TOSHIBA AMERICA, INC.

8 BIT MICROCONTROLLER

TLCS
48
90

Notice to Potential TLCS-90* Customers:

This Databook contains a preliminary Data Sheet for Toshiba's new TLCS-90 Series 8-bit Microcontrollers. Although the Data contained therein is generally believed to be current, it is recommended that you contact you local Toshiba Sales Person or Office to obtain the latest Specs prior to any actual System Design.

Thus you will be ensured of having the latest Device Electrical Specs to incorporate into your System Design Process.

* The TLCS-90 Series currently includes the following devices:

- TMP90C840N
- TMP80C840F
- TMP90C841N
- TMP90C841N
- TMP90C840E

Future TLCS-90 Series Devices will be added to this expanding Family.

8-BIT MICROCONTROLLER

TLCS 48, 90

Additional Literature Available:

Literature Description	Device Families
• Microcomputer Products Summary	All CPU Products
• 4-Bit Microcontroller	TLCS-42, TLCS-47/470
• 8-Bit Microcontroller	TLCS-48, TLCS-90
• 8-Bit Microprocessor	TLCS-Z80, TLCS-85
• 16-Bit Microprocessor	TLCS-68000
• Speech Devices	Speech Products
• Micro Peripherals	Other Micro Peripherals

Plus Various Development System Manuals

November 1988

INDEX

PART 1

TLCS-48

LSI DEVICES

CONTENTS

NMOS 8-BIT MICROCONTROLLER (TLCS-48)

TMP8048AP/TMP8035AP
TMP8049AP/TMP8039AP
TMP8048AT/TMP8035AT
TMP8049AT/TMP8039AT

GENERAL DESCRIPTION

The TMP8048AP/TMP8049AP, from here on referred to as the TMP8048A except in case of no need to specify each parts, is a single chip microcontroller internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, RAM data memory, ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

The TMP8048A is particularly efficient as a controller. It has extensive bit handing capability as well as facilities for both binary and BCD arithmetic.

The TMP8035A/TMP8039A is the equivalent of a TMP8048A/TMP8049A without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP8048AP/TMP8035AP, TMP8049AP/TMP8039AP are packaged in a standard 40 pin Dual Inline Plastic Package.

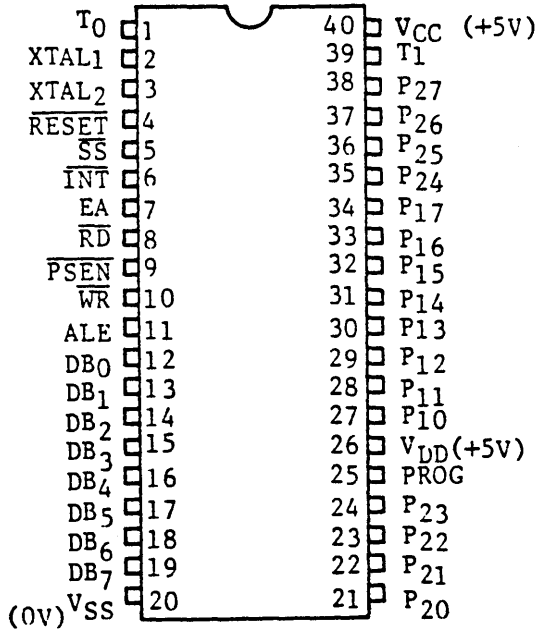
The TMP8048AT/TMP8035AT, TMP8049AT/TMP8039AT are packaged in the JEDEC standard type 44pin PLCC (Plastic Leaded Chip Carrier).

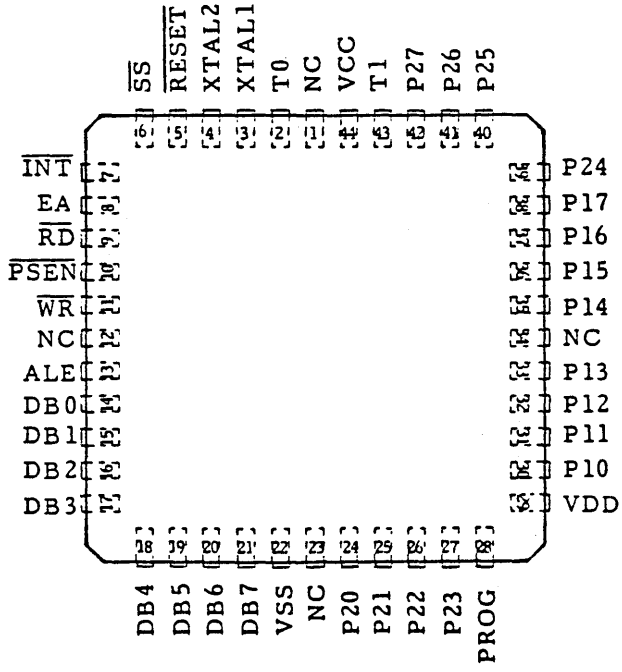
FEATURES

- 1.36 us Instruction Cycle
- All instruction 1 or 2 cycles
- Over 90 instructions; 70% single byte
- Easy expandable memory and I/O
- 27 I/O lines
- Interval Time/Event Counter
- Single level interrupt
- Single 5V supply

Parts Number	Program Memory (ROM)	Data Memory (RAM)
TMP8048A	1 k Byte	64 Byte
TMP8049A	2 k Byte	128 Byte
TMP8035A	-	64 Byte
TMP8039A	-	128 Byte

PIN CONNECTIONS (Top View)





PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VDD (Power Supply)
+5V during operation Low power standby pin for TMP8048A RAM

VCC (Main Power Supply)
+5V during operation

PROG (Output)
Output strobe for the TMP8243P I/O exapnder

P10-P17 (Input/Output) Port 1
8-bit quasi-bidirectional port (Internal Pullup = 50 kohm).

P20-P27 (Input/Output) Port 2
8-bit quasi-bidrectional port (Internal Pullup = 50 kohm).
P20-P23 Contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0-DB7 (Input/Output, 3 State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$ and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

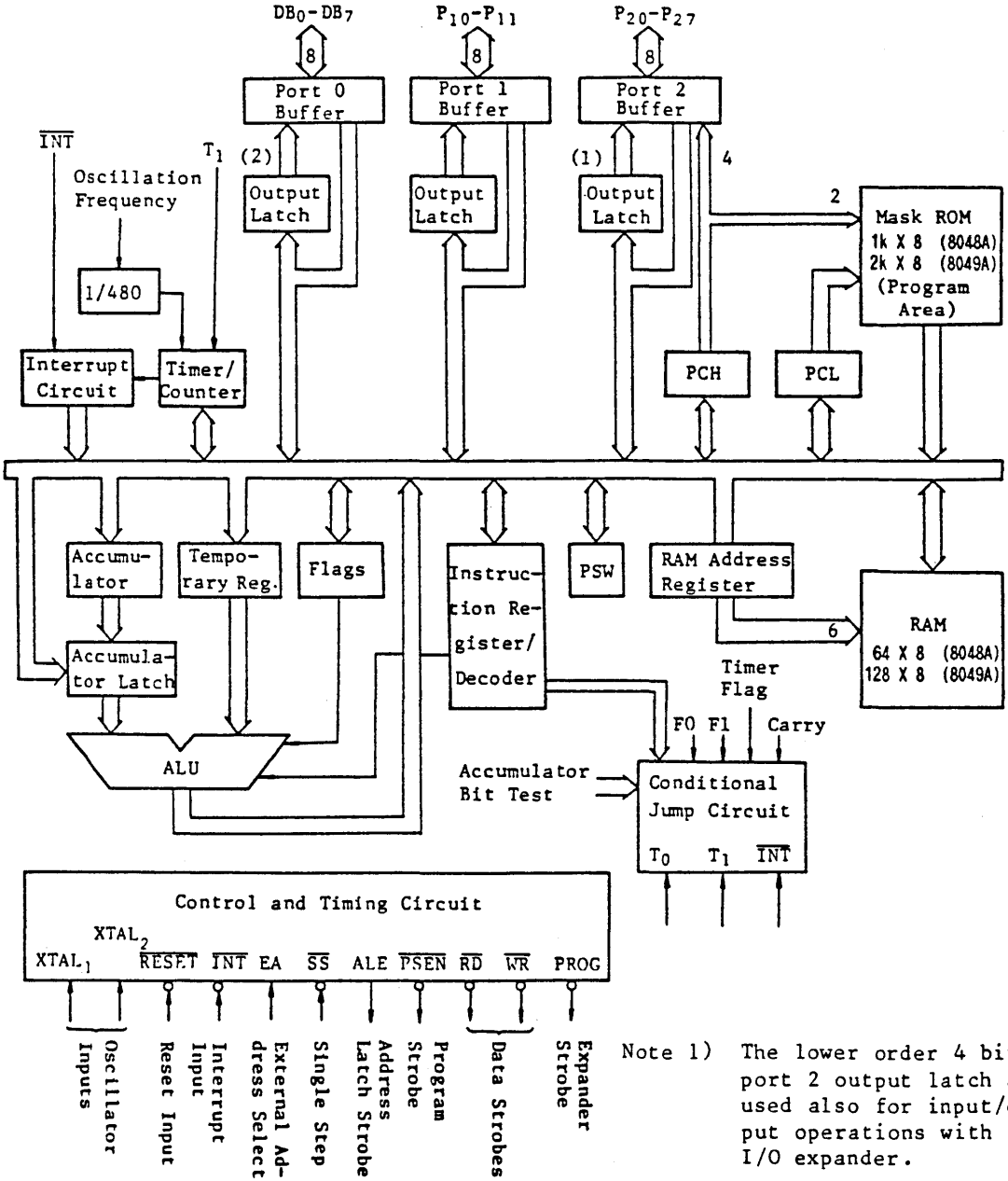
T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active Low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low). Used as a Write Strobe to External Data Memory.

BLOCK DIAGRAM



Note 1) The lower order 4 bits of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

$\overline{\text{RESET}}$ (Input)
Active Low signal which is used to initialize the Processor. Also used during Power down.

ALE (Output)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single Step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being excuted.

EA (Input)
External Access input which forces all program memory fetchs to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High).

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

FUNCTIONAL DESCRIPTION

1. System Configuration

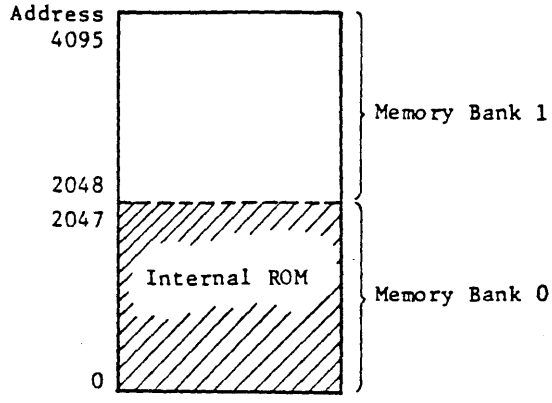
. The following system functions of the TMP8048A are described in detail.

(1)	Program Memory	(6)	Stack (Stack Pointer)
(2)	Data Memory	(7)	Flag 0, Flag 1
(3)	I/O Port	(8)	Program Status Word (PSW)
(4)	Timer/Counter	(9)	Reset
(5)	Interrupt Control Circuit	(10)	Oscillator Circuit

(1) Program Memory

. The maximum memory that can be directly addressed by the TMP8048A is 4096 bytes. The first 1024 bytes from location 0 through 1023 (TMP8048A) or the first 2048 bytes from location 0 to 2047 (TMP8049A) can be internal resident mask ROM. The rest of the 3072 bytes or the 2048 bytes of addressable memory are external to the chip. The TMP8035A and TMP8039A have has no internal resident memory; all memory must be external.

There are three locations in Program Memory of special importance.



Program Memory Area

. Location 0
Activating the Reset line of the processor causes the first instruction to be fetched from Location 0.

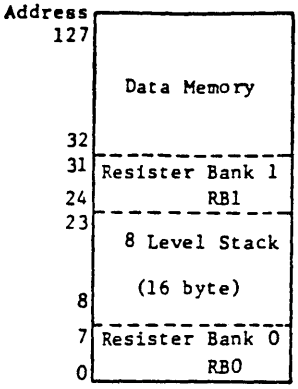
. Location 3
Activating the interrupt line of the processor (if interrupt enabled) causes a jump to subroutine defined by address held in Location 3.

. Location 7
A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

. Program address 0-2047 and 2048-4095 are called memory banks 0 and 1 respectively switching of memory banks is achieved by changing the most significant bit of the program counter (PC) during execution of an uncoditional jump instruction or call instruction executed after using SEL MB0 or SEL MB1.
Reset operation automatically selects Bank 0.

(2) Data Memory

. Resident Data Memory (volatile RAM) is organized as 64 words (TMP8048A) or 128 words (TMP8049A) by 8-bits wide.
. The first 8 locations (0 - 7) of the memory array are designed as working registers and are directly addressable by several instructions. By executing a Register Bank switch instruction (SEL RB1) locations 24 - 31 are designated as the working registers in palce of 0 - 7.



Internal Data Memory Area

- RAM locations 8 - 23 serve a dual role in that they contain the program counter stack which is a stack 2 bytes wide by 8 levels deep. These locations store returning addresses from subroutines. If the level of subroutine nesting is less than the permitted 8, you free up 2 bytes of RAM for general use for every level of nesting not utilized.

- All 64 (TMP8048A) or 128 (TMP8049A) locations are indirectly addressable through either of two RAM Pointer Registers which reside at R0 and R1 of the Register array.

- The TMP8048A architecture allows extension of the Data Memory to 256 words.

(3) Input/Output Ports

- The TMP8048A has 27 I/O lines which can be used for either input or output. These I/O lines are grouped into 3 ports each having 8 bidirectional lines and 3 "test" inputs which can after program sequences when tested by conditional jump instructions.

- Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction.

- All lines of Ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure (illustrated in Figure 1). Each line is continuously pulled to a +5V level through a high impedance resistive device (50kohm) which is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. In order to speed up the "0" to "1" transition a low impedance device (5kohm) is switched in momentarily whenever a "1" is written to line. When "0" is written to line a low impedance device overecomes the pullup and provides TTL current sinking capability.

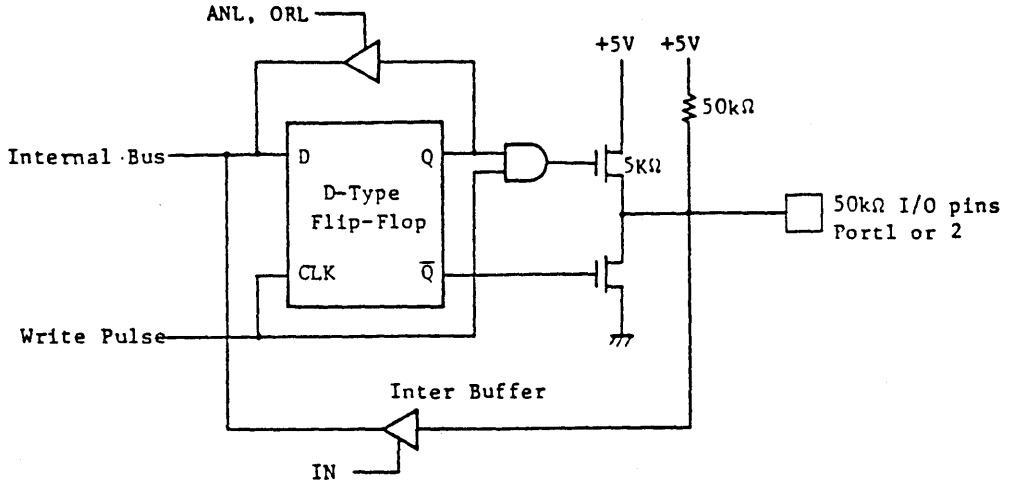


Fig.1 Input/Output Circuit of Port 1, Port 2

- Reset initializes all lines to a high impedance "1" state.

- When external data memory area is not addressed during execution of an internal program, Port 0 (DB0 - DB7) becomes a true bidirectional port (bus) with associated input and output strobes. If bidirectional feature not needed Bus can serve as either a statically latched output port or a non-latched input port. However, I/O lines of this port cannot be intermixed.

- As a static port data is written and latched using the OUTL instruction and inputted using the INS instruction these two commands generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ strobe lines.

- As a bidirectional port the MOVX instructions are used to read and write the port which generate the $\overline{RD}$ and $\overline{WR}$ strobes.

- When not being written or read, the Bus lines are in a high impedance state.

(4) Timer/Event Counter

- The 8-bit binary up counter can use either of the following frequency inputs

  (1) Internal clock (1/480 of OSC frequency)
  ..... Timer mode

(2) External input clock form T1 terminal
(minimum cycle time 3 x ALE cycle)
..... Event Counter mode

The counter is presettable and readable with two MOV instructions which transfer the content of the accumulator to the counter and vice versa. The counter content is not affected by a Reset and is initialized solely by the MOVT, A instruction. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started by STRT T instruction or as an event counter by a STRT CNT. One started the counter will increment to its maximum count (FF) and overflow to Zero continuing its count until stopped by a STOP TCNT instruction or RESET.

The increment from maximum count to Zero (overflow) results in the setting of an overflow flag and the generation of an interrupt request. When interrupt acknowledged a subroutine call to Location 7 will be initiated. Location 7 should store the starting address of the timer or counter service routine. The state of the overflow flag is testable with the conditional Jump (JTF). The flag is reset by excuting a JTF or by RESET.

Figure 2 illustrates the concept of the timer circuit.

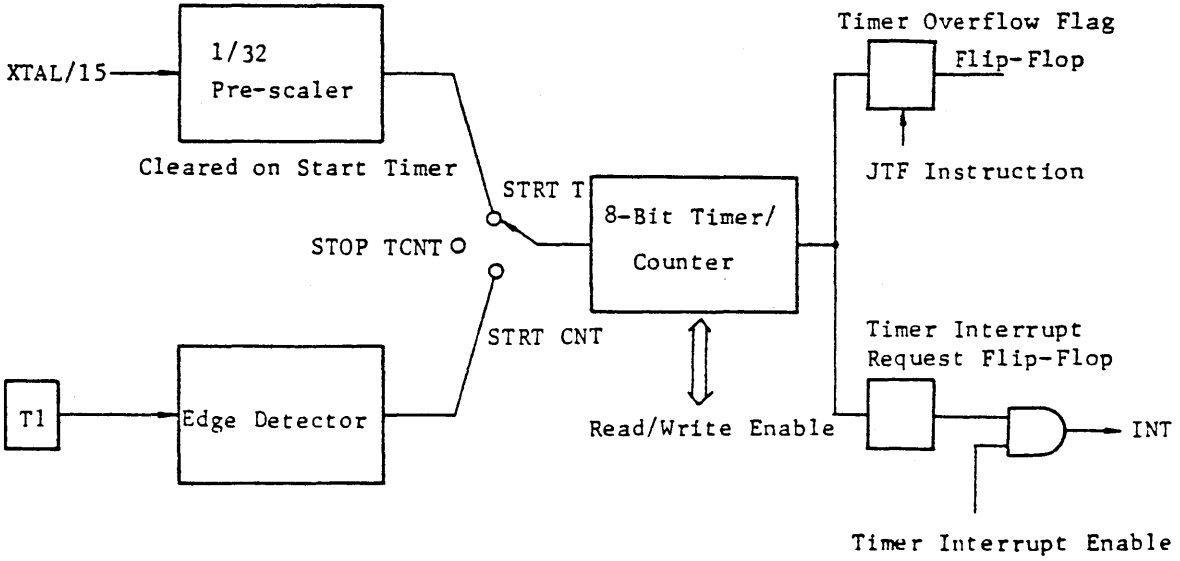


Fig. 2 Concept of Timer Circuit

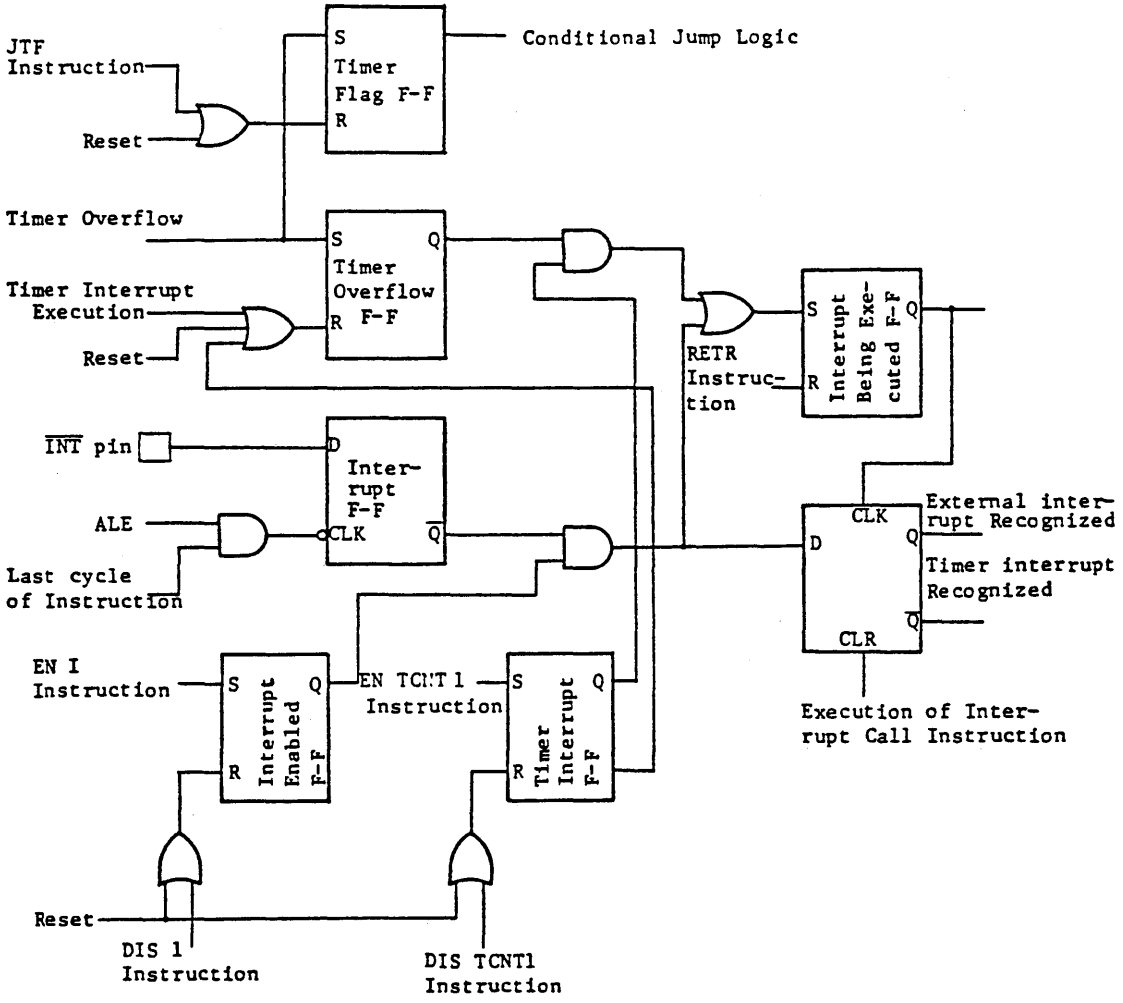


Fig. 3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

. There are two distinct types of Interrupts in the TMP8048A.

(1) External Interrupt from the $\overline{\text{INT}}$ terminal

(2) Timer Interrupt caused by timer overflow

. The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR (which should occur at the end of an interrupt service routine) reenables the interrupt logic.

. An interrupt sequence is initiated by applying a low level "0" to the $\overline{INT}$ pin. $\overline{INT}$ is level triggered and active low which allows "Wire Oring" of several interrupt sources. The interrupt level is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at Location 3. As in any call to subroutine, the Program Counter and Program Status Word are saved in the stack.

. When an overflow occurs in the internal timer/event counter an interrupt request is generated which is reserviced as outlined in previous paragraph except that a jump to Location 7 is used instead of 3. If $\overline{INT}$ and times overflow occur simultaneously then external request $\overline{INT}$ takes precedence.

. If an extra external interrupt is needed in addition to $\overline{INT}$ this can be achieved by enabling the counter interrupt, loading FFH in the counter (one less than the terminal count), and enabling the event counter mode. A "1" to "0" transition on T1 will cause an interrupt vector to Location 7.

. The interrupt service routine pointed to be addresses in Location 3 or 7 must reside in memory between 0 and 2047, i.e., Bank 0.

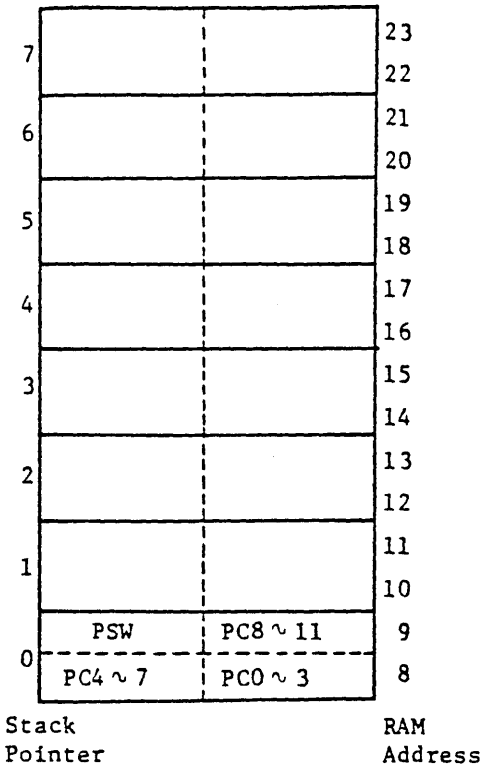
Figure 3 illustrates the concept of the interrupt control circuit.

(6) Stack (Stack Pointer)

. An interrupt or Call to subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit stack pointer which is part of the Program Status Words (PSW explained in section (8)). Data RAM locations, 8 through 23 are available as stack registers and are used to store the program counter and 4-bits of PSW as shown in the figure.

. The stack pointer when initialized points to RAM location 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to Locations 8 and 9. Then the stack pointer is incremented by one to point to Locations 10 and 11. Eight levels of subroutine are obviously possible.

. At the end of a subroutine signalled by a RET or RETR causes the stack pointer to be decremented by one and the contents of the resulting pair to be transferred to the Program Counter.



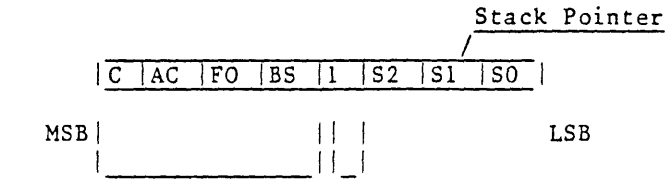
(7) Flag 0, Flag 1 (F0, F1)

. The TMP8048A has two flags F0 and F1 which are used for conditional jump. These flags can be set, reset and tested with the conditional jump instruction JF0.

. F0 is a part of the program status word (PSW) and is saved in the stack area when a subroutine is called.

(8) Program Status Word (PSW)

. An 8-bit status word which can be loaded to and from the accumulator exists called the Program Status Word (PSW). The PSW is read by a MOV A, PWS and written to by a MOV PSW, A. The information available in the PSW is shown in the diagram below.



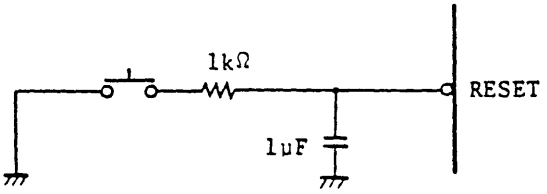
Bits 0 - 2 : Stack Pointer Bits (S0, S1, S2)
Bit 3 : Not used ("1" level when read.)
Bit 4 : Working Register Bank Switch Bit (BS)

0 = Bank 0
1 = Bank 1

Bit 5 : Flag 0 (F0)
Bit 6 : Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA, A (AC)
Bit 7 : Carry (C) flag which indicates that the previous operation has resulted in the accumulator. (C)

(9) Reset

. The reset input provides a means for initialization of the processor. This Schmitt trigger input has an internal pullup register which in combination with an external 1uF capacitor provides an internal reset pulse sufficient length to guarantee that all internal logic is initialized.



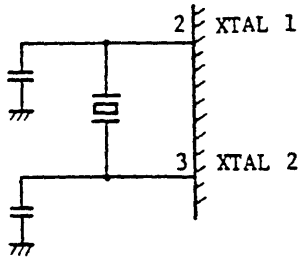
If the pulse is generated externally the reset pin must be held at ground ($\leq$0.8V) for at least 10mS after the power supply is within tolerance.

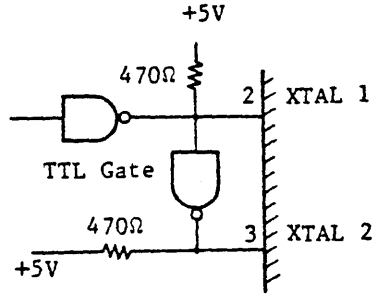
. Reset performs the following functions within the chip:

(i) Sets PC to Zero.
(ii) Sets Stack Pointer to Zero.
(iii) Selects Register Bank 0.
(iv) Selects Memory Bank 0.
(v) Sets BUS (DB0 - DB7) to high impedance state. (Except when EA=5V)
(vi) Sets Ports 1 and 2 to input mode.
(vii) Disables interrupts (timer and external).
(viii) Stops Timer.
(ix) Clears Timer Flag.
(x) Clears F0 and F1.
(xi) Disables clock output from T0.

(10) Oscillator Circuit

. TMP8048A can be operated by the external clock input in addition to crystal oscillator as shown below.





(a) Crystal Parameters and External Capacitance

The frequency of the oscillator will be calculated from the following formula.

$$f = (1+C0/2(CL+C))/2\pi\sqrt{L \times C0}$$

Load Capacitance CL

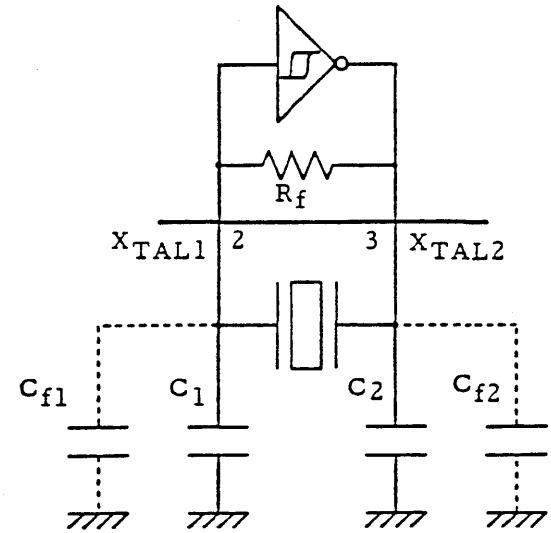
$CL=(C1+Cf1)(C2+Cf2)/((C1+Cf1)+(C2+Cf2))$

Cf1 : Input Capacitance (4pF Typ.)+Stray Capacitance (less than 5pF)

Cf2 : Output Capacitance(6pF Typ.)+Stray Capacitance (less than 5pF)

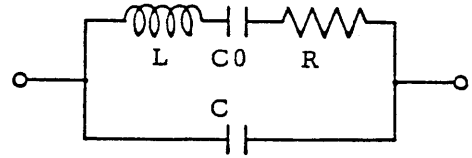
However the recommended value in the following table will be used better by the reason of the start of the oscillation will depend on the Equivalent Series Resistance Rl and the External Capacitances Cl+Cfl, C2+Cf2.

Frequency f(MHz)	Equivalent Series Res. R(ohm)	External Capacitance Cl=C2(pF) Recommended Value	Typical Allowance
11	25 Max.	10	5 to 17
11	30 Max.	10	5 to 15
10	25 Max.	10	5 to 20
10	30 Max.	10	5 to 17
8	30 Max.	15	5 to 25
8	40 Max.	10	5 to 20
6	40 Max.	20	5 to 35
6	80 Max.	10	5 to 17
5	50 Max.	20	5 to 40
5	80 Max.	15	5 to 25
4	100 Max.	15	5 to 30
4	150 Max.	10	5 to 20
3	100 Max.	20	5 to 40
3	200 Max.	15	5 to 25
2	400 Max.	25	5 to 40
2	500 Max.	15	5 to 25
1	800 Max.	25	10 to 40



(b) Ceramic Resonator and External Capacitance

Frequency f(MHz)	External Capacitance Recommended Value Cl=C2 (pF)
3 to 11	33
1 to 3	100



2. Basic Operation and Timing

The following basic operations and timing are explained

(1) Instruction Cycle
(2) External Memory Access Timing
(3) Interface with I/O Expander TMP8243P
(4) Internal Program Verify (Read) Timing
(5) Single Step Operation Timing
(6) Low Power Stand-by Mode

(1) Instruction Cycle

. The instruction of TMP8048A are executed in one or two machine cycles, and one machine cycle contents of five states.

. Fig. 4 illustrates its relationship with the clock input to CPU.

. $\phi 2$ clock shown in Fig. 4 is derived to outside by ENTO CLK instruction.

. ALE can be also used as the clock to indicate the machine cycle as well as giving the external address latch timing.

(2) External Memory Access Timing

(i) Program Memory Access

. TMP8048A programs are executed in the following three modes.

(1) Execution of internal program only.
(2) Execution of both external and internal programs.
(3) Execution of external program only.

The external program memory is accessed (instructions are fetched) automatically when the internal ROM address is exceeded in mode (2) and from initial start address 0 in mode (3).

. In the external program memory access operation, the following will occur
  . The contents of the 12-bit program counter will be output on BUS(DB0 - DB7) and the lower 4-bits of Port 2.
  . Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.
  . Program Store Enable ($\overline{\text{PSEN}}$) indicates that an external instruction fetch is in progress and serves to enable the external memory device.
  . Bus (DB0 - DB7) reverts to Input mode and the processor accepts its 8-bit contents as an Instruction Word.

. Figure 5 illustrates the timing.

(ii) Access of External Data Memory

. In the extended data memory access operation during READ/WRITE cycle the following occurs
  . The contents of R0 R1 is output onto BUS (DB0 - DB7).
  . ALE indicates address is valid. The trailing edge of ALE is used to latch the address externally.
  . A read $\overline{\text{RD}}$ or write $\overline{\text{WR}}$ pulse on the corresponding output pins indicates the type of data memory access in progress. Output data valid at trailing edge of $\overline{\text{WR}}$ and input data must be valid at trailing edge of $\overline{\text{RD}}$.
  . Data (8-bits) is transferred over BUS.

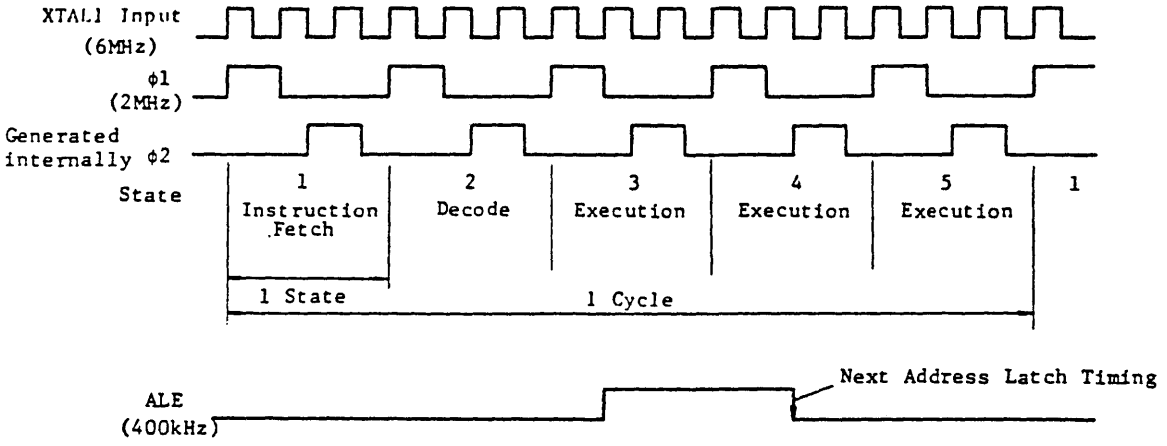


Fig. 4 Instruction Cycle Timing

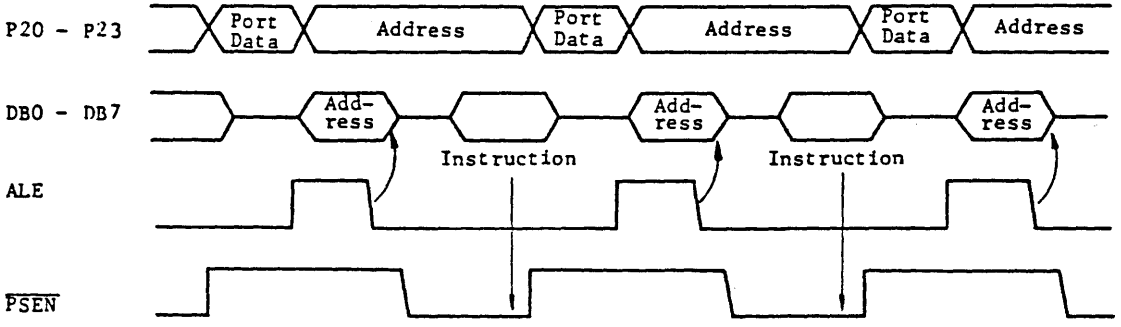
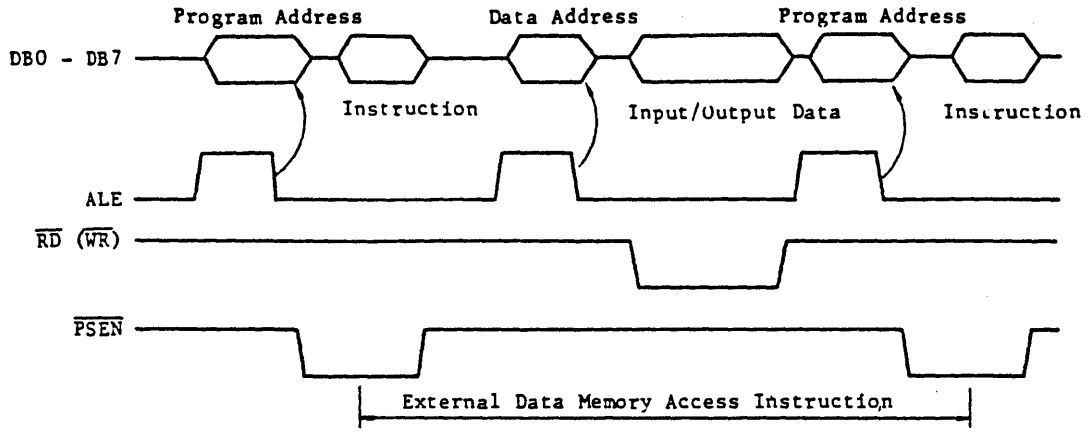


Fig. 5 Timing of External Program Memory Access



Suggest we have two diagrams

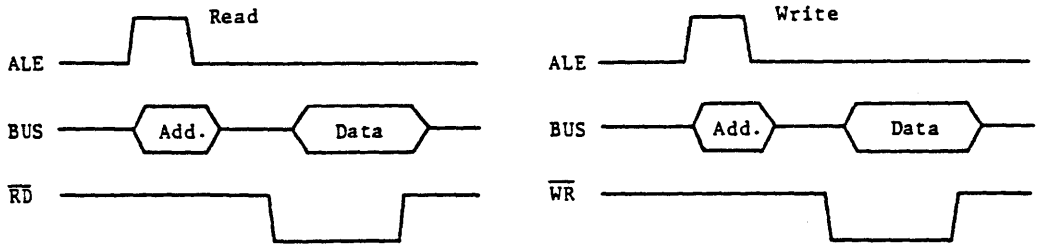


Fig. 6 Timing of Accessing External Data Memory

. Figure 6 illustrates the timing of accessing the external data memory during execution of external program.

(3) Interface with I/O Expander (TMP8243P)

. The TMP8048A I/O can be easily expanded using the TMP8243 I/O Expander. This device uses only the lower half 4-bits of Port 2 for communication with the TMP8048A. The TMP8243 contains for 4-bit I/O ports which serve as extensions of one chip I/O and are addressed as Ports (4-7). All communication takes place over the lower half of port 2 (P20 - P23) with timing provided by an output pulse on the PROG pin. Each transfer consists of two 4-bit nibbles the first containing the "OP Code" and port address and second containing the actual 4-bits of data.

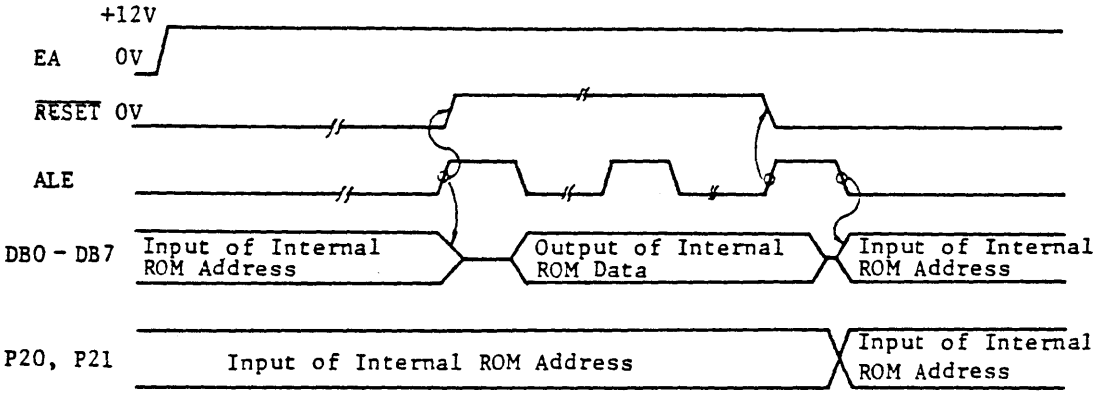


Fig. 7 Timing of Reading Internal Program Memory

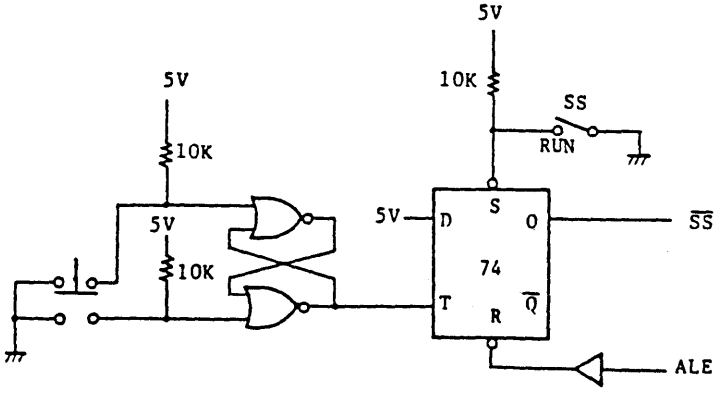


Fig. 8(a) Single Step Circuit

Reading of Internal Program Memory

- The processor is placed in the READ mode by applying +12V to the EA pin and 0V to the RESET pin. The address of the location to be read is then applied to BUS and the low order 2 or 3 bits of Port 2. The address is latched by a 0 to 1 transition on RESET and the high level causes the contents of program memory location addressed to appear on the eight lines of BUS.

- Figure 7 illustrates the timing diagram for this operation.

(5) Single Step Operation.

- A single step feature useful for debug can be implemented by utilizing a circuit shown in Figure 8 (a) combined with the SS pin and ALE pin.

- A D-type flip flop with set and reset is used to generate SS. In the run mode SS is held high by keeping the flip flop set. To enter single step, set is removed allowing ALE to bring SS low via reset input. The next instruction is started by clocking a "1" into the FF which will not appear on SS unless ALE is high removing reset. In response to SS going high the processor begins an instruction fetch which brings ALE low resetting FF and causing the processor to again enter the stopped state.

- The timing diagram in this case is as shown in Figure 8 (b). (EA=5V)

(6) Lower Power Stand-by Mode

- The Lower TMP8048A has been organized to allow power to be removed from all but the volatile, 64 x 8 or 128 x 8 data RAM array. In power down mode the contents of data RAM can be maintained while drawing typically 10 - 15% of normal operating power requirements.

- VCC serves as the 5V supply for the bulk of the TMP8048A while the VDD supplies only the RAM array. In standby mode VCC is reduced to 0V but VDD is kept at 5V. Applying a low level to reset inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from VCC.

