

# M37732S4FP, M37732S4AFP M37732S4BFP

16-BIT CMOS MICROCOMPUTER

**DESCRIPTION**

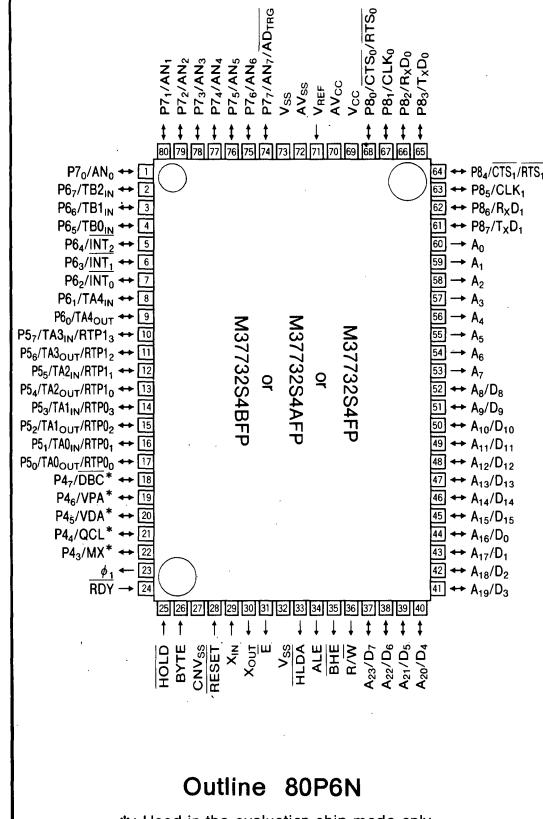
The M37732S4FP is a single-chip microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 80-pin plastic molded QFP. This single-chip microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction 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 controller that require high-speed processing of large data.

The differences between M37732S4FP, M37732S4AFP and M37732S4BFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37732S4FP unless otherwise noted.

Type name	ROM size	External clock input frequency
M37732S4FP	External	8 MHz
M37732S4AFP	External	16MHz
M37732S4BFP	External	25MHz

**FEATURES**

- Number of basic instructions ..... 103
- Memory size RAM ..... 2048 bytes
- Instruction execution time
  - M37732S4FP
    - (The fastest instruction at 8MHz frequency) ..... 500ns
  - M37732S4AFP
    - (The fastest instruction at 16MHz frequency) ..... 250ns
  - M37732S4BFP
    - (The fastest instruction at 25MHz frequency) ..... 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8MHz frequency)
  - ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) ..... 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  - (ports P4, P5, P6, P7, P8) ..... 37
- Pulse output port ..... 4-bit×2

**PIN CONFIGURATION (TOP VIEW)****Outline 80P6N**

\*: Used in the evaluation chip mode only

**APPLICATION**

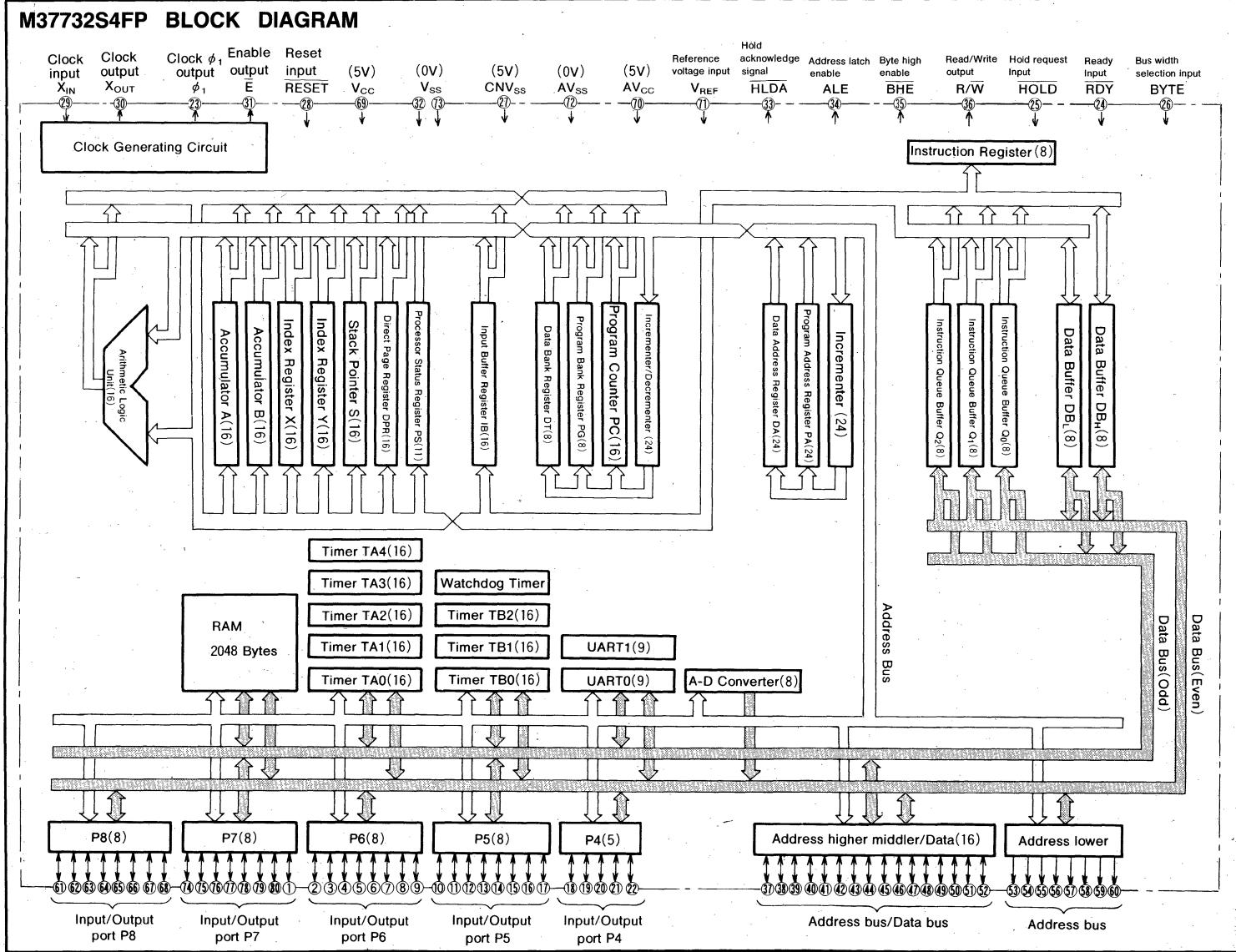
Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication and measuring instruments.

# MITSUBISHI MICROCOMPUTERS

## M37732S4FP, M37732S4A FP M37732S4BFP

### 16-BIT CMOS MICROCOMPUTER



**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****FUNCTIONS OF M37732S4FP**

Parameter	Functions
Number of basic instructions	103
Instruction execution time	M37732S4FP 500ns (the fastest instructions, at 8MHz frequency)
	M37732S4APP 250ns (the fastest instructions, at 16MHz frequency)
	M37732S4BFP 160ns (the fastest instructions, at 25MHz frequency)
Memory size	RAM 2048 bytes
Input/Output ports	P5~P8 8-bit×4
	P4 5-bit×1
Multi-function timers	TA0, TA1, TA2, TA3, TA4 16-bit×5
	TB0, TB1, TB2 16-bit×3
Serial I/O	(UART or clock synchronous serial I/O)×2
A-D converter	8-bit×1 (8 channels)
Watchdog timer	12-bit×1
Interrupts	3 external types, 16 internal types (Each interrupt can be set the priority levels to 0~7.)
Clock generating circuit	Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage	5V±10%
Power dissipation	30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage 5V
	Output current 5mA
Memory expansion	16M bytes
Operating temperature range	-20~85°C
Device structure	CMOS high-performance silicon gate process
Package	80-pin plastic molded QFP

**M37732S4FP, M37732S4A **FP  
M37732S4BFP******16-BIT CMOS MICROCOMPUTER****PIN DESCRIPTION**

Pin	Name	Input/Output	Functions
V <sub>CC</sub> , V <sub>SS</sub>	Power supply		Supply 5 V±10% to V <sub>CC</sub> and 0 V to V <sub>SS</sub> .
CNV <sub>SS</sub>	CNV <sub>SS</sub> input	Input	Connect to V <sub>CC</sub> .
RESET	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X <sub>IN</sub>	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X <sub>IN</sub> and X <sub>OUT</sub> . When an external clock is used, the clock source should be connected to the X <sub>IN</sub> pin and the X <sub>OUT</sub> pin should be left open.
X <sub>OUT</sub>	Clock output	Output	
E	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV <sub>CC</sub> , AV <sub>SS</sub>	Analog supply input		Power supply for A-D converter. Connect AV <sub>CC</sub> to V <sub>CC</sub> , and AV <sub>SS</sub> to V <sub>SS</sub> externally.
V <sub>REF</sub>	Reference voltage input	Input	This is reference voltage input pin for A-D converter.
φ <sub>1</sub>	Clock output	Output	This pin outputs the clock φ <sub>1</sub> which is divided the clock to X <sub>IN</sub> pin by 2.
RDY	Ready	Input	This is ready input pin. This is an input pin for the RDY signal. Internal clock stops while this signal is "L".
HOLD	Hold request input	Input	This is an input pin for HOLD request signal. The microcomputer enters into hold state while this signal is "L".
HLDA	Hold acknowledge output	Output	This is an output pin for HLDA signal, indicates the hold state.
R/W	Read/Write output	Output	"H" indicates the read status and "L" indicates the write status.
BHE	Byte high enable output	Output	"L" is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to retrieve only the address data from address data and data multiplex signal.
A <sub>0</sub> ~A <sub>7</sub>	Address (low-order) output	Output	Address (A <sub>7</sub> ~A <sub>0</sub> ) is output.
A <sub>8</sub> /D <sub>8</sub> ~A <sub>15</sub> /D <sub>15</sub>	Address (middle-order) output/Data (high-order) I/O	I/O	In case the BYTE pin is "L" and an external data bus is 16-bit width, high-order data (D <sub>15</sub> ~D <sub>8</sub> ) is input or output when E output is "L" and an address (A <sub>15</sub> ~A <sub>8</sub> ) is output when E output is "H". In case the BYTE pin is "H" and an external data bus is 8-bit width, only address (A <sub>15</sub> ~A <sub>8</sub> ) is output.
A <sub>16</sub> /D <sub>0</sub> ~A <sub>23</sub> /D <sub>7</sub>	Address (high-order) output/Data (low-order) I/O	I/O	Low-order data (D <sub>7</sub> ~D <sub>0</sub> ) is input or output when E output is "L", and an address (A <sub>23</sub> ~A <sub>16</sub> ) is output when E output is "H".
P4 <sub>3</sub> ~P4 <sub>7</sub>	I/O port P4	I/O	Port P4 is a 5-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset.
P5 <sub>0</sub> ~P5 <sub>7</sub>	I/O port P5	I/O	Port P5 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as I/O pins for timer A0, timer A1, timer A2 and timer A3.
P6 <sub>0</sub> ~P6 <sub>7</sub>	I/O port P6	I/O	In addition to having the same functions as port P5, these pins also function as I/O pins for timer A4, external interrupt input INT <sub>0</sub> , INT <sub>1</sub> and INT <sub>2</sub> pins, and input pin for timer B0, timer B1 and timer B2.
P7 <sub>0</sub> ~P7 <sub>7</sub>	I/O port P7	I/O	Port P7 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is input mode when reset. These pins also function as analog input AN <sub>0</sub> ~AN <sub>7</sub> input pins. P7 <sub>7</sub> also has an A-D conversion trigger input function.
P8 <sub>0</sub> ~P8 <sub>7</sub>	I/O port P8	I/O	Port P8 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as Rx <sub>D</sub> , Tx <sub>D</sub> , CLK, CTS/RTS pins for UART0 and UART1.

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### BASIC FUNCTION BLOCKS

The M37732S4FP contains the following devices on a single chip: RAM for storing instructions and data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, A-D converter, and other peripheral devices such as I/O ports. Each of these devices are described below.

### MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses  $0_{16}$  to  $FFFFF_{16}$ . The address space is divided into 64K bytes units called banks. The banks are numbered from  $0_{16}$  to  $FF_{16}$ .

Built-in RAM and control registers for built-in peripheral devices are assigned to bank  $0_{16}$ .

Addresses  $FFD6_{16}$  to  $FFFF_{16}$  are the RESET and interrupt vector addresses and contain the interrupt vectors. Use ROM for memory of this address. Refer to the section on interrupts for details.

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

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

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

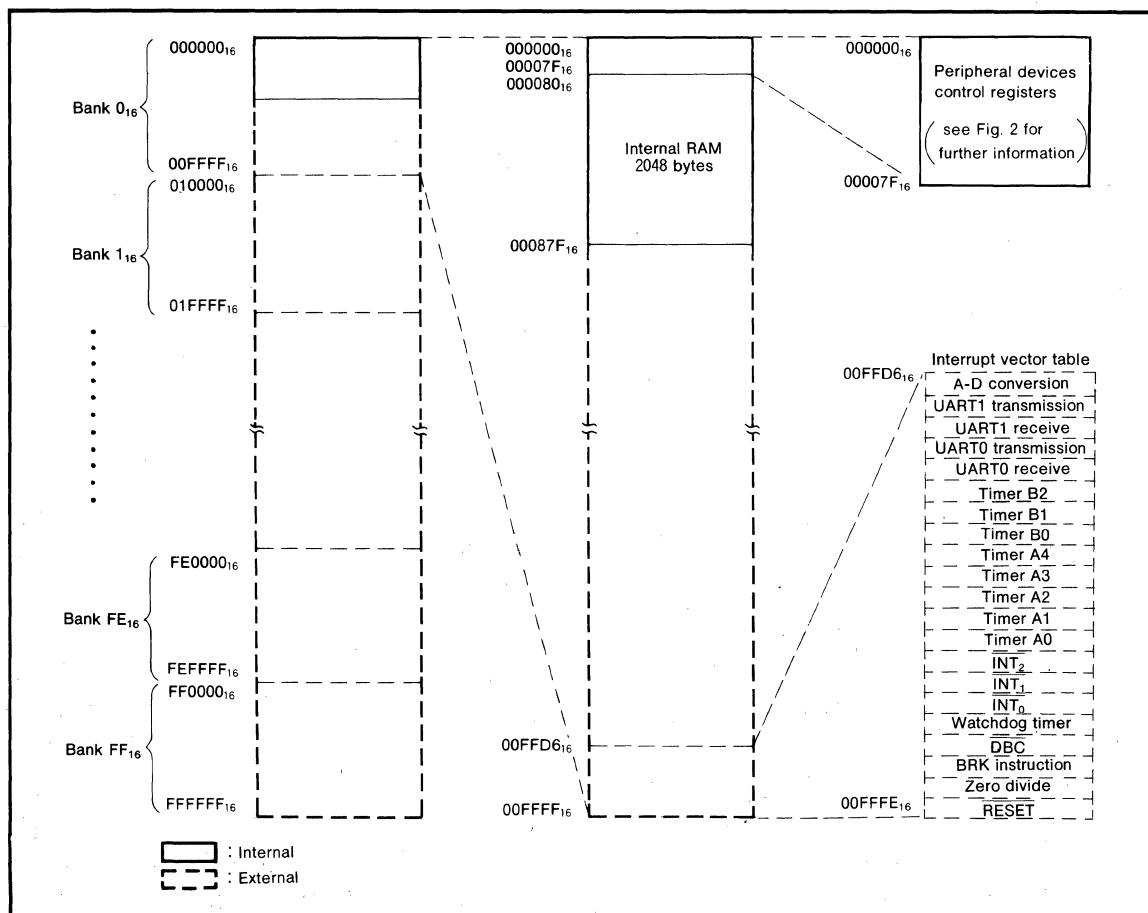


Fig. 1 Memory map

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

## Address (Hexadecimal notation)

000000	
000001	
000002	
000003	
000004	
000005	
000006	
000007	
000008	
000009	
00000A	Port P4
00000B	Port P5
00000C	Port P4 data direction register
00000D	Port P5 data direction register
00000E	Port P6
00000F	Port P7
000010	Port P6 data direction register
000011	Port P7 data direction register
000012	Port P8
000013	
000014	Port P8 data 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 bit 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 bit 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	
000047	
000048	
000049	
00004A	Timer A0
00004B	
00004C	Timer A1
00004D	
00004E	Timer A2
00004F	
000050	Timer A3
000051	
000052	Timer A4
000053	
000054	Timer B0
000055	
000056	Timer B1
000057	
000058	Timer B2
000059	
00005A	Timer A0 mode register
00005B	Timer A1 mode register
00005C	Timer A2 mode register
00005D	Timer A3 mode register
00005E	Timer A4 mode register
00005F	Timer B0 mode register
000060	Timer B1 mode register
000061	Timer B2 mode register
000062	Processor mode register
000063	
000064	Watchdog timer
000065	Watchdog timer frequency selection flag
000066	Waveform output mode register
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	
000071	A-D conversion interrupt control register
000072	UART0 transmission interrupt control register
000073	UART0 receive interrupt control register
000074	UART1 transmission interrupt control register
000075	UART1 receive interrupt control register
000076	Timer A0 interrupt control register
000077	Timer A1 interrupt control register
000078	Timer A2 interrupt control register
000079	Timer A3 interrupt control register
00007A	Timer A4 interrupt control register
00007B	Timer B0 interrupt control register
00007C	Timer B1 interrupt control register
00007D	Timer B2 interrupt control register
00007E	INT <sub>0</sub> interrupt control register
00007F	INT <sub>1</sub> interrupt control register
	INT <sub>2</sub> interrupt control register

Fig. 2. Location of peripheral devices and interrupt control registers

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### CENTRAL PROCESSING UNIT (CPU)

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

#### ACCUMULATOR A (A)

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

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

#### ACCUMULATOR B (B)

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

#### INDEX REGISTER X (X)

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

In index addressing mode, register X is used as the index register and the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction MVP or MVN, the contents of index register X indicates the low-order 16 bits of the source data address. The third byte of the MVP and MVN is the high-order 8 bits of the source data address.

#### INDEX REGISTER Y (Y)

Index register Y 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 16-bit register or as 8-bit register. It is used as a 16-bit register when flag x is "0" and as an 8-bit register when flag x is "1". Flag x is a part of the processor status register (PS) which is described later.

In index addressing mode, register Y is used as the index register and the contents of this address is added to obtain the real address.

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 address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

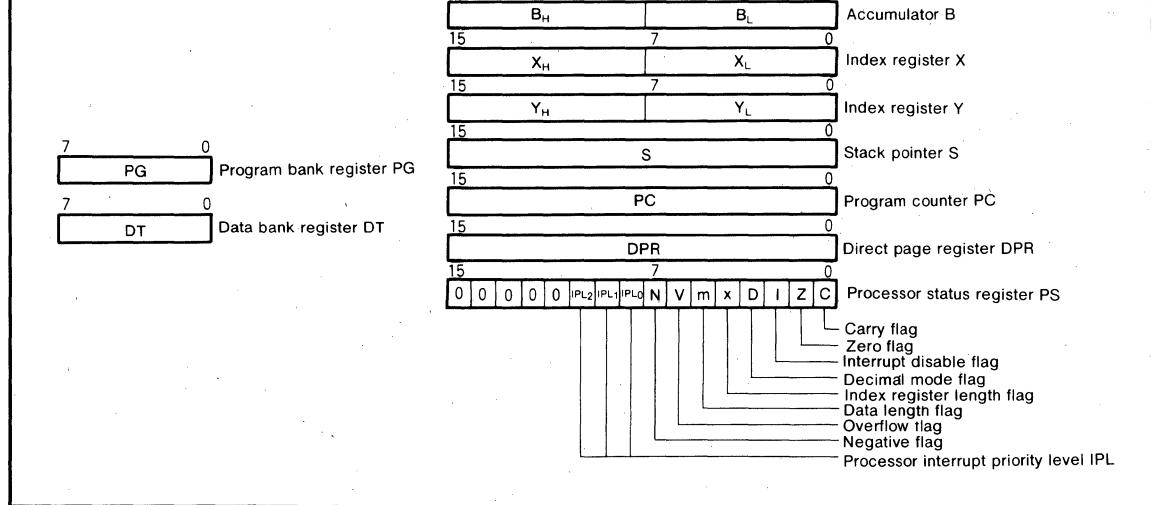


Fig. 3 Register structure

**STACK POINTER (S)**

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

**PROGRAM COUNTER (PC)**

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

**PROGRAM BANK REGISTER (PG)**

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

**DATA BANK REGISTER (DT)**

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

**DIRECT PAGE REGISTER (DPR)**

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

**PROCESSOR STATUS REGISTER (PS)**

Processor status register (PS) is an 11-bit register. It consists of a flag to indicate the result of operation and CPU interrupt levels.

Branch operations can be performed by testing the flags C, Z, V, and N.

The details of each processor status register bit are described below.

**1. Carry flag (C)**

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

**2. Zero flag (Z)**

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

**3. Interrupt disable flag (I)**

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

**4. Decimal mode flag (D)**

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

## 5. Index register length flag (x)

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

#### **6. Data length flag (m)**

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

## 7. Overflow flag (V)

The overflow flag has meaning when addition or subtraction is performed a word as signed binary number. When the data length flag m is "0", the overflow flag is set when the result of addition or subtraction is outside the range between -32768 and +32767. When the data length flag m is "1", the overflow flag is set when the result of addition or subtraction is outside the range between -128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

#### **8. Negative flag (N)**

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

## 9. Processor interrupt priority level (IPL)

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

## **BUS INTERFACE UNIT**

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency  $f_{(X_{IN})}$  by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and prefetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

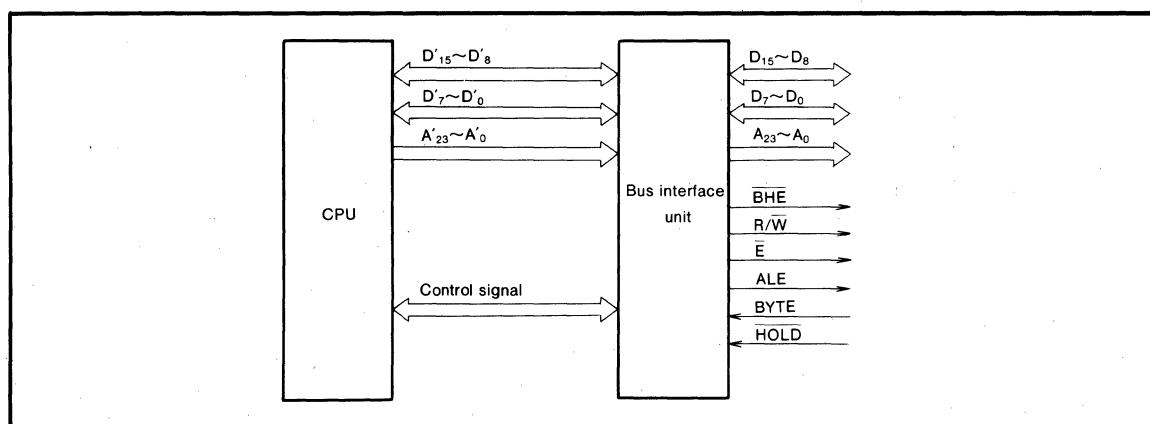


Fig. 4 Relationship between the CPU and the bus interface unit

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

The bus interface unit operates using one of the waveforms (1) to (6) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

The  $\bar{E}$  signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/W signal. Read is performed when the R/W signal is "H" state and write is performed when it is "L" state.

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

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

The signals  $A_0$  and  $\overline{BHE}$  in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses. The  $A_0$  signal that is the address bit 0 is "L" when an even number address is accessed. The  $\overline{BHE}$  signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register (address  $5E_{16}$ ) is the wait bit. When this bit is set to "0", the "L" width of  $\bar{E}$  signal is 2 times as long when accessing an external memory area. However, the "L" width of  $\bar{E}$  signal is not extended when an internal memory area is accessed. When the wait bit is "1", the "L" width of  $\bar{E}$  signal is not extended for any access. Waveform (3) is an expansion of the "L" width of  $\bar{E}$  signal in waveform (1). Waveform (4), (5), and (6) are expansion of each "L" width of  $\bar{E}$  signal in waveform (2), first half of waveform (2), and the last half of waveform (2) respectively.

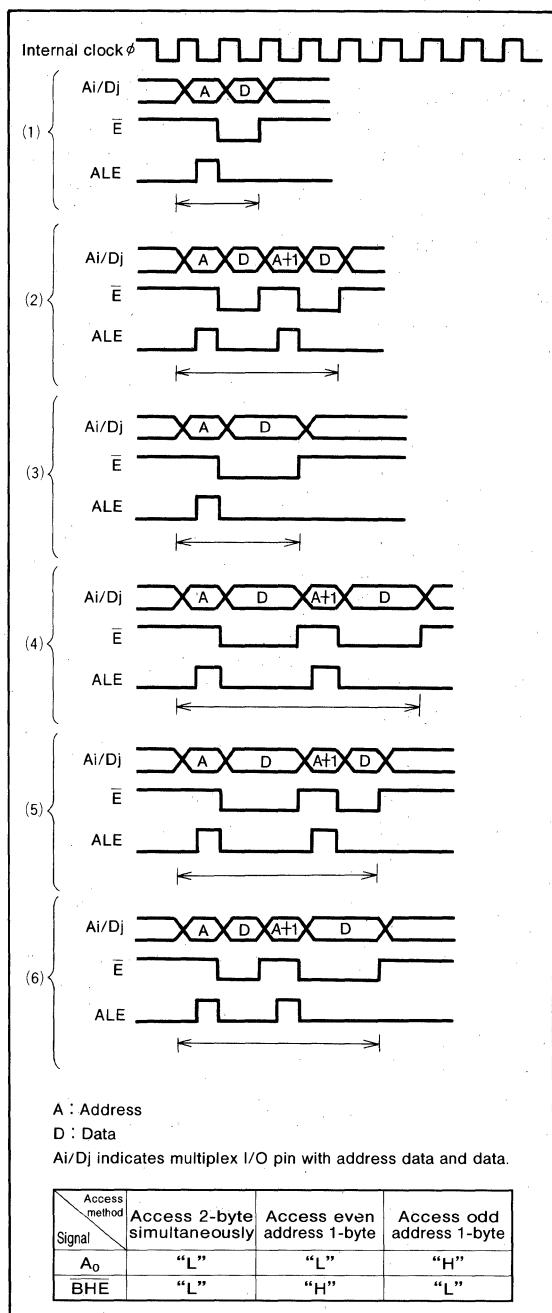


Fig. 5 Relationship between access method and signals  $A_0$  and  $\overline{BHE}$

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

Instruction code read, data read, and data write are described below.

Instruction code read will be described first.

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

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

This is referred to as instruction pre-fetching.

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

However, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1) or (3) in Figure 5 is used for instruction code read.

Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (6) in Figure 5 to perform the operation.

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

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

# M37732S4FP, M37732S4A<sup>F</sup>P M37732S4B<sup>F</sup>P

## 16-BIT CMOS MICROCOMPUTER

### INTERRUPTS

Table 1 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

DBC is an interrupt used during debugging.

Interrupts other than reset, DBC, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

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

INT<sub>2</sub> to INT<sub>0</sub> are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

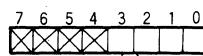
Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

reset > DBC > watchdog timer > other interrupts

Table 1. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses
A-D conversion	00FFD6 <sub>16</sub> 00FFD7 <sub>16</sub>
UART1 transmit	00FFD8 <sub>16</sub> 00FFD9 <sub>16</sub>
UART1 receive	00FFDA <sub>16</sub> 00FFDB <sub>16</sub>
UART0 transmit	00FFDC <sub>16</sub> 00FFDD <sub>16</sub>
UART0 receive	00FFDE <sub>16</sub> 00FFDF <sub>16</sub>
Timer B2	00FFE0 <sub>16</sub> 00FFE1 <sub>16</sub>
Timer B1	00FFE2 <sub>16</sub> 00FFE3 <sub>16</sub>
Timer B0	00FFE4 <sub>16</sub> 00FFE5 <sub>16</sub>
Timer A4	00FFE6 <sub>16</sub> 00FFE7 <sub>16</sub>
Timer A3	00FFE8 <sub>16</sub> 00FFE9 <sub>16</sub>
Timer A2	00FFEA <sub>16</sub> 00FEB <sub>16</sub>
Timer A1	00FFEC <sub>16</sub> 00FFED <sub>16</sub>
Timer A0	00FFEE <sub>16</sub> 00FFEF <sub>16</sub>
<u>INT</u> <sub>2</sub> external interrupt	00FFF0 <sub>16</sub> 00FFF1 <sub>16</sub>
<u>INT</u> <sub>1</sub> external interrupt	00FFF2 <sub>16</sub> 00FFF3 <sub>16</sub>
<u>INT</u> <sub>0</sub> external interrupt	00FFF4 <sub>16</sub> 00FFF5 <sub>16</sub>
Watchdog timer	00FFF6 <sub>16</sub> 00FFF7 <sub>16</sub>
<u>DBC</u> (unusable)	00FFF8 <sub>16</sub> 00FFF9 <sub>16</sub>
Break instruction	00FFFA <sub>16</sub> 00FFFB <sub>16</sub>
Zero divide	00FFFC <sub>16</sub> 00FFFD <sub>16</sub>
Reset	00FFFE <sub>16</sub> 00FFFF <sub>16</sub>



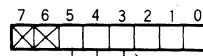
Interrupt priority

Interrupt request bit

0 : No interrupt

1 : Interrupt

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



Interrupt priority

Interrupt request bit

0 : No interrupt

1 : Interrupt

Polarity selection bit

0 : Set interrupt request bit at "H" level for level sense and when changing from "H" to "L" level for edge sense.

1 : Set interrupt request bit at "L" level for level sense and when changing from "L" to "H" level for edge sense.

Level sense/edge sense selection bit

0 : Edge sense

1 : Level sense

Interrupt control register configuration for INT<sub>2</sub>~INT<sub>0</sub>.

Fig. 6 Interrupt control register configuration

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

Table 2. Addresses of interrupt control registers

Interrupt control registers	Addresses
A-D conversion interrupt control register	000070 <sub>16</sub>
UART0 transmit interrupt control register	000071 <sub>16</sub>
UART0 receive interrupt control register	000072 <sub>16</sub>
UART1 transmit interrupt control register	000073 <sub>16</sub>
UART1 receive interrupt control register	000074 <sub>16</sub>
Timer A0 interrupt control register	000075 <sub>16</sub>
Timer A1 interrupt control register	000076 <sub>16</sub>
Timer A2 interrupt control register	000077 <sub>16</sub>
Timer A3 interrupt control register	000078 <sub>16</sub>
Timer A4 interrupt control register	000079 <sub>16</sub>
Timer B0 interrupt control register	00007A <sub>16</sub>
Timer B1 interrupt control register	00007B <sub>16</sub>
Timer B2 interrupt control register	00007C <sub>16</sub>
INT <sub>0</sub> interrupt control register	00007D <sub>16</sub>
INT <sub>1</sub> interrupt control register	00007E <sub>16</sub>
INT <sub>2</sub> interrupt control register	00007F <sub>16</sub>

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

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

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

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

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

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

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

For reset, DBC, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

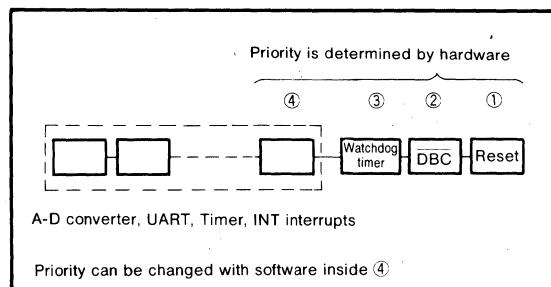


Fig. 7 Interrupt priority

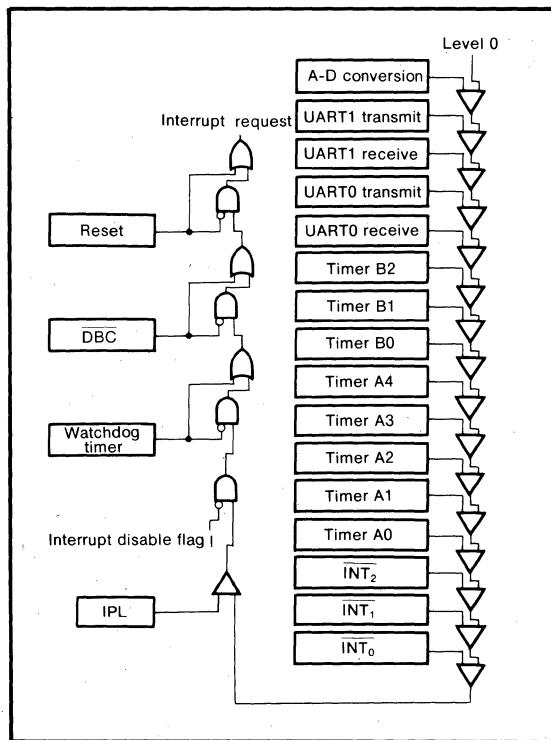


Fig. 8 Interrupt priority resolution

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

As shown in Figure 9, there are three different interrupt priority resolution time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.

The time is selected with bits 4 and 5 of the processor mode register (address 5E<sub>16</sub>) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00<sub>16</sub>" and therefore, the longest time is selected.

However, the shortest time may be selected by software.

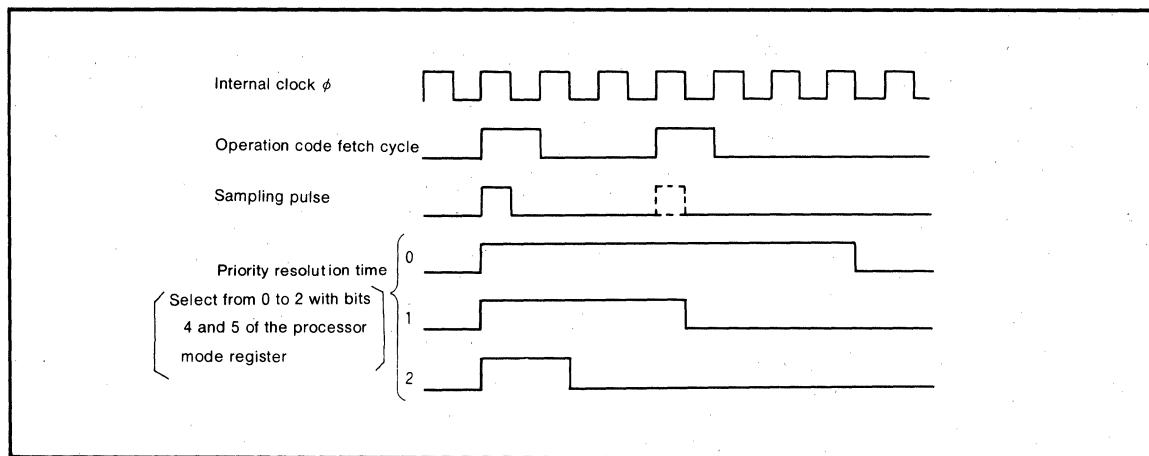
**Table 3. Value set in processor interrupt level (IPL) during an interrupt**

Interrupt types	Setting value
Reset	0
DBC	7
Watchdog timer	7
Zero divide	Not change value of IPL
BRK instruction	Not change value of IPL

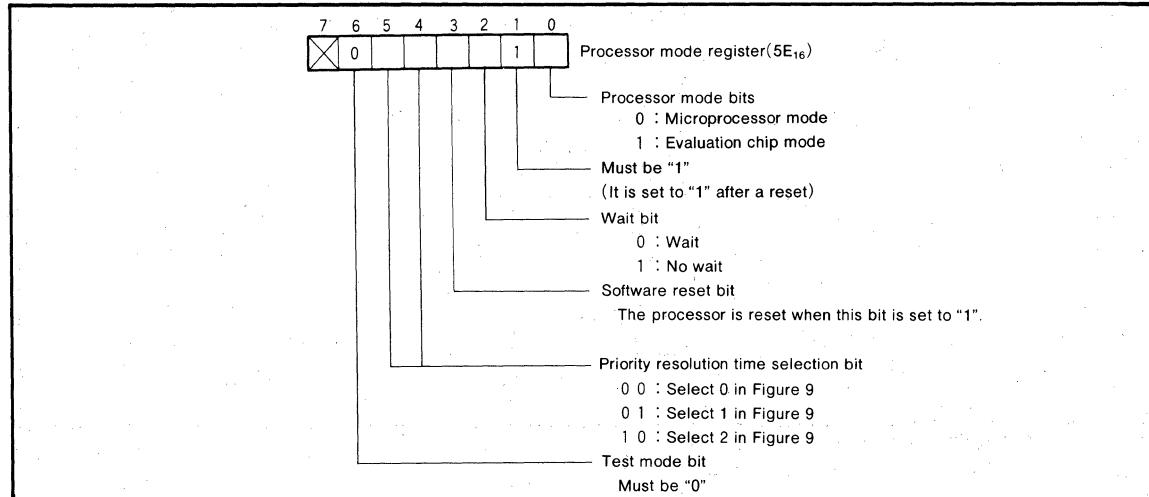
**Table 4. Relationship between priority level evaluation time selection bit and number of cycles**

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of $\phi$
0	1	4 cycles of $\phi$
1	0	2 cycles of $\phi$

$\phi$  : internal clock



**Fig. 9 Interrupt priority resolution time**



**Fig. 10 Processor mode register configuration**

## TIMER

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

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

## TIMER A

Figure 11 shows a block diagram of timer A.

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

### (1) Timer mode [00]

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

Bits 6 and 7 are used to select the timer counter source. The counting of the selected clock starts when the count start flag is "1" and stops when it is "0".

Figure 13 shows the bit configuration of the count start flag. The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes  $0000_{16}$ . At the same time, the contents of the reload register is transferred to the counter and count is continued.

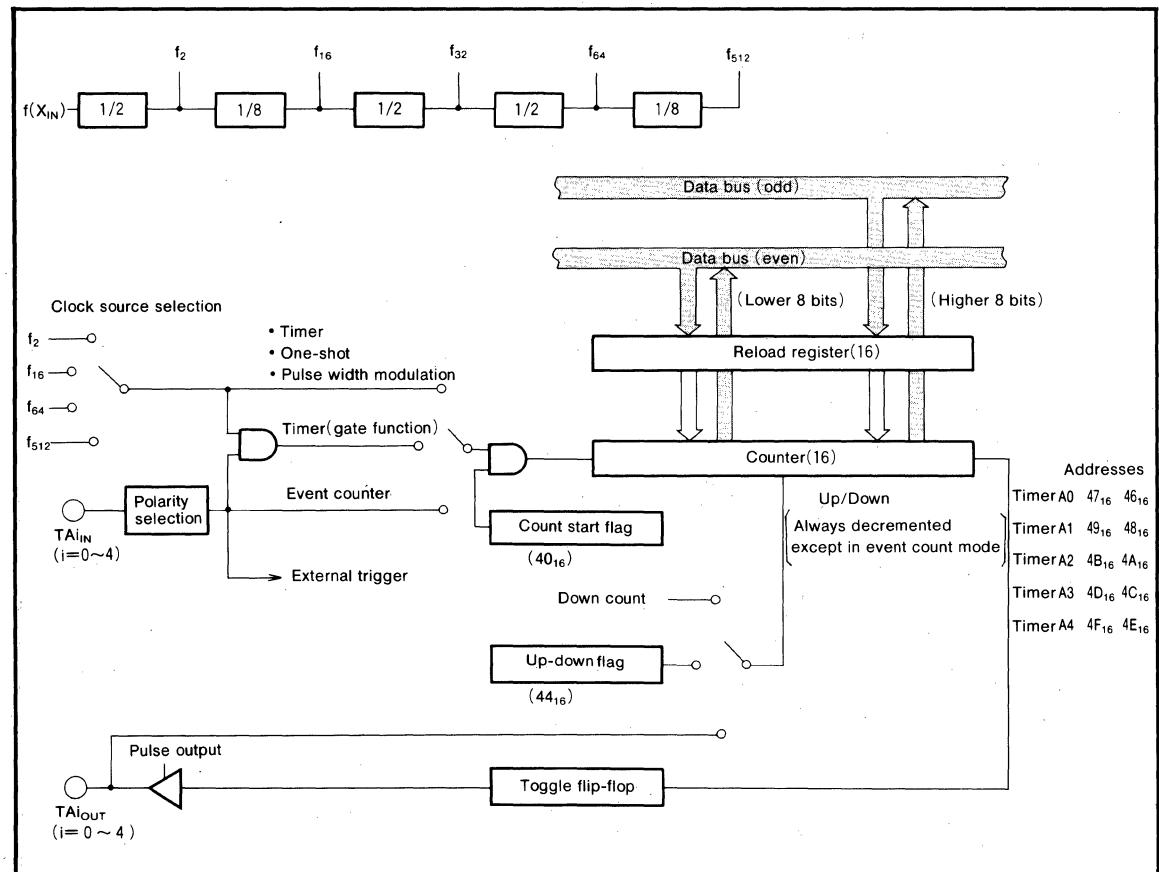


Fig. 11 Block diagram of timer A

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

When bit 2 of the timer Ai mode register is "1", the output is generated from TAi<sub>OUT</sub> pin. The output is toggled each time the contents of the counter reaches to 0000<sub>16</sub>. When the contents of the count start flag is "0", "L" is output from TAi<sub>OUT</sub> pin.

When bit 2 is "0", TAi<sub>OUT</sub> can be used as a normal port pin.

When bit 4 is "0", TAi<sub>IN</sub> can be used as a normal port pin.

When bit 4 is "1", counting is performed only while the input signal from the TAi<sub>IN</sub> pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the TAi<sub>IN</sub> input signal. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAi<sub>IN</sub> pin input

signal is "H" and if bit 3 is "0", counting is performed while it is "L".

Note that the duration of "H" or "L" on the TAi<sub>IN</sub> pin must be two or more cycles of the timer count source.

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

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

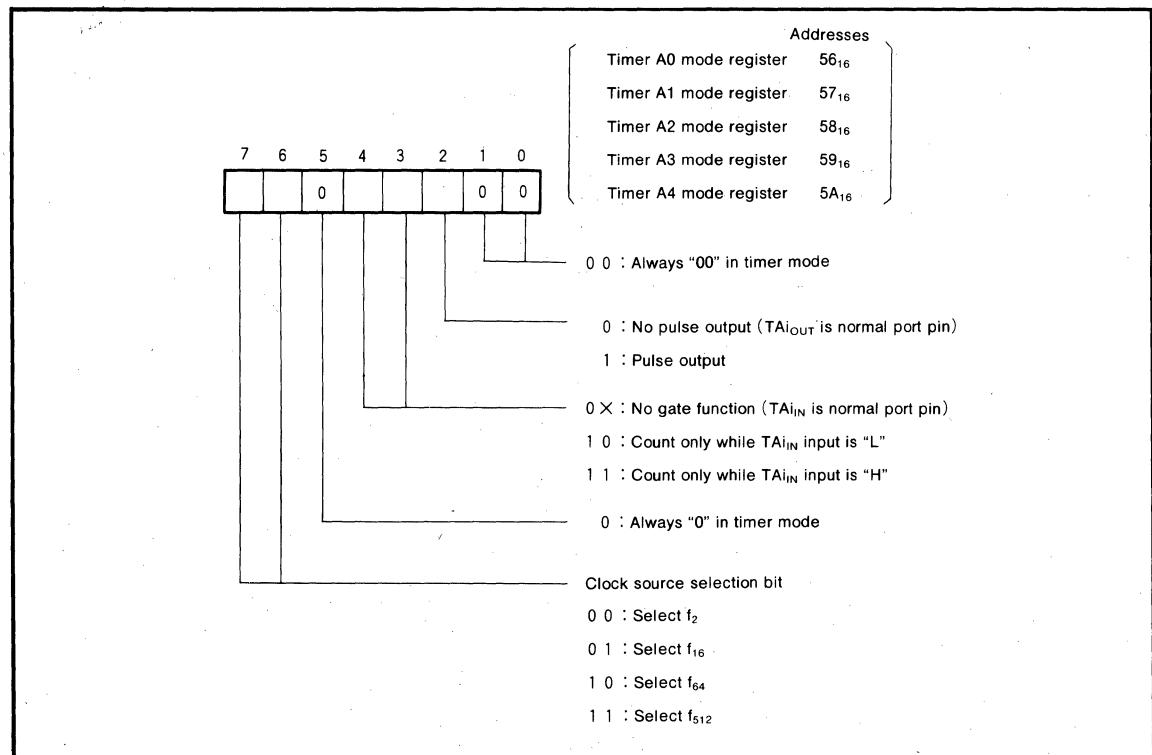


Fig. 12 Timer Ai mode register bit configuration during timer mode

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

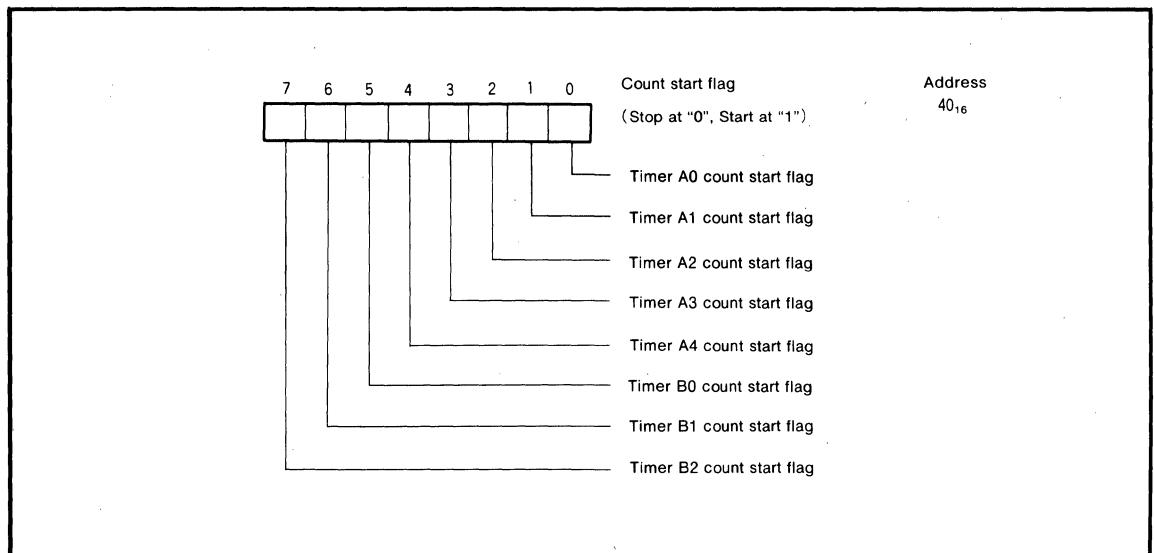


Fig. 13 Count start flag bit configuration

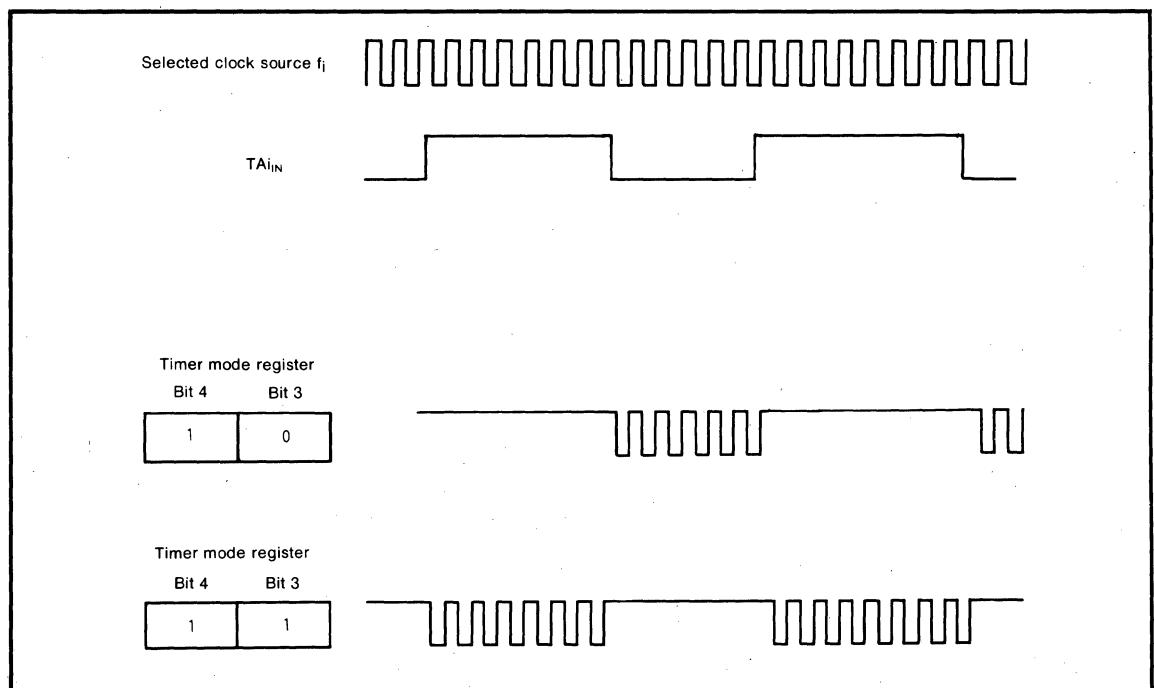


Fig. 14 Count waveform when gate function is available

# M37732S4FP, M37732S4A FP M37732S4B FP

## 16-BIT CMOS MICROCOMPUTER

### (2) Event counter mode [01]

Figure 15 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, the bit 0 of the timer Ai mode register must be "1" and bit 1 and 5 must be "0".

The input signal from the TAi<sub>IN</sub> pin is counted when the count start flag shown in Figure 13 is "1" and counting is stopped when it is "0".

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

In event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAi<sub>OUT</sub> pin.

When bit 4 of the timer Ai mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAi<sub>OUT</sub> pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be "0" if bit 4 is "1" because if bit 2 is "1", TAi<sub>OUT</sub> pin becomes an output pin with pulse output.

The count is decremented when the input signal from the TAi<sub>OUT</sub> pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAi<sub>OUT</sub> pin before valid edge is input to the TAi<sub>IN</sub> pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000<sub>16</sub> (decrement count) or FFFF<sub>16</sub> (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is "1" and the counter reaches 0000<sub>16</sub> (decrement count) or FFFF<sub>16</sub> (increment count), the waveform reversing polarity is output from TAi<sub>OUT</sub> pin.

If bit 2 is "0", TAi<sub>OUT</sub> pin can be used as a normal port pin. However, if bit 4 is "1" and the TAi<sub>OUT</sub> pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 should be "0" unless the output from the TAi<sub>OUT</sub> pin is to be used to select the count direction.

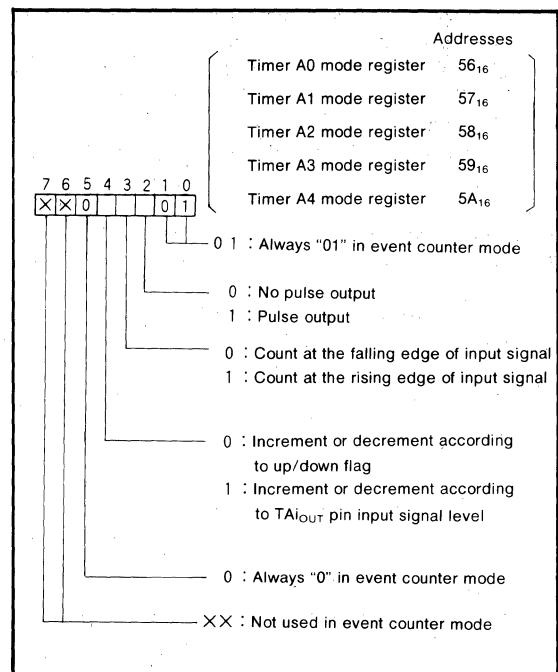


Fig. 15 Timer Ai mode register bit configuration during event counter mode

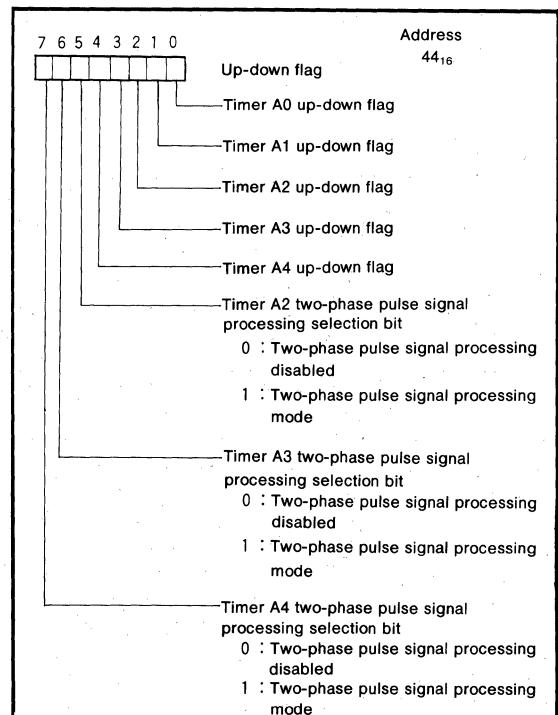


Fig. 16 Up-down flag bit configuration

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

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

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A<sub>2</sub>, A<sub>3</sub>, or A<sub>4</sub>. There are two types of two-phase pulse processing operations. One uses timers A<sub>2</sub> and A<sub>3</sub>, and the other uses timer A<sub>4</sub>. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAK<sub>j</sub><sub>OUT</sub> ( $j=2$  to 4) pin and TAK<sub>j</sub><sub>IN</sub> pin.

When timers A<sub>2</sub> and A<sub>3</sub> are used, as shown in Figure 17, the count is incremented when a rising edge is input to the TAK<sub>k</sub><sub>IN</sub> pin after the level of TAK<sub>k</sub><sub>OUT</sub> ( $k=2, 3$ ) pin changes from "L" to "H", and when the falling edge is inserted, the count is decremented.

For timer A<sub>4</sub>, as shown in Figure 18, when a phase related pulse with a rising edge input to the TA4<sub>IN</sub> pin is input after the level of TA4<sub>OUT</sub> pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the TA4<sub>OUT</sub> pin and TA4<sub>IN</sub> pin.

When a phase related pulse with a falling edge input to the TA4<sub>OUT</sub> pin is input after the level of TA4<sub>IN</sub> pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the TA4<sub>IN</sub> pin and TA4<sub>OUT</sub> pin. When performing this two-phase pulse signal proce-

sing, timer A<sub>j</sub> mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0". Bits 6 and 7 are ignored. Note that bits 5, 6, and 7 of the up-down flag register (44<sub>16</sub>) are the two-phase pulse signal processing selection bit for timer A<sub>2</sub>, A<sub>3</sub>, and A<sub>4</sub> respectively. Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

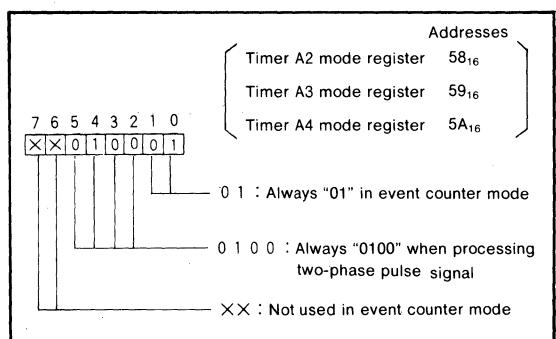


Fig. 19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

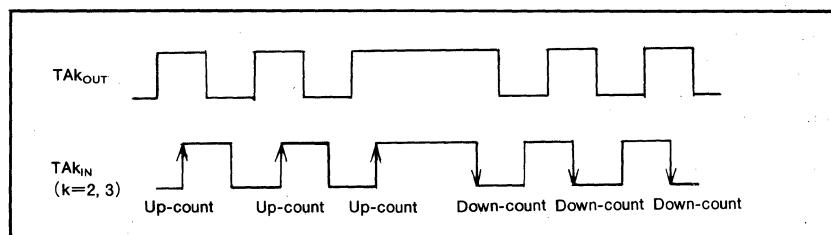


Fig. 17 Two-phase pulse processing operation of timer A2 and timer A3

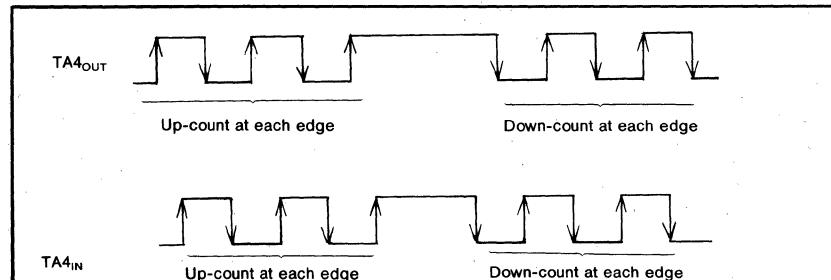


Fig. 18 Two-phase pulse processing operation of timer A4

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

### (3) One-shot pulse mode [10]

Figure 20 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the  $T_{AiN}$  pin. Software trigger is selected when bit 4 is "0" and the input signal from the  $T_{AiN}$  pin is used as the trigger when it is "1".

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

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

Figure 21 shows the bit configuration of the one-shot start flag.

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not  $0000_{16}$ , the  $T_{AiOUT}$  pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches  $0001_{16}$ , The  $T_{AiOUT}$  pin goes "L" and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

1

pulse frequency of the selected clock

$\times$  (counter's value at the time of trigger).

If the count start flag is "0",  $T_{AiOUT}$  goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start flag.

As shown in Figure 23, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.

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

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

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

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

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

Undefined data is read when timer Ai is read.

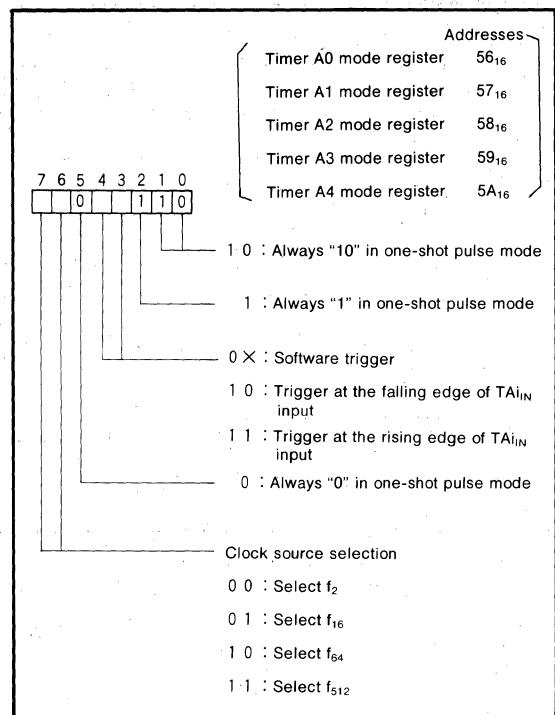


Fig. 20 Timer Ai mode register bit configuration during one-shot pulse mode

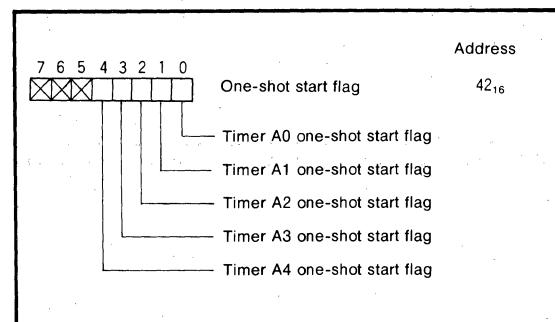


Fig. 21 One-shot start flag bit configuration

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP  
M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

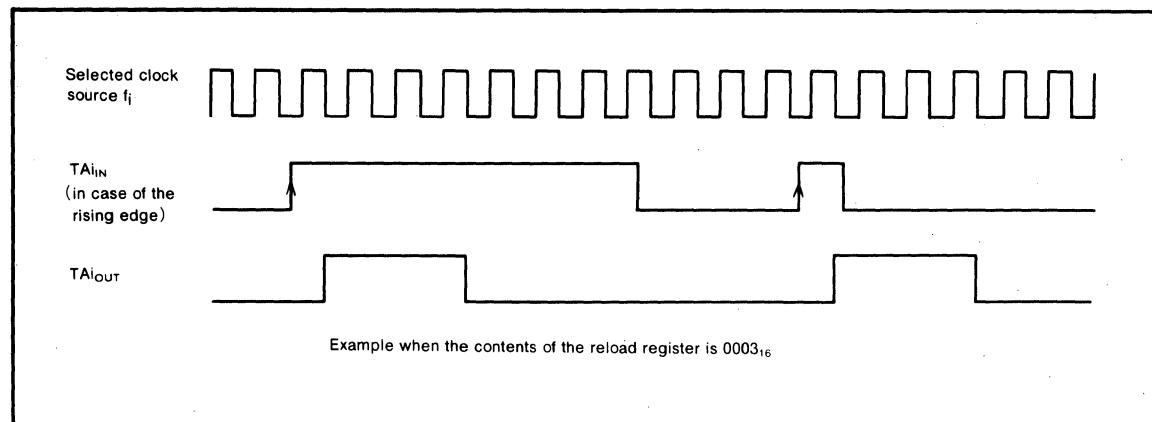


Fig. 22 Pulse output example when external rising edge is selected

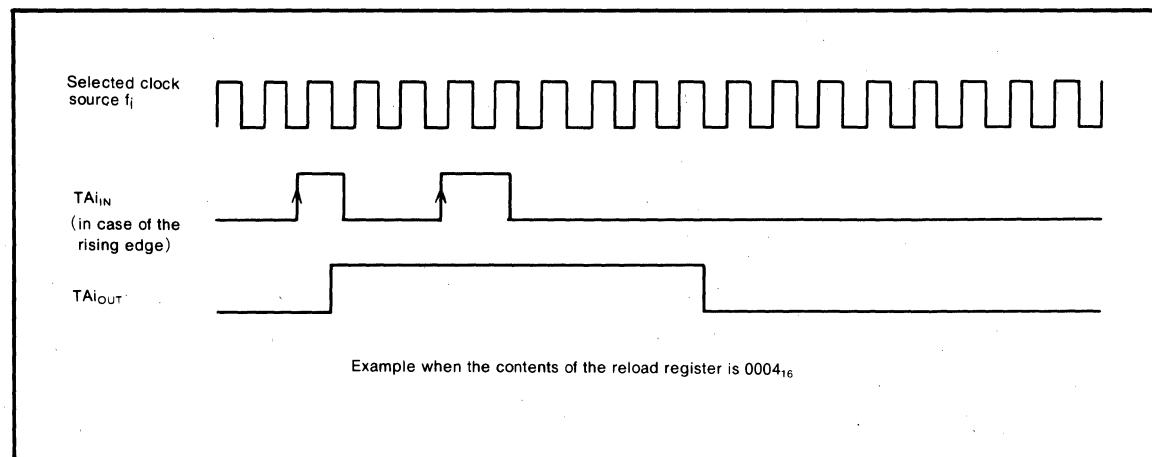


Fig. 23 Example when trigger is re-issued during pulse output

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### (4) Pulse width modulation mode [11]

Figure 24 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1". Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is performed when bit 5 is "0" and 8-bit length pulse width modulator is performed when it is "1". The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi<sub>IN</sub> pin (external trigger).

