# M50740-XXXSP

## SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

## DESCRIPTION

The M50740-XXXSP is a single-chip 8-bit microcomputer fabricated using CMOS technology and housed in a 52-pin shrink plastic molded DIL package. It is designed to suit for controlling home electrical appliances and consumer equipment with a simple instruction where the ROM and RAM use the same memory area.

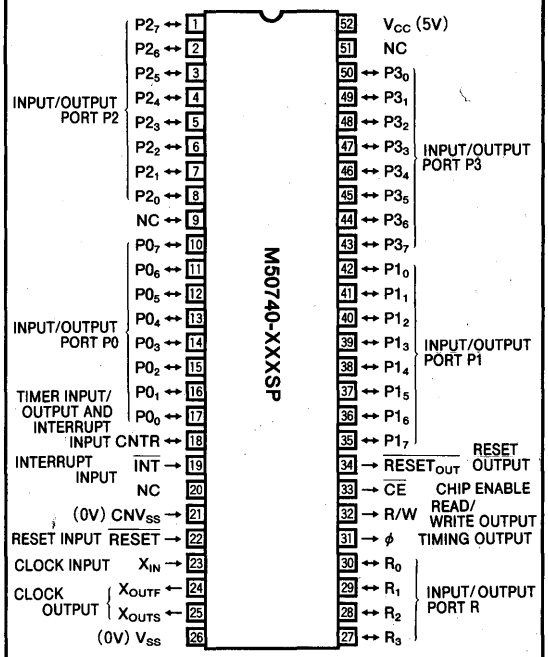
## FEATURES

- Basic machine instructions ..... 70
- Memory capacity ROM ..... 3072 bytes
  RAM ..... 96 bytes
- Instruction execution time (with shortest instruction, at 4MHz) ..... 2$\mu$s (min.)
- Single power supply ..... 5V ±10%
- Low power dissipation .. 15mW during operation (4MHz)
- Subroutine nesting ..... 48 levels (max.)
- Interrupts ..... 5 sources, 32 levels (max.)
- 8-bit timers ..... 3
- Programmable input/output (ports P0, P1, P2, P3) ..... 32

## APPLICATION

VTRs, tuners and audio equipment

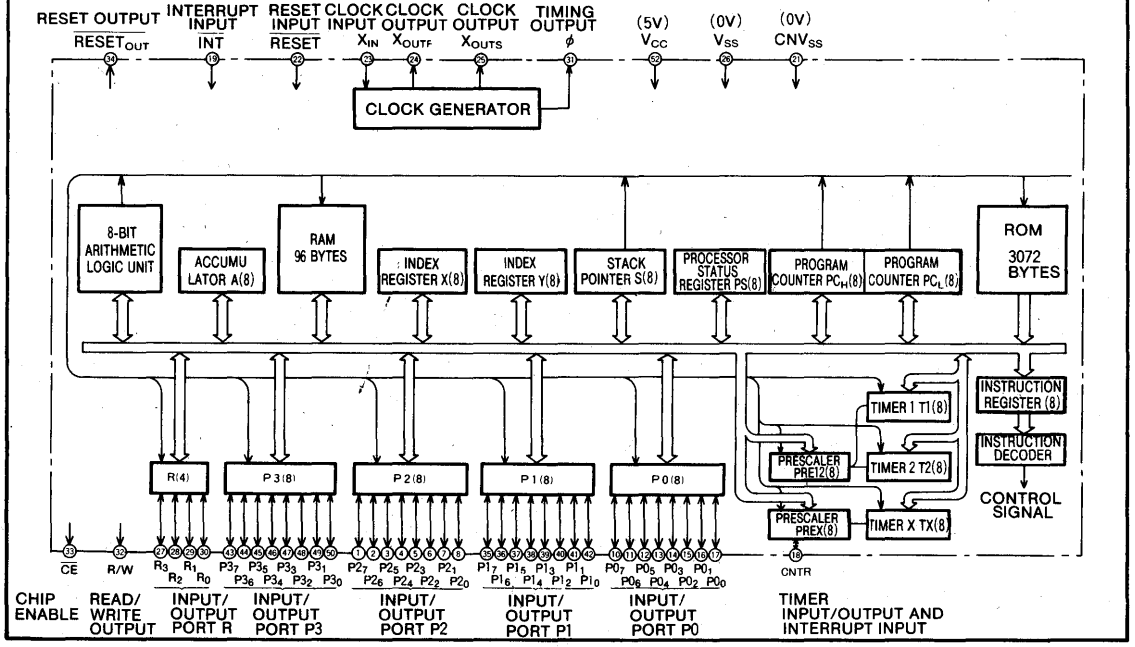
## PIN CONFIGURATION (TOP VIEW)




Outline 52P4B

NC : NO CONNECTION

## BLOCK DIAGRAM



## PERFORMANCE SPECIFICATIONS

| Parameter | | | Performance |
|---|---|---|---|
| Number of basic instructions | | | 70 |
| Execution time of basic instruction | | | 2$\mu$s (with shortest instructions, 4MHz clock frequency) |
| Clock frequency | | | 4MHz |
| Memory capacity | ROM | | 3072 bytes |
| | RAM | | 96 bytes |
| Input/output ports | $\overline{INT}$ | Input | 1 bit × 1 |
| | P0, P1, P2, P3 | Input/output | 8 bits × 4 |
| | R | Input/output | 4 bits × 1 |
| | CNTR | Input/output | 1 bit × 1 |
| Timers | | | 8-bit prescalers x 2 + 8-bit timers x 3 |
| Subroutine nesting | | | Max. 48 level |
| Interrupts | | | External interrupts (2), internal timer interrupts (3) |
| Clock generator | | | Built-in (externally connected RC circuit, ceramic or quartz resonator) |
| Supply voltage | During operation | | 5V ± 10% |
| Power dissipation | High-speed operation | | 15 mW (at 4MHz clock frequency) |
| | Low-speed operation | | 100 $\mu$W (at 20kHz clock frequency) |
| Input/output characteristics | Input/output withstanding voltage | | 12V (Ports P0, P1, P2, $\overline{INT}$, CNTR) |
| | Output current | | 10mA (Ports P0, P1, P2, P3) |
| Memory expansion | | | Possible |
| Ambient operating temperature | | | −10~70℃ |
| Device structure | | | CMOS silicon gate process |
| Package | | | 52-pin shrink plastic molded DIL |

## PIN DESCRIPTION

| Pin | Name | Input or output | Function |
|---|---|---|---|
| $V_{CC}$<br>$V_{SS}$ | Supply voltage | In | 5V ±10% supplied to $V_{CC}$, 0V supplied to $V_{SS}$ |
| $\overline{CNV_{SS}}$ | $CNV_{SS}$ input | In | To be connected to $V_{SS}$. |
| $\overline{RESET}$ | Reset input | In | When this input is kept low for at least 2$\mu$s, the reset state is enabled. |
| $X_{IN}$ | Clock input | In | The clock generator circuit is built-in. For setting the oscillation frequency, either connect the external RC circuit to $X_{IN}$ and $X_{OUTS}$ or $X_{OUTF}$ or connect a ceramic or quartz resonator across $X_{IN}$ and $X_{OUTS}$. When using an external clock source connect the clock generator source to $X_{IN}$, leaving $X_{OUTF}$ and $X_{OUTS}$ open. For details, refer to the section on the clock generator circuit. |
| $X_{OUTS}$ | Clock output | Out | Internal clock generator output. An RC circuit or ceramic or quartz resonator is connected between this output and $X_{IN}$ to control the oscillation frequency. For details, refer to the section on the clock generator circuit. |
| $X_{OUTF}$ | Clock output | Out | Internal clock generator output. An RC circuit is connected between this output and $X_{IN}$ to control the oscillation frequency. For details, refer to the section on the clock generator circuit. |
| $\phi$ | Timing output | Out | Timing output |
| CNTR | Timer input/output and interrupt input | In/out | Timer X input/output pin and interrupt input pin. |
| $\overline{INT}$ | Interrupt input | In | Interrupt input pin. |
| $P0_0$~$P0_7$ | Input/output port P0 | In/out | This 8-bit input/output port has a direction register and for each bit the port is programmed to serve for input or output. The input mode is established during resetting. N-channel open-drain circuits are used for the outputs. For details, refer to the section on the input/output pins. |
| $P1_0$~$P1_7$ | Input/output port P1 | In/out | This is an 8-bit input/output port with virtually the same functions as those of port P0. |
| $P2_0$~$P2_7$ | Input/output port P2 | In/out | This is an 8-bit input/output port with virtually the same functions as those of port P0. |
| $P3_0$~$P3_7$ | Input/output port P3 | In/out | This is an 8-bit input/output port with virtually the same functions as those of port P0, but p-channel open-drain circuits are used for the outputs. |
| $R_0$~$R_3$ | Input/output port R | In/out | This 4-bit input/output port is used for connection with the I/O expander. |
| R/W | Read/write output | Out | Read/write signal output for I/O expander. |
| $\overline{CE}$ | Chip enable output | Out | Chip enable signal output for I/O expander. |
| $\overline{RESET_{OUT}}$ | Reset output | Out | Reset signal output for I/O expander. |

## BASIC FUNCTION BLOCKS

### Memory

Fig. 1 shows the memory map. The 3072-byte ROM extends from $1400_{16}$ to $1FFF_{16}$. The area from $1F00_{16}$ to $1FFF_{16}$ includes special addresses, and when the special page addressing mode is used with the JSR instruction, subroutines on these pages can be called with two bytes. The area from $1FF4_{16}$ to $1FFF_{16}$ includes the reset and interrupt vector addresses. For details, refer to the section on interrupts.

The addresses from $0000_{16}$ to $00FF_{16}$ are own as the zero page and access to this page can be achieved with two bytes by using the zero page addressing mode, which reduces the number of programming steps. The memories used frequently, such as the RAM, input/output ports and timers, are allocated to the zero page.

From $0000_{16}$ to $005F_{16}$ is the RAM space and the size is 96 bytes. Apart from storing data, the RAM is also used as a stack for subroutine calls or interrupts.

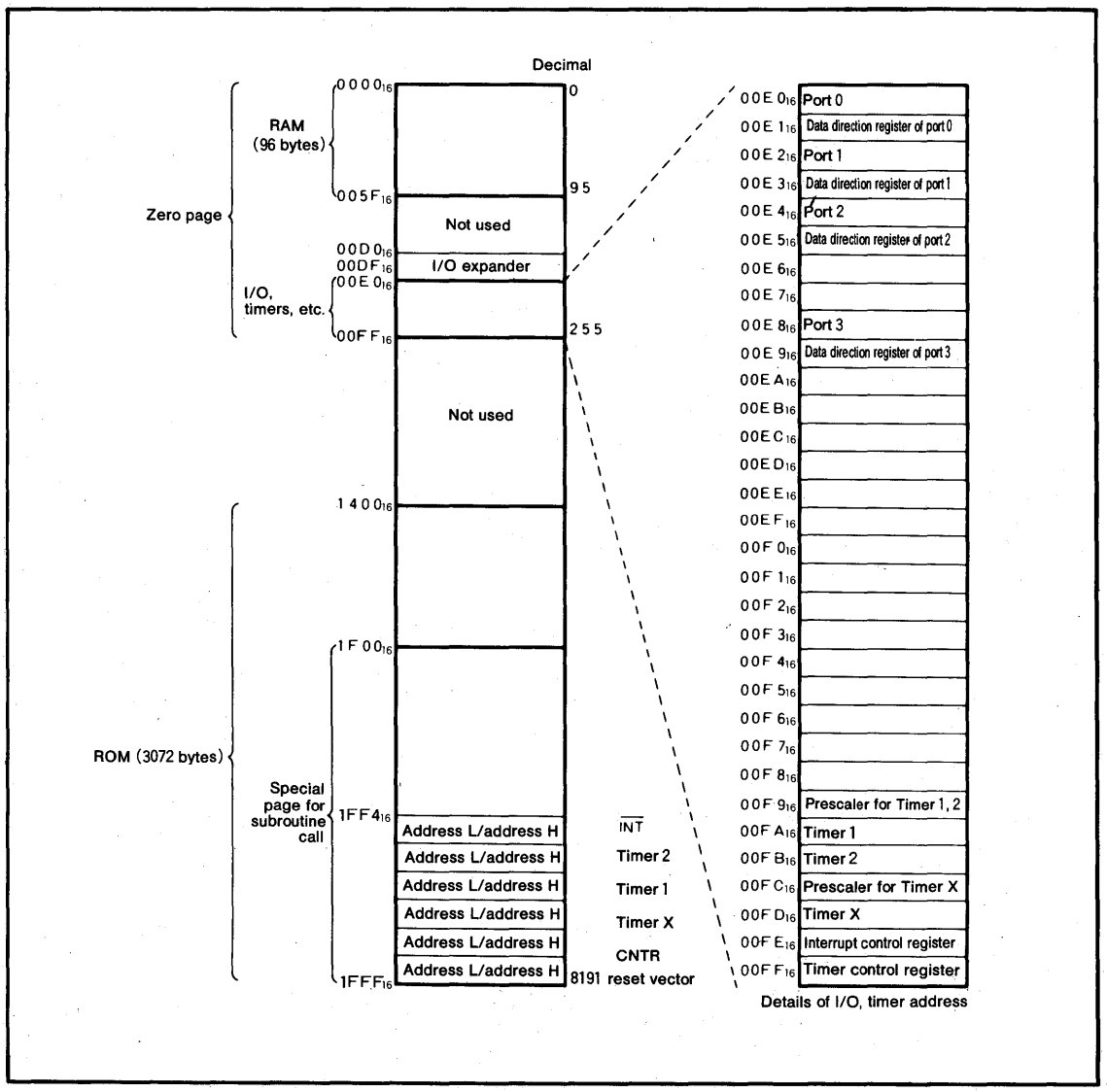



Fig.1 Memory layout

## CPU

Six registers, as shown in Fig. 2, are contained inside the CPU. Each of these register is now described in turn.

## Accumulator A

The accumulator is the 8-bit register and is heart of the microcomputer. Arithmetic and logic operations, transfers and processing of input/output and other data are performed centering on this register.

## Index Register X

This is an 8-bit register. In the index addressing mode where this register serves as the index register, the contents of this register and the contents of the program counter is added and the result is actual address. When flag T in the program status register is "1," the contents of index register X become the other operand address.

## Index Register Y

This is also an 8-bit register. In the index addressing mode where this register serves as the index register, the contents of this register and the contents of the program counter is added and the result is actual address.