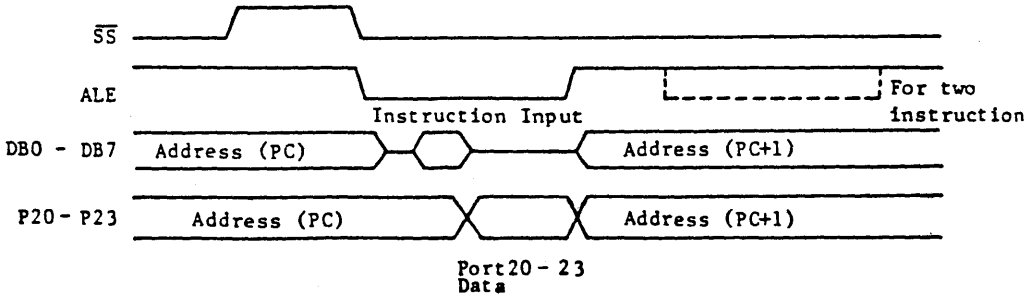


Fig. 8(b) Single Step Operation Timing

INSTRUCTION

ACCUMULATOR INSTRUCTION

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
ADD A,Rr	0	1	1	0	1	r	r	r	(A)<-(A)+(Rr) r = 0 - 7	1	1	o	o
ADD A,@Rr	0	1	1	0	0	0	0	r	(A)<-(A)+((Rr)) r = 0,1	1	1	o	o
ADD A,#Data	0	0	0	0	0	0	1	1	(A)<-(A)+Data	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ADDC A,Rr	0	1	1	1	1	r	r	r	(A)<-(A)+(Rr)+(C) r = 0 - 7	1	1	o	o
ADDC A,@Rr	0	1	1	1	0	0	0	r	(A)<-(A)+((Rr))+(C) r = 0,1	1	1	o	o
ADDC A,#Data	0	0	0	1	0	0	1	1	(A)<-(A)+Data+(C)	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ANL A,Rr	0	1	0	1	1	r	r	r	(A)<-(A)and(Rr) r = 0 - 7	1	1	-	-
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)<-(A)and((Rr)) r = 0,1	1	1	-	-
ANL A,#Data	0	1	0	1	0	0	1	1	(A)<-(A)and Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
ORL A,Rr	0	1	0	0	1	r	r	r	(A)<-(A)or(Rr) r = 0 - 7	1	1	-	-
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)<-(A)or((Rr)) r = 0,1	1	1	-	-
ORL A,#Data	0	1	0	0	0	0	1	1	(A)<-(A)or Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
XRL A, Rr	1	1	0	1	1	r	r	r	(A)<-(A)Eor(Rr) r = 0 - 7	1	1	-	-
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)<-(A)Eor((Rr)) r = 0,1	1	1	-	-
XRL A,#Data	0	1	0	0	0	0	1	1	(A)<-(A)Eor Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
INC A	0	0	0	1	0	1	1	1	(A)<-(A)+1	1	1	-	-
DEC A	0	0	0	0	0	1	1	1	(A)<-(A)-1	1	1	-	-
CLR A	0	0	1	0	0	1	1	1	(A)<-0	1	1	-	-
CPL A	0	0	1	1	0	1	1	1	(A)<-NOT (A)	1	1	-	-
DA A	0	1	0	1	0	1	1	1	Decimal Adjust Accumulator	1	1	o	-
SWAP A	0	1	0	0	0	1	1	1	(A4-7)->(A0-3) <-	1	1	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
RL A	1	1	1	0	0	1	1	1	(An+1)<-(An)	1	1	-	-
									n = 0 - 6				
RLC A	1	1	1	1	0	1	1	1	(An+1)<-(An)	1	1	-	-
									n = 0 - 6				
									(C)<-(A7)				
									(A0)<-(C)				
RR A	0	1	1	1	0	1	1	1	(An)<-(An+1)	1	1	-	-
									n = 0 - 6				
									(A7)<-(A0)				
RRC A	0	1	1	0	0	1	1	1	(An)<-(An+1)	1	1	-	-
									n = 0 - 6				
									(C)<-(A0)				
									(A7)<-(C)				

Input/Output Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
IN A,Pp	0	0	0	0	1	0	P	P	(A)<-(Pp)	1	2	-	-
									p = 1, 2				
OUTL Pp,A	0	0	1	1	1	0	P	P	(Pp)<-(A)	1	2	-	-
									p = 1, 2				
ANL Pp,#Data	1	0	0	1	1	0	P	P	(Pp)<-(Pp)and Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	p = 1, 2				
ORL Pp,#Data	1	0	0	0	1	0	P	P	(Pp)<-(Pp)or Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	p = 1, 2				
INS A,BUS	0	0	0	0	1	0	0	0	(A)<-(BUS)	1	2	-	-
OUTL BUS,A	0	0	0	0	0	0	1	0	(BUS)<-(A)	1	2	-	-
ANL BUS,#Data	1	0	0	1	1	0	0	0	(BUS)<-	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	(BUS)and Data				
ORL BUS,#Data	1	0	0	0	1	0	0	0	(BUS)<-	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	(BUS) or Data				
MOVD A,Pp	0	0	0	0	1	1	P	P	(A0-3)<-(Pp)	1	2	-	-
									(A4-7)<- 0				
									p = 4 - 7				
MOVD Pp,A	0	0	1	1	1	1	P	P	(Pp)<-(A0-3)	1	2	-	-
									p = 4 - 7				
ANLD Pp,A	1	0	0	1	1	1	P	P	(Pp)<-	1	2	-	-
									(Pp)and(A0-3)				
									p = 4 - 7				
ORLD Pp,A	1	0	0	0	1	1	P	P	(Pp)<-	1	2	-	-
									(Pp)or (A0-3)				
									P = 4 - 7				

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
INC Rr	0	0	0	1	1	r	r	r	(Rr)<--(Rr)+1 r = 0 - 7	1	1	-	-
INC @Rr	0	0	0	1	0	0	0	r	((Rr))<-((Rr))+1 r = 0, 1	1	1	-	-
DEC Rr	1	1	0	0	1	r	r	r	(Rr)<--(Rr)-1 r = 0 - 7	1	1	-	-

Branch Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JMP Address	a10	a9	a8	0	0	1	0	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC8-10)<--(a8-10				
									(PC11)<--DBF				
JMPP @A	1	0	1	1	0	0	1	1	(PC0-7)<--((A))	1	2	-	-
DJNZ Rr	1	1	1	0	1	r	r	r	(Rr)<--(Rr)-1	2	2	-	-
Address	a7	a6	a5	a4	a3	a2	a1	a0	if Rr not 0				
									(PC0-7)<--(a0-7)				
JC Address	1	1	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if C = 1				
									(PC)<--(PC)+2				
									if C = 0				
JNC Address	1	1	1	0	0	1	1	0	(PC0-7)<-(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if C = 0				
									(PC)<--(PC)+2				
									if C = 1				
JZ Address	1	1	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if (A) = 0				
									(PC)<--(PC)+2				
									if (A).NEQ.0				
JNZ Adress	1	0	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if (A).NEQ.0				
									(PC)<--(PC)+2				
									if (A) = 0				
JT0 Adress	0	0	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T0 = 1				
									(PC)<--(PC)+2				
									if T0 = 0				
JNT0 Address	0	0	1	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T0 = 0				
									(PC)<--(PC)+2				
									if T0 = 1				
JT1 Address	0	1	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T1 = 1				
									(PC)<--(PC)+2				
									if T1 = 0				
JNT1 Address	0	1	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T1 = 0				
									(PC)<--(PC)+2				
									if T1 = 1				

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JF0 Address	1	0	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F0 = 1				
									(PC)<--(PC)+2				
									if F0 = 0				
JF1 Address	0	1	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F1 = 1				
									(PC)<--(PC)+2				
									if F1 = 0				
JTF Address	0	0	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if TF = 1				
									(PC)<--(PC)+2				
									if TF = 0				
JNI Address	1	0	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if INT = 0				
									(PC)<--(PC)+2				
									if INT = 1				
JBb Address	b2	b1	b0	1	0	0	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if (Bb) = 1				
									(PC)<--(PC)+2				
									if (Bb) = 0				
									(b = 0 - 7)				
CALL Address	a10	a9	a8	1	0	1	0	0	((SP))<--	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC),(PSW4-7)				
									(SP)<--(SP)+1				
									(PC8-10)<--(a8-10)				
									(PC0-7)<--(a0-7)				
									(PC11)<--DBF				
RET	1	0	0	0	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
RETR	1	0	0	1	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
									(PSW4-7)<--((SP))				

Flag Manipulation Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
CLR C	1	0	0	1	0	1	1	1	(C)<-- 0	1	1	0	-
CPL C	1	0	1	0	0	1	1	1	(C)<--NOT(C)	1	1	0	-
CLR F0	1	0	0	0	0	1	0	1	(F0)<-- 0	1	1	-	-
CPL F0	1	0	0	1	0	1	0	1	(F0)<--NOT(F0)	1	1	-	-
CLR F1	1	0	1	0	0	1	0	1	(F1)<-- 0	1	1	-	-
CPL F1	1	0	1	1	0	1	0	1	(F1)<--NOT(F1)	1	1	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A, Rr	1	1	1	1	1	r	r	r	(A)<-- (Rr) r = 0 - 7	1	1	-	-
MOV A, @Rr	1	1	1	1	0	0	0	r	(A)<-- ((Rr)) r = 0, 1	1	1	-	-
MOV A, #Data	0	0	1	0	0	0	1	1	(A)<-- Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
MOV Rr, A	1	0	1	0	1	r	r	r	(Rr)<--(A) r = 0 - 7	1	1	-	-
MOV@Rr,A	1	0	1	0	0	0	0	r	((Rr))<-(A) r = 0, 1	1	1	-	-
MOV Rr,#Data	1	0	1	1	1	r	r	r	(Rr)<--Data r = 0 - 7	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
MOV@Rr,#Data	1	0	1	1	0	0	0	r	((Rr))<--Data r = 0, 1	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
MOV A,PSW	1	1	0	0	0	1	1	1	(A)<-- (PSW)	1	1	-	-
MOV PSW, A	1	1	0	1	0	1	1	1	(PSW)<--(A)	1	1	-	-
XCH A, Rr	0	0	1	0	1	r	r	r	(A)-->(Rr) <-- r = 0 - 7	1	1	-	-
XCH A,@Rr	0	0	1	0	0	0	0	r	(A)-->((Rr)) <-- r = 0, 1	1	1	-	-
XCHD A,@Rr	0	0	1	1	0	0	0	r	(A0-3)-->((Rr0-3)) <-- r = 0, 1	1	1	-	-
MOVX A,@Rr	1	0	0	0	0	0	0	r	(A)<--((Rr)) r = 0, 1	1	2	-	-
MOVX @Rr,A	1	0	0	1	0	0	0	r	((Rr))<--(A) r = 0, 1	1	2	-	-
MOVP A,@A	1	0	1	0	0	0	1	1	(PC0-7)<--(A) (A)<--((PC))	1	2	-	-
MOVP3 A,@A	1	1	1	0	0	0	1	1	(PC0-7)<--(A) (PC8-11)<--0011 (A)<--((PC))	1	2	-	-

Timer/Counter Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A,T	0	1	0	0	0	0	1	0	(A)<--(T)	1	1	-	-
MOV T,A	0	1	1	0	0	0	1	0	(T)<--(A)	1	1	-	-
STRT T	0	1	0	1	0	1	0	1	Counting is started in the timer mode	1	1	-	-
STRT CNT	0	1	0	0	0	1	0	1	Counting is started in the event counter mode	1	1	-	-
STOP TCNT	0	1	1	0	0	1	0	1	Stop both time accumulation and event counting	1	1	-	-
EN TCNTI	0	0	1	0	0	1	0	1	Timer interrupt is enabled	1	1	-	-
DIS TCNTI	0	0	1	1	0	1	0	1	Timer interrupt is disabled	1	1	-	-

Control Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
EN I	0	0	0	0	0	1	0	1	External interrupt is enabled	1	1	-	-
DIS I	0	0	0	1	0	1	0	1	External interrupt is disabled	1	1	-	-
SEL RB0	1	1	0	0	0	1	0	1	(BS)<-- 0	1	1	-	-
SEL RB1	1	1	0	1	0	1	0	1	(BS)<-- 1	1	1	-	-
SEL MB0	1	1	1	0	0	1	0	1	(DBF)<-- 0	1	1	-	-
SEL MB1	1	1	1	1	0	1	0	1	(DBF)<-- 1	1	1	-	-
ENTO CLK	0	1	1	1	0	1	0	1	T0 is enabled to act as the clock output	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-

SYMBOL	ITEM	RATTING
VDD	VDD Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to +7V
VINB	Input Voltage (Only EA)	-0.5V to +13V
PD	Power Dissipation (Ta = 70°C)	1.5W
TSOLDER	Soldering Temperature (Soldering Time 10 sec)	260°C
TSTG	Storage Temperature	-55°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS

T_A=0°C to 70°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage		-0.5	-	0.8	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{\text{RESET}}$)		2.0	-	VCC	V
VIH1	Input High Voltage (XTAL1, XTAL2, $\overline{\text{RESET}}$)		3.8	-	VCC	V
VOL	Output Low Voltage (BUS)	IOL=2.0mA	-	-	0.45	V
VOL1	Output Low Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	IOL=1.8mA	-	-	0.45	V
VOL2	Output Low Voltage (PROG)	IOL=1.0mA	-	-	0.45	V
VOL3	Output Low Voltage (For other output pins)	IOL=1.6mA	-	-	0.45	V
VOH	Output High Voltage (BUS)	IOH=-400uA	2.4	-	-	V
VOH1	Output High Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	IOH=-100uA	2.4	-	-	V
VOH2	Output High Voltage (For other output pins)	IOH=-40uA	2.4	-	-	V
ILI	Input Leak Current (T1, $\overline{\text{INT}}$)	VSS ≦ VIN ≦ VCC	-	-	±10	uA
ILI1	Input Leak Current (P10-17, P20-27, EA, $\overline{\text{SS}}$)	VSS+0.45≦VIN≦ VCC			-500	uA
ILO	Output Leak Current (BUS, T0) (High impedance condition	VSS+0.45≦VIN≦ VCC			±10	uA

DC CHARACTERISTICS

T_A=0°C to 70°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL	PARAMETER	TMP8048A/ TMP8035A			TMP8049A/ TMP8039A			UNIT
		MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
IDD	VDD Supply Current	-	10	15	-	15	20	mA
ICC+IDD	Total Supply Current	-	65	90	-	85	120	mA

AC CHARACTERISTICS

T_A=0°C to 70°C, V_{CC}=V_{DD}=+5V±10%, V_{SS}=0V, Unless Otherwise Noted.

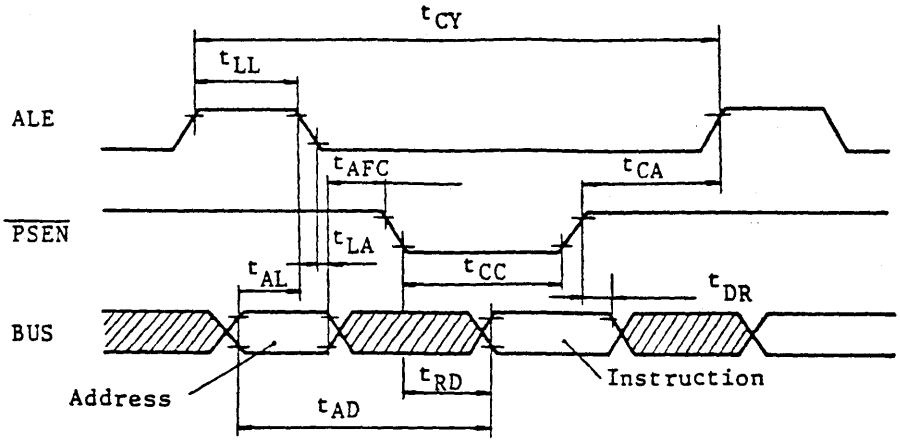
SYMBOL	PARAMETER	TEST CONDITION	f (t)	11 MHz		UNIT
				MIN.	MAX.	
t	Clock Cycle Time		1/xtal f	90	1000	ns
tLL	ALE Pulse Width		3.5t-170	150	-	ns
tAL	Address Setup Time (ALE)		2t-110	70	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	50	-	ns
tCC1	Control Pulse Width ($\overline{RD}$,$\overline{WR}$)		7.5t-200	480	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	350	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	390	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	40	-	ns
tDR	Data Hold Time ($\overline{PSEN}$,$\overline{RD}$)	CL=20pF	1.5t-30	0	110	ns
tRD1	Data Input Read Time ($\overline{RD}$)		6t-170	-	375	ns
tRD2	Data Input Read Time($\overline{PSEN}$)		4.5t-170	-	240	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	300	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10.5t-220	-	730	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7.5t-200	-	460	ns
tAFC1	Address Float Time ($\overline{RD}$)	CL=20pF	2t-40	140	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	10	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$,$\overline{WR}$)		3t-75	200	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	60	-	ns
tCA1	Control to ALE Time ($\overline{RD}$,$\overline{WR}$,PROG)		t-65	25	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	290	-	ns
tCY	Cycle Time		15t	1.36	15.0	us
tCP	Port Control Setup Time (PROG)		1.5t-80	50	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	100	-	ns
tPR	Port 2 Input Data Set Time (PROG)		8.5t-120	-	650	ns

SYMBOL	PARAMETER	TEST CONDITION	f (t)	11 MHz		UNIT
				MIN.	MAX.	
tDP	Port2 Output Data Setup Time (PROG)		6t-290	250	-	ns
tPD	Port2 Output Data Hold Time (PROG)		1.5t-90	40	-	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	140	ns
tPP	PROG Pulse Width		10.5t-250	700	-	ns
tPL	Port2 I/O Data Setup Time(ALE)		4t-200	160	-	ns
tLP	Port2 I/O Data Hold Time (ALE)		0.5t-30	15	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	510	ns
tOPRR	Output Clock Cycle Time (TO)		3t	270	-	ns

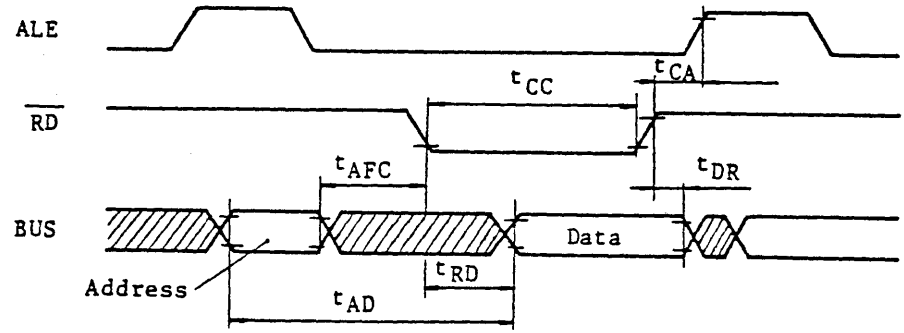
1. Control Outputs : CL=80pF, BUS Output : CL=150pF.
2. Address Float Time: BUS Hi-Impedance, CL=20pF
3. f(t): Assume the 50% duty clock is inputed to X1, X2.

TIMING WAVEFORM

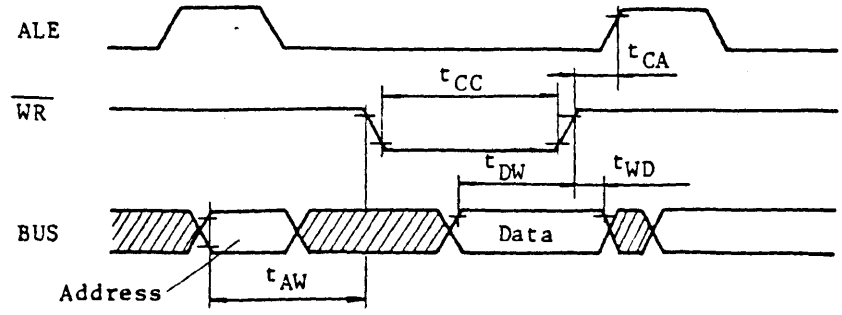
A. Instruction Fetch from External Program Memory



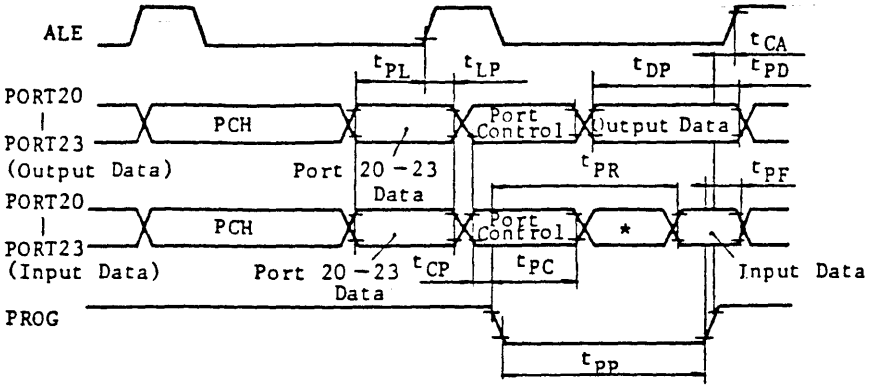
B. Read from External Data Memory



C. Write into External Data Memory

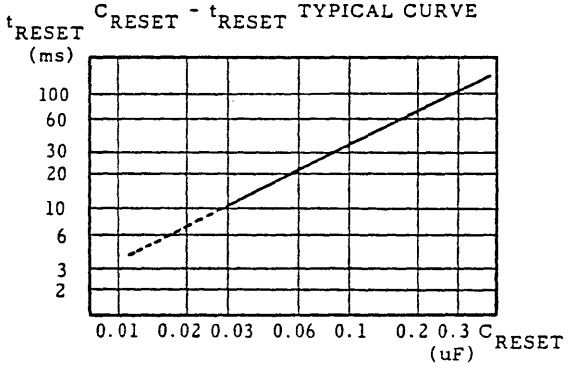


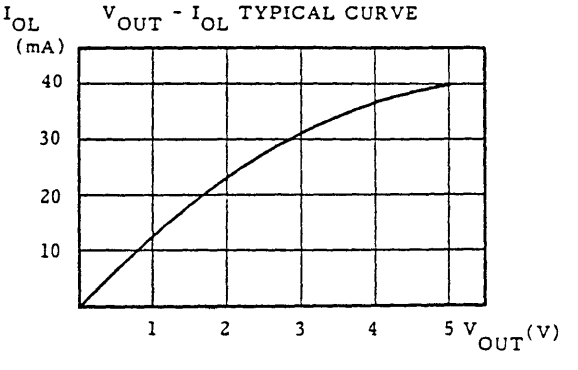
D. Timing of Port 2 during Expander Instruction Execution

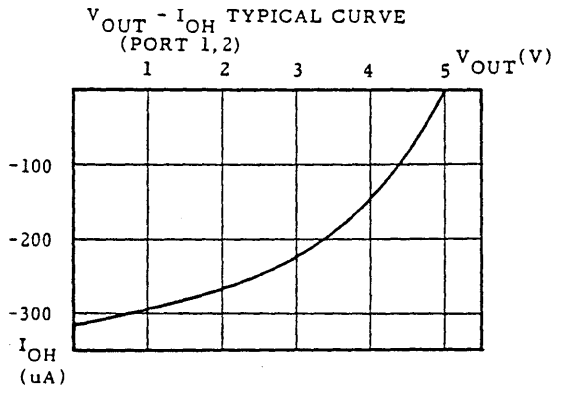


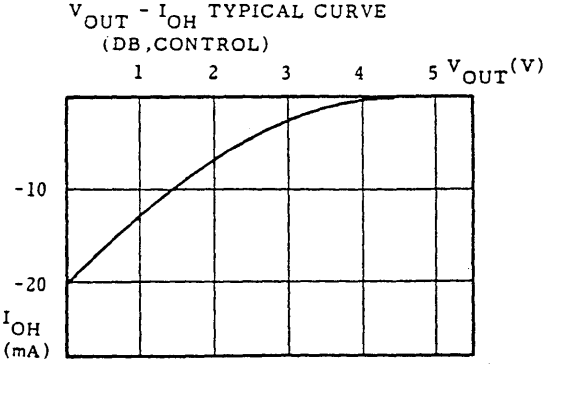
* Input Enabled State

TYPICAL CHARACTERISTICS





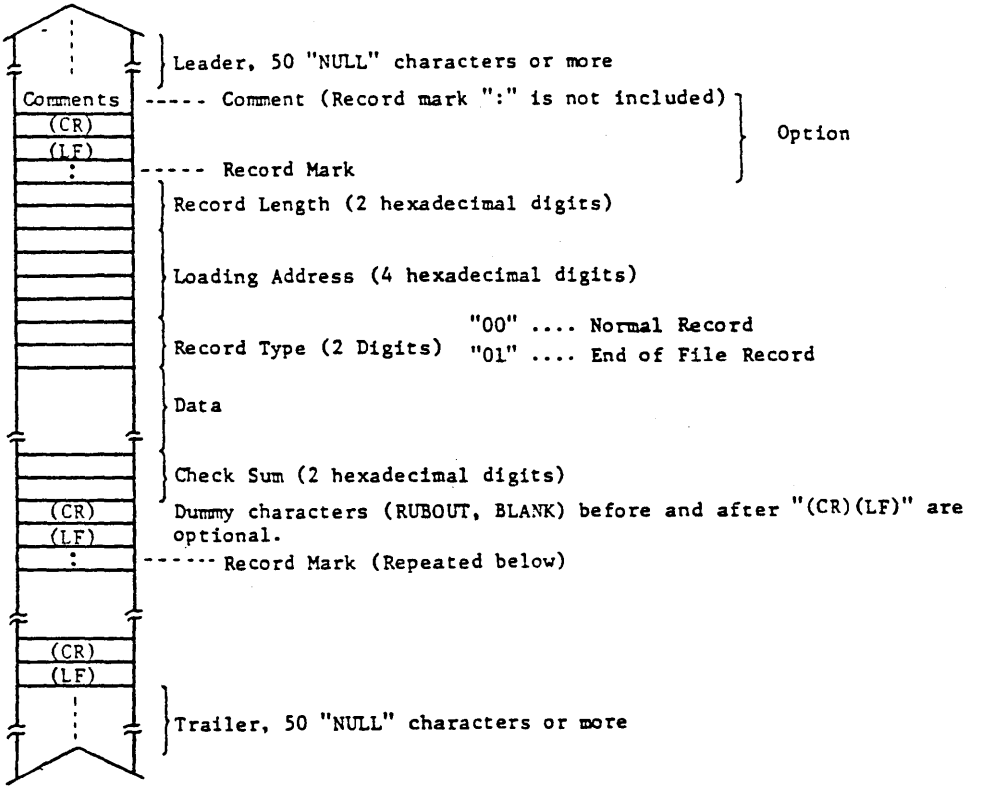




PROGRAM TAPE FORMAT

TMP8048A programs are delivered in the form of paper tape with the following format and it required to attach the tape list. The format of paper tape is same as the Intel type object tape (hexadecimal tape output by Intel MDS system, PROMPT 48 Development Tool, etc.)

(1) Tape Format



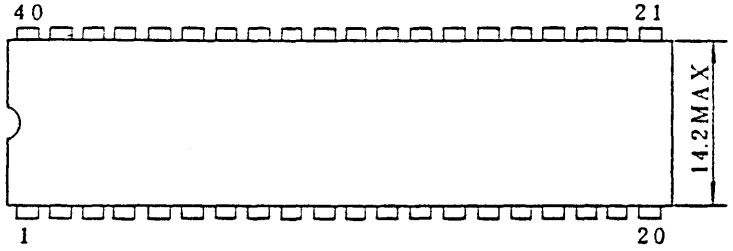
(2) Example of Tape List (TMP8048AP)

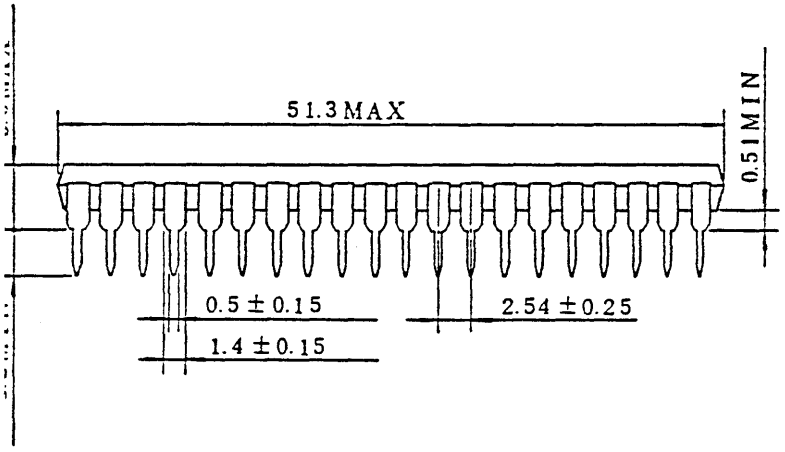
```
TOSHIBA MICRO COMPUTER TLCS-48
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B61380 60B20VC372BF60BD6
:00000001FF
```

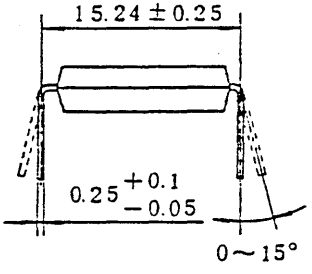
OUTLINE DRAWING

TMP8048AP/TMP8035AP, TMP8049AP/TMP8039AP

Unit in mm







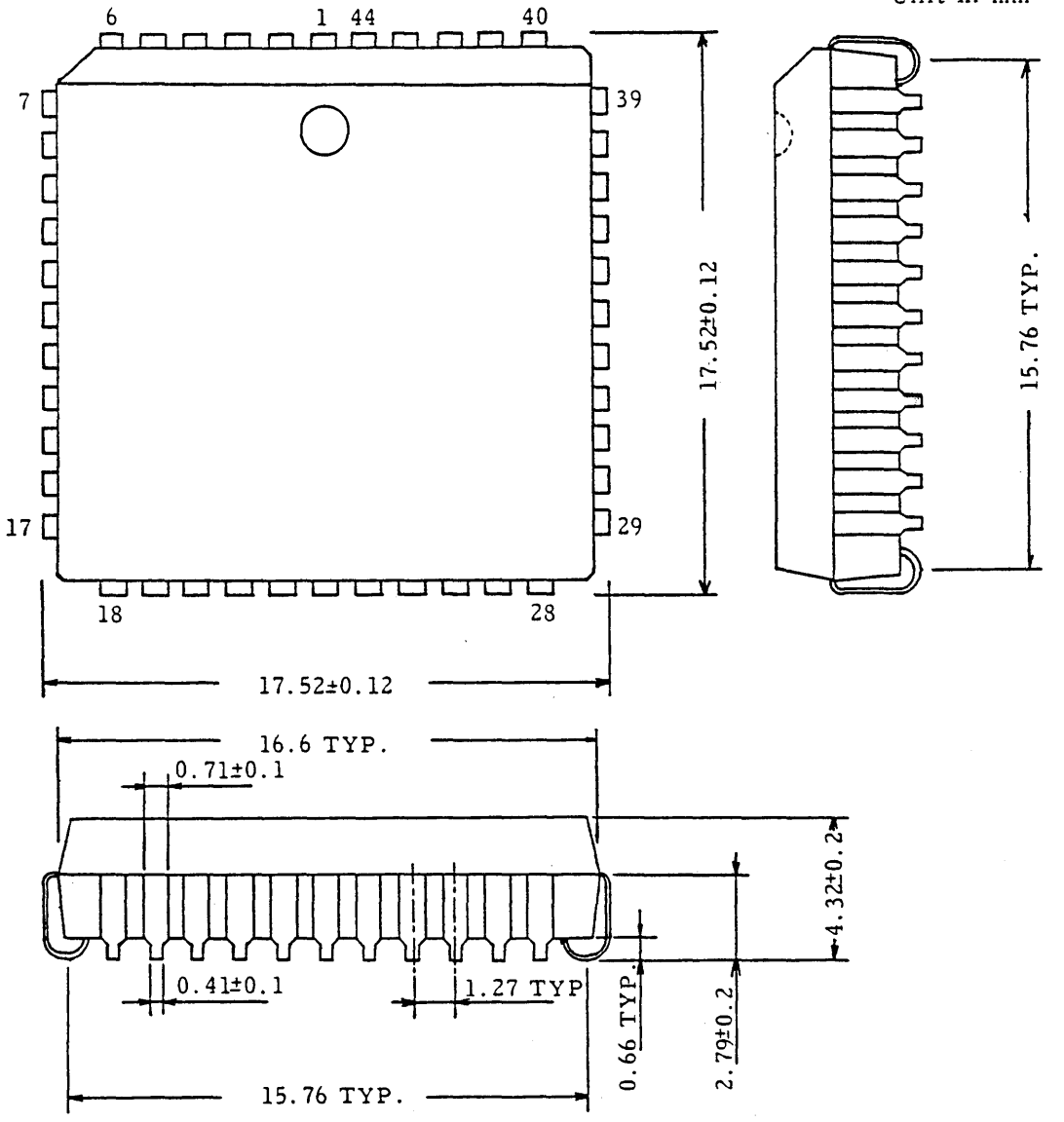
Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

OUTLINE DRAWING

TMP8048AT/TMP8035AT, TMP8049AT/TMP8039AT

PRELIMINARY

Unit in mm



8-BIT SINGLE-CHIP MICROCOMPUTER

TMP80C48AP/TMP80C48AP-6
TMP80C35AP/TMP80C35AP-6
TMP80C48AF/TMP80C48AF-6
TMP80C48AT/TMP80C35AT

GENERAL DESCRIPTION

The TMP80C48A is a single chip microcomputer fabricated in Silicon Gate CMOS technology which provides internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, 64 x 8 RAM data memory, 1K x 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

The TMP80C48A is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

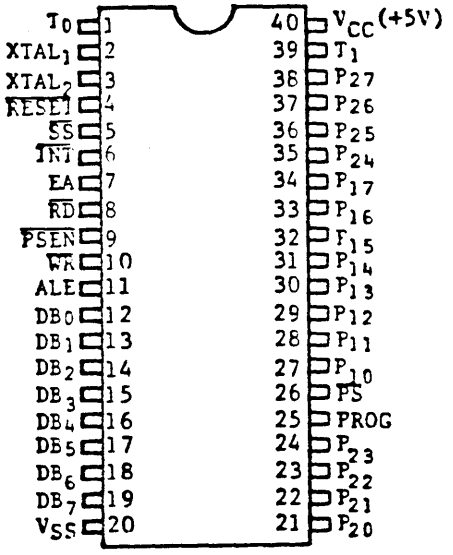
The TMP80C35A/-6 is the equivalent of a TMP80C48A/-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP80C48AP/-6 and TMP80C35AP/-6 are in a standard Dual Inline Package.
The TMP80C48AF/-6 is in a 44-pin Flat Package.
The TMP80C48AT and TMP80C35AT are packaged in the JEDEC standard type 44pin PLCC (Plastic Leaded Chip Carrier)

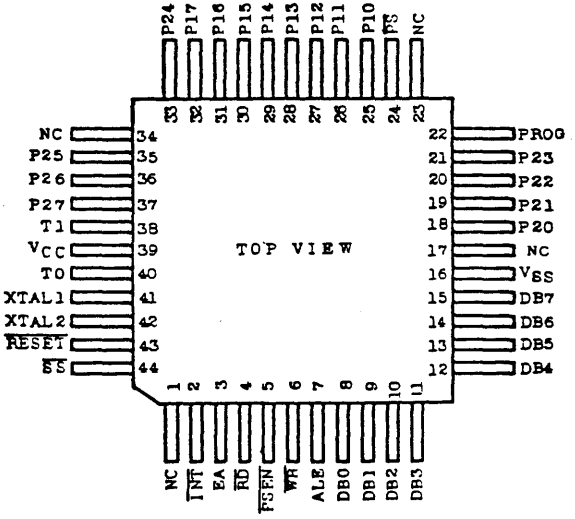
FEATURES

.TMP80C48AP/TMP80C35AP/TMP80C48AF/
TMP80C48AT/TMP80C35AT
1.36μs Instruction Cycle Time
0°C to 70°C, 5V ± 10%
. TMP80C48AP-6/TMP80C35AP-6/TMP80C48AF-6
2.5 μs Instruction Cycle Time
-40°C to 85°C, 5V ± 20%
. Software Upward Compatible with
TMP8048AP/INTEL's 8048
. HALT Instruction (Additional Instruction)
. 1K x 8 masked ROM
. 64 x 8 RAM
. 27 I/O lines
. Interval Timer/Event Counter
. Low Power
10mA MAX. in Normal Operation
(VCC=5V, fXTAL=6MHz)
10μA Max. in Power Down Mode
(VCC=5V, fXTAL : DC)
. Single Power Supply
. Power Down Mode (Stand-by Mode)
. Halt Mode (Idle Mode)

PIN CONNECTIONS (TOP VIEW)

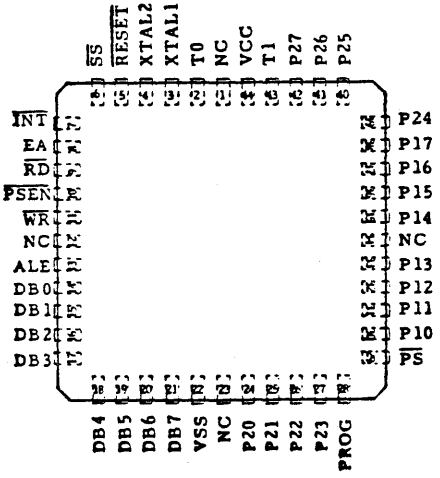


Flat Package

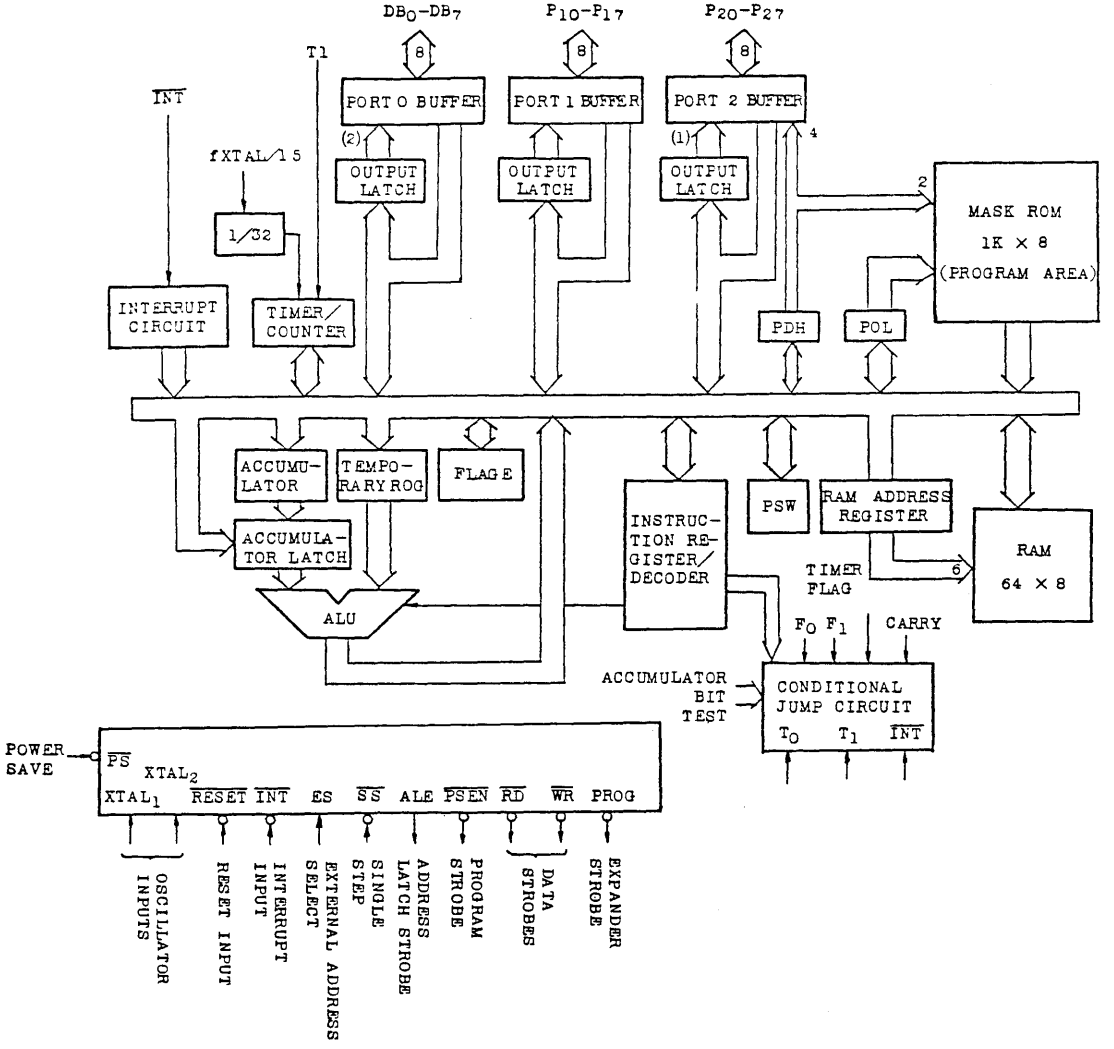


NC: No Connection

PLCC



BLOCK DIAGRAM



Note 1) The lower order 4 bit of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VCC (Power Supply)
+5V during operation

$\overline{PS}$ (Input)
The control signal for the power saving at the power down mode (Active Low)

PROG (Output)
Output strobe for the TMP82C43P I/O expander.

P10 - P17 (Input/Output) Port 1
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P27 (Input/Output) Port 2
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0 - DB7 (Input/Output, Tri-State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low). Used as a Write Strobe to External Data Memory.

$\overline{\text{RESET}}$ (Input)
Active Low signal which is used to initialize the Processor. Also used during the power down mode.

ALE (Oupput)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being executed. Also used during the power down mode.

