} direction selection bit		0	○	○
3	Port P_{i3} direction selection bit		0	○	○
4	Port P_{i4} direction selection bit		0	○	○
5	Port P_{i5} direction selection bit		0	○	○
6	Port P_{i6} direction selection bit		0	○	○
7	Port P_{i7} direction selection bit		0	○	○

Note: The high-order 4 bits of port P3 dierction register are write prohibited and these bits are fixed to "0" at reading.

2.A-D control register

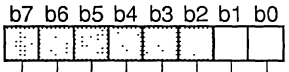


A-D control register (Address $1E_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Analog input selection bits	b2b1b0 0 0 0 : Select AN_0 0 0 1 : Select AN_1 0 1 0 : Select AN_2 0 1 1 : Select AN_3 1 0 0 : Select AN_4 1 0 1 : Select AN_5 1 1 0 : Select AN_6 1 1 1 : Select AN_7 (Note)	Undefined	○	○
1			Undefined	○	○
2			Undefined	○	○
3	A-D mode selection bits	b4b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweet mode	0	○	○
4			0	○	○
5	Trigger selection bit	0 : Software trigger (internal trigger) 1 : $\overline{AD_{TRG}}$ input trigger (external trigger)	0	○	○
6	A-D conversion start flag	0 : Stop A-D conversion 1 : Start A-D conversion	0	○	○
7	A-D conversion frequency (ϕ_{AD}) selection flag	0 : Select $f(X_{IN})/8$ 1 : Select $f(X_{IN})/4$	0	○	○

Note. Pin AN_7 cannot be used as analog voltage input pin when an external trigger is selected.

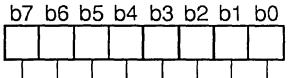
3.A-D sweep pin selection register



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

Bit	Bit name	Functions	At reset	R	W
0	A-D sweep pin selection bits	b1b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0–AN_3 (4 pins) 1 0 : AN_0–AN_5 (6 pins) 1 1 : AN_0–AN_7 (8 pins)	1	○	○
1			1	○	○
2	These bits cannot be written and are undefined at reading.		Undefined	×	×
3			Undefined	×	×
4			Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

4.UARTi transmit/receive mode registers (i=0, 1)

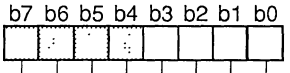


UART0 transmit/receive mode register (Address 30_{16})
UART1 transmit/receive mode register (Address 38_{16})

Bit	Bit name	Functions	At reset	R	W
0	Serial I/O mode selection bits	b2b1b0 0 0 0 : Serial I/O prohibited 0 0 1 : Clock synchronous serial I/O 0 1 0 : This cannot be available. 0 1 1 : This cannot be available. 1 0 0 : 7-bit UART 1 0 1 : 8-bit UART 1 1 0 : 9-bit UART 1 1 1 : This cannot be available.	0	○	○
1			0	○	○
2			0	○	○
3	Internal/external clock selection bits	0 : Internal clock 1 : External clock	0	○	○
4	Stop bit length selection bit (in UART mode)	0 : One stop bit 1 : Two stop bits	0	○	○
5	Odd/even parity selection bit (in UART mode)	0 : Odd parity 1 : Even parity	0	○	○
6	Parity enable bit (in UART mode)	0 : Parity disabled 1 : Parity enabled	0	○	○
7	Sleep function selection bit (in UART mode)	0 : Sleep function disabled 1 : Sleep function enabled	0	○	○

Note: Bits 4 to 6 are ignored in clock synchronous mode.
Bit 7 must be "0" when using clock synchronous mode.

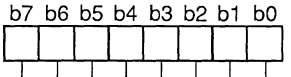
5.UARTi transmit/receive control register 0 (i=0, 1)



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

Bit	Bit name	Functions	At reset	R	W
0	BRG count source selection bits	b1b0 0 0 : Select f(X_{IN})/2 (f_2) 0 1 : Select f(X_{IN})/16 (f_{16}) 1 0 : Select f(X_{IN})/64 (f_{64}) 1 1 : Select f(X_{IN})/512 (f_{512})	0	○	○
1			0	○	○
2	$\overline{CTS}/\overline{RTS}$ function selection bit	0 : Select $\overline{CTS}$ function 1 : Select $\overline{RTS}$ function	0	○	○
3	Transmission register empty flag	0 : Data in transmission register (transmitting) 1 : No data in transmission register (transmit complete)	1	○	×
4	These bits cannot be written and are undefined at reading.		Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

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

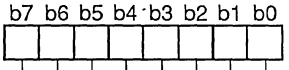


UART0 transmit/receive control register 1 (Address 35_{16})
UART1 transmit/receive control register 1 (Address $3D_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Transmit enable bit	0 : Transmission disable 1 : Transmission enable	0	○	○
1	Transmission buffer empty flag	0 : Data in transmission buffer register 1 : No data in transmission buffer register	1	○	×
2	Receive enable bit	0 : Receive disable 1 : Receive enable	0	○	○
3	Receive completion flag	0 : No data in receive buffer register 1 : Data in receive buffer register	0	○	×
4	Overrun error flag	0 : No overrun error 1 : Overrun error	0	○	×
5	Framing error flag (in UART mode)	0 : No framing error 1 : Framing error	0	○	×
6	Parity error flag (in UART mode)	0 : No parity error 1 : Parity error	0	○	×
7	Error sum flag (in UART mode)	0 : No error 1 : Error	0	○	×

Note: Bits 5 to 7 are ignored in clock synchronous mode.
Each error flag is cleared to "0" when the receive buffer register is read.

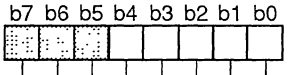
7.Count start flag



Count start flag (Address 40_{16})

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 count start flag	0 : Count stop 1 : Count start	0	○	○
1	Timer A1 count start flag		0	○	○
2	Timer A2 count start flag		0	○	○
3	Timer A3 count start flag		0	○	○
4	Timer A4 count start flag		0	○	○
5	Timer B0 count start flag		0	○	○
6	Timer B1 count start flag		0	○	○
7	Timer B2 count start flag		0	○	○

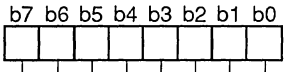
8.One-shot start flag



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

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 one-shot start flag	1 : One-shot start	0	X	○
1	Timer A1 one-shot start flag		0	X	○
2	Timer A2 one-shot start flag		0	X	○
3	Timer A3 one-shot start flag		0	X	○
4	Timer A4 one-shot start flag		0	X	○
5	These bits cannot be written and are undefined at reading.		Undefined	X	X
6			Undefined	X	X
7			Undefined	X	X

9.Up-down flag

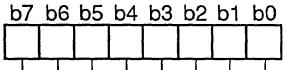


Up-down flag (Address 44_{16})

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 up-down flag	0 : Down count 1 : Up count	0	○	○
1	Timer A1 up-down flag		0	○	○
2	Timer A2 up-down flag		0	○	○
3	Timer A3 up-down flag		0	○	○
4	Timer A4 up-down flag		0	○	○
5	Timer A2 two-phase signal processing selection bit	0 : Two-phase pulse signal processing disable 1 : Two-phase pulse signal processing enable	0	×	○
6	Timer A3 two-phase signal processing selection bit		0	×	○
7	Timer A4 two-phase signal processing selection bit		0	×	○

Note: Data must be written using **LDM** or **STA** instruction for bits 5–7.

10.Timer Ai mode registers (i=0–4)

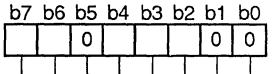


Timer Ai mode register (Addresses 56_{16}–$5A_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Operating mode selection bits	b1b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : PWM mode	0	○	○
1			0	○	○
2	These bits' functions depend on operating mode.		0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6	Count source selection bits	b7b6 0 0 : Select f(X_{IN})/2 (f_2) 0 1 : Select f(X_{IN})/16 (f_{16}) 1 0 : Select f(X_{IN})/64 (f_{64}) 1 1 : Select f(X_{IN})/512 (f_{512})	0	○	○
7			0	○	○

Note: In event counter mode, bits 6 and 7 are ignored.