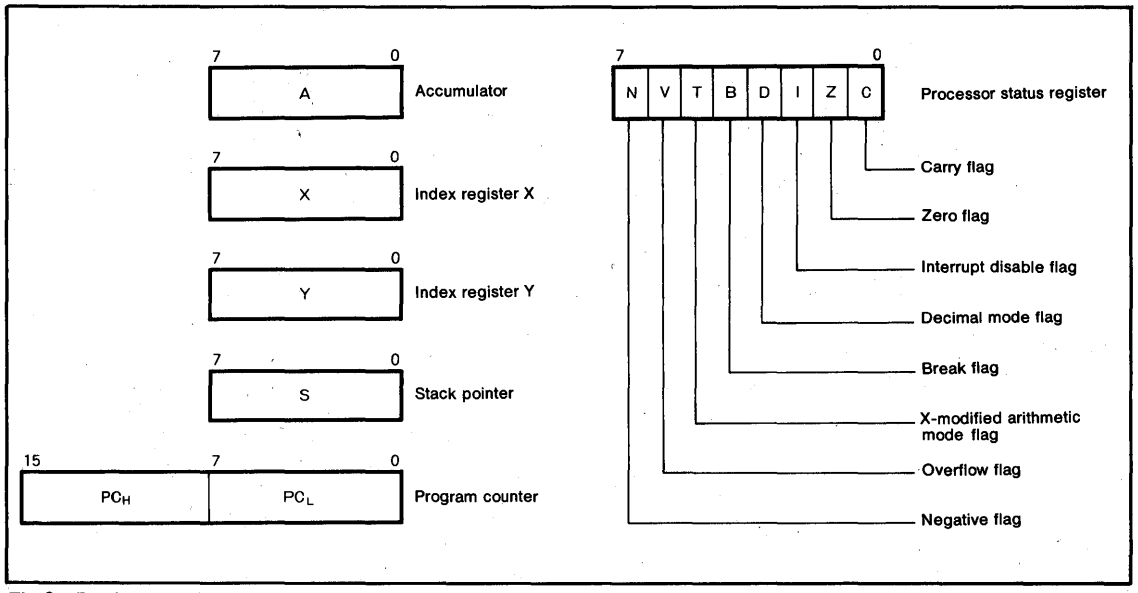



Fig.2 Register configuration

## Stack Pointer S

The stack pointer is an 8-bit register used for calling subroutines and for interrupts. When an interrupt is acknowledged, the high-order contents of the program counter are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," the contents of the stack pointer are then decremented by 1, the low-order contents of the program counter are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," the contents of the stack pointer are then further decremented by 1, and the contents of the program status register are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0."

This operation is performed automatically when an interrupt is acknowledged. The RTI instruction is used to return from the interrupt routine, and when it is executed, the stack pointer is incremented by 1 and returned in the reverse sequence to that described above. Since the contents of accumulator are not saved automatically, the PHA instruction must be used for this purpose. When the PHA instruction is executed, the contents of the accumulator are saved in the address where the low-order address are the contents of the stack pointer and the high-order address are "0," and the contents of the stack pointer are decremented by 1. The accumulator is returned by the PLA instruction. When this instruction is executed, the contents of the stack pointer are incremented by 1 and the contents of the address where the low-order address are the stack pointer contents and the high-order address are "0" enter the accumulator.

Similarly, the contents of the program status register are saved and returned by the PHP and PLP instructions respectively. With a subroutine call, program counter saving only is performed and this necessitates saving on the program for registers which must not be destroyed. The RTS instruction is employed to return from the subroutine.

## Program Counter PC

This is a 16-bit counter consisting of $PC_H$ and $PC_L$, both is 8 bit register. $PC_H$ is 8 bit register, but only 5 bits are actually used. The program counter specifies the address of the program memory which is to be executed next.

## Processor Status Register PS

This 8-bit register consists of the flags that hold the status immediately after arithmetic and logic operations. The C, Z, V and N flags can be tested and branched using the branch instructions. Each bit of the register is described in detail below.

**1. Carry Flag C**

The carry flag C is used to store carry or overflow after execution of arithmetic and logic operations by the arithmetic logic unit. It also undergoes change with the shift and rotate instructions. It can be set or reset directly using the SEC and CLC instructions.

**2. Zero Flag Z**

This flag is set when the results of data transfer or arithmetic and logic operations are "0" and reset when they are not "0."

**3. Interrupt Disable Flag I**

This flag disables all interrupts when its contents are "1." When an interrupt is acknowledged, its contents are automatically made "1." The flag can be set or reset by the program using the SEI and CLI instructions.

**4. Decimal Mode Flag D**

This flag determines whether additions and subtractions are to be undertaken by the binary or decimal mode. The ordinary binary mode is used when its contents are "0", while 1 word is processed as a 2-digit decimal number when its contents are "1." Decimal corrections are performed automatically. The SED and CLD instructions are used for setting and resetting.

**5. Break Flag B**

Operation is the same for interrupts when the BRK instruction is executed. This instruction is used for debugging programs. The BRK instruction interrupt vector and the interrupt vector of the lowest order of priority are located in the same address. In order to discriminate whether or not an interrupt has occurred with the BRK instruction, the contents of flag B are set to "1" when interrupted by the BRK instruction; at all other times, the contents are set to "0" and saved. It is possible to ascertain whether an interrupt has occurred with BRK by investigating the bit saved in the interrupt routine.

**6. X-modified Arithmetic Mode Flag**

Arithmetic and logic operations are performed between the accumulator and memory when the flag T contents is "0." When this bit is "1," the accumulator is bypassed and operations are performed directly between the memories. The results of such operations between memory 1 and memory 2 enter memory 1. The memory 1 address is specified by the contents of index register X; the memory 2 address is specified by the ordinary addressing mode. The SET and CLT instructions are used for setting and resetting flag T.

**7. Overflow Flag V**

This flag is significant when in the addition and subtruction a single word is treated as a signed binary number. It is set when the results of an addition or subtraction exceed +127 or −128. Apart from this, the 6th bit of the memory subject to the execution of the BIT instruction enters the overflow flag when this instruction is executed. The CLV instruction is used to clear the overflow flag. A setting instruction is not provided.

**8. Negative Flag N**

This flag is set when the results of an arithmetic or logic operation or of data transfer are negative (7th bit is "1"). In addition, the 7th bit of the memory subject to the BIT instruction enters the negative flag when this instruction is executed. Instructions to set and reset this flag are not provided.

**Table 1 Interrupt vector addresses and priority**

| Interrupt source | Priority | Vector address |
|---|---|---|
| $\overline{\text{RESET}}$ | 1 | 1FFF, 1FFE |
| CNTR | 2 | 1FFD, 1FFC |
| Timer X | 3 | 1FFB, 1FFA |
| Timer 1 | 4 | 1FF9, 1FF8 |
| Timer 2 | 5 | 1FF7, 1FF6 |
| $\overline{\text{INT}}$ (BRK) | 6 | 1FF5, 1FF4 |

## INTERRUPTS

Interrupts include the interrupt from pin CNTR, the timer X interrupts, timer 1 interrupt, timer 2 interrupt, the interrupt from pin $\overline{\text{INT}}$ and the interrupt based on the BRK instruction. The interrupts are vector interrupts and Table 1 shows the vector table and priority. Resetting take the same action as interrupt and so it is descried here.

When an interrupt is acknowledged, the registers are saved, as described in the above section on stack pointer S, the interrupt disable flag I is set and a jump is made to the address indicated by the contents of the vector table. The interrupt request bit is automatically cleared. Resetting is not disabled by any condition. Interrupts (exclusive of resetting) are not acknowledged when the interrupt disable flag has been set. The interrupts from pin CNTR, timer X, timer 1, timer 2 and $\overline{\text{INT}}$ can be controlled individually by the interrupt control and timer control registers. This is shown in Fig. 3. When the interrupt enable bit is "1," when the interrupt request bit is "1" and when the interrupt disable flag I is "0," the interrupt is acknowledged. When the level of pins CNTR and $\overline{\text{INT}}$ change from high to low or when the contents of timer X, timer 1 or timer 2 reach to "0," the corresponding interrupt request bits are set.

These bits can be reset by programming but cannot be set. The interrupt enable bit can be set and reset by programming. Whether interrupt is caused by the BRK instruction, can be verified by checking break flag B which has been saved, as mentioned in the section on the break flag B.

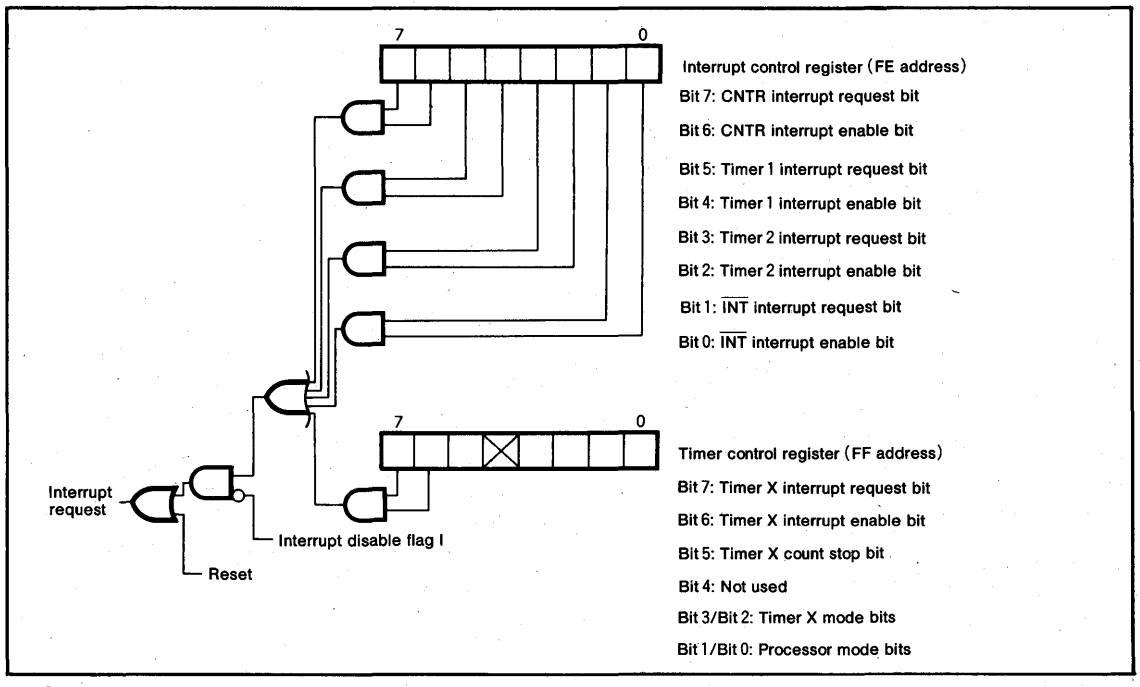



**Fig.3 Interrupt control**

## Timers

There are 3 timers: timer X, timer 1 and timer 2. Timer X has four modes which are selected by the value of the timer X mode bits (bit 2 and bit 3) in the timer control register. When the timer count stop bit (bit 5) is set to "1," all four timer X modes stop. Fig. 4 is a block diagram of timers X, 1 and 2. Timer 1 and timer 2 have a common prescaler composed of 8 bits. The frequency division ratio is determined by the prescaler contents. This ratio is $1/(n+2)$ when the prescaler latch contents are made n decimally. All the timers have 8-bit timer latches. The countdown system is featured for the timers, and the timer latch contents are reloaded into the timer at the following cycle when the counter contents reach to "0."

When the timer contents reach to "0," the interrupt request bit (on the interrupt control register or on the timer control register located in the $FE_{16}$ or $FF_{16}$ address respectively) corresponding to the timer is set to "1." Any number except "0" should be entered in the prescaler latch and timer latch.

Refer to the section on interrupts for details. The four modes of timer X are now described.

(1) Timer mode (00)
In this mode the frequency produced by dividing the oscillation frequency by 16, is counted. When the timer contents reach to "0," the interrupt request bit is set to "1," the timer latch contents are re-loaded and the count is continued.

(2) Pulse output mode (01)
Every time the timer contents reach to "0," the signal on the pin CNTR changes the polarity.

(3) Event counter mode (10)
Operation is the same as in the timer mode except for counting the signal from pin CNTR.

3

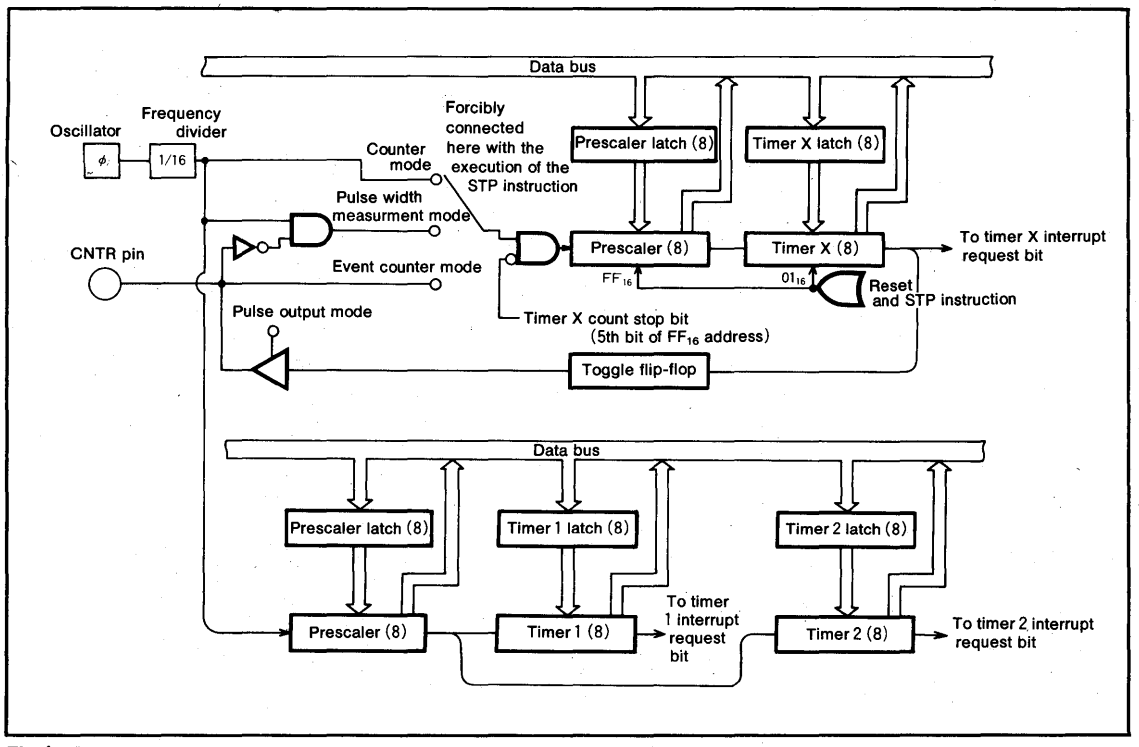



**Fig.4 Block diagram of timer X, timer 1, timer 2**

(4) Pulse width measurment mode (11)

The frequency, produced by dividing the oscillation frequency by 16, is counted only while the pin CNTR level is low. When the counter contents reach to "0," the interrupt request bit is set to "1," the timer latch contents are re-loaded and the count is continued.