EA (Input)
External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High)

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

TLCS-48 LIST OF INSTRUCTIONS (1/4)

Class	Assembler Mnemonic	Object Code (1st) (2nd) Bin.	Hex.	Function		Flag C, AC	Cycle
Accumulator Instruction	ADD A, Rr	01101rrr	68+r	(A)←(A)+(Rr)	r=0~7	↕ ↕	1
	ADD A, @Rr	0110000r	60+r	(A)←(A)+[(Rr)]	r=0, 1	↕ ↕	1
	ADD A, #i	00000011 iiiiiiii	03 ii	(A)←(A)+i		↕ ↕	2
	ADDC A, Rr	01111rrr	78+r	(A)←(A)+(Rr)+(C)	r=0~7	↕ ↕	1
	ADDC A, @Rr	0111000r	70+r	(A)←(A)+[(Rr)]+(C)	r=0, 1	↕ ↕	1
	ADDC A, #i	00010011 iiiiiiii	13 ii	(A)←(A)+i+(C)		↕ ↕	2
	ANL A, Rr	01011rrr	58+r	(A)←(A) ∧(Rr)	r=0~7		1
	ANL A, @Rr	0101000r	50+r	(A)←(A) ∧[(Rr)]	r=0, 1		1
	ANL A, #i	01010011 iiiiiiii	53 ii	(A)←(A) ∧i			2
	ORL A, Rr	01001rrr	48+r	(A)←(A) ∨(Rr)	r=0~7		1
	ORL A, @Rr	0100000r	40+r	(A)←(A) ∨[(Rr)]	r=0, 1		1
	ORL A, #i	01000011 iiiiiiii	43 ii	(A)←(A) ∨i			2
	XRL A, Rr	11011rrr	D8+r	(A)←(A) ∀(Rr)	r=0~7		1
	XRL A, @Rr	1101000r	D0+r	(A)←(A) ∀[(Rr)]	r=0, 1		1
	XRL A, #i	11010011 iiiiiiii	D3 ii	(A)←(A) ∀i			2
Instruction	INC A	00010111	17	(A)←(A)+1			1
	DEC A	00000111	07	(A)←(A)-1			1
	CLR A	00100111	27	(A)←0			1
	CPL A	00110111	37	(A)←NOT(A)			1
	DA A	01010111	57	(A)←(A)BCD		↕	1
	SWAP A	01000111	47	(A)<7:4> ↔(A)<3:0>			1
	RL A	11100111	E7	(A)<n+1> ←(A)<n> (A)<0> ←(A)<7>	n=0~6		1
	RLC A	11110111	F7	(A)<n+1> ←(A)<n> (C)←(A)<7> (A)<0> ←(C)	n=0~6	↕	1
	RR A	01110111	77	(A)<n> ←(A)<n+1> (A)<7> ←(A)<0>	n=0~6		1
	RRC A	01100111	67	(A)<n> ←(A)<n+1> (C)←(A)<0> (A)<7> ←(C)	n=0~6	↕	1
I/O	IN A, Pd	000010pp	08+p	(A)←(Pp)	P=1, 2		2

TLCS-48 LIST OF INSTRUCTIONS (2/4)

Class	Assembler Mnemonic	Object Code (1st) (2nd) Bin.	Hex.	Function		Flag C, AC	Cycle
INPUT/OUTPUT Instruction	OUTL Pp, A	001110pp	38+p	(Pp) ←(A)	P=1, 2		2
	ANL Pp, #i	100110pp iiiiiiii	98+p ii	(Pp) ←(Pp)∧i	P=1, 2		2
	ORL Pp, #i	100010pp iiiiiiii	88+p ii	(Pp) ←(Pp)∨i	P=1, 2		2
	INS A, BUS	00001000	08	(A)←(BUS)			2
	OUTL BUS, A	00000010	02	(BUS)←(AC)			2
	ANL BUS, #i	10011000 iiiiiiii	98 ii	(BUS)←(BUS) ∧i			2
	ORL BUS, #i	10001000 iiiiiiii	88 ii	(BUS)←(BUS) ∨i			2
	MOVD A, Pp	000011pp	0C+p	(A)<3:0> ←(Pp) (A)<7:4> ←0	p=4~7		2
	MOVD Pp, A	001111pp	3C+p	(Pp) ←(A)<3:0>	p=4~7		2
	ANLD Pp, A	100111pp	9C+p	(Pp) ←(Pp)∧(A)<3:0>	p=4~7		2
	ORLD Pp, A	100011pp	8C+p	(Pp) ←(Pp)∨(A)<3:0>	p=4~7		2
(1)	INC Rr	00011rrr	18+r	(Rr) ←(Rr)+1	r=0~7		1
(1)	INC @Rr	0001000r	10+r	[(Rr)] ←[(Rr)]+1	r=0, 1		1
(1)	DEC Rr	11001rrr	C8+r	(Rr) ←(Rr)-1	r=0~7		1
Branch Instruction	JMP a	a10 0100 aHL	a10+4	(PC)<10:0> ←a (PC)<11> ←(DBF)			2
	JMPP @A	10110011	B3	(PC)<7:0>←PRO[(PC)<11:8>・(A)]			2
	DJNZ Rr, a	11101rrr aHL	E8+r	(Rr) ←(Rr)-1 if(Rr) ≠0then(PC)<7:0>←aHL else no operation	r=0~7		2
	JC a	11110110 aHL	F6	if(C)=1 then(PC)<7:0>←aHL else no operation			2
	JNC a	11100110 aHL	E6	if(C)=0 then(PC)<7:0>←aHL else no operation			2
	JZ a	11000110 aHL	C6	if(A)=0 then(PC)<7:0>←aHL else no operation			2
	JNZ a	10010110 aHL	96	if(A)≠0 then(PC)<7:0> ←aHL else no operation			2
	JTO a	00110110 aHL	36	if TO=1 then(PC)<7:0>←aHL else no operation			2

(1) ····· Register Instruction

Class	Assembler Mnemonic		Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C,AC	Cycle
Branch Instruction	JNTO	a	00100110 aHL	26	if T0=0 then(PC)<7:0>←aHL else no operation		2
	JT1	a	01010110 aHL	56	if T1=1 then(PC)<7:0>←aHL else no operation		2
	JNT1	a	01000110 aHL	46	if T1=0 then(PC)<7:0>←aHL else no operation		2
	JF0	a	10110110 aHL	B6	if F0=1 then(PC)<7:0>←aHL else no operation		2
	JF1	a	01110110 aHL	76	if F1=1 then(PC)<7:0>←aHL else no operation		2
	JTF	a	00010110 aHL	16	if TF=1 then(PC)<7:0>←aHL else no operation		2
	JNI	a	10000110 aHL	86	if INT =0 then(PC)<7:0>←aHL else no operation		2
	JBb	a	bbb10010 aHL	b*12	if (A)<b>=1 then (PC)<7:0>←aHL else no operation b=0~7		2
(2)	CALL	a	aH10100 aHL	aH*14	[(SP)] ←(PSW)<7:4> • (PC) (SP) ←(SP)+1 (PC)<10:0> ←a (PC)<11> ←(DBF)		2
	RET		10000011	83	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0>		2
	RETR		10010011	93	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0> (PSW)<7:4> ←[(SP)]<15:12>	↕ ↕	2
(3)	CLR	C	10010111	97	(C)←0		1
	CPL	C	10100111	A7	(C)←NOT(C)		1
	CLR	F0	10000101	85	(F0) ←0		1
	CPL	F0	10010101	95	(F0) ←NOT(F0)		1
	CLR	F1	10100101	A5	(F1) ←0		1
	CPL	F1	10110101	B5	(F1) ←NOT(F1)		1
(4)	MOV	A , Rr	11111rrr	F8+r	(A)←(Rr) r=0~7		1
	MOV	A ,@Rr	1111000r	F0+r	(A)←[(Rr)] r=0,1		1
	MOV	A ,#i	00100011 iiiiiiii	23 ii	(A) ←i		2
	MOV	Rr, A	10101rrr	A8+r	(Rr)←(A) r=0~7		1
	MOV	@Rr, A	1010000r	A0+r	[(Rr)]←(A) r=0,1		1
Move Instruction	MOV	Rr,#i	10111rrr iiiiiiii	B8+r ii	(Rr)←i r=0~7		2
	MOV	@Rr,#i	1011000r iiiiiiii	B0+r ii	[(Rr)]←i r=0,1		2
	MOV	A,PSW	11000111	C7	(A) ←(PSW)		1
	MOV	PSW,A	11010111	D7	(PSW) ←(A)		1
	XCH	A,Rr	00101rrr	28+r	(A) ↔(Rr) r=0~7		1
	XCH	A,@Rr	0010000r	20+r	(A) ↔[(Rr)] r=0,1		1
	XCHD	A,@Rr	0011000r	30+r	(A)<3:0>↔[(Rr)<3:0>] r=0,1		1
	MOVX	@Rr,A	1001000r	90+r	EXT[(Rr)] ←(A) r=0,1		1
	MOVX	A,@Rr	1000000r	80+r	(A) ←EXT[(Rr)] r=0,1		1
	MOVP	A,@A	10100011	A3	(A) ←PRO[(PC)<11:8> • (A)]		1
	MOVP3	A,@A	11100011	E3	(A) ←PRO[(PC)<11> • 011 • (A)]		1
TC/Counter	MOV	A,T	01000010	42	(A) ←(TR)		1
	MOV	T,A	01100010	62	(TR)←(A)		1
	STRT	T	01010101	55	Start Timer		1
	STRT	CNT	01000101	45	Start counter		1
	STOP	TCNT	01100101	65	Stop Timer/Counter		1
	EN	TCNTI	00100101	25	Enable Timer/Counter Interrupt		1
	DIS	TCNTI	00110101	35	Disable Timer/Counter Interrupt		1
(5)							
Control	EN	I	00000101	05	Enable External Interrupt		1
	DIS	I	00010101	15	Disable External Interrupt		1
	SEL	RB0	11000101	C5	(BS)← 0		1
	SEL	RB1	11010101	D5	(BS)← 1		1
	SEL	MB0	11100101	E5	(DBF) ← 0		1
	SEL	MB1	11110101	F5	(DBF) ← 1		1
	ENTO	CLK	01110101	75	Enable Clock Output on T0		1
	HALT		00000001	01	Halt		1
(6)	NOP		00000000	00	no opration		1

(2) ····· Subroutine Instruction (3) ····· Flag Instruction
(4) ····· Move Instruction (5) ····· A/D Converter Instruction (6) ····· Other

TMP80C48AP/TMP80C35AP/TMP80C48AF/TMP80C48AT/TMP80C35AT
ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=70°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.8	V
VIL1	Input Low Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$)			-0.5	-	0.6	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
VIH1	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	
ICC2	VCC Supply Current	Normal operation	VCC=5V, fXTAL=11MHz	-	-	15	mA
ICCH2		HALT Mode	VIH=VCC-0.2V VIL=0.2V	-	-	4.0	

TMP80C48AP/TMP80C35AP/TMP80C48AF/TMP80C48AT/TMP80C35AT

AC CHARACTRISTICS

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
t	Clock Period	Note 2	1/xtal f	90.9	1000	ns
tLL	ALE Pulse Width		3.5t-170	150	-	ns
tAL	Address Setup Time (ALE)		2t-110	70	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	50	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	480	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	350	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	390	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	40	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	110	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	375	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	240	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	300	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	730	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	460	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	140	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	10	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	200	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	60	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	25	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	290	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	50	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	100	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	650	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	140	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	250	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	40	-	ns

TMP80C48AP/TMP80C35AP/TMP80C48AF/TMP80C48AT/TMP80C35AT

AC CHARACTRISTICS (CONTINUE)

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	700	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	160	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	15	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	510	ns
t0PRR	T0 Clock Period		3t	270	-	ns
tCY	Cycle Time		15t	1.36	15.0	μs

Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 11MHz. and the Min. Clock frequency is 1MHz.

TMP80C48AP-6/TMP80C35AP-6/TMP80C48AF-6 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=85°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS (I)

TOPR = -40°C to 85°C, VCC=+5V±10%, VSS=0V, unless otherwise noted

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage			-0.5	-	0.8	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
VIH1	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current ($\overline{T1}$, $\overline{INT}$, $\overline{EA}$, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current (BUS, $\overline{TO}$) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	

TMP80C48AP-6/TMP80C35AP-6/TMP80C48AF-6 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (II)

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage		-0.5	-	0.15 x VCC	V
VIH	Input High Voltage (Except XTAL1, XTAL2, RESET, PS)		0.5 x VCC	-	VCC	V
VIH1	Input High Voltage (XTAL1, XTAL2, RESET, PS)		0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)	IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)	IOL=1.2mA	-	-	0.45	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)	IOH=-400μA	VCC-0.8	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)	IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current (T1, INT, EA, PS)	VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current (SS, RESET)	VSS≦VIN≦VCC	-	-	-VCC/0.1	μA
ILI2	Input Leak Current (P10-P17, P20-P27)	VSS+0.45V≦VIN≦VCC	-	-	-VCC/0.01	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)	VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current — Normal Operation	VCC=5V, fXTAL=6MHz	-	-	10	mA
ICCH1	VCC Supply Current — HALT Mode	VIH=VCC-0.2V, VIH=0.2V	-	-	2.5	mA

TMP80C48AP-6/TMP80C35AP-6/TMP80C48AF-6

AC CHARACTRISTICS

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
t	Clock.Period	Note 2	1/xtal f	166.6	1000	ns
tLL	ALE Pulse Width		3.5t-170	410	-	ns
tAL	Address Setup Time (ALE)		2t-110	220	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	120	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	1050	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	800	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	880	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	120	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	220	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	800	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	550	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	680	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	1500	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	1000	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	290	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	40	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	420	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	175	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	100	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	590	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	170	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	400	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	1290	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	250	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	710	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	160	-	ns

TMP80C48AP-6/TMP80C35AP-6/TMP80C48AF-6

AC CHARACTRISTICS (CONTINUE)

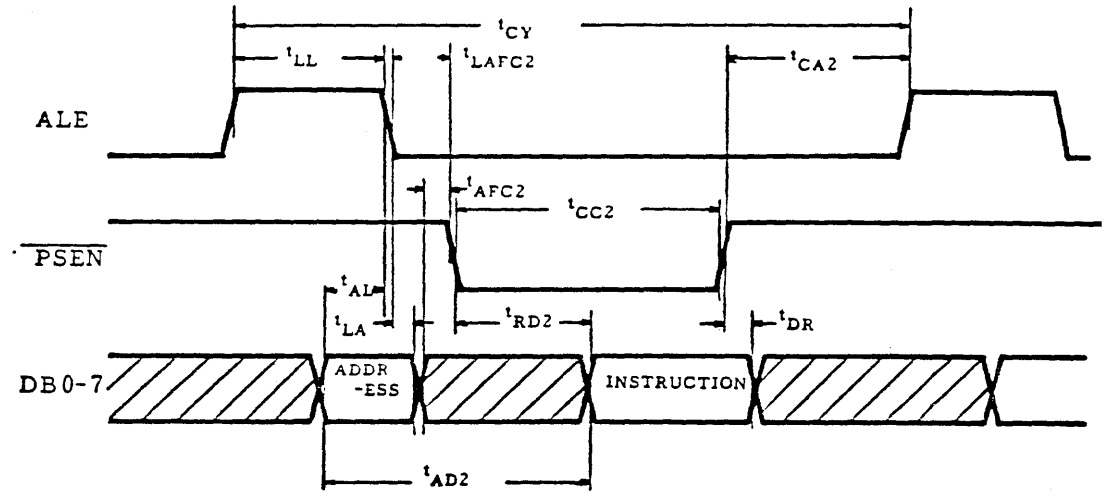
TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	1500	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	460	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	130	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	850	ns
t0PRR	T0 Clock Period		3t	500	-	ns
tCY	Cycle Time		15t	2.5	15.0	μs

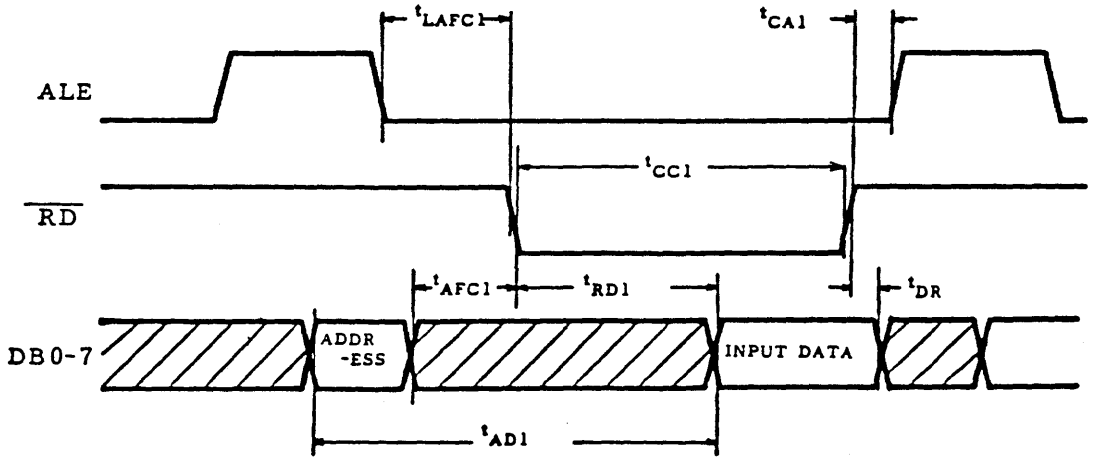
Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 6MHz. and the Min. Clock frequency is 1MHz.

TIMING WAVEFORM

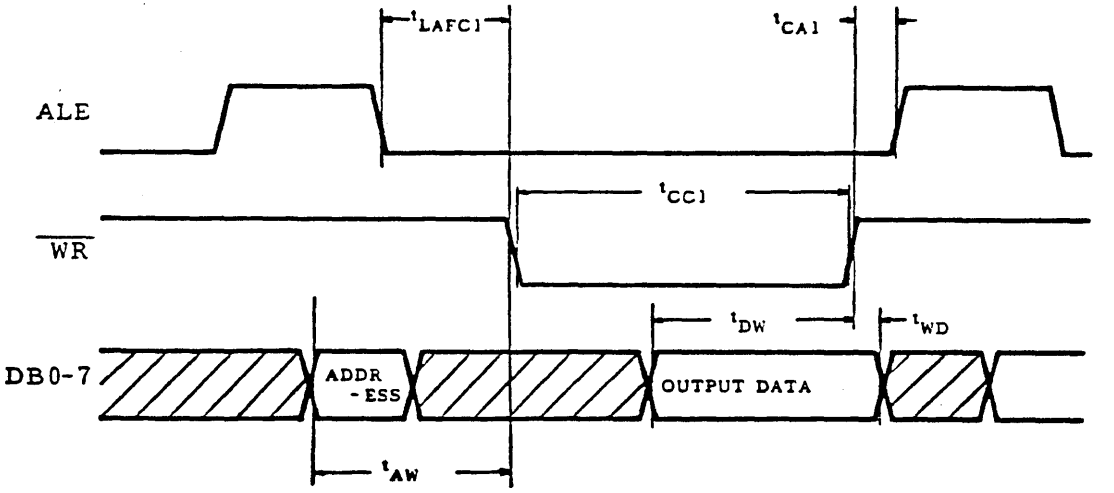
A. Instruction Fetch from External Program Memory



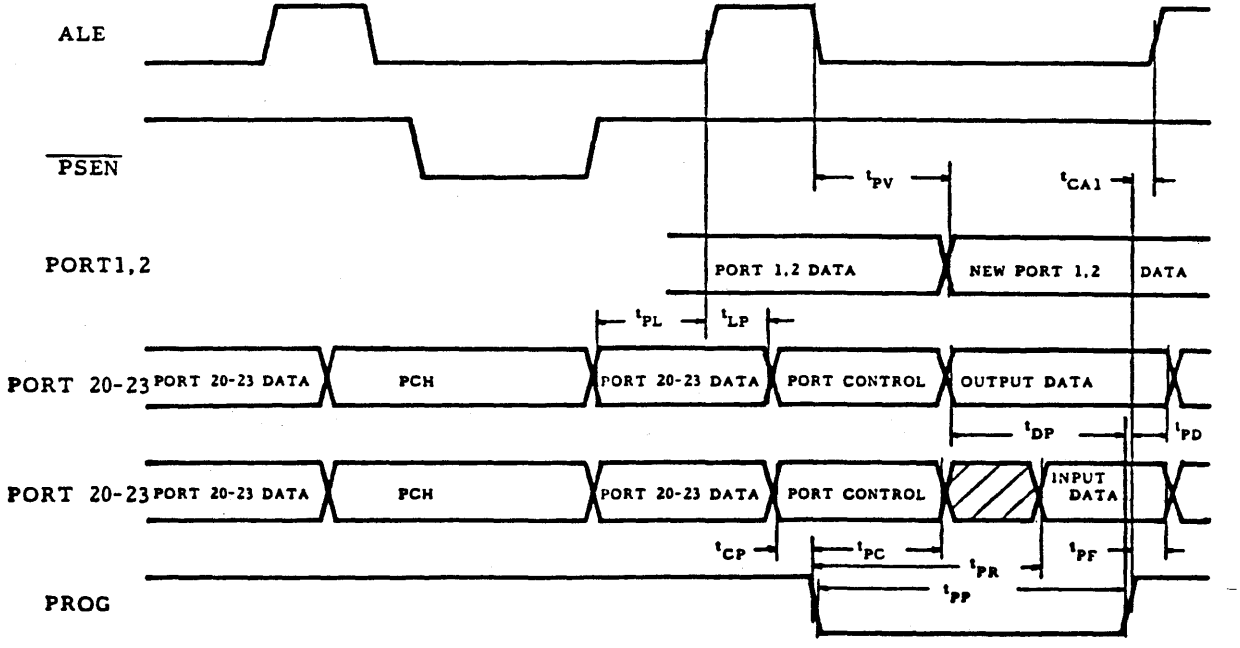
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



POWER DOWN MODE (I) ..... Data Hold Mode in RAM

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{RESET}$ terminal has been set to low level. Consequently, all the data in RAM area can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{RESET}$ terminal is set to high level, thus, the normal mode is restarted from the initialize operation (address 0).

DC CHARACTERISTICS

TMP80C48AP/35AP,TMP80C48AF,TMP80C48AT/35AT: TOPR=0°C to 70°C, VSS=0V
TMP80C48AP-6/35AP-6,TMP80C48AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB1	Standby Voltage(1)		2.0	-	6.0	V
ISB1	Standby Current(1)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

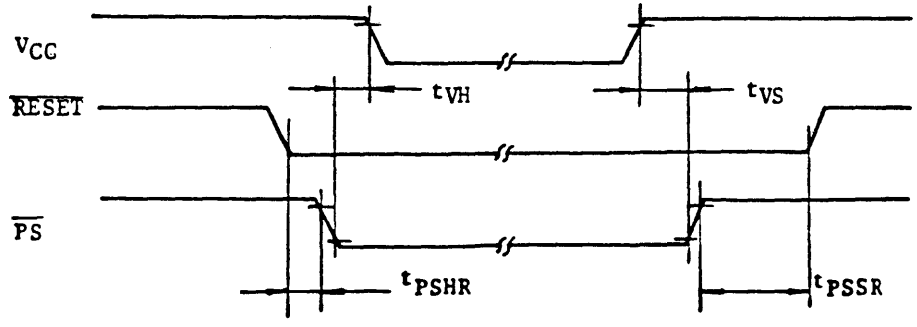
AC CHARACTERISTICS

TMP80C48AP/35AP,TMP80C48AF,TMP80C48AT/35AT: TOPR=0°C to 70°C, VSS=0V
TMP80C48AP-6/35AP-6,TMP80C48AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHR	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μS
tPSSR	Power Save Setup Time ($\overline{RESET}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



POWER DOWN MODE (II) ..... ALL Data Hold Mode

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{SS}$ terminal has been set to low level. Consequently, all data can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 3V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{SS}$ terminal is set to high level, thus, the normal mode is restarted continuously from the state just before the power down mode (II).

DC CHARACTERISTICS

TMP80C48AP/35AP,TMP80C48AF,TMP80C48AT/35AT: TOPR=0°C to 70°C, VSS=0V
TMP80C48AP-6/35AP-6,TMP80C48AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB2	Standby Voltage(2)		3.0	-	6.0	V
ISB2	Standby Current(2)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

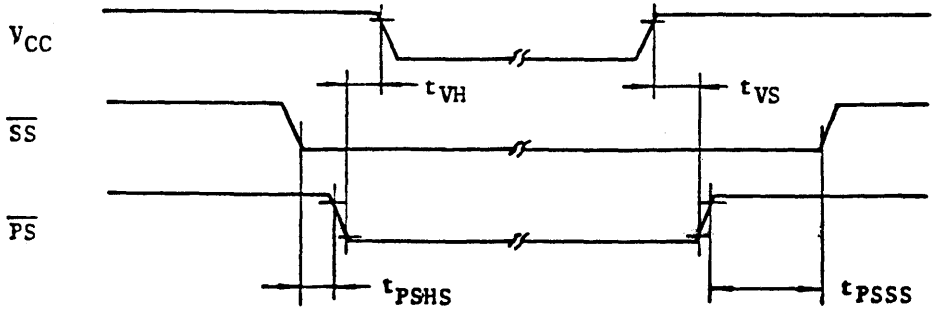
AC CHARACTERISTICS

TMP80C48AP/35AP,TMP80C48AF,TMP80C48AT/35AT: TOPR=0°C to70°C,VSS=5V±10%,VSS=0V
TMP80C48AP-6/35AP-6,TMP80C48AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHS	Power Save Hold Time ($\overline{SS}$)		10	-	-	μS
tPSSS	Power Save Setup Time ($\overline{SS}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



HALT MODE

. 1 HALT INSTRUCTION

OP code is "01H". HALT INSTRUCTION is an additional instruction to the standard 8048/8049 instruction set.

. 2 Entry to HALT MODE

On the execution of HALT INSTRUCTION, TMP80C48A/TMP80C35A enter HALT MODE.

. 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logic values just prior to the execution of HALT INSTRUCTION are maintained. Power consumption in HALT MODE is less than 50% of normal operation. The status of each pins are described in the following table.

. 4 Release from HALT MODE

HALT MODE is released by either of two signals ($\overline{\text{RESET}}$,$\overline{\text{INT}}$).

(1) $\overline{\text{RESET}}$ Release Mode : An active $\overline{\text{RESET}}$ input signal causes the normal reset function. TMP80C48A/TMP80C35A start the program at address "000 H".

(2) $\overline{\text{INT}}$ Release Mode : An active $\overline{\text{INT}}$ input signal causes the normal operation.

In case of interrupt enable mode (EI MODE), TMP80C48A/TMP80C35A execute the interrupt service routine, after the execution of one instruction which is located at the next address after HALT INSTRUCTION.

In case of interrupt disable mode (DI MODE), TMP80C48A/TMP80C35A execute normal operation from the next address after HALT INSTRUCTION.

. 5 Supply Voltage Range in HALT MODE

The operating supply voltage range and the operating temperature range are same as in normal operation.

PIN STATUS IN POWER DOWN MODE (I) (II)

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	High impedance Input disabled
T0	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{\text{RESET}}$, $\overline{\text{SS}}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{\text{INT}}$, EA	Input disabled when oscilltor is stopped.
$\overline{\text{RD}}$, $\overline{\text{WR}}$, ALE PROG, $\overline{\text{PSEN}}$	High impedance

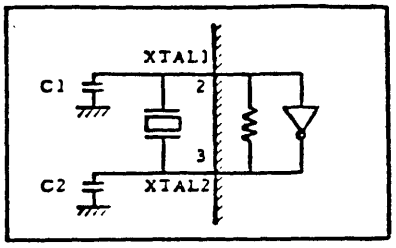
PIN STATUS IN HALT MODE

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	Values prior to the execution of HALT INSTRUCTION are maintained.
T0	Status prior to the execution of HALT INSTRUCTION is maintained.
T1	Input disabled
XTAL1, XTAL2	Continue oscillation
$\overline{\text{RESET}}$, $\overline{\text{INT}}$	Input enabled
$\overline{\text{SS}}$, $\overline{\text{EA}}$	Input disabled
$\overline{\text{RD}}$, $\overline{\text{WR}}$, PROG, $\overline{\text{PSEN}}$	Output "High" level
ALE	Output "Low" level

OSCILATOR

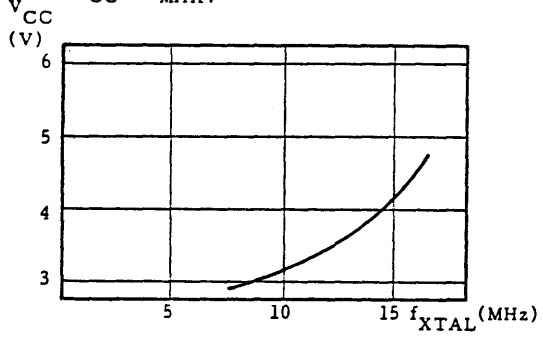
QUARTZ CRYSTAL
f = 1MHz to 4MHz : C1 = C2 = 30pF
f = 4MHz to 11MHz : C1 = C2 = 20pF

CERAMIC RESONATOR
f = 1MHz to 3MHz : C1 = C2 = 100pF
f = 3MHz to 11MHz : C1 = C2 = 30pF

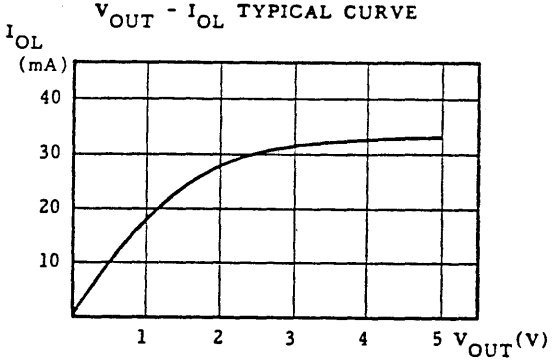


TYPICAL CHARACTARISTICS: Vcc = 5V, Ta = 25°C, unless otherwise noted.

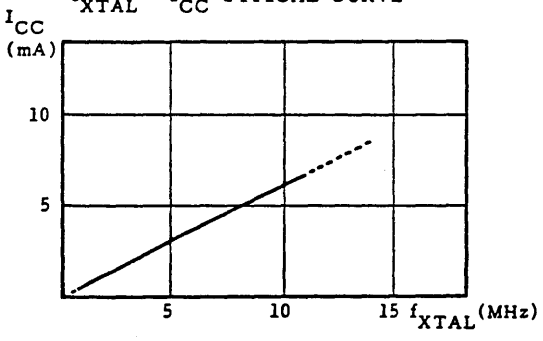
V_{CC} - $f_{MAX.}$ TYPICAL CURVE



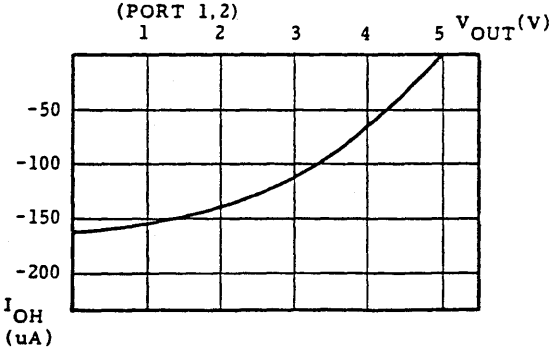
V_{OUT} - I_{OL} TYPICAL CURVE



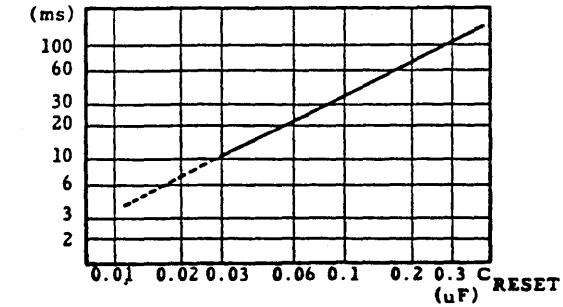
f_{XTAL} - I_{CC} TYPICAL CURVE



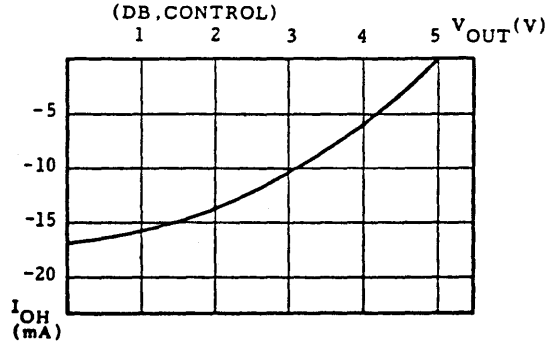
V_{OUT} - I_{OH} TYPICAL CURVE (PORT 1,2)



t_{RESET} C_{RESET} - t_{RESET} TYPICAL CURVE

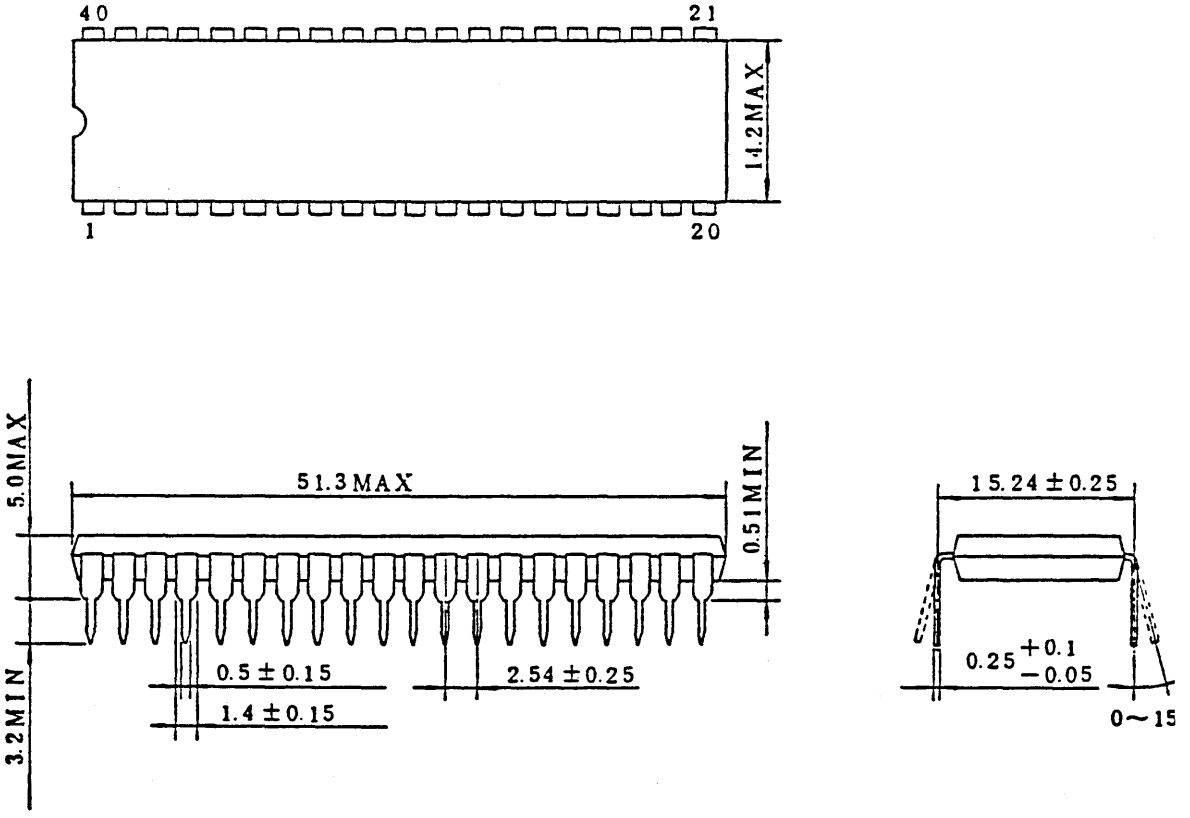


V_{OUT} - I_{OH} TYPICAL CURVE (DB, CONTROL)



OUTLINE DRAWING (DUAL INLINE PACKAGE)

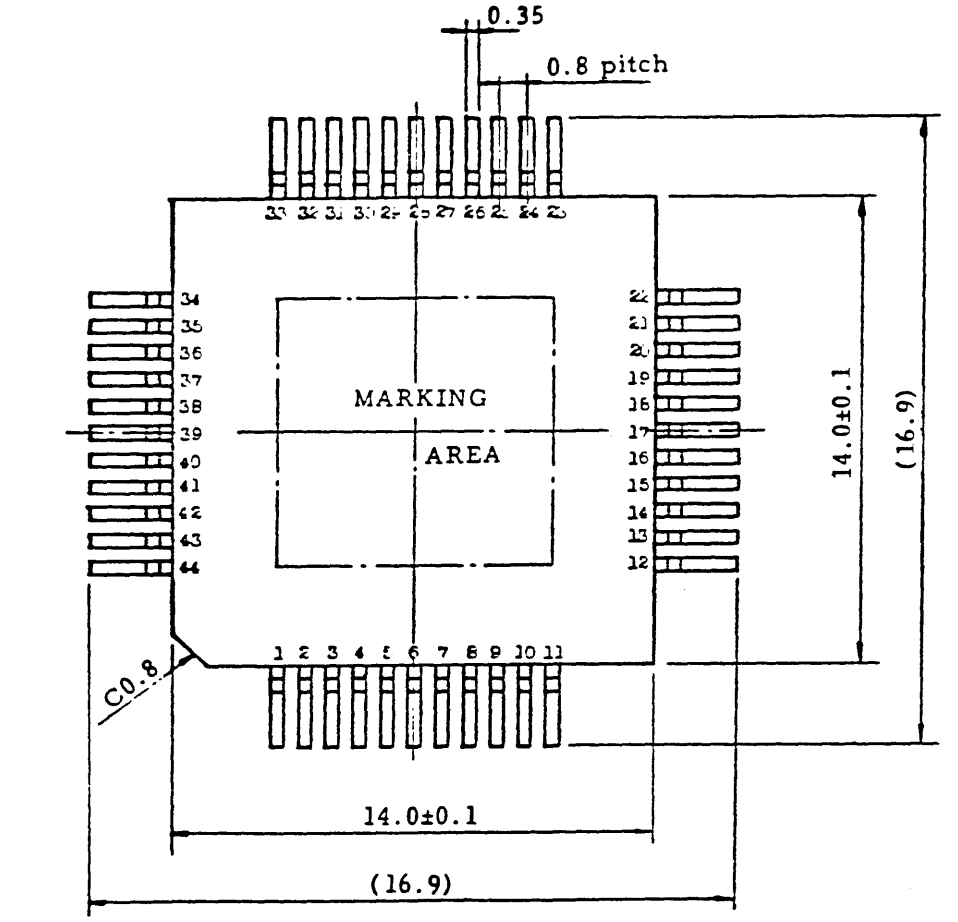
Unit in mm

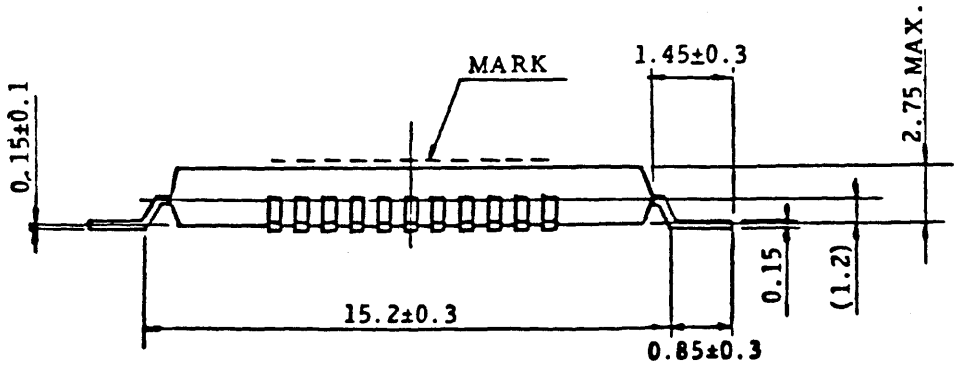


Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

OUTLINE DRAWING (FLAT PACKAGE)

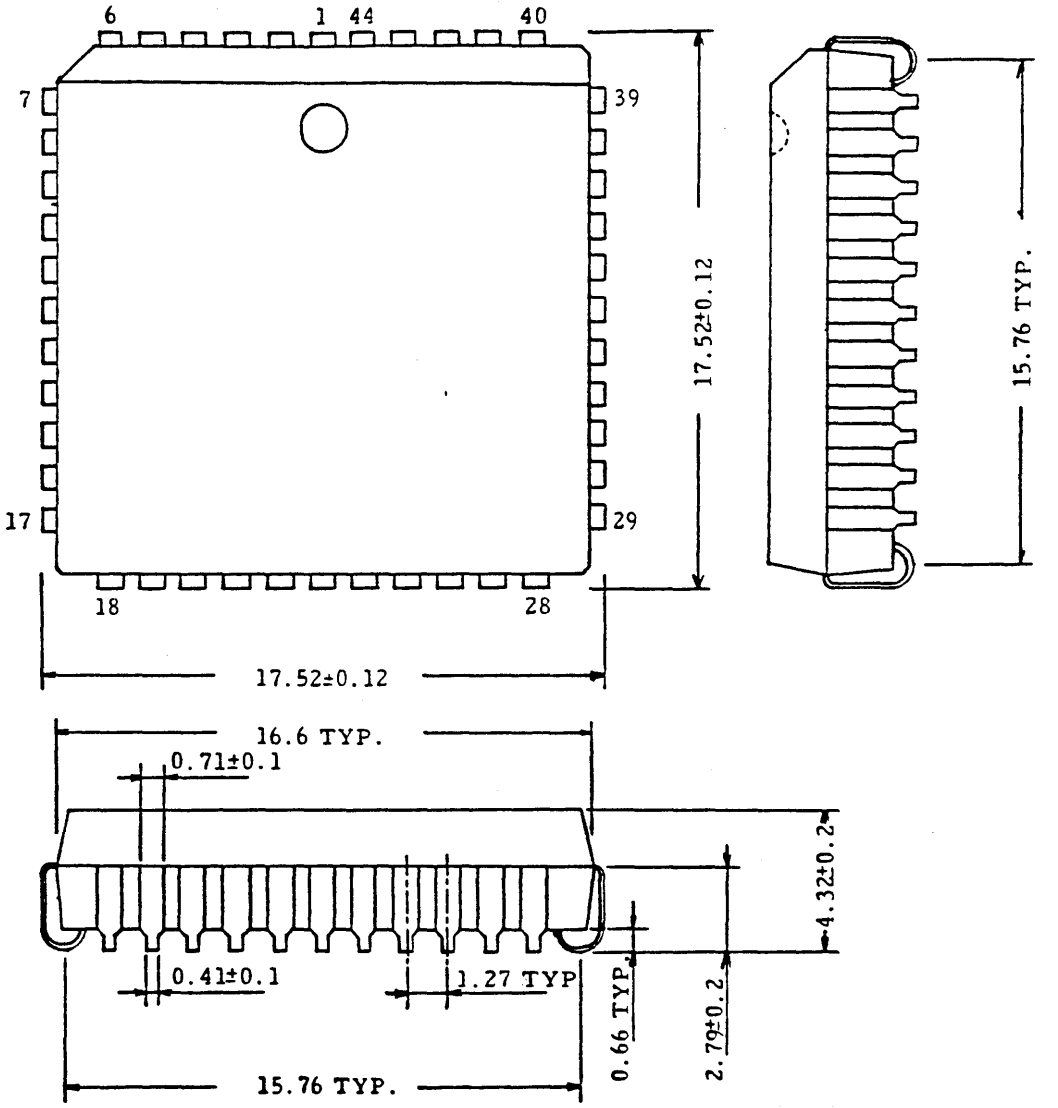
Unit in mm





OUTLINE DRAWING (Plastic Leaded Chip Carrier)

unit in mm



8-BIT SINGLE-CHIP MICROCOMPUTER

TMP80C49AP/TMP80C49AP-6
TMP80C39AP/TMP80C39AP-6
TMP80C49AF/TMP80C49AF-6
TMP80C49AT/TMP80C39AT

GENERAL DESCRIPTION

The TMP80C49A is a single chip microcomputer fabricated in Silicon Gate CMOS technology which provides internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, 128 x 8 RAM data memory, 2K x 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

The TMP80C49A is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

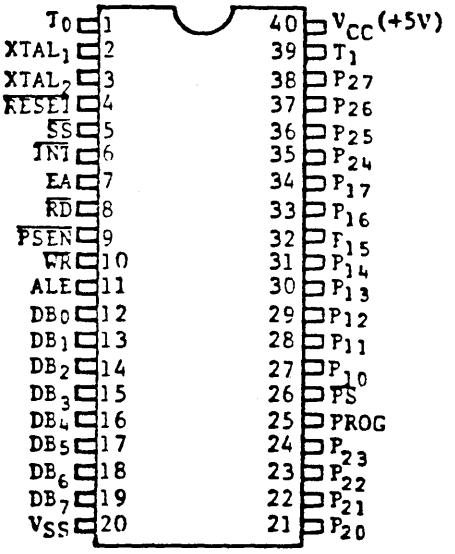
The TMP80C39A/-6 is the equivalent of a TMP80C49A/-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP80C49AP/-6 and TMP80C39AP/-6 are in a standard Dual Inline Package.
The TMP80C49AF/-6 is in a 44-pin Flat Package.
The TMP80C49AT and TMP80C39AT are packaged in the JEDEC standard type 44pin PLCC (Plastic leaded Chip Carrier).

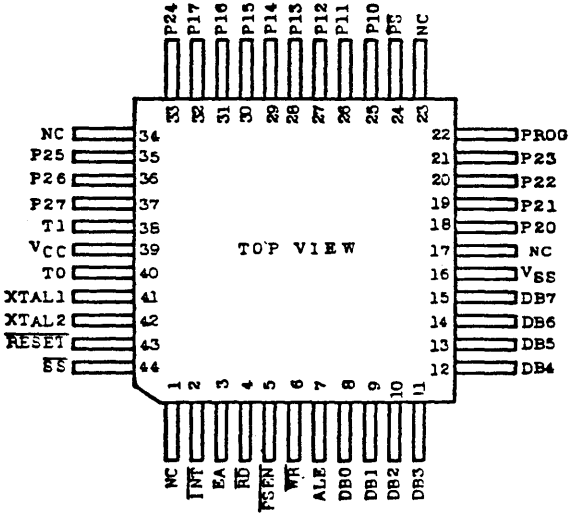
FEATURES

- TMP80C49AP/TMP80C39AP/TMP80C49AF/ TMP80C49AT/TMP80C39AT
  1.36μs Instruction Cycle Time
  0°C to 70°C, 5V ± 10%
- TMP80C49AP-6/TMP80C39AP-6/TMP80C49AF-6
  2.5 μs Instruction Cycle Time
  -40°C to 85°C, 5V ± 20%
- Software Upward Compatible with TMP8049AP/TMP80C49P-6/INTEL's 8049
- HALT Instruction (Additional Instruction)
- 2K x 8 masked ROM
- 128 x 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Low Power
  10mA MAX. in Normal Operation (VCC=5V, fXTAL=6MHz)
  10μA Max. in Power Down (VCC=5V, fXTAL: DC)
- Single Power Supply
- Power Down Mode (Stand-by Mode)
- Halt Mode (Idle Mode)

PIN CONNECTIONS (TOP VIEW)

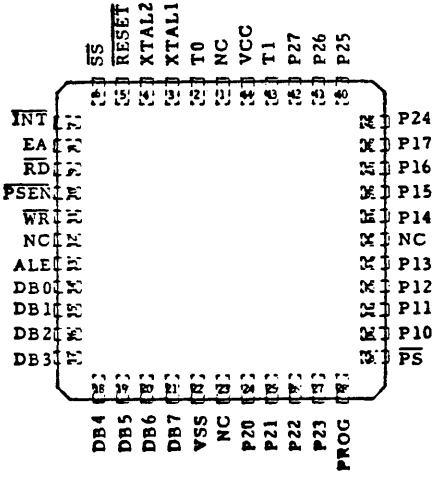


Flat Package

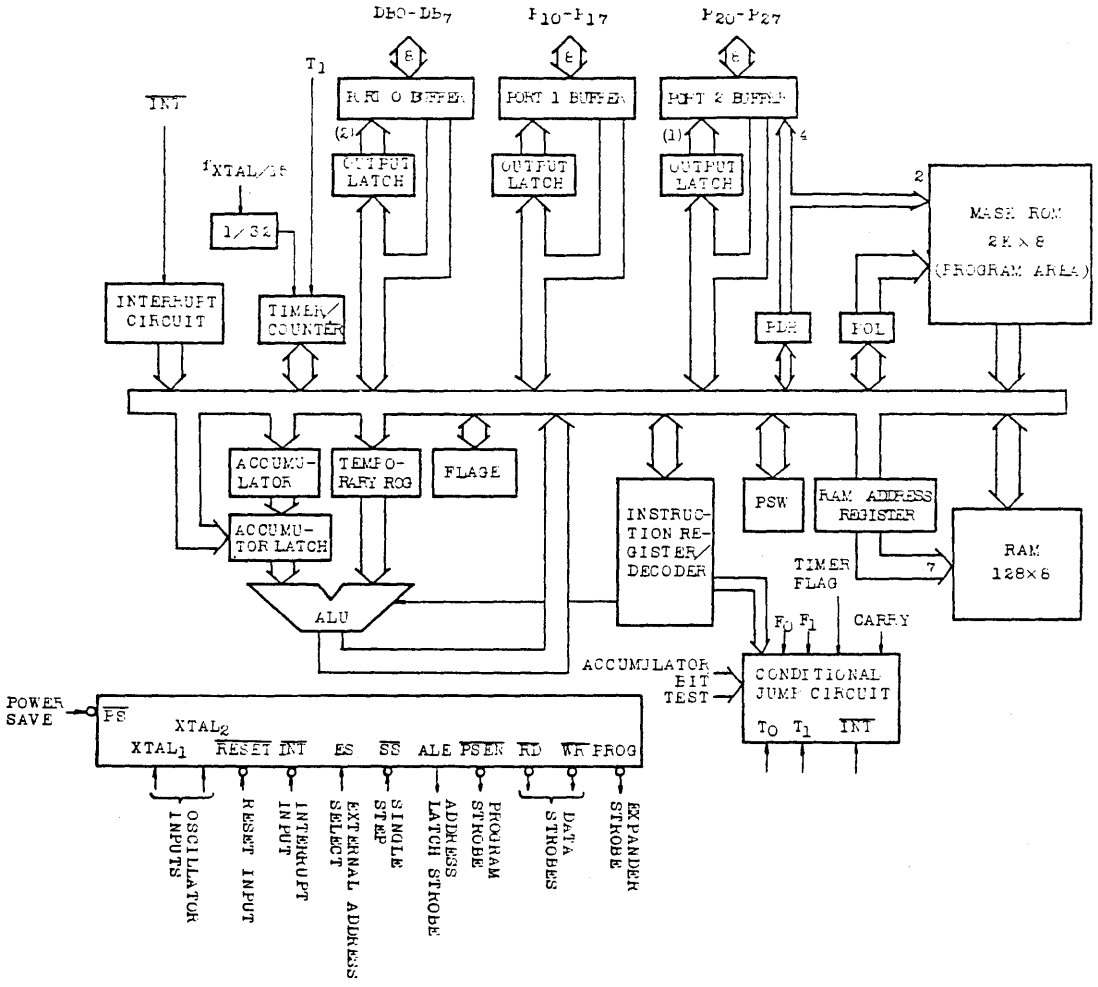


NC: No Connection

PLCC



BLOCK DIAGRAM



Note 1) The lower order 4 bit of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VCC (Power Supply)
+5V during operation

$\overline{PS}$ (Input)
The control signal for the power saving at the power down mode (Active Low)

PROG (Output)
Output strobe for the TMP82C43P I/O expander.