The software trigger mode is selected when bit 4 is "0". Pulse width modulator is started and pulse is output from TAi<sub>OUT</sub> when the timer Ai start flag is set to "1".

The external trigger mode is selected when bit 4 is "1". Pulse width modulator starts when a trigger signal is input from the TAi<sub>IN</sub> pin when the timer Ai start flag is "1". Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is "0" and at the rise when it is "1".

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

Then when the time Ai start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration "H" of pulse is

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

and the output pulse period is

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

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

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

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

Undefined data is read when timer Ai is read.

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

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

The reload register and the counter are both divided into 8-bit halves.

The low order 8 bits function as a prescaler and the high

order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000<sub>16</sub> as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

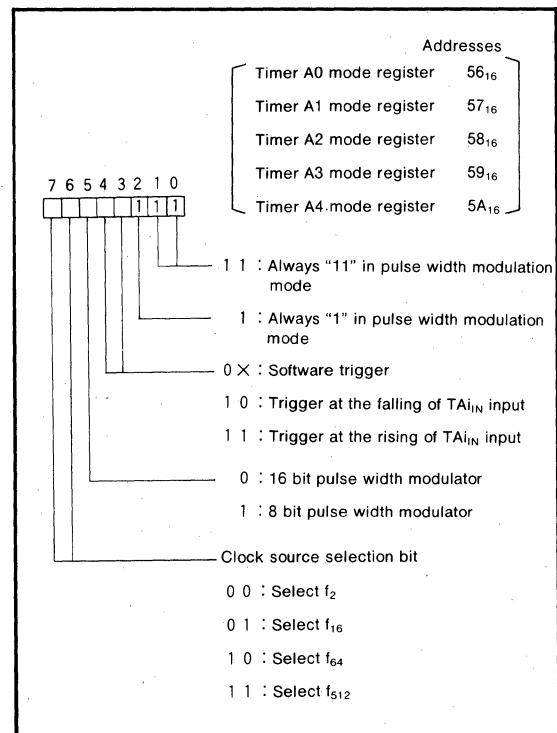


Fig. 24 Timer Ai mode register bit configuration during pulse width modulation mode

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

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

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

The high order 8-bit function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that

the length is 8 bits. If the high order 8-bit of the reload register is m, the duration "H" of pulse is

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

And the output pulse period is

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

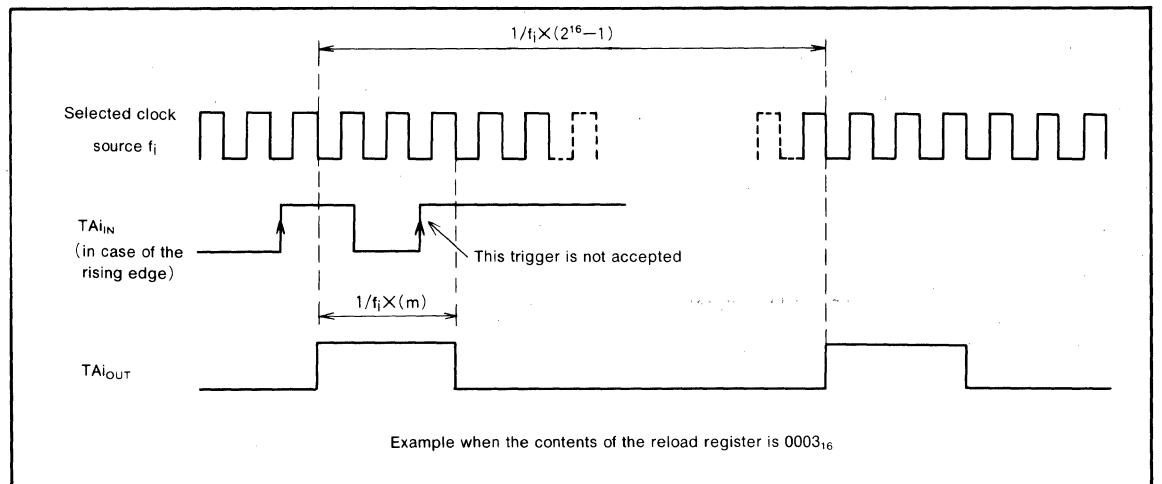


Fig. 25 16-bit length pulse width modulator output pulse example

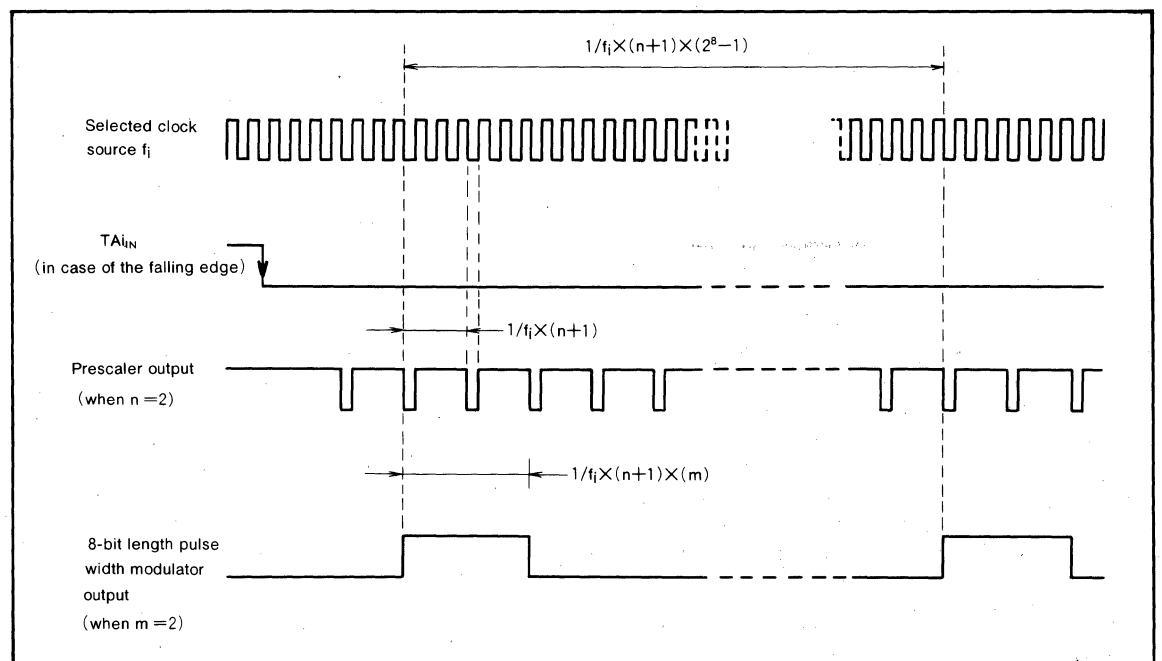


Fig. 26 8-bit length pulse width modulator output pulse example

**TIMER B**

Figure 27 shows a block diagram of timer B.

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

**(1) Timer mode [00]**

Figure 28 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0, and 1 of the timer Bi mode register must always be "0" in timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag "1" and stops when "0".

As shown in Figure 13, the timer Bi count start flag is at the same address as the timer Ai count start flag. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes  $0000_{16}$ . At the same time, the contents of the reload register is stored in the counter and count is continued.

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

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

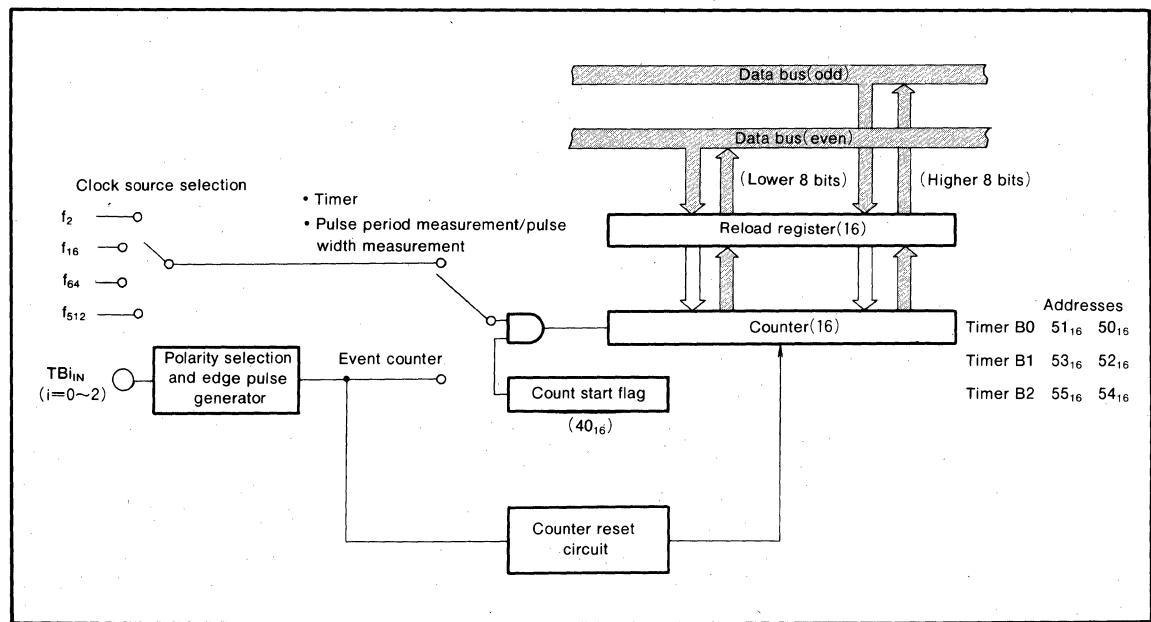


Fig. 27 Timer B block diagram

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

**(2) Event counter mode [01]**

Figure 29 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, the bit 0 in the timer Bi mode register must be "1" and bit 1 must be "0".

The input signal from the  $TB_{BiN}$  pin is counted when the count start flag is "1" and counting is stopped when it is "0". Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

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

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

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

Figure 30 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode.

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

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

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

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 31, when the fall of the input signal from  $TB_{BiN}$  pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

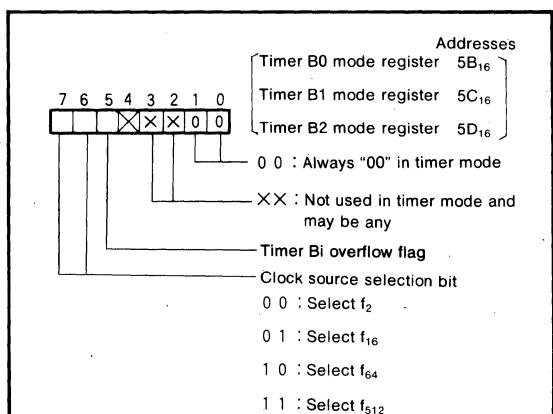


Fig. 28 Timer Bi mode register bit configuration during timer mode

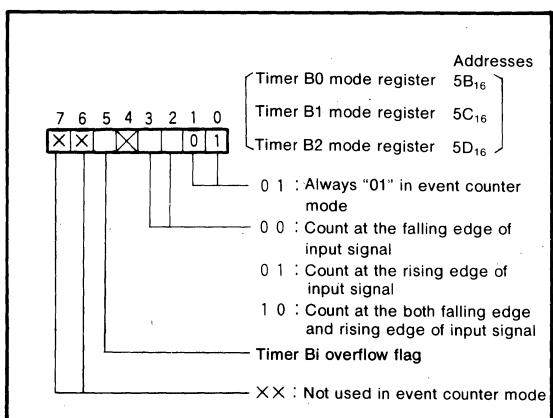


Fig. 29 Timer Bi mode register bit configuration during event counter mode

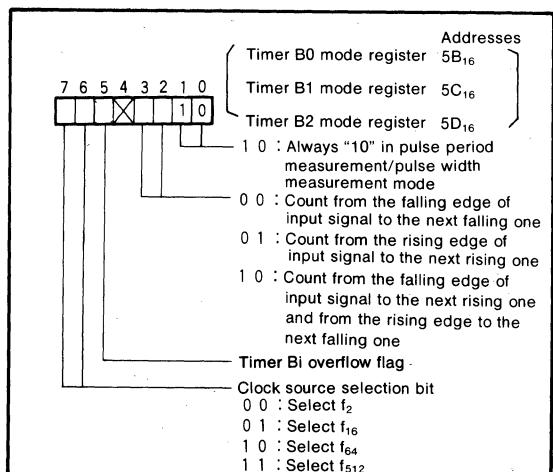


Fig. 30 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to "1". When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to pulse period measurement mode except that the clock is counted from the fall of the  $TBi_{IN}$  pin input signal to the next rise or from the rise of the input signal to the next fall as

shown in Figure 32.

When timer Bi is read, the contents of the reload register is read.

Note that in this mode, the interval between the fall of the  $TBi_{IN}$  pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000<sub>16</sub>. This flag is cleared by writing to corresponding timer Bi mode register. This bit is set to "1" at reset.

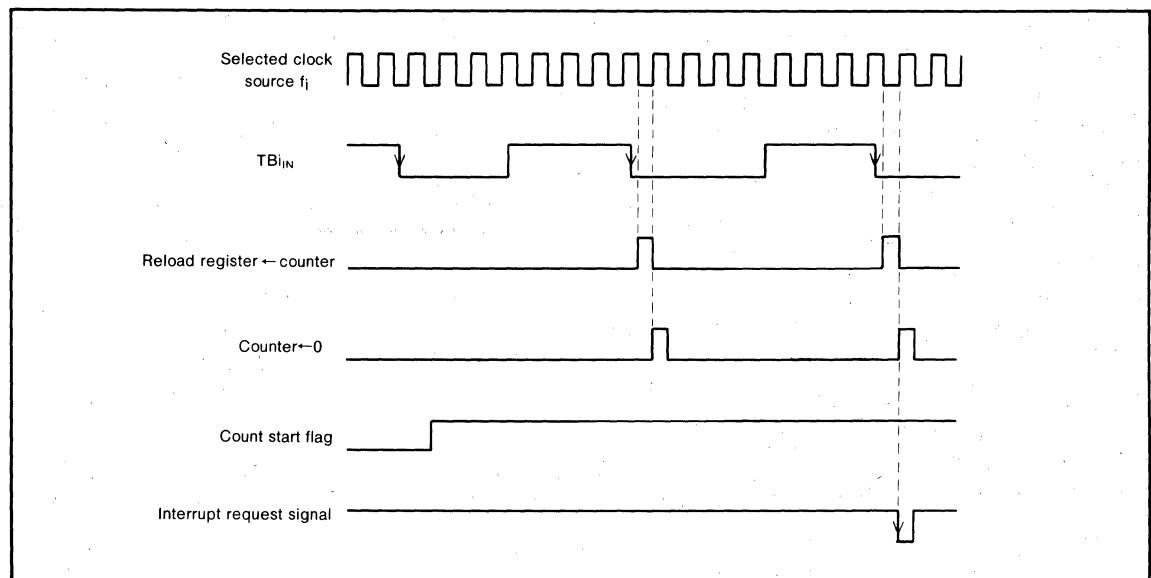


Fig. 31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

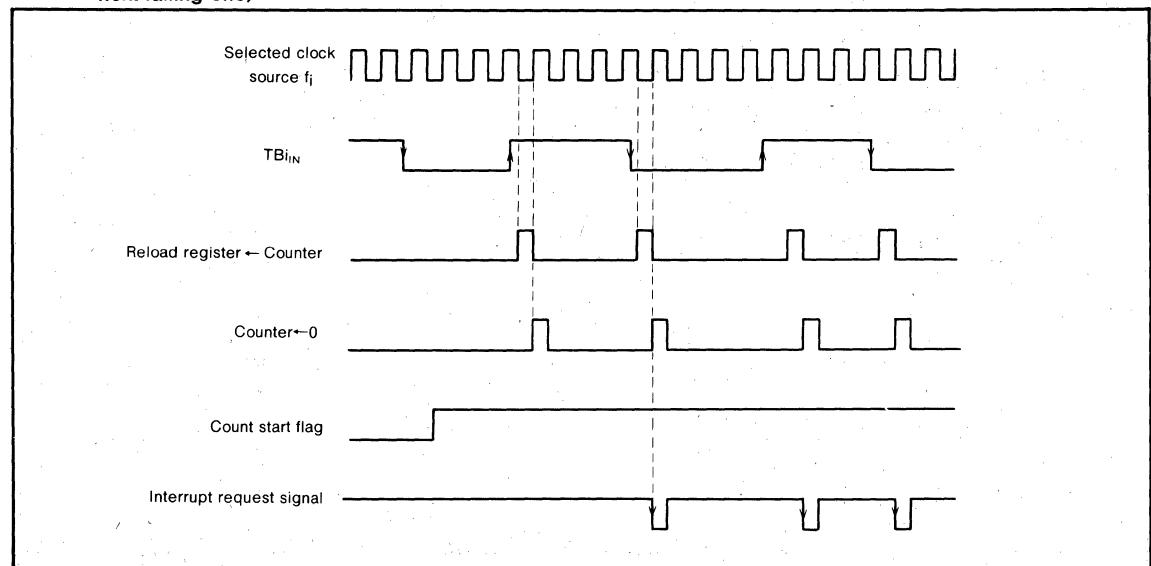


Fig. 32 Pulse width measurement mode operation

**Pulse output port mode**

Figure 33 shows a block diagram for pulse output port mode. In the pulse output port mode, two pairs of four-bit pulse output ports are used. Whether using pulse output port or not can be selected by waveform output selection bit (bit 0, bit 1) of waveform output mode register ( $62_{16}$  address) shown in Figure 34. When bit 0 of waveform output selection bit is set to "1", ports P5<sub>7</sub>, P5<sub>6</sub>, P5<sub>5</sub> and P5<sub>4</sub> are used as pulse output ports (RTP1 selected), and when bit 1 of waveform output selection bit is set to "1", ports P5<sub>3</sub>, P5<sub>2</sub>, P5<sub>1</sub>, and P5<sub>0</sub> are used as pulse output ports (RTP0 selected). When bits 1 and 0 of waveform output selection bit are set to "1", ports P5<sub>7</sub>, P5<sub>6</sub>, P5<sub>5</sub>, and P5<sub>4</sub>, and ports P5<sub>3</sub>, P5<sub>2</sub>, P5<sub>1</sub> and P5<sub>0</sub> are used as pulse output ports (RTP1 and RTP0 selected).

The ports not used as pulse output ports can be used as normal parallel ports or timer input/output.

In the pulse output port mode, set timers A2 and A0 to timer mode as timers A2 and A0 are used. Figure 35 shows the bit configuration of timer A0, A2 mode registers in pulse output port mode.

Data can be set in each bit of the pulse output data regis-

ter corresponding to four ports selected as pulse output ports. Figure 36 shows the bit configuration of the pulse output data register. The contents of the pulse output data register 1 (low-order four bits of  $64_{16}$  address) corresponding to ports P5<sub>7</sub>, P5<sub>6</sub>, P5<sub>5</sub> and P5<sub>4</sub> is output to the ports each time the counter of timer A2 becomes  $0000_{16}$ . The contents of the pulse output data register 0 (low-order four bits of  $65_{16}$  address) corresponding to ports P5<sub>3</sub>, P5<sub>2</sub>, P5<sub>1</sub>, and P5<sub>0</sub> is output to the ports each time the counter of timer A0 becomes  $0000_{16}$ .

When "0" is written to a specified bit of the pulse output data register, "L" level is output to the corresponding pulse output port when the counter of corresponding timer becomes  $0000_{16}$ , and when "1" is written, "H" level is output to the pulse output port.

Pulse width modulation can be applied to each pulse output port. Since pulse width modulation involves the use of timers A3 and A1, activate these timers in pulse width modulation mode. When a certain bit of the pulse output register is "1", pulse width modulation is output from the pulse output port when the counter of the corresponding timer becomes  $0000_{16}$ .

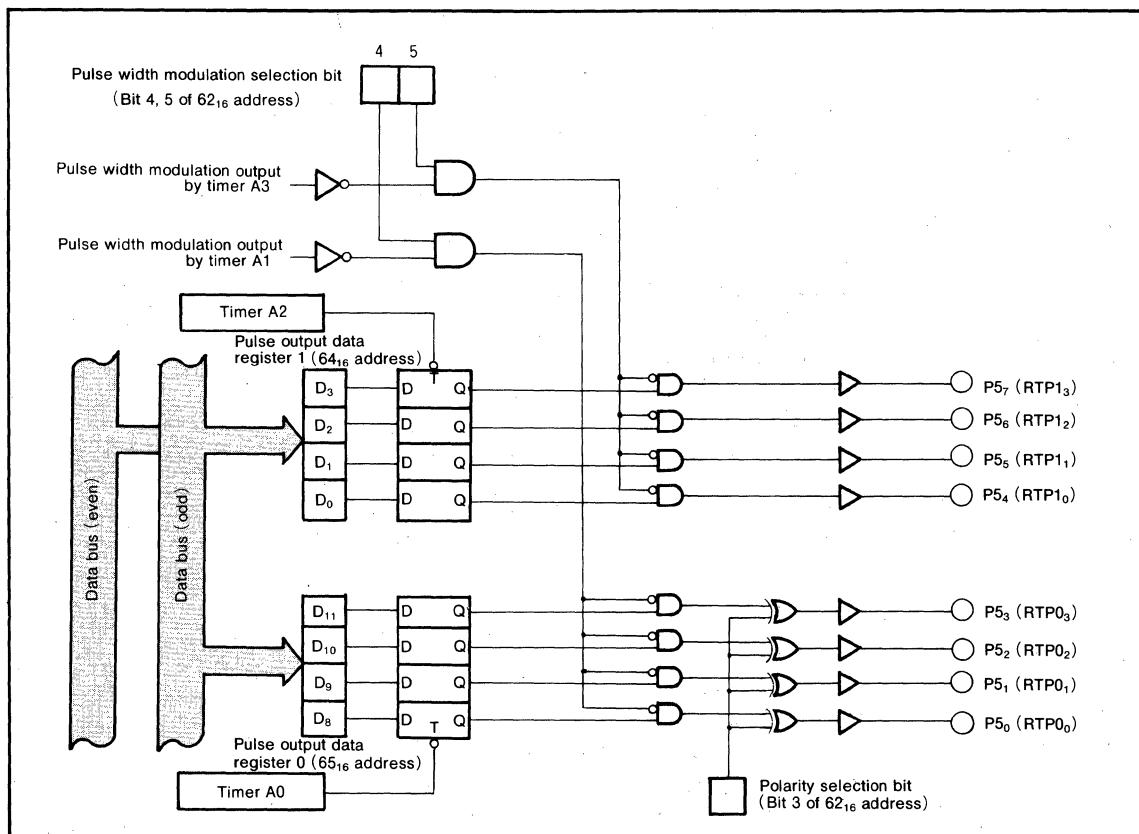


Fig. 33 Block diagram for pulse output port mode

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

Ports P5<sub>7</sub>, P5<sub>6</sub>, P5<sub>5</sub> and P5<sub>4</sub> are applied pulse width modulation by timer A3 by setting the pulse width modulation selection bit by timer A3 (bit 5) of the waveform output mode register to "1".

Ports P5<sub>3</sub>, P5<sub>2</sub>, P5<sub>1</sub> and P5<sub>0</sub> are applied pulse width modulation by timer A1 by setting the pulse width modulation selection bit by timer A1 (bit 4) of the waveform output mode register to "1".

The contents of the pulse output data register 0 can be reversed and output to pulse output ports P5<sub>3</sub>, P5<sub>2</sub>, P5<sub>1</sub> and P5<sub>0</sub> by the polarity selection bit (bit 3) of the waveform output mode register. When the polarity selection bit is "0", the contents of the pulse output data register 0 is output unchangeably, and when "1", the contents of the pulse output data register 0 is reversed and output. When pulse width modulation is applied, likewise the polarity reverse to pulse width modulation can be selected by the polarity selection bit.

Figure 37 shows example of waveforms in pulse output port mode.

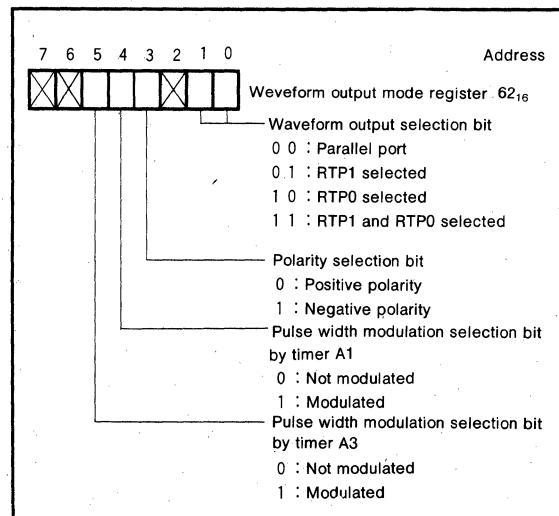


Fig. 34 Waveform output mode register bit configuration

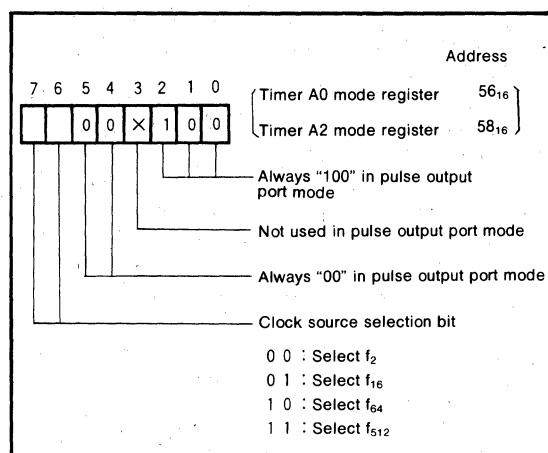


Fig. 35 Timer A0, A2 mode register bit configuration in pulse output port mode

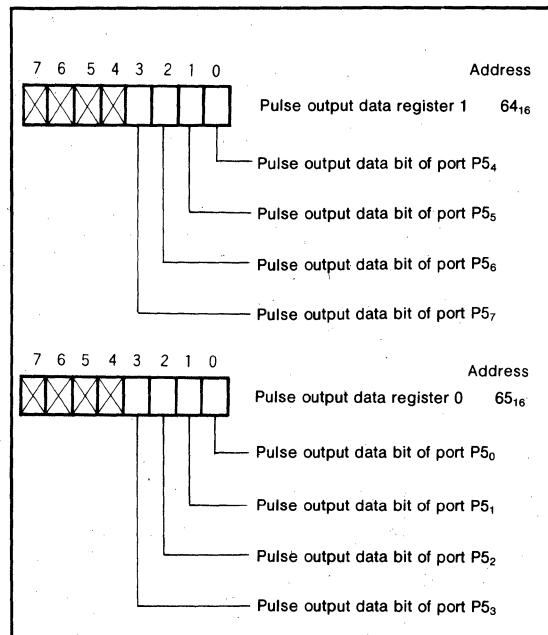


Fig. 36 Pulse output data register bit configuration

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

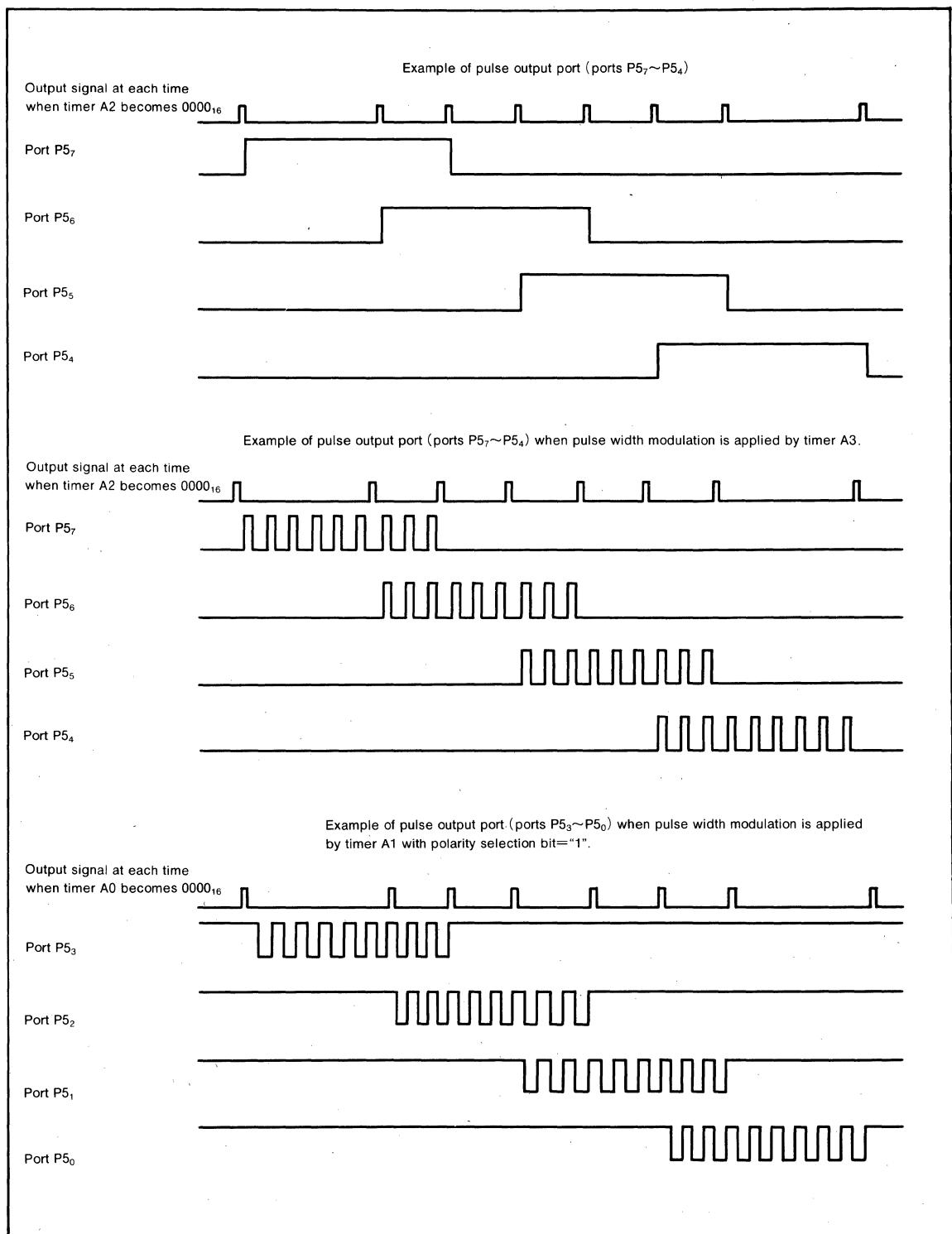


Fig. 37 Example of waveforms in pulse output port mode

## SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 38 shows a block diagram of the serial I/O ports.

Bits 0, 1, and 2 of the UART*i* (*i*=0, 1) Transmit/Receive mode register shown in Figure 39 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port us-

ing start and stop bits.

Figures 40 and 41 show the connections of receiver/transmitter according to the mode.

Figure 42 shows the bit configuration of the UART*i* transmit/receive control register.

Each communication method is described below.

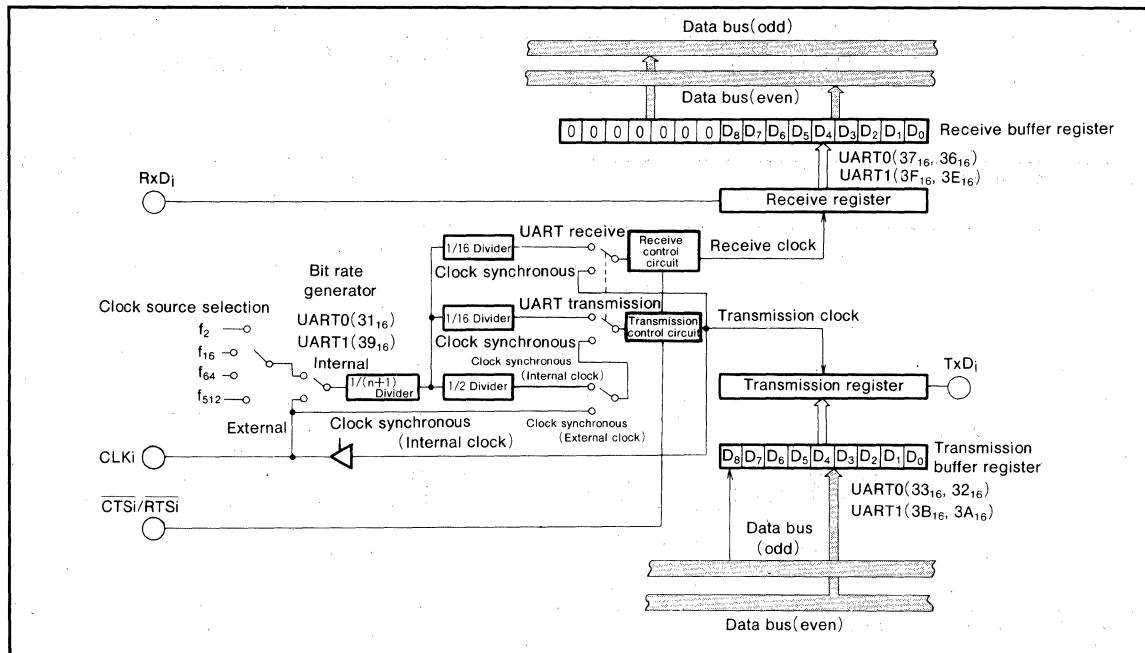


Fig. 38 Serial I/O port block diagram

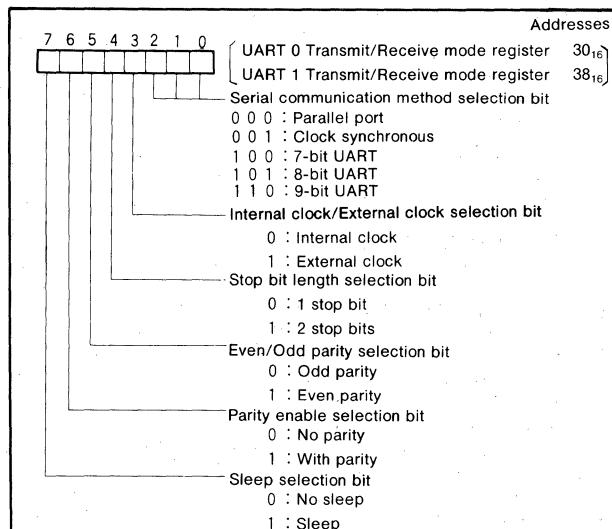
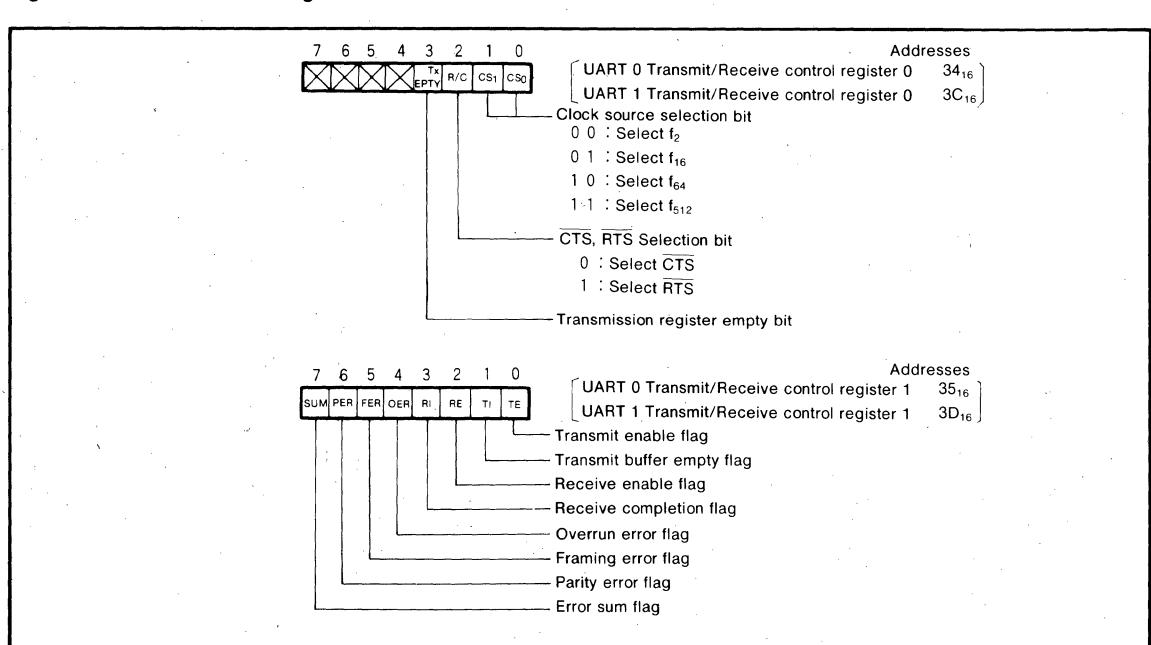
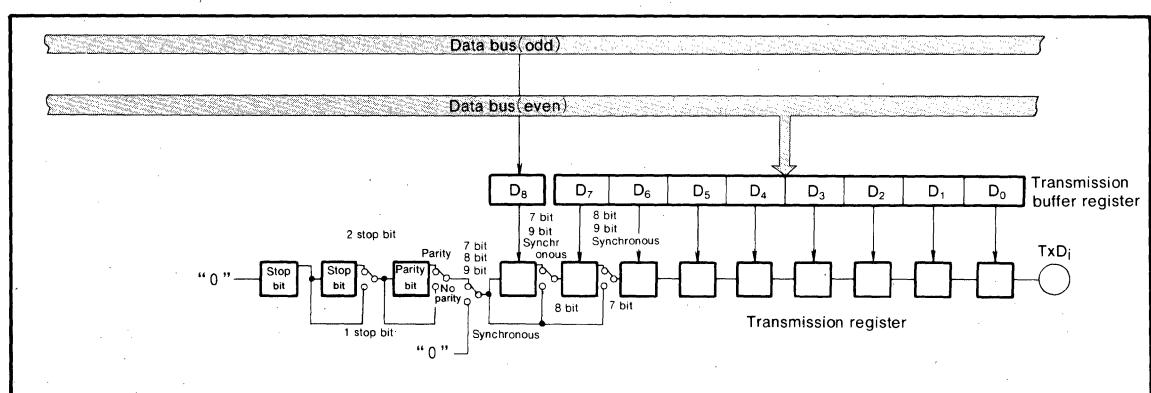
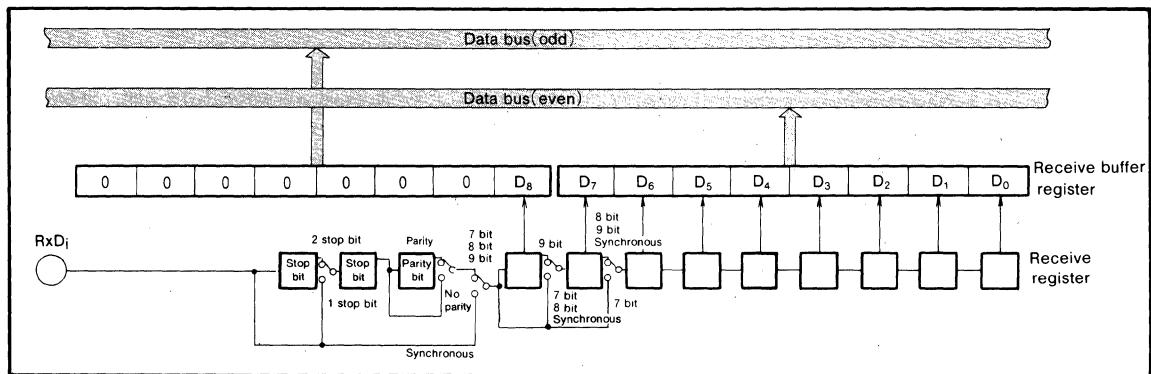


Fig. 39 UART*i* Transmit/Receive mode register bit configuration

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**



## CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 43 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

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

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

The clock source is selected by bit 0 ( $CS_0$ ) and bit 1 ( $CS_1$ ) of the clock sending side UARTj transmit/receive control register 0. As shown in Figure 38, the selected clock is divided by  $(n+1)$ , then by 2, passed through a transmission control circuit, and output as transmission clock  $CLK_j$ . Therefore, when the selected clock is  $f_i$ ,

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

On the clock receiving side, the  $CS_0$  and  $CS_1$  bits of the UARTk transmit/receive control register are ignored because an external clock is selected.

The bit 2 of the clock sending side UARTj transmit/receive control register is clear to "0" to select  $\overline{CTS}_j$  input. The bit 2 of the clock receiving side is set to "1" to select  $\overline{RTS}_k$  output.  $\overline{CTS}_j$  and  $\overline{RTS}_k$  signals are described later.

## Transmission

Transmission is started when the bit 0 (TEj flag) of UARTj transmit/receive control register 1 is "1", bit 1 is (Tlj flag) of one is "0", and  $\overline{CTS}_j$  input is "L". As shown in Figure 44, data is output from TxDj pin when transmission clock  $CLK_j$  changes from "H" to "L". The data is output from the least significant bit.

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

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of UARTj transmit/receive control register 0 is "1",  $\overline{CTS}_j$  input is ignored and transmission start is controlled only by the TEj flag and  $Tlj$  flag. Once transmission has started, the TEj flag,  $Tlj$  flag, and  $\overline{CTS}_j$  signals are ignored until data transmission completes. Therefore, trans-

mission is not interrupt when  $\overline{CTS}_j$  input is changed to "H" during transmission.

The transmission start condition indicated by  $TE_j$  flag,  $Tlj$  flag, and  $\overline{CTS}_j$  is checked while the  $T_{ENDj}$  signal shown in Figure 44 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and  $Tlj$  flag is cleared to "0" before the  $T_{ENDj}$  signal goes "H".

The bit 3 (TxEMPTYj flag) of UARTj transmit/receive control register 0 changes to "1" at the next cycle after the  $T_{ENDj}$  signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the  $Tlj$  flag changes from "0" to "1", the interrupt request bit in the UARTj transmission interrupt control register is set to "1".

## Receive

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

The  $\overline{RTS}_k$  output is "H" when the  $RE_k$  flag is "0" and goes "L" when the  $RE_k$  flag changes to "1". It goes back to "H" when receive starts. Therefore, the  $\overline{RTS}_k$  output can be used to determine whether the receive register is ready to receive. It is ready when  $\overline{RTS}_k$  output is "L".

The data from the RxDk pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock  $CLK_j$  changes from "L" to "H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (Rlk flag) of UARTk transmit/receive control register 1 is set to "1". In other words, the setting of the Rlk flag indicates that the receive buffer register contains the received data. At this point,  $\overline{RTS}_j$  output goes "L" to indicate that the next data can be received. When the Rlk flag changes from "0" to "1", the interrupt request bit in the UARTk receive interrupt control register is set to "1". Bit 4 (OERk flag) of UARTk transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while Rlk flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. Rlk and OERk flags are cleared automatically to "0" when the low-order byte of the receive buffer register is read. The OERk flag is also cleared when the REk flag is cleared. Bit 5 (FERk flag), bit 6 (PERk flag), and bit 7 (SUMk flag) are ignored in clock synchronous mode.

As shown in Figure 38, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from UARTk to UARTj.

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

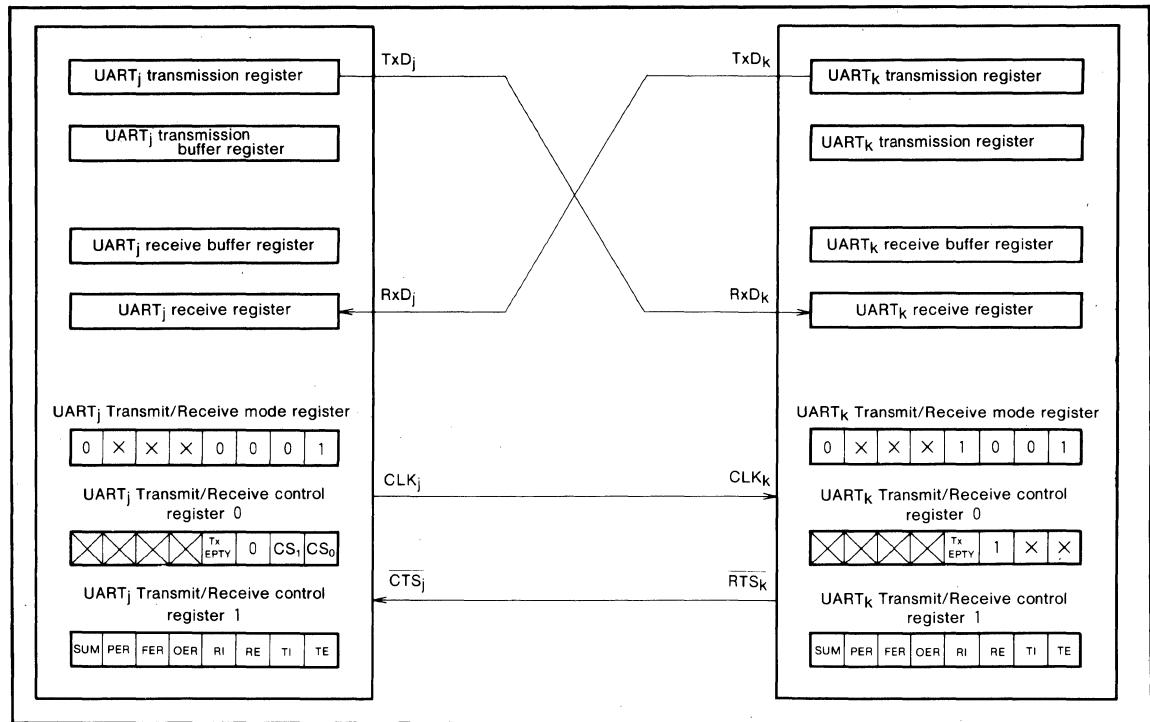


Fig. 43 Clock synchronous serial communication

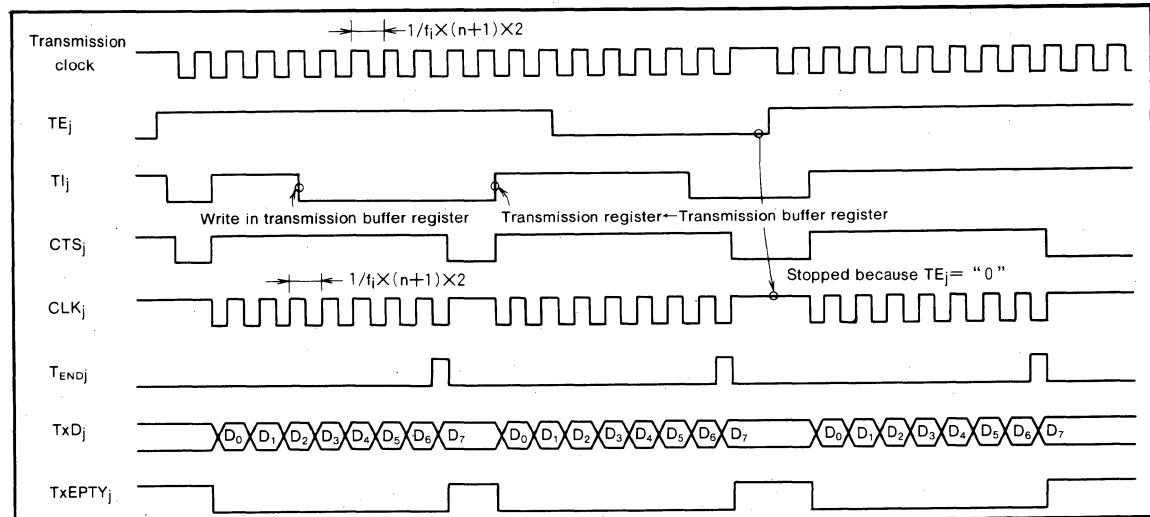


Fig. 44 Clock synchronous serial I/O timing

## ASYNCHRONOUS SERIAL COMMUNICATION

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

With 8-bit asynchronous communication, the bit 0 of UART<sub>i</sub> transmit/receive mode register is "1", the bit 1 is "0", and the bit 2 is "1".

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, the bit 0 (CS<sub>0</sub>) and bit 1 (CS<sub>1</sub>) of UART<sub>i</sub> transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK pin can be used as a normal I/O pin.

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

Therefore, the transmission speed can be changed by changing the contents n of the bit rate generator. If the selected clock is an internal clock f<sub>i</sub> or an external clock f<sub>EXT</sub>,

$$\text{Bit Rate} = (f_i \text{ or } f_{\text{EXT}}) / \{ (n+1) \times 16 \}$$

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

The bit 5 is a selection bit of odd parity or even parity.

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

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

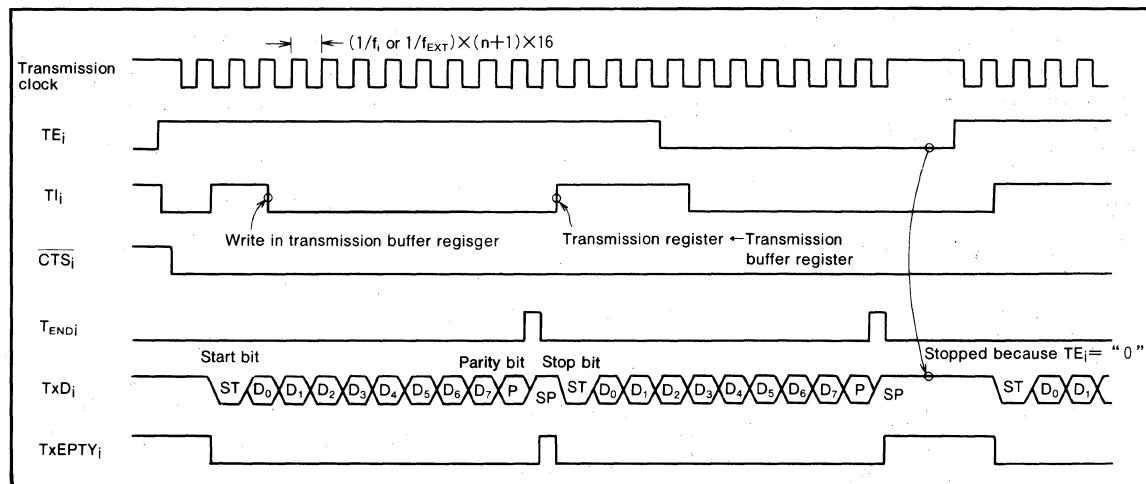


Fig. 45 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

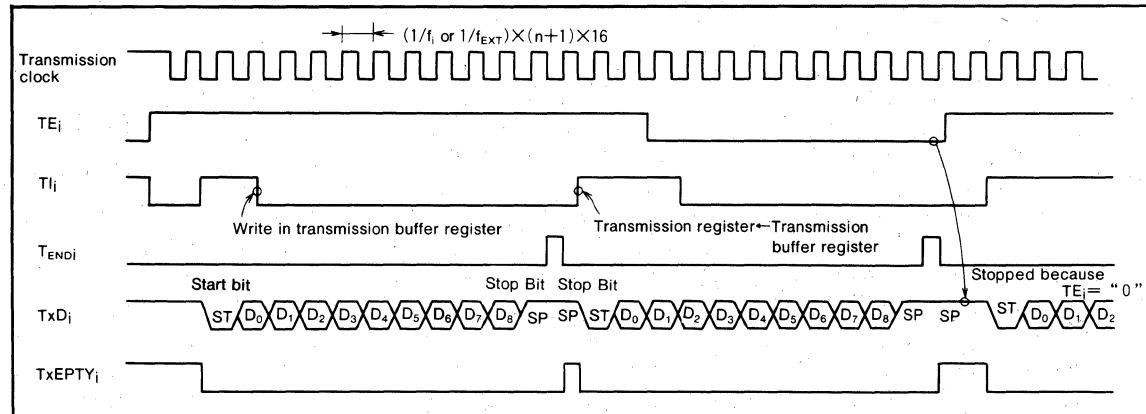


Fig. 46 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

**MITSUBISHI MICROCOMPUTERS**  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

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

Bits 4 to 6 should be set or reset according to the data format of the communicating devices.

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

The  $\overline{\text{UART}_i}$  transmit/receive control register 0 bit 2 is used to determine whether to use  $\overline{\text{CTS}_i}$  input or  $\overline{\text{RTS}_i}$  output.  $\overline{\text{CTS}_i}$  input used if bit 2 is "0" and  $\overline{\text{RTS}_i}$  output is used if bit 2 is "1".

If  $\overline{\text{CTS}_i}$  input is selected, the user can control whether to stop or start transmission by external  $\overline{\text{CTS}_i}$  input.  $\overline{\text{RTS}_i}$  will be described later.

### Transmission

Transmission is started when the bit 0 ( $\overline{\text{TE}_i}$  flag) of  $\overline{\text{UART}_i}$  transmit/receive control register 1 is "1", the bit 1 ( $\overline{\text{TI}_i}$  flag) is "0", and  $\overline{\text{CTS}_i}$  input is "L" if  $\overline{\text{CTS}_i}$  input is selected. As shown in Figure 45 and 46, data is output from the  $\overline{\text{TxD}_i}$  pin with the stop bit and parity bit specified by the bits 4 to 6 of  $\overline{\text{UART}_i}$  transmit/receive mode register bits. The data is output from the least significant bit.

The  $\overline{\text{TI}_i}$  flag indicates whether the transmission buffer is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

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

Once transmission has started, the  $\overline{\text{TE}_i}$  flag,  $\overline{\text{TI}_i}$  flag, and  $\overline{\text{CTS}_i}$  signal (if  $\overline{\text{CTS}_i}$  input is selected) are ignored until data transmission is completed.

Therefore, transmission does not stop until it completes even if the  $\overline{\text{TE}_i}$  flag is cleared during transmission.

The transmission start condition indicated by  $\overline{\text{TE}_i}$  flag,  $\overline{\text{TI}_i}$  flag, and  $\overline{\text{CTS}_i}$  is checked while the  $\overline{\text{TEND}_i}$  signal shown in Figure 45 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and  $\overline{\text{TI}_i}$  flag is cleared to 0 before the  $\overline{\text{TEND}_i}$  signal goes "H".

The bit 3 ( $\overline{\text{TxEMPTY}_i}$  flag) of  $\overline{\text{UART}_i}$  transmit/receive control register 0 changes to "1" at the next cycle after the  $\overline{\text{TEND}_i}$  signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission is completed.

When the  $\overline{\text{TI}_i}$  flag changes from "0" to "1", the interrupt request bit in the  $\overline{\text{UART}_i}$  transmission interrupt control register is set to "1".

### Receive

Receive is enabled when the bit 2 ( $\overline{\text{RE}_i}$  flag) of  $\overline{\text{UART}_i}$  transmit/receive control register 1 is set. As shown in Figure 47, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

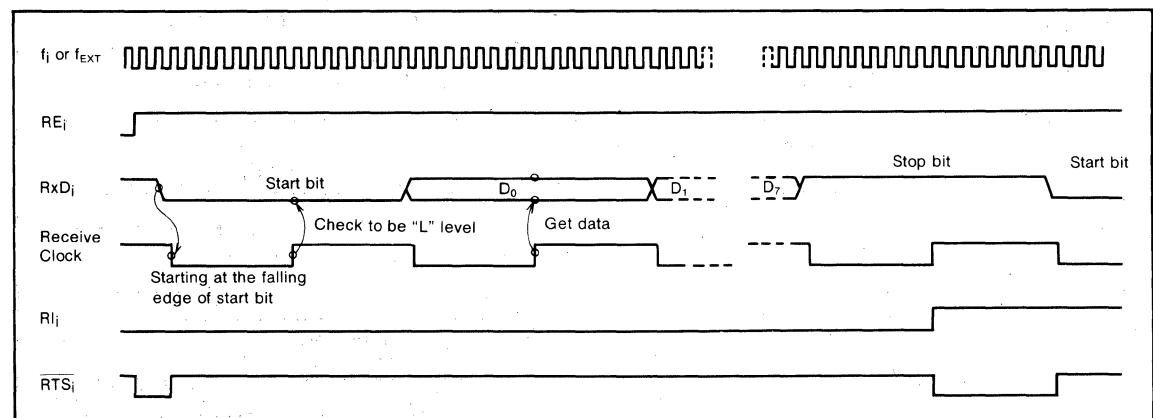


Fig. 47 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

If RTS<sub>i</sub> output is selected by setting the bit 2 of UART<sub>i</sub> transmit/receive control register 0 to "1", the RTS<sub>i</sub> output is "H" when the RE<sub>i</sub> flag is "0". When the RE<sub>i</sub> flag changes to "1", the RTS<sub>i</sub> output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, RTS<sub>i</sub> output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 40. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UART<sub>i</sub> transmit/receive control register 1 is set. In other words, the RI<sub>i</sub> flag indicates that the receive buffer register contains data when it is set. If RTS<sub>i</sub> output is selected, RTS<sub>i</sub> output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UART<sub>i</sub> receive interrupt control register is set when the RI<sub>i</sub> flag changes from "0" to "1".

The bit 4 (OER<sub>i</sub> flag) of UART<sub>i</sub> transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI<sub>i</sub> flag is "1". In other words when an overrun error occurs. If the OER<sub>i</sub> flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive buffer register has been read.

Bit 5 (FER<sub>i</sub> flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER<sub>i</sub> flag) is set when a parity error occurs.

Bit 7 (SUM<sub>i</sub> flag) is set when either the OER<sub>i</sub> flag, FER<sub>i</sub> flag, or the PER<sub>i</sub> flag is set. Therefore, the SUM<sub>i</sub> flag can be used to determine whether there is an error.

The setting of the RI<sub>i</sub> flag, OER<sub>i</sub> flag, FER<sub>i</sub> flag, and the PER<sub>i</sub> flag is performed while transferring the contents of the receive register to the receive buffer register. The RI<sub>i</sub>, OER<sub>i</sub>, FER<sub>i</sub>, PER<sub>i</sub>, and SUM<sub>i</sub> flags are cleared when the low order byte of the receive buffer register is read or when the RE<sub>i</sub> flag is cleared.

### Sleep mode

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

The sleep mode is entered when the bit 7 of UART<sub>i</sub> transmit/receive mode register is set.

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

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is "0". Also the RI<sub>i</sub>, OER<sub>i</sub>, FER<sub>i</sub>, PER<sub>i</sub>, and the SUM<sub>i</sub> flag are unchanged. Therefore, the interrupt request bit of the UART<sub>i</sub> receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

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

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

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### A-D CONVERTER

The A-D converter is an 8-bit successive approximation converter.

Figure 48 shows a block diagram of the A-D converter and Figure 49 shows the bit configuration of the A-D control register. The frequency of the A-D converter operating clock  $\phi_{AD}$  is selected by the bit 7 of the A-D control register. When bit 7 is "0",  $\phi_{AD}$  is the clock frequency divided by 8. That is,  $\phi_{AD} = f(X_{IN})/8$ . When bit 7 is "1",  $\phi_{AD}$  is the clock frequency divided by 4 and  $\phi_{AD}$  is  $=f(X_{IN})/4$ . The  $\phi_{AD}$  during A-D conversion must be 250kHz minimum because the comparator consists of a capacity coupling amplifier.

The operating mode is selected by the bits 3 and 4 of A-D control register. The available operating modes are one-shot, repeat, single sweep, and repeat sweep.

The bit of data direction register bit corresponding to the A-D converter pin must be "0" (input mode) because the analog input port is shared with port P7.

The operation of each mode is described below.