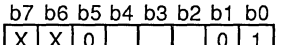
(1)Timer mode



Timer Ai mode register <Timer mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 0 : Timer mode
1		
2	Pulse output function selection bit	0 : No pulse output 1 : Pulse output
3	Gate function selection bits	b4b3 0 X : No gate function 1 0 : Count while TA_{iIN} input level is "L" 1 1 : Count while TA_{iIN} input level is "H"
4		
5	This bit must be fixed to "0".	
6	Count source selection bits	b7b6 0 0 : Select f(X_{IN})/2 (f_2) 0 1 : Select f(X_{IN})/16 (f_{16}) 1 0 : Select f(X_{IN})/64 (f_{64}) 1 1 : Select f(X_{IN})/512 (f_{512})
7		

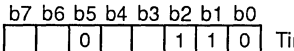
(2)Event counter mode



Timer Ai mode register <Event counter mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 1 : Event counter mode
1		
2	Pulse output function selection bit	0 : No pulse output 1 : Pulse output
3	Count polarity selection bit	0 : Count at the falling edge of the input signal 1 : Count at the rising edge of the input signal
4	Up-down switching factor selection bit	0 : Content of the up-down flag 1 : Input signal of the TAiOUT pin
5	This bit must be fixed to "0".	
6	These bits are ignored (may be "0" or "1").	
7		

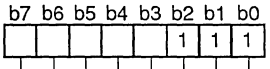
(3)One-shot pulse mode



Timer Ai mode register <One-shot pulse mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 0 : One-shot pulse mode
1		
2	This bit must be fixed to "1".	
3	Trigger selection bits	b4b3 0 X : Internal trigger (Writing operation to the one-shot start flag) 1 0 : Falling edge of the input signal toTAiIN input 1 1 : Rising edge of the input signal to TAiIN input
4		
5	This bit must be fixed to "0".	
6	Count source selection bits	b7b6 0 0 : Select f(XIN)/2 (f_2) 0 1 : Select f(XIN)/16 (f_{16}) 1 0 : Select f(XIN)/64 (f_{64}) 1 1 : Select f(XIN)/512 (f_{512})
7		

(4)PWM mode



Timer Ai mode register <PWM mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 1 : PWM mode
1		
2	This bit must be fixed to "1".	
3	Trigger selection bits	b4b3 0 X : Internal trigger (Writing operation to the one-shot start flag) 1 0 : Falling edge of the input signal TAiIN input 1 1 : Rising edge of the input signal TAiIN input
4		
5	16/8-bit PWM mode selection bit	0 : 16-bit PWM mode 1 : 8-bit PWM mode
6	Count source selection bits	b7b6 0 0 : Select f(XIN)/2 (f2) 0 1 : Select f(XIN)/16 (f16) 1 0 : Select f(XIN)/64 (f64) 1 1 : Select f(XIN)/512 (f512)
7		

11.Timer Bi mode registers (i=0–2)

b7 b6 b5 b4 b3 b2 b1 b0

Timer Bi mode register (Address $5B_{16}$–$5D_{16}$)

Bit	Bit name	Functions	At reset	R	W
0	Operating mode selection bits	b1b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : This cannot be available.	0	○	○
1			0	○	○
2	These bits' functions depend on operating mode.		0	○	○
3			0	○	○
4	This bit cannot be written and is undefined at reading.		Undefined	×	×
5	Timer Bi overflow flag	0 : No overflow or underflow 1 : Overflow or underflow	1	○	× (Note)
6	Count source selection bits	b7b6 0 0 : Select f(XIN)/2 (f2) 0 1 : Select f(XIN)/16 (f16) 1 0 : Select f(XIN)/64 (f64) 1 1 : Select f(XIN)/512 (f512)	0	○	○
7			0	○	○

Note: The timer Bi overflow flag which is set to "1" is cleared to "0" by writing to this register.

(1)Timer mode

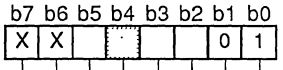
b7 b6 b5 b4 b3 b2 b1 b0

				X	X	0	0

Timer Bi mode register <Timer mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 0 : Timer mode
1		
2	These bits are ignored (may be "0" or "1").	
3		
4	This bit cannot be written and is undefined at reading.	
5	Timer Bi overflow flag	0 : No underflow 1 : Underflow
6	Count source selection bits	b7b6 0 0 : Select f(XIN)/2 (f2) 0 1 : Select f(XIN)/16 (f16) 1 0 : Select f(XIN)/64 (f64) 1 1 : Select f(XIN)/512 (f512)
7		

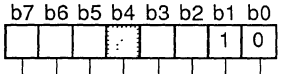
(2)Event counter mode



Timer Bi mode register <Event counter mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 1 : Event counter mode
1		
2	Count polarity selection bits	b3b2 0 0 : Count at the falling edge of the input signal 0 1 : Count at the rising edge of the input signal 1 0 : Count at both edges of the input signal 1 1 : This cannot be available.
3		
4	This bit cannot be written and is undefined at reading.	
5	Timer Bi overflow flag	0 : No underflow 1 : Underflow
6	These bits are ignored (may be "0" or "1").	
7		

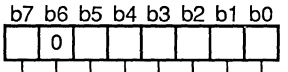
(3)Pulse period/pulse width measurement mode



Timer Bi mode register <Pulse period/pulse width measurement mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 0 : Pulse period/pulse width measurement mode
1		
2	Measurement mode selection bits	b3b2 0 0 : Pulse period measurement mode (between falling edge and the next falling edge) 0 1 : Pulse period measurement mode (between rising edge and the next rising edge) 1 0 : Pulse width measurement mode 1 1 : This cannot be available.
3		
4	This bit cannot be written and is undefined at reading.	
5	Timer Bi overflow flag	0 : No overfolw 1 : Overflow
6	Count source selection bits	b7b6 0 0 : Select f(X_{IN})/2 (f_2) 0 1 : Select f(X_{IN})/16 (f_{16}) 1 0 : Select f(X_{IN})/64 (f_{64}) 1 1 : Select f(X_{IN})/512 (f_{512})
7		

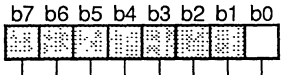
12.Processor mode register



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

Bit	Bit name	Functions	At reset	R	W
0	Processor mode bits	b1b0 0 0 : Single-chip mode 0 1 : Memory expansion mode 1 0 : Microprocessor mode 1 1 : This cannot be available.	0	○	○
1			0	○	○
2	Wait bit	0 : Wait during external access 1 : No wait	0	○	○
3	Software reset bit	Software reset activated by writing "1".	0	×	○
4	Interrupt priority detection time selection bits	b5b4 0 0 : Select 7 cycles at internal clock ϕ 0 1 : Select 4 cycles at internal clock ϕ 1 0 : Select 2 cycles at internal clock ϕ 1 1 : This cannot be available.	0	○	○
5			0	○	○
6	This bit must be fixed to "0".		0	○	○
7	Clock ϕ_1 output selection bit	0 : ϕ_1 output disabled (P4$_2$ is normal I/O port.) 1 : ϕ_1 output enable (P4$_2$ is ϕ_1 output pin.)	0	○	○

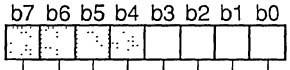
13.Watchdog timer frequency selection flag



Watchdog timer frequency selection flag (Address 61_{16})

Bit	Bit name	Functions	At reset	R	W
0	Watchdog timer frequency selection flag	0 : Select f(X$_{IN}$)/512 (f$_{512}$) 1 : Select f(X$_{IN}$)/32 (f$_{32}$)	0	○	○
1	These bits cannot be written and are undefined at reading.		Undefined	×	×
2			Undefined	×	×
3			Undefined	×	×
4			Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

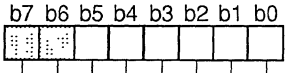
14.A-D conversion, UART 0 and 1 transmission, UART 0 and 1 receive, timers A0–A4, timers B0–B2, interrupt control registers



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

Bit	Bit name	Functions	At reset	R	W
0	Interrupt priority level selection bits	b2b1b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	○	○
1			0	○	○
2			0	○	○
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	○
4	These bits cannot be written and are undefined at reading.		Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

15.$\overline{INT_0}$–$\overline{INT_2}$ interrupt control registers



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

Bit	Bit name	Functions	At reset	R	W
0	Interrupt priority level selection bits	b2b1b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	0	○	○
1			0	○	○
2			0	○	○
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	○
4	Level/edge selection bit	0 : Set request bit at "H" level for level sense and the falling edge for edge sense. 1 : Set request bit at "L" level for level sense and the rising edge for edge sense.	0	○	○
5	Level/edge sense selection bit	0 : Edge sense 1 : Level sense	0	○	○
6	These bits cannot be written and are undefined at reading.		Undefined	×	×
7			Undefined	×	×

Appendix 4. Stop, wait, one-wait, Ready, Hold state

1. Stop, wait, one-wait, Ready, Hold state

Table 1 shows the stop, wait, one-wait, Ready, and Hold state.
The following are some notes for items in Table 1.

(1) Oscillation

Timer A, timer B, serial I/O, and A-D converter can be used when the oscillator is operating.

(2) STP instruction

For mask ROM version, whether to enable or disable the **STP** instruction is selected with the **STP** instruction option on the mask ROM order confirmation form. The **STP** instruction is always valid for PROM version and external ROM version.
Table 2 shows the external interrupts used to remove the state after executing the **STP** instruction (stop mode).

Table 2 External interrupts used to remove stop mode

External interrupt	Interrupt source
External input signal	$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$
Serial I/O using external clock (clock synchronous/asynchronous)	UART0 receive, UART0 transmission UART1 receive, UART1 transmission
Timer interrupts in event counter mode	Timer A0, Timer A1, Timer A2, Timer A3, Timer A4, Timer B0, Timer B1, Timer B2

(3) STP, WIT instructions

The reset used to remove the state after executing the **STP** instruction (stop mode) or the state after executing the **WIT** instruction (wait mode) is a hardware reset. If a hardware reset is used to remove a stop mode or wait mode, the contents of the internal RAM are the contents before executing these instructions. The status of the other internal registers are the same as described in section "3.1.2 Internal status at reset". The contents of the internal RAM is not retained if a hardware reset is performed in cases other than stop mode or wait mode.
If **STP** or **WIT** instruction is to be executed after writing to internal RAM, SFR, external memory, or peripheral IC, insert **NOP** instructions in front of these instructions. Table 3 shows the number of **NOP** instructions to insert.