P10 - P17 (Input/Output) Port 1
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P27 (Input/Output) Port 2
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0 - DB7 (Input/Output, Tri-State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low). Used as a Write Strobe to External Data Memory.

$\overline{\text{RESET}}$ (Input)
Active Low signal which is used to initialize the Processor. Also used during the power down mode.

ALE (Oupput)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being executed. Also used during the power down mode.

EA (Input)
External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High)

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

TLCS-48 LIST OF INSTRUCTIONS (1/4)

Class	Assembler Mnemonic	Object Code (1st) (2nd) Bin.	Hex.	Function	Flag C, AC	Cycle
Accumulator Instruction	ADD A, Rr	01101rrr	68+r	(A)←(A)+(Rr) r=0~7	↕ ↕	1
	ADD A, @Rr	0110000r	60+r	(A)←(A)+[(Rr)] r=0,1	↕ ↕	1
	ADD A, #i	00000011 iiiiiiii	03 ii	(A)←(A)+i	↕ ↕	2
	ADDC A, Rr	01111rrr	78+r	(A)←(A)+(Rr)+(C) r=0~7	↕ ↕	1
	ADDC A, @Rr	0111000r	70+r	(A)←(A)+[(Rr)]+(C) r=0,1	↕ ↕	1
	ADDC A, #i	00010011 iiiiiiii	13 ii	(A)←(A)+i+(C)	↕ ↕	2
	ANL A, Rr	01011rrr	58+r	(A)←(A) ∧(Rr) r=0~7		1
	ANL A, @Rr	0101000r	50+r	(A)←(A) ∧[(Rr)] r=0,1		1
	ANL A, #i	01010011 iiiiiiii	53 ii	(A)←(A) ∧i		2
	ORL A, Rr	01001rrr	48+r	(A)←(A) ∨(Rr) r=0~7		1
	ORL A, @Rr	0100000r	40+r	(A)←(A) ∨[(Rr)] r=0,1		1
	ORL A, #i	01000011 iiiiiiii	43 ii	(A)←(A) ∨i		2
	XRL A, Rr	11011rrr	D8+r	(A)←(A) ∀(Rr) r=0~7		1
	XRL A, @Rr	1101000r	D0+r	(A)←(A) ∀[(Rr)] r=0,1		1
	XRL A, #i	11010011 iiiiiiii	D3 ii	(A)←(A) ∀i		2
	INC A	00010111	17	(A)←(A)+1		1
	DEC A	00000111	07	(A)←(A)-1		1
	CLR A	00100111	27	(A)←0		1
	CPL A	00110111	37	(A)←NOT(A)		1
	DA A	01010111	57	(A)←(A)BCD	↕	1
	SWAP A	01000111	47	(A)<7:4> ↔(A)<3:0>		1
	RL A	11100111	E7	(A)<n+1> ←(A)<n> (A)<0> ←(A)<7> n=0~6		1
	RLC A	11110111	F7	(A)<n+1> ←(A)<n> (C)←(A)<7> (A)<0> ←(C) n=0~6	↕	1
	RR A	01110111	77	(A)<n> ←(A)<n+1> n=0~6 (A)<7> ←(A)<0>		1
	RRC A	01100111	67	(A)<n> ←(A)<n+1> (C)←(A)<0> (A)<7> ←(C) n=0~6	↕	1
I/O	IN A, Pd	000010pp	08+p	(A)←(Pd) P=1,2		2
	OUTL Pd, A	001110pp	38+p	(Pd) ←(A) P=1,2		2
	ANL Pd, #i	100110pp iiiiiiii	98+p ii	(Pd) ←(Pd)∧i P=1,2		2
	ORL Pd, #i	100010pp iiiiiiii	88+p ii	(Pd) ←(Pd)∨i P=1,2		2
	INS A, BUS	00001000	08	(A)←(BUS)		2
	OUTL BUS, A	00000010	02	(BUS)←(AC)		2
	ANL BUS, #i	10011000 iiiiiiii	98 ii	(BUS)←(BUS) ∧i		2
	ORL BUS, #i	10001000 iiiiiiii	88 ii	(BUS)←(BUS) ∨i		2
	MOVD A, Pp	000011pp	0C+p	(A)<3:0> ←(Pp) p=4~7 (A)<7:4> ←0		2
	MOVD Pp, A	001111pp	3C+p	(Pp) ←(A)<3:0> p=4~7		2
	ANLD Pp, A	100111pp	9C+p	(Pp) ←(Pp)∧(A)<3:0> p=4~7		2
	ORLD Pp, A	100011pp	8C+p	(Pp) ←(Pp)∨(A)<3:0> p=4~7		2
	INC Rr	00011rrr	18+r	(Rr) ←(Rr)+1 r=0~7		1
(1)	INC @Rr	0001000r	10+r	[(Rr)] ←[(Rr)]+1 r=0,1		1
	DEC Rr	11001rrr	C8+r	(Rr) ←(Rr)-1 r=0~7		1
Branch Instruction	JMP a	a11 00100 aHL	a11+4	(PC)<10:0> ←a (PC)<11> ←(DBF)		2
	JMPP @A	10110011	B3	(PC)<7:0>←PROM[(PC)<11:8>・(A)]		2
	DJNZ Rr, a	11101rrr aHL	E8+r	(Rr) ←(Rr)-1 r=0~7 if(Rr) ≠0then(PC)<7:0>←aHL else no operation		2
	JC a	11110110 aHL	F6	if(C)=1 then(PC)<7:0>←aHL else no operation		2
	JNC a	11100110 aHL	E6	if(C)=0 then(PC)<7:0>←aHL else no operation		2
	JZ a	11000110 aHL	C6	if(A)=0 then(PC)<7:0>←aHL else no operation		2
	JNZ a	10010110 aHL	96	if(A)≠0 then(PC)<7:0> ←aHL else no operation		2
	JTO a	00110110	36	if T0=1 then(PC)<7:0>←aHL		2

TLCS-48 LIST OF INSTRUCTIONS (2/4)

TLCS-48 LIST OF INSTRUCTIONS (3/4)

Clocks	Assembler Mnemonic		Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C,AC	Cycle
Branch Instruction	JNTO	a	00100110 aHL	26	if T0=0 then(PC)<7:0>←aHL else no operation		2
	JT1	a	01010110 aHL	56	if T1=1 then(PC)<7:0>←aHL else no operation		2
	JNT1	a	01000110 aHL	46	if T1=0 then(PC)<7:0>←aHL else no operation		2
	JF0	a	10110110 aHL	B6	if F0=1 then(PC)<7:0>←aHL else no operation		2
	JF1	a	01110110 aHL	76	if F1=1 then(PC)<7:0>←aHL else no operation		2
	JTF	a	00010110 aHL	16	if TF=1 then(PC)<7:0>←aHL else no operation		2
	JNI	a	10000110 aHL	86	if INT =0 then(PC)<7:0>←aHL else no operation		2
	JBb	a	bbb10010 aHL	b+12	if (A)<b>=1 then (PC)<7:0>←aHL else no operation b=0~7		2
(2)	CALL	a	aH10100 aHL	aH+14	[(SP)] ←(PSW)<7:4> • (PC) (SP) ←(SP)+1 (PC)<10:0> ←a (PC)<11> ←(DBF)		2
	RET		10000011	83	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0>		2
	RETR		10010011	93	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0> (PSW)<7:4> ←[(SP)]<15:12>	↕ ↕	2
(3)	CLR	C	10010111	97	(C)←0		1
	CPL	C	10100111	A7	(C)←NOT(C)		1
	CLR	F0	10000101	85	(F0) ←0		1
	CPL	F0	10010101	95	(F0) ←NOT(F0)		1
	CLR	F1	10100101	A5	(F1) ←0		1
	CPL	F1	10110101	B5	(F1) ←NOT(F1)		1
(4)	MOV	A , Rr	11111rrr	F8+r	(A)←(Rr) r=0~7		1
	MOV	A ,@Rr	1111000r	F0+r	(A)←[(Rr)] r=0,1		1
	MOV	A ,#i	00100011 iiiiiiii	23 ii	(A) ←i		2
	MOV	Rr, A	10101rrr	A8+r	(Rr)←(A) r=0~7		1
	MOV	@Rr, A	1010000r	A0+r	[(Rr)]←(A) r=0,1		1

(2) ····· Subroutine Instruction (3) ····· Flag Instruction
(4) ····· Move Instruction

TLCS-48 LIST OF INSTRUCTIONS (4/4)

Clocks	Assembler Mnemonic		Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C,AC	Cycle
Move Instruction	MOV	Rr,#i	10111rrr iiiiiiii	B8+r ii	(Rr)←i r=0~7		2
	MOV	@Rr,#i	1011000r iiiiiiii	B0+r ii	[(Rr)]←i r=0,1		2
	MOV	A,PSW	11000111	C7	(A) ←(PSW)		1
	MOV	PSW,A	11010111	D7	(PSW) ←(A)		1
	XCH	A,Rr	00101rrr	28+r	(A) ↔(Rr) r=0~7		1
	XCH	A,@Rr	0010000r	20+r	(A) ↔[(Rr)] r=0,1		1
	XCHD	A,@Rr	0011000r	30+r	(A)<3:0>↔[(Rr)<3:0>] r=0,1		1
	MOVX	@Rr,A	1001000r	90+r	EXT[(Rr)] ←(A) r=0,1		1
	MOVX	A,@Rr	1000000r	80+r	(A) ←[XT][(Rr)] r=0,1		1
	MOVP	A,@A	10100011	A3	(A) ←PRO[(PC)<11:8> • (A)]		1
	MOVP3	A,@A	11100011	E3	(A) ←PRO[(PC)<11> • 011 • (A)]		1
TC io	MOV	A,T	01000010	42	(A) ←(TR)		1
	MOV	T,A	01100010	62	(TR)←(A)		1
mu	STRT	T	01010101	55	Start Timer		1
en	STRT	CNT	01000101	45	Start counter		1
rt	STOP	TCNT	01100101	65	Stop Timer/Counter		1
o	EN	TCNTI	00100101	25	Enable Timer/Counter Interrupt		1
r	DIS	TCNTI	00110101	35	Disable Timer/Counter Interrupt		1
(5)							
CPU Control	EN	I	00000101	05	Enable External Interrupt		1
	DIS	I	00010101	15	Disable External Interrupt		1
	SEL	RB0	11000101	C5	(BS)← 0		1
	SEL	RB1	11010101	D5	(BS)← 1		1
	SEL	MB0	11100101	E5	(DBF) ← 0		1
	SEL	MB1	11110101	F5	(DBF) ← 1		1
	ENTO	CLK	01110101	75	Enable Clock Output on T0		1
	HALT		00000001	01	Halt		1
(6)	NOP		00000000	00	no opration		1

(5) ····· A/D Converter Instruction (6) ····· Other

TMP80C49AP/TMP80C39AP/TMP80C49AF/TMP80C49AT/TMP80C39AT
ELECTRICAL CHARACTERISTICS
ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=70°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS
TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.8	V
VIL1	Input Low Voltage ($\overline{RESET}$)			-0.5	-	0.6	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
VIH1	Input High Voltage ($\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	
ICC2	VCC Supply Current	Normal operation	VCC=5V, fXTAL=11MHz	-	-	15	mA
ICCH2		HALT Mode	VIH=VCC-0.2V VIL=0.2V	-	-	4.0	

TMP80C49AP/TMP80C39AP/TMP80C49AF/TMP80C49AT/TMP80C39AT

AC CHARACTRISTICS

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
t	Clock Period	Note 2	1/xtal f	90.9	1000	ns
tLL	ALE Pulse Width		3.5t-170	150	-	ns
tAL	Address Setup Time (ALE)		2t-110	70	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	50	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	480	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	350	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	390	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	40	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	110	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	375	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	240	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	300	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	730	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	460	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	140	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	10	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	200	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	60	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	25	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	290	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	50	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	100	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	650	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	140	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	250	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	40	-	ns

TMP80C49AP/TMP80C39AP/TMP80C49AF/TMP80C49AT/TMP80C39AT

AC CHARACTRISTICS (CONTINUE)
TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	700	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	160	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	15	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	510	ns
t0PRR	T0 Clock Period		3t	270	-	ns
tCY	Cycle Time		15t	1.36	15.0	μs

Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 11MHz. and the Min. Clock frequency is 1MHz.

TMP80C49AP-6/TMP80C39AP-6/TMP80C49AF-6 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=85°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS (I)

TOPR = -40°C to 85°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage			-0.5	-	0.8	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
V_{IH1}	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current ($\overline{T1}$, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	

TMP80C49AP-6/TMP80C39AP-6/TMP80C49AF-6 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (II)

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage			-0.5	-	0.15 x VCC	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.5 x VCC	-	VCC	V
VIH1	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current ($\overline{T1}$, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-VCC/0.1	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-VCC/0.01	μA
ILO	Output Leak Current(BUS,TO) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal Operation	VCC=5V, fXTAL=6MHz	-	-	10	mA
ICCH1		HALT Mode	VIH=VCC-0.2V, VIH=0.2V	-	-	2.5	

TMP80C49AP-6/TMP80C39AP-6/TMP80C49AF-6

AC CHARACTRISTICS

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
t	Clock Period	Note 2	1/xtal f	166.6	1000	ns
tLL	ALE Pulse Width		3.5t-170	410	-	ns
tAL	Address Setup Time (ALE)		2t-110	220	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	120	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	1050	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	800	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	880	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	120	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	220	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	800	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	550	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	680	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	1500	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	1000	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	290	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	40	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	420	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	175	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	100	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	590	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	170	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	400	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	1290	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	250	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	710	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	160	-	ns

TMP80C49AP-6/TMP80C39AP-6/TMP80C49AF-6

AC CHARACTRISTICS (CONTINUE)

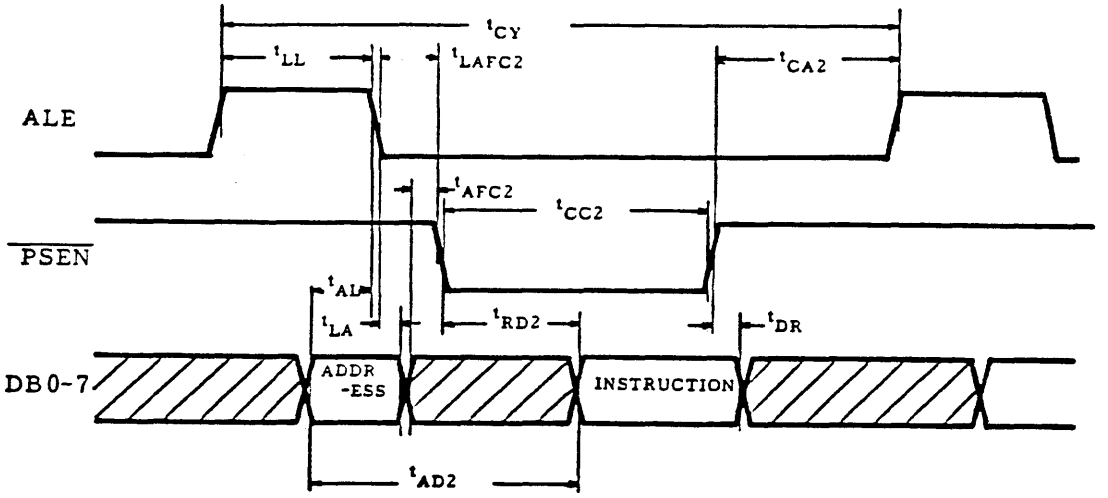
TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	1500	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	460	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	130	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	850	ns
tOPRR	T0 Clock Period		3t	500	-	ns
tCY	Cycle Time		15t	2.5	15.0	μs

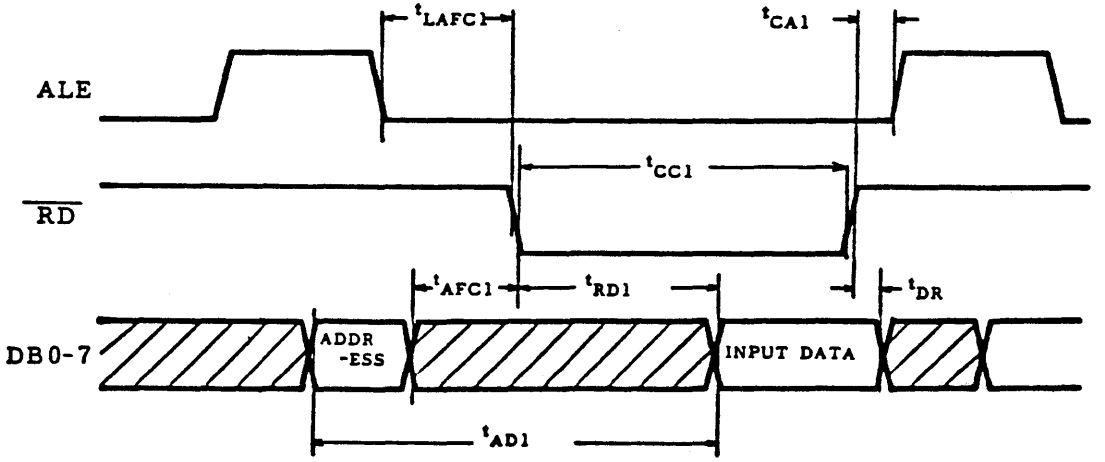
Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 6MHz. and the Min. Clock frequency is 1MHz.

TIMING WAVEFORM

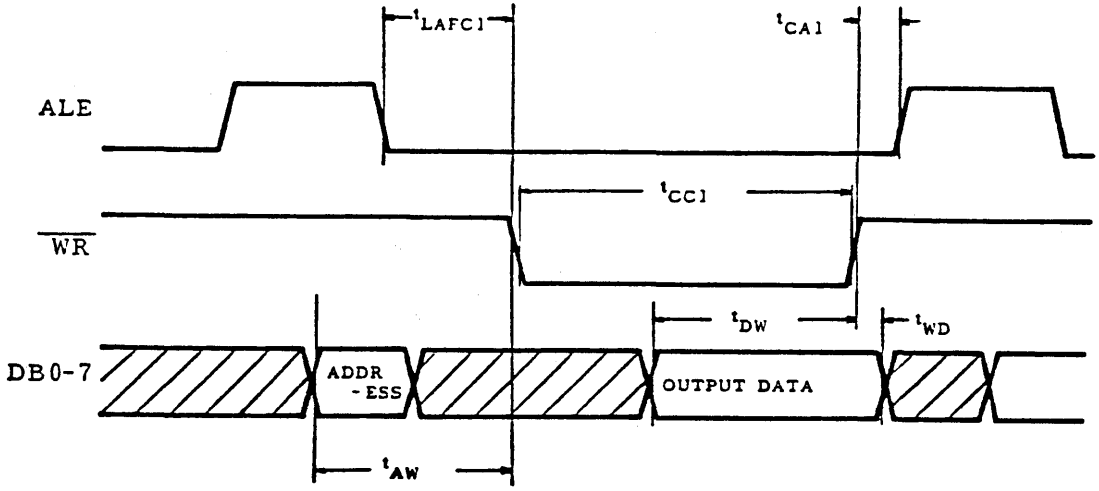
A. Instruction Fetch from External Program Memory



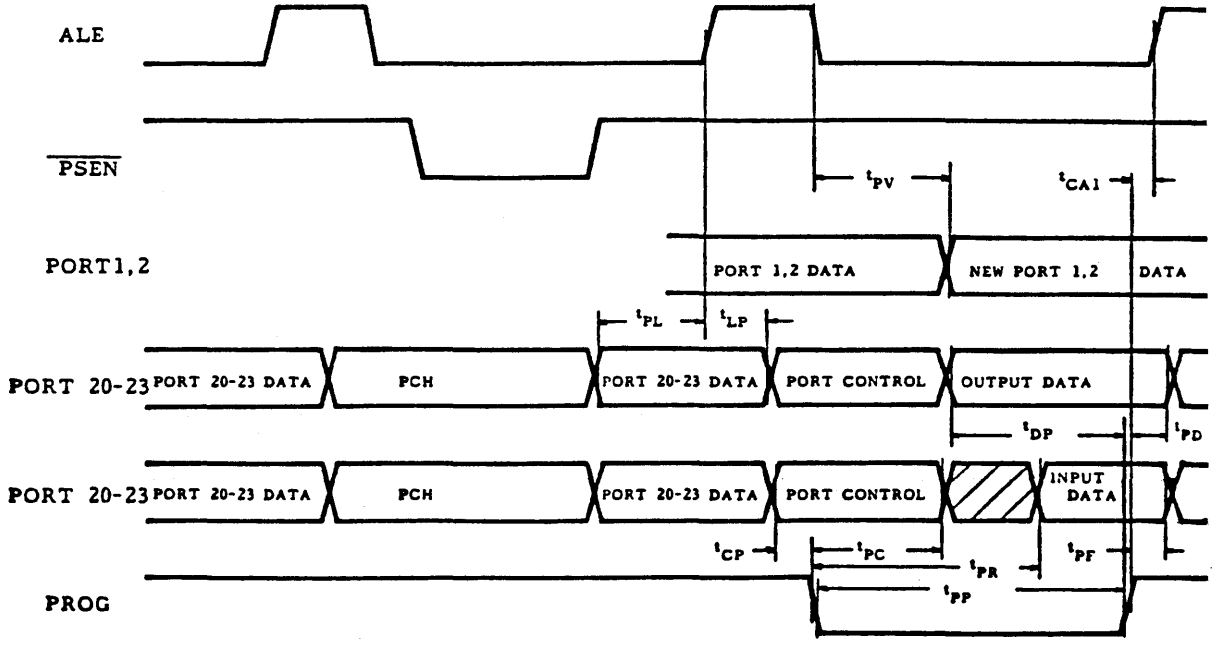
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



POWER DOWN MODE (I) ..... Data Hold Mode in RAM

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{RESET}$ terminal has been set to low level. Consequently, all the data in RAM area can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{RESET}$ terminal is set to high level, thus, the normal mode is restarted from the initialize operation (address 0).

DC CHARACTERISTICS

TMP80C49AP/39AP,TMP80C49AF,TMP80C49AT/35AT: TOPR=0°C to 70°C, VSS=0V
TMP80C49AP-6/39AP-6,TMP80C49AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB1	Standby Voltage(1)		2.0	-	6.0	V
ISB1	Standby Current(1)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

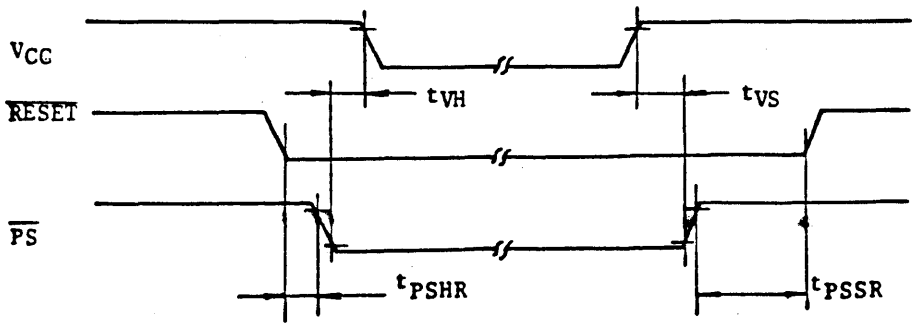
AC CHARACTERISTICS

TMP80C49AP/39AP,TMP80C49AF,TMP80C49AT/39AT:TOPR=0°C to 70°C,VCC=5V±10%,VSS=0V
TMP80C49AP-6/39AP-6,TMP80C49AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHR	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μS
tPSSR	Power Save Setup Time ($\overline{RESET}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



POWER DOWN MODE (II) ..... ALL Data Hold Mode

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{SS}$ terminal has been set to low level. Consequently, all data can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 3V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{SS}$ terminal is set to high level, thus, the normal mode is restarted continuously from the state just before the power down mode (II).

DC CHARACTERISTICS

TMP80C49AP/39AP,TMP80C49AF,TMP80C49AT/39AT: TOPR=0°C to 70°C, VSS=0V
TMP80C49AP-6/39AP-6,TMP80C49AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB2	Standby Voltage(2)		3.0	-	6.0	V
ISB2	Standby Current(2)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

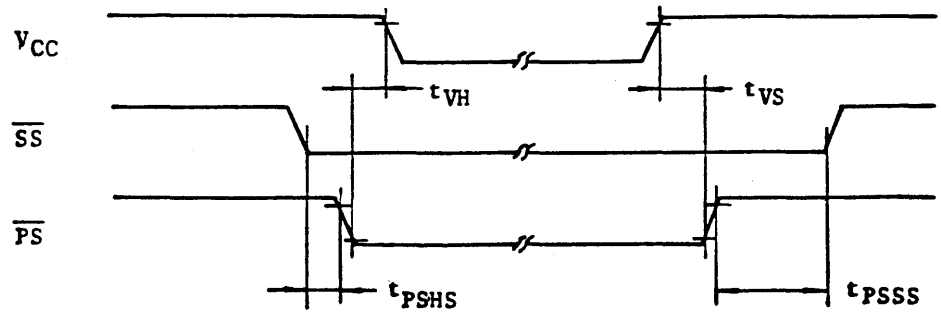
AC CHARACTERISTICS

TMP80C49AP/39AP,TMP80C49AF,TMP80C49AT/39AT:TOPR=0°C to 70°C,VSS=5V±10%,VSS=0V
TMP80C49AP-6/39AP-6,TMP80C49AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHS	Power Save Hold Time ($\overline{SS}$)		10	-	-	μS
tPSSS	Power Save Setup Time ($\overline{SS}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



HALT MODE

. 1 HALT INSTRUCTION

OP code is "01H". HALT INSTRUCTION is an additional instruction to the standard 8048/8049 instruction set.

. 2 Entry to HALT MODE

On the execution of HALT INSTRUCTION, TMP80C49A/TMP80C39A enter HALT MODE.

. 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logic values just prior to the execution of HALT INSTRUCTION are maintained. Power consumption in HALT MODE is less than 50% of normal operation. The status of each pins are described in the following table.

. 4 Release from HALT MODE

HALT MODE is released by either of two signals ($\overline{\text{RESET}}$,$\overline{\text{INT}}$).

(1) $\overline{\text{RESET}}$ Release Mode : An active $\overline{\text{RESET}}$ input signal causes the normal reset function. TMP80C49A/TMP80C39A start the program at address "000 H".

(2) $\overline{\text{INT}}$ Release Mode : An active $\overline{\text{INT}}$ input signal causes the normal operation.

In case of interrupt enable mode (EI MODE), TMP80C49A/TMP80C39A execute the interrupt service routine, after the execution of one instruction which is located at the next address after HALT INSTRUCTION.

In case of interrupt disable mode (DI MODE), TMP80C49A/TMP80C39A execute normal operation from the next address after HALT INSTRUCTION.

. 5 Supply Voltage Range in HALT MODE

The operating supply voltage range and the operating temperature range are same as in normal operation.

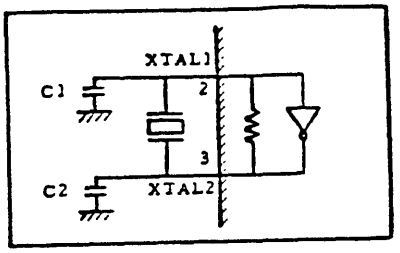
PIN STATUS IN POWER DOWN MODE (I) (II)

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	High impedance Input disabled
T0	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{\text{RESET}}$, $\overline{\text{SS}}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{\text{INT}}$, EA	Input disabled when oscilltor is stopped.
$\overline{\text{RD}}$, $\overline{\text{WR}}$, ALE PROG, $\overline{\text{PSEN}}$	High impedance

PIN STATUS IN HALT MODE

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	Values prior to the execution of HALT INSTRUCTION are maintained.
T0	Status prior to the execution of HALT INSTRUCTION is maintained.
T1	Input disabled
XTAL1, XTAL2	Continue oscillation
$\overline{\text{RESET}}$, $\overline{\text{INT}}$	Input enabled
$\overline{\text{SS}}$, $\overline{\text{EA}}$	Input disabled
$\overline{\text{RD}}$, $\overline{\text{WR}}$, PROG, $\overline{\text{PSEN}}$	Output "High" level
ALE	Output "Low" level

OSCILATOR

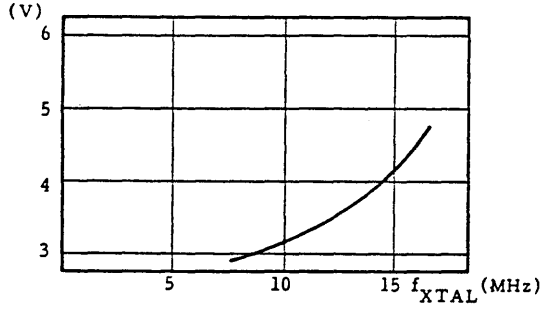


QUARTZ CRYSTAL
f = 1MHz to 4MHz : C1 = C2 = 30pF
f = 4MHz to 11MHz : C1 = C2 = 20pF

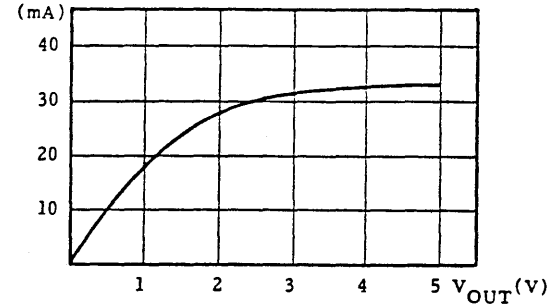
CERAMIC RESONATOR
f = 1MHz to 3MHz : C1 = C2 = 100pF
f = 3MHz to 11MHz : C1 = C2 = 30pF

TYPICAL CHARACTARISTICS : Vcc = 5V, Ta = 25°C, unless Otherwise noted.

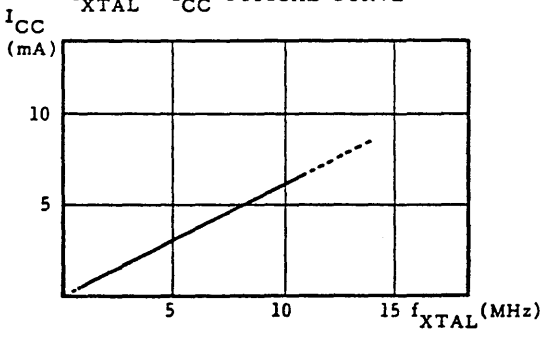
V_{CC} - $f_{MAX.}$ TYPICAL CURVE



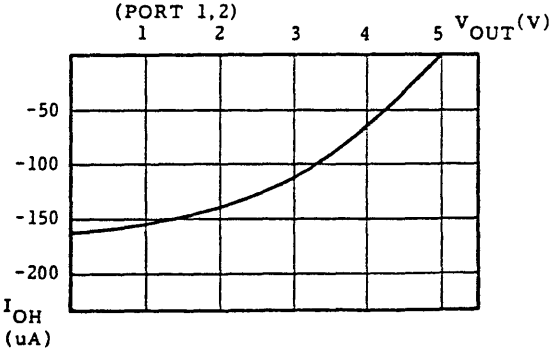
V_{OUT} - I_{OL} TYPICAL CURVE



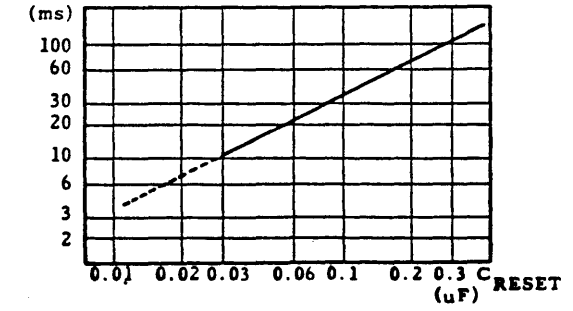
f_{XTAL} - I_{CC} TYPICAL CURVE



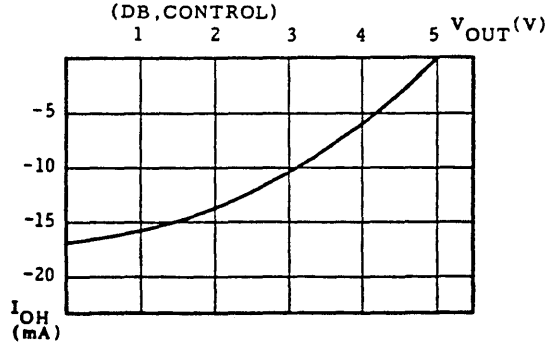
V_{OUT} - I_{OH} TYPICAL CURVE
(PORT 1,2)



t_{RESET} C_{RESET} - t_{RESET} TYPICAL CURVE

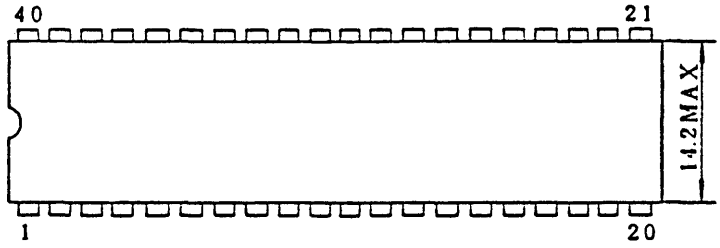


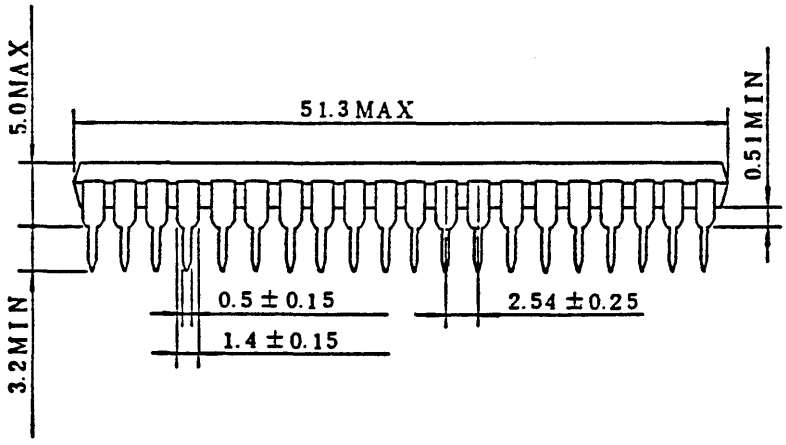
V_{OUT} - I_{OH} TYPICAL CURVE
(DB, CONTROL)

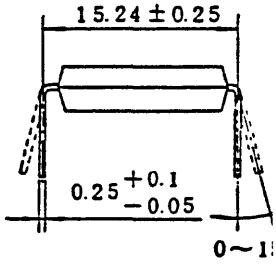


OUTLINE DRAWING (DUAL INLINE PACKAGE)

Unit in mm

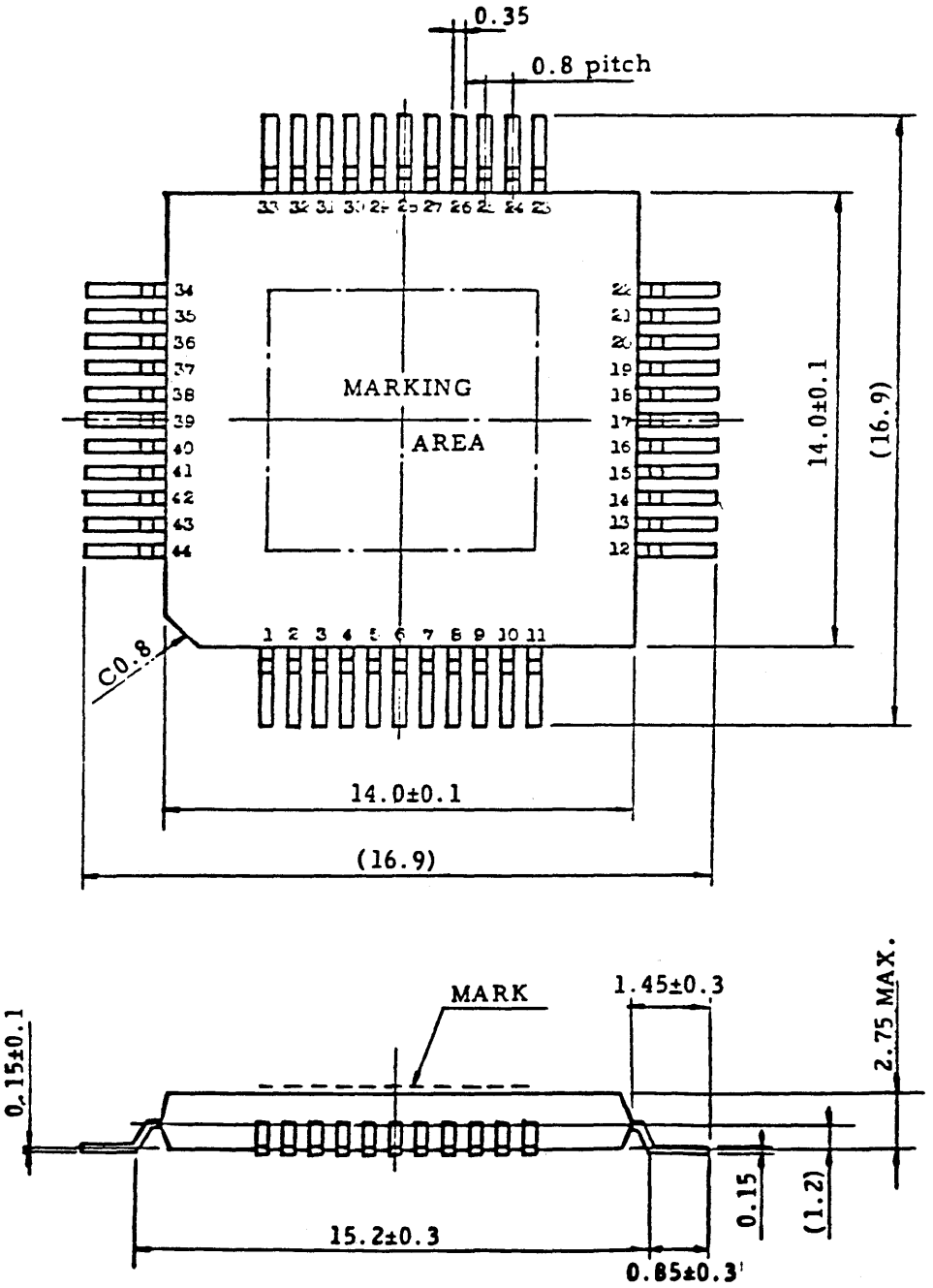






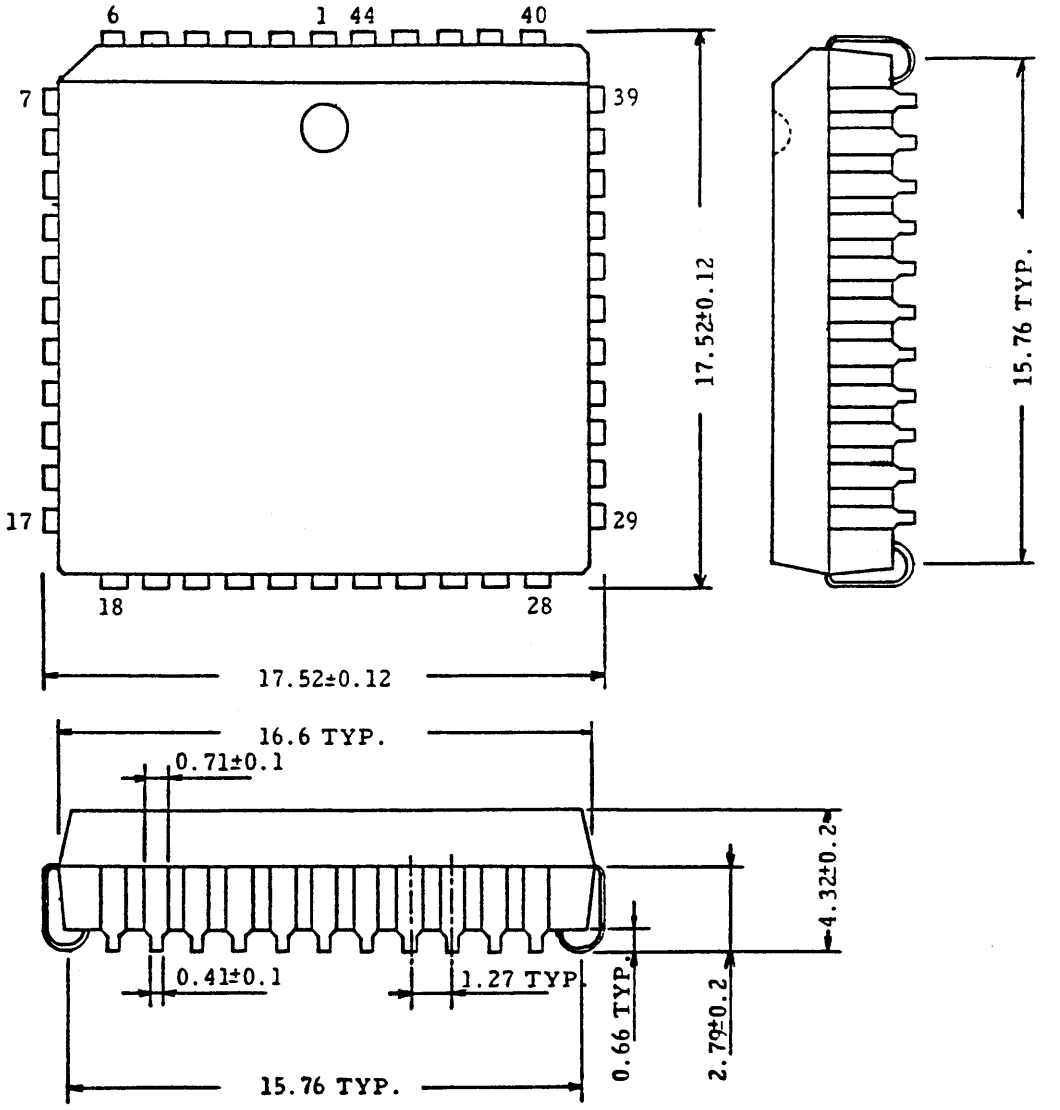
Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

OUTLINE DRAWING (FLAT PACKAGE)



OUTLINE DRAWING (Plastic Leaded Chip Carrier)

unit in mm



8-BIT SINGLE-CHIP MICROCOMPUTER

TMP80C50AP/TMP80C50AP-6
TMP80C40AP/TMP80C40AP-6
TMP80C50AF/TMP80C50AF-6
TMP80C50AT/TMP80C40AT

GENERAL DESCRIPTION

The TMP80C50A is a single chip microcomputer fabricated in Silicon Gate CMOS technology which provides internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, 256 x 8 RAM data memory, 4K x 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

The TMP80C50A is particularly efficient as a controller. It has extensive bit TOSHIBAg capability as well as facilities for both binary and BCD arithmetic.

