USER'S MANUAL

M37450M2-XXXSP/FP

MITSUBISHI SEMICONDUCTORS

M37450M2-XXXSP/FP

USER'S MANUAL

MITSUBISHI
ELECTRIC

We hereby introduce the manual for CMOS 8-bit microcomputer M37450M2-XXXSP/FP hardware. This manual is prepared for the users who should understand fully the functions and features of M37450M2-XXXSP/FP so that they can utilze this product to its fullest capacity. A detailed explanations of the specifications and applications regarding the hardware is hereby provided; for the software, consult MELPS 740 programming manual.

The contents of this user's manual are subject to change for the reasons of later improvement of the features.

The information, charts, and other data in this user's manual are correct and reliable; however, Mitsubishi Electric Corporation is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

MELPS and MELCS are the trademarks of Mitsubishi Electric Corporation.

CONTENTS

1. DESCRIPTION

1.1 Summary

M37450M2-XXXSP/FP is a single-chip microcomputer designed with CMOS sillcon gate technology. M37450M2-XXXSP is housed in a 64-pin shrink plastic molded DIP and M37450M2-XXXFP is housed in an 80-pin plastic molded QFP. In this single-chip microcomputer, ROM, RAM and I/O are placed on the same memory map in addition to simple instruction set. It is applicable to the office equipment and industrial controller. The use of CMOS process enables extremely low power consumption and is most suitable for battery drive. M37450M2-XXXSP/FP is equipped with 4K bytes of ROM and 128 bytes of RAM.

In addition, the following are the built-in functions:

- Master CPU bus interface function
- 16-bit multi-functional timer
- Serial I/O (Clock synchronous/UART is selectable by program)
- A-D converter (Successive approximation)
- D-A converter (R-2R type)
- PWM output
- High-speed multiplier/divider

In accordance with the built-in multiplier/divider, M37450M2-XXXSP/FP has multiply and divide instructions in addition to MELPS 740 basic instruction sets.

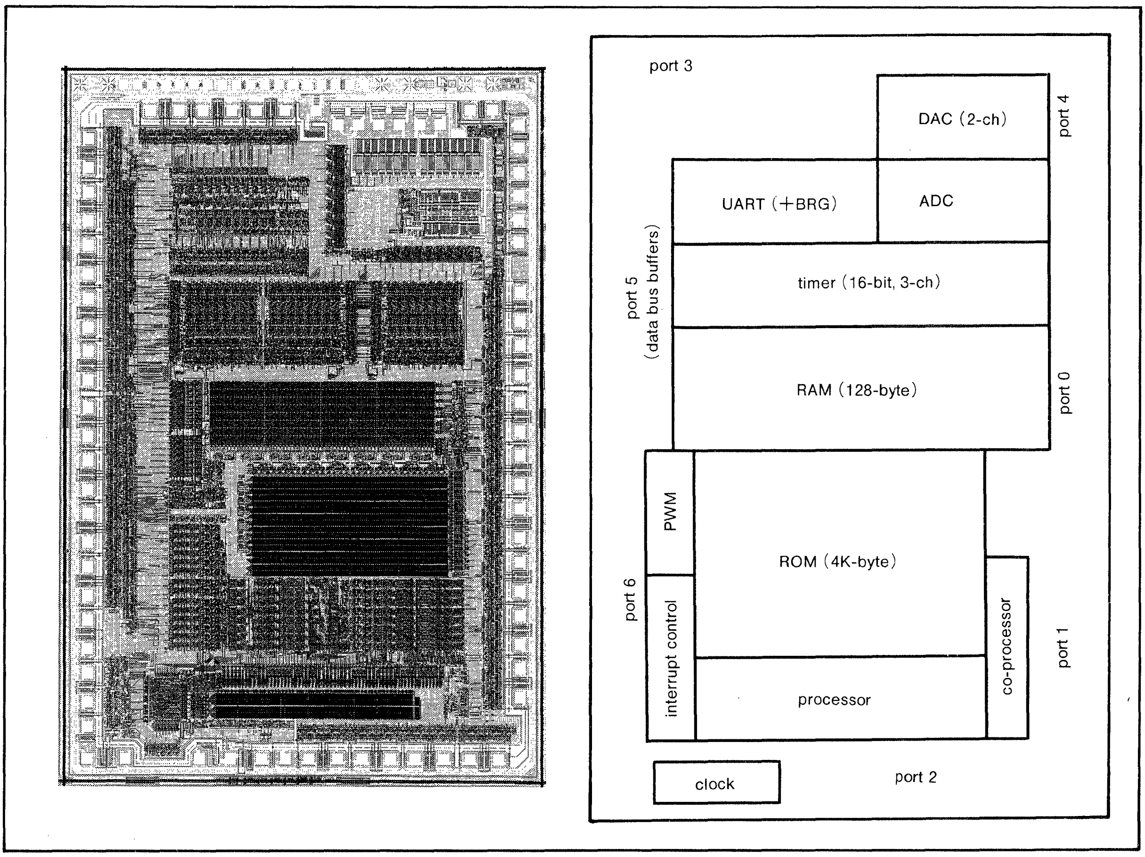


Fig. 1.1.1 M37450M2-XXXSP/FP chip floor plan

The difference between M37450M2-XXXSP and M37450M2-XXXFP lies only in the package (pin numbers) and port 4 (common with analog input port) ; and in whether $\overline{RD}$, $\overline{WR}$, RESET$_{OUT}$, AV$_{CC}$, ADV$_{REF}$, DAV$_{REF}$, and V$_{REF}$ pins exist or not. Therefore, in this user's manual, explanation applies only to M37450M2-XXXSP (abbreviated as M37450), unless noted otherwise. Also, when referring to the difference between M37450M2-XXXSP and M37450M2-XXXFP, 64-pin model (SP) and 80-pin model (FP) will be used to distinguish them.

Table 1.1.1 Functions of M37450M2-XXXSP/FP

Parameter			Functions
Number of basic instructions			71(69 MELPS 740 basic instructions+2)
Instruction execution time			0.8μs (minimum instructions, at 10MHz of frequency)
Clock frequency			10MHz (max.)
Memory size		ROM	4096 bytes
		RAM	128 bytes
Input/Output port	P0~P3, P5, P6	I/O	8-bit×6
	P4	Input	3-bit×1(8-bit×1 for 80-pin model)
	D-A	Output	2-bit×1
Serial I/O			UART or clock synchronous
Timers			16-bit timer×3, 8-bit timer (serial I/O baud rate generator)×1
A-D converter			8-bit×3 channels (8 channels for 80-pin model)
D-A converter			8-bit×2 channels
Pulse width modulator			8-bit or 16-bit×1
Data bus buffer			1-byte input and output each
Subroutine nesting			64-levels (max. for M37450M2)
			96-levels (max. for M37450M4, M37450M8)
Interrupts			6 external interrupts, 8 internal interrupts 1 software interrupt
Clock generating circuit			Built-in (ceramic or quarts crystal oscillator)
Supply voltage			5V±10%
Power dissipation			30mW (at 10MHz frequency)
Input/Output characteristics	Input/Output voltage		5V
	Output current		±5mA (max.)
Memory expansion			Possible
Operating temperature range			−10~70℃
Device structure			CMOS silicon gate
Package		M37450M2-XXXSP	64-pin shrink plastic molded DIP
		M37450M2-XXXFP	80-pin plastic molded QFP

1.2 M37450 family

M37450M2-XXXSP/FP as the base chip, the M37450 family will be developed as follows:

Table 1.2.1 M37450 family

Type	ROM	RAM	Additional
M37450M2-XXXSP/FP	4K bytes	128 bytes	
M37450M4-XXXSP/FP	8K bytes	256 bytes	M37450M2 ROM expansion
M37450M8-XXXSP/FP	16K bytes	384 bytes	M37450M2 ROM expansion
M37450S1SP/FP	—	128 bytes	M37450M2 external ROM
M37450S2SP/FP	—	256 bytes	M37450M4 external ROM
M37450S4SP/FP	—	448 bytes	M37450M8 external ROM
M37450E4-XXXSP/FP	8K bytes	256 bytes	M37450M4 one-time programable version
M37450E8-XXXSP/FP★★	16K bytes	384 bytes	M37450M8 one-time programable version
M37450E4SS/FS	8K bytes	256 bytes	M37450M4 EPROM version
M37450E8SS/FS★★	16K bytes	384 bytes	M37450M8 EPROM version
M37450PSS/FS	—	448 bytes	M37450 piggyback type

★★Under development

These are all pin compatible and developed in terms of memory size and characteristics so that the user can select the most suitable device according to the system.

This manual expains M37450M2-XXXSP/FP, but it is applicable to other microcomputers in the M37450 family as long as the user remembers the difference in memory.

1.3 Pin configuration

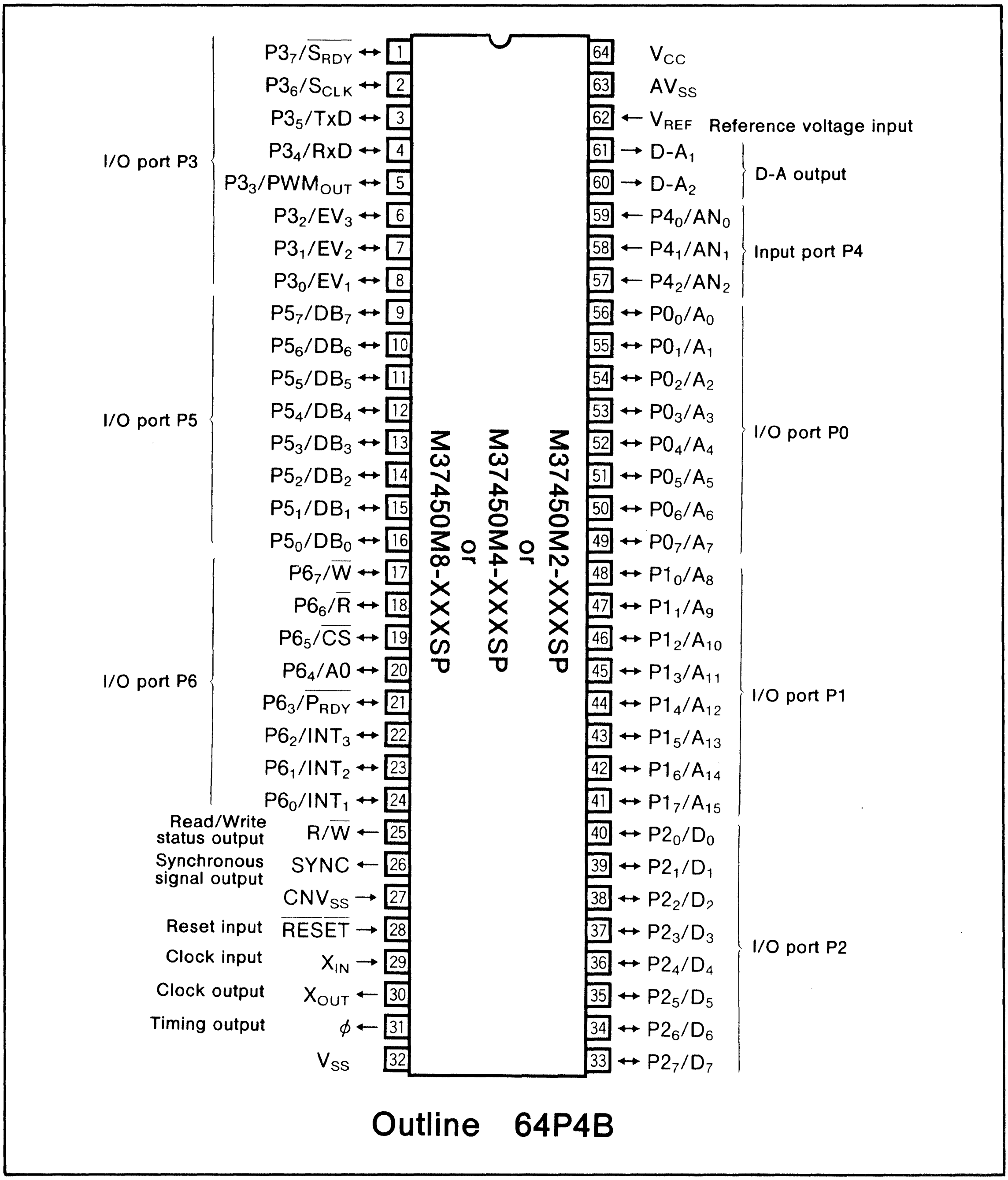
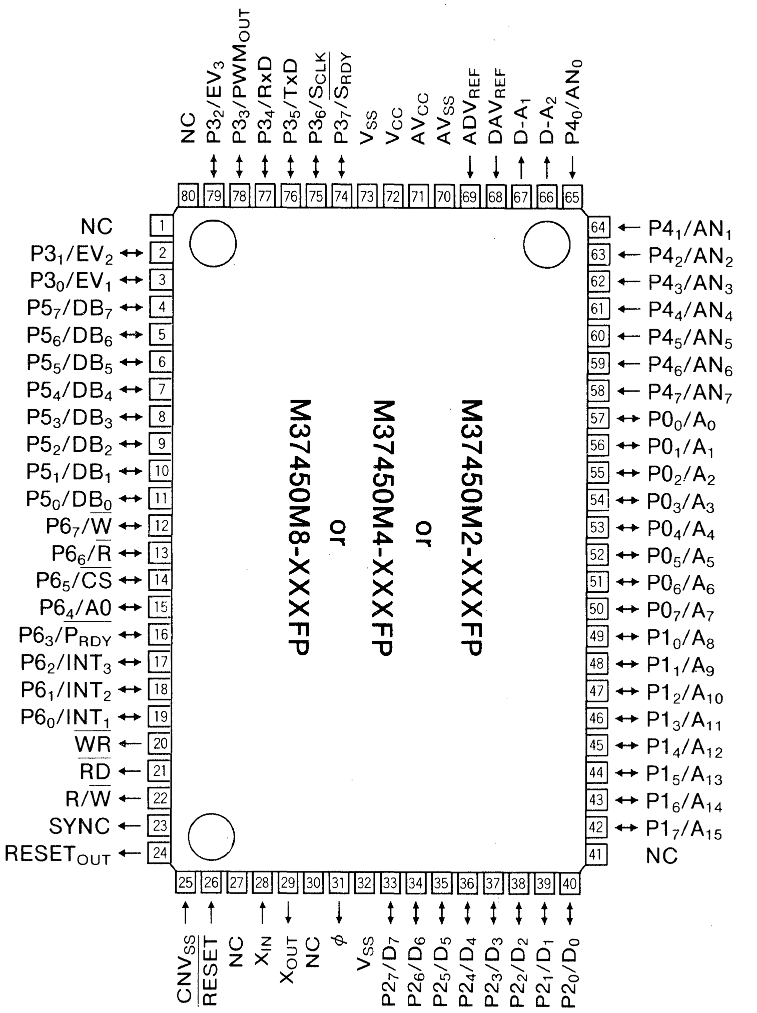


Fig 1.3.1 M37450M2-XXXSP, M37450M4-XXXSP, M37450M8-XXXSP pin configuration



Outline 80P6

NC : No connection

Fig 1.3.2 M37450M2-XXXFP, M37450M4-XXXFP, M37450M8-XXXFP pin configuration

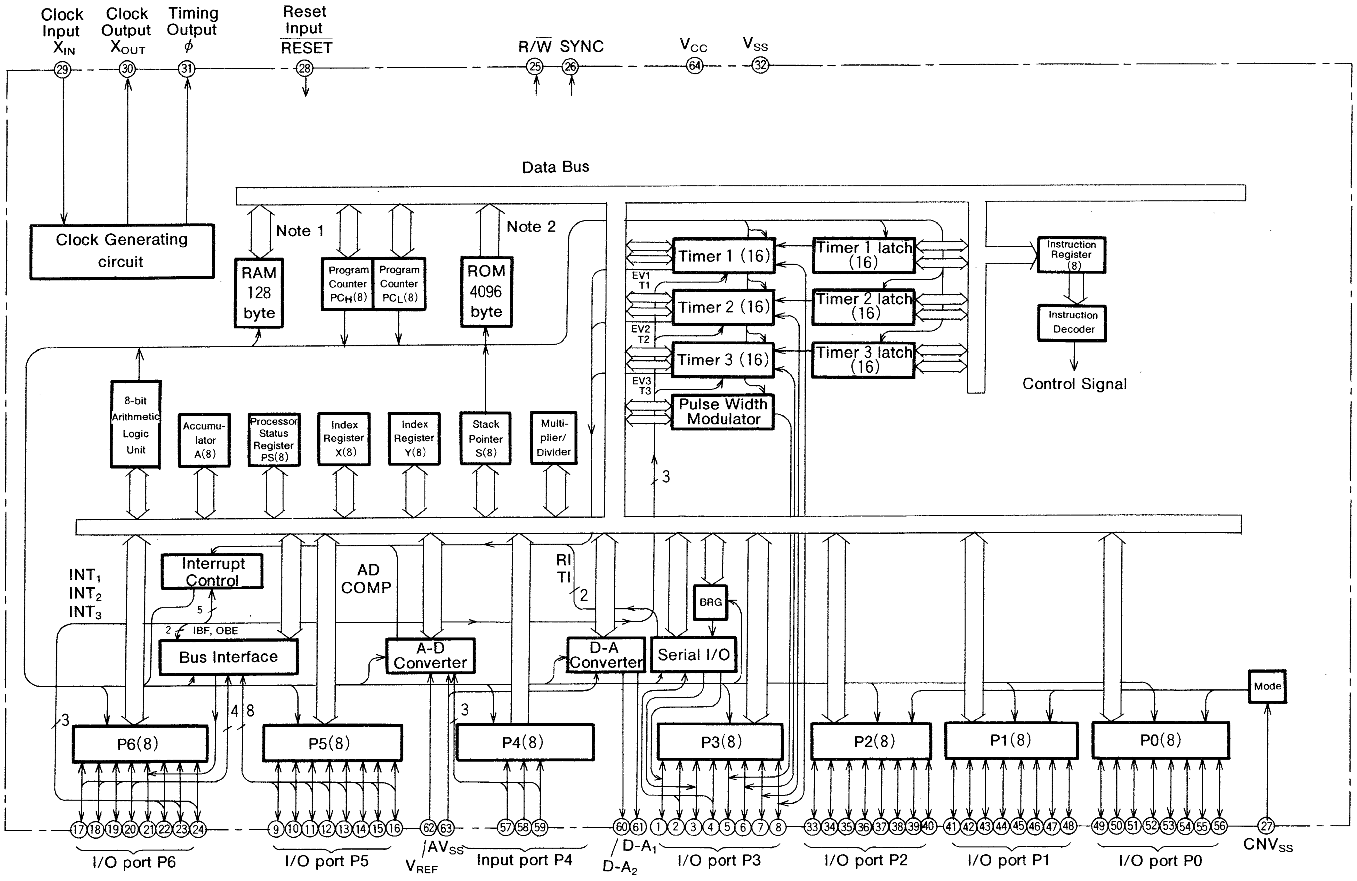
1.4 Pin description

Table 1.4.1 Pin description

Pin	Name	Input/ Output	Functions
V_{CC}, V_{SS}	Supply voltage		Power supply inputs 5V±10% to V_{CC}, and 0V to V_{SS}.
CNV_{SS}	CNV_{SS}		Controls the processor mode of the chip. Normally connected to V_{SS} or V_{CC}.
$\overline{\text{RESET}}$	Reset input	Input	To enter the reset state, the reset input pin must be kept at a "L" for more than 2μs (under normal V_{CC} conditions). If more time is needed for the crystal oscillator to stabilize, this "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	This chip has an internal clock generating circuit. To control generating frequency, an external ceramic or a quartz crystal oscillator is connected between the X_{IN} and X_{OUT} pins. If an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
ϕ	Timing output	Output	Outputs signal consisting of oscillating frequency divided by four.
SYNC	Synchronous signal output	Output	This signal is output "H" during operation code fetch and is used to control single stepping of programs.
R/$\overline{\text{W}}$	Read/Write status output	Output	This signal determines the direction of the data bus. It is "H" during read and "L" during write.
$P0_0$~$P0_7$	I/O port P0	I/O	Port P0 is an 8-bit I/O port with directional registers allowing each I/O bit to be individually programmed as input or output. The output structure is CMOS output. The low-order bits of the address are output except in single-chip mode.
$P1_0$~$P1_7$	I/O port P1	I/O	Port P1 is an 8-bit I/O port and has basically the same functions as port P0. The high-order bits of the address are output except in single-chip mode.
$P2_0$~$P2_7$	I/O port P2	I/O	Port P2 is an 8-bit I/O port and has basically the same functions as port P0. Used as data bus except in single-chip mode.
$P3_0$~$P3_7$	I/O port P3	I/O	Port P3 is an 8-bit I/O port and has basically the same functions as port P0. Serial I/O, PWM output, or event I/O function can be selected with a program.
$P4_0$~$P4_2$ ($P4_0$~$P4_7$)	Input port P4	Input	Analog input pin for the A-D converter. The 64-pin model has three pins and the 80-pin model has eight pins. They may also be used as digital input pins.
$P5_0$~$P5_7$	I/O port P5	I/O	Port P5 is an 8-bit I/O port and has basically the same functions as port P0. This port functions as an 8-bit data bus for the master CPU when slave mode is selected with a program.
$P6_0$~$P6_7$	I/O port P6	I/O	Port P6 is an 8-bit I/O port and has basically the same function as port P0. Pins $P6_3$~$P6_7$ change to a control bus for the master CPU when slave mode is selected with a program. Pins $P6_0$~$P6_2$ may be programmed as external interrupt input pins.
D-A_1, D-A_2	D-A output	Output	Analog signal from D-A converter is output.
V_{REF}	Reference voltage input	Input	Reference voltage input pin for A-D and D-A converter. This pin is for 64-pin model only.
ADV_{REF}	A-D reference voltage input	Input	Reference voltage input pin for A-D converter. This pin is for 80-pin model only.
DAV_{REF}	D-A reference voltage input	Input	Reference voltage input pin for D-A converter. This pin is for 80-pin model only.
AV_{SS}	Analog power supply		Ground level input pin for A-D and D-A converter. Same voltage as V_{SS} is applied.
AV_{CC}	Analog power supply		Power supply input pin for A-D converter. This pin is for 80-pin model only. Same voltage as V_{CC} is applied. In the case of the 64-pin model, AV_{CC} is connected to V_{CC} internally.
$\overline{\text{RD}}$	Read signal output	Output	Control signal output as active "L" when valid data is read from data bus. This pin is for 80-pin model only.
$\overline{\text{WR}}$	Write signal output	Output	Control signal output as active "L" when writing data from data bus to external component. This pin is for 80-pin model only.
$RESET_{OUT}$	Reset output	Output	Control signal output as active "H" during reset. It is used as a reset output signal for peripheral components. This pin is for 80-pin model only.

1.5 Block diagram

M37450M2-XXXSP

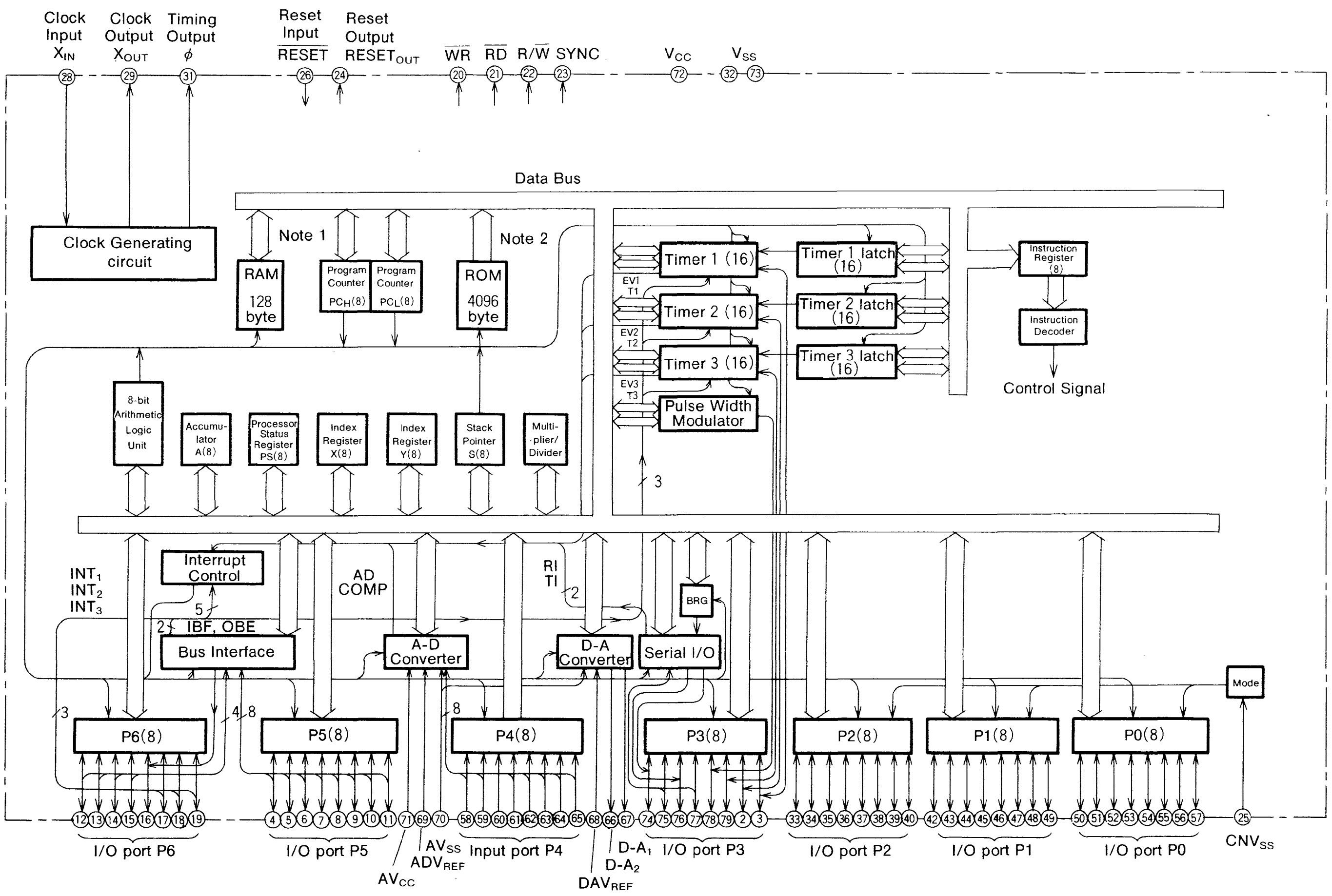


Note 1 : 256 bytes for M37450M2-XXXSP and 384 bytes for M37450M8-XXXSP.

2 : 8192 bytes for M37450M4-XXXSP and 16384 bytes for M37450M8-XXXSP.

Fig. 1.5.1 M37450M2-XXXSP block diagram

M37450M2-XXXFP



Note 1 : 256 bytes for M37450M4-XXXFP and 384 bytes for M37450M8-XXXFP.

2 : 8192 bytes for M37450M4-XXXFP and 16284 bytes for M37450M8-XXXFP.

Fig. 1.5.2 M37450M2-XXXFP block diagram

2. FUNCTIONS

2.1 Central processing unit (CPU)

6 registers are built-in the CPU of M37450. Accumulator (A), Index register X(X), Index register Y(Y), Stack pointer (S), Processor Status register (PS), all consisting 8-bit. Program Counter (PC) includes PC_H and PC_L, each of which consists of 8-bit registers.
Other than the I flag to be set "1" immediately, the content of these registers are undefined after reset. Figure 2.1.1 shows the structure of the registers for M37450.

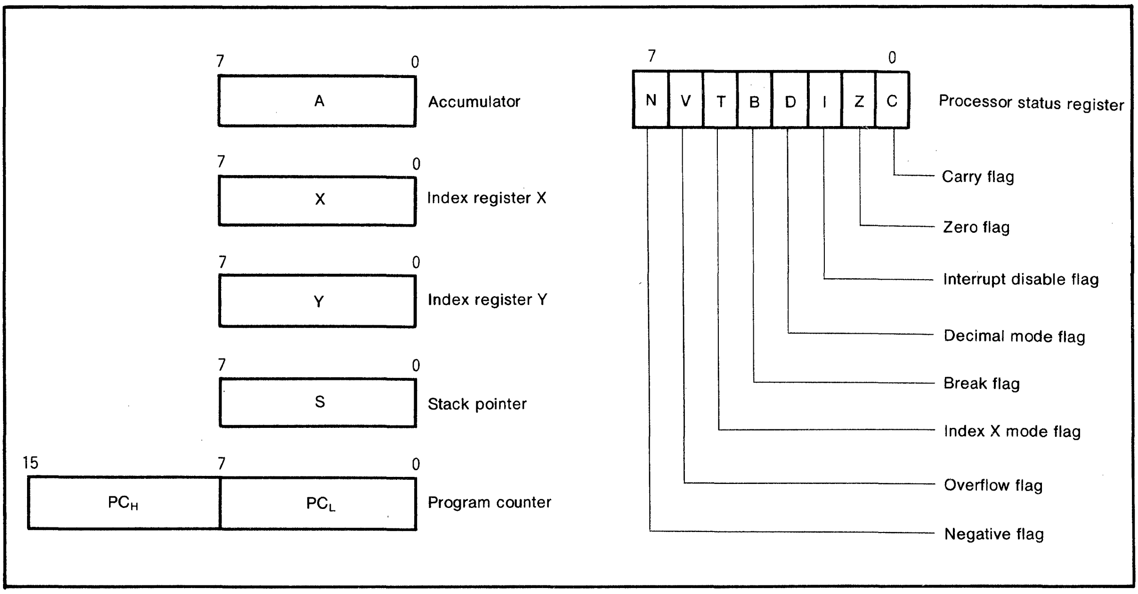


Fig. 2.1.1 Register structure

2.1.1 Accumulator (A)

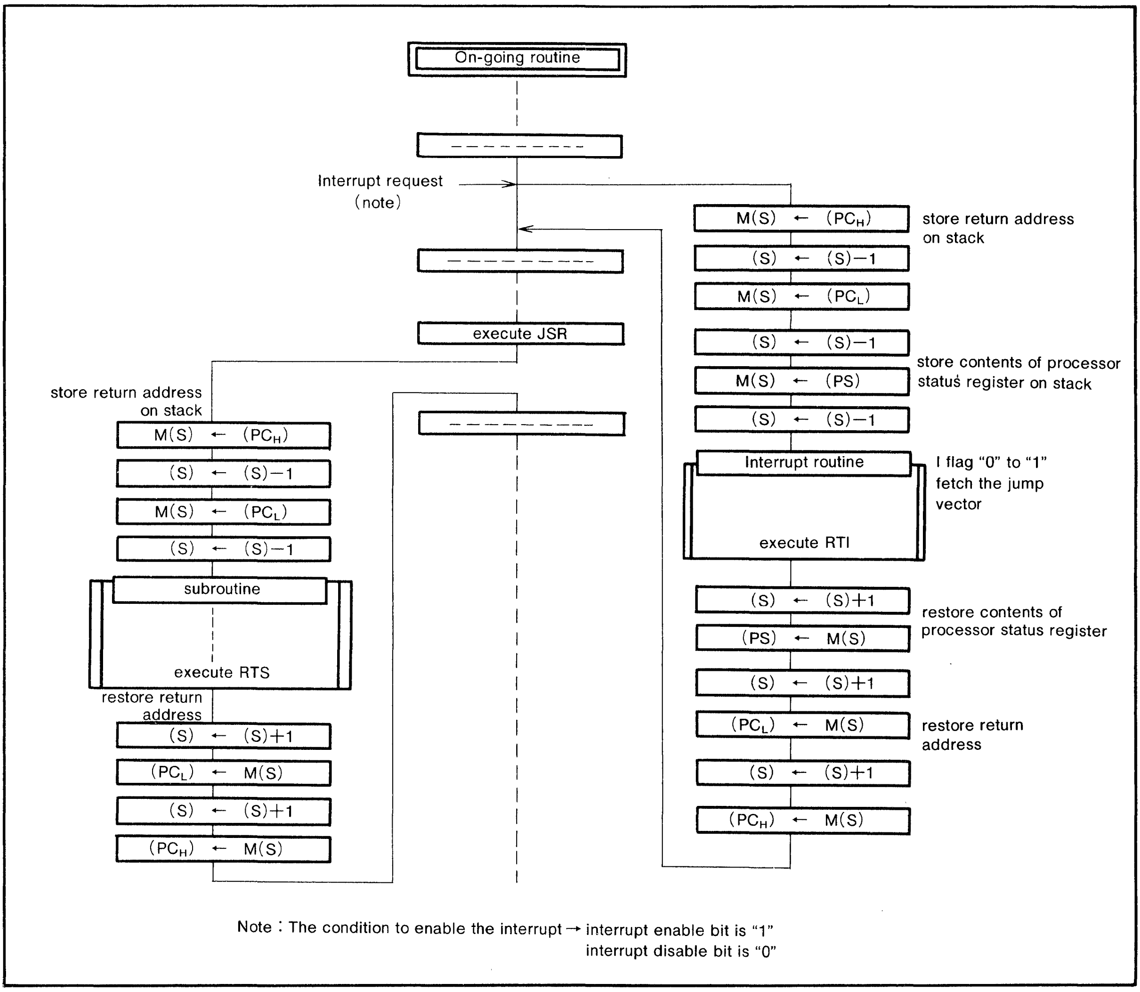
The accumulator (A) is the 8-bit general-purpose register. This is a highly used register for data operation, data transfer, temporary saves and conditional judgement.

2.1.2 Index register X(X), Index register Y(Y)

Index register X and Index register Y are both 8-bit registers. In the addressing mode using these index registers, memory access is made by adding the contents of the registers to the contents of the designated address. This addressing mode is used for subroutine table reference and memory table reference, etc.

Also, the index register has the increment, decrement, comparision, and data transfer functions which can be used as if it were an accumulator.

In index register X, its content shows the OPERAND address when the T flag in the processor status register is "1".



Note : The condition to enable the interrupt → interrupt enable bit is "1"
interrupt disable bit is "0"

Fig. 2.1.2 Stack store and restore sequence when executing interrupt and subroutine calls

2.1.3 Stack pointer (S)

Stack pointer is an 8-bit register used during subroutine calling and interrupts. When branching out from an on-going routine to subroutine and interrupt routine, it is necessary to remember the return address. Normally, internal RAM is used for storing the return address and this area is called stack area. Stack pointer is the pointer to show where the stack data are stored within the stack area.
Figure 2.1.2 shows the data store and restore sequence to the stack area.
Registers other than program counter and processor status register will not be stored automatically (during subroutine calling, the program counter only will be stored). Therefore, it is necessary, to store necessary registers by programming. (For the accumulator store and return, PHA and PLA instructions are used; for the store/return of processor status register, PHP and PLP instructions are used.)
Normally, a zero page RAM(0000_{16}~$00FF_{16}$) is used for the stack area. By setting the bit 7(SPS) of MISRG2(address $00DF_{16}$ mentioned in the latter part) to "1", the 1-page RAM (0100_{16}~$01FF_{16}$) can be used as the stack area. (However, this function cannot be used in the single-chip mode of M37450M2-XXXSP/FP, therefoe, do not set SPS to "1").

2.1.4 Program counter (PC)

Program counter is a 16-bit counter which consists of 8-bit register, PC_H and PC_L. This counter indicates the address of the next instruction to be executed.

2.1.5 Processor status register (PS)

Processor status register is an 8-bit register which is composed of flags to maintain the condition of the processor immediately after an operation.
After reset, I flag is set to "1", but other flags are unknown. T, D flags will directly affect operation so they must be always initialized.
Each bit of the processor status register is explained below :

(1) Carry flag C
The carry flag maintains the carry or borrow generated by the Arithmetic Logic Unit (ALU) immediately after an operation, it is also changed by the shift and rotate instructions. SEC, CLC instructions allow direct access for setting and resetting.

(2) Zero flag Z
This flag is set when the result of an operation and data transfer is "0" and reset when the result is other than "0".

(3) Interrupt disable flag I
This flag is used to disable all interrupts (except the ones caused by BRK instructions). When this flag is "1", it means interrupt disable condition. When an interrupt is accept, this flag is automatically set to "1" thereby preventing other interrupts. The SEI and CLI insturctions are used by way of programming to set and reset this flag, respectively.

(4) Decimal mode flag D
This flag is used to define whether addition and subtraction are executed in binary or decimal. If the flag is "0", the operation is executed in binary. When the content of the flag is "1", the operation is executed in decimal. Decimal correction is automatically executed. (ADC and SBC is the only instruction which can be performed in decimal mode). SED and CLD instructions are used to set and reset this flag, respectively. This can directly effects CPU operation ; always initialize after reset.

(5) Break flag B
The operation of a BRK instruction is similar to an interrupt. The BRK instruction is a non-maskable software interrupt that is used during program debugging. The break flag can be checked only by checking the content of the processor status register (PS) saved during an interrupt. The content of the processor status register (PS) is saved after setting flag B to "1" when the BRK instruction is used as an interrupt. It is cleared to "0" for other interrupts.

(6) Index X mode flag T
When this flag is "0", the operation between accumulator and memory is executed. When this flag is "1", the operation between memories, memory and I/O, I/O and I/O is executed directly and direct transfer of data is possible also without passing the accumulator. That is, the operation resulting between memories 1 and 2 are stored in memory 1. The address of memory 1 is specified by the contents of index register X and that of the memory 2 is specified by the normal addressing mode. Flag T is set and reset by the SET and CLT instructions. This can directly effects CPU; always initialize after reset.

(7) Overflow flag V
The overflow flag functions when one word is added or subtracted in binary with the sign. When the result exceeds +127 or −128, the overflow flag is set. Besides the above conditions, when BIT instruction is executed, the bit 6 of the memory, is input into the overflow flag. The overflow flag is reset by CLV insturction and no set instruction exists. This flag is non-function during the decimal operation mode.

(8) Negative flag N
The negative flag is set when the result of the data transfer or operation is negative (bit 7 is "1"). After execution of BIT instructions, bit 7 of the memory where the BIT instruction is executed, is stored into the negative flag. The negative flag can be neither set nor reset directly. This flag is nonfunctional during the decimal operation mode.

2.2 Access area

In M37450, all ROM, RAM, I/O, control registers are placed in the same memory area. Therefore, same instructions enable both data transfer and operation without the need to distinguish memory and I/O .
M37450 program counter consists of 16-bit and the accessible memory space is 64K byte from 0000_{16} to $FFFF_{16}$.
Within the 64K-byte memory area, the lowest 256-byte and the highest 256-byte are called zero-page and special page area, respectively. Both are accessible with 2-byte by using the individual special addressing mode.

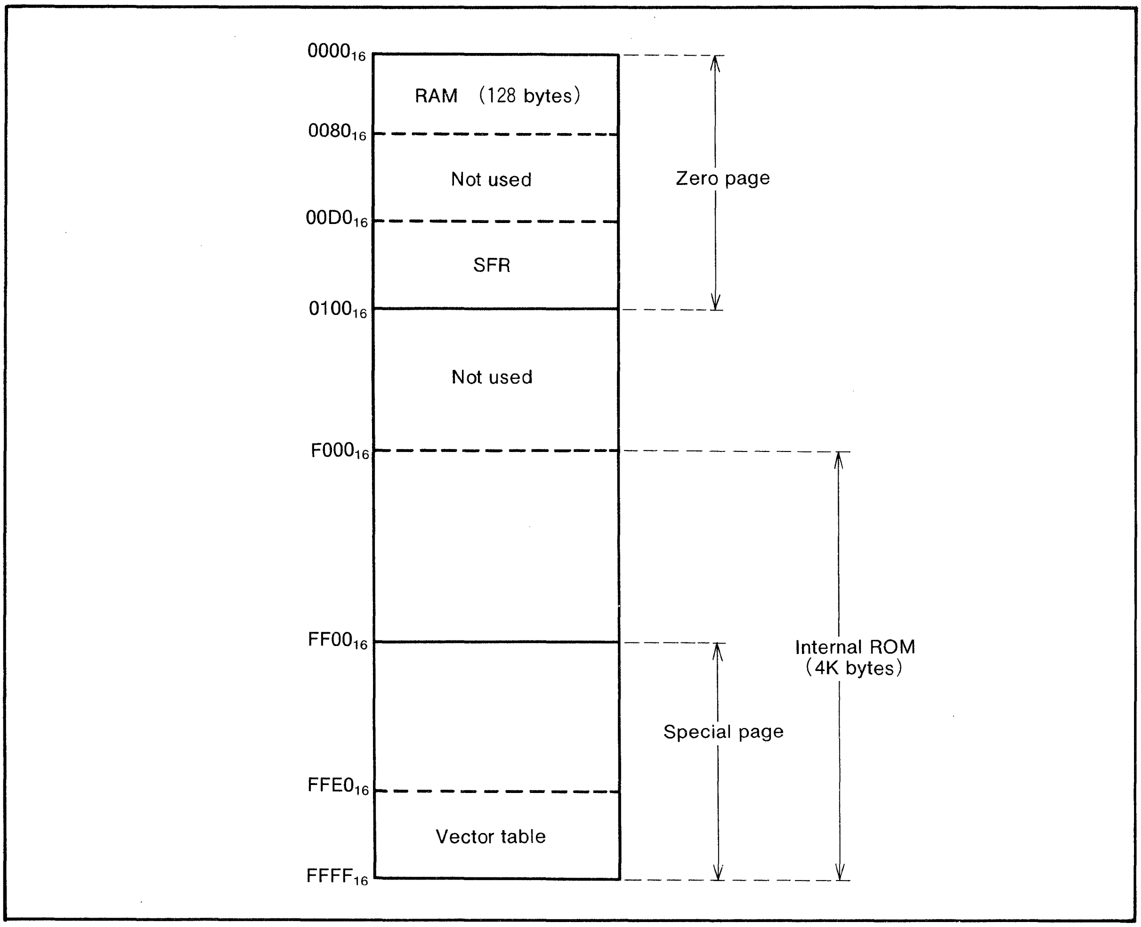


Fig. 2.2.1 Block diagram of memory access area (M37450M2-XXXSP/FP)

2.2.1 Zero page (0000_{16}~$00FF_{16}$)

The 256-byte addresses from 0000_{16} to $00FF_{16}$ are called zero page area where built-in RAM and Special Function Register (SFR) are assigned.

To specify the memory or register in the zero page area, use the zero page addressing mode shown in the Figure 2.2.2. Especially in this area, if using the instruction in the zero page specific addressing mode, we can access the zero page area with shorter instruction cycles.

2.2.2 Special page ($FF00_{16}$~$FFFF_{16}$)

The 256-byte addresses from $FF00_{16}$ to $FFFF_{16}$ are called special page area. To specify the memory in the special page area, use the special page addressing mode shown in the Figure 2.2.2. Especially in this area, if using the instruction in the special page specific addressing mode, we can access the special page area with shorter instruction cycles. Normally the highly used subroutines enter in this area.

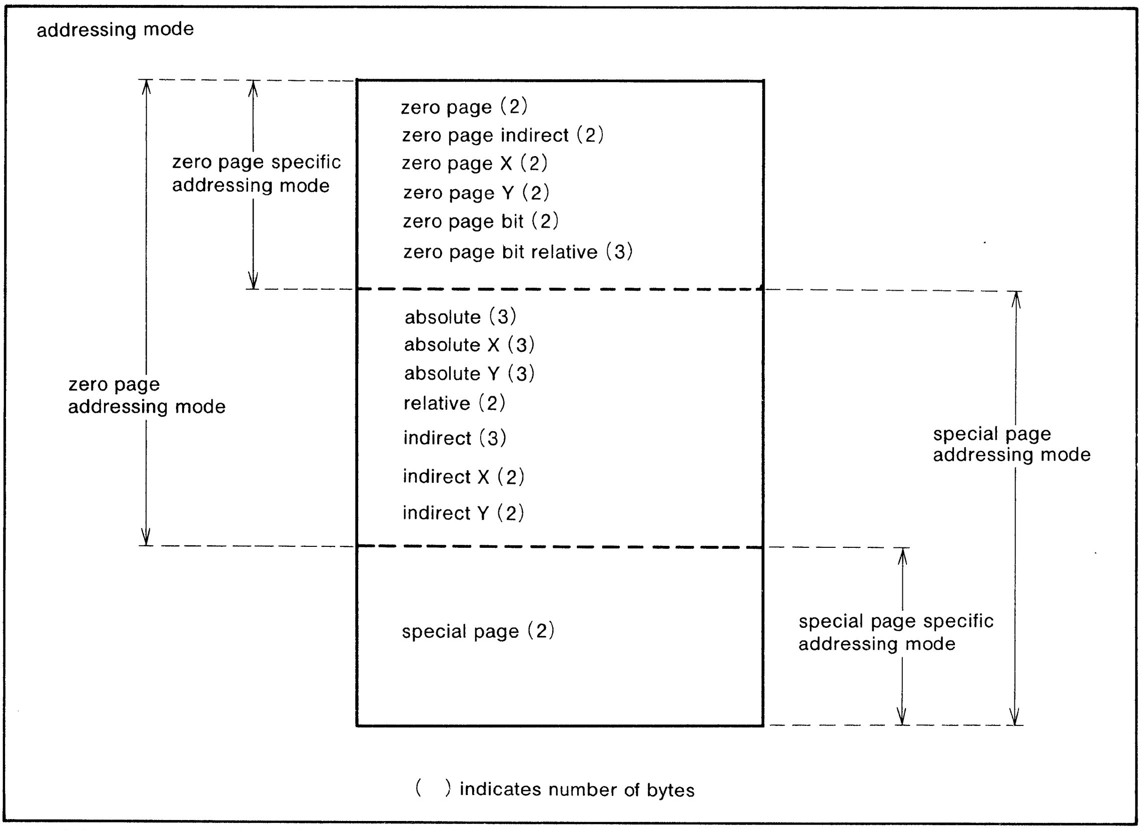


Fig. 2.2.2 Zero page and special page addressing modes

2.3 Memory map

A memory map for the M37450 (in single-chip mode) is shown in Figure 2.3.1. The following explains the memory and I/O assigned to the memory area in the single-chip mode.

(1) RAM (0000_{16}~$007F_{16}$)
In M37450, the static RAM with the 128 X 8-bit capacity is assigned in the addresses 0000_{16} to $007F_{16}$. The internal RAM is used for data storage, subroutine calling or the stack area when interrupts occur. When RAM is used as the stack area, the depth of the subroutine 'nesting' and the interrupt levels should be kept in mind in order to avoid destruction of the RAM contents.

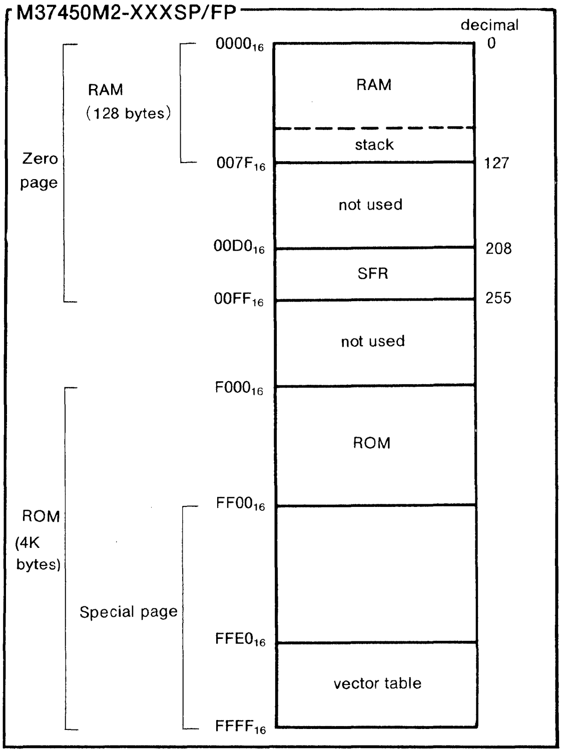
(2) ROM ($F000_{16}$~$FFFF_{16}$)
In M37450, the mask ROM with the 4K X 8-bit is assigned to the addresses $F000_{16}$ to $FFFF_{16}$. In the internal ROM, addresses $FFE0_{16}$ to $FFFF_{16}$ are assigned as the vector area of reset and for interrupts.

(3) Special function register; SFR ($00D0_{16}$~$00FF_{16}$)
Addresses $00D0_{16}$~$00FF_{16}$ are assigned for SFR (special function register) which includes input/output ports, timer, serial I/O, bus interface, A-D converter, D-A converter, PWM, miscellaneous registers related to interrupts and CPU modes.

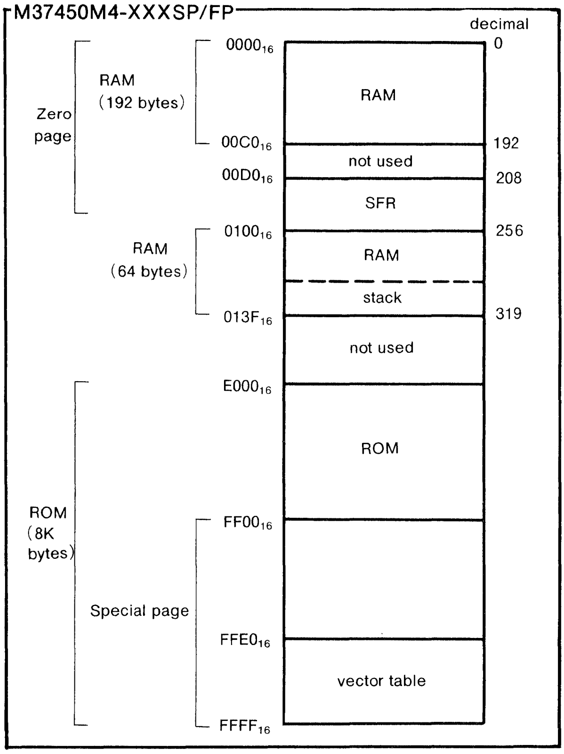
$00D0_{16}$	P0 data register	$00E8_{16}$	Serial I/O control register
$00D1_{16}$	P0 directional register	$00E9_{16}$	UART control register
$00D2_{16}$	P1 data register	$00EA_{16}$	Baud rate generator
$00D3_{16}$	P1 directional register	$00EB_{16}$	PWM register (L)
$00D4_{16}$	P2 data register	$00EC_{16}$	PWM register (H)
$00D5_{16}$	P2 directional register	$00ED_{16}$	Timer 1 control register 1
$00D6_{16}$	P3 data register	$00EE_{16}$	Timer 2 control register 2
$00D7_{16}$	P3 directional register	$00EF_{16}$	Timer 3 control register 3
$00D8_{16}$	P4	$00F0_{16}$	Timer 1 register (L)
$00D9_{16}$	Reserved	$00F1_{16}$	Timer 1 register (H)
$00DA_{16}$	P5 data register	$00F2_{16}$	Timer 1 latch (L)
$00DB_{16}$	P5 directional register	$00F3_{16}$	Timer 1 latch (H)
$00DC_{16}$	P6 data register	$00F4_{16}$	Timer 2 register (L)
$00DD_{16}$	P6 directional register	$00F5_{16}$	Timer 2 register (H)
$00DE_{16}$	Miscellaneous register 1	$00F6_{16}$	Timer 2 latch (L)
$00DF_{16}$	Miscellaneous register 2	$00F7_{16}$	Timer 2 latch (H)
$00E0_{16}$	D-A$_1$ register	$00F8_{16}$	Timer 3 register (L)
$00E1_{16}$	D-A$_2$ register	$00F9_{16}$	Timer 3 register (H)
$00E2_{16}$	A-D successive approximation register	$00FA_{16}$	Timer 3 latch (L)
$00E3_{16}$	A-D control register	$00FB_{16}$	Timer 3 latch (H)
$00E4_{16}$	Data bus buffer (DBBIN/DBBOUT)	$00FC_{16}$	Interrupt request register 1
$00E5_{16}$	Data bus buffer status register	$00FD_{16}$	Interrupt request register 2
$00E6_{16}$	Transmit/Receive buffer register	$00FE_{16}$	Interrupt control register 1
$00E7_{16}$	Serial I/O status register	$00FF_{16}$	Interrupt control register 2

Fig. 2.3.1 SFR map

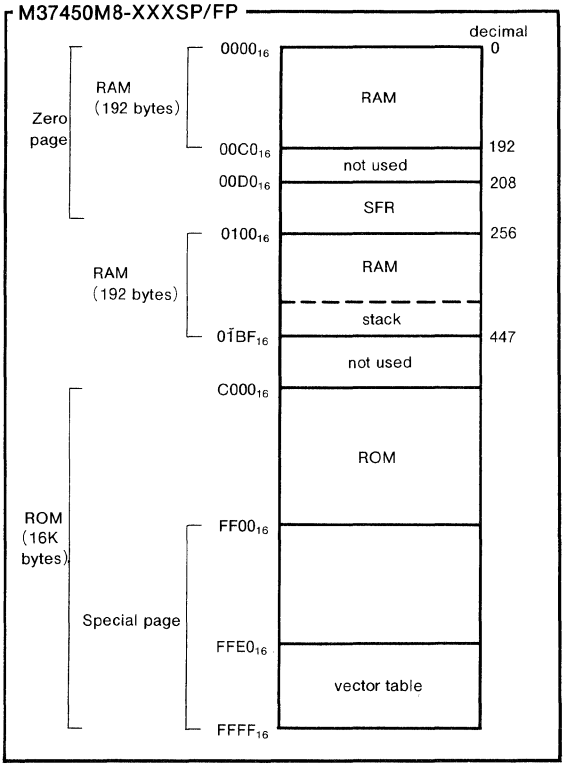
M37450M2-XXXSP/FP



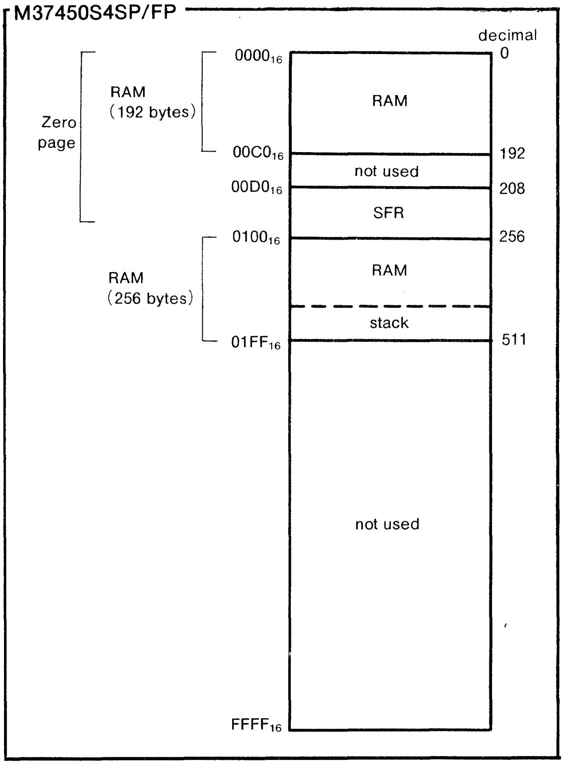
M37450M4-XXXSP/FP



M37450M8-XXXSP/FP



M37450S4SP/FP



2.3.1 Processor mode

M37450 can select three processor modes by changing the contents of the processor mode bit (bit 0 and 1 of address $00DF_{16}$) while CNVss pin is connected to Vss.
The relationship between the processor mode bit and each processor mode is shown in Table 2.3.1.
Also, these modes can be controlled by the input level of the CNV_{SS} pin.
Microprocessor mode starts when CNV_{SS} pin is connected to V_{CC}, and the processor mode bits (b1, b0) are automatically set to (1, 0).
In this case, only microprocessor mode can be selected, and do not change the processor mode bits.
CNV_{SS} pin must be connected to V_{CC} in case of external ROM version.
The Table 2.3.2 shows the relationship between the CNV_{SS} pin input level and the processor mode.

(1) Single-chip mode
The all input/output ports of M37450 have the original function.

(2) Memory expanding mode
This mode is used when the internal memory, I/O, and the peripheral functions are not enough. In this mode, the register area for port P0~P2 cannot be used, but all other memory and the peripheral functions can be used.
When memory expanding mode is selected, the functions of Port P0~P2 change as shown in Table 2.3.3.
External memory can be placed in any area of the 64K bytes address. In the area where the external memory and the internal memory overlap, the internal memory has the priority to be read out. When the CPU reads out this area, therefore, the data from the external memory is not read into the CPU. The write-in in this area, however, will be made in the respective internal and external areas.

(3) Microprocessor mode
Same as the memory expanding mode except the access to the internal ROM area being prohibited. In this mode, ROM can easily be attached externally. Therefore this mode is used for small production or as the proto-type model before mass production. Figure.2.3.3 shows the memory map for each mode except the single-chip mode.

Table 2.3.1 Relationship between the processor mode and the processor mode bit

		b1	b0	
DBBM	DBBE	CM1	CM0	MISRG2; address $00DF_{16}$

b1	b0	Processor mode bit
0	0	Single-chip mode
0	1	Memory expanding mode
1	0	Microprocessor mode
1	1	Inhibit use

Table 2.3.2 Relationship between CNV$_{SS}$ pin input level and the processor mode

CNV$_{SS}$ pin	Processor mode
Connect to V$_{SS}$	Start as the single-chip mode after reset. Three modes can be realized by the processor mode bit.
Connect to V$_{CC}$	Microprocessor mode starts after reset. Only microprocessor mode can be selected.

Table 2.3.3 The function of each port during memory expanding mode

Port name	Function
Port P0	Output the lower 8 bits of address.
Port P1	Output the higher 8 bits of address.
Port P2	Works as input/output pin for data $D_7 \sim D_0$ (including instruction code). (The direction of the data bus will be determined by R/$\overline{W}$ status.)

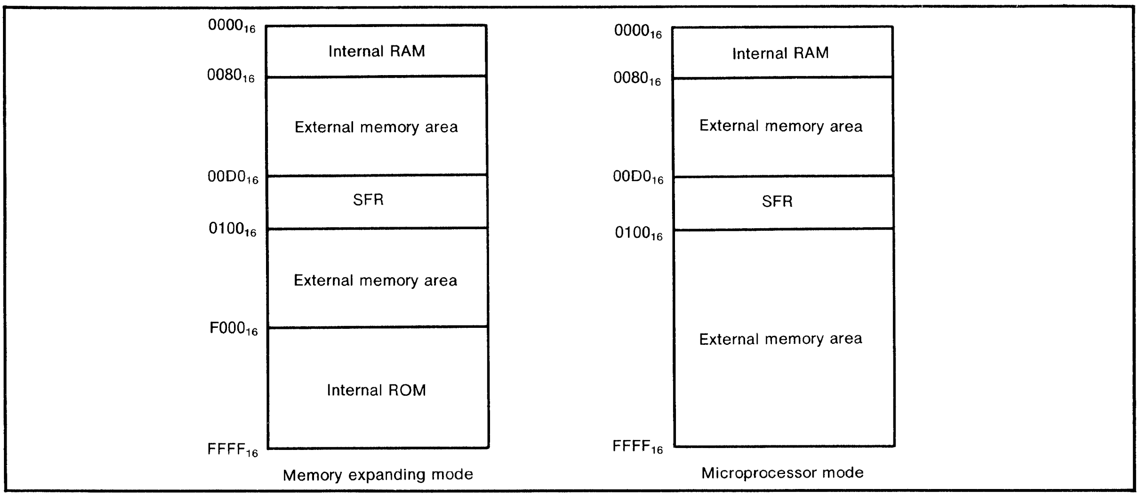


Fig. 2.3.3 Memory map for each processor mode (except single-chip mode)

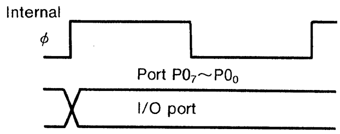
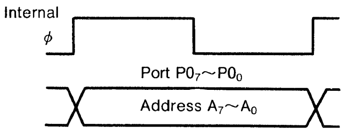
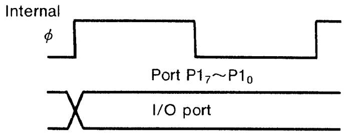
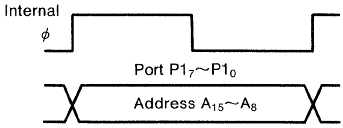
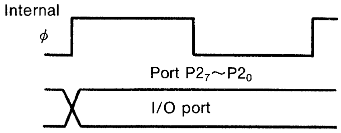
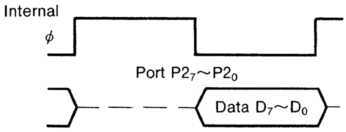
Port \ Mode	Single-chip mode	Memory expanding mode	Microprocessor mode
CM1	0	0	1
CM0	0	1	0
Port P0			Same as left
Port P1			Same as left
Port P2			Same as left

Fig. 2.3.4 The function of ports P0~P2 in each processor mode

2.3.2 Bus control during memory expansion

M37450 is equipped with the software wait insertion flag, RDYE (bit 6 of address 00DF$_{16}$) which enables easy access in case of memory expanding and microprocessor mode with expansion of external memory and I/O.

When the RDYE flag is set, the bus operation is two times as slow as the original bus cycle ϕ. The bus cycle changes at the next cycle after access is made to the RDYE flag. (see FIgure 2.3.5)

- Normal operation (RDYE=0) : ϕ=2.5MHz (f(X_{IN})=10MHz)
- Slow operation (RDYE=1) : ϕ=1.25MHz (f(X_{IN})=10MHz)

The flag setting affects the bus cycle even in the single-chip mode such as memory accessing and the peripheral function units except the timers, PWM, UART, etc.
The RDYE flag is set to "0" at reset and the operation starts in no-software wait mode.

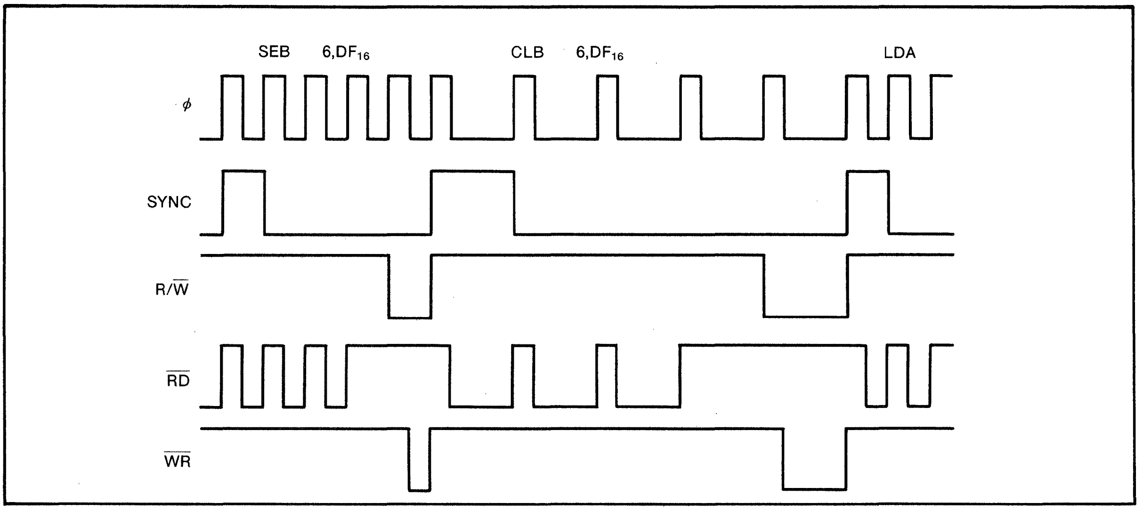
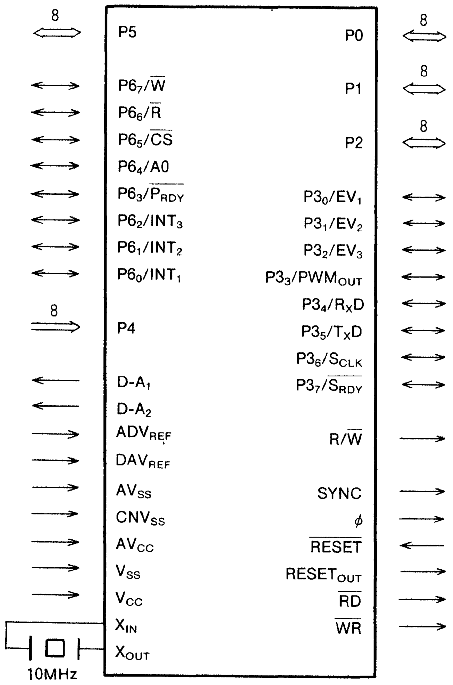


Fig. 2.3.5 Bus-cycle changing by RDYE flag

2.4 Input/Output pin

M37450 contains 53 input/output ports (58 in the 80-pin model).
Figure 2.4.1 shows the input/output ports of the 80-pin model.



Note : In the 64-pin model, P4 has 3 pins and no $\overline{RD}$, $\overline{WR}$, AV_{CC}, $RESET_{OUT}$ pins.
The reference voltage input pins for A-D and D-A converters are common in 64-pin model.

Fig. 2.4.1 Input/Output ports (80-pin model)

2.4.1 Input/Output ports

M37450 has the input/output ports shown in Table 2.4.1.

Table 2.4.1 Input/Output ports

Port		Number of pins
Programmable I/O port	Port P0~P3, P5, P6	48

Input port	Port 4	3(64-pin model)
		8(80-pin model)

Output port	D-A	2

Port P3, P4, and P6 are double-functional ports and the function can be selected by program.

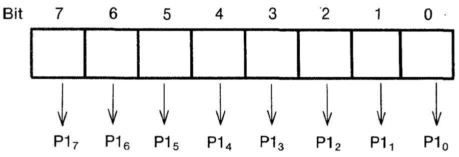
The following shows how to determine the port direction of the programmable I/O port.

Data that is written on the programmed output pin is stored in the port latch and is transferred to the output pin. When data is input to the programmed pin, data is read not from output pin but from output latch. Therefore, previously output data can be read correctly regardless of the logical level of the pin due to output loading.

Because the programmed input pin is floating, the value of the pin can be read correctly. When data is written to the programmed input pin, it is written only to the port latch and the pin remains floating.

The directional register can be utilized as same as the memory in the zero page; therefore using the zero page addressing mode achieves a short access cycle.

The directional register corresponding to each port is located in the SFR area assigned to address $00D0_{16}$~ $00FF_{16}$. Each bit of this directional register determines the corresponding port direction. Bit and pin correspond in the follwing manner:



Port P1 directional register (address $00D3_{16}$)

On the corresponding bit of the pin:

When "0" is written in, the corresponding port is an input pin.
When "1" is written in, the corresponding port is an output pin.

At reset, each directional register is initialized to "00_{16}", the I/O port is input.

Example: When "$6B_{16}$" is written in the P0 directinal register ($00D1_{16}$).

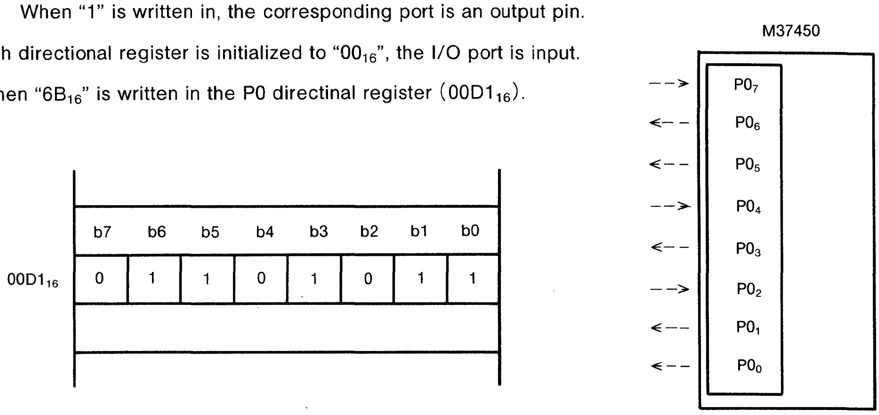
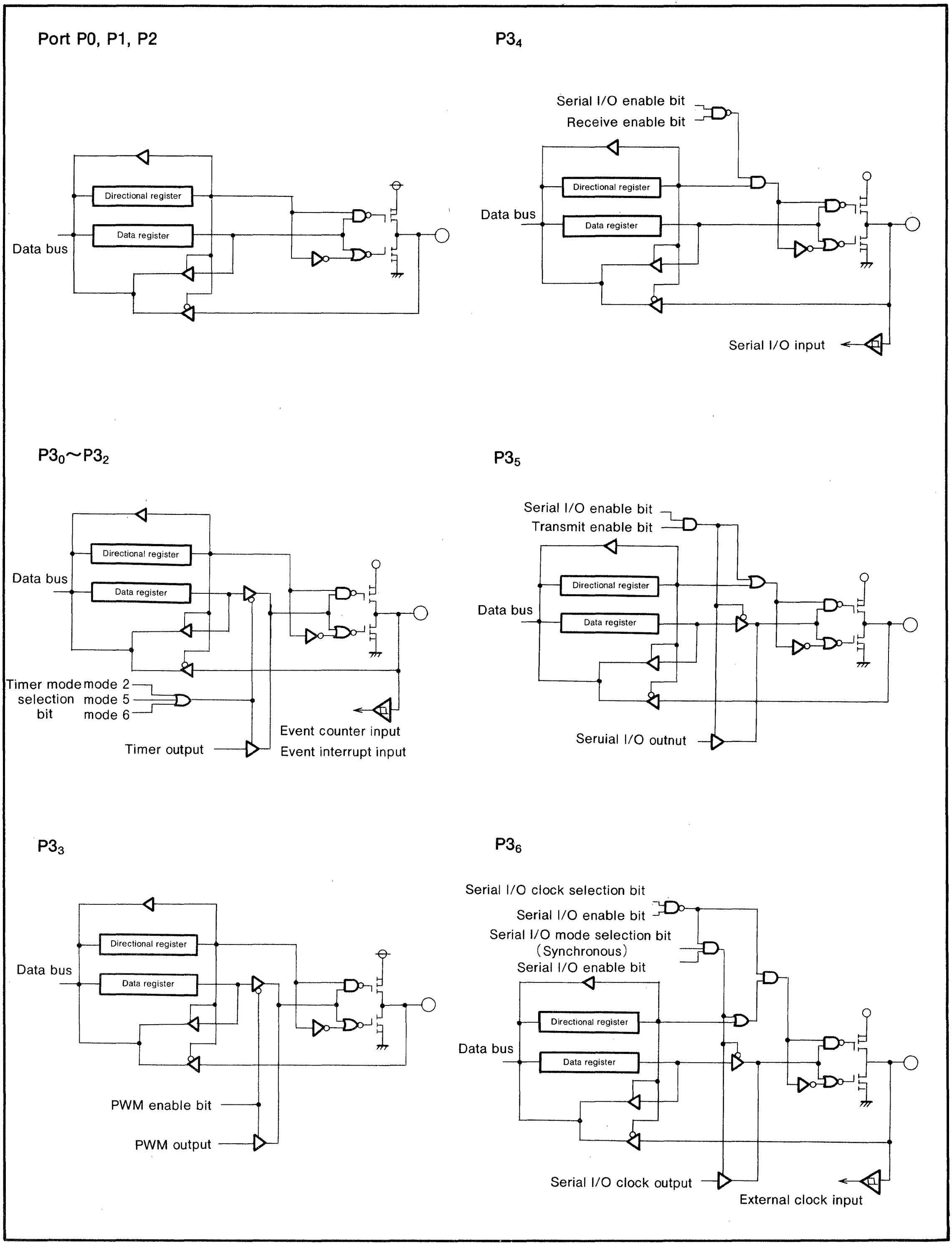


Table 2.4.2 Port description

Port P0	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible	Except in the single-chip mode, outputs the lower address byte.
Port P1	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible	Except in the single-chip mode, outputs the upper address byte.
Port P2	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible	Except in the single-chip mode, the data bus information is input and output.
Port P3	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible (However, when EV1~EV3, R_XD and S_{CLK} pins are selected, they have schmitt input and not TTL compatible)	This port is not affected by the processor mode. All ports are double-functional ports in which functions can be selected by program.
Port P4	3-bit input port···64-pin model 8-bit input port···80-pin model Input format : TTL compatible	This port can be used as analog input pin and also as digital input port. When this port is read in 64-pin model, the upper 5 bits are unknown.
Port P5	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible	This port is not affected by the processor mode. Port P5 can become automatically the data bus for the host CPU when the slave mode is selected. (bit 2 of address $00DF_{16}$ as "1").
Port P6	8-bit programmable input/output port Output format : CMOS form Input format : TTL compatible (However, when INT_1~INT_3 are selected, they have schmitt input and not TTL compatible)	This port is not affected by the processor mode. Within this port, $P6_3$~$P6_7$ 5-bit can become automatically the control bus for the host CPU by selecting the slave mode. Port $P6_0$ ~ $P6_2$ 3-bit port are double functions and functions can be selected by program.
$D\text{-}A_1$ $D\text{-}A_2$	2-bit analog output port	The value written-in the D-A converter register ($00E0_{16}$~$00E1_{16}$ address) generates the corresponding analog voltage.