Fig. 5 shows the relationship between the timer control register contents and the timer modes.

Also shown are the processor mode and other bits.

When reset or the STP instruction is executed, the timer X prescaler is set in $FF_{16}$ and the timer X latch is set in $01_{16}$. When the STP instruction is executed, the frequency produced by dividing the oscillation frequency by 16 serves as the timer X prescaler input, regardless of the timer X mode bit. This mode is released either when the timer X interrupt request bit is set to "1" or when resetting is accomplished and resume the mode determined by the timer X mode bit. For details on the operation of the STP instruction, reference should be made to the section on the oscillator circuit.

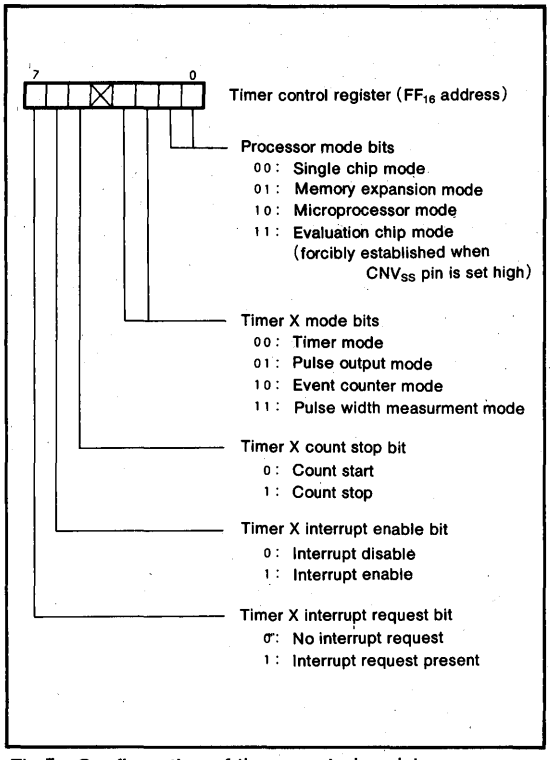



Fig.5 Configuration of timer control register

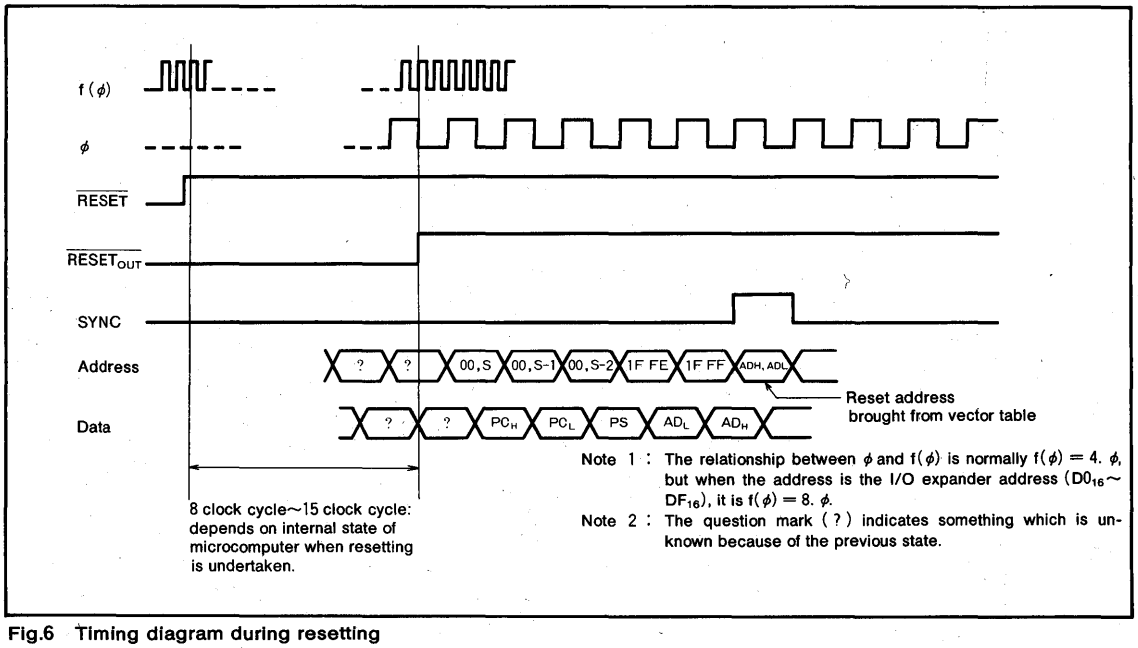



Fig.6 Timing diagram during resetting

## Reset Circuit

When a supply voltage of 5V ±10% is being supplied to the M50740-XXXSP and the $\overline{\text{RESET}}$ pin is returned to the high level after being kept at the low level for 2μs or more, the reset is released in accordance with the sequence shown in Fig. 6, and the program starts from the address which is derived from the contents of the address $1FFF_{16}$ and $1FFE_{16}$ ,high-order address is the contents of address $1FFF_{16}$ and low-order address is the contents of address $1FFE_{16}$. When resetting is accomplished, the internal state of the microcomputer is as shown in Fig. 7.

Fig. 8 shows an example of the reset circuit.

The reset input voltage should be set to less than 0.6V at that point when the supply voltage is passing through 4.5V.

| | | Address | |
|---|---|---|---|
| (1) | Data direction register of port 0 | ( $E1_{16}$ ) ··· | $00_{16}$ |
| (2) | Data direction register of port 1 | ( $E3_{16}$ ) ··· | $00_{16}$ |
| (3) | Data direction register of port 2 | ( $E5_{16}$ ) ··· | $00_{16}$ |
| (4) | Data direction register of port 3 | ( $E9_{16}$ ) ··· | $00_{16}$ |
| (5) | Prescaler X | ( $FC_{16}$ ) ··· | $FF_{16}$ |
| (6) | Timer X | ( $FD_{16}$ ) ··· | $01_{16}$ |
| (7) | Interrupt control register | ( $FE_{16}$ ) ··· | $00_{16}$ |
| (8) | Timer control register | ( $FF_{16}$ ) ··· | $00_{16}$ |
| (9) | Interrupt disable flag on the processor status register | ( PS ) ··· | 1 |
| (10) | Program counter | ( $PC_H$ ) ··· | contents of the address $1FFF_{16}$ |
| | | ( $PC_L$ ) ··· | contents of the address $1FFE_{16}$ |
| (11) | The oscillator output is connected to pin $X_{OUTF}$ as with the state established after the FST instruction has been executed. | | |

Fig.7 Internal state of microcomputer after resetting

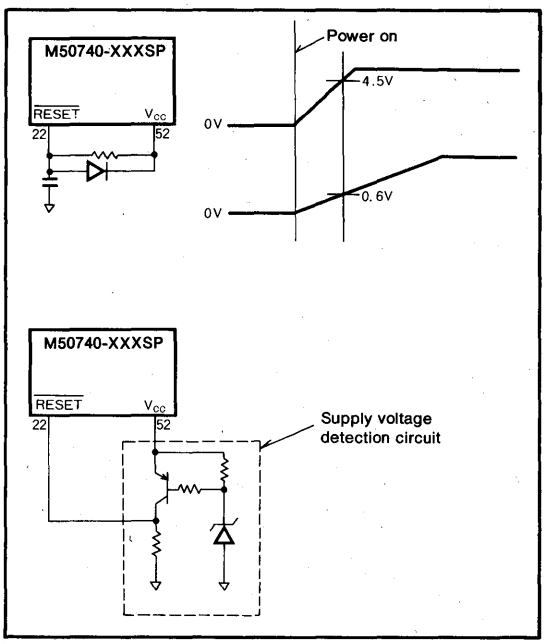



Fig.8 Example of reset circuit

## Input/Output Pins

(1) Port P0

This port is an 8-bit input/output port with n-channel open-drain outputs. As shown in the memory map of Fig. 1, port P0 is treated as the memory of address $E0_{16}$ on the zero page. Port P0 has a data direction register (address $E1_{16}$ on zero page) and programming can be undertaken for individual bit to use the port for input or output. The pins where the data direction register is programmed to "1" are for output and those where the register is programmed to "0" are for input. The data written into the pin programmed as an output pin are written into the port latch and supplied direct to the output pin. When reading the data from a pin programmed as an output pin, it is not the output pin contents which are read but the port latch contents. Consequently, since the LED or other similar part is driven directly, the value output previously can be read correctly even if the low-level output voltage rises. The pin programmed as an input pin remains floating, so external signal can be read. When data are written, they are written into the port latch only and the pin remains floating.

(2) Port P1
This has the same functions as port P0.

(3) Port P2
This has the same functions as port P0.

(4) Port P3
Apart from the fact that this port has p-channel open-drain outputs, its functions are the same as those for port P0. Fig. 9 is a block diagram of port P0~P3.
Also indicated are the output structure of port R, CNTR, $\phi$, R/W, $\overline{CE}$ and $\overline{RESET}_{OUT}$.

(5) Port R
This port is for exchanging data with the I/O expander. When $\phi$ is high, the port address of the I/O expander is sent; when it is low, data are sent to or received from the expander. The above data and addresses are effective only when pin $\overline{CE}$ is low. Fig. 10 is a timing diagram.

(6) $\overline{CE}$
This pin is set low when the address becomes the I/O expander address ($D0_{16}$~$DF_{16}$). It is used to inform the I/O expander that the port R address or data is effective.

(7) R/W
This is set low while writing is being executed, and it is used to inform the I/O expander that either writing or reading is being undertaken.

(8) $\phi$
Normally output to this pin is a signal with a frequency produced by dividing the clock frequency by 4. However, when pin $\overline{CE}$ is low, an output with a frequency which is one-eighth of the clock frequency is output. The pin is used to provide synchronization with the I/O expander.

(9) $\overline{RESET}_{OUT}$
When the $\overline{RESET}$ pin is set low, this pin also goes low. When the $\overline{RESET}$ pin is set high, the pin also goes high after between 8 and 15 clock cycles (this depends on the internal state of the microcomputer). The $\overline{RESET}_{OUT}$ pin itself is used to reset the I/O expander.

(10) $\overline{INT}$
When an input which changes the level from high to low is applied to this interrupt input pin, the $\overline{INT}$ interrupt request bit (bit 1 of address $FE_{16}$) is set to "1."

(11) CNTR
This pin serves both as the timer X input/output pin and as the interrupt input pin. When an input which changes its level from high to low is applied, the CNTR interrupt request bit (bit 7 of address $FE_{16}$) is set to "1." The pin serves as the external pulse input pin in the event counter mode. In the pulse output mode a pulse which reverses its polarity is output every time the timer X contents are reach to "0." In the pulse width measument mode, the pulse to be measured is supplied to this pin.

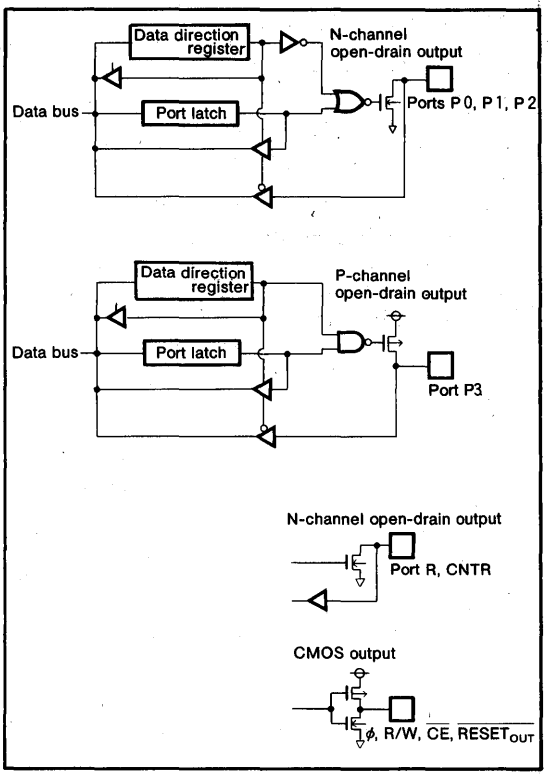



Fig.9 Block diagram of port P0~P3 (single chip mode) and output formats of port R, CNTR, R/W, $\overline{CE}$ and $\overline{RESET}_{OUT}$

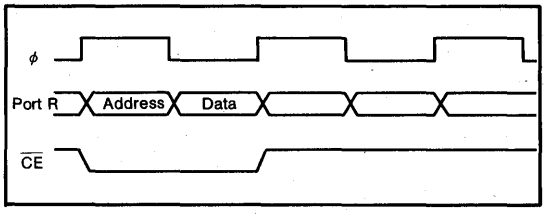



Fig.10 Timing diagram of port R

## Clock Generator Circuit

The clock generator circuit is built-in, as shown in Fig. 11. When the STP instruction is executed, oscillation is stopped with the internal clock $\phi$ in the high-level. Furthermore, $FF_{16}$ is set in prescaler X and $01_{16}$ in timer X, and the output, one-sixteenth of the oscillator output, is forcibly connected to the prescaler X input. This connection is released when, as mentioned in the timer section, timer X overflows or when resetting is accomplished. Oscillation re-starts when an interrupt is acknowledged but the internal clock $\phi$ remains high until timer X overflows. Only when timer X overflows is the internal clock $\phi$ supplied. This is because time is required for the oscillation to rise when a ceramic resonator or similar part is employed.

When the FST instruction is executed, $SW_{OSC}$ closes and when the SLW instruction is executed, it opens. These instructions are used when RC oscillation is emploied and the oscillation frequency is changed. $SW_{OSC}$ is closed during resetting.