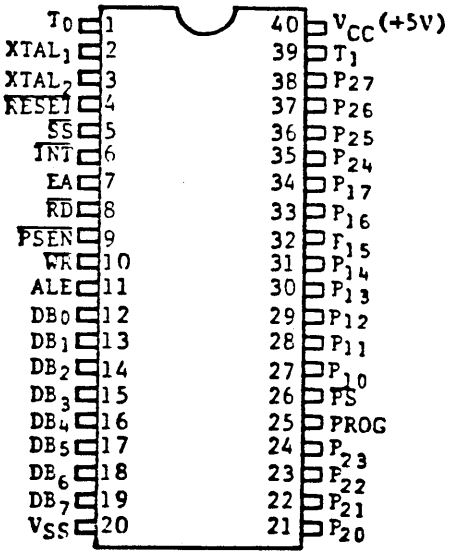
The TMP80C40A/-6 is the equivalent of a TMP80C50A/-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP80C50AP/-6 and TMP80C40AP/-6 are in a standerd Dual Inline Package.
The TMP80C50AF/-6 is in a 44-pin Flat Package.
The TMP80C50AT and TMP80C40AT are packaged in the JEDEC standard type 44pin PLCC (Plastic Leaded Chip Carrier).

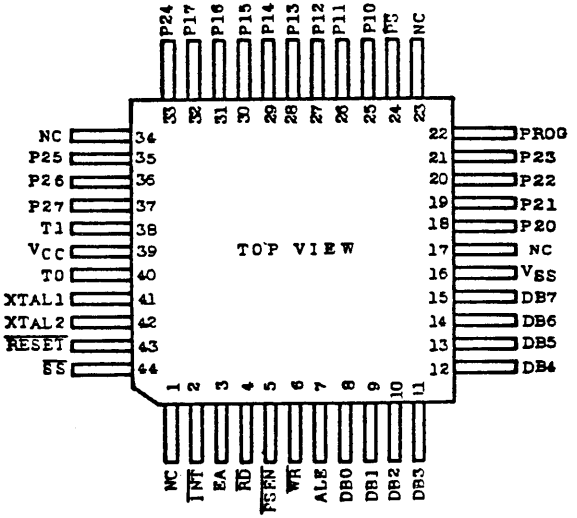
FEATURES

- TMP80C50AP/TMP80C40AP/TMP80C50AF/ TMP80C50AT/TMP80C40AT
  1.36μs Instruction Cycle Time
  0°C to 70°C, 5V ± 10%
- TMP80C50AP-6/TMP80C40AP-6/TMP80C50AF-6
  2.5 μs Instruction Cycle Time
  -40°C to 85°C, 5V ± 20%
- Software Upward Compatible with TMP8049AP/TMP80C49AP-6/INTEL's 8049.
- HALT Instruction (Additional Instruction)
- 4K x 8 masked ROM
- 256 x 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Low Power
  10mA MAX. in Normal Operation
  (VCC=5V, fXTAL=6MHz)
  10μA Max. in Power Down Mode
  (VCC=5V, fXTAL : DC)
- Single Power Supply
- Power Down Mode (Stand-by Mode)
- Halt Mode (Idle Mode)

PIN CONNECTIONS (TOP VIEW)

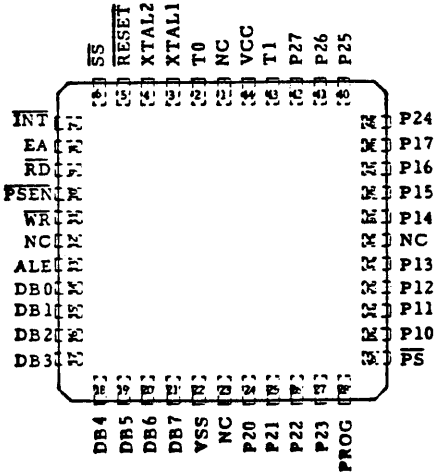


Flat Package

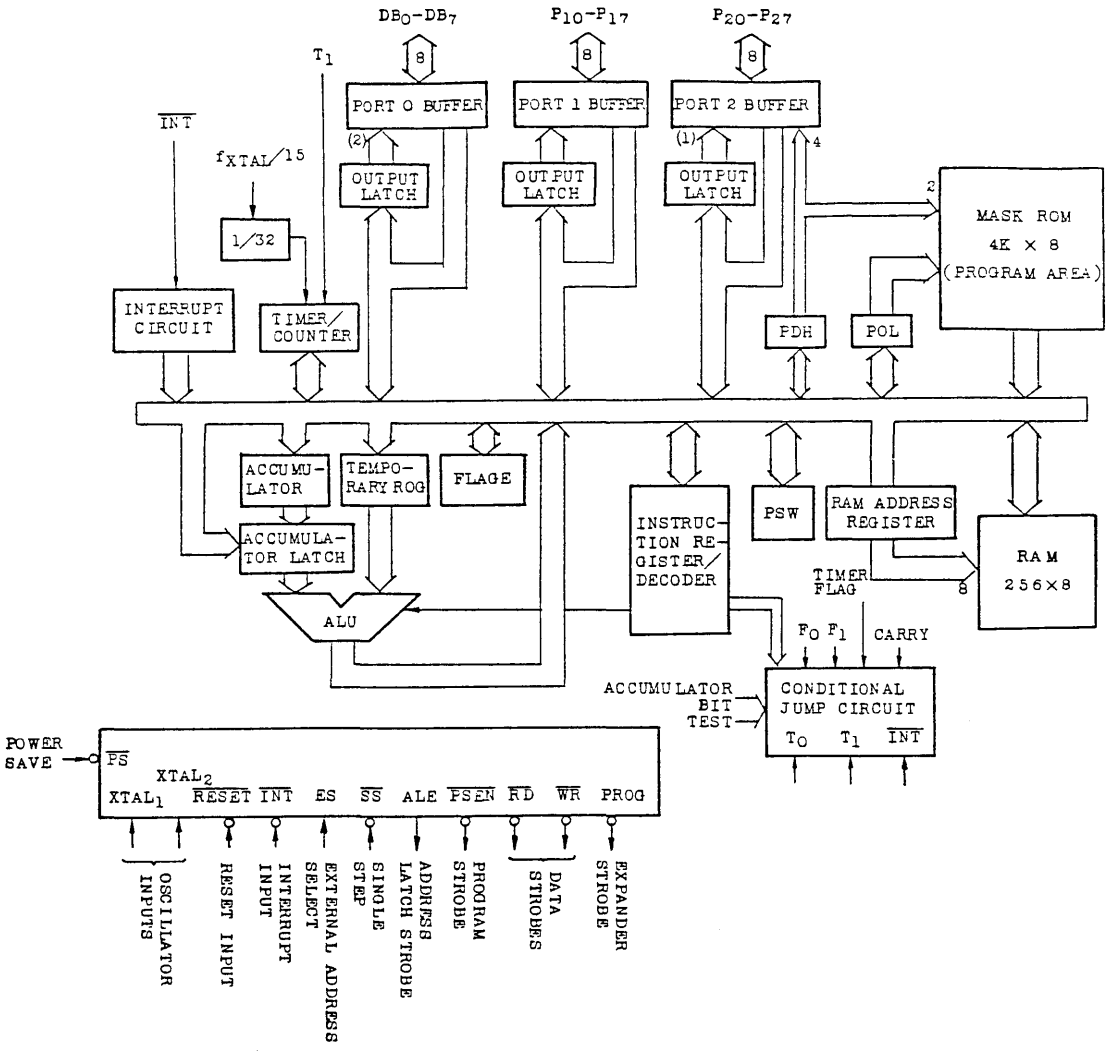


NC: No Connection

PLCC



BLOCK DIAGRAM



Note 1) The lower order 4 bit of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VCC (Power Supply)
+5V during operation

$\overline{PS}$ (Input)
The control signal for the power saving at the power down mode (Active Low)

PROG (Output)
Output strobe for the TMP82C43P I/O expander.

P10 - P17 (Input/Output) Port 1
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P27 (Input/Output) Port 2
8-bit quasi-bidirectional port (Internal Pullup = 50KΩ).

P20 - P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0 - DB7 (Input/Output, Tri-State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low). Used as a Write Strobe to External Data Memory.

$\overline{\text{RESET}}$ (Input)
Active Low signal which is used to initialize the Processor. Also used during the power down mode.

ALE (Oupput)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being executed. Also used during the power down mode.

EA (Input)
External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High)

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

TLCS-48 LIST OF INSTRUCTIONS (1/4)

Instruction Class	Assembler Mnemonic	Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C, AC	Cycle
Accumulator Instruction	ADD A, Rr	01101rrr	68+r	(A)←(A)+(Rr) r=0~7	↕ ↕	1
	ADD A, @Rr	0110000r	60+r	(A)←(A)+[(Rr)] r=0,1	↕ ↕	1
	ADD A, #i	00000011 iiiiiiii	03 ii	(A)←(A)+i	↕ ↕	2
	ADDC A, Rr	01111rrr	78+r	(A)←(A)+(Rr)+(C) r=0~7	↕ ↕	1
	ADDC A, @Rr	0111000r	70+r	(A)←(A)+[(Rr)]+(C) r=0,1	↕ ↕	1
	ADDC A, #i	00010011 iiiiiiii	13 ii	(A)←(A)+i+(C)	↕ ↕	2
	ANL A, Rr	01011rrr	58+r	(A)←(A) ∧(Rr) r=0~7		1
	ANL A, @Rr	0101000r	50+r	(A)←(A) ∧[(Rr)] r=0,1		1
	ANL A, #i	01010011 iiiiiiii	53 ii	(A)←(A) ∧i		2
	ORL A, Rr	01001rrr	48+r	(A)←(A) ∨(Rr) r=0~7		1
	ORL A, @Rr	0100000r	40+r	(A)←(A) ∨[(Rr)] r=0,1		1
	ORL A, #i	01000011 iiiiiiii	43 ii	(A)←(A) ∨i		2
	XRL A, Rr	11011rrr	D8+r	(A)←(A) ∀(Rr) r=0~7		1
	XRL A, @Rr	1101000r	D0+r	(A)←(A) ∀[(Rr)] r=0,1		1
	XRL A, #i	11010011 iiiiiiii	D3 ii	(A)←(A) ∀i		2
	INC A	00010111	17	(A)←(A)+1		1
	DEC A	00000111	07	(A)←(A)-1		1
	CLR A	00100111	27	(A)←0		1
	CPL A	00110111	37	(A)←NOT(A)		1
	DA A	01010111	57	(A)←(A)BCD	↕	1
	SWAP A	01000111	47	(A)<7:4> ↔(A)<3:0>		1
	RL A	11100111	E7	(A)<n+1> ←(A)<n> (A)<0> ←(A)<7> n=0~6		1
	RLC A	11110111	F7	(A)<n+1> ←(A)<n> (C)←(A)<7> (A)<0> ←(C) n=0~6	↕	1
	RR A	01110111	77	(A)<n> ←(A)<n+1> n=0~6 (A)<7> ←(A)<0>		1
	RRC A	01100111	67	(A)<n> ←(A)<n+1> (C)←(A)<0> (A)<7> ←(C) n=0~6	↕	1
I/O	IN A, Pp	000010pp	08+p	(A)←(Pp) P=1,2		2

TLCS-48 LIST OF INSTRUCTIONS (2/4)

Instruction Class	Assembler Mnemonic	Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C, AC	Cycle
INPUT/OUTPUT Instruction	OUTL Pp, A	001110pp	38+p	(Pp) ←(A) P=1,2		2
	ANL Pp, #i	100110pp iiiiiiii	98+p ii	(Pp) ←(Pp)∧ i P=1,2		2
	ORL Pp, #i	100010pp iiiiiiii	88+p ii	(Pp) ←(Pp)∨ i P=1,2		2
	INS A, BUS	00001000	08	(A)←(BUS)		2
	OUTL BUS, A	00000010	02	(BUS)←(AC)		2
	ANL BUS, #i	10011000 iiiiiiii	98 ii	(BUS)←(BUS) ∧i		2
	ORL BUS, #i	10001000 iiiiiiii	88 ii	(BUS)←(BUS) ∨i		2
	MOVD A, Pp	000011pp	0C+p	(A)<3:0> ←(Pp) p=4~7 (A)<7:4> ←0		2
	MOVD Pp, A	001111pp	3C+p	(Pp) ←(A)<3:0> p=4~7		2
	ANLD Pp, A	100111pp	9C+p	(Pp) ←(Pp)∧(A)<3:0> p=4~7		2
	ORLD Pp, A	100011pp	8C+p	(Pp) ←(Pp)∨(A)<3:0> p=4~7		2
(1)	INC Rr	00011rrr	18+r	(Rr) ←(Rr)+1 r=0~7		1
(1)	INC @Rr	0001000r	10+r	[(Rr)] ←[(Rr)]+1 r=0,1		1
(1)	DEC Rr	11001rrr	C8+r	(Rr) ←(Rr)-1 r=0~7		1
Branch Instruction	JMP a	a10 0100 aHL	a1+4	(PC)<10:0> ←a (PC)<11> ←(DBF)		2
	JMPP @A	10110011	B3	(PC)<7:0>←PRO[(PC)<11:8>・(A)]		2
	DJNZ Rr, a	11101rrr aHL	E8+r	(Rr) ←(Rr)-1 r=0~7 if(Rr) ≠0then(PC)<7:0>←aHL else no operation		2
	JC a	11110110 aHL	F6	if(C)=1 then(PC)<7:0>←aHL else no operation		2
	JNC a	11100110 aHL	E6	if(C)=0 then(PC)<7:0>←aHL else no operation		2
	JZ a	11000110 aHL	C6	if(A)=0 then(PC)<7:0>←aHL else no operation		2
	JNZ a	10010110 aHL	96	if(A)≠0 then(PC)<7:0> ←aHL else no operation		2
	JTO a	00110110 aHL	36	if TO=1 then(PC)<7:0>←aHL else no operation		2

(1) ..... Register Instruction

TLCS-48 LIST OF INSTRUCTIONS (3/4)

Classes	Assembler Mnemonic	Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C,AC	Cycle
Branch Instruction	JNTO a	00100110 aML	26	if T0=0 then(PC)<7:0>←aML else no operation		2
	JT1 a	01010110 aML	56	if T1=1 then(PC)<7:0>←aML else no operation		2
	JNT1 a	01000110 aML	46	if T1=0 then(PC)<7:0>←aML else no operation		2
	JF0 a	10110110 aML	B6	if F0=1 then(PC)<7:0>←aML else no operation		2
	JF1 a	01110110 aML	76	if F1=1 then(PC)<7:0>←aML else no operation		2
	JTF a	00010110 aML	16	if TF=1 then(PC)<7:0>←aML else no operation		2
	JNI a	10000110 aML	86	if INT =0 then(PC)<7:0>←aML else no operation		2
	JBb a	bbb10010 aML	b*12	if (A)<b>=1 then (PC)<7:0>←aML else no operation b=0~7		2
(2)	CALL a	aH10100 aML	aH*14	[(SP)] ←(PSW)<7:4> · (PC) (SP) ←(SP)+1 (PC)<10:0> ←a (PC)<11> ←(DBF)		2
	RET	10000011	83	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0>		2
	RETR	10010011	93	(SP) ←(SP)-1 (PC) ←[(SP)]<11:0> (PSW)<7:4> ←[(SP)]<15:12>	↕ ↕	2
(3)	CLR C	10010111	97	(C)←0		1
	CPL C	10100111	A7	(C)←NOT(C)		1
	CLR F0	10000101	85	(F0) ←0		1
	CPL F0	10010101	95	(F0) ←NOT(F0)		1
	CLR F1	10100101	A5	(F1) ←0		1
	CPL F1	10110101	B5	(F1) ←NOT(F1)		1
(4)	MOV A , Rr	11111rrr	F8+r	(A)←(Rr) r=0~7		1
	MOV A ,@Rr	1111000r	F0+r	(A)←[(Rr)] r=0,1		1
	MOV A ,#i	00100011 iiiiiiii	23 ii	(A) ←i		2
	MOV Rr, A	10101rrr	A8+r	(Rr)←(A) r=0~7		1
	MOV @Rr, A	1010000r	A0+r	[(Rr)]←(A) r=0,1		1

(2) ····· Subroutine Instruction (3) ····· Flag Instruction
(4) ····· Move Instruction

TLCS-48 LIST OF INSTRUCTIONS (4/4)

Classes	Assembler Mnemonic	Object Code (1st) Bin.	Object Code (2nd) Hex.	Function	Flag C,AC	Cycle
Move Instruction	MOV Rr,#i	10111rrr iiiiiiii	B8+r ii	(Rr)←i r=0~7		2
	MOV @Rr,#i	1011000r iiiiiiii	B0+r ii	[(Rr)]←i r=0,1		2
	MOV A,PSW	11000111	C7	(A) ←(PSW)		1
	MOV PSW,A	11010111	D7	(PSW) ←(A)		1
	XCH A,Rr	00101rrr	28+r	(A) ↔(Rr) r=0~7		1
	XCH A,@Rr	0010000r	20+r	(A) ↔[(Rr)] r=0,1		1
	XCHD A,@Rr	0011000r	30+r	(A)<3:0>↔[(Rr)<3:0>] r=0,1		1
	MOVX @Rr,A	1001000r	90+r	EXT[(Rr)] ←(A) r=0,1		1
	MOVX A,@Rr	1000000r	80+r	(A) ←[XT][(Rr)] r=0,1		1
	MOVP A,@A	10100011	A3	(A) ←PRO[(PC)<11:8> · (A)]		1
	MOVP3 A,@A	11100011	E3	(A) ←PRO[(PC)<11> · 011 · (A)]		1
(5)	MOV A,T	01000010	42	(A) ←(TR)		1
	MOV T,A	01100010	62	(TR)←(A)		1
	STRT T	01010101	55	Start Timer		1
	STRT CNT	01000101	45	Start counter		1
	STOP TCNT	01100101	65	Stop Timer/Counter		1
	EN TCNTI	00100101	25	Enable Timer/Counter Interrupt		1
	DIS TCNTI	00110101	35	Disable Timer/Counter Interrupt		1
Control	EN I	00000101	05	Enable External Interrupt		1
	DIS I	00010101	15	Disable External Interrupt		1
	SEL RB0	11000101	C5	(BS)← 0		1
	SEL RB1	11010101	D5	(BS)← 1		1
	SEL MB0	11100101	E5	(DBF) ← 0		1
	SEL MB1	11110101	F5	(DBF) ← 1		1
	ENTO CLK	01110101	75	Enable Clock Output on T0		1
	HALT	00000001	01	Halt		1
(6)	NOP	00000000	00	no opration		1

(5) ····· A/D Converter Instruction (6) ····· Other

TMP80C50AP/TMP80C40AP/TMP80C50AF/TMP80C50AT/TMP80C40AT
ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=70°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.8	V
VIL1	Input Low Voltage ($\overline{RESET}$)			-0.5	-	0.6	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
VIH1	Input High Voltage ($\overline{RESET}$, $\overline{PS}$)			0.7 VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	
ICC2	VCC Supply Current	Normal operation	VCC=5V, fXTAL=11MHz	-	-	15	mA
ICCH2		HALT Mode	VIH=VCC-0.2V VIL=0.2V	-	-	4.0	

TMP80C50AP/TMP80C40AP/TMP80C50AF/TMP80C50AT/TMP80C40AT

AC CHARACTRISTICS

TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
t	Clock Period	Note 2	1/xtal f	90.9	1000	ns
tLL	ALE Pulse Width		3.5t-170	150	-	ns
tAL	Address Setup Time (ALE)		2t-110	70	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	50	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	480	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	350	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	390	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	40	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	110	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	375	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	240	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	300	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	730	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	460	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	140	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	10	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	200	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	60	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	25	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	290	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	50	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	100	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	650	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	140	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	250	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	40	-	ns

TMP80C50AP/TMP80C40AP/TMP80C50AF/TMP80C50AT/TMP80C40AT

AC CHARACTRISTICS (CONTINUE)
TOPR=0°C to 70°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	11 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	700	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	160	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	15	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	510	ns
t0PRR	T0 Clock Period		3t	270	-	ns
tCY	Cycle Time		15t	1.36	15.0	μs

Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 11MHz. and the Min. Clock frequency is 1MHz.

TMP80C50AP-6/TMP80C40AP-6/TMP80C50AF-6 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to +7V
VINA	Input Voltage (Except EA)	-0.5V to VCC+0.5V
VINB	Input Voltage (Only EA)	-0.5V to 13V
PD	Power Dissipation (Ta=85°C)	250mW
TSOLDER	Soldering Temperature (Soldering Timer 10 sec)	260°C
TSTG	Storage Temperature	-65°C to 150°C
TOPR	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS (I)

TOPR = -40°C to 85°C, VCC=+5V±10%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
$\overline{VIL}$	Input Low Voltage			-0.5	-	0.8	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	VCC	V
VIH1	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH11	Output High Voltage (Except P10-P17, P20-P27)		IOH=-1.6mA	2.4	-	-	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH21	Output High Voltage (P10-P17, P20-P27)		IOH=-50μA	2.4	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current ($\overline{T1}$, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-50	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-500	μA
ILO	Output Leak Current(BUS,$\overline{TO}$) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal operation	VCC=5V,fXTAL=6MHz VIH=VCC-0.2V	-	-	10	mA
ICCH1		HALT Mode	VIL=0.2V	-	-	2.5	

TMP80C50AP-6/TMP80C40AP-6/TMP80C50AF-6 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (II)

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage			-0.5	-	0.15 x VCC	V
VIH	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.5 x VCC	-	VCC	V
VIH1	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7 x VCC	-	VCC	V
VOL	Output Low Voltage (Except P10-P17, P20-P27)		IOL=1.6mA	-	-	0.45	V
VOL1	Output Low Voltage (P10-P17, P20-P27)		IOL=1.2mA	-	-	0.45	V
VOH12	Output High Voltage (Except P10-P17, P20-P27)		IOH=-400μA	VCC-0.8	-	-	V
VOH22	Output High Voltage (P10-P17, P20-P27)		IOH=-25μA	VCC-0.8	-	-	V
ILI	Input Leak Current ($\overline{T1}$, $\overline{INT}$, EA, $\overline{PS}$)		VSS≦VIN≦VCC	-	-	±10	μA
ILI1	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		VSS≦VIN≦VCC	-	-	-VCC/0.1	μA
ILI2	Input Leak Current (P10-P17, P20-P27)		VSS+0.45V≦VIN≦VCC	-	-	-VCC/0.01	μA
ILO	Output Leak Current (BUS, $\overline{TO}$) (High impedance condition)		VSS+0.45V≦VIN≦VCC	-	-	±10	μA
ICC1	VCC Supply Current	Normal Operation	VCC=5V, fXTAL=6MHz VIH=VCC-0.2V, VIH=0.2V	-	-	10	mA
ICCH1		HALT Mode		-	-	2.5	

TMP80C50AP-6/TMP80C40AP-6/TMP80C50AF-6

AC CHARACTRISTICS

TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
t	Clock Period	Note 2	1/xtal f	166.6	1000	ns
tLL	ALE Pulse Width		3.5t-170	410	-	ns
tAL	Address Setup Time (ALE)		2t-110	220	-	ns
tLA	Address Hold Time (ALE)	CL=20pF	t-40	120	-	ns
tCC1	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)		7.5t-200	1050	-	ns
tCC2	Control Pulse Width ($\overline{PSEN}$)		6t-200	800	-	ns
tDW	Data Setup Time ($\overline{WR}$)		6.5t-200	880	-	ns
tWD	Data Hold Time ($\overline{WR}$)	CL=20pF	t-50	120	-	ns
tDR	Data Hold Time ($\overline{RD}$, $\overline{PSEN}$)	CL=20pF	1.5t-30	0	220	ns
tRD1	Data Input Read Time ($\overline{RD}$)		5.5t-120	-	800	ns
tRD2	Data Input Read Time ($\overline{PSEN}$)		4t-120	-	550	ns
tAW	Address Setup Time ($\overline{WR}$)		5t-150	680	-	ns
tAD1	Address Setup Time ($\overline{RD}$)		10t-170	-	1500	ns
tAD2	Address Setup Time ($\overline{PSEN}$)		7t-170	-	1000	ns
tAFC1	Address Float Time ($\overline{RD}$, $\overline{WR}$)	CL=20pF	2t-40	290	-	ns
tAFC2	Address Float Time ($\overline{PSEN}$)	CL=20pF	0.5t-40	40	-	ns
tLAFC1	ALE to Control Time ($\overline{RD}$, $\overline{WR}$)		3t-75	420	-	ns
tLAFC2	ALE to Control Time ($\overline{PSEN}$)		1.5t-75	175	-	ns
tCA1	Control to ALE Time ($\overline{RD}$, $\overline{WR}$, PROG)		t-65	100	-	ns
tCA2	Control to ALE Time ($\overline{PSEN}$)		4t-70	590	-	ns
tCP	Port Control Setup Time (PROG)		1.5t-80	170	-	ns
tPC	Port Control Hold Time (PROG)		4t-260	400	-	ns
tPR	Port 2 Input Data Setup Time (PROG)		8.5t-120	-	1290	ns
tPF	Port 2 Input Data Hold Time (PROG)		1.5t	0	250	ns
tDP	Port 2 Output Data Setup Time (PROG)		6t-290	710	-	ns
tPD	Port 2 Output Data Hold Time (PROG)		1.5t-90	160	-	ns

TMP80C50AP-6/TMP80C40AP-6/TMP80C50AF-6

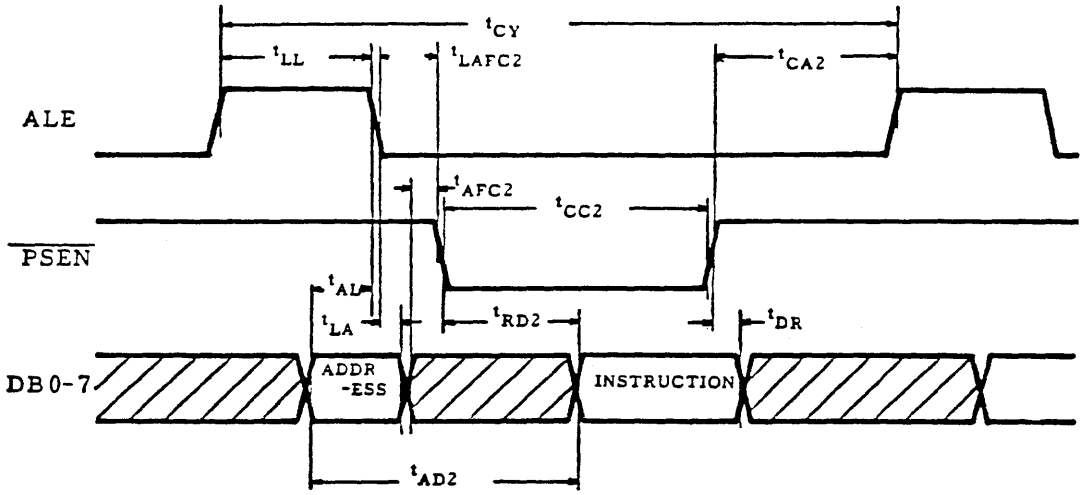
AC CHARACTRISTICS (CONTINUE)
TOPR=-40°C to 85°C, VCC=+5V±20%, VSS=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITION	f(t)	6 MHz		UNIT
				MIN.	MAX.	
tPP	PROG Pulse Width		10.5t-250	1500	-	ns
tPL	Port 2 I/O Data Setup Time(ALE)		4t-200	460	-	ns
tLP	Port 2 I/O Data Hold Time (ALE)		0.5t-30	130	-	ns
tPV	Port Output Delay Time (ALE)		4.5t+100	-	850	ns
t0PRR	T0 Clock Period		3t	500	-	ns
tCY	Cycle Time		15t	2.5	15.0	μs

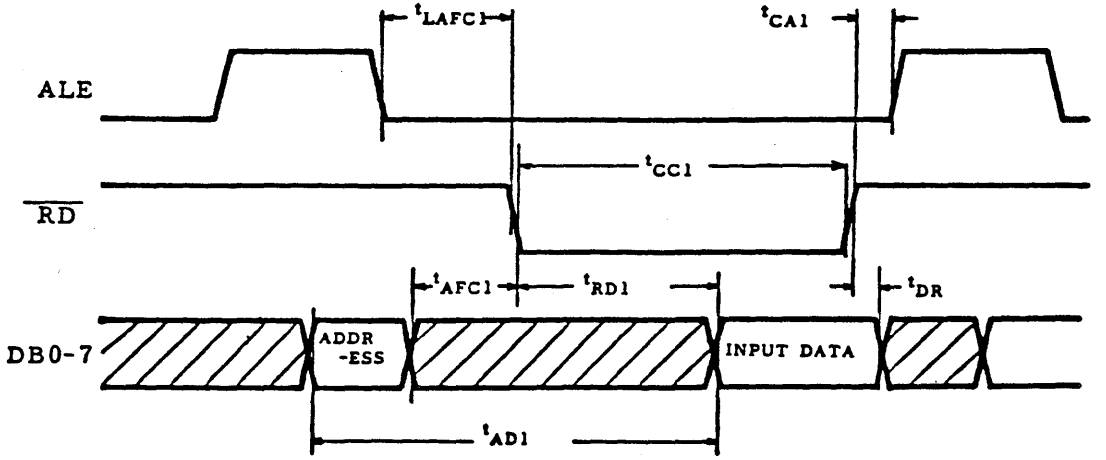
Note : 1. Control Output CL=80pF. BUS Output CL=150pF.
2. The f(t) assumes 50% duty cycle on XTAL1 and XTAL2.
The Max. Clock frequency is 6MHz. and the Min. Clock frequency is 1MHz.

TIMING WAVEFORM

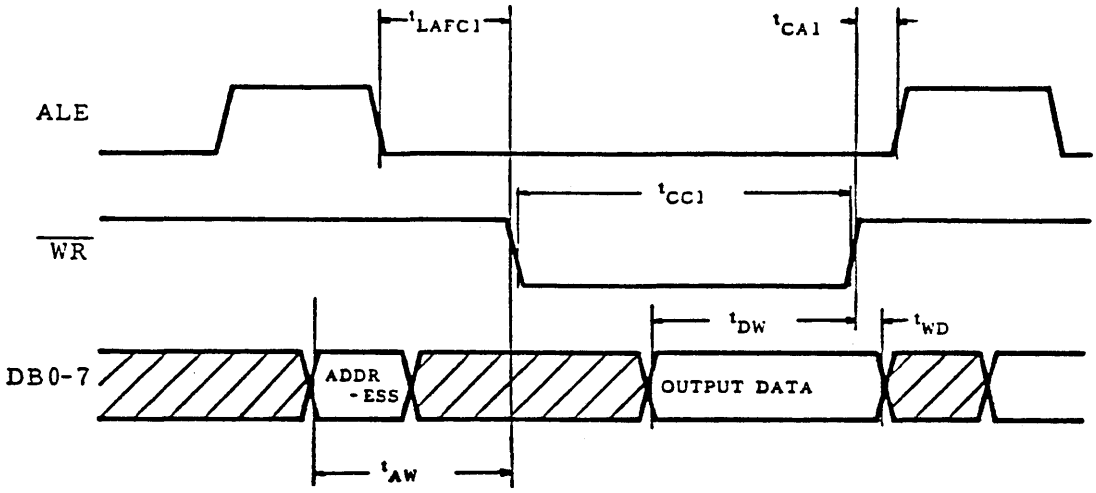
A. Instruction Fetch from External Program Memory



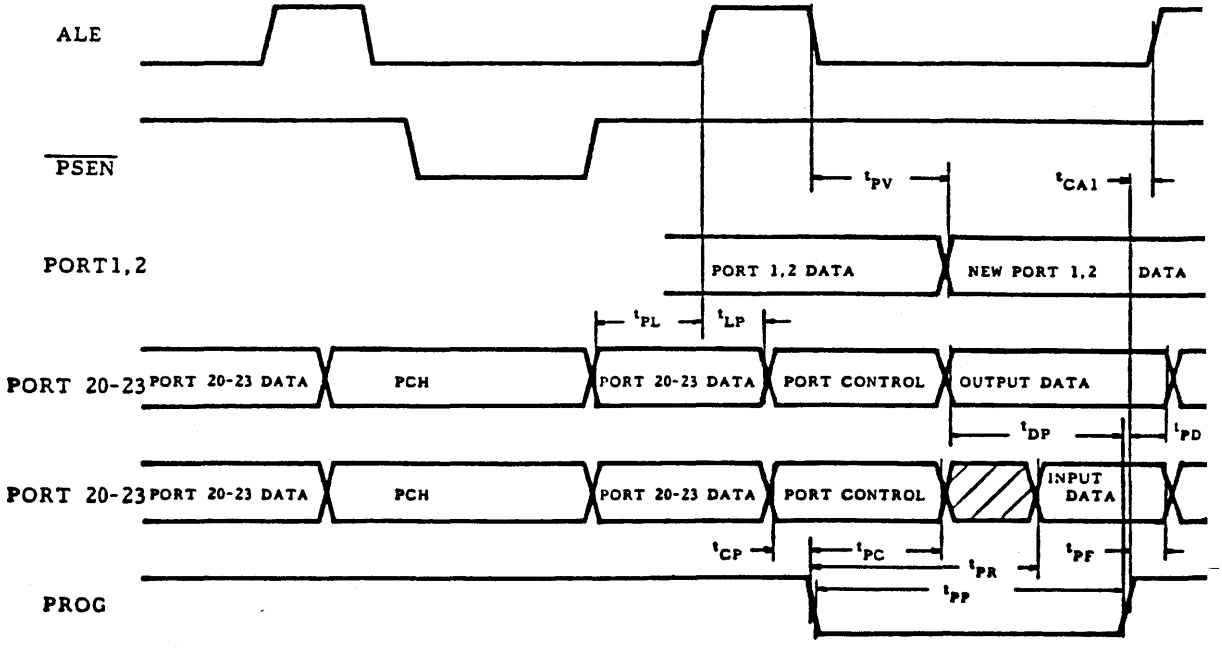
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



POWER DOWN MODE (I) ..... Data Hold Mode in RAM

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{RESET}$ terminal has been set to low level. Consequently, all the data in RAM area can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{RESET}$ terminal is set to high level, thus, the normal mode is restarted from the initialize operation (address 0).

DC CHARACTERISTICS

TMP80C50AP/40AP,TMP80C50AF,TMP80C50AT/40AT: TOPR=0°C to 70°C, VSS=0V
TMP80C50AP-6/40AP-6,TMP80C50AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB1	Standby Voltage(1)		2.0	-	6.0	V
ISB1	Standby Current(1)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

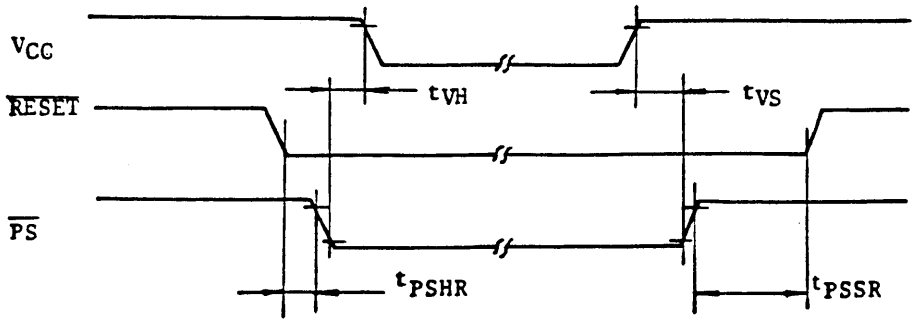
AC CHARACTERISTICS

TMP80C50AP/40AP,TMP80C50AF,TMP80C50AT/40AT:TOPR=0°C to 70°C,VCC=5V±10%,VSS=0V
TMP80C50AP-6/40AP-6,TMP80C50AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHR	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μS
tPSSR	Power Save Setup Time ($\overline{RESET}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



POWER DOWN MODE (II) ..... ALL Data Hold Mode

The operation of oscillation circuit is suspended by setting $\overline{PS}$ terminal to low level after $\overline{SS}$ terminal has been set to low level. Consequently, all data can be held in low power consumption.

The minimum hold voltage of VCC in this mode is 3V.

$\overline{PS}$ terminal is set to high level to resume oscillation after VCC has been reset to 5V, and then $\overline{SS}$ terminal is set to high level, thus, the normal mode is restarted continuously from the state just before the power down mode (II).

DC CHARACTERISTICS

TMP80C50AP/40AP,TMP80C50AF,TMP80C50AT/40AT: TOPR=0°C to 70°C, VSS=0V
TMP80C50AP-6/40AP-6,TMP80C50AF-6 : TOPR=-40°C to 85°C, VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VSB2	Standby Voltage(2)		3.0	-	6.0	V
ISB2	Standby Current(2)	VCC=5V,VIH=VCC-0.2V, VIL=0.2V	-	0.5	10	μA

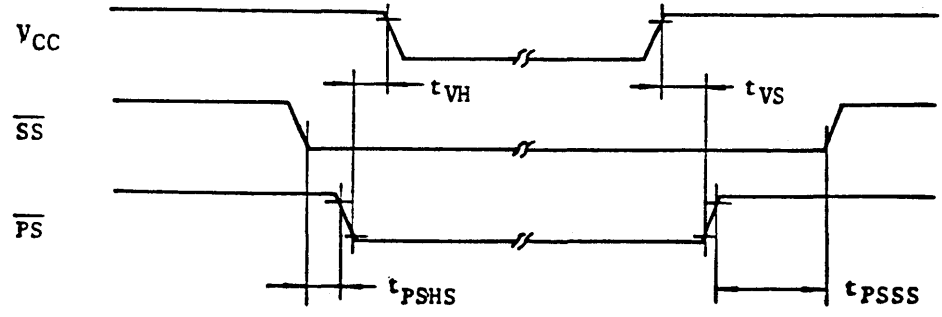
AC CHARACTERISTICS

TMP80C50AP/40AP,TMP80C50AF,TMP80C50AT/40AT:TOPR=0°C to 70°C,VSS=5V±10%,VSS=0V
TMP80C50AP-6/40AP-6,TMP80C50AF-6 : TOPR=-40°C to 85°C,VCC=5V±20%,VSS=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
tPSHS	Power Save Hold Time ($\overline{SS}$)		10	-	-	μS
tPSSS	Power Save Setup Time ($\overline{SS}$)		10	-	-	mS
tVH	VCC Hold Time ($\overline{PS}$)		5	-	-	μS
tVS	VCC Setup Time ($\overline{PS}$)		5	-	-	μS

Note: tCY=2.5μS (fXTAL=6MHz)

TIMING WAVEFORM



HALT MODE

. 1 HALT INSTRUCTION

OP code is "01H". HALT INSTRUCTION is an additional instruction to the standard 8048/8049 instruction set.

. 2 Entry to HALT MODE

On the execution of HALT INSTRUCTION, TMP80C50A/TMP80C40A enter HALT MODE.

. 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logic values just prior to the execution of HALT INSTRUCTION are maintained. Power consumption in HALT MODE is less than 50% of normal operation. The status of each pins are described in the following table.

. 4 Release from HALT MODE

HALT MODE is released by either of two signals ($\overline{\text{RESET}}$,$\overline{\text{INT}}$).

(1) $\overline{\text{RESET}}$ Release Mode : An active $\overline{\text{RESET}}$ input signal causes the normal reset function. TMP80C50A/TMP80C40A start the program at address "000 H".

(2) $\overline{\text{INT}}$ Release Mode : An active $\overline{\text{INT}}$ input signal causes the normal operation.

In case of interrupt enable mode (EI MODE), TMP80C50A/TMP80C40A execute the interrupt service routine, after the execution of one instruction which is located at the next address after HALT INSTRUCTION.

In case of interrupt disable mode (DI MODE), TMP80C50A/TMP80C40A execute normal operation from the next address after HALT INSTRUCTION.

. 5 Supply Voltage Range in HALT MODE

The operating supply voltage range and the operating temperature range are same as in normal operation.

PIN STATUS IN POWER DOWN MODE (I) (II)

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	High impedance Input disabled
TO	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{\text{RESET}}$, $\overline{\text{SS}}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{\text{INT}}$, EA	Input disabled when oscilltor is stopped.
$\overline{\text{RD}}$, $\overline{\text{WR}}$, ALE PROG, $\overline{\text{PSEN}}$	High impedance

PIN STATUS IN HALT MODE

PIN NAME	STATUS
DB0 - DB7 P10 - P17 P20 - P27	Values prior to the execution of HALT INSTRUCTION are maintained.
TO	Status prior to the execution of HALT INSTRUCTION is maintained.
T1	Input disabled
XTAL1, XTAL2	Continue oscillation
$\overline{\text{RESET}}$, $\overline{\text{INT}}$	Input enabled
$\overline{\text{SS}}$, $\overline{\text{EA}}$	Input disabled
$\overline{\text{RD}}$, $\overline{\text{WR}}$, PROG, $\overline{\text{PSEN}}$	Output "High" level
ALE	Output "Low" level

OSCILATOR

QUARTZ CRYSTAL

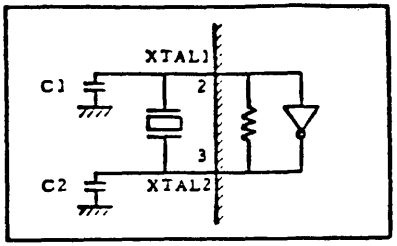
f = 1MHz to 4MHz : C1 = C2 = 30pF

f = 4MHz to 11MHz : C1 = C2 = 20pF

CERAMIC RESONATOR

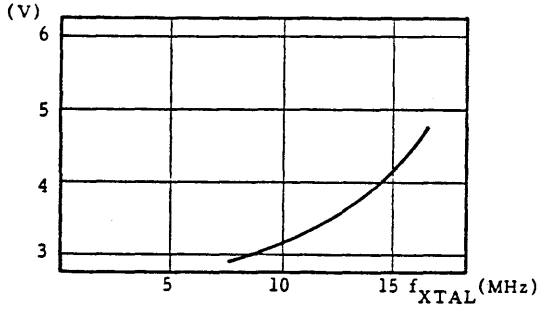
f = 1MHz to 3MHz : C1 = C2 = 100pF

f = 3MHz to 11MHz : C1 = C2 = 30PF

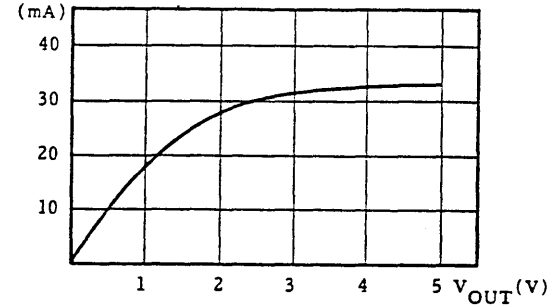


TYPICAL CHARACTARISTICS: Vcc = 5V, Ta = 25°C, unless otherwise noted.