Port P0, P1, P2

$P3_4$

$P3_0$~$P3_2$

$P3_5$

$P3_3$

$P3_6$

Fig. 2.4.2 Block diagram of ports P0~P6 (single-chip mode) and output only pin output format(1)

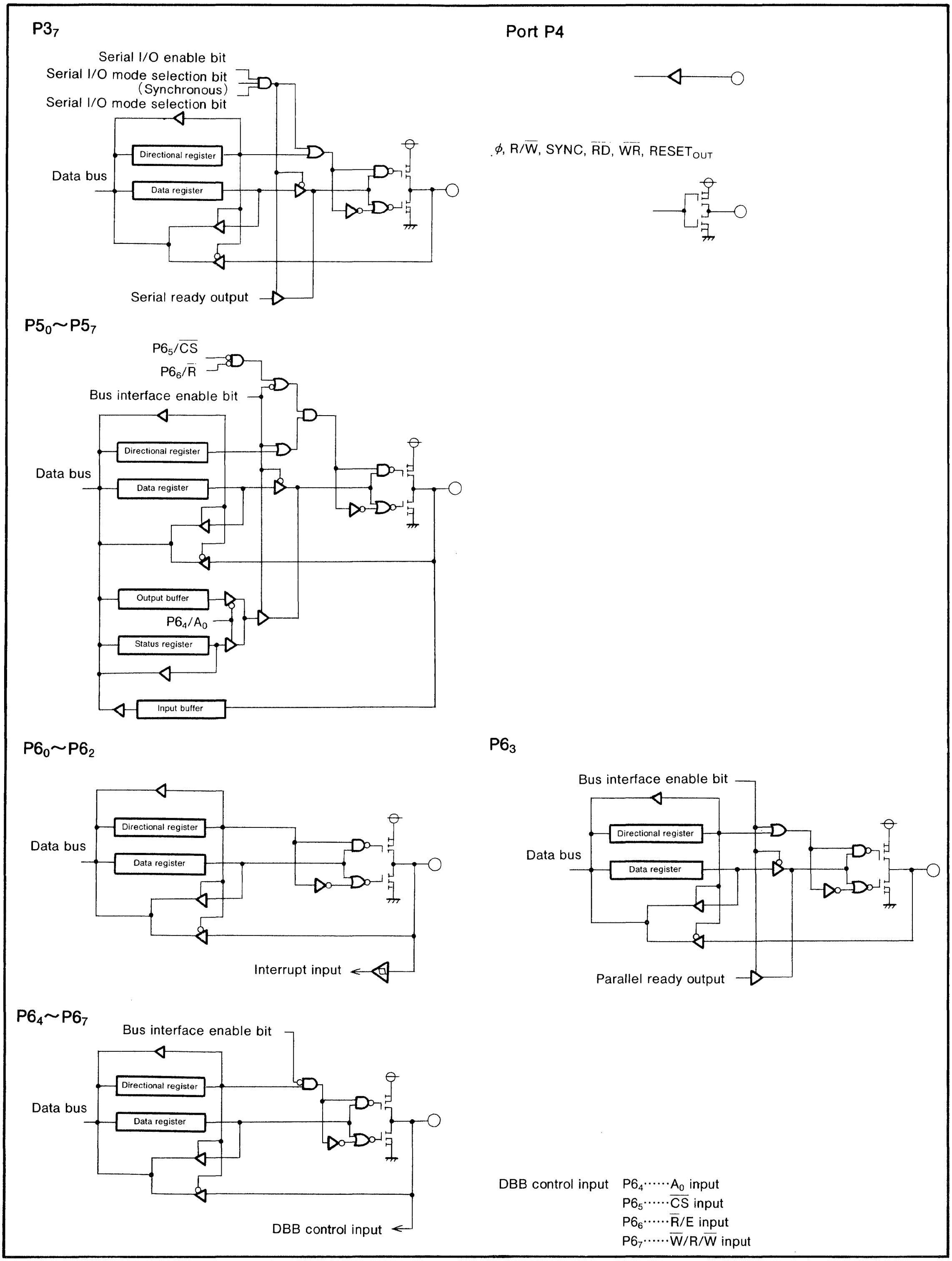


Fig. 2.4.3 Block diagram of ports P0～P6 (single-chip mode) and output only pin output format (2)

2.4.2 Pin descriptions

(1) R/$\overline{\text{W}}$ pin
R/$\overline{\text{W}}$ pin is the signal which determines the direction of the data bus. When the pin level is "H", the data bus is in read state and when the level is "L", it is write state. Ordinarily, this signal is combined with ϕ signal and used to control $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals in both memory expanding and microprocessor modes. Also this pin can be used as an alternate switching signal to change the direction of the external bus buffer. Figure 2.4.4 shows an example for such a use.

(2) $\overline{\text{RD}}$, $\overline{\text{WR}}$ pin (Only in 80-pin model)
The $\overline{\text{WR}}$ signal output from $\overline{\text{WR}}$ pin is functionally the same signal as the one that results by combining the R/$\overline{\text{W}}$ and ϕ signals as shown in Figure 2.4.4. A "L" signal is output when the bus is in a write cycle.
The $\overline{\text{RD}}$ signal output from $\overline{\text{RD}}$ pin is approximately the same as the $\overline{\text{RD}}$ signal shown in Figure 2.4.4. A "L" signal is output only when CPU is actually reading data from the data bus.
Using this model design with peripheral devices is easily accommodated due to the separation of the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals. Also the $\overline{\text{RD}}$ signal is not generated in a read cycle in which a dummy address is output, again allowing for easy connection to peripheral devices in the memory expanding and microprocessor mode.

(3) X_{IN}, X_{OUT} pin
X_{IN} and X_{OUT} are clock input/output pins. M37450 has a built-in clock generating circuit. The oscilation frequency is determined by a ceramic or crystal oscilator. When using an external clock, the clock source should be connected to X_{IN} pin and the X_{OUT} pin should be left open.

(4) ϕ pin
This pin outputs the internal system clock (the oscillation frequency between X_{IN} and X_{OUT} divided by 4). During a STP and WIT instructions, operation ceases and ϕ is kept in a "H" state.

(5) SYNC pin
This pin outputs a "H" signal for one ϕ cycle at op-code fetch. It is used to control single-step operation of the program. Since this signal is also output when the CPU accesses internal memory and I/O, it can be used to monitor the internal operation.

(6) $\overline{\text{RESET}}$ pin
To enter the reset state, the reset input pin must be kept at a "L" for more than 2μs (under normal V_{CC} conditions). If more time needed for the crystal oscillator to stabilize, this "L" condition should be maintained for the required time.

(7) RESET_{OUT} pin
This pin becomes active "H" during reset and can be used as a reset signal for the peripheral devices. This is only in the 80-pin model.

(8) V_{REF} pin
This pin inputs the reference voltage for the built-in A-D and D-A converters. This is only in the 64-pin model.

(9) ADV_{REF} pin
This pin is used to input the reference voltage for the built-in A-D converter. It is only in the 80-pin model.

(10) DAV_{REF} pin
This pin is used to input the reference voltage for the built-in D-A converter. It is only in the 80-pin model.

(11) CNV_{SS} pin
Depending on the level input to this pin, the CPU will operate in a specific mode at reset (please refer to 2.3.1 for details). But by changing the level of this pin after reset, this mode remains unchanged.

(12) V_{SS}, V_{CC}, AV_{SS}, AV_{CC} pin

Power supply pins for this device and for the built-in A-D, D-A converters. AV_{CC} pin is only in the 80-pin model and is connected to the built-in A-D converter. In the 64-pin model, it is internally connected to V_{CC}.

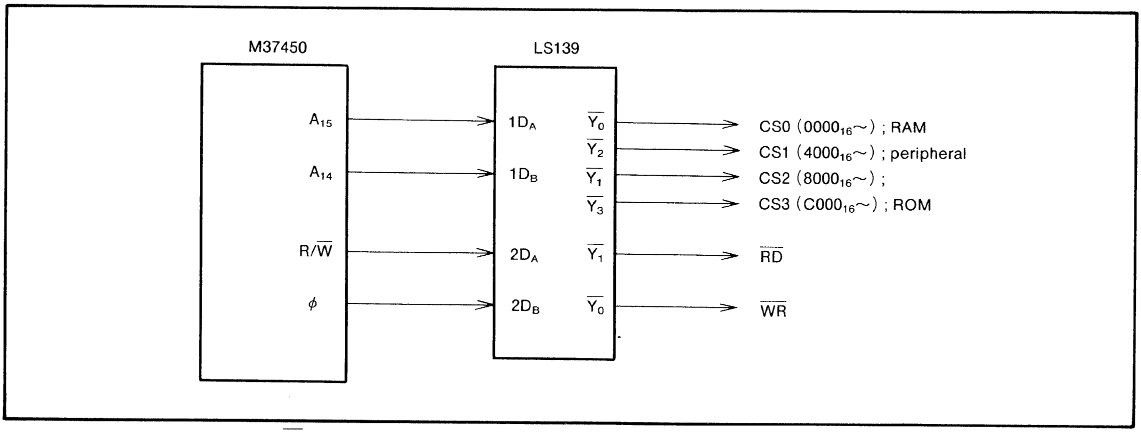


Fig. 2.4.4 Example use of $R/\overline{W}$ signal

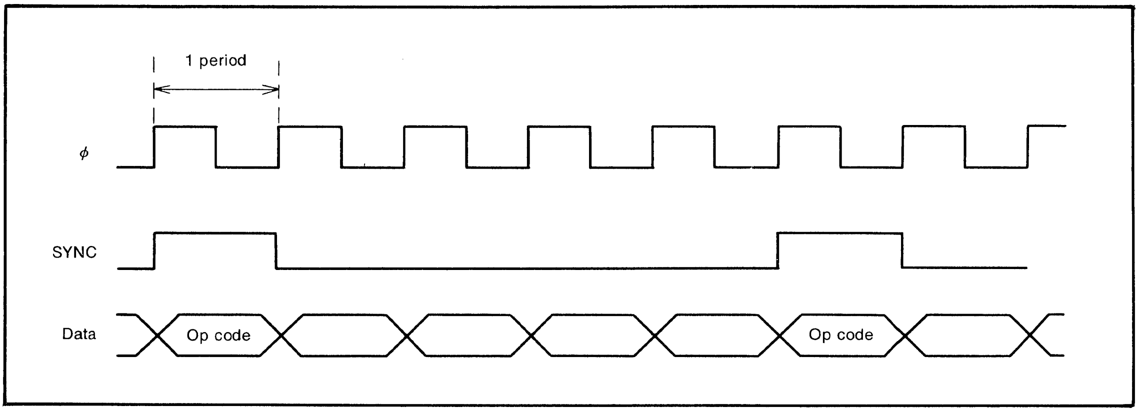


Fig. 2.4.5 SYNC output

2.5 Interrupt

Interrupts are usually used in the following cases:

- When the processing routine has higher priority than the on-going program.
- When a routine must be executed at specific interval.

The M37450 has 15 interrupt sources. Table 2.5.1 shows the priority and vector address of these interrupts. Each interrupt has its own jump vector address and a fixed priority.
These 15 interrupts are prioritized as shown in Table 2.5.1 (Reset input has the highest priority among the interrupts). When two or more interrupt requests are generated at the same sampling point, the interrupt having the higher priority is accepted. The priority order is determined by hardware. However, multiple priority processing through software is possible by using interrupt control flags (interrupt enable bit, interrupt disable flag) .

Table 2.5.1 Interrupt vector table and priority

Priority	Interrupt source	Vector address		Remarks
		Upper	Lower	
1	RESET (see note)	FFFF$_{16}$	FFFE$_{16}$	Non-maskable
2	Input buffer full interrupt	FFFD$_{16}$	FFFC$_{16}$	Valid only when data bus interface function is selected
3	Output buffer empty interrupt	FFFB$_{16}$	FFFA$_{16}$	
4	INT$_1$ interrupt	FFF9$_{16}$	FFF8$_{16}$	Polarity programmable
5	INT$_2$ interrupt	FFF7$_{16}$	FFF6$_{16}$	Polarity programmable
6	INT$_3$ interrupt	FFF5$_{16}$	FFF4$_{16}$	Polarity programmable
7	Timer 1 interrupt	FFF3$_{16}$	FFF2$_{16}$	
8	Timer 2 interrupt	FFF1$_{16}$	FFF0$_{16}$	
9	Timer 3 interrupt	FFEF$_{16}$	FFEE$_{16}$	
10	EV$_1$ (external event) interrupt	FFED$_{16}$	FFEC$_{16}$	Polarity programmable
11	EV$_2$ (external event) interrupt	FFEB$_{16}$	FFEA$_{16}$	Polarity programmable
12	EV$_3$ (external event) interrupt	FFE9$_{16}$	FFE8$_{16}$	Polarity programmable
13	Serial I/O receive interrupt	FFE7$_{16}$	FFE6$_{16}$	Valid only when serial I/O function is selected
14	Serial I/O transmit interrupt	FFE5$_{16}$	FFE4$_{16}$	
15	A-D conversion completion interrupt	FFE3$_{16}$	FFE2$_{16}$	
16	BRK instruction interrupt	FFE1$_{16}$	FFE0$_{16}$	Non-maskable software interrupt

Note : Reset is included in the table since it operates the same as other interrupts with the exception that it is non-maskable

2.5.1 Interrupt sources

The following explains each interrupt source:

(1) INT$_1$, INT$_2$, INT$_3$ interrupts

An interrupt request is generated when a level-change from "H" to "L" or from "L" to "H" of the INT$_1$, INT$_2$, or INT$_3$ pin is detected (However, a pulse width of 160ns or more for "H" or "L" is necessary at the effective polarity). These polarities are selected by bits 0∼2 of MISRG1 (address 00DE$_{16}$) .
INT$_1$∼INT$_3$ are common with P6$_0$∼P6$_2$ pins. They always detect the levels of port P6$_0$∼P6$_2$.
At reset, MISRG1 is cleared to "00$_{16}$". So, INT$_1$∼INT$_3$ interrupt request is generated at the falling edge.

(2) Timer 1∼Timer 3 interrupts

An interrupt request is generated at the next clock pulse after the timer 1, timer 2 or timer 3 reaches "0000$_{16}$".

(3) EV$_1$, EV$_2$, EV$_3$ interrupts

An interrupt request is generated at the valid edge of the event wave form. (However, a pulse width of 160ns or more for the "H" or "L" is necessary at the effective polarity.) Selection of polarities is decided by bit 5 of the timer 1∼3 control register.

(4) Input buffer full (IBF) interrupt (Valid only when the data bus interface function is selected)

The state of data bus is latched into the input data bus buffer (DBBIN) at the rising edge of $\overline{W}$ after both $\overline{CS}$ and $\overline{W}$ are simultaneously "0" . At this point the IBF interrupt request is generated.

(5) Output buffer empty (OBE) interrupt (Valid only when the data bus interface function is selected.)

When all $\overline{CS}$, $\overline{R}$ and A$_0$ are simultaneously "0" , the data from the output data bus buffer is output to the data bus. Then at the rising edge of $\overline{R}$ signal, OBE interrupt request is generated.

(6) Receive interrupt (valid only when serial I/O is selected.)

An interrupt request is generated when all data is received in the receive shift register and its contents are moved into the receive buffer.

(7) Transmission interrupt (valid only when serial I/O is selected.)

In the case of the transmission interrupt, the interrupt timing can be selected by the transmission interrupt control bit (TIC) , bit 3 of the serial I/O control register (See Table 2.5.2).

(8) A-D conversion completion interrupt

Interrupt request is generated at the completion of the A-D conversion.

(9) BRK interrupt

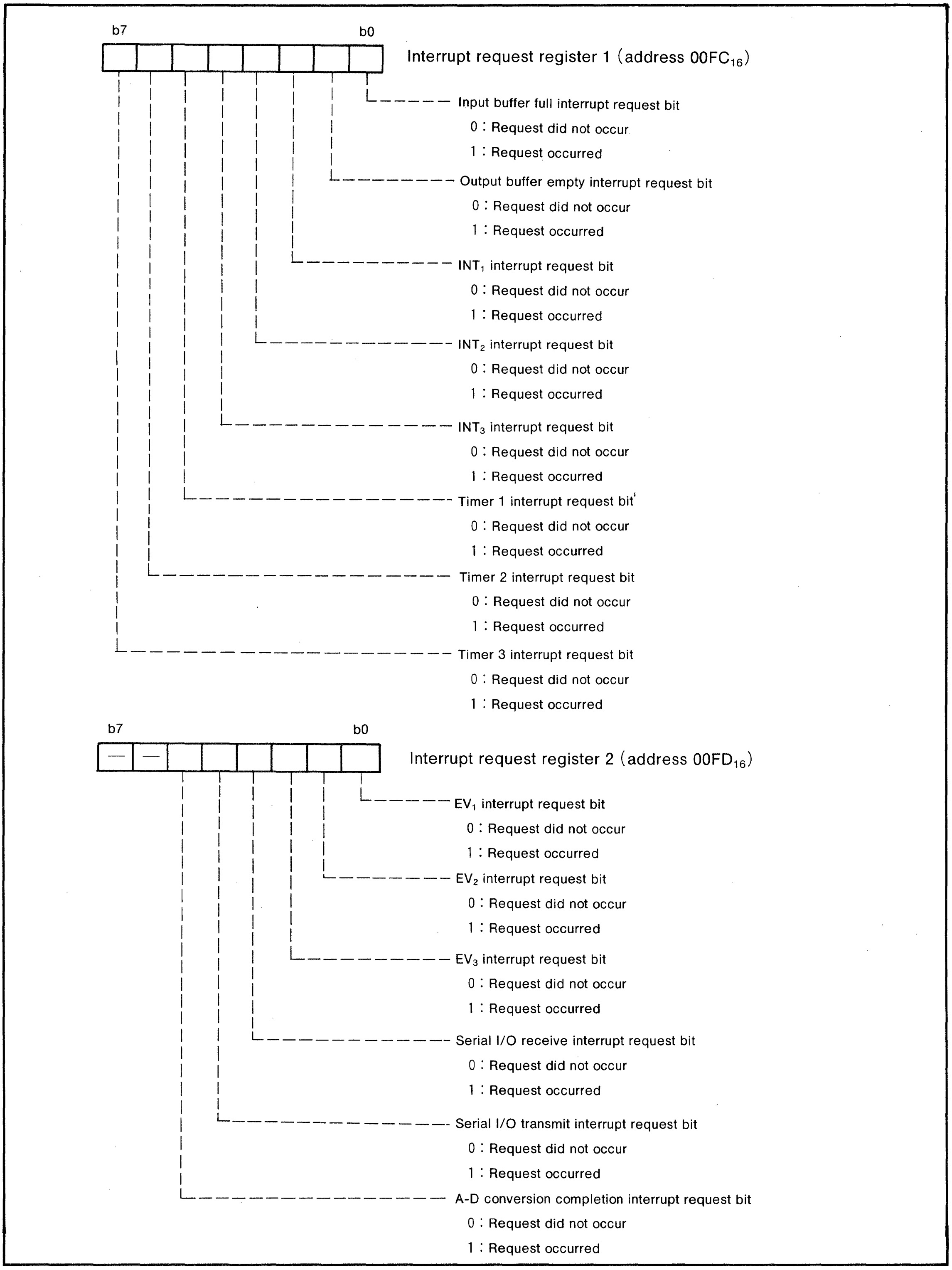
This interrupt is the lowest priority order software interrupt. It does not have a corresponding interrupt enable flag nor it is effected by the interrupt disable flag (non-maskable). If some interrupt requests are pending by interrupt enable bits or interrupt inhibit flag at BRK instruction execution, the service routine of the highest priority among the pending one is executed.

For detailed explanation of each interrupt, please refer to the section on each function.

Table 2.5.2 Selection of interrupt source by TIC

TIC	Interrupt sources
"0"	The data written into the transmission buffer is transferred to the transmit shift register and when the transmission buffer becomes empty, the interrupt request is generated.
"1"	When the shift operation of the transmission shift register is completed, the interrupt request is generated.

Table 2.5.3 Structure of registers related to interrupt (1)



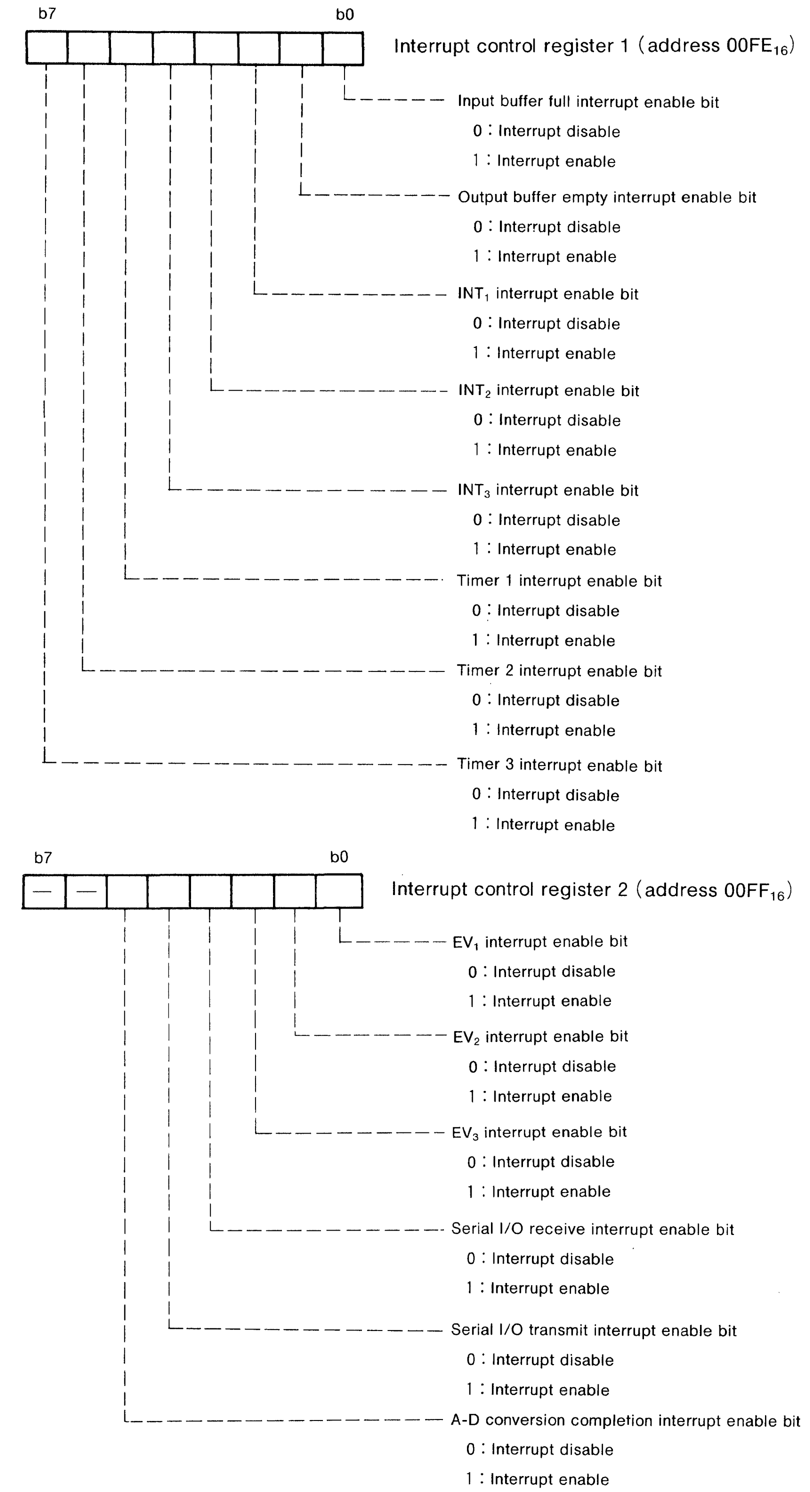
Interrupt request register 1 (address $00FC_{16}$)

- Input buffer full interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Output buffer empty interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- INT_1 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- INT_2 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- INT_3 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Timer 1 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Timer 2 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Timer 3 interrupt request bit
  0 : Request did not occur
  1 : Request occurred

Interrupt request register 2 (address $00FD_{16}$)

- EV_1 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- EV_2 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- EV_3 interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Serial I/O receive interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- Serial I/O transmit interrupt request bit
  0 : Request did not occur
  1 : Request occurred
- A-D conversion completion interrupt request bit
  0 : Request did not occur
  1 : Request occurred

Table. 2.5.4 Structure of registers related to interrupt (2)



Interrupt control register 1 (address $00FE_{16}$)

Input buffer full interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Output buffer empty interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

INT_1 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

INT_2 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

INT_3 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Timer 1 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Timer 2 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Timer 3 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Interrupt control register 2 (address $00FF_{16}$)

EV_1 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

EV_2 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

EV_3 interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Serial I/O receive interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

Serial I/O transmit interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

A-D conversion completion interrupt enable bit
0 : Interrupt disable
1 : Interrupt enable

2.5.2 Interrupt control

The interrupt control model for M37450 is shown in Figure 2.5.1.

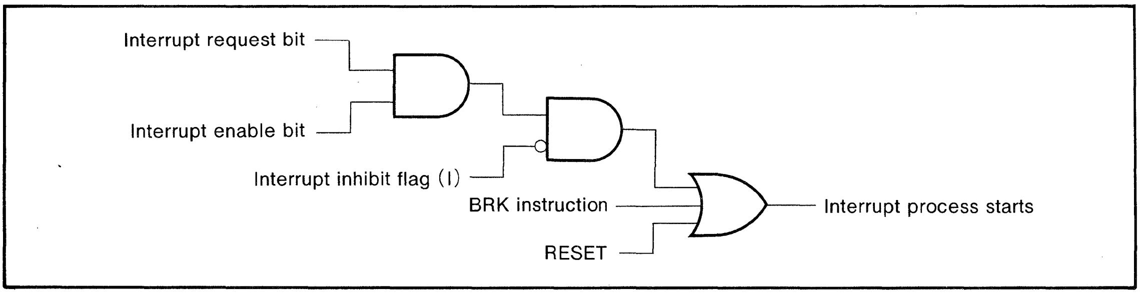


Fig. 2.5.1 Interrupt control

The interrupts for M37450, as shown in Figure 2.5.1, are controlled by the interrupt request bit, interrupt enable bit, and interrupt inhibit flag (I) except the software interrupt by BRK instruction.
These control bits and control flags are explained below:
(1) Interrupt request bit
Whenever interrupt request is generated, the request bit corresponding to the respective interrupt is set to "1".
The request bit remains "1" until the interrupt is accepted, at which point it is automatically cleared. This bit can be cleared in program, but can not be set. The interrupt request bits are in the interrupt request registers 1 and 2 (addresses $00FC_{16}$ and $00FD_{16}$).

(2) Interrupt enable bit
This bit determines whether an interrupt will be accepted or not. When the enable bit is "0" an interrupt by the corresponding interrupt source is prohibited. When this bit is "1" an interrupt by the corresponding source is allowed.

(3) Interrupt inhibit flag (I)
Interrupt inhibit flag (I) is in the bit 2 of the processor status register. This flag inhibits all interrupts except the BRK instruction interrupt.
By setting the interrupt inhibit flag to "1" , all interrupts are inhibited. By clearing it to "0" , the interrupt is allowed. To set and reset interrupt inhibit flag, SEI, CLI instructions are used respectively.
When an interrupt is accepted, and program flow goes to the interrupt service routine, the interrupt inhibit flag is automatically set and multiple interrupts are inhibited. In order to allow multiple interrupts, this flag must be cleared by the CLI instruction within the interrupt routine.
Each of these control flags and control bits is independent and have no effect on the other two flags. When the interrupt request and the interrupt enable bits are "1" , and the interrupt inhibit flag is "0" , the corresponding interrupt is generated.

2.5.3 Interrupt sequence

When an interrupt is accepted, the on-going process is stopped and the interrupt service routine is executed. After the interrupt service routine is completed it is necessary to restore everything to the state before the interrupt occurred.

In M37450, as soon as an interrupt is accepted, the contents of the program counter and the processor status register are saved in the stack area. At the same time, the content of the vector address corresponding to the accepted interrupt, which is in the interrupt vector table, enters into the program counter and interrupt service routine is executed.

In the interrupt service routine, the corresponding interrupt request bit is cleared to "0" and interrupt inhibit flag becomes "1" , thereby inhibiting multiple interrupts. (In order to allow multiple interrupts, the interrupt inhibit flag needs to be cleared by the CLI instruction.)

In order to execute the interrupt service routine, it is necessary to write the jump addresses (the first address of the interrupt service routine) in the vector table corresponding to each interrupt. The vector addresses of each interrupt are shown in the Table 2.5.1.

Figure 2.5.2 shows the change in the stack pointer and the program counter at the time when an interrupt is accepted.

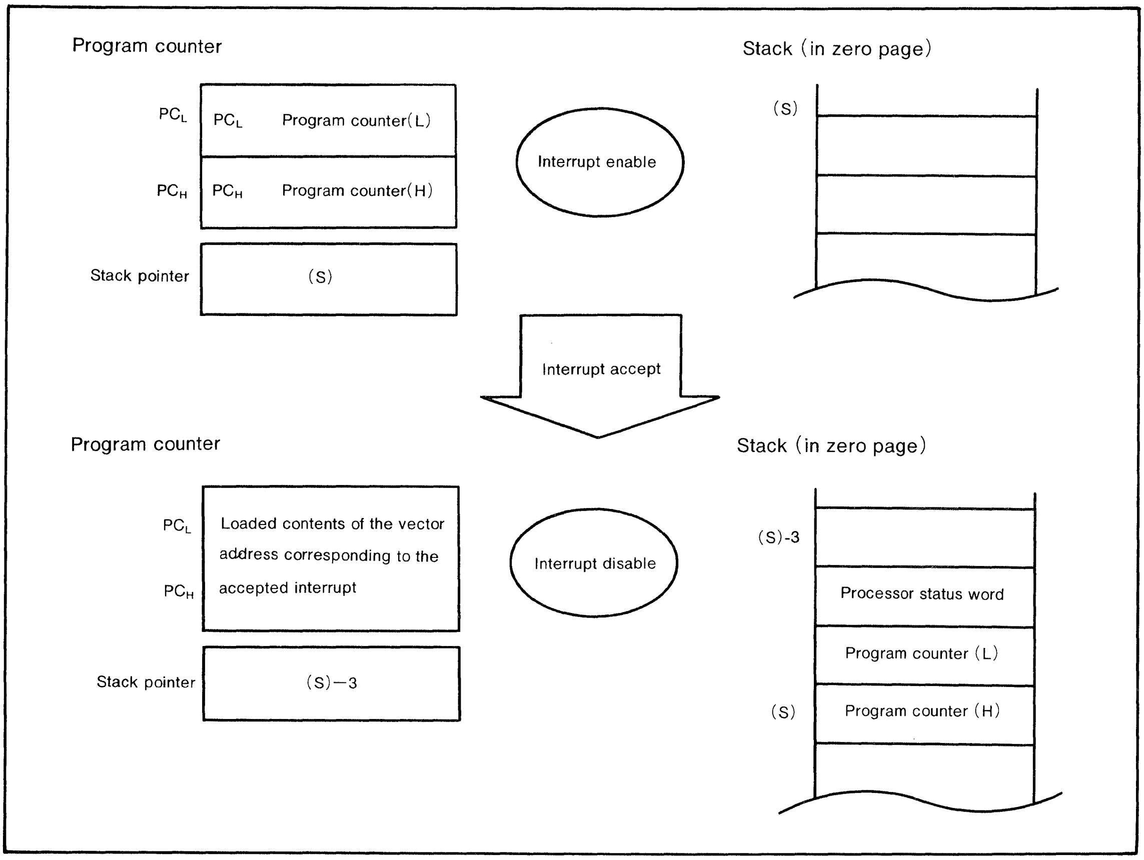
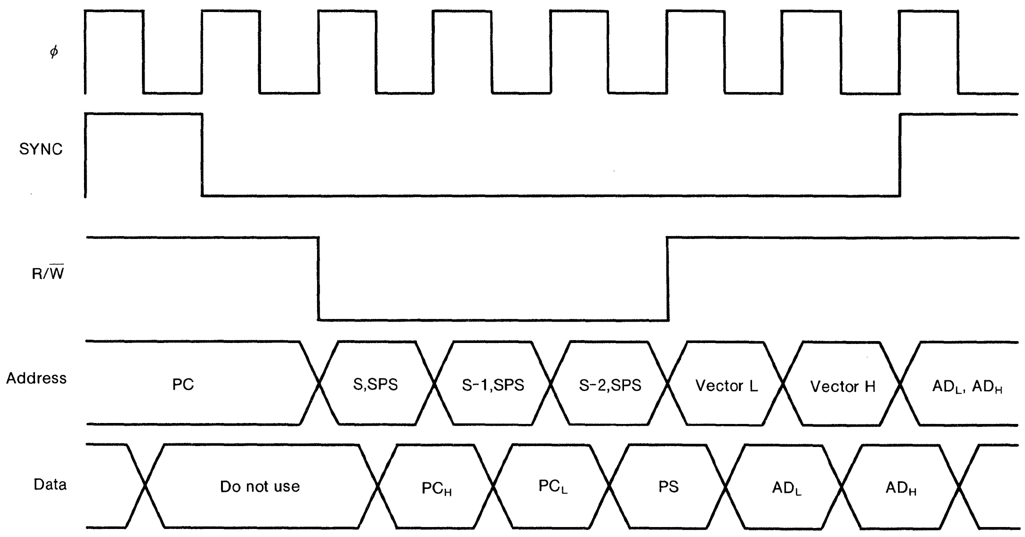


Fig. 2.5.2 Program counter and stack pointer change in interrupt sequence

2.5.4 Timing after interrupt

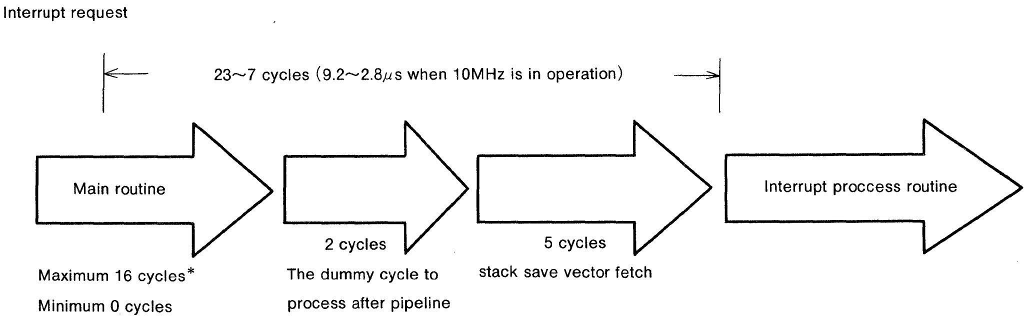
Figure 2.5.3 shows the timing sequence when an interrupt is accepted.
Figure 2.5.4 shows the processing time involved in the execution of interrupt service routine.



Note : Vector L and vector H are the vector address of each interrupt, and AD_H, AD_L are the contents of these vector address (jump addresses). See Table 2:5.1 for the vector address of each interrupt.

Fig. 2.5.3 Timing chart after interrupt

The interrupt service routine will start after the completion of the instruction being executed when the request occurs. The two conditions that allow for the interrupt to be accepted is the interrupt inhibit flag must be "0" and the interrupt enable bit must be "1" (excluding the BRK instruction interrupt) .



* At the DIV instruction

Fig. 2.5.4 Execution time prior to interrupt process routine

2.6 Timer

M37450 has three multi-functional 16-bit timers and one 8-bit timer for generating serial I/O bit rate (BRG). BRG will be explained under serial I/O section.
Each of the three 16-bit timers is independent. The following seven modes are program selectable.

- 16-bit timer mode (with timer latch)
- Event counter mode (event polarity programmable)
- Pulse output mode
- Pulse period measurement mode
- Pulse width measurement mode
- Programmable waveform generation mode
- Programmable one-shot generation mode

Figure 2.6.1 shows the block diagram of the timers.

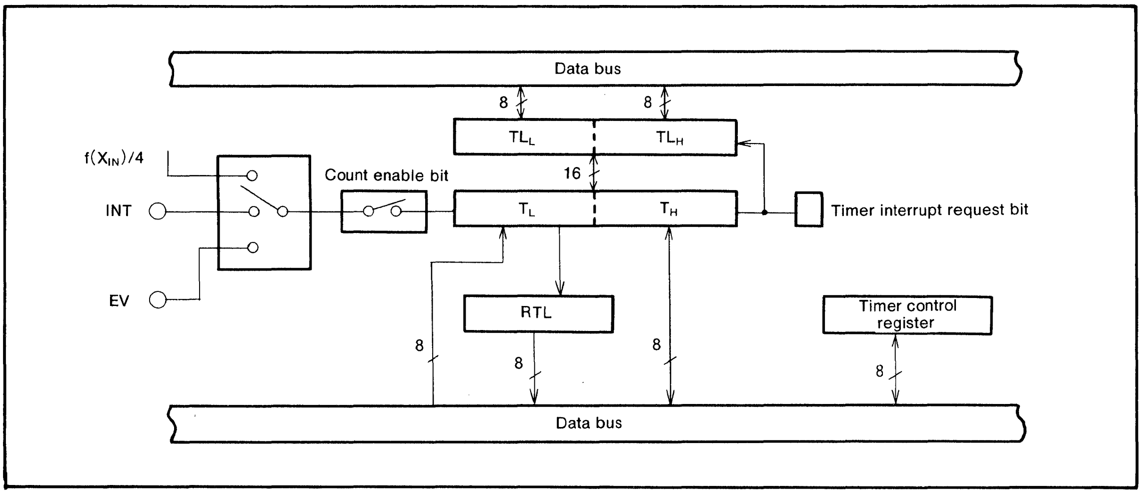


Fig. 2.6.1 Timer block diagram

2.6.1 Block explanations

(1) 16-bit timer (T)

The 16-bit timer T consists of two 8-bit timers, T_H and T_L. The timer operates as a down counter, and the timer interrupt request is generated at the next clock pulse after the timer reaches "0". At the same time the timer interrupt request flag is set to "1" , and since the contents of the timer latch is transferred into the timer except in pulse period measurement mode and in pulse width measurement mode.

In M37450, since the timer and the timer latch are independent, if the timer is set to a specific value the timer latch does not change. Data can be written into the timer at any time.

In order to write a value directly into the timer, the count enable bit must be set to disabled ("0") . First, the lower 8 bits (T_L) must be written into the timer, and then the upper 8 bits (T_H) must be written immediately following.

The timer counts down from its value until it underflows at the next count of which the interrupt request flag is set to "1".

When reading the value of the timer, since it is 16 bits, the upper 8 bits of the timer must be read (T_H) . At the same time the contents of the lower timer byte (T_L) are latched into the read-out timer latch (RTL) which can be read. Then if the lower timer byte (T_L) is read, the read-out timer latch (RTL) output the data. Therefore, we are reading both T_H and T_L at the same time.

(2) 16-bit timer latch (TL)

The 16-bit timer latch TL consists of two 8-bit timer latches TL_H or TL_L. The data in this latch is read into the timer every time the timer overflows except in pulse period measurement mode and in pulse width measurement mode. At any time data can be sent to the timer latch. There is no specific required order when writing a value into the timer latch. The division ratio is 1/ (N+1) , where N is the value loaded into the timer latch.

(3) Read-out timer latch (RTL)

The read-out timer latch RTL latches the contents of the T_L when the T_H is read out.

(4) Count enable bit (C)

The count enable bits, C1～C3, are bit 4～bit 6 of MISRG1. Each count bit C1～C3 corresponds to the specific timer as shown in the Figure 2.6.2.

"0" : Inhibit to input the count source to timer.

"1" : Cornect the count source to timer.

Immediately after reset these bits are set to "0" disable to count. If these bits, C1～C3 are set to "1" at the same time, timer 1～timer 3 will start counting simultaneously.

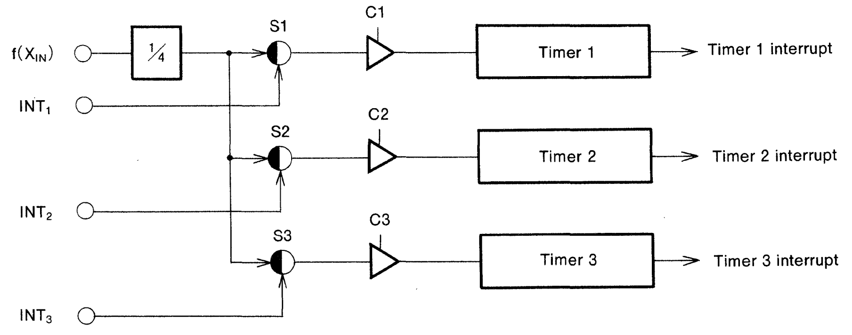


Fig. 2.6.2 Correspondence between count enable bit and count source selection bit

(5) Count source selection bit (S)
The count source selection bits, S1～S3 are bit 3 of the timer 1～timer 3 control registers respectively. The count source of the timer can be selected by this bit.

"0" : oscillation frequency f (X_{IN}) divided by 4
"1" : external input INT pin.

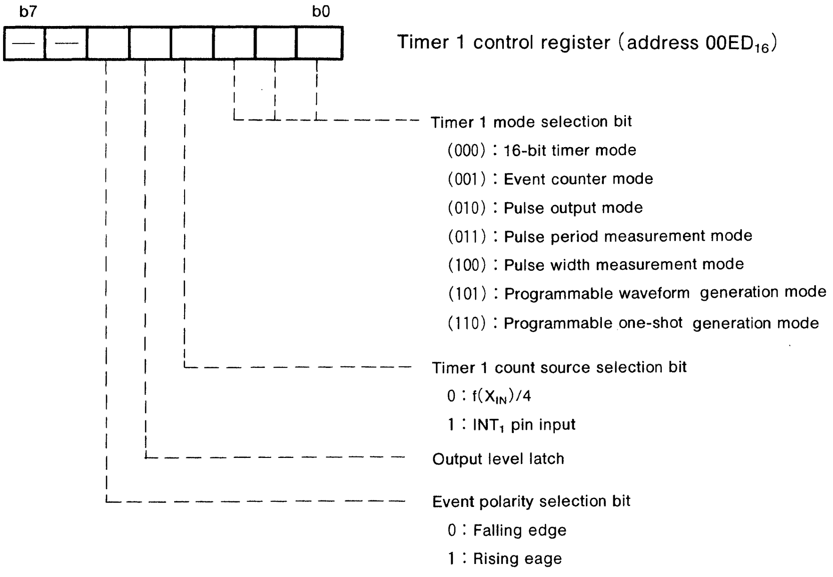
At 16-bit timer mode, event counter mode and programmable one-shot generation mode, this bit may be ignored.

(6) Timer mode selection bit (TMS)
The timer mode selection bits are bit 0～2 of timer 1～timer 3 control registers respectively. The correspondence between the timer mode selection bit and the timer mode is shown in Table 2.6.1.
However, the timer 1 will be forced to be 16-bit timer mode after the execution of STP instruction and returns to the original timer mode after re-start (return from the STP state).

Table 2.6.1 Relationship between TMS and the timer mode

TMS			Selection mode
b2	b1	b0	
0	0	0	16-bit timer mode
0	0	1	Event counter mode
0	1	0	Pulse output mode
0	1	1	Pulse period measurement mode
1	0	0	Pulse width measurement mode
1	0	1	Programmable wave form generation mode
1	1	0	Programmable one-shot generation mode
1	1	1	Selection disable

Table 2.6.2 Timer related bit structure



Timer 1 control register (address $00ED_{16}$)

Timer 1 mode selection bit
(000) : 16-bit timer mode
(001) : Event counter mode
(010) : Pulse output mode
(011) : Pulse period measurement mode
(100) : Pulse width measurement mode
(101) : Programmable waveform generation mode
(110) : Programmable one-shot generation mode

Timer 1 count source selection bit
0 : $f(X_{IN})/4$
1 : INT_1 pin input

Output level latch

Event polarity selection bit
0 : Falling edge
1 : Rising eage

Note : Timer 2 and timer 3 control registers consist of the same bit structure as timer 1 control 1 register.

2.6.2 16-bit timer mode (TMS"000")

(1) Summary

Interrupt request is generated each time when the timer overflows.

Resolution——16-bit
Division ratio—1/(N+1)
When the value N (N=0~65535) is written into the timer latch.
Count source—Oscillation frequency f (X_{IN}) divided by 4.

Figure 2.6.3 shows the structure of the timer mode.

(2) Explanation of the timer mode operation

The timer starts counting by setting the count enable bit to "1" . The timer counts down from its value until it underflows at which point the interrupt request flag is set to "1" . At the same time the content of the timer latch is loaded into the timer.

The count source for the timer is connected to the oscillation clock f(X_{IN}) divided by 4 independent of the count source selection bit (S) .

If the timer register is set to a value N, the timer counts N times and on the next clock cycle loads the value of the timer latch into the timer.

(3) Note

The timer and the timer latch are each independent therefore, necessary to initially set a value in the timer register and the timer latch.

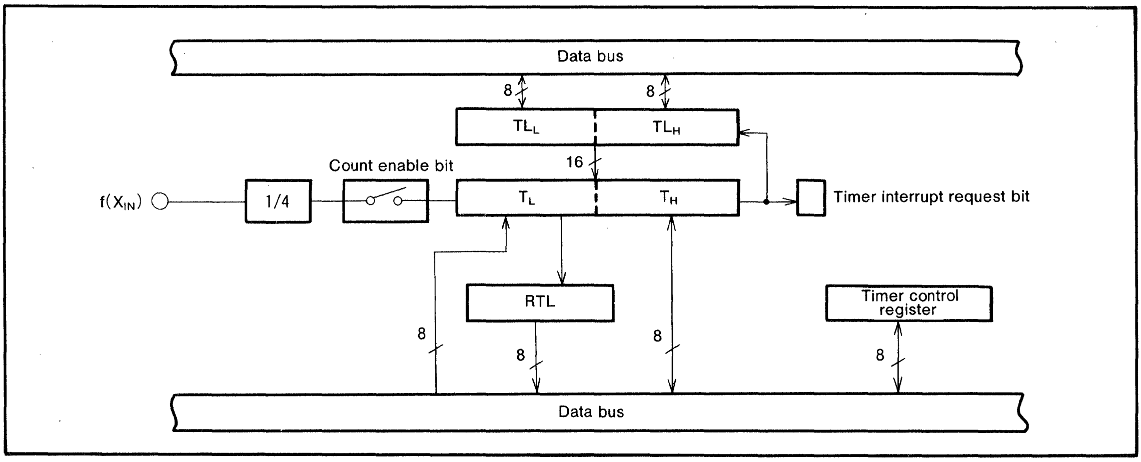


Fig. 2.6.3 16-bit timer mode structure

2.6.3 Event counter mode (TMS "001")

(1) Summary

In this mode, the timer counts an extenal event input from EV port as the count source.

Resolution——16-bit
Division ratio—1/(N+1) when the value N (N=0~65535) is written into the timer latch.
Count source—External event input from EV port.
(Event polarity can be selected by program.)

Maximum event input frequency— 1MHz at $f(X_{IN})=10MHz$

Figure 2.6.4 shows the structure of event counter mode.

(2) Explanation of the event counter mode

Event counter mode operates in the same way as the 16-bit timer mode except it counts the external event input from EV port. The count source selection bit (S) does not effect this mode. Event polarity is selected by bit 5 of each timer control register.

(3) Note

The pulse widths of the event input signal period (either "H" or "L") must be $(4/f(X_{IN})+100ns)$ or more (where $f(X_{IN})$ is the oscillation frequency) .

The event input from EV port is input into the timer synchronously with the internal clock, except the timers 2 and 3 count asynchronously while the STP instruction is executed.

Do not change the input status immediately before stop condition. Also, depending on the condition of EV port immediately before the change into stop condition and the value of the event polarity, the following lags in the count value occur as shown in Table 2.6.3.

Table 2.6.3 Change in count value depending on the event polarity and the EV port

Event polarity	EV port	Change in count value
0	0	1 count down immediately after execution of STP instruction
0	1	No change in the count value after execution of STP instruction
1	0	No change in the count value after execution of STP instruction
1	1	1 count down immediately after execution of STP instruction

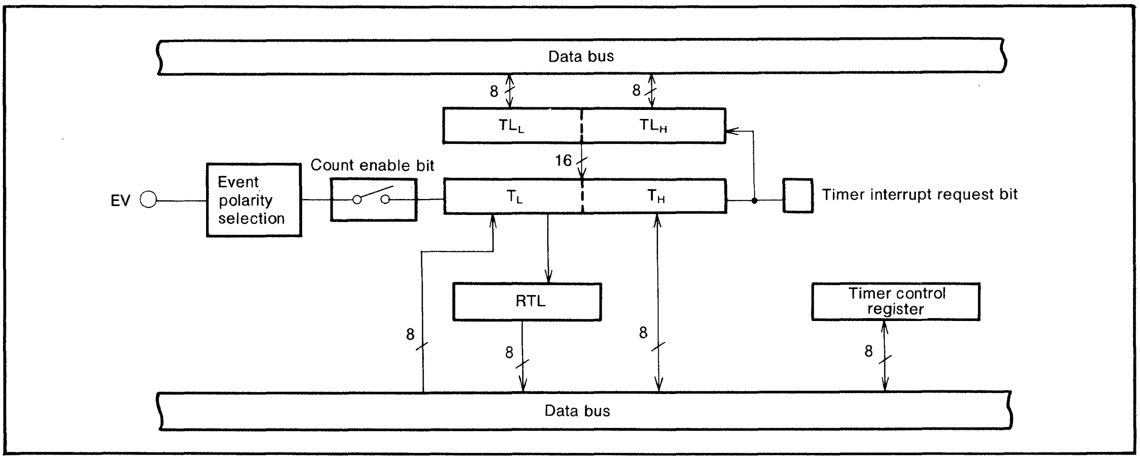


Fig. 2.6.4 Event counter mode structure

2.6.4 Pulse output mode (TMS "010")

(1) Summary

This mode outputs a variable period pulse on the EV port.

Resolution——16-bit
Count source—Selected by the count source selection bit.

Figure 2.6.5 shows the structure of pulse output mode.

(2) Explanation of pulse output mode operation

In this mode, a pulse width 50% cycle duty is output to the EV port and reverse the output each time the timer underflows. This creates a pulse period which is two times that of the timer underflow cycle.

(3) Note

EV port is initialized to "0" when "010" is written in TMS.

The pulse widths, (either "H" or "L") must be (6/f(X_{IN})+100ns) or more (where f(X_{IN}) is the oscillation frequency).

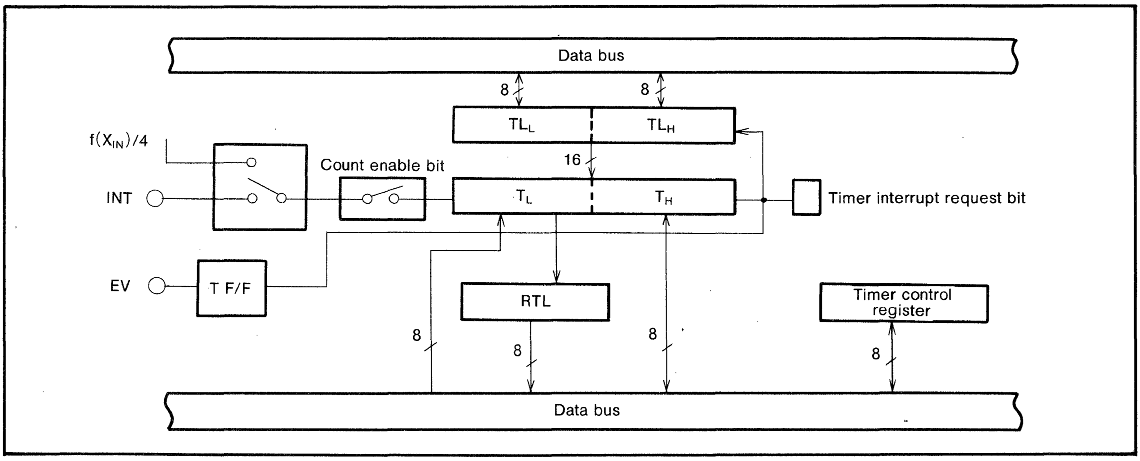


Fig. 2.6.5 Pulse output Mode block diagram

2.6.5 Pulse period measurement mode (TMS "011")

(1) Summary
This mode measures the period of the event wave form input from EV port.

Resolution——16-bit
Count source—Selected by the count source selection bit.

Figure 2.6.6 shows the structure of pulse period measurement mode.

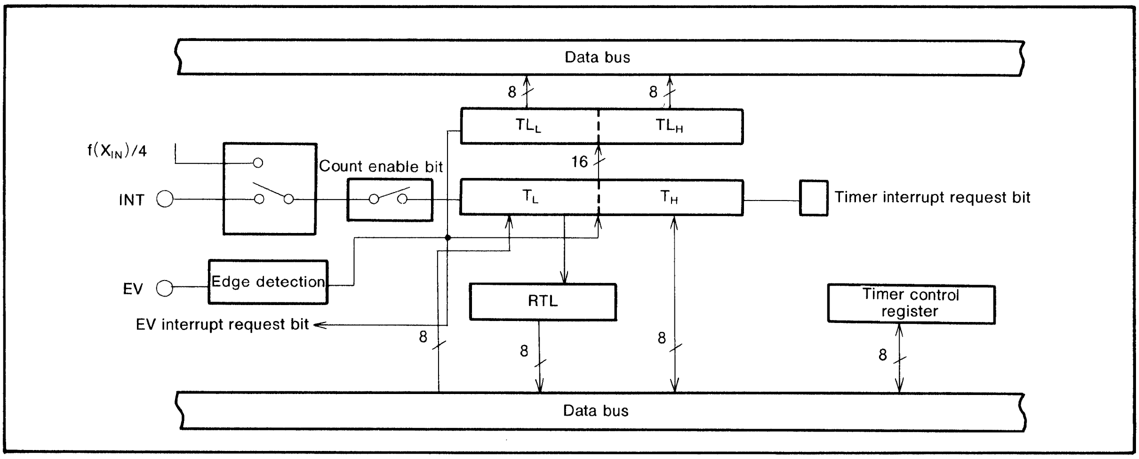


Fig. 2.6.6 Pulse period measurement mode structure

(2) Explanation of the pulse period measurement mode operation
The measured pulse period can be selected either from rising edge to rising edge of the event waveform or from falling edge to falling edge by the EV polarity selection bit (EP) . In this mode, event interrupt request is generated at the valid edge of the event waveform. At this time the value which is 1's complement of the timer is loaded into the timer latch, the contents of the timer is set to "$FFFF_{16}$" and the counting continues. Also when timer underflows the counting starts from "$FFFF_{16}$" instead of the value read from timer latch.

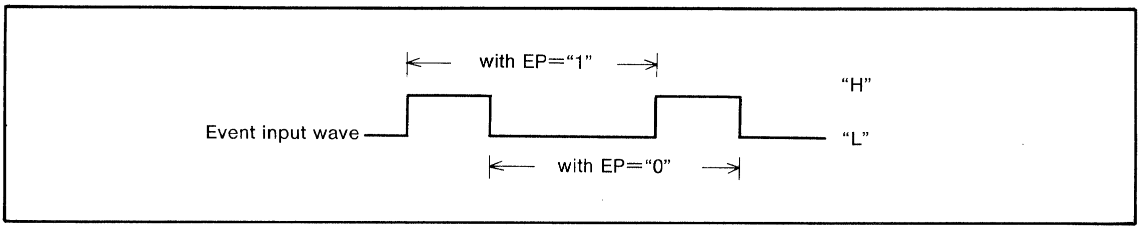


Fig. 2.6.7 Measurement area selection by event polarity

(3) Note
EV interrupt cannot be accepted during the STP mode. Also during pulse period measurement mode, data should not be written into the timer latch. The pulse widths of input pulse from EV_{pin} (either "H" or "L") must be (4/f(X_{IN})+100ns) or more. Also the pulse widths of count source input signal from INT pin must be (6/f(X_{IN})+100ns) or more (where f(X_{IN}) is the oscillation frequency).

2.6.6 Pulse width measurement mode (TMS "100")

(1) Summary

This mode measures the pulse width of the event waveform input to the EV port.

Resolution——16-bit
Count source—Selected by the count source selection bit.

Figure 2.6.8 shows the structure of the pulse width measurement mode.

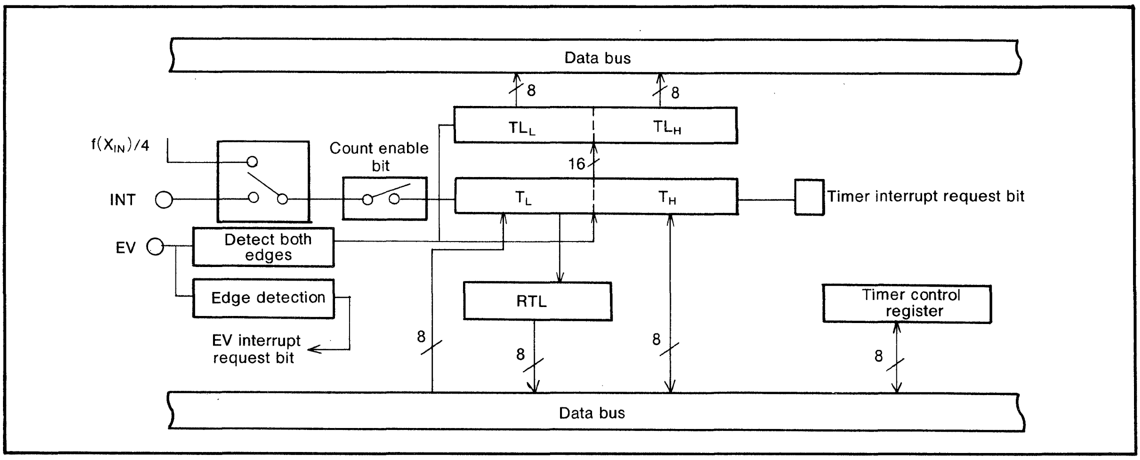


Fig. 2.6.8 Pulse width measurement mode structure

(2) Explanation of pulse width measurement mode operation

The measured pulse width can be selected as either the width during the "H" period or the "L" period by the EV polarity selection bit. If EV polarity is selected as falling edge (EP="0") , the width of the "H" period is measured; if it is selected as rising edge (EP="1") , the width of the "L" period, is measured. The function of the timer in this mode is the same as the pulse period measurement mode, but timer is set to "$FFFF_{16}$" at both edges of event waveform.

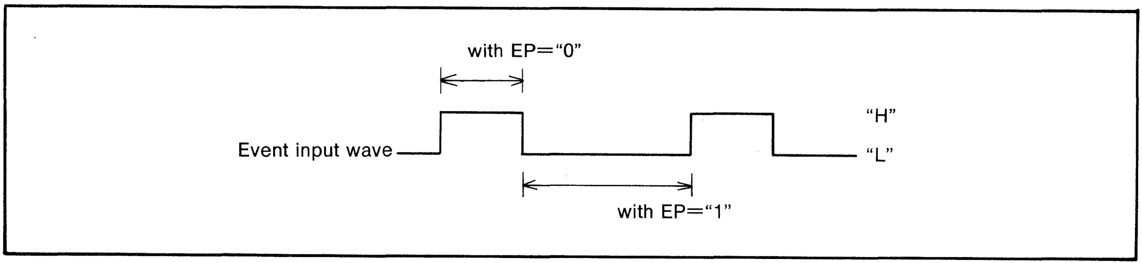


Fig. 2.6.9 Measurement area selection by event polarity

(3) Note

EV interrupt cannot be accepted during the STP mode. Also during pulse width measurement mode, data should not be written into the timer latch. The pulse widths of input pulse from EV pin (either "H" or "L") must be ($4/f(X_{IN})$+100ns) or more. Also the pulse width of count source input signal from INT pin must be ($6/f(X_{IN})$+100ns) or more (where $f(X_{IN})$ is the oscillation frequency).

2.6.7 Programmable waveform generating mode (TMS "101")

(1) Summary
This mode output to the EV port the contents of the output level latch (OLL), bit 4 of the timer control register, whenever the timer underflows.

Resolution——16-bit
Count source—Selected by the count source selection bit.

Figure 2.6.10 shows the structure of the programmable waveform generating mode.

(2) Explanation of programmable waveform generating mode operation
The operation is the same as the pulse output mode except it outputs the content of the output level latch (OLL) to the EV port whenever the timer underflows. A programmable waveform can be generated after the timer underflows by changing the value of the output level latch and the timer latch.

(3) Note
EV port is initialized to "1" when "101" is written in TMS. The pulse width of count source input signal from INT pin must be $(6/f(X_{IN})+100ns)$ or more (where $f(X_{IN})$ is the oscillation frequency).

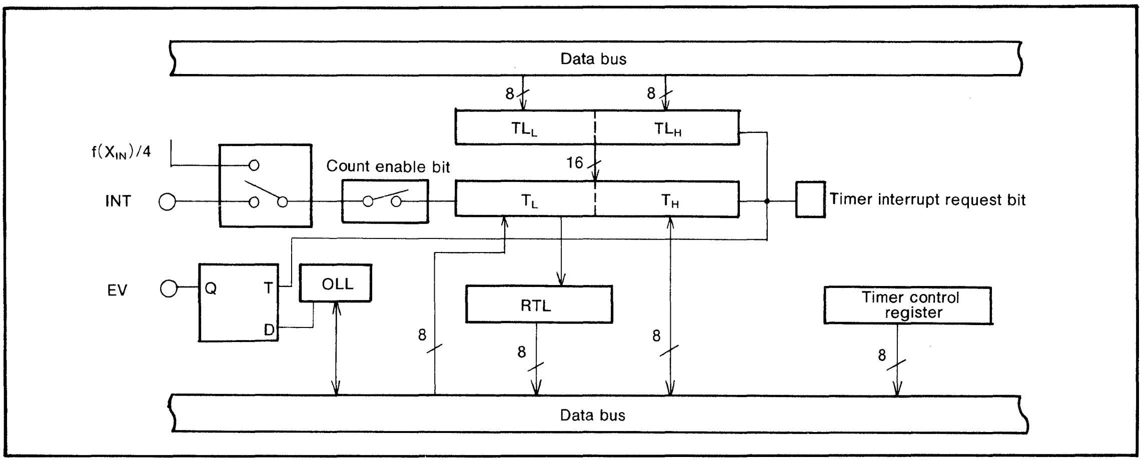


Fig. 2.6.10 Programmable wave form generating mode structure

2.6.8 Programmable one-shot generation mode (TMS "110")

(1) Summary

Resolution——16-bit
Count source—Oscillation frequency $f(X_{IN})$ divided by 4

Figure 2.6.11 shows the block diagram to the programmable one-shot generation mode.

(2) Explanation of the programmable one-shot generation mode operation
In this mode, a trigger signal from the INT pin leads the contents of the timer latch into the timer and starts the timer counting. At the same time as the trigger, the EV port becomes "H". When the timer underflows, this port becomes "L". The count source is the oscillation frequency divided by 4, regardless of the value of S. The maximum one-shot width is 26ms. ($f(X_{IN})=10MHz$). The polarity of the trigger is programmable.

(3) Note
When "110" is written in TMS, EV pin is initialized to "0". The rising output from EV_{pin} is synchronized with the trigger input signal of INT pin. The pulse width of count source input signal from INT pin must be $(6/f(X_{IN})+100ns)$ or more (where $f(X_{IN})$ is the oscillation frequency).

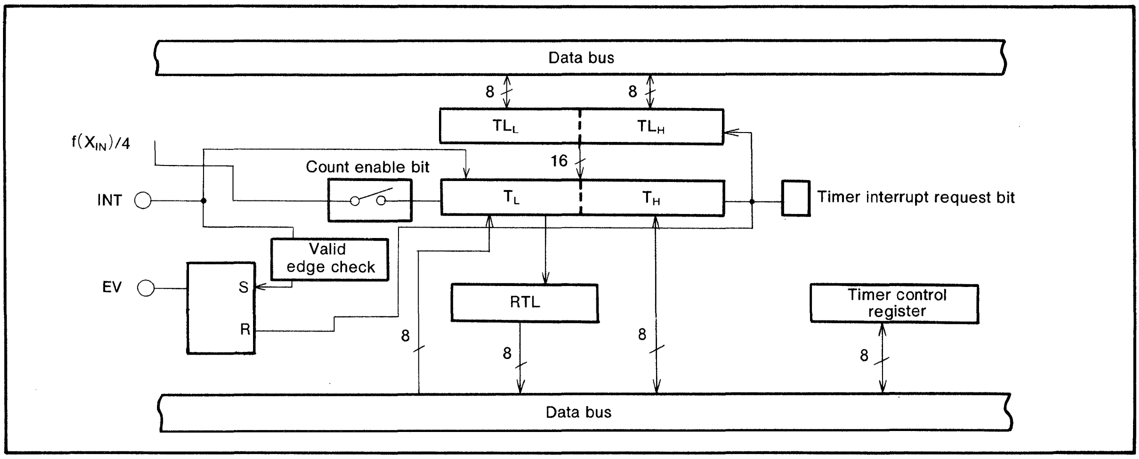


Fig. 2.6.11 Programmable one-shot generation mode structure

2.7 Serial I/O

M37450 has a built-in serial interface which can operate in either clock synchronous or asynchronous (UART) mode. There is also a built-in timer (BRG) for generating the baud rate specifically for the serial I/O function.

2.7.1 Baud rate generator (BRG)

BRG is an 8-bit timer with a timer latch used for generating baud rate of the serial I/O. Bit 0 of the serial I/O control register selects whether $f(X_{IN})/2$ or $f(X_{IN})/8$ is used as the clock source for the BRG generator.

BRG division ratio——1/(n+1)
(n : the value programmed in address $00EA_{16}$)

Figure 2.7.1 shows the structure of BRG.

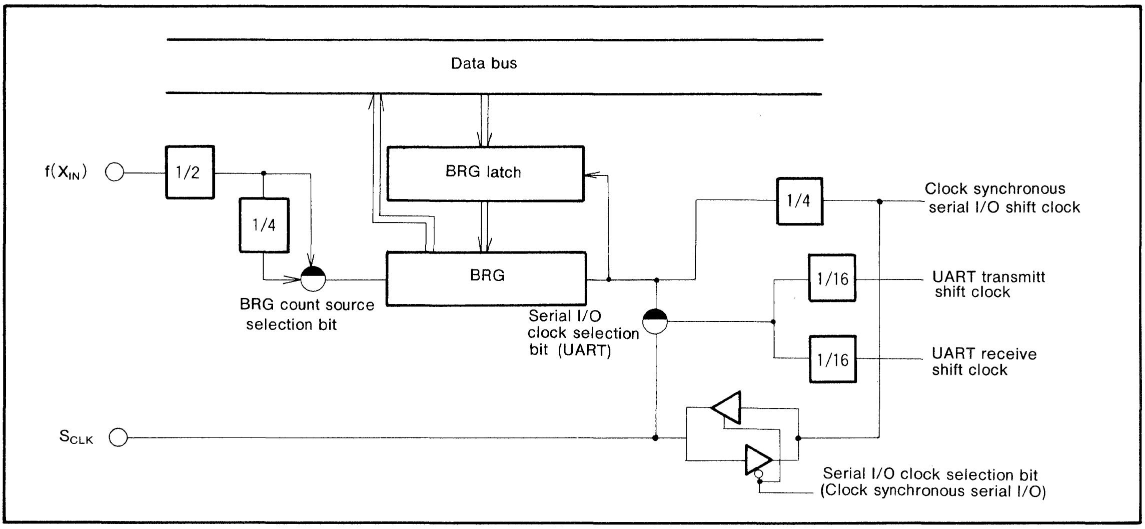

Data bus
f(X_IN)
1/2
1/4
BRG latch
BRG
BRG count source selection bit
Serial I/O clock selection bit (UART)
1/4
Clock synchronous serial I/O shift clock
1/16
UART transmitt shift clock
1/16
UART receive shift clock
S_CLK
Serial I/O clock selection bit (Clock synchronous serial I/O)

Fig. 2.7.1 Baud rate generator block diagram

2.7.2 Clock synchronous serial I/O

In M37450, the clock synchronous serial I/O mode can be selected by setting the serial I/O mode selection bit (bit 6 of the serial I/O control register)to "1".

(1) Summary

M37450 has a built-in clock synchronous serial I/O with the following features:

Protocol data communication—LSB first half duplex or full duplex
Synchronous clock—BRG/4 or external clock input from S_{CLK} pin
(Max external input clock should be 1.25MHz)

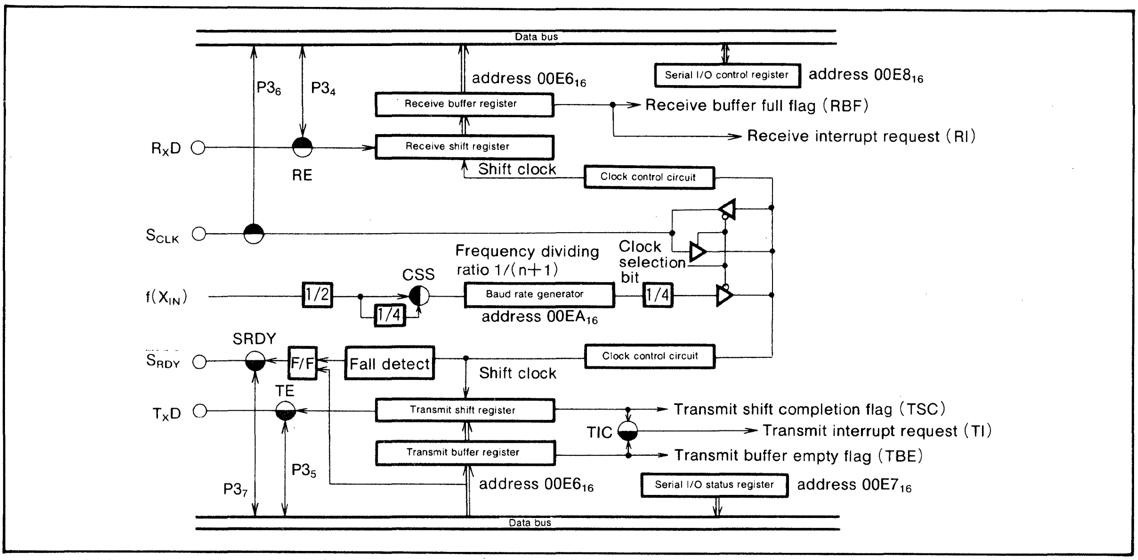
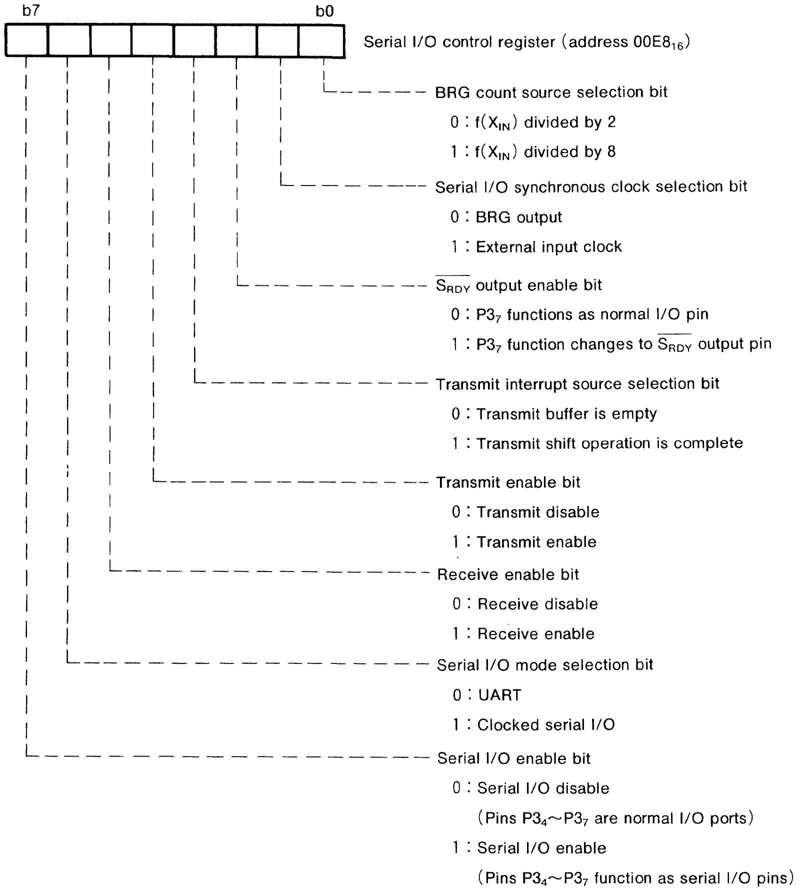


Fig. 2.7.2 Clock synchronous serial I/O block diagram

(2) Block explanations

[Serial I/O control register]
Serial I/O control register consists of 8-bit and Table 2.7.1 shows the bit structure of the serial I/O control register.
All bits of the serial I/O control register can be read or written to the program. At reset this register is cleared to "00_{16}" disabiling the serial I/O function.

Table 2.7.1 Bit structure of serial I/O control register



b7 b0

Serial I/O control register (address $00E8_{16}$)

BRG count source selection bit
0 : f(X_{IN}) divided by 2
1 : f(X_{IN}) divided by 8

Serial I/O synchronous clock selection bit
0 : BRG output
1 : External input clock

$\overline{S_{RDY}}$ output enable bit
0 : $P3_7$ functions as normal I/O pin
1 : $P3_7$ function changes to $\overline{S_{RDY}}$ output pin

Transmit interrupt source selection bit
0 : Transmit buffer is empty
1 : Transmit shift operation is complete

Transmit enable bit
0 : Transmit disable
1 : Transmit enable

Receive enable bit
0 : Receive disable
1 : Receive enable

Serial I/O mode selection bit
0 : UART
1 : Clocked serial I/O

Serial I/O enable bit
0 : Serial I/O disable
(Pins $P3_4$~$P3_7$ are normal I/O ports)
1 : Serial I/O enable
(Pins $P3_4$~$P3_7$ function as serial I/O pins)

● Bit 0 : BRG count source selection bit
This bit selects the input clock for the baud rate generator between f(X_{IN}) divided by 2 and f(X_{IN}) divided by 8.

● Bit 1 : Serial I/O synchronous clock selection bit
This bit selects whether BRG output or the external input clock issued as the clock. When an external input clock is selected, a clock of 1.25MHz or below must be used. When this bit is set to "0" in the synchronous serial I/O mode, the BRG output divided by 4 is selected as the synchronous clock.
When this bit is set to "0" in UART mode, the BRG output divided by 16 is selected as the synchronous clock.