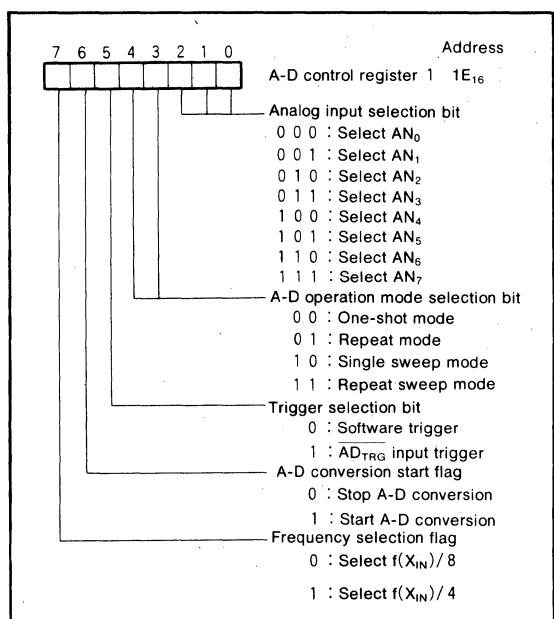


Fig. 49 A-D control register bit configuration

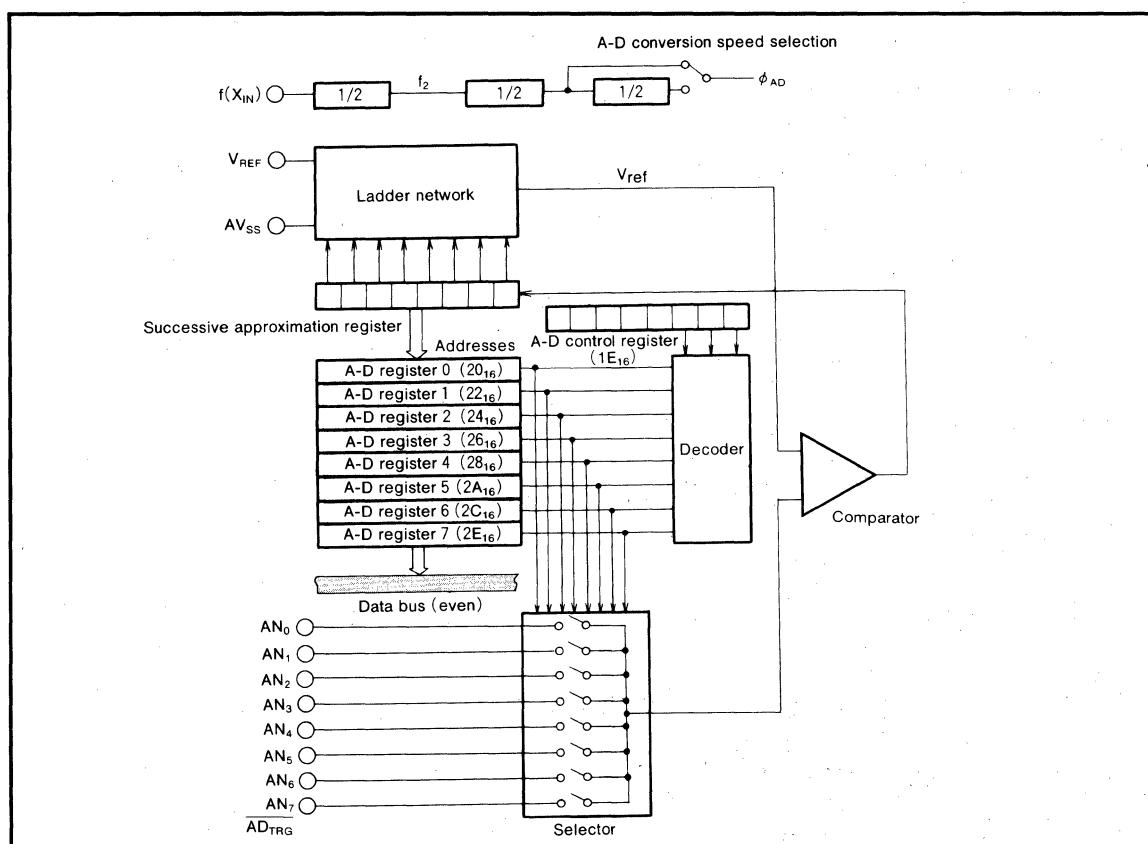


Fig. 48 A-D converter block diagram

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### (1) One-shot mode [00]

The A-D conversion pins are selected with the bit 0 to 2 of A-D control register. A-D conversion can be started by a software trigger or by an external trigger.

A software trigger is selected when the bit 5 of A-D control register is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set. A-D conversion ends after  $57 \phi_{AD}$  cycles and an interrupt request bit is set in the A-D conversion interrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the  $AD_{TRG}$  input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are  $AN_0$  to  $AN_6$  because the  $AD_{TRG}$  pin is shared with the analog voltage input pin  $AN_7$ . The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

### (2) Repeat mode [01]

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated. Also, no interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The contents of the A-D register can be read at any time.

### (3) Single sweep mode [10]

In the sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D sweep pin selection register (1F<sub>16</sub> address) shown in Figure 50. Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

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

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

When a software trigger is selected, A-D conversion is started when A-D control register bit 6 (A-D conversion start flag) is set. When A-D conversion of all selected pins end, an interrupt request bit is set in the A-D conversion in-

terrupt control register. At the same time, A-D control register bit 6 (A-D conversion start flag) is cleared and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the  $AD_{TRG}$  input changes from "H" to "L". In this case, the A-D conversion result of the trigger input itself is stored in the A-D register 7 because the  $AD_{TRG}$  pin is shared with  $AN_7$  pin. The operation is the same as done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

### (4) Repeat sweep mode [11]

The difference with the single sweep mode is that A-D conversion does not stop after converting from the  $AN_0$  pin to the selected pins, but repeats again from the  $AN_0$  pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The A-D register can be read at any time.

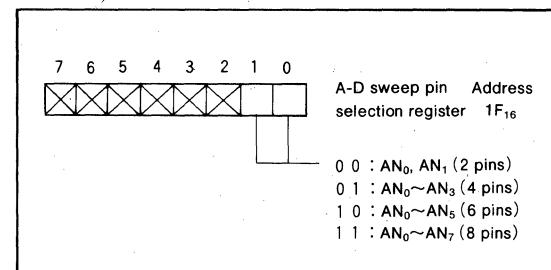


Fig. 50 A-D sweep pin selection register configuration

**MITSUBISHI MICROCOMPUTERS**  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

## WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away.

Figure 51 shows a block diagram of the watchdog timer. The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 ( $f_{32}$ ) or by 512 ( $f_{512}$ ). Whether to count  $f_{32}$  or  $f_{512}$  is determined by the watchdog timer frequency selection flag shown in Figure 52.  $f_{512}$  is selected when the flag is "0" and  $f_{32}$  is selected when it is "1". The flag is cleared after reset.  $FFF_{16}$  is set in the watchdog timer when "L" or  $2V_{CC}$  is applied to the RESET pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After  $FFF_{16}$  is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency  $f_{32}$  or  $f_{512}$ , and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and  $FFF_{16}$  is preset in the watchdog timer.

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

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register described in Figure 10 in the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the RESET pin voltage is raised to double the  $V_{CC}$  voltage.

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

The watchdog timer hold the contents during a hold state and the frequency is stopped to input.

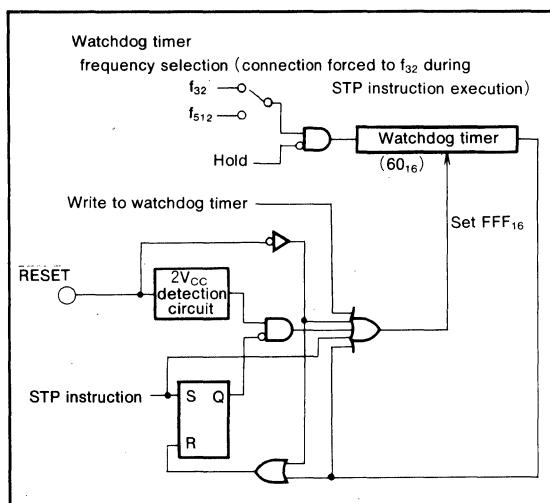


Fig. 51 Watchdog timer block diagram

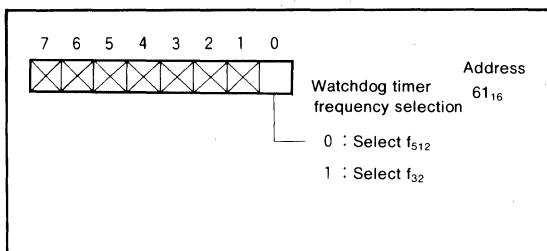


Fig. 52 Watchdog timer frequency selection flag

## RESET CIRCUIT

Reset occurs when the RESET pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V  $\pm 10\%$ . Program execution starts at the address formed by setting the address pins A<sub>23</sub>~A<sub>16</sub> to 00<sub>16</sub>, A<sub>15</sub>~A<sub>8</sub> to the contents of address FFFF<sub>16</sub>, and A<sub>7</sub>~A<sub>0</sub> to the contents of address FFFE<sub>16</sub>.

Figure 53 shows the status of the internal registers when a reset occurs.

Figure 54 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

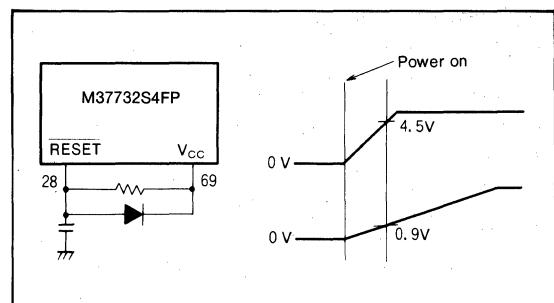


Fig. 54 Example of a reset circuit (perform careful evaluation at the system design level before using)

Address		Address
(1) Port P4 data directional register	(0C <sub>16</sub> )...	00 <sub>16</sub>
(2) Port P5 data directional register	(0D <sub>16</sub> )...	00 <sub>16</sub>
(3) Port P6 data directional register	(10 <sub>16</sub> )...	00 <sub>16</sub>
(4) Port P7 data directional register	(11 <sub>16</sub> )...	00 <sub>16</sub>
(5) Port P8 data directional register	(14 <sub>16</sub> )...	00 <sub>16</sub>
(6) A-D control register	(1E <sub>16</sub> )...	0 0 0 0 0 ? ? ?
(7) A-D sweep pin selection register	(1F <sub>16</sub> )...	X X X X X 1 1
(8) UART 0 Transmit/Receive mode register	(30 <sub>16</sub> )...	00 <sub>16</sub>
(9) UART 1 Transmit/Receive mode register	(38 <sub>16</sub> )...	00 <sub>16</sub>
(10) UART 0 Transmit/Receive control register 0	(34 <sub>16</sub> )...	X X X X 1 0 0 0
(11) UART 1 Transmit/Receive control register 0	(3C <sub>16</sub> )...	X X X X 1 0 0 0
(12) UART 0 Transmit/Receive control register 1	(35 <sub>16</sub> )...	0 0 0 0 0 0 1 0
(13) UART 1 Transmit/Receive control register 1	(3D <sub>16</sub> )...	0 0 0 0 0 0 1 0
(14) Count start flag	(40 <sub>16</sub> )...	00 <sub>16</sub>
(15) One-shot start flag	(42 <sub>16</sub> )...	X X X 0 0 0 0 0
(16) Up-down flag	(44 <sub>16</sub> )...	00 <sub>16</sub>
(17) Timer A0 mode register	(56 <sub>16</sub> )...	00 <sub>16</sub>
(18) Timer A1 mode register	(57 <sub>16</sub> )...	00 <sub>16</sub>
(19) Timer A2 mode register	(58 <sub>16</sub> )...	00 <sub>16</sub>
(20) Timer A3 mode register	(59 <sub>16</sub> )...	00 <sub>16</sub>
(21) Timer A4 mode register	(5A <sub>16</sub> )...	00 <sub>16</sub>
(22) Timer B0 mode register	(5B <sub>16</sub> )...	0 0 1 X 0 0 0 0
(23) Timer B1 mode register	(5C <sub>16</sub> )...	0 0 1 X 0 0 0 0
(24) Timer B2 mode register	(5D <sub>16</sub> )...	0 0 1 X 0 0 0 0
(25) Processor mode register	(5E <sub>16</sub> )...	X 0 0 0 0 0 1 0
(26) Watchdog timer	(60 <sub>16</sub> )...	FFF <sub>16</sub>
(27) Watchdog timer frequency selection flag	(61 <sub>16</sub> )...	X X X X X X X 0
		Contents of other registers and RAM are not initialized and should be initialized by software.
		0000 <sub>16</sub>
		00 <sub>16</sub>

Fig. 53 Microcomputer internal status during reset

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****INPUT/OUTPUT PINS**

Ports P8 to P4 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

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

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

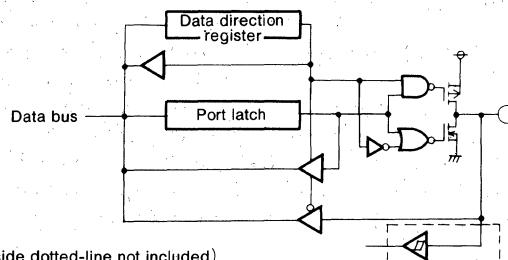
Figure 55 shows a block diagram of ports P8 to P4 and the E pin output.

In evaluation chip mode, port P4 is also used as control signal pins.

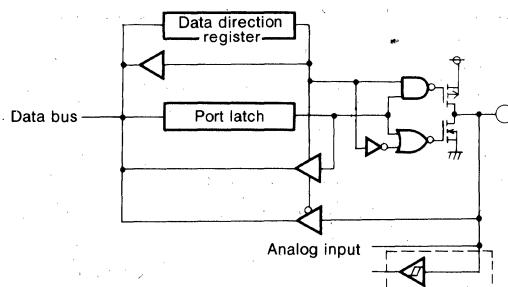
Refer to the section on processor modes for more details.

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

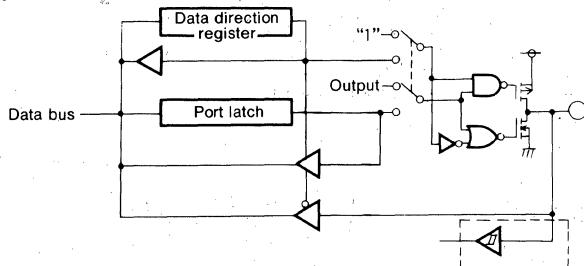
- Ports P4<sub>6</sub>~P4<sub>3</sub> (Inside dotted-line not included)
- Ports P4<sub>7</sub>, P5<sub>7</sub>, P6<sub>7</sub>~P6<sub>1</sub>, P8<sub>2</sub>, P8<sub>6</sub> (Inside dotted-line included, but P8<sub>2</sub>, P8<sub>6</sub> are without hysteresis)



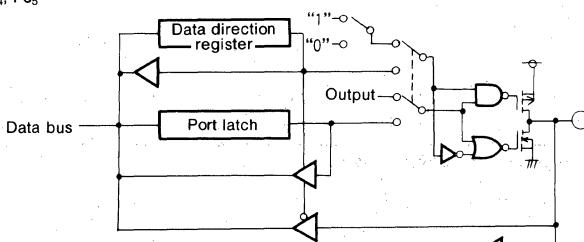
- Ports P7<sub>6</sub>~P7<sub>0</sub> (Inside dotted-line not included)
- Port P7<sub>7</sub> (Inside dotted-line included)



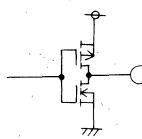
- Ports P8<sub>3</sub>, P8<sub>7</sub> (Inside dotted-line not included)
- Ports P5<sub>0</sub>~P5<sub>6</sub>, P6<sub>0</sub> (Inside dotted-line included)



- Ports P8<sub>0</sub>, P8<sub>1</sub>, P8<sub>4</sub>, P8<sub>5</sub>



- E



**Fig. 55 Block diagram for ports P8 to P4 and the E pin output**

**PROCESSOR MODE**

The bit 0 of processor mode register as shown in Figure 56 is used to select either, microprocessor mode, or evaluation chip mode.

Figure 57 shows the functions of A<sub>0</sub> to A<sub>7</sub> pins, A<sub>8</sub>/D<sub>8</sub> to A<sub>23</sub>/D<sub>7</sub> pins, and port P4 in each mode.

The external memory area changes when the mode changes.

Figure 58 shows the memory map for each mode.

The accessing of the external memory is affected by the BYTE pin and the bit 2 (wait bit) of processor mode register. These will be described next.

**•BYTE pin**

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

The data bus width is 8 bits when the level of the BYTE pin is "H" and A<sub>16</sub>/D<sub>0</sub> to A<sub>23</sub>/D<sub>7</sub> become the data I/O pin.

The data bus width is 16 bits when the level of the BYTE pin is "L" and A<sub>16</sub>/D<sub>0</sub> to A<sub>23</sub>/D<sub>7</sub> pins and A<sub>8</sub>/D<sub>8</sub> to A<sub>15</sub>/D<sub>15</sub> pins become the data I/O pins.

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

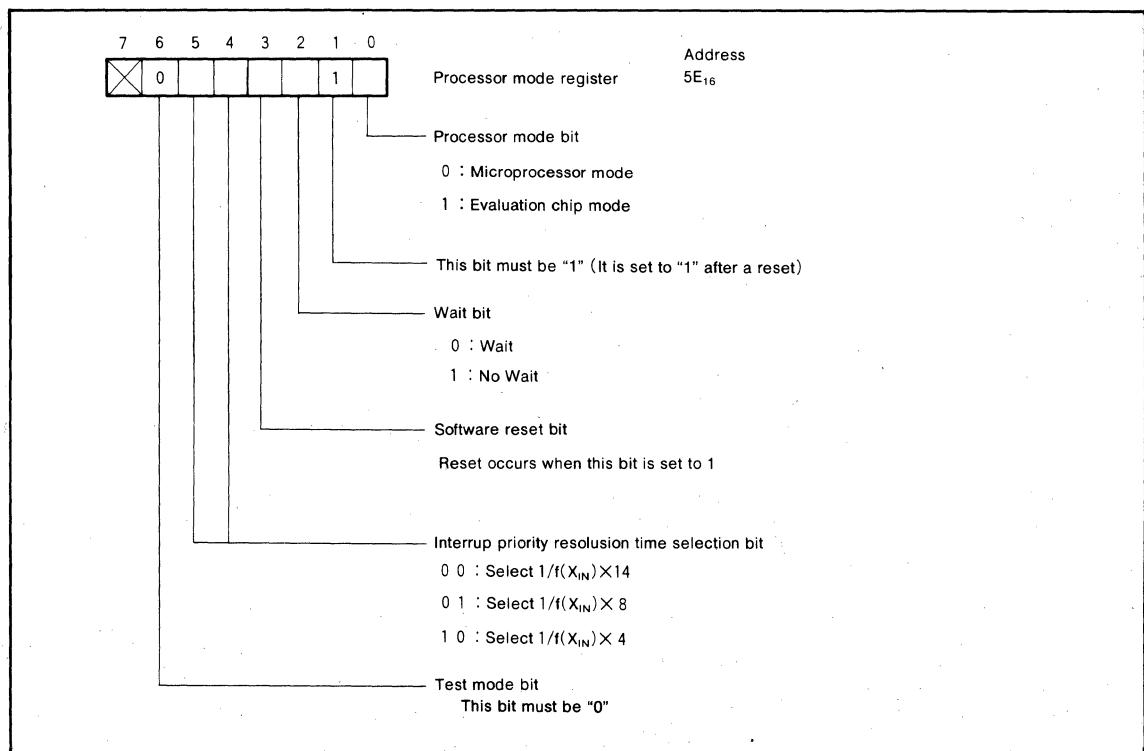


Fig. 56 Processor mode register bit configuration

**M37732S4FP, M37732S4AFP  
M37732S4BFP**
**16-BIT CMOS MICROCOMPUTER**

CM <sub>1</sub>	1	1
CM <sub>0</sub>	0	1
Mode	Microprocessor Mode	
A <sub>0</sub> ~A <sub>7</sub>		Same as left
BYTE = "L"		Same as left
A <sub>8</sub> /D <sub>8</sub> A <sub>15</sub> /D <sub>15</sub>		 Ports P4, P5 and their direction registers are treated as 16-bit wide bus.
BYTE = "L"		Same as left
BYTE = "H"		 Same as A <sub>8</sub> /D <sub>8</sub> to A <sub>15</sub> /D <sub>15</sub>
Port P4		

Fig. 57 Processor mode and A<sub>0</sub> to A<sub>7</sub> pins, A<sub>8</sub>/D<sub>8</sub> to A<sub>23</sub>/D<sub>7</sub> pins and port P4 functions

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

### • Wait bit

As shown in Figure 59, when the external memory area is accessed with the processor mode register bit 2 (wait bit) cleared to "0", the "L" width of  $\bar{E}$  signal becomes twice compared with no wait (the wait bit is "1"). The wait bit is cleared during reset.

The accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

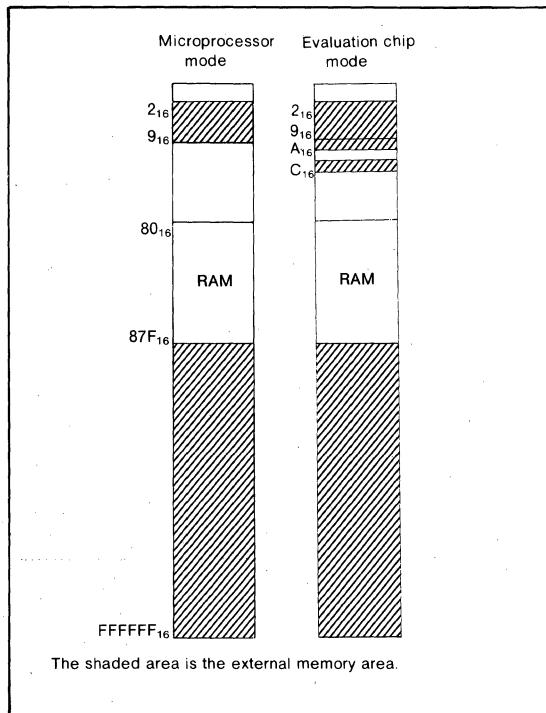


Fig. 58 External memory area for each processor mode

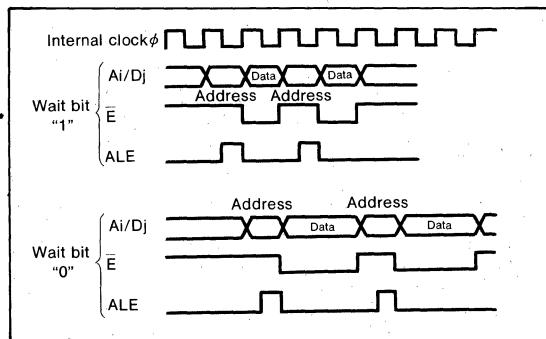


Fig. 59 Relationship between wait bit and access time

### (1) Microprocessor mode [10]

Microprocessor mode is entered by connecting the CNV<sub>SS</sub> pin to V<sub>CC</sub> and starting from reset.

A<sub>8</sub>/D<sub>8</sub> to A<sub>15</sub>/D<sub>15</sub> pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A<sub>8</sub>/D<sub>8</sub> to A<sub>15</sub>/D<sub>15</sub> pins function as an address output pin while  $\bar{E}$  is "H" and as an odd address data I/O pin while  $\bar{E}$  is "L". However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

When the BYTE pin level "H", A<sub>8</sub>/D<sub>8</sub> to A<sub>15</sub>/D<sub>15</sub> pins function as an address output pin.

A<sub>16</sub>/D<sub>0</sub> to A<sub>23</sub>/D<sub>7</sub> pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A<sub>16</sub>/D<sub>0</sub>~A<sub>23</sub>/D<sub>7</sub> pins function as an address output pin while  $\bar{E}$  is "H" and as an even address data I/O pin while  $\bar{E}$  is "L". However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

When the BYTE pin level is "H", A<sub>16</sub>/D<sub>0</sub>~A<sub>23</sub>/D<sub>7</sub> pins functions as an address output pin while  $\bar{E}$  is "H" and as an even and odd address data I/O pin while  $\bar{E}$  is "L". However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

R/W is a read/write signal which indicates a read when it is "H" and a write when it is "L".

BHE is a byte high enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously if address A<sub>0</sub> is "L" and BHE is "L".

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

HLDA is a hold acknowledge signal and is used to notify externally when the microcomputer receives HOLD input and enters into hold state.

HOLD is a hold request signal. It is an input signal used to put the microcomputer in hold state. HOLD input is accepted when the internal clock  $\phi$  falls from "H" level to "L" level while the bus is not used.  $A_0$  to  $A_7$  pins,  $A_8/D_8$  to  $A_{23}/D_7$  pins, R/W pin and BHE pin are floating while the microcomputer stays in hold state. These ports are floating after one cycle of the internal clock  $\phi$  later than HLDA signal changes to "L" level. At the removing of hold state, these ports are removed from floating state after one cycle of  $\phi$  later than HLDA signal changes to "H" level.

RDY is a ready signal. If this signal goes "L", the internal clock  $\phi$  stops at "L".  $\phi_1$  output from clock  $\phi_1$  output pin doesn't stop. RDY is used when slow external memory is attached.

## (2) Evaluation chip mode [11]

Evaluation chip mode is entered by applying voltage twice the  $V_{CC}$  voltage to the CNV<sub>ss</sub> pin. This mode is normally used for evaluation tools.

$A_8/D_8$  to  $A_{15}/D_{15}$  functions as an address output pin while  $\bar{E}$  is "H" and as data I/O pin of odd addresses while  $\bar{E}$  is "L" regardless of the BYTE pin level. However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

$A_{16}/D_0$  to  $A_{23}/D_7$  function as an address output pin while  $\bar{E}$  is "H" and as data I/O pin of even addresses while  $\bar{E}$  is "L" when the BYTE pin level is "L". However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

When the BYTE pin level is "H",  $A_{16}/D_0$  to  $A_{23}/D_7$  functions as an address output pin while  $\bar{E}$  is "H" and as data I/O pin of even and odd addresses while  $\bar{E}$  is "L". However, if an internal memory is read, external data is ignored while  $\bar{E}$  is "L".

Port P4 and its data direction register which are located at address  $0A_{16}$  and  $0C_{16}$  are treated differently in evaluation chip mode. When these addresses are accessed, the data bus width is treated as 16 bits regardless of the BYTE pin level, and the access cycle is treated as internal memory regardless of the wait bit.

Ports P4<sub>3</sub> to P4<sub>6</sub> become MX, QCL, VDA, and VPA output pins respectively. Port P4<sub>7</sub> becomes the DBC input pin.

The MX signal normally contains the contents of flag m, but the contents of flag x is output if the CPU is using flag x.

QCL is the queue buffer clear signal. It becomes "H" when the instruction queue buffer is cleared, for example, when a jump instruction is executed.

VDA is the valid data address signal. It becomes "H" while the CPU is reading data from data buffer or writing data to data buffer. It also becomes "H" when the first byte of the

instruction (operation code) is read from the instruction queue buffer.

VPA is the valid program address signal. It becomes "H" while the CPU is reading an instruction code from the instruction queue buffer.

DBC is the debug control signal and is used for debugging. Table 5 shows the relationship between the CNV<sub>ss</sub> pin input levels and processor modes.

Table 5. Relationship between the CNV<sub>ss</sub> pin input levels and processor modes

CNV <sub>ss</sub>	Mode	Description
$V_{CC}$	<ul style="list-style-type: none"> <li>• Microprocessor</li> <li>• Evaluation chip</li> </ul>	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2 \cdot V_{CC}$	• Evaluation chip	• Evaluation chip mode only.

**CLOCK GENERATING CIRCUIT**

Figure 60 shows a block diagram of the clock generator. When an STP instruction is executed, the internal clock  $\phi$  stops oscillating at "L" level. At the same time, FFF<sub>16</sub> is written to watchdog timer and the watchdog timer input connection is forced to f<sub>32</sub>. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation resumes when an interrupt is received, but the internal clock  $\phi$  remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator.

When a WIT instruction is executed, the internal clock  $\phi$  stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

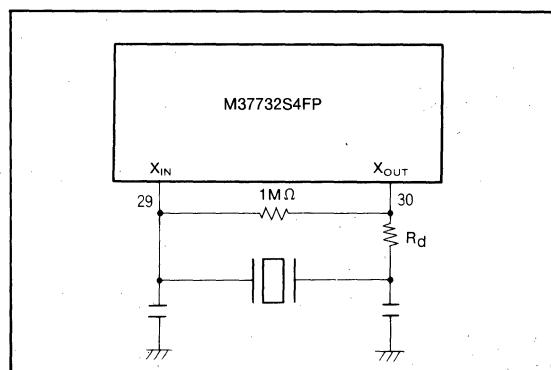


Fig. 61 Circuit using a ceramic resonator

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 61 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 62 shows an example of using an external clock signal.

**ADDRESSING MODES**

The M37732S4FP has 28 powerful addressing modes.

Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

**MACHINE INSTRUCTION LIST**

The M37732S4FP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

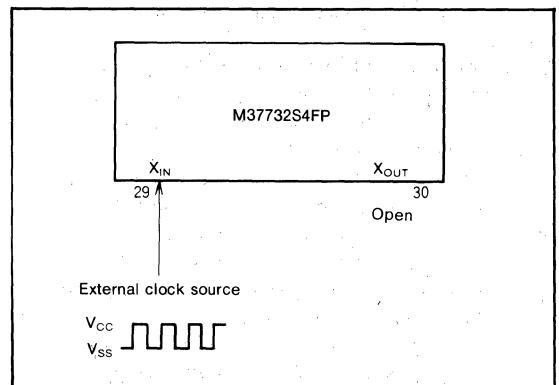


Fig. 62 External clock input circuit

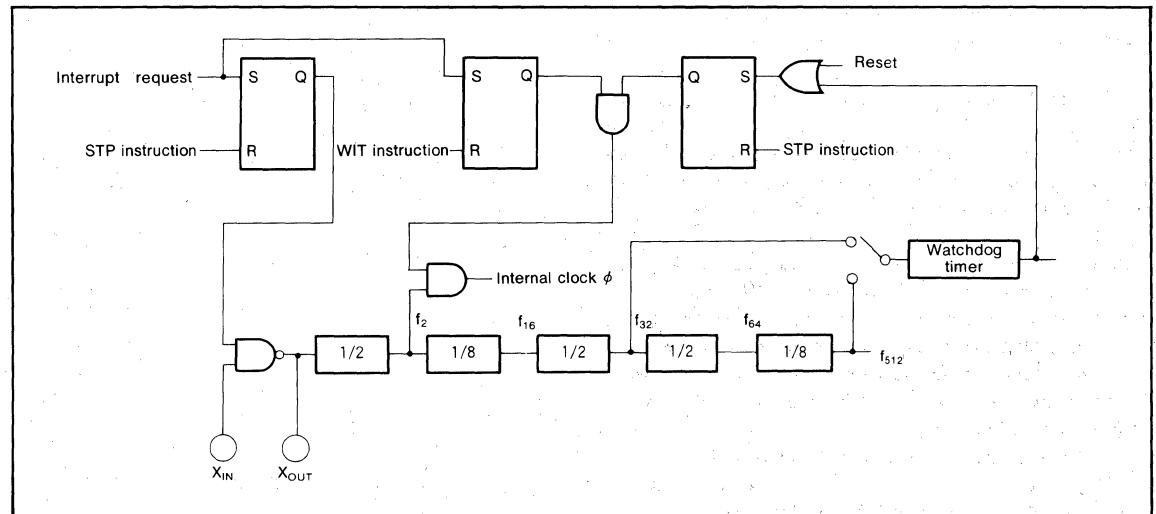


Fig. 60 Block diagram of a clock generator

**M37732S4FP, M37732S4AFP  
M37732S4BFP**
**16-BIT CMOS MICROCOMPUTER****ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Conditions	Ratings	Unit
$V_{CC}$	Supply voltage		-0.3~7	V
$AV_{CC}$	Analog supply voltage		-0.3~7	V
$V_I$	Input voltage RESET, CNV <sub>SS</sub> , BYTE		-0.3~12	V
$V_I$	Input voltage $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7, P_7 \sim P_7,$ $P_8 \sim P_8, V_{REF}, X_{IN}, HOLD, RDY$		-0.3~ $V_{CC}+0.3$	V
$V_O$	Output voltage $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7, P_7 \sim P_7,$ $P_8 \sim P_8, X_{OUT}, \bar{E}, \phi_1, HLDA, ALE, BHE, R/W$		-0.3~ $V_{CC}+0.3$	V
$P_d$	Power dissipation	$T_a=25^\circ C$	300	mW
$T_{opr}$	Operating temperature		-20~85	°C
$T_{stg}$	Storage temperature		-40~150	°C

**RECOMMENDED OPERATING CONDITIONS** ( $V_{CC}=5V \pm 10\%$ ,  $T_a=-20 \sim 85^\circ C$ , unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$V_{CC}$	Supply voltage	4.5	5.0	5.5	V
$AV_{CC}$	Analog supply voltage	$V_{CC}$			V
$V_{SS}$	Supply voltage	0			V
$AV_{SS}$	Analog supply voltage	0			V
$V_{IH}$	High-level input voltage $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, X_{IN}, RESET,$ $CNV_{SS}, BYTE, HOLD, RDY$	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IL}$	High-level input voltage $A_0/D_8 \sim A_{23}/D_7$	0.5 $V_{CC}$		$V_{CC}$	V
$V_{IL}$	Low-level input voltage $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, X_{IN}, RESET,$ $CNV_{SS}, BYTE, HOLD, RDY$	0		0.2 $V_{CC}$	V
$V_{IL}$	Low-level input voltage $A_0/D_8 \sim A_{23}/D_7$	0		0.16 $V_{CC}$	V
$I_{OH(peak)}$	High-level peak output current $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, \phi_1,$ $HLDA, ALE, BHE, R/W$			-10	mA
$I_{OH(avg)}$	High-level average output current $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, \phi_1,$ $HLDA, ALE, BHE, R/W$			-5	mA
$I_{OL(peak)}$	Low-level peak output current $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, \phi_1,$ $HLDA, ALE, BHE, R/W$			10	mA
$I_{OL(avg)}$	Low-level average output current $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7,$ $P_4 \sim P_7, P_5 \sim P_7, P_6 \sim P_7,$ $P_7 \sim P_7, P_8 \sim P_8, \phi_1,$ $HLDA, ALE, BHE, R/W$			5	mA
$f(X_{IN})$	M37732S4FP			8	MHz
	M37732S4AFP			16	
	M37732S4BFP			25	

Note 1. Average output current is the average value of a 100ms interval.

2. The sum of  $I_{OL(peak)}$  for ports  $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7, HLDA, ALE, BHE, R/W$ , and  $P_8$  must be 80mA or less, the sum of  $I_{OH(peak)}$  for ports  $A_0 \sim A_7, A_8/D_8 \sim A_{23}/D_7, HLDA, ALE, BHE, R/W$ , and  $P_8$  must be 80mA or less, the sum of  $I_{OL(peak)}$  for ports  $P_4, P_5, P_6, P_7, \phi_1$  must be 80mA or less, and the sum of  $I_{OH(peak)}$  for ports  $P_4, P_5, P_6, P_7, \phi_1$  must be 80mA or less.

**M37732S4FP, M37732S4AFP  
M37732S4BFP**

16-BIT CMOS MICROCOMPUTER

**M37732S4FP****ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P_4 \sim P_7$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage ALE	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P_4 \sim P_7$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage ALE	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA4 <sub>IN</sub> , TB0 <sub>IN</sub> ~TB2 <sub>IN</sub> , INT <sub>0</sub> ~INT <sub>2</sub> , AD <sub>TRG</sub> , CTS <sub>0</sub> , CTS <sub>1</sub> , CLK <sub>0</sub> , CLK <sub>1</sub>			0.4	1	V
$V_{T+}-V_{T-}$	Hysteresis RESET			0.2	0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>			0.1	0.3	V
$I_{IH}$	High-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P_4 \sim P_7$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P_4 \sim P_7$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	RAM hold voltage	When clock is stopped.	2			V
$I_{CC}$	Power supply current	Output only pin is open and other pins are $V_{SS}$ during reset.	$f(X_{IN})=8MHz$ , square waveform	6	12	mA
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k $\Omega$
$t_{CONV}$	Conversion time		28.5			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****M37732S4APP****ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P_{43} \sim P_{47}$ , $P_{50} \sim P_{57}$ , $P_{60} \sim P_{67}$ , $P_{70} \sim P_{77}$ , $P_{80} \sim P_{87}$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage ALE	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P_{43} \sim P_{47}$ , $P_{50} \sim P_{57}$ , $P_{60} \sim P_{67}$ , $P_{70} \sim P_{77}$ , $P_{80} \sim P_{87}$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage ALE	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA0_{IN} \sim TA4_{IN}$ , $TB0_{IN} \sim TB2_{IN}$ , $INT_0 \sim INT_2$ , $AD_{TRG}$ , $CTS_0$ , $CTS_1$ , $CLK_0$ , $CLK_1$ , CNV <sub>SS</sub> , BYTE, HOLD, RDY			0.4	1	V
$V_{T+}-V_{T-}$	Hysteresis RESET			0.2	0.5	V
$V_{T+}-V_{T-}$	Hysteresis $X_{IN}$			0.1	0.3	V
$I_{IH}$	High-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P_{43} \sim P_{47}$ , $P_{50} \sim P_{57}$ , $P_{60} \sim P_{67}$ , $P_{70} \sim P_{77}$ , $P_{80} \sim P_{87}$ , $X_{IN}$ , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P_{43} \sim P_{47}$ , $P_{50} \sim P_{57}$ , $P_{60} \sim P_{67}$ , $P_{70} \sim P_{77}$ , $P_{80} \sim P_{87}$ , $X_{IN}$ , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	RAM hold voltage	When clock is stopped.		2		V
$I_{CC}$	Power supply current	Output only pin is open and other pins are $V_{SS}$ during reset.	$f(X_{IN})=16MHz$ , square waveform	12	24	$mA$
			$T_a=25^\circ C$ when clock is stopped.		1	
			$T_a=85^\circ C$ when clock is stopped.		20	$\mu A$

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k $\Omega$
$t_{CONV}$	Conversion time		14.25			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37732S4FP, M37732S4AFP  
M37732S4BFP**
**16-BIT CMOS MICROCOMPUTER****M37732S4BFP****ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P4_0 \sim P4_7$ , $P5_0 \sim P5_7$ , $P6_0 \sim P6_7$ , $P7_0 \sim P7_7$ , $P8_0 \sim P8_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage ALE	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-10mA$	3.4			V
$V_{OH}$	High-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $P4_3 \sim P4_7$ , $P5_0 \sim P5_7$ , $P6_0 \sim P6_7$ , $P7_0 \sim P7_7$ , $P8_0 \sim P8_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $A_0 \sim A_7$ , $A_8/D_8 \sim A_{23}/D_7$ , $\phi_1$ , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage ALE	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA4 <sub>IN</sub> , TB0 <sub>IN</sub> ~TB2 <sub>IN</sub> , INT <sub>0</sub> ~INT <sub>2</sub> , AD <sub>TRG</sub> , CTS <sub>0</sub> , CTS <sub>1</sub> , CLK <sub>0</sub> , CLK <sub>1</sub>			0.4	1	V
$V_{T+}-V_{T-}$	Hysteresis RESET			0.2	0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>			0.1	0.3	V
$I_{IH}$	High-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P4_3 \sim P4_7$ , $P5_0 \sim P5_7$ , $P6_0 \sim P6_7$ , $P7_0 \sim P7_7$ , $P8_0 \sim P8_7$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_I=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $A_8/D_8 \sim A_{23}/D_7$ , $P4_3 \sim P4_7$ , $P5_0 \sim P5_7$ , $P6_0 \sim P6_7$ , $P7_0 \sim P7_7$ , $P8_0 \sim P8_7$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, HOLD, RDY	$V_I=0V$			-5	$\mu A$
$V_{RAM}$	RAM hold voltage	When clock is stopped.		2		V
$I_{CC}$	Power supply current	Output only pin is open and other pins are V <sub>SS</sub> during reset.	$f(X_{IN})=25MHz$ , square waveform	19	38	mA
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k $\Omega$
$t_{CONV}$	Conversion time		9.12			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**TIMING REQUIREMENTS ( $V_{CC}=5V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ , unless otherwise noted)**External clock input**

Symbol	Parameter	Limits						Unit	
		8 MHz		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	

**Microprocessor mode**

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{SU(DH-E)}$	Data high-order input setup time	60		45		30		ns	
$t_{SU(DL-E)}$	Data low-order input setup time	60		45		30		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	
$t_{SU(RDY-\phi_1)}$	RDY input setup time	70		60		55		ns	
$t_{SU(HOLD-\phi_1)}$	HOLD input setup time	70		60		55		ns	
$t_h(E-DH)$	Data high-order input hold time	0		0		0		ns	
$t_h(E-DL)$	Data low-order 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	
$t_h(\phi_1-RDY)$	RDY input hold time	0		0		0		ns	
$t_h(\phi_1-HOLD)$	HOLD input hold time	0		0		0		ns	

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****Timer A input** (Count input in event counter mode)

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TA)}$	TA <sub>iIN</sub> input cycle time	250		125		80		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	125		62		40		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> input low-level pulse width	125		62		40		ns	

**Timer A input** (Gating input in timer mode)

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TA)}$	TA <sub>iIN</sub> input cycle time	1000		500		320		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> input low-level pulse width	500		250		160		ns	

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

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TA)}$	TA <sub>iIN</sub> input cycle time	500		250		160		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> input low-level pulse width	250		125		80		ns	

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

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> input low-level pulse width	250		125		80		ns	

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

Symbol	Parameter	Limits						Unit	
		8 MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(UP)}$	TA <sub>iOUT</sub> input cycle time	5000		2500		2000		ns	
$t_{W(UPH)}$	TA <sub>iOUT</sub> input high-level pulse width	2500		1250		1000		ns	
$t_{W(UPL)}$	TA <sub>iOUT</sub> input low-level pulse width	2500		1250		1000		ns	
$t_{SU(UP-TIN)}$	TA <sub>iOUT</sub> input setup time	1000		500		400		ns	
$t_{H(TIN-UP)}$	TA <sub>iOUT</sub> input hold time	1000		500		400		ns	

**M37732S4FP, M37732S4A FP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****Timer B input** (Count input in event counter mode)

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time (one edge count)	250		125		80		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width (one edge count)	125		62		40		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> input low-level pulse width (one edge count)	125		62		40		ns	
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time (both edges count)	500		250		160		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width (both edges count)	250		125		80		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(AD)}$	AD <sub>TRG</sub> input cycle time (minimum allowable trigger)	2000		1000		1000		ns	
$t_{W(ADL)}$	AD <sub>TRG</sub> 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 <sub>I</sub> input cycle time	500		250		200		ns	
$t_{W(CKH)}$	CLK <sub>I</sub> input high-level pulse width	250		125		100		ns	
$t_{W(CKL)}$	CLK <sub>I</sub> input low-level pulse width	250		125		100		ns	
$t_{d(c-q)}$	TxD <sub>I</sub> output delay time			150		90		80 ns	
$t_{h(c-q)}$	TxD <sub>I</sub> hold time	30		30		30		ns	
$t_{su(d-c)}$	RxD <sub>I</sub> input setup time	60		30		20		ns	
$t_{h(c-d)}$	RxD <sub>I</sub> input hold time	90		90		90		ns	

**External interrupt INT<sub>I</sub> input**

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{W(INH)}$	INT <sub>I</sub> input high-level pulse width	250		250		250		ns	
$t_{W(INL)}$	INT <sub>I</sub> input low-level pulse width	250		250		250		ns	

**MITSUBISHI MICROCOMPUTERS**  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

**SWITCHING CHARACTERISTICS** ( $V_{CC}=5V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ , unless otherwise noted)

**Microprocessor mode** (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits						Unit	
			8 MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 63	100		30		12		ns	
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			110		70		45	ns	
$t_{PZX(E-DHZ)}$	Floating start delay time (BYTE="L")			5		5		5	ns	
$t_{d(AM-E)}$	Address middle-order output delay time		100		30		12		ns	
$t_{d(AM-ALE)}$	Address middle-order output delay time		80		24		5		ns	
$t_{d(E-DLO)}$	Data low-order output delay time			110		70		45	ns	
$t_{PZX(E-DLZ)}$	Floating start delay time			5		5		5	ns	
$t_{d(AH-E)}$	Address high-order output delay time		100		30		12		ns	
$t_{d(AH-ALE)}$	Address high-order output delay time		80		24		5		ns	
$t_{d(\phi_1-HLDA)}$	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	
$t_{d(BHE-E)}$	BHE output delay time			100		30		20	ns	
$t_{d(R/W-E)}$	R/W output delay time			100		30		20	ns	
$t_{d(E-\phi_1)}$	$\phi_1$ output delay time			0	30	0	20	0	18	ns
$t_{h(E-AL)}$	Address low-order hold time			50		25		18		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")			9		9		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")			50		25		18		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")			50		25		18		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")			50		25		18		ns
$t_{h(ALE-AH)}$	Address high-order hold time			9		9		9		ns
$t_{h(E-DLO)}$	Data low-order hold time			50		25		18		ns
$t_{PZX(E-DLZ)}$	Floating release delay time			50		25		18		ns
$t_{h(E-BHE)}$	BHE hold time			18		18		18		ns
$t_{h(E-R/W)}$	R/W hold time			18		18		18		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
$t_{W(EL)}$	$\bar{E}$ pulse width				220		95		50	ns

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

**Microprocessor mode** (when wait bit = "0", and external memory area is accessed)

Symbol	Parameter	Test conditions	Limits						Unit	
			8 MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_d(AL-E)$	Address low-order output delay time	Fig.63	100		30		12		ns	
$t_d(E-DHQ)$	Data high-order output delay time (BYTE="L")			110		70		45	ns	
$t_{PZX}(E-DHZ)$	Floating start delay time (BYTE="L")			5		5		5	ns	
$t_d(AM-E)$	Address middle-order output delay time		100		30		12		ns	
$t_d(AM-ALE)$	Address middle-order output delay time		80		24		5		ns	
$t_d(E-DLO)$	Data low-order output delay time			110		70		45	ns	
$t_{PZX}(E-DLZ)$	Floating start delay time			5		5		5	ns	
$t_d(AH-E)$	Address high-order output delay time		100		30		12		ns	
$t_d(AH-ALE)$	Address high-order output delay time		80		24		5		ns	
$t_d(\bar{A}_1-HLDA)$	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	
$t_d(BHE-E)$	BHE output delay time		100		30		20		ns	
$t_d(R/W-E)$	R/W output delay time		100		30		20		ns	
$t_d(E-\phi_1)$	$\phi_1$ output delay time		0	30	0	20	0	18	ns	
$t_h(E-AL)$	Address low-order hold time		50		25		18		ns	
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")		9		9		9		ns	
$t_h(E-DHQ)$	Data high-order hold time (BYTE="L")		50		25		18		ns	
$t_{PZX}(E-DHZ)$	Floating release delay time (BYTE="L")		50		25		18		ns	
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")		50		25		18		ns	
$t_h(ALE-AH)$	Address high-order hold time		9		9		9		ns	
$t_h(E-DLO)$	Data low-order hold time		50		25		18		ns	
$t_{PZX}(E-DLZ)$	Floating release delay time		50		25		18		ns	
$t_h(E-BHE)$	BHE hold time		18		18		18		ns	
$t_h(E-R/W)$	R/W hold time		18		18		18		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	
$t_W(E)$	E pulse width		470		220		130		ns	

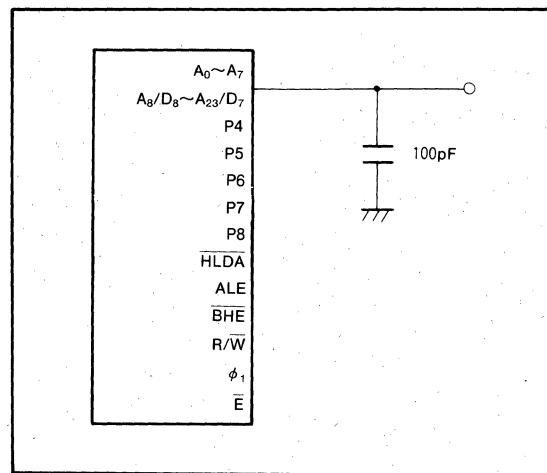
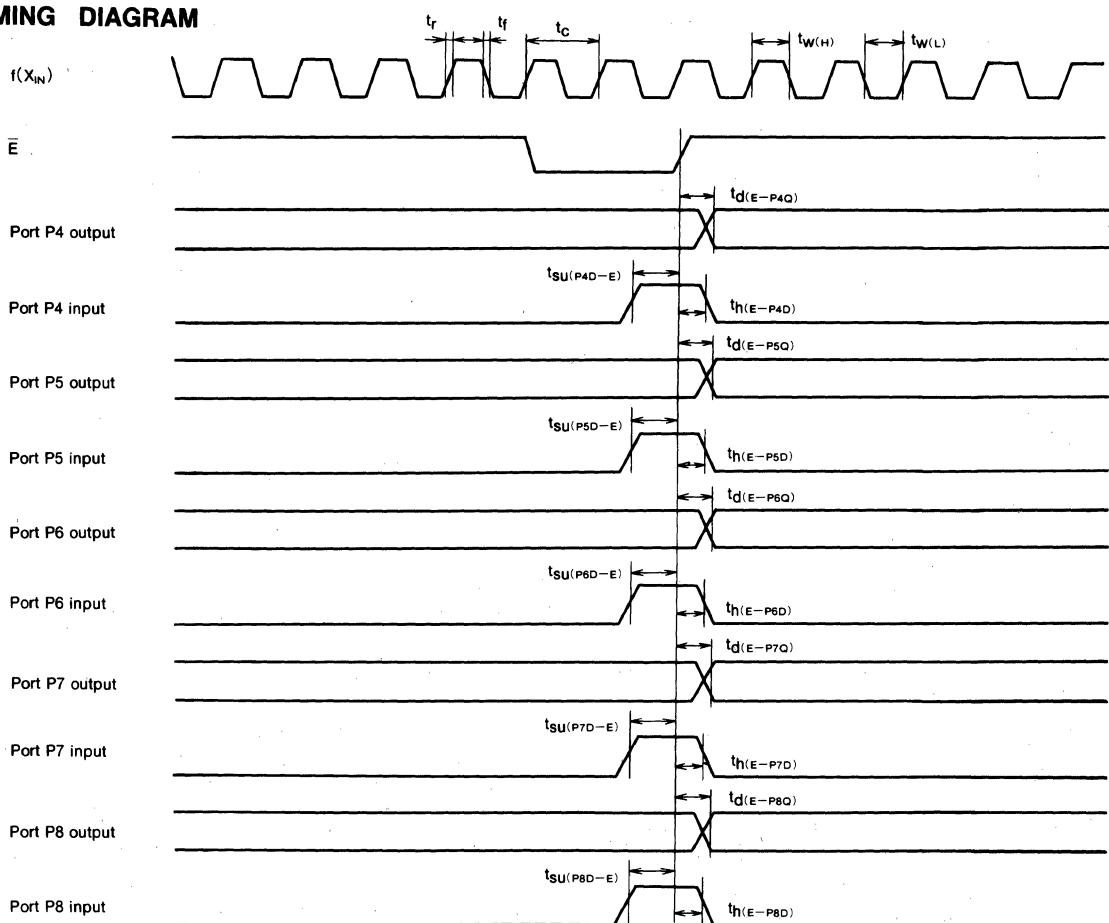


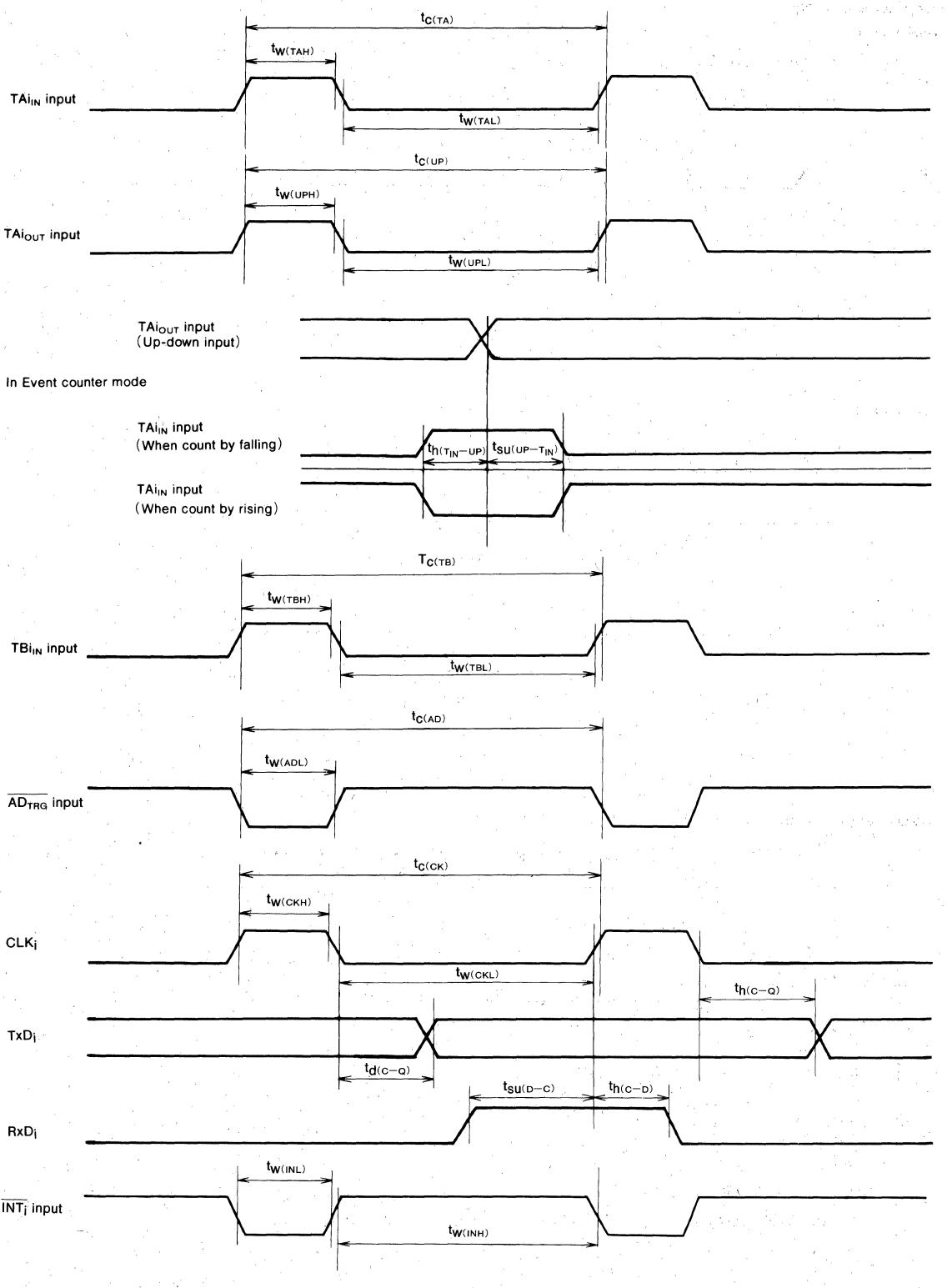
Fig. 63 Testing circuit for each terminal

MITSUBISHI MICROCOMPUTERS  
**M37732S4FP, M37732S4AFP**  
**M37732S4BFP**

**16-BIT CMOS MICROCOMPUTER**

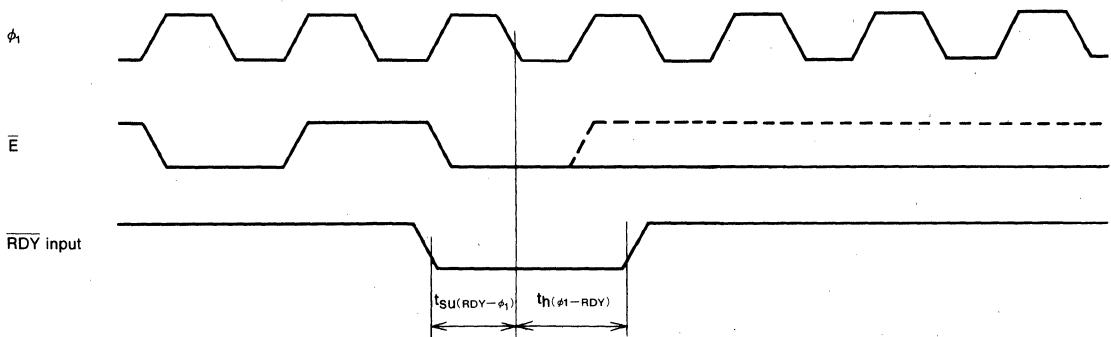
**TIMING DIAGRAM**



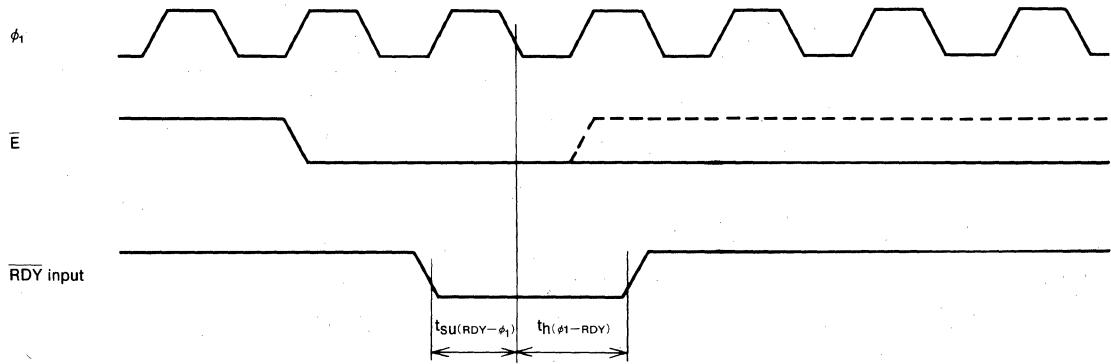
**M37732S4FP, M37732S4AFP  
M37732S4BFP**
**16-BIT CMOS MICROCOMPUTER**

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER****Microprocessor mode**

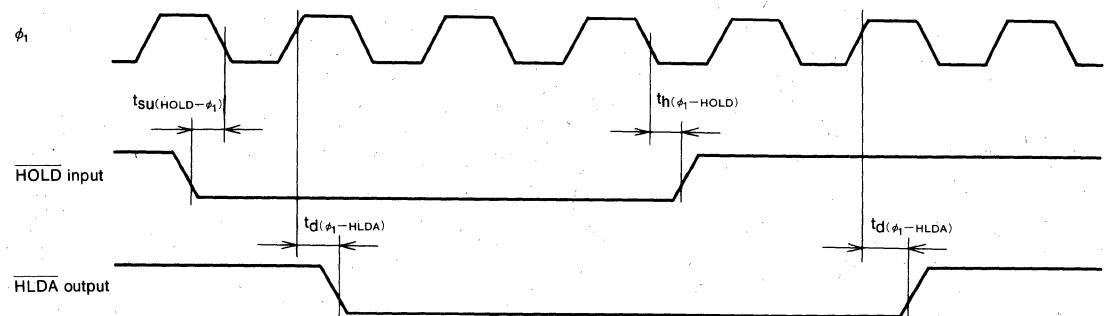
(When wait bit = "1")



(When wait bit = "0")



(When wait bit = "1" or "0" in common)

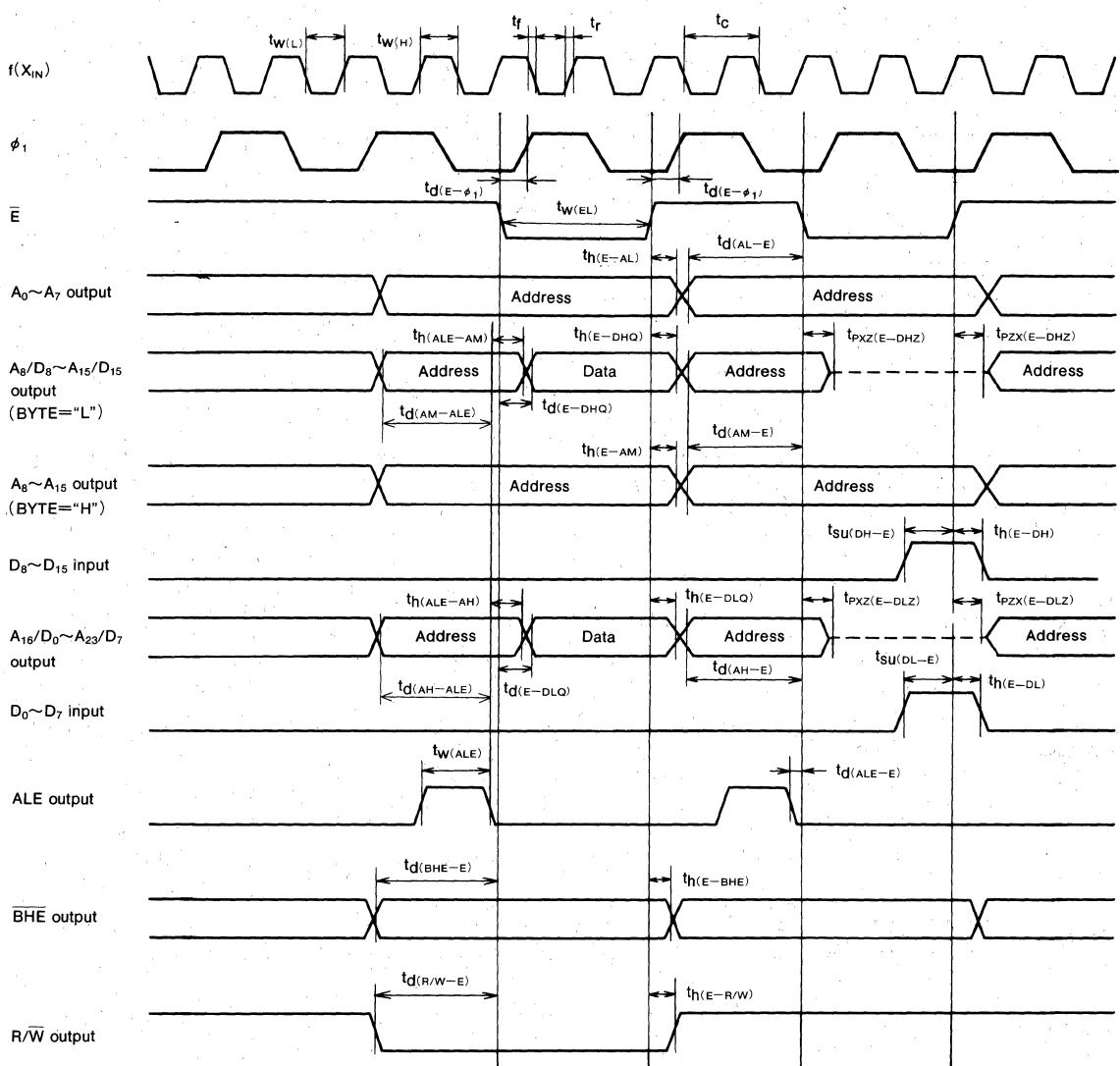
**Test conditions**

- $V_{CC} = 5V \pm 10\%$
- Input timing voltage :  $V_{IL} = 1.0V, V_{IH} = 4.0V$
- Output timing voltage :  $V_{OL} = 0.8V, V_{OH} = 2.0V$

# M37732S4FP, M37732S4AFP M37732S4BFP

## 16-BIT CMOS MICROCOMPUTER

Microprocessor mode (When wait bit = "1")

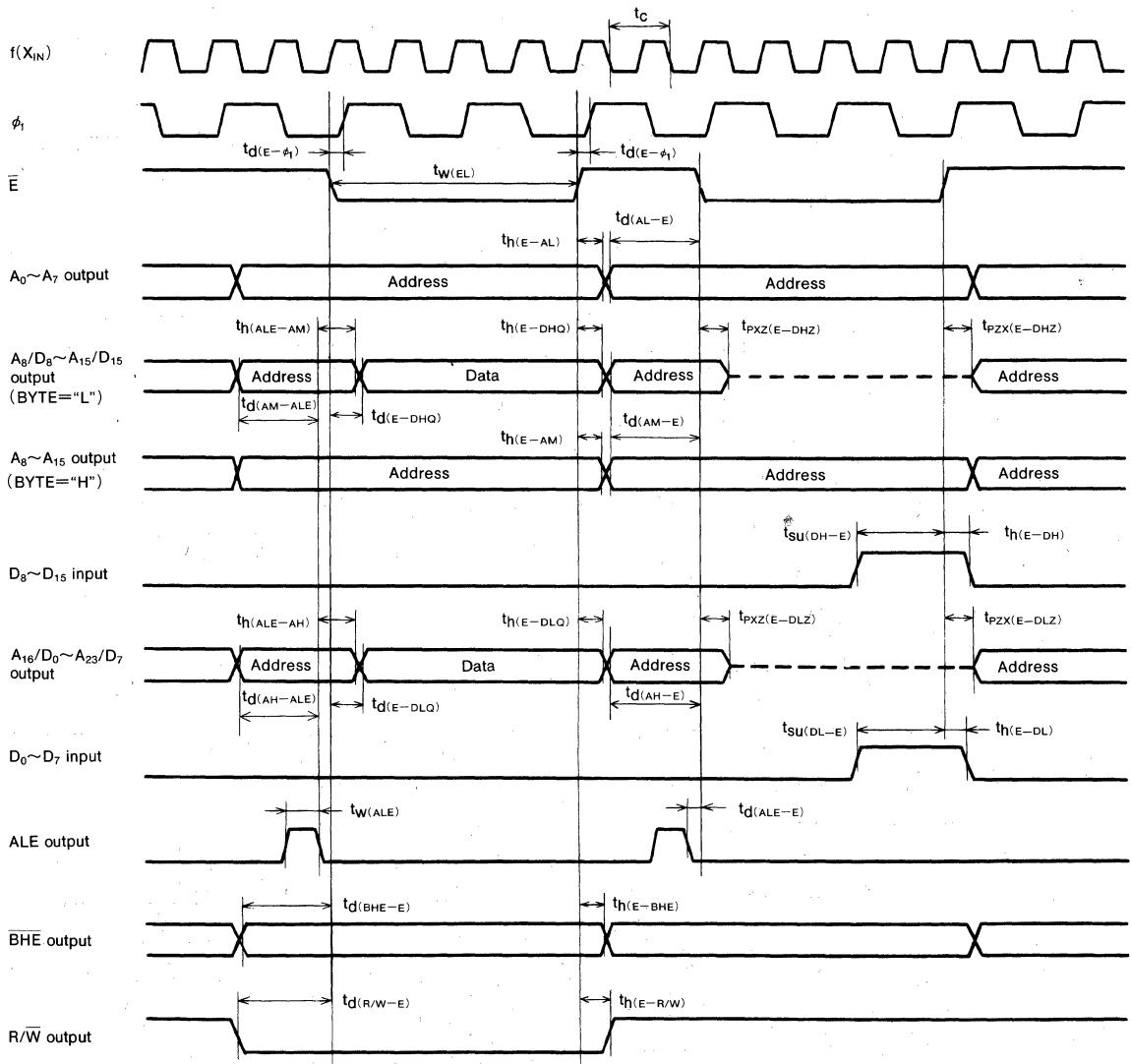


## Test conditions

- $V_{CC} = 5 \text{ V} \pm 10\%$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$
- $D_0 \sim D_{15}$  input :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.5 \text{ V}$

**M37732S4FP, M37732S4AFP  
M37732S4BFP****16-BIT CMOS MICROCOMPUTER**

Microprocessor mode (when wait bit = "0", and external memory area is accessed)



**Test conditions**

- $V_{CC} = 5 \text{ V} \pm 10\%$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$
- $D_0 \sim D_{15}$  input :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.5 \text{ V}$

# MELPS 7700

## ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

### ADDRESSING MODES

The MELPS 7700 microcomputers support 28 different addressing modes, offering extremely versatile and powerful memory accessing capability.