Table 3 Number of NOP instruction to insert

Condition	**NOP** instructions
After writing to internal RAM and SFR	1
After writing to external memory and peripheral I/C when the wait bit (bit 2 at address $5E_{16}$) is "1"	1
After writing to external memory and peripheral I/O when the wait bit is "0" (one-wait mode)	3

APPENDIX

Appendix 4. Stop, wait, one-wait, Ready, Hold state

Table 1 Stop, wait, one-wait, Ready, Hold state

Parameter \ Source	**STP** instruction (stop mode)	**WIT** instruction (wait mode)	Wait bit (one-wait mode)	$\overline{RDY}$ input (Ready state)	$\overline{HOLD}$ input (Hold state)
Enabling condition	Enabled in all operating modes	Enabled in all operating modes	Access external area with processor mode register bit 2 set to "0".	In memory expansion mode or microprocessor mode	In memory expansion mode or microprocessor mode
Oscillator	Stopped	Operating	Operating	Operating	Operating
ϕ_1 output	Stop at "L" level.	Operating	Operating	Operating	Operating
$\overline{E}$ output	Stop at "H" or "L" level.	Stop at "H" or "L" level.	"L" pulse width becomes twice at external access.	Stop at "H" or "L" level.	Stop at "H" level.
Port status	Retain bus and port status when **STP** instruction is executed.	Retain bus and port status when **WIT** instruction is executed.	———	Retain bus and port status when "L" level is applied.	Ports P0, P1, P2, $P3_0$, $P3_1$ are floating. Ports $P3_2$, $P3_3$ stop at "L" level. Ports $P4_3$–$P4_7$, P5, P6, P7, P8 retain status when "L" level is applied.
Watchdog timer state	Stopped (set "FFF_{16}" in watchdog timer and select count source f_{32})	Operating	Operating	Operating	Stopped
Removing of state	Hardware reset or accepting external interrupt	Hardware reset or accepting interrupt	Set processor mode register bit 2 to "1".	Return $\overline{RDY}$ input to "H" level.	Return $\overline{HOLD}$ input to "H" level.

Appendix 5. Package outlines

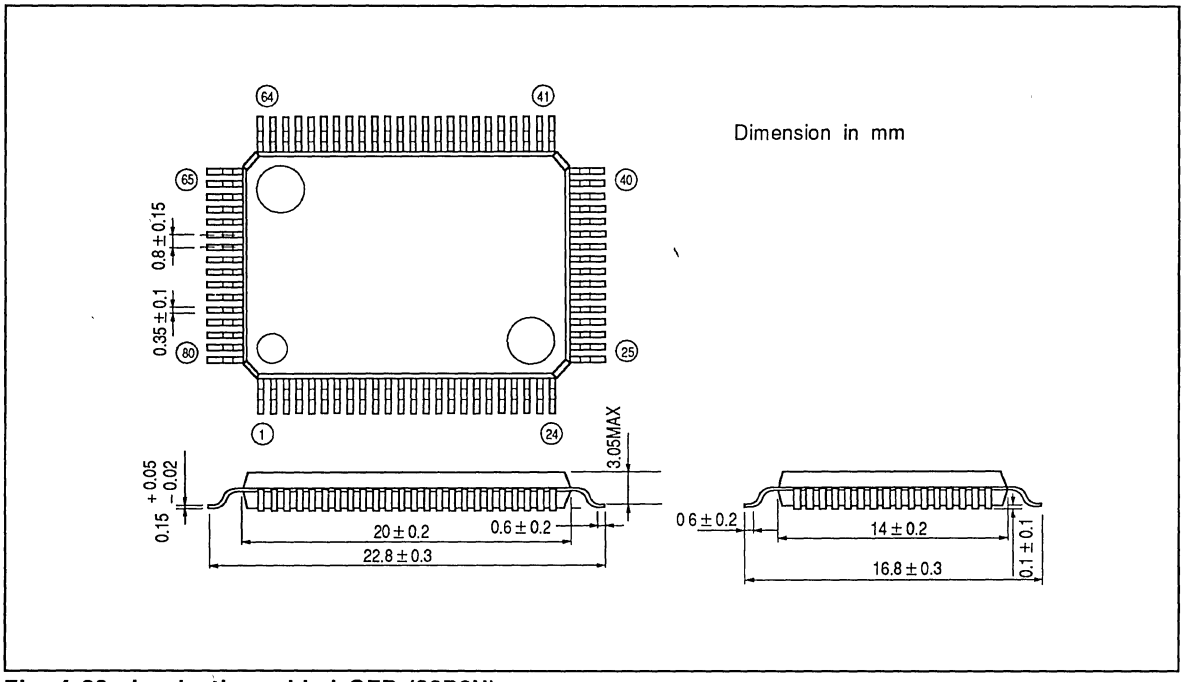


Fig. 4 80-pin plastic molded QFP (80P6N)

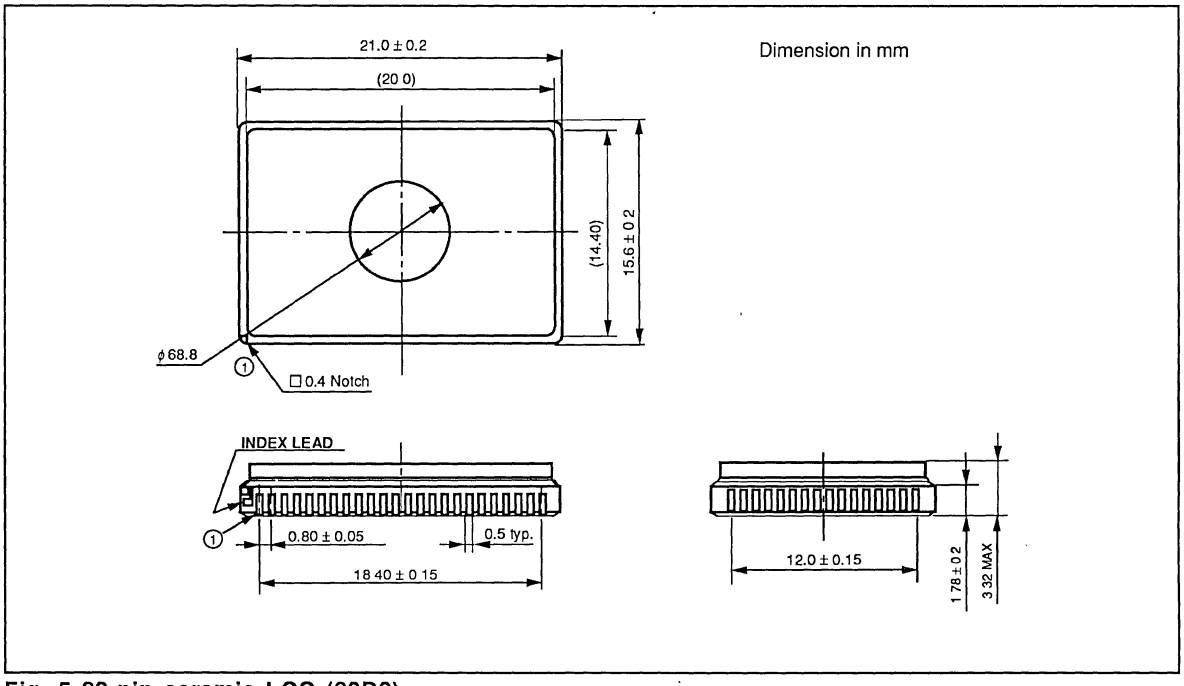


Fig. 5 80-pin ceramic LCC (80D0)

Appendix 6. Setting of unused pins

Table 4 Setting example of unused pins in single-chip mode

Pin	Setting
Ports P0–P8	Set to input mode and connect to Vss through a resistor (pull-down).
$\overline{E}$, X_{OUT} (Note 1)	Open.
AV_{CC}	Connect to V_{CC}.
AV_{SS}, V_{REF}, BYTE	Connect to V_{SS}.

Note 1 : When external clock is input to X_{IN}.

Table 5 Setting example of unused pins in memory expansion and microprocessor mode

Pin	Setting
Ports $P4_2$–$P4_7$, P5–P8	Set to input mode and connect to Vss through a resistor (pull-down).
$\overline{BHE}$ (Note 1), ALE (Note 2), $\overline{HLDA}$, X_{OUT} (Note 3)	Open.
$\overline{HOLD}$, $\overline{RDY}$	Connect to V_{CC} through a resistor (pull-up).
AV_{CC}	Connect to V_{CC}.
AV_{SS}, V_{REF}	Connect to V_{SS}.

Note 1 : When BYTE="H".
Note 2 : When BYTE="H" and address space is 64K bytes.
Note 3 : When external clock is input to X_{IN} pin.