- Bit 2 : $\overline{S_{RDY}}$ output enable bit
  This bit controls $\overline{S_{RDY}}$ output. When $\overline{S_{RDY}}$ output is selected at serial I/O enable and in clock synchronous serial I/O mode, port $P3_7$ becomes output pin and output $\overline{S_{RDY}}$ signal.
  When UART function is selected, port $P3_7$ works as a normal input/output port regardless of the value of this bit.

- Bit 3 : Transmit interrupt source selection bit (TIC)
  This bit selects the interrupt trigger point in the transmission process. If the bit is set to "0" , the transmit interrupt occurs when the transmit buffer becomes empty. If the bit is set to "1" the interrupt occurs when the transmit shift register has completed shifting.

- Bit 4 : Transmit enable bit (TE)
  This bit enables the transmission operation of the serial I/O.

- Bit 5 : Receive enable bit (RE)
  This bit enables the receive operation of the serial I/O.

- Bit 6 : Serial I/O mode selection bit
  This bit selects the clock asynchronous or clock synchronous serial I/O mode.

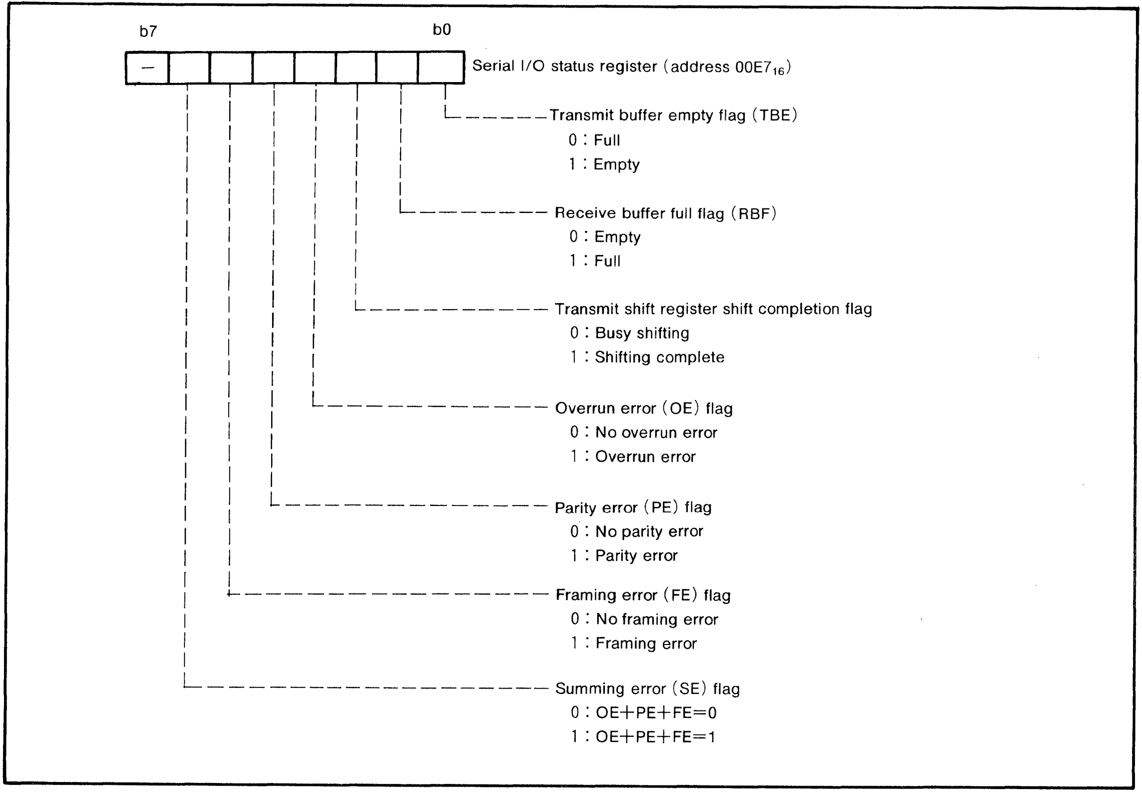
- Bit 7 : Serial I/O enable bit
  This bit controls the serial I/O function. When serial I/O is enabled, the port $P3_4 \sim P3_7$ can be selected as serial I/O function pins. The changes in the pin functions when the serial I/O is enabled are shown below.

$P3_4 \sim P3_7$ pins change in functions

- $P3_4$ pin: Serial data input pin (RxD)
  This is used as the normal I/O port when receive enable bit is disable("0").
- $P3_5$ pin: Serial data output pin (TxD)
  This is used as the normal I/O port when transmit enable bit is disable("0").
- $P3_6$ pin: This is used as the normal I/O port when internal clock is selected at UART.
- $P3_7$ pin: Receive-enable signal output pin ($\overline{S_{RDY}}$)
  This is used as the normal I/O port when UART is selected or $\overline{S_{RDY}}$ output function is not selected.

[Serial I/O status register]
Serial I/O status register is the read-only register consisting of 7 flags which show the state of transmit and receive of the serial I/O.
At reset, this register is initialized to "00_{16}", but when the transmit enable bit of the serial I/O control register is "1" (enable), this register will be set to "05_{16}".

Table 2.7.2 Serial I/O status register



Serial I/O status register (address $00E7_{16}$)

Transmit buffer empty flag (TBE)
0 : Full
1 : Empty

Receive buffer full flag (RBF)
0 : Empty
1 : Full

Transmit shift register shift completion flag
0 : Busy shifting
1 : Shifting complete

Overrun error (OE) flag
0 : No overrun error
1 : Overrun error

Parity error (PE) flag
0 : No parity error
1 : Parity error

Framing error (FE) flag
0 : No framing error
1 : Framing error

Summing error (SE) flag
0 : OE+PE+FE=0
1 : OE+PE+FE=1

● Bit 0 : Transmit buffer empty flag (TBE)
This flag indicates the state of the transmit buffer. It is set to "1" when the data written in the transmit buffer is shifted to the transmit shift register. This flag is cleared automatically when the data is written into the transmit buffer. Also this flag is cleared when transmit enable bit is set to "0".

● Bit 1 : Receive buffer full flag (RBF)
This flag shows the state of the receive buffer. It is set to "1" when all bits are completely shifted from the receive shift register to the receive buffer. This flag is cleared automatically when the receive buffer is read out. Also this flag is cleared when receive enable bit is set to "0".

● Bit 2 : Transmit shift register shift completion flag (TSC)
This bit is cleared to "0" when the data in the transmission buffer register is transferred to the transmit shift register and set to "1" when data shift completes. It is also set to "1" when TE=0.

● Bit 3 : Overrun error flag (OE)
When continuously receiving serial data, this bit is set when the next data fill the receive shift register before the data in the receive buffer register has been read.

● Bit 4 : Parity error flag (PE)
When receiving serial data with parity, this bit is set to "1" if the parity of the received data differs from the specified parity

● Bit 5 : Framing error flag (FE)
This bit is set to "1" when the stop bit of the receive data is not detected at the setting timing. This detects the only first bit, and does not check the second one.

● Bit 6 : Summing error flag (SE)
This bit is set when either overrun, a parity, or a framing error occurs.
Tests for these errors are performed as soon as the data is transferred from the receive shift register to the receive buffer register and at the same time the receive buffer full flag is set. The error flags (OE, PE, FE and SE) are cleared when any data is written in the serial I/O status register. Also, all status flags including error flags are cleared when SIOE=0.

[Receive shift register (RSR) and receive buffer (RB)]
The receive shift register is the serial-parallel conversion register for data receive only. It consists of 8-bit and cannot be accessed directly.
When all bits are received in the receive shift register, they are shifted to the receive buffer register(RB). Since there are a receive shift register and a receive buffer register, the M37450 can receive continuously.

[Receive-ready signal ($\overline{S_{RDY}}$) generating circuit]
M37450 has the capability to show outside that the serial I/O is receive-ready state by using the $\overline{S_{RDY}}$ signal. $\overline{S_{RDY}}$ signal is enabled by setting bit 2 of the serial I/O control register ($\overline{S_{RDY}}$ output enable bit) to "1".
If the $\overline{S_{RDY}}$ output enable bit is "1" and a signal is written to the transmit/receive buffer register, the $\overline{S_{RDY}}$ signal goes "L". At this point the CPU is in the"ready" condition. On the falling edge of the first shift clock, the $\overline{S_{RDY}}$ signal returns to "H" state. The transmit enable bit must be set to "1" when $\overline{S_{RDY}}$ signal is used. This signal is not used in UART mode.

[Transmit shift regester (TSR) and transmit buffer register(TB)]
The transmit buffer register(TB) consists of 8-bit. When the data are written in this buffer register, they are transferred to the transmit shift register (TSR) and output at the TxD pin from the least significant bit of the data.
Once the data in the transmit buffer is transferred to the transmit shift register, the transmit buffer can hold the next data to be transmitted during data transmission.

(3) Serial data receive operation (When the external clock is selected)
In order to set-up the serial I/O receive mode, the serial I/O control register must be initialized as follow : the serial I/O enable bit must be set, the external clock must be selected, receive enable bit, $\overline{S_{RDY}}$ and transmit enable bit must be set. If the $\overline{S_{RDY}}$ signal is not used or the internal clock is selected for the synchronous clock, it is not necessary to be set the transmission enable bit.
After clearing the receive interrupt request bit, the receive interrupt enable bit should be set. The $\overline{S_{RDY}}$ pin level changes from "H" to "L" by writing dummy data in the receive buffer register (00E6$_{16}$ address). This signal tells to the transmission side that the M37450 is ready to receive serial data. The transmission side detects the "L" signal and starts transmission.
The M37450 statrts to receive the serial data with the shift clock from the transmission side. The $\overline{S_{RDY}}$ pin return to "H" at the falling edge of the first shift clock.
The receive data input to the R_XD pin is read into the receive shift register bit by bit at the rising edge of the shift clock. The receive data is shifted from the most significant bit of the receive shift register and shifted one bit at a time.
After repeating this operation 8 times and receiving of 1 byte data in the receive shift register, the contents are shifted to the receive buffer. At this time, the receive buffer full flag is set and the receive interrupt request is generated. The receive buffer flag is cleared when the receive buffer register is read out.

The receive interrupt request flag is automatically cleared when the receive interrupt is accepted. This flag can be also cleared by software. When the new receive data is shifted into the receive shift register before reading the last data in the receive buffer register, the over-run flag of the serial I/O status register is set. In this case, the data is retained in the receive shift register and not sent to receive buffer register. And also the data in the receive shift register is not sent to receive buffer register when the data in the receive buffer register is read out.
The above is an example of serial I/O transfer using the external clock as, but the internal clock can also be used. When the external clock is selected since, the receive shift register continues to shift the data until the shift clock stops. Clock must stop after 8 shifts (the same thing applies in the case of the transmission). The clock starts by writing data into the transmit/receive buffer register when the internal clock is selected.

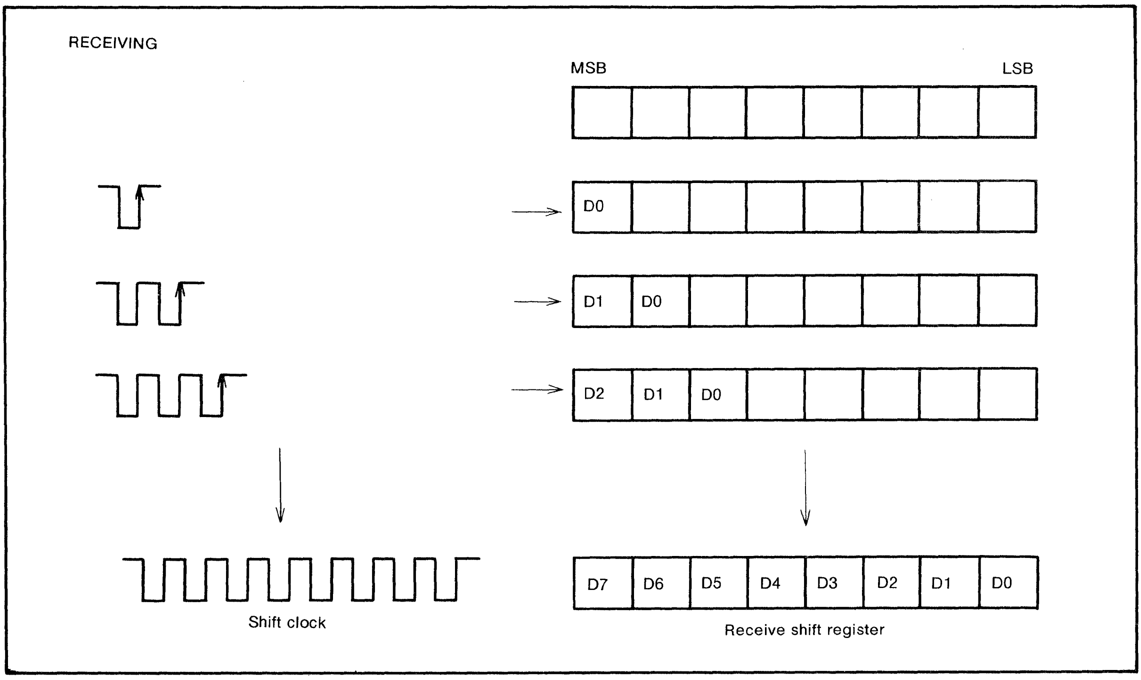


Fig. 2.7.3 Serial I/O receiving operation

(4) Serial data transmit operation (when the internal clock is selected)
In order to set-up the serial I/O transmit mode, the serial I/O control register must be initialized as follows: serial I/O enable bit must be set, internal clock must be selected, transmit enable bit must be enabled, and $\overline{S_{RDY}}$ output and receive enable bit must be disabled.
After clearing the transmit interrupt request bit, the transmit interrupt enable bit must be set and data written into the transmit buffer. When the data written in the transmit buffer is transferred to the transmit shift register, the transmit buffer empty flag (TBE) is set that the transmit buffer is empty. If the transmit interrupt source selection bit is "0" , the transmit interrupt is generated simultaneously. At the point when the transmission buffer empty flag is set, it is possible to write new transmit data into the transmit buffer. The transmit buffer empty flag is automatically cleared when the data is written in the transmit buffer.
When the internal clock is selected, the transmission clock, BRG output divided by 4, is input into the transmit shift register through S_{CLK} pin. Data transmission starts at writing data into the transmit buffer. At this time, eight-shift clock generates. As soon as the transmission starts, the transmit shift register shift completion flag in the serial I/O status register is cleared to "0". When the transmit interrupt source selection bit is "1", the transmit interrupt is generated at the completion of the transmission operation.

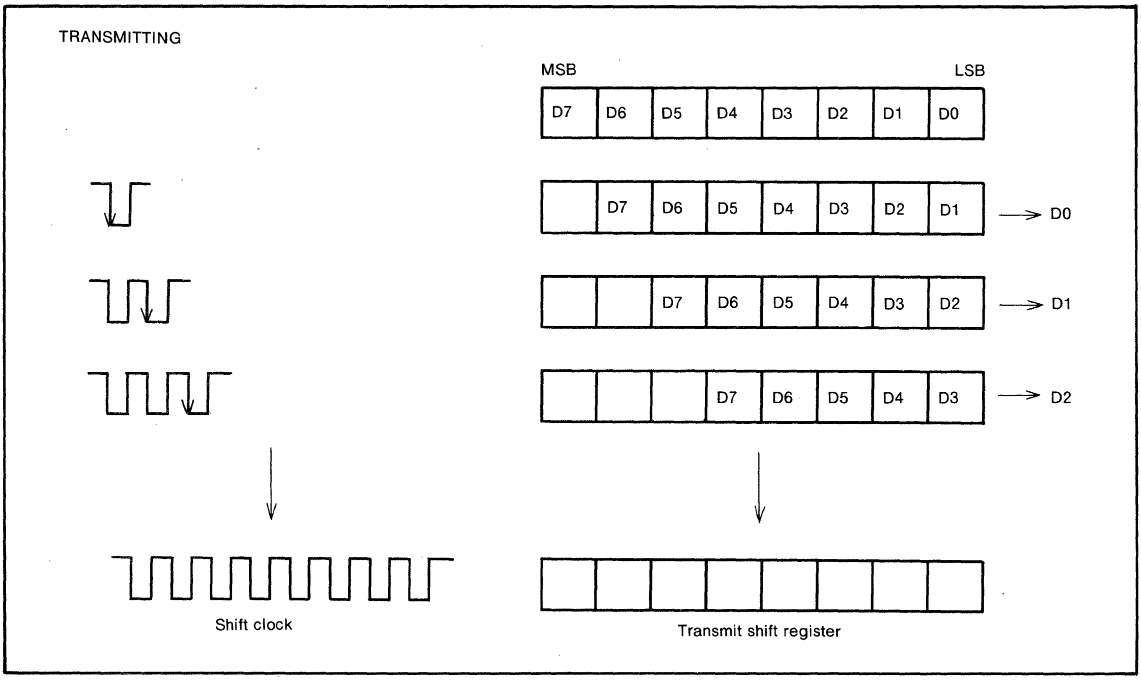
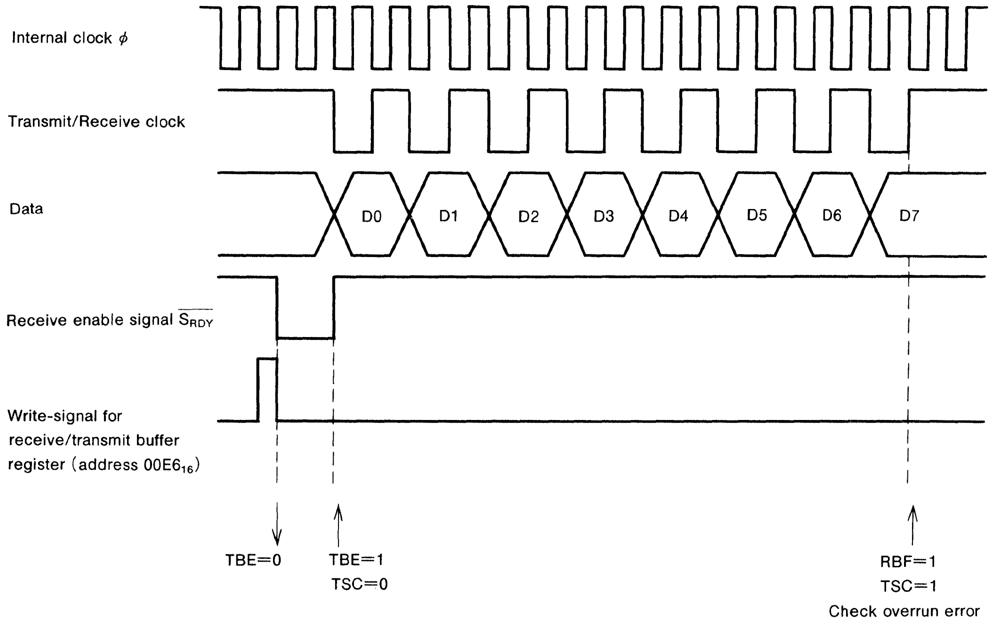


Fig. 2.7.4 Serial I/O transmission operation


Internal clock ϕ
Transmit/Receive clock
Data
D0
D1
D2
D3
D4
D5
D6
D7
Receive enable signal $\overline{S_{RDY}}$
Write-signal for receive/transmit buffer register (address $00E6_{16}$)
TBE=0
TBE=1
TSC=0
RBF=1
TSC=1
Check overrun error

Note 1 : The transmit interrupt (TI) selects TBE=1 when the transmit buffer is emptied (or TSC=1 when the transmit shift operation is complete) by the transmit interrupt source selection bit of the serial I/O control register.

Note 2 : When the data is written in the transmit buffer register and TSC=0, the transmission clock is continuously generated and the serial data is continuously output from the T_XD port.

Note 3 : The receive interrupt (RI) is set when the receive buffer full flag (RBF) is "1".

Note 4 : S_{CLK} generates after 0.5~1.5 shift clock cycle from writing transmit data to transmit buffer register.

Fig. 2.7.5 Clock synchronous serial I/O timing diagram

(5) Examples of clock synchronous serial I/O connection

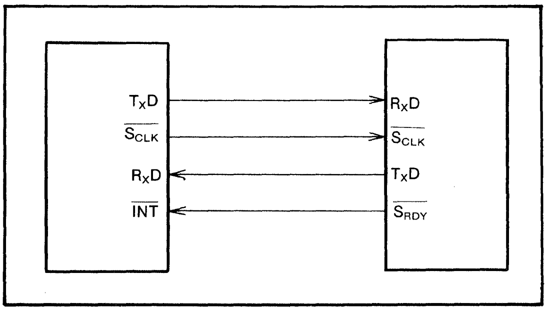


Fig. 2.7.6 Example of full duplex swap connection

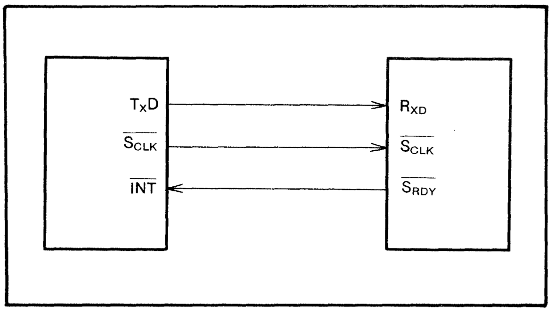


Fig. 2.7.7 Example of half duplex data communication connection

2.7.3 Clock asynchronous serial I/O (UART)

(1) Summary

M37450 has a built-in UART with the following features:

Baud rate selection		312.5 kbps(max.)
Data format selection	Start bit	1
	Data bit	7 or 8
	Parity bit	0 or 1
	Stop bit	1 or 2
Shift clock selection	External clock input from S_{CLK} or BRG/16	
Error detection	4 types (overrun, parity, framing, summing) Error or no-error can recognized by the summing error bit only.	

Figure 2.7.8 shows the structure of the UART-form serial I/O.

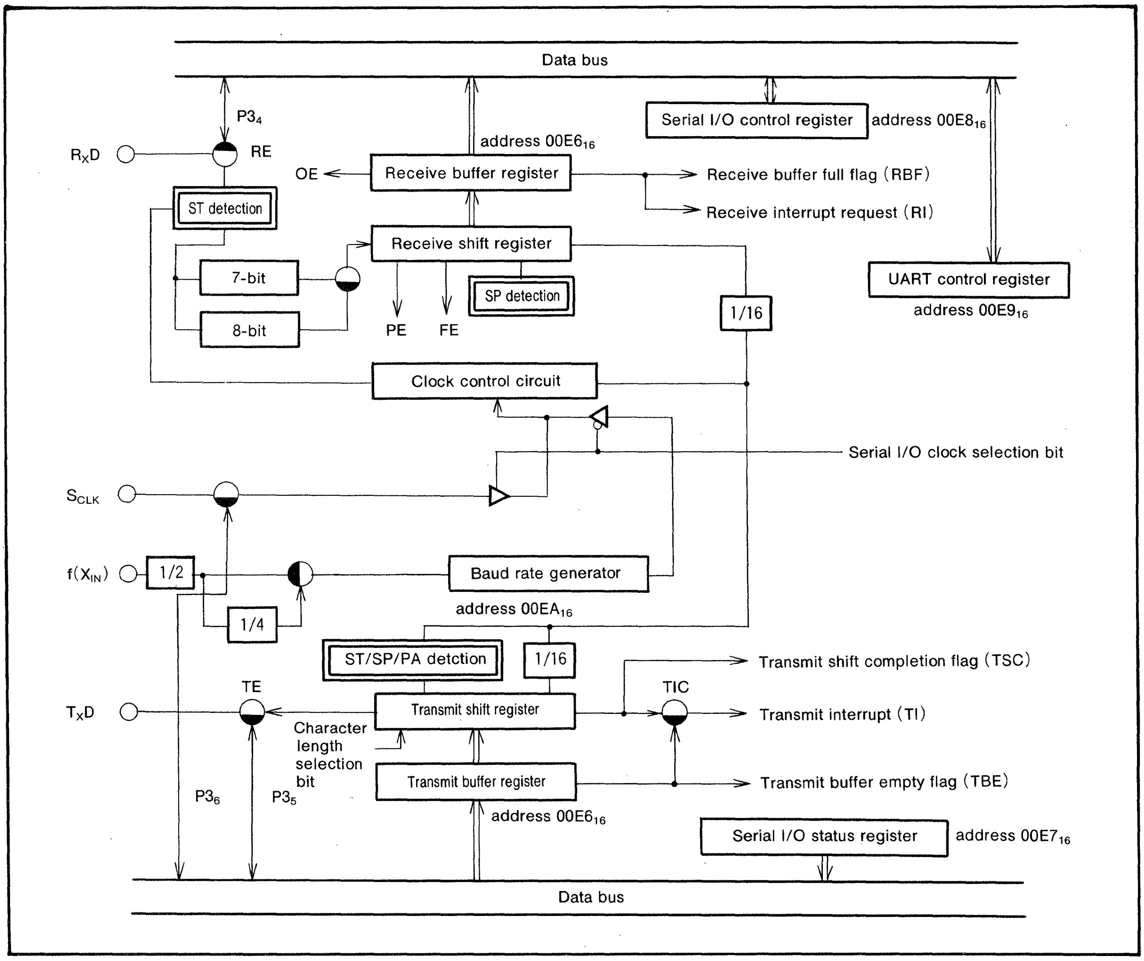


Fig. 2.7.8 UART block diagram

(2) Block explanations

[Serial I/O control register]

Serial I/O control register consists of 8-bit and Table 2.7.3 shows the bit structure of the serial I/O control register.

All bits of the serial I/O control register can be read or written to the program. At reset this register is cleared to "00_{16}" disabiling the serial I/O function.

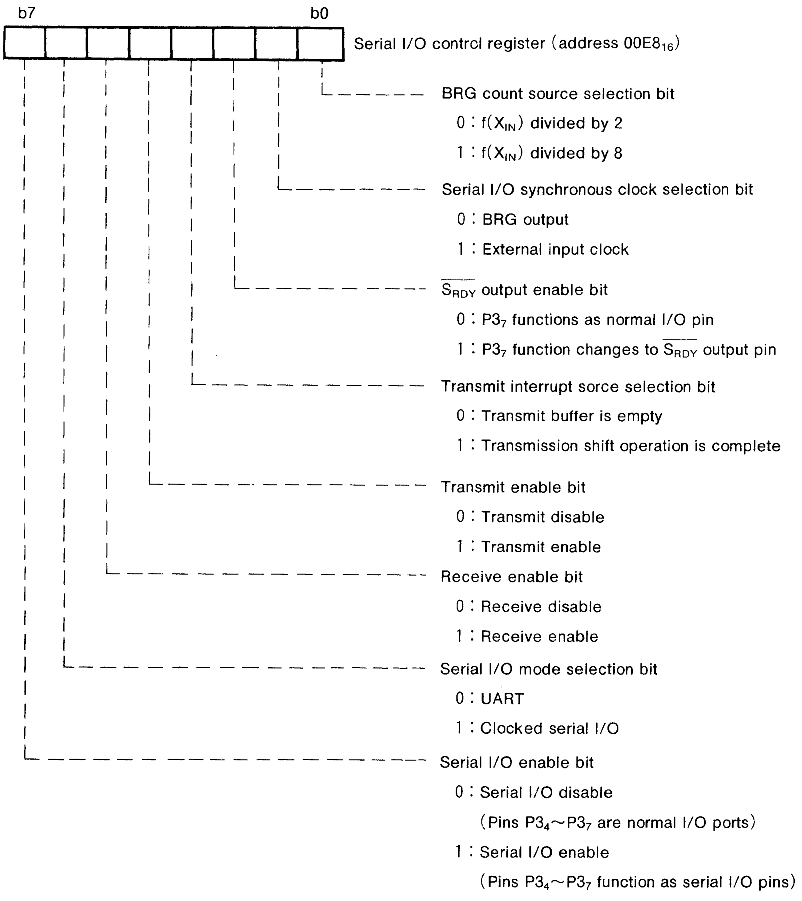
When the serial I/O enable bit is set to "1" , the functions of the ports $P3_4$~$P3_6$ are shown as follows:

$P3_4$ pin → Serial data input pin (RxD)
(When the receive function is not used, this pin can be used as a normal input/output port.)

$P3_5$ pin → Serial data output pin (TxD)
(When the transmit function is not used, this pin can be used as a normal input/output port.)

$P3_6$ pin → Clock input/output pin (S_{CLK})
(When the internal clock is used as the sampling clock, this pin can be used as a nomal input/output port.)

Table 2.7.3 Bit structure of serial I/O control register



Serial I/O control register (address $00E8_{16}$)

BRG count source selection bit
0 : $f(X_{IN})$ divided by 2
1 : $f(X_{IN})$ divided by 8

Serial I/O synchronous clock selection bit
0 : BRG output
1 : External input clock

$\overline{S_{RDY}}$ output enable bit
0 : $P3_7$ functions as normal I/O pin
1 : $P3_7$ function changes to $\overline{S_{RDY}}$ output pin

Transmit interrupt sorce selection bit
0 : Transmit buffer is empty
1 : Transmission shift operation is complete

Transmit enable bit
0 : Transmit disable
1 : Transmit enable

Receive enable bit
0 : Receive disable
1 : Receive enable

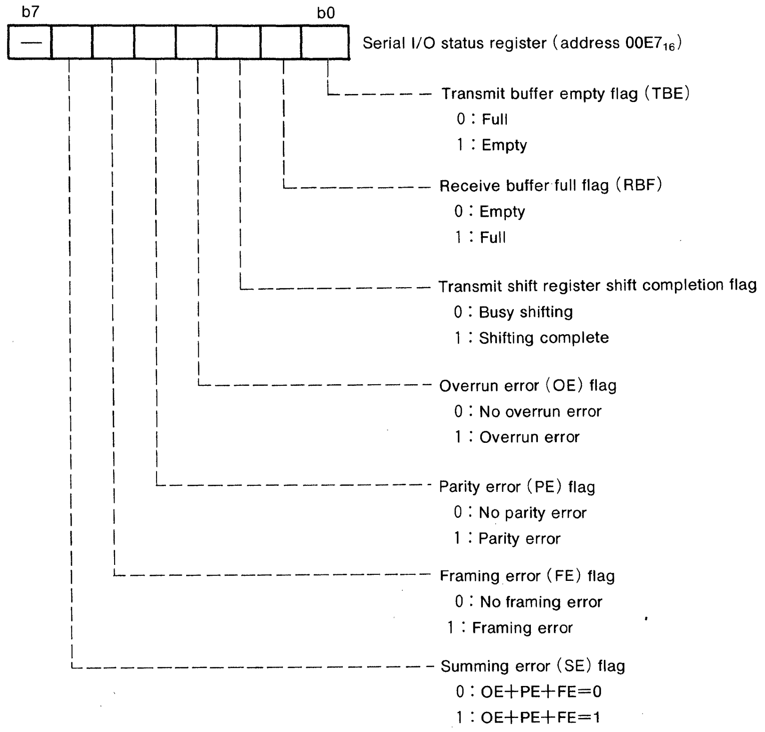
Serial I/O mode selection bit
0 : UART
1 : Clocked serial I/O

Serial I/O enable bit
0 : Serial I/O disable
(Pins $P3_4$~$P3_7$ are normal I/O ports)
1 : Serial I/O enable
(Pins $P3_4$~$P3_7$ function as serial I/O pins)

[Serial I/O status register]

The serial I/O status register is a read-only register consisting of 7 flags showing which show the state of the transmit/recive of the serial I/O.
At reset, this register is initialized to “00_{16}” , but is set to “05_{16}” when the transmit enable bit of the serial I/O control register is set to “1” (enable) .
The flags designated by bits 4~6 are only valid in the UART mode.

Table 2.7.4 Bit structure of serial I/O status register



Serial I/O status register (address $00E7_{16}$)

Transmit buffer empty flag (TBE)
0 : Full
1 : Empty

Receive buffer full flag (RBF)
0 : Empty
1 : Full

Transmit shift register shift completion flag
0 : Busy shifting
1 : Shifting complete

Overrun error (OE) flag
0 : No overrun error
1 : Overrun error

Parity error (PE) flag
0 : No parity error
1 : Parity error

Framing error (FE) flag
0 : No framing error
1 : Framing error

Summing error (SE) flag
0 : OE+PE+FE=0
1 : OE+PE+FE=1

[UART control register]

UART control register defines the UART's formats. This register can be read or written by software (see Table 2.7.5) . By setting the UART control register, 8 types of the serial data transfer formats can be selected (see Table 2.7.6) .

Table 2.7.5 Bit structure of UART control register

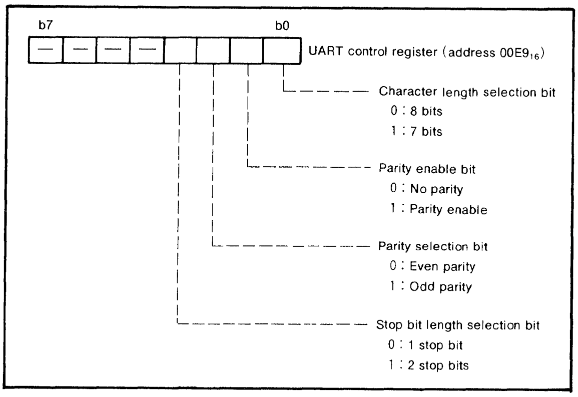


Table 2.7.6 Correspondence between UARTCON and data format

UART control register				Serial data transfer format
b3	b2	b1	b0	
0	×	0	0	1ST-8DATA-1SP
0	×	0	1	1ST-7DATA-1SP
0	×	1	0	1ST-8DATA-1PAR-1SP
0	×	1	1	1ST-7DATA-1PAR-1SP
1	×	0	0	1ST-8DATA-2SP
1	×	0	1	1ST-7DATA-2SP
1	×	1	0	1ST-8DATA-1PAR-2SP
1	×	1	1	1ST-7DATA-1PAR-2SP

ST : Start bit
DATA : Serial data
SP : Stop bit
PAR : Parity bit

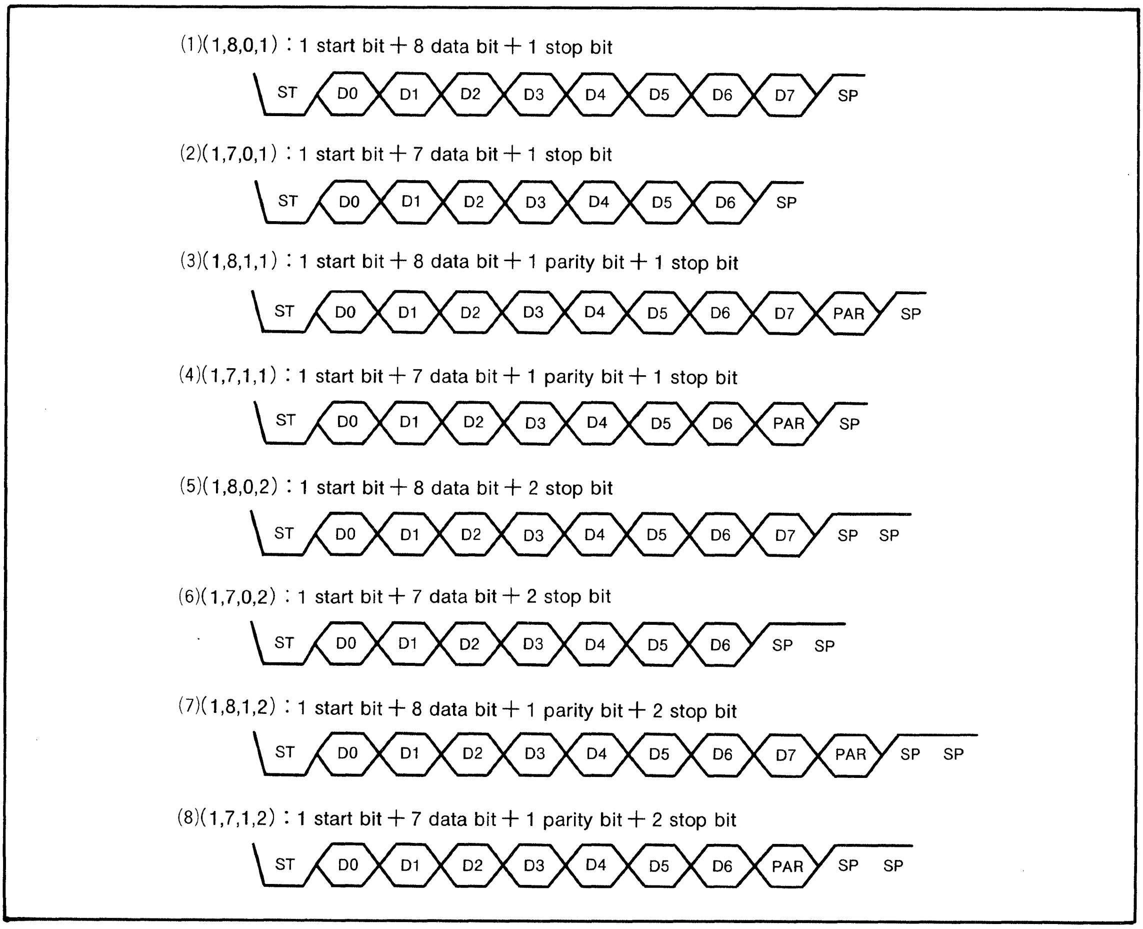

(1)(1,8,0,1) : 1 start bit + 8 data bit + 1 stop bit
ST D0 D1 D2 D3 D4 D5 D6 D7 SP
(2)(1,7,0,1) : 1 start bit + 7 data bit + 1 stop bit
ST D0 D1 D2 D3 D4 D5 D6 SP
(3)(1,8,1,1) : 1 start bit + 8 data bit + 1 parity bit + 1 stop bit
ST D0 D1 D2 D3 D4 D5 D6 D7 PAR SP
(4)(1,7,1,1) : 1 start bit + 7 data bit + 1 parity bit + 1 stop bit
ST D0 D1 D2 D3 D4 D5 D6 PAR SP
(5)(1,8,0,2) : 1 start bit + 8 data bit + 2 stop bit
ST D0 D1 D2 D3 D4 D5 D6 D7 SP SP
(6)(1,7,0,2) : 1 start bit + 7 data bit + 2 stop bit
ST D0 D1 D2 D3 D4 D5 D6 SP SP
(7)(1,8,1,2) : 1 start bit + 8 data bit + 1 parity bit + 2 stop bit
ST D0 D1 D2 D3 D4 D5 D6 D7 PAR SP SP
(8)(1,7,1,2) : 1 start bit + 7 data bit + 1 parity bit + 2 stop bit
ST D0 D1 D2 D3 D4 D5 D6 PAR SP SP

Fig. 2.7.9 The communication data format for UART

[Receive shift register (RSR) and receive buffer (RB)]
The basic operation of the receive shift register and the receive buffer is the same as in the clock synchronous serial I/O. However, the shift clock supplied to the receive shift register is either the external clock (input from S_{CLK} pin) or the BRG clock divided by 16. The examination of the start bit consists of testing the center level of the start bit after the first falling edge of RxD input. If this level is high, the bit is ignored as noise, but if the bit level is low, the bit is recognized as a normal start bit and the CPU starts receiving data.
The character bit length can be selected as either 8-bit or 7-bit by bit 0 of the UART control register.
When 7-bit mode is selected, the most significant bit in the receive buffer register is set to “0” .
The over-run error flag OE operates in the same manner as in clock synchronous mode but there are other flags such as: parity error bit (PE) , framing error bit (FE) and the summing error bit (SE) which is the result of the logical sum of OE,PE, and FE bits. These four error flags are bit 3~6 of the serial I/O status register and are cleared when the receive enable bit is cleared or dummy data is written into the serial I/O status register.

[Transmit shift register (TSR) and transmit buffer (TB)]
The basic operation of the transmit shift register and the transmit buffer is also the same as in the clock synchronous mode. The data is transmitted from TxD pin. Bit 3 of the UART control register selects the stop bit length as either 1 or 2 bits. Parity is created by hardware and set by bits 1 and 2 of the UART control register. When bit 1 of the UART control register is “0” meaning non-parity mode, the parity bit is not generated.
Data transmission begins by writing data into the transmit buffer after setting the transmit enable bit.

[Clock generating circuit]
Table 2.7.7. shows the baud rate selection list when UART is selected.

Table 2.7.7 Baud rate selection chart

Baud rate (bps)	BRG count source	BRG value	Baud rate at $f(X_{IN})=10MHz$
300	$f(X_{IN})/8$	$255(FF_{16})$	305. 17578
600	$f(X_{IN})/8$	$127(7F_{16})$	610. 35156
1200	$f(X_{IN})/8$	$63(3F_{16})$	1220. 7031
2400	$f(X_{IN})/8$	$31(1F_{16})$	2441. 4063
4800	$f(X_{IN})/8$	$15(0F_{16})$	4882. 8125
9600	$f(X_{IN})/8$	$7(07_{16})$	9765. 625
19200	$f(X_{IN})/8$	$3(03_{16})$	19531. 25
38400	$f(X_{IN})/8$	$1(01_{16})$	39062. 5
76800	$f(X_{IN})/2$	$3(03_{16})$	78125
153600	$f(X_{IN})/2$	$1(01_{16})$	156250
307200	$f(X_{IN})/2$	$0(00_{16})$	312500

(3) UART receive operation

In order the UART to receive data an initialization sequence must be followed. After selecting the clock source and serial I/O mode, bit 1 and 6 of serial I/O control register respectively, serial I/O enable bit must be set to "1" . By writing dummy data into the serial I/O status register, it will be initialized to "00_{16}" . Lastly the data format must be set in the UART control register.

After the receive enable bit of the serial I/O control register is set, the M37450 is ready to receive the data.

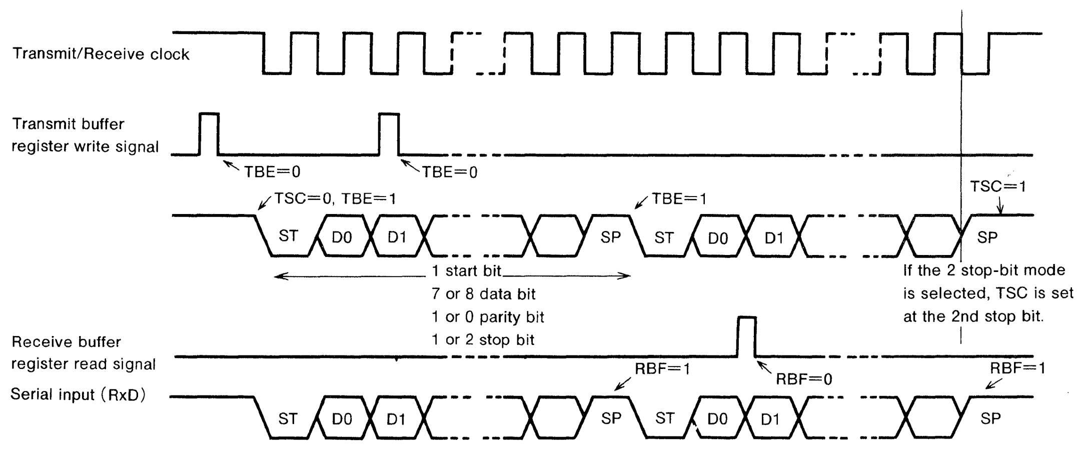
Once the start bit is recognized, receiving action will begin. When the first stop bit is recognized, the receive buffer full flag is set and the receive interrupt request is generated. If there is an error in the receiving operation, the error flag will be set at this time.

(4) UART transmission operation

In order the UART to transmit data an initialization sequence must be followed. After selecting the clock source and the serial I/O mode, bit 1 and 6 of serial I/O control register respectively, serial I/O enable bit must be set to "1" . By writing data into the serial I/O status register, it will be initialized to "00_{16}" . Lastly the data format must be set in the UART control register.

The shift clock is generated by writing the transmission data into the transmit buffer register after the transmit enable bit of the serial I/O control register is set to enable. At this time the transmission shift register shift completion flag is cleared to "0" and the transmit buffer empty flag is set to "1" . The next transmission data can be written into the transmission buffer. The transmit interrupt selection bit of the serial I/O control register can select the transmit interrupt timing. If the selection bit is "0", the transmit interrupt occurs when the transmit buffer empty flag becomes "1" . If the selection bit is "1" , the transmit interrupt occurs when the transmit shift register shift completion flag becomes "1" . (The shift completion flag is set by the second stop bit when the stop bit length is chosen to be 2 at 2 stop bits.

UART timing diagram is shown in Figure 2.7.10.



Note 1 : The error flag are detected at the same time RBF flag setting (RBF=1, at the first stop bit in data receiving mode.).

Note 2 : The transmit interrupt is programmed to interrupt at either transmit buffer being empty (TBE=1) or the transmit being completed (TSC=1).

Note 3 : The receive interrupt is generated by RBF=1.

Fig. 2.7.10 UART data transmit/receive timing

2.8 Bus interface

The slave microcomputer is defined as the processor which operates by the master processor's instruction.

The M37450 has a built-in bus interface which the master processor can directly access through its data bus. This bus interface allows the master CPU to access the M37450 without any special hardware or protocol. The master CPU can handle the slave M37450 the same as a standard peripheral LSI.

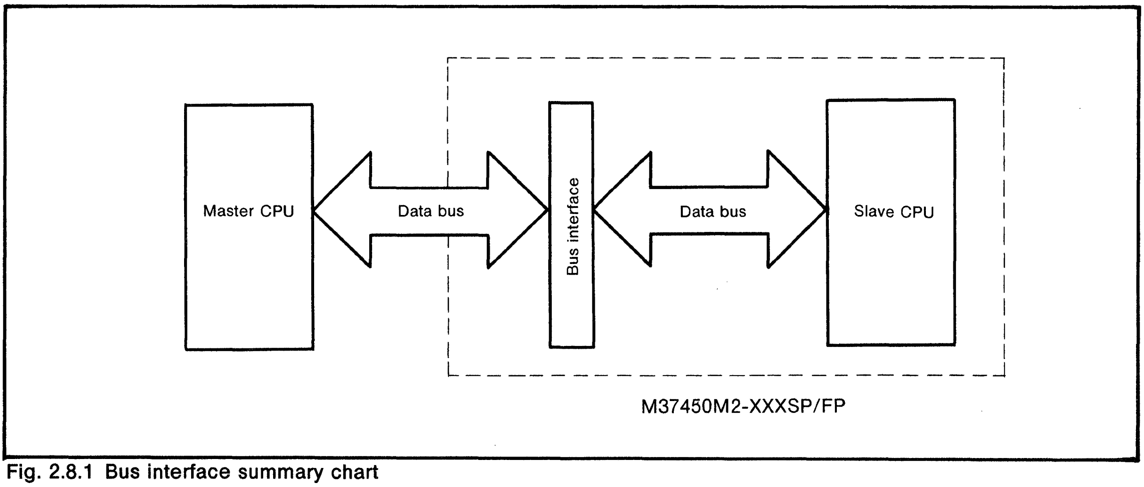


Fig. 2.8.1 Bus interface summary chart

2.8.1 Bus interface function

M37450 has the following built-in bus interface features:

- 8-bit data bus
- Data bus buffer (1-level common with input and output)
- Direct connection to “$\overline{R}$ and $\overline{W}$ separation” or “R/$\overline{W}$” type bus architecture
- Output the input/output buffer condition on the data bus buffer

Figure 2.8.2 shows the block diagram of the bus interface circuit.

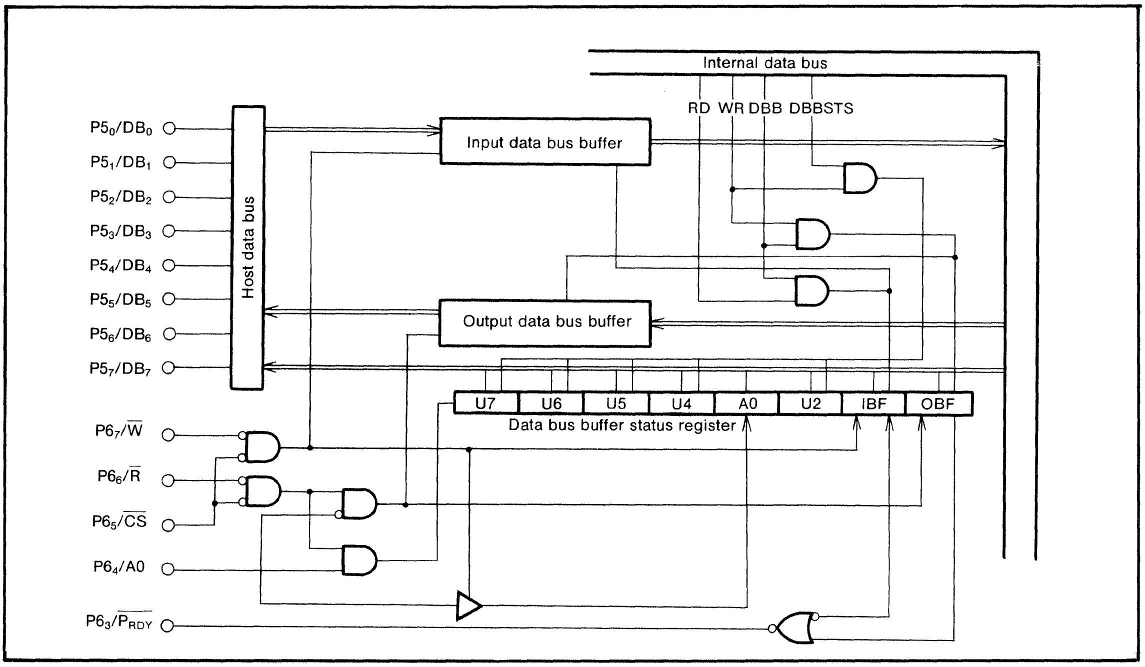


Fig. 2.8.2 Bus interface circuit block diagram

2.8.2 Block explanation

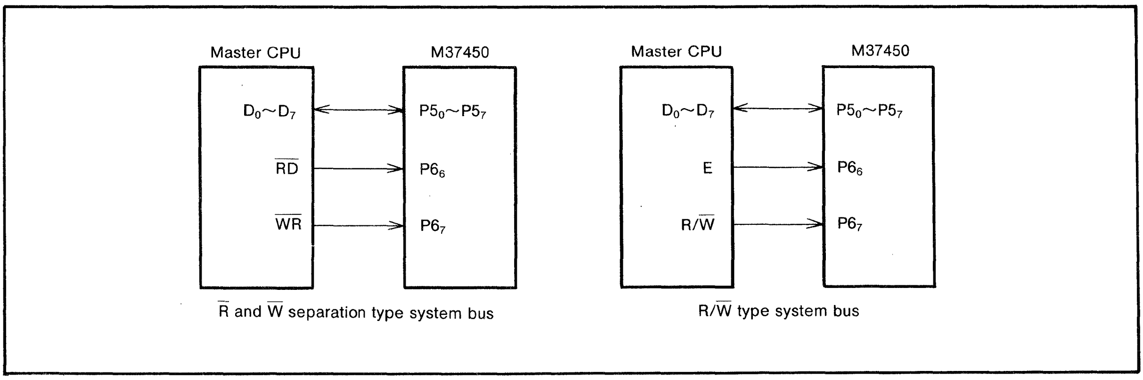
(1) Bus interface enable bit (DBBE)
If the bus interface enable bit, bit 2 of MISRG2 (address $00DF_{16}$) , is set to "1" , port P5 and the part of port P6 change their functions as shown in Table 2.8.1.

(2) Bus interface mode selection bit (DBBM)
Bus interface mode selection bit, bit 3 of MISRG2, defines the bus interface function type. If the bit is set to "0" , the bus is $\overline{R}$ and $\overline{W}$ separation type with separated $\overline{RD}$ or $\overline{WR}$ signals. If the bit is set to "1", the bus interface is R/$\overline{W}$ type with multiplexed R/$\overline{W}$ signals.

(3) Input data bus buffer (DBBIN)
This buffer is a single level buffer used to store the input from the master CPU data bus (port P5) . When the logical sum of $\overline{CS}$ and $\overline{W}$ is "0" , the data bus condition (at the rising edge of the $\overline{W}$) is latched into the input data bus buffer (DBBIN). At the same time, the input buffer full flag (IBF) which is in bit 1 of the data bus buffer status register (DBBSTS), is set to "1" . Then, the input buffer full interrupt request is generated and bit 0 of the interrupt request register 1 (address $00FC_{16}$) is set to "1" . Also in this timing, the A0 status is copied to bit 3 of the data bus status register. The status of this bit shows whether the content of the data bus buffer is data or a command. The input buffer full flag is cleared to "0" automatically by reading the input data bus buffer.

Table 2.8.1 Functions of bus interface enable bit

Port	DBBE	"0", disable	"1", enable	
	DBBM		"0" $\overline{RD}$ and $\overline{WR}$ separation type	"1" R/$\overline{W}$ type
Ports $P5_0$~$P5_7$		I/O port	System data bus DB0~DB7	
Ports $P6_3$		I/O port	$\overline{P_{RDY}}$ output pin	
Ports $P6_4$		I/O port	A_0 input pin	
Ports $P6_5$		I/O port	$\overline{CS}$ input pin	
Ports $P6_6$		I/O port	$\overline{RD}$ input pin	E signal input pin
Ports $P6_7$		I/O port	$\overline{WR}$ input pin	R/$\overline{W}$ input pin



$\overline{R}$ and $\overline{W}$ separation type system bus

R/$\overline{W}$ type system bus

Fig. 2.8.3 Bus connection block diagram

(4) Output data bus buffer (DBBOUT)

This buffer is a single level buffer, used to store the output to the master CPU data bus (port P5) . When data is written to the output data buffer (address $00E4_{16}$) , the output buffer full flag (OBF) , bit 0 of the data bus status register, is set to “1”.

When the logical sum of $\overline{\mathrm{CS}}$, $\overline{\mathrm{R}}$, and A0 is “0” (see Table 2.8.2 ① condition) , the contents of the output data bus buffer is output to the master CPU data bus. At the rising edge of the $\overline{\mathrm{R}}$ signal, the output buffer full flag is cleared to “0”, the output buffer empty interrupt request is generated and the output buffer empty interrupt request flag (bit 1 of address $00FC_{16}$) is set to “1”. The input data bus buffer and the output data bus buffer are in the same address, and data can be read from the DBB_{IN} and write to DBBOUT.

(5) Data bus buffer status and user definable flag (DBBSTS)

The data bus buffer status register, address $00E5_{16}$, houses data bus status flags and user definable flags. Bits 0, 1, and 3 are flags output buffer full flag, input buffer full flag, (IBF) , and address input (A0) flags respectively. These bits can not be set or clear by program. The other bits are user definable flags that can be set or clear by program. The A0 flag indicates where the data is actual data or a command; “0” meaning data and “1” meaning command. The content of the data bus status register is output to the data bus when the logical sum of $\overline{\mathrm{CS}}$ and $\overline{\mathrm{R}}$ is “0” and A0 is “1” (see the condition ② in the Table 2.8.2).

The logical sum of OBF and $\overline{\mathrm{IBF}}$ is defined as the $\overline{P_{RDY}}$ signal (parallel ready) if bus interface used, this signal is output from $P6_3$. This signal acts as a request to the master CPU for data receiving or transmission.

The master CPU cannot write to the data bus status register but it can read from it. On the other hand, the slave processor can read out from this register and write to bit U7, U6, U5, U4, and U2 in this register.

The data bus status register is initialized to “01_{16}” at reset. In order to complete the initialization process of the data bus status register, bus interface enable bit (DBBE) must be set to “1” , which causes the output buffer full flag to be cleared to “0” . At the same time that the output buffer full flag is cleared to “0” , the output buffer empty interrupt request is generated. Data to be sent should be written to the DBBOUT after IBF is set to “1” , because $\overline{P_{RDY}}$ does not change to “H” when IBF=“0”.

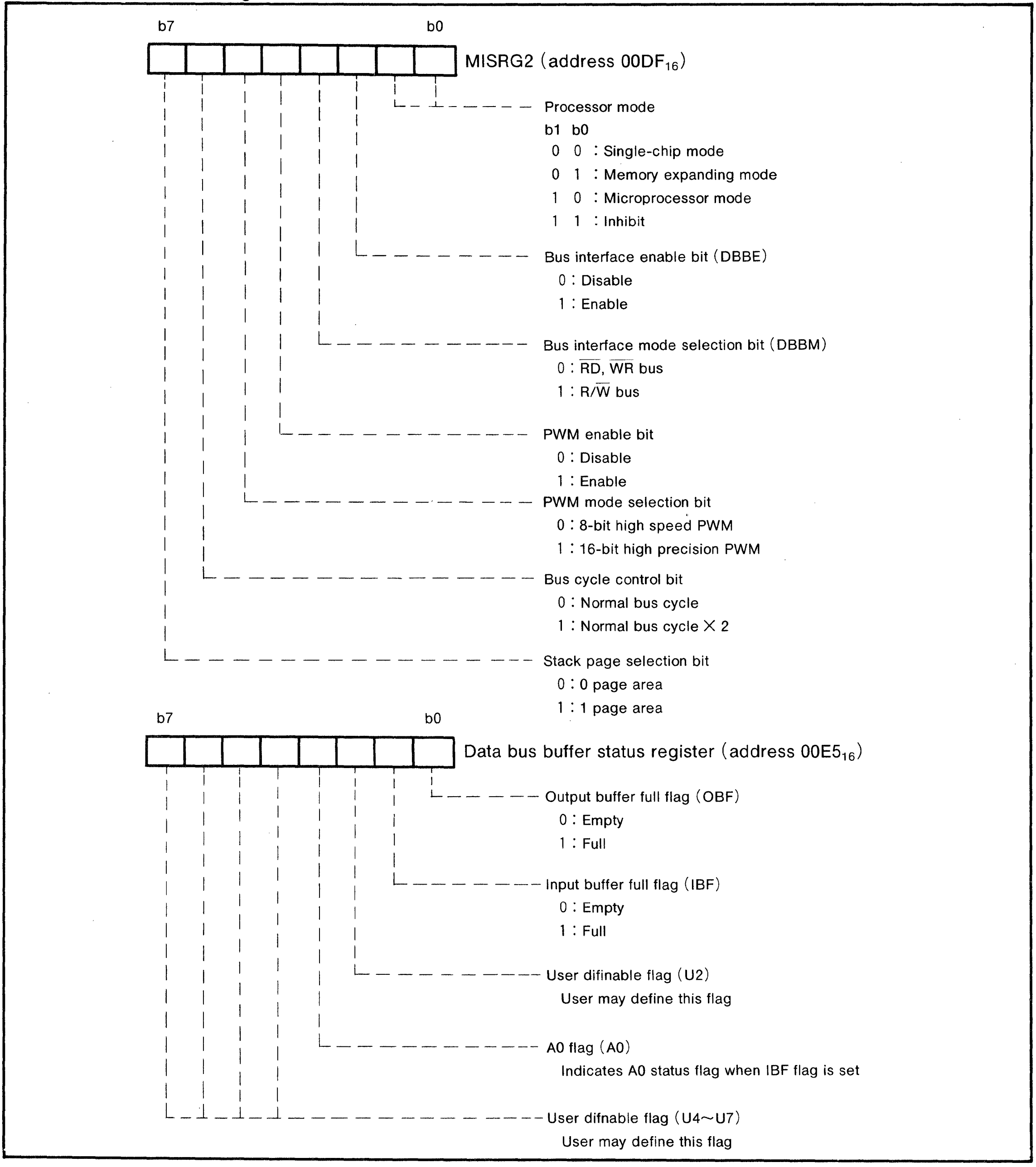
Table 2.8.2 Bus control signal and data bus status ($\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$ separation type)

	$\overline{\mathrm{CS}}$	$\overline{\mathrm{R}}$	$\overline{\mathrm{W}}$	$\overline{\mathrm{A0}}$	Data bus status	Data on data bus
①	0	0	1	0	Read	DBBOUT
②	0	0	1	1	Read	DBBSTS
	0	1	0	0	Write	DBBIN (data)
	0	1	0	1	Write	DBBIN (command)
	0	×	×	×	High impedance	—

Table 2.8.3 Bus control signal and data bus status (R/$\overline{\mathrm{W}}$ type)

$\overline{\mathrm{CS}}$	R/$\overline{\mathrm{W}}$	E	$\overline{\mathrm{A0}}$	Data bus status	Data on data bus
0	1	1	0	Read	DBBOUT
0	1	1	1	Read	DBBSTS
0	0	1	0	Write	DBBIN (data)
0	0	1	1	Write	DBBIN (command)
1	×	×	×	High impedance	—

Table 2.8.4 Structure of registers related to bus interface



MISRG2 (address $00DF_{16}$)

Processor mode

b1 b0
0 0 : Single-chip mode
0 1 : Memory expanding mode
1 0 : Microprocessor mode
1 1 : Inhibit

Bus interface enable bit (DBBE)
0 : Disable
1 : Enable

Bus interface mode selection bit (DBBM)
0 : $\overline{RD}$, $\overline{WR}$ bus
1 : R/$\overline{W}$ bus

PWM enable bit
0 : Disable
1 : Enable

PWM mode selection bit
0 : 8-bit high speed PWM
1 : 16-bit high precision PWM

Bus cycle control bit
0 : Normal bus cycle
1 : Normal bus cycle X 2

Stack page selection bit
0 : 0 page area
1 : 1 page area

Data bus buffer status register (address $00E5_{16}$)

Output buffer full flag (OBF)
0 : Empty
1 : Full

Input buffer full flag (IBF)
0 : Empty
1 : Full

User difinable flag (U2)
User may define this flag

A0 flag (A0)
Indicates A0 status flag when IBF flag is set

User difnable flag (U4~U7)
User may define this flag

2.8.3 The use of bus interface

Figure 2.8.3 shows an example of the bus interface connection.

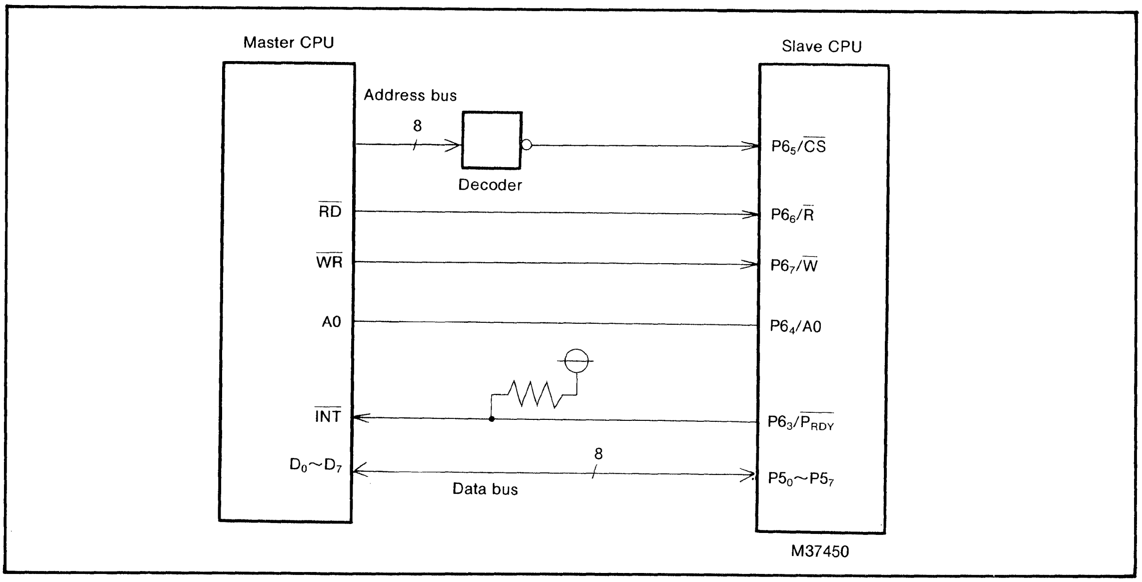


Fig. 2.8.3 Example of connection with the master CPU

(1) Initialization

The bus interface function is accessed by setting the bus interface enable bit in the M37450. However, the master processor needs to know that the slave processor is ready to be accessed.
The $\overline{P_{RDY}}$ which is common with $P6_3$ pin is output as "0" for telling the master processor that the slave processor is ready.

After reset sequence, $\overline{P_{RDY}}$ pin will be in input mode (high impedance), this pin must be pulled-up.

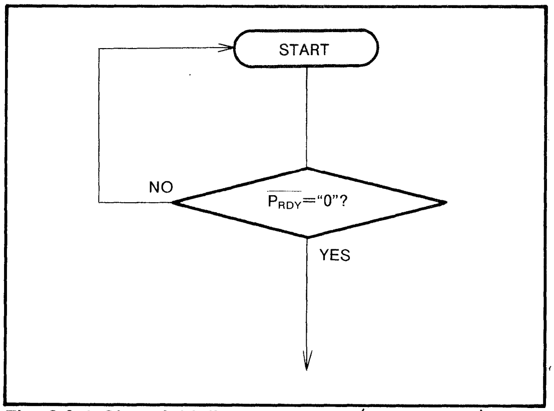


Fig. 2.8.4 Slave initialization routine (master side)

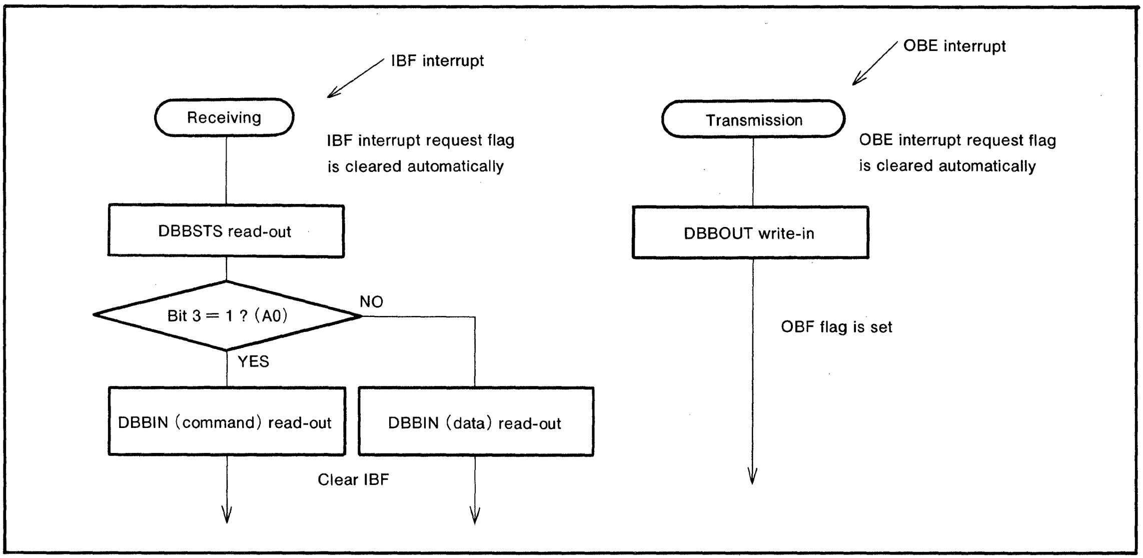
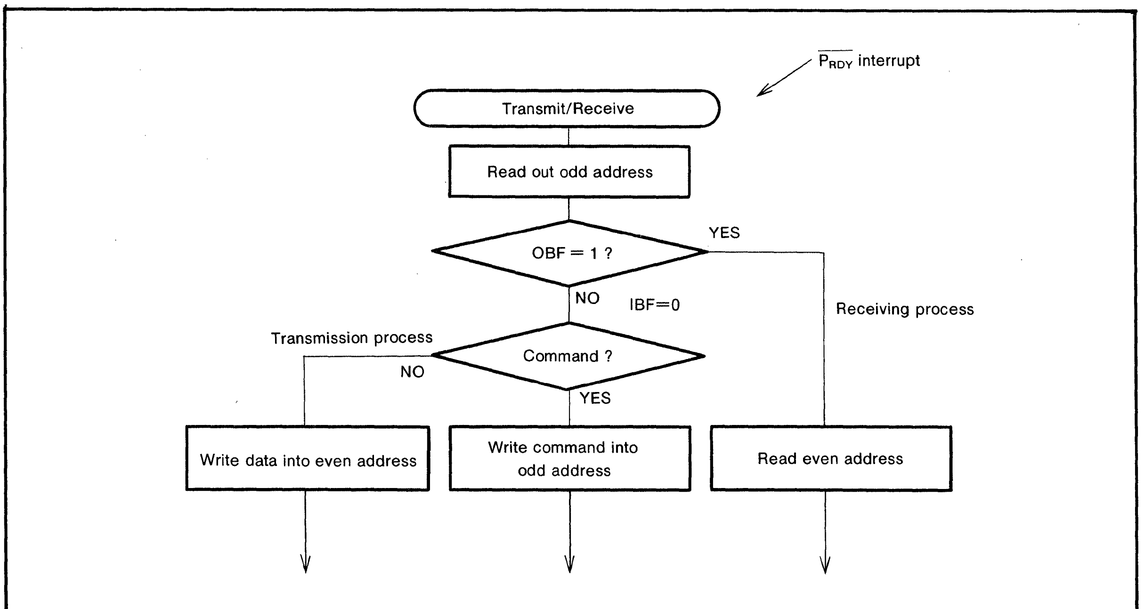


Fig. 2.8.5 Example of the slave side transmit/receive software



Note : In the connection example in Figure 2.8.3, the slave-side A0 status input is connected to the master-side address signal (A0) . The data or command can be distinguished by selecting even or odd address.

Fig. 2.8.6 Example of the host-side transmit/receive software

2.9 A-D converter

M37450 has a built-in A-D converter with the features shown below.

Analog input pin (common with P4)	——8channels(80-pin model) 3channels(64-pin model)
Architecture	——Successive approximation A-D conversion circuit with a multiplexer
Resolution	——8-bit
Absolute precision	——±3LSB
Conversion speed	——19.8μs (when f(X_{IN})=10MH$_z$)

The block diagram of A-D converter is shown in Figure 2.9.1.

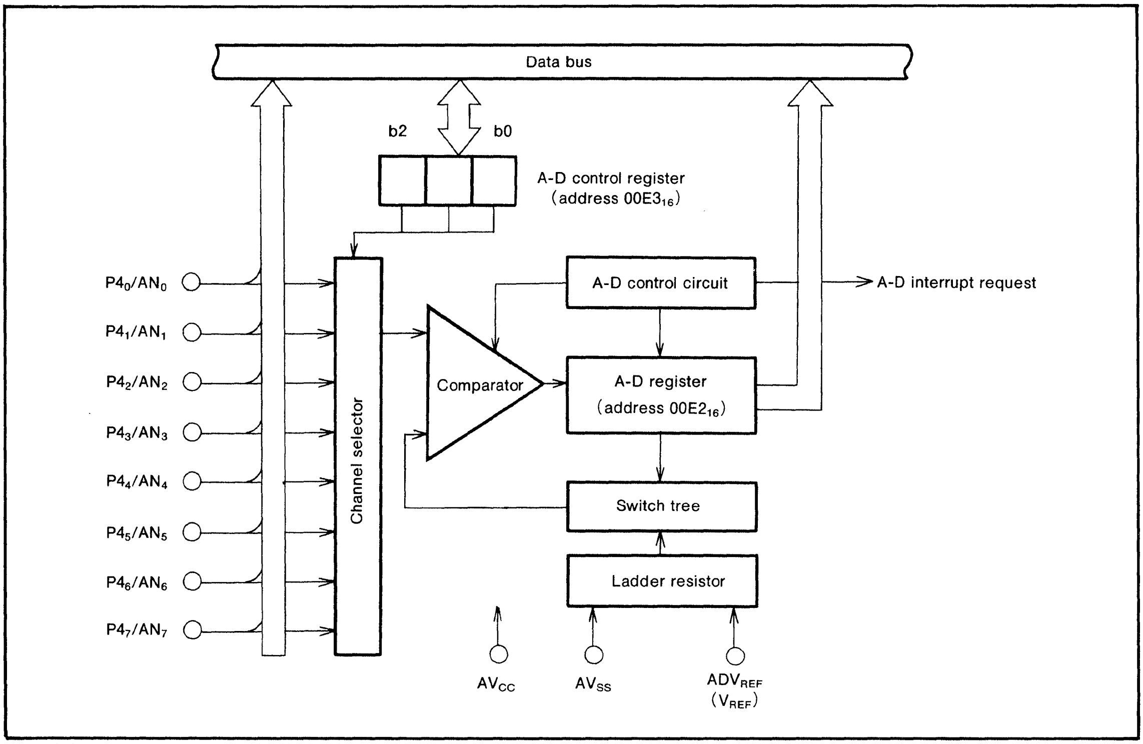


Fig. 2.9.1 A-D converter block diagram

2.9.1 Block explanations

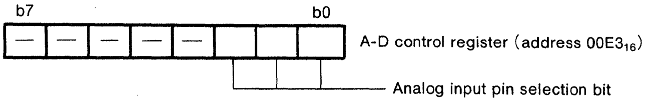
The function of each block of the A-D converter is explained below.

(1) A-D register
This is a read-only register where the result of A-D conversion is stored. The content of this register should not be read out during the A-D conversion.

(2) A-D control register
This register is to select the analog input pin. The value of the three least significant bits choses among AN_0~AN_7. The other analog input pins which are not used for A-D conversion can be used as normal port. Once the data is written to this register, A-D conversion will begin. In the 64-pin model there is only three analog input ports AN_0~AN_2, therefore the value of this register must be from 0~2.
At reset, this register is undefined.

(3) Comparator and control circuit
The analog input voltage and the reference voltage are compared and the result is stored in the A-D register.
The A-D interrupt request bit (bit5 of address $00FD_{16}$) is set to "1" at the completion of A-D conversion. The comparator consists of capacitance coupling so that $f(X_{IN})$ should be $1MH_Z$ or more during the A-D conversion.

Table 2.9.1 A-D control register bit structure


b7
b0
—
—
—
—
—
A-D control register (address $00E3_{16}$)
Analog input pin selection bit

	b2	b1	b0	
64-pin model	0	0	0	: AN_0
	0	0	1	: AN_1
	0	1	0	: AN_2
	0	1	1	: AN_3
	1	0	0	: AN_4
	1	0	1	: AN_5
	1	1	0	: AN_6
	1	1	1	: AN_7

Note : In the 64-pin model, only bits AN_0~AN_2 are selected.