When executing an instruction, the address of the memory location from which the data required for arithmetic opera-

tion is to be retrieved or to which the result of arithmetic operation is to be stored must be specified address during program execution. Addressing refers to the method of specifying the memory address.

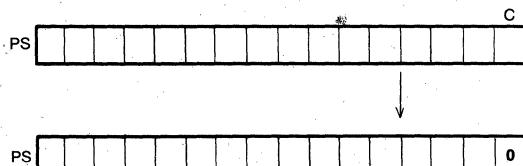
Actual addressing modes are now described by type.

**Mode : Implied addressing mode**

**Function : The single-instruction inherently address an internal register.**

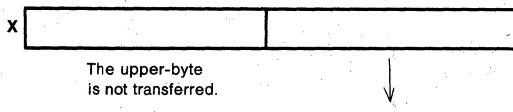
**Instruction : BRK, CLC, CLI, CLM, CLV,  
DEX, DEY, INX, INY, NOP,  
RTI, RTL, RTS, SEC, SEI,  
SEM, STP, TAD, TAS, TAX,  
TAY, TBD, TBS, TBX, TBY,  
TDA, TDB, TSA, TSB, TSX,  
TXA, TXB, TXS, TXY, TYA,  
TYB, TYX, WIT, XAB**

**ex. : Mnemonic CLC Machine code  $18_{16}$**



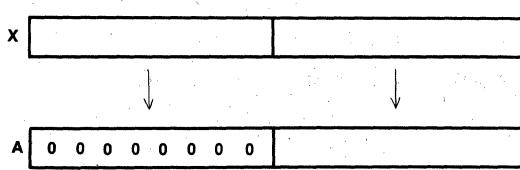
**ex. : Mnemonic TXA Machine code  $8A_{16}$**

( $m = 1, x = 0$ )



**ex. : Mnemonic TXA Machine code  $8A_{16}$**

( $m = 0, x = 1$ )



(Note) When the data length differ between the transfer-from and transfer-to locations, data is transferred at the data length for the transfer-to location. If, however, the index register is specified as the transfer-to location and the x flag is set to 1, 0016 is sent as the upper byte value.

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

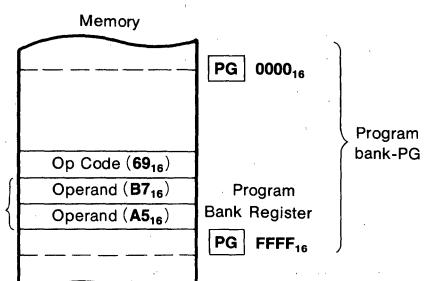
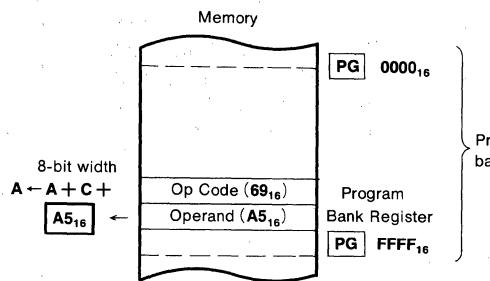
Mode : Immediate addressing mode

Function : A portion of the instruction is the actual data.  
 Such instruction code may cross over the bank boundary.

Instruction : ADC, AND, CLP, CMP, CPX,  
 CPY, DIV, EOR, LDA, LDT,  
 LDX, LDY, MPY, ORA, RLA,  
 SBC, SEP

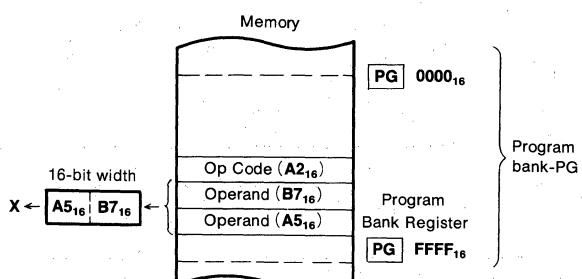
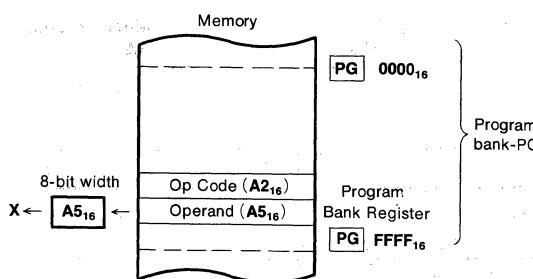
ex. : Mnemonic                      Machine code  
**ADC A,<#0A5H**                       $69_{16} \ A5_{16}$   
 $(m = 1)$

ex. : Mnemonic                      Machine code  
**ADC A,<#0A5B7H**                       $69_{16} \ B7_{16} \ A5_{16}$   
 $(m = 0)$



ex. : Mnemonic                      Machine code  
**LDX #0A5H**                       $A2_{16} \ A5_{16}$   
 $(x = 1)$

ex. : Mnemonic                      Machine code  
**LDX #0A5B7H**                       $A2_{16} \ B7_{16} \ A5_{16}$   
 $(x = 0)$



MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

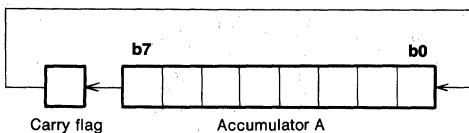
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

Mode : Accumulator addressing mode

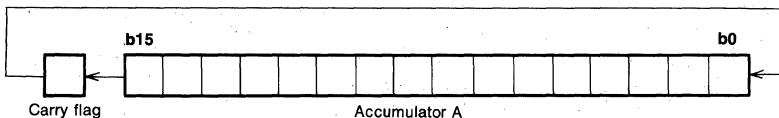
Function : The contents of accumulator are the actual data.

Instruction : ASL, DEC, INC, LSR, ROL, ROR

ex. : Mnemonic                      Machine code  
      ROL A                           $2A_{16}$   
      ( $m = 1$ )



ex. : Mnemonic                      Machine code  
      ROL A                           $2A_{16}$   
      ( $m = 0$ )



# MELPS 7700

## ADDRESSING MODES

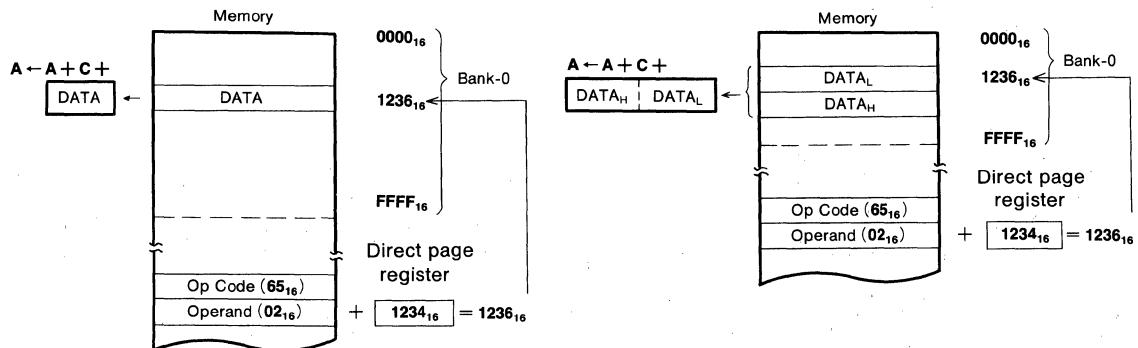
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

**Mode** : Direct addressing mode

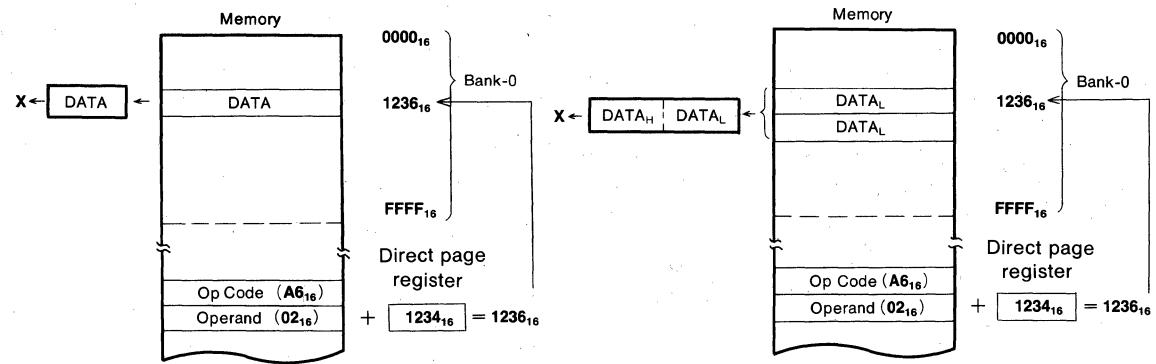
**Function** : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction to the contents of the direct page register become the actual data. If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1

**Instruction** : ADC, AND, ASL, CMP, CPX, CPY, DEC, DIV, EOR, INC, LDA, LDM, LDX, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STX, STY

<b>ex.</b>	: Mnemonic ADC A,02H (m = 1)	Machine code $65_{16} \ 02_{16}$	<b>ex.</b>	: Mnemonic ADC A,02H (m = 0)	Machine code $65_{16} \ 02_{16}$
------------	------------------------------------	-------------------------------------	------------	------------------------------------	-------------------------------------



<b>ex.</b>	: Mnemonic LDX 02H (x = 1)	Machine code $A6_{16} \ 02_{16}$	<b>ex.</b>	: Mnemonic LDX 02H (x = 0)	Machine code $A6_{16} \ 02_{16}$
------------	----------------------------------	-------------------------------------	------------	----------------------------------	-------------------------------------



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

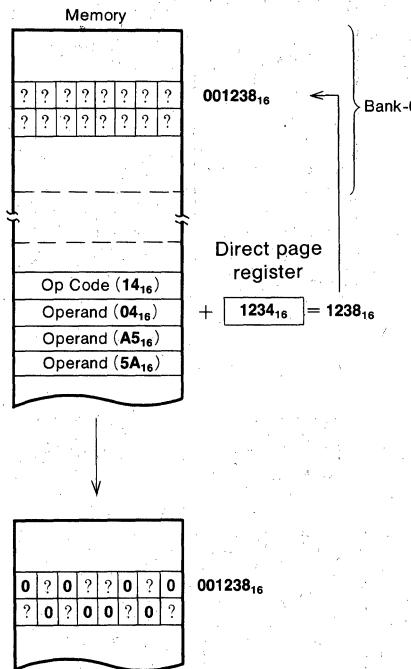
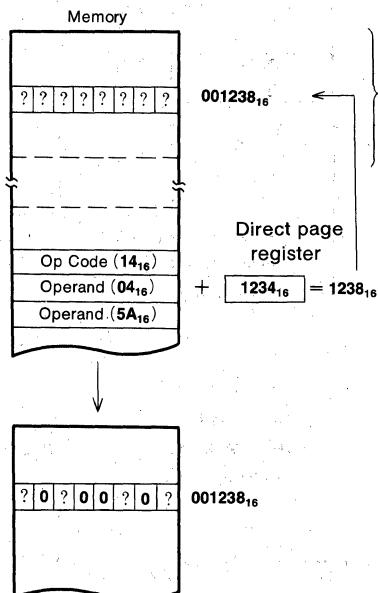
**Mode** : Direct bit addressing mode

**Function** : Specifies the bank-0 memory location by the value obtained by adding the instruction's second byte to the direct page register's contents, and specifies the position's of multiple bits in the memory location by the bit pattern in the third and fourth bytes of the instruction (third byte only when the m flag is set to 1). If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1

**Instruction** : CLB, SEB

**ex.** : Mnemonic      Machine code  
 CLB #5AH, 04H       $14_{16} \ 04_{16} \ 5A_{16}$   
 ( $m = 1$ )

**ex.** : Mnemonic      Machine code  
 CLB #5AA5H, 04H       $14_{16} \ 04_{16} \ A5_{16} \ 5A_{16}$   
 ( $m = 0$ )



# MELPS 7700

## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

**Mode** : Direct indexed X addressing mode

**Function** : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction, the contents of the direct page register and the contents of the index register X become the actual data. If, however, addition of the instruction's second byte, the direct page register's contents and the index register X's contents results in a value that exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2.

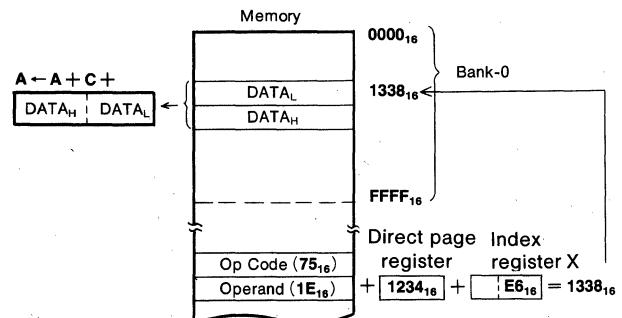
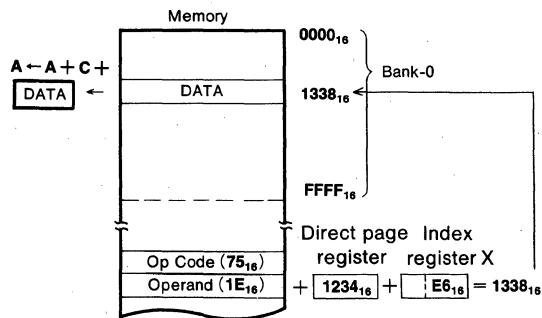
**Instruction** : ADC, AND, ASL, CMP, DEC, DIV, EOR, INC, LDA, LDM, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STY

**ex.** : Mnemonic  
**ADC A,1EH,X**  
 $(m = 1, x = 1)$

Machine code  
 $75_{16} \ 1E_{16}$

**ex.** : Mnemonic  
**ADC A,1EH,X**  
 $(m = 0, x = 1)$

Machine code  
 $75_{16} \ 1E_{16}$

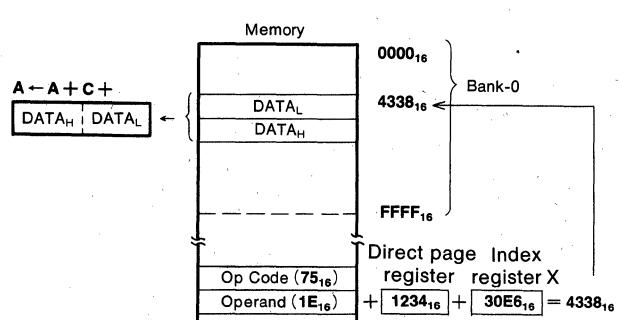
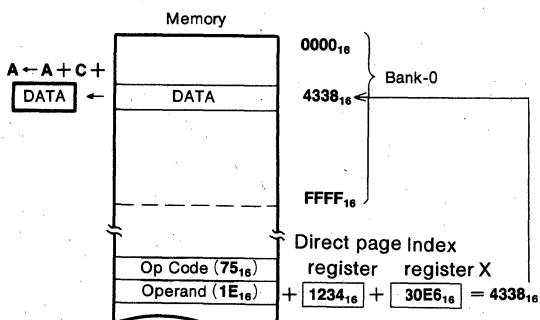


**ex.** : Mnemonic  
**ADC A,1EH,X**  
 $(m = 1, x = 0)$

Machine code  
 $75_{16} \ 1E_{16}$

**ex.** : Mnemonic  
**ADC A,1EH,X**  
 $(m = 0, x = 0)$

Machine code  
 $75_{16} \ 1E_{16}$

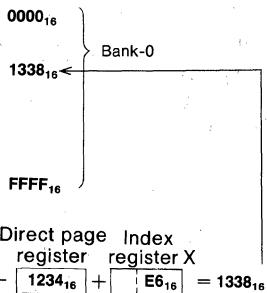
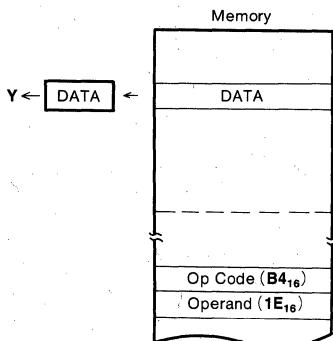


MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

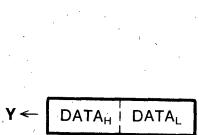
ex. : Mnemonic  
LDY 1EH,X  
(x = 1)

Machine code  
 $B4_{16} \ 1E_{16}$



ex. : Mnemonic  
LDY 1EH,X  
(x = 0)

Machine code  
 $B4_{16} \ 1E_{16}$



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

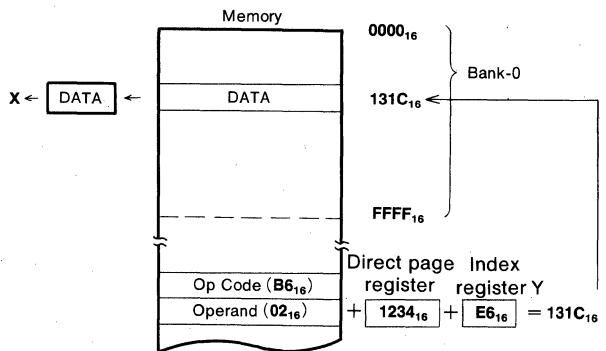
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Direct indexed Y addressing mode

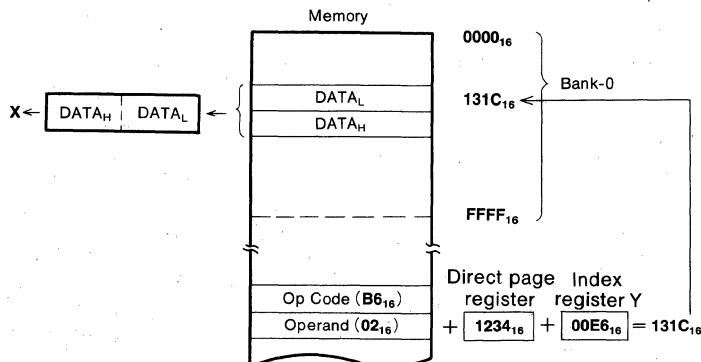
Function : The contents of the bank-0 memory location specified by the result of adding the second byte of the instruction, the contents of the direct page register and the contents of the index register Y become the actual data. If, however, addition of the instruction's second byte, the direct page register's contents and the index register Y's contents results in a value that exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2.

Instruction : LDX, STX

ex. : Mnemonic                      Machine code  
**LDX 02H,Y**                      **B6<sub>16</sub> 02<sub>16</sub>**  
(x = 1)



ex. : Mnemonic                      Machine code  
**LDX 02H,Y**                      **B6<sub>16</sub> 02<sub>16</sub>**  
(x = 0)



# **MITSUBISHI MICROCOMPUTERS**

## **MELPS 7700**

### **ADDRESSING MODES**

# **SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

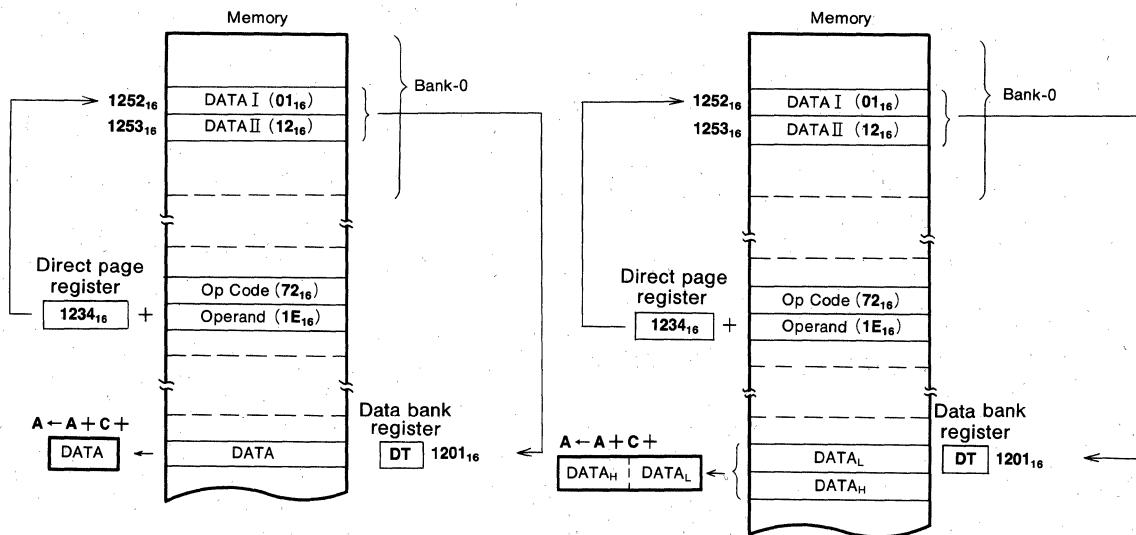
**Mode** : Direct indirect addressing mode

**Function** : The value obtained by adding the instruction's second byte to the contents of the direct page register specifies 2 adjacent bytes in memory bank-0, and the contents of these bytes in memory bank-DT (DT is contents of data bank register) become the actual data. If, however, the value obtained by adding the instruction's second byte and the direct page register's contents exceeds the bank-0 range, the specified location will be in bank-1.

**Instruction : ADC, AND, CMP, DIV, EOR,  
LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic Machine code  
**ADC A,(1EH)** **72<sub>16</sub> 1E<sub>16</sub>**  
 (m = 1)

ex. : Mnemonic Machine code  
**ADC A, (1EH)** **72<sub>16</sub> 1E<sub>16</sub>**  
           ( $m = 0$ )



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

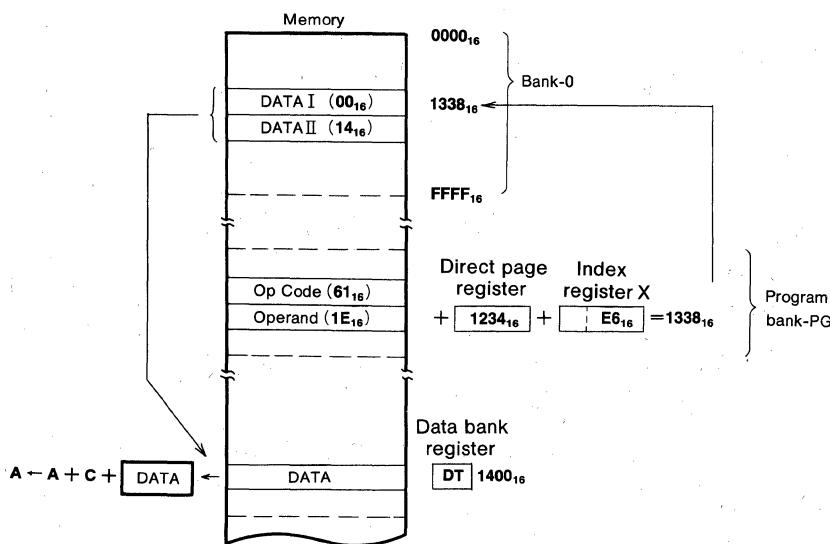
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Direct indexed X indirect addressing mode

Function : The value obtained by adding the instruction's second byte, the contents of the direct page register and the contents of the index register X specifies 2 adjacent bytes in memory bank-0, and the contents of these bytes in memory bank-0, and the contents of these bytes in memory bank-DT (DT is contents of data bank register) become the actual data. If, however, the value obtained by adding the instruction's second byte, the direct page register's contents and the index register X's contents exceeds the bank-0 or bank-1 range, the specified location will be in bank-1 or bank-2

ex. : Mnemonic                      Machine code  
 ADC A, (1EH, X)  
 $61_{16} \ 1E_{16}$   
 $(m = 1, x = 1)$

Instruction : ADC, AND, CMP, DIV, EOR,  
 LDA, MPY, ORA, SBC, STA

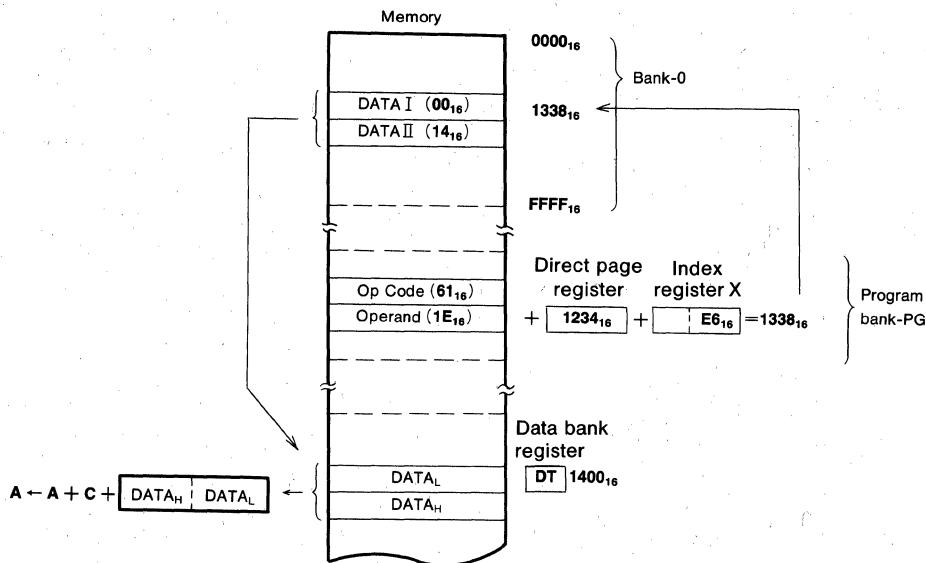


MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

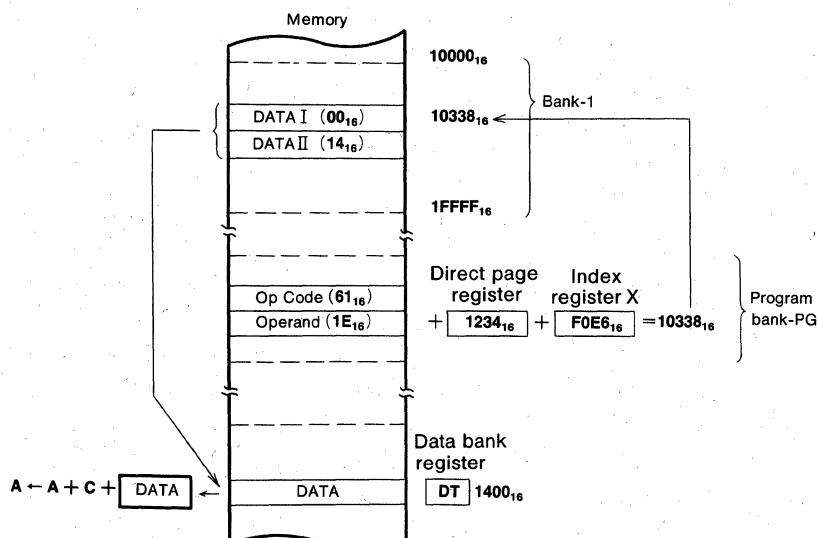
ex. : Mnemonic  
ADC A, (1EH, X)  
(m = 0, x = 1)

Machine code  
 $61_{16} \ 1E_{16}$



ex. : Mnemonic  
ADC A, (1EH, X)  
(m = 1, x = 0)

Machine code  
 $61_{16} \ 1E_{16}$

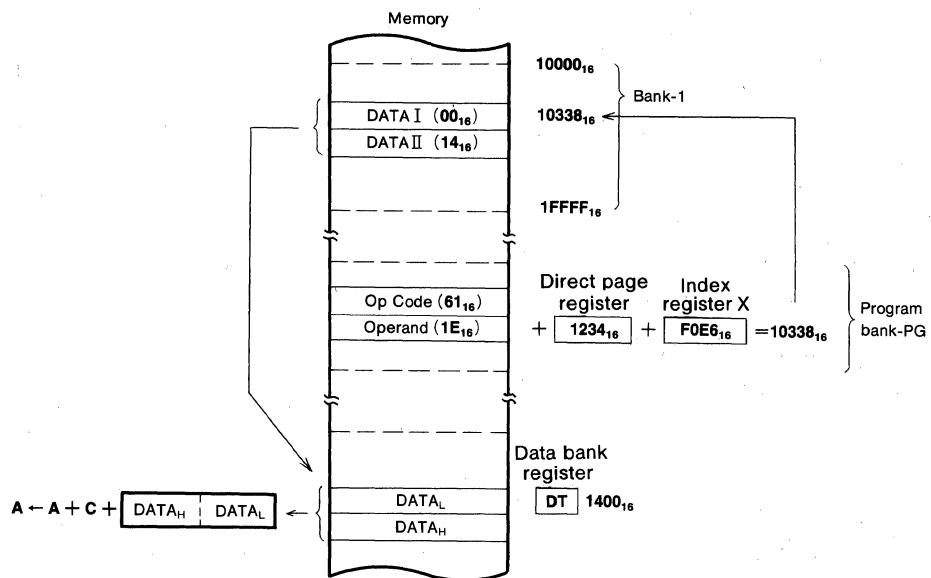


MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic  
ADC A, (1EH, X)  
(m = 0, x = 0 )

Machine code  
 $61_{16} \ 1E_{16}$

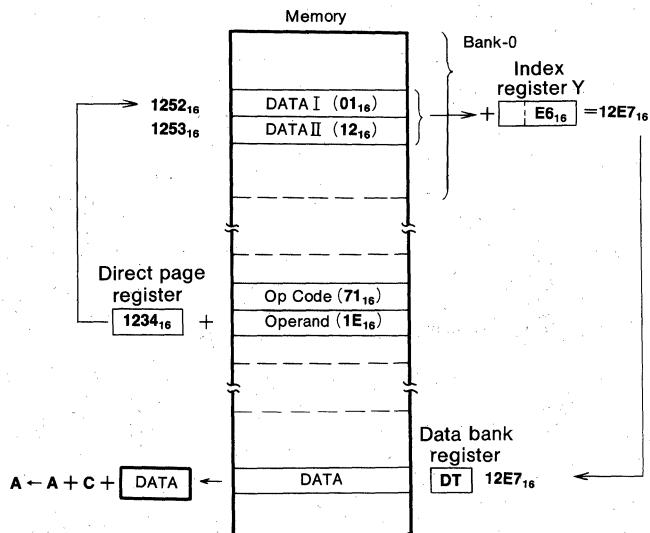


**Mode** : Direct indirect indexed Y addressing mode

**Function** : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 2 adjacent bytes in memory bank-0. The value obtained by adding the contents of these bytes and the contents of the index register Y specifies address of the actual data in memory bank-DT (DT is contents of data bank register). If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. Also, if addition of the contents of memory and index register Y generate a carry, the bank number will be 1 larger than the contents of the data bank register.

**Instruction** : ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA

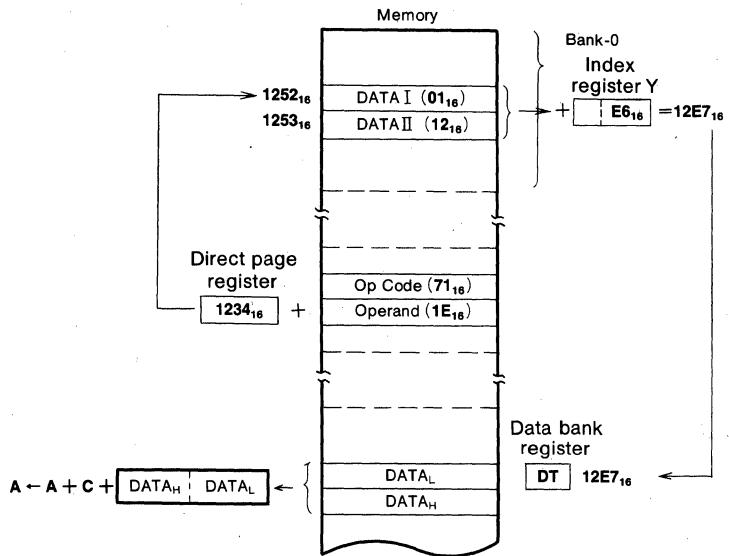
**ex.** : Mnemonic                      Machine code  
**ADC A, (1EH),Y**                  **71<sub>16</sub> 1E<sub>16</sub>**  
 $(m = 1, x = 1)$



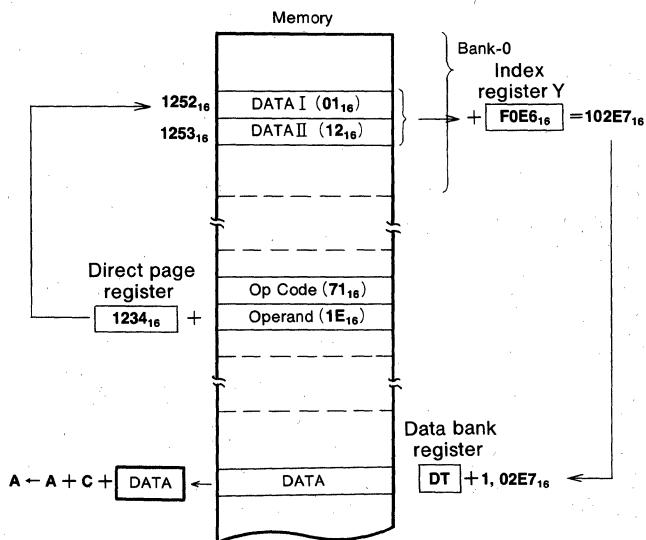
MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic                      Machine code  
ADC A, (1EH), Y                    71<sub>16</sub> 1E<sub>16</sub>  
(m = 0, x = 1)



ex. : Mnemonic                      Machine code  
ADC A, (1EH), Y                    71<sub>16</sub> 1E<sub>16</sub>  
(m = 1, x = 0)



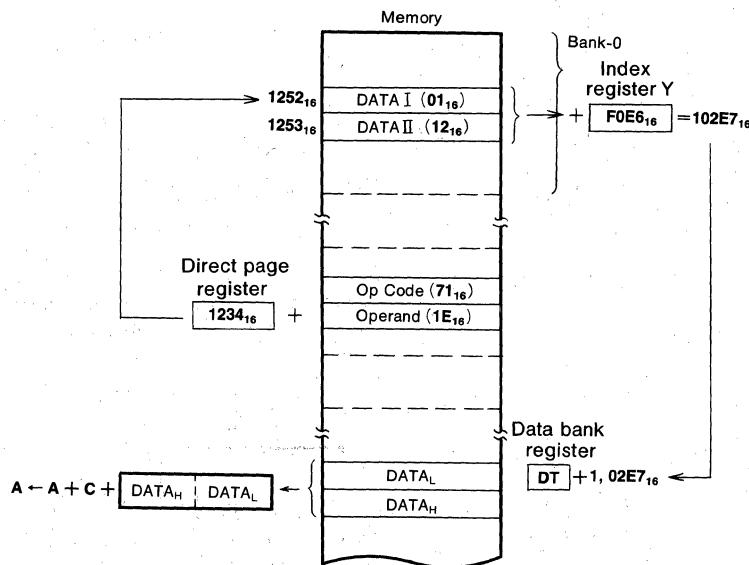
# **SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

**ex.** : Mnemonic

**ADC A, (1EH), Y**  
**(m = 0, x = 0)**

### Machine code

Machine 3



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

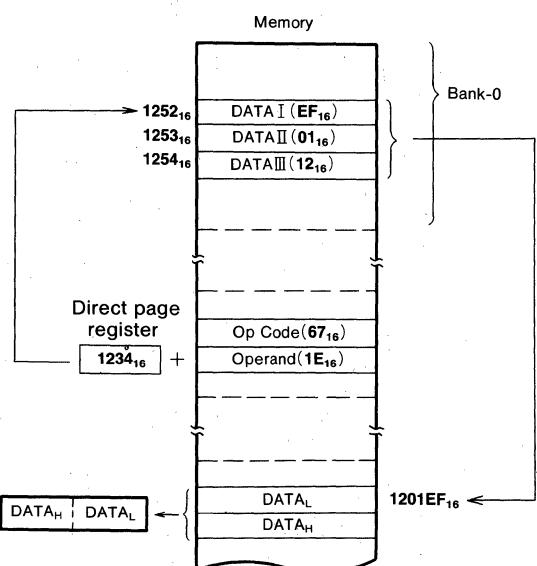
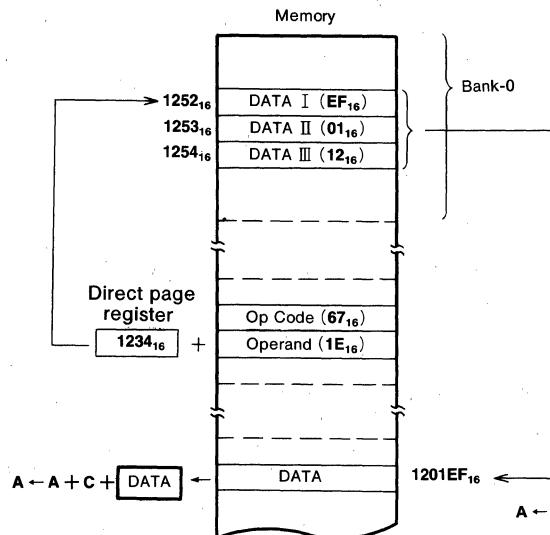
Mode : Direct indirect long addressing mode

Function : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 3 adjacent bytes in memory bank-0, and the contents of these bytes specify the address of the memory location that contains the actual data. If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. The 3 adjacent bytes memory location may be spread over two different banks.

Instruction : ADC, AND, CMP, DIV, EOR,  
 LDA, MPY, ORA, SBC, STA

ex. : Mnemonic                                      Machine code  
**ADCL A, (1EH)**                              **67<sub>16</sub> 1E<sub>16</sub>**  
 (m=1)

ex. : Mnemonic                                      Machine code  
**ADCL A, (1EH)**                              **67<sub>16</sub> 1E<sub>16</sub>**  
 (m=0)



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

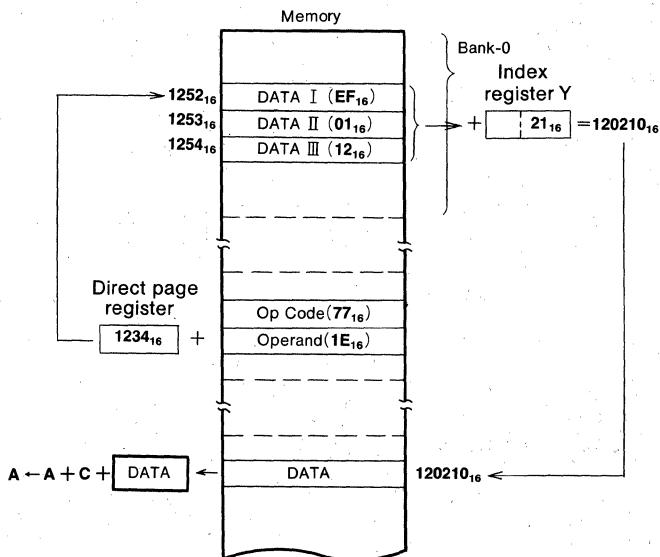
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Direct indirect long indexed Y addressing mode

Function : The value obtained by adding the instruction's second byte and the contents of the direct page register specifies 3 adjacent bytes in memory bank-0, and the value obtained by adding the contents of these bytes and the contents of the index register Y specifies the address of the memory location where the actual data is stored. If, however, the value obtained by adding the contents of the instruction's second byte and the direct page register exceeds the bank-0 range, the specified location will be in bank-1. The 3 adjacent bytes memory location may be spread over two different banks.

Instruction : ADC, AND, CMP, DIV, EOR,  
 LDA, MPY, ORA, SBC, STA

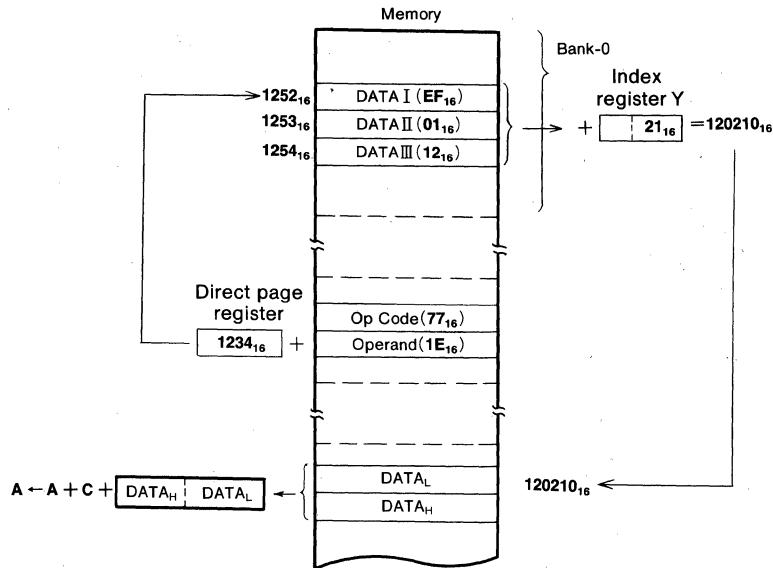
ex. : Mnemonic                      Machine code  
**ADCL A,(1EH), Y**              **77<sub>16</sub> 1E<sub>16</sub>**  
 (m=1, x=1)



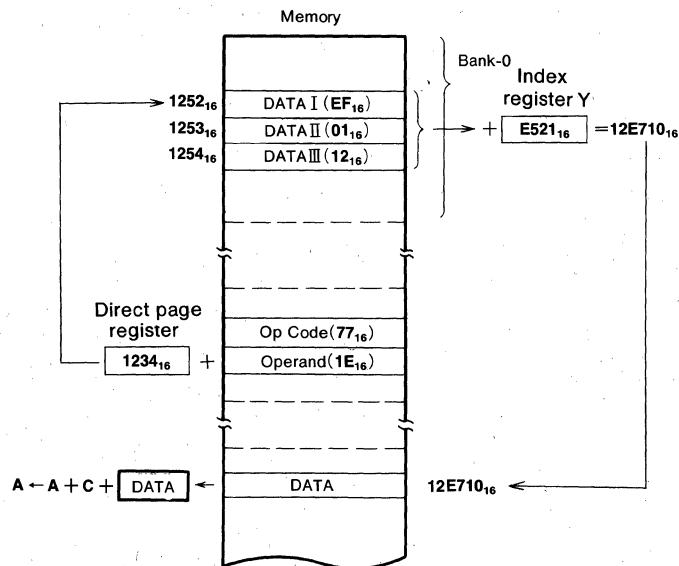
MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic                      Machine code  
ADCL A,(1EH), Y                  77<sub>16</sub> 1E<sub>16</sub>  
(m=0, x=1)



ex. : Mnemonic                      Machine code  
ADCL A,(1EH), Y                  77<sub>16</sub> 1E<sub>16</sub>  
(m=1, x=0)



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

ex.

: Mnemonic  
**ADCL A,(1EH), Y**  
 $(m=0, x=0)$

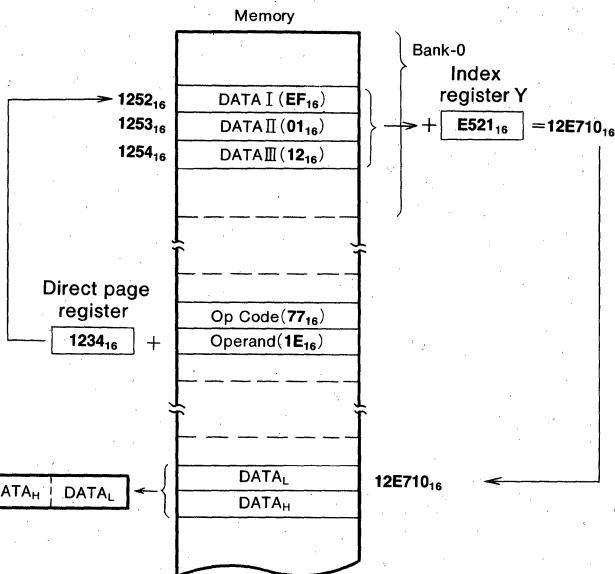
Machine code

$77_{16} \ 1E_{16}$

$A \leftarrow$

$A + C +$

$[DATA_H \quad DATA_L]$



# MELPS 7700

## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

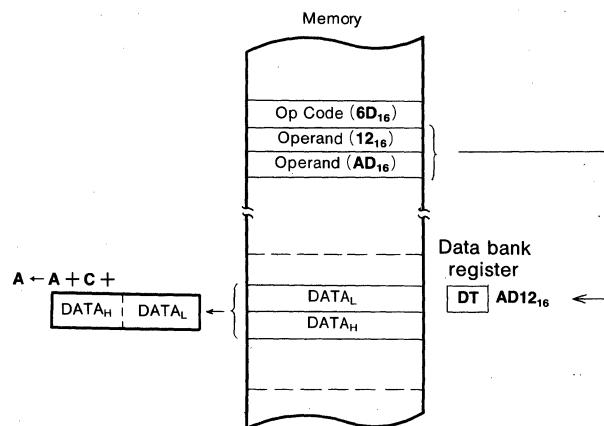
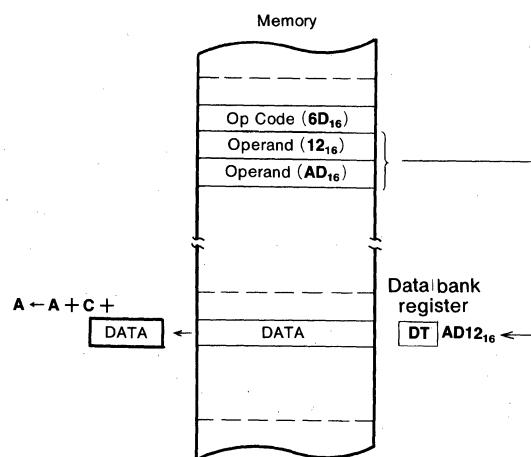
**Mode** : Absolute addressing mode

**Function** : The contents of the memory locations specified by the instruction's second and third bytes and the contents of the data bank register are the actual data. Note that, in the cases of the JMP and JSR instructions, the instructions' second and third byte contents are transferred to the program counter.

**Instruction** : ADC, AND, ASL, CMP, CPX, CPY, DEC, DIV, EOR, INC, JMP, JSR, LDA, LDM, LDX, LDY, LSR, MPY, ORA, ROL, ROR, SBC, STA, STX, STY

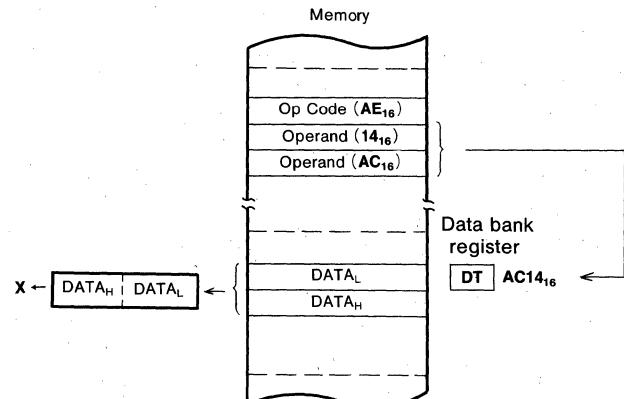
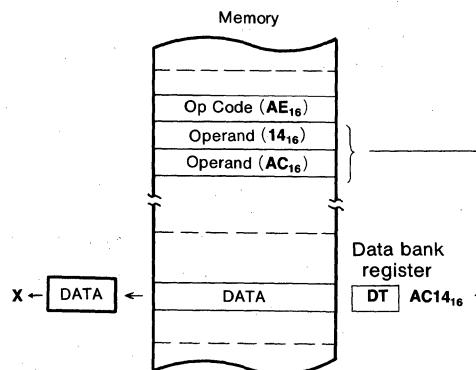
**ex.** : Mnemonic  
ADC A, 0AD12H  
(m=1)  
Machine code  
 $6D_{16} 12_{16} AD_{16}$

**ex.** : Mnemonic  
ADC A, 0AD12H  
(m=0)  
Machine code  
 $6D_{16} 12_{16} AD_{16}$



**ex.** : Mnemonic  
LDX 0AC14H  
(x=1)  
Machine code  
 $AE_{16} 14_{16} AC_{16}$

**ex.** : Mnemonic  
LDX 0AC14H  
(x=0)  
Machine code  
 $AE_{16} 14_{16} AC_{16}$



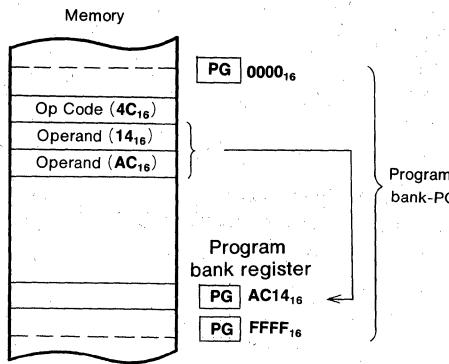
MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic  
**JMP 0AC14H**

Machine code  
**4C<sub>16</sub> 14<sub>16</sub> AC<sub>16</sub>**

Address to be  
executed next.



Program bank register contents are not affected.

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Absolute bit addressing mode

Function : The contents of the instruction's second and third bytes and the contents of the data bank register specify the memory locations, and data for multiple bit positions in the memory locations are specified by a bit pattern specified in the instruction's fourth and fifth bytes (the fourth byte only if the m flag is set to 1).

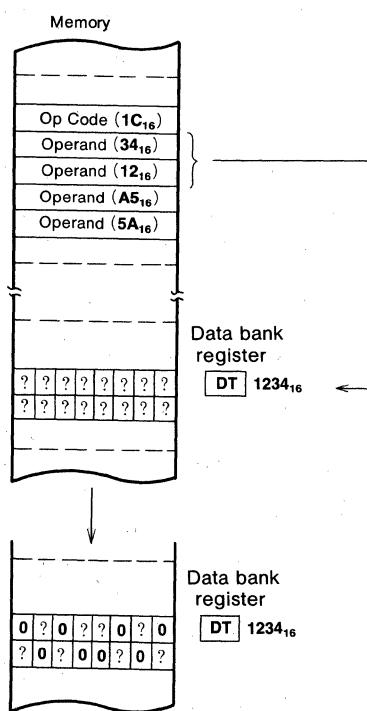
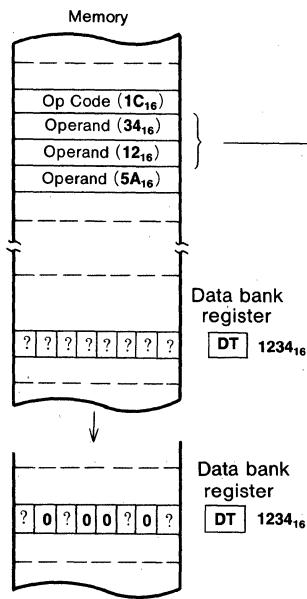
Instruction : CLB, SEB

ex. : Mnemonic  
**CLB #5AH, 1234H**  
 $(m=1)$

Machine code  
 $1C_{16} 34_{16} 12_{16} 5A_{16}$

ex. : Mnemonic  
**CLB #5AA5H, 1234H**  
 $(m=0)$

Machine code  
 $1C_{16} 34_{16} 12_{16} A5_{16} 5A_{16}$



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Absolute indexed X addressing mode

Function : The contents of the memory locations specified by a value resulting from addition of a 16-bit numeric value expressed by the instruction's second and third bytes with the contents of the index register X and the contents of the data bank register are the actual data. If, however, addition of the numeric value expressed by the instruction's second and third bytes with the contents of the index register X generates a carry, the bank number will be 1 larger than the contents of the data bank register.

ex.

: Mnemonic

**ADC A,0AD12H,X**  
 $(m=1, x=1)$

: Machine code

**7D<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**

Instruction : **ADC ,AND ASL, CMP, DEC,  
DIV, EOR, INC, LDA, LDM,  
LDY, LSR, MPY, ORA, ROL,  
ROR, SBC, STA**

: Mnemonic

**ADC A,0AD12H,X**  
 $(m=0, x=1)$

: Machine code

**7D<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**

$$A \leftarrow A + C +$$

**DATA**

**DATA**

Memory  
Op Code (7D<sub>16</sub>)  
Operand (12<sub>16</sub>)  
Operand (AD<sub>16</sub>)

$$\} + \boxed{EE_{16}} = AE00_{16}$$

Data bank register  
DT AE00<sub>16</sub>

$$A \leftarrow A + C +$$

**DATA<sub>H</sub>**

**DATA<sub>L</sub>**

Memory

Op Code (7D<sub>16</sub>)  
Operand (12<sub>16</sub>)  
Operand (AD<sub>16</sub>)

$$\} + \boxed{EE_{16}} = AE00_{16}$$

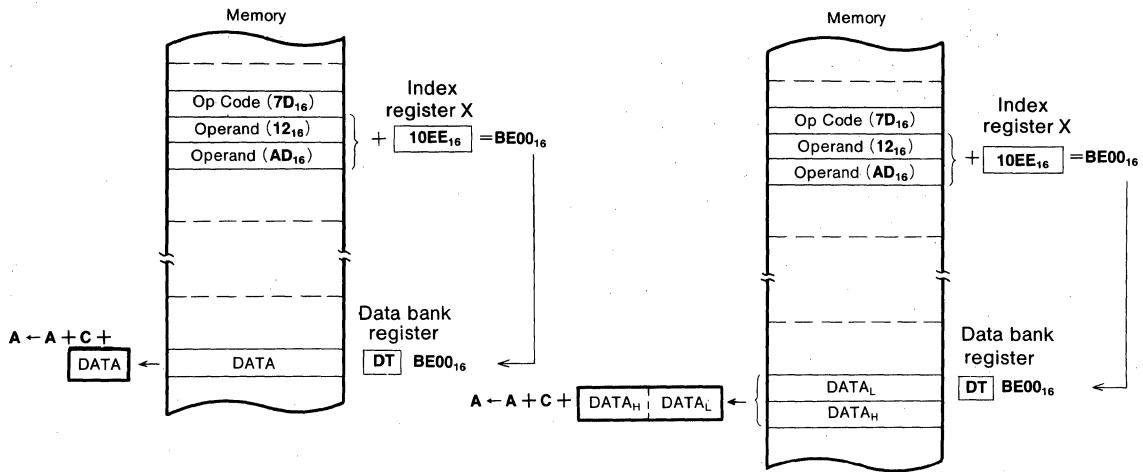
Memory  
Data bank register  
DT AE00<sub>16</sub>

# MELPS 7700

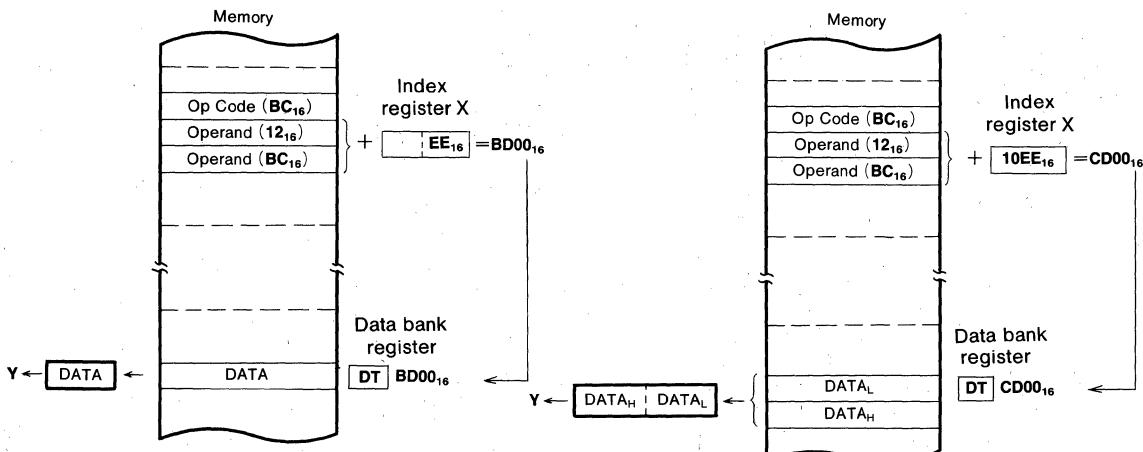
## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex.	: Mnemonic ADC A, 0AD12H, X (m=1, x=0)	Machine code 7D <sub>16</sub> 12 <sub>16</sub> AD <sub>16</sub>	ex.	: Mnemonic ADC A, 0AD12H, X (m=0, x=0)	Machine code 7D <sub>16</sub> 12 <sub>16</sub> AD <sub>16</sub>
-----	--	--	-----	--	--



ex.	: Mnemonic LDY 0BC12H, X (x=1)	Machine code BC <sub>16</sub> 12 <sub>16</sub> BC <sub>16</sub>	ex.	: Mnemonic LDY 0BC12H, X (x=0)	Machine code BC <sub>16</sub> 12 <sub>16</sub> BC <sub>16</sub>
-----	--------------------------------------	--	-----	--------------------------------------	--



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

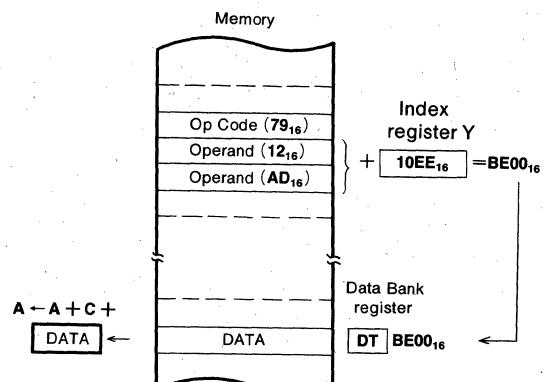
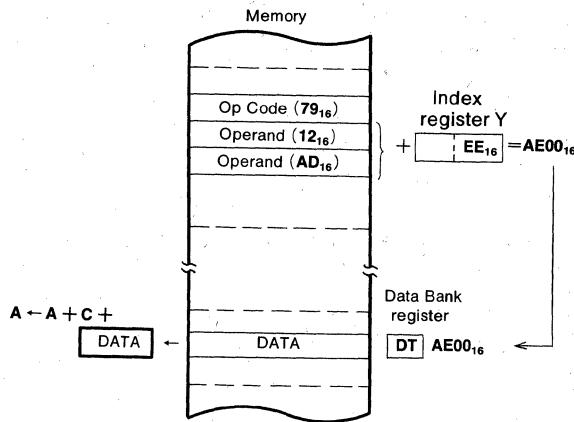
Mode : Absolute indexed Y addressing mode

Function : The contents of the memory locations specified by a value resulting from addition of a 16-bit numeric value expressed by the instruction's second and third bytes with the contents of the index register Y and the contents of the data bank register are the actual data. If, however, addition of the numeric value expressed by the instruction's second and third bytes with the contents of the index register Y generates a carry, the bank number will be 1 larger than the contents of the data bank register.