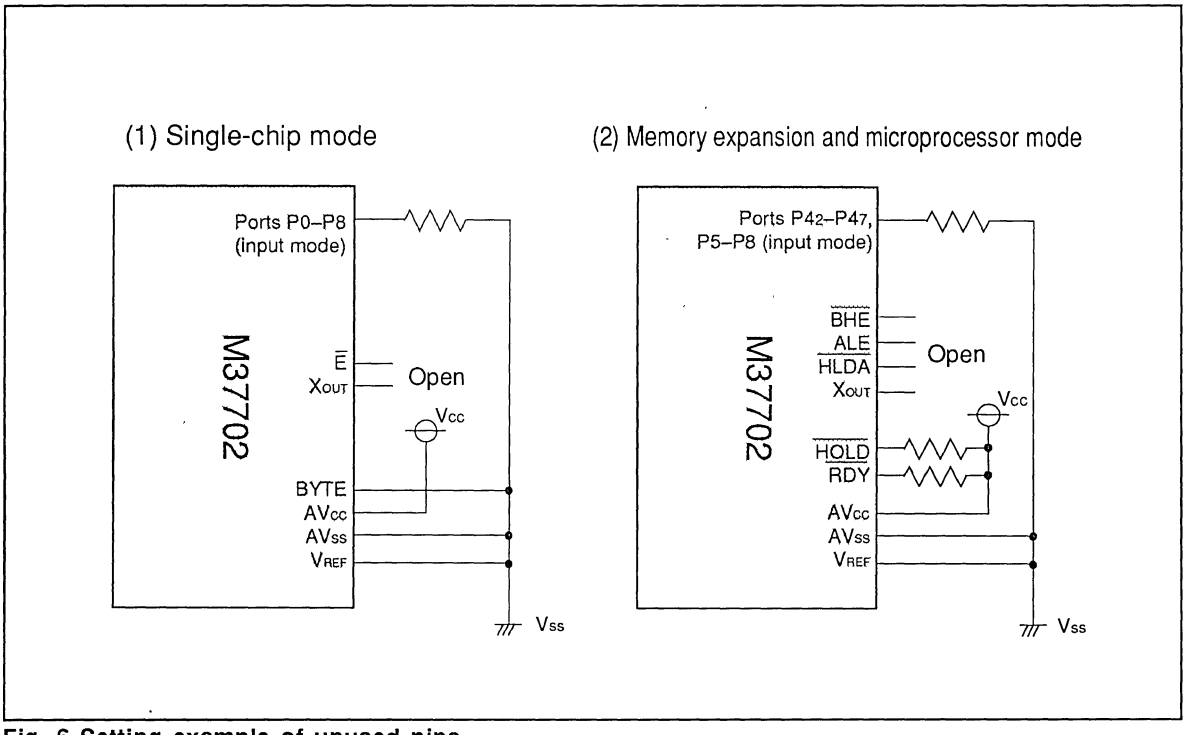


Fig. 6 Setting example of unused pins

Appendix. 7 ROM ordering method

Please submit the information described below when ordering Mask ROM and One time PROM.

[Mask ROM]

① Mask ROM Order Confirmation Form ..... 1 set
(There is a specific form to be used for each model.)
② Data to be written into mask ROM ..... EPROM
(Please provide three sets containing the identical data.)
③ Mark Specification Form (described on page 309) ..... 1 set

[One time PROM]

① Writing to PROM Order Confirmation Form ..... 1 set
(There is a specific form to be used for each model.)
② Data to be written into PROM built in ..... EPROM
(Please provide three sets containing the identical data.)
③ Mark Specification Form (described on page 309) ..... 1 set

GZZ—SH02—45A〈99A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

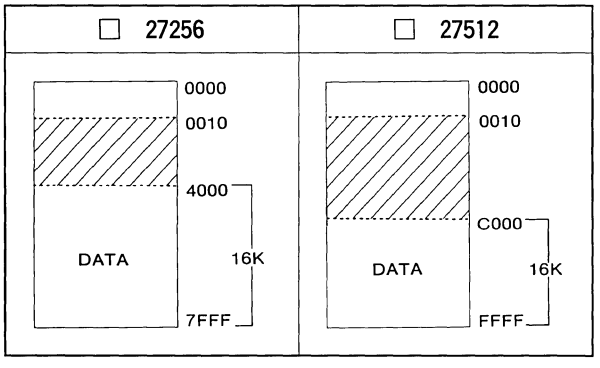
Note : Please fill in all items marked※

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

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2-XXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—46A〈99A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

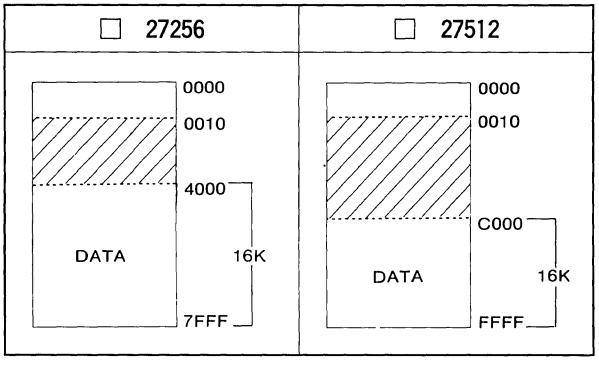
Note : Please fill in all items marked※

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

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2AXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—47A〈99A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2BXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

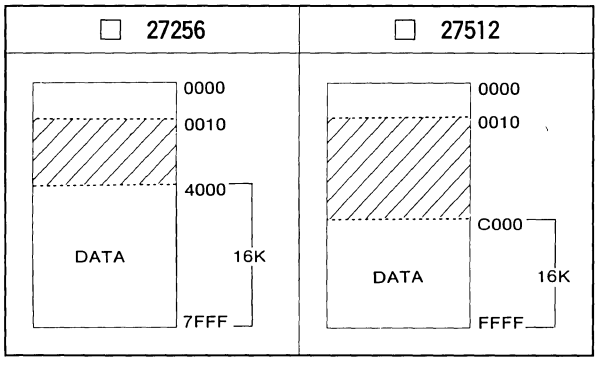
Note : Please fill in all items marked※

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

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
32	7	FF	F		

※2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M2BXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

GZZ—SH02—36A〈98A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

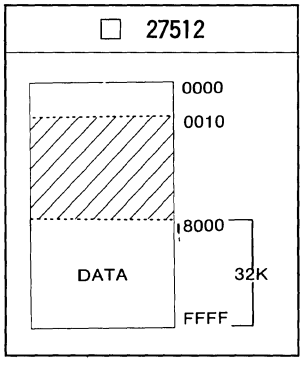
※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date :			

※1. Confirmation

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	2D	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4-XXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

APPENDIX

GZZ—SH02—37A〈98A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

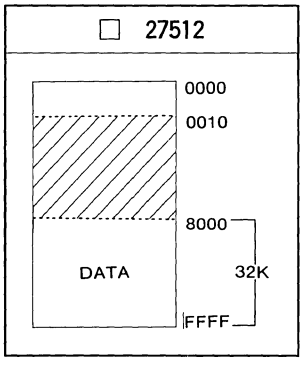
※ Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor	
	Date issued	Date :				

※1. Confirmation

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	41	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable [01_{16}] Address 10_{16}
☐ STP instruction disable [00_{16}] Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4AXXXFP) and attach to the Mask ROM Order Confirmation Form.

※4. Comments

APPENDIX

Appendix. 7 ROM ordering method

GZZ—SH02—38A〈98A0〉

Mask ROM number	

MELPS 7700 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4BXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

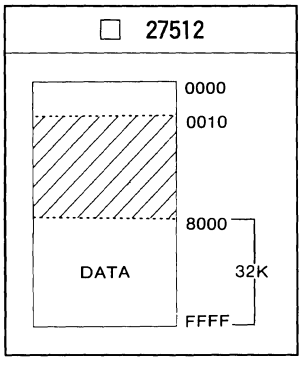
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options. This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
30	4	FF	C		
32	5	FF	D		
4D	6	FF	E		
34	7	FF	F		

※ 2. STP instruction option
One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered. Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※ 3. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702M4BXXXFP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

GZZ—SH02—54A〈99A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

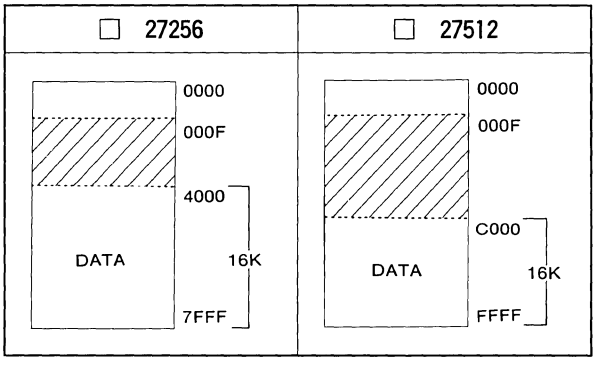
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2-XXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH02—55A〈99A0〉

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2AXXXFP
MITSUBISHI ELECTRIC

ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

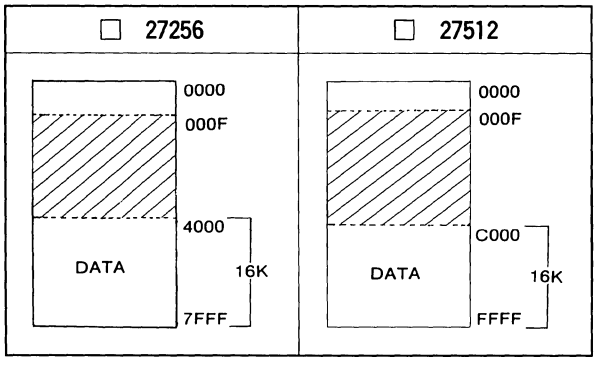
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2AXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH04—07A〈0ZA0〉

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E2BXXXFP
MITSUBISHI ELECTRIC

ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

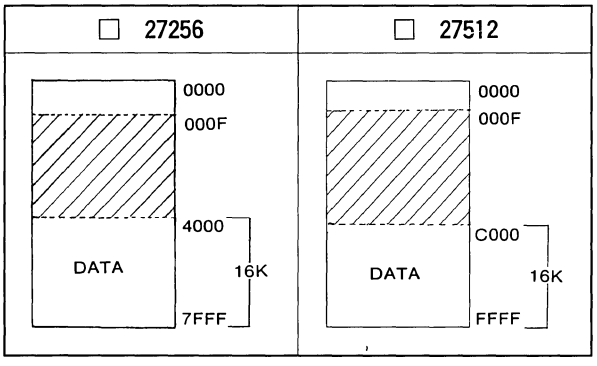
Note : Please fill in all items marked※

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

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

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

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
32	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E2BXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH03—67A〈07A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

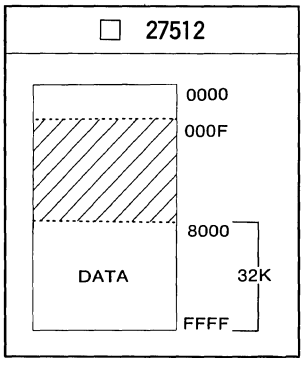
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	2D	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4-XXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH03—70A〈07A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4AXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

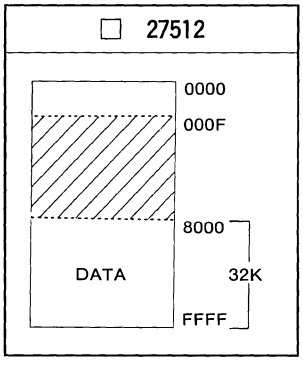
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	41	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4AXXXFP) and attach to the Writing to PROM Order Confirmation Form.