2.9.2 Internal operation during A-D conversion

The A-D conversion starts by writing data into the A-D control register. When A-D conversion starts, the A-D register is cleared to "00_{16}". Next, the most significant bit of the A-D register is set to "1" and the comparative voltage V_{ref} is input in the comparator. Here the analog input voltage V_{IN} is compared with V_{ref}.

If $V_{ref} < V_{IN}$, the bit is not cleared.
If $V_{ref} > V_{IN}$, the bit is cleared to "0".

By repeating this operation until the lowest bit of the A-D register, the analog value is converted to the digital value. At 198 clock cycles (when $f(X_{IN})=10MHz$, 19.8μs) after the start of the A-D conversion, the conversion is completed and the result of conversion is stored in the A-D register. At the same time the A-D interrupt request is generated and the A-D interrupt request bit (bit 5 of address $00FD_{16}$) is set to "1". ADV_{REF} is V_{REF} in the 64 pin model.

Relational formula between
V_{ref} and ADV_{REF} (80-pin model)

$$V_{ref}=\frac{ADV_{REF}{}^{*}}{256}\times(n-0.5) \text{ at } n=1\sim255$$

$V_{ref}=0$ at $n=0$ $*V_{REF}$ in the 64-pin model

Table 2.9.2 Register contents in the A-D conversion process

	Change in A-D register	Comparative voltage
At start of conversion	0 0 0 0 0 0 0 0	0
1st comparison	1 0 0 0 0 0 0 0	$\frac{ADV_{REF}}{2}-\frac{ADV_{REF}}{512}$
2nd comparison	1st 1 0 0 0 0 0 0. The result of the 1st conversion	$\frac{ADV_{REF}}{2}\pm\frac{ADV_{REF}}{4}-\frac{ADV_{REF}}{512}$
3rd comparison	1st 2nd 1 0 0 0 0 0	$\frac{ADV_{REF}}{2}\pm\frac{ADV_{REF}}{4}\pm\frac{ADV_{REF}}{8}-\frac{ADV_{REF}}{512}$
After 8th completion	A-D conversion result 1st 2nd 3rd 4th 5th 6th 7th 8th	

Note : ADV_{REF} is V_{REF} in the 64-pin model

2.9.3 A-D conversion method

The method for A-D conversion is explained below.

(1) Clear the A-D interrupt request bit (bit 5) of the interrupt request register 2.

(2) When the A-D interrupt is used, set the A-D interrupt enable bit to "1" and clear the interrupt disable flag to "0".

(3) Select the analog input by the analog input selection bit of the A-D control register. A-D conversion begins by this write operation, (do not read the content of the A-D register during A-D conversion).

(4) Confirm completion of conversion by checking the A-D interrupt request bit or the A-D interrupt.

(5) The result of the conversion is obtained by reading out the A-D register.

In case of not using the interrupt process the A-D interrupt request bit is not cleared to "0" automatically. It must be cleared to "0" before A-D conversion. At reset, this bit is reset to "0".

2.9.4 A-D converter circuit

Figure 2.9.2 shows the A-D converter circuit and Figure 2.9.3 shows the A-D conversion timing chart.

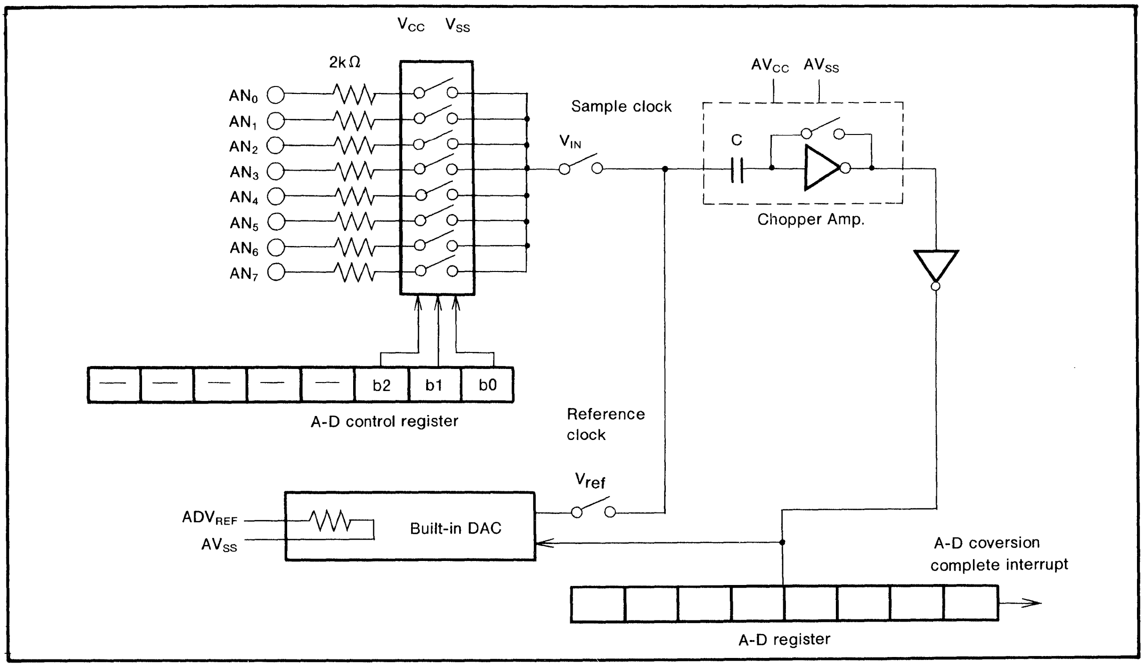


Fig. 2.9.2 A-D converter equivalent circuit

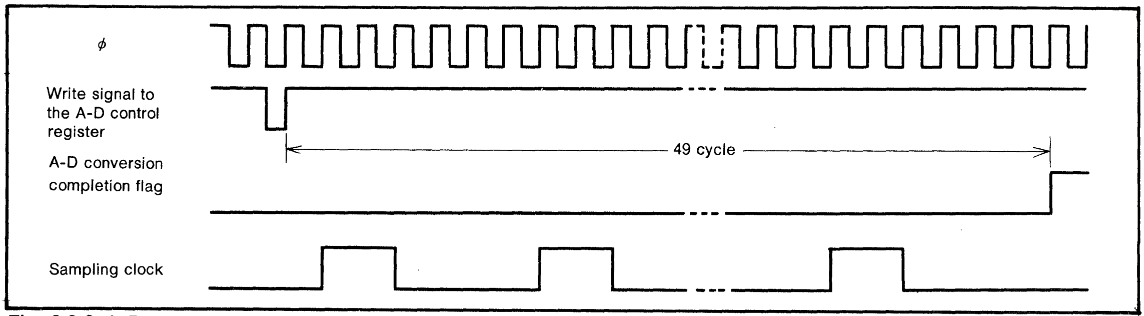

φ
Write signal to the A-D control register
49 cycle
A-D conversion completion flag
Sampling clock

Fig. 2.9.3 A-D conversion timing chart

2.9.5 Measuring method of each standard of the A-D converter

(1) Absolute precision

The difference between the result code through a real A-D converter and one of a ideal featured A-D converter is indicated by LSB (least significant bit) unit. The measured input voltage is determined at center point of the output voltage output by the ideal A-D converter.
For example, when $ADV_{REF} = 5.12V$, 1LSB width is 20mV and 0mV, 20mV, 40mV, and 60 mV ...are selected as the input voltages. The range of actual input voltage is form AV_{SS} to ADV_{REF}. When using the ADV_{REF} lower than AV_{CC}, the output code of input voltage becomes "FF_{16}" from ADV_{REF} to AV_{CC}, but the precision becomes worse. The absolute precision of ±3 LSB means that, the input voltage at 100mV, the output code obtained by the ideal A-D converter is 05_{16} indicating the range of 02_{16}~08_{16}.
The range of the input voltage is AV_{SS}~ADV_{REF}. If the lower ADV_{REF} is used, the absolute precision shown by LSB becomes poor. The absolute precision includes zero error and the full-scale error.

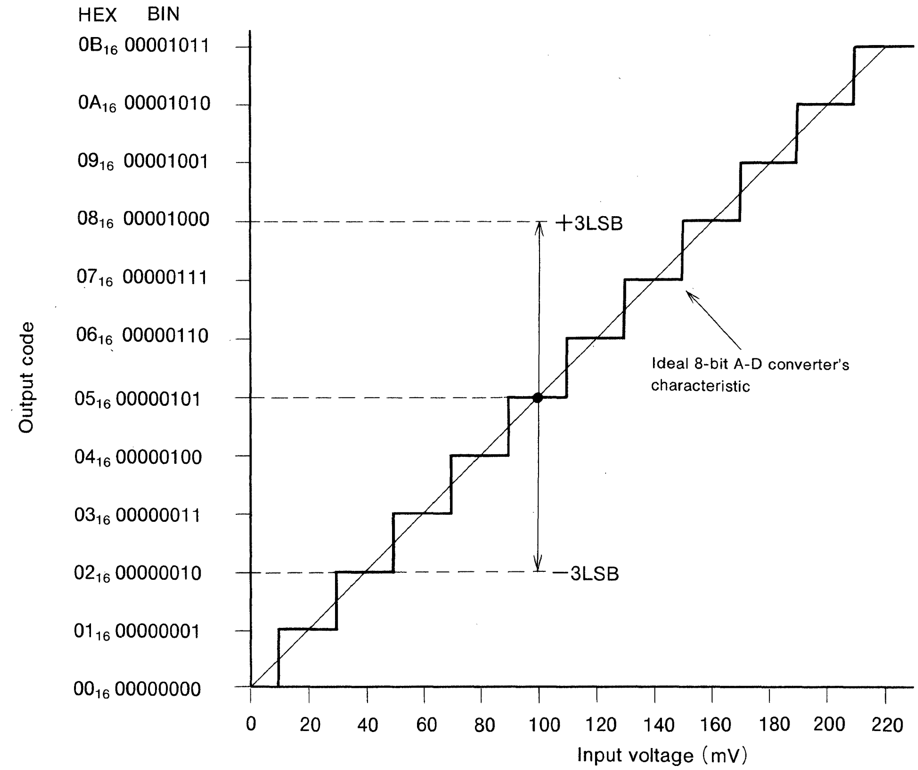


Fig. 2.9.4 A-D converter absolute precision

(2) Differential non-linearity

Differential non-linearity shows the difference between the step width of 1 LSB of the ideal A-D converter and the measured step width. For example, when $ADV_{REF}=5.12V$, 1LSB is 20mV. Here the maximum differential non-linearity deviation is ±1LSB, that means the step width of the input voltage which gets the same output code is in the range 0~40mV.

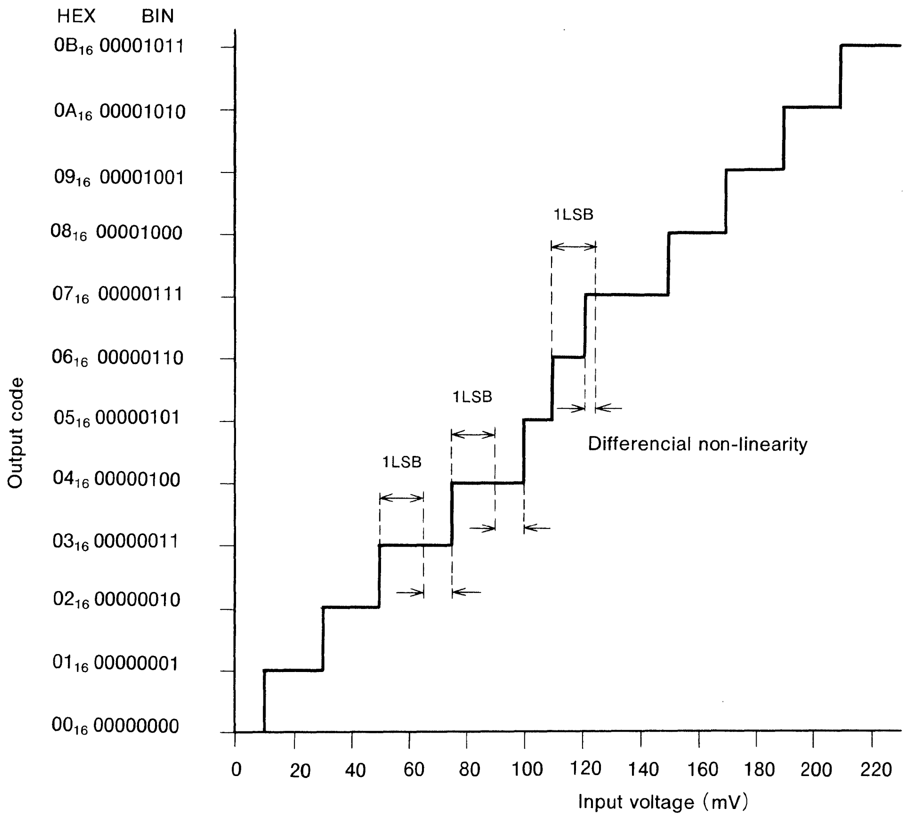


Fig. 2.9.5 A-D converter absolute precision

Note : Each port of M37450 has a built-in protective circuit for the static electricity, but each analog-type port (AV_{CC}, AV_{SS}, ADV_{REF}, DAV_{REF}, V_{REF}, D-A_1, D-A_2) are not on the same protective level; caution regarding the static electricity is necessary when handling.

2.10 D-A converter

M37450 has a built-in D-A converter with features shown below.

Analog output pin	——2-channel
Architecture	——R-2R type
Resolution	——8-bit

The block diagram of D-A converter is shown in Figure 2.10.1.

By writing the data to the D-A register, the voltage between DAV_{REF} and AV_{SS} corresponding to the contents of the D-A register is output from the D-A pin. ($D\text{-}A_1$ register corresponds with $D\text{-}A_1$ pin; $D\text{-}A_2$ register corresponds with $D\text{-}A_2$ pin.)

The relationship between the output voltage and the D-A register's value is as follows (80-pin model).

$V = DAV_{REF} \times n/256$ ($n=0\sim255$)

V; Output voltage

DAV_{REF}; Reference voltage (DAV_{REF} is V_{REF} in 64-pin model)

n; D-A conversion register value(Decimal representation)

Since both $D\text{-}A_1$ and $D\text{-}A_2$ registers are cleared to "00_{16}" at reset, the output voltage from $D\text{-}A_1$ and $D\text{-}A_2$ is 0V after reset.

The equivalent circuit of D-A converter is shown in Figure 2.10.2. Both $D\text{-}A_1$ and $D\text{-}A_2$ have the same structure and output an un-buffered voltage. The resistor ladder is directly connected to D-Ai pin, so buffer amplifer is need to output current.

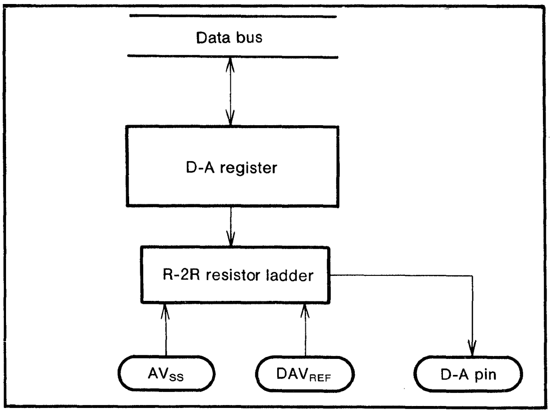
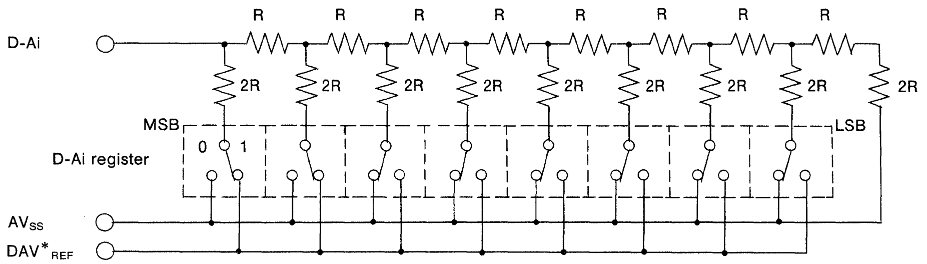


Fig. 2.10.1 D-A converter block diagram



This combination outputs a voltage corresponding to "$C0_{16}$" in the D-Ai register.

*DAV_{REF} pin corresponds to V_{REF} pin in the 64-pin model

Fig. 2.10.2 D-A converter equivalent circuit

2.11 PWM

M37450 has a built-in PWM generator with two modes which are software selectable.

(1) High speed mode (Resolution 8-bit)

Period $\frac{255\times2}{f(X_{IN})}$ (fixed period) (at $f(X_{IN})$=10MHz)

(2) High resolution mode (Resolution 16-bit)

Period $\frac{65535\times2}{f(X_{IN})}$ (fixed period) (at $f(X_{IN})$=10MHz)

The structure of PWM generator is shown in Figure 2.11.1

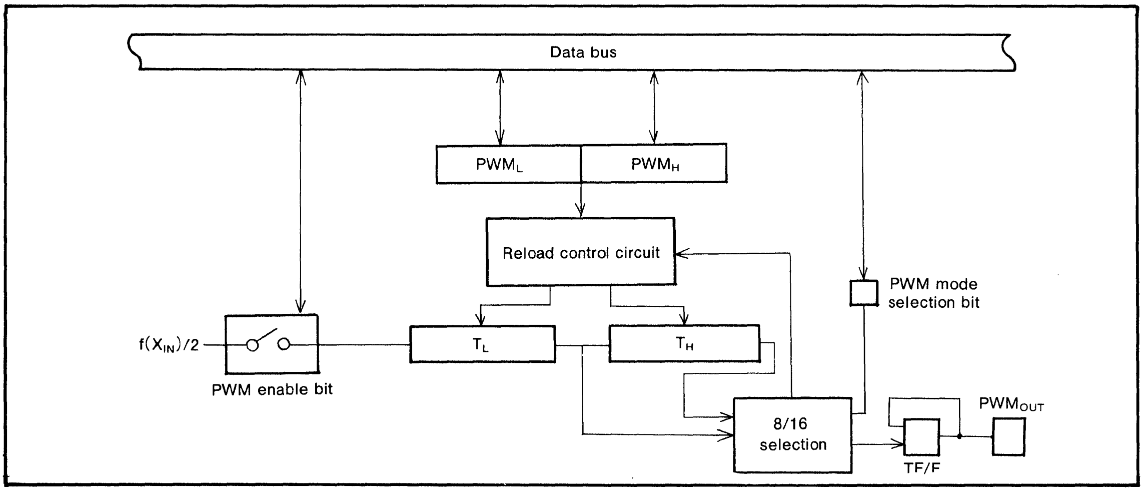


Fig. 2.11.1 PWM block diagram

2.11.1 Block explanations

(1) PWM enable bit, PWM mode selection bit

PWM enable bit and PWM mode selection bit are bit 4 and bit 5 of MISRG2 respectively.
PWM enable bit controls the connection of count clock to the PWM generator. The PWM generator operates when this bit is “1”.
PWM mode selection bit selects the PWM operation mode. When PWM mode selection bit is “0”, the 8-bit high speed mode is selected. When it is “1”, the 16-bit high resolution mode is selected.
Since MISRG2 is initialized to “00$_{16}$” at reset, the PWM generator is disabled at reset.

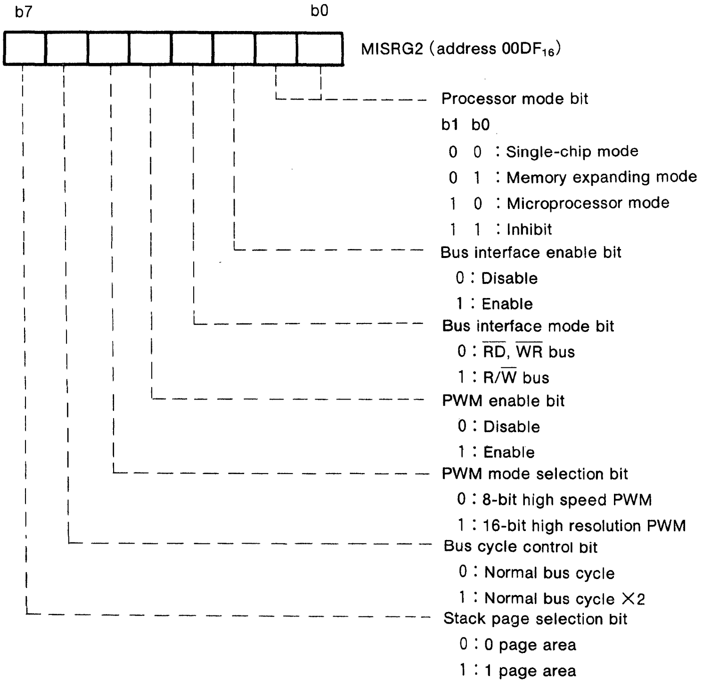
(2) PWM register

PWM register is made up of two 8-bit registers of PWM$_H$ and PWM$_L$. By setting a value to this register, the pulse width corresponding with the contents of this register is generated.

(3) PWM$_{OUT}$ pin

To use P3$_3$/PWM$_{OUT}$ pin as PWM$_{OUT}$ pin, the directional register for P3$_3$ pin must be set in the output condition (bit3 of address 00D7$_{16}$). Then by setting the PWM enable bit to “1”, the enable condition, P3$_3$ pin becomes PWM$_{OUT}$ automatically.

Table 2.11.1 PWM related register



MISRG2 (address 00DF$_{16}$)

Processor mode bit
b1 b0
0 0 : Single-chip mode
0 1 : Memory expanding mode
1 0 : Microprocessor mode
1 1 : Inhibit

Bus interface enable bit
0 : Disable
1 : Enable

Bus interface mode bit
0 : $\overline{RD}$, $\overline{WR}$ bus
1 : R/$\overline{W}$ bus

PWM enable bit
0 : Disable
1 : Enable

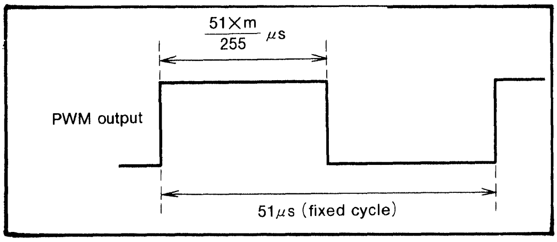
PWM mode selection bit
0 : 8-bit high speed PWM
1 : 16-bit high resolution PWM

Bus cycle control bit
0 : Normal bus cycle
1 : Normal bus cycle ×2

Stack page selection bit
0 : 0 page area
1 : 1 page area

2.11.2 Explanation of the PWM operation

PWM output starts by writing data into the PWM register and the PWM enable bit is set to "1". PWM output in each mode is explained below.



(1) High speed PWM mode
This mode is selected by writing "0" in the PWM mode selection bit. In this mode, the "H" output period on the PWM pin is determined by the value m (m=0~255) written in the PWM_L register.

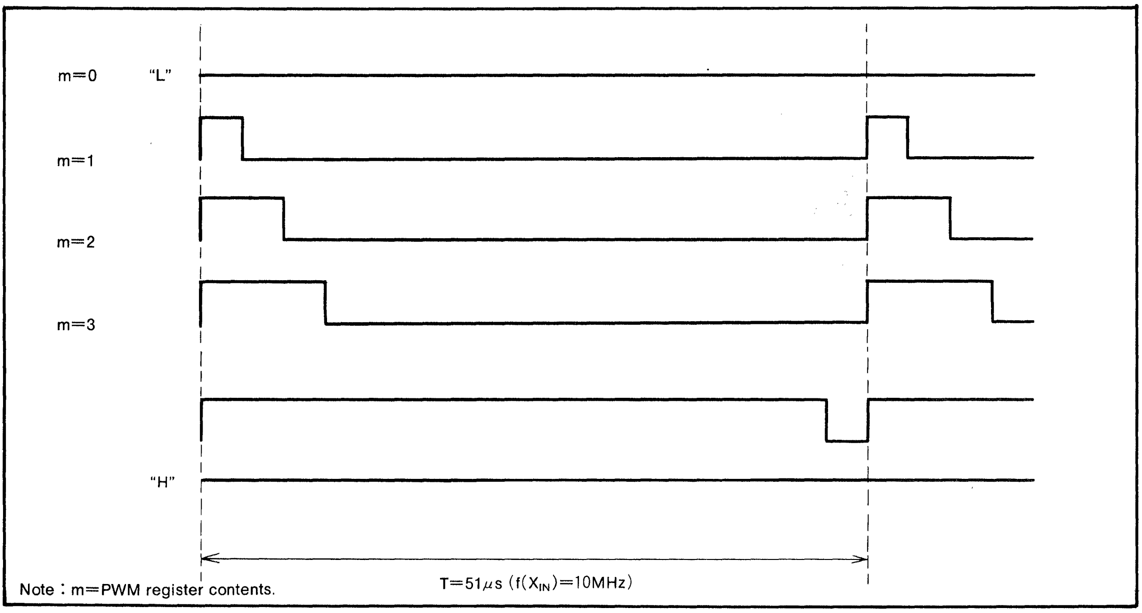


Fig. 2.11.2 Relationship between the PWM register value and the PWM output

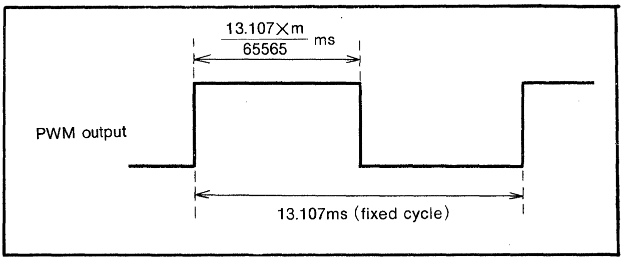
(2) High resolution PWM mode

This mode is selected by writing "1" in the PWM mode selection bit. In this mode, the "H" output period on the PWM pin is determined by the value m (m=0~65535) in the PWM_H and PWM_L registers.

To change all 16 bits during the PWM output period, the value of the PWM_H must be changed first, then the value of PWM_L. In order to change the lower 8 bits only, the new value can be written into the PWM_L without rewriting the value to PWM_H. In order to change only the upper 8 bits, the new value must be written to the PWM_H, the value of PWM_L must be read out and the same value must be rewritten into PWM_L register.

This must be done since the write action to PWM_L signifies the completion of the write cycle to the PWM register.

When the value of the PWM register is changed during both high speed and high resolution PWM modes, the output changes(B) after PWM cycle (A) is completed.



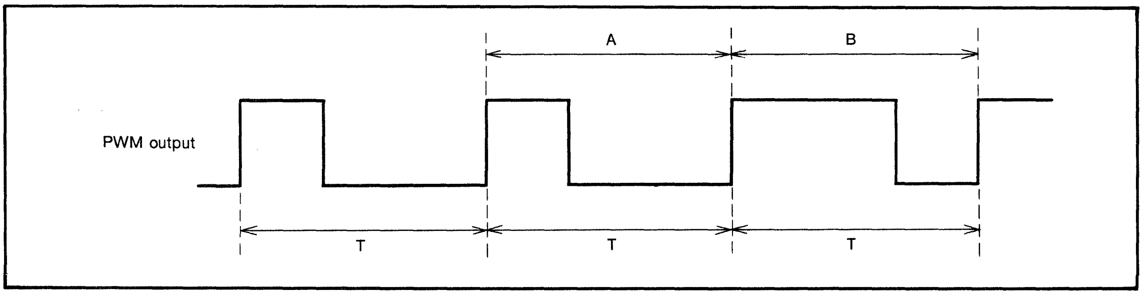


Fig. 2.11.3 PWM output renewal

2.12 Multiplier/Divider unit

M37450 has a built-in hardware multiplier/divider unit with the featrues shown below.

Multiplier	8-bit×8-bit 6μs (at the clock frequency 10MHz)
Divider	16-bit÷8-bit 6.4μs (at the clock frequency 10MHz)

In M37450 with the built-in multiplier/divider unit, the multiplication/division instructions have been established in addition to the 69 basic instructions in the conventional series MELPS 740. These instructions are explained below.

2.12.1 Multiplication instruction (MUL)

Table 2.12.1 shows the functions of the MUL instruction.

Table 2.12.1 Multiplication instruction

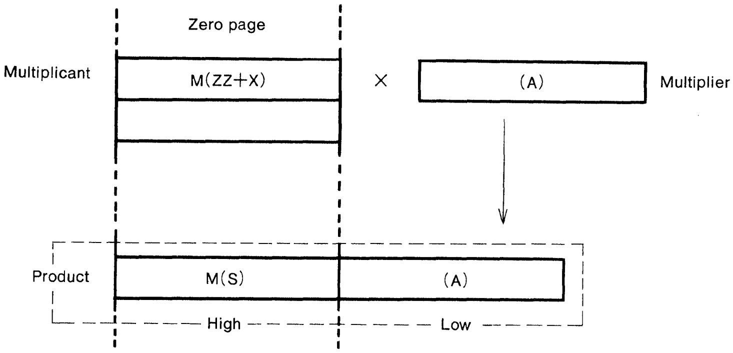
MUL (MULtiply accumulator and memory)

Function : $M(S) \cdot (A) \leftarrow (A) \times M(ZZ+X)$
$(S) \leftarrow (S) - 1$

Explanation : Multiplies accumulator with the memory specified by zero page X addressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.

Status flag : No change

Register : Accumulator changes
$(S) \leftarrow (S) - 1$
No change in the state of the other registers



Addressing mode	Mnemonic	Machine code	Byte number	Cycle number
Zero page, X	MUL $ZZ, X	62_{16}, ZZ_{16}	2	15

Note : When this instruction is executed, the contents of the accumulator and the stack pointer will change.

2.12.2 Division instruction

Table 2.12.2 shows the functions of the division instruction.
In the division instruction of the M37450, for there is no check for a divide by "0" or for a quotient exceeding 8 bits. Caution must also be paid to the remainder since it is 1's complement.
Precautions can be taken as shown in Figure 2.12.1.

Table 2.12.2 Division instruction

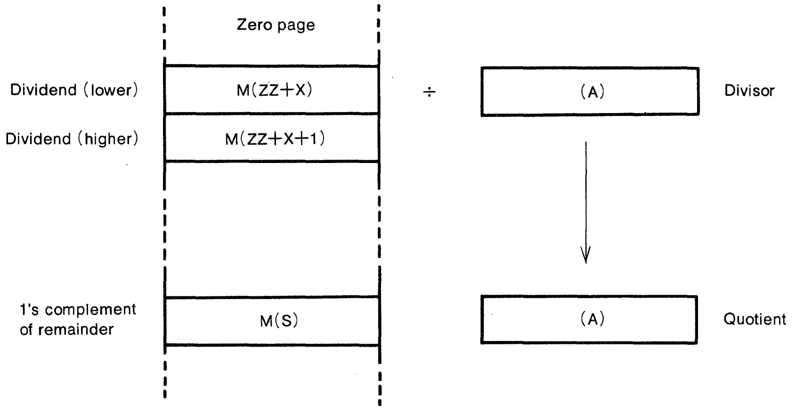
DIV (DIVide memory by accumulator)

Function : (A) ← (M(ZZ+X+1), M(ZZ+X))/(A)
M(S) ← 1's complement of remainder.
(S) ← (S) − 1

Explanation : Divides by acumulator the 16-bit data that is the cotents of M(ZZ+X+1) for high byte and contents of the next address memory for low byte, and stores the quotient in the accumulator and the remainder on the stack as 1's complement.

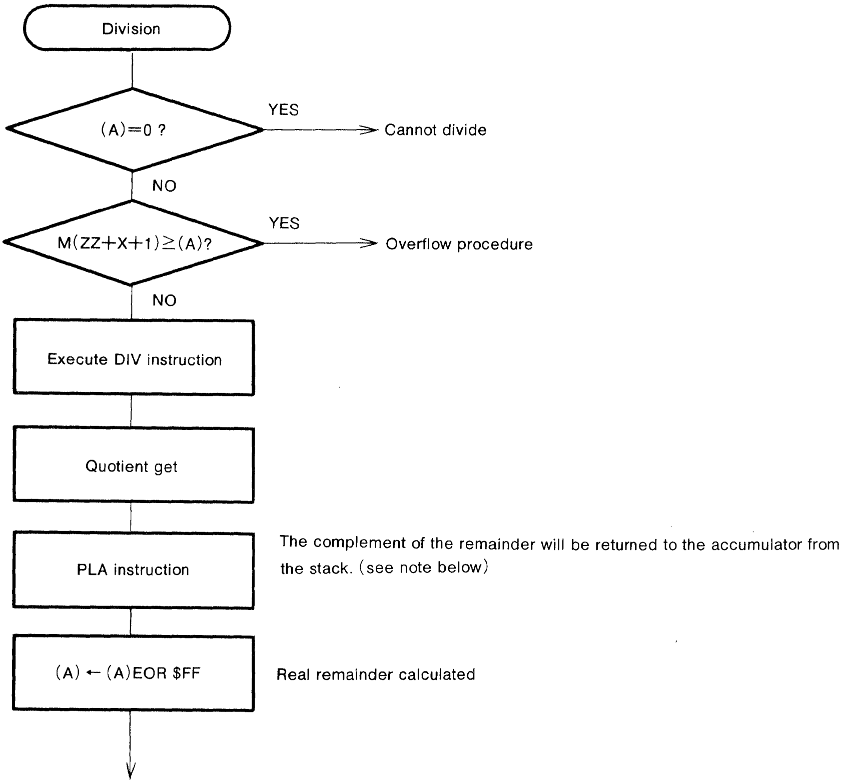
Status flag : No change

Register : Accumulator changes
(S) ← (S) − 1
No change in the state of the other registers



Addressing mode	Mnemonic	Machine code	Byte number	Cycle number
Zero page, X	DIV $ZZ, X	$E2_{16}$, ZZ_{16}	2	16

Note : The instruction does not check for an overflow of the quotient or zero division. A software check is necessary before the DIV instruction.
The contents of the accumulator and the stack pointer will change.



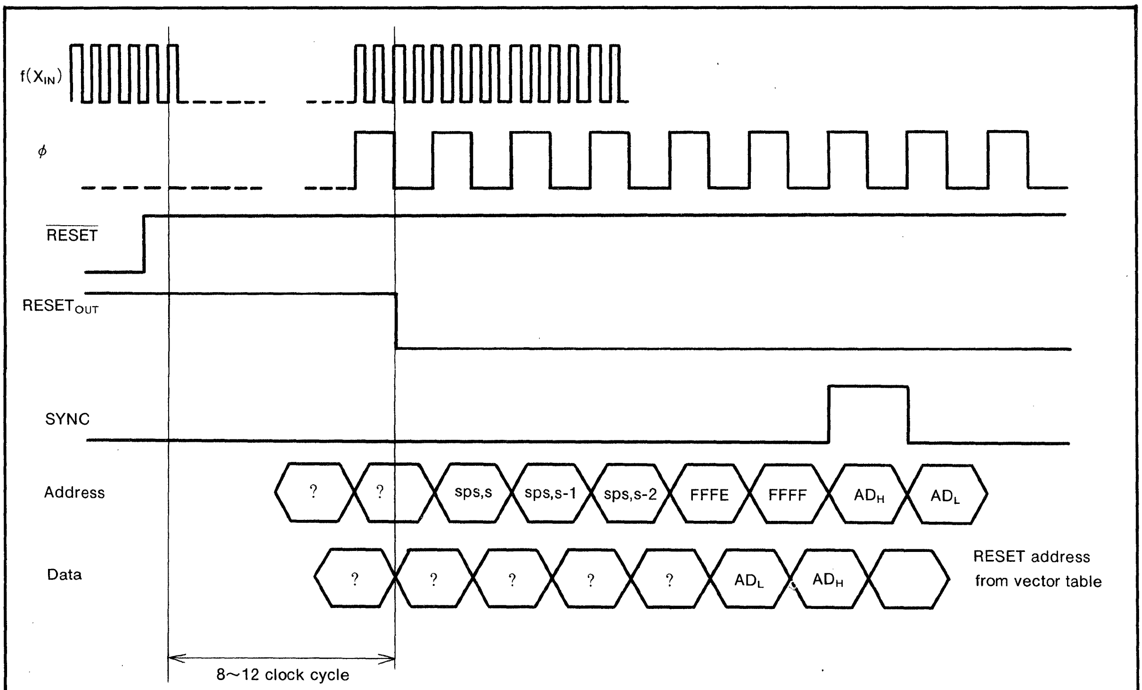
Note : When the division instruction is executed, the stack pointer changes. Therefore if a stack pointer manipulation is used such as a PLA instruction, it must be executed after the question is obtained even if the remainder is not used.

Fig. 2.12.1 Data processing before DIV instruction

3. RESET CIRCUIT

3.1 Reset operation

When the M37450 is supplied 5V±10% and $\overline{\text{RESET}}$ pin is in "L" level more than 8 ϕ clocks, the M37450 is reset. If more time is needed for the crystal oscillator to stabilize, this "L" condition should be maintained for the reguired time. When the $\overline{\text{RESET}}$ pin becomes high, the sequence shown in Figure 3.1.1 starts. The program will start at the address indicated by the $\overline{\text{RESET}}$ vector, address, $FFFE_{16}$ and $FFFF_{16}$.



Note 1 : Clock output signal, ϕ, is $f(X_{IN})$ divided by 4.
2 : The mark " ? " indicates an undefined address.

Fig. 3.1.1 Timing diagram at reset

3.2 Internal condition of microcomputer after reset

Figures 3.2.1 and 3.2.2 shows the internal state following reset.

	Address	Register contents
Port P0 directional register	$00D1_{16}$	00_{16}
Port P1 directional register	$00D3_{16}$	00_{16}
Port P2 directional register	$00D5_{16}$	00_{16}
Port P3 directional register	$00D7_{16}$	00_{16}
Port P5 directional register	$00DB_{16}$	00_{16}
Port P6 directional register	$00DD_{16}$	00_{16}
MISRG 1	$00DE_{16}$	□ 0 0 0 □ 0 0 0
MISRG 2	$00DF_{16}$	00_{16}
D-A_1 register	$00E0_{16}$	00_{16}
D-A_2 register	$00E1_{16}$	00_{16}
Data bus buffer status register	$00E5_{16}$	□ □ □ □ □ □ 0 1
Serial I/O status register	$00E7_{16}$	□ 0 0 0 0 0 0 0
Serial I/O control register	$00E8_{16}$	00_{16}
UART control register	$00E9_{16}$	□ □ □ □ 0 0 0 0
Timer 1 control register	$00ED_{16}$	□ □ 0 0 0 0 0 0
Timer 2 control register	$00EE_{16}$	□ □ 0 0 0 0 0 0
Timer 3 control register	$00EF_{16}$	□ □ 0 0 0 0 0 0

Fig. 3.2.1 Internal state of microcomputer at reset (1)

	Address	Register contents
Timer 1 register (lower)	$00F0_{16}$	FF_{16}
Timer 1 register (upper)	$00F1_{16}$	03_{16}
Interrupt request register 1	$00FC_{16}$	00_{16}
Interrupt request register 2	$00FD_{16}$	□ □ 0 0 0 0 0 0
Interrupt control register 1	$00FE_{16}$	00_{16}
Interrupt control register 2	$00FF_{16}$	□ □ 0 0 0 0 0 0
Processor status register	(PS)	□ □ □ □ □ 1 □ □
Program counter	(PC_H)	Contents of address $FFFF_{16}$
	(PC_L)	Contents of address $FFFE_{16}$

Note : The other registers and the RAM are undefined at reset, it is necessary to set initial values.

Fig. 3.2.2 Internal state of microcomputer at reset (2)

3.3 Reset circuit

The reset input voltage must be kept below 0.6V until the supply voltage surpass 4.5V.

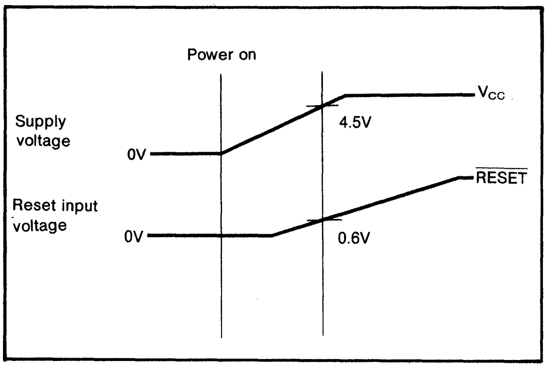


Fig. 3.3.1 Power on reset condition

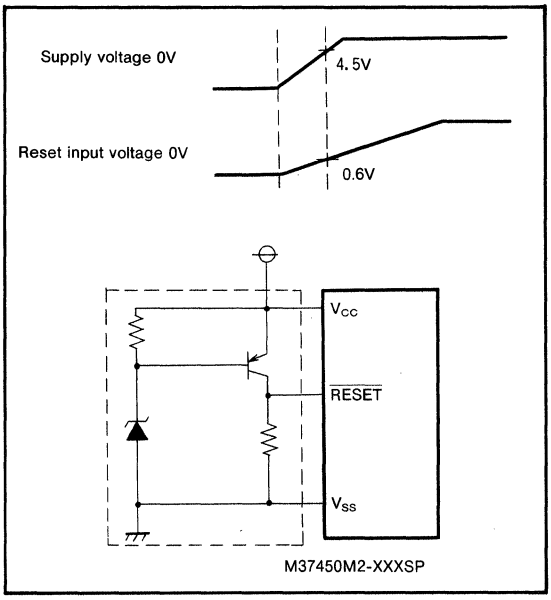


Fig. 3.3.2 Power on reset circuit

4. OSCILLATION CIRCUIT

4.1 Oscillation circuit

M37450 has an internal oscillation circuit. The internal clock ϕ is the frequency input into the clock input pin X_{IN} divided by 4. A ceramic or crystal resonator can be used for the element to be attached to this circuit.

(1) The oscillation circuit using ceramic resonator or crystal resonator
Figures 4.1.1～4.1.3 show the case where a ceramic or crystal resonator is used. As shown in the figure, the oscillation circuit can be formed by connecting the resonator between X_{IN} and X_{OUT}. In this case, the values Rf, C_{IN}, C_{OUT}, should be set to the values recommended by the manufacturer of the oscillator.

(2) External clock input circuit.
M37450 also can accept an external clock. A example of the external clock input is shown in Figure 4.1.4. In this case the X_{OUT} is left open.

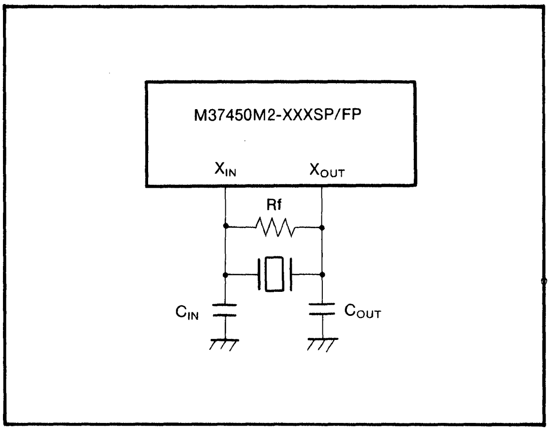


Fig. 4.1.1 Oscillation circuit example with a ceramic resonator

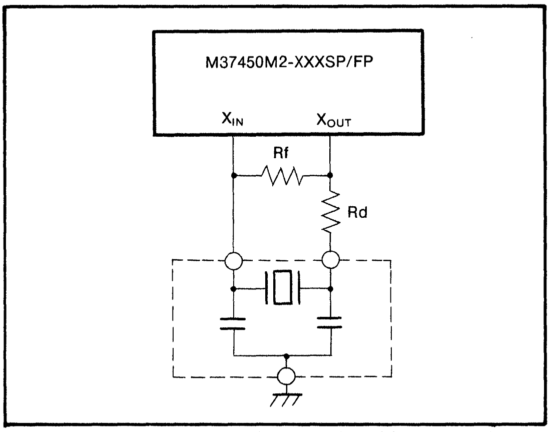


Fig. 4.1.2 Oscillation circuit example with a ceramic resonator (Built-in capacitor type)

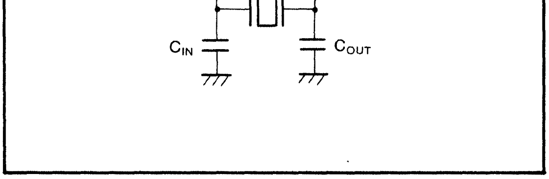


Fig. 4.1.3 Oscillation circuit example with a crystal resonator

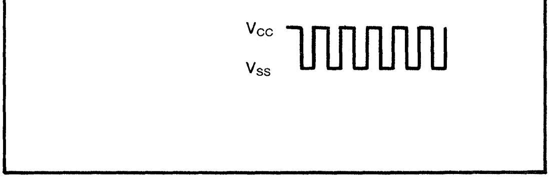


Fig. 4.1.4 Oscillation circuit example with a external oscillation source

4.2 Clock generating circuit

The oscillation circuit consists of an oscillating gate which works as an amplifier to obtain necessary gain for oscillation and the oscillation control flip-flop, which is used to stop or restart oscillation when necessary.

M37450 has a built-in clock generating circuit as shown in Figure 4.2.1.

In M37450, the oscillation can be stopped by executing STP instruction while retaining the contents of all registers except for timer 1 register (stop mode). When operation resumes after the STP, it can start-up in the same state as when the oscillation stopped.

When the STP instruction is executed, the oscillation stops with the internal clock ϕ in a "H" state. Also, timer 1 is set automatically, the lower byte to "FF_{16}" and the upper byte to "03_{16}". The count source of the timer 1 is forcibly connected to $f(X_{IN})/4$. This count source is rearranged to the original source when the stop mode is cancelled.

The oscillation will restart by a reset or an external interrupt. When an external interrupt is accepted, timer 1 starts counting. When it overflows, the internal clock ϕ is supplied. The delay caused by timer 1 is used since it takes time for the oscillation to stabilize when a ceramic resonator, etc. is used.

Before executing STP instruction timer 1 interrupt enable bit must be set to "0" (disable) and the timer 1 count enable bit must be set to "1" (enable). Also, one of the external interrupts must be in the interrupt enable condition (interrupt enable bit to "1"; and interrupt disable flag set to "0").

By using the WIT instruction the internal clock ϕ stops without stopping the oscillator (wait mode). In this case, since the oscillator is not in the stop condition, it is not necessary to establish a wait period to allow the oscillator to stabilize; it can resume operation immediately after cancellation of the wait condition. In order to recover from the wait condition, a reset or an interrupt is needed. Since the oscillator has not stopped, it is possible to use the internal interrupt such as the timer to restart.

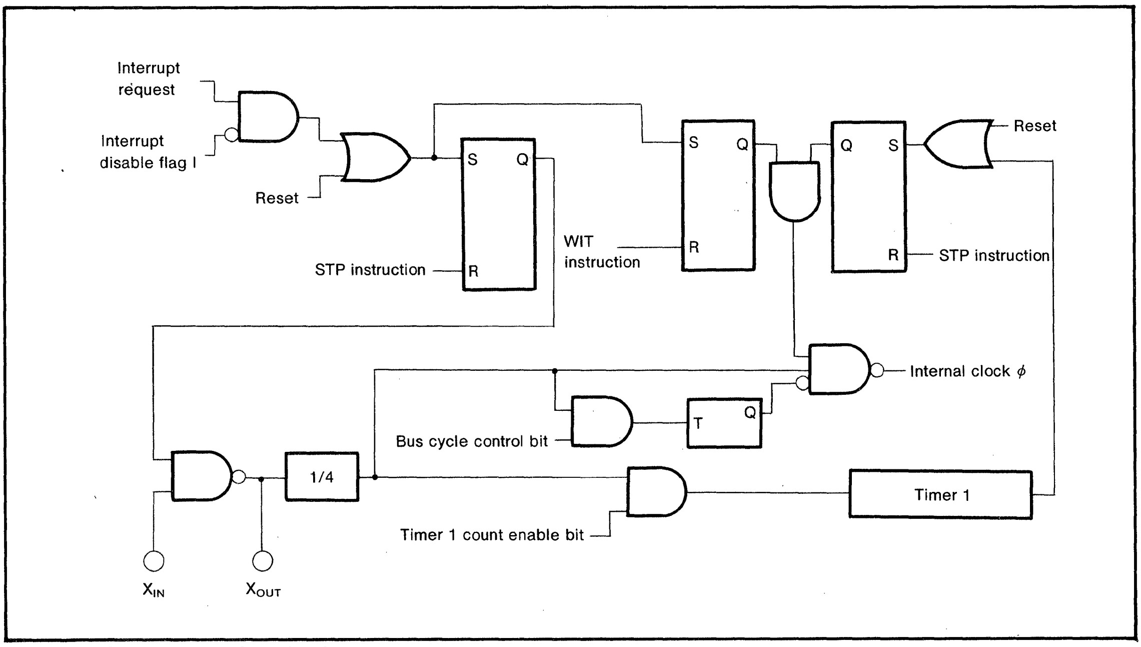


Fig. 4.2.1 Clock generating circuit

5. ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage	With respect to V_{SS} output transistors are at "off" state.	−0.3~7	V
V_I	Input voltage X_{IN}, $\overline{RESET}$		−0.3~7	V
V_I	Input voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_7$, $P4_0$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, ADV_{REF}, DAV_{REF}, V_{REF}, AV_{CC}		−0.3~V_{CC}+0.3	V
V_I	Input voltage CNV_{SS}		−0.3~13	V
V_O	Output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$, $P3_0$~$P3_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, X_{OUT}, φ R/$\overline{W}$, $\overline{RD}$, $\overline{WR}$, SYNC, $RESET_{OUT}$		−0.3~V_{CC}+0.3	V
Pd	Power dissipation	T_a = 25℃	1000(Note 1)	mW
Topr	Operating temperature		−10~70	℃
Tstg	Storage temperature		−40~125	℃

Note 1 : 500mW in case of the flat package

RECOMMENDED OPERATING CONDITIONS

(V_{CC}=5V±10%, Ta=−10~70℃ unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Nom.	Max.	
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{SS}	Supply voltage		0		V
V_{IH}	"H" input voltage $\overline{RESET}$, X_{IN}, CNV_{SS} (Note 2)	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$ (expect Note 2)	2.0		V_{CC}	V
V_{IL}	"L" input voltage CNV_{SS} (Note 2)	0		0.2V_{CC}	V
V_{IL}	"L" input voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$ (expect Note 2)	0		0.8	V
V_{IL}	"L" input voltage $\overline{RESET}$	0		0.12V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.16V_{CC}	V
$I_{OL(peak)}$	"L" peak output current $P0_0$~$P0_7$, $P1_0$~$P1_7$ $P2_0$~$P2_7$, $P3_0$~$P3_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$			10	mA
$I_{OL(avg)}$	"L" average output current $P0_0$~$P0_7$, $P1_0$~$P1_7$ $P2_0$~$P2_7$, $P3_0$~$P3_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$ (Note 3)			5	mA
$I_{OH(peak)}$	"H" peak output current $P0_0$~$P0_7$, $P1_0$~$P1_7$ $P2_0$~$P2_7$, $P3_0$~$P3_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$			−10	mA
$I_{OH(avg)}$	"H" average output current $P0_0$~$P0_7$, $P1_0$~$P1_7$ $P2_0$~$P2_7$, $P3_0$~$P3_7$ $P5_0$~$P5_7$, $P6_0$~$P6_7$ (Note 3)			−5	mA
$f(X_{IN})$	Internal clock oscillating frequency	1		10	MHz

Note 2 : Ports operating as special function pins INT_1~INT_3($P6_0$~$P6_2$), EV_1~EV_3($P3_0$~$P3_2$), R_XD($P3_4$), S_{CLK}($P3_6$)

Note 3 : $I_{OL(avg)}$ and $I_{OH(avg)}$ are the average current in 100ms.

Note 4 : The total of I_{OL} of port P0, P1 and P2 should be 40mA (max.).
The total of I_{OL} of port P3, P5, P6, R/$\overline{W}$ SYNC, $RESET_{OUT}$, $\overline{RD}$, $\overline{WR}$ and φ should be 40mA (max.).
The total of I_{OH} of port P0, P1, and P2 should be 40mA (max.).
The total of I_{OH} of port P3, P5, P6, R/$\overline{W}$, SYNC, $RESET_{OUT}$, $\overline{RD}$, $\overline{WR}$, and φ should be 40mA (max.).

ELECTRICAL CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}= 0 V, T_a=−10~70℃, f(X_{IN})=10MHz)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, SYNC, $RESET_{OUT}$, ϕ	I_{OH} =− 2 mA	V_{CC}−1			V
V_{OH}	"H" output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$	I_{OH} =− 5 mA	V_{CC}−1			V
V_{OL}	"L" output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$ $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, SYNC, $RESET_{OUT}$, ϕ	I_{OL} =2 mA			0. 45	V
V_{OL}	"L" output voltage $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$	I_{OL} =5 mA			1	V
V_{T+} − V_{T-}	Hysteresis INT_1~INT_3($P6_0$~$P3_2$), EV_1~EV_3($P3_0$~$P3_2$) R_XD($P3_4$), S_{CLK}($P3_6$)	Function input level	0. 3		1	V
V_{T+} − V_{T-}	Hysteresis $\overline{RESET}$				0. 7	V
V_{T+} − V_{T-}	Hysteresis X_{IN}		0. 1		0. 5	V
I_{IL}	"L" input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $\overline{RESET}$, X_{IN}	V_I =V_{SS}	−5		5	μA
I_{IH}	"H" input current $P0_0$~$P0_7$, $P1_0$~$P1_7$, $P2_0$~$P2_7$ $P3_0$~$P3_7$, $P4_0$~$P4_7$, $P5_0$~$P5_7$ $P6_0$~$P6_7$, $\overline{RESET}$, X_{IN}	V_I =V_{CC}	−5		5	μA
V_{RAM}	RAM retention voltage	At stop mode	2			V
I_{CC}	Supply current	f(X_{IN})=10MHz At system operation		6	10	mA
		At stop mode (Note 5)		1	10	μA

Note 5 : The terminals $\overline{RD}$, $\overline{WR}$, SYNC, R/$\overline{W}$, $RESET_{OUT}$, ϕ, X_{OUT}, D-A_1 and D-A_2 are all open.The other ports, which are in the input mode, are connected to V_{SS}. A-D converter is in the A-D completion state. The current through ADV_{REF} and DAV_{REF} is not included. (Fig. 5.4)

A-D CONVERTER CHARACTERISTICS (V_{CC}=AV_{CC}= 5 V, V_{SS}=AV_{SS}= 0 V, T_a =25℃, f(X_{IN})=10MHz unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute accuracy	V_{CC}=AV_{CC}=ADV_{REF}=5. 12V		±1. 5	±3	LSB
t_{CONV}	Conversion time				49	$t_{c(\phi)}$
V_{IA}	Analog input voltage		AV_{SS}		AV_{CC}	V
V_{ADVREF}	Reference input voltage		2		V_{CC}	V
R_{LADDER}	Ladder resistance value	ADV_{REF}= 5 V	2	7. 5	10	kΩ
$I_{IADVREF}$	Reference input current	ADV_{REF}= 5 V	0. 5	0. 7	2. 5	mA
V_{AVCC}	Analog power supply input voltage			V_{CC}		V
V_{AVSS}	Analog power supply input voltage			0		V

D-A CONVERTER CHARACTERISTICS (V_{CC}= 5 V, V_{SS}=AV_{SS}= 0 V, T_a =25℃ unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Full scale deviation	V_{CC}=DAV_{REF}=5V			1. 0	%
t_{SU}	Set time				3	μs
R_O	Output resistance		1	2	4	kΩ
V_{AVSS}	Analog power supply input voltage			0		V
V_{DAVREF}	Reference input voltage		4		V_{CC}	V
I_{DAVREF}	Reference power input current		0	2. 5	5	mA

TIMING REQUIREMENTS

Port/single-chip mode (V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{su(P0D-\phi)}$	Port P0 input setup time	Fig. 5.1	200			ns
$t_{su(P1D-\phi)}$	Port P1 input setup time		200			ns
$t_{su(P2D-\phi)}$	Port P2 input setup time		200			ns
$t_{su(P3D-\phi)}$	Port P3 input setup time		200			ns
$t_{su(P4D-\phi)}$	Port P4 input setup time		200			ns
$t_{su(P5D-\phi)}$	Port P5 input setup time		200			ns
$t_{su(P6D-\phi)}$	Port P6 input setup time		200			ns
$t_{h(\phi-P0D)}$	Port P0 input hold time		40			ns
$t_{h(\phi-P1D)}$	Port P1 input hold time		40			ns
$t_{h(\phi-P2D)}$	Port P2 input hold time		40			ns
$t_{h(\phi-P3D)}$	Port P3 input hold time		40			ns
$t_{h(\phi-P4D)}$	Port P4 input hold time		40			ns
$t_{h(\phi-P5D)}$	Port P5 input hold time		40			ns
$t_{h(\phi-P6D)}$	Port P6 input hold time		40			ns
$t_c(X_{IN})$	External clock input cycle time		100		1000	ns
$t_w(X_{INL})$	External clock input "L" pulse width		30			ns
$t_w(X_{INH})$	External clock input "H" pulse width		30			ns
$t_r(X_{IN})$	External clock rising edge time				20	ns
$t_f(X_{IN})$	External clock falling edge time				20	ns

Master CPU bus interface timing ($\overline{R}$ and $\overline{W}$ separation type mode)

(V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{su(CS-R)}$	$\overline{CS}$ setup time	Fig. 5.2	0			ns
$t_{su(CS-W)}$	$\overline{CS}$ setup time		0			ns
$t_{h(R-CS)}$	$\overline{CS}$ hold time		0			ns
$t_{h(W-CS)}$	$\overline{CS}$ hold time		0			ns
$t_{su(A-R)}$	A_0 setup time		40			ns
$t_{su(A-W)}$	A_0 setup time		40			ns
$t_{h(R-A)}$	A_0 hold time		10			ns
$t_{h(W-A)}$	A_0 hold time		10			ns
$t_{w(R)}$	Read pulse width		160			ns
$t_{w(W)}$	Write pulse width		160			ns
$t_{su(D-W)}$	Date input setup time before write		100			ns
$t_{h(W-D)}$	Date input hold time after write		10			ns

Master CPU bus interface timing (R/$\overline{W}$ type mode)

(V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{su(CS-E)}$	$\overline{CS}$ setup time	Fig. 5.2	0			ns
$t_{h(E-CS)}$	$\overline{CS}$ hold time		0			ns
$t_{su(A-E)}$	A_0 setup time		40			ns
$t_{h(E-A)}$	A_0 hold time		10			ns
$t_{su(RW-E)}$	R/$\overline{W}$ setup time		40			ns
$t_{h(E-RW)}$	R/$\overline{W}$ hold time		10			ns
$t_{w(EL)}$	Enable clock "L" pulse width		160			ns
$t_{w(EH)}$	Enable clock "H" pulse width		160			ns
$t_{r(E)}$	Enable clock rising edge time				25	ns
$t_{f(E)}$	Enable clock falling edge time				25	ns
$t_{su(D-E)}$	Data input setup time before write		100			ns
$t_{h(E-D)}$	Data input hold time after write		10			ns

Local bus/memory expansion mode, microprocessor mode

(V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{su(D-\phi)}$	Data input setup time	Fig. 5. 3	130			ns
$t_{h(\phi-D)}$	Data input hold time		0			ns
$t_{su(D-RD)}$	Data input setup time		130			ns
$t_{h(RD-D)}$	Data input hold time		0			ns

SWITCHING CHARACTERISTICS

Port/single-chip mode (V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(\phi-P0Q)}$	Port P0 data output delay time	Fig. 5. 1			200	ns
$t_{d(\phi-P1Q)}$	Port P1 data output delay time				200	ns
$t_{d(\phi-P2Q)}$	Port P2 data output delay time				200	ns
$t_{d(\phi-P3Q)}$	Port P3 data output delay time				200	ns
$t_{d(\phi-P5Q)}$	Port P5 data output delay time				200	ns
$t_{d(\phi-P6Q)}$	Port P6 data output delay time				200	ns
$t_{c(\phi)}$	Cycle time		400		4000	ns
$t_{W(\phi H)}$	ϕ clock pulse width ("H" level)		190			ns
$t_{W(\phi L)}$	ϕ clock pulse width ("L" level)		170			ns
$t_{r(\phi)}$	ϕ clock rising edge time				20	ns
$t_{f(\phi)}$	ϕ clock falling edge time				20	ns

Master CPU bus interface ($\overline{R}$ and $\overline{W}$ separation type mode)

(V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{a(R-D)}$	Data output enable time after read	Fig. 5. 2			120	ns
$t_{v(R-D)}$	Data output disable time after read		10		85	ns
$t_{PLH(R-PR)}$	$\overline{P_{RDY}}$ output transmission time after read				150	ns
$t_{PLH(W-PR)}$	$\overline{P_{RDY}}$ output transmission time after write				150	ns

Master CPU bus interface (R/$\overline{W}$ type mode) (V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{a(E-D)}$	Data output enable time after read	Fig. 5. 2			120	ns
$t_{v(E-D)}$	Data output disable time after read		10		85	ns
$t_{PLH(E-PR)}$	$\overline{P_{RDY}}$ output transmission time after E clock				150	ns

Local bus/memory expansion mode, microprocessor mode

(V_{CC}=5V±10%, V_{SS}=0V, T_a=−10~70℃, unless otherwise noted)

Symbol	Parameter	Test condition	Limits			Unit
			Min.	Typ.	Max.	
$t_{d(\phi-A)}$	Address delay time after ϕ	Fig. 5. 3			150	ns
$t_{v(\phi-A)}$	Address effective time after ϕ		10			ns
$t_{v(RD-A)}$	Address effective time after $\overline{RD}$		10			ns
$t_{v(WR-A)}$	Address effective time after $\overline{WR}$		10			ns
$t_{d(\phi-D)}$	Data output delay time after ϕ				160	ns
$t_{d(WR-D)}$	Data output delay time after $\overline{WR}$				160	ns
$t_{v(\phi-D)}$	Data output effective time after ϕ		20			ns
$t_{v(WR-D)}$	Data output effective time after $\overline{WR}$		20			ns
$t_{d(\phi-RW)}$	R/$\overline{W}$ delay time after ϕ				150	ns
$t_{d(\phi-SYNC)}$	SYNC delay time after ϕ				150	ns
$t_{W(RD)}$	$\overline{RD}$ pulse width		170			ns
$t_{W(WR)}$	$\overline{WR}$ pulse width		170			ns

Test Condition

Input voltage level :	V_{IH}	2.4V
	V_{IL}	0.45V
Output test level :	V_{OH}	2.0V
	V_{OL}	0.8V

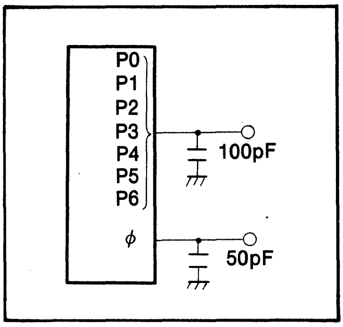


Fig. 5.1 Test circuit in single-chip mode

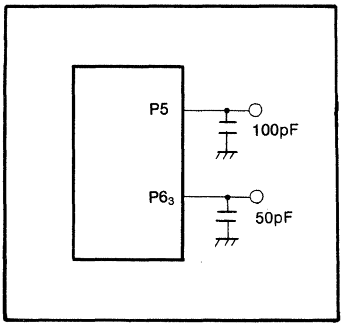


Fig. 5.2 Master CPU bus interface test circuit

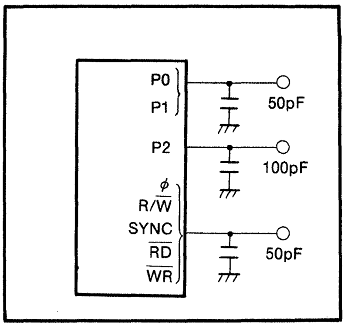


Fig. 5.3 Local bus test circuit

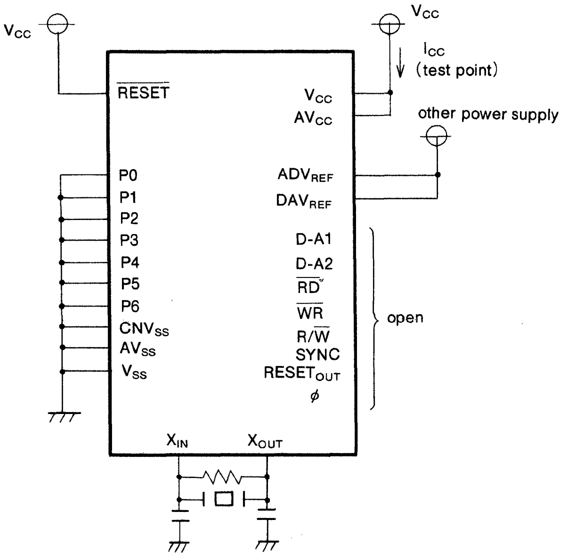
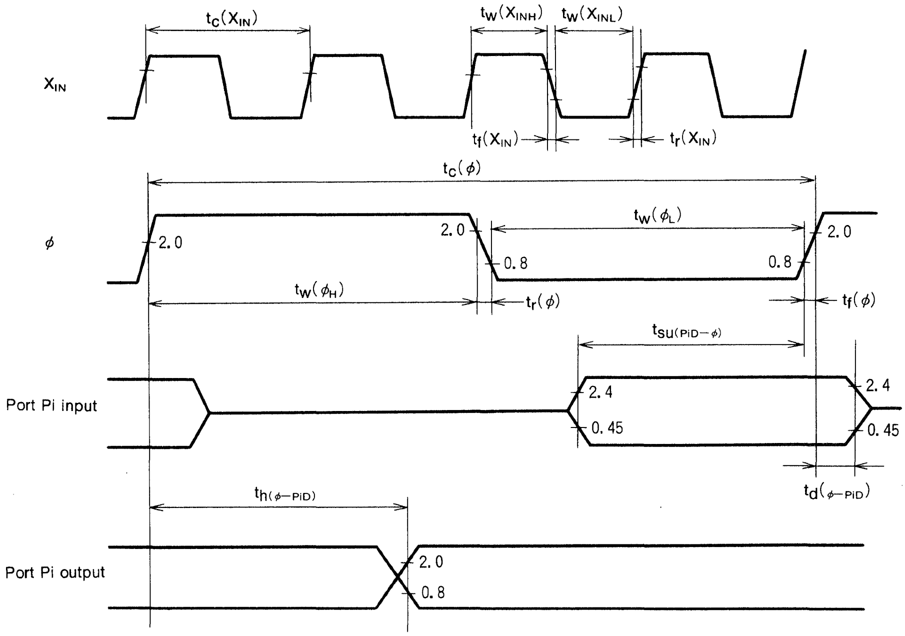


Fig. 5.4 I_{CC} (at stop mode) test condition

TIMING DIAGRAM

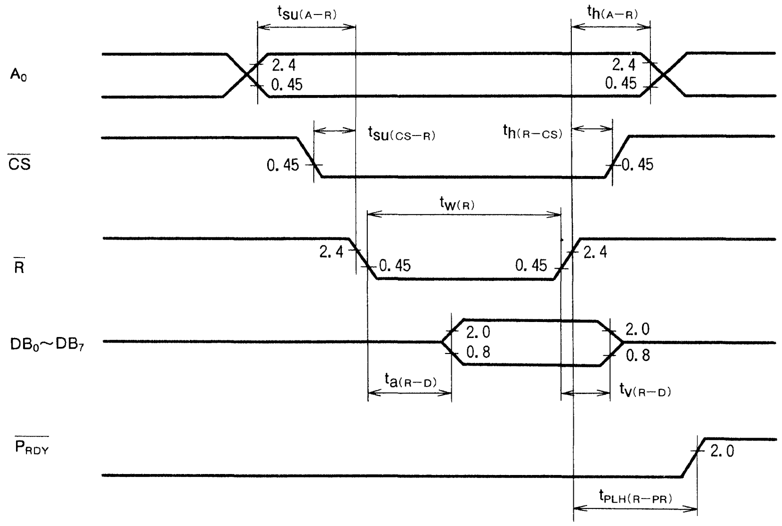
Port/Single-chip mode timing diagram



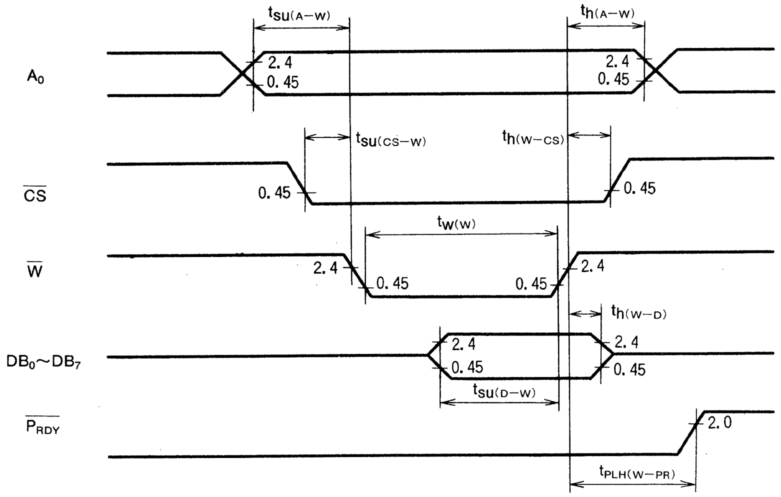
Note : $V_{IH}=0.8V_{CC}$, $V_{IL}=0.16V_{CC}$ of X_{IN}

Master CPU bus interface/$\overline{R}$ and $\overline{W}$ separation type timing diagram

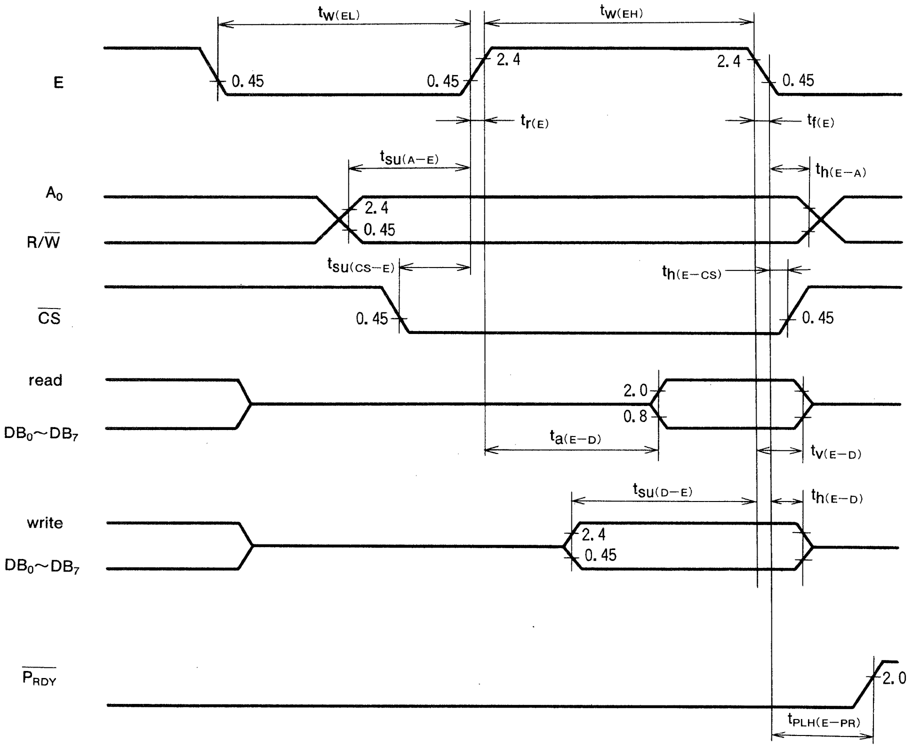
read



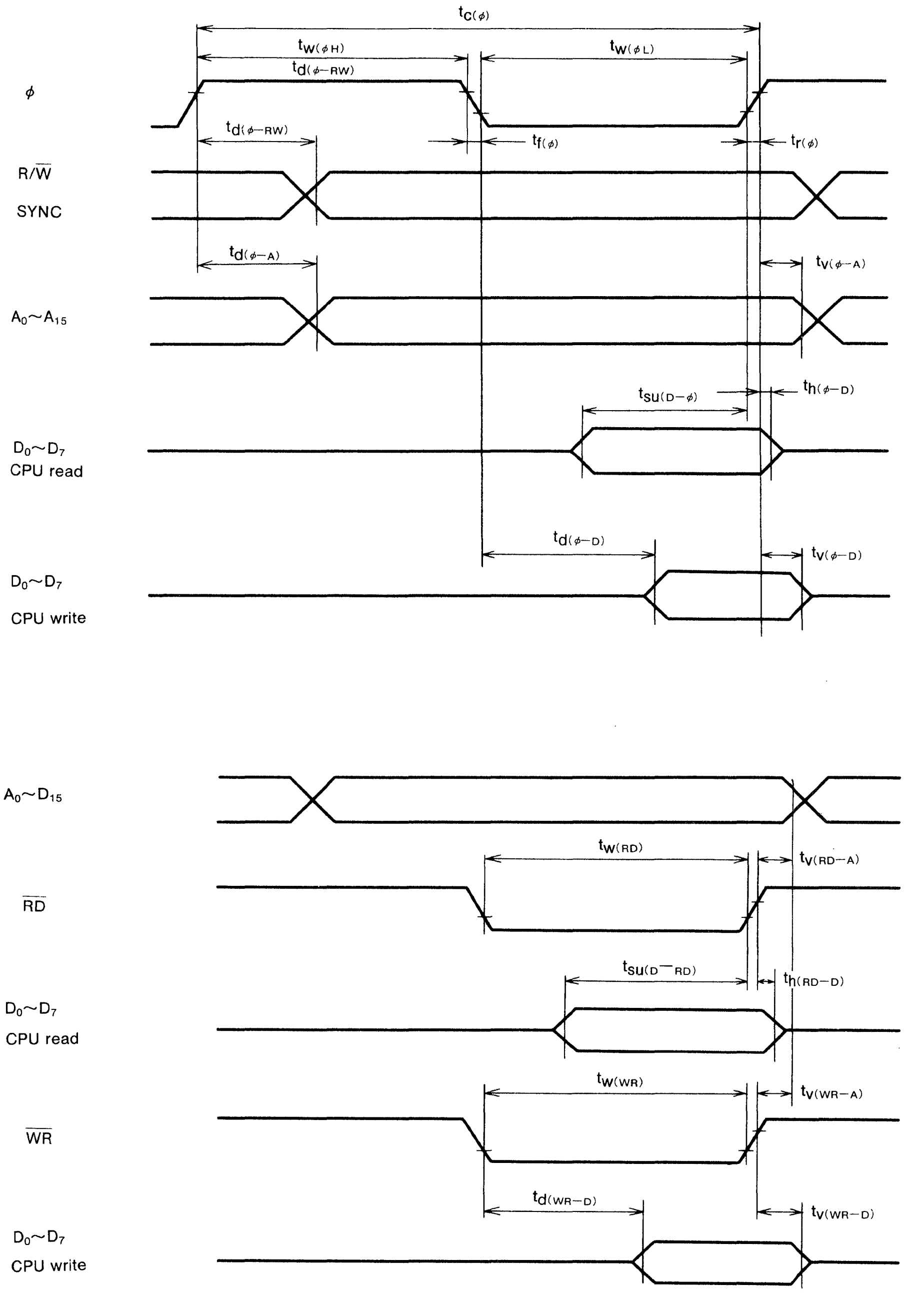
write



Master CPU interface/ R/$\overline{W}$ type timing diagram



Local bus timing diagram



6. TYPICAL CHARACTERISTICS

6.1 Port typical characteristics

Figure 6.1.1 and 6.1.2 shows the typical characteristics of the ports.