Instruction : ADC, AND, CMP, DIV, EOR,  
 LDA, LDX, MPY, ORA, SBC,  
 STA

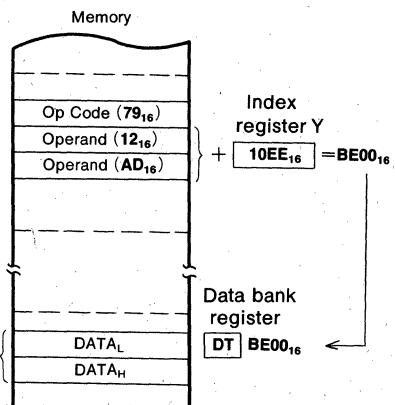
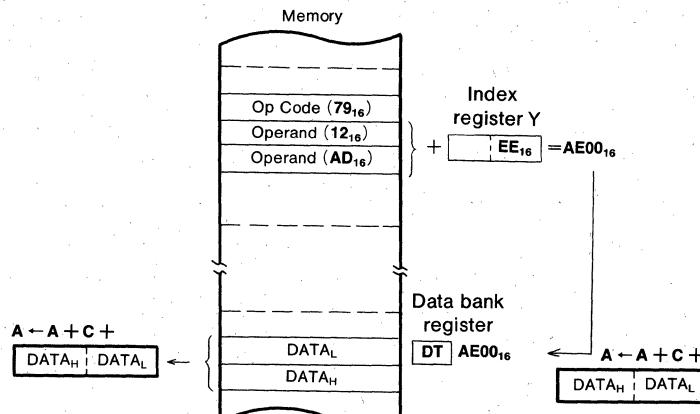
ex. : Mnemonic                      Machine code  
**ADC A, 0AD12H, Y**            **79<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**  
 (m=1, x=1)

ex. : Mnemonic                      Machine code  
**ADC A, 0AD12H, Y**            **79<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**  
 (m=1, x=0)



ex. : Mnemonic                      Machine code  
**ADC A, 0AD12H, Y**            **79<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**  
 (m=0, x=1)

ex. : Mnemonic                      Machine code  
**ADC A, 0AD12H, Y**            **79<sub>16</sub> 12<sub>16</sub> AD<sub>16</sub>**  
 (m=0, x=0)

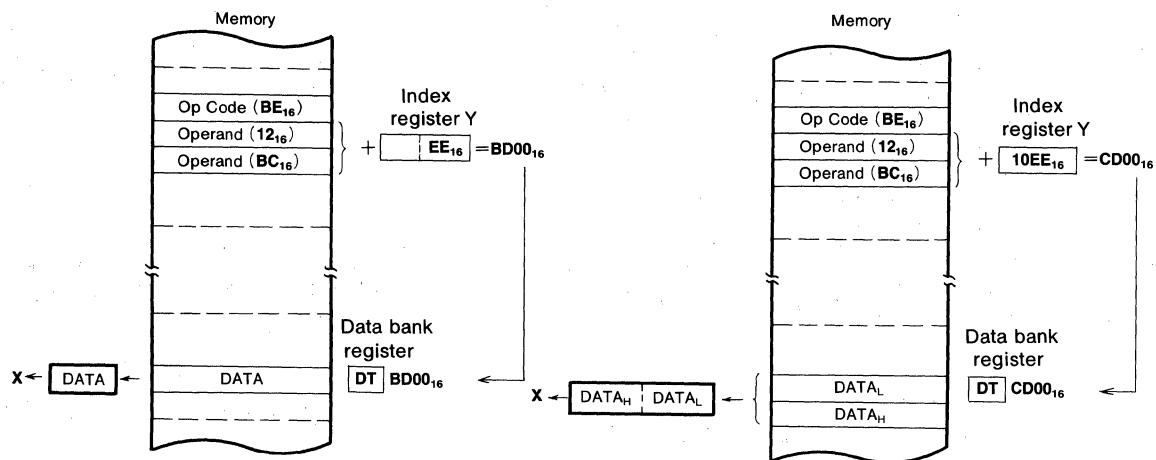


# MELPS 7700

## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex.	Mnemonic LDX 0BC12H, Y (x=1)	Machine code $BE_{16} 12_{16} BC_{16}$	ex.	Mnemonic LDX 0BC12H, Y (x=0)	Machine code $BE_{16} 12_{16} BC_{16}$
-----	------------------------------------	---	-----	------------------------------------	---



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

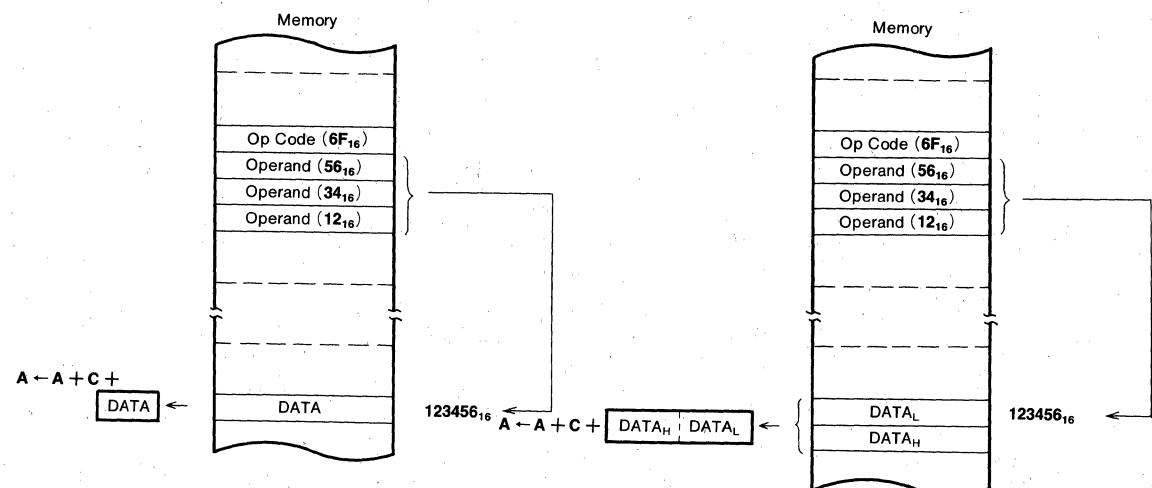
Mode : Absolute long addressing mode

Function : The contents of the memory locations specified by the instruction's second, third and fourth bytes become the actual data. Note that, in the cases of the JMP and JSR instructions, the instructions' second and third byte contents are transferred to the program counter and the fourth byte contents are transferred to the program bank register.

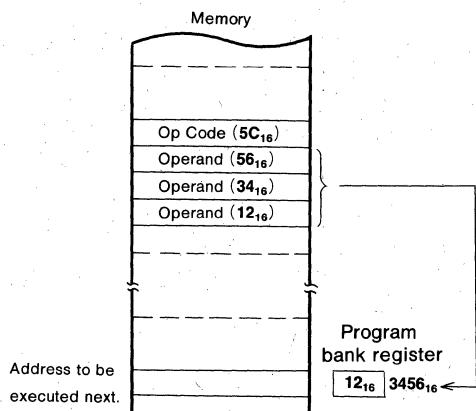
ex. : Mnemonic                      Machine code  
**ADC A, 123456H**                 $6F_{16} 56_{16} 34_{16} 12_{16}$   
 $(m=1)$

Instruction : **ADC, AND, CMP, DIV, EOR, JMP, JSR, LDA, MPY, ORA, SBC, STA**

ex. : Mnemonic                      Machine code  
**ADC A, 123456H**                 $6F_{16} 56_{16} 34_{16} 12_{16}$   
 $(m=0)$



ex. : Mnemonic                      Machine code  
**JMP 123456H**                 $5C_{16} 56_{16} 34_{16} 12_{16}$



Program bank register contents are replaced by the third operand.

# MELPS 7700

## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

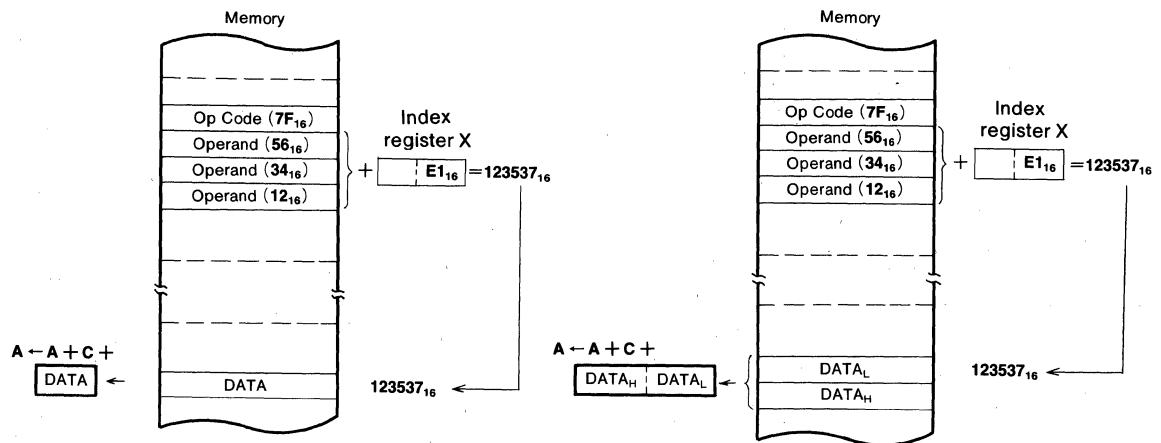
**Mode** : Absolute long indexed X addressing mode

**Function** : The contents of the memory location specified by adding the numeric value expressed by the instruction's second, third and fourth bytes with the contents of the index register X are the actual data.

**ex.** : Mnemonic                          Machine code  
**ADC A, 123456H, X**                     $7F_{16} \ 56_{16} \ 34_{16} \ 12_{16}$   
 $(m=1, x=1)$

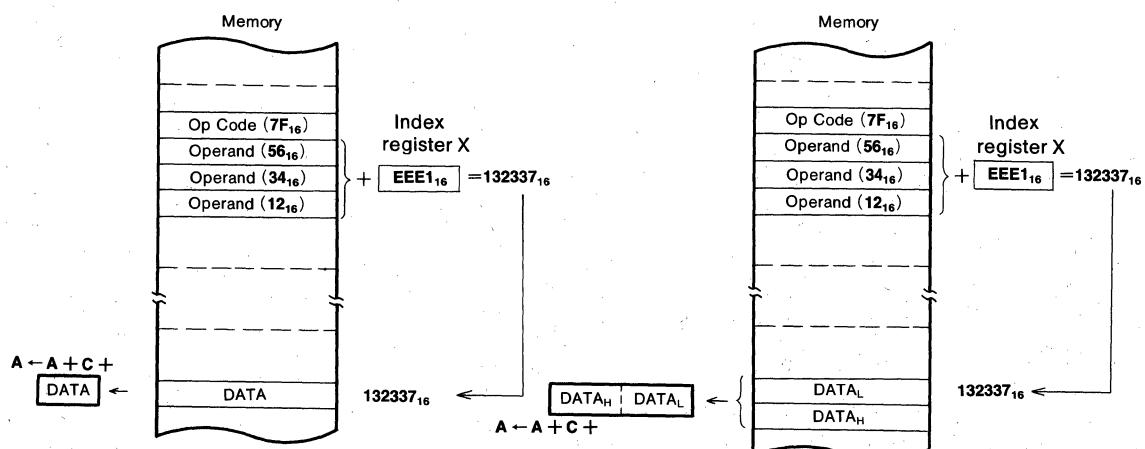
**Instruction** : ADC, AND, CMP, DIV, EOR,  
LDA, MPY, ORA, SBC, STA

**ex.** : Mnemonic                          Machine code  
**ADC A, 123456H, X**                     $7F_{16} \ 56_{16} \ 34_{16} \ 12_{16}$   
 $(m=0, x=1)$



**ex.** : Mnemonic                          Machine code  
**ADC A, 123456H, X**                     $7F_{16} \ 56_{16} \ 34_{16} \ 12_{16}$   
 $(m=0, x=1)$

**ex.** : Mnemonic                          Machine code  
**ADC A, 123456H, X**                     $7F_{16} \ 56_{16} \ 34_{16} \ 12_{16}$   
 $(m=0, x=0)$



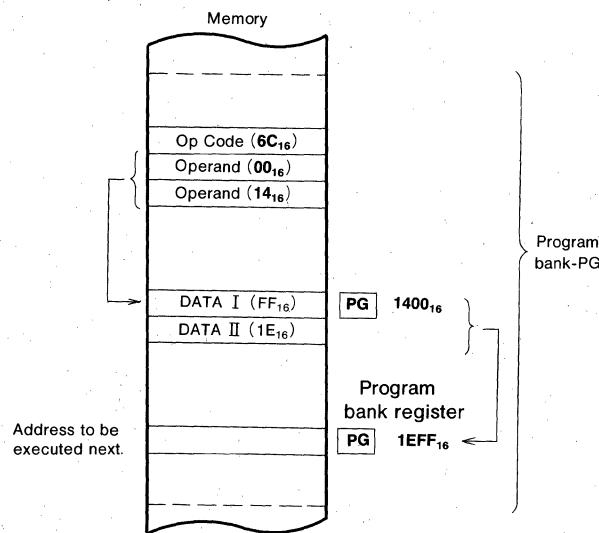
**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

**Mode** : Absolute indirect addressing mode

**Function** : The instruction's second and third bytes specify 2 adjacent bytes in memory, and the contents of these bytes specify the address within the same program bank to which a jump is to be made.

**ex.** :: Mnemonic                      Machine code  
**JMP(1400H)**                       $6C_{16} \ 00_{16} \ 14_{16}$



**Instruction : JMP**

MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

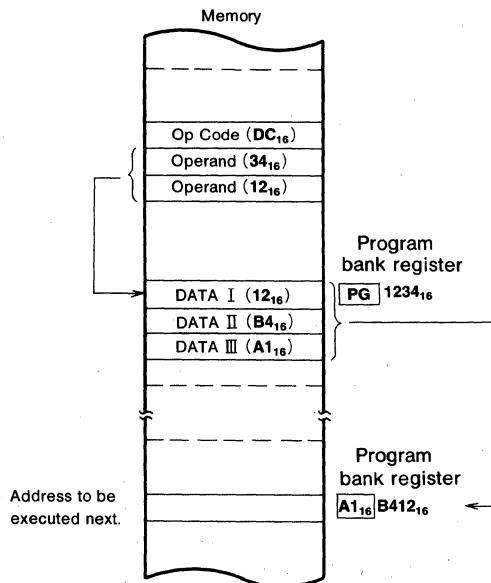
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

Mode : Absolute indirect long addressing mode

Function : The instruction's second and third bytes specify 3 adjacent bytes in memory, and the contents of these bytes specify the address to which a jump is to be made.

Instruction : JMP

ex. : Mnemonic                      Machine code  
JMPL(1234H)                      DC<sub>16</sub> 34<sub>16</sub> 12<sub>16</sub>



DATA III is loaded in the program bank register.

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

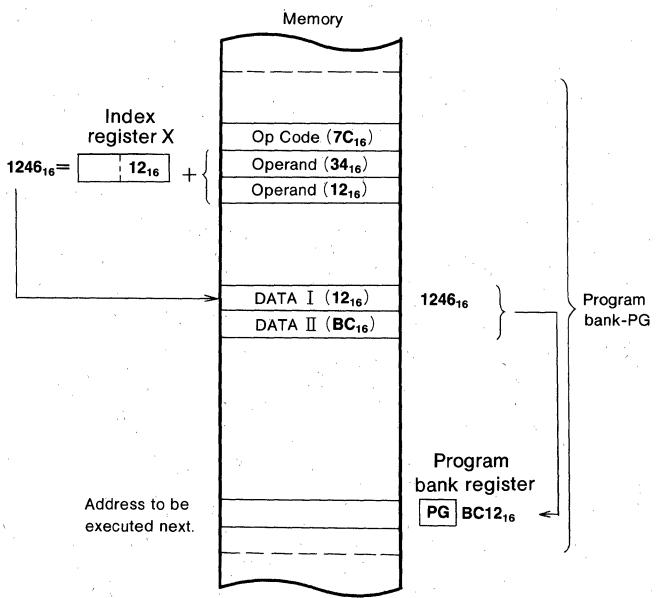
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Absolute indexed X indirect addressing mode

Function : The value obtained by adding the instruction's second and third bytes and the contents of the index register X specifies 2 adjacent bytes in memory, and the contents of these bytes specify the address to which a jump is to be made.

ex. : Mnemonic                      Machine code  
**JMP(1234H, X)**                  **7C<sub>16</sub> 34<sub>16</sub> 12<sub>16</sub>**  
(x=1)

Instruction : **JMP, JSR**



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Stack addressing mode

Function : Register contents are saved to or restored from the memory location specified by the stack pointer. The stack pointer is set in bank-0.

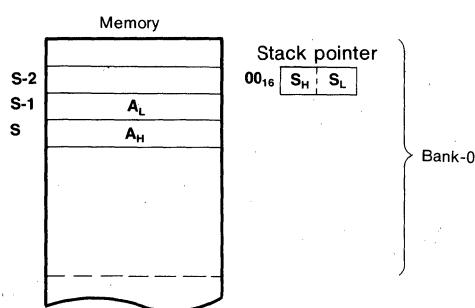
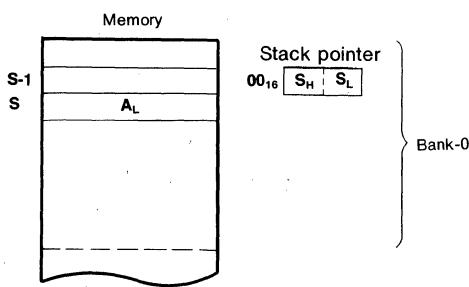
Instruction : PEA, PEI, PER, PHA, PHB,  
 PHD, PHG, PHP, PHT, PHX,  
 PHY, PLA, PLB, PLD, PLP,  
 PLT, PLX, PLY, PSH, PUL

ex. : Mnemonic  
**PHA**  
 (m=1)

Machine code  
 $48_{16}$

ex. : Mnemonic  
**PHA**  
 (m=0)

Machine code  
 $48_{16}$

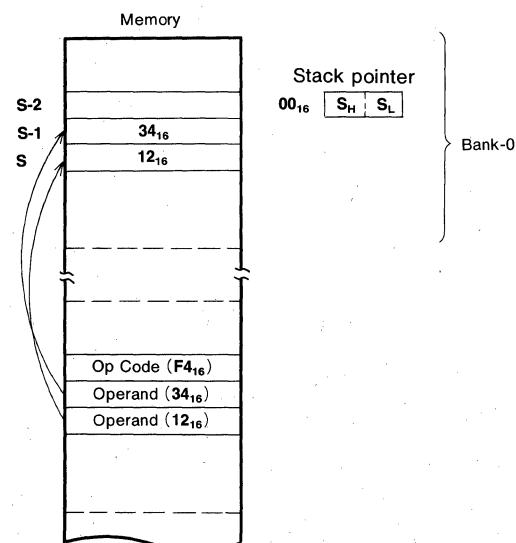
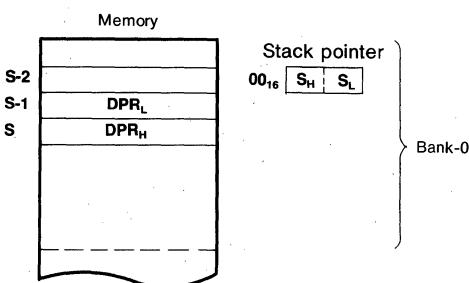


ex. : Mnemonic  
**PHD**

Machine code  
 $0B_{16}$

ex. : Mnemonic  
**PEA # 1234H**

Machine code:  
 $F4_{16} 34_{16} 12_{16}$

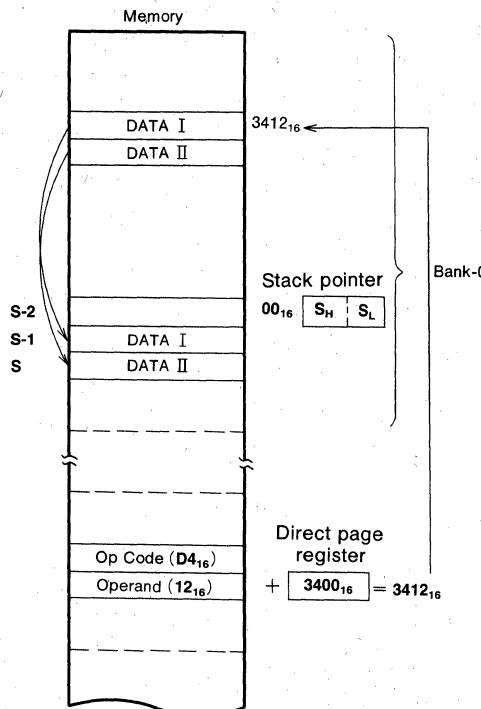


**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

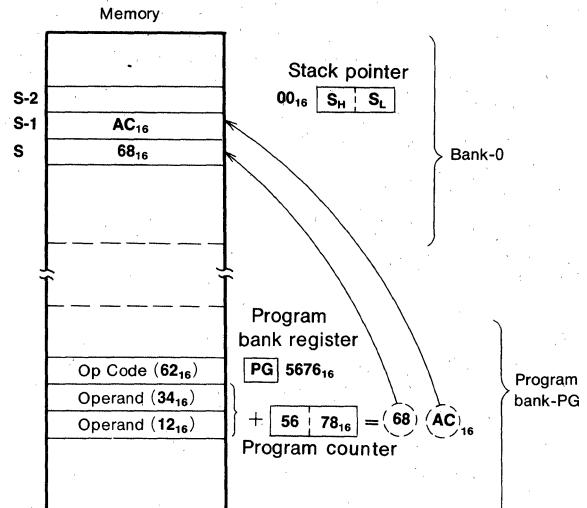
ex. : Mnemonic  
 PEI # 12H

Machine code  
 $D4_{16} \ 12_{16}$



ex. : Mnemonic  
 PER # 1234H

Machine code  
 $62_{16} \ 34_{16} \ 12_{16}$



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

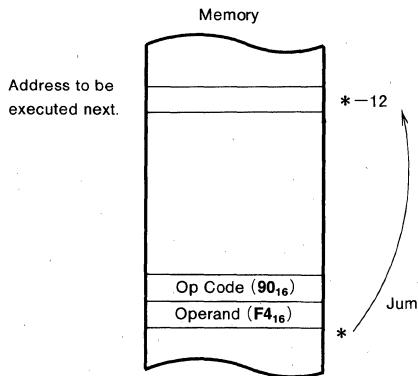
Mode : Relative addressing mode

Function : Branching occurs to the address specified by the value resulting from addition of the contents of the program counter and the instruction's second byte. In the case of a long branch by the BRA instruction, a 15-bit signed numeric value formed by the contents of the instruction's second and third bytes is added to the program counter contents. If the addition generates a carry or borrow, 1 is added to or subtracted from the program bank register.

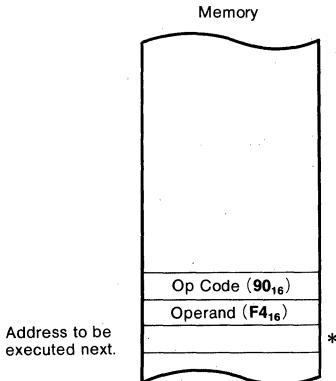
Instruction : **BCC, BCS, BEQ, BMI, BNE, BPL, BRA, BVC, BVS**

ex. : Mnemonic                      Machine code  
**BCC \*—12**                       $90_{16} F4_{16}$

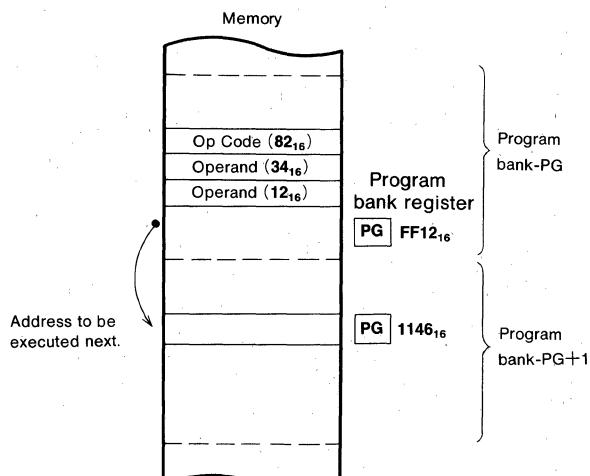
Branches to the address \*—12 if the carry flag (C) has been cleared.



Advances to the address \* if the carry flag (C) has been set.



ex. : Mnemonic                      Machine code  
**BRA 1234H**                       $82_{16} 34_{16} 12_{16}$



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

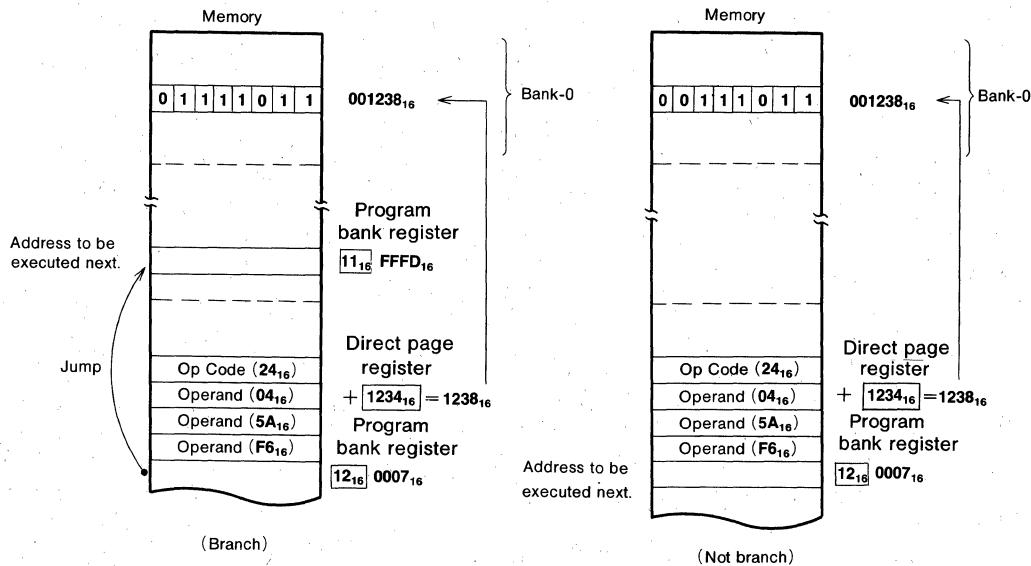
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Direct bit relative addressing mode

Function : Specifies the bank-0 memory location by the value obtained by adding the instruction's second byte to the direct page register's contents, and specifies the positions of multiple bits in the memory location by the bit pattern in the third and fourth bytes (the third byte only if the m flag is set to 1). Then, if the specified bits all satisfy the branching conditions, the instruction's fifth byte (or the fourth byte if the m flag is set to 1) is added to the program counter as a signed value, generating the branching destination address. If, however, addition of the instruction's second byte to the direct page register's contents result in a value that exceeds the bank-0 range, the specified location will be in bank-1.

Instruction : BBC, BBS

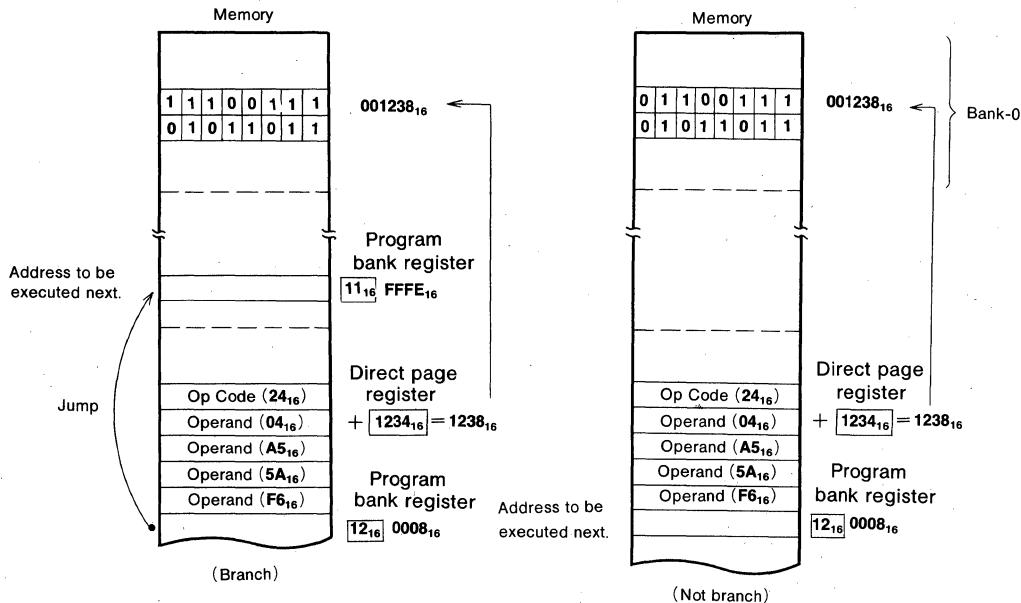
ex. : Mnemonic                      Machine code  
**BBS #5AH, 04H, 0F6H 24<sub>16</sub> 04<sub>16</sub> 5A<sub>16</sub> F6<sub>16</sub>**  
(m=1)



# MELPS 7700 ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic                      Machine code  
**BBS #5AA5H, 04H, 0F6H**    **24<sub>16</sub> 04<sub>16</sub> A5<sub>16</sub> 5A<sub>16</sub> F6<sub>16</sub>**  
 (m=0)



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

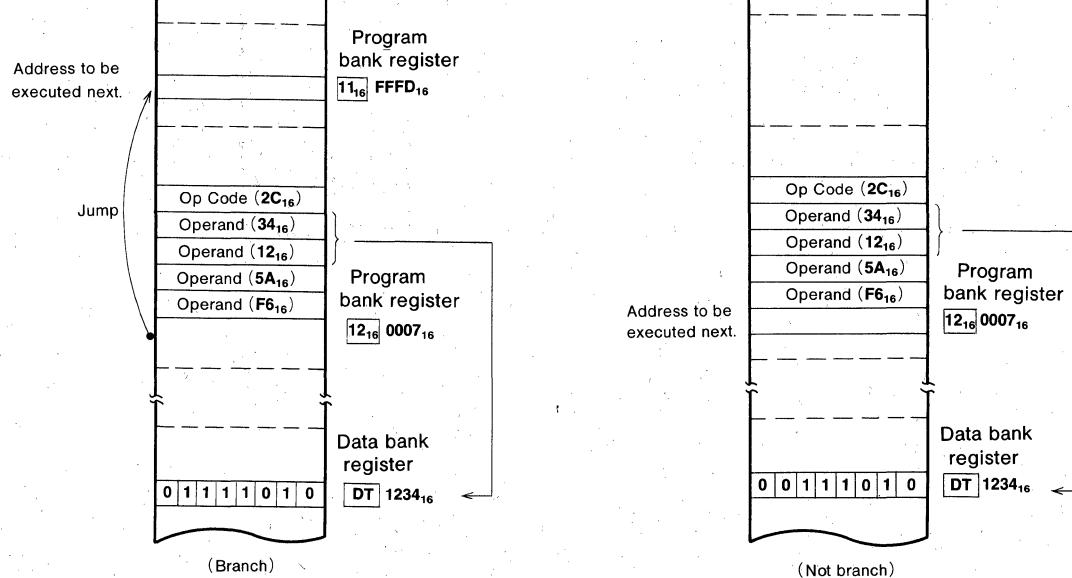
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Absolute bit relative addressing mode

Function : The instruction's second and third bytes and the contents of the data bank register specify the memory location, and data for the memory location's multiple bits is specified by a bit pattern in the instruction's fourth and fifth bytes (the fourth byte only if the m flag is set to 1). Then, if the specified bits all satisfy the branching conditions, the instruction's sixth byte (or the fifth byte if the m flag is set to 1) is added to the program counter as a signed value, generating the branching destination address.

ex. : Mnemonic                              Machine code  
**BBS #5AH, 1234H, 0F6H**    **2C<sub>16</sub> 34<sub>16</sub> 12<sub>16</sub> 5A<sub>16</sub> F6<sub>16</sub>**  
(m=1)

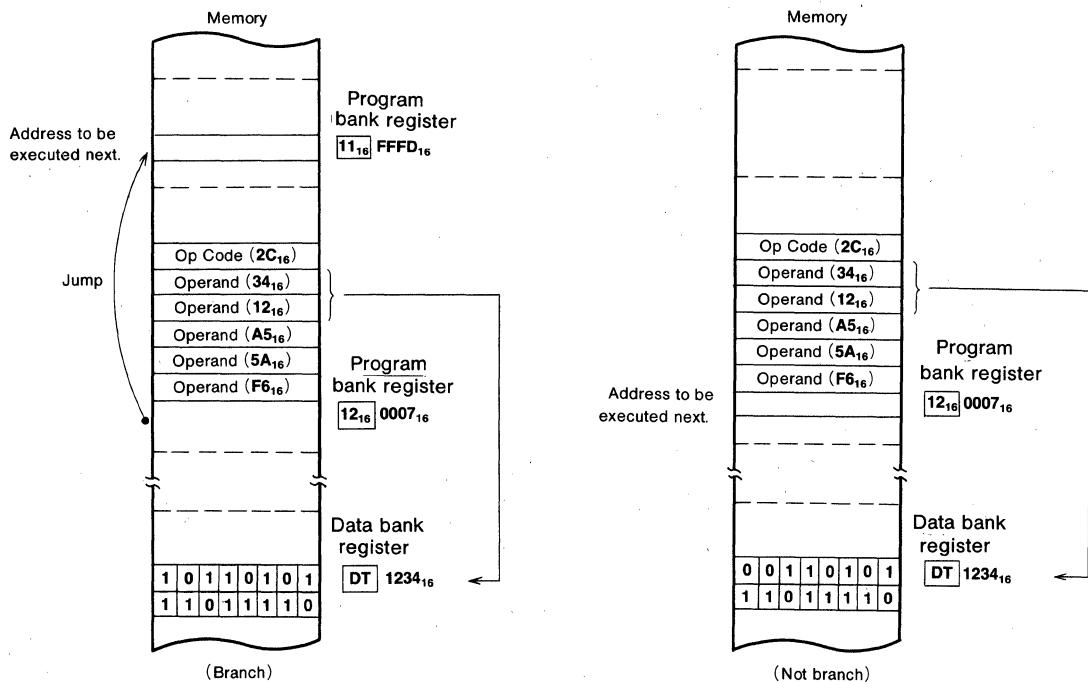
Instruction : BBC, BBS



MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic                      Machine code  
BBS #5AA5H, 1234H, 0F6H       $2C_{16} 34_{16} 12_{16} A5_{16} 5A_{16} F6_{16}$   
(m=0)



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Stack pointer relative addressing mode

Function : The contents of a bank-0 memory location specified by the value resulting from addition of the instruction's second byte and the contents of the stack pointer become the actual data. If, however, the value obtained by adding the contents of the instruction's second byte and the stack pointer's contents exceeds the bank-0 range, the specified location will be in bank-1.

Instruction : ADC, AND, CMP, DIV, EOR,  
LDA, MPY, ORA, SBC, STA

ex.

: Mnemonic

ADC A, 02H, S  
(m=1)

Machine code

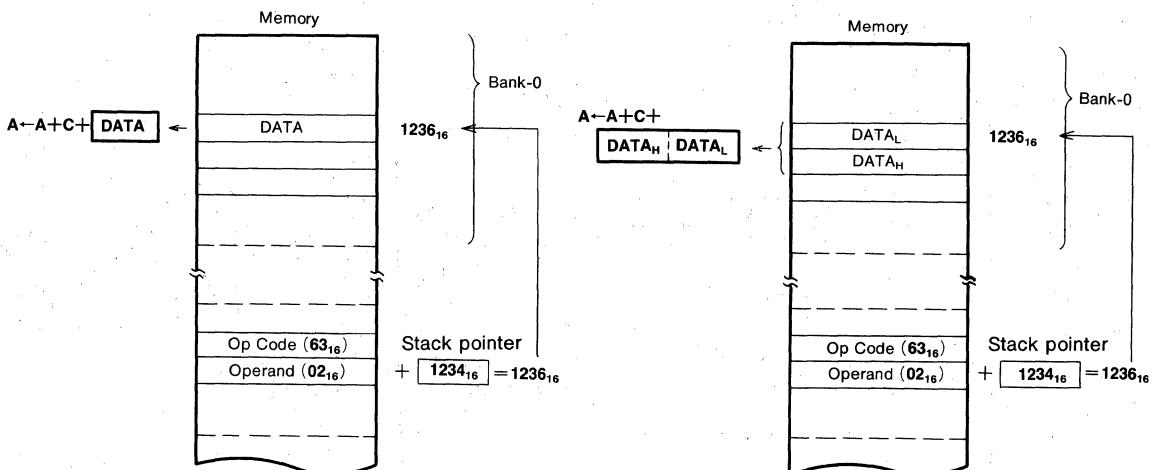
$63_{16} \ 02_{16}$

ex. : Mnemonic

ADC A, 02H, S  
(m=0)

Machine code

$63_{16} \ 02_{16}$



# MELPS 7700

## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

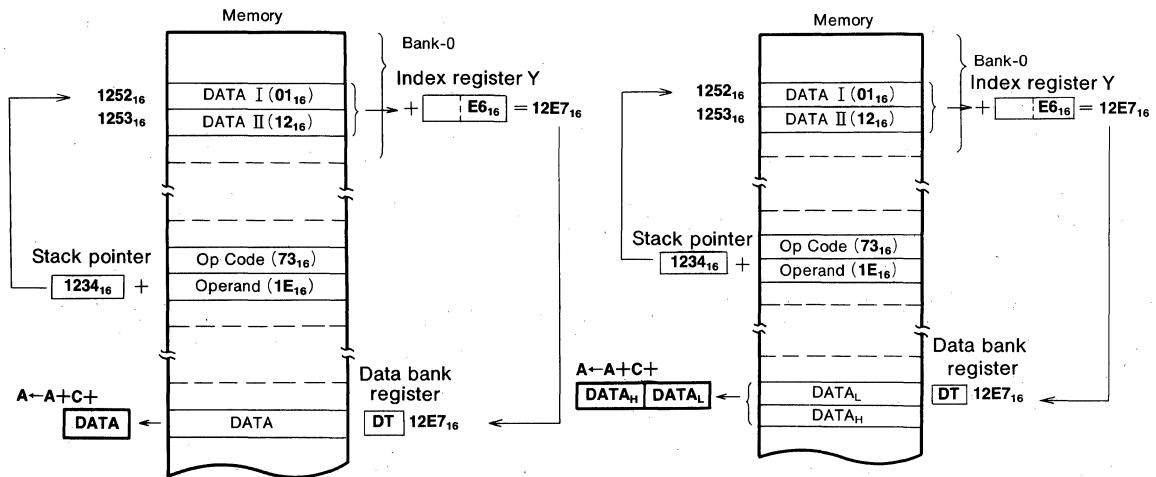
**Mode** : Stack pointer relative indirect indexed Y addressing mode

**Function** : The value obtained by adding the instruction's second byte and the contents of the stack pointer specifies 2 adjacent bytes in memory. The value obtained by adding the contents of these bytes and the contents of the index register Y specifies address of the actual data in memory bank-DT (DT is contents of data bank register). If addition of the 2 bytes in memory with the contents of the index register Y generate a carry, the bank number will be 1 larger than the contents of the data bank register.

**Instruction** : ADC, AND, CMP, DIV, EOR, LDA, MPY, ORA, SBC, STA

**ex.** : Mnemonic                      Machine code  
ADC A,(1EH, S), Y                73<sub>16</sub> 1E<sub>16</sub>  
(m=1, x=1)

**ex.** : Mnemonic                      Machine code  
ADC A,(1EH, S), Y                73<sub>16</sub> 1E<sub>16</sub>  
(m=0, x=1)

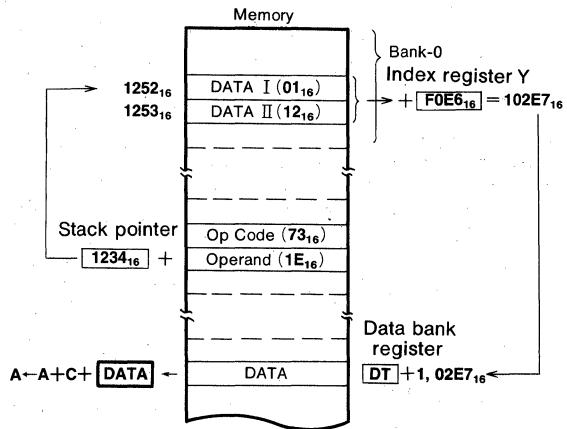


# MELPS 7700

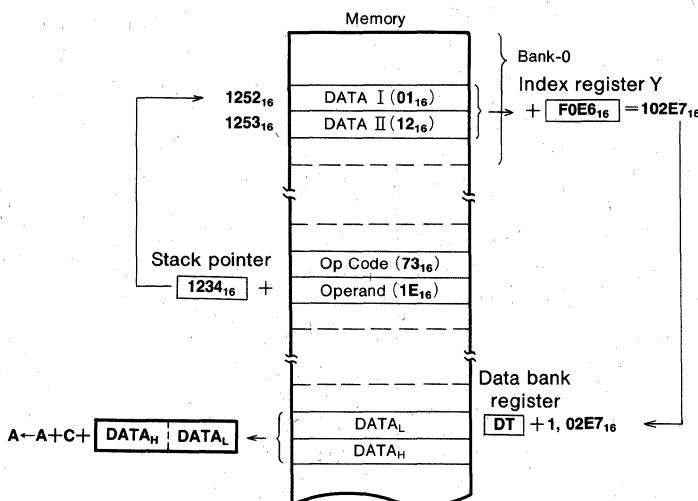
## ADDRESSING MODES

### SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic                      Machine code  
**ADC A, (1EH, S), Y**             $73_{16} \ 1E_{16}$   
 $(m=1, x=0)$



ex. : Mnemonic                      Machine code  
**ADC A, (1EH, S), Y**             $73_{16} \ 1E_{16}$   
 $(m=0, x=0)$



**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700**  
**ADDRESSING MODES**

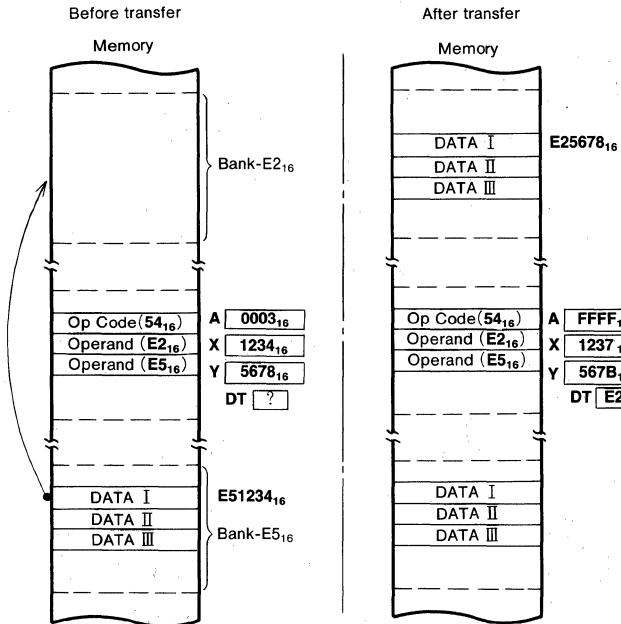
**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

Mode : Block transfer addressing mode

Function : The instruction's second byte specifies the transfer-to data bank, and the contents of the index register Y specify the transfer - to address within the data bank. The instruction's third byte specifies the transfer-from data bank, and the contents of the index register X specify the address in the data bank where the data to be transferred is stored. The contents of the accumulator A constitute the number of bytes to be transferred. Upon termination of transfer, the contents of the data bank register will specify the transfer-to data bank. The MVN instruction is used for transfer to lower address location. In this case, the contents of the index registers X and Y are incremented each time data is transferred. The MVP instruction is used for transfer to higher address location. In this case, the contents of the index registers X and Y are decremented each time data is transferred. The block of data to be transferred may cross over the bank boundary.

Instruction : MVN, MVP

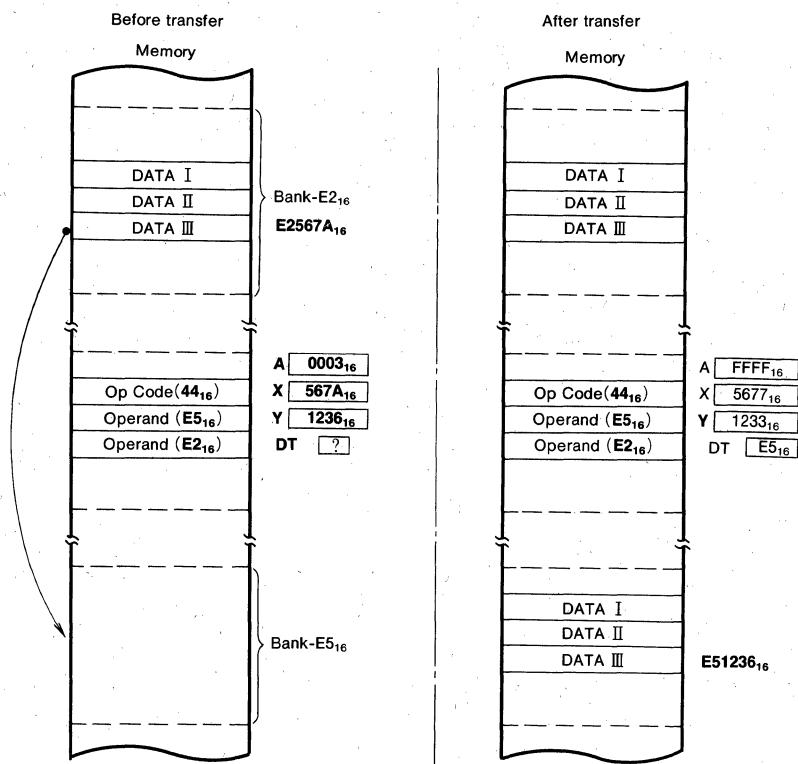
ex. : Mnemonic                      Machine code  
 MVN 0E2H, 0E5H                    54<sub>16</sub> E2<sub>16</sub> E5<sub>16</sub>



MITSUBISHI MICROCOMPUTERS  
MELPS 7700  
ADDRESSING MODES

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

ex. : Mnemonic      : Machine code  
MVP 0E5H, 0E2H      44<sub>16</sub> E5<sub>16</sub> E2<sub>16</sub>



# MELPS 7700 INSTRUCTION CODE TABLE

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

## INSTRUCTION CODE TABLE-1

		D <sub>3</sub> ~D <sub>0</sub>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA A,(DIR,X)		ORA A,SR	SEB DIR,b	ORA A,DIR	ASL DIR	'ORA A,L(DIR)	PHP A,IMM	ORA A	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 A,ABS,Y	ORA A	DEC A	TAS	CLB ABS,b	ORA A,ABS,X	ASL ABS,X	ORA A,ABL,X	
0010	2	JSR	AND A,(DIR,X)	JSR	AND A,SR	BBS DIR,b,R	AND A,DIR	ROL DIR	AND A,L(DIR)	PLP A,IMM	AND A	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 A,ABS,Y	AND A	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)	eor A,SR	Note 1	MVP A,DIR	eor DIR	LSR A,L(DIR)	eor A,ABS,Y	PHA A,IMM	eor A	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 A,DIR,X	eor DIR,X	LSR A,L(DIR),Y	eor A,ABS,Y	CLI A,ABS,Y	eor A	PHY TAD	JMP ABL	eor A,ABS,X	lsr ABS,X	eor A,ABL,X		
0110	6	RTS	ADC A,(DIR,X)	PER A,SR	ADC DIR	LDM A,DIR	ADC DIR	ROR A,L(DIR)	ADC A,ABS,Y	PLA A,IMM	adc A	ROR A	RTL	JMP (ABS)	adc A,ABS	rор 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 A,ABS,Y	adc A,ABS,Y	PLY TDA	JMP (ABS,X)	adc A,ABS,X	rор ABS,X	adc A,ABL,X		
1000	8	BRA	STA A,(DIR,X)	BRA	STA A,SR	STY DIR	STA A,DIR	STX DIR	STA A,L(DIR)	DEY Note 2	sta A,ABS,Y	TXA TDX	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 A,ABS,Y	sta A,ABS,Y	TXS TXY	TXY	LDM ABS	STA A,ABS,X	LDM ABS,X	STA A,ABL,X	
1010	A	LDY	LDA A,(DIR,X)	LDY IMM	LDA A,SR	LDY DIR	LDA A,DIR	LDY DIR	LDA A,L(DIR)	TAY A,IMM	LDA A,IMM	TAX A	PLT	LDY ABS	LDA A,ABS	LDY 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	LDY DIR,Y	LDA A,L(DIR),Y	CLV A,ABS,Y	LDA A,ABS,Y	TSX TXY	TYX	LDY ABS,X	LDA A,ABS,X	LDY ABS,Y	LDA A,ABL,X	
1100	C	CPY	CMP IMM	CLP A,(DIR,X)	CMP IMM	CMP A,SR	CPY DIR	CMP A,DIR	DEC DIR	CMP A,L(DIR)	INY A,IMM	CMP A,IMM	DEX A	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 DIR,X	CMP DIR,X	DEC DIR	CMP A,L(DIR),Y	CLM A,ABS,Y	CMP A,ABS,Y	PHX INX	STP	JMP L(ABS)	CMP A,ABS,X	DEC ABS,X	CMP A,ABL,X	
1110	E	CPX	SBC IMM	SEP A,(DIR,X)	SBC A,SR	CPX DIR	SBC A,DIR	INC DIR	SBC A,L(DIR)	INX A,IMM	SBC A,IMM	NOP A	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 DIR,X	SBC DIR,X	INC DIR,X	SBC A,L(DIR),Y	SEM A,ABS,Y	SBC A,ABS,Y	PLX PUL	PUL	JSR (ABS,X)	SBC A,ABS,X	INC ABS,X	SBC A,ABL,X	

Note 1 : 42<sub>16</sub> specifies the contents of the INSTRUCTION CODE TABLE-2.

About the second word's codes, refer to the INSTRUCTION CODE TABLE-2.

2 : 89<sub>16</sub> specifies the contents of the INSTRUCTION CODE TABLE-3.

About the third word's codes, refer to the INSTRUCTION CODE TABLE-2.

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 INSTRUCTION**  
**CODE TABLE**

**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER**

**INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42<sub>16</sub>)**

		D <sub>3</sub> ~D <sub>0</sub>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		D <sub>7</sub> ~D <sub>4</sub> Hexadecimal notation	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	

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 INSTRUCTION**  
**CODE TABLE**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

**INSTRUCTION CODE TABLE-3 (The first word's code of each instruction is 89<sub>16</sub>)**

D <sub>7</sub> ~D <sub>4</sub>	D <sub>3</sub> ~D <sub>0</sub>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		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						
0101	5										IMM						
0110	6																
0111	7																
1000	8																
1001	9																
1010	A																
1011	B																
1100	C				LDT IMM												
1101	D																
1110	E																
1111	F																

MITSUBISHI MICROCOMPUTERS  
**MELPS 7700 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	#					
ADC (Note 1,2)	Acc,C $\leftarrow$ Acc+M+C	Adds the carry, the accumulator and the memory contents. The result is entered into the accumulator. When the D flag is "0", binary addition 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	4	3	42	6	3	42	7	3	42	8	3	42	9	3	42	10	3	
				69			65			75			72			61			71			
AND (Note 1,2)	Acc $\leftarrow$ Acc $\wedge$ 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	4	3	42	6	3	42	7	3	42	8	3	42	9	3	42	10	3	
				29			25			35			32			21			31			
ASL (Note 1)	m=0 [b <sub>15</sub> ...b <sub>0</sub> ] $\leftarrow$ 0 m=1 [b <sub>15</sub> ...b <sub>0</sub> ] $\leftarrow$ 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	4	2														
						0A																
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 $\leftarrow$ PC $\pm$ offset PG $\leftarrow$ PG + 1 (carry occurred) PG $\leftarrow$ PG - 1 (borrow occurred)	Jumps to the address indicated by the program counter plus the offset value.																				
BRK	PC $\leftarrow$ PC + 2 M(S) $\leftarrow$ PG S $\leftarrow$ S - 1 M(S) $\leftarrow$ PC <sub>H</sub> S $\leftarrow$ S - 1 M(S) $\leftarrow$ PC <sub>L</sub> S $\leftarrow$ S - 1 M(S) $\leftarrow$ PS <sub>H</sub> S $\leftarrow$ S - 1 M(S) $\leftarrow$ PS <sub>L</sub> S $\leftarrow$ S - 1 I $\leftarrow$ 1 PC <sub>L</sub> $\leftarrow$ AD <sub>L</sub> PC <sub>H</sub> $\leftarrow$ AD <sub>H</sub> PG $\leftarrow$ 00 <sub>16</sub>	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 $\leftarrow$ 0	Makes the contents of the specified bit in the memory "0".											14	8	3							
CLC	C $\leftarrow$ 0	Makes the contents of the C flag "0".	18	2	1																	
CLI	I $\leftarrow$ 0	Makes the contents of the I flag "0".	58	2	1																	
CLM	m $\leftarrow$ 0	Makes the contents of the m flag "0".	08	2	1																	
CLP	PSb $\leftarrow$ 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 $\leftarrow$ 0	Makes the contents of the V flag "0".	88	2	1																	
CMP (Note 1,2)	Acc-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
						42	4	3				42	6	3			D5	7	3	42	8	3
						C9						C5					D5			42	9	3
																	D2			42	10	3
																	C1			D1		

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE**  
**INSTRUCTIONS**

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE**  
**INSTRUCTIONS**

Symbol	Function	Details	Addressing mode									
			IMP	IMM	A	DIR	DIR.b	DIR.X	DIR.Y	(DIR)	(DIR,X)	(DIR,Y)
			op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #
CPX (Note 2)	X-M	Compares the contents of the index register X with the contents of the memory.		E0 2 2		E4 4 2						
CPY (Note 2)	Y-M	Compares the contents of the index register Y with the contents of the memory.		C0 2 2		C4 4 2						
DEC (Note 1)	Acc <sup>-</sup> Acc-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 4 2							
					1A							
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) $\leftarrow$ 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 27 3 29		89 29 3 25		89 30 3 35		89 31 3 32	89 32 3 21	89 33 3 31
EOR (Note 1,2)	Acc <sup>-</sup> Acc $\vee$ 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 42 4 3 49		45 4 2 42 6 3 45		55 5 2 42 7 3 55		52 6 2 41 7 2 51 8 2 42 8 3 42 9 3 42 10 3 52 31 41 51		
INC (Note 1)	Acc <sup>-</sup> Acc+1 or M+M+1	Increments the contents of the accumulator or memory by 1.			3A 2 1 42 4 2 3A	E5 7 2		F6 7 2				
INX	X $\leftarrow$ X+1	Increments the contents of the index register X by 1.	E8 2 1									
INY	Y $\leftarrow$ Y+1	Increments the contents of the index register Y by 1.	C8 2 1									
JMP	ABS PC <sub>L</sub> $\leftarrow$ AD <sub>L</sub> PC <sub>H</sub> $\leftarrow$ AD <sub>H</sub>  ABL PC <sub>L</sub> $\leftarrow$ AD <sub>L</sub> PC <sub>H</sub> $\leftarrow$ AD <sub>H</sub> PG $\leftarrow$ AD <sub>G</sub>  (ABS) PC <sub>L</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> ) PC <sub>H</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +1)  L(ABS) PC <sub>L</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> ) PC <sub>H</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +1) PG $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +2)  (ABS, X) PC <sub>L</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +X) PC <sub>H</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +X+1)	Places a new address into the program counter and jumps to that new address.										
JSR	ABS M(S) $\leftarrow$ PC <sub>H</sub> S $\leftarrow$ S-1 M(S) $\leftarrow$ PC <sub>L</sub> S $\leftarrow$ S-1 PC <sub>L</sub> $\leftarrow$ AD <sub>L</sub> PC <sub>H</sub> $\leftarrow$ AD <sub>H</sub>  ABL M(S) $\leftarrow$ PG S $\leftarrow$ S-1 M(S) $\leftarrow$ PC <sub>H</sub> S $\leftarrow$ S-1 M(S) $\leftarrow$ PC <sub>L</sub> S $\leftarrow$ S-1 PC <sub>L</sub> $\leftarrow$ AD <sub>L</sub> PC <sub>H</sub> $\leftarrow$ AD <sub>H</sub> PG $\leftarrow$ AD <sub>G</sub>  (ABS, X) M(S) $\leftarrow$ PC <sub>H</sub> S $\leftarrow$ S-1 M(S) $\leftarrow$ PC <sub>L</sub> S $\leftarrow$ S-1 PC <sub>L</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +X) PC <sub>H</sub> $\leftarrow$ (AD <sub>H</sub> , AD <sub>L</sub> +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.										

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE**  
**INSTRUCTIONS**

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 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	#
LDA (Note 1,2)	Acc ← M	Enters the contents of the memory into the accumulator.		A9	2	2			A5	4	2		B5	5	2		B2	6	2	
				42	4	3			42	6	3		42	7	3		42	8	3	
				A9					A5				B5				A1	7	2	
																	B1	8	2	
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	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 <sub>15</sub> ...b <sub>0</sub> ]→C m=1 0→[b <sub>7</sub> ...b <sub>0</sub> ]→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	4	2					4A						
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	16	3			89	18	3		89	19	3		89	20	3	
				09					05				15				12	20	1	
																01	21	3		
																11	22	3		
MVN (Note 8)	Mn+i←Mm+i	Transmits the data block. The transmission is done from the lower order address of the block.																		
MVP (Note 9)	Mn-i←Mm-i	Transmits the data block. Transmission is done form the higher order address of the data block.																		
NOP	PC←PC+1	Advances the program counter, but performs nothing else.	EA	2	1															
ORA (Note 1,2)	Acc←AccVM	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
					42	4	3			42	6	3		42	7	3		42	8	3
					09					05				15				12	01	11
																	11	20	3	
PEA	M(S)←IMM <sub>2</sub> S←S-1 M(S)←IMM <sub>1</sub> S←S-1	The 3rd and the 2nd bytes of the instruction are saved into the stack, in this order.																		
PEI	M(S)←M((DPR)+IMM +1) S←S-1 M(S)←M((DPR)+IMM) S←S-1	Specifies 2 sequential bytes in the direct page in the 2nd byte of the instruction, and saves the contents into the stack.																		
PER	EAR←PC+IMM <sub>2</sub> ;IMM <sub>1</sub> M(S)←EAR <sub>H</sub> S←S-1 M(S)←EAR <sub>L</sub> S←S-1	Regards the 2nd and 3rd bytes of the instruction as 16-bit numerals, adds them to the program counter, and saves the result into the stack.																		
PHA	m=0 M(S)←A <sub>H</sub> S←S-1 M(S)←A <sub>L</sub> S←S-1  m=1 M(S)←A <sub>L</sub> S←S-1	Saves the contents of accumulator A into the stack.																		
PHB	m=0 M(S)←B <sub>H</sub> S←S-1 M(S)←B <sub>L</sub> S←S-1  m=1 M(S)←B <sub>L</sub> S←S-1	Saves the contents of accumulator B into the stack.																		