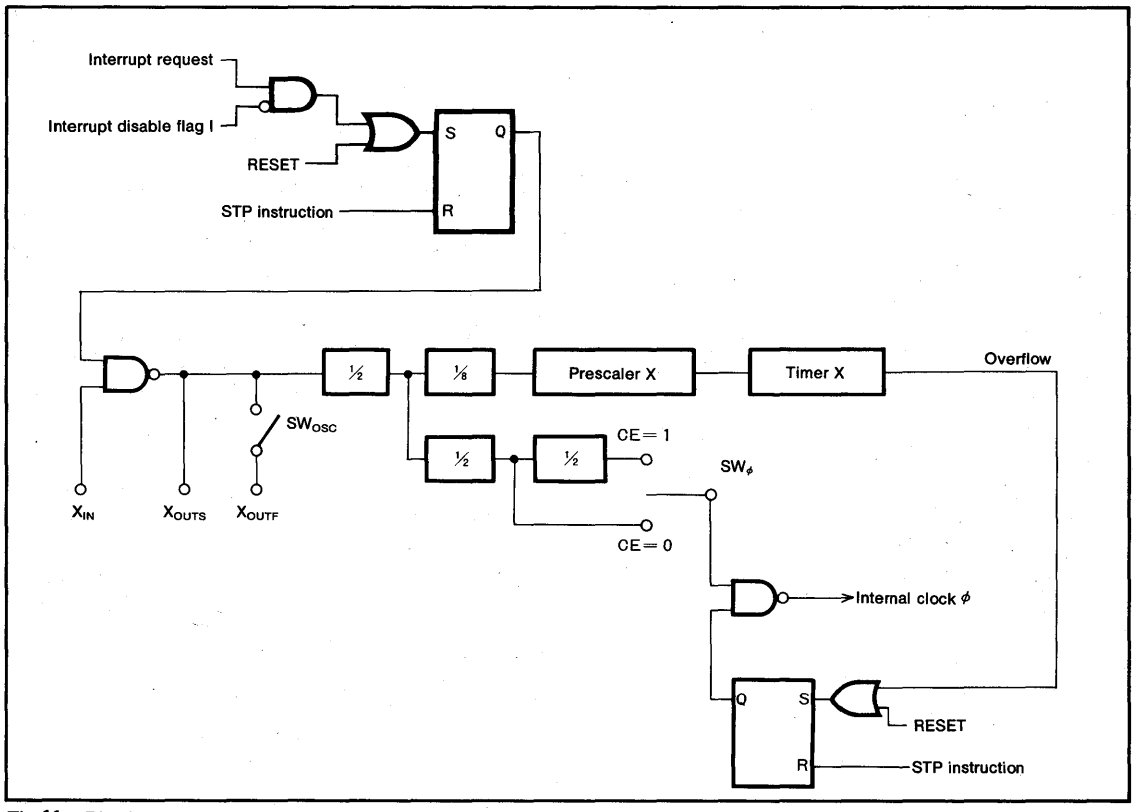



Fig.11 Block diagram of clock generator circuit

When the address becomes the I/O expander address ($D0_{16} \sim DF_{16}$), SW$\phi$ is connected to the output (CE = 1) which is one-eighth of the oscillation frequency and at all other times it is connected to the output (CE = 0) which is one-fourth of the same frequency. This is because a margin in terms of time is given to the signal exchange with the I/O expander.

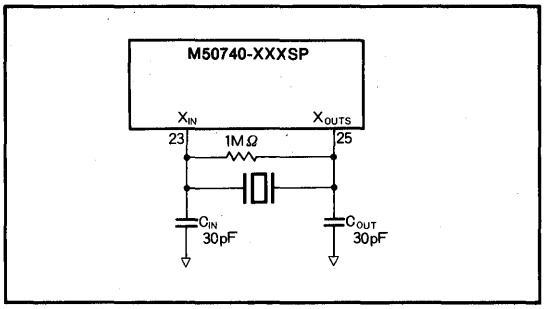



Fig.12 Externally connected ceramic resonator circuit

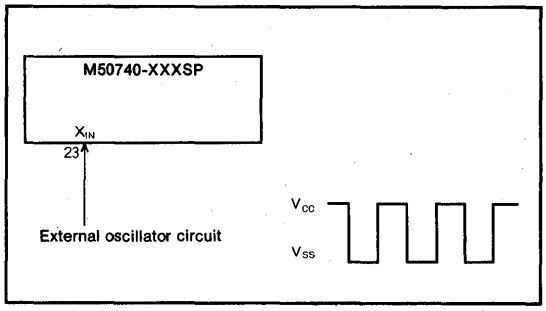



Fig.13 External clock input circuit

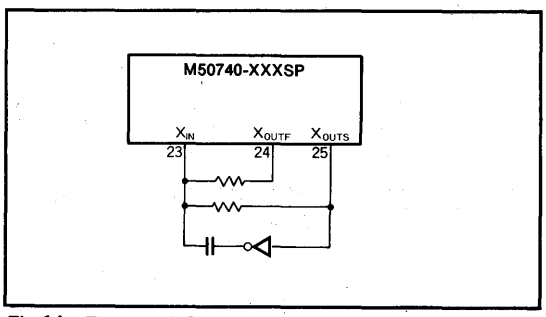



Fig.14 External RC circuit

Figs. 12~14 give examples of clock generator circuits. The clock signal is produced if a ceramic resonator (or quartz crystal) is externally connected. $X_{OUTF}$ is left open. The capacitance and other constants depend on the resonator itself and the values recommended by the manufacturer in question should be used.

When the external clock source is used it should be applied to the $X_{IN}$ pin with pins $X_{OUTS}$ and $X_{OUTF}$ left open.

An inverter is required externally for RC oscillation.

## Addressing Modes

The M50740-XXXSP has 17 addressing modes and an extremely powerful memory access capability.

When extracting data required for arithmetic and logic operations from the memory or when storing the results of such operations in a memory using the appropriate instructions for this purpose, the memory address must be specified. Even when jumping to an address during a program, that particular address must be specified. The specification of the memory address is called addressing. The data required for addressing and the registers involved are now described. The M50740-XXXSP's instructions can be classified into three kinds, as shown in Fig. 15, by the byte number in the program memory required for configuring the instruction: 1-byte, 2-byte and 3-byte instructions. In each case, the first byte is known as the "operation code" which forms the basis of the instruction. The second or third byte is called the "operand" which affects the addressing. The contents of index registers X and Y also effect the addressing.

However many the addressing modes, there is no difference in the sense that a particular memory is specified. What differs is whether the operand or the index register contents or a combination of both should be used to specify the memory or jump destination. Based on these 3 methods, the range of variation is increased and the M50740-XXXSP's operation is enhanced by combinations of the bit operation instructions, jump instruction and arithmetic instructions. The accumulator or register is specified with a 1-byte instruction and so there is no operand byte, which is the part specifying the memory.

Actual addressing modes are now described by type.

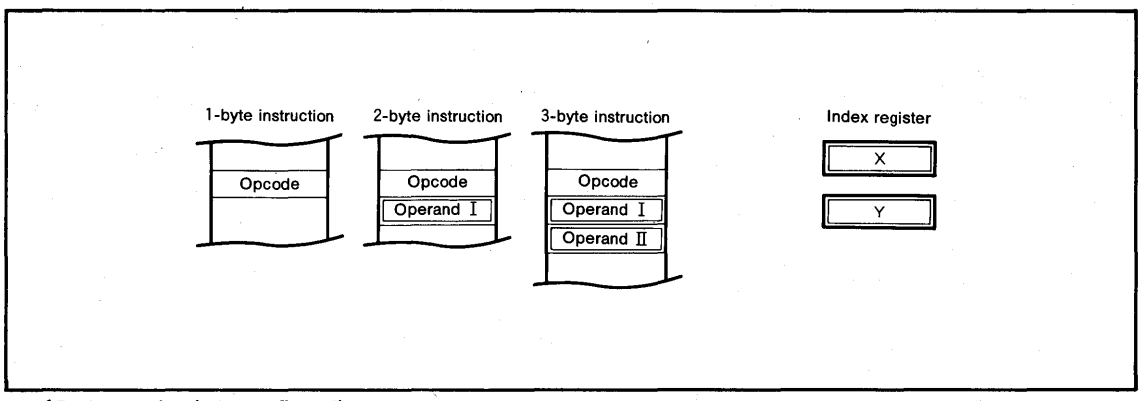
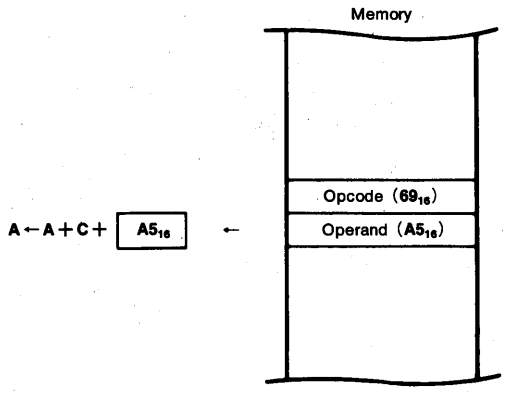


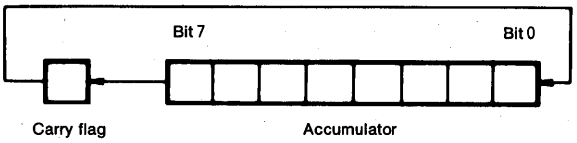

Fig.15 Instruction byte configuration

Name : Immediate addressing mode
Function : Operand follow immediate after opcode.
Instructions : **ADC, AND, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, SBC**
Example : Mnemonic — Machine code
**ADC #$A5** — **$69_{16}$ $A5_{16}$**

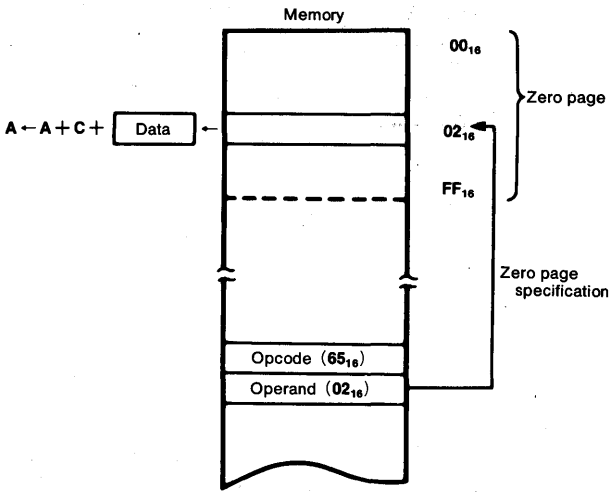
* This symbol designates the immediate addressing mode.



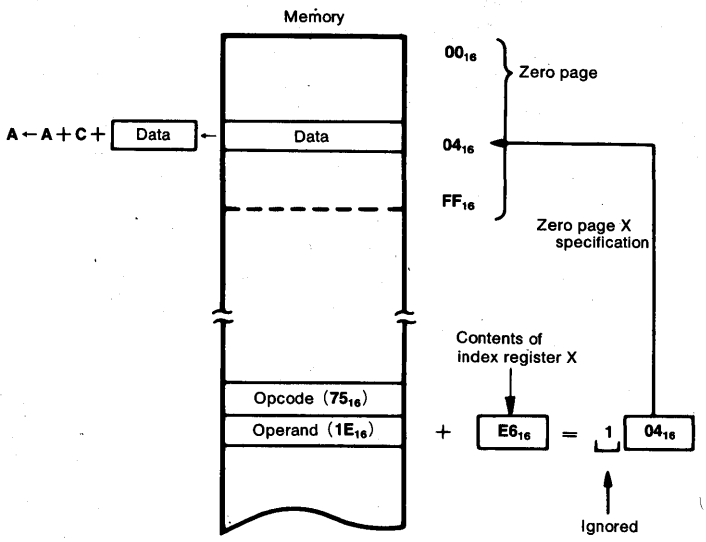

Name : Accumulator addressing mode
Function : Operation is performed on accumulator.
Instructions : **ASL, DEC, INC, LSR, ROL, ROR**
Example : Mnemonic — Machine code
**ROL A** — **$2A_{16}$**



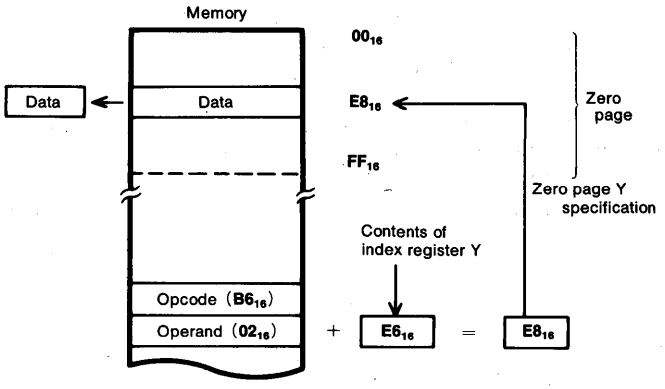
Name : Zero page addressing mode
Function : Operation is performed on the zero page memory ($00_{16}$~$FF_{16}$)
Instructions : **ADC, AND, ASL, BIT, CMP, COM, CPX, CPY, DEC, EOR, INC, LDA, LDM, LDX, LDY, LSR, ORA, ROL, ROR, RRF, SBC, STA, STX, STY, TST**
Example : Mnemonic — Machine code
**ADC $02** — $65_{16}$ $02_{16}$



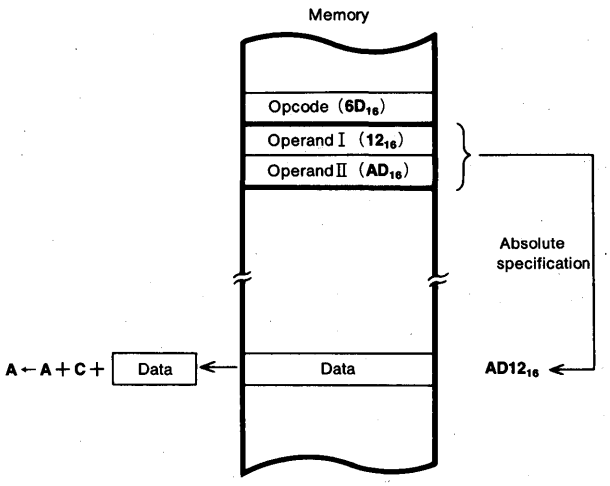

Name : Zero page X addressing mode
Function : Operation is performed on the memory which address is specified by adding the operand and contents of index register X.
Instructions : **ADD, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA, STY**
Example : Mnemonic — Machine code
**ADC $1E,X** — $75_{16}$ $1E_{16}$