【Port P0$_0$ I_{OH}—V_{OH} characteristics (P-ch drive)】

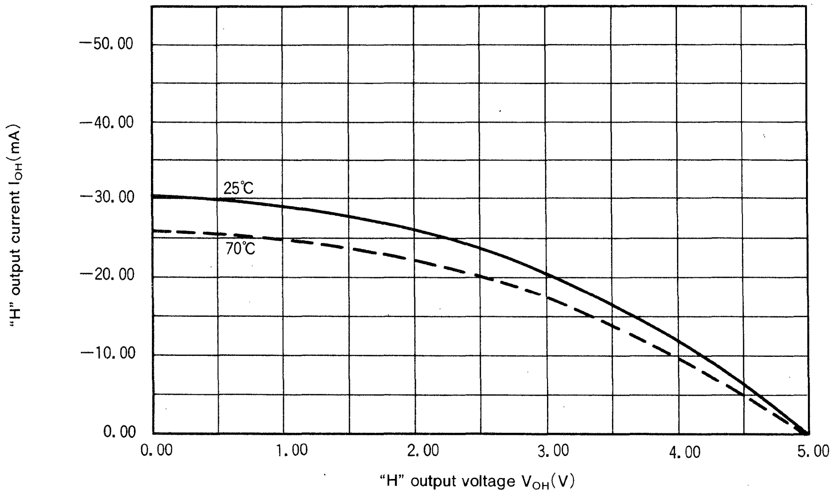


Fig.6.1.1 Port P0 typical characteristics at P-ch drive

【Port P0$_0$ I_{OL}—V_{OL} characteristics (N-ch drive)】

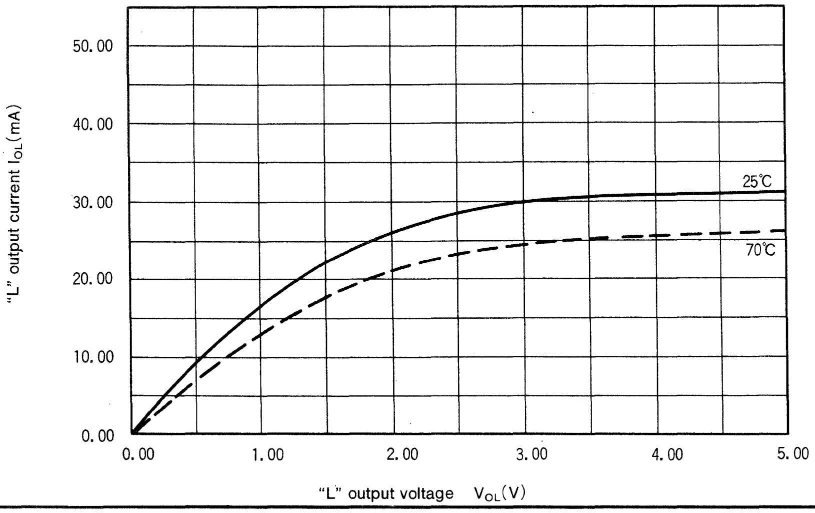


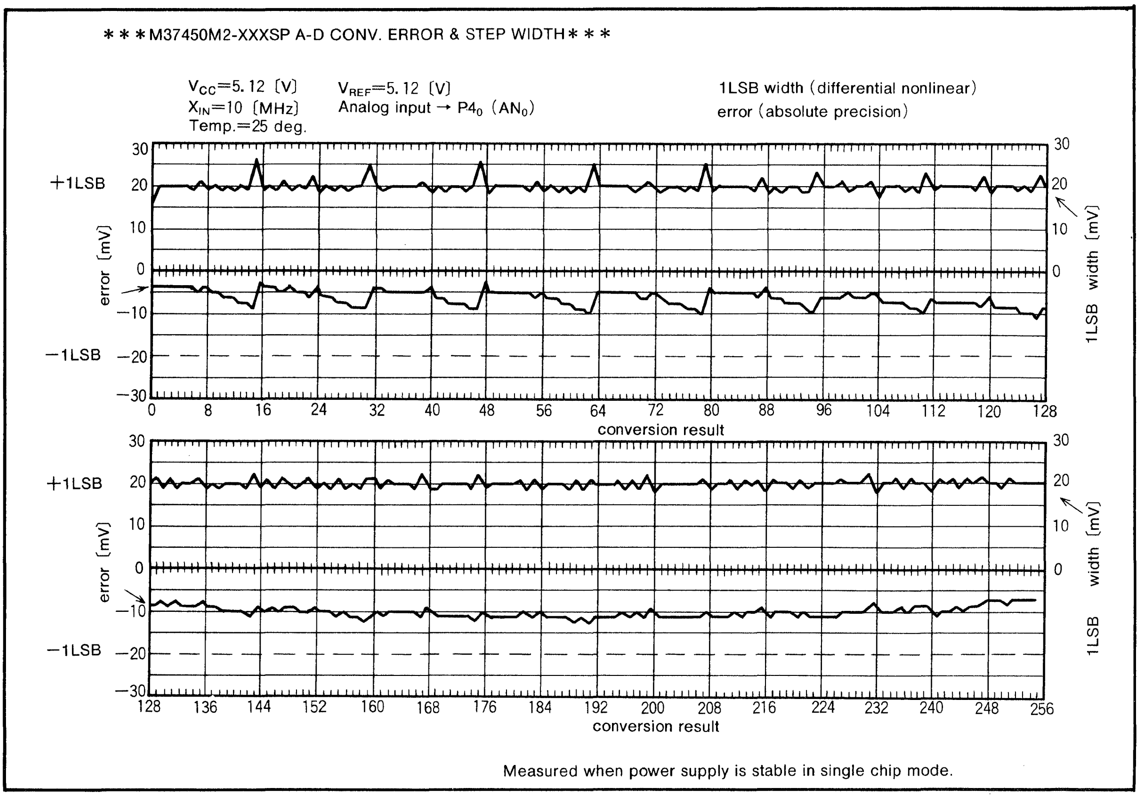
Fig.6.1.2 Port P0 typical characteristics at N-ch drive

6.2 A-D conversion typical characteristics

Figure 6.2.1 shows the typical chatacteristics of the A-D converter.

The bottom line in the graph shows the absolute precision error, this error being the shift from the ideal point of output code conversion. For example, the conversion of output code 00_{16} to 01_{16} ideally takes place at $AN_1 = 10mV$. However, since the measurement is $-4mV$, the measured point of conversion is $10-4=6mV$.

The top line in the graph shows the widths of input voltages equivalent to output codes. For example, the measured width of the input voltage for output code $0F_{16}$ is 26mV, so this line therfore shows that the differential nonlinear error is $26-20=6mV$ (0.3LSB).



Measured when power supply is stable in single chip mode.

Fig.6.2.1 A-D conversion typical characteristics

6.3 D-A conversion typical characteristics

Figure 6.3.1 shows the typical characteristics of the D-A converter.
The bottom line of the graph shows the absolute precision error, this error being the difference between the ideal analog output corresponding to the input code and the measured value.
The top line in the graph shows the width of the output analog value corresponding to the 1-bit change in the input code.

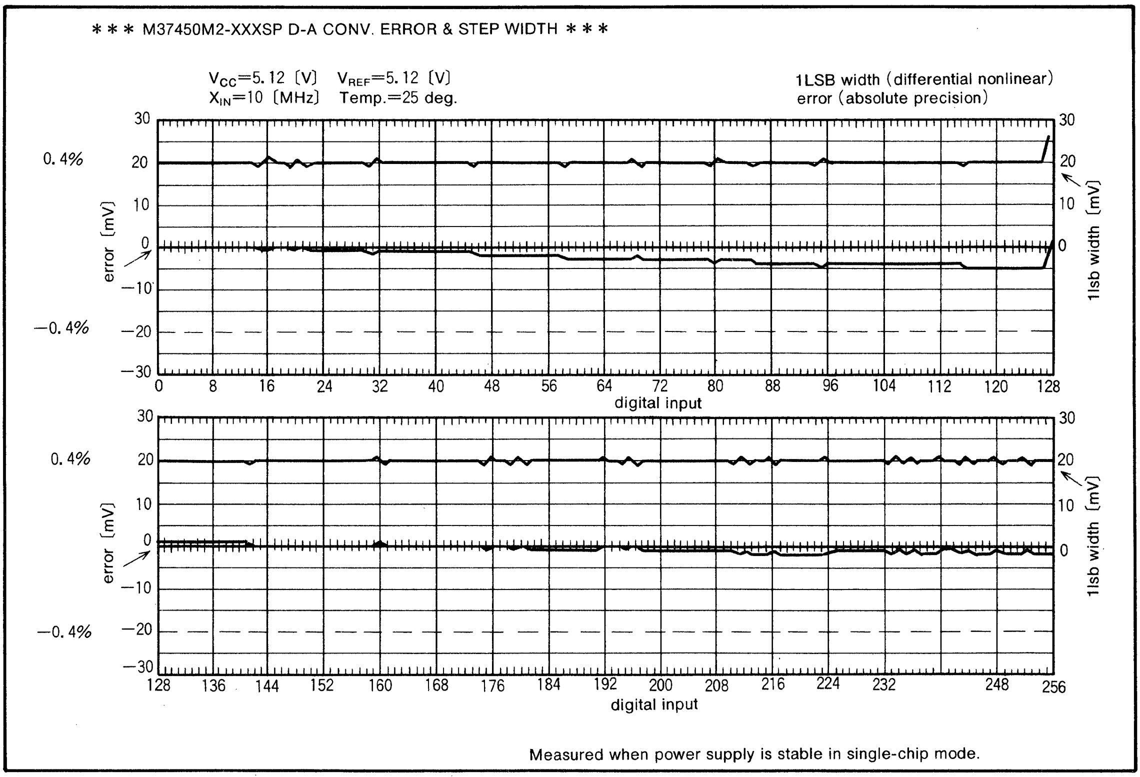


Fig.6.3.1 D-A conversion typical characteristics

7. APPLICATION

7.1 Memory expansion

The following figures explain the memory and port expansion methods when the M37450 is in memory expanding and microprocessor mode.

7.1.1 ROM expansion

Figure 7.1.1 and 7.1.2 shows the example of ROM expansion for 80-pin model and 64-pin model respectively.

Avalable EPROMs are determined by these caluculations:

CE access time (include decode time)

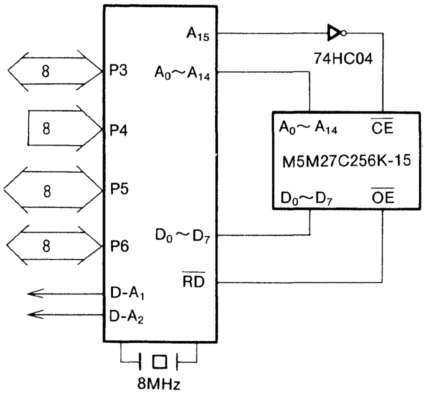
$$t_{a(CE)} \leqq \frac{4000}{f(X_{IN})} - (t_{r(\phi)} + t_{d(\phi-A)} + t_{su(D-\phi)})$$

OE access time

$$t_{a(OE)} \leqq \frac{2000}{f(X_{IN})} - 30) - t_{su(D-\phi)}$$

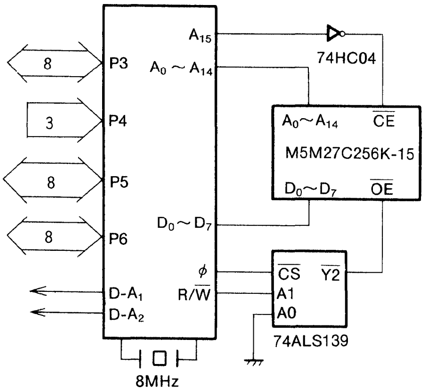
In 10MHz no-wait condition, the EPROM needs these timings:

$t_{a(CE)} \leqq 100\text{ns}$, $t_{a(OE)} \leqq 40\text{ns}$



Decode Address A_{15} and connect 32K-byte EPROM.
Use an EPROM which has less than 150ns access time for CPU in no-wait condition (RDYE = "0") when the CPU is working in f (X_{IN})=8MHz operation

Fig.7.1.1 M37450M2-XXXFP ROM expansion example

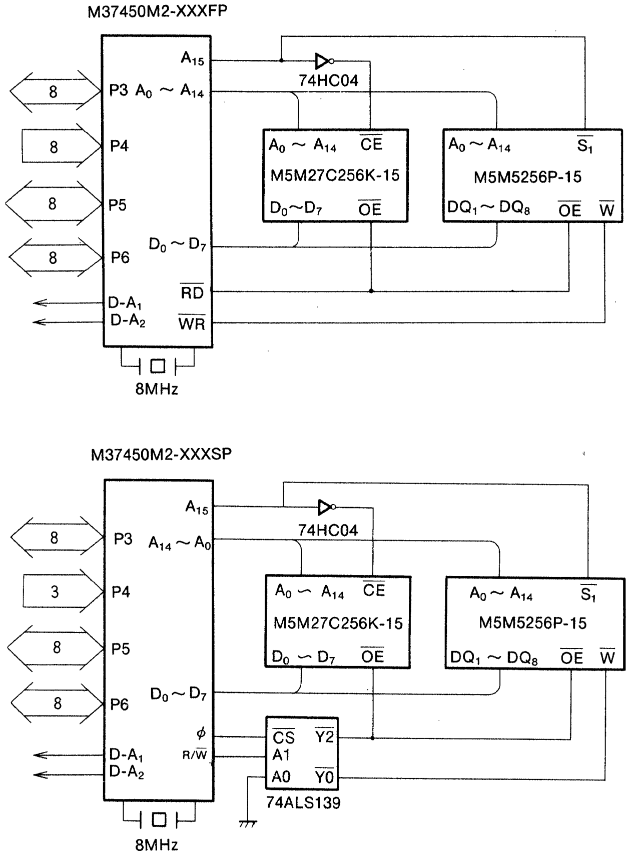


The signal which is almost same as $\overline{RD}$ can be generated by ϕ and R/$\overline{W}$ through 74ALS139.

Fig.7.1.2 M37450M2-XXXSP ROM expansion example

7.1.2 ROM and RAM expansion

Figure 7.1.3 shows the example of 32KB ROM expansion and 32KB RAM expansion.



Note. The 32K-byte RAM overlapped by the internal RAM and SFR, is write-only.

Available EPROMS are determined by these caluclations :

Read model

S1 access time (include decode time)

$$t_{a(S1)} \leqq \frac{4000}{f(X_{IN})} - (t_{r(\phi)} + t_{d(\phi-A)} + t_{su(D-\phi)})$$

OE access time

$$t_{a(OE)} \leqq \left(\frac{2000}{f(X_{IN})} - 30\right) - t_{su(D-\phi)}$$

Write model

$$t_{su(D)} \leqq \left(\frac{2000}{f(X_{IN})} - 30\right) - t_{d(\phi-D)}$$

Fig.7.1.3 Example for ROM and RAM expansion of M37450M2-XXXSP

7.1.3 Memory and port expansion

(1) Interface to M5M82C55AP-5

Figure 7.1.4 shows the port expansion circuit with the M5M82C55AP-5 and memories. The M5M82C55AP-5 must be used in mode 0. When using the M37450M2-XXXSP in 8MHz operation, use the ROM and RAM which satisfy the M37450M2-XXXSP's AC characteristics. However, in this case, the M5M82C55AP-5 can not satisfy the AC chatacteristics. To access the M5M82C55AP-5, set the RDYE flag (bit 6 of address 00DF$_{16}$) of the M37450M2-XXXSP to "1". By setting RDYE, the timing period of the M37450M2-XXXSP is expaned by two. This causes the M37450 to satisfy the M5M82C55AP-5's timing characteristics.

When using the M37450M2-XXXFP (80-pin type), the CPU outputs the $\overline{RD}$ and $\overline{WR}$ signals, so there is no need to separate the R/$\overline{W}$ signal.

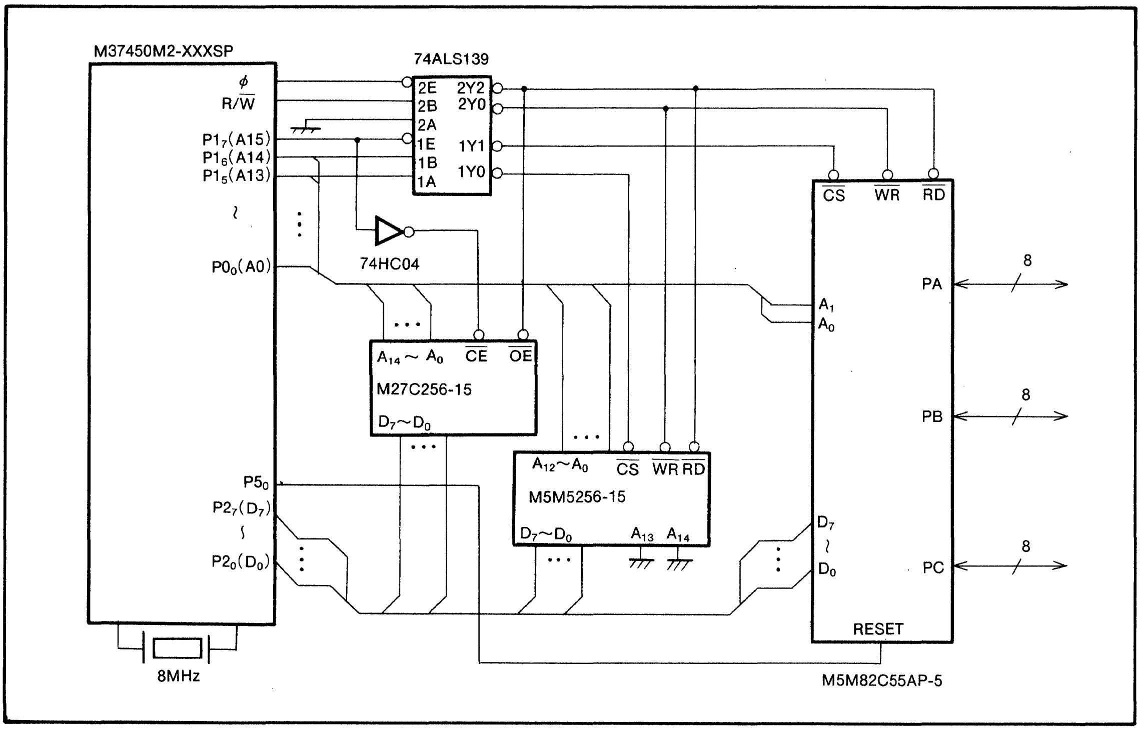


Fig.7.1.4 Example for interface to M5M82C55AP-5

(2) Timer function upgrade

Figure 7.1.5 explains how the timer LSI upgraded, M5M82C54, and EPROM interface to the M37450FP. The addresses of the M5M82C54P are addresses 4000_{16} to 4003_{16}. The RDYE flag must be set to "1" for the M37450 to synchronize with the M5M82C54P.

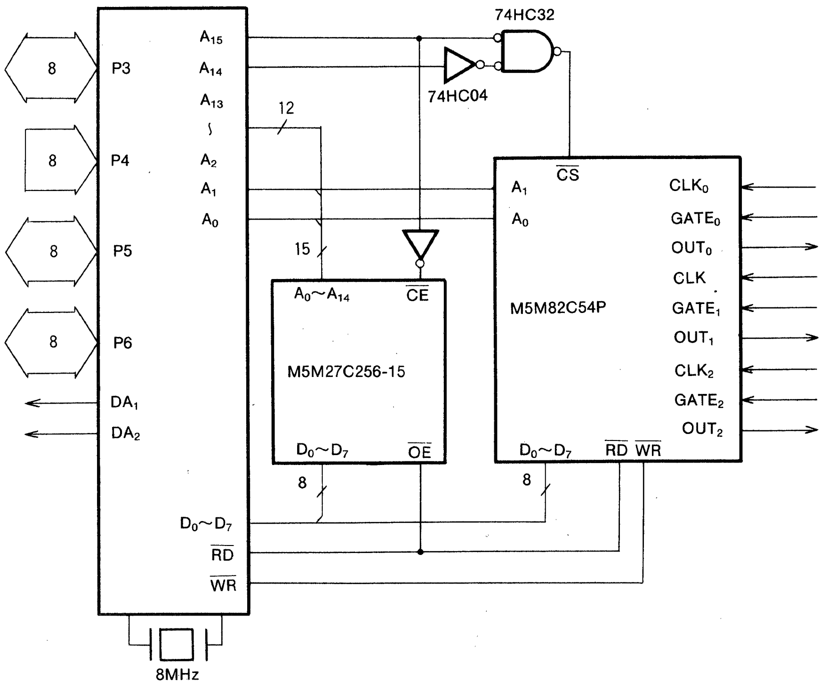
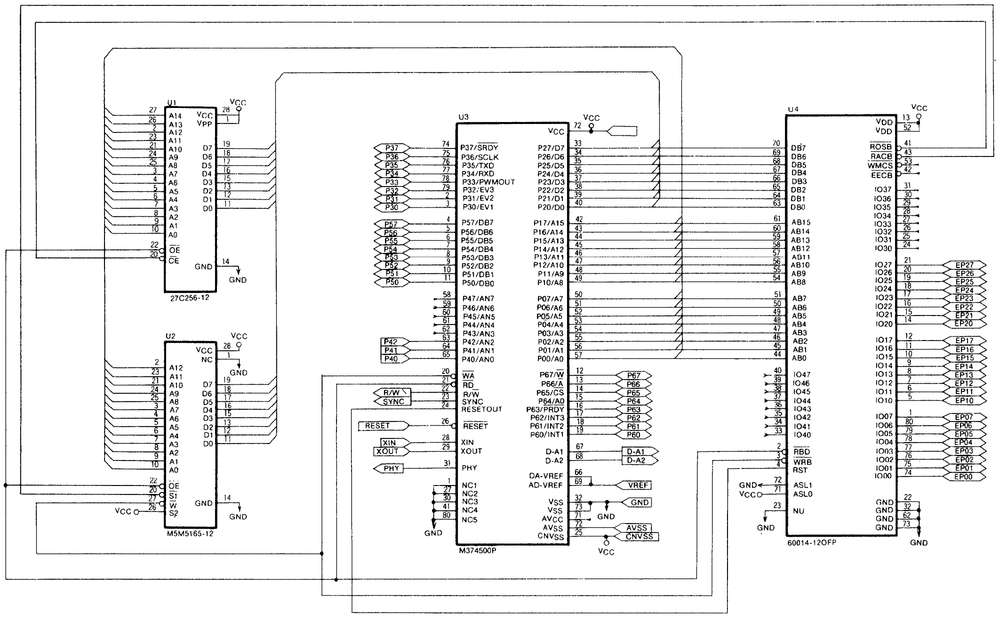


Fig.7.1.5 Example for timer function upgrade

(3) Port expansion using gate arrays

The M60014-0120FP is a gate array in the M37450 family, which is designed for port expansion. In microprocessor mode or memory-expanding mode of the M37450 family, it provides the system with three 8-bit (24 bits) or five 8-bit (40 bits) bi-directional ports. For details, see the M60014-0120FP specifications in Appendix 12.

Figure 7.1.6 is a system wiring diagram for the small-eva-board (PCA7450PGB), which uses the microprocessor mode M37450M2-XXXFP and M60014-0120FP, for evaluating the 7450 system.


U1
27C256-12
A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
OE CE
VCC VPP
D7 D6 D5 D4 D3 D2 D1 D0
GND
U2
M5M5165-12
A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
OE S1 W S2
VCC NC
GND
U3
M374500P
P37/SRDY P36/SCLK P35/TXD P34/RXD P33/PWMOUT P32/EV3 P31/EV2 P30/EV1
P57/DB7 P56/DB6 P55/DB5 P54/DB4 P53/DB3 P52/DB2 P51/DB1 P50/DB0
P47/AN7 P46/AN6 P45/AN5 P44/AN4 P43/AN3 P42/AN2 P41/AN1 P40/AN0
WA RD R/W SYNC RESETOUT RESET XIN XOUT PHY
NC1 NC2 NC3 NC4 NC5
P27/D7 P26/D6 P25/D5 P24/D4 P23/D3 P22/D2 P21/D1 P20/D0
P17/A15 P16/A14 P15/A13 P14/A12 P13/A11 P12/A10 P11/A9 P10/A8
P07/A7 P06/A6 P05/A5 P04/A4 P03/A3 P02/A2 P01/A1 P00/A0
P67/W P66/A P65/CS P64/A0 P63/PRDY P62/INT3 P61/INT2 P60/INT1
D-A1 D-A2 DA-VREF AD-VREF VSS VSS AVCC AVSS CNVSS
P37 P36 P35 P34 P33 P32 P31 P30
P57 P56 P55 P54 P53 P52 P51 P50
P42 P41 P40
R/W SYNC RESET XIN XOUT PHY
P67 P66 P65 P64 P63 P62 P61 P60
D-A1 D-A2 VREF GND AVSS CNVSS
U4
60014-12OFP
DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
AB15 AB14 AB13 AB12 AB11 AB10 AB9 AB8
AB7 AB6 AB5 AB4 AB3 AB2 AB1 AB0
IO47 IO46 IO45 IO44 IO43 IO42 IO41 IO40
RBD WRB RST ASL1 ASL0 NU
VDD VDD ROSB RACB WMCS EECB
IO37 IO36 IO35 IO34 IO33 IO32 IO31 IO30
IO27 IO26 IO25 IO24 IO23 IO22 IO21 IO20
IO17 IO16 IO15 IO14 IO13 IO12 IO11 IO10
IO07 IO06 IO05 IO04 IO03 IO02 IO01 IO00
GND GND GND GND
EP27 EP26 EP25 EP24 EP23 EP22 EP21 EP20
EP17 EP16 EP15 EP14 EP13 EP12 EP11 EP10
EP07 EP06 EP05 EP04 EP03 EP02 EP01 EP00
VCC
GND

Note : The oscillation is 8MHz.

Fig.7.1.6 PCA7450PGB system wiring diagram

7.2 Applicaiton circuit

7.2.1 Electric typewriter application

This application uses three M37450s. As the master processor, the M37450 uses the 7.37MHz clock frequency. Others uses are mechanical controllers and FLT display drivers.

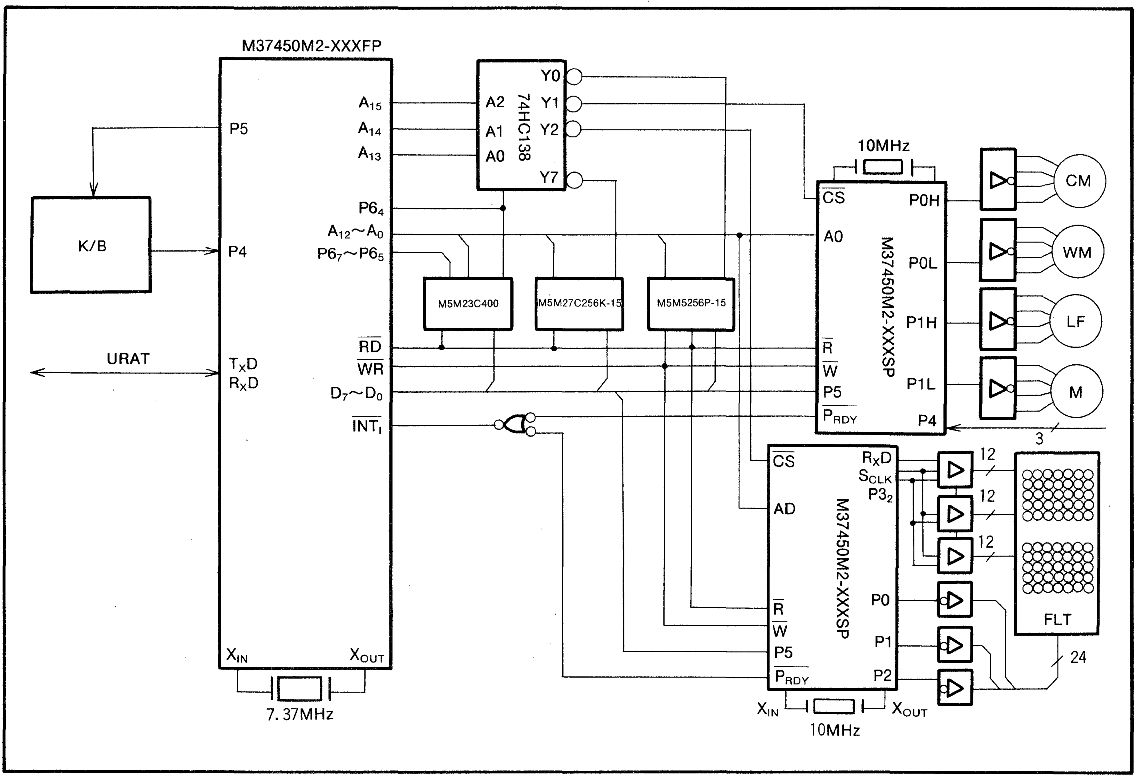


Fig.7.2.1 Application for electric typewriter

7.2.2 Hard disk drive application

The M37450 has a high-speed A-D converter, a 2-channel D-A converter, high-speed PWM and 3 timers. Therefore, the M37450 can be used as hard disk controller which operates VCM servo (voice coil motor) or STM servo (stepping motor).

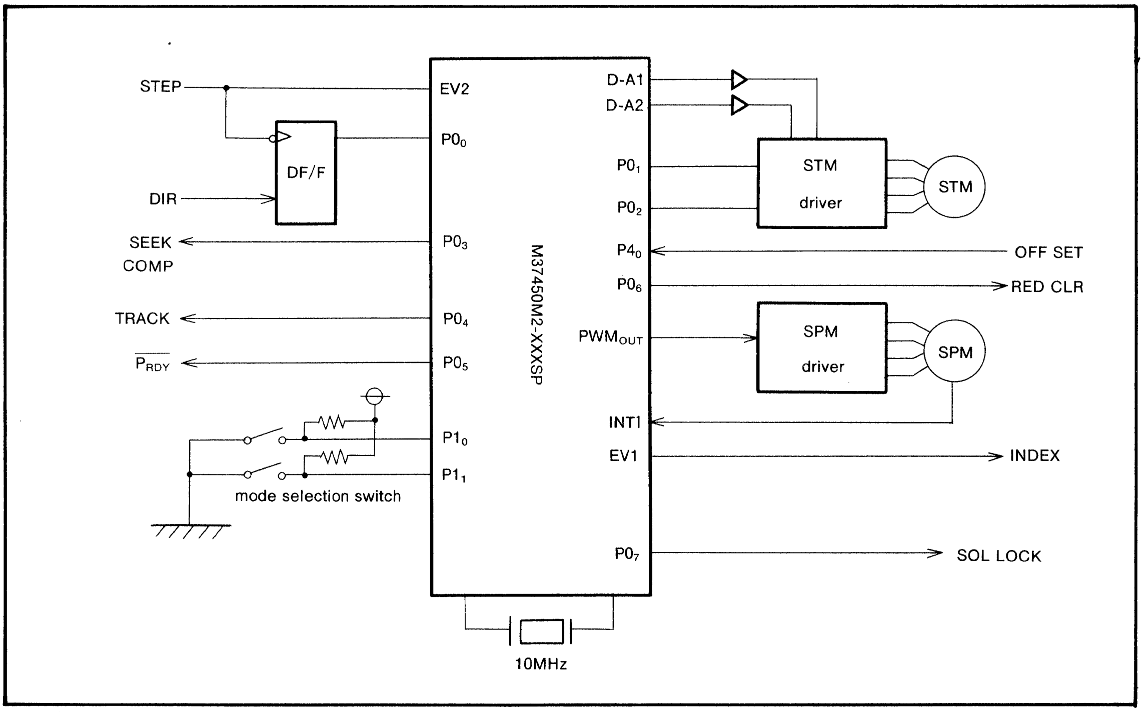


Fig.7.2.2 Application for hard disk drive

7.2.3 Intelligent modem application

The M37450 has advanced 16-bit timers which can be applied to the intelligent modem controller shown in Figure 7.2.3.

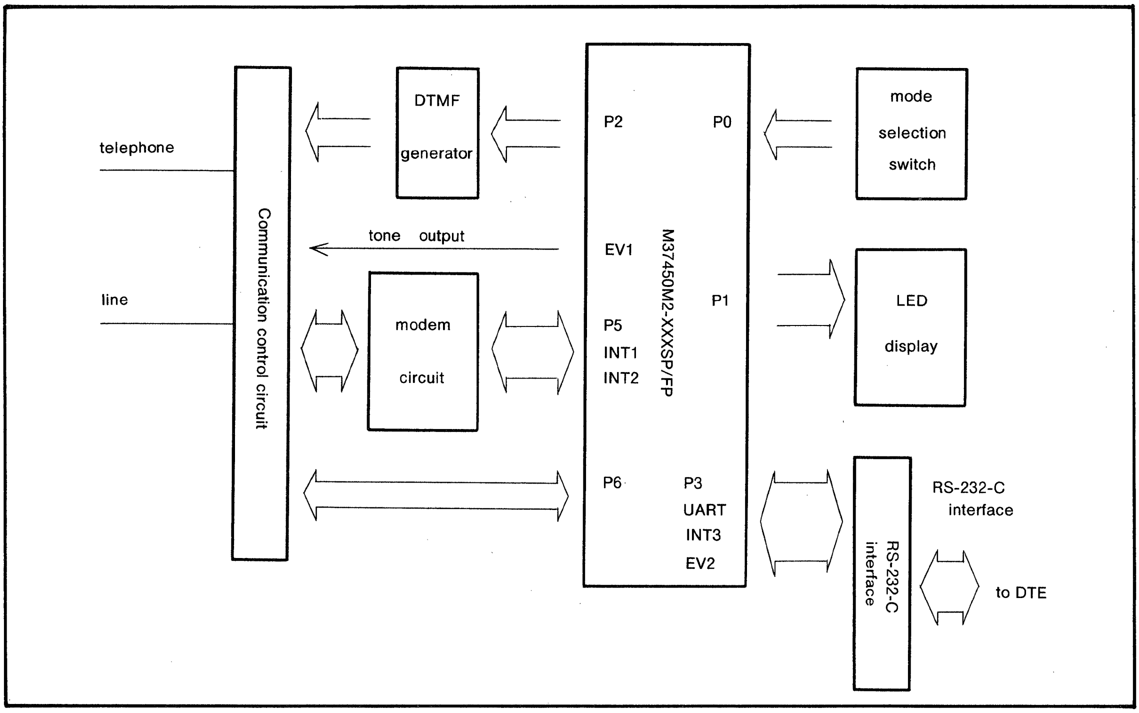


Fig.7.2.3 Application for intelligent modem

7.2.4 Voice digital recording/playback system application

By using the high speed A-D and D-A converters of the M37450, voice digital recording/playback system can be realized.

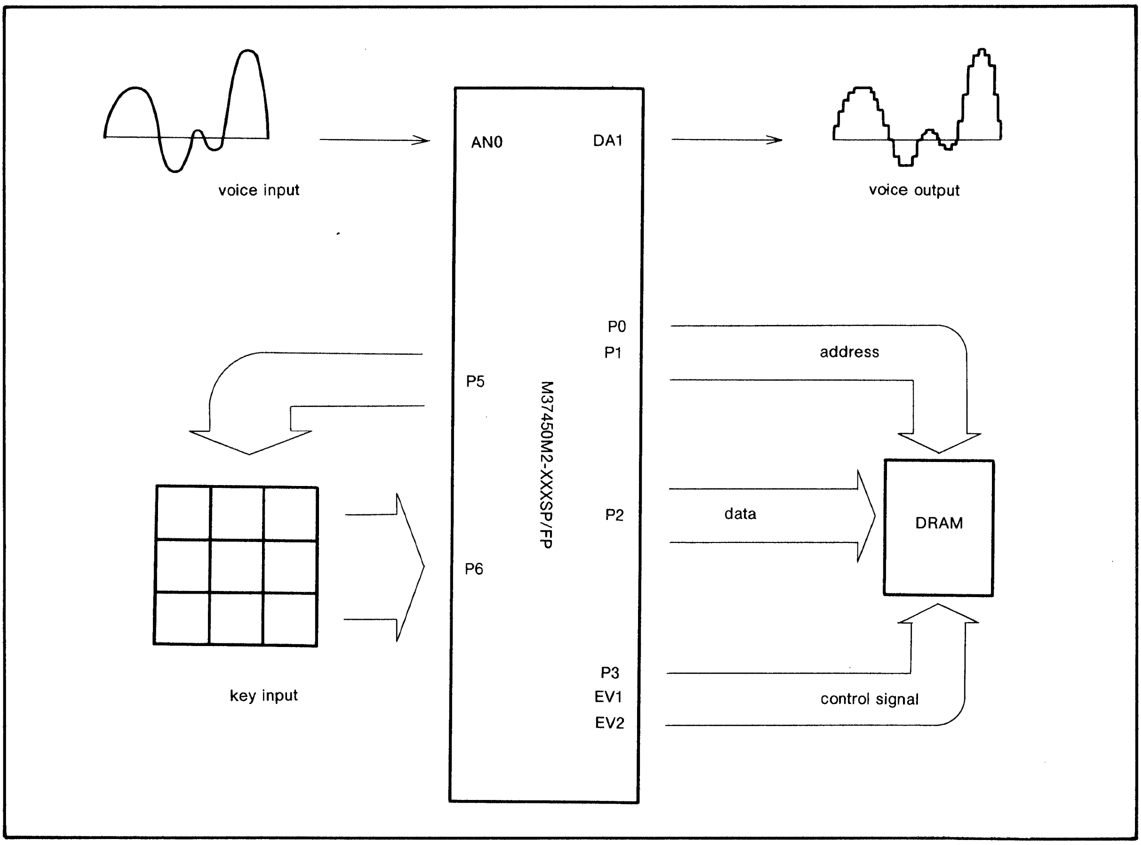


Fig.7.2.4 Application for voice digital recording/playback system

```
* M50745 ASSEMBLER V.2.01C *                                   P. 002

E  SEQ.  LOC.   OBJ..  ' ....*....1....*....2....*....SOURCE STATEMENT....5....*.

      1                 ;***************************************  VER 0.8(5/26)
      2                 ;        ***  S F R EQUATION  ***      **
      3                 ;************************************,***
      4  00D0           PORT0    =       $D0                  ;*
      5  00D1           PORT0D   =       $D1                  ;*
      6  00D2           PORT1    =       $D2                  ;*
      7  00D3           PORT1D   =       $D3                  ;*
      8  00D4           PORT2    =       $D4                  ;*
      9  00D5           PORT2D   =       $D5                  ;*
     10  00D6           PORT3    =       $D6                  ;*
     11  00D7           PORT3D   =       $D7                  ;*
     12  00D8           PORT4    =       $D8                  ;*
     13  00DA           PORT5    =       $DA                  ;*
     14  00DB           PORT5D   =       $DB                  ;*
     15  00DC           PORT6    =       $DC                  ;*
     16  00DD           PORT6D   =       $DD                  ;*
     17                 ;****************************************
     18  00DE           MISRG1   =       $DE                  ;*
     19  0,DE           I1P      =       0,MISRG1             ;*
     20  1,DE           I2P      =       1,MISRG1             ;*
     21  2,DE           I3P      =       2,MISRG1             ;*
     22  4,DE           C1       =       4,MISRG1             ;*
     23  5,DE           C2       =       5,MISRG1             ;*
     24  6,DE           C3       =       6,MISRG1             ;*
     25  00DF           MISRG2   =       $DF                  ;*
     26  0,DF           CM0      =       0,MISRG2             ;*
     27  1,DF           CM1      =       1,MISRG2             ;*
     28  2,DF           DBBE     =       2,MISRG2             ;*
     29  3,DF           DBBM     =       3,MISRG2             ;*
     30  4,DF           PWME     =       4,MISRG2             ;*
     31  5,DF           PWMM     =       5,MISRG2             ;*
     32  6,DF           RDYE     =       6,MISRG2             ;*
     33  7,DF           SPS      =       7,MISRG2             ;*
     34  00E0           D_A1     =       $E0                  ;*
     35  00E1           D_A2     =       $E1                  ;*
     36  00E2           ADSR     =       $E2                  ;*
     37  00E3           ADCON    =       $E3                  ;*
     38  00E4           DBB      =       $E4                  ;*
     39  00E5           DBBSTS   =       $E5                  ;*
     40  0,E5           OBF      =       0,DBBSTS             ;*
     41  1,E5           IBF      =       1,DBBSTS             ;*
     42  3,E5           A0       =       3,DBBSTS             ;*
     43  00E6           RB_TB    =       $E6                  ;*
     44  00E7           SIOSTS   =       $E7                  ;*
     45  0,E7           TBE      =       0,SIOSTS             ;*
     46  1,E7           RBF      =       1,SIOSTS             ;*
     47  2,E7           TSC      =       2,SIOSTS             ;*
     48  3,E7           OE       =       3,SIOSTS             ;*
     49  4,E7           PE       =       4,SIOSTS             ;*
     50  5,E7           FE       =       5,SIOSTS             ;*
     51  6,E7           SE       =       6,SIOSTS             ;*
     52  00E8           SIOCON   =       $E8                  ;*
     53  0,E8           FOSC2_8  =       0,SIOCON             ;*
     54  1,E8           BRG_EXC  =       1,SIOCON             ;*
```

```
* M50745 ASSEMBLER V.2.01C *                                         P. 003

E  SEQ.  LOC.  OBJ..       ....*....1....*....2....*....SOURCE STATEMENT....5....*.

     55  2,E8              SREADYE =        2,SIOCON                 ;*
     56  3,E8              TIC     =        3,SIOCON                 ;*
     57  4,E8              TE      =        4,SIOCON                 ;*
     58  5,E8              RE      =        5,SIOCON                 ;*
     59  6,E8              UAR_CIO =        6,SIOCON                 ;*
     60  7,E8              SIOEN   =        7,SIOCON                 ;*
     61  00E9              UACON   =        $E9                      ;*
     62  0,E9              CH7_8   =        0,UACON                  ;*
     63  1,E9              PARE_D  =        1,UACON                  ;*
     64  2,E9              EVEN_OD =        2,UACON                  ;*
     65  3,E9              SP1_2   =        3,UACON                  ;*
     66  00EA              BRG     =        $EA                      ;*
     67  00EB              PWML    =        $EB                      ;*
     68  00EC              PWMH    =        $EC                      ;*
     69  00ED              TMR1    =        $ED                      ;*
     70  0,ED              TMS10   =        0,TMR1                   ;*
     71  1,ED              TMS11   =        1,TMR1                   ;*
     72  2,ED              TMS12   =        2,TMR1                   ;*
     73  3,ED              S1      =        3,TMR1                   ;*
     74  4,ED              OLL1    =        4,TMR1                   ;*
     75  5,ED              EP1     =        5,TMR1                   ;*
     76  00EE              TMR2    =        $EE                      ;*
     77  0,EE              TMS20   =        0,TMR2                   ;*
     78  1,EE              TMS21   =        1,TMR2                   ;*
     79  2,EE              TMS22   =        2,TMR2                   ;*
     80  3,EE              S2      =        3,TMR2                   ;*
     81  4,EE              OLL2    =        4,TMR2                   ;*
     82  5,EE              EP2     =        5,TMR2                   ;*
     83  00EF              TMR3    =        $EF                      ;*
     84  0,EF              TMS30   =        0,TMR3                   ;*
     85  1,EF              TMS31   =        1,TMR3                   ;*
     86  2,EF              TMS32   =        2,TMR3                   ;*
     87  3,EF              S3      =        3,TMR3                   ;*
     88  4,EF              OLL3    =        4,TMR3                   ;*
     89  5,EF              EP3     =        5,TMR3                   ;*
     90                    ;*****************************************
     91  00F0              T1L     =        $F0                      ;*
     92  00F1              T1H     =        $F1                      ;*
     93  00F2              RL1L    =        $F2                      ;*
     94  00F3              RL1H    =        $F3                      ;*
     95  00F4              T2L     =        $F4                      ;*
     96  00F5              T2H     =        $F5                      ;*
     97  00F6              RL2L    =        $F6                      ;*
     98  00F7              RL2H    =        $F7                      ;*
     99  00F8              T3L     =        $F8                      ;*
    100  00F9              T3H     =        $F9                      ;*
    101  00FA              RL3L    =        $FA                      ;*
    102  00FB              RL3H    =        $FB                      ;*
    103  00FC              IRQ1    =        $FC                      ;*
    104  0,FC              IBFR    =        0,IRQ1                   ;*
    105  1,FC              OBER    =        1,IRQ1                   ;*
    106  2,FC              I1R     =        2,IRQ1                   ;*
    107  3,FC              I2R     =        3,IRQ1                   ;*
    108  4,FC              I3R     =        4,IRQ1                   ;*
```

```
* M50745 ASSEMBLER V.2.01C *                                       P. 004

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    109  5,FC            T1R      =       5,IRQ1                     ;*
    110  6,FC            T2R      =       6,IRQ1                     ;*
    111  7,FC            T3R      =       7,IRQ1                     ;*
    112  00FD            IRQ2     =       $FD                        ;*
    113  0,FD            E1R      =       0,IRQ2                     ;*
    114  1,FD            E2R      =       1,IRQ2                     ;*
    115  2,FD            E3R      =       2,IRQ2                     ;*
    116  3,FD            RIR      =       3,IRQ2                     ;*
    117  4,FD            TIR      =       4,IRQ2                     ;*
    118  5,FD            ADCOMPR  =       5,IRQ2                     ;*
    119  00FE            ICON1    =       $FE                        ;*
    120  0,FE            IBFE     =       0,ICON1                    ;*
    121  1,FE            OBEE     =       1,ICON1                    ;*
    122  2,FE            I1E      =       2,ICON1                    ;*
    123  3,FE            I2E      =       3,ICON1                    ;*
    124  4,FE            I3E      =       4,ICON1                    ;*
    125  5,FE            T1E      =       5,ICON1                    ;*
    126  6,FE            T2E      =       6,ICON1                    ;*
    127  7,FE            T3E      =       7,ICON1                    ;*
    128  00FF            ICON2    =       $FF                        ;*
    129  0,FF            E1E      =       0,ICON2                    ;*
    130  1,FF            E2E      =       1,ICON2                    ;*
    131  2,FF            E3E      =       2,ICON2                    ;*
    132  3,FF            RIE      =       3,ICON2                    ;*
    133  4,FF            TIE      =       4,ICON2                    ;*
    134  5,FF            ADCOMPE  =       5,ICON2                    ;*
    135                  ;****************************************
    136  0000            W1       =       $00                        ;*
    137                  ;****************************************
    138                           .NLIST                             ;*
    158                           .LIST
```

```
E SEQ.  LOC.  OBJ..    ....•....1....•....2....•....SOURCE STATEMENT....5....•.

                                 .PAGE
                       ;•••••••••••••••••••••••••••••••••••••
                       ;   •••    TIMER              •••     •
                       ;•••••••••••••••••••••••••••••••••••••
                       ;* * *   MODE 0 (TMS=000B : Timer mode)
                       ;
                       ; This is the normal timer mode. When count value N is set in the timer latch, the actual counting number becomes (N+1), due to
                       ; the reload action needing 1 count-cycle. However, the timer does not need to be reloaded action until the first overflow. Therefore
                       ; the initial timer value must be (N+1) so the same counting time will be used when the value N was loaded.
                       ; The timer must be loaded from the low-byte address first.
                       ;
                       TMS000:                                         ;
      F018 1FED                  CLB      TMS10                        ;SET MODE
      F01A 3FED                  CLB      TMS11                        ;
      F01C 5FED                  CLB      TMS12                        ;
                       ;                                               ;  ••• TIMER REGISTER SETTING •••
      F01E 3CAAF2                LDM      #$AA,RL1L                    ;       RL1H | RL1L
      F021 3CBBF3                LDM      #$BB,RL1H                    ;     ( BBH    AAH  )
      F024 3CABF0                LDM      #$AB,T1L                     ;       T1H  | T1L
      F027 3CBBF1                LDM      #$BB,T1H                     ;     ( BBH    ABH  )
                       ;                                               ;
      F02A BFFC                  CLB      T1R                          ;CLEAR REQUEST FLAG
      F02C 8FDE                  SEB      C1                           ;TIMER START
      F02E B7FCFD                BBC      T1R,•                        ;WAIT FOR TIMER_OVERFLOW
                       ;                                               ; (400ns x BBABH WHEN fosc = 10MHz)
      F031 BFFC                  CLB      T1R                          ;CLEAR REQUEST FLAG
      F033 EA                    NOP                                   ;
      F034 B7FCFD                BBC      T1R,•                        ;WAIT FOR TIMER_OVERFLOW
      F037 9FDE                  CLB      C1                           ; (400ns x (BBAA+1)H WHEN fosc = 10 MHz)
      F039 60                    RTS                                   ;
                       TMS0002:                                        ;TMS000 OF TYPE_2
      F03A 1FED                  CLB      TMS10                        ;SET MODE
      F03C 3FED                  CLB      TMS11                        ;
      F03E 5FED                  CLB      TMS12                        ;
                       ;                                               ;  ••• TIMER REGISTER SETTING •••
      F040 3CAAF2                LDM      #$AA,RL1L                    ;SET    RL1H | RL1L
      F043 3CBBF3                LDM      #$BB,RL1H                    ;     ( BBH    AAH  )
      F046 3CABF0                LDM      #$AB,T1L                     ;SET    T1H  | T1L
      F049 3CBBF1                LDM      #$BB,T1H                     ;     ( BBH    ABH  )
                       ;                                               ;
      F04C BFFC                  CLB      T1R                          ;CLEAR REQUEST FLAG
      F04E 8FDE                  SEB      C1                           ;TIMER START
      F050 B7FCFD                BBC      T1R,•                        ;WAIT FOR TIMER OVER_FLOW
                       ;                                               ; (400ns x BBABH WHEN fosc = 10MHz)
      F053 BFFC                  CLB      T1R                          ;CLEAR REQUEST FLAG
      F055 3C22F2                LDM      #$22,RL1L                    ;SET    RL1H | RL1L
      F058 3C33F3                LDM      #$33,RL1H                    ;     ( 33H    22H  )
      F05B B7FCFD                BBC      T1R,•                        ;WAIT FOR TIMER_OVERFLOW
      F05E BFFC                  CLB      T1R                          ; (400ns x (BBAA+1)H WHEN fosc = 10 MHz)
      F060 EA                    NOP                                   ;
      F061 B7FCFD                BBC      T1R,•                        ;WAIT FOR TIMER_OVERFLOW
                       ;                                               ; (400ns x (3322+1)H WHEN fosc = 10 MHz)
      F064 9FDE                  CLB      C1                           ;
      F066 60                    RTS                                   ;
```

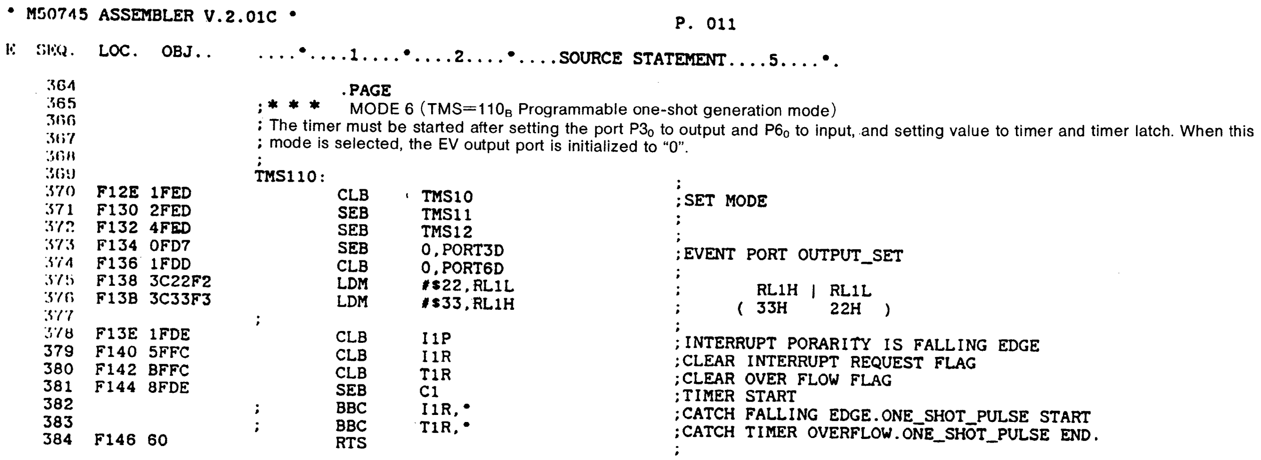
```
• M50745 ASSEMBLER V.2.01C •                              P. 006

E SEQ.  LOC.  OBJ..    ....•....1....•....2....•....SOURCE STATEMENT....5....•.

                                .PAGE
               ;* * *    MODE 1 (TMS=$001_B$ : Event counter mode)
               ; This mode has the same function as the timer mode except for using the EV port as count source. It is necessary to set port $P3_0$ to
               ; input mode and set the polarity of the EV port. The EV input signal synchronizes with the internal clock, and the timer contents can
               ; be read anytime. In this case, the timers high byte (T1H) must be read first, and T1L second.
               ;
               ;
               TMS001:
F067 0FED              SEB     TMS10                   ;SET MODE
F069 3FED              CLB     TMS11                   ;
F06B 5FED              CLB     TMS12                   ;
F06D 1FD7              CLB     0,PORT3D                ;
F06F BFED              CLB     EP1                     ;PORARITY IS FALLING EDGE
               ;                                       ;  *** TIMER REGISTER SETTING ***
F071 3CFFF2            LDM     #$FF,RL1L               ;       RL1H | RL1L
F074 3CFFF3            LDM     #$FF,RL1H               ;     ( FFH    FFH  )
F077 3CFFF0            LDM     #$FF,T1L                ;       T1H  | T1L
F07A 3CFFF1            LDM     #$FF,T1H                ;     ( FFH    FFH  )
               ;                                       ;
F07D BFFC              CLB     T1R                     ;CLEAR REQUEST FLAG
F07F 8FDE              SEB     C1                      ;TIMER START
               ;        :                              ;
               ;        :                              ;
F081 A5F1              LDA     T1H                     ;    |         |  :Read timer_High
F083 49FF              EOR     #$FF                    ;    |---------|  :Get 1's complement of timer_High
F085 8501              STA     W1+1                    ;W1  | timer_L |
F087 A5F0              LDA     T1L                     ;    |---------|  :Read timer_Low
F089 49FF              EOR     #$FF                    ;W1+1| timer_H |  :Get 1's complement of timer_Low
F08B 8500              STA     W1                      ;    |---------|
               ;                                       ;    |         |
F08D 18                CLC                             ;
F08E B7FC01            BBC     T1R,TMS00101            ;CHECK OVER FLOW
F091 38                SEC                             ;
               TMS00101:                               ;
F092 60                RTS                             ;
```

```
• M50745 ASSEMBLER V.2.01C •                                            P. 007

E  SEQ.  LOC.  OBJ..      ....•....1....•....2....•....SOURCE STATEMENT....5....•.

   248                            .PAGE
   249                    ; * * *  MODE 2 (TMS=010B : Pulse output mode)-
   250                    ; In this mode,.the timer must be started after setting port P30 to output mode, setting value to the timer latch, and selecting the
   251                    ; count source. When this mode is selected, the EV port is initialized to "0" automatically.
   252                    ;
   253                    TMS010:                                    ;
   254   F093 1FED                CLB     TMS10                      ;SET MODE
   255   F095 2FED                SEB     TMS11                      ;
   256   F097 5FED                CLB     TMS12                      ;
   257   F099 0FD7                SEB     0,PORT3D                   ;EVENT PORT OUTPUT_SET
   258   F09B 7FED                CLB     S1                         ;COUNT SOURCE - fosc/4
   259                    ;                                          ;  *** TIMER REGISTER SETTING ***
   260   F09D 3CAAF2              LDM     #$AA,RL1L                  ;       RL1H | RL1L
   261   F0A0 3CBBF3              LDM     #$BB,RL1H                  ;     ( BBH    AAH   )
   262   F0A3 3CABF0              LDM     #$AB,T1L                   ;       T1H  | T1L
   263   F0A6 3CBBF1              LDM     #$BB,T1H                   ;     ( BBH    ABH   )
   264                    ;                                          ;
   265   F0A9 BFFC                CLB     T1R                        ;CLEAR REQUEST FLAG
   266   F0AB 8FDE                SEB     C1                         ;TIMER START
   267                    ;       BBC     T1R,*                      ;OUTPUT Low  Pulse (BBABH * 400ns)
   268   F0AD BFFC                CLB     T1R                        ;CLEAR REQUEST FLAG
   269                    ;       NOP                                ;
   270                    ;       BBC     T1R,*                      ;OUTPUT High Pulse ((BBAA+1)H * 400ns)
   271   F0AF 60                  RTS                                ;REPEAT
```

```
• M50745 ASSEMBLER V.2.01C •                                P. 008

E  SEQ.  LOC.  OBJ..      ....•....1....•....2....•....SOURCE STATEMENT....5....•.

   272                               PAGE
   273                    ;* * *     MODE 3 (TMS=011_B Pulse period measurement mode)
   274                    ;
   275                    ; The timer must be started after setting port P3_0 to input mode, setting the polarity of input pulse and selecting the count source.
   276                    ; The end of the measurement is recognized by E1R to become "1". The period of the pulse can be determined by reading the con-
   277                    ; tents of the timer latch. (The timer latch can be read in any order.)
   278                    ;
   279                    TMS011:                                  ;
   280   F0B0 0FED                SEB     TMS10                    ;SET MODE
   281   F0B2 2FED                SEB     TMS11                    ;
   282   F0B4 5FED                CLB     TMS12                    ;
   283   F0B6 1FD7                CLB     0,PORT3D                 ;EVENT PORT INPUT_SET
   284   F0B8 7FED                CLB     S1                       ;COUNT SOURCE - fosc/4
   285   F0BA AFED                SEB     EP1                      ;PORARITY IS RISING EDGE
   286                    ;                                        ;
   287   F0BC 1FFD                CLB     E1R                      ;CLEAR EVENT_INTERRUPT_REQUEST_FLAG
   288   F0BE 8FDE                SEB     C1                       ;TIMER START
   289   F0C0 17FDFD              BBC     E1R,*                    ;Pulse Phase COUNT END AND DUST FIRST TIME
   290   F0C3 1FFD                CLB     E1R                      ;CLEAR EVENT_INTERRUPT_REQUEST_FLAG
   291   F0C5 EA                  NOP                              ;
   292   F0C6 17FDFD              BBC     E1R,*                    ;
   293   F0C9 A5F3                LDA     RL1H                     ;READ TIMER_LATCH High
   294   F0CB A6F2                LDX     RL1L                     ;READ TIMER_LATCH Low
   295   F0CD 9FDE                CLB     C1                       ;TIMER STOP
   296   F0CF 60                  RTS                              ;
```

```
• M50745 ASSEMBLER V.2.01C •                              P. 009

E SEQ.  LOC.  OBJ..    ....•....1....•....2....•....SOURCE STATEMENT....5....•.

   297                           .PAGE
   298                 ;* * *     MODE 4 (TMS=100B Pulse width measurement mode)
   299                 ;
   300                 ; The timer must be started after setting the port P30 to input, setting the polarity of input pulse and selecting the count source. The
   301                 ; end of measurement is recognized by E1R to become "1". The width of the pulse can be determined by reading the contents of
   302                 ; the timer latch. (The timer latch can be read in any order.)
   303                 ;
   304                 TMS100:                                    ;
   305  F0D0 1FED               CLB     TMS10                      ;SET MODE
   306  F0D2 3FED               CLB     TMS11                      ;
   307  F0D4 4FED               SEB     TMS12                      ;
   308  F0D6 1FD7               CLB     0,PORT3D                   ;SET EVENT_PORT INPUT
   309  F0D8 7FED               CLB     S1                         ;COUNT SOURCE - fosc/4
   310  F0DA BFED               CLB     EP1                        ;PORARITY IS FALLING EDGE
   311                 ;                                          ;MEASURES PULSE High
   312  F0DC 1FFD               CLB     E1R                        ;CLEAR EVENT_INTERRUPT_REQUEST_FLAG
   313  F0DE 8FDE               SEB     C1                         ;TIMER START
   314  F0E0 17FDFD             BBC     E1R,•                      ;COUNT Pulse Width END AND DUST FIRST TIME
   315  F0E3 1FFD               CLB     E1R                        ;CLEAR EVENT_INTERRUPT_REQUEST_FLAG
   316  F0E5 EA                 NOP                                ;
   317  F0E6 17FDFD             BBC     E1R,•                      ;
   318  F0E9 A5F3               LDA     RL1H                       ;READ TIMER_LATCH High
   319  F0EB A6F2               LDX     RL1L                       ;READ TIMER_LATCH Low
   320  F0ED 9FDE               CLB     C1                         ;TIMER STOP
   321  F0EF 60                 RTS                                ;
```

```
• M50745 ASSEMBLER V.2.01C •                                  P. 010

E  SEQ. LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    322                          .PAGE
    323                  ;* * *   MODE 5 (TMS=101B Programmable waveform generation mode)
    324                  ;The timer must be started after setting the port P30 to input, setting the polarity of input pulse and selecting the count source.
    325                  ;When this mode is selected, the EV output port is initialized to "1".
    326                  ;
    327                  TMS101:                                ;
    328 F0F0 0FED                SEB     TMS10                  ;SET MODE
    329 F0F2 3FED                CLB     TMS11                  ;
    330 F0F4 4FED                SEB     TMS12                  ;
    331 F0F6 0FD7                SEB     0,PORT3D               ;SET EVENT PORT OUTPUT
    332 F0F8 7FED                CLB     S1                     ;COUNT SOURCE - fosc/4
    333 F0FA 3C22F2              LDM     #$22,RL1L              ;       RL1H | RL1L
    334 F0FD 3C33F3              LDM     #$33,RL1H              ;     ( 33H    22H  )
    335 F100 3CAAF0              LDM     #$AA,T1L               ;       T1H  | T1L
    336 F103 3CBBF1              LDM     #$BB,T1H               ;     ( BBH    AAH  )
    337                  ;                                      ;                            <-BBAAH->
    338 F106 9FED                CLB     OLL1                   ;CLEAR OUTPUT_LEVEL_LATCH    ---------   --
    339 F108 BFFC                CLB     T1R                    ;CLEAR OVERFLOW FLAG         |       |   |
    340 F10A 8FDE                SEB     C1                     ;TIMER START                         -----
    341 F10C B7FCFD              BBC     T1R,*                  ;EVENT OUTPUT HIGH PULSE             <--->
    342                  ;                                      ;                                  (3322+1)H
    343 F10F BFFC                CLB     T1R                    ;CLEAR OVER FLOW FLAG
    344 F111 8FED                SEB     OLL1                   ;
    345 F113 B7FCFD              BBC     T1R,*                  ;EVENT OUTPUT LOW PULSE
    346                  ;                                      ;
    347 F116 9FDE                CLB     C1                     ;TIMER STOP
    348 F118 60                  RTS                            ;
    349                  ;;;;;;;;                               ;
    350                  TMS1012:                               ;TMS101 OF TYPE_2(LIKE ONE-SHOT-PULSE)
    351 F119 0FED                SEB     TMS10                  ;SET MODE
    352 F11B 3FED                CLB     TMS11                  ;
    353 F11D 4FED                SEB     TMS12                  ;
    354 F11F 0FD7                SEB     0,PORT3D               ;SET EVENT PORT OUTPUT       <-BBAAH->
    355 F121 3CAAF0              LDM     #$AA,T1L               ;       T1H  | T1L           ---------
    356 F124 3CBBF1              LDM     #$BB,T1H               ;     ( BBH    AAH  )        |       |
    357                  ;                                      ;COUNT SOURCE CHANGE                 --------
    358 F127 6FED                SEB     S1                     ;COUNT SOURCE - INT_PORT CLOCK
    359                  ;                                      ;
    360 F129 9FED                CLB     OLL1                   ;
    361 F12B 8FDE                SEB     C1                     ;TIMER START
    362                  ;                                      ;
    363 F12D 60                  RTS                            ;
```



```
* M50745 ASSEMBLER V.2.01C *                                      P. 011
E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.
  364                            .PAGE
  365                     ;* * *   MODE 6 (TMS=110B Programmable one-shot generation mode)
  366                     ; The timer must be started after setting the port P30 to output and P60 to input, and setting value to timer and timer latch. When this
  367                     ; mode is selected, the EV output port is initialized to "0".
  368                     ;
  369                     TMS110:                                          ;
  370    F12E 1FED                 CLB     TMS10                         ;SET MODE
  371    F130 2FED                 SEB     TMS11                         ;
  372    F132 4FED                 SEB     TMS12                         ;
  373    F134 0FD7                 SEB     0,PORT3D                      ;EVENT PORT OUTPUT_SET
  374    F136 1FDD                 CLB     0,PORT6D                      ;
  375    F138 3C22F2               LDM     #$22,RL1L                     ;       RL1H | RL1L
  376    F13B 3C33F3               LDM     #$33,RL1H                     ;     ( 33H    22H  )
  377                     ;                                                ;
  378    F13E 1FDE                 CLB     I1P                           ;INTERRUPT PORARITY IS FALLING EDGE
  379    F140 5FFC                 CLB     I1R                           ;CLEAR INTERRUPT REQUEST FLAG
  380    F142 BFFC                 CLB     T1R                           ;CLEAR OVER FLOW FLAG
  381    F144 8FDE                 SEB     C1                            ;TIMER START
  382                     ;        BBC     I1R,*                         ;CATCH FALLING EDGE.ONE_SHOT_PULSE START
  383                     ;        BBC     T1R,*                         ;CATCH TIMER OVERFLOW.ONE_SHOT_PULSE END.
  384    F146 60                   RTS                                   ;
```

```
* M50745 ASSEMBLER V.2.01C *                                    P. 012

E  SEQ.  LOC.  OBJ..    ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    385                         .PAGE
    386                 ;******************************************************************
    387                 ;   ***    INITIAL SET FOR PULSE WIDTH MODULATER FUNCTION ***     *
    388                 ;******************************************************************
    389                 ; In order to start the PWM function, set port P3₃ to output mode, select the mode (resolution), set the PWM register, and set the
    390                 ; PWME bit to "1". By setting the PWME to "0" or writing "0" to the PWM register, the PWM output will stop.
    391                 ;
    392                 PWM000:                                 ;SET FOR PWM
    393  F147 6FD7              SEB     3,PORT3D                ;SET PPORT3_3 OUTPUT
    394  F149 3CAAEB            LDM     #$AA,PWML               ;
    395  F14C 3C55EC            LDM     #$55,PWMH               ;
    396  F14F 60                RTS                             ;
    397                 ;*****************************************
    398                 ;   ***    PLUSE WIDTH MODULATER ***      *
    399                 ;*****************************************
    400                 PWM8                                    ;
    401                 ;                                       ;
    402  F150 BFDF              CLB     PWMM                    ;PWM MODE IS 8_BIT_MODE
    403  F152 8FDF              SEB     PWME                    ;PWM START |---  (CYCLE IS 51.2us)  ---|
    404                 ;                                       ;OUTPUT    H-----------HL-------------L
    405                 ;                                       ;  PULSE   |-  34  us  -|-  17.2  us  -|
    406                 ;       CLB     PWME                    ;PWM STOP
    407  F154 60                RTS                             ;
    408                 ;                                       ;
    409                 PWM16:                                  ;
    410  F155 AFDF              SEB     PWMM                    ;PWM MODE IS 16_BIT_MODE
    411  F157 8FDF              SEB     PWME                    ;PWM START |---  (CYCLE IS 13.1ms)  ---|
    412                 ;        :                              ;OUTPUT    H-----------HL-------------L
    413                 ;        :                              ;  PULSE   |- 4.386 ms -|-  8.714 ms  -|
    414  F159 3C11EC            LDM     #$11,PWMH               ;CHANGE PWM's VALUE
    415  F15C 3C22EB            LDM     #$22,PWML               ;PWM PULSE CHANGE FROM NEXT CYCLE
    416                 ;       CLB     PWME                    ;PWM STOP
    417  F15F 60                RTS                             ;
```

```
* M50745 ASSEMBLER V.2.01C *                                  P. 013

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   418                            .PAGE
   419                    ;************************************
   420                    ;    ***    A-D CONVERTER         ***     *
   421                    ;************************************
   422                    ; After writing the port number to ADCON, the A-D conversion will start through this port. The conversion completes in 19.8μsec
   423                    ; when using 10MHz clock frequency. The ADCOMPR flag becoming "1" signals the A-D conversion completion.
   424                    ;
   425                    AD000:
   426   F160  A207               LDX      #$07                 ;
   427                    AD010:                                  ;
   428   F162  BFFD               CLB      ADCOMPR              ;CLEAR REQUEST FLAG
   429   F164  8A                 TXA                           ;
   430   F165  85E3               STA      ADCON                ;SELECT A-D PIN AND START A-D CONVERT
   431   F167  B7FDFD             BBC      ADCOMPR,*            ;CHECK CONVERT_END
   432   F16A  A5E2               LDA      ADSR                 ;READ AD VALUE
   433   F16C  9500               STA      W1,X                 ;
   434   F16E  CA                 DEX                           ;
   435   F16F  10F1               BPL      AD010                ;
   436   F171  60                 RTS                           ;
```

```
• M50745 ASSEMBLER V.2.01C •                                P. 014

E  SEQ.  LOC.  OBJ..     ....•....1....•....2....•....SOURCE STATEMENT....5....•.

    437                            .PAGE
    438                  ;•••••••••••••••••••••••••••••••••••••••
    439                  ;   •••     D-A CONVERTER          •••     •
    440                  ;•••••••••••••••••••••••••••••••••••••••
    441                  ;•••••••••••••••••••••••••••••••••••••••
    442                  ; By setting the D-A1, 2 register, the D-A1, 2 port outputs the corresponding voltage. The output voltage can be calculated as
    443                  ; (DAVref)·(D-A/256) [V]
    444                  ;
    445                  DA000:                                    ;
    446   F172 A900                LDA      #$00                  ;
    447                  DA010:                                    ;
    448   F174 85E0                STA      D_A1                  ;(D_A1/256) • Vref [V] OUTPUT
    449   F176 85E1                STA      D_A2                  ;(D_A2/256) • Vref [V] OUTPUT
    450   F178 3A                  INC      A                     ;
    451   F179 D0F9                BNE      DA010                 ;0 - 255
    452   F17B 60                  RTS                            ;
    453                            .NLIST
    470                            .LIST
```

```
* M50745 ASSEMBLER V.2.01C *                                    P. 015

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    471                             .PAGE
    472                  ;*************************************************
    473                             *=     $FFE0                        ;*
    474  FFE0 7CF1                  .WORD  BRK                          ;* BRK
    475  FFE2 7CF1                  .WORD  ADCOMP                       ;* ADCOMP
    476  FFE4 7CF1                  .WORD  TI                           ;* TI
    477  FFE6 7CF1                  .WORD  RI                           ;* RI
    478  FFE8 7CF1                  .WORD  EV3                          ;* EV3
    479  FFEA 7CF1                  .WORD  EV2                          ;* EV2
    480  FFEC 7CF1                  .WORD  EV1                          ;* EV1
    481  FFEE 7CF1                  .WORD  T3                           ;* T3
    482  FFF0 7CF1                  .WORD  T2                           ;* T2
    483  FFF2 7CF1                  .WORD  T1                           ;* T1
    484  FFF4 7CF1                  .WORD  INT3                         ;* INT3
    485  FFF6 7CF1                  .WORD  INT2                         ;* INT2
    486  FFF8 7CF1                  .WORD  INT1                         ;* INT1
    487  FFFA 7CF1                  .WORD  OBEI                         ;* OBE
    488  FFFC 7CF1                  .WORD  IBFI                         ;* IBF
    489  FFFE 00F0                  .WORD  RESET                        ;* RESET
    490                  ;*************************************************
    491                  ;                 ***  END  ***                **
    492                  ;*************************************************
    493                             .END

ERROR COUNT  0000
TOTAL LINE   0493 LINES
COMMENT LINE 0147 LINES
OBJECT SIZE  0413 BYTES
```

```
• M50745 ASSEMBLER V.2.01C •                                   P. 002

E  SEQ.  LOC.  OBJ..     ....•....1....•....2....•....SOURCE STATEMENT....5....•.


     138                 ;••••••••••••••••••••••••••••••••••••••••••
     139                 ;     •••  CLOCKED SERIAL I/O  •••       ••
     140                 ;••••••••••••••••••••••••••••••••••••••••••
     141                 ; This program sets the work area for the clocked serial I/O of the sample program.
     142  0000           FLAGS    -      $00                      ;•
     143  0,00           ERRF0    -      0,FLAGS                  ;•
     144  1,00           ERRF1    -      1,FLAGS                  ;•
     145  2,00           ERRF2    -      2,FLAGS                  ;•
     146  0001           BRGFLG3 -       $01                      ;•FLAGS SELECT BIT RATE(bps)
     147  0,01           FB1250   -      0,BRGFLG3                ;•1.25MHz
     148  1,01           FB250    -      1,BRGFLG3                ;•250KHz   when fosc - 10.0MHz
     149  2,01           FB50     -      2,BRGFLG3                ;•50KHz
     150  3,01           FB10     -      3,BRGFLG3                ;•10KHz
     151  4,01           FB25     -      4,BRGFLG3                ;•12.5KHz
     152  5,01           FB12H    -      5,BRGFLG3                ;•6.25KHz
     153  6,01           FB5      -      6,BRGFLG3                ;•2.5KHz
     154  7,01           FB2H     -      7,BRGFLG3                ;•1.25KHz
     155  0003           WORKI    -      $03                      ;• RECEIVER    REGISTER
     156  0009           WORKO    -      $09                      ;• TRANSMITTER REGISTER
```