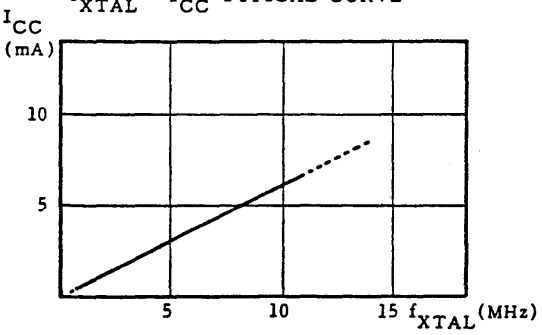
V_{CC} - $f_{MAX.}$ TYPICAL CURVE



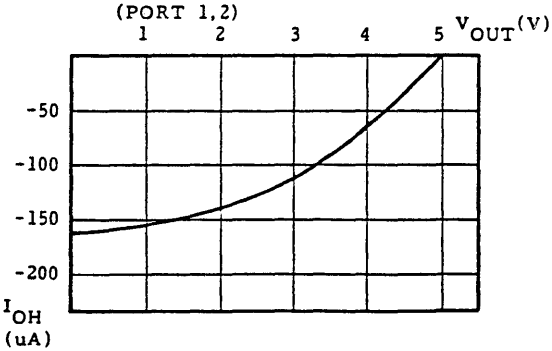
V_{OUT} - I_{OL} TYPICAL CURVE



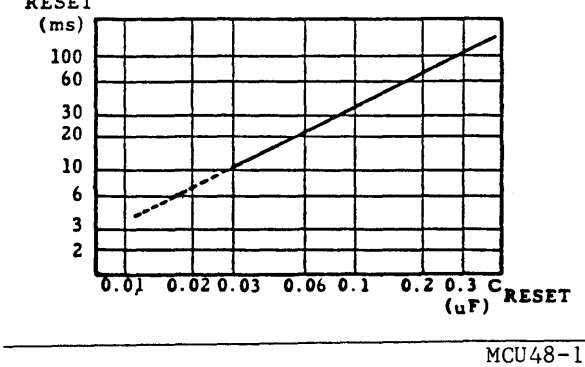
f_{XTAL} - I_{CC} TYPICAL CURVE



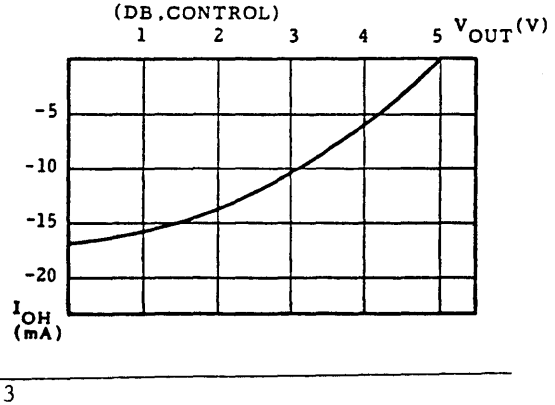
V_{OUT} - I_{OH} TYPICAL CURVE
(PORT 1,2)



t_{RESET} C_{RESET} - t_{RESET} TYPICAL CURVE

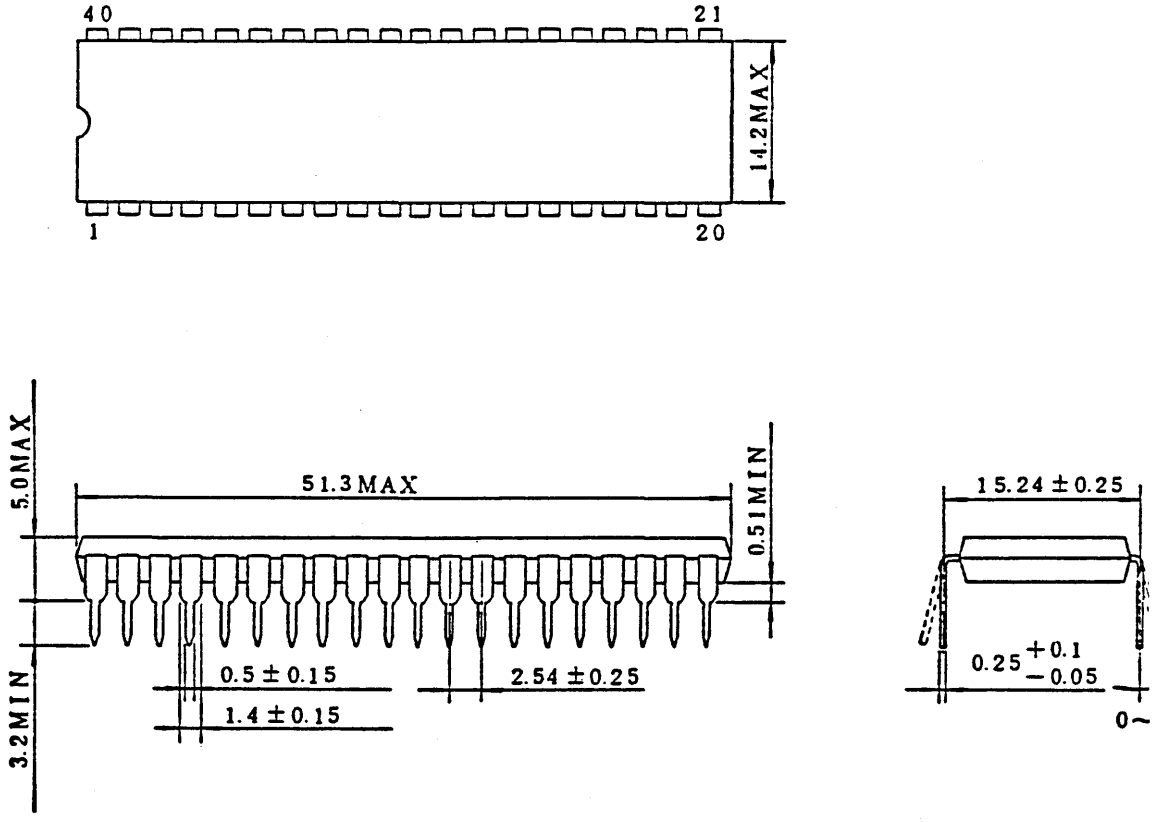


V_{OUT} - I_{OH} TYPICAL CURVE
(DB.CONTROL)



OUTLINE DRAWING (DUAL INLINE PACKAGE)

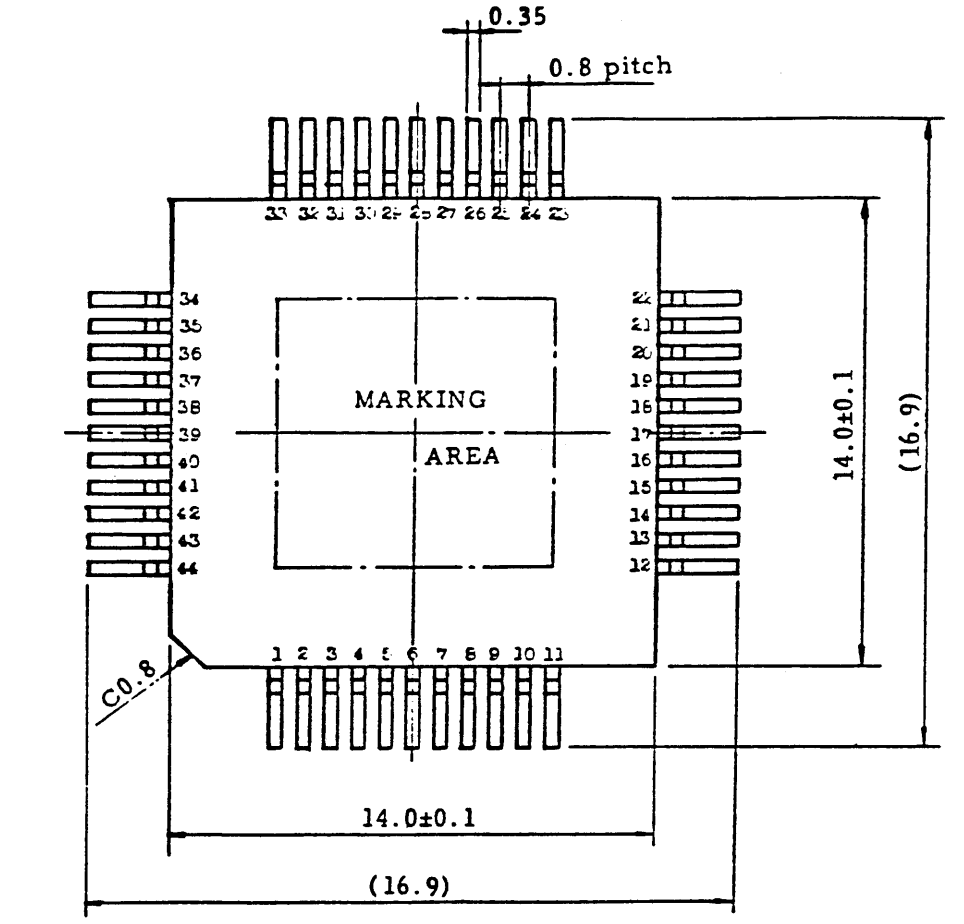
Unit in mm

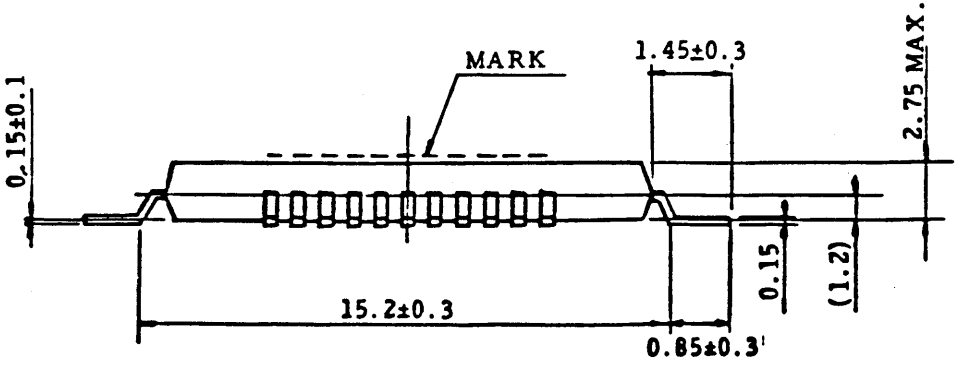


Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

OUTLINE DRAWING (FLAT PACKAGE)

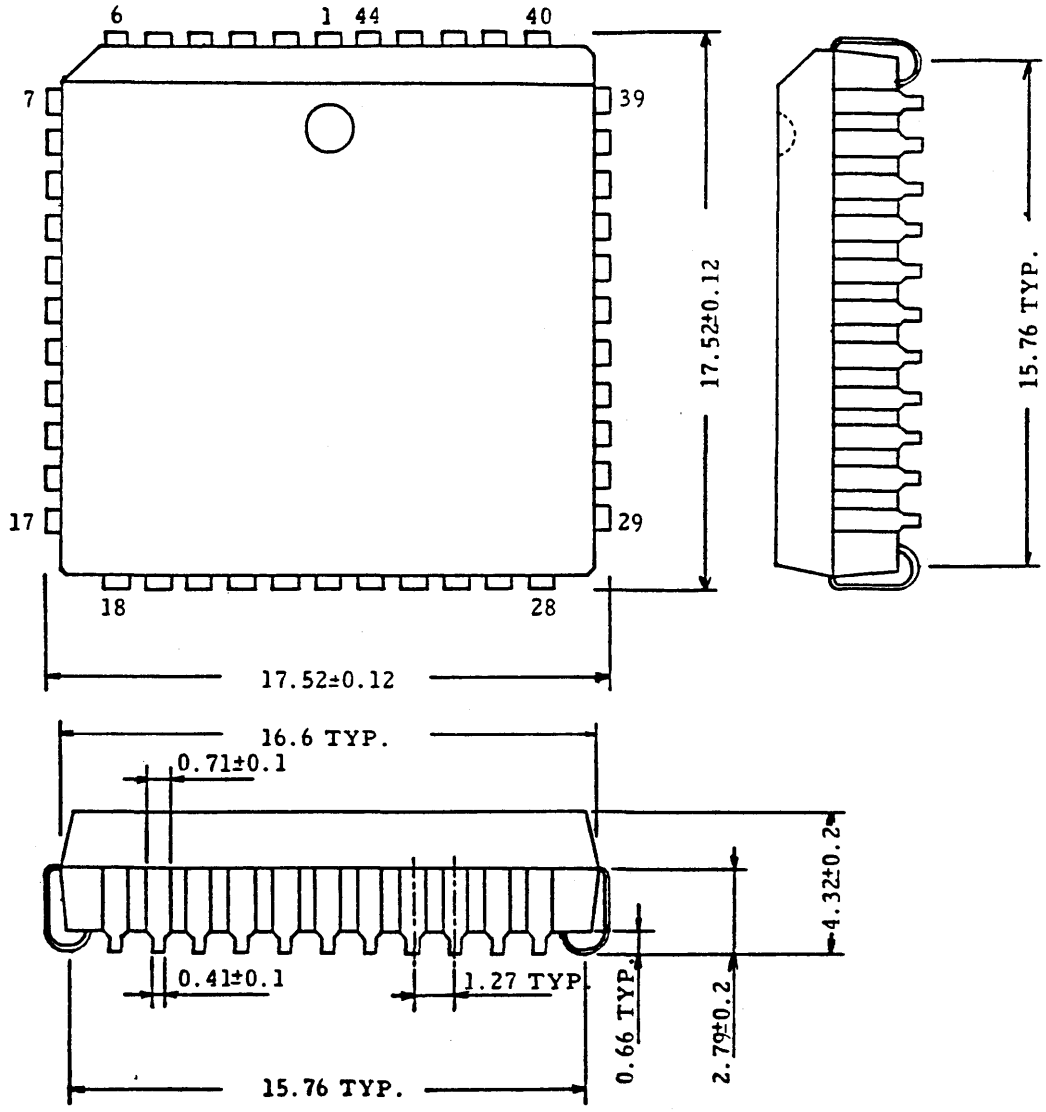
Unit in mm





OUTLINE DRAWING (Plastic Leaded Chip Carrier)

unit in mm



8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP8048PI, from here on referred to as the TMP8048, is a single chip microcomputer fabricated in N-channel Silicon Gate MOS technology which provides internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, 64 x 8 RAM data memory, 1K x 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

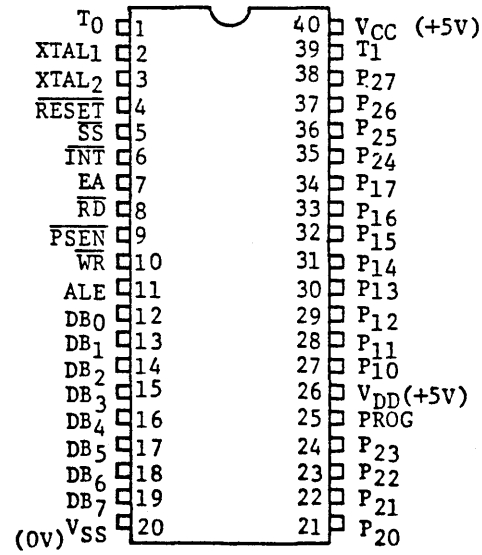
The TMP8048 is particularly efficient as a controller. It has extensive bit handing capability as well as facilities for both binary and BCD arithmetic.

The TMP8035PI is the equivalent of a TMP8048 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

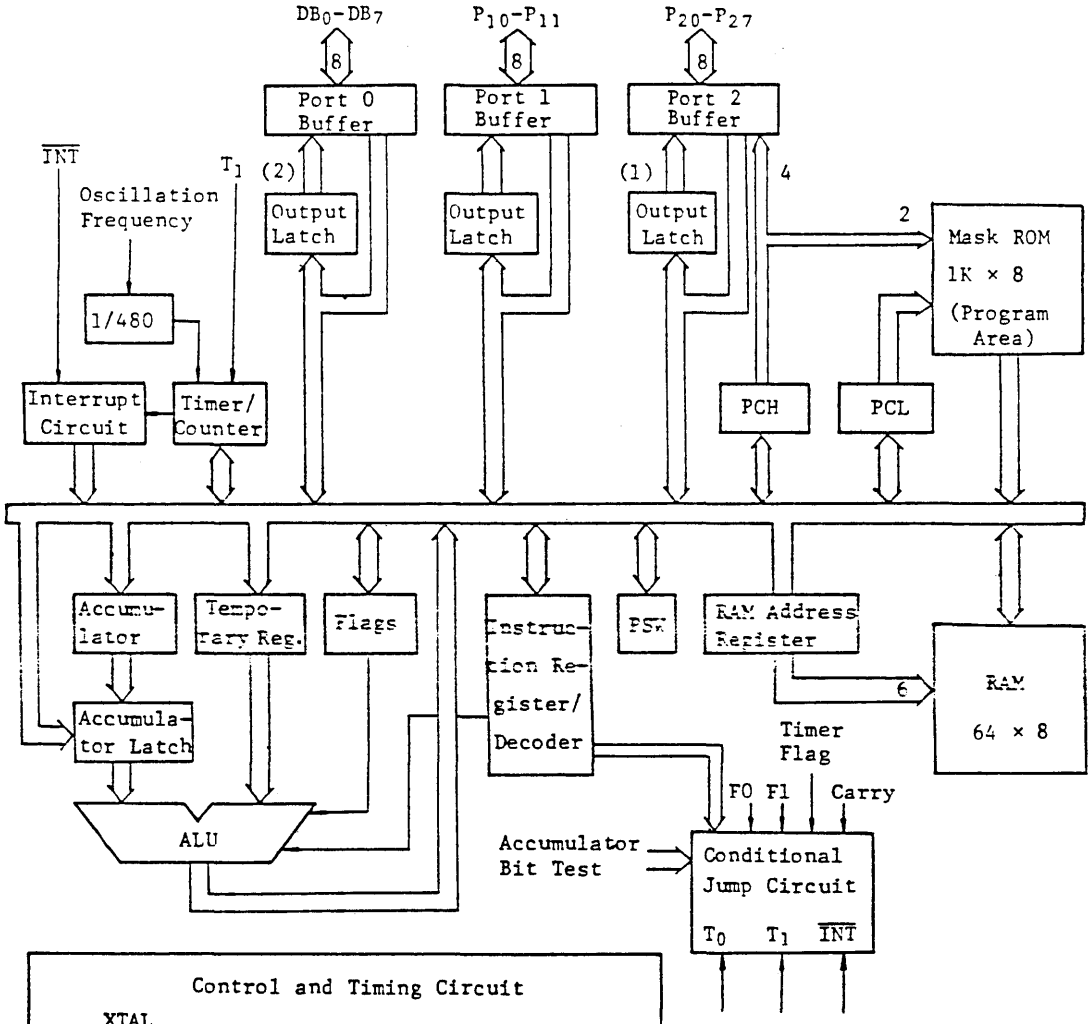
FEATURES

- 2.5 μS Instruction Cycle
- All instruction 1 or 2 cycles
- Over 90 instructions; 70% single byte
- Easy expandable memory and I/O
- 1K x 8 masked ROM
- 64 x 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Single level interrupt
- Single 5V supply
  -40°C to +85°C Operation

PIN CONNECTIONS (Top View)



BLOCK DIAGRAM



Note 1) The lower order 4 bits of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VDD (Power Supply)
+5V during operation Low power standby pin for TMP8048 RAM

VCC (Main Power Supply)
+5V during operation

PROG(Output)
Output strobe for the TMP8243P I/O expander

P10-P17 (Input/Output) Port 1
8-bit quasi -bidirectional port (Internal Pullup=50kΩ).

P20-P27 (Input/Output) Port 2
8-bit quasi-bidirectional port (Internal Pullup=50kΩ).
P20-P23 Contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0-DB7 (Input/Output, 3 State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active Low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low) Used as a Write Strobe to External Data Memory.

RESET (Input)
Active Low signal which is used to initialize the Processor. Also used during Power down.

ALE (Output)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being excuted.

EA (Input)
External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High).

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

FUNCTIONAL DESCRIPTION

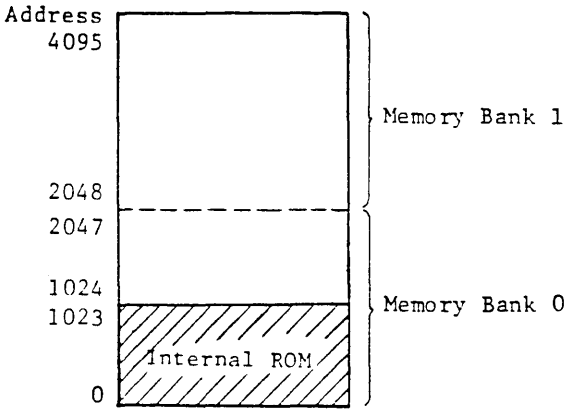
1. System Configuration

The following system functions of the TMP8048 are described in detail.

(1) Program Memory	(6) Stack (Stack Pointer)
(2) Data Memory	(7) Flag 0, Flag 1
(3) I/O Port	(8) Program Status Word (PSW)
(4) Timer/Counter	(9) Reset
(5) Interrupt Control Circuit	(10) Oscillator Circuit

(1) Program Memory
. The maximum memory that can be directly addressed by the TMP8048 is 4096 bytes. The first 1024 bytes from location 0 through 1023 can be internal resident mask ROM. The rest of the 3072 bytes of addressable memory are external to the chip. The TMP8035 has no internal resident memory; all memory must be external.

There are three locations in Program Memory of special importance.



Program Memory Area

. Location 0
Activating the Reset line of the processor causes the first instruction to be fetched from Location 0.

. Location 3
Activating the interrupt line of the processor (if interrupt enabled) causes a jump to subroutine defined by address held in Location 3.

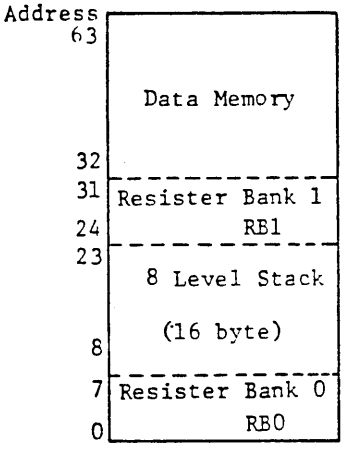
. Location 7
A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

. Program address 0-2047 and 2048-4095 are called memory banks 0 and 1 respectively switching of memory banks is achieved by changing the most significant bit of the program counter (PC) during execution of an unconditional jump instruction or call instruction executed after using SEL MB0 or SEL MB1.

Reset operation automatically selects Bank 0.

(2) Data Memory

. Resident Data Memory (volatile RAM) is organized as 64 words by 8-bits wide.
. The first 8 locations (0 -7) of the memory array are designated as working registers and are directly addressable by several instructions. By executing a Register Bank switch instruction (SEL RB1) locations 24 - 31 are designated as the working registers in place of 0 - 7.



Internal Data Memory Area

- RAM locations 8 - 23 serve a dual role in that they contain the program counter stack which is a stack 2 bytes wide by 8 levels deep. These locations store returning addresses from subroutines. If the level of subroutine nesting is less than the permitted 8, you free up 2 bytes of RAM for general use for every level of nesting not utilized.

- ALL 64 locations are indirectly addressable through either of two RAM Pointer Registers which reside at R0 and R1 of the Register array.

- The TMP8048 architecture allows extension of the Data Memory to 256 words.

(3) Input/Output Ports

- The TMP8048 has 27 I/O lines which can be used for either input or output. These I/O lines are grouped into 3 ports each having 8 bidirectional lines and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

- Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction.

- All lines of Ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure (illustrated in Figure 1). Each line is continously pulled to a +5V level through a high impedance resistive device (50kΩ) which is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. In order to speed up the "0" to "1" transition a low impedance device (5kΩ) is switched in momentarily whenever a "1" is written to line. When a "0" is written to line a low impedance device overcomes the pullup and provides TTL current sinking capability.

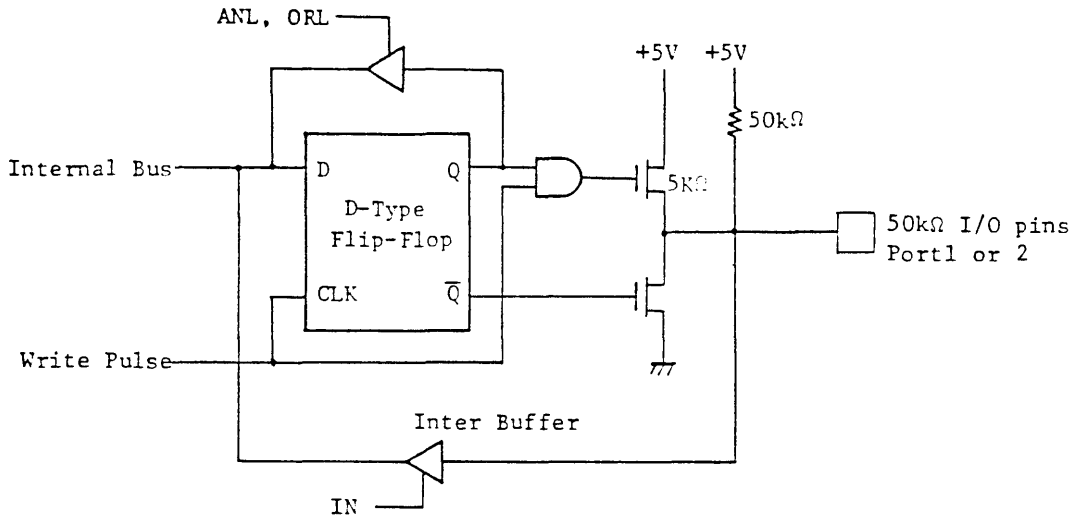


Fig.1 Input/Output Circuit of Port 1, Port 2

. Reset initializes all lines to a high impedance "1" state.

. When external data memory area is not addressed during excution of an internal program, Port 0 (DB0 - DB7) becomes a true bidirectional port (bus) with associated input and output strobes. If bidirectional feature not needed Bus can serve as either a statically latched output port or a non-latched input port. However, I/O lines of this port cannot be intermixed.

. As a static port data is written and latched using the OUTL instruction and inputted using the INS instruction these two commands generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ strobe lines.

. As a bidirectional port the MOVX instructions are used to read and write the port which generate the $\overline{RD}$ and $\overline{WR}$ strobes.

. When not being written or read, the Bus lines are in a high impedance state.

(4)Timer/Event Counter

. The 8-bit binary up counter can use either of the following frequency inputs

(1) Internal clock (1/480 of OSC frequency)
..... Timer mode

(2) External input clock form T1 terminal
(minimum cycle time 3 x ALE cycle)
..... Event Counter mode

The counter is presettable and readable with two MOV instructions which transfer the content of the accumulator to the counter and vice versa. The counter content is not affected by a Reset and is initialized solely by the MOVT, A instruction. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started by START T instruction or as an event counter by a START CNT. Once started the counter will increment to its maximum count (FF) and overflow to Zero continuing its count until stopped by a STOP TCNT instruction or RESET.

The increment from maximum count to Zero (overflow) results in the setting of an overflow flag and the generation of an interrupt request. When interrupt acknowledged a subroutine call to Location 7 will be initiated. Location 7 should store the starting address of the timer or counter service routine. The state of the overflow flag is testable with the conditional JUMP (JTF). The flag is reset by excuting a JTF or by RESET. Figure 2 illustrates the concept of the timer circuit.

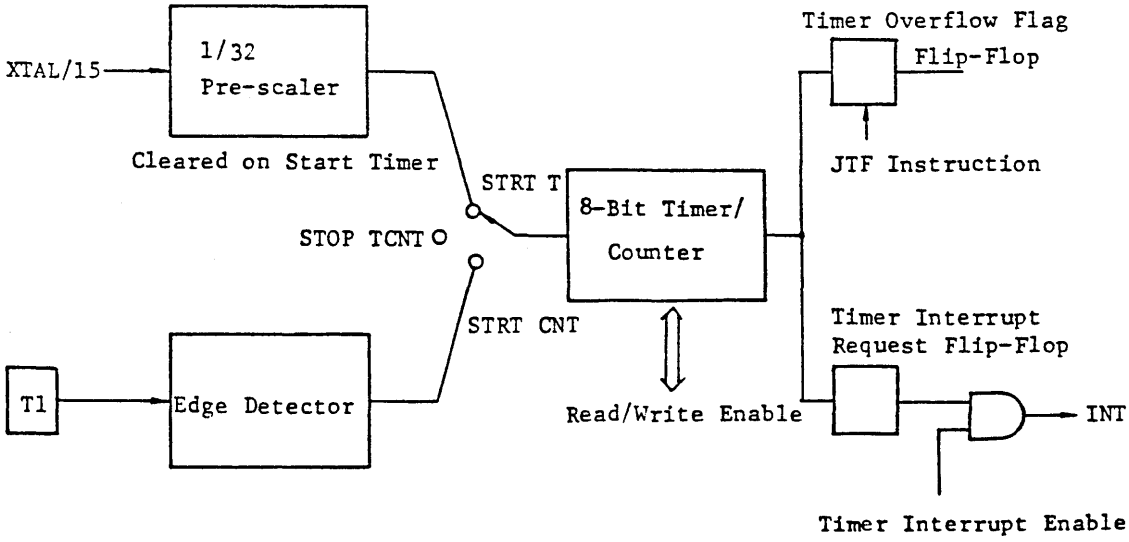


Fig.2 Concept of Timer Circuit

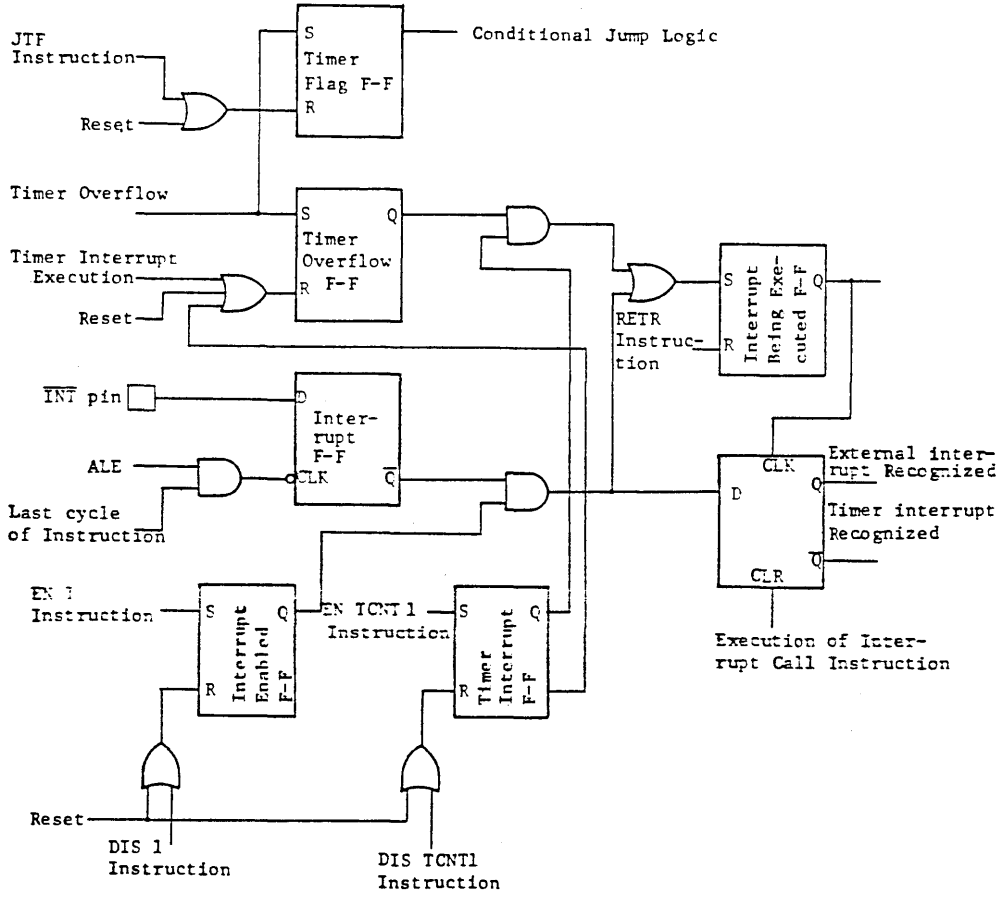


Fig.3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

. There are two distinct types of Interrupts in the TMP8048.

(1) External Interrupt from the $\overline{\text{INT}}$ terminal

(2) Timer Interrupt caused by timer overflow

The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR (which should occur at the end of an interrupt service routine) reenables the interrupt input logic.

. An interrupt sequence is initiated by applying a low level "0" to the $\overline{INT}$ pin. $\overline{INT}$ is level triggered and active low which allows "Wire Oring" of several interrupt sources. The interrupt level is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at Location 3. As in any call to subroutine, the Program Counter and Program Status Word are saved in the stack.

. When an overflow occurs in the internal timer/event counter an interrupt request is generated which is reserviced as outlined in previous paragraph except that a jump to Location 7 is used instead of 3. If $\overline{INT}$ and times overflow occur simultaneously then external request $\overline{INT}$ takes precedence.

. If an extra external interrupt is needed in addition to $\overline{INT}$ this can be achieved by enabling the counter interrupt, loading FFH in the counter (one less than the terminal count), and enabling the event counter mode. A "1" to "0" transition on T1 will cause an interrupt vector to Location 7.

. The interrupt service routine pointed to be addresses in Location 3 or 7 must reside in memory between 0 and 2047, i.e., Bank 0.

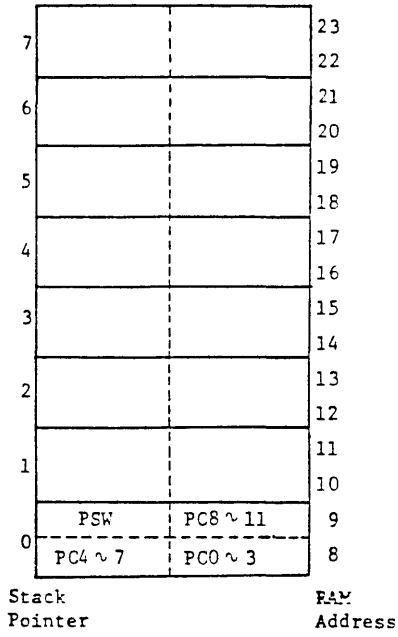
Figure 3 illustrates the concept of the interrupt control circuit.

(6) Stack (stack Pointer)

. An interrupt or Call to subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit stack pointer which is part of the Program Status Words (PSW explained in section (8)). Data RAM locations, 8 through 23 are available as stack registers and are used to store the program counter and 4-bits of PSW as shown in the figure.

. The stack pointer when initialized points to RAM location 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to Locations 8 and 9. Then the stack pointer is incremented by one to point to Locations 10 and 11. Eight levels of subroutine are obviously possible.

. At the end of a subroutine signalled by a RET or RETR causes the stack pointer to be decremented by one and the contents of the resulting pair to be transferred to the Program Counter.



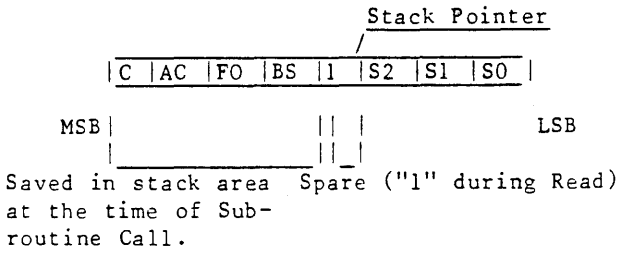
(7) Flag 0, Flag 1, (F0, F1)

. The TMP8048 has two flags F0 and F1 which are used for conditional jump. These flags can be set, reset and tested with the conditional jump instruction JF0.

. F0 is a part of the program status word (PSW) and is saved in the stack area when a subroutine is called.

(8) Program Status Word (PSW)

. An 8-bit status word which can be loaded to and from the accumlator exists called the Program Status Word (PSW). The PSW is read by a MOV A, PSW and written to by a MOV PSW, A. The information available in the PSW is shown in the diagram below.



Bits 0 - 2 : Stack Pointer Bits(S0, S1, S2)
Bit 3 : Not used ("1" level when read.)
Bit 4 : Working Register Bank Switch Bit (BS)

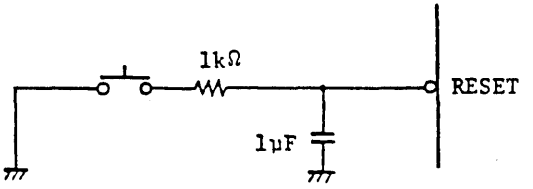
0 = Bank 0
1 = Bank 1

Bit 5 : Flag 0 (F0)
Bit 6 : Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA, A (AC)

Bit 7 : Carry (C) flag which indicates that the previous operation has resulted in the accumulator. (C)

(9) Reset

. The reset input provides a means for initialization of the processor. This Schmitt trigger input has an internal pullup registor which in combination with an external 1μF capacitor provides an internal reset pulse sufficient length to guarantee that all internal logic is initialized.



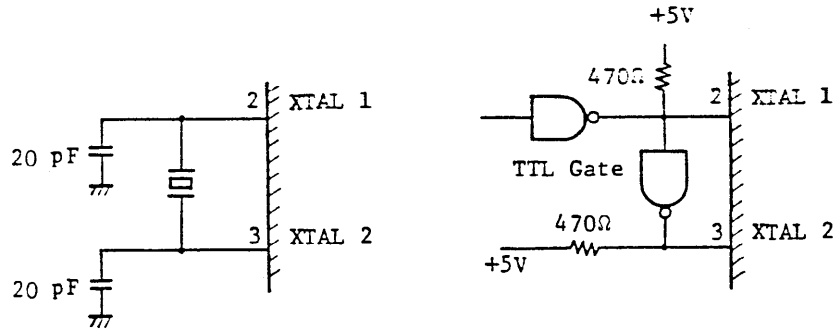
If the pulse is generated externally the reset pin must be held at ground ($\leq$0.5V) for at least 50mS after the power supply is within tolerance.

. Reset performs the following functions within the chip:

(i) Sets PC to Zero.
(ii) Sets Stack Pointer to Zero.
(iii) Selects Register Bank 0.
(iv) Selects Memory Bank 0.
(v) Sets BUS (DB0 - DB 7) to high impedance state. (Except when EA = 5V)
(vi) Sets Ports 1 and 2 to input mode.
(vii) Disables interrupts (timer and external).
(viii) Stops Timer.
(ix) Clears Timer Flag.
(x) Clears F0 and F1.
(xi) Disables clock output from T0.

(10) Oscillator Circuit

. TMP8048 can be operated by the external clock input in addition to crystal oscillator as shown below.



2. Basic Operation and Timing

The following basic operations and timing are explained
(1) Instruction Cycle
(2) External Memory Access Timing
(3) Interface with I/O Expander TMP8243P
(4) Internal Program Verify (Read) Timing
(5) Single Step Operation Timing
(6) Low Power Stand-by Mode

(1) Instruction Cycle

. The instructions of TMP8048 are executed in one or two machine cycles, and one machine cycle contents of five states.

. Fig.4 illustrates its relationship with the clock input to CPU.

. $\phi 2$ clock shown in Fig.4 is derived to outside by ENTO CLK instruction.

. ALE can be also used as the clock to indicate the machine cycle as well as giving the external address latch timing.

(2) External Memory Access Timing

(i) Program Memory Access

. TMP8048 programs are excuted in the following three modes.

(1) Execution of internal program only.
(2) Execution of both external and internal programs.
(3) Execution of external program only.

The external program memory is accessed (instructions are fetched) automatically when the internal ROM address is exceeded in mode (2) and from initial start address 0 in mode (3).

. In the external program memory access operation, the following will occur

. The contents of the 12-bit program counter will be output on BUS(DB0 - DB7) and the lower 4-bits of Port 2.
. Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.
. Program Store Enable (PSEN) indicates that an external instruction fetch is in progress and serves to enable the external memory device.
. BUS (DB0 - DB7) reverts to Input mode and the processor accepts its 8-bit contents as an Instruction Word.

. Figure 5 illustrates the timing.

(ii) Access of External Data Memory

. In the extended data memory access operation during READ/WRITE cycle the following occurs
. The contents of R0 R1 is output onto BUS (DB0 - DB7).
. ALE indciates address is valid. The trailing edge of ALE is used to latch the address externally.
. A read $\overline{RD}$ or write $\overline{WR}$ pulse on the corresponding output pins indicates the type of data memory access in progress. Output data valid at trailing edge of $\overline{WR}$ and input data must be valid at trailing edge of $\overline{RD}$.
. Data (8-bits) is transferred over BUS.

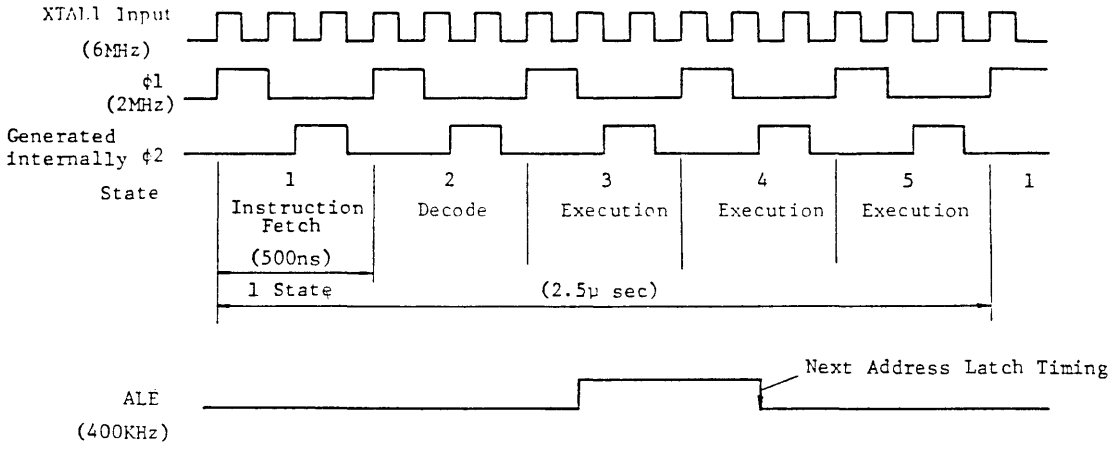


Fig.4 Instruction Cycle Timing

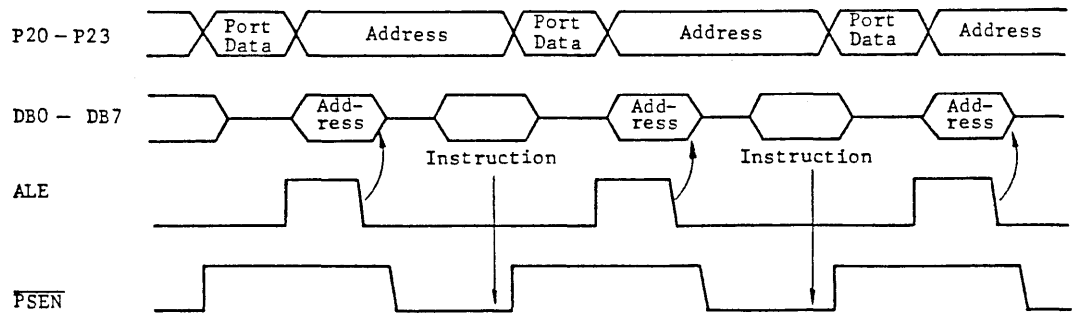
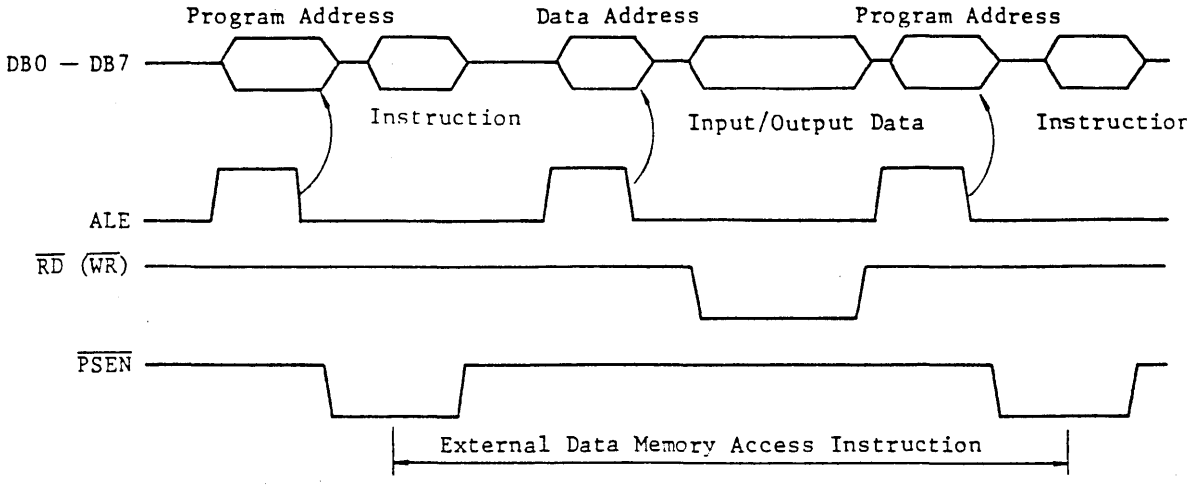


Fig. 5 Timing of External Program Memory Access



Suggest we have diagrams

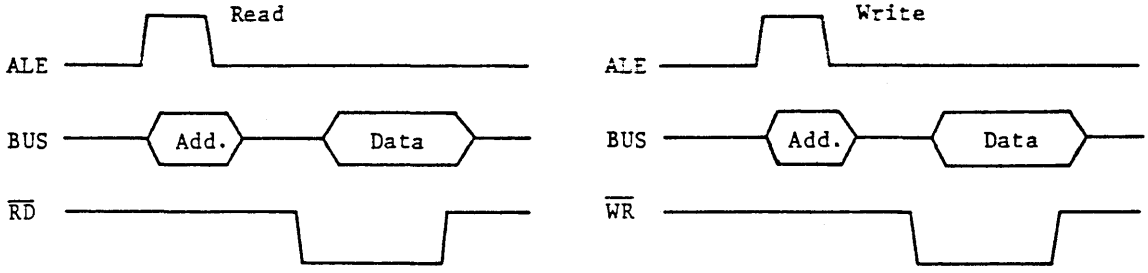


Fig.6 Timing of Accessing External Data Memory

. Figure 6 illustrates the timing of accessing the external data memory during execution of external program.

(3) Interface with I/O Expander (TMP8243P)

. The TMP8048 I/O can be easily expanded using the TMP8243 I/O Expander. THis device uses only the lower half 4-bits of Port 2 for commuication with the TMP8048. The TMP8243 contains four 4-bit I/O ports which serve as extensions of one chip I/O and are addressed as Ports (4-7). All communication takes place over the lower half of port 2 (P20 - P23) with timing provided by an output pulse on the PROG pin. Each transfer consists of two 4-bit nibbles the first containing the "OP Code" and port address and the second containing the actual 4-bits of data.