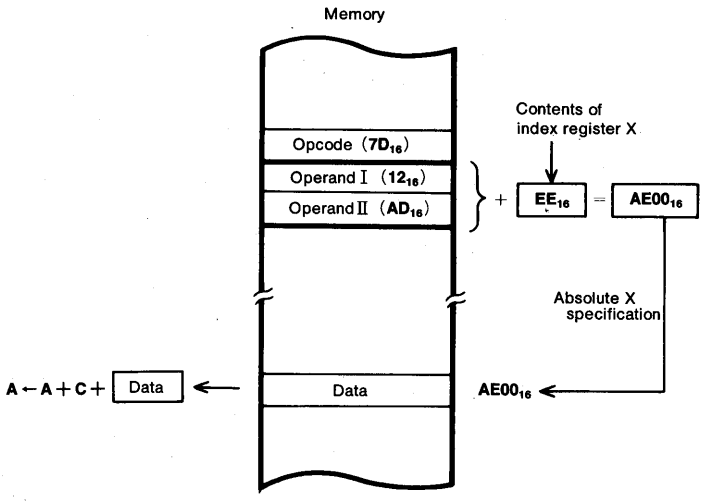
Name : Zero page Y addressing mode
Function : Operation is performed on the memory which address is specified by adding the operand and contents of index register Y.
Instructions : **LDX, STX**
Example : Mnemonic Machine code
**LDX $02,Y** $B6_{16}$ $02_{16}$



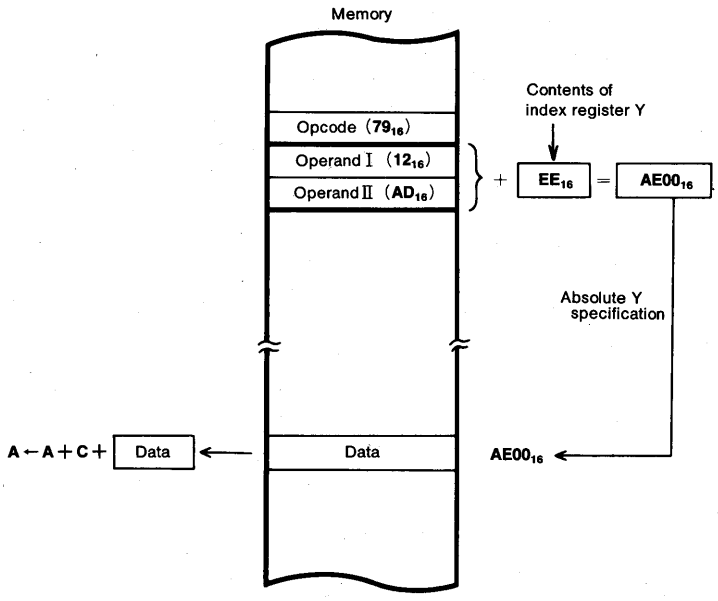

Name : Absolute addressing mode
Function : Operation is performed on the memory which address is specified by first and second operand.
Instructions : **ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, EOR, INC, JMP, JSR, LDA, LDX, LDY, LSR, ORA, ROL, ROR, SBC, STA, STX, STY**
Example : Mnemonic Machine code
**ADC $AD12** $6D_{16}$ $12_{16}$ $AD_{16}$



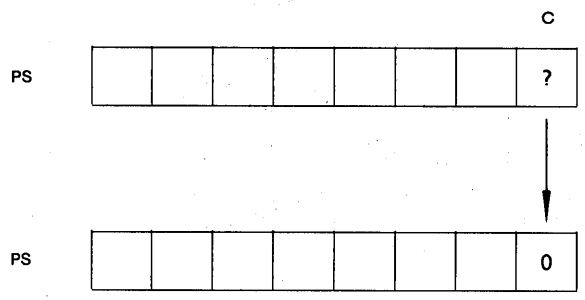
Name : Absolute X addressing mode
Function : Operation is performed on the memory which address is specified by adding the contents of index register X and value indicated first and second operand.
Instructions : **ADC, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA**
Example : Mnemonic Machine code
**ADC $AD12,X** $7D_{16}$ $12_{16}$ $AD_{16}$




Name : Absolute Y addressing mode
Function : Operation is performed on the memory which address is specified by adding the contents of index register Y and value indicated first and second operand.
Instructions : **ADC, AND, CMP, EOR, LDA, LDX, ORA, SBC, STA**
Example : Mnemonic Machine code
**ADC $AD12,Y** $79_{16}$ $12_{16}$ $AD_{16}$



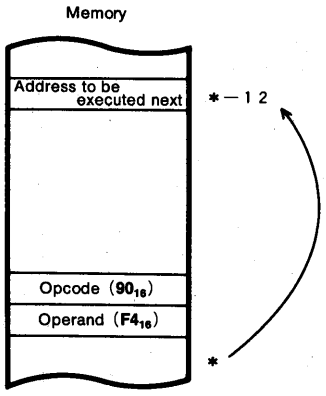
Name : Implied addressing mode
Function : Implied addressing mode need no operand.
Instructions : **BRK, CLC, CLD, CLI, CLT, CLV, DEX, DEY, FST, INX, INY, NOP, PHA, PHP, PLA, PLP, RTI, RTS, SEC, SED, SEI, SET, SLW, STP, TAX, TSX, TAY, TXA, TYA**
Example : Mnemonic Machine code
**CLC** $18_{16}$



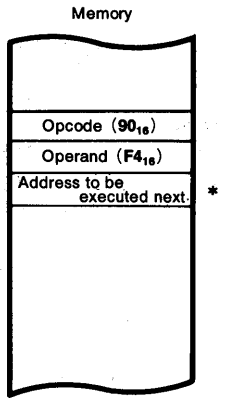

Carry flag reset

Name : Relative addressing mode
Function : Jumps to address which is produced by adding the contents of program counter and the contents of operand.
Instructions : **BCC, BCS, BEQ, BMI, BNE, BPL, BRA, BVC, BVS**
Example : Mnemonic Machine code
**BCC ＊−12** $90_{16}$ $F4_{16}$

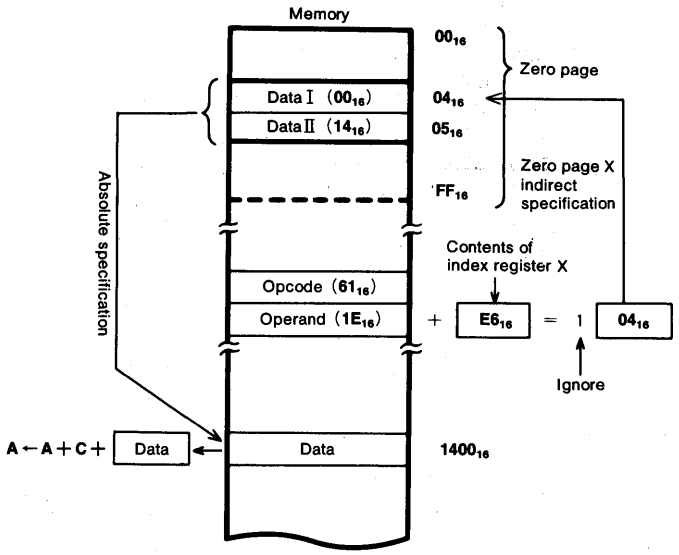
Jumps to − 12 address when carry flag(c) is cleared.




Proceed to next address when carry flag(c) is set.

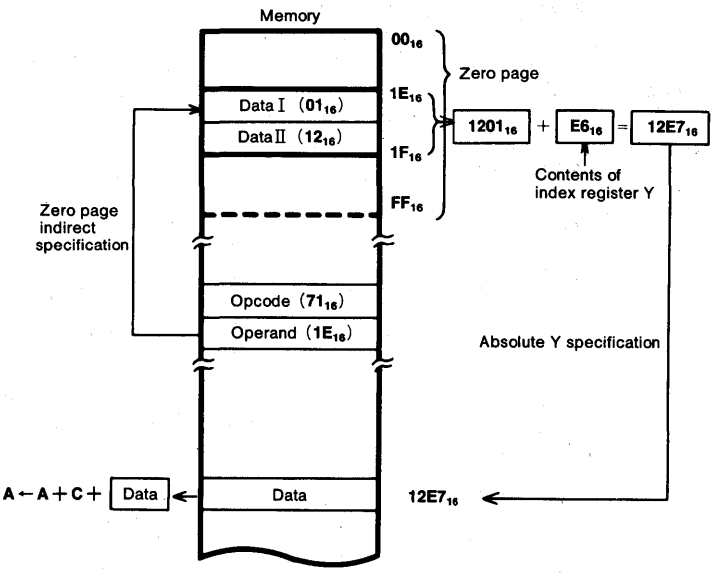


Name : Indirect X addressing mode
Function : Operation is performed on the memory at address indicated by contents of consecutive 2 byte memory which first address is formed by adding operand and contents of index register X.
Instructions : **ADC, AND, CMP, EOR, LDA, ORA, SBC, STA**
Example : Mnemonic Machine code
**ADC ($1E,X)** $61_{16}$ $1E_{16}$



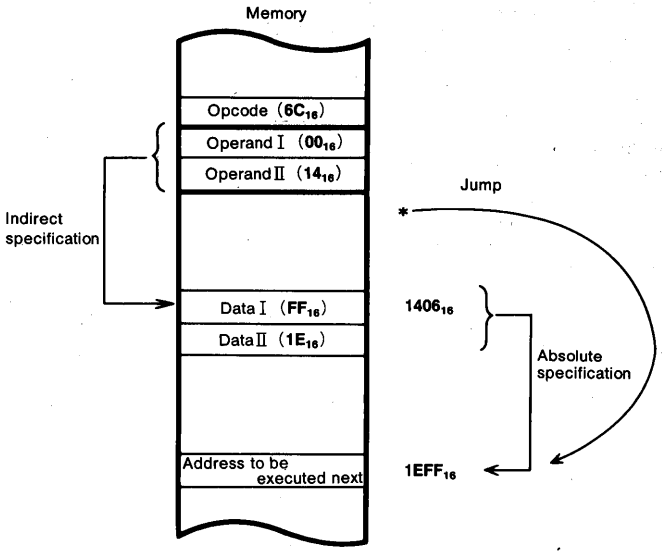

In this example, $00_{16}$ as data I and $14_{16}$ as data II have been stored beforehand.

Name : Indirect Y addressing mode
Function : Operation is performed on the memory addressed by adding the contents of index register Y and contents of consecutive 2 byte zero page memory which first address is specified by operand.
Instructions : **ADC, AND, CMP, EOR, LDA, ORA, SBC, STA**
Example : Mnemonic Machine code
**ADC ($1E),Y** $71_{16}$ $1E_{16}$



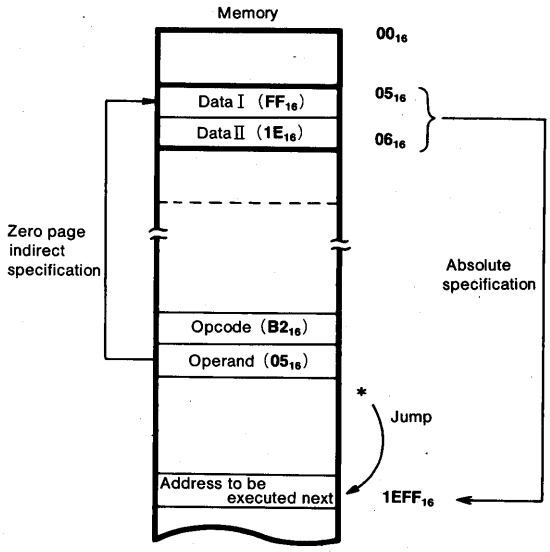

In this example, $00_{16}$ as data I and $12_{16}$ as Data II have been stored beforehand.

Name : Indirect absolute addressing mode
Function : Specifies consecutive 2-byte memories by contents of first and second operand and jumps to address indicated by contents of these memories.
Instructions : **JMP, JSR**
Example : Mnemonic Machine code
**JMP $1400** **$6C_{16}$ $00_{16}$ $14_{16}$**



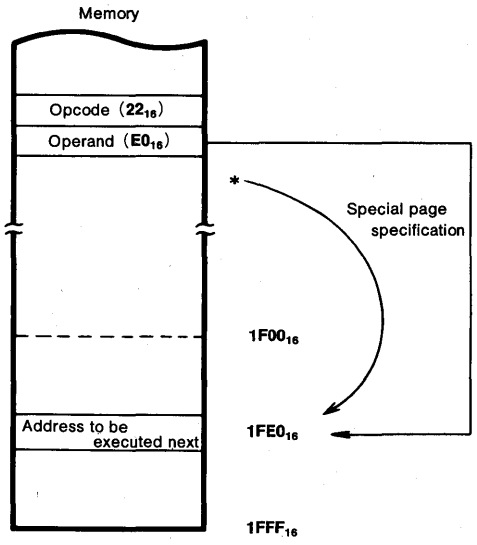

In this example, $FF_{16}$ as data I and $1E_{16}$ as data II have been stored beforehand.

Name : Zero page indirect absolute addressing mode
Function : Specifies consecutive 2-byte memories in zero page area by operand contents and jumps to adddress indicated by contents of these memories.
Instructions : **JMP, JSR**
Example : Mnemonic Machine code
**JMP $05** **$B2_{16}$ $05_{16}$**

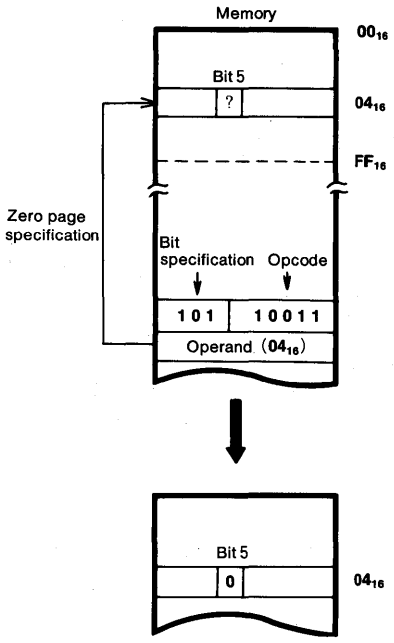



In this example, $FF_{16}$ as data I and $1E_{16}$ as data II have been stored beforehand.

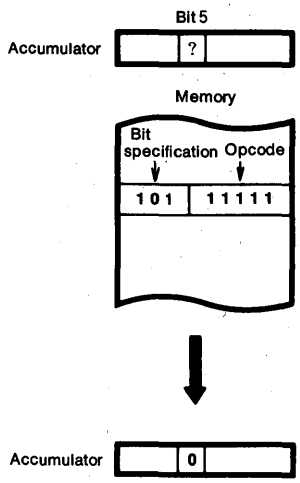
Name : Special page addressing mode
Function : Jumps to address in special page area. 8 high-order address and 8 low-order address of jump distination is $1F_{16}$ and contents of operand respectively.
Instruction : **JSR**
Example : Mnemonic Machine code
**JSR ¥$1FE0** $22_{16}$ $E0_{16}$
↑
*This symbol denotes special page mode.