APPLICATION

```
* M50745 ASSEMBLER V.2.01C *                                      P. 003

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   191                            .PAGE
   192                    ;**************************************
   193                    ;       ***  SERIAL I/O  ***          **
   194                    ;**************************************
   195                    ; This program shows the CPU receives 1 byte data via the clocked serial I/O. The received data is stored in memory WORK I.
   196                    ;
   197                    RCV08:                                  ;*                                    ------|
   198  F041  3C80D7              LDM     #$80,PORT3D             ;* %10000000 = PORT3D          Srdy |------>
   199                    ;                                       ;* SET SIOCON %11100110        Sclk |<------
   200  F044  EFE8                SEB     SIOEN                   ;* ENABLE SERIAL I/O           Si   |<------
   201  F046  CFE8                SEB     UAR_CIO                 ;* ENABLE CLOLKED SERIAL      (hardware)
   202  F048  AFE8                SEB     RE                      ;* ENABLE RECEIVER
   203  F04A  4FE8                SEB     SREADYE                 ;* ENABLE SREADY OUTPUT
   204  F04C  2FE8                SEB     BRG_EXC                 ;* SELECT EXC_CLOCK
   205  F04E  3C00E6              LDM     #$00,RB_TB              ;* WRITE DUMMY DATA AND Srdy IS CLEARED LOW
   206  F051  77FDFD              BBC     RIR,*                   ;* RIR = 1 ? -- (RBE = 1 ?)
   207  F054  A5E6                LDA     RB_TB                   ;* READ RB_TB
   208  F056  8503                STA     WORKI                   ;*
   209  F058  7FFD                CLB     RIR                     ;*
   210  F05A  FFE8                CLB     SIOEN                   ;*
   211  F05C  60                  RTS                             ;*
   212                    ;                                       ;*
```

```
* M50745 ASSEMBLER V.2.01C *                              P. 004

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   213                             .PAGE
   214                    ;*************************************
   215                    ;          ***  SERIAL I/O  ***      **
   216                    ;*************************************
   217                    ; This program shows the CPU transmits 1 byte data via the clocked serial I/O. The transmitted data must be set in memory
   218                    ; WORK0   before the transmission.
   219                    TRN08:                                 ;*                          -----,
   220   F05D 3C60D7              LDM      #$60,PORT3D           ;* %01100000 - PORT3D      Sclk |------>
   221                    ;                                      ;*                          So   |------>
   222   F060 1FE8                CLB      FOSC2_8               ;* fosc / (2*4)             INT  |<------ Srdy
   223                    ;                                      ;* when fosc - 10 MHz
   224   F062 3C00EA              LDM      #$00,BRG              ;* SET 1.25MHz
   225                    ;                                      ;* SET SIOCON %11010000
   226   F065 1FDD                CLB      0,PORT6D              ;* P6_0 IS INPUT PORT
   227   F067 EFE8                SEB      SIOEN                 ;* ENABLE SERIAL I/O
   228   F069 CFE8                SEB      UAR_CIO               ;* ENABLE CLOLKED SERIAL
   229   F06B 3FE8                CLB      BRG_EXC               ;* SELECT BRG_CLOCK
   230   F06D 8FE8                SEB      TE                    ;* ENABLE TRANSMITTER
   231   F06F 9FFD                CLB      TIR                   ;*
   232   F071 07DCFD              BBS      0,PORT6,*             ;* CHECK INT PORT
   233   F074 B509                LDA      WORKO,X               ;*
   234   F076 85E6                STA      RB_TB                 ;* SET RB_TB
   235   F078 97FDFD              BBC      TIR,*                 ;* TIR - 1 ? -- (TBE - 1 ?)
   236   F07B 9FFD                CLB      TIR                   ;*
   237   F07D FFE8                CLB      SIOEN                 ;*
   238   F07F 60                  RTS                            ;*
```

```
* M50745 ASSEMBLER V.2.01C *                                        P. 005

E  SEQ.  LOC.  OBJ..    ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   239                          .PAGE
   240                  ;*************************************
   241                  ;       ***  SERIAL I/O  ***         **
   242                  ;*************************************
   243                  ; This 2 pages program transmits and receives 2-byte data in the full-duplex method. Send the transmitted data to WORK and
   244                  ; WORK+1 memories and select the baud rate flags (FB1250~FB2H), then execute the SERIAL00 program. This page shows the
   245                  ; initial value and baud rate setting.
   246                  ;
   247                  SERIAL00:                               ;*                           |
   248   F080  1F00             CLB     ERRF0                   ;*                     So    |--->--|
   249   F082  3F00             CLB     ERRF1                   ;*                           |      |
   250   F084  5F00             CLB     ERRF2                   ;*                     Sin   |---<--|
   251   F086  3CE0D7           LDM     #$E0,PORT3D             ;* %11100000 = PORT3D        |
   252   F089  3C00FD           LDM     #$00,IRQ2               ;*                       (hardware)
   253   F08C  0FE8             SEB     FOSC2_8                 ;* fosc / (2*4*4)
   254                  SERIAL01:                               ;* when fosc = 10 MHz
   255   F08E  F70105           BBC     FB2H,SERIAL02           ;*
   256   F091  3CF9EA           LDM     #$F9,BRG                ;* SET 1.25KHz
   257   F094  803D             BRA     SERIALEN                ;*
   258                  SERIAL02:                               ;*
   259   F096  D70105           BBC     FB5,SERIAL03            ;*
   260   F099  3C7CEA           LDM     #$7C,BRG                ;* SET 2.5KHz
   261   F09C  8035             BRA     SERIALEN                ;*
   262                  SERIAL03:                               ;*
   263   F09E  B70105           BBC     FB12H,SERIAL04          ;*
   264   F0A1  3C31EA           LDM     #$31,BRG                ;* SET 6.25KHz
   265   F0A4  802D             BRA     SERIALEN                ;*
   266                  SERIAL04:                               ;*
   267   F0A6  970105           BBC     FB25,SERIAL05           ;*
   268   F0A9  3C18EA           LDM     #$18,BRG                ;* SET 12.5KHz
   269   F0AC  8025             BRA     SERIALEN                ;*
   270                  SERIAL05:                               ;*
   271   F0AE  1FE8             CLB     FOSC2_8                 ;* fosc / (2*4)
   272   F0B0  770105           BBC     FB10,SERIAL06           ;*
   273   F0B3  3C7CEA           LDM     #$7C,BRG                ;* SET 10KHz
   274   F0B6  801B             BRA     SERIALEN                ;*
   275                  SERIAL06:                               ;*
   276   F0B8  570105           BBC     FB50,SERIAL07           ;*
   277   F0BB  3C18EA           LDM     #$18,BRG                ;* SET 50KHz
   278   F0BE  8013             BRA     SERIALEN                ;*
   279                  SERIAL07:                               ;*
   280   F0C0  370105           BBC     FB250,SERIAL08          ;*
   281   F0C3  3C04EA           LDM     #$04,BRG                ;* SET 250KHz
   282   F0C6  800B             BRA     SERIALEN                ;*
   283                  SERIAL08:                               ;*
   284   F0C8  170105           BBC     FB1250,SERIAL09         ;*
   285   F0CB  3C00EA           LDM     #$00,BRG                ;* SET 1.25MHz
   286   F0CE  8003             BRA     SERIALEN                ;*
   287                  SERIAL09:                               ;*
   288   F0D0  0F00             SEB     ERRF0                   ;*
   289   F0D2  60               RTS                             ;*
```

```
 • M50745 ASSEMBLER V.2.01C •                                      P. 006

E  SEQ.  LOC.  OBJ..      ....•....1....•....2....•....SOURCE STATEMENT....5....•.

   290                            .PAGE
   291                    ; This program transmits 2-byte data and at the same time receiving the 2-byte data in full-duplex.
   292                    ;
   293                    SERIALEN:                               ;• SET SIOCON %1111X1XX (X IS SELECT_BIT)
   294   F0D3 EFE8                SEB     SIOEN                   ;•
   295   F0D5 CFE8                SEB     UAR_CIO                 ;•
   296   F0D7 AFE8                SEB     RE                      ;•
   297   F0D9 8FE8                SEB     TE                      ;•
   298   F0DB 4FE8                SEB     SREADYE                 ;•
   299                    ;                                       ;•
   300   F0DD 9FFD                CLB     TIR                     ;•
   301   F0DF A509                LDA     WORKO                   ;•
   302   F0E1 85E6                STA     RB_TB                   ;• SET RB_TB
   303                    ;                                       ;•
   304   F0E3 97FDFD              BBC     TIR,•                   ;• TIR = 1 ? -- (TBE = 1 ?)
   305   F0E6 9FFD                CLB     TIR                     ;•
   306   F0E8 A50A                LDA     WORKO+1                 ;•
   307   F0EA 85E6                STA     RB_TB                   ;• SET RB_TB AND TBE IS CLEARED AUTOMATICALLY
   308                    ;                                       ;•
   309   F0EC 77FDFD              BBC     RIR,•                   ;• RIR = 1 ? -- (RBF = 1 ?)
   310   F0EF 7FFD                CLB     RIR                     ;•
   311   F0F1 A5E6                LDA     RB_TB                   ;• RBF IS CLEARED  AUTOMATICALLY
   312   F0F3 8503                STA     WORKI                   ;•
   313                    ;                                       ;•
   314   F0F5 C509                CMP     WORKO                   ;•
   315   F0F7 F002                BEQ     SERIAL20                ;•
   316   F0F9 2F00                SEB     ERRF1                   ;•
   317                    SERIAL20:                               ;•
   318   F0FB 77FDFD              BBC     RIR,•                   ;• RIR = 1 ?
   319   F0FE A5E6                LDA     RB_TB                   ;• RBF IS CLEARED AUTOMATICALLY
   320   F100 8504                STA     WORKI+1                 ;•
   321   F102 C50A                CMP     WORKO+1                 ;•
   322   F104 F002                BEQ     SERIAL30                ;•
   323   F106 4F00                SEB     ERRF2                   ;•
   324                    SERIAL30:                               ;•
   325   F108 FFE8                CLB     SIOEN                   ;•
   326   F10A 60                  RTS                             ;•
```

```
* M50745 ASSEMBLER V.2.01C *                                   P. 004

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    235                             .PAGE
    236                   ;************************************
    237                   ;       ***  SERIAL I/O  ***         **
    238                   ;************************************
    239                   ; This program transmits 1-byte data. Before execution, the transmitted data must be set in WORK0.
    240                   ;
    241                   SIO08:                                ;*                        (hardware)
    242  F09A 3CE0D7                LDM     #$E0,PORT3D         ;* %11100000 = PORT3D          |
    243                   ;                                     ;*                        So    |------>
    244  F09D 1FE8                  CLB     FOSC2_8             ;* fosc / (2*4)           Sclk  |------>
    245                   ;                                     ;* when fosc = 10 MHz
    246  F09F 3C00EA                LDM     #$00,BRG            ;* SET 1.25MHz
    247                   ;                                     ;* SET SIOCON %11X1XXXX (X IS SELECT_BIT)
    248  F0A2 EFE8                  SEB     SIOEN               ;* ENABLE SERIAL I/O
    249  F0A4 CFE8                  SEB     UAR_CIO             ;* ENABLE CLOLKED SERIAL
    250  F0A6 8FE8                  SEB     TE                  ;* ENABLE TRANSMITTER
    251  F0A8 6FE8                  SEB     TIC                 ;* SET INTERRUPT REQUEST FROM TSC
    252  F0AA 9FFD                  CLB     TIR                 ;*
    253  F0AC A515                  LDA     WORK0               ;*
    254  F0AE 85E6                  STA     RB_TB               ;* SET RB_TB
    255  F0B0 97FDFD                BBC     TIR,*               ;* TIR = 1 ? -- (TSC = 1 ?)
    256  F0B3 9FFD                  CLB     TIR                 ;*
    257  F0B5 FFE8                  CLB     SIOEN               ;*
    258  F0B7 60                    RTS                         ;*
    259                             .NLIST                      ;
```

```
* M50745 ASSEMBLER V.2.01C *                                    P. 008

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   350                              .PAGE
   351                   ;**************************************
   352                   ;       ***  SERIAL I/O  ***          **
   353                   ;**************************************
   354                   ; This program transmits 32-bit data continuously. Before the execution, the transmitted data must be set in WORK0 to WORK0+3.
   355                   ;
   356                   SIO32:                                 ;*                          (hardware)
   357   F127 3CE0D7             LDM      #$E0,PORT3D           ;* %11100000 - PORT3D             |
   358                   ;                                      ;*                          So    |------>
   359   F12A 1FE8               CLB      FOSC2_8               ;* fosc / (2*4)             Sclk  |------>
   360                   ;                                      ;* when fosc - 10 MHz
   361   F12C 3C00EA             LDM      #$00,BRG              ;* SET 1.25MHz
   362                   ;                                      ;* SET SIOCON %11X1XXXX (X IS SELECT_BIT)
   363   F12F EFE8               SEB      SIOEN                 ;* ENABLE SERIAL I/O
   364   F131 CFE8               SEB      UAR_CIO               ;* ENABLE CLOLKED SERIAL
   365   F133 8FE8               SEB      TE                    ;* ENABLE TRANSMITTER
   366   F135 A203               LDX      #$03                  ;*
   367                   SIO33:                                 ;*
   368   F137 B509               LDA      WORK0,X               ;*
   369   F139 85E6               STA      RB_TB                 ;* SET RB_TB(when 8 bits)
   370   F13B CA                 DEX                            ;*
   371   F13C B509               LDA      WORK0,X               ;*
   372   F13E 85E6               STA      RB_TB                 ;* SET RB_TB(when 16 bits)
   373   F140 CA                 DEX                            ;*
   374                   SIO34:                                 ;*
   375   F141 B509               LDA      WORK0,X               ;*
   376   F143 17E7FD             BBC      TBE,*                 ;*
   377   F146 85E6               STA      RB_TB                 ;* SET RB_TB(when 24,32 bits)
   378   F148 CA                 DEX                            ;*
   379   F149 10F6               BPL      SIO34                 ;*
   380   F14B 57E7FD             BBC      TSC,*                 ;*
   381   F14E FFE8               CLB      SIOEN                 ;*
   382   F150 60                 RTS                            ;*
```

```
E  SEQ.   LOC.   OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   138                      ;*************************************
   139                      ;      *** UART TYPE SERIAL I/O ***     **
   140                      ;*************************************
   141                      ;   This program sets the work area for UART of sample program.
   142    0000              WORKI    =        $00               ;*
   143    0002              WORKO    =        $02               ;*
   144    0004              FLAGS    =        $04               ;*
   145    0,04              ERRF0    =        0,FLAGS           ;*
   146    1,04              ERRF1    =        1,FLAGS           ;*
   147    2,04              ERRF2    =        2,FLAGS           ;*
   148    0005              UES      =        $05               ;*UART ERROR STATUS
   149    3,05              UESOE    =        3,UES             ;*
   150    4,05              UESPE    =        4,UES             ;*
   151    5,05              UESFE    =        5,UES             ;*
   152    6,05              UESSE    =        6,UES             ;*
   153    0006              BRGFLG   =        $06               ;*FLAGS SELECT BIT RATE(bps)
   154    0,06              FB3      =        0,BRGFLG          ;*300
   155    1,06              FB6      =        1,BRGFLG          ;*600      when fosc = 9.8304MHz
   156    2,06              FB12     =        2,BRGFLG          ;*1200
   157    3,06              FB24     =        3,BRGFLG          ;*2400
   158    4,06              FB48     =        4,BRGFLG          ;*4800
   159    5,06              FB96     =        5,BRGFLG          ;*9600
   160    6,06              FB192    =        6,BRGFLG          ;*19200
   161    7,06              FB384    =        7,BRGFLG          ;*38400
   162    0007              BRGFLG2  =        $07               ;*   CONTINUE
   163    0,07              FB768    =        0,BRGFLG2         ;*76800
   164    1,07              FB1536   =        1,BRGFLG2         ;*153600
   165    2,07              FB3072   =        2,BRGFLG2         ;*307200
   166    7,07              FPEV_OD  =        7,BRGFLG2         ;*THIS FLAG SELECTS ODD PARITY OR OR EVEN ONE.
   167    0008              UARTDF   =        $08               ;*FLAGS SELECT UART DATA FORMAT
   168    0,08              FUF701   =        0,UARTDF          ;*1.7.0.1
   169    1,08              FUF711   =        1,UARTDF          ;*1.7.1.1
   170    2,08              FUF702   =        2,UARTDF          ;*1.7.0.2
   171    3,08              FUF712   =        3,UARTDF          ;*1.7.1.2
   172    4,08              FUF801   =        4,UARTDF          ;*1.8.0.1
   173    5,08              FUF811   =        5,UARTDF          ;*1.8.1.1
   174    6,08              FUF802   =        6,UARTDF          ;*1.8.0.2
   175    7,08              FUF812   =        7,UARTDF          ;*1.8.1.2
   176                      ; Changing the address of the flags to port address causes the port input signal to define the UART's baud rate
   177                      ; and transmission format.
```

```
• M50745 ASSEMBLER V.2.01C •                                     P. 003

E  SEQ.  LOC.  OBJ..       ....•....1....•....2....•....SOURCE STATEMENT....5....•.

    246                                .PAGE
    247                    ;•••••••••••••••••••••••••••••••••••••••
    248                    ;      •••  UART TYPE SERIAL I/O  •••     ••
    249                    ;•••••••••••••••••••••••••••••••••••••••
    250                    ;
    251                    ; This 3 pages program shows the 2-byte data transmission and receiving program in the full-duplex method. Before executing the
    252                    ; program UARTS00, set the transmitted data to WORK, WORK+1, the baud rate flags (FB3~FB3072) and UART format flags
    253                    ; (FUF701~FUF812). This program page sets the UART's bit rate with the flags.
    254                    ;
    255                    UARTTS00:                                      ;•
    256  F074 1F04                    CLB      ERRF0                  ;•
    257  F076 0FE8                    SEB      FOSC2_8                ;•
    258  F078 170605                  BBC      FB3,UARTTS01           ;•
    259  F07B 3CFFEA                  LDM      #$FF,BRG               ;•      300bps  SET
    260  F07E 8052                    BRA      UARTTSEN               ;•
    261                    UARTTS01:                                      ;•
    262  F080 370605                  BBC      FB6,UARTTS02           ;•
    263  F083 3C7FEA                  LDM      #$7F,BRG               ;•      600bps  SET
    264  F086 804A                    BRA      UARTTSEN               ;•
    265                    UARTTS02:                                      ;•
    266  F088 570605                  BBC      FB12,UARTTS03          ;•
    267  F08B 3C3FEA                  LDM      #$3F,BRG               ;•    1,200bps  SET
    268  F08E 8042                    BRA      UARTTSEN               ;•
    269                    UARTTS03:                                      ;•
    270  F090 770605                  BBC      FB24,UARTTS04          ;•
    271  F093 3C1FEA                  LDM      #$1F,BRG               ;•    2,400bps  SET
    272  F096 803A                    BRA      UARTTSEN               ;•
    273                    UARTTS04:                                      ;•
    274  F098 970605                  BBC      FB48,UARTTS05          ;•
    275  F09B 3C0FEA                  LDM      #$0F,BRG               ;•    4,800bps  SET
    276  F09E 8032                    BRA      UARTTSEN               ;•
    277                    UARTTS05:                                      ;•
    278  F0A0 B70605                  BBC      FB96,UARTTS06          ;•
    279  F0A3 3C07EA                  LDM      #$07,BRG               ;•    9,600bps  SET
    280  F0A6 802A                    BRA      UARTTSEN               ;•
    281                    UARTTS06:                                      ;•
    282  F0A8 D70605                  BBC      FB192,UARTTS07         ;•
    283  F0AB 3C03EA                  LDM      #$03,BRG               ;•   19,200bps  SET
    284  F0AE 8022                    BRA      UARTTSEN               ;•
    285                    UARTTS07:                                      ;•
    286  F0B0 F70605                  BBC      FB384,UARTTS08         ;•
    287  F0B3 3C01EA                  LDM      #$01,BRG               ;•   38,400bps  SET
    288  F0B6 801A                    BRA      UARTTSEN               ;•
    289                    UARTTS08:                                      ;•
    290  F0B8 1FE8                    CLB      FOSC2_8                ;•
    291  F0BA 170705                  BBC      FB768,UARTTS09         ;•
    292  F0BD 3C03EA                  LDM      #$03,BRG               ;•   76,800bps  SET
    293  F0C0 8010                    BRA      UARTTSEN               ;•
    294                    UARTTS09:                                      ;•
    295  F0C2 370705                  BBC      FB1536,UARTTS10        ;•
    296  F0C5 3C01EA                  LDM      #$01,BRG               ;•  153,600bps  SET
    297  F0C8 8008                    BRA      UARTTSEN               ;•
    298                    UARTTS10:                                      ;•
    299  F0CA 570708                  BBC      FB3072,UARTTS11        ;•
    300  F0CD 3C00EA                  LDM      #$00,BRG               ;•  307,200bps  SET
    301  F0D0 8000                    BRA      UARTTSEN               ;•
    302                    UARTTSEN:                                      ;•
    303  F0D2 20D8F0                  JSR      UART                   ;•
    304                    UARTTS11:                                      ;•
    305  F0D5 0F04                    SEB      ERRF0                  ;•
    306  F0D7 60                      RTS                             ;•
```

```
E  SEQ.  LOC.  OBJ..     ....•....1....•....2....•....SOURCE STATEMENT....5....•.

   307                                  .PAGE
   308                          ;This program sets the UART's transmission format with the flags.
   309                          ;••••••••••••••••••••••••••••••••••••••
   310                          ;              UART SUBROUTINE        ••
   311                          ;•••••••••••••••••••••••••••••••••••••• (START-BIT . CHAR[7,8] . PARITY[0,1]
   312                          UART:                                 ;•  . STOP-BIT[1,2]) [ ,ODD,EVEN]
   313                          UART701:                              ;•
   314   F0D8 170805                     BBC     FUF701,UART711       ;•
   315   F0DB 3C01E9                     LDM     #$01,UACON           ;• UACON SET     %0001
   316   F0DE 803B                       BRA     UARTEN               ;•               (1.7.0.1)
   317                          UART711:                              ;•
   318   F0E0 370805                     BBC     FUF711,UART702       ;•
   319   F0E3 3C03E9                     LDM     #$03,UACON           ;• UACON SET     %0011
   320   F0E6 802E                       BRA     UARTPA               ;•               (1.7.1.1) EVEN
   321                          UART702:                              ;•
   322   F0E8 570805                     BBC     FUF702,UART712       ;•
   323   F0EB 3C09E9                     LDM     #$09,UACON           ;• UACON SET     %1001
   324   F0EE 802B                       BRA     UARTEN               ;•               (1.7.0.2)
   325                          UART712:                              ;•
   326   F0F0 770805                     BBC     FUF712,UART801       ;•
   327   F0F3 3C0BE9                     LDM     #$0B,UACON           ;• UACON SET     %1011
   328   F0F6 801E                       BRA     UARTPA               ;•               (1.7.1.2) EVEN
   329                          UART801:                              ;•
   330   F0F8 970805                     BBC     FUF801,UART811       ;•
   331   F0FB 3C00E9                     LDM     #$00,UACON           ;• UACON SET     %0000
   332   F0FE 801B                       BRA     UARTEN               ;•               (1.8.0.1)
   333                          UART811:                              ;•
   334   F100 B70805                     BBC     FUF811,UART802       ;•
   335   F103 3C02E9                     LDM     #$02,UACON           ;• UACON SET     %0010
   336   F106 800E                       BRA     UARTPA               ;•               (1.8.1.1) EVEN
   337                          UART802:                              ;•
   338   F108 D70805                     BBC     FUF802,UART812       ;•
   339   F10B 3C08E9                     LDM     #$08,UACON           ;• UACON SET     %1000
   340   F10E 800B                       BRA     UARTEN               ;•               (1.8.0.2)
   341                          UART812:                              ;•
   342   F110 F7080B                     BBC     FUF812,UART822       ;•
   343   F113 3C0AE9                     LDM     #$0A,UACON           ;• UACON SET     %1010
   344                          UARTPA:                               ;•
   345   F116 F70702                     BBC     FPEV_OD,UARTEN       ;•
   346   F119 4FE9                       SEB     EVEN_OD              ;•PARITY IS ODD
   347                          UARTEN:                               ;•
   348   F11B 2021F1                     JSR     UART10               ;•
   349                          UART822:                              ;•
   350   F11E 0F04                       SEB     ERRF0                ;•
   351   F120 60                         RTS                          ;•
```

```
 • M50745 ASSEMBLER V.2.01C •                                P. 006

 E  SEQ.  LOC.  OBJ..    ....•....1....•....2....•....SOURCE STATEMENT....5....•.

    352                            .PAGE
    353                   ;  This program shows the 2-byte data transmission and receiving program in the full-duplex method.
    354                   UART10:                                   ;• UART SET %1011000X - SIOCON     TxD |-->--|
    355   F121 3CA0D7              LDM      #$A0,PORT3D             ;•              %10100000 - PORT3D       |     |
    356   F124 EFE8                SEB      SIOEN                   ;• ENABLE SERIAL I/O                  RxD |--<--|
    357   F126 DFE8                CLB      UAR_CIO                 ;• ENABLE CLOLKED SERIAL                  |
    358   F128 AFE8                SEB      RE                      ;• ENABLE RECEIVER                     (hardware)
    359   F12A 8FE8                SEB      TE                      ;• ENABLE TRANSMITTER
    360   F12C 3C0005              LDM      #$00,UES                ;•
    361   F12F 3F04                CLB      ERRF1                   ;•
    362   F131 5F04                CLB      ERRF2                   ;•
    363                   ;                                         ;•
    364   F133 A502                LDA      WORKO                   ;•
    365   F135 85E6                STA      RB_TB                   ;• SET RB_TB
    366                   ;                                         ;•
    367   F137 17E7FD              BBC      TBE,•                   ;•
    368   F13A A503                LDA      WORKO+1                 ;•
    369   F13C 85E6                STA      RB_TB                   ;• SET RB_TB
    370                   UART12:                                   ;•
    371   F13E 37E7FD              BBC      RBF,•                   ;•
    372   F141 D7E704              BBC      SE,UART120              ;•
    373   F144 A5E7                LDA      SIOSTS                  ;•
    374   F146 8505                STA      UES                     ;•
    375                   UART120:                                  ;•
    376   F148 A5E6                LDA      RB_TB                   ;•
    377   F14A 8500                STA      WORKI                   ;•
    378   F14C C502                CMP      WORKO                   ;•
    379   F14E F002                BEQ      UART13                  ;•
    380   F150 2F04                SEB      ERRF1                   ;•
    381                   UART13:                                   ;•
    382   F152 A903                LDA      #$03                    ;•
    383   F154 25E7                AND      SIOSTS                  ;•
    384   F156 85E7                STA      SIOSTS                  ;• WRITE DUMMY DATA AND CLEAR ERROR_FLAGS
    385   F158 37E7FD              BBC      RBF,•                   ;•
    386   F15B A5E6                LDA      RB_TB                   ;•
    387   F15D 8501                STA      WORKI+1                 ;•
    388   F15F C503                CMP      WORKO+1                 ;•
    389   F161 F002                BEQ      UART14                  ;•
    390   F163 4F04                SEB      ERRF2                   ;•
    391                   UART14:                                   ;•
    392   F165 206BF1              JSR      UART_E_C                ;• CHECK UART ERROR
    393   F168 FFE8                CLB      SIOEN                   ;•
    394   F16A 60                  RTS                              ;•
```

```
* M50745 ASSEMBLER V.2.01C *                            P. 007

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    395                           .PAGE
    396                  ; This program examines the errors of the received data.
    397                  UART_E_C:                                  ;* CHECK UART ERROR
    398  F169 27041D              BBS     ERRF1,ERROR               ;*
    399  F16C 47041A              BBS     ERRF2,ERROR               ;*
    400  F16F C7E717              BBS     SE,ERROR                  ;*  SE = 1 ?
    401  F172 A7E714              BBS     FE,ERROR                  ;*  FE = 1 ?
    402  F175 87E711              BBS     PE,ERROR                  ;*  PE = 1 ?
    403  F178 67E70E              BBS     OE,ERROR                  ;*  OE = 1 ?
    404  F17B C7050B              BBS     UESSE,ERROR               ;*  FIRST SE = 1 ?
    405  F17E A70508              BBS     UESFE,ERROR               ;*  FIRST FE = 1 ?
    406  F181 870505              BBS     UESPE,ERROR               ;*  FIRST PE = 1 ?
    407  F184 670502              BBS     UESOE,ERROR               ;*  FIRST OE = 1 ?
    408  F187 8010                BRA     NOERROR                   ;*
    409                  ;*****************************************:*
    410                  ERROR:                                     ;*
    411  F189 17DCFD              BBC     0,PORT6,ERROR             ;*  INT ON ?
    412  F18C 2200                JSR     ¥T100m                    ;*  WAIT 100 ms
    413  F18E 17DCF8              BBC     0,PORT6,ERROR             ;*  INT ON ?
    414                  ERROR1:                                    ;*
    415  F191 07DCFD              BBS     0,PORT6,ERROR1            ;*  INT ON ?
    416  F194 2200                JSR     ¥T100m                    ;*  WAIT 100 ms
    417  F196 07DCF8              BBS     0,PORT6,ERROR1            ;*  INT ON ?
    418                  NOERROR:
    419  F199 60                  RTS                               ;*
```

```
• M50745 ASSEMBLER V.2.01C •                              P. 002

E  SEQ.  LOC.  OBJ..    ....•....1....•....2....•....SOURCE STATEMENT....5....•.

     1                              .NLIST
   140                              .LIST
   141                  ; This program sets the work area for the multiply/divide calculation.
   142   0003           n           =       $03                  ;
   143   0008           W1          =       $08                  ;
   144   000D           W2          =       $0D                  ;
   145   000F           DIVISOR     =       $0F                  ;
   146   0010           SP          =       $10                  ;
   147   0015           XO          =       $15                  ;
   148                              .NLIST
   171                              .LIST
   172                  ; This program divides n-byte data with 1-byte data (DIVISOR). The dividend is stored in the n-byte memories which start from
   173                  ; address W1. The quotient is stored in the n-byte area which starts from address W2. Also, the surplus data is stored in address W1.
   174                  ; "n" must be:
   175                  ;     #n=1    when    8 bits/8 bits
   176                  ;     #n=2    when    16 bits/8 bits
   177                  ;
   178                  DIVM: #n=3    when    24 bits/8 bits     ,              |        :       |
   179   F021  A900                 LDA     #0                   ;(2)        W1  |----------------|
   180                  ;    ※                                   ;             +1|----------------|
   181   F023  A202                 LDX     #n-1                 ;(2)          +2|----------------|
   182                  DIV1:                                    ;               |----------------|
   183   F025  9509                 STA     W1+1,X               ;(5)            |----------------|
   184   F027  A50F                 LDA     DIVISOR              ;(3)            |       :        |
   185   F029  E208                 DIV     W1,X                 ;(16)       W2  |----------------|
   186   F02B  950D                 STA     W2,X                 ;(5)          +1|----------------|
   187   F02D  68                   PLA                          ;(4)          +2|----------------|
   188   F02E  49FF                 EOR     #$FF                 ;(2)            |----------------|
   189   F030  9508                 STA     W1,X                 ;(5)            |----------------|
   190   F032  CA                   DEX                          ;(2)            |       :        |
   191   F033  10F0                 BPL     DIV1                 ;(2/4)   DIVISOR|----------------|
   192   F035  60                   RTS                          ;(6)            |----------------|
   193                  ;                                        ;               |       :        |
   194                  ; The execution time as the calculation of n-byte division: 46 (n—1)+54 mc (n≧1)
```

```
 • M50745 ASSEMBLER V.2.01C •                              P. 003

E  SEQ.  LOC.  OBJ..      ....•....1....•....2....•....SOURCE STATEMENT....5....•.

    195                           .PAGE
    196                   ;
    197                   ;                       n-1
    198                   ;  This program shows  Σ aiXi (1≦n≦64).
    199                   ;                       i=0
    200                   ;  "X" is a variable which specifies RAM area (X0~Xn-1), "a" is a constant which specifies ROM area (a0~an-1). All are 8-bit data.
    201                   ;  The result is stored in the 3-byte data specified by address SP.
    202                   ;  When setting "n", #n of the ※ part must be in the specified condition.
    203                   SOFP:                                  ;
    204  F036 3C0010              LDM     #$00,SP                ;(4)          ---    RAM     ---
    205  F039 3C0011              LDM     #$00,SP+1              ;(4)       X0 |---------------|
    206  F03C 3C0012              LDM     #$00,SP+2              ;(4)       X1 |---------------|
    207                   ;   ※                                  ;             |---------------|
    208  F03F A202                LDX     #n-1                   ;(2)          |       :       |
    209                   SOFP1:                                 ;        Xn-1 |---------------|
    210  F041 BD5AF0              LDA     a0,X                   ;(4)          |---------------|
    211  F044 6215                MUL     X0,X                   ;(15)         |               |
    212  F046 18                  CLC                            ;(2)          |               |
    213  F047 6510                ADC     SP                     ;(3)       SP |---------------|
    214  F049 8510                STA     SP                     ;(4)       +1 |---------------|
    215  F04B 68                  PLA                            ;(4)       +2 |---------------|
    216  F04C 6511                ADC     SP+1                   ;(3)          |---------------|
    217  F04E 8511                STA     SP+1                   ;(4)          |               |
    218  F050 A512                LDA     SP+2                   ;(3)          ---    ROM     ---
    219  F052 6900                ADC     #$00                   ;(2)       A0 |---------------|
    220  F054 8512                STA     SP+2                   ;(4)       A1 |---------------|
    221  F056 CA                  DEX                            ;(2)          |---------------|
    222  F057 10E8                BPL     SOFP1                  ;(2/4)        |       :       |
    223  F059 60                  RTS                            ;(6)     An-1 |---------------|
    224                   ;                                      ;             |---------------|
    225                   a0:                                    ;             |               |
    226  F05A 01020304            .BYTE   1,2,3,4,5,6,7,8        ; any
         F05E 05060708
    227  F062 0B0C0D0E            .BYTE   11,12,13,14,15,16,17,18 ;
         F066 0F101112
    228  F06A 15161718            .BYTE   21,22,23,24,25,26,27,28 ;
         F06E 191A1B1C
    229  F072 1F202122            .BYTE   31,32,33,34,35,36,37,38 ;
         F076 23242526
    230  F07A 292A2B2C            .BYTE   41,42,43,44,45,46,47,48 ;
         F07E 2D2E2F30
    231  F082 33343536            .BYTE   51,52,53,54,55,56,57,58 ;
         F086 3738393A
    232  F08A 3D3E3F40            .BYTE   61,62,63,64,65,66,67,68 ;
         F08E 41424344
    233  F092 4748494A            .BYTE   71,72,73,74,75,76,77,78 ;
         F096 4B4C4D4E
    234                   ; The execution time is shown as: 52 (n−1)+70mc (n≧0)    here n is the calculation time.
```

```
* M50745 ASSEMBLER V.2.01C *                          P. 003

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

     199                  ;
     200                  ;                   M A S T E R                   S L A V E
     201                  ;                  -------------┐              ┌-------------
     202                  ;                               |              |
     203                  ;                            WR |------------->| P67/W
     204                  ;                               |              |
     205                  ;                            RD |------------->| 66/R
     206                  ;                               |   ┌-------┐  |
     207                  ;       M37450S-FP    A15-A13 |-->|DECODER|-->| 65/CS    M37450M2-XXXSP
     208                  ;                               |   └-------┘  |
     209                  ;                            A0 |------------->| P64/A0
     210                  ;                               |              |
     211                  ;                         D7-D0 |<------------>| P 5
     212                  ;                               |              |
     213                  ;                  -------------┘              └-------------
     214                  ;                                  (hardware)
     215                  ;
```

```
* M50745 ASSEMBLER V.2.01C *                                    P. 002

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

   141                    ;  This program sets the work area for the master microcomputer data I/O buffer.
   142   0000             WCOM     =        $00                  ;*
   143   0.00             FA0      =        0,WCOM               ;* USER SELECT FLAG   1 = COMMAND , 0 = DATA
   144   0010             WCMD     =        $10                  ;* COMMAND DATA WORK AREA
   145   0020             WDAT     =        $20                  ;* DATA         WORK AREA
   146   2000             MDAT     =        $2000                ;* SLAVE's DATA    ADDRESS
   147   2001             MCMD     =        $2001                ;* SLAVE's COMMAND ADDRESS
   148                             *=       $F000                ;*

   159                    ; This is the data bus buffer I/O sample program of the master microcomputer.
   160                    ; This part is the main routine. It also shows the 1-byte read/write.
   161   F006 CFDF                 SEB      RDYE                 ;* MASTER HARD WAIT (WHEN 10MHz)
   162   F008 3C00DD               LDM      #$00,PORT6D          ;*
   163   F00B 3C00FE               LDM      #$00,ICON1           ;*
   164   F00E 3C00FF               LDM      #$00,ICON2           ;*
   165   F011 3C0000               LDM      #$00,WCOM            ;*
   166                    MAS:                                   ;*
   167   F014 07DCFD               BBS      0,PORT6,*            ;*
   168   F017 201DF0               JSR      RDWR                 ;*
   169   F01A 80F8                 BRA      MAS                  ;*
   170   F01C 42                   STP                           ;*
```

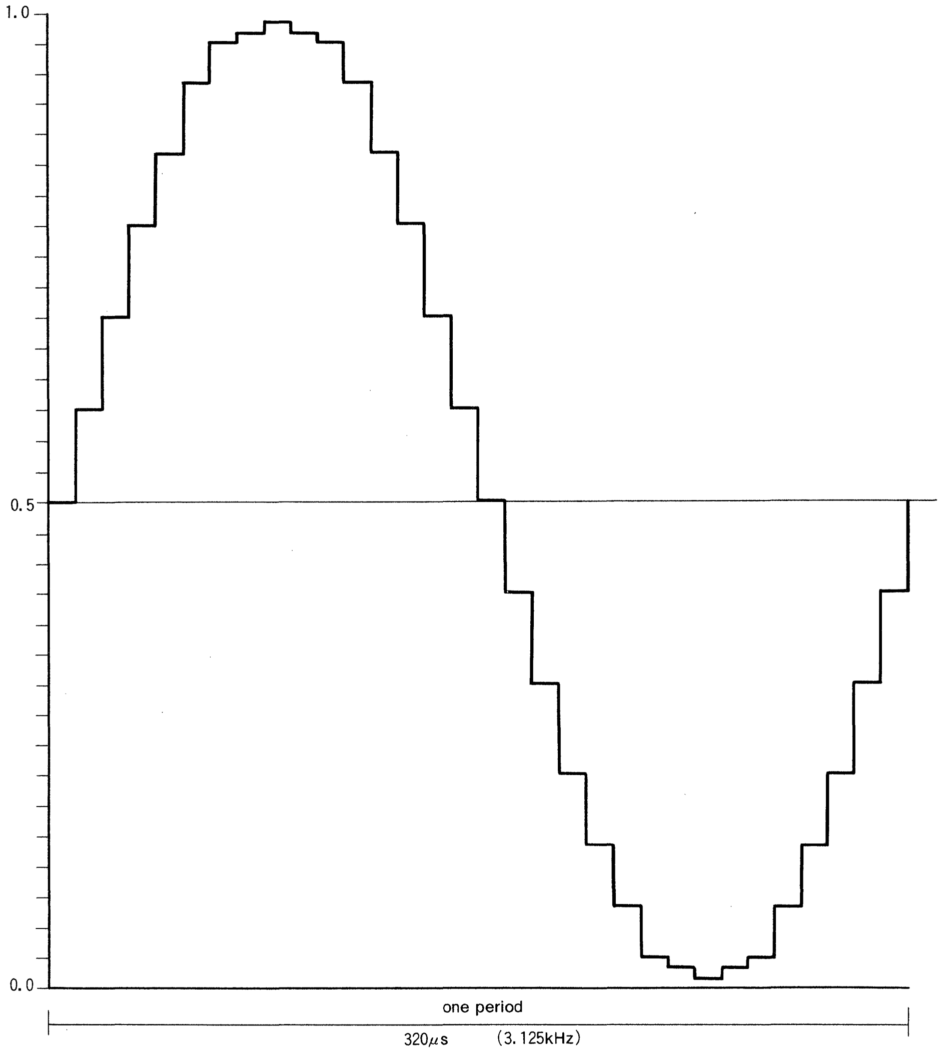
```
* M50745 ASSEMBLER V.2.01C *                        P. 003

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    172                   ; This part is the routine which the slave microcomputer reads and writes.
    173                   RDWR:                                    ;*
    174  F01D AD0120              LDA     MCMD                     ;* READ $2001
    175  F020 8500                STA     WCOM                     ;*
    176                   ;                                        ;*
    177  F022 0304                BBS     0,A,RD                   ;* SLAVE IS REQUESTING OUTPUT
    178  F024 330A                BBC     1,A,WR                   ;*
    179  F026 8017                BRA     RDWREND                  ;*
    180                   RD:                                      ;*
    181  F028 AD0020              LDA     MDAT                     ;* READ $2000
                                                                   ;*
                                                                   ;*
    185  F02C 8520                STA     WDAT                     ;* STORE WORK AREA
    186  F02E 800F                BRA     RDWREND                  ;*
    187                   WR:                                      ;* MASTER WRITE SLAVE
    188  F030 070007              BBS     FA0,WRCMD                ;*
    189  F033 A520                LDA     WDAT                     ;*
    190  F035 8D0020              STA     MDAT                     ;* WRITE $2000
    191  F038 8005                BRA     RDWREND                  ;*
    192                   WRCMD:                                   ;* SLAVE REQUEST COMMAND
    193  F03A A510                LDA     WCMD                     ;*
    194  F03C 8D0120              STA     MCMD                     ;* WRITE $2001
    195                   RDWREND:                                 ;*
    196  F03F 60                  RTS                              ;*
```

```
 * M50745 ASSEMBLER V.2.01C *                                P. 002

E  SEQ.  LOC.  OBJ..      ....*....1....*....2....*....SOURCE STATEMENT....5....*.


    141                   ; This program sets the work area of the slave microcomputer's I/O data bus buffer.
    142                   ;
    143  0000             WCMD    =       $00                  ; COMMAND DATA WORK AREA
    144  0010             WDAT    =       $10                  ; DATA         WORK AREA
    145                           *=      $F000                ;


    165                   ; This is the data bus buffer I/O sample program of the slave processor. This part is the main routine. It also shows
    166                   ; the 1-byte read/ write.
    167  F01E 3C08DC              LDM     #$08,PORT6           ;
    168  F021 3C08DD              LDM     #$08,PORT6D          ; SET P63/Prdy OUTPUT
    169  F024 7FDF                CLB     DBBM                 ; 80 MODE
    170  F026 4FDF                SEB     DBBE                 ; SET DATA BUS BUFFER(DBB)
    171                   SLV:                                 ;
    172  F028 37E506              BBC     IBF,SLV1             ;
    173  F02B 203FF0              JSR     IN                   ; INPUT 1 BYTES


    177                   SLV1:                                ;
    178  F031 07E5F4              BBS     OBF,SLV              ;
    179  F034 203AF0              JSR     OUT                  ; OUTPUT 1 BYTES
    180  F037 80EF                BRA     SLV                  ;
    181  F039 42                  STP                          ;
    182                   ; This routine outputs data to the data bus buffer.
    183                   OUT:                                 ;
    184  F03A A510                LDA     WDAT                 ;
    185  F03C 85E4                STA     DBB                  ; OBF ON
    186  F03E 60                  RTS                          ;
    187                   ; This routine inputs data from the data bus buffer.
    188                   IN:                                  ;
    189  F03F 67E506              BBS     A0,IN010             ;
    190  F042 A5E4                LDA     DBB                  ;
    191  F044 8510                STA     WDAT                 ;
    192  F046 8004                BRA     IN020                ;
    193                   IN010:                               ; STORE COMMAND
    194  F048 A5E4                LDA     DBB                  ;
    195  F04A 8500                STA     WCMD                 ;
    196                   IN020:                               ;
    197  F04C 60                  RTS                          ;
```

Sine-wave output by D-A conversion function.

The next program samples the value from the ROM table's data every 1/32 period, and outputs the data to ports D-A_1 and D-A_2 through D-A conversion.

The figure shown below shows the expected sine-wave.



ROM data

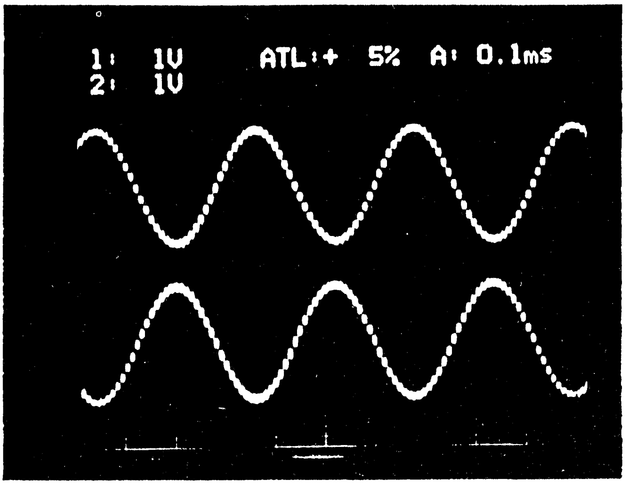
N : counter value

$$Y=\frac{\sin(90^\circ \times N/8)+1}{2} \qquad \Delta Y=Y-Y'$$

N	Y (calculated value)	data (HEX)	Y' (set value)	ΔY
0	0.5	80	0.5	0
1	0.598	99	0.598	0
2	0.692	B1	0.691	0.001
3	0.778	C7	0.777	0.001
4	0.854	DA	0.851	0.003
5	0.916	EA	0.914	0.002
6	0.962	F6	0.961	0.001
7	0.990	FD	0.988	0.002
8	1	FF	1	1
9	0.990	FD	0.988	0.002
10	0.962	F6	0.961	0.001
11	0.916	EA	0.914	0.002
12	0.854	DA	0.851	0.003
13	0.778	C7	0.777	0.001
14	0.962	B1	0.691	0.001
15	0.598	99	0.598	0
16	0.5	80	0.5	0
17	0.402	66	0.398	0.004
18	0.309	4F	0.309	0
19	0.222	38	0.218	0.004
20	0.146	25	0.145	0.001
21	0.084	15	0.082	0.002
22	0.038	09	0.035	0.003
23	0.010	02	0.008	0.002
24	0	00	0	0
25	0.010	02	0.008	0.002
26	0.038	09	0.035	0.003
27	0.084	15	0.082	0.002
28	0.146	25	0.145	0.001
29	0.222	38	0.218	0.004
30	0.309	4F	0.309	0
31	0.402	60	0.398	0.004
32	0.5	80	0.5	0

```
* M50745 ASSEMBLER V.2.01C *                                       P. 005

E  SEQ.  LOC.  OBJ..     ....*....1....*....2....*....SOURCE STATEMENT....5....*.

    143                            .PAGE                               ;*
    144                            *=        $F000                     ;*
    145                  ;*************************************************
    146                  :         ***       INITIAL       ***         **
    147                  ;*************************************************
    148                  RESET:                                        ;*
    149  F000 78                   SEI                                 ;*
    150  F001 A27F                 LDX       #$7F                      ;*
    151  F003 9A                   TXS                                 ;*
    152  F004 12                   CLT                                 ;*
    153  F005 D8                   CLD                                 ;*
    154  F006 3C1CFE               LDM       #$1C,ICON1                ;*
    155  F009 A900                 LDA       #$00                      ;*
    156  F00B 85FF                 STA       ICON2                     ;*
    157                  ;*************************************************
    158                  :    ***    D-A CONVERTER          ***        *
    159                  ;*************************************************
    160                  SUBSIN:
    161  F00D A220                 LDX       #32
    162                  L2:
    163  F00F BD1FF0               LDA       SINDAT,X
    164  F012 85E0                 STA       D_A1
    165  F014 49FF                 EOR       #$FF
    166  F016 85E1                 STA       D_A2
    167  F018 CA                   DEX
    168  F019 D002                 BNE       L1
    169  F01B 80F0                 BRA       SUBSIN
    170                  L1:
    171  F01D 80F0                 BRA       L2
    172                  SINDAT:
    173                  ;                   0  ,1  ,2  ,3  ,4  ,5  ,6  ,7  ,8  ,9
    174  F01F 8099B1C7             .BYTE     $80,$99,$B1,$C7,$DA,$EA,$F6,$FD,$FF,$FD
         F023 DAEAF6FD
         F027 FFFD
    175  F029 F6EADAC7             .BYTE     $F6,$EA,$DA,$C7,$B1,$99,$80,$66,$4F,$38
         F02D B1998066
         F031 4F38
    176  F033 25150902             .BYTE     $25,$15,$09,$02,$00,$02,$09,$15,$25,$38
         F037 00020915
         F03B 2538
    177  F03D 4F6680               .BYTE     $4F,$66,$80
```

Output waveform



APPENDIX 1. Programming notes

(1) Processor status register

- Initialization

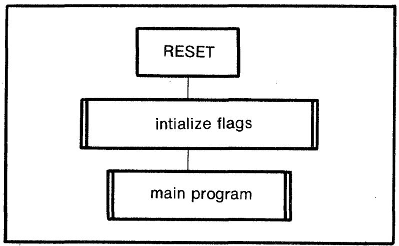
Except for the interrupt inhibit flag (I) being set to "1", the contents of the processor status register (PS) is unpredictable after reset. Therefore, flags affecting program execution must be intialized.

The T flag and D flag which affect arithmetic operations, must be initialized.

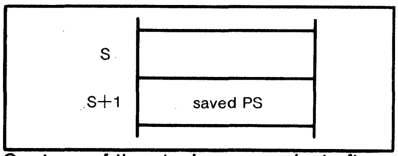
- Reference and save

To reference the PS, the contents of the PS must first be saved to the stack using the PHP instruction, then read the contents of the S+1.

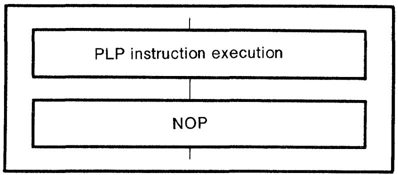
The PLP instruction can be used to return the saved PS value to the PS if required. However, a NOP instruction must be used after the execution of the PLP instruction.



Initialize flags



Contens of the stack memory just after the execution of the PHP instruction

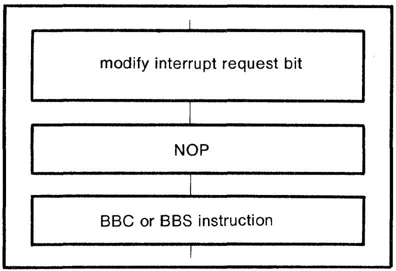


Notes on PLP instruction

(2) Interrupts

Even though the BBC and BBS instructions are executed just after the interrupt request bits are modified (by the program), those instructions are only valid for the contents before the modification.

At least one instruction cycle must be used (such as a NOP) between the modification of the interrupt request bits and the execution of BBC and BBS instructions.



Notes on modification of interrupt request bit

(3) Decimal operations

- Execution

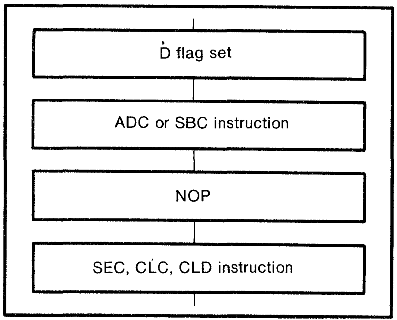
Decimal operations are performed by setting the decimal mode flag (D) and executing the ADC or SBC instruction. In this case, there must be at least one instruction following the ADC or SBC instruction before executing the SEC, CLC, or CLD instruction.

- Notes on flags in decimal mode

The N (Negative), V (Overflow), and Z (Zero) flags are ignored during decimal mode.

Also note that the C (Carry) flag is set to "1" if a digit is carried as the result of the operation, but is reset to "0" if the digit is borrowed. Therefore it can be used as a flag for judging of carried or borrowed.

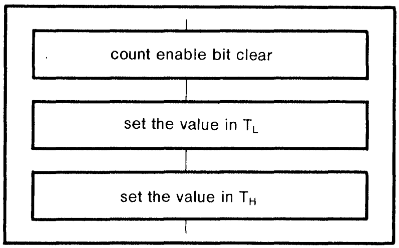
It should be initialized before the calculations.



Notes on operation in decimal mode

(4) Timer

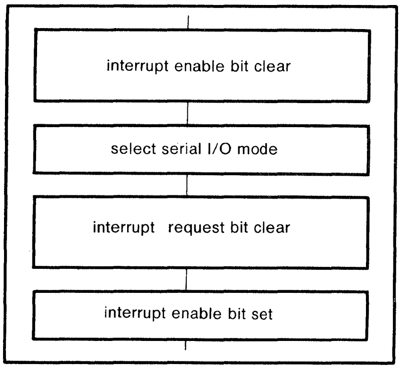
- The frequency division ratio
  The frequency division ratio when n (0~65535) is written in the timer latch is 1/(n+1).
- Write to timer
  When directly writing a value in the timer, clear the count enable bit to count disable ("0") and write in the low-order byte first and then in the high-order byte.
- Read from timer
  The timer value must be read from the high-order byte first.



Notes on setting the value to timer

(5) Serial I/O

- $\overline{S_{RDY}}$ output
  In clock synchronous serial I/O mode, if the receiver is to output an $\overline{S_{RDY}}$ using an external clock, the receive enable bit, $\overline{S_{RDY}}$ output enable bit, and transmission enable bit must be set to "1".
- Initial setting of serial I/O interrupt
  When using the serial I/O interrupt, it should be set using the following procedure:
  ①clear the interrupt enable bit to "0" by CLB instruction
  ②select serial I/O
  ③after waiting for one or more instructions use the CLB instruction to set the interrupt request bit to "0".
  ④set the interrupt enable bit to "1".



Serial I/O interrupt initialize

(6) A-D conversion

The comparator consists of coupling that lose their charge when the clock frequency is low. Therefore, f (X_{IN}) must be no less than 1MHz during A-D conversion. (If the bus cycle control bit is "1", the bus cycle is doubled and the A-D conversion time is also doubled, therefore, f (X_{IN}) must not be less than 2MHz.) Also, the STP and WIT instructions must not be executed during A-D conversion.

(7) STP instruction

The STP instruction must be executed after setting the timer 1 count enable bit (bit 4 of address $00DE_{16}$) to enable ("1").

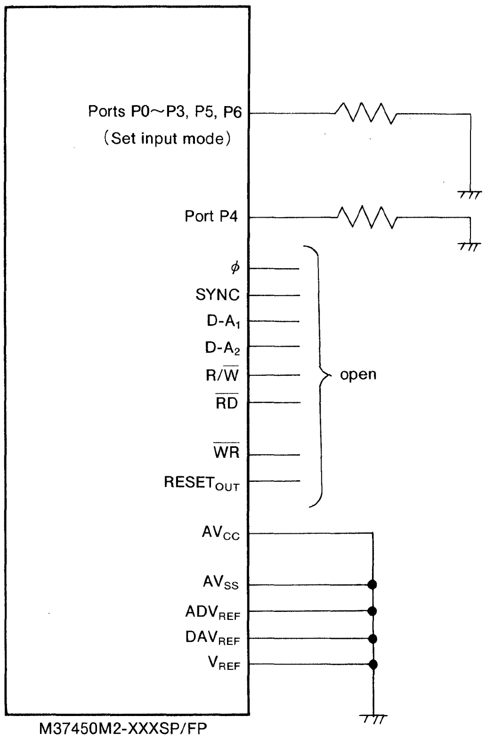
(8) Multiply/Divide instructions

1. The MUL and DIV instructions are not affected by the T and D flags.
2. The contents of the processor status register are unaffected by multiply or divide instructions.

APPENDIX 2. Handling of unused pins

Table 8.3.1 Example of unused pins handling of M37450M2-XXXSP/FP

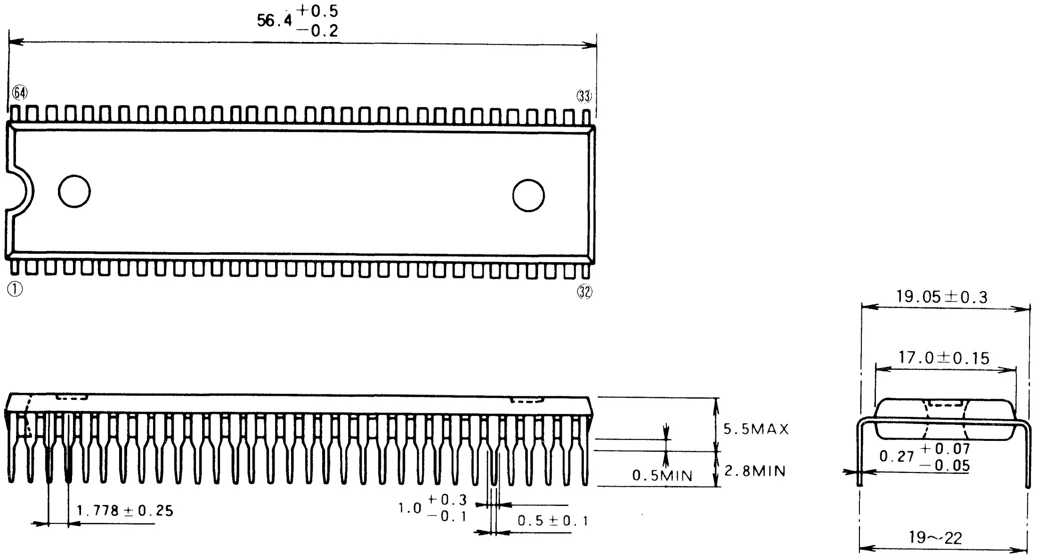
Pin or port	Handling
Ports P0~P3, P5, P6	set input mode and pull-down through resistor.
Port P4	pull-down through resistor.
ϕ, SYNC, D-A$_1$, D-A$_2$, R/$\overline{W}$, $\overline{RD}$ $\overline{WR}$, RESET$_{OUT}$	open
AV$_{CC}$	V$_{CC}$ or V$_{SS}$ level
AV$_{SS}$, ADV$_{REF}$, DAV$_{REF}$, V$_{REF}$	V$_{SS}$ level



APPENDIX 3. Package outlines

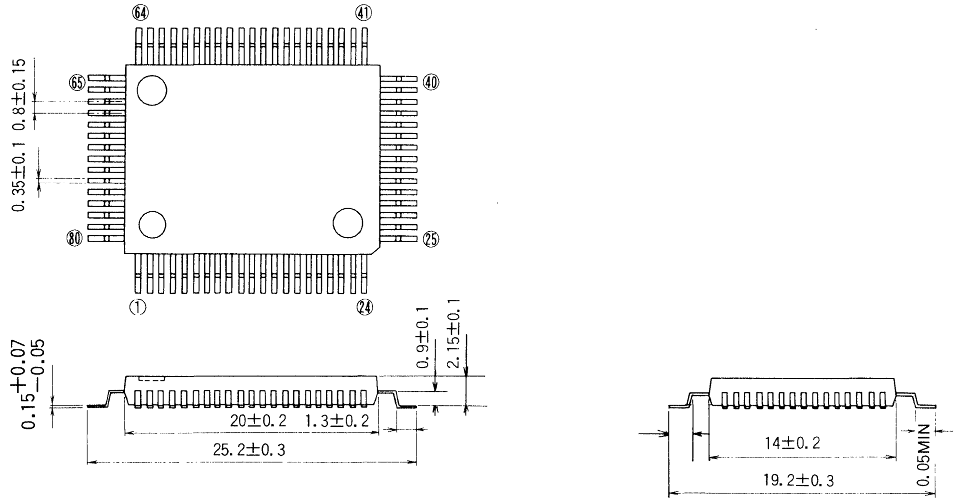
M37450M2-XXXSP package outline

TYPE 64P4B 64-PIN MOLDED PLASTIC DIP(LEAD PITCH 1.778mm) Dimension in mm



M37450M2-XXXFP package outline

TYPE 80P6 80PIN MOLDED PLASTIC FLAT Dimension in mm



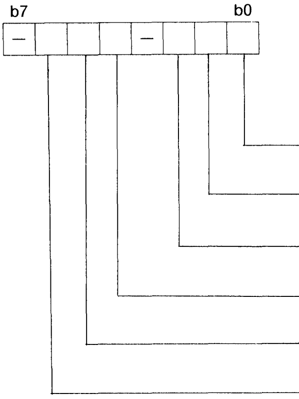
APPENDIX 4. SFR memory map

<table>
<tr><td>$00D0_{16}$</td><td colspan="8">P0 register</td><td>P0</td></tr>
<tr><td>$00D1_{16}$</td><td colspan="8">P0 directional register</td><td>P0D</td></tr>
<tr><td>$00D2_{16}$</td><td colspan="8">P1 register</td><td>P1</td></tr>
<tr><td>$00D3_{16}$</td><td colspan="8">P1 directional register</td><td>P1D</td></tr>
<tr><td>$00D4_{16}$</td><td colspan="8">P2 register</td><td>P2</td></tr>
<tr><td>$00D5_{16}$</td><td colspan="8">P2 directional register</td><td>P2D</td></tr>
<tr><td>$00D6_{16}$</td><td colspan="8">P3 register</td><td>P3</td></tr>
<tr><td>$00D7_{16}$</td><td colspan="8">P3 directional register</td><td>P3D</td></tr>
<tr><td>$00D8_{16}$</td><td colspan="4">——————</td><td colspan="4">P4(64-pin model)</td><td>P4</td></tr>
<tr><td></td><td colspan="8">P4(80-pin model)</td><td></td></tr>
<tr><td>$00D9_{16}$</td><td colspan="8">Reserved</td><td></td></tr>
<tr><td>$00DA_{16}$</td><td colspan="8">P5 register</td><td>P5</td></tr>
<tr><td>$00DB_{16}$</td><td colspan="8">P5 directional register</td><td>P5D</td></tr>
<tr><td>$00DC_{16}$</td><td colspan="8">P6 register</td><td>P6</td></tr>
<tr><td>$00DD_{16}$</td><td colspan="8">P6 directional register</td><td>P6D</td></tr>
<tr><td>$00DE_{16}$</td><td colspan="8">MISRG1</td><td>MISRG1</td></tr>
<tr><td></td><td>—</td><td>C3</td><td>C2</td><td>C1</td><td>—</td><td>I3P</td><td>I2P</td><td>I1P</td><td></td></tr>
<tr><td>$00DF_{16}$</td><td colspan="8">MISRG2</td><td>MISRG2</td></tr>
<tr><td></td><td>SPS</td><td>RDYE</td><td>PWMM</td><td>PWME</td><td>DBBM</td><td>DBBE</td><td>CM1</td><td>CM0</td><td></td></tr>
<tr><td>$00E0_{16}$</td><td colspan="8">D-A1 register</td><td>D-A1</td></tr>
<tr><td>$00E1_{16}$</td><td colspan="8">D-A2 register</td><td>D-A2</td></tr>
</table>

00E2$_{16}$	A-D register								ADSR
00E3$_{16}$	A-D control register								ADCON
	—	—	—	—	—	ANC2	ANC1	ANC0	
00E4$_{16}$	Data bus buffer register								DBB
00E5$_{16}$	Data bus buffer status register								DBBSTS
	U7	U6	U5	U4	A0	U2	IBF	OBF	
00E6$_{16}$	Receive/transmit buffer register								RB/TB
00E7$_{16}$	Serial I/O status register								SIOSTS
	—	SE	FE	PE	OE	TSC	RBF	TBE	
00E8$_{16}$	Serial I/O control register								SIOCON
	SIOE	SIOM	RE	TE	TIC	SRDY	SCS	CSS	
00E9$_{16}$	UART control register								UACON
	—	—	—	—	STPS	PARS	PARE	CHAS	
00EA$_{16}$	Baud rate generator								BRG
00EB$_{16}$	PWM register(low byte)								PWML
00EC$_{16}$	PWM register(high byte)								PWMH
00ED$_{16}$	Timer 1 control register								TMR1
	—	—	EP1	OLL1	S1	TMS12	TMS11	TMS10	
00EE$_{16}$	Timer 2 control register								TMR2
	—	—	EP2	OLL2	S2	TMS22	TMS21	TMS20	
00EF$_{16}$	Timer 3 control register								TMR3
	—	—	EP3	OLL3	S3	TMS32	TMS31	TMS30	

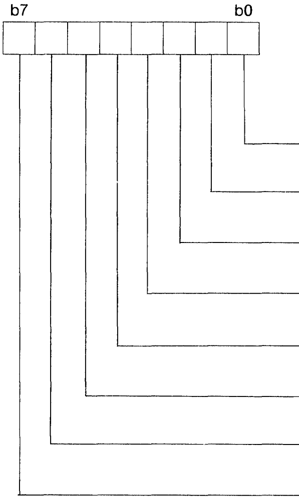
<table>
<tr><td>00F0$_{16}$</td><td colspan="8">Timer 1 register (low-order)</td><td>T1L</td></tr>
<tr><td>00F1$_{16}$</td><td colspan="8">Timer 1 register (high-order)</td><td>T1H</td></tr>
<tr><td>00F2$_{16}$</td><td colspan="8">Timer 1 latch (low-order)</td><td>TL1L</td></tr>
<tr><td>00F3$_{16}$</td><td colspan="8">Timer 1 latch (high-order)</td><td>TL1H</td></tr>
<tr><td>00F4$_{16}$</td><td colspan="8">Timer 2 register (low-order)</td><td>T2L</td></tr>
<tr><td>00F5$_{16}$</td><td colspan="8">Timer 2 register (high-order)</td><td>T2H</td></tr>
<tr><td>00F6$_{16}$</td><td colspan="8">Timer 2 latch (low-order)</td><td>TL2L</td></tr>
<tr><td>00F7$_{16}$</td><td colspan="8">Timer 2 latch (high-order)</td><td>TL2H</td></tr>
<tr><td>00F8$_{16}$</td><td colspan="8">Timer 3 register (low-order)</td><td>T3L</td></tr>
<tr><td>00F9$_{16}$</td><td colspan="8">Timer 3 register (high-order)</td><td>T3H</td></tr>
<tr><td>00FA$_{16}$</td><td colspan="8">Timer 3 latch (low-order)</td><td>TL3L</td></tr>
<tr><td>00FB$_{16}$</td><td colspan="8">Timer 3 latch (high-order)</td><td>TL3H</td></tr>
<tr><td>00FC$_{16}$</td><td colspan="8">Interrupt request register 1</td><td>IRQ1</td></tr>
<tr><td></td><td>T3R</td><td>T2R</td><td>T1R</td><td>I3R</td><td>I2R</td><td>I1R</td><td>OBER</td><td>IBFR</td><td></td></tr>
<tr><td>00FD$_{16}$</td><td colspan="8">Interrupt request register 2</td><td>IRQ2</td></tr>
<tr><td></td><td>—</td><td>—</td><td>ADR</td><td>TIR</td><td>RIR</td><td>E3R</td><td>E2R</td><td>E1R</td><td></td></tr>
<tr><td>00FE$_{16}$</td><td colspan="8">Interrupt control register 1</td><td>ICON1</td></tr>
<tr><td></td><td>T3E</td><td>T2E</td><td>T1E</td><td>I3E</td><td>I2E</td><td>I1E</td><td>OBEE</td><td>IBFE</td><td></td></tr>
<tr><td>00FF$_{16}$</td><td colspan="8">Interrupt control register 2</td><td>ICON2</td></tr>
<tr><td></td><td>—</td><td>—</td><td>ADE</td><td>TIE</td><td>RIE</td><td>E3E</td><td>E2E</td><td>E1E</td><td></td></tr>
</table>

APPENDIX 5. Control registers



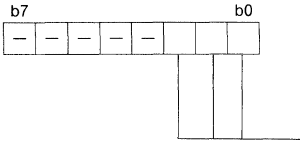
MISRG 1 (address 00DE$_{16}$)

bit	name	function	status at reset
0	INT$_1$ input polarity selection bit	0 : falling edge 1 : rising edge	0
1	INT$_2$ input polarity selection bit	0 : falling edge 1 : rising edge	0
2	INT$_3$ input polarity selection bit	0 : falling edge 1 : rising edge	0
4	Timer 1 count enable bit	0 : count disable 1 : count enable	0
5	Timer 2 count enable bit	0 : count disable 1 : count enable	0
6	Timer 3 count enable bit	0 : count disable 1 : count enable	0



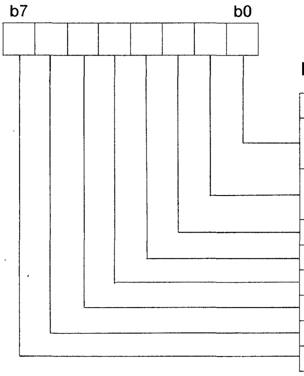
MISRG 2 (address 00DF$_{16}$)

bit	name	function	status at reset
0	Processor mode bit	b1 b0 0 0 : single-chip mode 0 1 : memory expanding mode 1 0 : microprocessor mode 1 1 : disable	0
1			0
2	Bus interface enable bit	0 : disable 1 : enable	0
3	Bus interface mode bit	0 : $\overline{RD}$, $\overline{WR}$ bus 1 : R/$\overline{W}$ bus	0
4	PWM enable bit	0 : disable 1 : enable	0
5	PWM mode selection bit	0 : 8-bit high speed PWM 1 : 16-bit high precision PWM	0
6	Bus cycle control bit	0 : normal bus cycle 1 : normal bus cycle×2	0
7	Stack page selection bit	0 : in 0 page area 1 : in 1 page area	0



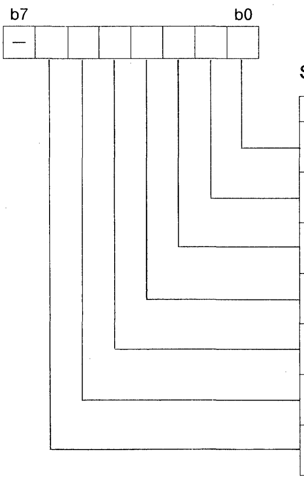
A-D control register (address 00E3$_{16}$)

bit	name	function	status at reset
2,1,0	Analog input pin selection bit	b2 b1 b0 0 0 0 : input from P4$_0$ pin 0 0 1 : input from P4$_1$ pin 0 1 0 : input from P4$_2$ pin 0 1 1 : input from P4$_3$ pin 1 0 0 : input from P4$_4$ pin 1 0 1 : input from P4$_5$ pin 1 1 0 : input from P4$_6$ pin 1 1 1 : input from P4$_7$ pin	U



Data bus buffer status register (address $00E5_{16}$)

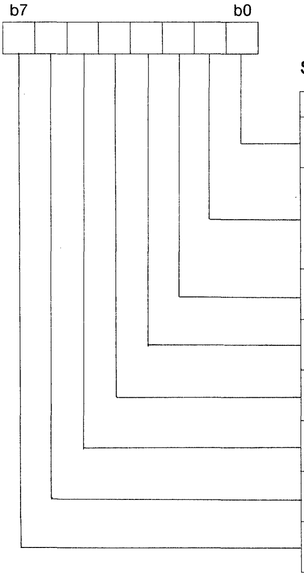
bit	name	function	status at reset
0	Output buffer full flag	0 : buffer empty 1 : buffer full	1
1	Input buffer full flag	0 : buffer empty 1 : buffer full	0
2	User definable flag	User may freely define this flag	U
3	A0 flag	Indicates the A0 status when the IBF flag is set.	0
4	User definable flag	User may freely define this flag	U
5	User definable flag	User may freely define this flag	U
6	User definable flag	User may freely define this flag	U
7	User definable flag	User may freely define this flag	U



Serial I/O status register (address $00E7_{16}$)

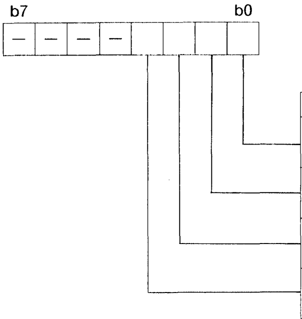
bit	name	function	status at reset
0	Transmit buffer empty flag	0 : buffer full 1 : buffer empty	0
1	Receive buffer full flag	0 : buffer empty 1 : buffer full	0
2	Transmit shift register shift completion flag	0 : busy shifting 1 : shift complete	0
3	Overrun error (OE) flag	0 : no 1 : yes	0
4	Parity error (PE) flag	0 : no 1 : yes	0
5	Framing error (FE) flag	0 : no 1 : yes	0
6	Summing error (SE) flag	0 : PE U OE U FE=0 1 : PE U OE U FE=1	0

Note : When the transmit enable bit of the serial I/O control register becomes "1" (enable), this register will be set to "05_{16}". Bits 4~6 flags are valid only in the UART mode.