※3. Comments

GZZ—SH03—38A〈01A0〉

ROM number	

MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37702E4BXXXFP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

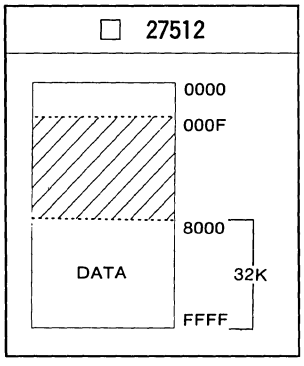
Note : Please fill in all items marked※

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

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

Checksum code for entire EPROM areas | | | | | (hexadecimal notation)

EPROM Type :



(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to $0F_{16}$ are the area for storing the data on model designation. This area must be written with the data shown below.
Address and data are written in hexadecimal notation.

	Address		Address
4D	0	42	8
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
32	5	FF	D
45	6	FF	E
34	7	FF	F

※2. Mark specification
Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37702E4BXXXFP) and attach to the Writing to PROM Order Confirmation Form.

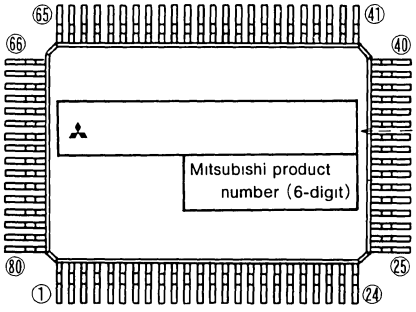
※3. Comments

80P6N (80-PIN QFP) MARK SPECIFICATION FORM

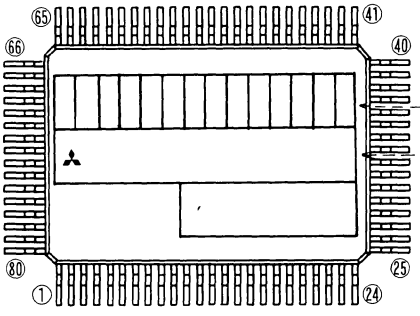
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi IC Catalog Name

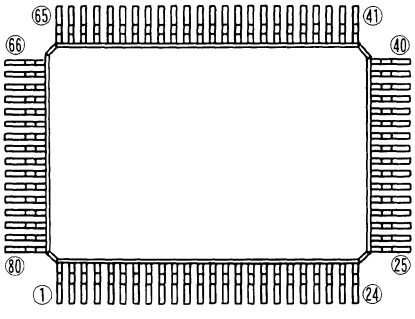


Note : The fonts and size of characters are standard Mitsubishi type.

Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 14 alphanumeric characters for capital letters, hyphens, commas, periods and so on.
4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible. Mitsubishi product number (6-digit) and Mask ROM number (3-digit) are always marked for sorting the products.
2 : If special character fonts (e.g., customer's trade mark logo) must be used in Special Mark, check the box below.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special character fonts required ☐

Appendix 8. IC socket

1. IC socket

EPROM version of M37702 group (with 80–pin ceramic LCC package) can use IC sockets described below.

●IC sockets for LCC package(80D0) provided by YAMAICHI ELECTRONICS Co., Ltd.

Type name : IC61–0804–046
Type name : IC61–0804–034 (low profile model for mounting on printed circuit board)

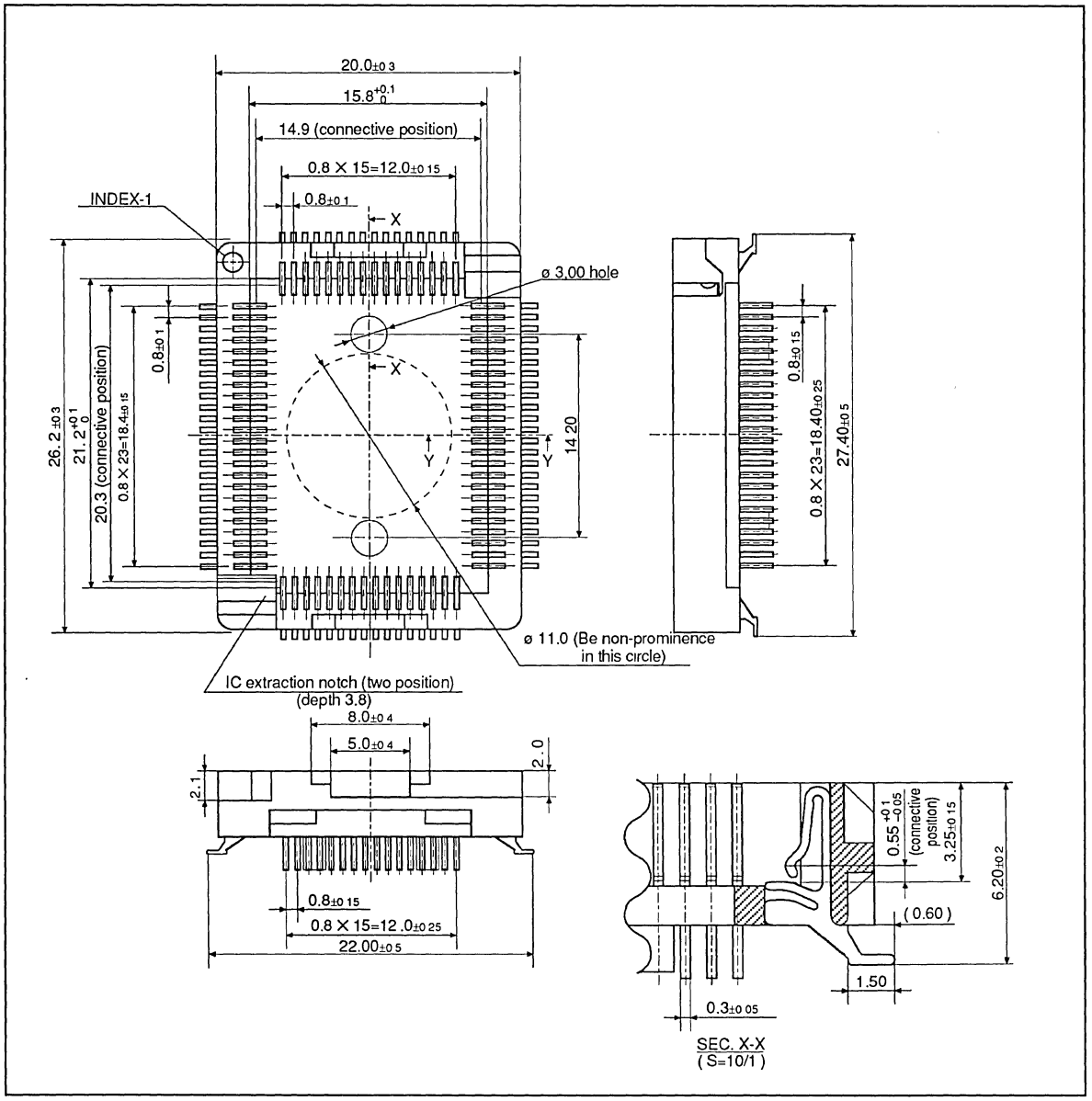


Fig. 7 IC socket (type name : IC61–0804–046) by YAMAICHI ELECTRONICS Co., Ltd. outline