Name : Zero page bit addressing mode
Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by operand.
Instructions : **CLB, SEB**
Example : Mnemonic Machine code
**CLB 5,$04** $BF_{16}$ $04_{16}$

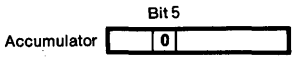


Name : Accumulator bit addressing mode
Function : Specifies bit in accumulator by 3 high-order bits of opcode.
Instructions : **CLB, SEB**
Example : Mnemonic Machine code
**CLB 5,A** **$BB_{16}$**

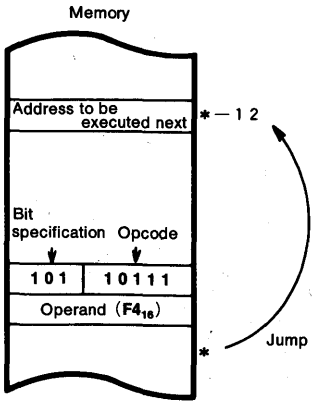



Name : Zero page bit addressing mode
Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by operand.
Instructions : **CLB, SEB**
Example : Mnemonic Machine code
**CLB 5,$04** **$BF_{16}$ $04_{16}$**

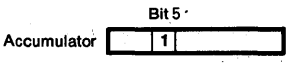
When accumulator bit 5 is cleared



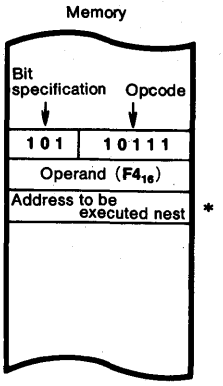

Jump to * −12 address




When accumulator bit 5 is set




Advance to * address



Name : Zero page bit relative addressing mode

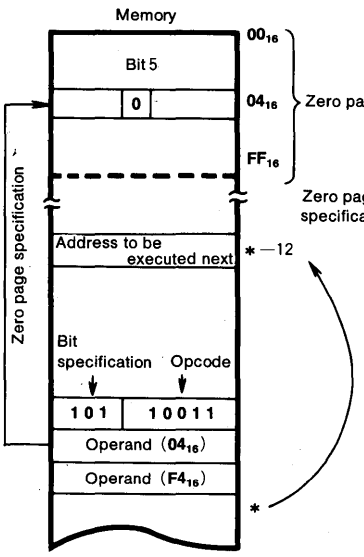
Function : Operation is performed on the bit specified by 3 high-order bits of opcode, memory address containing this bit is specified by first operand and, depending on the state of this special bit, jumps to the address indicated by the value produced by adding the second operand contents to the contents of the program counter.

Instructions : **BBC, BBS**

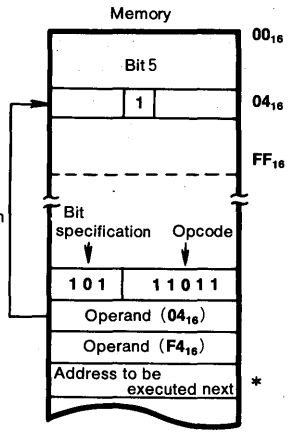
Example : Mnemonic Machine code

**BBC 5,$04,−12** $B7_{16}$ $04_{16}$ $F4_{16}$

Jump to * − 12 address when $04_{16}$ address bit 5 is cleared.




Advance to * address when $04_{16}$ address bit 5 is set.




## Documentation Required for Ordering a Custom Mask

The following information should be provided when ordering a custom mask:

(1) M50740-XXXSP mask confirmation sheet

(2) ROM data EPROM 3 sets

## Programming Precautions

(1) The frequency division ratio of the timers and prescalers is not 1/(n+1) but 1/(n+2).

(2) Select any numerical value except 0 for the set values of the timers and prescalers.

(3) Even when the BBC or BBS instruction is executed immediately after the contents of the interrupt request bit has been changed by the program, the execution is still valid for the contents prior to the change. This means that for execution keyed to the contents subsequent to the change, the instruction should be executed after one of more instructions.

(4) Data should be read from the timers and prescalers while there is no change in the prescaler input.

(5) The decimal mode flag D is set to "1" and the ADC or SBC instruction is executed with decimal arithmetic and logic operations. In this case, the SEC or CLC instruction should be executed after one or more instructions from the ADC or SBC instruction.

## MACHINE INSTRUCTIONS

| Symbol | Function | Details | Addressing mode | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IMP | | | IMM | | | A | | | BIT,A | | | ZP | | | BIT,ZP | | |
| | | | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # |
| ADC (Note 1) | When T=0<br>A←A+M+C<br><br>When T=1<br>M(X)←M(X)+M+C | Adds the carry, accumulator and memory contents. The results are entered into the accumulator.<br>Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing modes in the columns on the right, and the contents of the carry. The results are entered into the memory at the address indicated by index register X. | | | | 69 | 2 | 2 | | | | | | | 65 | 3 | 2 | | | |
| AND (Note 1) | When T=0<br>A←A∧M<br><br>When T=1<br>M(X)←M(X)∧M | "AND-s" the accumulator and memory contents. The results are entered into the accumulator.<br>"AND-s" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing modes in the columns on the right. The results are entered into the memory at the address indicated by index register X. | | | | 29 | 2 | 2 | | | | | | | 25 | 3 | 2 | | | |
| ASL | 7 0<br>C ← □ ←0 | 1-bit shifts the contents of accumulator or contents of memory to the left. "0" enters 0th bit of memory or accumulator and the contents of the 7th bit enter carry flag. | | | | | | | 0A | 2 | 1 | | | | 06 | 5 | 2 | | | |
| BBC (Note 4) | $A_b$ or $M_b$=0? | Branches when the contents of the bit specified in the accumulator or memory are "0". | | | | | | | | | | 13+2i | 4 | 2 | | | | 17+2i | 5 | 3 |
| BBS (Note 4) | $A_b$ or $M_b$=1? | Branches when the contents of the bit specified in the accumulator or memory are "1". | | | | | | | | | | 03+2i | 4 | 2 | | | | 07+2i | 5 | 3 |
| BCC (Note 4) | C=0? | Branches when the contents of carry flag are "0". | | | | | | | | | | | | | | | | | | |
| BCS (Note 4) | C=1? | Branches when the contents of carry flag are "1". | | | | | | | | | | | | | | | | | | |
| BEQ (Note 4) | Z=1? | Branches when the contents of zero flag are "1". | | | | | | | | | | | | | | | | | | |
| BIT | A∧M | "AND-s" the contents of accumulator and memory. The results are not entered anywhere. | | | | | | | | | | | | | 24 | 3 | 2 | | | |
| BMI (Note 4) | N=1? | Branches when the contents of negative flag are "1". | | | | | | | | | | | | | | | | | | |
| BNE (Note 4) | Z=0? | Branches when the contents of zero flag are "0". | | | | | | | | | | | | | | | | | | |
| BPL (Note 4) | N=0? | Branches when the contents of negative flag are "0". | | | | | | | | | | | | | | | | | | |
| BRA | PC←PC±offset | Jumps to address where offset has been added to the program counter. | | | | | | | | | | | | | | | | | | |
| BRK | B←1<br>$M(S)←PC_H$<br>S←S−1<br>$M(S)←PC_L$<br>S←S−1<br>M(S)←PS<br>S←S−1<br>$PC_L←AD_L$<br>$PC_H←AD_H$ | Executes software interrupt. | 00 | 7 | 1 | | | | | | | | | | | | | | | |

| Addressing mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | Processor status register | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZP,X | | | ZP,Y | | | ABS | | | ABS,X | | | ABS,Y | | | IND | | | ZP,IND | | | IND,X | | | IND,Y | | | REL | | | SP | | | 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 | # | N | V | T | B | D | I | Z | C |
| 75 | 4 | 2 | | | | 6D | 4 | 3 | 7D | 5 | 3 | 79 | 5 | 3 | | | | | | | 61 | 6 | 2 | 71 | 6 | 2 | | | | | | | N | V | • | • | • | • | Z | C |
| 35 | 4 | 2 | | | | 2D | 4 | 3 | 3D | 5 | 3 | 39 | 5 | 3 | | | | | | | 21 | 6 | 2 | 31 | 6 | 2 | | | | | | | N | • | • | • | • | • | Z | • |
| 16 | 6 | 2 | | | | 0E | 6 | 3 | 1E | 7 | 3 | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 90 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | B0 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | F0 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | 2C | 4 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | $M_7$ | $M_6$ | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 30 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | D0 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 10 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 80 | 4 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | 1 | • | 1 | • | • |

3

| Symbol | Function | Details | Addressing mode | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IMP | | | IMM | | | A | | | BIT,A | | | ZP | | | BIT,ZP | | |
| | | | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # |
| BVC (Note 4) | V=0? | Branches when the contents of overflow flag are "0." | | | | | | | | | | | | | | | | | | |
| BVS (Note 4) | V=1? | Branches when the contents of overflow flag are "1." | | | | | | | | | | | | | | | | | | |
| CLB | $A_b$ or $M_b$←0 | Clears the contents of the bit specified in the accumulator or memory to "0." | | | | | | | | | | 1B+2i | 2 | 1 | | | | 1F+2i | 5 | 2 |
| CLC | C←0 | Clears the contents of the carry flag to "0." | 18 | 2 | 1 | | | | | | | | | | | | | | | |
| CLD | D←0 | Clears the contents of decimal mode flag to "0." | D8 | 2 | 1 | | | | | | | | | | | | | | | |
| CLI | I←0 | Clears the contents of interrupt disable flag to "0." | 58 | 2 | 1 | | | | | | | | | | | | | | | |
| CLT | T←0 | Clears the contents of X-modified arithmetic mode flag to "0." | 12 | 2 | 1 | | | | | | | | | | | | | | | |
| CLV | V←0 | Clears the contents overflow flag to "0." | B8 | 2 | 1 | | | | | | | | | | | | | | | |
| CMP (Note 3) | When T=0<br>A−M<br>When T=1<br>M(X)−M | Compares the contents of accumulator and memory.<br>Compares the contents of the memory specified by addressing modes in the columns on the right with the contents of the address indicated by index register X. | | | | C9 | 2 | 2 | | | | | | | C5 | 3 | 2 | | | |
| COM | M←$\overline{M}$ | Formes one's complement of contents of memory, and store it into memory. | | | | | | | | | | | | | 44 | 5 | 2 | | | |
| CPX | X−M | Compares the contents of index register X and memory. | | | | E0 | 2 | 2 | | | | | | | E4 | 3 | 2 | | | |
| CPY | Y−M | Compares the contents of index register Y and memory. | | | | C0 | 2 | 2 | | | | | | | C4 | 3 | 2 | | | |
| DEC | A←A−1 or<br>M←M−1 | Decrements the contents of accumulator or memory by 1. | | | | | | | 1A | 2 | 1 | | | | C6 | 5 | 2 | | | |
| DEX | X←X−1 | Decrements the contents of index register X by 1. | CA | 2 | 1 | | | | | | | | | | | | | | | |
| DEY | Y←Y−1 | Decrements the contents of index register Y by 1. | 88 | 2 | 1 | | | | | | | | | | | | | | | |
| EOR (Note 1) | When T=0<br>A←A∀M<br><br>When T=1<br>M(X)←M(X)∀M | "Exclusive-ORs" the contents of accumulator and memory. The results are stored into the accumulator.<br>"Exclusive-ORs" the contents of the memory specified by the addressing modes in the columns on the right and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X. | | | | 49 | 2 | 2 | | | | | | | 45 | 3 | 2 | | | |
| FST | | Connects oscillator output to $X_{OUTF}$. | E2 | 2 | 1 | | | | | | | | | | | | | | | |
| INC | A←A+1 or<br>M←M+1 | Increments the contents of accumulator or memory by 1. | | | | | | | 3A | 2 | 1 | | | | E6 | 5 | 2 | | | |
| INX | X←X+1 | Increments the contents of index register X by 1. | E8 | 2 | 1 | | | | | | | | | | | | | | | |
| INY | Y←Y+1 | Increments the contents of index register Y by 1. | C8 | 2 | 1 | | | | | | | | | | | | | | | |

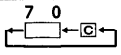
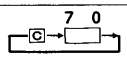
| Addressing mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | Processor status register | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZP,X | | | ZP,Y | | | ABS | | | ABS,X | | | ABS,Y | | | IND | | | ZP,IND | | | IND,X | | | IND,Y | | | REL | | | SP | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | N | V | T | B | D | I | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 50 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | 70 | 2 | 2 | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | 0 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | 0 | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | 0 | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | 0 | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | 0 | • | • | • | • | • | • |
| D5 | 4 | 2 | | | | CD | 4 | 3 | DD | 5 | 3 | D9 | 5 | 3 | | | | | | | C1 | 6 | 2 | D1 | 6 | 2 | | | | | | | N | • | • | • | • | • | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | EC | 4 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | C |
| | | | | | | CC | 4 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | C |
| D6 | 6 | 2 | | | | CE | 6 | 3 | DE | 7 | 3 | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| 55 | 4 | 2 | | | | 4D | 4 | 3 | 5D | 5 | 3 | 59 | 5 | 3 | | | | | | | 41 | 6 | 2 | 51 | 6 | 2 | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| F6 | 6 | 2 | | | | EE | 6 | 3 | FE | 7 | 3 | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |

3

| Symbol | Function | Details | Addressing mode | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IMP | | | IMM | | | A | | | BIT,A | | | ZP | | | BIT,ZP | | |
| | | | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # |
| JMP | If addressing mode is ABS<br>$PC_L \leftarrow AD_L$<br>$PC_H \leftarrow AD_H$<br>If addressing mode is IND<br>$PC_L \leftarrow (AD_H, AD_L)$<br>$PC_H \leftarrow (AD_H, AD_L+1)$<br>If addressing mode is ZP, IND<br>$PC_L \leftarrow (00, AD_L)$<br>$PC_H \leftarrow (00, AD_L+1)$ | Jumps to new address. | | | | | | | | | | | | | | | | | | |
| JSR | $M(S) \leftarrow PC_H$<br>$S \leftarrow S-1$<br>$M(S) \leftarrow PC_L$<br>$S \leftarrow S-1$<br>After executing the above,<br>If addressing mode is ABS,<br>$PC_L \leftarrow AD_L$<br>$PC_H \leftarrow AD_H$<br>If addressing mode is SP,<br>$PC_L \leftarrow AD_L$<br>$PC_H \leftarrow FF$<br>If addressing mode is ZP, IND,<br>$PC_L \leftarrow (00, AD_L)$<br>$PC_H \leftarrow (00, AD_L+1)$ | After storing contents of program counter in stack, and jumps to new address. | | | | | | | | | | | | | | | | | | |
| LDA<br>(Note 2) | When T=0<br>A←M<br>When T=1<br>M(X)←M | Load accumulator with contents of memory.<br><br>Load memory indicated by index register X with contents of memory specified by addressing mode shown in right column. | | | | A9 | 2 | 2 | | | | | | | A5 | 3 | 2 | | | |
| LDM | M←IMM | Load memory with immediate value. | | | | | | | | | | | | | 3C | 4 | 3 | | | |
| LDX | X←M | Load index register X with contents of memory. | | | | A2 | 2 | 2 | | | | | | | A6 | 3 | 2 | | | |
| LDY | Y←M | Load index register Y with contents of memory. | | | | A0 | 2 | 2 | | | | | | | A4 | 3 | 2 | | | |
| LSR | 7 0<br>0 → □ → C | Shift the contents of accumulator or memory to the right by one bit.<br>0th bit of accumulator or memory is stored in carry, 7th bit is cleared. | | | | | | | 4A | 2 | 1 | | | | 46 | 5 | 2 | | | |
| NOP | PC←PC+1 | No operation. | EA | 2 | 1 | | | | | | | | | | | | | | | |
| ORA<br>(Note 1) | When T=0<br>A←AVM<br><br>When T=1<br>M(X)←M(X)VM | Produce the logical OR of the contents of memory and accumulator. The result is stored in accumulator.<br>produce the logical OR of contents of memory indicated by index register X and contents of memory specified by addressing mode shown in right column. The result is stored in memory of address specified by index register X. | | | | 09 | 2 | 2 | | | | | | | 05 | 3 | 2 | | | |

| Addressing mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Processor status register | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZP,X | | | ZP,Y | | | ABS | | | ABS,X | | | ABS,Y | | | IND | | | ZP,IND | | | IND,X | | | IND,Y | | | REL | | | SP | | | 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 | # | N | V | T | B | D | I | Z | C |
| | | | | | | 4C | 3 | 3 | | | | | | | 6C | 5 | 3 | B2 | 4 | 2 | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | 20 | 6 | 3 | | | | | | | | | | 02 | 7 | 2 | | | | | | | | | | 22 | 5 | 2 | • | • | • | • | • | • | • | • |
| B5 | 4 | 2 | | | | AD | 4 | 3 | BD | 5 | 3 | B9 | 5 | 3 | | | | | | | A1 | 6 | 2 | B1 | 6 | 2 | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | B6 | 4 | 2 | AE | 4 | 3 | | | | BE | 5 | 3 | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| B4 | 4 | 2 | | | | AC | 4 | 3 | BC | 5 | 3 | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| 56 | 6 | 2 | | | | 4E | 6 | 3 | 5E | 7 | 3 | | | | | | | | | | | | | | | | | | | | | | 0 | • | • | • | • | • | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| 15 | 4 | 2 | | | | 0D | 4 | 3 | 1D | 5 | 3 | 19 | 5 | 3 | | | | | | | 01 | 6 | 2 | 11 | 6 | 2 | | | | | | | N | • | • | • | • | • | Z | • |

3

| Symbol | Function | Details | Addressing mode | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IMP | | | IMM | | | A | | | BIT,A | | | ZP | | | BIT,ZP | | |
| | | | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # |
| PHA | M(S)←A<br>S←S−1 | Saves the contents of the accumulator in the memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1. | 48 | 3 | 1 | | | | | | | | | | | | | | | |
| PHP | M(S)←PS<br>S←S−1 | Saves the contents of processor status register in the memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1. | 08 | 3 | 1 | | | | | | | | | | | | | | | |
| PLA | S←S+1<br>A←M(S) | Increments the contents of stack pointer by 1 and pulls from the memory at the address indicated by the stack pointer, and store it in accumulator. | 68 | 4 | 1 | | | | | | | | | | | | | | | |
| PLP | S←S+1<br>PS←M(S) | Increments the contents of stack pointer by 1 and pulls from the memory at the address indicated by the stack pointer, and store it in processor status register. | 28 | 4 | 1 | | | | | | | | | | | | | | | |
| ROL |  | Connects the carry flag and the accumulator or memory and rotates the contents to the left by 1 bit. | | | | | | | 2A | 2 | 1 | | | | 26 | 5 | 2 | | | |
| ROR |  | Connects the carry flag and the accumulator or memory and rotates the contents to the right by 1 bit. | | | | | | | 6A | 2 | 1 | | | | 66 | 5 | 2 | | | |
| RRF |  | Rotates the contents of memory to the right by 4 bits. | | | | | | | | | | | | | 82 | 8 | 2 | | | |
| RTI | S←S+1<br>PS←M(S)<br>S←S+1<br>$PC_L$←M(S)<br>S←S+1<br>$PC_H$←M(S) | Returns from the interrupt routine to the main routine. | 40 | 6 | 1 | | | | | | | | | | | | | | | |
| RTS | S←S+1<br>$PC_L$←M(S)<br>S←S+1<br>$PC_H$←M(S) | Returns from the subroutine to the main routine. | 60 | 6 | 1 | | | | | | | | | | | | | | | |
| SBC<br>(Note 1) | When T=0<br>A←A−M−C<br><br>When T=1<br>M(X)←M(X)−M−C | Subtracts the contents of memory and carry flag from the contents of accumulator. The results are stored into the accumulator.<br>Subtracts contents of carry flag and contents of the memory indicated by the addressing modes shown in the columns on the right from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X. | | | | E9 | 2 | 2 | | | | | | | E5 | 3 | 2 | | | |
| SEB | $A_b$ or $M_b$←1 | Sets the specified bit contents of accumulator or memory to "1." | | | | | | | | | | 0B+2i | 2 | 1 | | | | 0F+2i | 5 | 2 |
| SEC | C←1 | Sets the contents of carry flag to "1." | 38 | 2 | 1 | | | | | | | | | | | | | | | |
| SED | D←1 | Sets the contents of decimal mode flag to "1." | F8 | 2 | 1 | | | | | | | | | | | | | | | |
| SEI | I←1 | Sets the contents of interrupt disable flag to "1." | 78 | 2 | 1 | | | | | | | | | | | | | | | |
| SET | T←1 | Sets the contents of X-modified arithmetic mode flag to "1." | 32 | 2 | 1 | | | | | | | | | | | | | | | |
| SLW | | Releases the connection between the oscillator output and pin $X_{OUTF}$. | C2 | 2 | 1 | | | | | | | | | | | | | | | |

| Addressing mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | Processor status register | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZP,X | | | ZP,Y | | | ABS | | | ABS,X | | | ABS,Y | | | IND | | | ZP,IND | | | IND,X | | | IND,Y | | | REL | | | SP | | | 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 | # | N | V | T | B | D | I | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (Value saved in stack) | | | | | | | |
| 36 | 6 | 2 | | | | 2E | 6 | 3 | 3E | 7 | 3 | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | C |
| 76 | 6 | 2 | | | | 6E | 6 | 3 | 7E | 7 | 3 | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (Value saved in stack) | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| F5 | 4 | 2 | | | | ED | 4 | 3 | FD | 5 | 3 | F9 | 5 | 3 | | | | | | | E1 | 6 | 2 | F1 | 6 | 2 | | | | | | | N | V | • | • | • | • | Z | C |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | 1 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | 1 | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | 1 | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | 1 | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |

3

| Symbol | Function | Details | Addressing mode | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IMP | | | IMM | | | A | | | BIT,A | | | ZP | | | BIT,ZP | | |
| | | | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # | OP | n | # |
| STA | M←A | Stores the contents of accumulator in the memory. | | | | | | | | | | | | | 85 | 4 | 2 | | | |
| STP | | Stops the oscillation of the oscillator. | 42 | 2 | 1 | | | | | | | | | | | | | | | |
| STX | M←X | Stores the contents of index register X in the memory. | | | | | | | | | | | | | 86 | 4 | 2 | | | |
| STY | M←Y | Stores the contents of index register Y in the memory. | | | | | | | | | | | | | 84 | 4 | 2 | | | |
| TAX | X←A | Transfers the contents of accumulator to index register X. | AA | 2 | 1 | | | | | | | | | | | | | | | |
| TAY | Y←A | Transfers the contents of accumulator to index register Y. | A8 | 2 | 1 | | | | | | | | | | | | | | | |
| TST | M=0? | Tests whether the contents of memory are "0" or not. | | | | | | | | | | | | | 64 | 3 | 2 | | | |
| TSX | X←S | Transfers the contents of stack pointer to index register X. | BA | 2 | 1 | | | | | | | | | | | | | | | |
| TXA | A←X | Transfers the contents of index register X to the accumulator. | 8A | 2 | 1 | | | | | | | | | | | | | | | |
| TXS | S←X | Transfers the contents of index register X to the stack pointer. | 9A | 2 | 1 | | | | | | | | | | | | | | | |
| TYA | A←Y | Transfers the contents of index register Y to the accumulator. | 98 | 2 | 1 | | | | | | | | | | | | | | | |

Note 1 : The number of cycles "n" is added by 3 when T is 1.
2 : The number of cycles "n" is added by 2 when T is 1.
3 : The number of cycles "n" is added by 1 when T is 1.
4 : The number of cycles "n" is added by 2 when branching has occurred.

| Symbol | Contents | Symbol | Contents |
|---|---|---|---|
| IMP | Implied addressing mode | + | Addition |
| IMM | Immediate addressing mode | − | Subtraction |
| A | Accumulator or accumulator addressing mode | ∧ | AND |
| | | ∨ | OR |
| BIT, A | Accumulator bit relative addressing mode | ∀ | Exclusive-OR |
| | | — | Negation |
| ZP | Zero page addressing mode | ← | Shows direction of data flow |
| BIT, ZP | Zero page bit relative addressing mode | X | Index register X |
| | | Y | Index register Y |
| ZP, X | Zero page X addressing mode | S | Stack pointer |
| ZP, Y | Zero page Y addressing mode | PC | Program counter |
| ABS | Absolute addressing mode | PS | Processor status register |
| ABS, X | Absolute X addressing mode | $PC_H$ | 8 high-order bits of program counter |
| ABS, Y | Absolute Y addressing mode | $PC_L$ | 8 low-order bits of program counter |
| IND | Indirect absolute addressing mode | $AD_H$ | 8 high-order bits of address |
| | | $AD_L$ | 8 low-order bits of address |
| ZP, IND | Zero page indirect absolute addressing mode | $(AD_H, AD_L)$ | Contents of memory at address indicated by $AD_H$ and $AD_L$, in $AD_H$ is 8 high-order bits and $AD_L$ is low-order bits. |
| IND, X | Indirect X addressing mode | $(00, AD_L)$ | Contents of address indicated by zero page $AD_L$ |
| IND, Y | Indirect Y addressing mode | FF | FF in Hexadecimal notation |
| REL | Relative addressing mode | M | Memory specified by address designation of any addressing mode |
| SP | Special page addressing mode | | |
| C | Carry flag | M (X) | Memory of address indicated by contents of index register X |
| Z | Zero flag | | |
| I | Interrupt disable flag | M (S) | Memory of address indicated by contents of stack pointer |
| D | Decimal mode flag | $A_b$ | 1 bit of accumulator |
| B | Break flag | $M_b$ | 1 bit of memory |
| T | X-modified arithmetic mode flag | OP | Opcode |
| V | Overflow flag | n | Number of cycles |
| N | Negative flag | # | Number of bytes |

| Addressing mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Processor status register | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZP,X | | | ZP,Y | | | ABS | | | ABS,X | | | ABS,Y | | | IND | | | ZP,IND | | | IND,X | | | IND,Y | | | REL | | | SP | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | 0P | n | # | N | V | T | B | D | I | Z | C |
| 95 | 5 | 2 | | | | 8D | 5 | 3 | 9D | 6 | 3 | 99 | 6 | 3 | | | | | | | 81 | 7 | 2 | 91 | 7 | 2 | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | 96 | 5 | 2 | 8E | 5 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| 94 | 5 | 2 | | | | 8C | 5 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | • | • | • | • | • | • | • | • |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | N | • | • | • | • | • | Z | • |

3

## LIST OF INSTRUCTION CODES

| $D_7$~$D_4$ \ $D_3$~$D_0$ (Hexadecimal notation) | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000 | 0 | BRK | ORA IND, X | JSR ZP, IND | BBS 0, A | — | ORA ZP | ASL ZP | BBS 0, ZP | PHP | ORA IMM | ASL A | SEB 0, A | — | ORA ABS | ASL ABS | SEB 0, ZP |
| 0001 | 1 | BPL | ORA IND, Y | CLT | BBC 0, A | — | ORA ZP, X | ASL ZP, X | BBC 0, ZP | CLC | ORA ABS, Y | DEC A | CLB 0, A | — | ORA ABS, X | ASL ABS, X | CLB 0, ZP |
| 0010 | 2 | JSR ABS | AND IND, X | JSR SP | BBS 1, A | BIT ZP | AND ZP | ROL ZP | BBS 1, ZP | PLP | AND IMM | ROL A | SEB 1, A | BIT ABS | AND ABS | ROL ABS | SEB 1, ZP |
| 0011 | 3 | BMI | AND IND, Y | SET | BBC 1, A | — | AND ZP, X | ROL ZP, X | BBC 1, ZP | SEC | AND ABS, Y | INC A | CLB 1, A | LDM ZP | AND ABS, X | ROL ABS, X | CLB 1, ZP |
| 0100 | 4 | RTI | EOR IND, X | STP | BBS 2, A | COM ZP | EOR ZP | LSR ZP | BBS 2, ZP | PHA | EOR IMM | LSR A | SEB 2, A | JMP ABS | EOR ABS | LSR ABS | SEB 2, ZP |
| 0101 | 5 | BVC | EOR IND, Y | — | BBC 2, A | — | EOR ZP, X | LSR ZP, X | BBC 2, ZP | CLI | EOR ABS, Y | — | CLB 2, A | — | EOR ABS, X | LSR ABS, X | CLB 2, ZP |
| 0110 | 6 | RTS | ADC IND, X | — | BBS 3, A | TST ZP | ADC ZP | ROR ZP | BBS 3, ZP | PLA | ADC IMM | ROR A | SEB 3, A | JMP IND | ADC ABS | ROR ABS | SEB 3, ZP |
| 0111 | 7 | BVS | ADC IND, Y | — | BBC 3, A | — | ADC ZP, X | ROR ZP, X | BBC 3, ZP | SEI | ADC ABS, Y | — | CLB 3, A | — | ADC ABS, X | ROR ABS, X | CLB 3, ZP |
| 1000 | 8 | BRA | STA IND, X | RRF ZP | BBS 4, A | STY ZP | STA ZP | STX ZP | BBS 4, ZP | DEY | — | TXA | SEB 4, A | STY ABS | STA ABS | STX ABS | SEB 4, ZP |
| 1001 | 9 | BCC | STA IND, Y | — | BBC 4, A | STY ZP, X | STA ZP, X | STX ZP, Y | BBC 4, ZP | TYA | STA ABS, Y | TXS | CLB 4, A | — | STA ABS, X | — | CLB 4, ZP |
| 1010 | A | LDY IMM | LDA IND, X | LDX IMM | BBS 5, A | LDY ZP | LDA ZP | LDX ZP | BBS 5, ZP | TAY | LDA IMM | TAX | SEB 5, A | LDY ABS | LDA ABS | LDX ABS | SEB 5, ZP |
| 1011 | B | BCS | LDA IND, Y | JMP ZP, IND | BBC 5, A | LDY ZP, X | LDA ZP, X | LDX ZP, Y | BBC 5, ZP | CLV | LDA ABS, Y | TSX | CLB 5, A | LDY ABS, X | LDA ABS, X | LDX ABS, Y | CLB 5, ZP |
| 1100 | C | CPY IMM | CMP IND, X | SLW | BBS 6, A | CPY ZP | CMP ZP | DEC ZP | BBS 6, ZP | INY | CMP IMM | DEX | SEB 6, A | CPY ABS | CMP ABS | DEC ABS | SEB 6, ZP |
| 1101 | D | BNE | CMP IND, Y | — | BBC 6, A | — | CMP ZP, X | DEC ZP, X | BBC 6, ZP | CLD | CMP ABS, Y | — | CLB 6, A | — | CMP ABS, X | DEC ABS, X | CLB 6, ZP |
| 1110 | E | CPX IMM | SBC IND, X | FST | BBS 7, A | CPX ZP | SBC ZP | INC ZP | BBS 7, ZP | INX | SBC IMM | NOP | SEB 7, A | CPX ABS | SBC ABS | INC ABS | SEB 7, ZP |
| 1111 | F | BEQ | SBC IND, Y | — | BBC 7, A | — | SBC ZP, X | INC ZP, X | BBC 7, ZP | SED | SBC ABS, Y | — | CLB 7, A | — | SBC ABS, X | INC ABS, X | CLB 7, ZP |