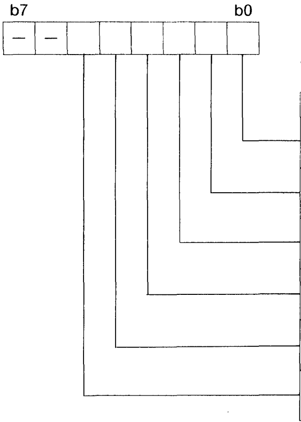
Serial I/O control register (address $00E8_{16}$)

bit	name	function	status at reset
0	BRG count source selection bit	0 : $f(X_{IN})/2$ 1 : $f(X_{IN})/8$	0
1	Serial I/O clock synchronous selection bit	0 : BRG output divided by 4 (when clock synchronous serial I/O is selected) or 16 (when UART mode is selected) 1 : external clock	0
2	$\overline{S_{RDY}}$ output enable bit	0 : $P3_7$ operates as normal I/O pin 1 : $P3_7$ change to $\overline{S_{RDY}}$ output pin	0
3	Transmit interrupt source selection bit	0 : when transmit buffer becomes empty 1 : when transmit shift operation is complete	0
4	Transmit enable bit	0 : transmit disable 1 : transmit enable	0
5	Receive enable bit	0 : transmit disable 1 : transmit enable	0
6	Serial I/O mode selection bit	0 : UART 1 : clock synchronous serial I/O	0
7	Serial I/O enable bit	0 : serial I/O disable 1 : serial I/O enable	0



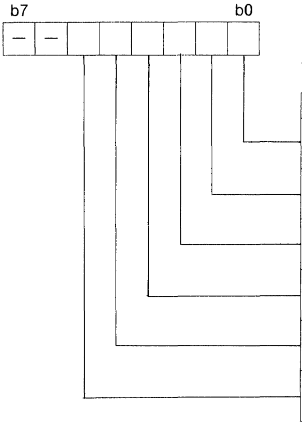
UART control register (address $00E9_{16}$)

bit	name	function	status at reset
0	Character length selection bit	0 : 8-bit 1 : 7-bit	0
1	Parity enable bit	0 : parity disable 1 : parity enable	0
2	Parity selection bit	0 : even parity 1 : odd parity	0
3	Stop bit length selection bit	0 : 1 stop bit 1 : 2 stop bit	0



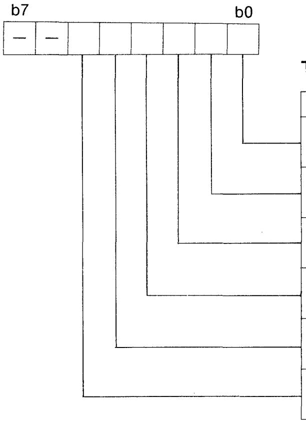
Timer 1 control register (address $00ED_{16}$)

bit	name	function	status at reset
0	Timer 1 mode selection bit	b2 b0 0 0 0 : 16-bit timer mode 0 0 1 : event counter mode 0 1 0 : pulse output mode 0 1 1 : pulse period measurement mode 1 0 0 : pulse width measurement mode 1 0 1 : programmable waveform generation mode 1 1 0 : programmable one-shot generation mode	0
1			0
2			0
3	Timer 1 count source selection bit	0 : $f(X_{IN})/4$ 1 : INT_1 pin input	0
4	Output level latch		0
5	Event polarity selection bit	0 : falling edge 1 : rising edge	0



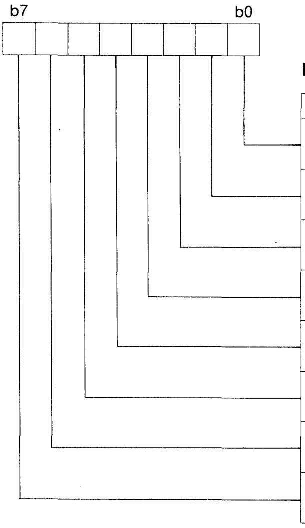
Timer 2 control register (address $00EE_{16}$)

bit	name	function	status at reset
0	Timer 2 mode selection bit	b2 b0 0 0 0 : 16-bit timer mode 0 0 1 : event counter mode 0 1 0 : pulse output mode 0 1 1 : pulse period measurement mode 1 0 0 : pulse width measurement mode 1 0 1 : programmable waveform generation mode 1 1 0 : programmable one-shot generation mode	0
1			0
2			0
3	Timer 2 count source selection bit	0 : $f(X_{IN})/4$ 1 : INT_2 pin input	0
4	Output level latch		0
5	Event polarity selection bit	0 : falling edge 1 : rising edge	0



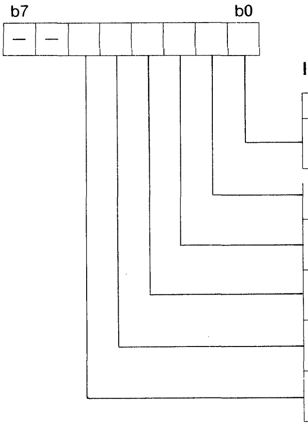
Timer 3 control register (address $00EF_{16}$)

bit	name	function	status at reset
0	Timer 3 mode selection bit	b2 b1 b0 0 0 0 : 16-bit timer mode 0 0 1 : event counter mode 0 1 0 : pulse output mode 0 1 1 : pulse period measurement mode 1 0 0 : pulse width measurement mode 1 0 1 : programmable waveform generation mode 1 1 0 : programmable one-shot generation mode	0
1			0
2			0
3	Timer 3 count source selection bit	0 : $f(X_{IN})/4$ 1 : INT_3 pin input	0
4	Output level latch		0
5	Event polarity selection bit	0 : falling edge 1 : rising edge	0



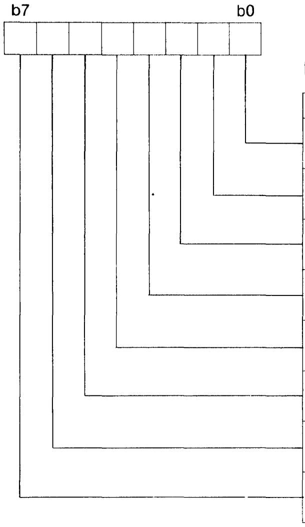
Interrupt request register 1 (address $00FC_{16}$)

bit	name	function	status at reset
0	Input buffer full interrupt request bit	0 : no request 1 : request occured	0
1	Output buffer empty interrupt request bit	0 : no request 1 : request occured	0
2	INT_1 interrupt request bit	0 : no request 1 : request occured	0
3	INT_2 interrupt request bit	0 : no request 1 : request occured	0
4	INT_3 interrupt request bit	0 : no request 1 : request occured	0
5	Timer 1 interrupt request bit	0 : no request 1 : request occured	0
6	Timer 2 interrupt request bit	0 : no request 1 : request occured	0
7	Timer 3 interrupt request bit	0 : no request 1 : request occured	0



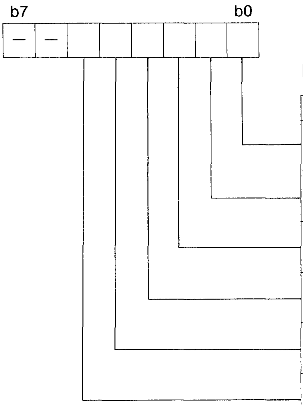
Interrupt request register 2 (address $00FD_{16}$)

bit	name	function	status at reset
0	EV_1 interrupt request bit	0 : no request 1 : request occured	0
1	EV_2 interrupt request bit	0 : no request 1 : request occured	0
2	EV_3 interrupt request bit	0 : no request 1 : request occured	0
3	Serial I/O receive interrupt request bit	0 : no request 1 : request occured	0
4	Serial I/O transmit interrupt request bit	0 : no request 1 : request occured	0
5	A-D interrupt request bit	0 : no request 1 : request occured	0



Interrupt control register 1 (address $00FE_{16}$)

bit	name	function	status at reset
0	Input buffer full interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
1	Output buffer empty interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
2	INT_1 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
3	INT_2 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
4	INT_3 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
5	Timer 1 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
6	Timer 2 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
7	Timer 3 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0



Interrupt control register 2 (address $00FF_{16}$)

bit	name	function	status at reset
0	EV_1 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
1	EV_2 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
2	EV_3 interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
3	Serial I/O receive interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
4	Serial I/O transmit interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0
5	A-D coversion completion interrupt enable bit	0 : interrupt disable 1 : interrupt enable	0

APPENDIX 6. Instruction cycles in each addressing mode

Series MELPS 740 microcomputers use the internal clock ϕ which is $f(X_{IN})$ divided by four as the system clock.

The SYNC signal is output when the contents of the PC indicates the location of the next op code. During the next half-cycle of ϕ, the op code is fetched and stored in the op code register. The instruction decoder of the CPU decodes this op code and determines the following sequence.

This instruction timings of all addressing modes are described on the following pages. In these figures, ADDR and DATA are internal signals of the single-chip microcomputer.

Therefore, these signals can be investigated only in the microprocessor mode.

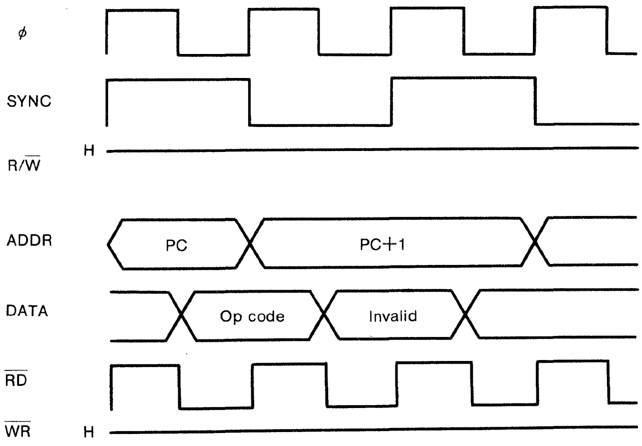
IMPLIED

Instructions : CLC△
CLD△
CLI△
CLT△
CLV△
DEX△
DEY△
INX△
INY△
NOP△
SEC△
SED△
SEI△
SET△
TAX△
TAY△
TSX△
TXA△
TXS△
TYA△

Byte length : 1

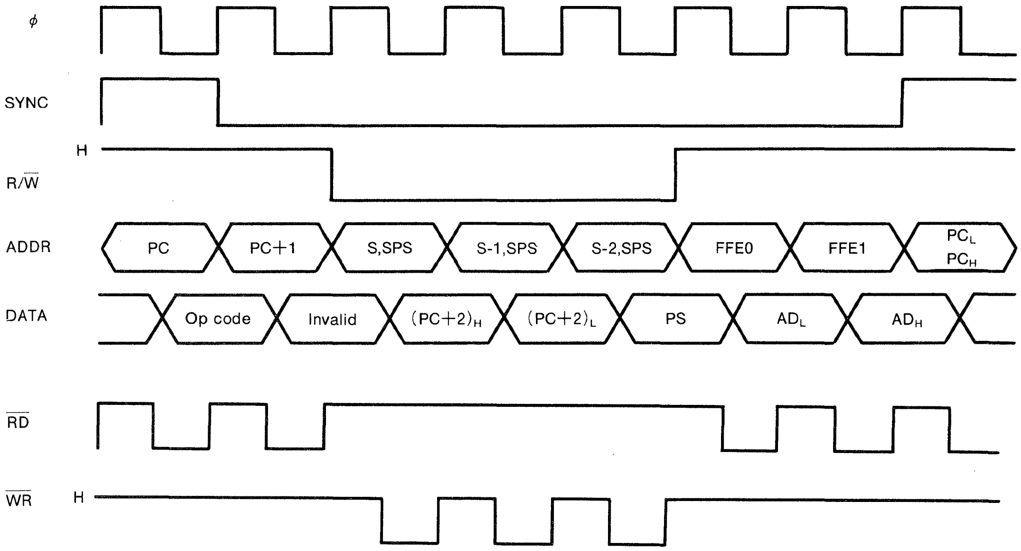
Cycle number : 2

Timing :



IMPLIED

Instructions	: BRK△
Byte length	: 1
Cycle number	: 7
Timing	:



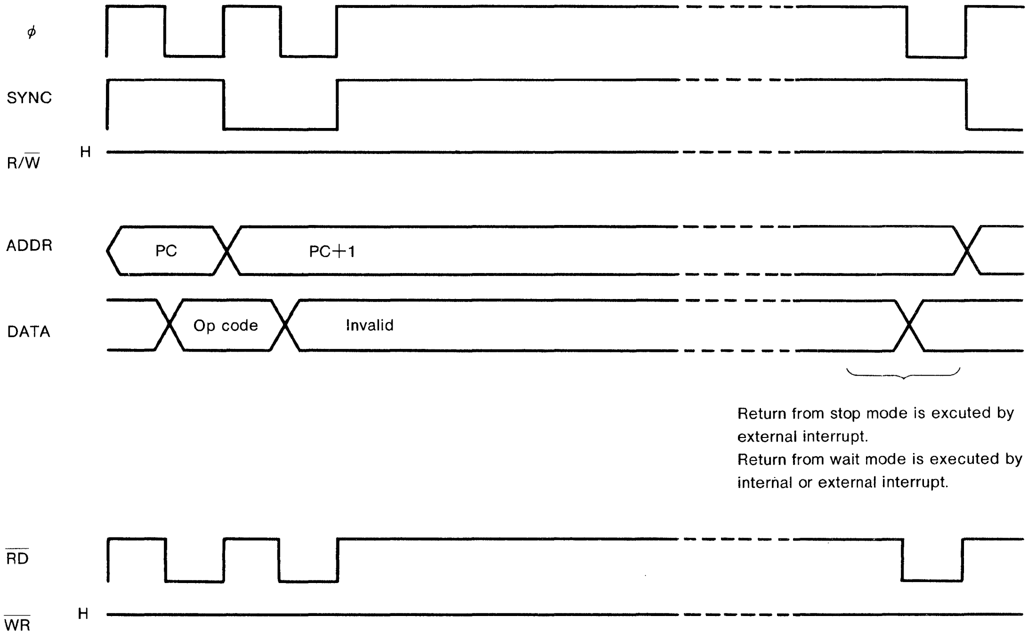
IMPLIED

Instructions : STP△
WIT△

Byte length : 1

Cycle number :

Timing :



Return from stop mode is excuted by external interrupt.
Return from wait mode is executed by internal or external interrupt.

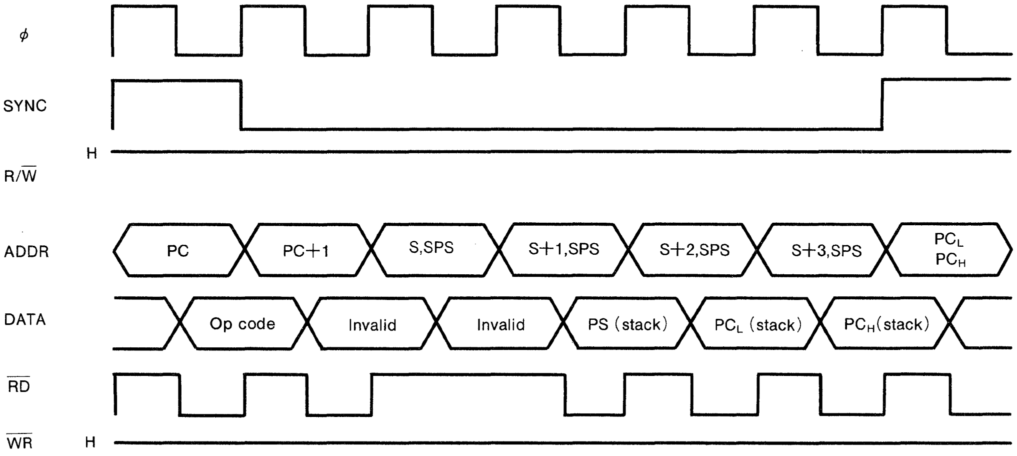
IMPLIED

Instructions : RTI△

Byte length : 1

Cycle number : 6

Timing :

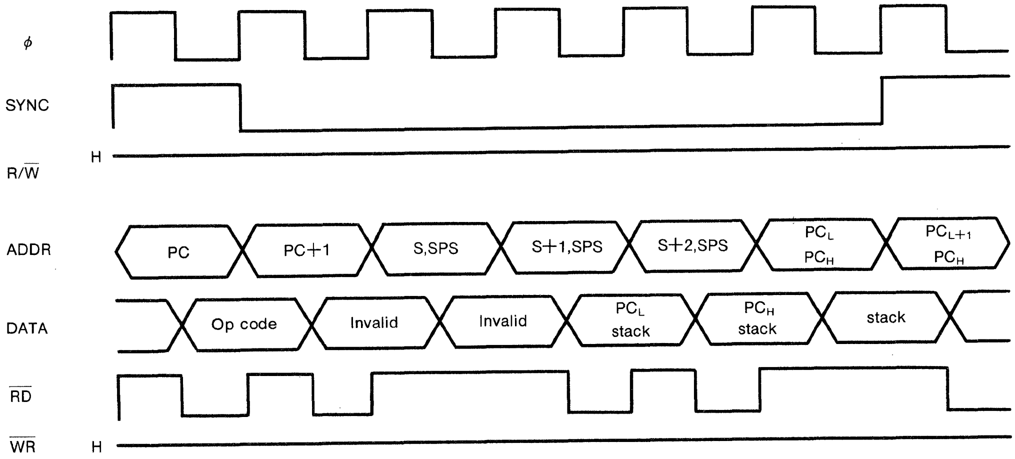


Instructions : RTS△

Byte length : 1

Cycle number : 6

Timing :



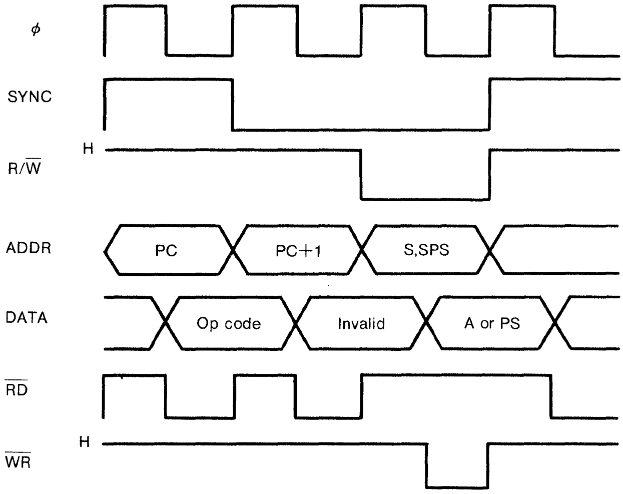
IMPLIED

Instructions : PHA△
PHP△

Byte length : 1

Cycle number : 3

Timing :

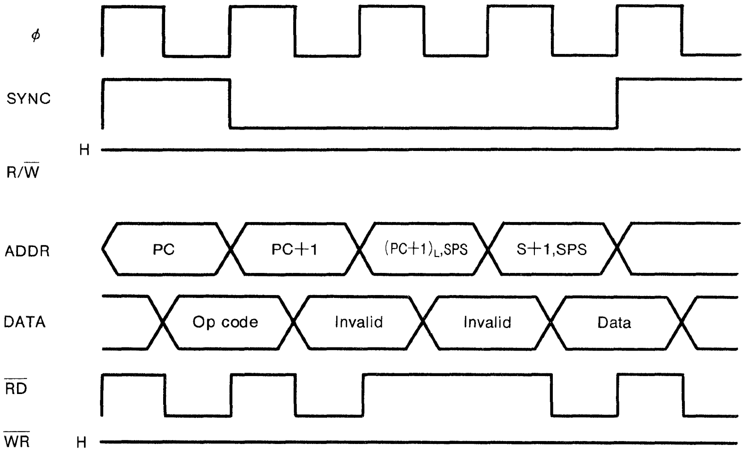


Instructions : PLA△
PLP△

Byte length : 1

Cycle number : 4

Timing :



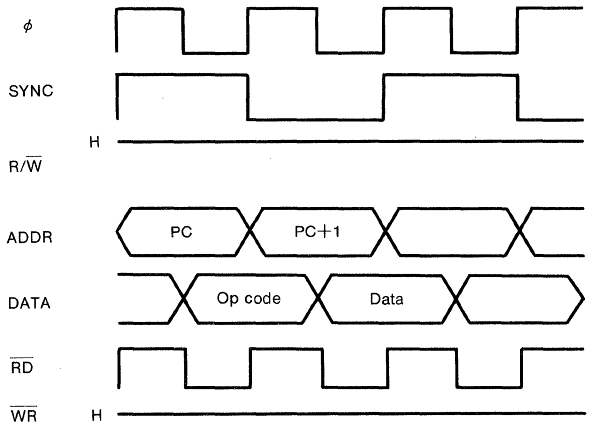
IMMEDIATE

Instructions : ADC△#$nn (T=0)
AND△#$nn (T=0)
CMP△#$nn (T=0)
CPX△#$nn
CPY△#$nn
EOR△#$nn (T=0)
LDA△#$nn (T=0)
LDX△#$nn
LDY△#$nn
ORA△#$nn (T=0)
SBC△#$nn (T=0)

Byte length : 2

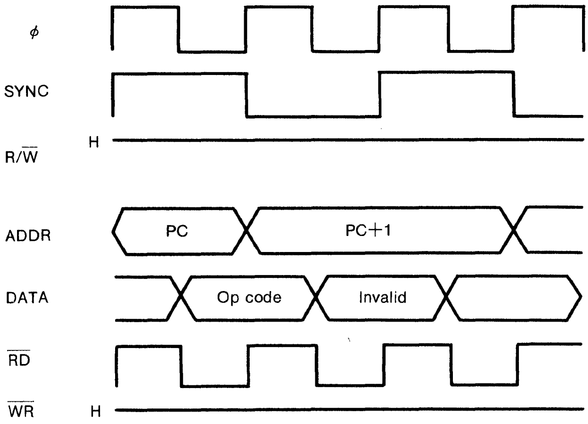
Cycle number : 2

Timing :



ACCUMULATOR

Instructions	: ASL△A DEC△A INC△A LSR△A ROL△A ROR△A
Byte length	: 1
Cycle number	: 2
Timing	:



ACCUMULATOR BIT RELATIVE

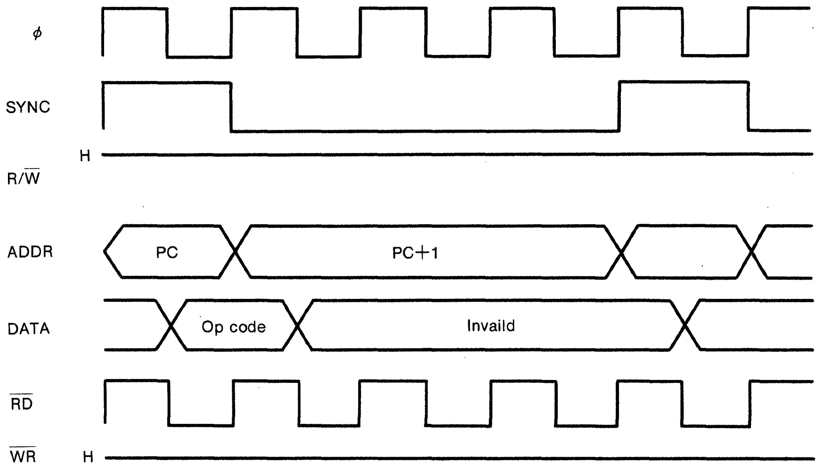
Instructions : BBC△i, A, $hhll
BBS△i, A, $hhll

Byte length : 2

①With no branch

Cycle number : 4

Timing :



ACCUMULATOR BIT RELATIVE

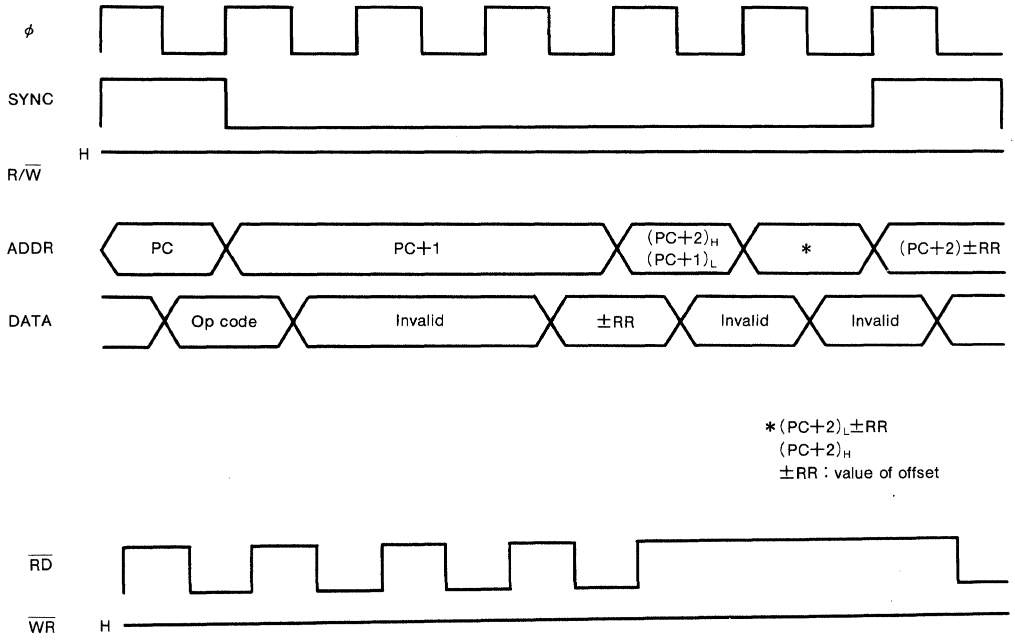
Instructions : BBC△i, A, $hhll
BBS△i, A, $hhll

Byte length : 2

②With branch

Cycle number : 6

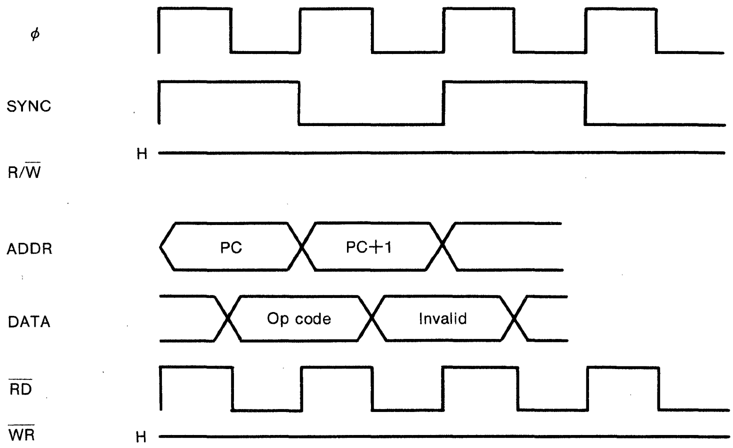
Timing :



*(PC+2)$_L$±RR
(PC+2)$_H$
±RR : value of offset

ACCUMULATOR BIT

Instructions	: CLB△i, A SEB△i, A
Byte length	: 1
Cycle number	: 2
Timing	:



BIT RELATIVE

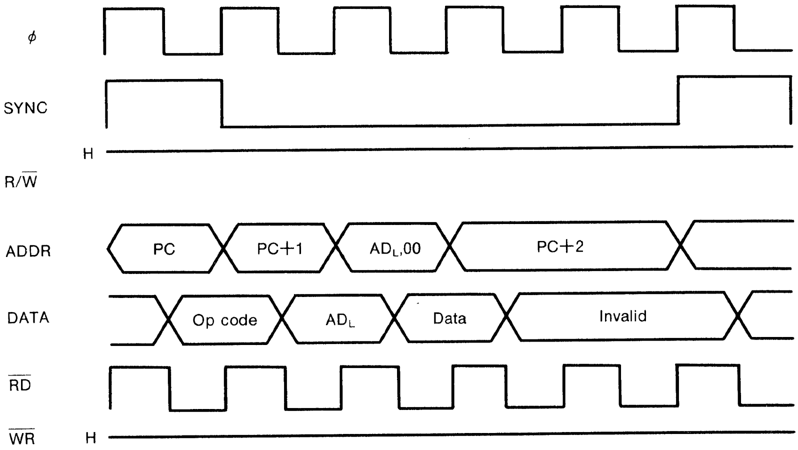
Instructions : BBC△i, $zz, $hhll
BBS△i, $zz, $hhll

Byte length : 3

①With no branch

Cycle number : 5

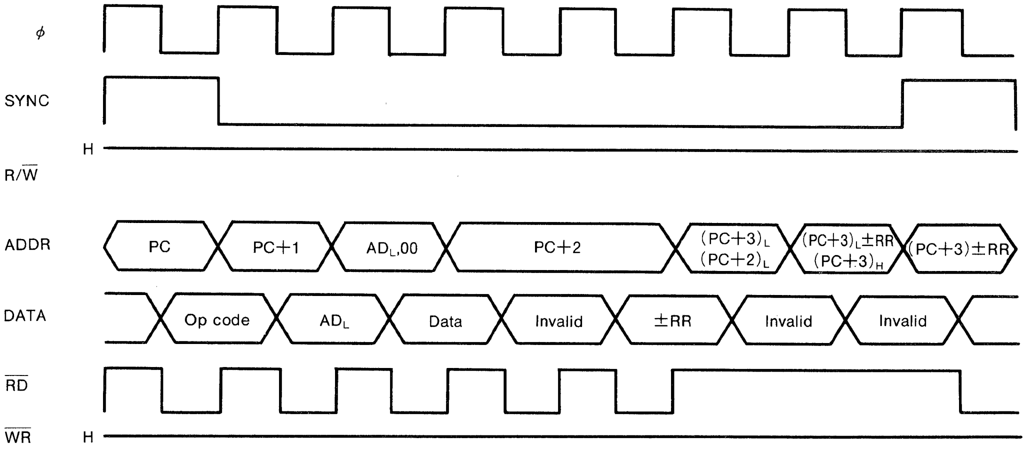
Timing :



②With branch

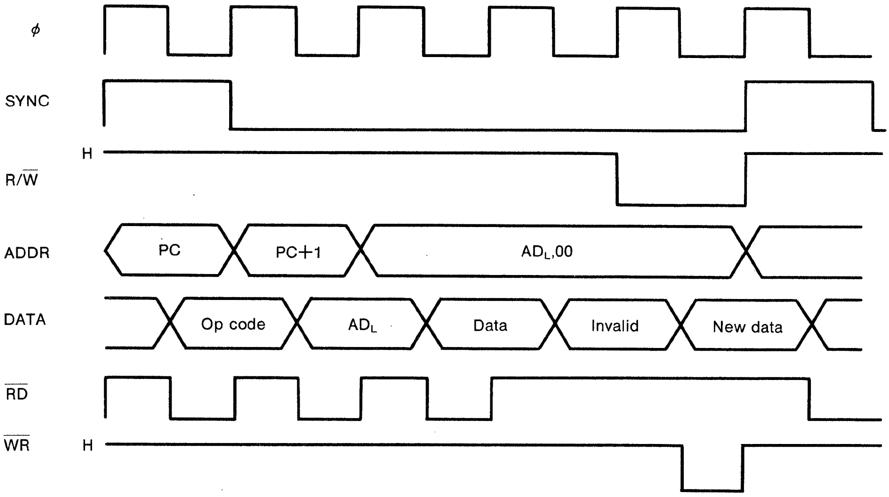
Cycle number : 7

Timing :



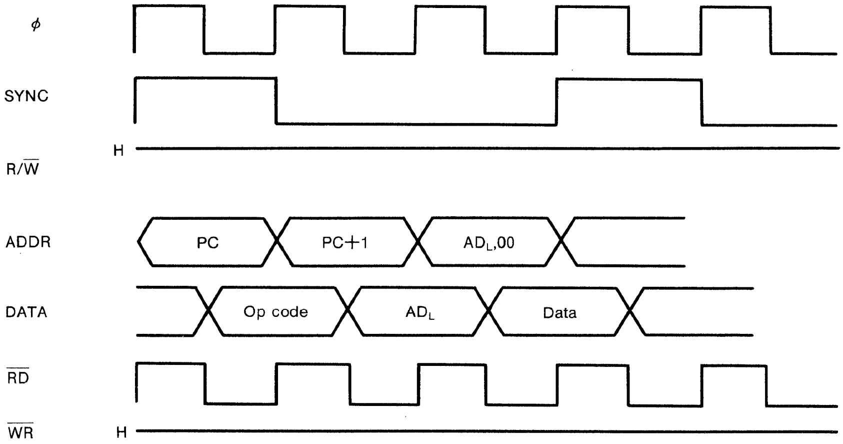
ZERO PAGE BIT

Instructions	: CLB△i, $zz SEB△i, $zz
Byte length	: 2
Cycle number	: 5
Timing	:



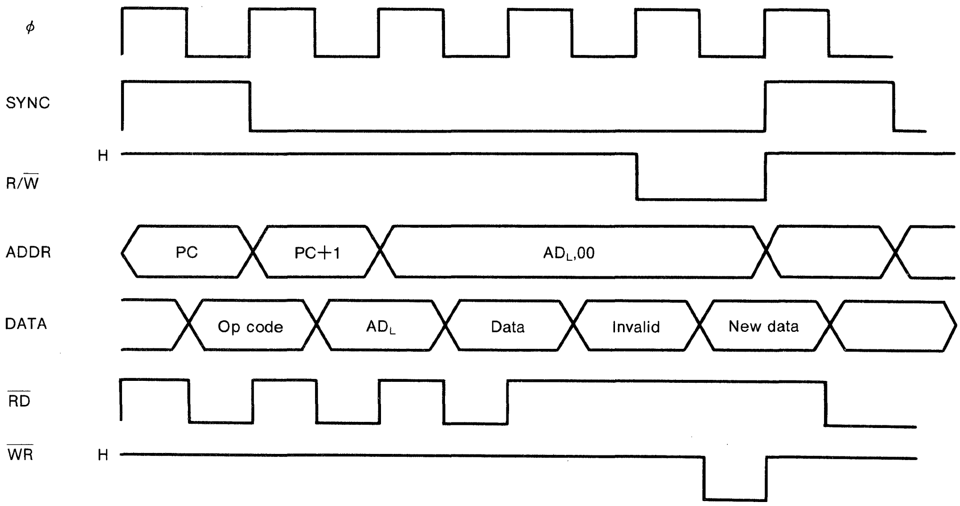
ZERO PAGE

Instructions	: ADC△$zz (T=0) AND△$zz (T=0) BIT△$zz CMP△$zz (T=0) CPX△$zz CPY△$zz EOR△$zz (T=0) LDA△$zz (T=0) LDX△$zz LDY△$zz ORA△$zz SBC△$zz (T=0) TST△$zz (T=0)
Byte length	: 2
Cycle number	: 3
Timing	:



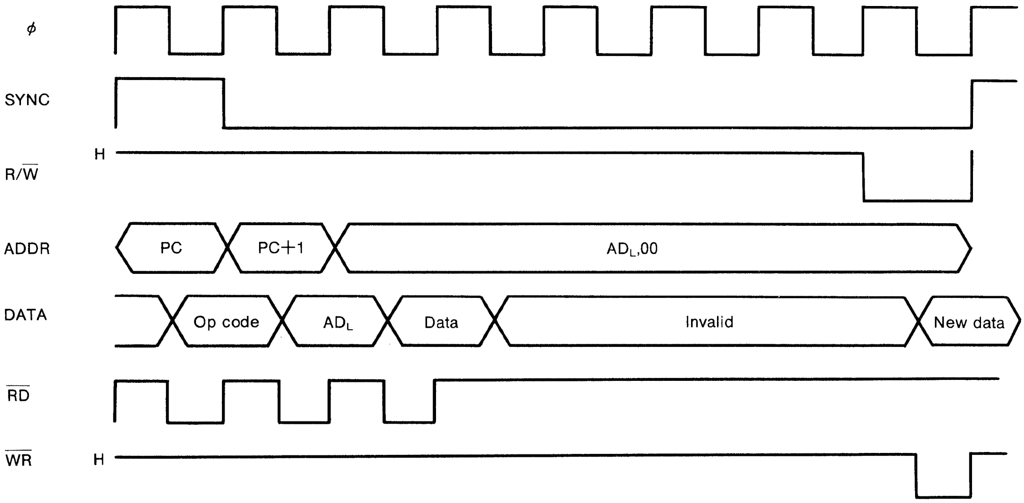
ZERO PAGE

Instructions	: ASL△$zz COM△$zz DEC△$zz INC△$zz LSR△$zz ROL△$zz ROR△$zz
Byte length	: 2
Cycle number	: 5
Timing	:



ZERO PAGE

Instructions	: RRF△$zz
Byte length	: 2
Cycle number	: 8
Timing	:



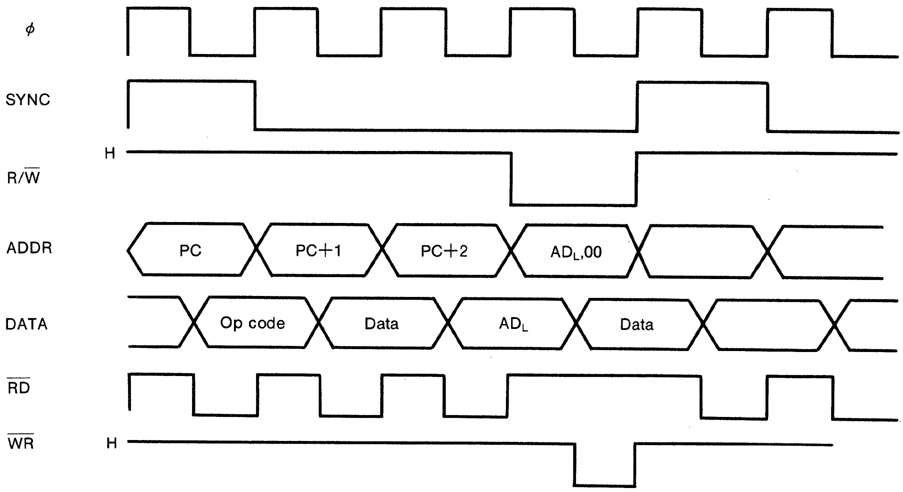
ZERO PAGE

Instructions : LDM△#$nn, $zz

Byte length : 3

Cycle number : 4

Timing :

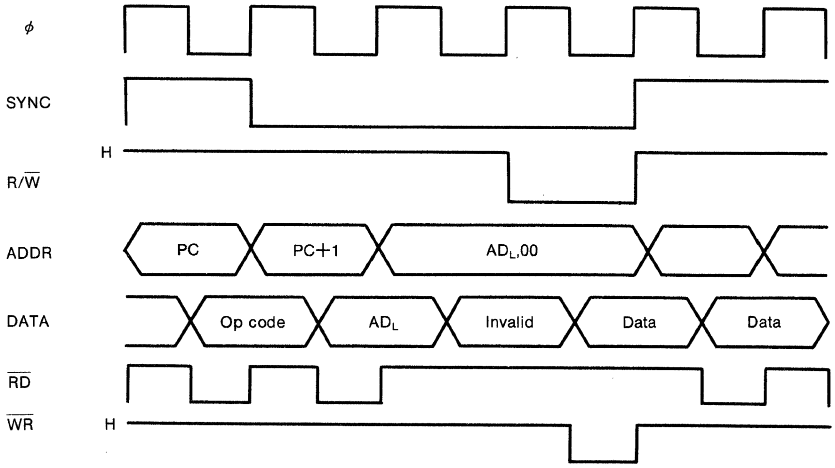


Instructions : STA△$zz
STX△$zz
STY△$zz

Byte length : 2

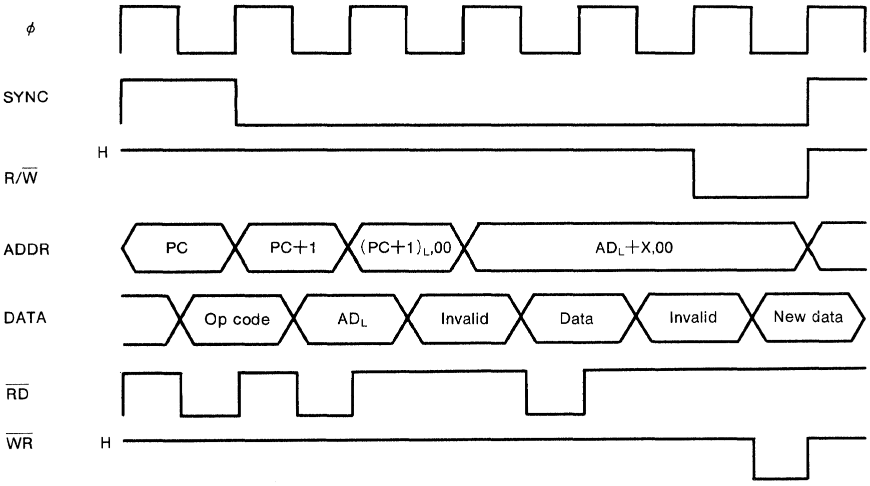
Cycle number : 4

Timing :



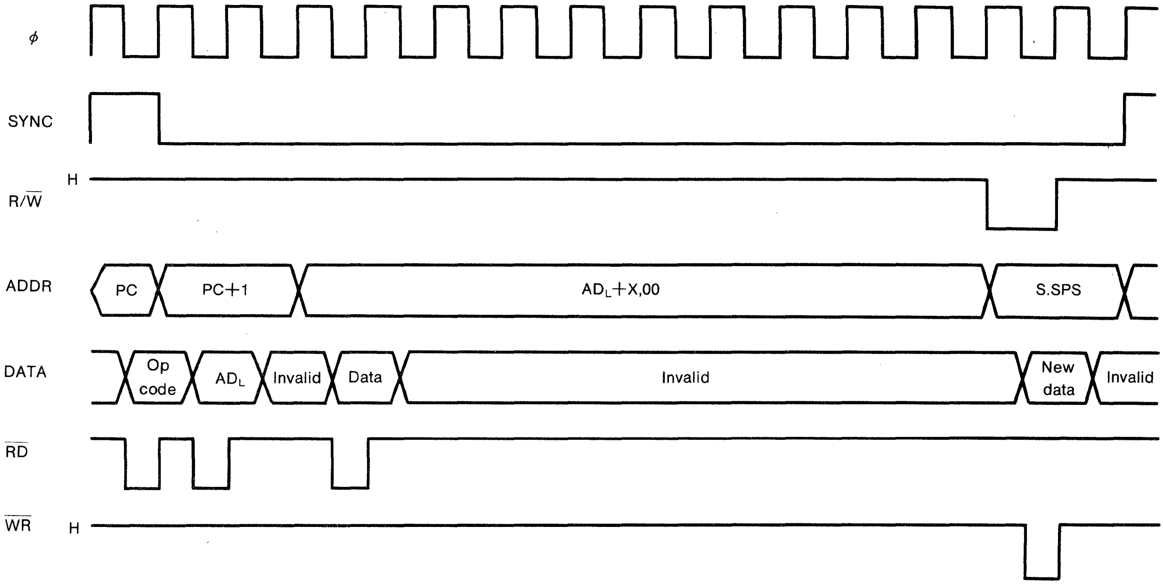
ZERO PAGE X

Instructions	: ASL△$zz, X DEC△$zz, X INC△$zz, X LSR△$zz, X ROL△$zz, X ROR△$zz, X
Byte length	: 2
Cycle number	: 6
Timing	:



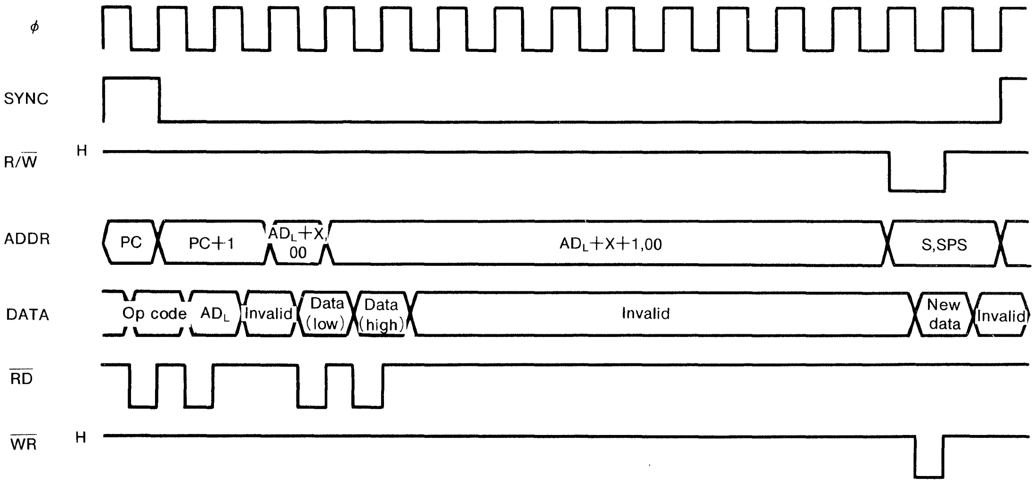
ZERO PAGE X

Instructions	: MUL△$zz, X
Byte length	: 2
Cycle number	: 15
Timing	:



ZERO PAGE X

Instructions	: DIV△$zz, X
Byte length	: 2
Cycle number	: 16
Timing	:



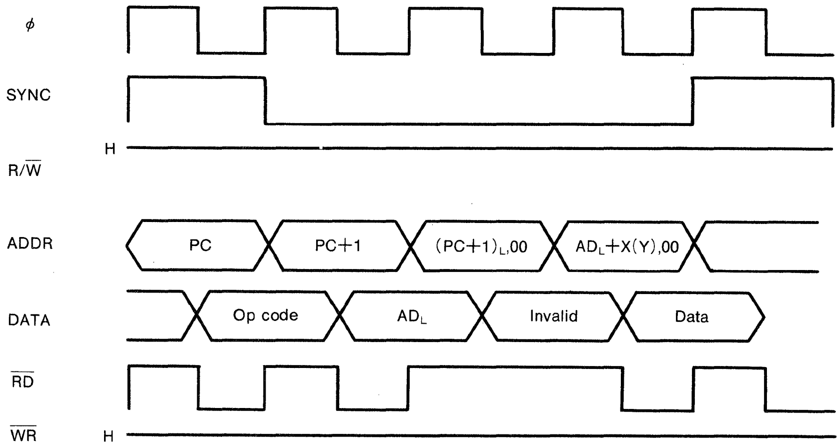
ZERO PAGE X
ZERO PAGE Y

Instructions : ADC△$zz,X (T=0)
AND△$zz, X (T=0)
CMP△$zz, X (T=0)
EOR△$zz, X (T=0)
LDA△$zz, X (T=0)
LDX△$zz, Y
LDY△$zz, X
ORA△$zz, X (T=0)
SBC△$zz, X (T=0)

Byte length : 2

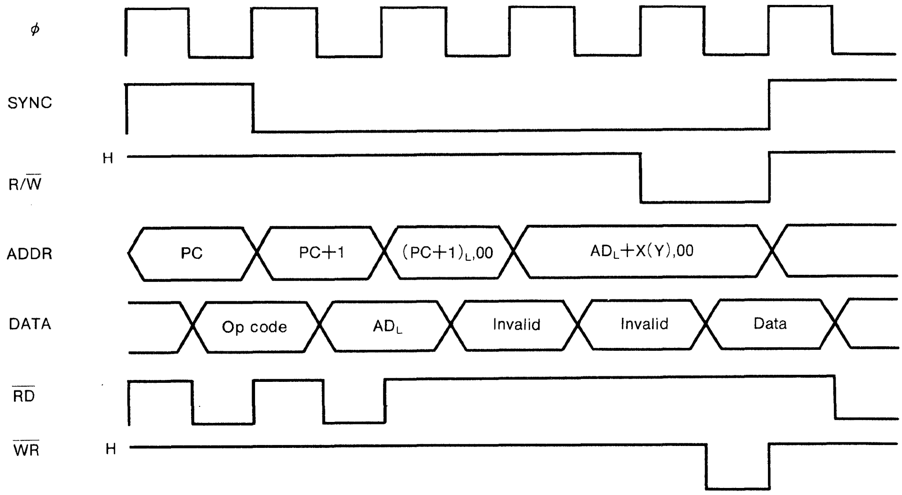
Cycle number : 4

Timing :



ZERO PAGE X
ZERO PAGE Y

Instructions	: STA△$zz, X STX△$zz, Y STY△$zz, X
Byte length	: 2
Cycle number	: 5
Timing	:



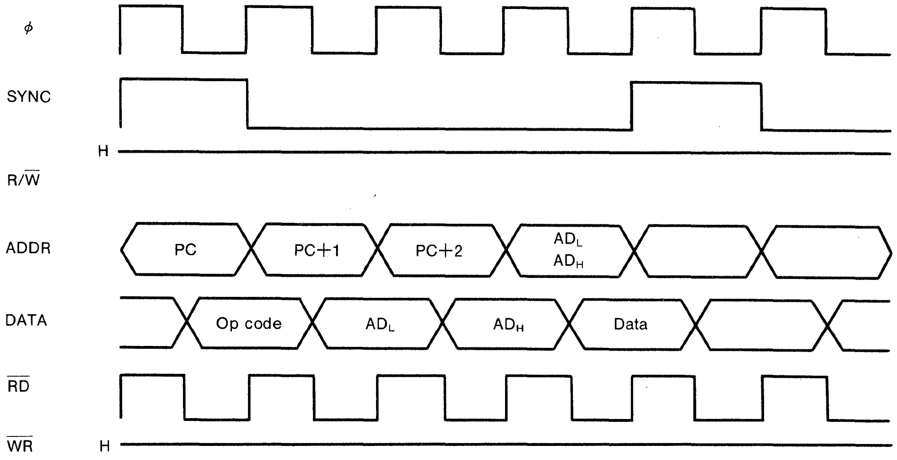
ABSOLUTE

Instructions : ADC△$hhll (T=0)
AND△$hhll (T=0)
BIT△$hhll
CMP△$hhll (T=0)
CPX△$hhll
CPY△$hhll
EOR△$hhll (T=0)
LDA△$hhll (T=0)
LDX△$hhll
LDY△$hhll
ORA△$hhll (T=0)
SBC△$hhll (T=0)

Byte length : 3

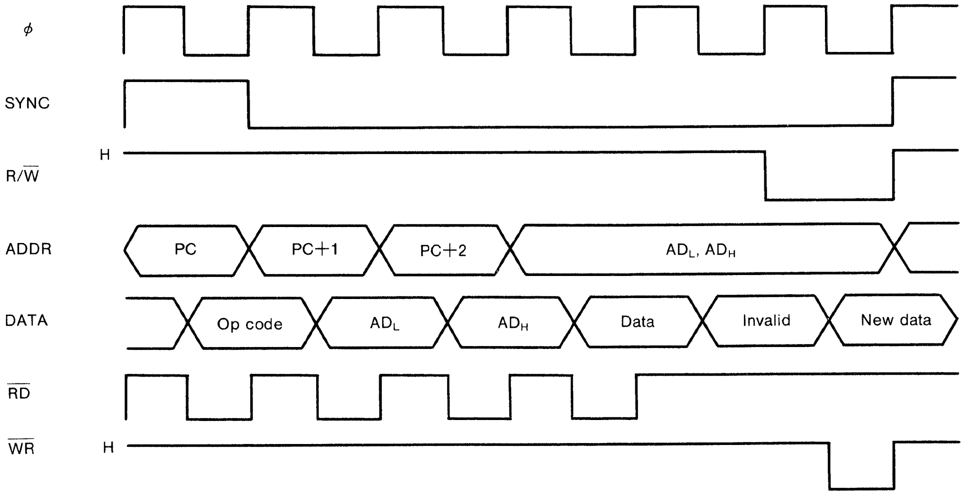
Cycle number : 4

Timing :



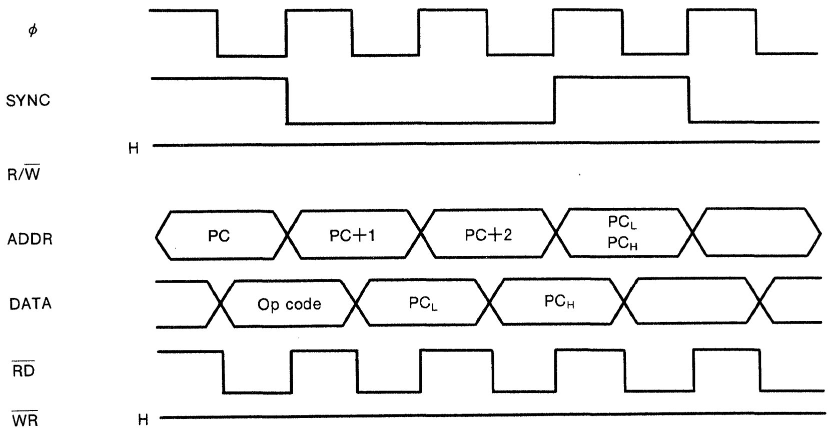
ABSOLUTE

Instructions	: ASL△$hhll DEC△$hhll INC△$hhll LSR△$hhll ROL△$hhll ROR△$hhll
Byte length	: 3
Cycle number	: 6
Timing	:

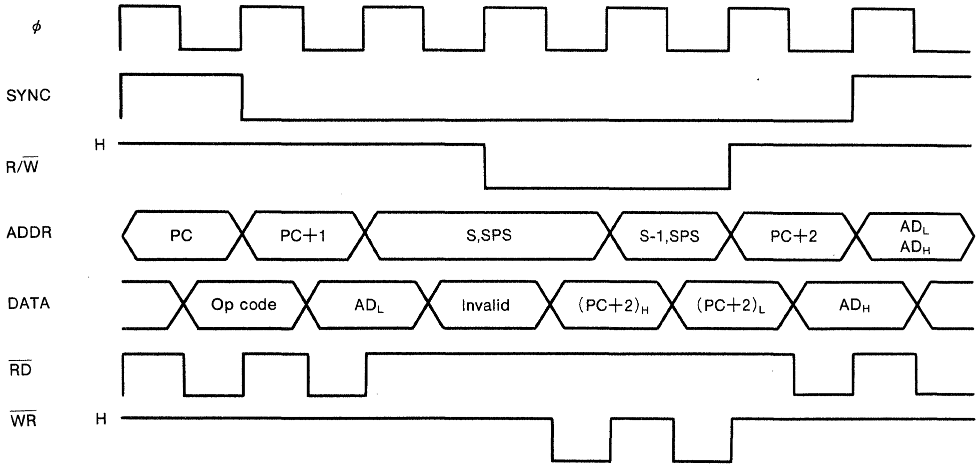


ABSOLUTE

Instructions	: JMP△$hhll
Byte length	: 3
Cycle number	: 3
Timing	:

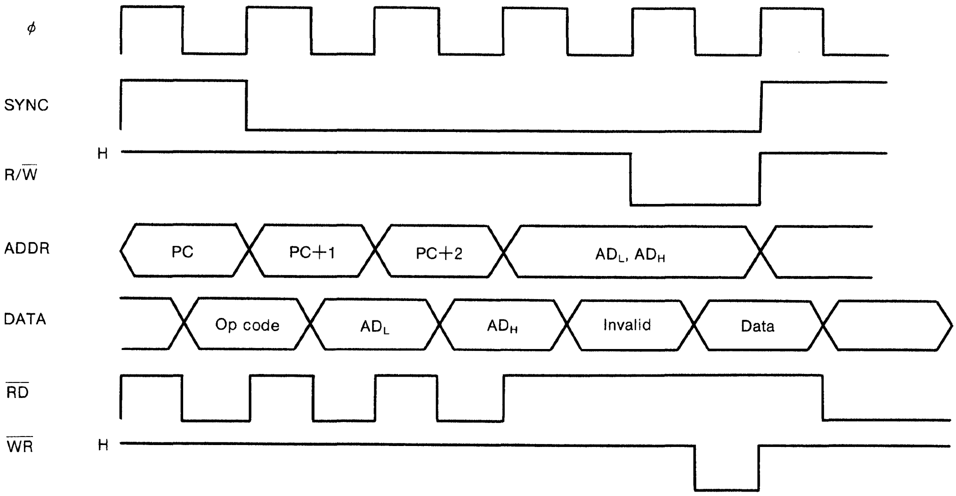


Instructions	: JSR△$hhll
Byte length	: 3
Cycle number	: 6
Timing	:



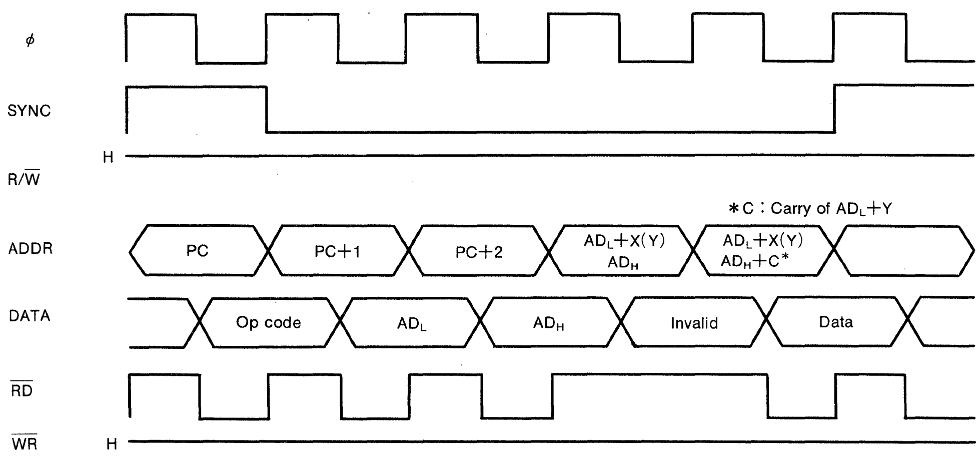
ABSOLUTE

Instructions	: STA△$hhll STX△$hhll STY△$hhll
Byte length	: 3
Cycle number	: 5
Timing	:



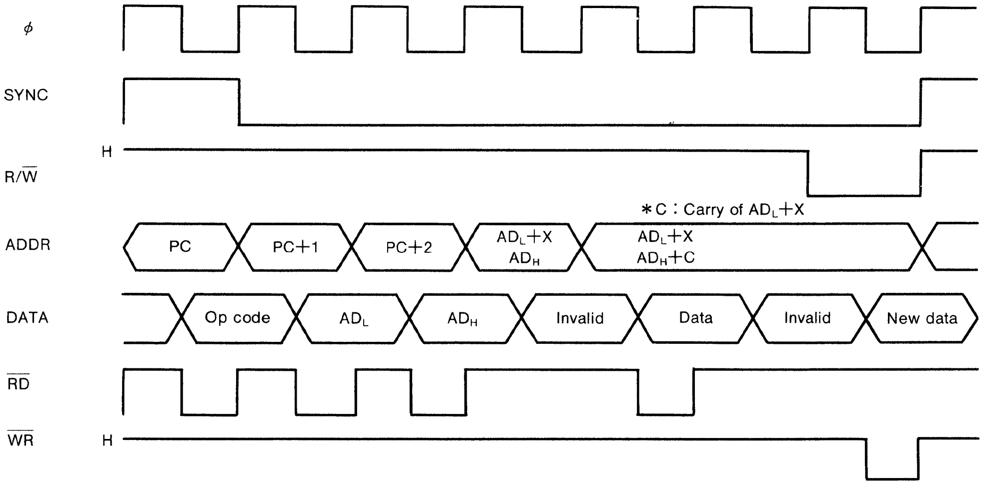
ABSOLUTE X
ABSOLUTE Y

Instructions	: ADC△$hhll, X or Y (T=0) AND△$hhll, X or Y (T=0) CMP△$hhll, X or Y (T=0) EOR△$hhll, X or Y (T=0) LDA△$hhll, X or Y (T=0) LDX△$hhll, Y LDY△$hhll, Y ORA△$hhll, X or Y (T=0) SBC△$hhll, X or Y (T=0)
Byte length	: 3
Cycle number	: 5
Timing	:



ABSOLUTE X

Instructions	: ASL△$hhll, X DEC△$hhll, X INC△$hhll, X LSR△$hhll, X ROL△$hhll, X ROR△$hhll, X
Byte length	: 3
Cycle number	: 7
Timing	:



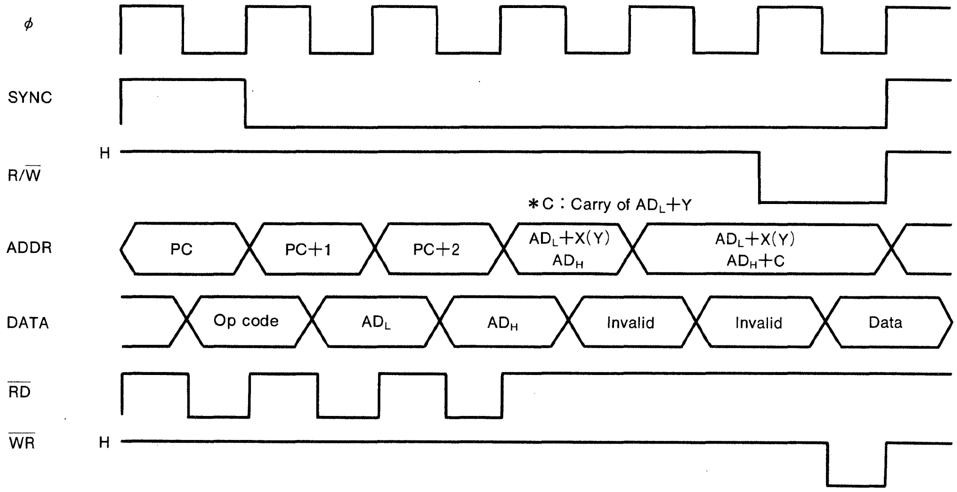
ABSOLUTE X
ABSOLUTE Y

Instructions : STA△$hhll, X or Y

Byte length : 3

Cycle number : 6

Timing :



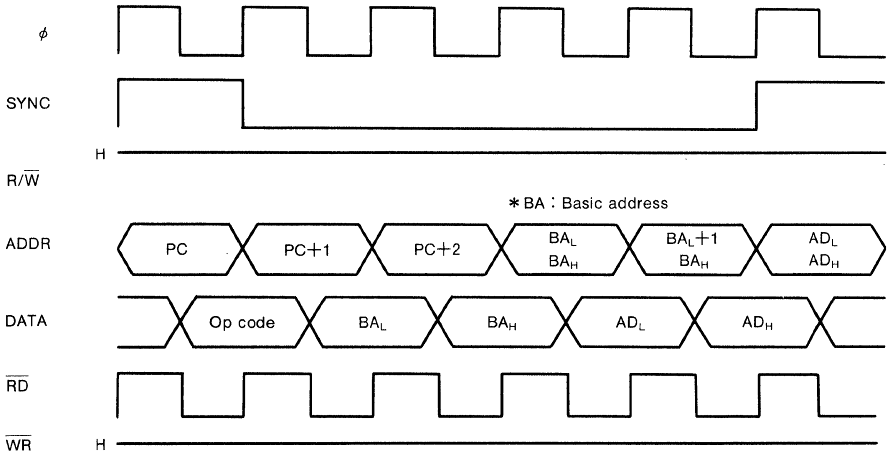
INDIRECT

Instructions : JMP△ ($hhll)

Byte length : 3

Cycle number : 5

Timing :



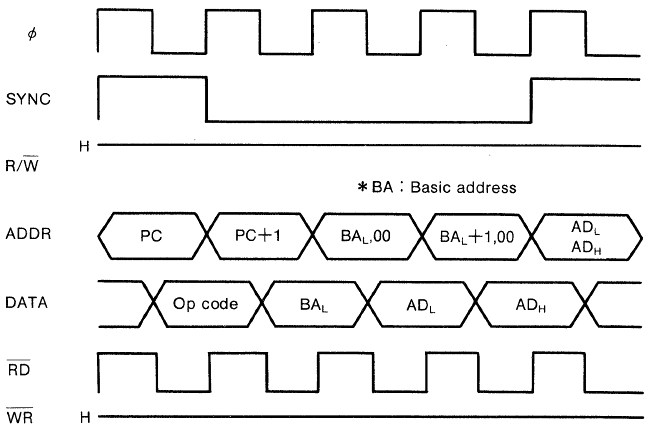
ZERO PAGE INDIRECT

Instructions : JMP△ ($zz)

Byte length : 2

Cycle number : 4

Timing :

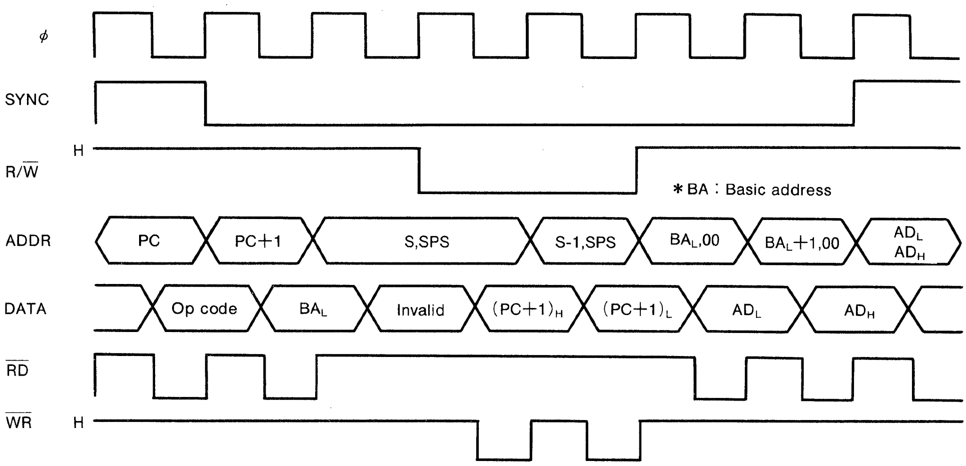


Instructions : JSR△ ($zz)

Byte length : 2

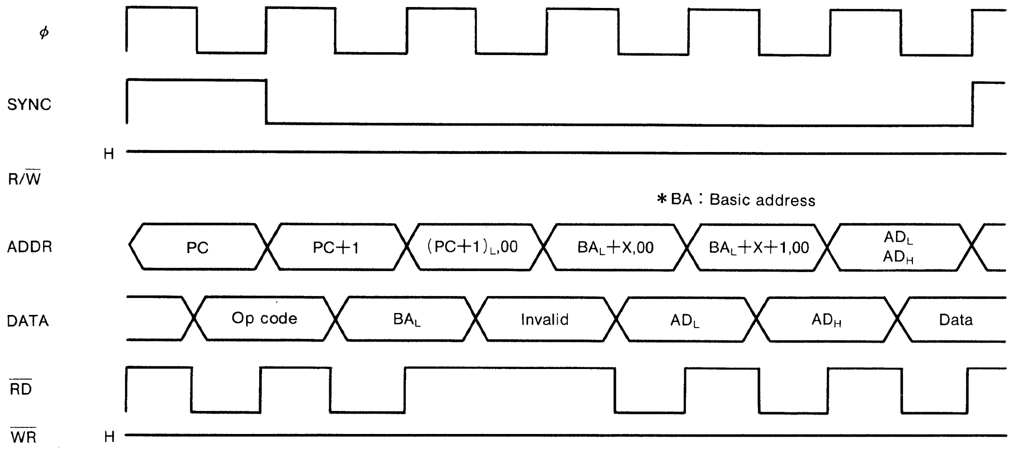
Cycle number : 7

Timing :



INDIRECT X

Instructions	: ADC△ ($zz, X) (T=0) AND△ ($zz, X) (T=0) CMP△ ($zz, X) (T=0) EOR△ ($zz, X) (T=0) LDA△ ($zz, X) (T=0) ORA△ ($zz, X) (T=0) SBC△ ($zz, X) (T=0)
Byte length	: 2
Cycle number	: 6
Timing	:



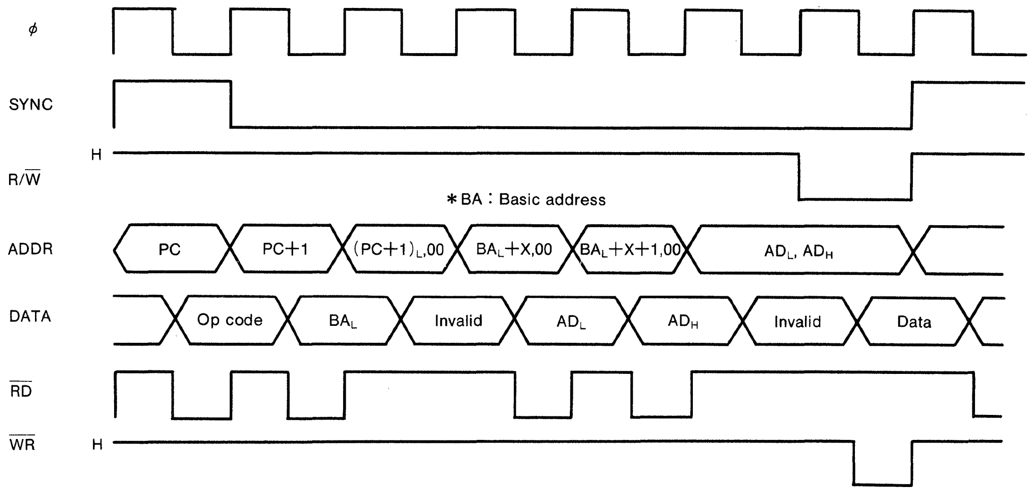
INDIRECT X

Instructions : STA△ ($zz, X)

Byte length : 2

Cycle number : 7

Timing :



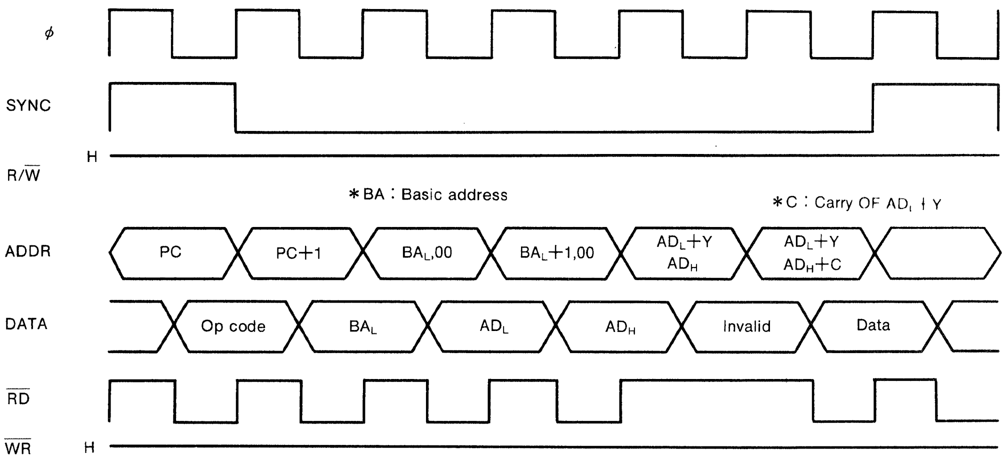
INDIRECT Y

Instructions : ADC△ ($zz), Y(T=0)
AND△ ($zz), Y(T=0)
CMP△ ($zz), Y(T=0)
EOR△ ($zz), Y(T=0)
LDA△ ($zz), Y(T=0)
ORA△ ($zz), Y(T=0)
SBC△ ($zz), Y(T=0)

Byte length : 2

Cycle number : 6

Timing :

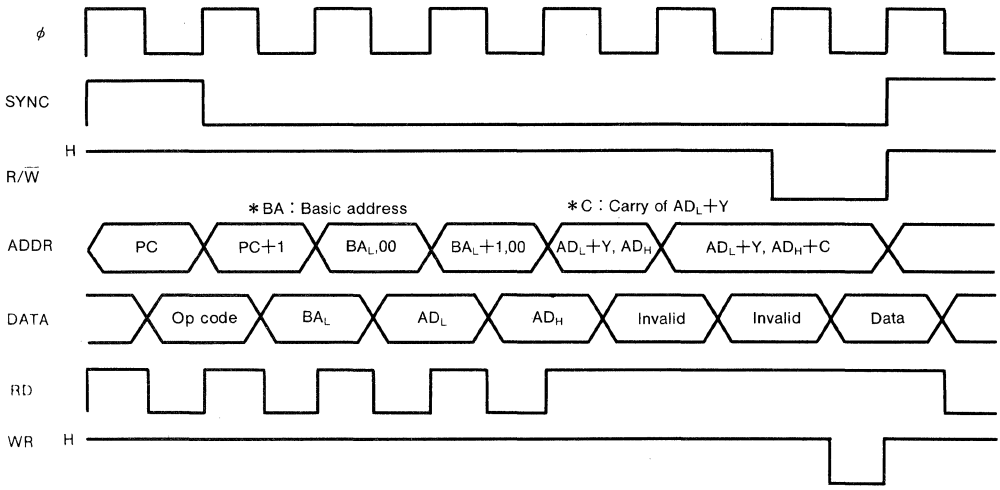


*BA : Basic address

*C : Carry OF AD_L + Y

INDIRECT Y

Instructions	: STA△ ($zz), Y
Byte length	: 2
Cycle number	: 7
Timing	:



RELATIVE

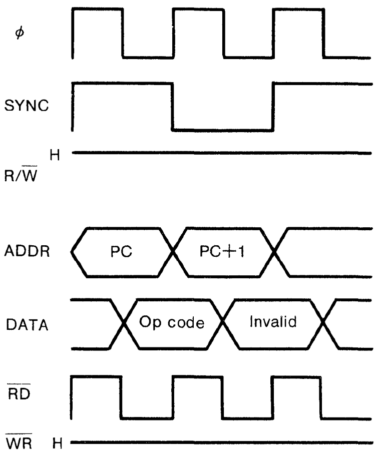
Instructions : BCC△$hhll
BCS△$hhll
BEQ△$hhll
BMI△$hhll
BNE△$hhll
BPL△$hhll
BVC△$hhll
BVS△$hhll