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE**  
**INSTRUCTIONS**

		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 #	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																			
42	12 3 B7	42 13 3 AD	42 6 4	42 8 4	42 8 4	42 8 5	42 9 5											42 7 3	42 10 3																			
A7				9C 5 4		9E 6 4																																
				AE 4 3		BE 6 3																																
				AC 4 3		BC 6 3																																
				4E 7 3		5E 8 3																																
89	24 3 07	89 25 3 17	89 18 4 0D		89 20 4 1D	89 20 4 19	89 20 5 0F	89 21 5 1F										89 19 3 03	89 22 3 13																			
07	10 2 07	17 11 2 17	0D 4 3 0D		1D 6 3	19 6 3	0F 6 4	1F 7 4											03 5 2	13 8 2																		
					42 8 4 1D	42 8 4 19	42 8 5 0F	42 9 5 1F										42 7 3 03	42 10 3 13																			
																		F4 5 3																				
																		D4 5 2																				
																		62 5 3																				
																		48 4 1																				
																		42 6 2 48																				

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 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 #	op n #	op n #
PHD	$M(S) \leftarrow DPR_H$ $S \leftarrow S - 1$ $M(S) \leftarrow DPR_L$ $S \leftarrow S - 1$	Saves the contents of the direct page register into the stack.										
PHG	$M(S) \leftarrow PG$ $S \leftarrow S - 1$	Saves the contents of the program bank register into the stack.										
PHP	$M(S) \leftarrow PS_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PS_L$ $S \leftarrow S - 1$	Saves the contents of the program status register into the stack.										
PHT	$M(S) \leftarrow DT$ $S \leftarrow S - 1$	Saves the contents of the data bank register into the stack.										
PHX	x=0 $M(S) \leftarrow X_H$ $S \leftarrow S - 1$ $M(S) \leftarrow X_L$ $S \leftarrow S - 1$  x=1 $M(S) \leftarrow X_L$ $S \leftarrow S - 1$	Saves the contents of the index register X into the stack.										
PHY	x=0 $M(S) \leftarrow Y_H$ $S \leftarrow S - 1$ $M(S) \leftarrow Y_L$ $S \leftarrow S - 1$  x=1 $M(S) \leftarrow Y_L$ $S \leftarrow S - 1$	Saves the contents of the index register Y into the stack.										
PLA	m=0 $S \leftarrow S + 1$ $A_L \leftarrow M(S)$ $S \leftarrow S + 1$ $A_H \leftarrow M(S)$  m=1 $S \leftarrow S + 1$ $A_L \leftarrow M(S)$	Restores the contents of the stack on the accumulator A.										
PLB	m=0 $S \leftarrow S + 1$ $B_L \leftarrow M(S)$ $S \leftarrow S + 1$ $B_H \leftarrow M(S)$  m=1 $S \leftarrow S + 1$ $B_L \leftarrow M(S)$	Restores the contents of the stack on the accumulator B.										
PLD	$S \leftarrow S + 1$ $DPR_L \leftarrow M(S)$ $S \leftarrow S + 1$ $DPR_H \leftarrow M(S)$	Restores the contents of the stack on the direct page register.										
PLP	$S \leftarrow S + 1$ $PS_L \leftarrow M(S)$ $S \leftarrow S + 1$ $PS_H \leftarrow M(S)$	Restores the contents of the stack on the processor status register.										
PLT	$S \leftarrow S + 1$ $DT \leftarrow M(S)$	Restores the contents of the stack on the data bank register.										
PLX	x=0 $S \leftarrow S + 1$ $X_L \leftarrow M(S)$ $S \leftarrow S + 1$ $X_H \leftarrow M(S)$  x=1 $S \leftarrow S + 1$ $X_L \leftarrow M(S)$	Restores the contents of the stack on the index register X.										

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MACHINE  
INSTRUCTIONS

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 #	•	•	•	•	•	•	•	•	•	•	
											0B 4 1							•	•	•	•	•	•	•	•	•	•	
											4B 3 1							•	•	•	•	•	•	•	•	•	•	
											0B 4 1							•	•	•	•	•	•	•	•	•	•	
											8B 3 1							•	•	•	•	•	•	•	•	•	•	
											DA 4 1							•	•	•	•	•	•	•	•	•	•	
											5A 4 1							•	•	•	•	•	•	•	•	•	•	
											6B 5 1							•	•	N	•	•	•	•	•	•	Z	•
											42 7 2							•	•	N	•	•	•	•	•	•	Z	•
											68																	
											2B 5 1							•	•	•	•	•	•	•	•	•	•	
											2B 6 1							Value saved in stack.										
											AB 6 1							•	•	N	•	•	•	•	•	Z	•	
											FA 5 1							•	•	N	•	•	•	•	•	Z	•	

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE INSTRUCTIONS**

Symbol	Function	Details	Addressing mode													
			IMP op n #	IMM op n #	A op n #	DIR op n #	DIR,b op n #	DIR,X op n #	DIR,Y (DIR) op n #	(DIR,X) op n #	(DIR,Y) op n #					
PLY	x=0 S←S+1 Y <sub>L</sub> ←M(S) S←S+1 Y <sub>H</sub> ←M(S)	Restores the contents of the stack on the index register Y.														
	x=1 S←S+1 Y <sub>L</sub> ←M(S)															
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	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	2	1	26	7	2		36	7	2		
					42	4	2									
					2A											
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	2	1	66	7	2		76	7	2		
					42	4	2									
					6A											
RTI	S←S+1 PS <sub>L</sub> ←M(S) S←S+1 PS <sub>H</sub> ←M(S) S←S+1 PC <sub>L</sub> ←M(S) S←S+1 PC <sub>H</sub> ←M(S) S←S+1 PG←M(S)	Returns from the interruption routine.	40	11	1											
RTL	S←S+1 PC <sub>L</sub> ←M(S) S←S+1 PC <sub>H</sub> ←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 <sub>L</sub> ←M(S) S←S+1 PC <sub>H</sub> ←M(S)	Returns from the subroutine. The contents of the program bank register are not restored.	60	5	1											
SBC (Note 1,2)	Acc, C←Acc-M-C	Subtracts the contents of the memory and the borrow from the contents of the accumulator.	E9	2	2			E5	4	2	F5	5	2	F2	6	2
			42	4	3			42	6	3	42	7	3	42	8	3
			E9					E5			F5			F2	E1	7
														42	10	3
														F1	8	2
														F2	E1	7
														F1	8	2

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MACHINE  
INSTRUCTIONS

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						
																		•	•	•	N	•	•	•	•	•	Z	•				
																		EB	12	2												
																		+	2h	+12												
																		FB	14	2												
																		+	3h	+142												
																		If restored the contents of PS, it becomes its value. And the other case is no change.														
																		•	•	•	•	•	•	•	•	•	•	•				
		2E 7 3		3E 8 3															•	•	•	N	•	•	•	•	•	Z	C			
		6E 7 3		7E 8 3															•	•	•	N	•	•	•	•	•	Z	C			
E7 10 2 F7 11 2 ED 4 3		FD 6 3	F9 6 3	EF 6 4	FF 7 4													E3 5 2	F3 8 2				•	•	•	N	V	•	•	•	Z	C
42 12 3 42 13 3 42 6 4		42 8 4	42 8 4	42 8 5	42 9 5													42 7 3	42 10 3													
E7	F7	ED		FD	F9	EF	FF										E3	F3														

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE  
INSTRUCTIONS**

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MACHINE  
INSTRUCTIONS**



# MELPS 7700 MACHINE INSTRUCTIONS

## Symbols in machine instructions table

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	▽	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	Acc	Accumulator
DIR, b	Direct bit addressing mode	AccH	Accumulator's upper 8 bits
DIR, X	Direct indexed X addressing mode	AccL	Accumulator's lower 8 bits
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A <sub>H</sub>	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A <sub>L</sub>	Accumulator A's lower 8 bits
(DIR, Y)	Direct indexed indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B <sub>H</sub>	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B <sub>L</sub>	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X <sub>H</sub>	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X <sub>L</sub>	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y <sub>H</sub>	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y <sub>L</sub>	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 <sub>H</sub>	Program counter's upper 8 bits
STK	Stack addressing mode	PC <sub>L</sub>	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 <sub>H</sub>	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR <sub>L</sub>	Direct page register's lower 8 bits
BLK	Block transfer addressing mode	PS	Processor status register
C	Carry flag	PS <sub>H</sub>	Processor status register's upper 8 bits
Z	Zero flag	PS <sub>L</sub>	Processor status register's lower 8 bits
I	Interrupt disable flag	PS <sub>b</sub>	Processor status register's b-th bit
D	Decimal operation mode flag	M(S)	Contents of memory at address indicated by stack pointer
x	Index register length selection flag	M <sub>b</sub>	b-th memory location
m	Data length selection flag	AD <sub>G</sub>	Value of 24-bit address's upper 8-bit ( $A_{23} \sim A_{16}$ )
V	Overflow flag	AD <sub>H</sub>	Value of 24-bit address's middle 8-bit ( $A_{15} \sim A_8$ )
N	Negative flag	AD <sub>L</sub>	Value of 24-bit address's lower 8-bit ( $A_7 \sim A_0$ )
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 <sub>1</sub> , i <sub>2</sub>	Number of registers pushed or pulled
^	Logical AND		
▽	Logical OR		

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 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  $DPR_L \neq 0$  must be incremented by 1. The number of cycles shown in the table differs according to the bytes fetched into the instruction queue buffer, or according to whether the memory read/write address is odd or even. It also differs when the external region memory is accessed by  $BYTE="H"$ .

Note 1. The operation code at the upper row is used for accumulator A, and the operation at the lower row is used for accumulator B.

Note 2. When setting flag  $m=0$  to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 3. The number of cycles increments by 2 when branching.

Note 4. The operation code on the upper row is used for branching in the range of  $-128 \sim +127$ , and the operation code on the lower row is used for branching in the range of  $-32768 \sim +32767$ .

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 transferred is even.

When the number of bytes to be transferred 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 transferred is even.

When the number of bytes to be transferred 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.



---

## **PROGRAMMABLE ROM MICROCOMPUTERS**

---

**3**



# M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP M37702E2FS, M37702E2AFS, M37702E2BFS

## PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP

### DESCRIPTION

The M37702E2-XXXFP, M37702E2AXXXFP and M37702E2BXXXFP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 80-pin plastic molded QFP. The features of these chips are similar to those of the M37702M2-XXXFP, M37702M2AXXXFP and M37702M2BXXXFP except that these chips have a 16K-byte PROM built in.

These single-chip microcomputers have a large 16M bytes address space, 3-byte instruction queue buffers, and 2-byte 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. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, this chip is suitable for small quantity production runs. The M37702E2FS (8MHz version), M37702E2AFS (16MHz version) and M37702E2BFS (25MHz version) with erasable ROM that are housed in a windowed ceramic LCC are also provided.

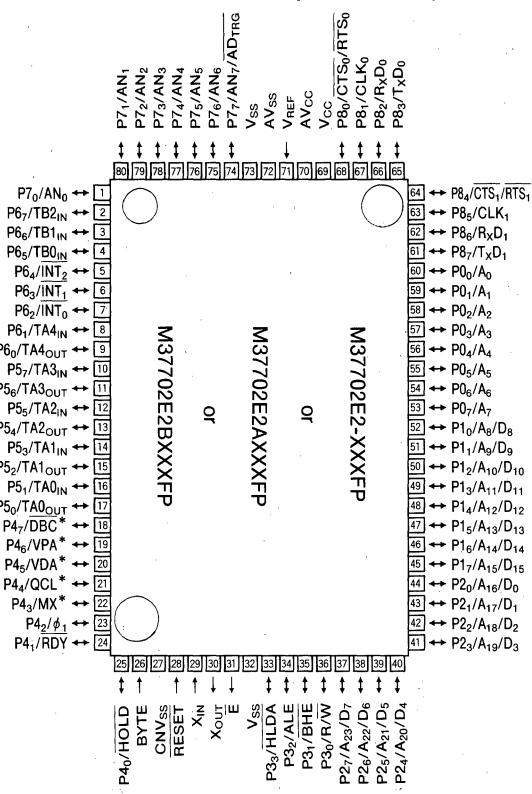
The differences between the M37702E2-XXXFP, M37702E2AXXXFP and M37702E2BXXXFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702E2-XXXFP unless otherwise noted.

Type name	External clock input frequency
M37702E2-XXXFP	8 MHz
M37702E2AXXXFP	16MHz
M37702E2BXXXFP	25MHz

### DISTINCTIVE FEATURES

- Number of basic instructions ..... 103
- Memory size PROM ..... 16K bytes
- RAM ..... 512 bytes
- Instruction execution time
  - M37702E2-XXXFP
    - (The fastest instruction at 8 MHz frequency) ..... 500ns
  - M37702E2AXXXFP
    - (The fastest instruction at 16 MHz frequency) ..... 250ns
  - M37702E2BXXXFP
    - (The fastest instruction at 25 MHz frequency) ..... 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)
  - ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) ..... 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  - (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) ..... 68

### PIN CONFIGURATION (TOP VIEW)



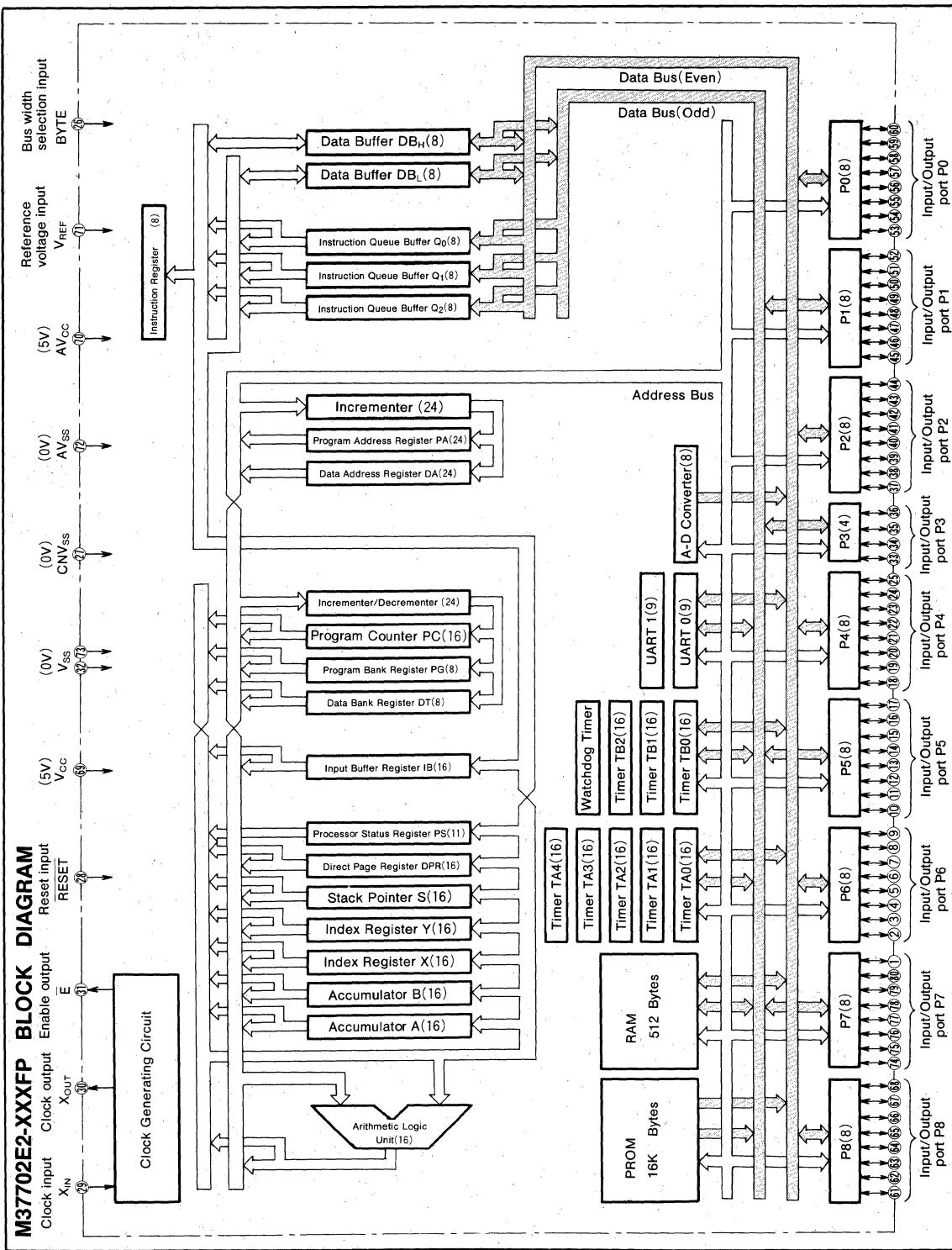
### Outline 80P6N

\*: Used in the evaluation chip mode only

### APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME and NC, communication, and measuring instruments

**M37702E2-XXXFP, M37702E2AXXFP, M37702E2BXXFP**  
**M37702E2FS, M37702E2AFS, M37702E2BFS**
**PROM VERSION of M37702M2-XXXFP, M37702M2AXXFP, M37702M2BXXFP**


**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**FUNCTIONS OF M37702E2-XXXFP**

Parameter		Functions
Number of basic instructions		103
Instruction execution time	M37702E2-XXXFP, M37702E2FS	500ns (the fastest instructions, at 8MHz frequency)
	M37702E2AXXXFP, M37702E2AFS	250ns (the fastest instructions, at 16MHz frequency)
	M37702E2BXXXFP, M37702E2BFS	160ns (the fastest instructions, at 25MHz frequency)
Memory size	PROM	16K bytes
	RAM	512 bytes
Input/Output ports	P0~P2, P4~P8	8-bit×8
	P3	4-bit×1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit×5
	TB0, TB1, TB2	16-bit×3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		8-bit×1 (8 channels)
Watchdog timer		12-bit×1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0~7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5V±10%
Power dissipation		30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		-20~85°C
Device structure		CMOS high-performance silicon gate process
Package	M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP	80-pin plastic molded QFP
	M37702E2FS, M37702E2AFS, M37702E2BFS	80-pin ceramic LCC (with a window)

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**PIN DESCRIPTION (NORMAL MODE)**

Pin	Name	Input/Output	Functions
V <sub>CC</sub> , V <sub>SS</sub>	Power supply		Supply 5 V±10% to V <sub>CC</sub> and 0 V to V <sub>SS</sub> .
CNV <sub>SS</sub>	CNV <sub>SS</sub> Input	Input	This pin controls the processor mode. Connect to V <sub>SS</sub> for single-chip mode.
<u>RESET</u>	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition which should be maintained for the required time.
X <sub>IN</sub>	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X <sub>IN</sub> and X <sub>OUT</sub> . When an external clock is used, the clock source should be connected to the X <sub>IN</sub> pin and the X <sub>OUT</sub> pin should be left open.
X <sub>OUT</sub>	Clock output	Output	
<u>E</u>	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	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 "L" signal inputs and 8 bits when "H" signal inputs.
AV <sub>CC</sub> , AV <sub>SS</sub>	Analog supply input		Power supply for the A-D converter. Connect AV <sub>CC</sub> to V <sub>CC</sub> and AV <sub>SS</sub> to V <sub>SS</sub> externally.
V <sub>REF</sub>	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
P0 <sub>0</sub> ~P0 <sub>7</sub>	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset. Address(A <sub>7</sub> ~A <sub>0</sub> ) is output in memory expansion mode or microprocessor mode.
P1 <sub>0</sub> ~P1 <sub>7</sub>	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data(D <sub>15</sub> ~D <sub>8</sub> ) is input or output when E output is "L" and an address(A <sub>15</sub> ~A <sub>8</sub> ) is output when E output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A <sub>15</sub> ~A <sub>8</sub> ) is output.
P2 <sub>0</sub> ~P2 <sub>7</sub>	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D <sub>7</sub> ~D <sub>0</sub> ) is input or output when E output is "L" and an address(A <sub>23</sub> ~A <sub>16</sub> ) is output when E output is "H".
P3 <sub>0</sub> ~P3 <sub>3</sub>	I/O port P3	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/W, BHE, ALE, and HLDA signals are output.
P4 <sub>0</sub> ~P4 <sub>7</sub>	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, P4 <sub>0</sub> and P4 <sub>1</sub> become HOLD and RDY input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode, port P4 <sub>2</sub> can be programmed for φ <sub>1</sub> output pin divided the clock to X <sub>IN</sub> pin by 2. In microprocessor mode, P4 <sub>2</sub> always has the function as φ <sub>1</sub> output pin.
P5 <sub>0</sub> ~P5 <sub>7</sub>	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, and timer A3.
P6 <sub>0</sub> ~P6 <sub>7</sub>	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A4, external interrupt input INT <sub>0</sub> , INT <sub>1</sub> , and INT <sub>2</sub> pins, and input pins for timer B0, timer B1, and timer B2.
P7 <sub>0</sub> ~P7 <sub>7</sub>	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN <sub>0</sub> ~AN <sub>7</sub> input pins. P7 <sub>7</sub> also has an A-D conversion trigger input function.
P8 <sub>0</sub> ~P8 <sub>7</sub>	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as Rx/D, TxD, CLK, CTS/RTS pins for UART 0 and UART 1.

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**PIN DESCRIPTION (EPROM MODE)**

Pin	Name	Input/Output	Functions
V <sub>CC</sub> , V <sub>SS</sub>	Power supply		Supply 5 V±10% to V <sub>CC</sub> and 0 V to V <sub>SS</sub> .
CNV <sub>SS</sub>	V <sub>PP</sub> input	Input	Connect to V <sub>PP</sub> when programming or verifying.
BYTE	V <sub>PP</sub> input	Input	Connect to V <sub>PP</sub> when programming or verifying.
RESET	Reset input	Input	Connect to V <sub>SS</sub> .
X <sub>IN</sub>	Clock input	Input	Connect a ceramic resonator between X <sub>IN</sub> and X <sub>OUT</sub> .
X <sub>OUT</sub>	Clock output	Output	
Ē	Enable output	Output	Keep open.
AV <sub>CC</sub> , AV <sub>SS</sub>	A-D power supply	Input	Connect AV <sub>CC</sub> to V <sub>CC</sub> and AV <sub>SS</sub> to V <sub>SS</sub> .
V <sub>REF</sub>	Reference voltage input	Input	Connect to V <sub>SS</sub> .
P0 <sub>0</sub> ~P0 <sub>7</sub>	Address input (A <sub>0</sub> ~A <sub>7</sub> )	Input	Port P0 functions as the lower 8 bits address input (A <sub>0</sub> ~A <sub>7</sub> ).
P1 <sub>0</sub> ~P1 <sub>7</sub>	Address input (A <sub>8</sub> ~A <sub>14</sub> )	Input	Port P1 <sub>0</sub> ~P1 <sub>6</sub> functions as the higher 7 bits address input (A <sub>8</sub> ~A <sub>14</sub> ). Connect P1 <sub>7</sub> to V <sub>CC</sub> .
P2 <sub>0</sub> ~P2 <sub>7</sub>	Data I/O (D <sub>0</sub> ~D <sub>7</sub> )	I/O	Port P2 functions as the 8 bits data bus (D <sub>0</sub> ~D <sub>7</sub> ).
P3 <sub>0</sub> ~P3 <sub>3</sub>	Input port P3	Input	Connect to V <sub>SS</sub> .
P4 <sub>0</sub> ~P4 <sub>7</sub>	Input port P4	Input	Connect to V <sub>SS</sub> .
P5 <sub>0</sub> ~P5 <sub>7</sub>	Control signal input	Input	P5 <sub>1</sub> and P5 <sub>2</sub> functions as OE and CE input pin. Connect P5 <sub>0</sub> , P5 <sub>3</sub> , P5 <sub>4</sub> and P5 <sub>5</sub> to V <sub>CC</sub> . Connect P5 <sub>6</sub> and P5 <sub>7</sub> to V <sub>SS</sub> .
P6 <sub>0</sub> ~P6 <sub>7</sub>	Input port P6	Input	Connect to V <sub>SS</sub> .
P7 <sub>0</sub> ~P7 <sub>7</sub>	Input port P7	Input	Connect to V <sub>SS</sub> .
P8 <sub>0</sub> ~P8 <sub>7</sub>	Input port P8	Input	Connect to V <sub>SS</sub> .

# M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP M37702E2FS, M37702E2AFS, M37702E2BFS

## PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP

### EPROM MODE

The M37702E2-XXXFP features an EPROM mode in addition to its normal modes. When the RESET signal level is "L", the chip automatically enters the EPROM mode. Table 1 list the correspondence between pins and Fig. 1 shows the pin connections in the EPROM mode.

When in the EPROM mode, ports P0, P1, P2, P5<sub>1</sub>, P5<sub>2</sub>, CNV<sub>SS</sub> and BYTE are used for the EPROM (equivalent to the M5M27C256K). When in this mode, the built-in PROM can be written to or read from using these pins in the same way as with the M5M27C256K.

This chip does not have Device Identifier Mode, so that set the corresponding program algorithm. The program area should specify address 4000<sub>16</sub> ~ 7FFF<sub>16</sub> for the M37702E2-XXXFP.

Connect the clock which is either ceramic resonator or external clock to X<sub>IN</sub> pin and X<sub>OUT</sub> pin.

### 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 and M37702E2BFS, 1M mode way becomes standard.

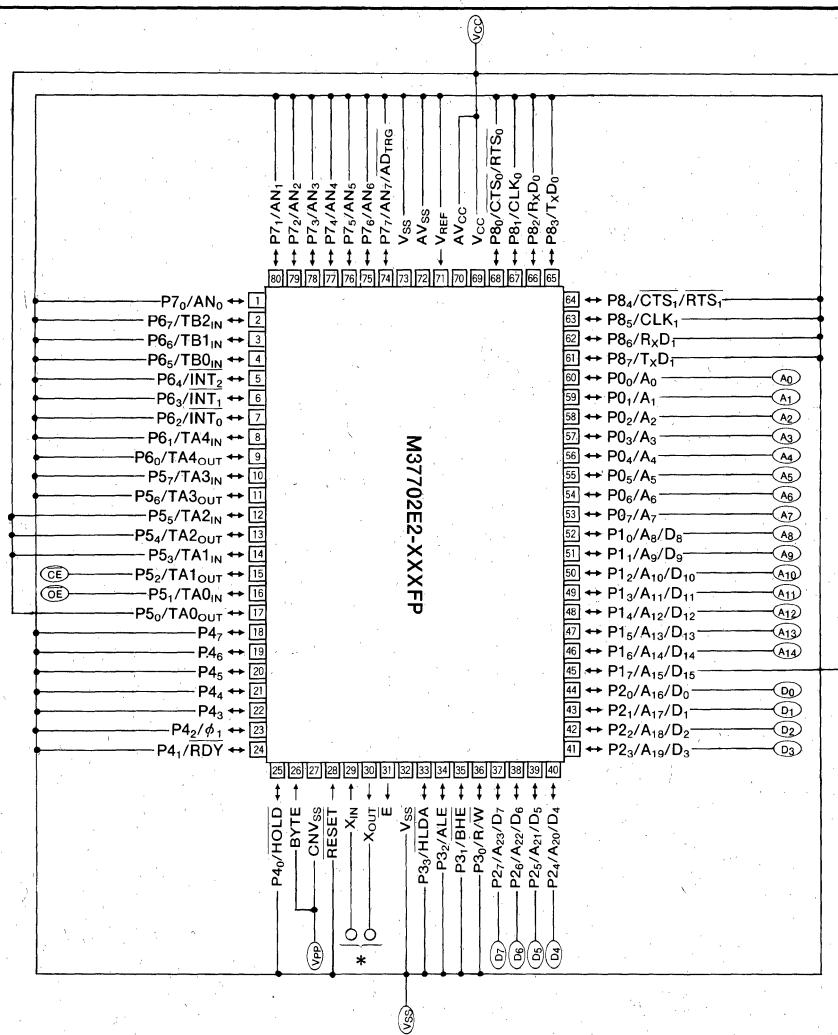


Fig. 1 Pin connection in EPROM programming mode

# M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP M37702E2FS, M37702E2AFS, M37702E2BFS

## PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP

Table 1 Pin function in EPROM programming mode

	M37702E2-XXXFP	M5M27C256K
V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>
V <sub>PP</sub>	CNV <sub>SS</sub> , BYTE	V <sub>PP</sub>
V <sub>SS</sub>	V <sub>SS</sub>	V <sub>SS</sub>
Address Input	Ports P0, P1 <sub>0</sub> ~P1 <sub>6</sub>	A <sub>0</sub> ~A <sub>14</sub>
Data I/O	Port P2	D <sub>0</sub> ~D <sub>7</sub>
CE	P5 <sub>2</sub>	CE
OE	P5 <sub>1</sub>	OE

### FUNCTION IN EPROM MODE

#### Reading

To read the EPROM, set the CE and OE pins to a "L" level. Input the address of the data (A<sub>0</sub>~A<sub>14</sub>) to be read and the data will be output to the I/O pins D<sub>0</sub>~D<sub>7</sub>. The data I/O pins will be floating when either the CE or OE pins are in the "H" state.

#### Writing

To write to the EPROM, set the OE pin to a "H" level. The CPU will enter the program mode when V<sub>PP</sub> is applied to the V<sub>PP</sub> pin. The address to be written to is selected with pins A<sub>0</sub>~A<sub>14</sub>, and the data to be written is input to pins D<sub>0</sub>~D<sub>7</sub>. Set the CE pin to a "L" level to begin writing.

#### Erasing

To erase data on this chip, use an ultraviolet light source with a 2537 Angstrom wave length. The minimum radiation power necessary for erasing is 15W·s/cm<sup>2</sup>.

(M37702E2FS, M37702E2AFS, M37702E2BFS)

### Program operation

#### AC ELECTRICAL CHARACTERISTICS (T<sub>A</sub>=25±5°C, V<sub>CC</sub>=6V±0.25V, V<sub>PP</sub>=12.5±0.3V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>AS</sub>	Address setup time			2		μs
t <sub>OES</sub>	OE setup time			2		μs
t <sub>DS</sub>	Data setup time			2		μs
t <sub>AH</sub>	Address hold time			0		μs
t <sub>DH</sub>	Data hold time			2		μs
t <sub>DFF</sub>	Output enable to output float delay			0	130	ns
t <sub>VCS</sub>	V <sub>CC</sub> setup time			2		μs
t <sub>VPS</sub>	V <sub>PP</sub> setup time			2		μs
t <sub>FPW</sub>	CE initial program pulse width		0.95	1	1.05	ms
t <sub>OPW</sub>	CE over program pulse width		2.85		78.75	ms
t <sub>OE</sub>	Data valid from OE				150	ns

### FAST PROGRAMMING ALGORITHM

To program the M37702E2-XXXFP with fast programming algorithm, first set V<sub>CC</sub>=6V, V<sub>PP</sub>=12.5, and set the address to "0". Apply a 1ms write pulse, check that the data can be read, and if it cannot be read OK, repeat the procedure, applying a 1ms write pulse and checking that the data can be read until it can be read OK. Record the accumulated number of pulse applied (N) before the data can be read OK, and then write the data again, applying a further three times this number of pulses (3×N ms).

When this series of write operations is complete, increment the address, and continue to repeat the procedure above until the last address has been reached.

Finally, when all addresses have been written, read with V<sub>CC</sub>=V<sub>PP</sub>=5V (or V<sub>CC</sub>=V<sub>PP</sub>=5.25V).

Table 2 I/O signal in each mode

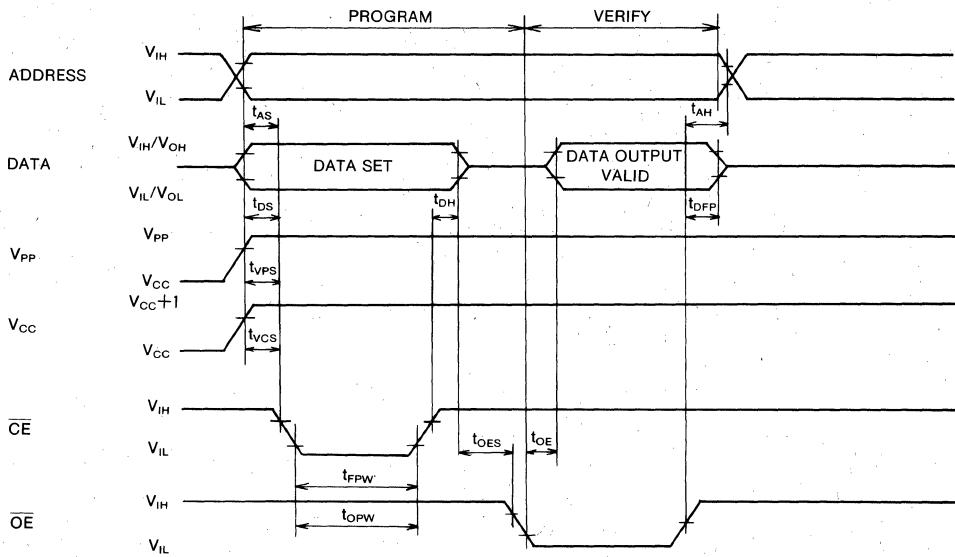
Pin Mode	CE	OE	V <sub>PP</sub>	V <sub>CC</sub>	Data I/O
Read-out	V <sub>IL</sub>	V <sub>IL</sub>	5V	5V	Output
Output	V <sub>IL</sub>	V <sub>IH</sub>	5V	5V	Floating
Disable	V <sub>IH</sub>	X	5V	5V	Floating
Programming	V <sub>IL</sub>	V <sub>IH</sub>	12.5V	6V	Input
Programming Verify	V <sub>IH</sub>	V <sub>IL</sub>	12.5V	6V	Output
Program Disable	V <sub>IH</sub>	V <sub>IH</sub>	12.5V	6V	Floating

Note 1 : An X indicates either V<sub>IL</sub> or V<sub>IH</sub>.

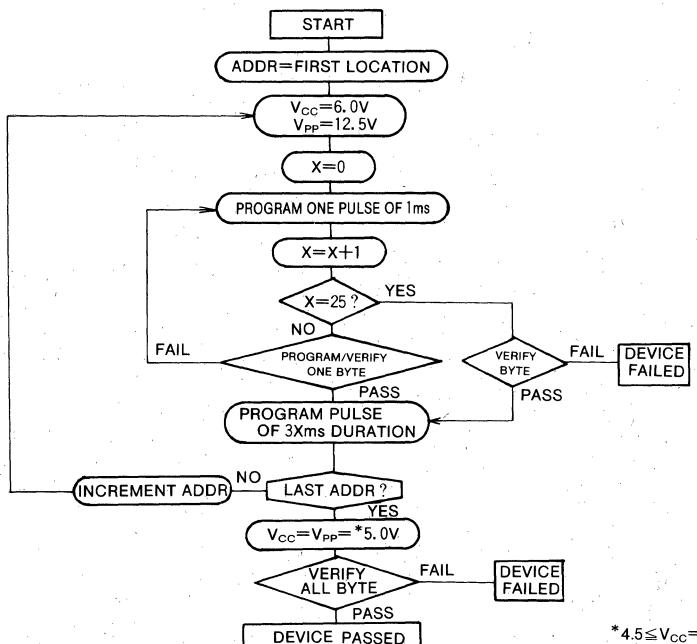
**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

### AC waveforms



### Fast programming algorithm flow chart



**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**SAFETY INSTRUCTIONS**

- (1) Sunlight and fluorescent lamp contain light that can erase written information. When using in read mode, be sure to cover the transparent glass portion with a seal or other materials (ceramic package product).
- (2) Mitsubishi Electric corp. provides the seal for covering the transparent glass. Take care that the seal does not touch the read pins (ceramic package product).
- (3) Clean the transparent glass before erasing. Fingers' flat and paste disturb the passage of ultraviolet rays and may affect badly the erasure capability (ceramic package product).
- (4) A high voltage is used for writing. Take care that over-voltage is not applied. Take care especially at power on.
- (5) The programmable M37702E2FP, M37702E2AFP and M37702E2BFP that are shipped in blank are also provided. For the M37702E2FP, M37702E2AFP and M37702E2BFP, Mitsubishi Electric corp. does not perform PROM write test and screening in the assembly process and following processes. To improve reliability after write, performing write and test according to the flow below before use is recommended.

**BASIC FUNCTION BLOCKS**

Since these processors operate in exactly the same way as the M37702M2-XXXFP, refer to the section on the M37702M2-XXXFP.

**ADDRESSING MODES**

The M37702E2-XXXFP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

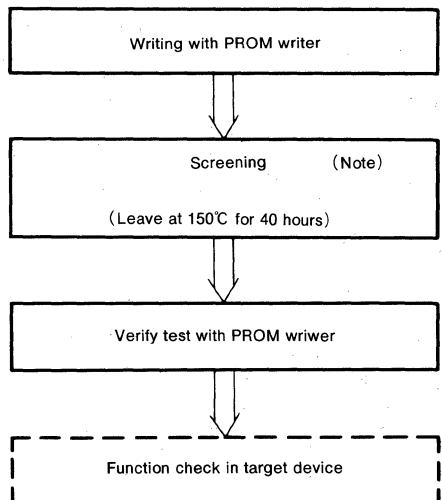
**MACHINE INSTRUCTION LIST**

The M37702E2-XXXFP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

**DATA REQUIRED FOR PROM ORDERING**

Please send the following data for writing to PROM.

- (1) M37702E2-XXXFP writing to PROM order confirmation form
- (2) 80P6N mark specification form
- (3) ROM data (EPROM 3sets)



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

- (6) Use a fit IC socket to mount the ceramic package product except for evaluation. Settle the ceramic package in an IC socket with silicon resin and the like, surely.

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Conditions	Ratings	Unit
$V_{CC}$	Supply voltage		-0.3~7	V
$AV_{CC}$	Analog supply voltage		-0.3~7	V
$V_I$	Input voltage RESET, CNV <sub>SS</sub> , BYTE		-0.3~12	V
$V_I$	Input voltage P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub> , V <sub>REF</sub> , X <sub>IN</sub>		-0.3~ $V_{CC}$ +0.3	V
$V_O$	Output voltage P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub> , X <sub>OUT</sub> , E		-0.3~ $V_{CC}$ +0.3	V
$P_d$	Power dissipation	$T_a=25^\circ C$	300	mW
$T_{opr}$	Operating temperature		-20~85	°C
$T_{stg}$	Storage temperature		-40~150	°C

**RECOMMENDED OPERATING CONDITIONS ( $V_{CC}=5V\pm10\%$ ,  $T_a=-20\sim85^\circ C$ , unless otherwise noted)**

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$V_{CC}$	Supply voltage	4.5	5.0	5.5	V
$AV_{CC}$	Analog supply voltage		$V_{CC}$		V
$V_{SS}$	Supply voltage		0		V
$AV_{SS}$	Analog supply voltage		0		V
$V_{IH}$	High-level input voltage P <sub>0</sub> ~P <sub>7</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	High-level input voltage P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> (in single-chip mode)	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	High-level input voltage P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> (in memory expansion mode and microprocessor mode)	0.5 $V_{CC}$		$V_{CC}$	V
$V_{IL}$	Low-level input voltage P <sub>0</sub> ~P <sub>7</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0		0.2 $V_{CC}$	V
$V_{IL}$	Low-level input voltage P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> (in single-chip mode)	0		0.2 $V_{CC}$	V
$V_{IL}$	Low-level input voltage P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> (in memory expansion mode and microprocessor mode)	0		0.16 $V_{CC}$	V
$I_{OH(peak)}$	High-level peak output current P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub>			-10	mA
$I_{OH(avg)}$	High-level average output current P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub>			-5	mA
$I_{OL(peak)}$	Low-level peak output current P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub>			10	mA
$I_{OL(avg)}$	Low-level average output current P <sub>0</sub> ~P <sub>7</sub> , P <sub>1</sub> ~P <sub>17</sub> , P <sub>2</sub> ~P <sub>27</sub> , P <sub>3</sub> ~P <sub>33</sub> , P <sub>4</sub> ~P <sub>47</sub> , P <sub>5</sub> ~P <sub>57</sub> , P <sub>6</sub> ~P <sub>67</sub> , P <sub>7</sub> ~P <sub>77</sub> , P <sub>8</sub> ~P <sub>87</sub>			5	mA
$f(X_{IN})$	M37702E2-XXXFP, M37702E2FS			8	MHz
	M37702E2AXXXFP, M37702E2AFS			16	
	M37702E2BXXXFP, M37702E2BFS			25	

Note 1. Average output current is the average value of a 100ms interval.

2. The sum of  $I_{OL(peak)}$  for ports P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, and P<sub>8</sub> must be 80mA or less,  
the sum of  $I_{OH(peak)}$  for ports P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, and P<sub>8</sub> must be 80mA or less,  
the sum of  $I_{OL(peak)}$  for ports P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>, and P<sub>7</sub> must be 80mA or less, and  
the sum of  $I_{OH(peak)}$  for ports P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>, and P<sub>7</sub> must be 80mA or less.

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**M37702E2-XXXFP**

**ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_1$ , $P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_1$ , $P_3$	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage $P_3$	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-10mA$	3.4			V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_1$ , $P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_1$ , $P_3$	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage $P_3$	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			0.43	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA0_{IN} \sim TA4_{IN}$ , $TB0_{IN} \sim TB2_{IN}$ , $INT_0 \sim INT_2$ , $AD_{TRG}$ , $CTS_0$ , $CTS_1$ , $CLK_0$ , $CLK_1$		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis $X_{IN}$		0.1		0.3	V
$I_{IH}$	High-level input current $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3 \sim P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $X_{IN}$ , RESET, $CNV_{SS}$ , BYTE	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3 \sim P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $X_{IN}$ , RESET, $CNV_{SS}$ , BYTE	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	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 $V_{SS}$ during reset.	$f(X_{IN})=8MHz$ , square waveform	6	12	$mA$
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
$t_{CONV}$	Conversion time		28.5			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**M37702E2AXXXFP**

**ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P3 <sub>3</sub> , P4 <sub>0</sub> ~P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>0</sub> ~P6 <sub>7</sub> , P7 <sub>0</sub> ~P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>7</sub>	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P3 <sub>3</sub>	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage P3 <sub>2</sub>	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-10mA$	3.4			V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P3 <sub>3</sub> , P4 <sub>0</sub> ~P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>0</sub> ~P6 <sub>7</sub> , P7 <sub>0</sub> ~P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>7</sub>	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P3 <sub>3</sub>	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage P3 <sub>2</sub>	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			0.43	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA4 <sub>IN</sub> , TB0 <sub>IN</sub> ~TB2 <sub>IN</sub> , INT0~INT2, ADTRG, CTS <sub>0</sub> , CTS <sub>1</sub> , CLK <sub>0</sub> , CLK <sub>1</sub>		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>		0.1		0.3	V
$I_{IH}$	High-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>3</sub> , P4 <sub>0</sub> ~P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>0</sub> ~P6 <sub>7</sub> , P7 <sub>0</sub> ~P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_I=5V$			5	$\mu A$
$I_{IL}$	Low-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>3</sub> , P4 <sub>0</sub> ~P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>0</sub> ~P6 <sub>7</sub> , P7 <sub>0</sub> ~P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_I=0V$			-5	$\mu A$
$V_{RAM}$	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 $V_{SS}$ during reset.	$f(X_{IN})=16MHz$ , square waveform	12	24	mA
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	$\mu A$

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k $\Omega$
$t_{CONV}$	Conversion time		14.25			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP**  
**M37702E2FS, M37702E2AFS, M37702E2BFS**
**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**
**M37702E2BXXXFP****ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_4$ , $P_5$ , $P_6$ , $P_7$ , $P_8 \sim P_8$	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_4$ , $P_5$ , $P_6$ , $P_7$ , $P_8 \sim P_8$	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage $P_{32}$	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_4$ , $P_5$ , $P_6$ , $P_7$ , $P_8 \sim P_8$	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3$ , $P_4$ , $P_5$ , $P_6$ , $P_7$ , $P_8 \sim P_8$	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage $P_{32}$	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.43	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA0_{IN} \sim TA4_{IN}$ , $TB0_{IN} \sim TB2_{IN}$ , $INT_0 \sim INT_2$ , AD <sub>TRG</sub> , CTS <sub>0</sub> , CTS <sub>1</sub> , CLK <sub>0</sub> , CLK <sub>1</sub>		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis $X_{IN}$		0.1		0.3	V
$I_{IH}$	High-level input current $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3 \sim P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $X_{IN}$ , RESET, CNV <sub>SS</sub> , BYTE	$V_I=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $P_0 \sim P_7$ , $P_1 \sim P_{17}$ , $P_2 \sim P_{27}$ , $P_3 \sim P_3$ , $P_4 \sim P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_8 \sim P_8$ , $X_{IN}$ , RESET, CNV <sub>SS</sub> , BYTE	$V_I=0V$			-5	$\mu A$
$V_{RAM}$	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 $V_{SS}$ during reset.	$f(X_{IN})=25MHz$ , square waveform	19	38	mA
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
$t_{CONV}$	Conversion time		9.12			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**TIMING REQUIREMENTS** ( $V_{CC}=5V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ , 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	
$t_{SU(RDY-\phi_1)}$	RDY input setup time	70		60		55		ns	
$t_{SU(HOLD-\phi_1)}$	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	
$t_h(\phi_1-RDY)$	RDY input hold time	0		0		0		ns	
$t_h(\phi_1-HOLD)$	HOLD input hold time	0		0		0		ns	

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**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 <sub>iIN</sub> input cycle time	250		125		80		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	125		62		40		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input cycle time	1000		500		320		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input cycle time	500		250		160		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iOUT</sub> input cycle time	5000		2500		2000		ns	
$t_{W(UPH)}$	TA <sub>iOUT</sub> input high-level pulse width	2500		1250		1000		ns	
$t_{W(UPL)}$	TA <sub>iOUT</sub> input low-level pulse width	2500		1250		1000		ns	
$t_{SU(UP-TIN)}$	TA <sub>iOUT</sub> input setup time	1000		500		400		ns	
$t_{H(TIN-UP)}$	TA <sub>iOUT</sub> input hold time	1000		500		400		ns	

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**Timer B input** (Count input in event counter mode)

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time (one edge count)	250		125		80		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width (one edge count)	125		62		40		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> input low-level pulse width (one edge count)	125		62		40		ns	
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time (both edges count)	500		250		160		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width (both edges count)	250		125		80		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(TB)}$	TBi <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TBi <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TBi <sub>IN</sub> 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.		
$t_{C(AD)}$	AD <sub>TRG</sub> input cycle time (minimum allowable trigger)	2000		1000		1000		ns	
$t_{W(ADL)}$	AD <sub>TRG</sub> 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 <sub>I</sub> input cycle time	500		250		200		ns	
$t_{W(CKH)}$	CLK <sub>I</sub> input high-level pulse width	250		125		100		ns	
$t_{W(CKL)}$	CLK <sub>I</sub> input low-level pulse width	250		125		100		ns	
$t_d(c-q)$	TxD <sub>I</sub> output delay time		150		90		80	ns	
$t_h(c-q)$	TxD <sub>I</sub> hold time	30		30		30		ns	
$t_{SU(D-C)}$	RxD <sub>I</sub> input setup time	60		30		20		ns	
$t_{H(c-d)}$	RxD <sub>I</sub> input hold time	90		90		90		ns	

**External interrupt INT<sub>i</sub> input**

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{W(INH)}$	INT <sub>i</sub> input high-level pulse width	250		250		250		ns	
$t_{W(INL)}$	INT <sub>i</sub> input low-level pulse width	250		250		250		ns	

**M37702E2-XXXFP, M37702E2AXXFP, M37702E2BXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXFP, M37702M2BXXFP**

**SWITCHING CHARACTERISTICS** ( $V_{CC}=5V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ , unless otherwise noted)

**Single-chip mode**

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 2		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	

**Memory expansion mode and microprocessor mode** (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 2	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	
$t_{d(\phi_1-HLDA)}$	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	
$t_{d(BHE-E)}$	BHE output delay time		100		30		20		ns	
$t_{d(R/W-E)}$	R/W output delay time		100		30		20		ns	
$t_{d(E-\phi_1)}$	$\phi_1$ 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	
$t_{pxz(E-P1Z)}$	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	
$t_{pxz(E-P2Z)}$	Port P2 floating release delay time		50		25		18		ns	
$t_h(E-BHE)$	BHE hold time		18		18		18		ns	
$t_h(E-R/W)$	R/W hold time		18		18		18		ns	
$t_w(EL)$	$\bar{E}$ pulse width		220		95		50		ns	

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**Memory expansion mode and microprocessor mode** (when wait bit = "0", and external memory area accessed)

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_d(P0A-E)$	Port P0 address output delay time	Fig. 2	100		30		12		ns	
$t_d(E-P1Q)$	Port P1 data output delay time (BYTE="L")			110		70		45	ns	
$t_{PZX}(E-P1Z)$	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	
$t_{PZX}(E-P2Z)$	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	
$t_d(\phi_1-HLDA)$	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	
$t_d(BHE-E)$	BHE output delay time		100		30		20		ns	
$t_d(R/W-E)$	R/W output delay time		100		30		20		ns	
$t_d(E-\phi_1)$	$\phi_1$ 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	
$t_h(E-BHE)$	BHE hold time		18		18		18		ns	
$t_h(E-R/W)$	R/W hold time		18		18		18		ns	
$t_W(EL)$	E pulse width		470		220		130		ns	

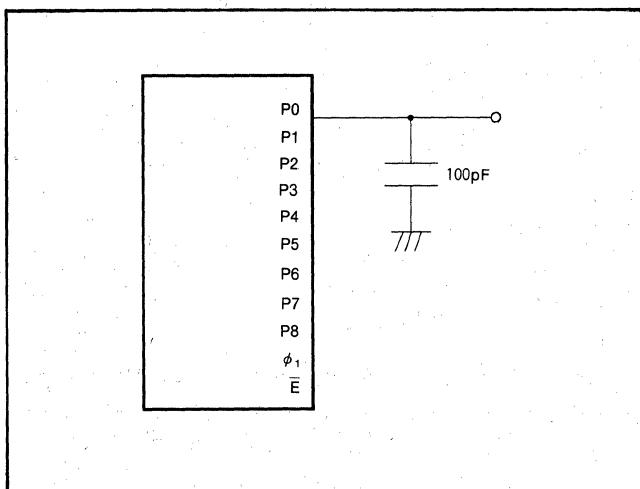


Fig. 2 Testing circuit for ports P0~P8,  $\phi_1$

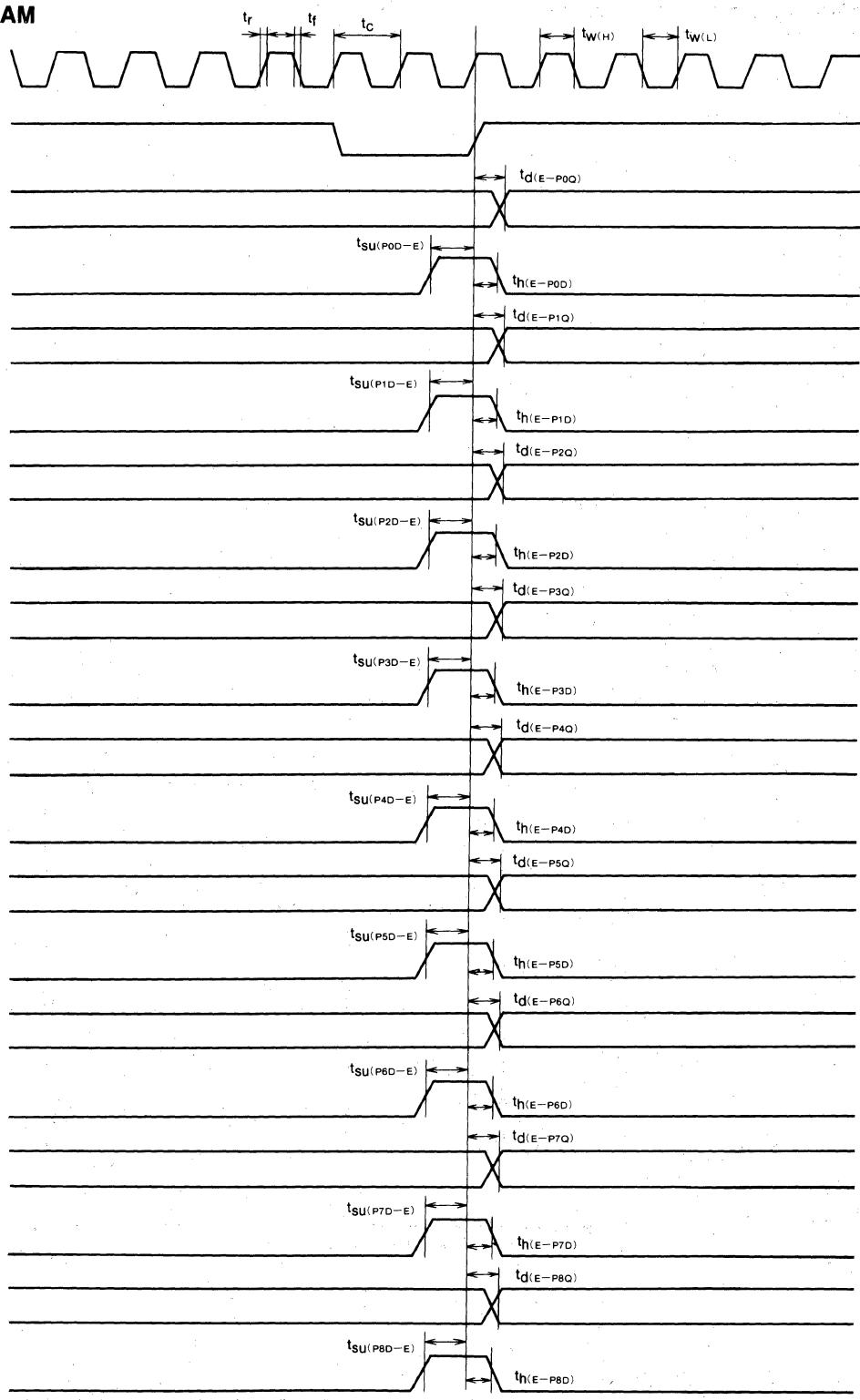
**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

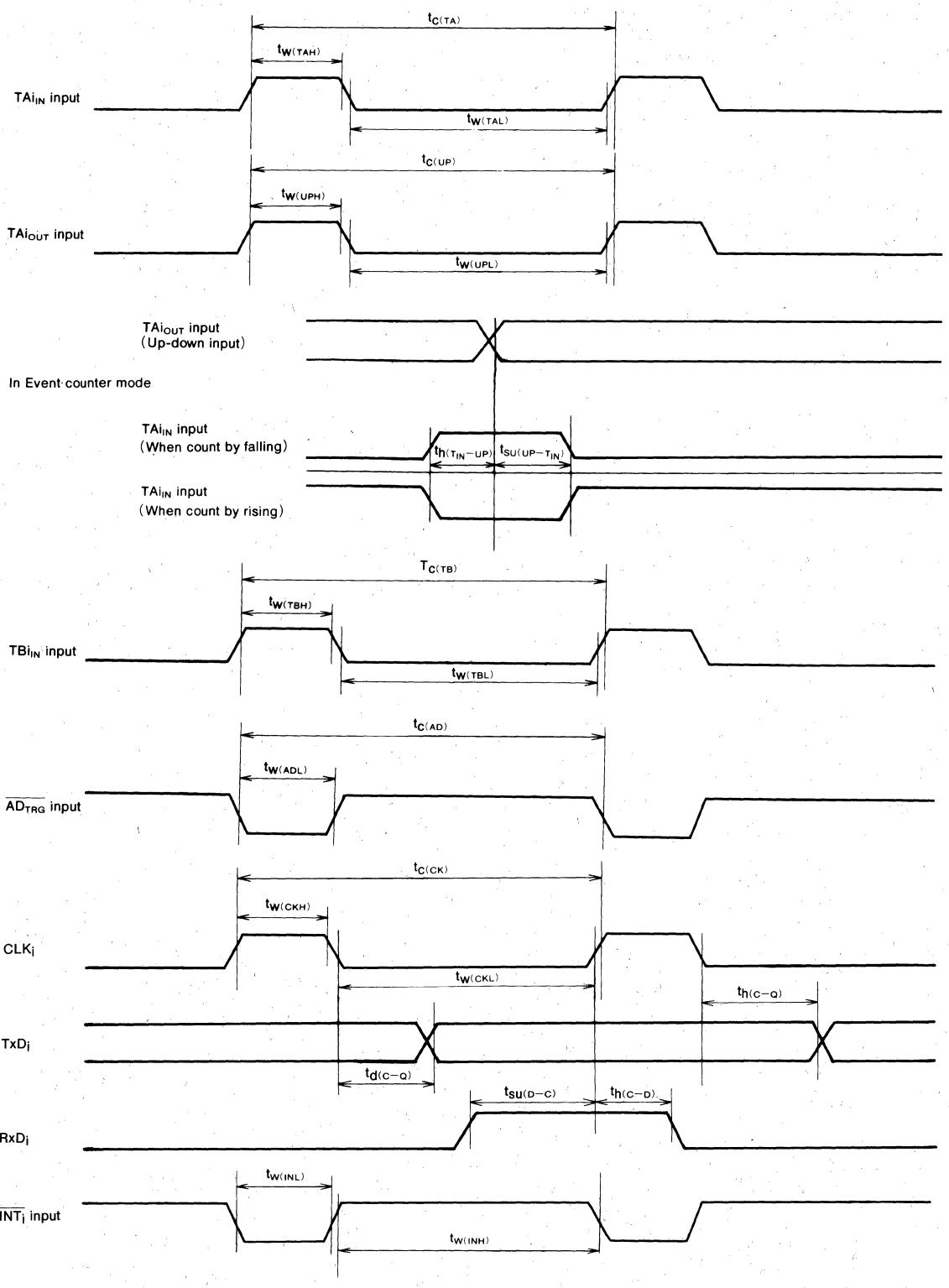
**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

**TIMING DIAGRAM**

Single-chip mode

$f(X_{IN})$



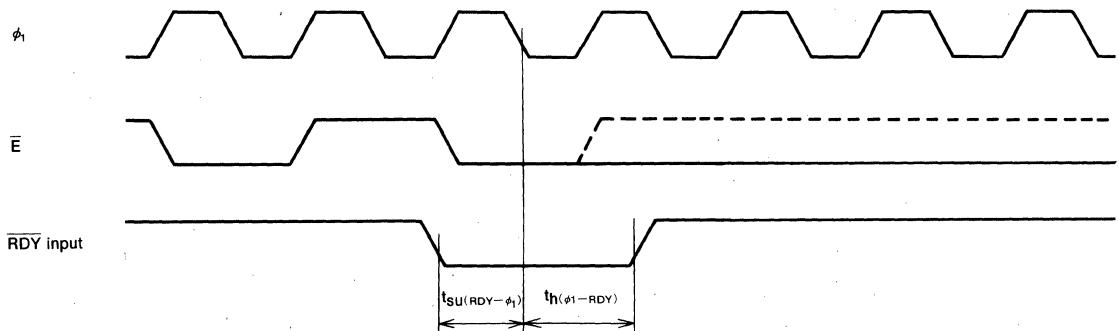
**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP**  
**M37702E2FS, M37702E2AFS, M37702E2BFS**
**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**


**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

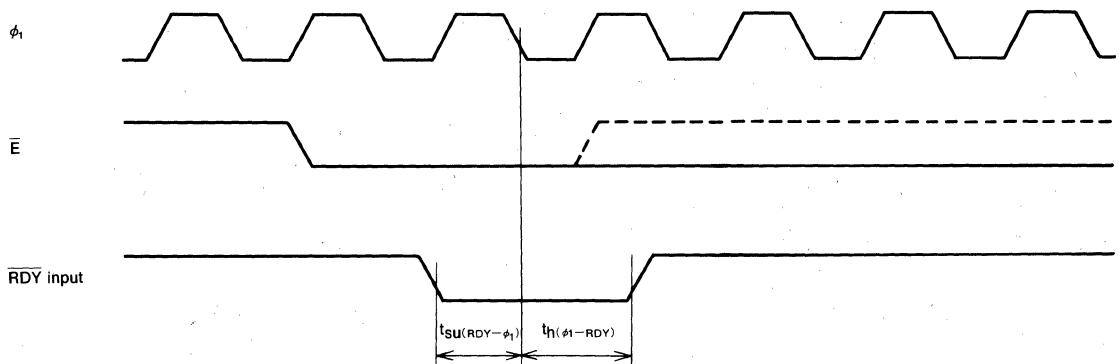
**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

Memory expansion mode and microprocessor mode

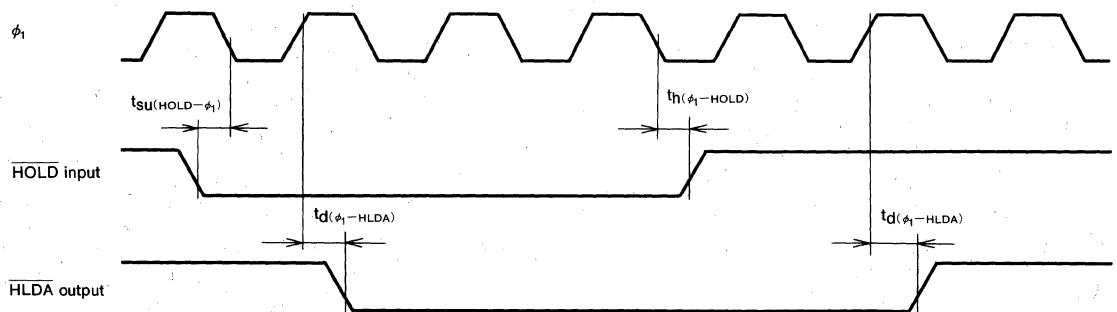
(When wait bit = "1")



(When wait bit = "0")



(When wait bit = "1" or "0" in common)



**Test conditions**

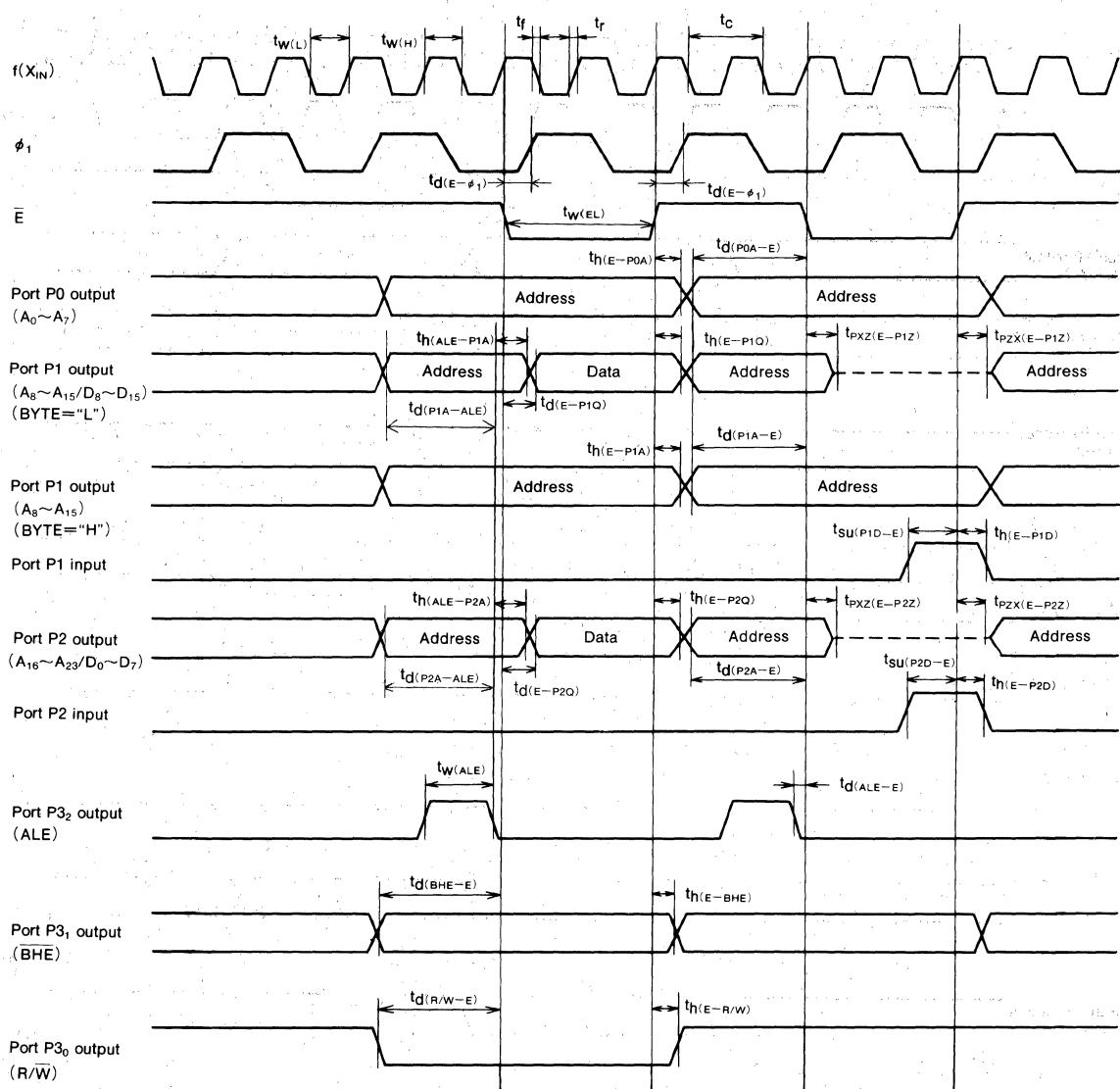
- $V_{CC} = 5 V \pm 10\%$
- Input timing voltage :  $V_{IL} = 1.0V$ ,  $V_{IH} = 4.0V$
- Output timing voltage :  $V_{OL} = 0.8V$ ,  $V_{OH} = 2.0V$

# M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP

## M37702E2FS, M37702E2AFS, M37702E2BFS

### PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP

**Memory expansion mode and microprocessor mode (When wait bit="1")**



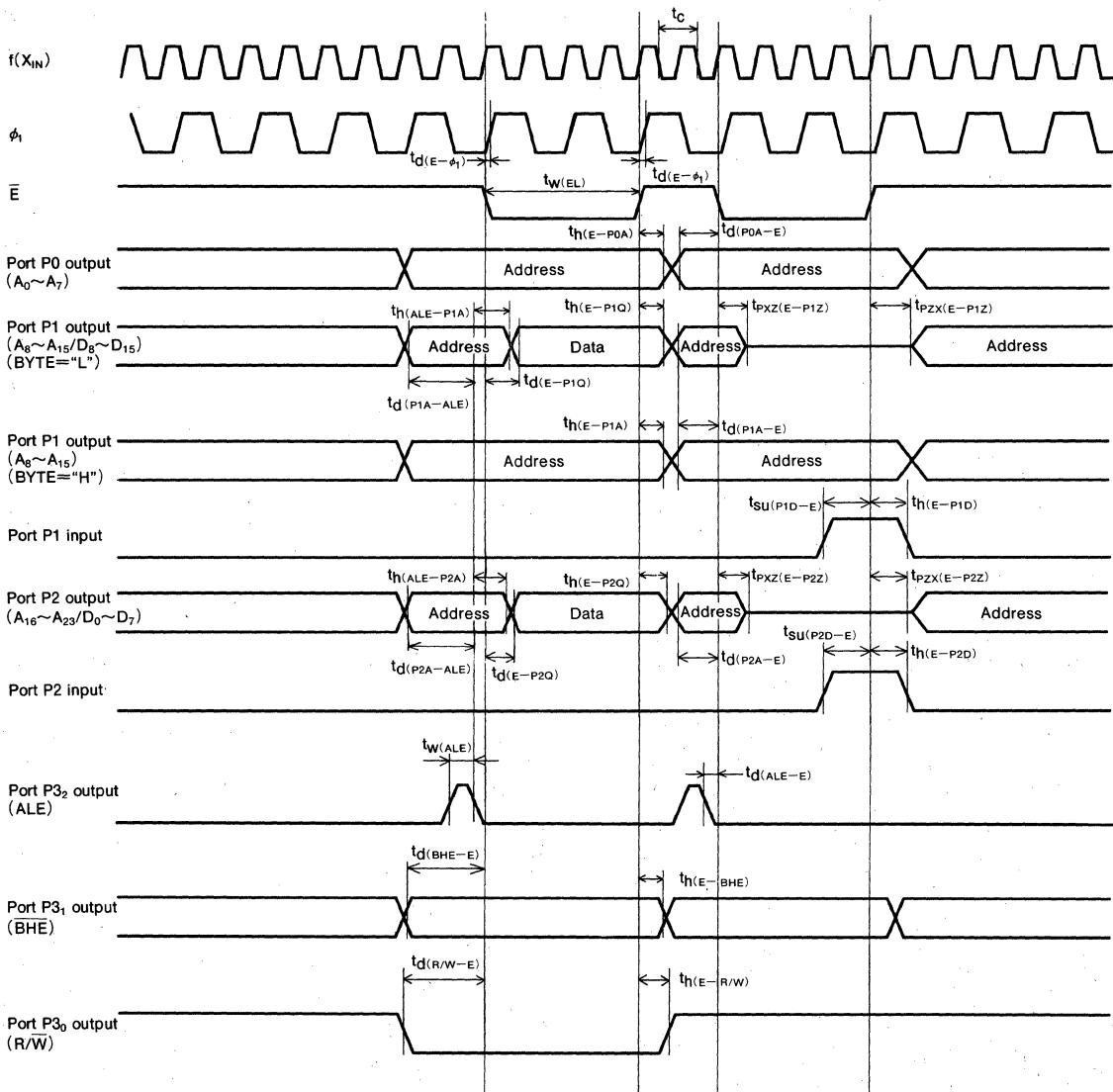
#### Test conditions

- $V_{CC} = 5 \text{ V} \pm 10\%$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$
- Ports P1, P2 input :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.5 \text{ V}$
- Port P4, input :  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$

**M37702E2-XXXFP, M37702E2AXXXFP, M37702E2BXXXFP  
M37702E2FS, M37702E2AFS, M37702E2BFS**

**PROM VERSION of M37702M2-XXXFP, M37702M2AXXXFP, M37702M2BXXXFP**

Memory expansion mode and microprocessor mode (When wait bit = "0", and external memory area is accessed)



Test conditions

- V<sub>CC</sub> = 5 V ± 10%
- Output timing voltage : V<sub>OL</sub> = 0.8V, V<sub>OH</sub> = 2.0V
- Ports P1, P2 input : V<sub>IL</sub> = 0.8V, V<sub>IH</sub> = 2.5V
- Port P4<sub>1</sub> input : V<sub>IL</sub> = 1.0V, V<sub>IH</sub> = 4.0V

# M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP

## M37702E4FS, M37702E4AFS, M37702E4BFS

PROM VERSION of M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP

### DESCRIPTION

The M37702E4-XXXFP, M37702E4AXXXFP and M37702E4BXXXFP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 80-pin plastic molded QFP. The features of these chips are similar to those of the M37702M4-XXXFP, M37702M4AXXXFP and M37702M4BXXXFP except that these chips have a 32K-byte PROM built in.