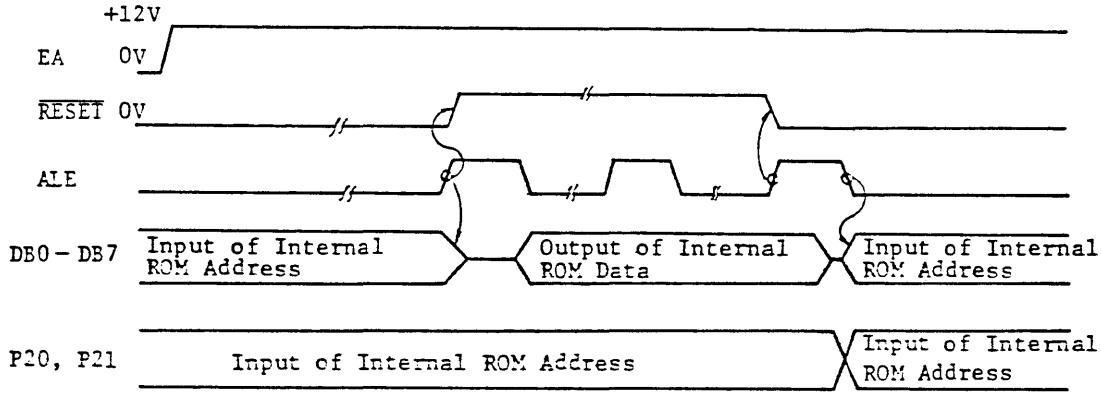


Fig.7 Timing of Reading Internal Program Memory

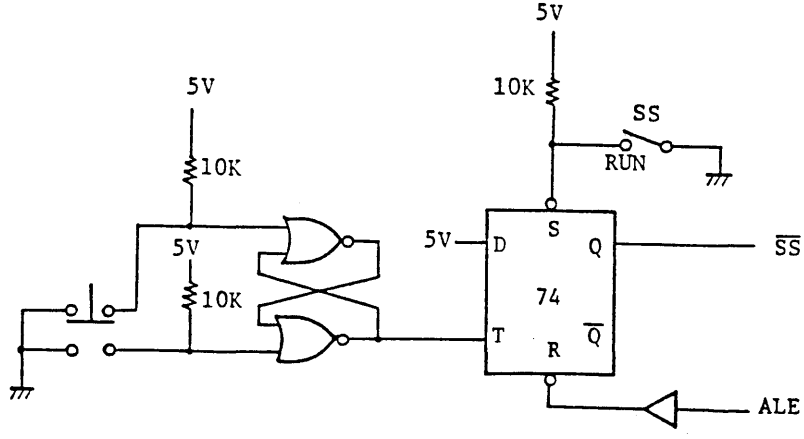


Fig.8 (a) Single Step Circuit

Reading of Internal Program Memory

. The processor is placed in the READ mode by applying +12V to the EA pin and 0V to the $\overline{\text{RESET}}$ pin. The address of the location to be read is then applied to BUS and the low order 2-bits of Port 2. The address is latched by a 0 to 1 transition on $\overline{\text{RESET}}$ and the high level causes the contents of program memory location addressed to appear on the eight lines of BUS.

. Figure 7 illustrates the timing diagram for this operation.

(5) Single Step Operation.

. A single step feature useful for debug can be implemented by utilizing a circuit shown in Figure 8 (a) combined with the $\overline{\text{SS}}$ pin and ALE pin.

. A D-type flip flop with set and reset is used to generate $\overline{\text{SS}}$. In the run mode $\overline{\text{SS}}$ is held high by keeping the flip flop set. To enter single step, set is removed allowing ALE to bring $\overline{\text{SS}}$ low via reset input. The next instruction is started by clocking a "1" into the FF which will not appear on $\overline{\text{SS}}$ unless ALE is high removing reset. In response to $\overline{\text{SS}}$ going high the processor begins an instruction fetch which brings ALE low resetting FF and causing the processor to again enter the stopped state.

. The timing diagram in this case is as shown in Figure 8 (b). (EA = 5V).

(6) Lower Power Stand-by Mode.

. The Lower TMP8048 has been organized to allow power to be removed from all but the volatile, 64 x 8 data RAM array. In power down mode the contents of data RAM can be maintained while drawing typically 10 - 15% of normal operating power requirements.

. VCC serves as the 5V supply for the bulk of the TMP8048 while the VDD supplies only the RAM array. In standby mode VCC is reduced to 0V but VDD is kept at 5V. Applying a low level to reset inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from VCC.

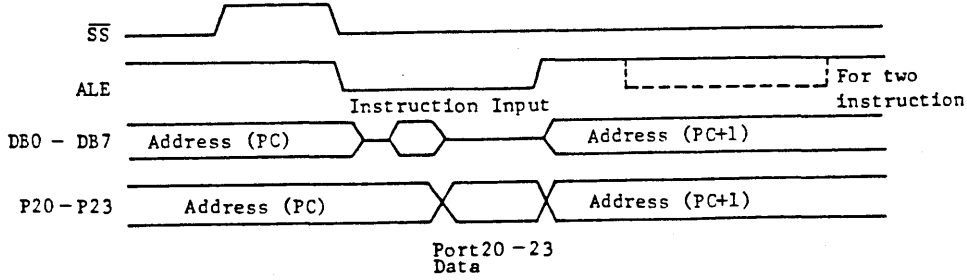


Fig.8(b) Single Step Operation Timing

INSTRUCTION

ACCUMULATOR INSTRUCTION

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
ADD A,Rr	0	1	1	0	1	r	r	r	(A)<-(A)+(Rr) r = 0 - 7	1	1	o	o
ADD A,@Rr	0	1	1	0	0	0	0	r	(A)<-(A)+((Rr)) r = 0, 1	1	1	o	o
ADD A,#Data	0	0	0	0	0	0	1	1	(A)<-(A)+Data	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ADDC A,Rr	0	1	1	1	1	r	r	r	(A)<-(A)+(Rr)+(C) r = 0 - 7	1	1	o	o
ADDC A,@Rr	0	1	1	1	0	0	0	r	(A)<-(A)+((Rr))+(C) r = 0, 1	1	1	o	o
ADDC A,#Data	0	0	0	1	0	0	1	1	(A)<-(A)+Data+(C)	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ANL A,Rr	0	1	0	1	1	r	r	r	(A)<-(A) and (Rr) r = 0 - 7	1	1	-	-
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)<-(A)and ((Rr)) r = 0, 1	1	1	-	-
ANL A,#Data	0	1	0	1	0	0	1	1	(A)<-(A) and Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
ORL A,Rr	0	1	0	0	1	r	r	r	(A)<-(A) or (Rr) r = 0 - 7	1	1	-	-
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)<-(A) or ((Rr)) r = 0, 1	1	1	-	-
ORL A,#Data	0	1	0	0	0	0	1	1	(A)<-(A) or Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
XRL A, Rr	1	1	0	1	1	r	r	r	(A)<-(A) EOR (Rr) r = 0 - 7	1	1	-	-
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)<-(A) EOR((Rr)) r = 0, 1	1	1	-	-
XRL A,#Data	1	1	0	1	0	0	1	1	(A)<-(A) EOR Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
INC A	0	0	0	1	0	1	1	1	(A)<-(A)+1	1	1	-	-
DEC A	0	0	0	0	0	1	1	1	(A)<-(A)-1	1	1	-	-
CLR A	0	0	1	0	0	1	1	1	(A)<-0	1	1	-	-
CPL A	0	0	1	1	0	1	1	1	(A)<-NOT (A)	1	1	-	-
DA A	0	1	0	1	0	1	1	1	Decimal Adjust Accumulator	1	1	o	-
SWAP A	0	1	0	0	0	1	1	1	(A4-7)->(A0-3) <-	1	1	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
RL A	1	1	1	0	0	1	1	1	(An+1)<-(An) n = 0 - 6 (A0)<-(A7)	1	1	-	-
RLC A	1	1	1	1	0	1	1	1	(An+1)<-(An) n = 0 - 6 (C)<-(A7) (A0)<-(C)	1	1	-	-
RR A	0	1	1	1	0	1	1	1	(An)<-(An+1) n = 0 - 6 (A7)<-(A0)	1	1	-	-
RRC A	0	1	1	0	0	1	1	1	(An)<-(An+1) n = 0 - 6 (C)<-(A0) (A7)<-(C)	1	1	-	-

Input/Output Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
IN A,Pp	0	0	0	0	1	0	P	P	(A)<-(Pp) P = 1, 2	1	2	-	-
OUTL Pp,A	0	0	1	1	1	0	P	P	(Pp)<-(A) P = 1, 2	1	2	-	-
ANL Pp,#Data	1 d7	0 d6	0 d5	1 d4	1 d3	0 d2	P d1	P d0	(Pp)<-(Pp)and Data P = 1, 2	2	2	-	-
ORL Pp,#Data	1 d7	0 d6	0 d5	0 d4	1 d3	0 d2	P d1	P d0	(Pp)<-(Pp)or Data P = 1, 2	2	2	-	-
INS A,BUS	0	0	0	0	1	0	0	0	(A)<-(BUS)	1	2	-	-
OUTL BUS,A	0	0	0	0	0	0	1	0	(BUS)<-(A)	1	2	-	-
ANL BUS,#Data	1 d7	0 d6	0 d5	1 d4	1 d3	0 d2	0 d1	0 d0	(BUS)<-(BUS) and Data	2	2	-	-
ORL BUS,#Data	1 d7	0 d6	0 d5	0 d4	1 d3	0 d2	0 d1	0 d0	(BUS)<-(BUS) or Data	2	2	-	-
MOVD A,Pp	0	0	0	0	1	1	P	P	(A0-3)<-(Pp) (A4-7)<-0 P = 4 - 7	1	2	-	-
MOVD Pp,A	0	0	1	1	1	1	P	P	(Pp)<-(A0-3) P = 4 - 7	1	2	-	-
ANLD Pp,A	1	0	0	1	1	1	P	P	(Pp)<-(Pp)and (A0-3) P = 4 - 7	1	2	-	-
ORLD Pp,A	1	0	0	0	1	1	P	P	(Pp)<-(Pp)or(A0-3) P = 4 - 7	1	2	-	-

Register Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
INC Rr	0	0	0	1	1	r	r	r	(Rr)<--(Rr)+1 r = 0 - 7	1	1	-	-
INC @Rr	0	0	0	1	0	0	0	r	((Rr))<-((Rr))+1 r = 0, 1	1	1	-	-
DEC Rr	1	1	0	0	1	r	r	r	(Rr)<--(Rr)-1 r = 0 - 7	1	1	-	-

Branch Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JMP Address	a10	a9	a8	0	0	1	0	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC8-10)<--(a8-10) (PC11)<--DBF				
JMPP @A	1	0	1	1	0	0	1	1	(PC0-7)<--((A))	1	2	-	-
DJNZ Rr,	1	1	1	0	1	r	r	r	(Rr)<--(Rr)-1	2	2	-	-
Address	a7	a6	a5	a4	a3	a2	a1	a0	if Rr not 0 (PC0-7)<--(a0-7)				
JC Address	1	1	1	1	0	1	1	0	(PC0-7)<--(a0-7) if C = 1	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC) = (PC)+2 if C = 0				
JNC Address	1	1	1	0	0	1	1	0	(PC0-7)<-(a0-7) if C = 0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC)<--(PC)+2 if C = 1				
JZ Address	1	1	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if (A) = 0 (PC)<--(PC)+2 if (A) .NEQ.0				
JNZ Address	1	0	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if (A) .NEQ.0 (PC)<--(PC)+2 if (A) = 0				
JTO Address	0	0	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if TO = 1 (PC)<--(PC)+2 if TO = 0				
JNTO Address	0	0	1	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if TO = 0 (PC)<--(PC)+2 if TO = 1				
JT1 Address	0	1	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T1 = 1 (PC)<--(PC)+2 if T1 = 0				
JNT1 Address	0	1	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if T1 = 0 (PC)<--(PC)+2 if T1 = 1				

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JF0 Address	1	0	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F0 = 1				
									(PC)<--(PC)+2				
									if F0 = 0				
JF1 Address	0	1	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F1 = 1				
									(PC)<--(PC)+2				
									if F1 = 0				
JTF Address	0	0	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if TF = 1				
									(PC)<--(PC)+2				
									if TF = 0				
JNI Address	1	0	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if INT = 0				
									(PC)<--(PC)+2				
									if INT = 1				
JBb Address	b2	b1	b0	1	0	0	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if Bb = 1				
									(PC)<-(PC)+2				
									if Bb = 0				
									(b = 0 - 7)				
CALL Address	a10	a9	a8	1	0	1	0	0	((SP))<--	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC),(PSW4-7)				
									(SP)<--(SP)+1				
									(PC8-10)<--(a8-10)				
									(PC0-7)<--(a0-7)				
									(PC11)<--DBF				
RET	1	0	0	0	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
RETR	1	0	0	1	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
									(PSW4-7)<--((SP))				

Flag Manipulation Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
CLR C	1	0	0	1	0	1	1	1	(C)<-- 0	1	1	0	-
CPL C	1	0	1	0	0	1	1	1	(C)<--NOT(C)	1	1	0	-
CLR F0	1	0	0	0	0	1	0	1	(F0)<-- 0	1	1	-	-
CPL F0	1	0	0	1	0	1	0	1	(F0)<--NOT(F0)	1	1	-	-
CLR F1	1	0	1	0	0	1	0	1	(F1)<-- 0	1	1	-	-
CPL F1	1	0	1	1	0	1	0	1	(F1)<--NOT(F1)	1	1	-	-

Data Transfer Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A, Rr	1	1	1	1	1	r	r	r	(A)<-- (Rr) r = 0 - 7	1	1	-	-
MOV A, @Rr	1	1	1	1	0	0	0	r	(A)<-- ((Rr)) r = 0, 1	1	1	-	-
MOV A, #Data	0 d7	0 d6	1 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)<--Data	2	2	-	-
MOV Rr, A	1	0	1	0	1	r	r	r	(Rr)<--(A) r = 0 - 7	1	1	-	-
MOV@Rr,A	1	0	1	0	0	0	0	r	((Rr))<-(A) r = 0, 1	1	1	-	-
MOV Rr,#Data	1 d7	0 d6	1 d5	1 d4	1 d3	r d2	r d1	r d0	(Rr)<--Data r = 0 - 7	2	2	-	-
MOV@Rr,#Data	1 a7	0 a6	1 a5	1 a4	0 a3	0 a2	0 a1	r a0	((Rr))<--Data r = 0, 1	2	2	-	-
MOV A,PSW	1	1	0	0	0	1	1	1	(A)<-- (PSW)	1	1	-	-
MOV PSW, A	1	1	0	1	0	1	1	1	(PSW)<--(A)	1	1	-	-
XCH A, Rr	0	0	1	0	1	r	r	r	(A)-->(Rr) r = 0 - 7	1	1	-	-
XCH A,@Rr	0	0	1	0	0	0	0	r	(A)-->((Rr)) r = 0, 1	1	1	-	-
XCHD A,@Rr	0	0	1	1	0	0	0	r	(A0-3)—>((Rr0-3)) r = 0, 1	1	1	-	-
MOVX A,@Rr	1	0	0	0	0	0	0	r	(A)<--((Rr)) r = 0, 1	1	2	-	-
MOVX @Rr,A	1	0	0	1	0	0	0	r	((Rr))<--(A) r = 0, 1	1	2	-	-
MOVP A, @A	1	0	1	0	0	0	1	1	(PC0-7)<--(A) (A)<--((PC))	1	2	-	-
MOVP3 A,@A	1	1	1	0	0	0	1	1	(PC0-7)<--(A) (PC8-11)<--0011 (A)<--((PC))	1	2	-	-

Timer/Counter Instruction

Mnemonic	Instruction Code									Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A,T	0	1	0	0	0	0	1	0	(A)<--(T)	1	1	-	-
MOV T,A	0	1	1	0	0	0	1	0	(T)<--(A)	1	1	-	-
STRT T	0	1	0	1	0	1	0	1	Counting is started in the timer mode	1	1	-	-
STRT CNT	0	1	0	0	0	1	0	1	Counting is started in the event counter mode	1	1	-	-
STOP TCNT	0	1	1	0	0	1	0	1	Stop both time accumulation and event counting	1	1	-	-
EN TCNTI	0	0	1	0	0	1	0	1	Timer interrupt is enabled	1	1	-	-
DIS TCNTI	0	0	1	1	0	1	0	1	Timer interrupt is disabled	1	1	-	-

Control Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
EN I	0	0	0	0	0	1	0	1	External interrupt is enabled	1	1	-	-
DIS I	0	0	0	1	0	1	0	1	External interrupt is disabled	1	1	-	-
SEL RB0	1	1	0	0	0	1	0	1	(BS)<-- 0	1	1	-	-
SEL RB1	1	1	0	1	0	1	0	1	(BS)<-- 1	1	1	-	-
SEL MB0	1	1	1	0	0	1	0	1	(DBF)<-- 0	1	1	-	-
SEL MB1	1	1	1	1	0	1	0	1	(DBF)<-- 1	1	1	-	-
ENTO CLK	0	1	1	1	0	1	0	1	T0 is enabled to act as the clock output	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-

TMP8048PI/8035PI: INDUSTRIAL SPECIFICATION

ABSOLUTE MAXIMUM RATTINGS

SYMBOL	ITEM	RATTING
VDD	VDD Supply Voltage (with respect to GND (VSS))	-0.5V to + 7V
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to + 7V
VINA	Input Voltage (Except EA)	-0.5V to + 7V
VINB	Input Voltage (Only EA)	-0.5V to + 13V
PD	Power Dissipation (Ta = 70°C)	1.5W
TSOLDER	Soldering Temperature (Soldering Time 10 sec)	260°C
TSTG	Storage Temperature	-55°C to 150°C
TOPR	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS

TA=-40°C to 85°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage (Except XTAL1,XTAL2,$\overline{\text{RESET}}$)		-0.5	-	0.7	V
VIL1	Input Low Voltage (XTAL1,XTAL2,$\overline{\text{RESET}}$)		-0.5	-	0.6	V
VIH	Input High Voltage (Except XTAL1,XTAL2,$\overline{\text{RESET}}$)		2.2	-	VCC	V
VIH1	Input High Voltage (XTAL1,XTAL2,$\overline{\text{RESET}}$)		3.8	-	VCC	V
VOL	Output Low Voltage (BUS)	IOL = 1.6 mA	-	-	0.45	V
VOL1	Output Low Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	IOL = 1.6 mA	-	-	0.45	V
VOL2	Output Low Voltage (PROG)	IOL = 0.8 mA	-	-	0.45	V
VOL3	Output Low Voltage (For other output pins)	IOL = 1.2 mA	-	-	0.45	V
VOH	Output High Voltage (BUS)	IOH =-280μA	2.4	-	-	V
VOH1	Output High Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE)	IOH =-80μA	2.4	-	-	V
VOH2	Output High Voltage (For other output pins)	IOH =-30μA	2.4	-	-	V
ILI	Input Leak Current (T1, $\overline{\text{INT}}$)	VSS $\leqq$ VIN $\leqq$ VCC	-	-	±10	μA
ILI1	Input Leak Current (P10-17, P20-P27, EA, $\overline{\text{SS}}$)	VSS+0.45$\leqq$VIN$\leqq$VCC	-	-	-700	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)	VSS+0.45$\leqq$VIN$\leqq$VCC	-	-	±10	μA
IDD	VDD Supply Current		-	-	20	mA
IDD+ICC	Total Supply Current		-	-	145	mA

AC CHARACTERISTICS

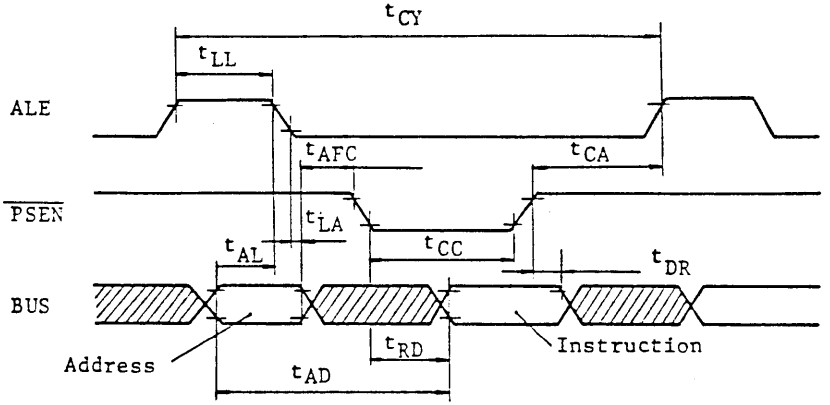
T_A=-40°C to 85°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
tLL	ALE Pulse Width		200	-	-	ns
tAL	Address Setup Time (ALE)		120	-	-	ns
tLA	Address Hold time (ALE)		80	-	-	ns
tCC	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		400	-	-	ns
tDW	Data Setup Time ($\overline{WR}$)		420	-	-	ns
tWD	Data Hold Time ($\overline{WR}$)		80	-	-	ns
tCY	Cycle Time		2.5	-	15.0	µs
tDR	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)	CL = 20 pF	0	-	200	ns
tRD	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	400	ns
tAW	Address Setup Time ($\overline{WR}$)		230	-	-	ns
tAD	Address Setup Time (Data Input)		-	-	600	ns
tAFC	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-40	-	-	ns
tCA	Internal between Control Pulse and ALE		10	-	-	ns
tCF	Port Control Setup Time (PROG)		115	-	-	ns
tPC	Port Control Hold Time (PROG)		65	-	-	ns
tPR	Port 2 Input Data Set Time (PROG)		-	-	860	ns
tDP	Output Data Setup Time (PROG)		230	-	-	ns
tPD	Output Data Hold Time (PROG)		25	-	-	ns
tPF	Port 2 Input Data Hold Time (PROG)		0	-	160	ns
tPP	PROG Pulse Width		920	-	-	ns
tPL	Port 2 I/O Data Setup Time		300	-	-	ns
tLP	Port 2 I/O Data Hold Time		120	-	-	ns

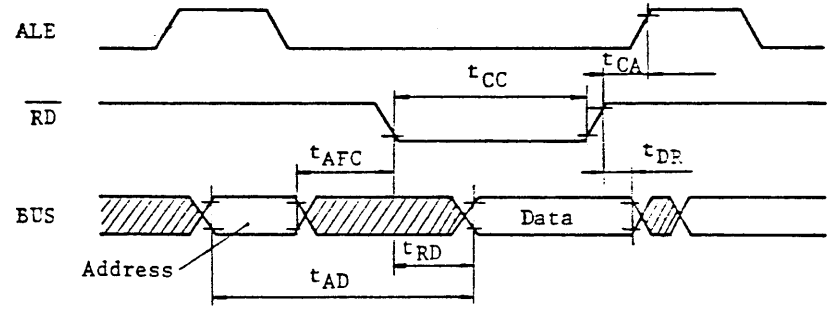
Note : tCY=2.5µs, Control Output: CL=80pF, BUS Output: CL=150pF, PORT20-23: CL=80pF.

TIMING WAVEFORM

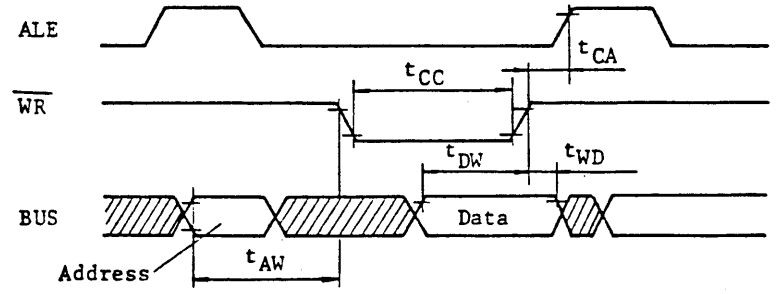
A. Instruction Fetch from External Program Memory



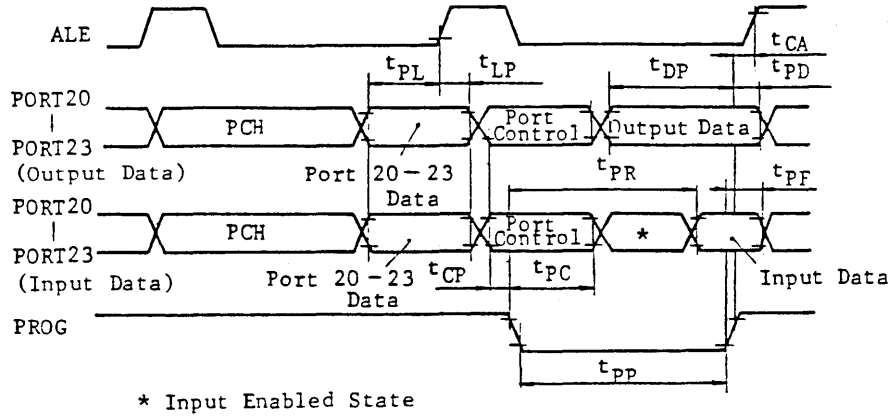
B. Read from External Data Memory



C. Write into External Data Memory



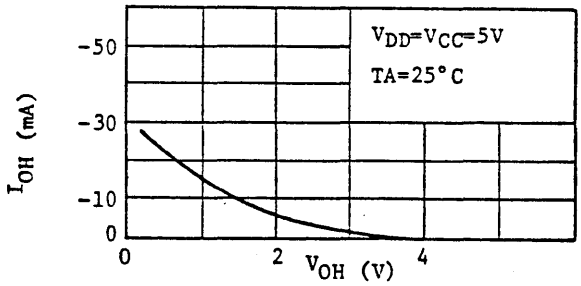
D. Timing of Port 2 during Expander Instruction Execution



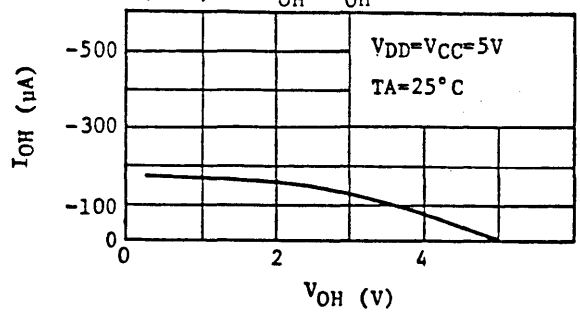
* Input Enabled State

TYPICAL CHARACTERISTICS

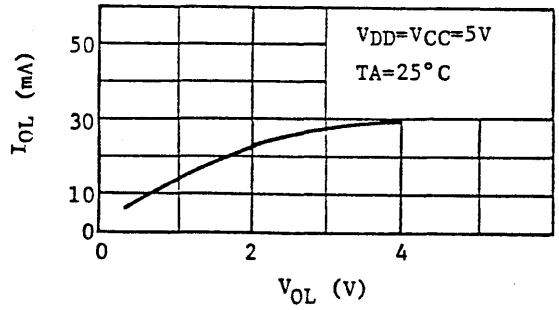
1) BUS: I_{OH} – V_{OH}



2) P1, P2: I_{OH} – V_{OH}



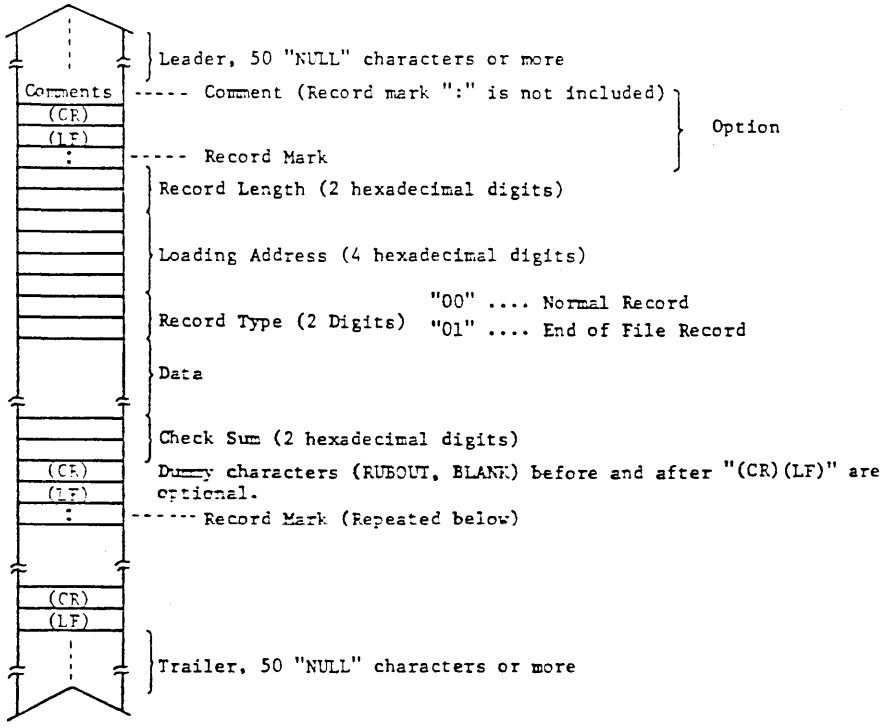
3) BUS, P1, P2: I_{OL} – V_{OL}



PROGRAM TAPE FORMAT

TMP8048 programs are delivered in the form of paper tape with the following format and it is required to attach the tape list. The format of paper tape is same as the Intel type object tape (hexadecimal tape output by Intel MDS system, PROMPT 48 Development Tool, etc.)