Byte length : 2

①With no branch

Cycle number : 2

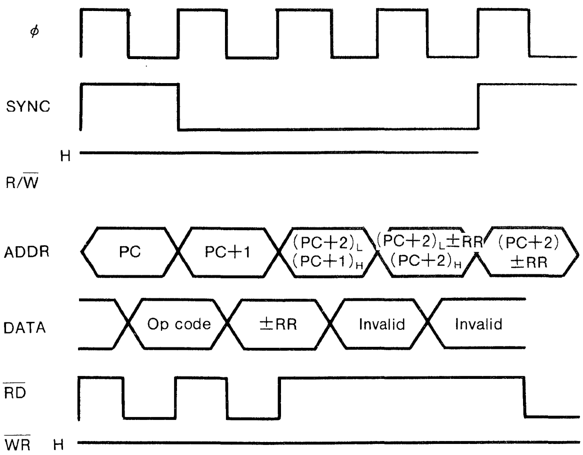
Timing :



②With branch

Cycle number : 4

Timing :



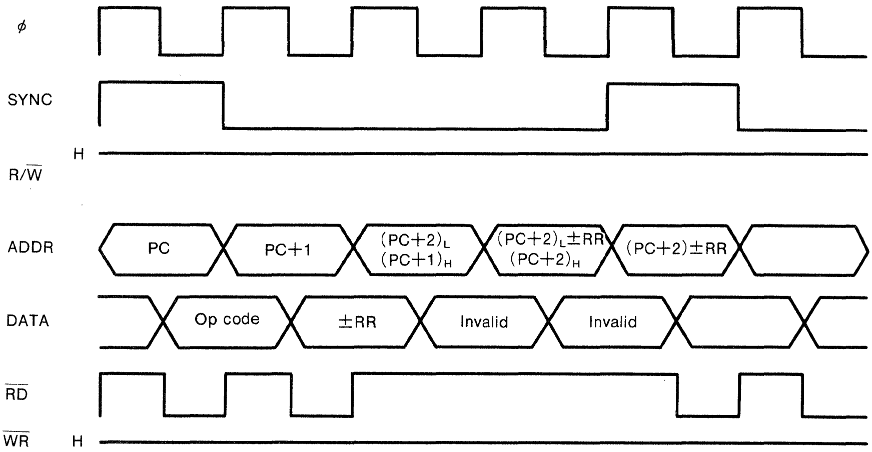
RELATIVE

Instructions : BRA△$hhll

Byte length : 2

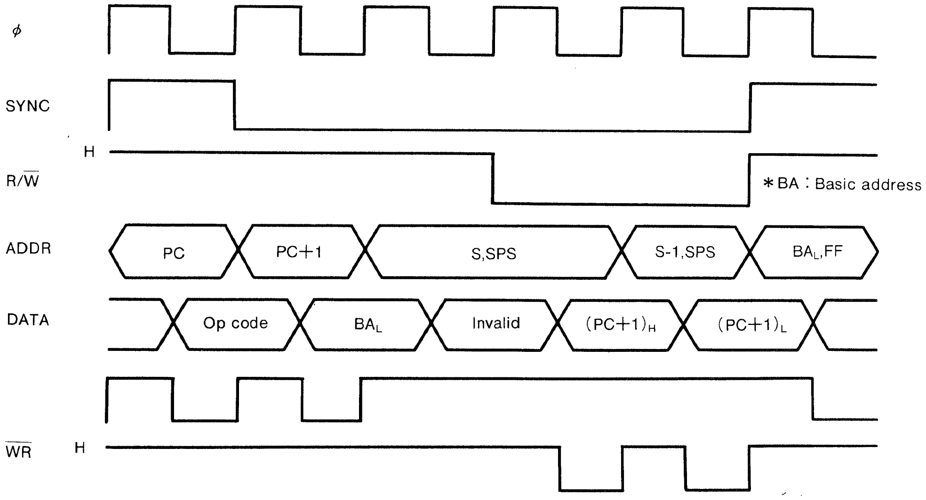
Cycle number : 4

Timing :



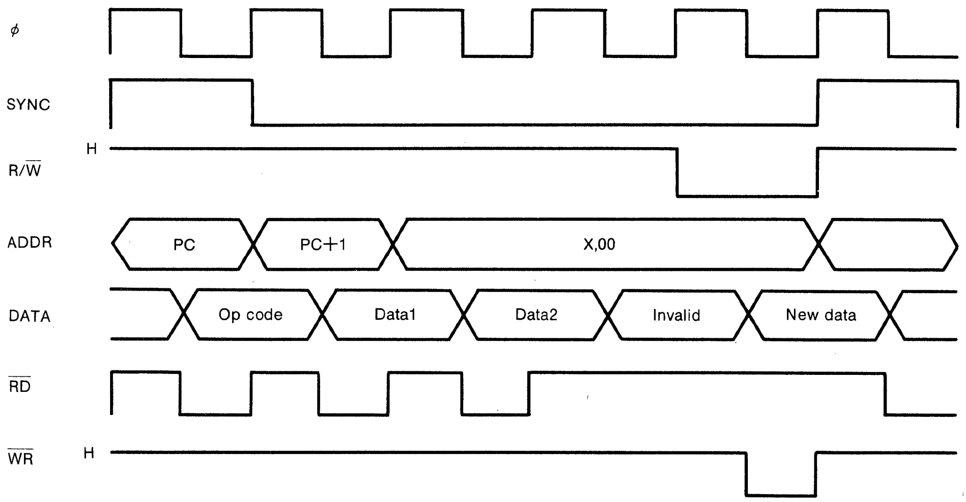
SPECIAL PAGE

Instructions	: JSR△¥$hhll
Byte length	: 2
Cycle number	: 5
Timing	:



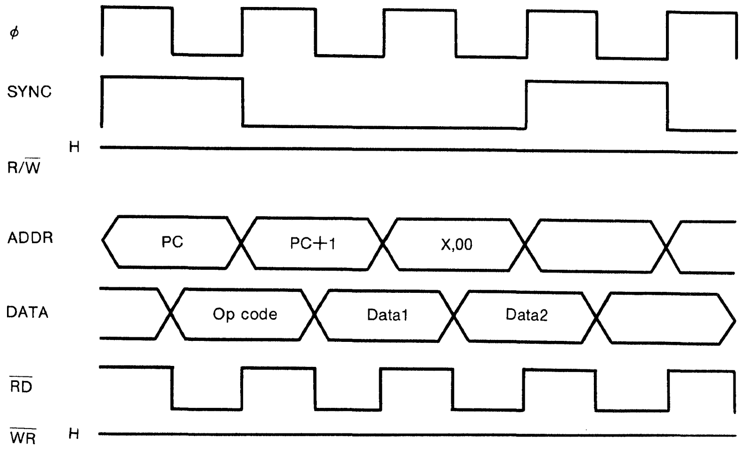
IMMEDIATE

Instructions	: ADC△#$nn (T=1) AND△#$nn (T=1) EOR△#$nn (T=1) ORA△#$nn (T=1) SBC△#$nn (T=1)
Byte length	: 2
Cycle number	: 5
Timing	:



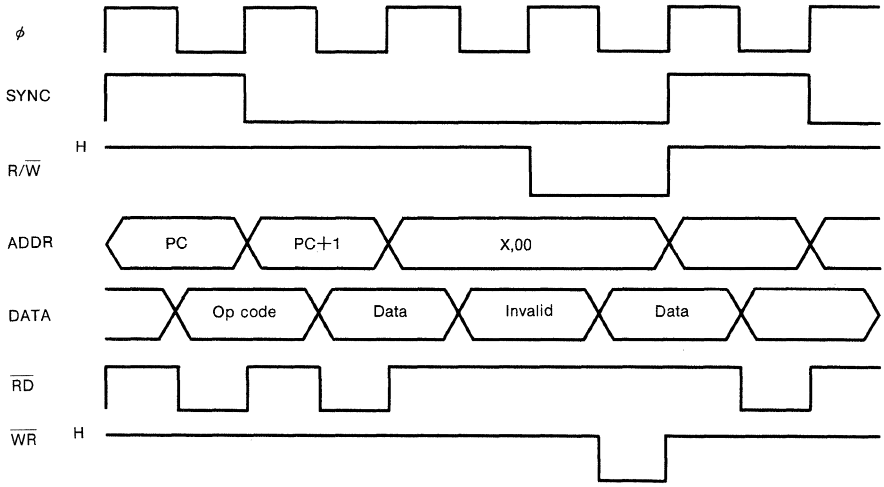
IMMEDIATE

Instructions	: CMP△#$nn (T=1)
Byte length	: 2
Cycle number	: 3
Timing	:



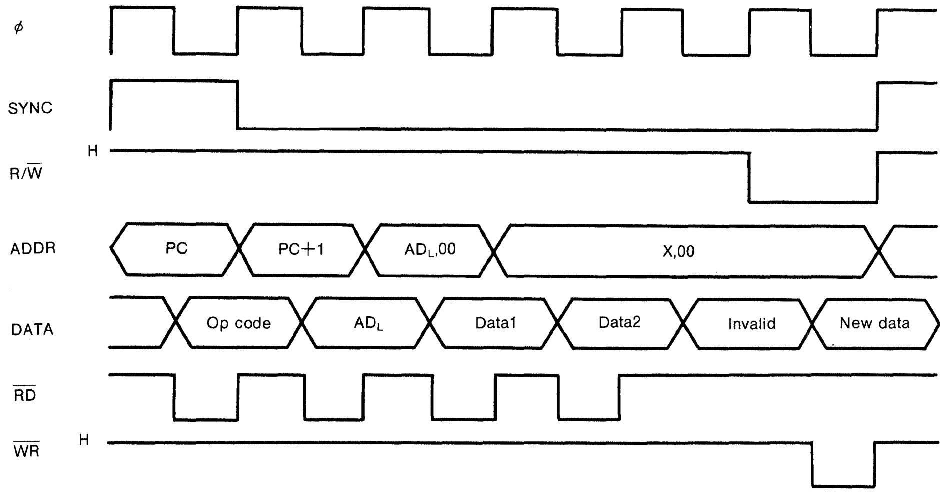
IMMEDIATE

Instructions	: LDA△#$nn (T=1)
Byte length	: 2
Cycle number	: 4
Timing	:



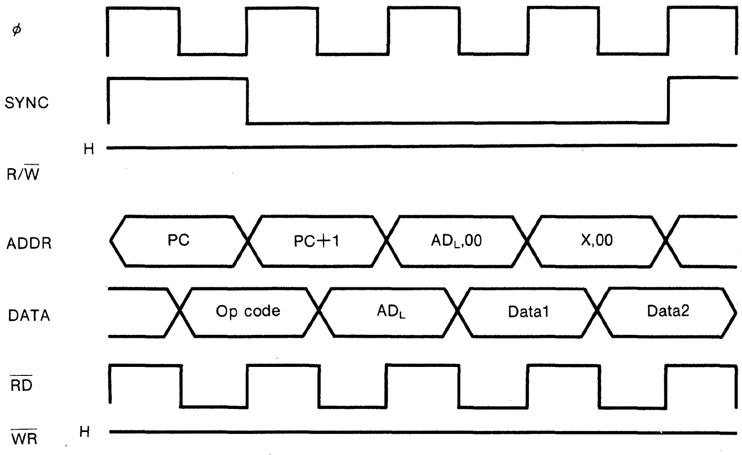
ZERO PAGE

Instructions	: ADC△$zz (T=1) AND△$zz (T=1) EOR△$zz (T=1) ORA△$zz (T=1) SBC△$zz (T=1)
Byte length	: 2
Cycle number	: 6
Timing	:



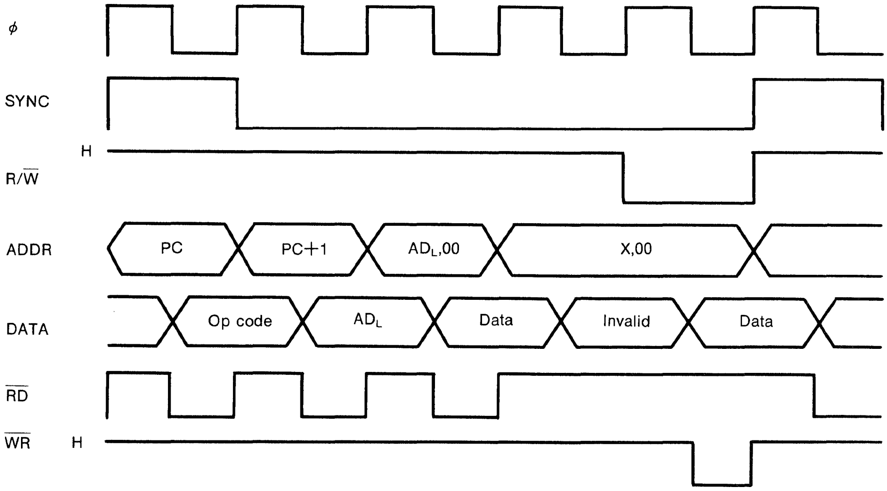
ZERO PAGE

Instructions	: CMP△$zz (T=1)
Byte length	: 2
Cycle number	: 4
Timing	:



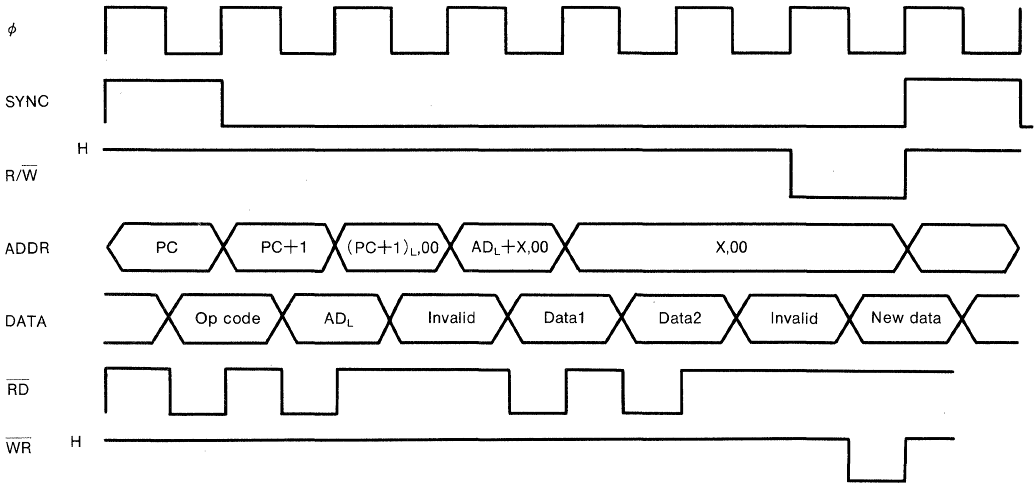
ZERO PAGE

Instructions	: LDA△$zz (T=1)
Byte length	: 2
Cycle number	: 5
Timing	:



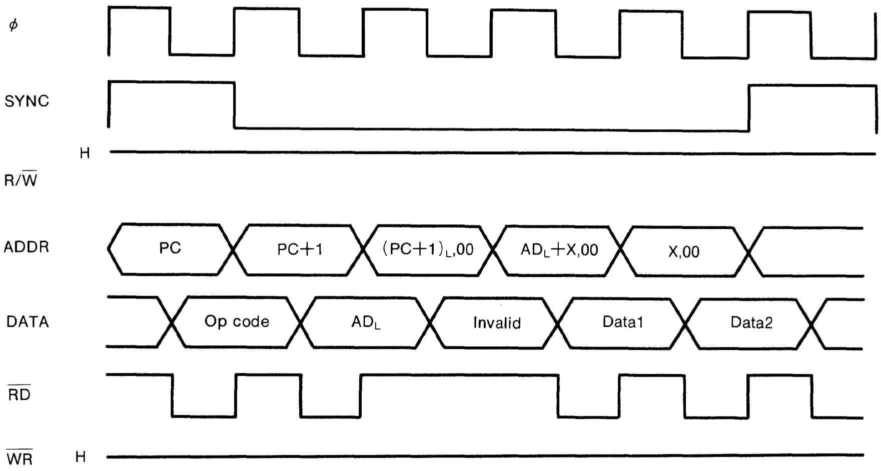
ZERO PAGE X

Instructions	: ADC△$zz, X (T=1) AND△$zz, X (T=1) EOR△$zz, X (T=1) ORA△$zz, X (T=1) SBC△$zz, X (T=1)
Byte length	: 2
Cycle number	: 7
Timing	:



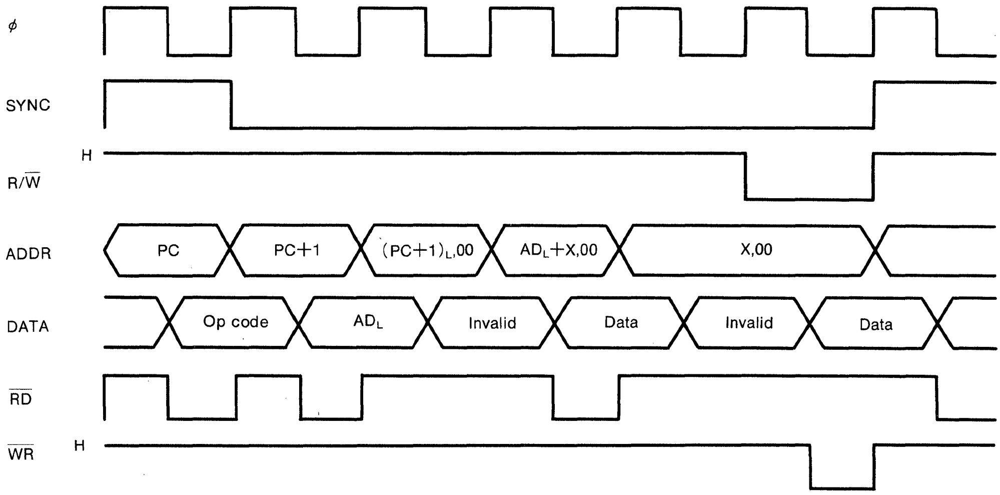
ZERO PAGE X

Instructions	: CMP△$zz, X (T=1)
Byte length	: 2
Cycle number	: 5
Timing	:



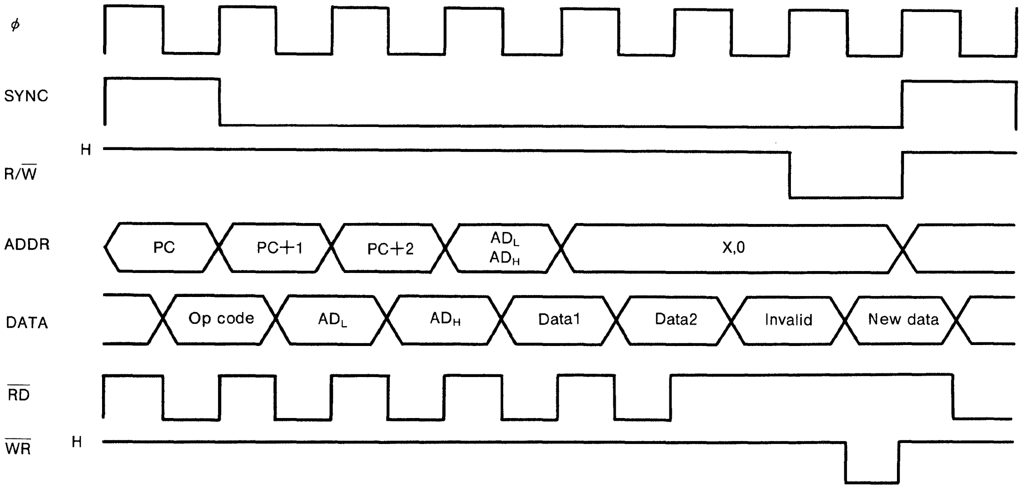
ZERO PAGE X

Instructions	: LDA△$zz, X (T=1)
Byte length	: 2
Cycle number	: 6
Timing	:



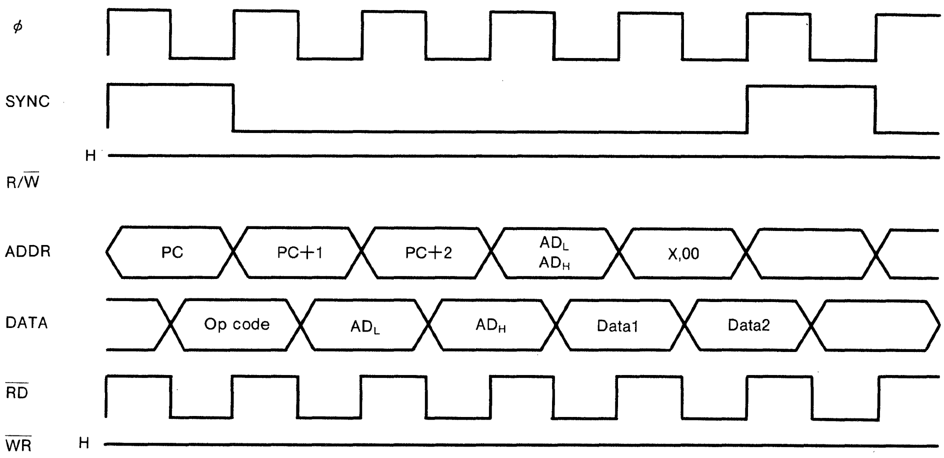
ABSOLUTE

Instructions	: ADC△$hhll (T=1) AND△$hhll (T=1) EOR△$hhll (T=1) ORA△$hhll (T=1) SBC△$hhll (T=1)
Byte length	: 3
Cycle number	: 7
Timing	:



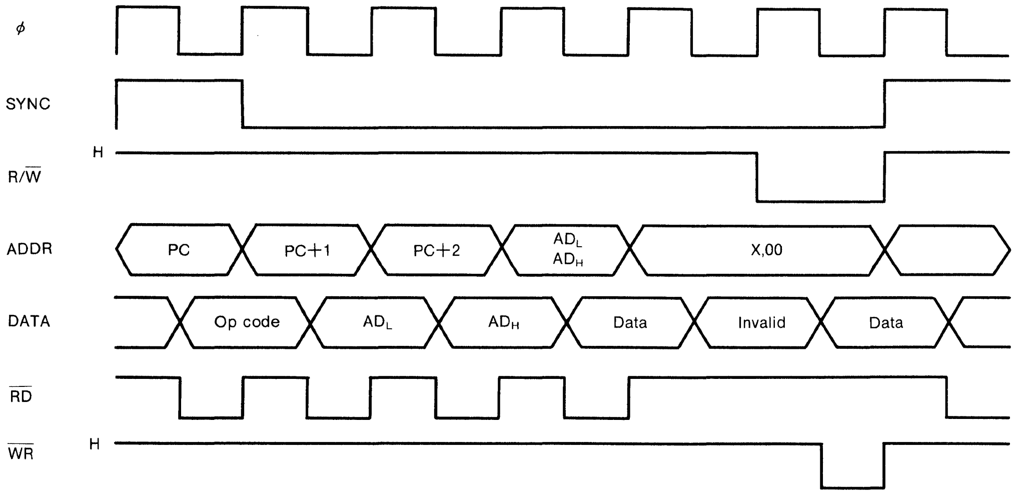
ABSOLUTE

Instructions	: CMP△$hhll (T=1)
Byte length	: 3
Cycle number	: 5
Timing	:



ABSOLUTE

Instructions	: LDA△$hhll (T=1)
Byte length	: 3
Cycle number	: 6
Timing	:



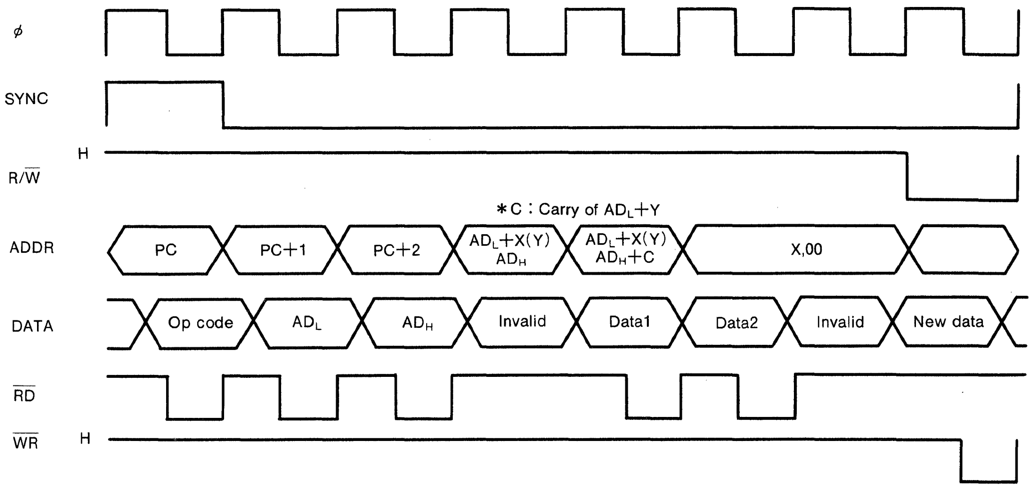
ABSOLUTE X
ABSOLUTE Y

Instructions : ADC△$hhll, X or Y (T=1)
AND△$hhll, X or Y (T=1)
EOR△$hhll, X or Y (T=1)
ORA△$hhll, X or Y (T=1)
SBC△$hhll, X or Y (T=1)

Byte length : 3

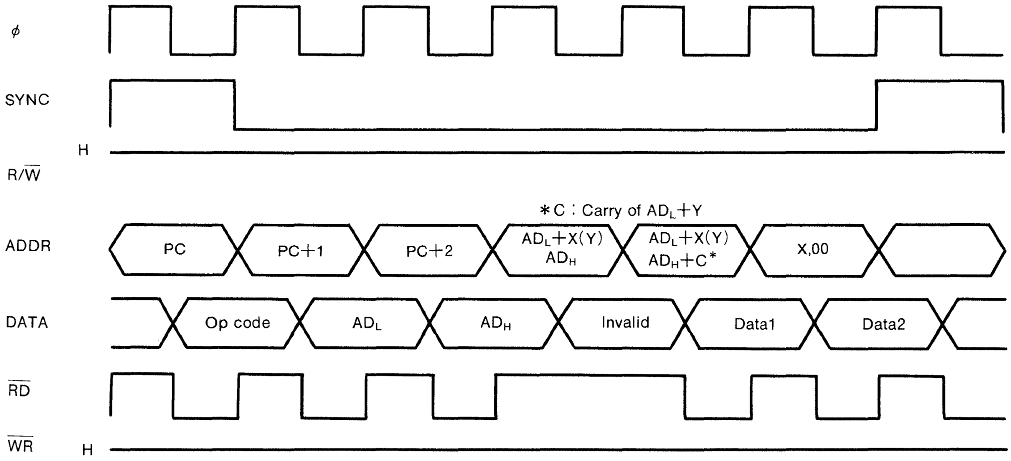
Cycle number : 8

Timing :



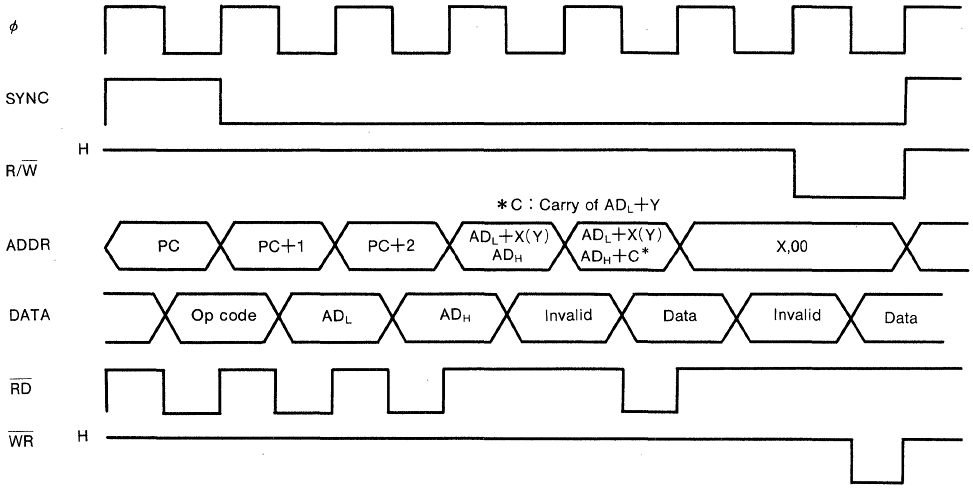
ABSOLUTE X
ABSOLUTE Y

Instructions	: CMP△$hhll, X or Y (T=1)
Byte length	: 3
Cycle number	: 6
Timing	:



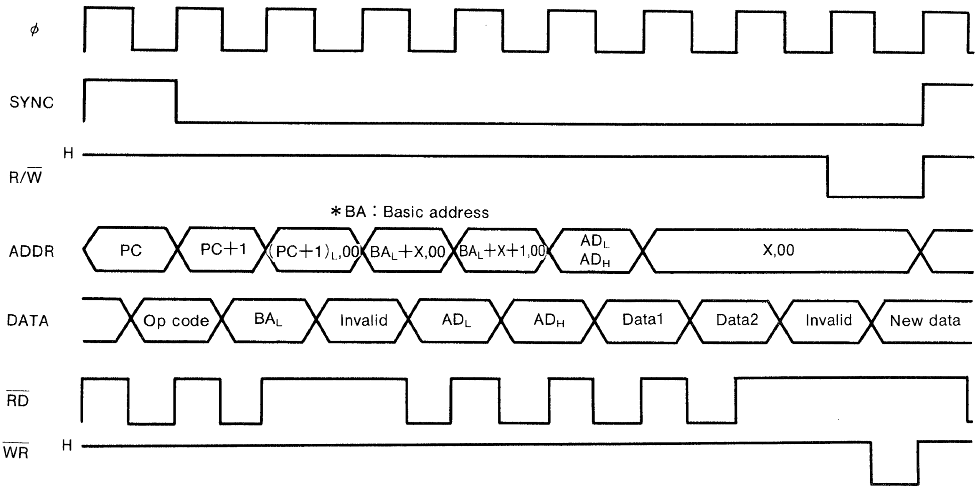
ABSOLUTE X
ABSOLUTE Y

Instructions	: LDA△$hhll, X or Y (T=1)
Byte length	: 3
Cycle number	: 7
Timing	:



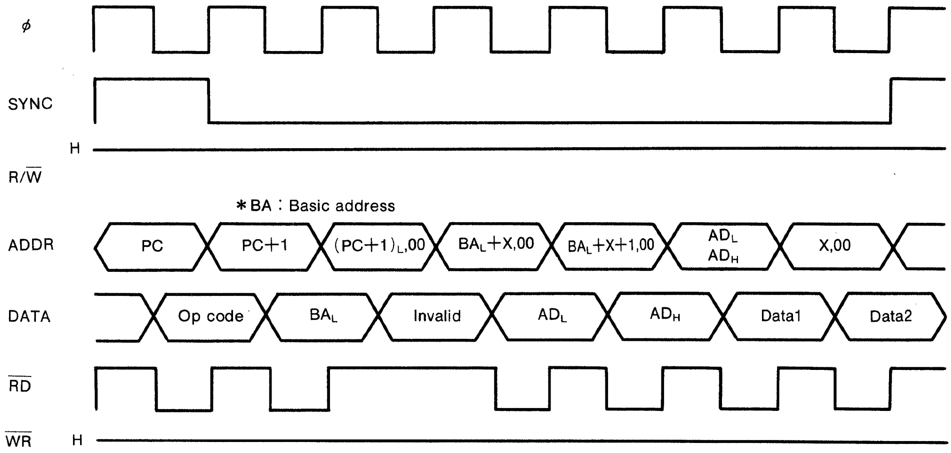
INDIRECT X

Instructions	: ADC△ ($zz, X) (T=1) AND△ ($zz, X) (T=1) EOR△ ($zz, X) (T=1) ORA△ ($zz, X) (T=1) SBC△ ($zz, X) (T=1)
Byte length	: 2
Cycle number	: 9
Timing	:



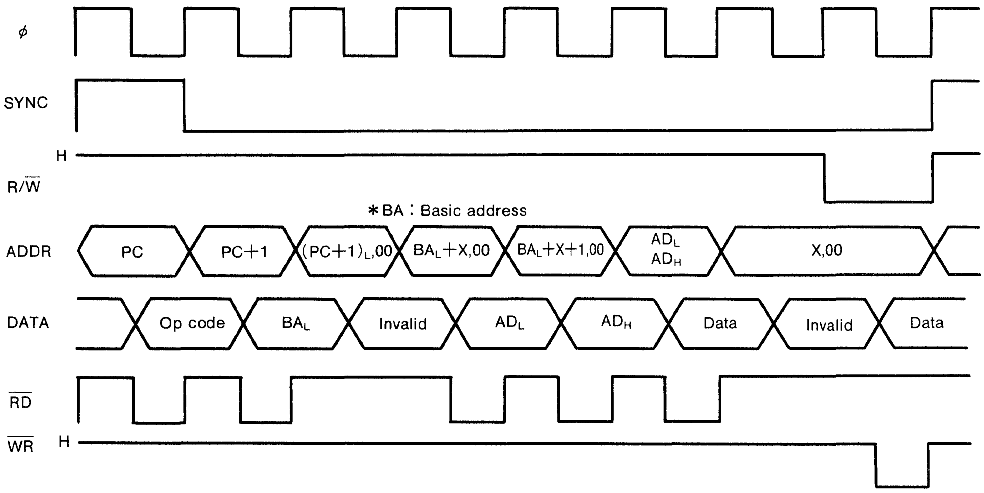
INDIRECT X

Instructions	: CMP△ ($zz, X) (T=1)
Byte length	: 2
Cycle number	: 7
Timing	:



INDIRECT X

Instructions	: LDA△ ($zz, X) (T=1)
Byte length	: 2
Cycle number	: 8
Timing	:



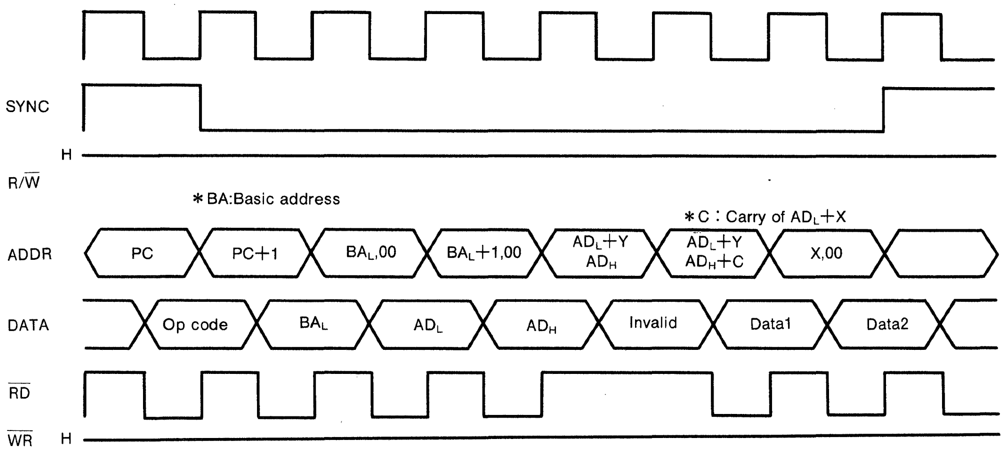
INDIRECT Y

Instructions : CMP△ ($zz), Y (T=1)

Byte length : 2

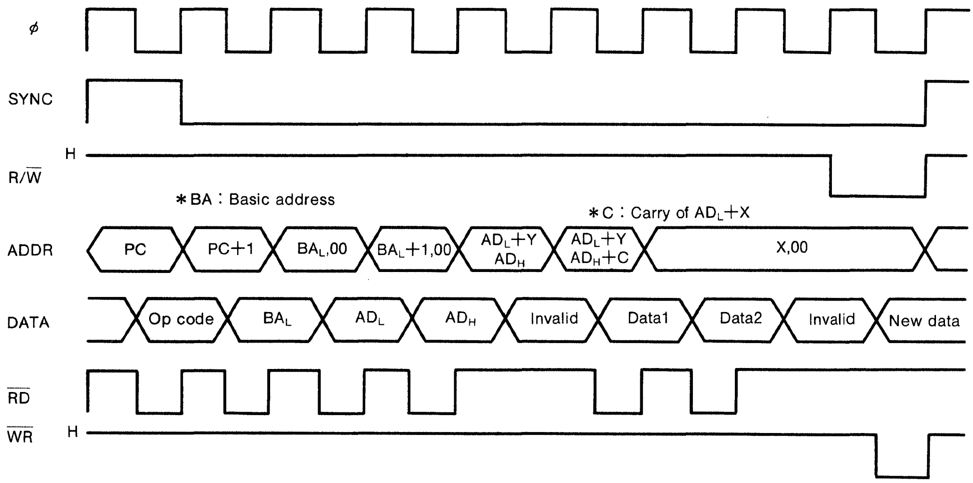
Cycle number : 7

Timing :



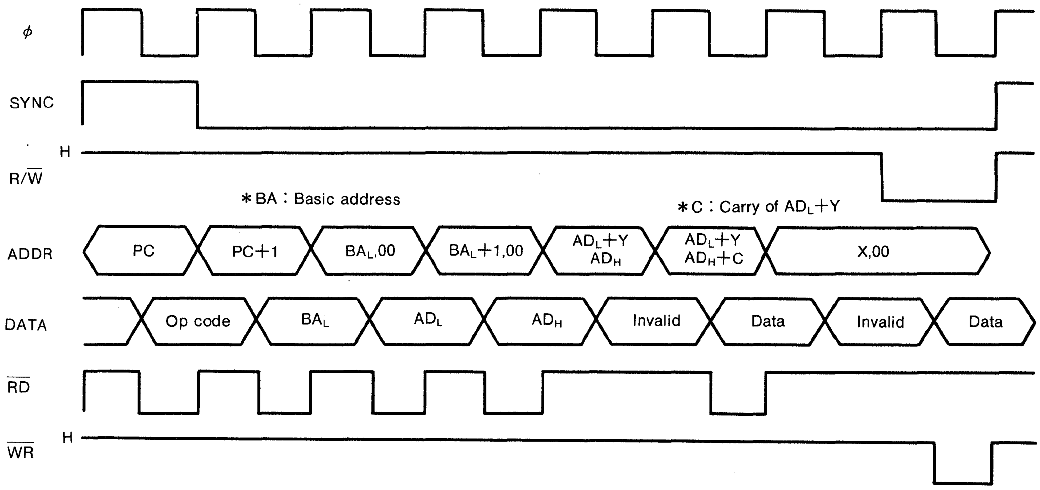
INDIRECT Y

Instructions	: ADC△ ($zz), Y (T=1) AND△ ($zz), Y (T=1) EOR△ ($zz), Y (T=1) ORA△ ($zz), Y (T=1) SBC△ ($zz), Y (T=1)
Byte length	: 2
Cycle number	: 9
Timing	:



INDIRECT Y

Instructions	: LDA△ ($zz), Y (T=1)
Byte length	: 2
Cycle number	: 8
Timing	:



APPENDIX 7. 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) (Note 5)	When T=0 A←A+M+C When T=1 M(X)←M(X)+M+C	Adds the carry, accumulator and memory contents. The results are entered into the accumulator. 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 A←A∧M When T=1 M(X)←M(X)∧M	"AND-s" the accumulator and memory contents. The results are entered into the accumulator. "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 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 M(S)←PC_H S←S−1 M(S)←PC_L S←S−1 M(S)←PS S←S−1 PC_L←AD_L PC_H←AD_H	Executes software interrupt.	00	7	1															

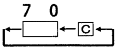
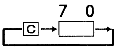
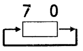
Addressing mode																																	Processor status register							
ZP,X			ZP,Y			ABS			ABS,X			ABS,Y			IND			ZP,IND			IND,X			IND,Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							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	•	•

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 A−M When T=1 M(X)−M	Compares the contents of accumulator and memory. 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 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															
DIV	A←(M(zz+X+1), M(zz+X))/A M (S) ← 1's complement of Remainder S←S−1	Divides by accumulator the 16-bit data that is the contents of M(zz+x+1) for high byte and the contents of the next address memory for low byte, and stores the quotient in the accumulator and the remainder on the stack as 1's complement.																		
EOR (Note 1)	When T=0 A←A∀M When T=1 M(X)←M(X)∀M	"Exclusive-ORs" the contents of accumulator and memory. The results are stored into the accumulator. "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			
INC	A←A+1 or 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
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
																											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	•
E2	16	2																															•	•	•	•	•	•	•	•
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	•

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 $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow (AD_H, AD_L)$ $PC_H \leftarrow (AD_H, AD_L+1)$ If addressing mode is ZP, IND $PC_L \leftarrow (00, AD_L)$ $PC_H \leftarrow (00, AD_L+1)$	Jumps to new address.																		
JSR	$M(S) \leftarrow PC_H$ $S \leftarrow S-1$ $M(S) \leftarrow PC_L$ $S \leftarrow S-1$ After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow (00, AD_L)$ $PC_H \leftarrow (00, AD_L+1)$	After storing contents of program counter in stack, and jumps to new address.																		
LDA (Note 2)	When T=0 $A \leftarrow M$ When T=1 $M(X) \leftarrow M$	Load accumulator with contents of memory. 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	$M \leftarrow IMM$	Load memory with immediate value.													3C	4	3			
LDX	$X \leftarrow M$	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	7 0 0 → □ → C	Shift the contents of accumulator or memory to the right by one bit. 0th bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	2	1				46	5	2			
MUL	$M(S) \cdot A \leftarrow A \times M(zz+X)$ $S \leftarrow S-1$	Multiplies accumulator with the memory specified by the zero page X addrressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.																		
NOP	$PC \leftarrow PC+1$	No operation.	EA	2	1															
ORA (Note 1)	When T=0 $A \leftarrow A \vee M$ When T=1 $M(X) \leftarrow M(X) \vee M$	Produce the logical OR of the contents of memory and accumulator. The result is stored in accumulator. 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
62	15	2																															•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•

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 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 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 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 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 PS←M(S) S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the interrupt routine to the main routine.	40	6	1															
RTS	S←S+1 PC_L←M(S) S←S+1 PC_H←M(S)	Returns from the subroutine to the main routine.	60	6	1															
SBC (Note 1) (Note 5)	When T=0 A←A−M−C When T=1 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. 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															

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	•	•	•	•	•

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															
WIT		Stops the internal clock.	C2	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.
5 : N, V and Z flags are invalid at decimal operation mode.

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
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	•
																																	•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	—	Subtraction
A	Accumulator or Accumulator addressing mode	∧	Logical OR
		∨	Logical AND
BIT, A	Accumulator bit relative addressing mode	⊻	Logical 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 8 low-order bits.
IND, X	Indirect X addressing mode		
IND, Y	Indirect Y addressing mode	$(00, AD_L)$	Contents of address indicated by zero page AD_L
REL	Relative addressing mode	FF	FF in Hexadecimal notation
SP	Special page addressing mode	M	Memory specified by address designation of any addressing mode
C	Carry flag		
Z	Zero flag	M (X)	Memory of address indicated by contents of index register X
I	Interrupt disable flag		
D	Decimal mode flag	M (S)	Memory of address indicated by contents of stack pointer
B	Break flag		
T	X-modified arithmetic mode flag	A_b	1 bit of accumulator
V	Overflow flag	M_b	1 bit of memory
N	Negative flag	OP	Opcode
		n	Number of cycles
		#	Number of bytes

APPENDIX 8. List of instruction codes

	D$_3$~D$_0$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
D$_7$~D$_4$	Hexadecimal notation	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	MUL	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	WIT	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	DIV	BBS 7, A	CPX ZP	SBC ZP	FST (Note2) DIV	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

APPENDIX 9. Mask ROM ordering method

Please send the following data for mask orders.

- mask ROM order confirmation form
- mark specification form
- ROM data······EPROM 3 sets

GZZ—SH00—95A＜75B0＞

Mask ROM number	

SERIES MELPS 740 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37450M2-XXXSP/FP
MITSUBISHI ELECTRIC

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked＊.

※ Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
	Date issued	Date :			

※ 1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

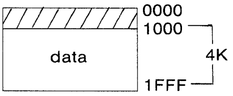
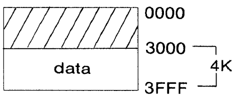
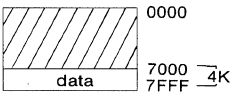
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).

If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in 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.

Microcomputer name : □ M37450M2-XXXSP □ M37450M2-XXXFP

Checksum code for entire EPROM areas					(hexadecimal notation)

EPROM type

□ 2764	□ 27128	□ 27256
		

Set "FF_{16}" in the shaded area.

※ 2 . Mark specification

Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (64P4B for M37450M2-XXXSP ; 80P6 for M37450M2-XXXFP) and attach to the mask ROM confirmation form.

※ 3 . Comments

GZZ−SH00−99A＜75B0＞

SERIES MELPS 740 MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M37450M4-XXXSP/FP
MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※ Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
	Date issued	Date :			

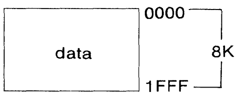
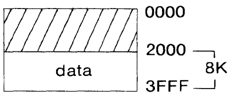
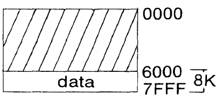
※ 1 . Confirmation

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 in 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.

Microcomputer name : □ M37450M4-XXXSP □ M37450M4-XXXFP

Checksum code for entire EPROM areas □□□□ (hexadecimal notation)

EPROM type

□ 2764	□ 27128	□ 27256
		

Set "FF_{16}" in the shaded area.

※ 2 . Mark specification

Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (64P4B for M37450M4-XXXSP ; 80P6 for M37450M4-XXXFP) and attach to the mask ROM confirmation form.

※ 3 . Comments

GZZ—SH01—00A<76B0>

SERIES MELPS 740 MASK ROM ORDER CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M37450M8-XXXSP/FP MITSUBISHI ELECTRIC

Mask ROM number	

Receipt	Date :	
	Section head signature	Supervisor signature

Note : Please fill in all items marked※.

※	Customer	Company name	TEL ( )	Issuance signature	Responsible officer	Supervisor
		Date issued	Date :			

※ 1 . Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

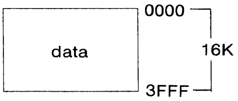
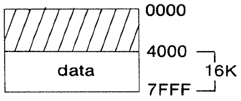
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).

If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based in 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.

Microcomputer name : □ M37450M8-XXXSP □ M37450M8-XXXFP

Checksum code for entire EPROM areas					(hexadecimal notation)

EPROM type

□ 27128	□ 27256
	

Set "FF$_{16}$" in the shaded area.

※ 2 . Mark specification

Mark specification must be submitted using the correct form for the type package being ordered fill out the appropriate mark specification form (64P4B for M37450M8-XXXSP ; 80P6 for M37450M8-XXXFP) and attach to the mask ROM confirmation form.

※ 3 . Comments

64P4B (64-PIN SHRINK DIP) MARK SPECIFICATION FORM

1. Standard Mitsubishi mark
2. Standard mark+Customer's parts number
3. Special mark required

For 2 :

← Up to 19 characters

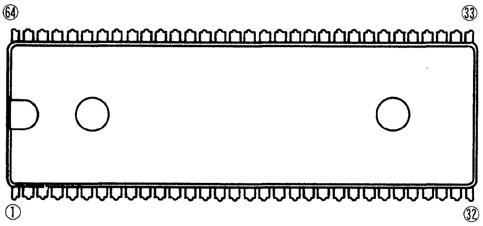
Mitsubishi IC catalog name

2-a. Mitsubishi logo required
2-b. Mitsubishi logo not required

Note 1 : The mark field should be written to the right.
2 : The identification mark can be up to 19 alphanumeric characters (except J, I and O) and hyphens.

For 3 :

Note 3 : If the special character fonts (ex. customer's trademark logo) must be used in special mark, a clean font original (ideally a logo drawing) must be submitted.
4 : If special mark is to be printed, indicate the desired layout on the package drawing below. The layout will be duplicated as closely as technically possible.



80P6 (80-PIN QFP) MARK SPECIFICATION FORM

1. Standard Mitsubishi mark
2. Standard mark+Customer's parts number
3. Special mark required

For 2 :

← Up to 12 characters

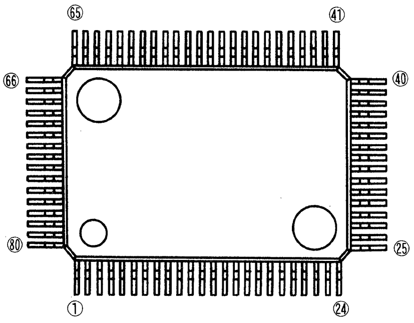
	Mitsubishi IC catalog name

2-a. Mitsubishi logo required
2-b. Mitsubishi logo not required

Note 1 : The mark field should be written to the right.
2 : The identification mark can be up to 12 alphanumeric characters (except J, I and O) and hyphens.

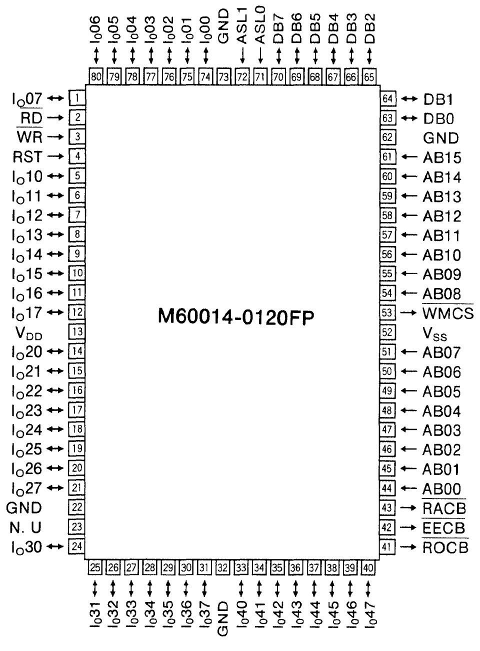
For 3 :

Note 3 : If the special character fonts (ex. customer's trademark logo) must be used in special mark, a clean font original (ideally a logo drawing) must be submitted.
4 : If special mark is to be printed, indicate the desired layout on the package drawing below. The layout will be duplicated as closely as technically possible.



APPENDIX 10. M60014-0120FP

Pin cofiguration



Outline

The M60014-012FP is a 2-micron CMOS gate array. As LSI for CPU I/O interfaces, it emulates ports P0, P1 and P2 of the M37450 in microprocessor mode.

This gate array operates on a 5V power supply and can .connect directly to the M37450 data bus and address bus. It has 40 I/O ports, these ports being assigned in groups of 3 to 5 addresses. As with the MELPS 740 series I/O ports, they can be defined in 1-bit units as input and output ports.

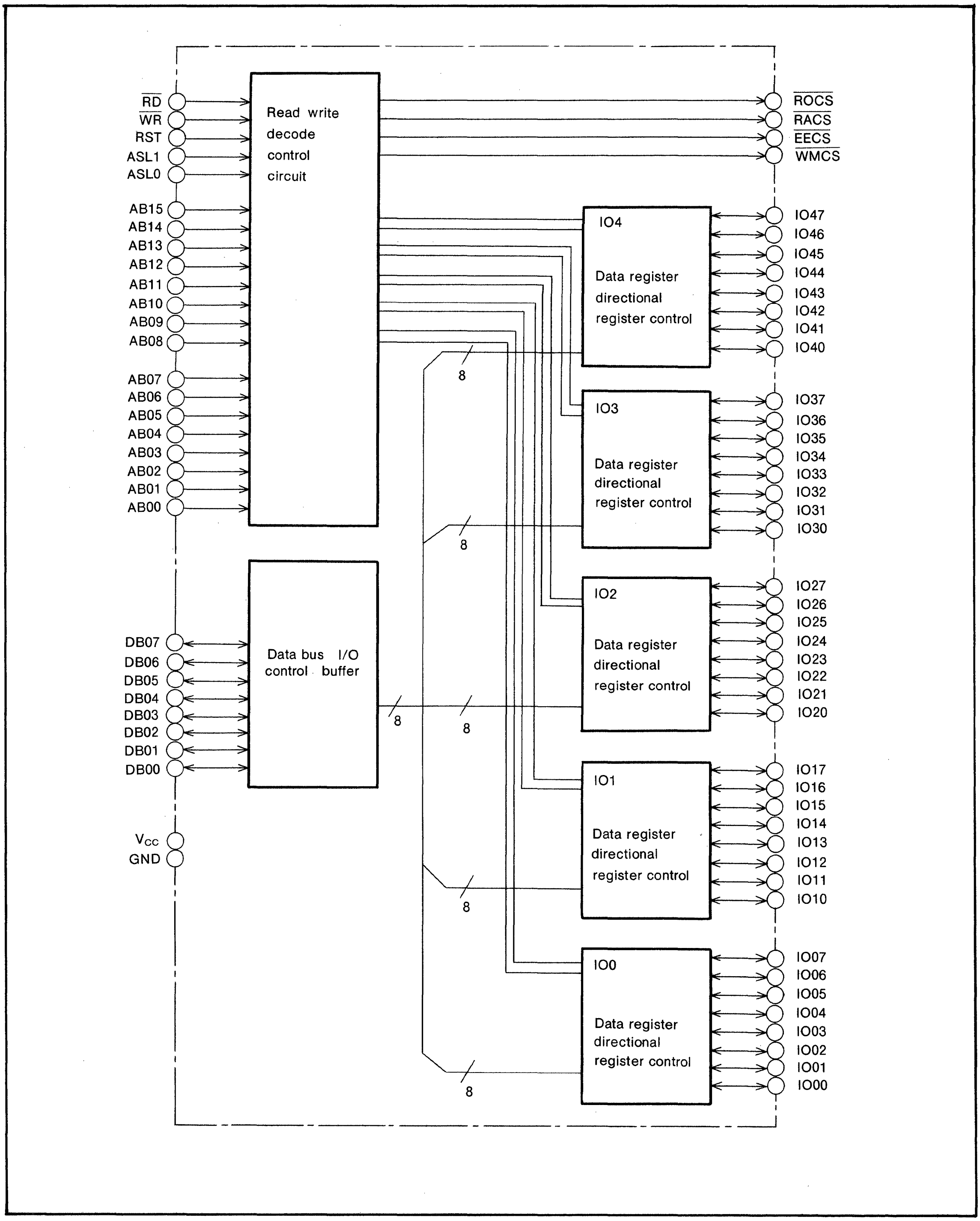
Features

- 5V power supply
- Four address decode ouptuts (for EPROM, RAM×2, and EEPROM)
- Port addresses selectable from a choice of four
- Direct connection to M37450 QFP type (can not for DIP type)
- No-wait connection to M37450 in 8MHz operation
- Forty built-in ports with almost similar electrical characteristics to M37450 I/O ports

Application

I/O port expansion when the M37450 and the MELPS 740 series with built-in CMOS output and TTL input I/O ports are in microporocessor mode.

Block diagram



Function outline

The 40 bits of the I/O pins are divided into five groups of 8 bits. These function as CMOS output and TTL input ports. As shown in Table 7.1, port IO0 is treated as input address $00D0_{16}$ when ASL1=0 and ASL0=1. Port IO0 has a directional register ($00D1_{16}$) allowing each I/O bit to be individually programmed as input or ouptut as with the MELPS 740.
Similarly, when ASL1=0 and ASL0=1:

- Port IO1 is treated as input address $00D2_{16}$ and has a directional register at address $00D3_{16}$.
- port IO2 is treated as input address $00D4_{16}$ and has a directional retister at address $00D5_{16}$.
- Port IO3 is treated as input address $00D6_{16}$ and has a directional register at address $00D7_{16}$.
- Port IO4 is treated as input addres $00D8_{16}$ and has a directional register at address $00D9_{16}$.

With ASL1 and ASL0, addresses A7 to A4 can be selected from C_{16}, D_{16}, E_{16}, or F_{16}.
If the RST pin is input "H", all I/O pins are initialized as input ports. Four output pins are for chip-select of external memory.
$\overline{\text{ROCS}}$ is the ROM chip select and selects a 32KB area (decodes 8000_{16}-$FFFF_{16}$).
$\overline{\text{RACS}}$ is the RAM chip select and selectes a 512B area (decodes 0000_{16}-$01FF_{16}$).
$\overline{\text{EECS}}$ is the EEPROM chip select and selects a 512B area (decodes 0200_{16}-$03FF_{16}$).
$\overline{\text{WMCS}}$ is the RAM chip select and selects a 32KB area (decodes 0000_{16}-$7FFF_{16}$).

Functions

[Input pins]

- $\overline{\text{RD}}$ (read) input
  When this pin is "L" and the addresses of the port are decoded, the content of this port is output to data bus.
- $\overline{\text{WR}}$ (write)
  When this pin is "L" and the addresses of the port are decoded, input the data from the data bus and latches the data in the port latch.
- AB00~AB15 (address) input
  These are used to select from I00, I02, I03 and I04 and to generate the chip-select output.
- RST (reset) input
  Clear the data latch of the port and set the directional register when this port is "H".
- ALS1, ALS0 (address selector) input
  Select the 12 most significant bits of the addresses of ports I00~I04 as shown in Table 7.1.

[I/O pins]

- DB07-DB00 (data bus) I/O
  Input and latch data using $\overline{\text{WR}}$.
  Output data of the port address has been decoded using $\overline{\text{RD}}$.
- I007-I000 (ports) I/O
  These I/O ports are decoded using the 12 most significant bits of the address and ASL1 and ASL0. The 4 least significant bits of the address are decoded as follows:
  0000_B as data register and 0001_B as directional register.
- I017-I010 (ports) I/O
  These I/O ports are decoded using the 12 most significant bits of the address and ASL1 and ASL0. The 4 least significant bits of the address are decoded as follows:
  0010_B as data register and 0011_B as directional register.
- I027-I020 (ports) I/O
  These I/O ports are decoded using the 12 most significant bits of the address and ASL1 and ASL0. The 4 least significant bits of the address are decoded as follows:
  0100_B as data register and 0101_B as directional register.
- I037-I030 (ports) I/O
  These I/O ports are decoded using the 12 most significant bits of the address and ASL1 and ASL0. The 4 least significant bits of the address are decoded as follows:
  0110_B as data register and 0111_B as directional register.

- I047-I040 (ports) I/O

  These I/O ports are decoded using the 12 most significant bits of the address and ASL1 and ASL0. The 4 least significant bits of the address are decoded as follows:

  1000_B as data register and 1001_B as directional register.

[Output pins]

- $\overline{\text{ROCS}}$ (chip select) output

  This chip select is set for ROM and selects a 32KB area (decodes 8000_{16}~$FFFF_{16}$).
- $\overline{\text{RACS}}$ (chip select) output

  This chip select is set for RAM and selects a 512B area (decodes 0000_{16}~$01FF_{16}$).
- $\overline{\text{EECS}}$ (chip select) output

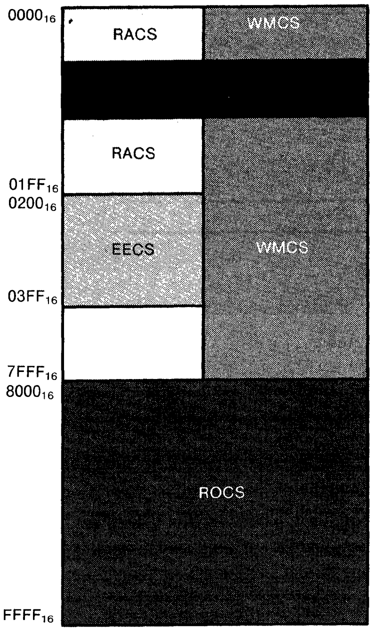
  This chip select is set for the EEPROM and selects a 512B area (decodes 0200_{16}~$03FF_{16}$).
- $\overline{\text{WMCS}}$ (chip select) output

  Thsi chip select is set for the ROM and selects a 32KB area (decodes 0000_{16}~$7FFF_{16}$).

Address select table

ASL1	ALS0	AB15~AB08	A7	A6	A5	A4	A3~A0	
0	0	0~0	1	1	0	0	X~X	($00CX_{16}$)
0	1	0~0	1	1	0	1	X~X	($00DX_{16}$)
1	0	0~0	1	1	1	0	X~X	($00EX_{16}$)
1	1	0~0	1	1	1	1	X~X	($00FX_{16}$)

Address map



ASL1=0, ASL0=0		ASL1=0, ASL0=1	
00C0$_{16}$	I00 data register	00D0$_{16}$	I00 data register
00C1$_{16}$	I00 directional register	00D1$_{16}$	I00 directional register
00C2$_{16}$	I01 data register	00D2$_{16}$	I01 data register
00C3$_{16}$	I01 directional register	00D3$_{16}$	I01 directional register
00C4$_{16}$	I02 data register	00D4$_{16}$	I02 data register
00C5$_{16}$	I02 directional register	00D5$_{16}$	I02 directional register
00C6$_{16}$	I03 data register	00D6$_{16}$	I03 data register
00C7$_{16}$	I03 directional register	00D7$_{16}$	I03 directional register
00C8$_{16}$	I04 data register	00D8$_{16}$	I04 data register
00C9$_{16}$	I04 directional register	00D9$_{16}$	I04 directional register

ASL1=0, ASL0=0		ASL1=0, ASL0=1	
00E0$_{16}$	I00 data register	00F0$_{16}$	I00 data register
00E1$_{16}$	I00 directional register	00F1$_{16}$	I00 directional register
00E2$_{16}$	I01 data register	00F2$_{16}$	I01 data register
00E3$_{16}$	I01 directional register	00F3$_{16}$	I01 directional register
00E4$_{16}$	I02 data register	00F4$_{16}$	I02 data register
00E5$_{16}$	I02 directional register	00F5$_{16}$	I02 directional register
00E6$_{16}$	I03 data register	00F6$_{16}$	I03 data register
00E7$_{16}$	I03 directional register	00F7$_{16}$	I03 directional register
00E8$_{16}$	I04 data register	00F8$_{16}$	I04 data register
00E9$_{16}$	I04 directional register	00F9$_{16}$	I04 directional register

□ RACS decode area

░ EECS decode area

▒ WMCS decode area

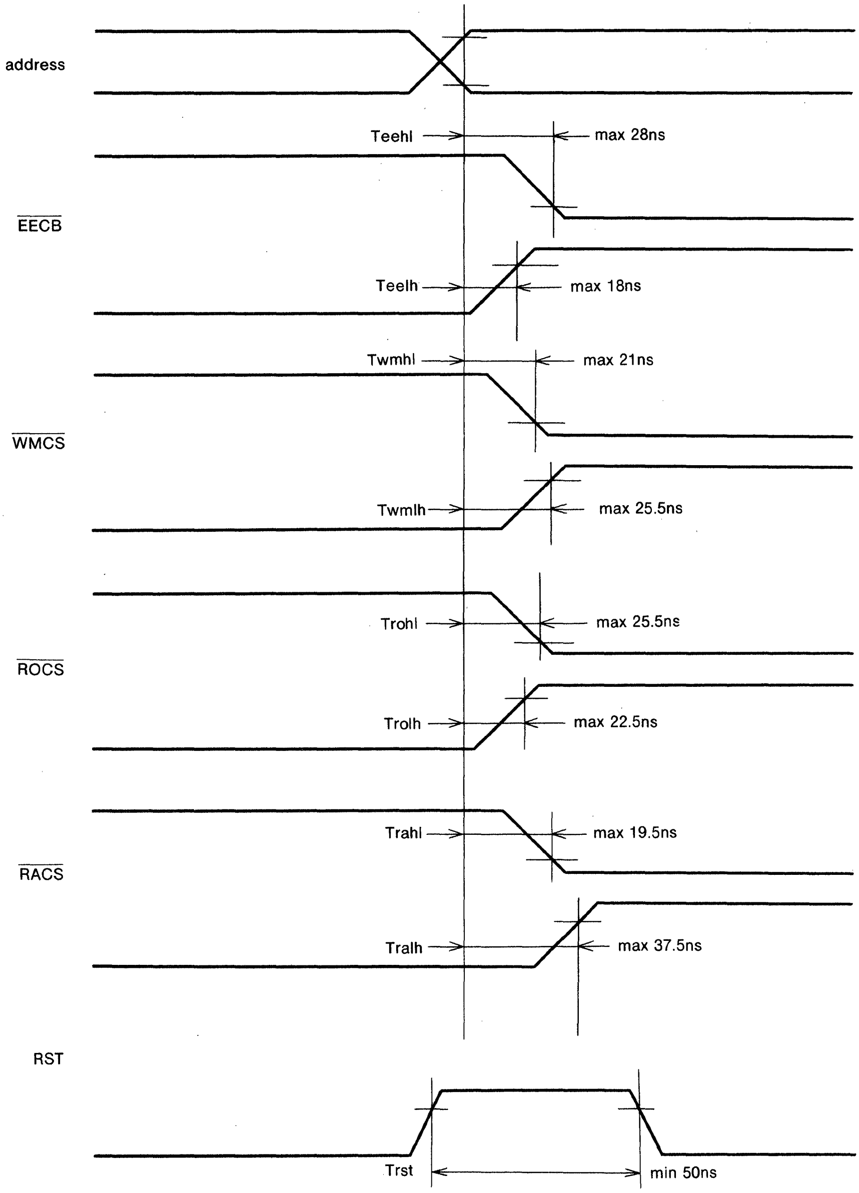
▓ ROCS decode area

■ is dynamically changed by ASL1 and ASL0. When this area is decoded, overlapping RACS and WMCS output "H".

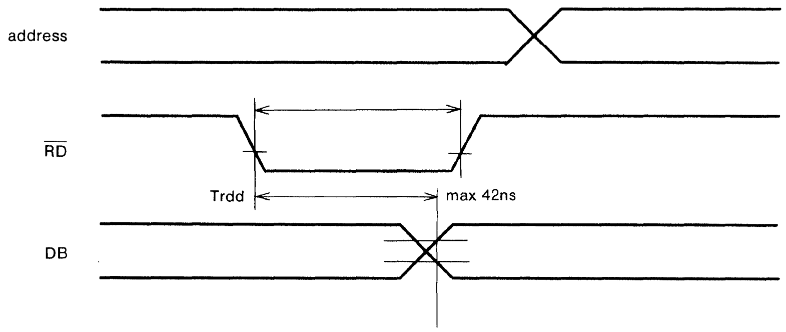
Note 1. If M37450M2-XXXFP and M60014-0120FP are set to address selectors ASL1=0 and ASL0=1 then connected, ports P0, P1 and P2 of M37450 emulate ports IO0, IO1 and IO2 of M60014-0120FP. Ports IO3 and IO4, which correspond to M37450 ports P3 and P4 are also assigned to the M60014-0102FP, but IO3 and IO4 do not emulate P3 and P4. (Ports IO3 and IO4 of the M60014-0120FP output the same data as P3 and P4 when they are set as output ports, but they cannot be used as input ports because, when P3 and P4 are set as input ports, the data input to ports IO3 and IO4 is not read by the M37450.)

Note 2. User cannot use the 00XA$_{16}$~00XF$_{16}$ (X=C, D, E, F) which are in the decoded area by ASL1 and ASL0.

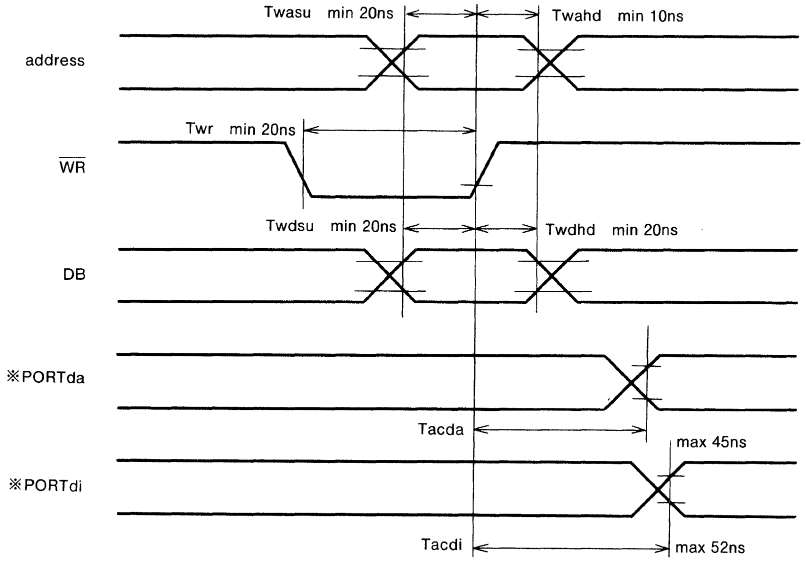
Timing chart



- at read operation



- at write operation



※PORTda at the time write to data register when the contents of the register is "1".
※PORTdi at the time write to directional register when the contents of the register is "1".

Electric characteristics

Symbol	Parameter	Condition	Limits		Unit
			max.	min.	
V_{IL}	input voltage (TTL interface)	V_{DD}=5V		0. 8	V
V_{IH}		V_{DD}=5V	2. 0		
V_{OL}	output voltage	V_{DD}=5V		0. 1	V
V_{OH}		I_O=0mA	4. 9		
I_{OL}	output current	V_{OL}=0.4V	6		mA
I_{OH}		V_{OH}=2.4V	−4		

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) 218-3473
(03) 218-3499
Facsimile: (03) 214-5570

Overseas Marketing Manager
Kita-Itami Works
4-1, Mizuhara, Itami-shi,
Hyogo-ken 664, Japan
Telex: 526408 KMELCO J
Telephone: (0727) 82-5131
Facsimile: (0727) 72-2329

HONG KONG

MITSUBISHI ELECTRIC (H.K.) LTD.
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 Arques Avenue
Sunnyvale, CA 94086
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

SAN DIEGO
Mitsubishi Electronics America, Inc.
11545 West Bernardo Court
Suite 100
San Diego, CA 92128
Telephone: (619) 592-1445
Facsimile: (619) 592-0242

DENVER
Mitsubishi Electronics America, Inc.
4600 South Ulster Street
Metropoint Building, 7th Floor
Denver, CO 80237
Telephone: (303) 740-6775
Facsimile: (303) 694-0613

SOUTHWEST
Mitsubishi Electronics America, Inc.
991 Knox Street
Torrance, CA 90502
Telephone: (213) 515-3993
Facsimile: (213) 217-5781

SOUTH CENTRAL
Mitsubishi Electronics America, Inc.
1501 Luna Road, Suite 124
Carrollton, TX 75006
Telephone: (214) 484-1919
Facsimile: (214) 243-0207

NORTHERN
Mitsubishi Electronics America, Inc.
15612 Highway 7 #243
Minnetonka, MN 55345
Telephone: (612) 938-7779
Facsimile: (612) 938-5125

NORTH CENTRAL
Mitsubishi Electronics America, Inc.
800 N. Bierman Circle
Mt. Prospect, IL 60056
Telephone: (312) 298-9223
Facsimile: (312) 298-0567

NORTHEAST
Mitsubishi Electronics America, Inc.
200 Unicorn Park Drive
Woburn, MA 01801
Telephone: (617) 932-5700
Facsimile: (617) 938-1075

MID-ATLANTIC
Mitsubishi Electronics America, Inc.
800 Cottontail Lane
Somerset, NJ 08873
Telephone: (201) 469-8833
Facsimile: (201) 469-1909

SOUTH ATLANTIC
Mitsubishi Electronics America, Inc.
2500 Gateway Center Blvd., Suite 300
Morrisville. NC 27560
Telephone: (404) 368-4850
Facsimile: (404) 662-5208

SOUTHEAST
Mitsubishi Electronics America, Inc.
Town Executive Center
6100 Glades Road #210
Boca Raton, FL 33433
Telephone: (407) 487-7747
Facsimile: (407) 487-2046

CANADA
Mitsubishi Electronics America, Inc.
6185 Ordan Drive, Unit #110
Mississauga, Ontario, Canada L5T 2E1
Telephone: (416) 670-8711
Facsimile: (416) 670-8715

Mitsubishi Electronics America, Inc.
300 March Road, Suite 302
Kanata, Ontario, Canada K2K 2E2
Telephone: (416) 670-8711
Facsimile: (416) 670-8715

WEST GERMANY

Mitsubishi Electric Europe GmbH
Headquarters:
Gothear Str. 8
4030 Ratingen 1, West Germany
Telex: 8585070 MED D
Telephone: (02102) 4860
Facsimile: (02102) 486-115

Munich Office:
Arabellastraße 31
8000 München 81, West Germany
Telex: 5214820
Telephone: (089) 919006-09
Facsimile: (089) 9101399

FRANCE

Mitsubishi Electric Europe GmbH
55, Avenue de Colmar
92563 Rueil Malmaison Cedex
Telex: 632326
Telephone: 47087871
Facsimile: 47513622

ITALY

Mitsubishi Electric Europe GmbH
Centro Direzionale Colleoni
Palazzo Cassiopea 1
20041 Agrate Brianza I-Milano
Telephone: (039) 636011
Facsimile: (039) 6360120

SWEDEN

Mitsubishi Electric Europe GmbH
Lastbilsvägen 6B
5-19149 Sollentuna, Sweden
Telex: 10877 (meab S)
Telephone: (08) 960468
Facsimile: (08) 966877

U.K.

Mitsubishi Electric (U.K.) Ltd.
Travellers Lane
Hatfield
Herts AL10 8XB, England, U.K.
Telephone: (0044) 7072 76100
Facsimile: (0044) 7072 78692

AUSTRALIA

Mitsubishi Electric Australia Pty. Ltd.
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 SEMICONDUCTORS USER'S MANUAL
M37450M2-XXXSP/FP

Oct. First Edition 1989
Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

MITSUBISHI SEMICONDUCTORS

M37450M2-XXXSP/FP

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.

H-E0040-A KI-8910 Printed in Japan (ROD)

New publication, effective Oct. 1989.
Specifications subject to change without notice.