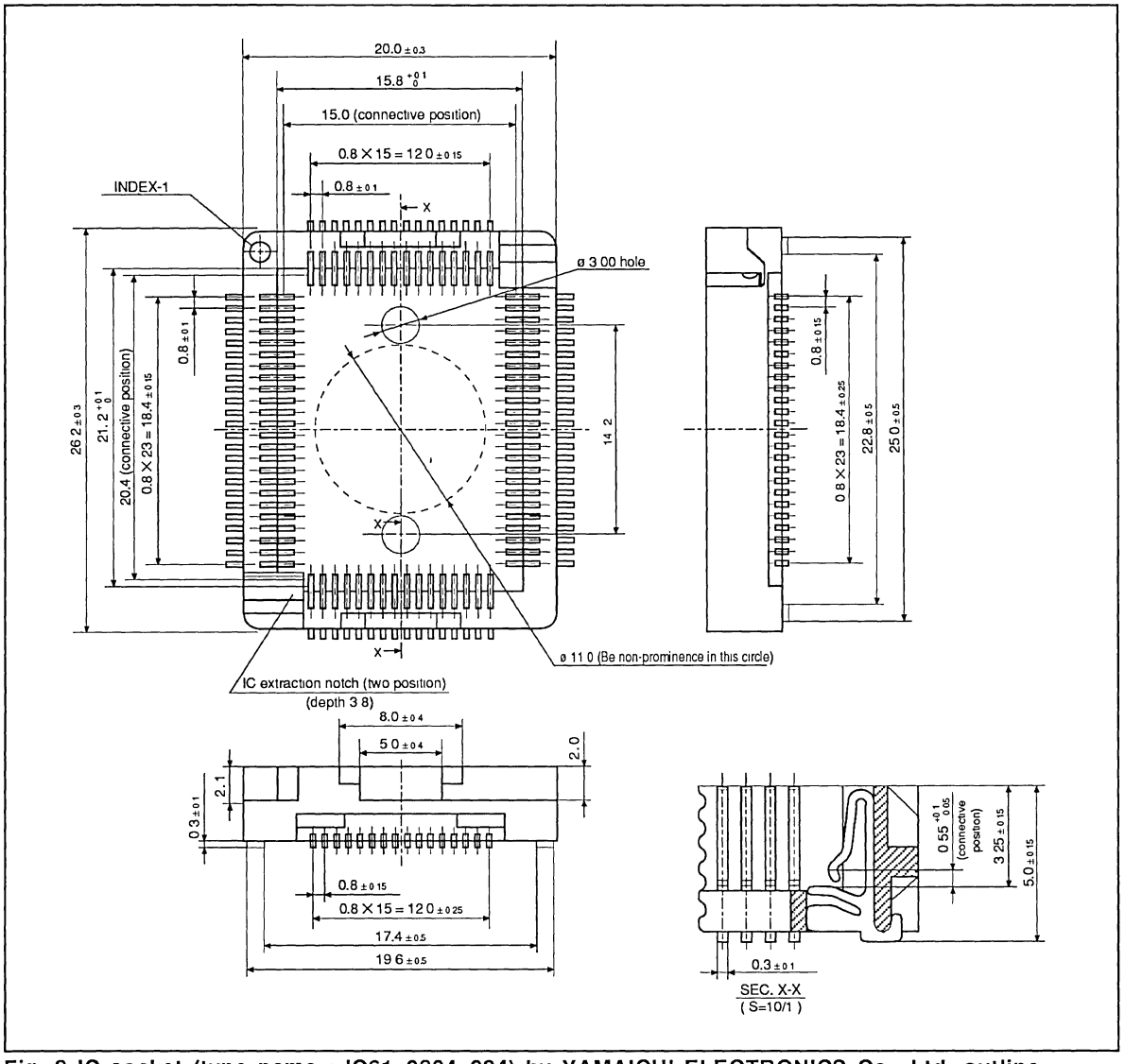


Fig. 8 IC socket (type name : IC61–0804–034) by YAMAICHI ELECTRONICS Co., Ltd. outline

[Precaution when using IC socket]

1. The profile of IC61–0804–046 is higher than IC61–0804–034. IC61–0804–046 is an exclusive use of mounting on printed circuit board and easy to solder.
2. Use IC61–0804–046 and IC61–0804–034 sockets for evaluation.
   Do not use these IC sockets mounting EPROM models for mass production.

2. Example of printed circuit board mount pad

Figure 9 shows the example of printed circuit board mount pad. This mount pad can mount 80P6N package, and IC61–0804–046 and IC61–0804–034 sockets provided by YAMAICHI ELECTRONICS Co., Ltd. which is described in "1. IC socket" .

The mount pad shown in Figure 9 is one of examples. The user should consider the mounting condition and the standard when specifying the dimension for mass production board.

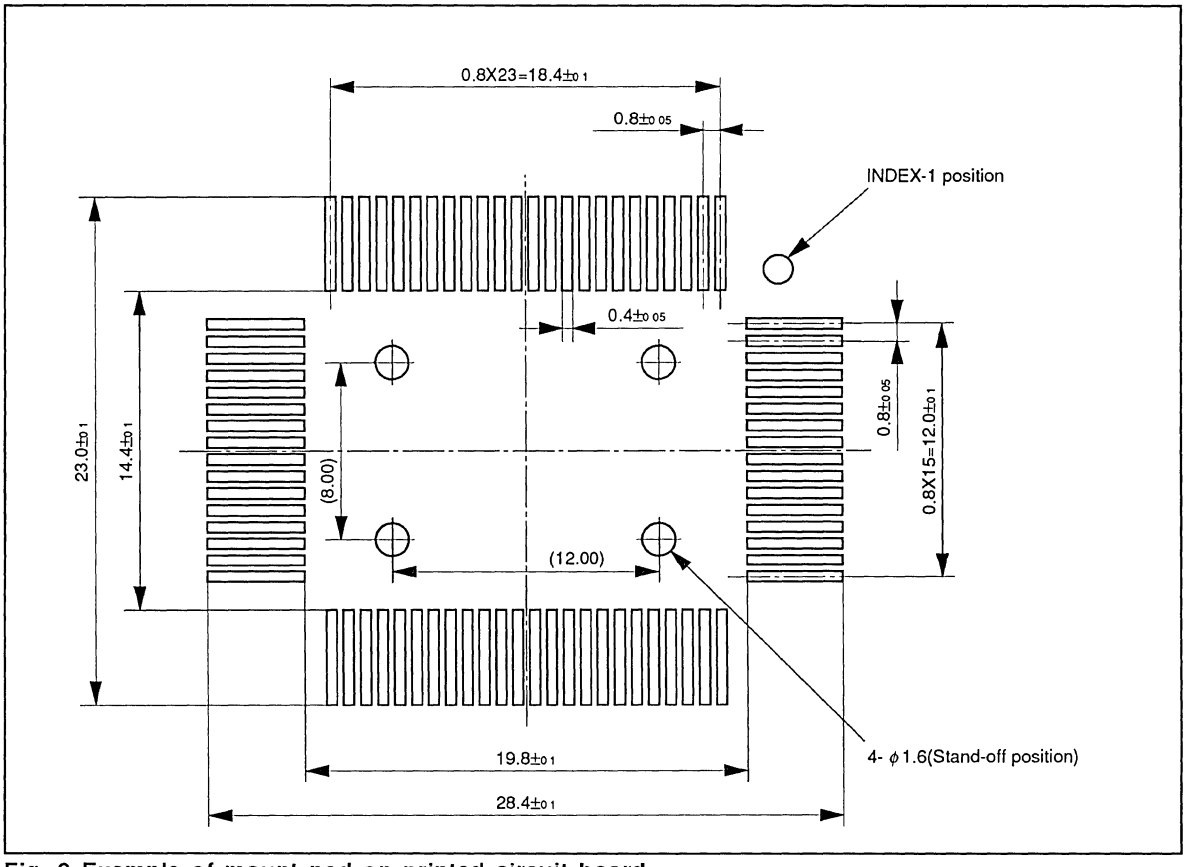


Fig. 9 Example of mount pad on printed circuit board

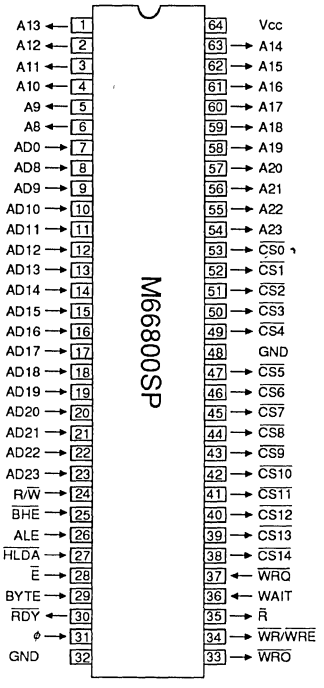
Appendix 9. M66800SP/FP

1. Overview

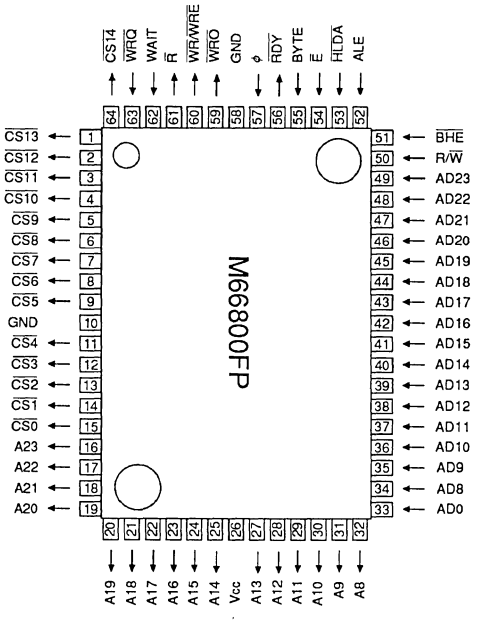
The M66800SP/FP is a 16M-byte external memory control IC for the M37702 group. It uses a MAST (Mitsubishi Advanced Schottky TTL) process to enable fast operation.

2. Features

- Supports no wait operation of the M37702 group under the following conditions:
  (1) Using 120ns (Max.) access time memory at source oscillating frequency of 12MHz
  (2) Using 100ns (Max.) access time memory at source oscillating frequency of 13MHz
- Internal read/write control circuit with independent read/write output
- Two internal octal latch circuits for address
- Internal chip selectable circuit that can fully decode 16M bytes
- Internal Ready signal generation circuit
- Pin assignment that facilitates connection to M37702 group



Outline 64P4B



Outline 64P6W

Fig. 10 M66800SP/FP pin configuration

Appendix 10. Instruction code table

INSTRUCTION CODE TABLE-1

D_7~D_4 \ D_3~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	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.

INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42_{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		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~D_4 \ D_3~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 11. Machine instructions
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	#	op	n	#
ADC (Note 1,2)	$A_{CC},C \leftarrow A_{CC}+M+C$	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	2	2				65	4	2				75	5	2				72	6	2	61	7	2	71	8	2
						42 69	4	3				42 65	6	3				42 75	7	3				42 72	8	3	42 61	9	3	42 71	10	3
AND (Note 1,2)	$A_{CC} \leftarrow A_{CC} \wedge M$	Obtains the logical product of the contents of the accumulator and the contents of the memory The result is entered into the accumulator				29	2	2				25	4	2				35	5	2				32	6	2	21	7	2	31	8	2
						42 29	4	3				42 25	6	3				42 35	7	3				42 32	8	3	42 21	9	3	42 31	10	3
ASL (Note 1)	m=0 C←[b_{15} … b_0]←0 m=1 C←[b_7 … b_0]←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	2	1	06	7	2				16	7	2												
									42 0A	4	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)	A_{CC}−M	Compares the contents of the accumulator with the contents of the memory				C9	2	2				C5	4	2				D5	5	2				D2	6	2	C1	7	2	D1	8	2
						42 C9	4	3				42 C5	6	3				42 D5	7	3				42 D2	8	3	42 C1	9	3	42 D1	10	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														

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}←A_{CC}−1 or M←M−1	Decrements the contents of the accumulator or memory by 1							1A	2	1	C6	7	2				D6	7	2												
									42 1A	4	2																					
DEX	X←X−1	Decrements the contents of the index register X by 1	CA	2	1																											
DEY	Y←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}←A_{CC}¥M	Logical exclusive sum is obtained of the contents of the accumulator and the contents of the memory The result is placed into the accumulator				49	2	2				45	4	2				55	5	2				52	6	2	41	7	2	51	8	2
						42 49	4	3				42 45	6	3				42 55	7	3				42 52	8	3	42 41	9	3	42 51	10	3
INC (Note 1)	A_{CC}←A_{CC}+1 or M←M+1	Increments the contents of the accumulator or memory by 1							3A	2	1	E6	7	2				F6	7	2												
									42 3A	4	2																					
INX	X←X+1	Increments the contents of the index register X by 1	E8	2	1																											
INY	Y←Y+1	Increments the contents of the index register Y by 1	C8	2	1																											
JMP	ABS PC_L ← AD_L PC_H ← AD_H ABL PC_L ← AD_L PC_H ← AD_H PG←AD_G (ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) L(ABS) PC_L ←(AD_H, AD_L) PC_H ←(AD_H, AD_L+1) PG ←(AD_H, AD_L+2) (ABS, X) PC_L ←(AD_H, AD_L+X) PC_H ←(AD_H, AD_L+X +1)	Places a new address into the program counter and jumps to that new address																													
JSR	ABS M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ← AD_L PC_H ← AD_H ABL M(S)← PG S←S−1 M(S)← PC_H S←S−1 M(S)←PC_L S ← S−1 PC_L ← AD_L PC_H ← AD_H PG ← AD_G (ABS, X) M(S)← PC_H S←S−1 M(S)← PC_L S←S−1 PC_L ←(AD_H, AD_L+X) PC_H ←(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																													

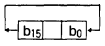
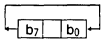
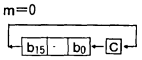
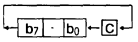
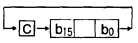
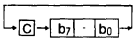
Addressing mode																																																						Processor status register										
L(DIR)			L(DIR),Y			ABS			ABS,b			ABS,X			ABS,Y			ABL			ABL,X			(ABS)			L(ABS)			(ABS,X)			STK			REL			DIR,b,R			ABS,b,R			SR			(SR),Y			BLK			10	9	8	7	6	5	4	3	2	1	0
op	n	#	op	n	#	op	n	#	op	n	#	op	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
						EC	4	3																																													•	•	•	N	•	•	•	•	•	Z	C
						CC	4	3																																													•	•	•	N	•	•	•	•	•	Z	C
						CE	7	3				DE	8	3																																							•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
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																						89 23	30	3	89 33	33	3				•	•	•	N	V	•	•	•	•	Z	C
47	10	2	57	11	2	4D	4	3				5D	6	3	59	6	3	4F	6	4	5F	7	4																						43	5	2	53	8	2				•	•	•	N	•	•	•	•	•	Z	•
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																						42 43	7	3	42 53	10	3														
						EE	7	3				FE	8	3																																							•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
																																																						•	•	•	N	•	•	•	•	•	Z	•
						4C	2	3										5C	4	4				6C	4	3	DC	8	3	7C	6	3																						•	•	•	•	•	•	•	•	•	•	•
						20	6	3										22	8	4										FC	8	3																						•	•	•	•	•	•	•	•	•	•	•

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)	$A_{CC} \leftarrow M$	Enters the contents of the memory into the accumulator				A9	2	2				A5	4	2				B5	5	2				B2	6	2	A1	7	2	B1	8	2						
						42 A9	4	3				42 A5	6	3				42 B5	7	3				42 B2	8	3	42 A1	9	3	42 B1	10	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	2	1	46	7	2				56	7	2																		
									42 4A	4	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)	$Mn+1 \leftarrow Mm+1$	Transmits the data block The transmission is done from the lower order address of the block																																			
MVP (Note 9)	$Mn-1 \leftarrow Mm-1$	Transmits the data block Transmission is done form the higher order address of the data block																																			
NOP	PC←PC+1	Advances the program counter, but performs nothing else	EA	2	1																																
ORA (Note 1,2)	$A_{CC} \leftarrow A_{CC} \vee M$	Logical sum per bit of the contents of the accumulator and the contents of the memory is obtained The result is entered into the accumulator				09	2	2				05	4	2				15	5	2				12	6	2	01	7	2	11	8	2						
						42 09	4	3				42 05	6	3				42 15	7	3				42 12	8	3	42 01	9	3	42 11	10	3						
PEA	$M(S) \leftarrow IMM_2$ $S \leftarrow S-1$ $M(S) \leftarrow IMM_1$ $S \leftarrow S-1$	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	$EAR \leftarrow PC+IMM_2,IMM_1$ $M(S) \leftarrow EAR_H$ $S \leftarrow S-1$ $M(S) \leftarrow EAR_L$ $S \leftarrow S-1$	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 $M(S) \leftarrow A_H$ $S \leftarrow S-1$ $M(S) \leftarrow A_L$ $S \leftarrow S-1$ m=1 $M(S) \leftarrow A_L$ $S \leftarrow S-1$	Saves the contents of accumulator A into the stack																																			
PHB	m=0 $M(S) \leftarrow B_H$ $S \leftarrow S-1$ $M(S) \leftarrow B_L$ $S \leftarrow S-1$ m=1 $M(S) \leftarrow B_L$ $S \leftarrow S-1$	Saves the contents of accumulator B into the stack																																			

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
A7	10	2	B7	11	2	AD	4	3				BD	6	3	B9	6	3	AF	6	4	BF	7	4																							A3	5	2	B3	8	2				•	•	•	N	•	•	•	•	•	Z	•
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																							42 A3	7	3	42 B3	10	3														
						9C	5	4				9E	6	4																																								•	•	•	•	•	•	•	•	•	•	•
																																																						•	•	•	•	•	•	•	•	•	•	•
						AE	4	3							BE	6	3																																						•	•	•	N	•	•	•	•	•	Z	•
						AC	4	3				BC	6	3																																									•	•	•	N	•	•	•	•	•	Z	•
						4E	7	3				5E	8	3																																									•	•	•	0	•	•	•	•	•	Z	C
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																							89 03	19	3	89 13	22	3				•	•	•	N	•	•	•	•	•	Z	0
																																																			54	$7+\frac{i}{2}\times 7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																			44	$9+\frac{i}{2}\times 7$	3	•	•	•	•	•	•	•	•	•	•	•
																																																							•	•	•	•	•	•	•	•	•	•	•
07	10	2	17	11	2	0D	4	3				1D	6	3	19	6	3	0F	6	4	1F	7	4																							03	5	2	13	8	2				•	•	•	N	•	•	•	•	•	Z	•
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																							42 03	7	3	42 13	10	3														
																																	F4	5	3																			•	•	•	•	•	•	•	•	•	•	•
																																	D4	5	2																			•	•	•	•	•	•	•	•	•	•	•
																																	62	5	3																			•	•	•	•	•	•	•	•	•	•	•
																																	48	4	1																			•	•	•	•	•	•	•	•	•	•	•
																																	42 48	6	2																			•	•	•	•	•	•	•	•	•	•	•

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																•	•	•	•	•	•	•	•	•	•	•
																																	DA	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	•

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  m=1 n bit rotate left 	Rotates the contents of the accumulator A, n bits to the left				89 49	6 + i	3																														
ROL (Note 1)	m=0  m=1 	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  m=1 	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) op	L(DIR) n	L(DIR) #	L(DIR),Y op	L(DIR),Y n	L(DIR),Y #	ABS op	ABS n	ABS #	ABS,b op	ABS,b n	ABS,b #	ABS,X op	ABS,X n	ABS,X #	ABS,Y op	ABS,Y n	ABS,Y #
						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
42 E7	12	3	42 F7	13	3	42 ED	6	4				42 FD	8	4	42 F9	8	4