These single-chip microcomputers have 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. These microcomputers are suitable for office, business, and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, these chips are suitable for small quantity production runs. The M37702E4FS (8MHz version), M37702E4AFS (16MHz version) and M37702E4BFS (25MHz version) with erasable ROM that are housed in a windowed ceramic LCC are also provided.

The differences between M37702E4-XXXFP, M37702E4AXXXFP and M37702E4BXXXFP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37702E4-XXXFP unless otherwise noted.

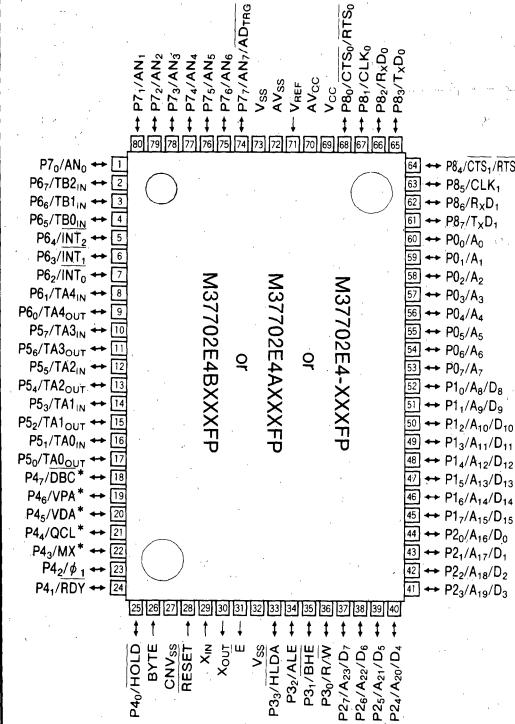
Type name	External clock input frequency
M37702E4-XXXFP	8 MHz
M37702E4AXXXFP	16MHz
M37702E4BXXXFP	25MHz

The M37702E4-XXXFP has the same functions as the M37702E2-XXXFP except for the memory size.

### FEATURES

- Number of basic instructions ..... 103
- Memory size PROM ..... 32K bytes
- RAM ..... 2048 bytes
- Instruction execution time  
M37702E4-XXXFP  
(The fastest instruction at 8 MHz frequency) ..... 500ns  
M37702E4AXXXFP  
(The fastest instruction at 16 MHz frequency) ..... 250ns  
M37702E4BXXXFP  
(The fastest instruction at 25 MHz frequency) ..... 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency) ..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) ..... 2
- 8-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output  
(ports P0, P1, P2, P3, P4, P5, P6, P7, P8) ..... 68

### PIN CONFIGURATION (TOP VIEW)



### Outline 80P6N

\*: Used in the evaluation chip mode only

### APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

### THE FUNCTIONS AND CHARACTERISTICS

The M37702E4-XXXFP has the same functions and characteristics as the M37702E2-XXXFP except for the PROM and RAM size. Refer to the section on the M37702E2-XXXFP.

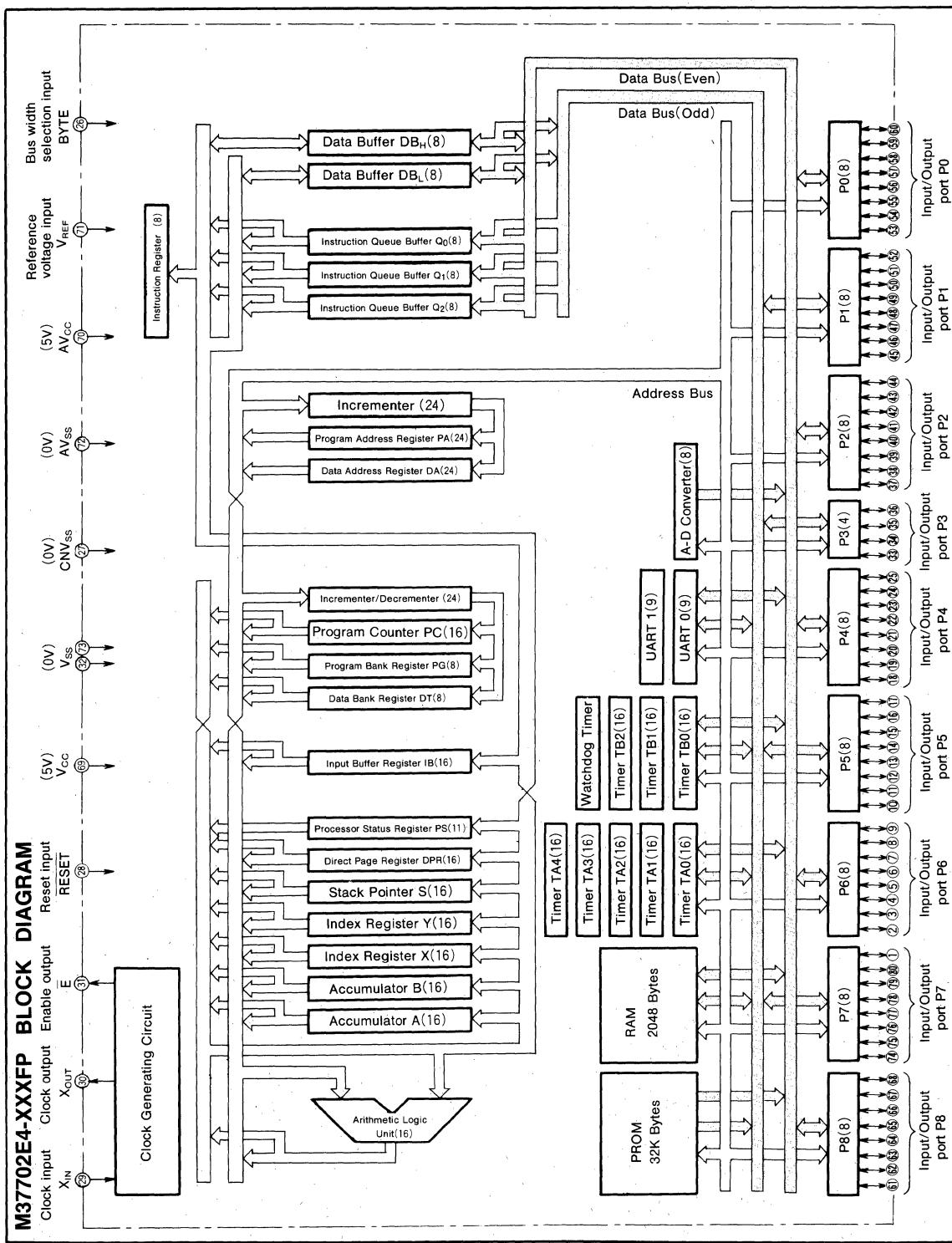
### DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM.

- (1) M37702E4-XXXFP writing to PROM order confirmation form
- (2) 80P6N mark specification form
- (3) ROM data (EPROM 3sets)

**M37702E4-XXXFP, M37702E4AXXXFP, M37702E4BXXXFP  
M37702E4FS, M37702E4AFS, M37702E4BFS**

**PROM VERSION of M37702M4-XXXFP, M37702M4AXXXFP, M37702M4BXXXFP**



# M37703E2-XXXSP, M37703E2AXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP

## DESCRIPTION

The M37703E2-XXXSP, M37703E2AXXSP and M37703E2BXXXSP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 64-pin shrink plastic molded DIP. The features of these chips are similar to those of the M37703M2-XXXSP, M37703M2AXXSP and M37703M2BXXXSP except that this chip has a 16K-byte PROM built in.

These single-chip microcomputers have 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. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, this chip is suitable for small quantity production runs.

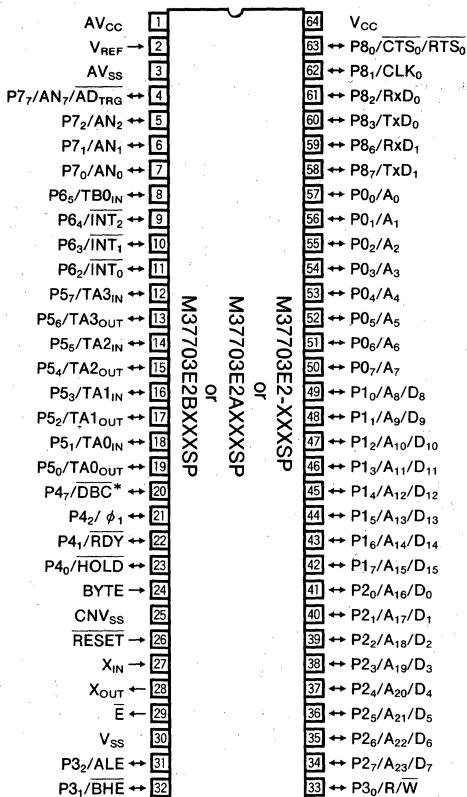
The differences between M37703E2-XXXSP, M37703E2AXXSP and M37703E2BXXXSP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703E2-XXXSP unless otherwise noted.

Type name	External clock input frequency
M37703E2-XXXSP	8 MHz
M37703E2AXXSP	16MHz
M37703E2BXXXSP	25MHz

## FEATURES

- Number of basic instructions ..... 103
- Memory size     PROM(one time) ..... 16K bytes
- RAM ..... 512 bytes
- Instruction execution time
  - M37703E2-XXXSP  
(The fastest instruction at 8 MHz frequency) ..... 500ns
  - M37703E2AXXSP  
(The fastest instruction at 16 MHz frequency) ..... 250ns
  - M37703E2BXXXSP  
(The fastest instruction at 25 MHz frequency) ..... 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)  
..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) ..... 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output  
(ports P0, P1, P2, P3, P4, P5, P6, P7, P8) ..... 53

## PIN CONFIGURATION (TOP VIEW)



## Outline 64P4B

\*: Used in the evaluation chip mode only

## APPLICATION

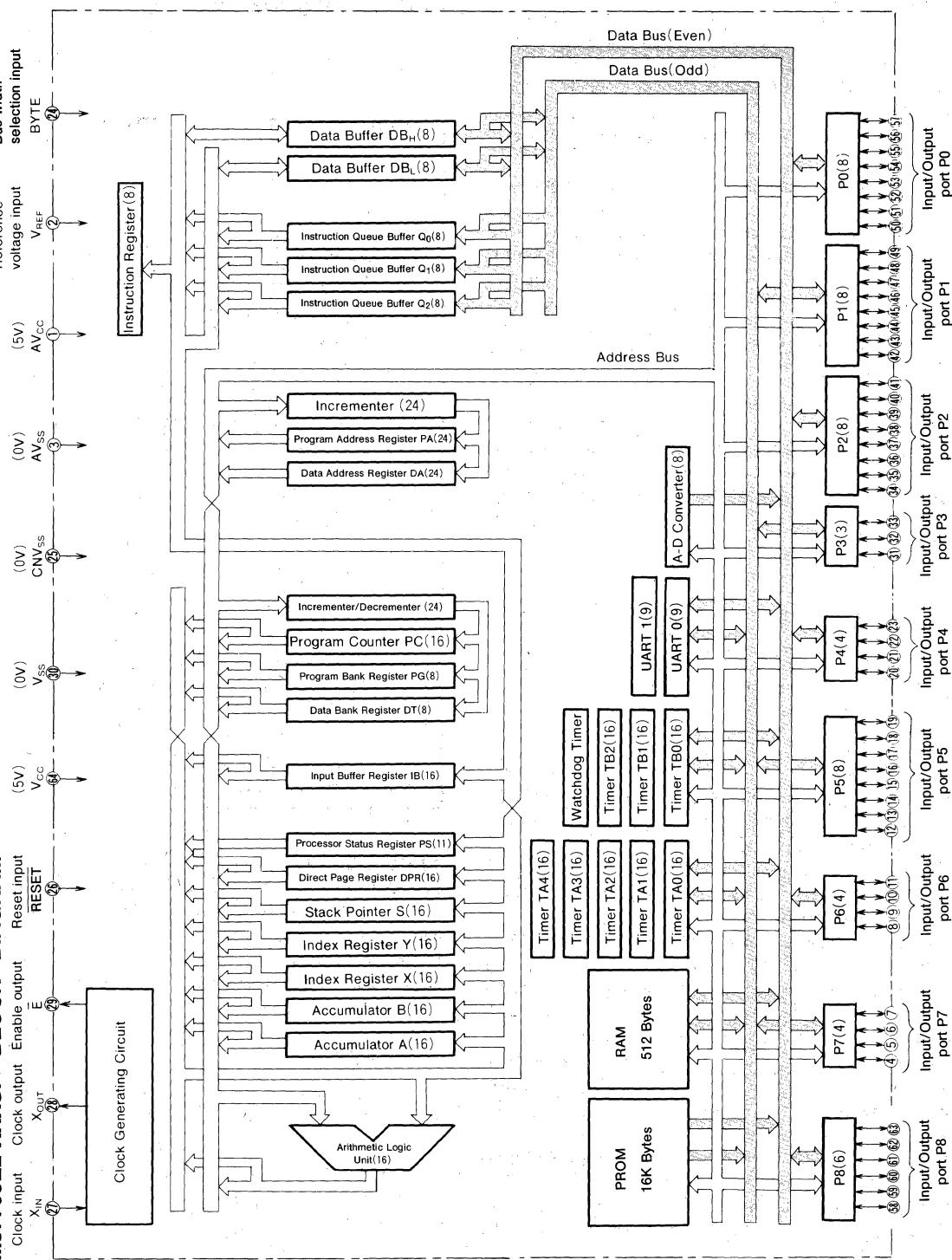
Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

## M37703E2-XXXSP BLOCK DIAGRAM



**M37703E2-XXXSP, M37703E2AXXXSP  
M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

**FUNCTIONS OF M37703E2-XXXSP**

Parameter	Functions
Number of basic instructions	103
Instruction execution time	M37703E2-XXXSP 500ns (the fastest instructions, at 8MHz frequency)
	M37703E2AXXXSP 250ns (the fastest instructions, at 16MHz frequency)
	M37703E2BXXXSP 160ns (the fastest instructions, at 25MHz frequency)
Memory size	PROM 16K bytes
	RAM 512 bytes
Input/Output ports	P0, P1, P2, P5 8-bit×4
	P8 6-bit×1
	P4, P6, P7 4-bit×3
	P3 3-bit×1
Multi-function timers	TA0, TA1, TA2, TA3, TA4 16-bit×5 (4 Input/Output functions)
	TB0, TB1, TB2 16-bit×3 (1 Input function)
Serial I/O	UART×2 (One can be set clock synchronous serial I/O.)
A-D converter	8-bit×1 (4 channels)
Watchdog timer	12-bit×1
Interrupts	3 external types, 16 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit	Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage	5 V±10%
Power dissipation	30mW(at external 8 MHz frequency)
Input/Output characteristic	Input/Output voltage 5 V
	Output current 5 mA
Memory expansion	Maximum 16M bytes
Operating temperature range	-20~85°C
Device structure	CMOS high-performance silicon gate process
Package	64-pin shrink plastic molded DIP

**M37703E2-XXXSP, M37703E2AXXXSP  
M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

**PIN DESCRIPTION (NORMAL MODE)**

Pin	Name	Input/Output	Functions
V <sub>CC</sub> , V <sub>SS</sub>	Power supply		Supply 5 V±10% to V <sub>CC</sub> and 0 V to V <sub>SS</sub> .
CNV <sub>SS</sub>	CNV <sub>SS</sub> input	Input	This pin controls the processor mode. Connect to V <sub>SS</sub> for single-chip mode.
RESET	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X <sub>IN</sub>	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X <sub>IN</sub> and X <sub>OUT</sub> . When an external clock is used, the clock source should be connected to the X <sub>IN</sub> pin
X <sub>OUT</sub>	Clock output	Output	and the X <sub>OUT</sub> pin should be left open.
E	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	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 "L" signal inputs and 8 bits when "H" signal inputs.
AV <sub>CC</sub> , AV <sub>SS</sub>	Analog supply input		Power supply for the A-D converter. Connect AV <sub>CC</sub> to V <sub>CC</sub> and AV <sub>SS</sub> to V <sub>SS</sub> externally.
V <sub>REF</sub>	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
P0 <sub>0</sub> ~P0 <sub>7</sub>	I/O port P0	I/O	In single-chip mode, port P0 becomes an 8-bit I/O port. An I/O directional register is available so that each pin can be programmed for input or output. These ports are in input mode when reset. Address(A <sub>7</sub> ~A <sub>0</sub> ) is output in memory expansion mode or microprocessor mode.
P1 <sub>0</sub> ~P1 <sub>7</sub>	I/O port P1	I/O	In single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in memory expansion mode or microprocessor mode and external data bus is 16-bit width, high-order data (D <sub>15</sub> ~D <sub>8</sub> ) is input or output when E output is "L" and an address (A <sub>15</sub> ~A <sub>8</sub> ) is output when E output is "H". If the BYTE pin is "H" that is an external data bus is 8-bit width, only address(A <sub>15</sub> ~A <sub>8</sub> ) is output.
P2 <sub>0</sub> ~P2 <sub>7</sub>	I/O port P2	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode low-order data(D <sub>7</sub> ~D <sub>0</sub> ) is input or output when E output is "L" and an address(A <sub>23</sub> ~A <sub>16</sub> ) is output when E output is "H".
P3 <sub>0</sub> ~P3 <sub>2</sub>	I/O port P3	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, R/W, BHE, and ALE signals are output.
P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub>	I/O port P4	I/O	In single-chip mode, these pins have the same functions as port P0. In memory expansion mode or microprocessor mode, P <sub>40</sub> and P <sub>44</sub> become HOLD and RDY input pin respectively. Functions of other pins are the same as in single-chip mode. In single-chip mode or memory expansion mode, port P4 <sub>2</sub> can be programmed for φ <sub>1</sub> output pin divided the clock to X <sub>IN</sub> pin by 2. In microprocessor mode, P4 <sub>2</sub> always has the function as φ <sub>1</sub> output pin.
P5 <sub>0</sub> ~P5 <sub>7</sub>	I/O port P5	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as I/O pins for timer A0, timer A1, timer A2, and timer A3.
P6 <sub>2</sub> ~P6 <sub>5</sub>	I/O port P6	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as external interrupt input INT <sub>0</sub> , INT <sub>1</sub> , and INT <sub>2</sub> pins, and input pins for timer B0.
P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub>	I/O port P7	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as analog input AN <sub>0</sub> ~AN <sub>2</sub> and AN <sub>7</sub> input pins. P7 <sub>7</sub> also has an A-D conversion trigger input function.
P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	I/O port P8	I/O	In addition to having the same functions as port P0 in single-chip mode, these pins also function as Rx <sub>D</sub> , TxD, CLK, CTS/RTS pins for UART 0, and as Rx <sub>D</sub> , TxD pins for UART 1.

**M37703E2-XXXSP, M37703E2AXXSP  
M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP**

**PIN DESCRIPTION (EPROM MODE)**

Pin	Name	Input/Output	Functions
V <sub>CC</sub> , V <sub>SS</sub>	Power supply		Supply 5V±10% to V <sub>CC</sub> and 0V to V <sub>SS</sub> .
CNV <sub>SS</sub>	V <sub>PP</sub> input	Input	Connect to V <sub>PP</sub> when programming or verifying.
BYTE	V <sub>PP</sub> input	Input	Connect to V <sub>PP</sub> when programming or verifying.
RESET	Reset input	Input	Connect to V <sub>SS</sub> .
X <sub>IN</sub>	Clock input	Input	Connect a ceramic resonator between X <sub>IN</sub> and X <sub>OUT</sub> .
X <sub>OUT</sub>	Clock output	Output	
Ē	Enable output	Output	Keep open.
AV <sub>CC</sub> , AV <sub>SS</sub>	A-D power supply		Connect AV <sub>CC</sub> to V <sub>CC</sub> and AV <sub>SS</sub> to V <sub>SS</sub> .
V <sub>REF</sub>	Reference voltage input	Input	Connect to V <sub>SS</sub> .
P0 <sub>0</sub> ~P0 <sub>7</sub>	Address input (A <sub>0</sub> ~A <sub>7</sub> )	Input	Port P0 functions as the lower 8 bits address input (A <sub>0</sub> ~A <sub>7</sub> ).
P1 <sub>0</sub> ~P1 <sub>7</sub>	Address input (A <sub>8</sub> ~A <sub>14</sub> )	Input	Port P1 <sub>0</sub> ~P1 <sub>6</sub> functions as the higher 7 bits address input (A <sub>8</sub> ~A <sub>14</sub> ). Connect P1 <sub>7</sub> to V <sub>CC</sub> .
P2 <sub>0</sub> ~P2 <sub>7</sub>	Data I/O (D <sub>0</sub> ~D <sub>7</sub> )	I/O	Port P2 functions as the 8 bits data bus (D <sub>0</sub> ~D <sub>7</sub> ).
P3 <sub>0</sub> ~P3 <sub>2</sub>	Input port P3	Input	Connect to V <sub>SS</sub> .
P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub>	Input port P4	Input	Connect to V <sub>SS</sub> .
P5 <sub>0</sub> ~P5 <sub>7</sub>	Control signal input	Input	P5 <sub>1</sub> and P5 <sub>2</sub> functions as OE and CE input pin. Connect P5 <sub>0</sub> , P5 <sub>3</sub> , P5 <sub>4</sub> and P5 <sub>5</sub> to V <sub>CC</sub> . Connect P5 <sub>6</sub> and P5 <sub>7</sub> to V <sub>SS</sub> .
P6 <sub>2</sub> ~P6 <sub>5</sub>	Input port P6	Input	Connect to V <sub>SS</sub> .
P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub>	Input port P7	Input	Connect to V <sub>SS</sub> .
P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	Input port P8	Input	Connect to V <sub>SS</sub> .

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

### EPROM MODE

The M37703E2-XXXSP features an EPROM mode in addition to its normal modes. When the RESET signal level is "L", the chip automatically enters the EPROM mode. Table 1 list the correspondence between pins and Fig. 1 gives the pin connections in the EPROM mode.

When in the EPROM mode, ports P0, P1, P2, P5<sub>1</sub>, P5<sub>2</sub>, CNV<sub>ss</sub> and BYTE are used for the EPROM (equivalent to the M5M27C256K). When in this mode, the built-in PROM can be written to or read from using these pins in the same way as with the M5M27C256K.

This chip does not have Device Identifier Mode, so that set the corresponding program algorithm. The program area should specify address 4000<sub>16</sub>~7FFF<sub>16</sub> for the M37703E2-

XXXSP.

Connect the clock which is either ceramic resonator or external clock to X<sub>IN</sub> pin and X<sub>OUT</sub> pin.

#### 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 M37703E2BXXXSP, 1M mode may become standard.

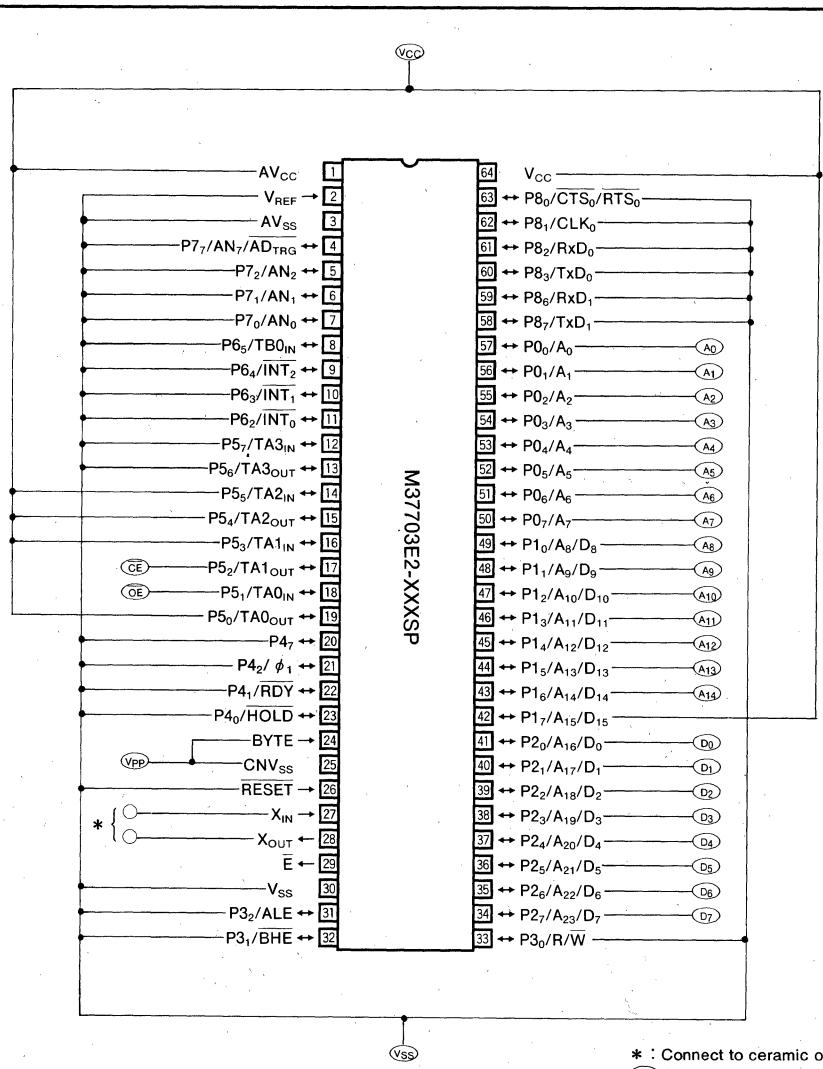


Fig. 1 Pin connection in EPROM programming mode

# M37703E2-XXXSP, M37703E2AXXSP M37703E2BXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXSP

Table 1. Pin function in EPROM programming mode

	M37703E2-XXXSP	M5M27C256K
V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>
V <sub>PP</sub>	CNV <sub>SS</sub> , BYTE	V <sub>PP</sub>
V <sub>SS</sub>	V <sub>SS</sub>	V <sub>SS</sub>
Address input	Ports P0, P1 <sub>0</sub> ~P1 <sub>6</sub>	A <sub>0</sub> ~A <sub>14</sub>
Data I/O	Port P2	D <sub>0</sub> ~D <sub>7</sub>
CE	P <sub>5</sub> <sub>2</sub>	'CE
OE	P <sub>5</sub> <sub>1</sub>	OE

## FUNCTION IN EPROM MODE

### Reading

To read the PROM, set the CE and OE pins to a "L" level. Input the address of the data (A<sub>0</sub>~A<sub>14</sub>) to be read and the data will be output to the I/O pins D<sub>0</sub>~D<sub>7</sub>. The data I/O pins will be floating when either the CE or OE pins are in the "H" state.

### Writing

To write to the PROM, set the OE pin to a "H" level. The CPU will enter the program mode when V<sub>PP</sub> is applied to the V<sub>PP</sub> pin. The address to be written to is selected with pins A<sub>0</sub>~A<sub>14</sub>, and the data to be written is input to pins D<sub>0</sub>~D<sub>7</sub>. Set the CE pin to a "L" level to begin writing.

## FAST PROGRAMMING ALGORITHM

To program the M37703E2-XXXSP with fast programming algorithm, first set V<sub>CC</sub>=6V, V<sub>PP</sub>=12.5, and set the address to "0". Apply a 1ms write pulse, check that the data can be read, and if it cannot be read OK, repeat the procedure, applying a 1ms write pulse and checking that the data can be read until it can be read OK. Record the accumulated number of pulse applied (N) before the data can be read OK, and then write the data again, applying a further three times this number of pulses (3N ms).

When this series of write operations is complete, increment the address, and continue to repeat the procedure above until the last address has been reached.

Finally, when all addresses have been written, read with V<sub>CC</sub>=V<sub>PP</sub>=5V (or V<sub>CC</sub>=V<sub>PP</sub>=5.25V).

Table 2. I/O signal in each mode

Pin Mode	CE	OE	V <sub>PP</sub>	V <sub>CC</sub>	Data I/O
Read-out	V <sub>IL</sub>	V <sub>IL</sub>	5V	5V	Output
Output	V <sub>IL</sub>	V <sub>IH</sub>	5V	5V	Floating
Disable	V <sub>IH</sub>	X	5V	5V	Floating
Programming	V <sub>IL</sub>	V <sub>IH</sub>	12.5V	6V	Input
Programming Verify	V <sub>IH</sub>	V <sub>IL</sub>	12.5V	6V	Output
Program Disable	V <sub>IH</sub>	V <sub>IH</sub>	12.5V	6V	Floating

Note 1 : An X indicates either V<sub>IL</sub> or V<sub>IH</sub>.

## Program operation

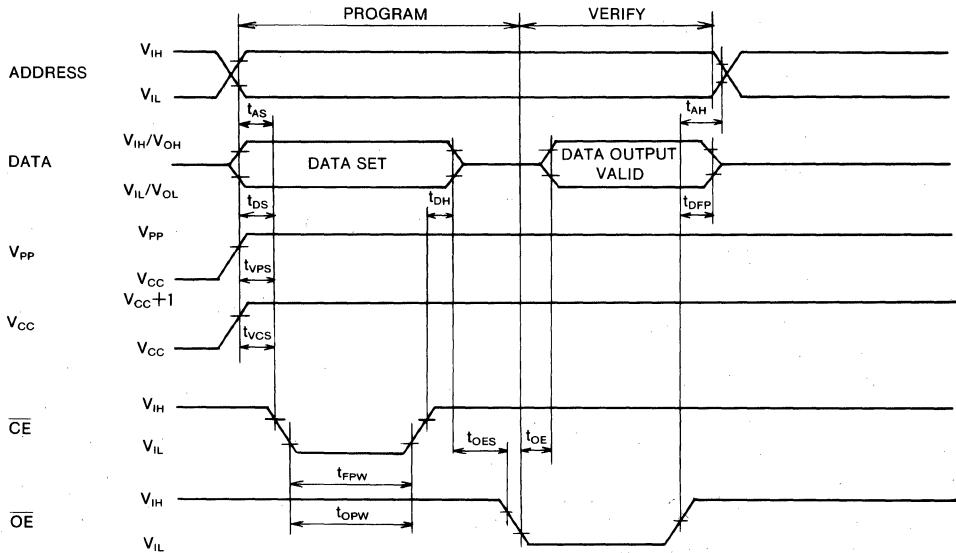
### AC ELECTRICAL CHARACTERISTICS (T<sub>a</sub>=25±5°C, V<sub>CC</sub>=6V±0.25V, V<sub>PP</sub>=12.5±0.3V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>AS</sub>	Address setup time			2		μs
t <sub>OE</sub>	OE setup time			2		μs
t <sub>DS</sub>	Data setup time			2		μs
t <sub>AH</sub>	Address hold time			0		μs
t <sub>DH</sub>	Data hold time			2		μs
t <sub>DFF</sub>	Output enable to output float delay			0	130	ns
t <sub>VCS</sub>	V <sub>CC</sub> setup time			2		μs
t <sub>VPS</sub>	V <sub>PP</sub> setup time			2		μs
t <sub>FPW</sub>	CE initial program pulse width		0.95	1	1.05	ms
t <sub>OPW</sub>	CE over program pulse width		2.85	78.75		ms
t <sub>OE</sub>	Data valid from OE				150	ns

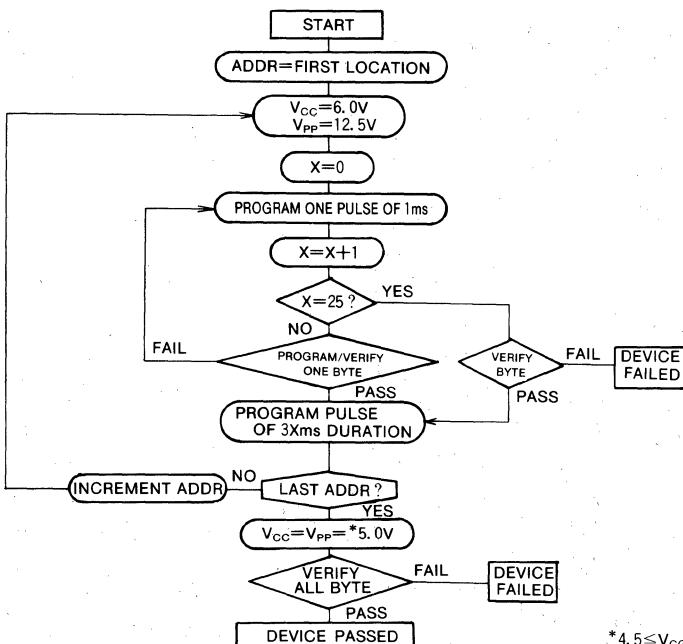
# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

## AC waveforms



## Fast programming algorithm flow chart



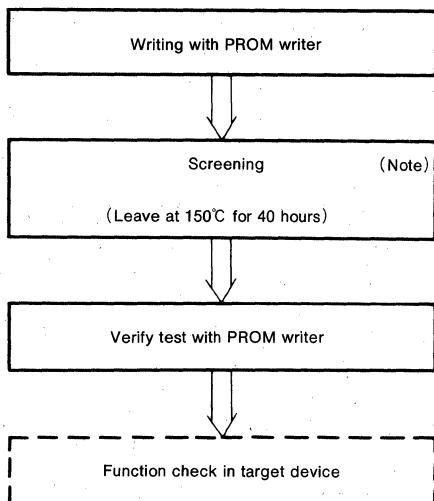
\* $4.5 \leq V_{CC} = V_{PP} \leq 5.5$  V

# M37703E2-XXXSP, M37703E2AXXSP M37703E2BXXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP

### CAUTION: UNITS SHIPPED AS BLANKS

The programmable M37703E2SP, M37703E2ASP and M37703E2ESP that are shipped in blank are also provided. For the M37703E2SP, M37703E2ASP and M37703E2BSP, Mitsubishi Electric corp. does not perform PROM write test and screening following the assembly processes. To improve reliability after write, performing write and test according to the flow below before use is recommended.



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

### BASIC FUNCTION BLOCKS

Since these processors operate in exactly the same way as the M37703M2-XXXSP, refer to the section on the M37703M2-XXXSP.

### ADDRESSING MODES

The M37703E2-XXXSP has 28 powerful addressing modes. Refer to the MELPS 7700 addressing mode description for the details of each addressing mode.

### MACHINE INSTRUCTION LIST

The M37703E2-XXXSP has 103 machine instructions. Refer to the MELPS 7700 machine instruction list for details.

### DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM .

- (1) M37703E2-XXXSP writing to PROM order confirmation form
- (2) 64P4B mark specification form for one time PROM
- (3) ROM data (EPROM 3sets)

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
$V_{CC}$	Supply voltage		-0.3~7	V
$AV_{CC}$	Analog supply voltage		-0.3~7	V
$V_I$	Input voltage RESET, CNV <sub>SS</sub> , BYTE		-0.3~12 (Note1)	V
$V_I$	Input voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , V <sub>REF</sub> , X <sub>IN</sub>		-0.3~ $V_{CC}$ +0.3	V
$V_O$	Output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>OUT</sub> , E		-0.3~ $V_{CC}$ +0.3	V
$P_d$	Power dissipation	T <sub>a</sub> =25°C	1000	mW
$T_{opr}$	Operating temperature		-20~85	°C
$T_{stg}$	Storage temperature		-40~150	°C

Note 1. Input voltage for CNV<sub>SS</sub> and BYTE pins is 13V in writing to PROM.

### RECOMMENDED OPERATING CONDITIONS ( $V_{CC}=5V \pm 5\%$ , $T_a=-20\sim 85^\circ C$ , unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$V_{CC}$	Supply voltage	4.5	5.0	5.5	V
$AV_{CC}$	Analog supply voltage		$V_{CC}$		V
$V_{SS}$	Supply voltage		0		V
$AV_{SS}$	Analog supply voltage		0		V
$V_{IH}$	High-level input voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	High-level input voltage P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> (in single-chip mode)	0.8 $V_{CC}$		$V_{CC}$	V
$V_{IH}$	High-level input voltage P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> (in memory expansion mode and microprocessor mode)	0.5 $V_{CC}$		$V_{CC}$	V
$V_{IL}$	Low-level input voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0		0.2 $V_{CC}$	V
$V_{IL}$	Low-level input voltage P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> (in single-chip mode)	0		0.2 $V_{CC}$	V
$V_{IL}$	Low-level input voltage P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> (in memory expansion mode and microprocessor mode)	0		0.16 $V_{CC}$	V
$I_{OH(peak)}$	High-level peak output current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>			-10	mA
$I_{OH(avg)}$	High-level average output current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>			-5	mA
$I_{OL(peak)}$	Low-level peak output current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>			10	mA
$I_{OL(avg)}$	Low-level average output current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>			5	mA
$f(X_{IN})$	External clock frequency input	M37703E2-XXXSP		8	MHz
		M37703E2AXXXSP		16	
		M37703E2BXXXSP		25	

Note 2. Average output current is the average value of a 100ms interval.

3. 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.

**M37703E2-XXXSP, M37703E2AXXSP  
M37703E2BXXXSP**
**PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP**
**M37703E2-XXXSP****ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3$ , $P_1$ , $P_4 \sim P_2$ , $P_4$ , $P_5 \sim P_7$ , $P_6 \sim P_5$ , $P_7 \sim P_7$ , $P_7$ , $P_8 \sim P_8$ , $P_8$ , $P_8$	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3$ , $P_1$	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage $P_3$	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage $P_3$	$I_{OH}=-400\mu A$	4.8			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-10mA$	3.4			V
$V_{OH}$	High-level output voltage $\bar{E}$	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3$ , $P_1$ , $P_4 \sim P_2$ , $P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_7$ , $P_8 \sim P_8$ , $P_8$ , $P_8$	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3$ , $P_1$	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage $P_3$	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage $P_3$	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA3 <sub>IN</sub> , TB0 <sub>IN</sub> , INT <sub>0</sub> ~INT <sub>2</sub> , AD <sub>TRG</sub> , CTS <sub>0</sub> , CLK <sub>0</sub>			0.4	1	V
$V_{T+}-V_{T-}$	Hysteresis RESET			0.2	0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>			0.1	0.3	V
$I_{IH}$	High-level input current $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3 \sim P_3$ , $P_4 \sim P_2$ , $P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_7$ , $P_8 \sim P_8$ , $P_8$ , $P_8$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current $P_0 \sim P_7$ , $P_1 \sim P_7$ , $P_2 \sim P_7$ , $P_3 \sim P_3$ , $P_4 \sim P_2$ , $P_4$ , $P_5 \sim P_5$ , $P_6 \sim P_6$ , $P_7 \sim P_7$ , $P_7$ , $P_8 \sim P_8$ , $P_8$ , $P_8$ , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	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 V <sub>SS</sub> during reset.	$f(X_{IN})=8MHz$ , square waveform	6	12	mA
$I_{CC}$		$T_a=25^\circ C$ when clock is stopped.			1	$\mu A$
$I_{CC}$		$T_a=85^\circ C$ when clock is stopped.			20	$\mu A$

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=8MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k $\Omega$
$t_{CONV}$	Conversion time		28.5			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**MITSUBISHI MICROCOMPUTERS**  
**M37703E2-XXXSP, M37703E2AXXXSP**  
**M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

**M37703E2AXXXSP**

**ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub>	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage P3 <sub>2</sub>	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage P3 <sub>2</sub>	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	High-level output voltage $\bar{E}$	$I_{OL}=-10mA$	3.4			V
$V_{OL}$	High-level output voltage $\bar{E}$	$I_{OL}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub>	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage P3 <sub>2</sub>	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage P3 <sub>2</sub>	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage $\bar{E}$	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA3 <sub>IN</sub> , TB0 <sub>IN</sub> , INT <sub>0</sub> ~INT <sub>2</sub> , AD <sub>TRG</sub> , CTS <sub>0</sub> , CLK <sub>0</sub>		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>		0.1		0.3	V
$I_{IH}$	High-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	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 V <sub>SS</sub> during reset.	$f(X_{IN})=16MHz$ , square waveform	12	24	mA
			$T_a=25^\circ C$ when clock is stopped.		1	$\mu A$
			$T_a=85^\circ C$ when clock is stopped.		20	

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=16MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
$t_{CONV}$	Conversion time		14.25			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

## M37703E2BXXXSP

**ELECTRICAL CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	$I_{OH}=-10mA$	3			V
$V_{OH}$	High-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub>	$I_{OH}=-400\mu A$	4.7			V
$V_{OH}$	High-level output voltage P3 <sub>2</sub>	$I_{OH}=-10mA$	3.1			V
$V_{OH}$	High-level output voltage P3 <sub>2</sub>	$I_{OH}=-400\mu A$	4.8			V
$V_{OH}$	High-level output voltage E	$I_{OH}=-10mA$	3.4			V
$V_{OH}$	High-level output voltage E	$I_{OH}=-400\mu A$	4.8			V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	$I_{OL}=10mA$			2	V
$V_{OL}$	Low-level output voltage P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub>	$I_{OL}=2mA$			0.45	V
$V_{OL}$	Low-level output voltage P3 <sub>2</sub>	$I_{OL}=10mA$			1.9	V
$V_{OL}$	Low-level output voltage P3 <sub>2</sub>	$I_{OL}=2mA$			0.43	V
$V_{OL}$	Low-level output voltage E	$I_{OL}=10mA$			1.6	V
$V_{OL}$	Low-level output voltage E	$I_{OL}=2mA$			0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA0 <sub>IN</sub> ~TA3 <sub>IN</sub> , TB0 <sub>IN</sub> , INT <sub>0</sub> ~INT <sub>2</sub> , AD <sub>TRG</sub> , CTS <sub>0</sub> , CLK <sub>0</sub>			0.4	1	V
$V_{T+}-V_{T-}$	Hysteresis RESET			0.2	0.5	V
$V_{T+}-V_{T-}$	Hysteresis X <sub>IN</sub>			0.1	0.3	V
$I_{IH}$	High-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=5V$			5	$\mu A$
$I_{IL}$	Low-level input current P0 <sub>0</sub> ~P0 <sub>7</sub> , P1 <sub>0</sub> ~P1 <sub>7</sub> , P2 <sub>0</sub> ~P2 <sub>7</sub> , P3 <sub>0</sub> ~P3 <sub>2</sub> , P4 <sub>0</sub> ~P4 <sub>2</sub> , P4 <sub>7</sub> , P5 <sub>0</sub> ~P5 <sub>7</sub> , P6 <sub>2</sub> ~P6 <sub>5</sub> , P7 <sub>0</sub> ~P7 <sub>2</sub> , P7 <sub>7</sub> , P8 <sub>0</sub> ~P8 <sub>3</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	$V_i=0V$			-5	$\mu A$
$V_{RAM}$	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 V <sub>SS</sub> during reset.	$f(X_{IN})=25MHz$ , square waveform	19	38	mA
			$T_a=25^\circ C$ when clock is stopped.		1	
			$T_a=85^\circ C$ when clock is stopped.		20	$\mu A$

**A-D CONVERTER CHARACTERISTICS** ( $V_{CC}=5V$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ ,  $f(X_{IN})=25MHz$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
$t_{CONV}$	Conversion time			9.12		$\mu s$
$V_{REF}$	Reference voltage			2		V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

**MITSUBISHI MICROCOMPUTERS**  
**M37703E2-XXXSP, M37703E2AXXSP**  
**M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP**

**TIMING REQUIREMENTS** ( $V_{CC}=5V \pm 10\%$ ,  $V_{SS}=0V$ ,  $T_a=25^\circ C$ , 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	
$t_{SU(RDY-\phi_1)}$	RDY input setup time	70		60		55		ns	
$t_{SU(HOLD-\phi_1)}$	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	
$t_h(\phi_1-RDY)$	RDY input hold time	0		0		0		ns	
$t_h(\phi_1-HOLD)$	HOLD input hold time	0		0		0		ns	

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

### 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 <sub>iIN</sub> input cycle time	250		125		80		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	125		62		40		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input cycle time	1000		500		320		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input cycle time	500		250		160		ns	
$t_{W(TAH)}$	TA <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iIN</sub> input high-level pulse width	250		125		80		ns	
$t_{W(TAL)}$	TA <sub>iIN</sub> 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 <sub>iOUT</sub> input cycle time	5000		2500		2000		ns	
$t_{W(UPH)}$	TA <sub>iOUT</sub> input high-level pulse width	2500		1250		1000		ns	
$t_{W(UPL)}$	TA <sub>iOUT</sub> input low-level pulse width	2500		1250		1000		ns	
$t_{SU(UP-TIN)}$	TA <sub>iOUT</sub> input setup time	1000		500		400		ns	
$t_{H(TIN-UP)}$	TA <sub>iOUT</sub> input hold time	1000		500		400		ns	

# M37703E2-XXXSP, M37703E2AXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP

### Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{C(TB)}$	TB0 <sub>IN</sub> input cycle time (one edge count)	250		125		80		ns	
$t_{W(TBH)}$	TB0 <sub>IN</sub> input high-level pulse width (one edge count)	125		62		40		ns	
$t_{W(TBL)}$	TB0 <sub>IN</sub> input low-level pulse width (one edge count)	125		62		40		ns	
$t_{C(TB)}$	TB0 <sub>IN</sub> input cycle time (both edges count)	500		250		160		ns	
$t_{W(TBH)}$	TB0 <sub>IN</sub> input high-level pulse width (both edges count)	250		125		80		ns	
$t_{W(TBL)}$	TB0 <sub>IN</sub> 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.		
$t_{C(TB)}$	TB0 <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TB0 <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TB0 <sub>IN</sub> 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.		
$t_{C(TB)}$	TB0 <sub>IN</sub> input cycle time	1000		500		320		ns	
$t_{W(TBH)}$	TB0 <sub>IN</sub> input high-level pulse width	500		250		160		ns	
$t_{W(TBL)}$	TB0 <sub>IN</sub> 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.		
$t_{C(AD)}$	AD <sub>TRG</sub> input cycle time (minimum allowable trigger)	2000		1000		1000		ns	
$t_{W(ADL)}$	AD <sub>TRG</sub> 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 <sub>0</sub> input cycle time	500		250		200		ns	
$t_{W(CKH)}$	CLK <sub>0</sub> input high-level pulse width	250		125		100		ns	
$t_{W(CKL)}$	CLK <sub>0</sub> input low-level pulse width	250		125		100		ns	
$t_{D(c-a)}$	TxD <sub>0</sub> output delay time		150		90		80	ns	
$t_{H(c-a)}$	TxD <sub>0</sub> hold time	30		30		30		ns	
$t_{SU(D-C)}$	RxD <sub>0</sub> input setup time	60		30		20		ns	
$t_{H(c-d)}$	RxD <sub>0</sub> input hold time	90		90		90		ns	

### External interrupt INT<sub>i</sub> input

Symbol	Parameter	Limits						Unit	
		8MHz		16MHz		25MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
$t_{W(INH)}$	INT <sub>i</sub> input high-level pulse width	250		250		250		ns	
$t_{W(INL)}$	INT <sub>i</sub> input low-level pulse width	250		250		250		ns	

# M37703E2-XXXSP, M37703E2AXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXSP, M37703M2BXXXSP

## SWITCHING CHARACTERISTICS ( $V_{CC}=5V \pm 10\%$ , $V_{SS}=0V$ , $T_a=25^\circ C$ , unless otherwise noted)

### Single-chip mode

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 1	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	

### Memory expansion mode and microprocessor mode (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_{d(P0A-E)}$	Port P0 address output delay time	Fig. 1	100		30		12		ns	
$t_{d(E-P1Q)}$	Port P1 data output delay time (BYTE="L")		110		70		45		ns	
$t_{PZX(E-P1Z)}$	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	
$t_{PZX(E-P2Z)}$	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	
$t_{d(ALE-E)}$	ALE output delay time		4		4		4		ns	
$t_{w(ALE)}$	ALE pulse width		90		35		22		ns	
$t_{d(BHE-E)}$	BHE output delay time		100		30		20		ns	
$t_{d(R/W-E)}$	R/W output delay time		100		30		20		ns	
$t_{d(E-\phi_1)}$	$\phi_1$ 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	
$t_h(E-BHE)$	BHE hold time		18		18		18		ns	
$t_h(E-R/W)$	R/W hold time		18		18		18		ns	
$t_{w(EL)}$	$\bar{E}$ pulse width		220		95		50		ns	

**M37703E2-XXXSP, M37703E2AXXXSP  
M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

**Memory expansion mode and microprocessor mode** (when wait bit = "0", and external memory area accessed)

Symbol	Parameter	Test conditions	Limits						Unit	
			8MHz		16MHz		25MHz			
			Min.	Max.	Min.	Max.	Min.	Max.		
$t_d(P0A-E)$	Port P0 address output delay time	Fig. 1	100		30		12		ns	
$t_d(E-P1Q)$	Port P1 data output delay time (BYTE="L")			110		70		45	ns	
$tp_{XZ}(E-P1Z)$	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	
$tp_{XZ}(E-P2Z)$	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	
$t_d(ALE-E)$	ALE output delay time		4		4		4		ns	
$t_w(ALE)$	ALE pulse width		90		35		22		ns	
$t_d(BHE-E)$	BHE output delay time		100		30		20		ns	
$t_d(R/W-E)$	R/W output delay time		100		30		20		ns	
$t_d(E-\phi_1)$	$\phi_1$ 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	
$tp_{ZX}(E-P1Z)$	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	
$tp_{ZX}(E-P2Z)$	Port P2 floating release delay time		50		25		18		ns	
$t_h(E-BHE)$	BHE hold time		18		18		18		ns	
$t_h(E-R/W)$	R/W hold time		18		18		18		ns	
$t_w(EL)$	E pulse width		470		220		130		ns	

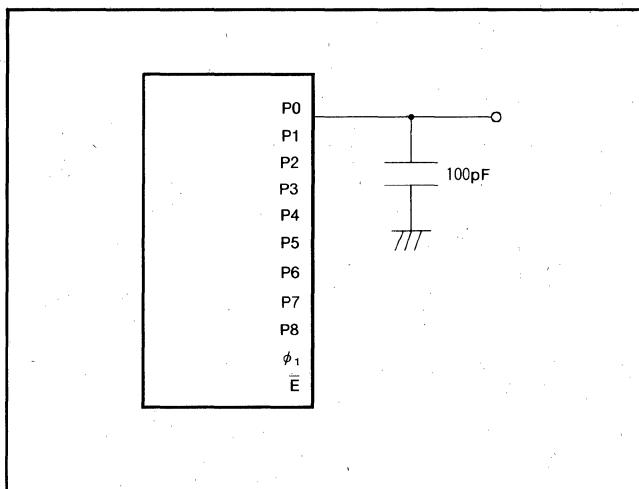


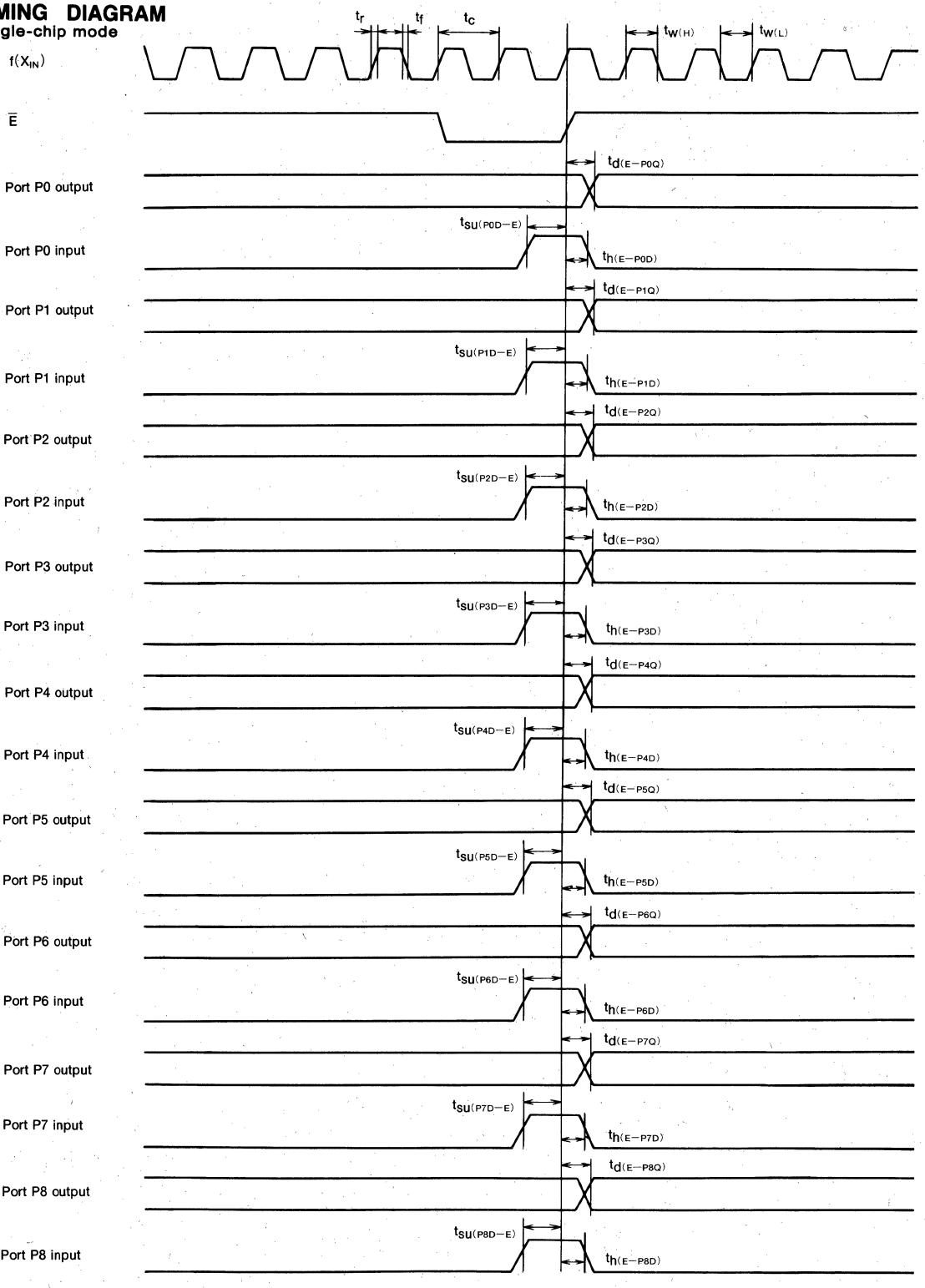
Fig. 1 Testing circuit for ports P0~P8,  $\phi_1$