3-byte instruction

2-byte instruction

## ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Conditions | Limits | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | With respect to VSS; output transistors cut-off | −0.3~7 | V |
| $V_I$ | Input voltage, $R_0$~$R_3$, $CNV_{SS}$, $\overline{RESET}$, $X_{IN}$ | | −0.3~7 | V |
| $V_I$ | Input voltage, $P3_0$~$P3_7$ | | −3.0~$V_{CC}$+0.3 | V |
| $V_I$ | Input voltage, $\overline{INT}$ $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, CNTR | | −0.3~13 | V |
| $V_O$ | Output voltage, $R_0$~$R_3$ | | −0.3~7 | V |
| $V_O$ | Output voltage, $P3_0$~$P3_7$, $X_{OUTF}$, $X_{OUTS}$, $\phi$, R/W, $\overline{CE}$, $\overline{RESET_{OUT}}$ | | −0.3~$V_{CC}$+0.3 | V |
| $V_O$ | Output voltage, $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ CNTR | | −0.3~13 | V |
| $P_d$ | Power dissipation | $T_a$ = 25℃ | 1000 | mW |
| $T_{opr}$ | Operating temperature | | −10~70 | ℃ |
| $T_{stg}$ | Storage temperature | | −40~125 | ℃ |

3

## RECOMMENDED OPERATING CONDITIONS ($T_a$ = −10~70℃, $V_{CC}$ = 5V±10%, unless otherwise noted)

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| $V_{CC}$ | Supply voltage | 4.5 | 5 | 5.5 | V |
| $V_{SS}$ | Supply voltage | | 0 | | V |
| $V_{IH}$ | High-level input voltage, $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_7$, $R_0$~$R_3$, $CNV_{SS}$ | 0.8$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IH}$ | High-level input voltage, CNTR, $\overline{INT}$ | 0.8$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IH}$ | High-level input voltage, $\overline{RESET}$ | 0.48$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IH}$ | High-level input voltage, $X_{IN}$ | 0.8$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | Low-level input voltage, $P0_0$~$P0_7$, $P1_0$~$P1_7$ $P2_0$~$P2_7$, $P3_0$~$P3_7$, $R_0$~$R_3$, $CNV_{SS}$ | 0 | | 0.2$V_{CC}$ | V |
| $V_{IL}$ | Low-level input voltage, CNTR, $\overline{INT}$ | 0 | | 0.2$V_{CC}$ | V |
| $V_{IL}$ | Low-level input voltage, $\overline{RESET}$ | 0 | | 0.12$V_{CC}$ | V |
| $V_{IL}$ | Low-level input voltage, $X_{IN}$ | 0 | | 0.2$V_{CC}$ | V |
| $f_{(\phi)}$ | Internal clock oscillation frequency | | | 4 | MHz |

Note 1 : A high-level input voltage for ports P0, P1, P2, CNTR and $\overline{INT}$ of up to +12V may be supplied.

## ELECTRICAL CHARACTERISTICS ($V_{CC}$ = 5V±10%, $V_{SS}$ = 0V, $f_{(\phi)}$ = 4MHz, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| $V_{OH}$ | High-level output voltage, $P3_0$~$P3_7$ | $V_{CC}$ = 5V, $T_a$ = 25℃, $I_{OH}$ = −10mA | 3 | | | V |
| $V_{OH}$ | High-level output voltage, $\phi$, R/W, $\overline{CE}$, $\overline{RESET_{OUT}}$ | $V_{CC}$ = 5V, $T_a$ = 25℃, $I_{OH}$ = −2.5mA | 3 | | | V |
| $V_{OL}$ | Low-level output voltage, $P0_0$~ $P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $R_0$~$R_3$, CNTR | $V_{CC}$ = 5V, $T_a$ = 25℃, $I_{OL}$ = 10mA | | | 2 | V |
| $V_{OL}$ | Low-level output voltage, $\phi$, R/W, $\overline{CE}$, $\overline{RESET_{OUT}}$ | $V_{CC}$ = 5V, $T_a$ = 25℃, $I_{OL}$ = 5mA | | | 2 | V |
| $V_{T+}-V_{T-}$ | Hysteresis, CNTR, $\overline{INT}$ | $V_{CC}$ = 5V, $T_a$ = 25℃ | 0.3 | | 1 | V |
| $V_{T+}-V_{T-}$ | Hysteresis, $\overline{RESET}$ | $V_{CC}$ = 5V, $T_a$ = 25℃ | | 0.5 | 0.7 | V |
| $V_{T+}-V_{T-}$ | Hysteresis, $X_{IN}$ | $V_{CC}$ = 5V, $T_a$ = 25℃ | 0.1 | | 0.5 | V |
| $I_{IL}$ | Input leakage current, $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $\overline{INT}$, CNTR | $V_{CC}$ = 5V, $T_a$ = 25℃, 0 ≦ $V_I$ ≦ 12V | −12 | | 12 | μA |
| $I_{IL}$ | Input leakage current, $P3_0$~$P3_7$, $R_0$~$R_3$, $CNV_{SS}$, $\overline{RESET}$, $X_{IN}$ | $V_{CC}$ = 5V, $T_a$ = 25℃, 0 ≦ $V_I$ ≦ 5V | −5 | | 5 | μA |
| $I_{CC}$ | Supply current | $P3_0$~$P3_7$: $V_{CC}$, output pins open; $V_{SS}$ for all input and output pins except $P3_0$~$P3_7$; $V_{CC}$ = 5V, $T_a$ = 25℃ | | 3 | 6 | mA |

## TIMING REQUIREMENTS
## SINGLE CHIP MODE ($V_{CC}$ = 5V±10%, $V_{SS}$ = 0V, $T_a$ = 25℃, $f_{(\phi)}$ = 4MHz, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| $t_{su(P0D-\phi)}$ | Port P0 input setup time | | 270 | | | ns |
| $t_{su(P1D-\phi)}$ | Port P1 input setup time | | 270 | | | ns |
| $t_{su(P2D-\phi)}$ | Port P2 input setup time | | 270 | | | ns |
| $t_{su(P3D-\phi)}$ | Port P3 input setup time | | 270 | | | ns |
| $t_{su(RD-\phi)}$ | Port R input setup time | | 330 | | | ns |
| $t_{h(\phi-P0D)}$ | Port P0 input hold time | | 0 | | | ns |
| $t_{h(\phi-P1D)}$ | Port P1 input hold time | | 0 | | | ns |
| $t_{h(\phi-P2D)}$ | Port P2 input hold time | | 0 | | | ns |
| $t_{h(\phi-P3D)}$ | Port P3 input hold time | | 0 | | | ns |
| $t_{h(\phi-RD)}$ | Port R input hold time | | 0 | | | ns |
| $t_C$ | External clock input cycle time | | 250 | | | ns |
| $t_W$ | External clock input pulse width | | 75 | | | ns |
| $t_r$ | External clock rise time | | | | 25 | ns |
| $t_f$ | External clock fall time | | | | 25 | ns |

## SWITCHING CHARACTERISTICS
## SINGLE CHIP MODE ($V_{CC}$ = 5V±10%, $V_{SS}$ = 0V, $T_a$ = 25℃, $f_{(\phi)}$ = 4MHz, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| $t_{d(\phi-P0Q)}$ | Port P0 data output delay time | Fig.16 | | | 230 | ns |
| $t_{d(\phi-P1Q)}$ | Port P1 data output delay time | Fig.16 | | | 230 | ns |
| $t_{d(\phi-P2Q)}$ | Port P2 data output delay time | Fig.16 | | | 230 | ns |
| $t_{d(\phi-P3Q)}$ | Port P3 data output delay time | Fig.17 | | | 200 | ns |
| $t_{d(\phi-RA)}$ | Port R address output delay time | Fig.16 | | | 200 | ns |
| $t_{d(\phi-RAF)}$ | Port R address output delay time | Fig.16 | 0 | | 200 | ns |
| $t_{d(\phi-RQ)}$ | Port R data output delay time | Fig.16 | | | 200 | ns |
| $t_{d(\phi-RQF)}$ | Port R data output delay time | Fig.16 | | | 200 | ns |
| $t_{d(\phi-CE)}$ | $\overline{CE}$ output delay time | Fig.18 | | | 200 | ns |
| $t_{d(\phi-RW)}$ | R/W output delay time | Fig.18 | | | 100 | ns |

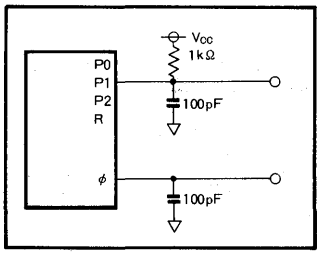



Fig.16 Port P0~P2, R test circuit

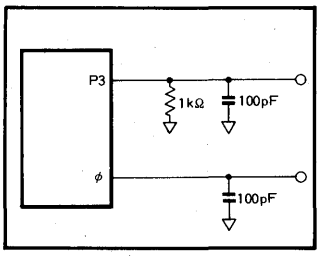



Fig.17 Port P3 test circuit

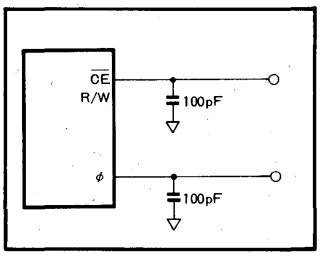
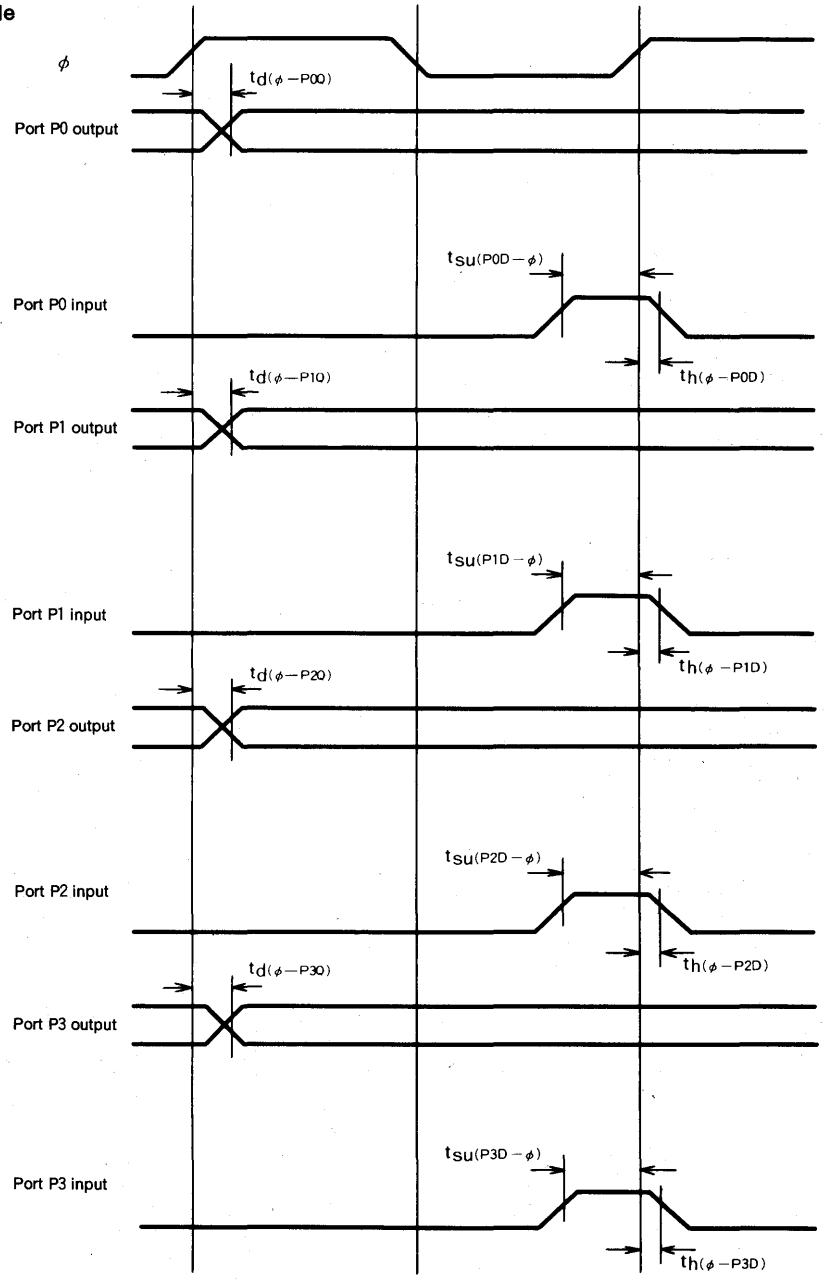



Fig.18 $\overline{CE}$, R/W test circuit

## TIMING DIAGRAMS

Single chip mode



Single chip mode (continued evaluation)