ABL op	ABL n	ABL #	ABL,X op	ABL,X n	ABL,X #	(ABS) op	(ABS) n	(ABS) #	L(ABS) op	L(ABS) n	L(ABS) #	(ABS,X) op	(ABS,X) n	(ABS,X) #	STK op	STK n	STK #
															7A	5	1
															EB	12 + $2i_1+i_2$	2
															FB	14 + $3i_1+4i_2$	2
EF	6	4	FF	7	4												
42 EF	8	5	42 FF	9	5												

REL op	REL n	REL #	DIR,b,R op	DIR,b,R n	DIR,b,R #	ABS,b,R op	ABS,b,R n	ABS,b,R #	SR op	SR n	SR #	(SR),Y op	(SR),Y n	(SR),Y #	BLK op	BLK n	BLK #
									E3	5	2	F3	8	2			
									42 E3	7	3	42 F3	10	3			

Processor status register

10 IPL	9 IPL	8 IPL	7 N	6 V	5 m	4 x	3 D	2 I	1 Z	0 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

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	4	2				95	5	2				92	7	2	81	7	2	91	7	2			
												42 85	6	3				42 95	7	3				42 92	9	3	42 81	9	3	42 91	9	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	•

Symbols in machine instructions table

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	$\forall$	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	A_{CC}	Accumulator
DIR, b	Direct bit addressing mode	A_{CCH}	Accumulator's upper 8 bits
DIR, X	Direct indexed X addressing mode	A_{CCL}	Accumulator's lower 8 bits
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A_H	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A_L	Accumulator A's lower 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B_H	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B_L	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X_H	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X_L	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y_H	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y_L	Index register Y's lower 8 bits
(ABS)	Absolute indirect addressing mode	S	Stack pointer
L (ABS)	Absolute indirect long addressing mode	PC	Program counter
(ABS, X)	Absolute indexed X indirect addressing mode	PC_H	Program counter's upper 8 bits
STK	Stack addressing mode	PC_L	Program counter's lower 8 bits
REL	Relative addressing mode	PG	Program bank register
DIR, b, REL	Direct bit relative addressing mode	DT	Data bank register
ABS, b, REL	Absolute bit relative addressing mode	DPR	Direct page register
SR	Stack pointer relative addressing mode	DPR_H	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR_L	Direct page register's lower 8 bits
		PS	Processor status register
BLK	Block transfer addressing mode	PS_H	Processor status register's upper 8 bits
C	Carry flag	PS_L	Processor status register's lower 8 bits
Z	Zero flag	PS_b	Processor status register's b-th bit
I	Interrupt disable flag	M(S)	Contents of memory at address indicated by stack pointer
D	Decimal operation mode flag		
x	Index register length selection flag	M_b	b-th memory location
m	Data length selection flag	AD_G	Value of 24-bit address's upper 8-bit (A_{23}~A_{16})
V	Overflow flag	AD_H	Value of 24-bit address's middle 8-bit (A_{15}~A_8)
N	Negative flag	AD_L	Value of 24-bit address's lower 8-bit (A_7~A_0)
IPL	Processor interrupt priority level	op	Operation code
+	Addition	n	Number of cycle
−	Subtraction	#	Number of byte
*	Multiplication	i	Number of transfer byte or rotation
/	Division	i_1, i_2	Number of registers pushed or pulled
$\wedge$	Logical AND		
$\vee$	Logical OR		

APPENDIX

Appendix 11. Machine instructions

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≠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

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN

Semiconductor Marketing Division
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telex: 24532 MELCO J
Telephone: (03) 3218-3473
(03) 3218-3499
Facsimile: (03) 3214-5570

Overseas Marketing Manager
Kita-Itami Works
4-1, Mizuhara, Itami-shi,
Hyogo-ken 664, Japan
Telex: 526408 KMELCO J
Telephone (0727) 82-5131
Facsimile: (0727) 72-2329

HONG KONG

Mitsubishi Electric (H K) Ltd.
41st fl , Manulife Tower, 169,
Electric Road, North Point, Hong Kong
Telex 60800 MELCO HX
Telephone 510-0555
Facsimile: 510-9830, 510-9822,
510-9803

SINGAPORE

MELCO SALES SINGAPORE PTE LTD
230 Upper Bukit Timah Road # 03-01/15
Hock Soon Industrial Complex
Singapore 2158
Telex RS 20845 MELCO
Telephone 4695255
Facsimile: 4695347

TAIWAN

MELCO-TAIWAN CO., Ltd.
1st fl , Chung-Ling Bldg.,
363, Sec. 2, Fu-Hsing S Road,
Taipei R O.C
Telephone. (02) 735-3030
Facsimile (02) 735-6771
Telex. 25433 CHURYO "MELCO-TAIWAN"

U.S.A.

NORTHWEST
Mitsubishi Electronics America, Inc
1050 East Arques Avenue
Sunnyvale, CA 94086
Telephone (408) 730-5900
Facsimile (408) 730-4972

SAN DIEGO
Mitsubishi Electronics America, Inc
16980 Via Tazon, Suite 220
San Diego, CA 92128
Telephone (619) 451-9618
Facsimile (619) 592-0242

DENVER
Mitsubishi Electronics America, Inc
4600 South Ulster Street
Metropoint Building, 7th Floor
Denver, CO 80237
Telephone (303) 740-6775
Facsimile: (303) 694-0613

SOUTHWEST
Mitsubishi Electronics America, Inc.
991 Knox Street
Torrance, CA 90502
Telephone: (213) 515-3993
Facsimile: (213) 217-5781

SOUTH CENTRAL
Mitsubishi Electronics America, Inc.
1501 Luna Road, Suite 124
Carrollton, TX 75006
Telephone: (214) 484-1919
Facsimile: (214) 243-0207

NORTHERN
Mitsubishi Electronics America, Inc
15612 Highway 7 #243
Minnetonka, MN 55345
Telephone: (612) 938-7779
Facsimile (612) 938-5125

NORTH CENTRAL
Mitsubishi Electronics America, Inc
800 N Bierman Circle
Mt Prospect, IL 60056
Telephone (312) 298-9223
Facsimile: (312) 298-0567

NORTHEAST
Mitsubishi Electronics America, Inc
200 Unicorn Park Drive
Woburn, MA 01801
Telephone (617) 932-5700
Facsimile. (617) 938-1075

MID-ATLANTIC
Mitsubishi Electronics America, Inc
800 Cottontail Lane
Somerset, NJ 08873
Telephone (201) 469-8833
Facsimile. (201) 469-1909

SOUTH ATLANTIC
Mitsubishi Electronics America, Inc
2500 Gateway Center Blvd., Suite 300
Morrisville NC 27560
Telephone (404) 368-4850
Facsimile: (404) 662-5208

SOUTHEAST
Mitsubishi Electronics America, Inc.
Town Executive Center
6100 Glades Road #210
Boca Raton, FL 33433
Telephone: (407) 487-7747
Facsimile: (407) 487-2046

CANADA
Mitsubishi Electronics America, Inc
6185 Ordan Drive, Unit #110
Mississauga, Ontario, Canada L5T 2E1
Telephone: (416) 670-8711
Facsimile (416) 670-8715

Mitsubishi Electronics America, Inc
300 March Road, Suite 302
Kanata, Ontario, Canada K2K 2E2
Telephone (416) 670-8711
Facsimile: (416) 670-8715

WEST GERMANY

Mitsubishi Electric Europe GmbH
Headquarters:
Gothear Str 8
4030 Ratingen 1, West Germany
Telex 8585070 MED D
Telephone: (02102) 4860
Facsimile (02102) 486-115

Munich Office:
Arabellastraße 31
8000 Munchen 81, West Germany
Telex 5214820
Telephone. (089) 919006-09
Facsimile (089) 9101399

FRANCE

Mitsubishi Electric Europe GmbH
55, Avenue de Colmar
92563 Rueil Malmaison Cedex
Telex 632326
Telephone 47087871
Facsimile. 47513622

ITALY

Mitsubishi Electric Europe GmbH
Centro Direzionale Colleoni
Palazzo Cassiopea 1
20041 Agrate Brianza I-Milano
Telephone: (039) 636011
Facsimile: (039) 6360120

SWEDEN

Mitsubishi Electric Europe GmbH
Lastbilsvagen 6B
5-19149 Sollentuna, Sweden
Telex 10877 (meab S)
Telephone: (08) 960468
Facsimile (08) 966877

U.K.

Mitsubishi Electric (U.K.) Ltd.
Travellers Lane
Hatfield
Herts AL10 8XB, England, U K.
Telephone: (0044) 7072 76100
Facsimile (0044) 7072 78692

AUSTRALIA

Mitsubishi Electric Australia Pty Ltd.
348 Victoria Road
Rydalmere Nsw 2116, Australia
Private Bag No 2 Rydalmere Nsw 2116
Telex: MESYDAA 126614
Telephone: (02) 684-7200
Facsimile (02) 638-7072

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL M37702/M37703 Group

Dec. First Edition 1991
Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.
©1991 **MITSUBISHI ELECTRIC CORPORATION**

MITSUBISHI SEMICONDUCTORS

M37702 Group
M37703 Group

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100 TELEX: J24532 CABLE: MELCO TOKYO

If these products or technologies fall under Japanese and/or COCOM strategic restrictions, diversion contrary thereto is prohibited.

H-EA004-A Printed in Japan (ROD)
© 1991 MITSUBISHI ELECTRIC CORPORATION

New publication, effective Dec. 1991.
Specifications subject to change without notice.