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

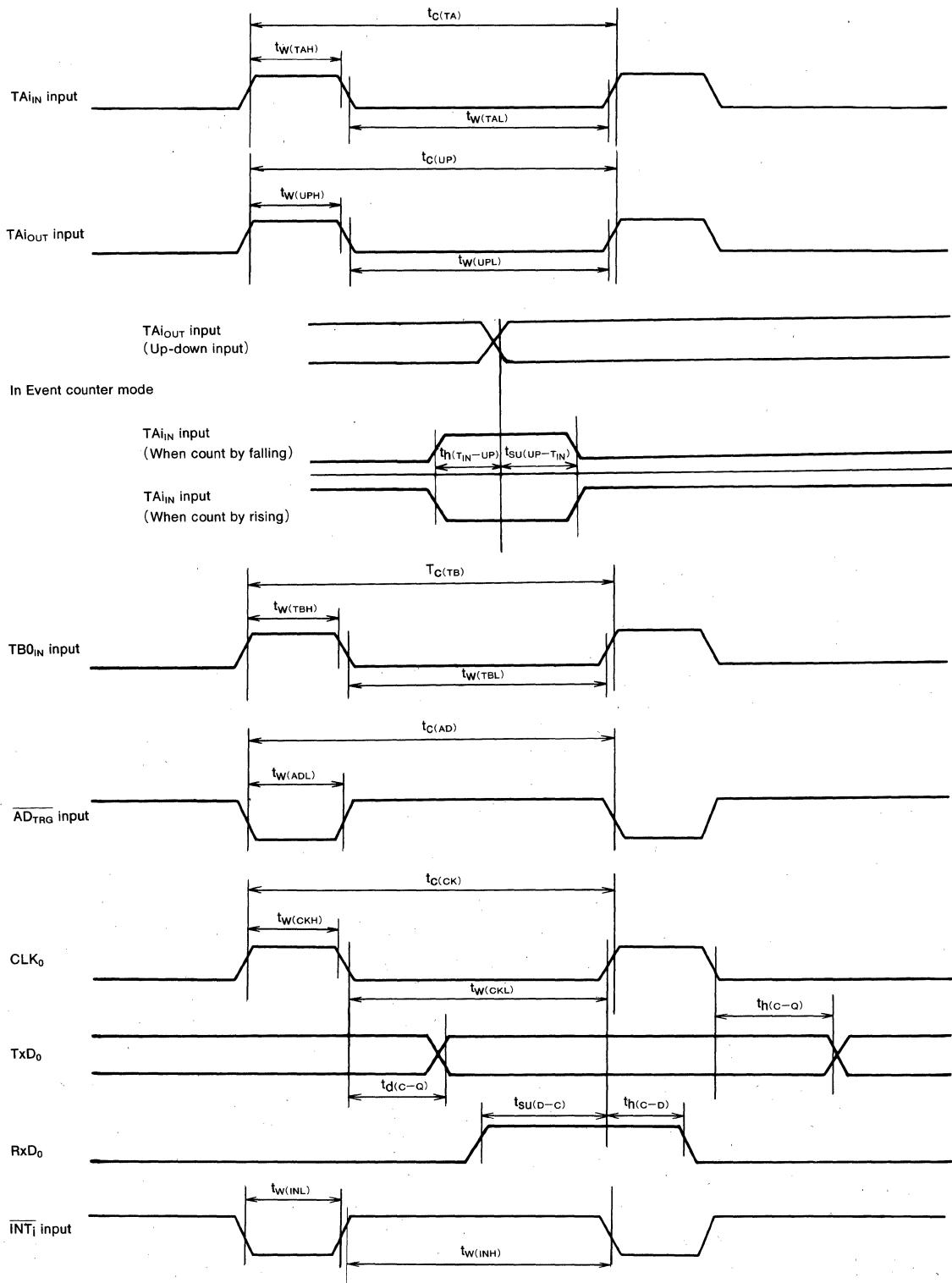
## TIMING DIAGRAM

Single-chip mode



# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

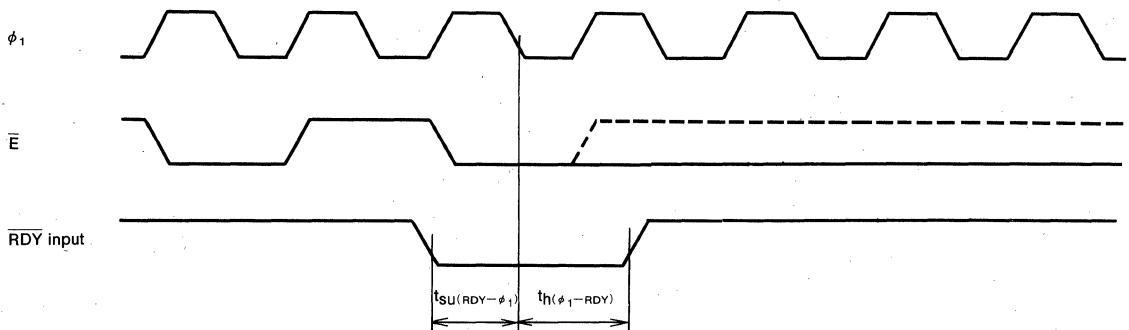


# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

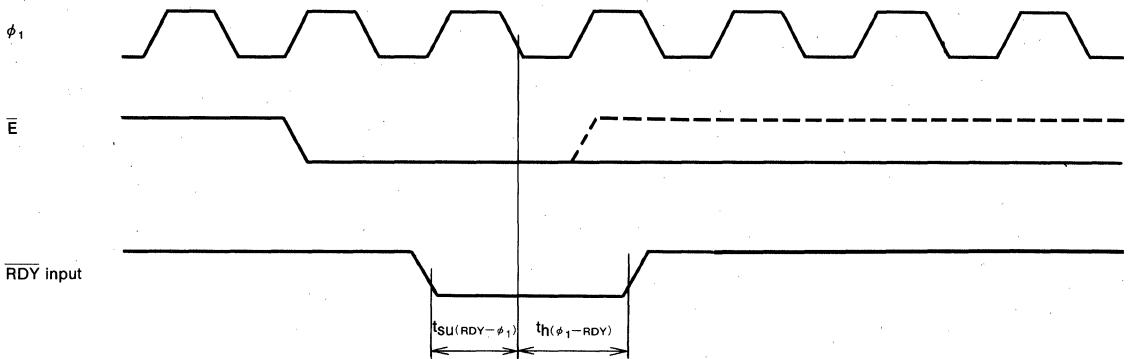
## PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

Memory expansion mode and microprocessor mode

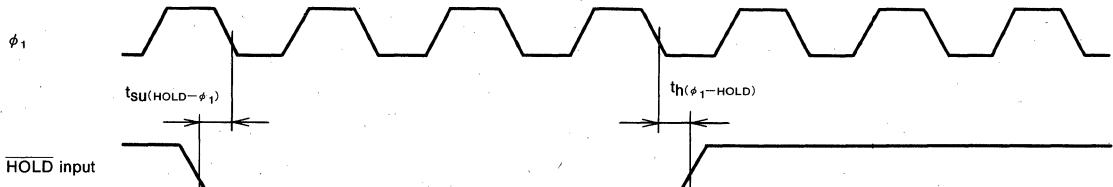
(When wait bit = "1")



(When wait bit = "0")



(When wait bit = "1" or "0" in common)



### Test conditions

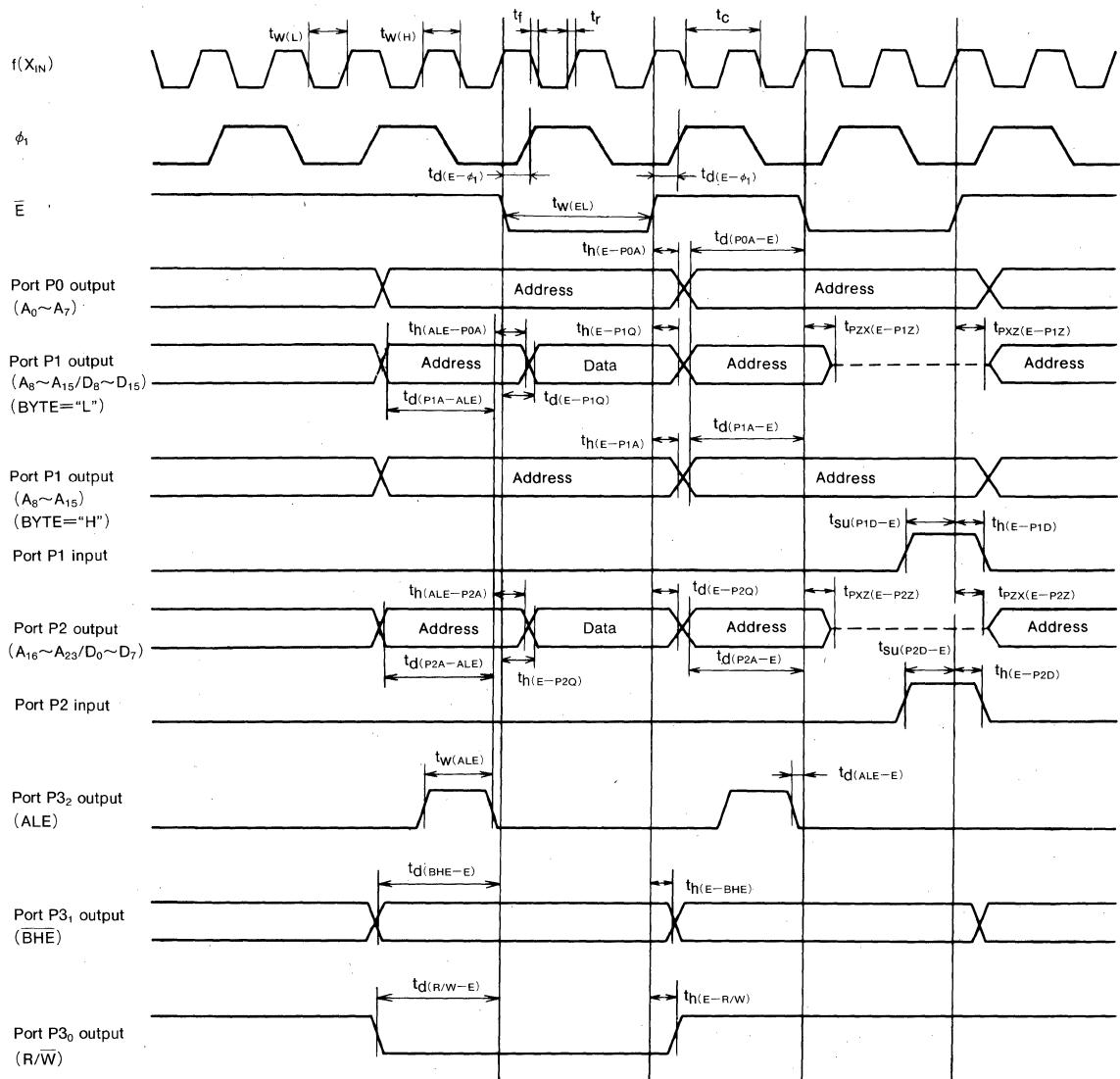
- $V_{CC} = 5 \text{ V} \pm 10\%$
- Input timing voltage :  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$

MITSUBISHI MICROCOMPUTERS

**M37703E2-XXXSP, M37703E2AXXXSP  
M37703E2BXXXSP**

**PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP**

Memory expansion mode and microprocessor mode (When wait bit = "1")



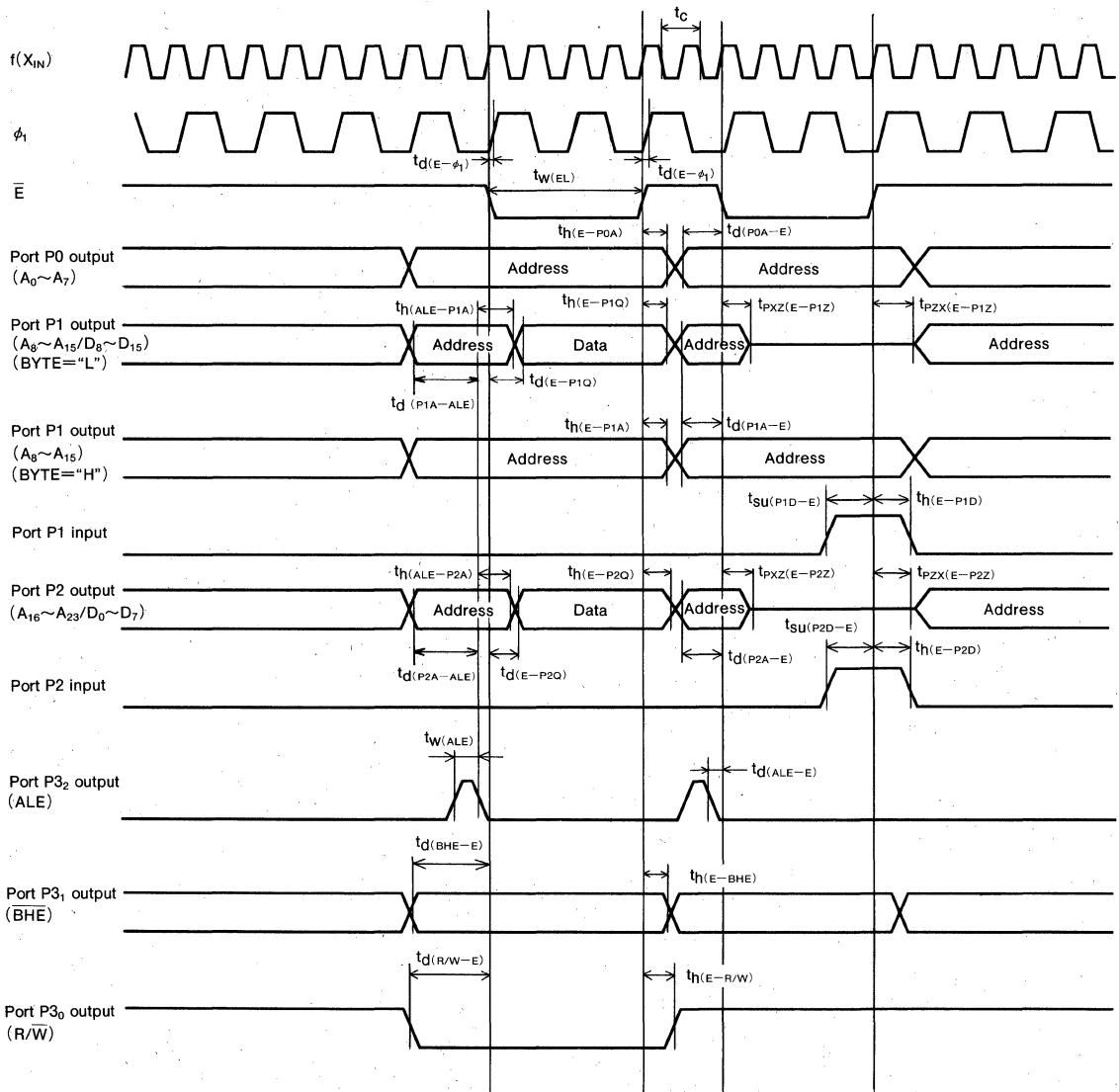
Test conditions

- $V_{CC} = 5 \text{ V} \pm 10\%$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$
- Port P1, P2 input :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.5 \text{ V}$
- Port P4<sub>i</sub> input :  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$

# M37703E2-XXXSP, M37703E2AXXXSP M37703E2BXXXSP

## PROM VERSION of M37703M2-XXXSP, M37703M2AXXXSP, M37703M2BXXXSP

Memory expansion mode and microprocessor mode (When wait bit = "0", and external memory area is accessed)



### Test conditions

- $V_{CC} = 5 \text{ V} \pm 10\%$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$
- Ports P1, P2 input :  $V_{IL} = 0.8 \text{ V}$ ,  $V_{IH} = 2.5 \text{ V}$
- Port P4<sub>1</sub> input :  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$

# M37703E4-XXXSP, M37703E4AXXXSP M37703E4BXXXSP

PROM VERSION of M37703M4-XXXSP, M37703M4AXXXSP, M37703M4BXXXSP

## DESCRIPTION

The M37703E4-XXXSP, M37703E4AXXXSP and M37703E4BXXXSP are single-chip microcomputers designed with high-performance CMOS silicon gate technology. These are housed in a 64-pin shrink plastic molded DIP. The features of these chips are similar to those of the M37703M4-XXXSP, M37703M4AXXXSP and M37703M4BXXXSP except that these chips have a 32K-byte PROM built in.

These single-chip microcomputers have 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. These microcomputers are suitable for office, business and industrial equipment controller that require high-speed processing of large data. Since general purpose PROM writers can be used for the built-in PROM, these chips are suitable for small quantity production runs.

The differences between M37703E4-XXXSP, M37703E4AXXXSP and M37703E4BXXXSP are the external clock input frequency as shown below. Therefore, the following descriptions will be for the M37703E4-XXXSP unless otherwise noted.

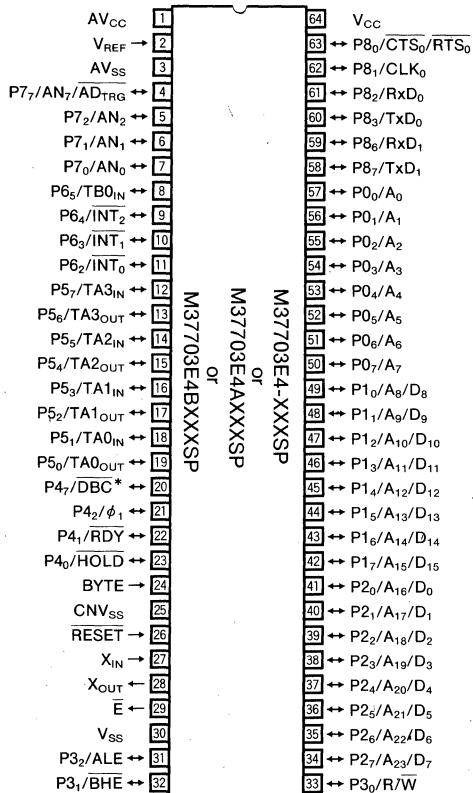
Type name	External clock input frequency
M37703E4-XXXSP	8 MHz
M37703E4AXXXSP	16MHz
M37703E4BXXXSP	25MHz

The M37703E4-XXXSP has the same functions as the M37703E2-XXXSP except for the memory size.

## FEATURES

- Number of basic instructions ..... 103
- Memory size PROM (one time) ..... 32K bytes
- RAM ..... 2048 bytes
- Instruction execution time
  - M37703E4-XXXSP  
(The fastest instruction at 8 MHz frequency) ..... 500ns
  - M37703E4AXXXSP  
(The fastest instruction at 16 MHz frequency) ..... 250ns
  - M37703E4BXXXSP  
(The fastest instruction at 25 MHz frequency) ..... 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 8 MHz frequency)  
..... 30mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
- UART (may also be synchronous) ..... 2
- 8-bit A-D converter ..... 4-channel inputs
- 12-bit watchdog timer
- Programmable input/output  
(ports P0, P1, P2, P3, P4, P5, P6, P7, P8) ..... 53

## PIN CONFIGURATION (TOP VIEW)



## Outline 64P4B

\* : Used in the evaluation chip mode only

## APPLICATION

Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as ME, NC, communication, and measuring instruments

## THE FUNCTIONS AND CHARACTERISTICS

The M37703E4-XXXSP has the same functions and characteristics as the M37703E2-XXXSP except for the ROM and RAM size. Refer to the section on the M37703E2-XXXSP.

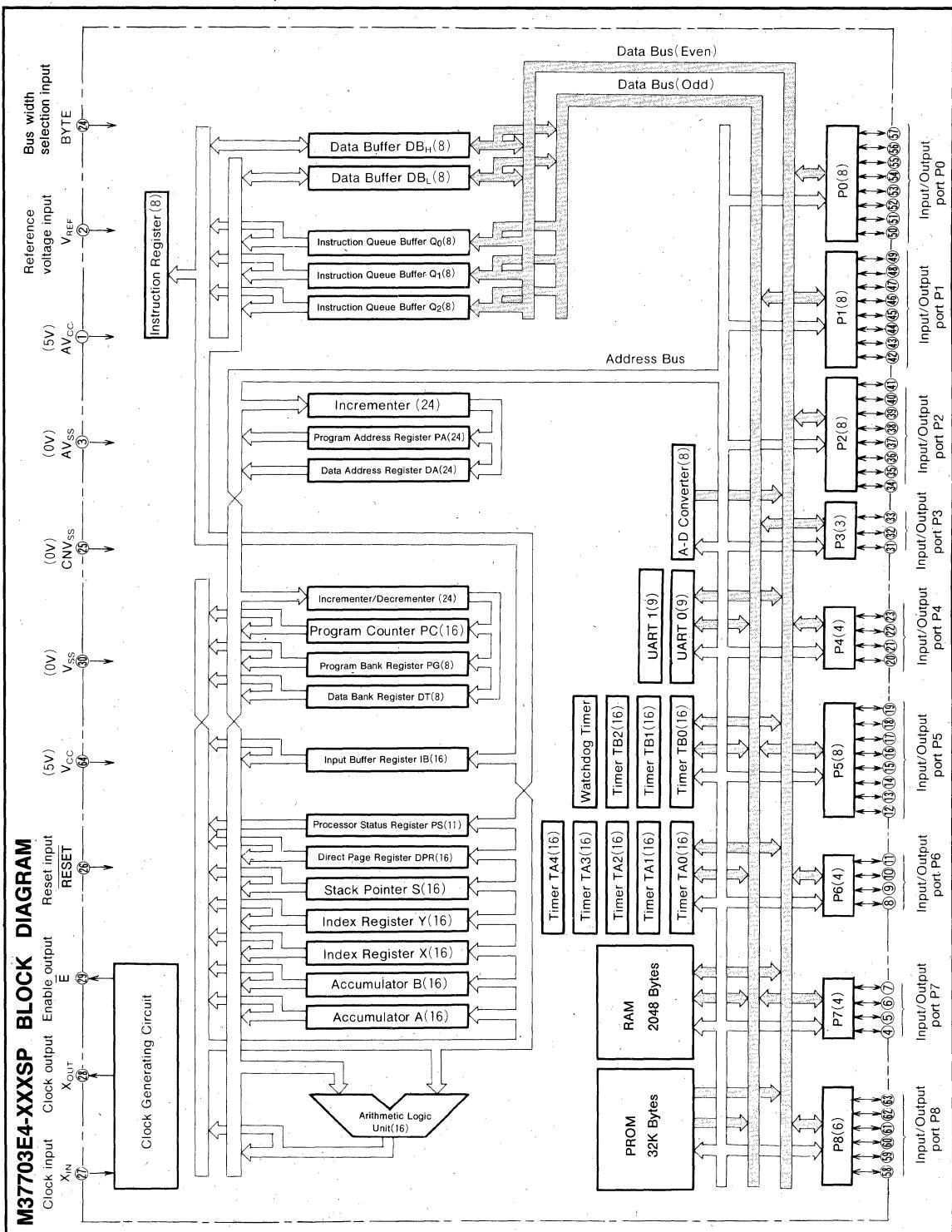
## DATA REQUIRED FOR PROM ORDERING

Please send the following data for writing to PROM.

- (1) M37703E4-XXXSP writing to PROM order confirmation form
- (2) 64P4B mark specification form for one time PROM
- (3) ROM data (EPROM 3sets)

**MITSUBISHI MICROCOMPUTERS**  
**M37703E4-XXXSP, M37703E4AXXXSP**  
**M37703E4BXXXSP**

**PROM VERSION of M37703M4-XXXSP, M37703M4AXXXSP, M37703M4BXXXSP**



---

## **APPENDICES**

---

**4**



# MELPS 7700 MASK ROM ORDERING METHOD

## MASK ROM ORDERING METHOD

Mitsubishi Electric corp. accepts order to transfer EPROM supplied program data into the mask ROMs in single-chip 16-bit microcomputers. When placing such order, please submit the information described below.

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 ..... 1 set

## NOTES

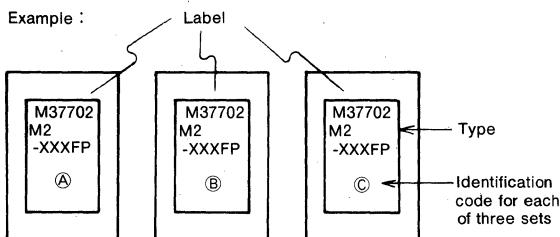
### (1) Acceptable EPROM type

Any EPROM made by Mitsubishi Electric corp. that is listed in the Mask ROM Order Confirmation Form may be used.

### (2) EPROM window labeling

Please write the model name and the identification code (A, B, C) on the label for each of the three sets of data EPROMs provided.

Example :



### (3) Calculation and indication of check sum code

Please calculate the total number of data in words in the EPROM, and write the number in 4-digit hexadecimal form in the check sum code field of the Mask ROM Order Confirmation Form.

### (4) Options

Refer to the appropriate data book entry and write the desired options on the Mask ROM Order Confirmation Form.

### (5) Marking specification method

The permissible marking specifications differ depending on the shape of package. Please fill out the Marking Specification Form and attach it to the Mask ROM Order Confirmation Form.

## OUTLINE OF ORDER PROCESSING

Mitsubishi Electric corp. will produce the mask ROM if at least two of the three EPROM sets submitted contain identical data.

If we find error when the submitted EPROMs are compared, we will contact your representative. Thus, we assume responsibility only when we produce the mask ROMs that contain data other than the data correctly provided by the customer.

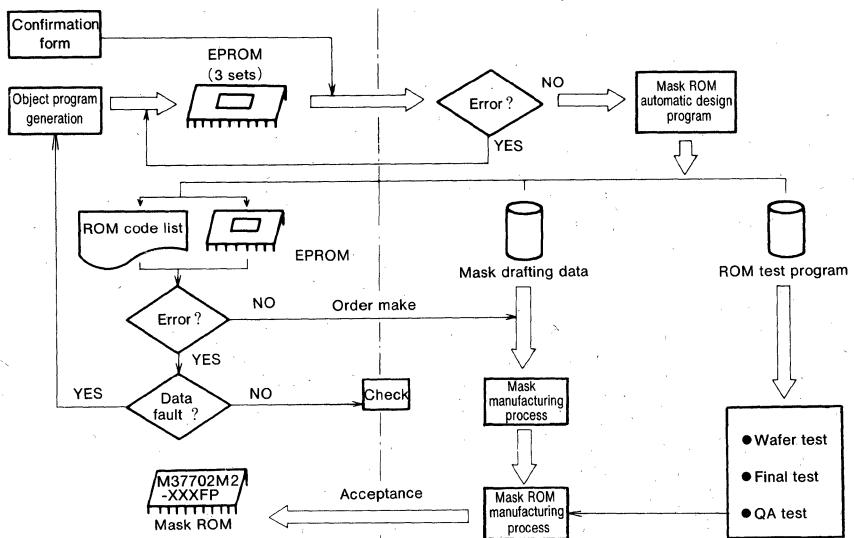
Mitsubishi Electric corp. uses an automatic mask ROM design program to generate the following:

- 1 : Drafting data for mask ROM production;
- 2 : ROM code listing or EPROM for mask ROM production error check work;
- 3 : Mask ROM test program.

The chart below shows the flow of mask ROM production.

## MELPS 7700 MASK ROM DEVELOPMENT CAD SYSTEM

From customer Mitsubishi Electric corp.



MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-45A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM  
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2-XXXFP  
MITSUBISHI ELECTRIC**

Mask 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).

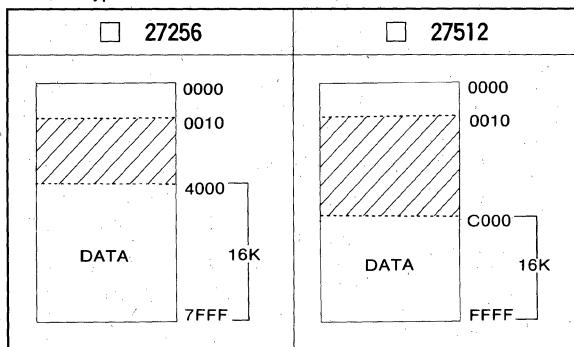
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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	Address	Option data
4D	0	2D	8	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<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable       Address 10<sub>16</sub>  
 STP instruction disable       Address 10<sub>16</sub>

※ 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

# MELPS 7700 MASK ROM ORDERING METHOD

GZZ-SH02-46A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM  
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2AXXXFP  
MITSUBISHI ELECTRIC**

Mask ROM number	
-----------------	--

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

※ Customer	Company name	( )	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).

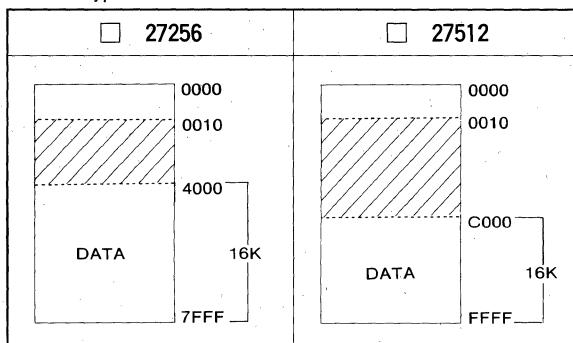
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

## EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
32	5	FF
4D	6	FF
32	7	FF

8      Option data      10

## ※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable      

01 <sub>16</sub>
------------------

 Address 10<sub>16</sub>
- STP instruction disable      

00 <sub>16</sub>
------------------

 Address 10<sub>16</sub>

## ※ 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

# MELPS 7700 MASK ROM ORDERING METHOD

GZZ-SH02-47A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M2BXXXFP**  
**MITSUBISHI ELECTRIC**

Mask ROM number	
-----------------	--

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

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

## ※ 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).

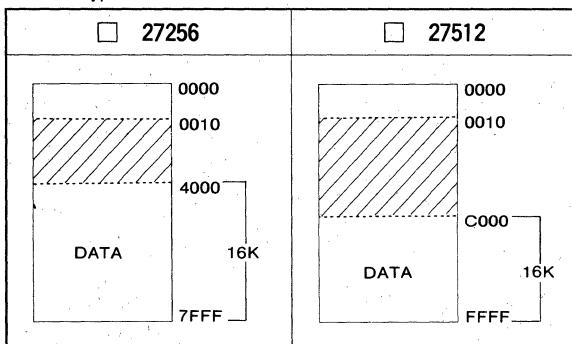
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
32	5	FF
4D	6	FF
32	7	FF

Option data 10

## ※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable      

01 <sub>16</sub>
------------------

 Address 10<sub>16</sub>  
 STP instruction disable      

00 <sub>16</sub>
------------------

 Address 10<sub>16</sub>

## ※ 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

# MELPS 7700 MASK ROM ORDERING METHOD

GZZ-SH02-36A&lt;98A0&gt;

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM  
SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4-XXXFP  
MITSUBISHI ELECTRIC**

Mask ROM number	
Receipt	Date :
	Section head 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).

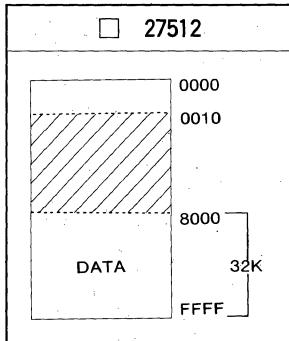
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
32	5	FF
4D	6	FF
34	7	FF

Option data

10

## ※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable       Address 10<sub>16</sub>  
 STP instruction disable       Address 10<sub>16</sub>

## ※ 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

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-37A(98A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4AXXXFP**  
**MITSUBISHI ELECTRIC**

Mask 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).

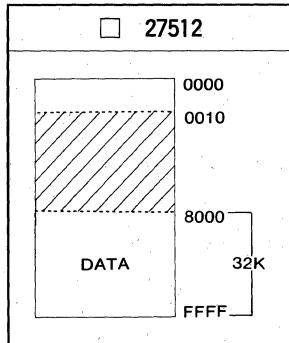
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
32	5	FF
4D	6	FF
34	7	FF

Option data

10

※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

※ 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

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-38A(98A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37702M4BXXXFP**  
**MITSUBISHI ELECTRIC**

Mask ROM number	
Receipt	Date :
	Section head 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).

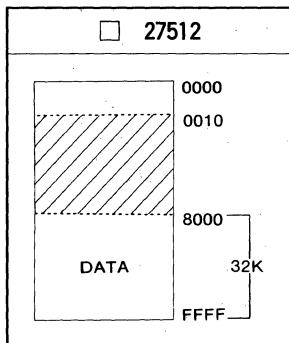
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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	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
4D	6	FF	E
34	7	FF	F

Option data

\* 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

\* 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

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-51A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2-XXXSP**  
**MITSUBISHI ELECTRIC**

Mask 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

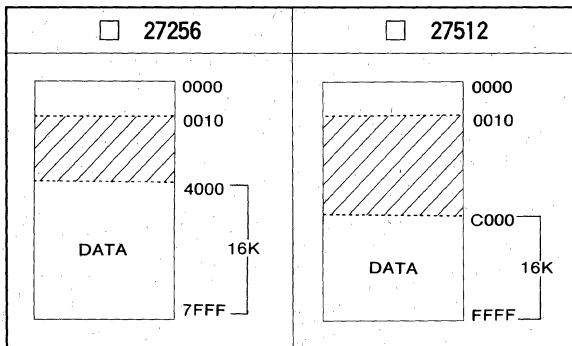
Specify the name of the product being ordered and the type of EPROMs submitted.

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 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<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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	Address
4D	0	2D	8
33	1	FF	Option data
37	2	FF	10
37	3	FF	
30	4	FF	
33	5	FF	
4D	6	FF	
32	7	FF	

※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable       Address 10<sub>16</sub>
- STP instruction disable       Address 10<sub>16</sub>

※ 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2-XXXSP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-52A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2AXXSP**  
**MITSUBISHI ELECTRIC**

Mask 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).

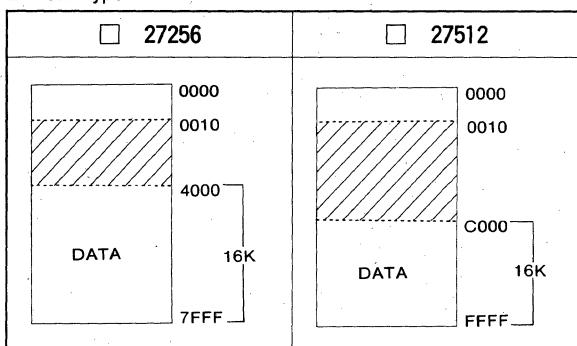
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
33	5	FF
4D	6	FF
32	7	FF

Option data 10

※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

※ 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2AXXSP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-53A(99A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M2BXXXSP**  
**MITSUBISHI ELECTRIC**

Mask ROM number	
-----------------	--

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

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

※ 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).

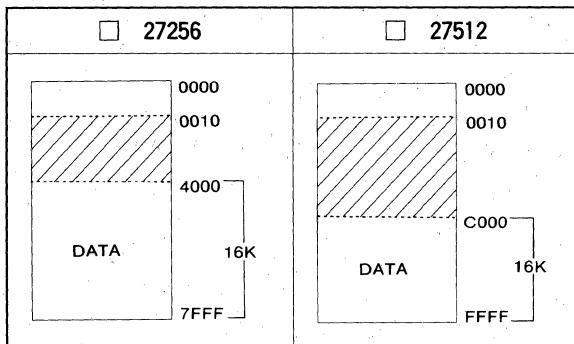
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
33	5	FF
4D	6	FF
32	7	FF

Option data 10

※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

※ 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M2BXXXSP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 MASK ROM**  
**ORDERING METHOD**

GZZ-SH02-42A < 98A0 >

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4-XXXSP**  
**MITSUBISHI ELECTRIC**

Mask 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).

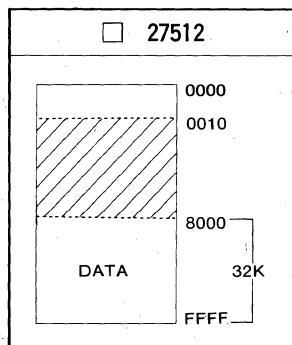
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas



(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
33	5	FF
4D	6	FF
34	7	FF

Option data

\* 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

\* 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4-XXXSP) and attach to the Mask ROM Order Confirmation Form.

\* 4. Comments

MITSUBISHI MICROCOMPUTERS  
**MELPS 7700 MASK ROM  
ORDERING METHOD**

GZZ-SH02-43A(98A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM  
SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4AXXXSP  
MITSUBISHI ELECTRIC**

Mask 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).

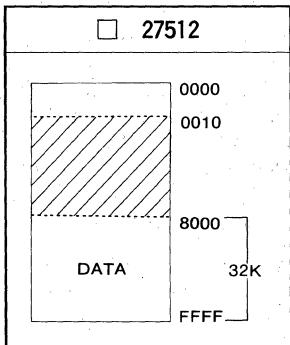
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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	Address
4D	0	41	Option data 10
33	1	FF	9
37	2	FF	A
37	3	FF	B
30	4	FF	C
33	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<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- |  |                  |                          |
|--|------------------|--------------------------|
| <input type="checkbox"/> STP instruction enable  | 01 <sub>16</sub> | Address 10 <sub>16</sub> |
| <input type="checkbox"/> STP instruction disable | 00 <sub>16</sub> | Address 10 <sub>16</sub> |

※ 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4AXXXSP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

MITSUBISHI MICROCOMPUTERS  
MELPS 7700 MASK ROM  
ORDERING METHOD

GZZ-SH02-44A(98A0)

**MELPS 7700 MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER M37703M4BXXXSP**  
**MITSUBISHI ELECTRIC**

Mask 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).

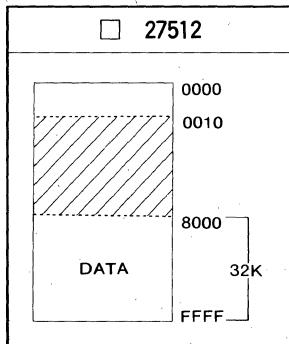
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas      



      (hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 10<sub>16</sub> 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
33	1	FF
37	2	FF
37	3	FF
30	4	FF
33	5	FF
4D	6	FF
34	7	FF
		Option data 10

※ 2. STP instruction option

One of the following sets of data should be written to the option data address (10<sub>16</sub>) of the EPROM you have ordered. Check @ in the appropriate box.

- STP instruction enable      



      Address 10<sub>16</sub>
- STP instruction disable      



      Address 10<sub>16</sub>

※ 3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form (for M37703M4BXXXSP) and attach to the Mask ROM Order Confirmation Form.

※ 4. Comments

# MELPS 7700 PROM ORDERING METHOD

## PROM ORDERING METHOD

Mitsubishi Electric corp. accepts order to transfer EPROM supplied program data into the one time PROMs in single-chip 16-bit microcomputers. When placing such order, please submit the information described below.

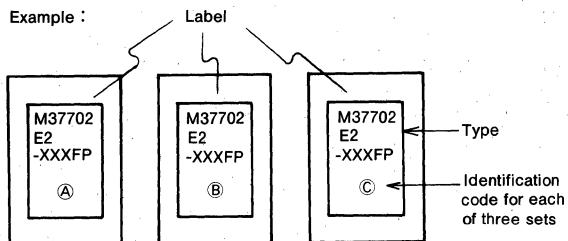
1. Writing to PROM Order Confirmation Form ..... 1 set  
(There is a specific form to be used for each model.)
2. Data to be written into PROM built in ..... EPROM  
(Please provide three sets containing the identical data.)
3. Mark Specification Form ..... 1 set

## NOTES

### (1) Acceptable EPROM type

Any EPROM made by Mitsubishi Electric corp. that is listed in the Writing to PROM Order Confirmation Form may be used.

Example :



### (2) EPROM window labeling

Please write the model name and the identification code (A, B, C) on the label for each of the three sets of data EPROMs provided.

### (3) Calculation and indication of check sum code

Please calculate the total number of data in words in the EPROM, and write the number in 4-digit hexadecimal form in the check sum code field of the Writing to PROM Order Confirmation Form.

### (4) Marking specification method

The permissible marking specifications differ depending on the shape of package. Please fill out the Marking Specification Form and attach it to the Writing to PROM Order Confirmation Form.

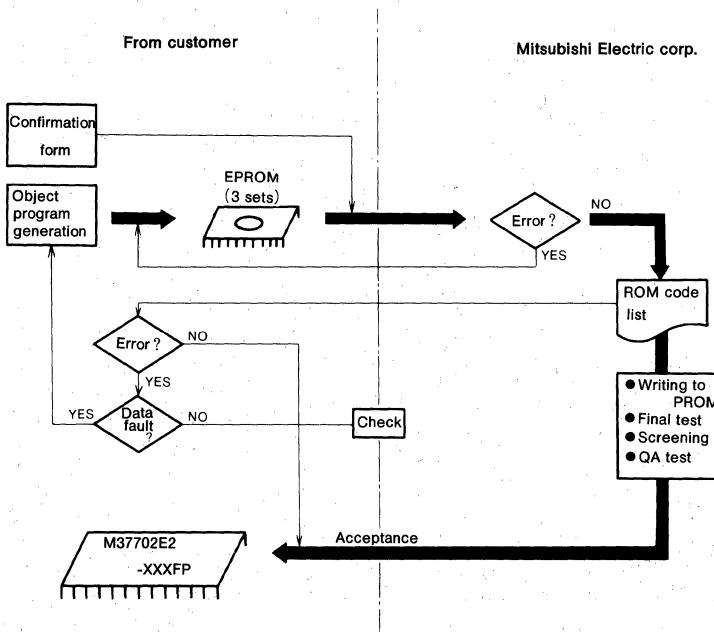
## OUTLINE OF ORDER PROCESSING

Mitsubishi Electric corp. will produce Writing to PROM if at least two of the three EPROM sets submitted contain identical data.

If we find error when the submitted EPROMs are compared, we will contact your representative. Thus, we assume responsibility only when we produce Writing to PROMs that contain data other than the data correctly provided by the customer.

The chart below shows the flow of one time PROM production.

## MELPS 7700 ONE TIME PROM DEVELOPMENT CAD SYSTEM



**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH02-54A(99A0)

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER**  
**M37702E2-XXXFP**  
**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).

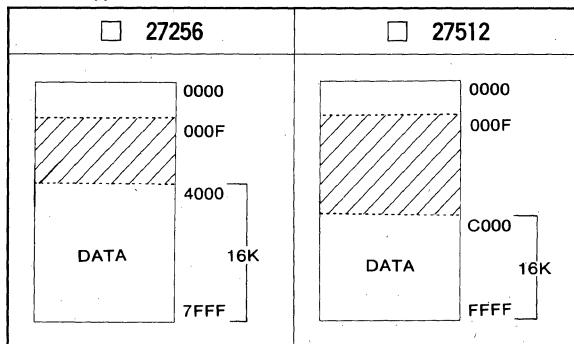
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
32	5
45	6
32	7

## ※ 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

## MELPS 7700 PROM ORDERING METHOD

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 ( )	Responsible officer	Supervisor
	Date issued			

## ※ 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).

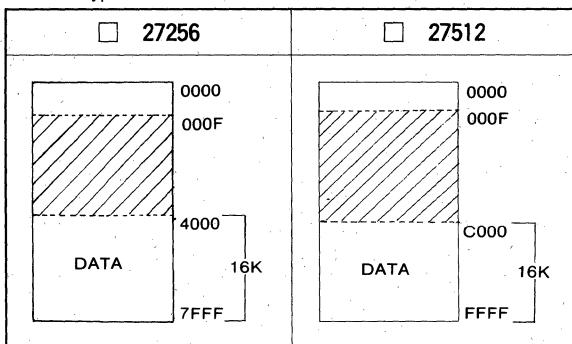
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas 

--	--	--	--

 (hexadecimal notation)

## EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
32	5
45	6
32	7

## ※ 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

**MELPS 7700 PROM ORDERING METHOD**

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 (      )	Responsible officer	Supervisor
	Date issued			

## ※ 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).

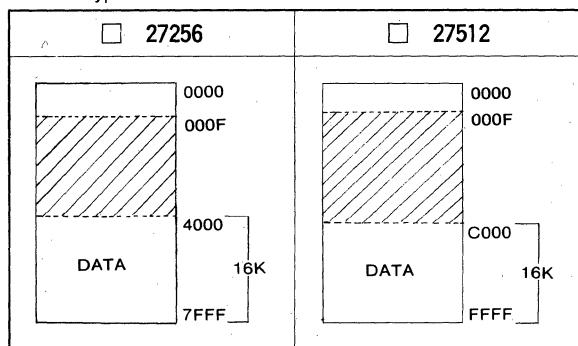
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

## EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
32	5
45	6
32	7

## ※ 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

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-67A&lt;07A0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM****SINGLE-CHIP 16-BIT MICROCOMPUTER****M37702E4-XXXFP****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).

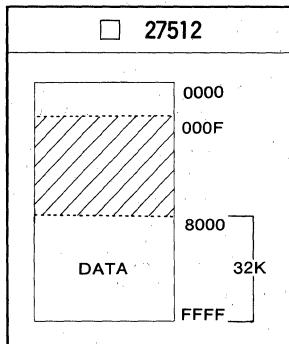
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas 

--	--	--	--

 (hexadecimal notation)

## EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
32	5
45	6
34	7

## ※ 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

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-70A&lt;07A0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER**  
**M37702E4AXXFP**  
**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).

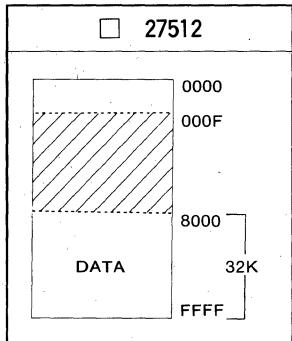
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	41
37	8
37	FF
30	9
32	A
45	B
34	C
	D
	E
	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 M37702E4AXXFP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-38A(01A0)

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER**  
**M37702E4BXXXFP**  
**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).

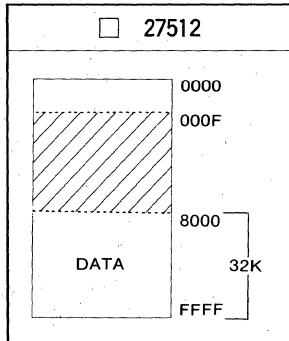
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
- (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
32	5
45	6
34	7
42	8
FF	9
FF	A
FF	B
FF	C
FF	D
FF	E
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

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH02-58A(99A0)

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER**  
**M37703E2-XXXSP**  
**MITSUBISHI ELECTRIC**

ROM number	
Receipt	Date :
	Section head signature

Note : Please fill in all items marked※

※ Customer	Company name	TEL (      )	Responsible officer Signature	Supervisor Signature
	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).

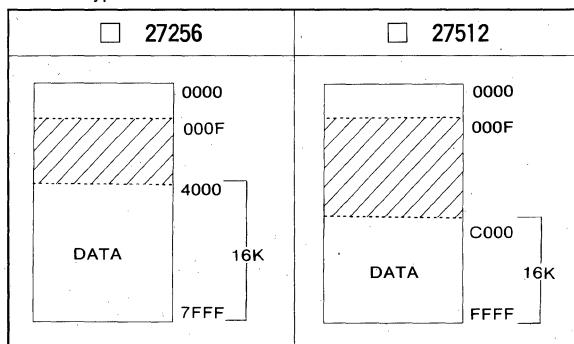
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas 

--	--	--	--

 (hexadecimal notation)

## EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
33	5
45	6
32	7

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2-XXXSP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

**MITSUBISHI MICROCOMPUTERS**  
**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH02-59A<99A0>

## **MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**

# **SINGLE-CHIP 16-BIT MICROCOMPUTER**

**M37703E2AXXSP**

MITSUBISHI ELECTRIC

ROM number			
Receipt	<p>Date :</p> <table border="1"> <tr> <td>Section head signature</td> <td>Supervisor signature</td> </tr> </table>	Section head signature	Supervisor signature
Section head signature	Supervisor signature		

Note : Please fill in all items marked※

Note : Please fill in all items marked *					
Customer	Company name	TEL		Responsible officer Issuance signatures	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 :**

<input type="checkbox"/> 27256	<input type="checkbox"/> 27512
<p>DATA</p> <p>16K</p>	<p>DATA</p> <p>16K</p>
0000 000F 4000 7FFF	0000 000F C000 FFFF

- (1) Set "FF<sub>16</sub>" in the shaded area.

(2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
33	5
45	6
32	7

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2AXXXSP) and attach to the Writing to PROM Order Confirmation Form.

### ※ 3. Comments

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH04-08A&lt;0ZA0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM****SINGLE-CHIP 16-BIT MICROCOMPUTER****M37703E2BXXXSP****MITSUBISHI ELECTRIC**

ROM number	
------------	--

Receipt	Date :
	Section head signature

Note : Please fill in all items marked\*

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

## ※ 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).

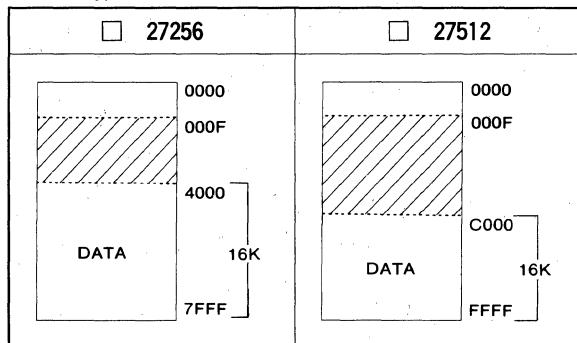
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	42
37	9
37	FF
30	A
33	FF
45	B
32	C
	D
	E
	F

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E2BXXXSP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-69A&lt;07A0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM****SINGLE-CHIP 16-BIT MICROCOMPUTER****M37703E4-XXXSP****MITSUBISHI ELECTRIC**

ROM number	
------------	--

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※

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

## ※ 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).

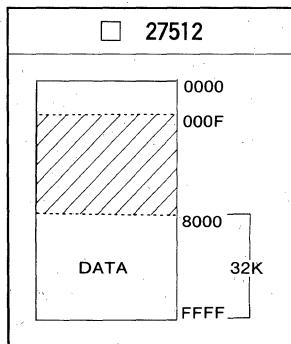
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas 

--	--	--	--

 (hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
33	5
45	6
34	7

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4-XXXSP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-68A&lt;07A0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP 16-BIT MICROCOMPUTER**  
**M37703E4AXXSP**  
**MITSUBISHI ELECTRIC**

Receipt	ROM number	
	Date :	Section head signature

Note : Please fill in all items marked※

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

## ※ 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).

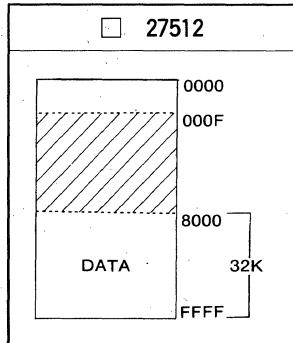
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas 

--	--	--	--

 (hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
33	5
45	6
34	7

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4AXXSP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

**MELPS 7700 PROM ORDERING METHOD**

GZZ-SH03-41A&lt;01A0&gt;

**MELPS 7700 WRITING TO PROM ORDER CONFIRMATION FORM****SINGLE-CHIP 16-BIT MICROCOMPUTER****M37703E4BXXXSP****MITSUBISHI ELECTRIC**

ROM number	
------------	--

Receipt	Date :
	Section head signature

Note : Please fill in all items marked※

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

## ※ 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).

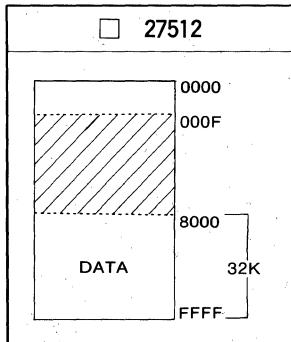
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce writing to PROM based on this data. We shall assume the responsibility for errors only if the written PROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas

--	--	--	--

(hexadecimal notation)

EPROM Type :



- (1) Set "FF<sub>16</sub>" in the shaded area.
  - (2) Address 0<sub>16</sub> to 0F<sub>16</sub> 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
33	1
37	2
37	3
30	4
33	5
45	6
34	7

## ※ 2. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 64P4B Mark Specification Form for one time PROM (for M37703E4BXXXSP) and attach to the Writing to PROM Order Confirmation Form.

## ※ 3. Comments

# MARK SPECIFICATION FORM

## MARK SPECIFICATION FORM

Mark specification format differs depending on the package type.

Fill out the Mark Specification Form for the package type being ordered, and submit the form with the Mask ROM Confirmation Form.

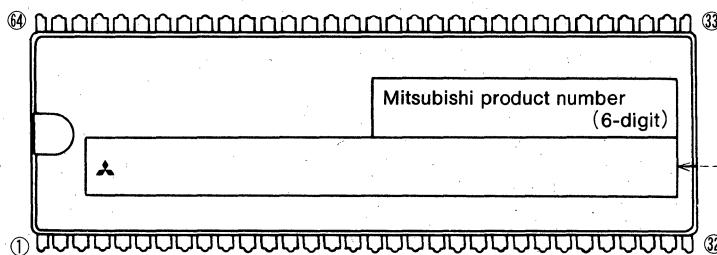
**MITSUBISHI MICROCOMPUTERS**  
**MARK SPECIFICATION FORM**

**64P4B (64-PIN SHRINK DIP) MARK SPECIFICATION FORM**

Mitsubishi IC catalog name

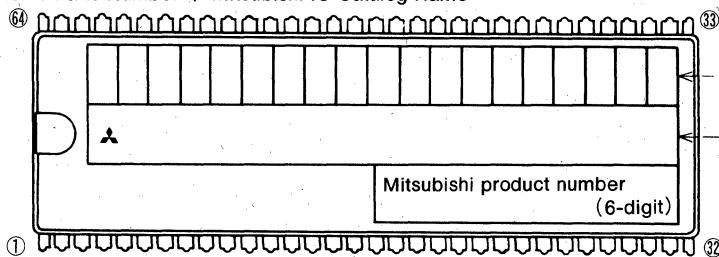
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi IC Catalog Name



Customer's Parts Number  
 Note: The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

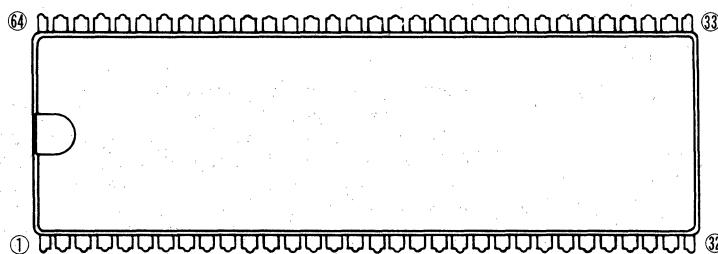
2 : The fonts and size of characters are standard Mitsubishi type.

3 : Customer's parts number can be up to 19 alphanumeric characters for capital letters, hyphens, commas, periods and so on.

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

Mitsubishi logo is not required

C. Special Mark Required



Mitsubishi logo is not required

Note1 : If special mark is to be printed, indicate the desired layout of the mark in the upper 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 on the right.

For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special character fonts required

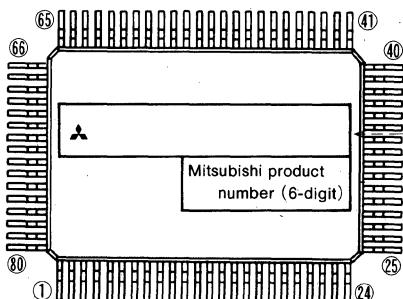
**MITSUBISHI MICROCOMPUTERS**  
**MARK SPECIFICATION FORM**

**80P6N (80-PIN QFP) MARK SPECIFICATION FORM**

Mitsubishi IC catalog name

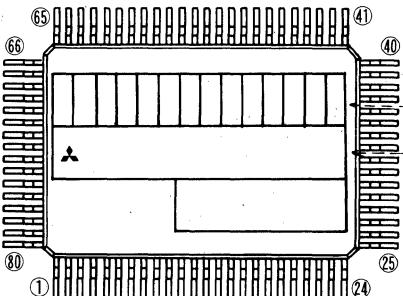
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

B. Customer's Parts Number + Mitsubishi IC Catalog Name



Customer's Parts Number

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

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

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

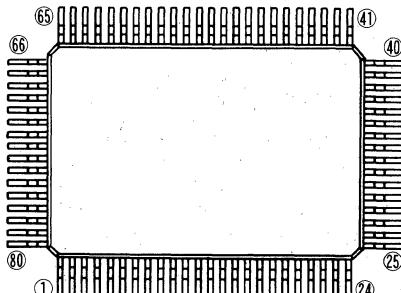
3 : Customer's parts number can be up to 14 alphanumeric characters for capital letters, hyphens, commas, periods and so on.

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



**MITSUBISHI MICROCOMPUTERS**  
**MARK SPECIFICATION FORM**

**64P4B(64-PIN SHRINK DIP) MARK SPECIFICATION FORM**  
for one time PROM version microcomputers

Enter the catalog number of the microcomputer for which this mark specification is intended.(If you do not know the ROM code number, enter XXX in its place.)

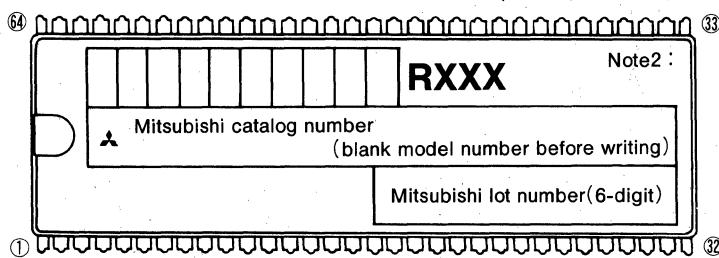
the catalog number of the microcomputer

**M**

**A. Standard Mitsubishi Mark**

Customer specified part number will be printed together with the ROM code number on the top line.

Enter the desired part number left aligned in the box below.(up to 10 characters)



Note1 : The following characters can be used in the part number :

Uppercase alphabet, numbers, ampersand, hyphen, period, comma, +, /, (, ), ©  
(©will be printed at 1.5X character width)

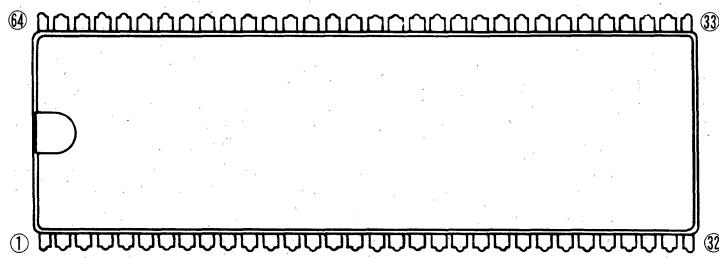
2 : XXX is the ROM code number.

**B. Special Mark Required**

If you desire anything other than the standard Mitsubishi mark, it will be treated as a special mark.

Special marks will take longer to produce and should be avoided if possible.

If a special mark is to be printed, indicate the desired layout of the mark in the figure below. The layout will be duplicated as closely as possible.



Note1 : If the customer's trademark logo must be used in the special mark, please submit a clean original logo.

Note that special marks require extra cost and time to produce.

# 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 KMELOCO J  
Telephone: (0727) 82-5131  
Facsimile: (0727) 72-2329

## HONG KONG

MITSUBISHI ELECTRIC (H.K.) LTD.  
25 Floor, Leighton Centre,  
77, Leighton Road, Causeway Bay.  
Hong Kong  
Telex: 60800 MELCO HX  
Telephone: (5) 773901-3  
Facsimile: (5) 895-3104

## 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 Argues 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

Mitsubishi Electronics America, Inc.  
6185 Ordan Drive, Unit # 110  
Mississauga, Ontario, Canada L5T 2E1  
Telephone: (416) 670-8711  
Facsimile: (416) 670-8715

## CANADA

Mitsubishi Electronics America, Inc.  
300 March Road, Suite 302  
Kanata, Ontario, Canada K2K 2E2  
Telephone: (416) 670-8711  
Facsimile: (416) 670-8715

## GERMANY

Mitsubishi Electric Europe GmbH  
Headquarters:  
Gotheer Str. 8  
4030 Ratingen 1, Germany  
Telex: 8585070 MED D  
Telephone: (02102) 4860  
Facsimile: (02102) 486-115

Munich Office:  
Arabellastraße 31  
8000 München 81, 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  
Lastbilsvägen 6B  
5-19149 Söllentuna, 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.  
73-75, Epping Road, North Ryde,  
P.O. Box 1567, Macquarie Centre,  
N.S.W., 2113, Australia  
Telex: MESYD AA 26614  
Telephone: (02) (888) 5777  
Facsimile: (02) (887) 3635

**MITSUBISHI DATA BOOK  
SINGLE-CHIP 16-BIT MICROCOMPUTERS**

---

Jun. First Edition 1991

Edited by

Committee of editing of Mitsubishi Semiconductor Data Book

Published by

Mitsubishi Electric Corp., Semiconductor Division

---

This book, or parts thereof, may not be reproduced in any form without permission of  
Mitsubishi Electric Corporation.

© 1991 MITSUBISHI ELECTRIC CORPORATION Printed in Japan

# MITSUBISHI SEMICONDUCTORS SINGLE-CHIP 16-BIT MICROCOMPUTERS

Enlarged  
edition



## MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO

These products or technologies  
are subject to Japanese and/or  
COCOM strategic restrictions, and  
diversion contrary thereto is  
prohibited.