(1) Tape Format

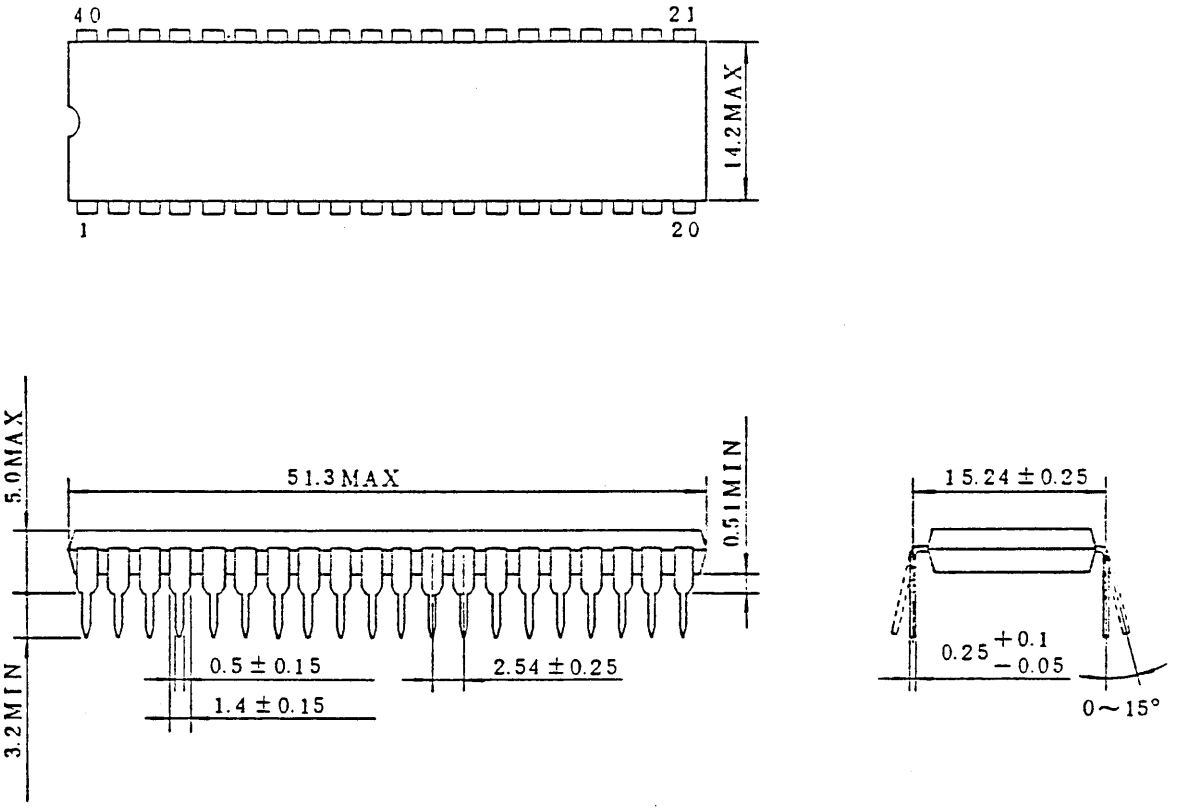


(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-48
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20VC372BF60BD6
:00000001FF
```

OUTLINE DRAWING

Unit in mm



Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP8049PI-6, from here on referred to as the TMP8049, is a single chip microcomputer fabricated in N-channel Silicon Gate MOS technology which provides internal 8-bit parallel architecture.

The following basic architectural functions of a computer have been included in a single chip; an 8-bit CPU, 128 x 8 RAM data memory, 2K x 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

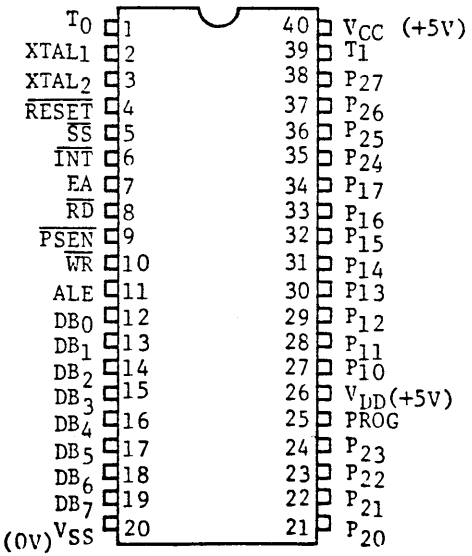
The TMP8049 is particularly efficient as a controller. It has extensive bit handing capability as well as facilities for both binary and BCD arithmetic.

The TMP8039PI is the equivalent of a TMP8049 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

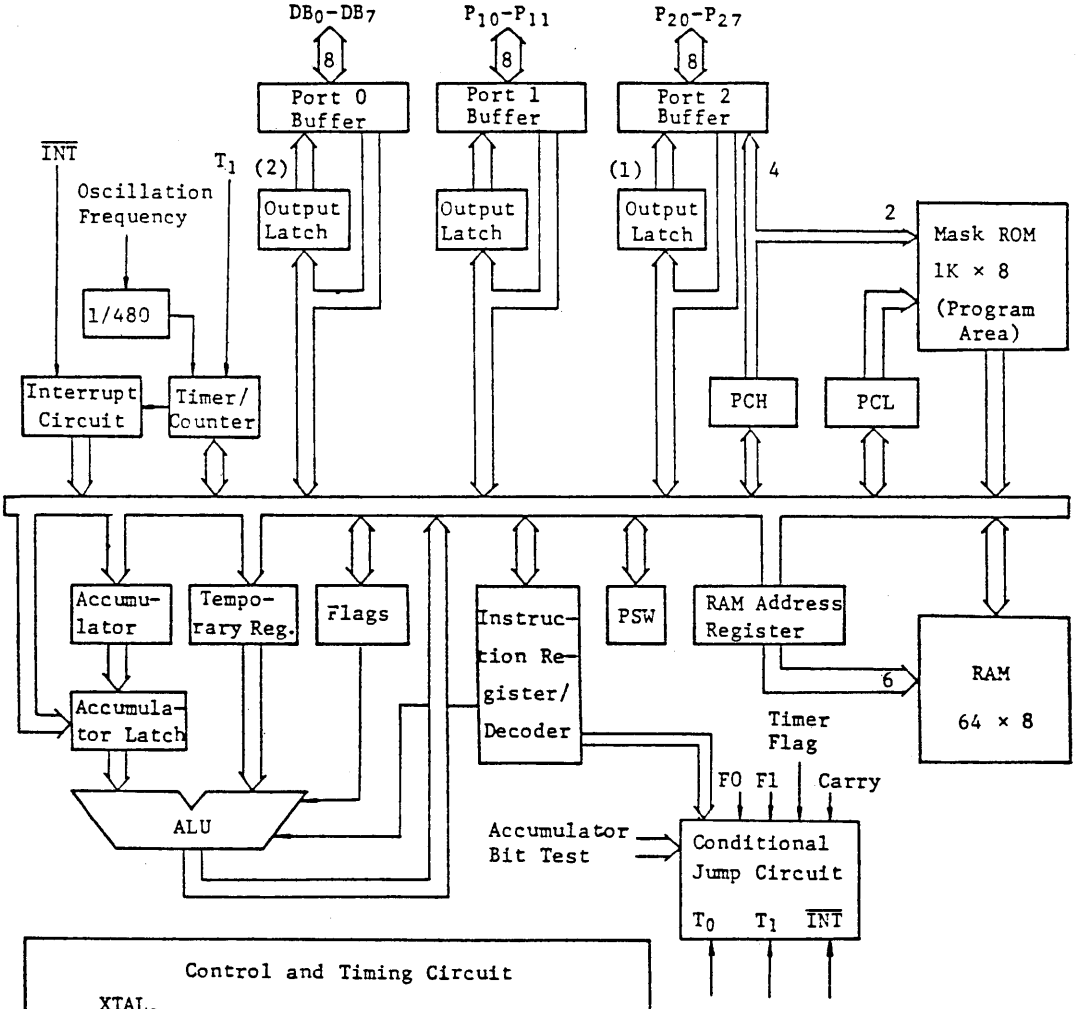
FEATURES

- 2.5 μS Instruction Cycle
- All instruction 1 or 2 cycles
- Over 90 instructions; 70% single byte
- Easy expandable memory and I/O
- 2K x 8 masked ROM
- 128 x 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Single level interrupt
- Single 5V supply
  -40°C to +85°C Operation

PIN CONNECTIONS (Top View)



BLOCK DIAGRAM



Note 1) The lower order 4 bits of port 2 output latch are used also for input/output operations with the I/O expander.

Note 2) The output latch of port 0 is also used for address output.

PIN NAMES AND PIN DESCRIPTION

VSS (Power Supply)
Circuit GND potential

VDD (Power Supply)
+5V during operation Low power standby pin for TMP8049 RAM

VCC (Main Power Supply)
+5V during operation

PROG(Output)
Output strobe for the TMP8243P I/O expander

P10-P17 (Input/Output) Port 1
8-bit quasi -bidirectional port (Internal Pullup=50kΩ).

P20-P27 (Input/Output) Port 2
8-bit quasi-bidirectional port (Internal Pullup=50kΩ).
P20-P23 Contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for the TMP8243P.

DB0-DB7 (Input/Output, 3 State)
True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.

T0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.

T1 (Input)
Input pin testable using the JT1 and JNT1 instruction. Can be designated the event counter input using the timer/STRT CNT instruction.

$\overline{INT}$ (Input)
External interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active Low)

$\overline{RD}$ (Output)
Output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory (Active Low).

$\overline{WR}$ (Output)
Output strobe during a Bus write (Active Low) Used as a Write Strobe to External Data Memory.

RESET (Input)
Active Low signal which is used to initialize the Processor. Also used during Power down.

ALE (Output)
Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

$\overline{\text{PSEN}}$ (Output)
Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)
Single step input can be used in conjunction with ALE to "single step" processor through each instruction when $\overline{\text{SS}}$ is low the CPU is placed into a wait state after it has completed the instruction being excuted.

EA (Input)
External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug and essential for testing and program verification. (Active High).

XTAL 1 (Input)
One side of crystal input for internal oscillator. Also input for external source.

XTAL 2 (Input)
Other side of crystal input.

FUNCTIONAL DESCRIPTION

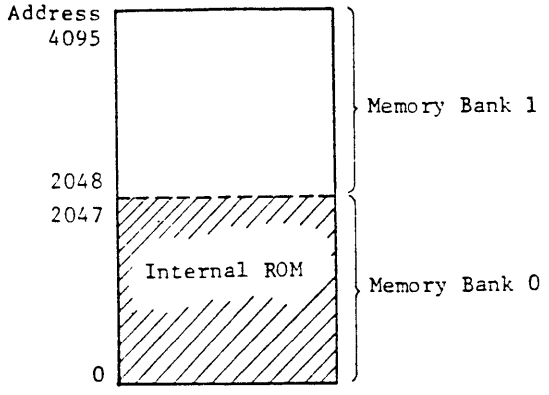
1. System Configuration

The following system functions of the TMP8049 are described in detail.

(1) Program Memory	(6) Stack (Stack Pointer)
(2) Data Memory	(7) Flag 0, Flag 1
(3) I/O Port	(8) Program Status Word (PSW)
(4) Timer/Counter	(9) Reset
(5) Interrupt Control Circuit	(10) Oscillator Circuit

(1) Program Memory
. The maximum memory that can be directly addressed by the TMP8049 is 4096 bytes. The first 2048 bytes from location 0 through 2047 can be internal resident mask ROM. The rest of the 2048 bytes of addressable memory are external to the chip. The TMP8039 has no internal resident memory; all memory must be external.

There are three locations in Program Memory of special importance.



Program Memory Area

. Location 0
Activating the Reset line of the processor causes the first instruction to be fetched from Location 0.

. Location 3
Activating the interrupt line of the processor (if interrupt enabled) causes a jump to subroutine defined by address held in Location 3.

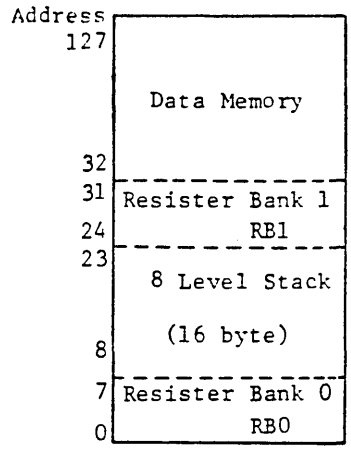
. Location 7
A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

. Program address 0-2047 and 2048-4095 are called memory banks 0 and 1 respectively switching of memory banks is achieved by changing the most significant bit of the program counter (PC) during execution of an unconditional jump instruction or call instruction executed after using SEL MB0 or SEL MB1.

Reset operation automatically selects Bank 0.

(2) Data Memory

. Resident Data Memory (volatile RAM) is organized as 128 words by 8-bits wide.
. The first 8 locations (0 -7) of the memory array are designated as working registers and are directly addressable by several instructions. By executing a Register Bank switch instruction (SEL RB1) locations 24 - 31 are designated as the working registers in place of 0 - 7.



Internal Data Memory Area

. RAM locations 8 - 23 serve a dual role in that they contain the program counter stack which is a stack 2 bytes wide by 8 levels deep. These locations store returning addresses from subroutines. If the level of subroutine nesting is less than the permitted 8, you free up 2 bytes of RAM for general use for every level of nesting not utilized.

. ALL 128 locations are indirectly addressable through either of two RAM Pointer Registers which reside at R0 and R1 of the Register array.

. The TMP8049 architecture allows extension of the Data Memory to 256 words.

(3) Input/Output Ports

. The TMP8049 has 27 I/O lines which can be used for either input or output. These I/O lines are grouped into 3 ports each having 8 bidirectional lines and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

. Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction.

. All lines of Ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure (illustrated in Figure 1). Each line is continously pulled to a +5V level through a high impedance resistive device (50kΩ) which is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. In order to speed up the "0" to "1" transition a low impedance device (5kΩ) is switched in momentarily whenever a "1" is written to line. When a "0" is written to line a low impedance device overcomes the pullup and provides TTL current sinking capability.

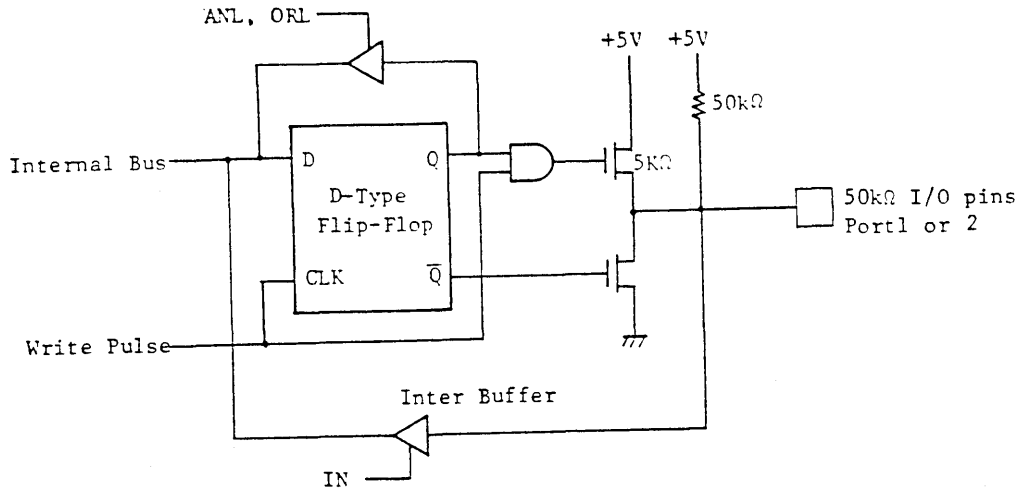


Fig.1 Input/Output Circuit of Port 1, Port 2

- Reset initializes all lines to a high impedance "1" state.

- When external data memory area is not addressed during excution of an internal program, Port 0 (DB0 - DB7) becomes a true bidirectional port (bus) with associated input and output strobes. If bidirectional feature not needed Bus can serve as either a statically latched output port or a non-latched input port. However, I/0 lines of this port cannot be intermixed.

- As a static port data is written and latched using the OUTL instruction and inputted using the INS instruction these two commands generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ strobe lines.

- As a bidirectional port the MOVX instructions are used to read and write the port which generate the $\overline{RD}$ and $\overline{WR}$ strobes.

- When not being written or read, the Bus lines are in a high impedance state.

(4)Timer/Event Counter

- The 8-bit binary up counter can use either of the following frequency inputs

  (1) Internal clock (1/480 of OSC frequency)
  ..... Timer mode

(2) External input clock form T1 terminal
(minimum cycle time 3 x ALE cycle)
..... Event Counter mode

The counter is presettable and readable with two MOV instructions which transfer the content of the accumulator to the counter and vice versa. The counter content is not affected by a Reset and is initialized solely by the MOVT, A instruction. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started by START T instruction or as an event counter by a START CNT. Once started the counter will increment to its maximum count (FF) and overflow to Zero continuing its count until stopped by a STOP TCNT instruction or RESET.

The increment from maximum count to Zero (overflow) results in the setting of an overflow flag and the generation of an interrupt request. When interrupt acknowledged a subroutine call to Location 7 will be initiated. Location 7 should store the starting address of the timer or counter service routine. The state of the overflow flag is testable with the conditional JUMP (JTF). The flag is reset by excuting a JTF or by RESET. Figure 2 illustrates the concept of the timer circuit.

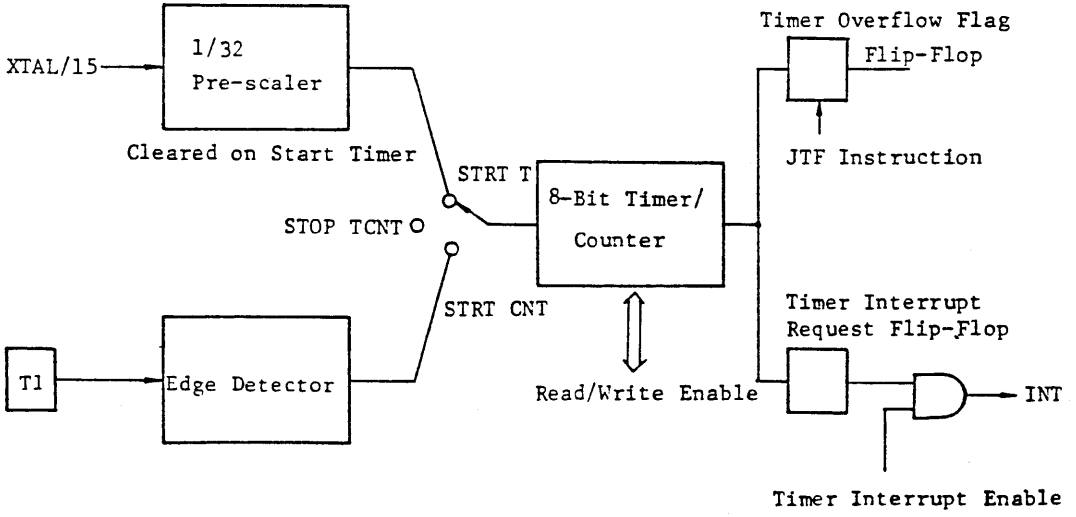


Fig.2 Concept of Timer Circuit

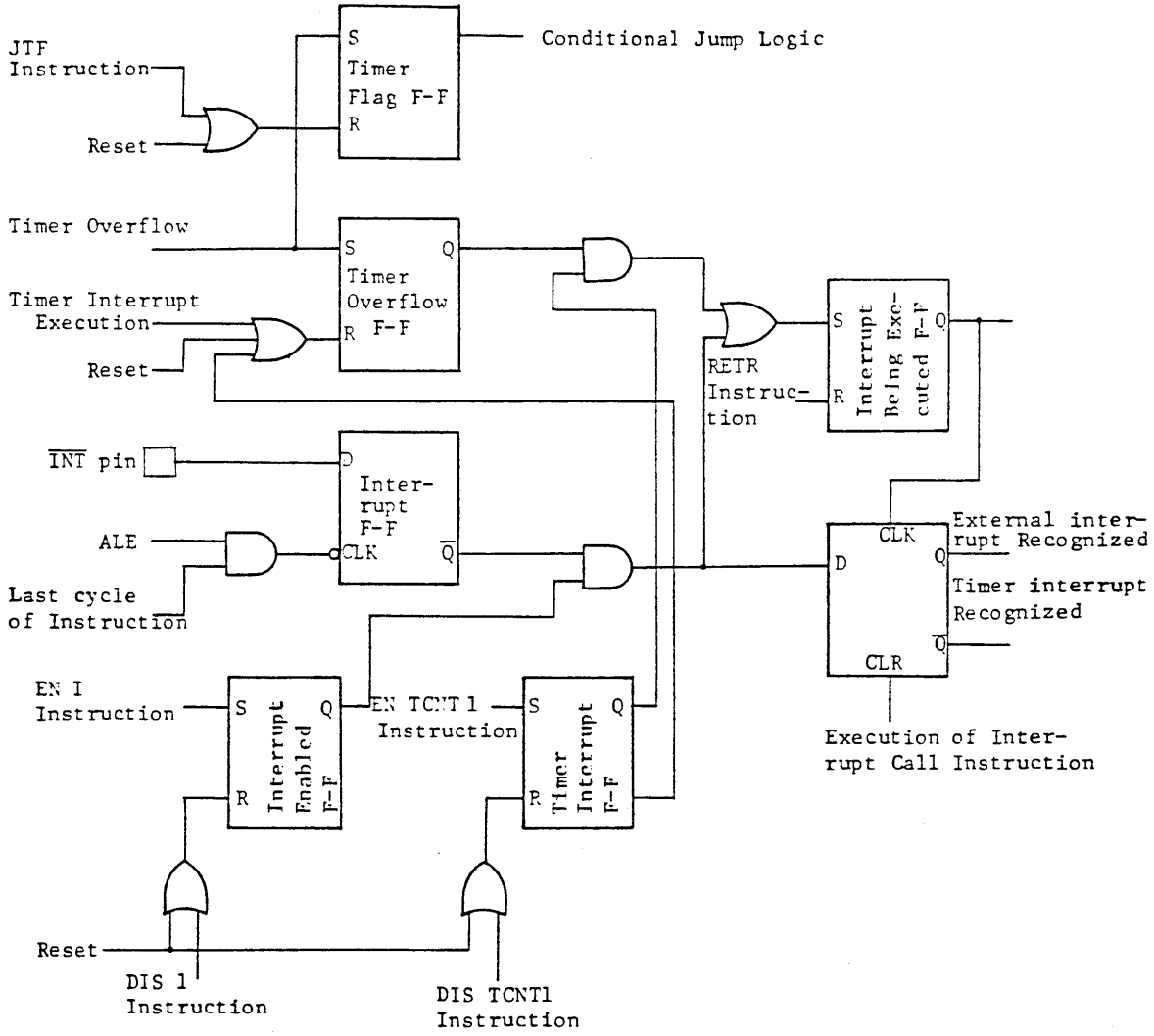


Fig.3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

. There are two distinct types of Interrupts in the TMP8049.

(1) External Interrupt from the $\overline{INT}$ terminal

(2) Timer Interrupt caused by timer overflow

The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR (which should occur at the end of an interrupt service routine) reenables the interrupt input logic.

- An interrupt sequence is initiated by applying a low level "0" to the $\overline{INT}$ pin. $\overline{INT}$ is level triggered and active low which allows "Wire Oring" of several interrupt sources. The interrupt level is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at Location 3. As in any call to subroutine, the Program Counter and Program Status Word are saved in the stack.

- When an overflow occurs in the internal timer/event counter an interrupt request is generated which is reserviced as outlined in previous paragraph except that a jump to Location 7 is used instead of 3. If $\overline{INT}$ and times overflow occur simultaneously then external request $\overline{INT}$ takes precedence.

- If an extra external interrupt is needed in addition to $\overline{INT}$ this can be achieved by enabling the counter interrupt, loading FFH in the counter (one less than the terminal count), and enabling the event counter mode. A "1" to "0" transition on T1 will cause an interrupt vector to Location 7.

- The interrupt service routine pointed to be addresses in Location 3 or 7 must reside in memory between 0 and 2047, i,e., Bank 0.

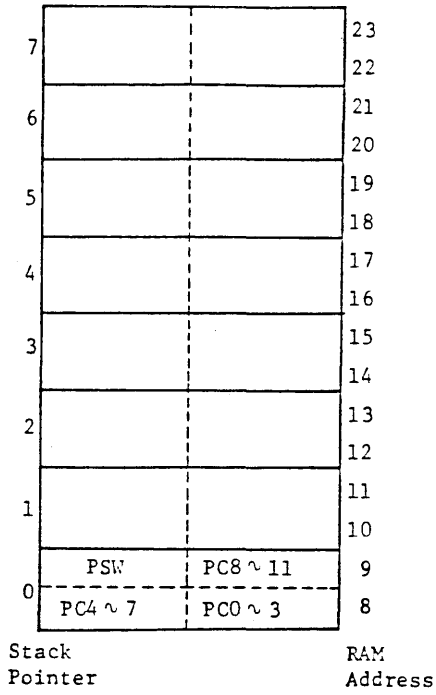
Figure 3 illustrates the concept of the interrupt control circuit.

(6) Stack (stack Pointer)

- An interrupt or Call to subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit stack pointer which is part of the Program Status Words (PSW explained in section (8)). Data RAM locations, 8 through 23 are available as stack registers and are used to store the program counter and 4-bits of PSW as shown in the figure.

- The stack pointer when initialized points to RAM location 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to Locations 8 and 9. Then the stack pointer is incremented by one to point to Locations 10 and 11. Eight levels of subroutine are obviously possible.

- At the end of a subroutine signalled by a RET or RETR causes the stack pointer to be decremented by one and the contents of the resulting pair to be transferred to the Program Counter.



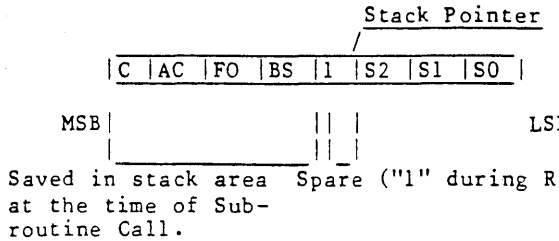
(7) Flag 0, Flag 1, (F0, F1)

. The TMP8049 has two flags F0 and F1 which are used for conditional jump. These flags can be set, reset and tested with the conditional jump instruction JF0.

. F0 is a part of the program status word (PSW) and is saved in the stack area when a subroutine is called.

(8) Program Status Word (PSW)

. An 8-bit status word which can be loaded to and from the accumlator exists called the Program Status Word (PSW). The PSW is read by a MOV A, PSW and written to by a MOV PSW, A. The information available in the PSW is shown in the diagram below.



Bits 0 - 2 : Stack Pointer Bits(S0, S1, S2)
Bit 3 : Not used ("1" level when read.)
Bit 4 : Working Register Bank Switch Bit (BS)

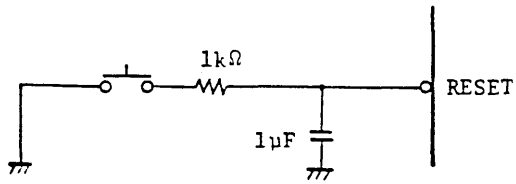
0 = Bank 0
1 = Bank 1

Bit 5 : Flag 0 (F0)
Bit 6 : Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA, A (AC)

Bit 7 : Carry (C) flag which indicates that the previous operation has resulted in the accumulator. (C)

(9) Reset

- The reset input provides a means for initialization of the processor. This Schmitt trigger input has an internal pullup registor which in combination with an external 1μF capacitor provides an internal reset pulse sufficient length to guarantee that all internal logic is initialized.



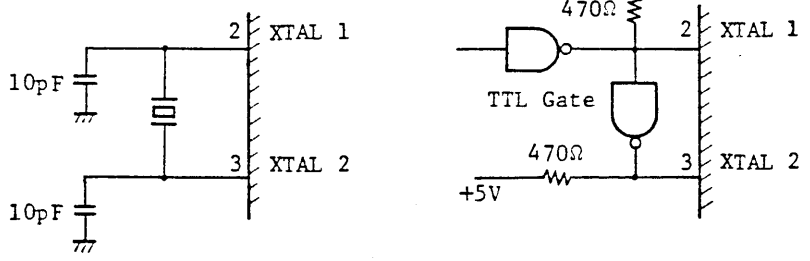
If the pulse is generated externally the reset pin must be held at ground ($\leq$0.5V) for at least 50mS after the power supply is within tolerance.

. Reset performs the following functions within the chip:

(i) Sets PC to Zero.
(ii) Sets Stack Pointer to Zero.
(iii) Selects Register Bank 0.
(iv) Selects Memory Bank 0.
(v) Sets BUS (DB0 - DB 7) to high impedance state. (Except when EA = 5V)
(vi) Sets Ports 1 and 2 to input mode.
(vii) Disables interrupts (timer and external).
(viii) Stops Timer.
(ix) Clears Timer Flag.
(x) Clears F0 and F1.
(xi) Disables clock output from T0.

(10) Oscillator Circuit

. TMP8049 can be operated by the external clock input in addition to crystal oscillator as shown below.



2. Basic Operation and Timing

The following basic operations and timing are explained
(1) Instruction Cycle
(2) External Memory Access Timing
(3) Interface with I/O Expander TMP8243P
(4) Internal Program Verify (Read) Timing
(5) Single Step Operation Timing
(6) Low Power Stand-by Mode

(1) Instruction Cycle

. The instructions of TMP8049 are executed in one or two machine cycles, and one machine cycle contents of five states.

. Fig.4 illustrates its relationship with the clock input to CPU.

. ϕ2 clock shown in Fig.4 is derived to outside by ENTO CLK instruction.

. ALE can be also used as the clock to indicate the machine cycle as well as giving the external address latch timing.

(2) External Memory Access Timing

(i) Program Memory Access

. TMP8049 programs are excuted in the following three modes.

(1) Execution of internal program only.
(2) Execution of both external and internal programs.
(3) Execution of external program only.

The external program memory is accessed (instructions are fetched) automatically when the internal ROM address is exceeded in mode (2) and from initial start address 0 in mode (3).

. In the external program memory access operation, the following will occur

. The contents of the 12-bit program counter will be output on BUS(DB0 - DB7) and the lower 4-bits of Port 2.
. Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.
. Program Store Enable (PSEN) indicates that an external instruction fetch is in progress and serves to enable the external memory device.
. BUS (DB0 - DB7) reverts to Input mode and the processor accepts its 8-bit contents as an Instruction Word.

. Figure 5 illustrates the timing.

(ii) Access of External Data Memory

. In the extended data memory access operation during READ/WRITE cycle the following occurs
. The contents of R0 R1 is output onto BUS (DB0 - DB7).
. ALE indciates address is valid. The trailing edge of ALE is used to latch the address externally.
. A read $\overline{RD}$ or write $\overline{WR}$ pulse on the corresponding output pins indicates the type of data memory access in progress. Output data valid at trailing edge of $\overline{WR}$ and input data must be valid at trailing edge of $\overline{RD}$.
. Data (8-bits) is transferred over BUS.

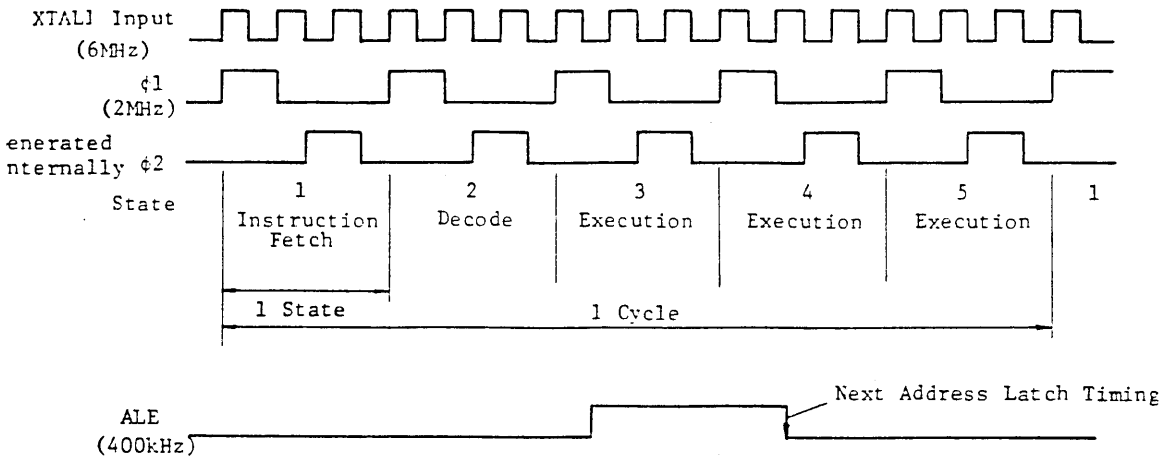


Fig.4 Instruction Cycle Timing

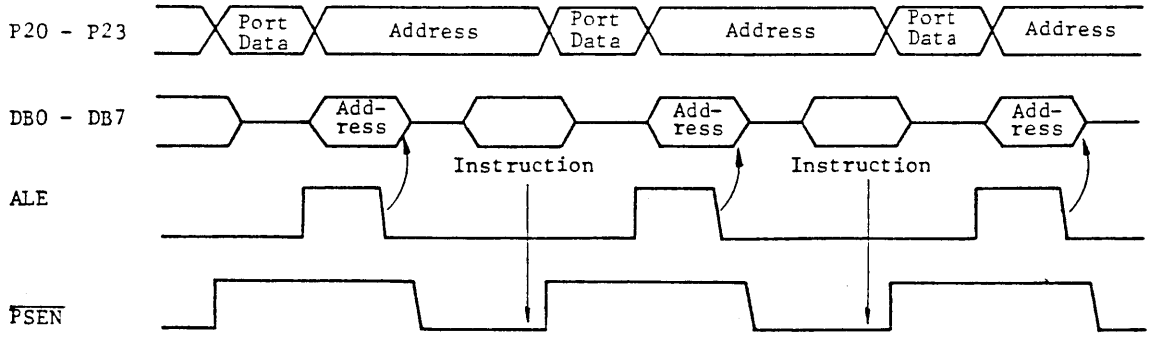
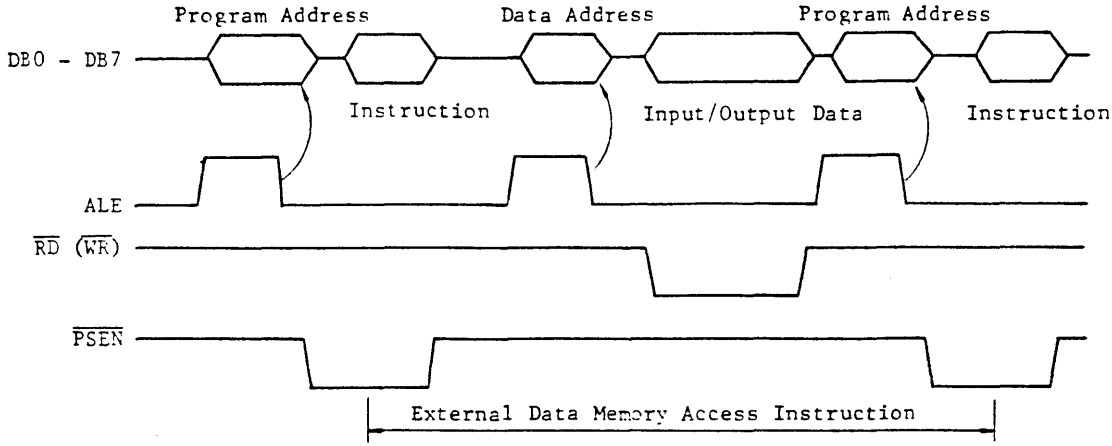


Fig. 5 Timing of External Program Memory Access



Suggest we have diagrams

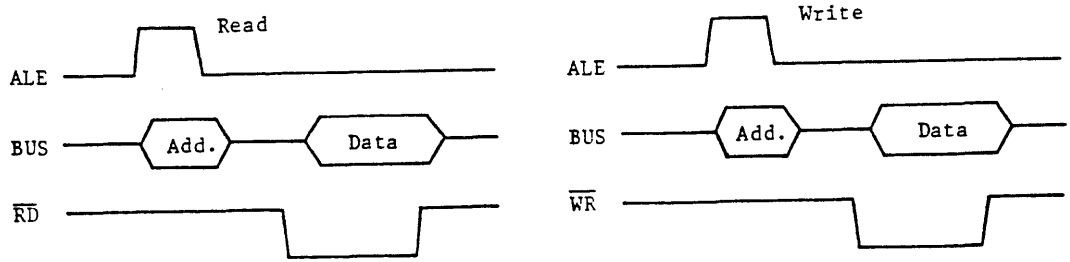


Fig. 6 Timing of Accessing External Data Memory

. Figure 6 illustrates the timing of accessing the external data memory during execution of external program.

(3) Interface with I/O Expander (TMP8243P)

. The TMP8049 I/O can be easily expanded using the TMP8243 I/O Expander. THis device uses only the lower half 4-bits of Port 2 for commuication with the TMP8049. The TMP8243 contains four 4-bit I/O ports which serve as extensions of one chip I/O and are addressed as Ports (4-7). All communication takes place over the lower half of port 2 (P20 - P23) with timing provided by an output pulse on the PROG pin. Each transfer consists of two 4-bit nibbles the first containing the "OP Code" and port address and the second containing the actual 4-bits of data.

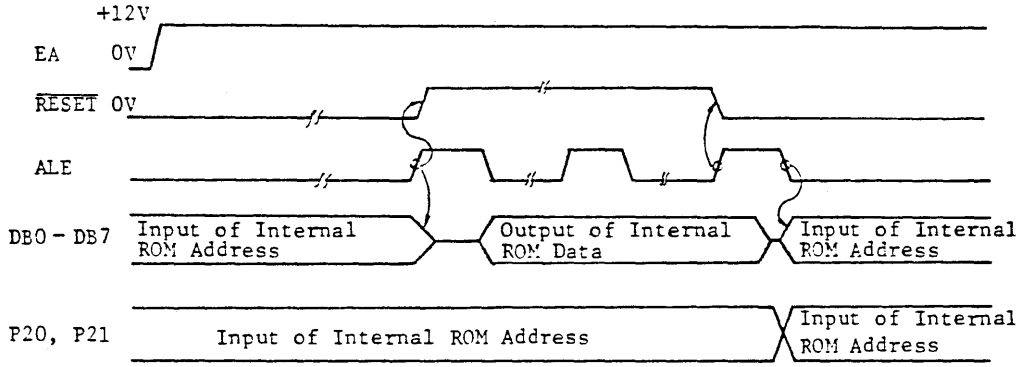


Fig.7 Timing of Reading Internal Program Memory

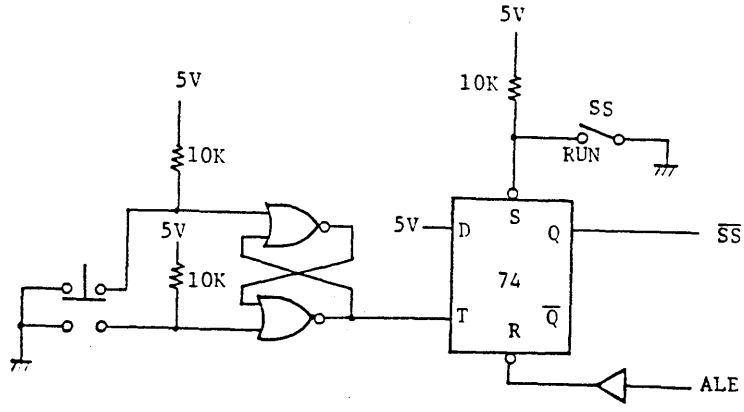


Fig.8 (a) Single Step Circuit

Reading of Internal Program Memory

. The processor is placed in the READ mode by applying +12V to the EA pin and 0V to the RESET pin. The address of the location to be read is then applied to BUS and the low order 2-bits of Port 2. The address is latched by a 0 to 1 transition on RESET and the high level causes the contents of program memory location addressed to appear on the eight lines of BUS.

. Figure 7 illustrates the timing diagram for this operation.

(5) Single Step Operation.

. A single step feature useful for debug can be implemented by utilizing a circuit shown in Figure 8 (a) combined with the SS pin and ALE pin.

. A D-type flip flop with set and reset is used to generate $\overline{SS}$. In the run mode $\overline{SS}$ is held high by keeping the flip flop set. To enter single step, set is removed allowing ALE to bring $\overline{SS}$ low via reset input. The next instruction is started by clocking a "1" into the FF which will not appear on $\overline{SS}$ unless ALE is high removing reset. In response to $\overline{SS}$ going high the processor begins an instruction fetch which brings ALE low resetting FF and causing the processor to again enter the stopped state.

. The timing diagram in this case is as shown in Figure 8 (b). (EA = 5V).

(6) Lower Power Stand-by Mode.

. The Lower TMP8049 has been organized to allow power to be removed from all but the volatile, 128 x 8 data RAM array. In power down mode the contents of data RAM can be maintained while drawing typically 10 - 15% of normal operating power requirements.

. VCC serves as the 5V supply for the bulk of the TMP8049 while the VDD supplies only the RAM array. In standby mode VCC is reduced to 0V but VDD is kept at 5V. Applying a low level to reset inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from VCC.

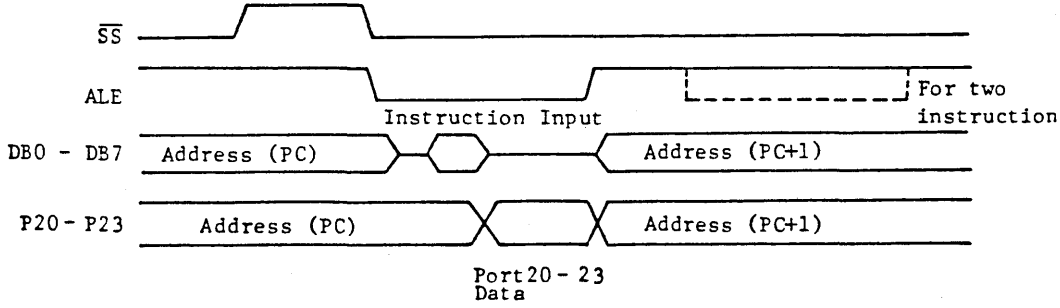


Fig.8(b) Single Step Operation Timing

INSTRUCTION

ACCUMULATOR INSTRUCTION

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
ADD A,Rr	0	1	1	0	1	r	r	r	(A)<-(A)+(Rr) r = 0 - 7	1	1	o	o
ADD A,@Rr	0	1	1	0	0	0	0	r	(A)<-(A)+((Rr)) r = 0, 1	1	1	o	o
ADD A,#Data	0	0	0	0	0	0	1	1	(A)<-(A)+Data	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ADDC A,Rr	0	1	1	1	1	r	r	r	(A)<-(A)+(Rr)+(C) r = 0 - 7	1	1	o	o
ADDC A,@Rr	0	1	1	1	0	0	0	r	(A)<-(A)+((Rr))+(C) r = 0, 1	1	1	o	o
ADDC A,#Data	0	0	0	1	0	0	1	1	(A)<-(A)+Data+(C)	2	2	o	o
	d7	d6	d5	d4	d3	d2	d1	d0					
ANL A,Rr	0	1	0	1	1	r	r	r	(A)<-(A) and (Rr) r = 0 - 7	1	1	-	-
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)<-(A)and ((Rr)) r = 0, 1	1	1	-	-
ANL A,#Data	0	1	0	1	0	0	1	1	(A)<-(A) and Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
ORL A,Rr	0	1	0	0	1	r	r	r	(A)<-(A) or (Rr) r = 0 - 7	1	1	-	-
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)<-(A) or ((Rr)) r = 0, 1	1	1	-	-
ORL A,#Data	0	1	0	0	0	0	1	1	(A)<-(A) or Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
XRL A, Rr	1	1	0	1	1	r	r	r	(A)<-(A) EOR (Rr) r = 0 - 7	1	1	-	-
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)<-(A) EOR((Rr)) r = 0, 1	1	1	-	-
XRL A,#Data	1	1	0	1	0	0	1	1	(A)<-(A) EOR Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
INC A	0	0	0	1	0	1	1	1	(A)<-(A)+1	1	1	-	-
DEC A	0	0	0	0	0	1	1	1	(A)<-(A)-1	1	1	-	-
CLR A	0	0	1	0	0	1	1	1	(A)<-0	1	1	-	-
CPL A	0	0	1	1	0	1	1	1	(A)<-NOT (A)	1	1	-	-
DA A	0	1	0	1	0	1	1	1	Decimal Adjust Accumulator	1	1	o	-
SWAP A	0	1	0	0	0	1	1	1	(A4-7)->(A0-3) <-	1	1	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
RL A	1	1	1	0	0	1	1	1	(An+1)<-(An)	1	1	-	-
									n = 0 - 6				
									(A0)<-(A7)				
RLC A	1	1	1	1	0	1	1	1	(An+1)<-(An)	1	1	-	-
									n = 0 - 6				
									(C)<-(A7)				
									(A0)<-(C)				
RR A	0	1	1	1	0	1	1	1	(An)<-(An+1)	1	1	-	-
									n = 0 - 6				
									(A7)<-(A0)				
RRC A	0	1	1	0	0	1	1	1	(An)<-(An+1)	1	1	-	-
									n = 0 - 6				
									(C)<-(A0)				
									(A7)<-(C)				

Input/Output Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
IN A,Pp	0	0	0	0	1	0	P	P	(A)<-(Pp)	1	2	-	-
									P = 1, 2				
OUTL Pp,A	0	0	1	1	1	0	P	P	(Pp)<-(A)	1	2	-	-
									P = 1, 2				
ANL Pp,#Data	1	0	0	1	1	0	P	P	(Pp)<-(Pp)and Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	P = 1, 2				
ORL Pp,#Data	1	0	0	0	1	0	P	P	(Pp)<-(Pp)or Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	P = 1, 2				
INS A,BUS	0	0	0	0	1	0	0	0	(A)<-(BUS)	1	2	-	-
OUTL BUS,A	0	0	0	0	0	0	1	0	(BUS)<-(A)	1	2	-	-
ANL BUS,#Data	1	0	0	1	1	0	0	0	(BUS)<-(BUS) and	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	Data				
ORL BUS,#Data	1	0	0	0	1	0	0	0	(BUS)<-(BUS) or	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	Data				
MOVD A,Pp	0	0	0	0	1	1	P	P	(A0-3)<-(Pp)	1	2	-	-
									(A4-7)<-0				
									P = 4 - 7				
MOVD Pp,A	0	0	1	1	1	1	P	P	(Pp)<-(A0-3)	1	2	-	-
									P = 4 - 7				
ANLD Pp,A	1	0	0	1	1	1	P	P	(Pp)<-(Pp)and	1	2	-	-
									(A0-3)				
									P = 4 - 7				
ORLD Pp,A	1	0	0	0	1	1	P	P	(Pp)<-(Pp)or(A0-3)	1	2	-	-
									P = 4 - 7				

Register Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
INC Rr	0	0	0	1	1	r	r	r	(Rr)<--(Rr)+1 r = 0 - 7	1	1	-	-
INC @Rr	0	0	0	1	0	0	0	r	((Rr))<-((Rr))+1 r = 0, 1	1	1	-	-
DEC Rr	1	1	0	0	1	r	r	r	(Rr)<--(Rr)-1 r = 0 - 7	1	1	-	-

Branch Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JMP Address	a10	a9	a8	0	0	1	0	0	(PC0-7)<--(a0-7) (PC8-10)<--(a8-10) (PC11)<--DBF	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JMPP @A	1	0	1	1	0	0	1	1	(PC0-7)<--((A))	1	2	-	-
DJNZ Rr, Address	1	1	1	0	1	r	r	r	(Rr)<--(Rr)-1 if Rr not 0 (PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JC Address	1	1	1	1	0	1	1	0	(PC0-7)<--(a0-7) if C = 1 (PC) = (PC)+2 if C = 0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JNC Address	1	1	1	0	0	1	1	0	(PC0-7)<-(a0-7) if C = 0 (PC)<--(PC)+2 if C = 1	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JZ Address	1	1	0	0	0	1	1	0	(PC0-7)<--(a0-7) if (A) = 0 (PC)<--(PC)+2 if (A) .NEQ.0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JNZ Address	1	0	0	1	0	1	1	0	(PC0-7)<--(a0-7) if (A) .NEQ.0 (PC)<--(PC)+2 if (A) = 0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JTO Address	0	0	1	1	0	1	1	0	(PC0-7)<--(a0-7) if TO = 1 (PC)<--(PC)+2 if TO = 0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JNTO Address	0	0	1	0	0	1	1	0	(PC0-7)<--(a0-7) if TO = 0 (PC)<--(PC)+2 if TO = 1	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JT1 Address	0	1	0	1	0	1	1	0	(PC0-7)<--(a0-7) if T1 = 1 (PC)<--(PC)+2 if T1 = 0	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					
JNT1 Address	0	1	0	0	0	1	1	0	(PC0-7)<--(a0-7) if T1 = 0 (PC)<--(PC)+2 if T1 = 1	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0					

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JF0 Address	1	0	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F0 = 1				
									(PC)<--(PC)+2				
									if F0 = 0				
JF1 Address	0	1	1	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if F1 = 1				
									(PC)<--(PC)+2				
									if F1 = 0				
JTF Address	0	0	0	1	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if TF = 1				
									(PC)<--(PC)+2				
									if TF = 0				
JNI Address	1	0	0	0	0	1	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if INT = 0				
									(PC)<--(PC)+2				
									if INT = 1				
JBb Address	b2	b1	b0	1	0	0	1	0	(PC0-7)<--(a0-7)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if Bb = 1				
									(PC)<-(PC)+2				
									if Bb = 0				
									(b = 0 - 7)				
CALL Address	a10	a9	a8	1	0	1	0	0	((SP))<--	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	(PC),(PSW4-7)				
									(SP)<--(SP)+1				
									(PC8-10)<--(a8-10)				
									(PC0-7)<--(a0-7)				
									(PC11)<--DBF				
RET	1	0	0	0	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
RETR	1	0	0	1	0	0	1	1	(SP)<--(SP)-1	1	2		
									(PC)<--((SP))				
									(PSW4-7)<--((SP))				

Flag Manipulation Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
CLR C	1	0	0	1	0	1	1	1	(C)<-- 0	1	1	0	-
CPL C	1	0	1	0	0	1	1	1	(C)<--NOT(C)	1	1	0	-
CLR F0	1	0	0	0	0	1	0	1	(F0)<-- 0	1	1	-	-
CPL F0	1	0	0	1	0	1	0	1	(F0)<--NOT(F0)	1	1	-	-
CLR F1	1	0	1	0	0	1	0	1	(F1)<-- 0	1	1	-	-
CPL F1	1	0	1	1	0	1	0	1	(F1)<--NOT(F1)	1	1	-	-

Data Transfer Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A, Rr	1	1	1	1	1	r	r	r	(A)<-- (Rr)	1	1	-	-
									r = 0 - 7			-	-
MOV A, @Rr	1	1	1	1	0	0	0	r	(A)<-- ((Rr))	1	1	-	-
									r = 0, 1				
MOV A, #Data	0	0	1	0	0	0	1	1	(A)<--Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
MOV Rr, A	1	0	1	0	1	r	r	r	(Rr)<--(A)	1	1	-	-
									r = 0 - 7				
MOV@Rr,A	1	0	1	0	0	0	0	r	((Rr))<-(A)	1	1	-	-
									r = 0, 1				
MOV Rr,#Data	1	0	1	1	1	r	r	r	(Rr)<--Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	r = 0 - 7				
MOV@Rr,#Data	1	0	1	1	0	0	0	r	((Rr))<--Data	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	r = 0, 1				
MOV A,PSW	1	1	0	0	0	1	1	1	(A)<-- (PSW)	1	1	-	-
MOV PSW, A	1	1	0	1	0	1	1	1	(PSW)<--(A)	1	1	-	-
XCH A, Rr	0	0	1	0	1	r	r	r	(A)-->(Rr)	1	1	-	-
									<--				
									r = 0 - 7				
XCH A,@Rr	0	0	1	0	0	0	0	r	(A)-->((Rr))	1	1	-	-
									<--				
									r = 0, 1				
XCHD A,@Rr	0	0	1	1	0	0	0	r	(A0-3)-->((Rr0-3))	1	1	-	-
									<--				
									r = 0, 1				
MOVX A,@Rr	1	0	0	0	0	0	0	r	(A)<--((Rr))	1	2	-	-
									r = 0, 1				
MOVX @Rr,A	1	0	0	1	0	0	0	r	((Rr))<--(A)	1	2	-	-
									r = 0, 1				
MOVP A, @A	1	0	1	0	0	0	1	1	(PC0-7)<--(A)	1	2	-	-
									(A)<--((PC))				
MOVP3 A,@A	1	1	1	0	0	0	1	1	(PC0-7)<--(A)	1	2	-	-
									(PC8-11)<--0011				
									(A)<--((PC))				

Timer/Counter Instruction

Mnemonic	Instruction Code									Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
MOV A,T	0	1	0	0	0	0	1	0	(A)<--(T)	1	1	-	-
MOV T,A	0	1	1	0	0	0	1	0	(T)<--(A)	1	1	-	-
STRT T	0	1	0	1	0	1	0	1	Counting is started in the timer mode	1	1	-	-
STRT CNT	0	1	0	0	0	1	0	1	Counting is started in the event counter mode	1	1	-	-
STOP TCNT	0	1	1	0	0	1	0	1	Stop both time accumulation and event counting	1	1	-	-
EN TCNTI	0	0	1	0	0	1	0	1	Timer interrupt is enabled	1	1	-	-
DIS TCNTI	0	0	1	1	0	1	0	1	Timer interrupt is disabled	1	1	-	-

Control Instruction

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
EN I	0	0	0	0	0	1	0	1	External interrupt is enabled	1	1	-	-
DIS I	0	0	0	1	0	1	0	1	External interrupt is disabled	1	1	-	-
SEL RB0	1	1	0	0	0	1	0	1	(BS)<-- 0	1	1	-	-
SEL RB1	1	1	0	1	0	1	0	1	(BS)<-- 1	1	1	-	-
SEL MB0	1	1	1	0	0	1	0	1	(DBF)<-- 0	1	1	-	-
SEL MB1	1	1	1	1	0	1	0	1	(DBF)<-- 1	1	1	-	-
ENTO CLK	0	1	1	1	0	1	0	1	T0 is enabled to act as the clock output	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-

TMP8049PI/8039PI: INDUSTRIAL SPECIFICATION

ABSOLUTE MAXIMUM RATTINGS

SYMBOL	ITEM	RATTING
VDD	VDD Supply Voltage (with respect to GND (VSS))	-0.5V to + 7V
VCC	VCC Supply Voltage (with respect to GND (VSS))	-0.5V to + 7V
VINA	Input Voltage (Except EA)	-0.5V to + 7V
VINB	Input Voltage (Only EA)	-0.5V to + 13V
PD	Power Dissipation (Ta = 70°C)	1.5W
TSOLDER	Soldering Temperature (Soldering Time 10 sec)	260°C
TSTG	Storage Temperature	-55°C to 150°C
TOPR	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS

TA=-40°C to 85°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage (Except XTAL1,XTAL2,$\overline{\text{RESET}}$)		-0.5	-	0.7	V
VIL1	Input Low Voltage (XTAL1,XTAL2,$\overline{\text{RESET}}$)		-0.5	-	0.6	V
VIH	Input High Voltage (Except XTAL1,XTAL2,$\overline{\text{RESET}}$)		2.2	-	VCC	V
VIH1	Input High Voltage (XTAL1,XTAL2,$\overline{\text{RESET}}$)		3.8	-	VCC	V
VOL	Output Low Voltage (BUS)	IOL = 1.6 mA	-	-	0.45	V
VOL1	Output Low Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, PSEN, ALE)	IOL = 1.6 mA	-	-	0.45	V
VOL2	Output Low Voltage (PROG)	IOL = 0.8 mA	-	-	0.45	V
VOL3	Output Low Voltage (For other output pins)	IOL = 1.2 mA	-	-	0.45	V
VOH	Output High Voltage (BUS)	IOH =-280μA	2.4	-	-	V
VOH1	Output High Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, PSEN, ALE)	IOH =-80μA	2.4	-	-	V
VOH2	Output High Voltage (For other output pins)	IOH =-30μA	2.4	-	-	V
ILI	Input Leak Current (T1, $\overline{\text{INT}}$)	VSS $\leq$ VIN $\leq$ VCC	-	-	±10	μA
ILI1	Input Leak Current (P10-17, P20-P27, EA, $\overline{\text{SS}}$)	VSS+0.45$\leq$VIN$\leq$VCC	-	-	-700	μA
ILO	Output Leak Current (BUS, TO) (High impedance condition)	VSS+0.45$\leq$VIN$\leq$VCC	-	-	±10	μA
IDD	VDD Supply Current		-	-	50	mA
IDD+ICC	Total Supply Current		-	-	170	mA

AC CHARACTERISTICS

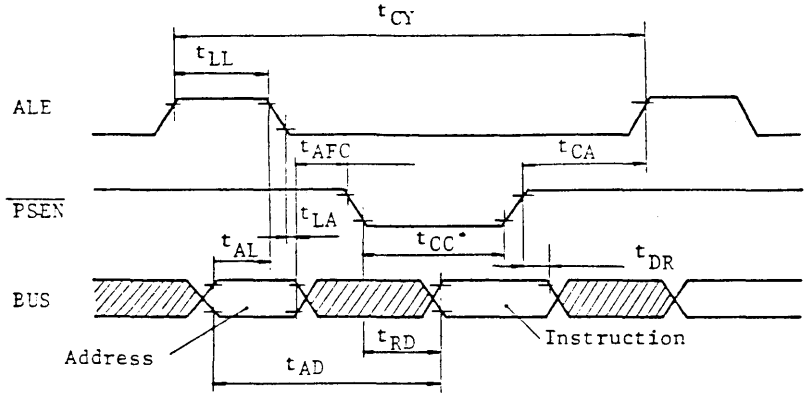
T_A=-40°C to 85°C, VCC=VDD=+5V±10%, VSS=0V, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
tLL	ALE Pulse Width		200	-	-	ns
tAL	Address Setup Time (ALE)		120	-	-	ns
tLA	Address Hold time (ALE)		80	-	-	ns
tCC	Control Pulse Width($\overline{PSEN}$,$\overline{RD}$,$\overline{WR}$)		400	-	-	ns
tDW	Data Setup Time ($\overline{WR}$)		420	-	-	ns
tWD	Data Hold Time ($\overline{WR}$)		80	-	-	ns
tCY	Cycle Time		2.5	-	15.0	μs
tDR	Data Hold Time ($\overline{PSEN}$,$\overline{RD}$)	CL = 20 pF	0	-	200	ns
tRD	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	400	ns
tAW	Address Setup Time ($\overline{WR}$)		230	-	-	ns
tAD	Address Setup Time (Data Input)		-	-	600	ns
tAFC	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-40	-	-	ns
tCA	Internal between Control Pulse and ALE		10	-	-	ns
tCP	Port Control Setup Time (PROG)		115	-	-	ns
tPC	Port Control Hold Time (PROG)		65	-	-	ns
tPR	Port 2 Input Data Set Time (PROG)		-	-	860	ns
tDP	Output Data Setup Time (PROG)		230	-	-	ns
tPD	Output Data Hold Time (PROG)		25	-	-	ns
tPF	Port 2 Input Data Hold Time (PROG)		0	-	160	ns
tPP	PROG Pulse Width		920	-	-	ns
tPL	Port 2 I/O Data Setup Time		300	-	-	ns
tLP	Port 2 I/O Data Hold Time		120	-	-	ns

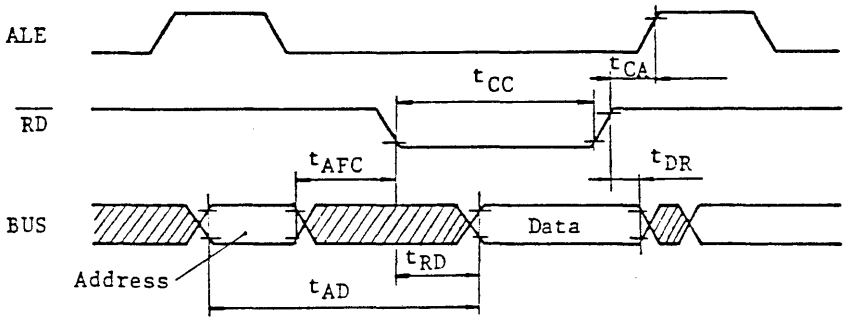
Note : tCY=2.5μs, Control Output: CL=80pF, BUS Output: CL=150pF, PORT20-23: CL=80pF.

TIMING WAVEFORM

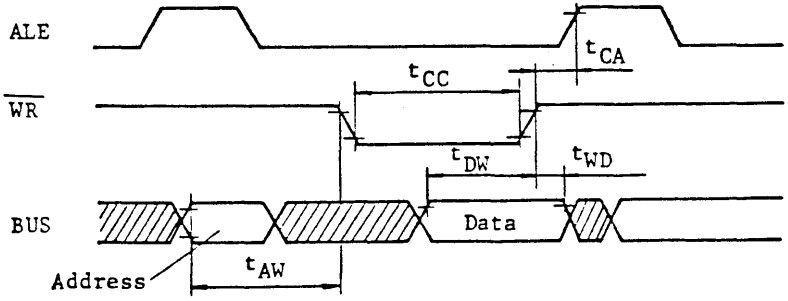
A. Instruction Fetch from External Program Memory



B. Read from External Data Memory

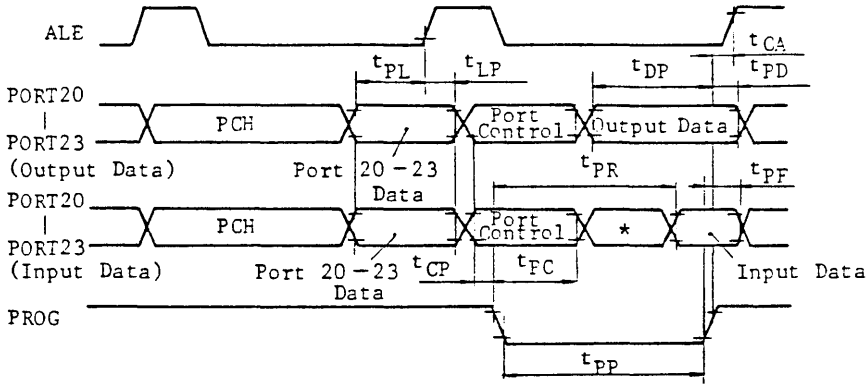


C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution

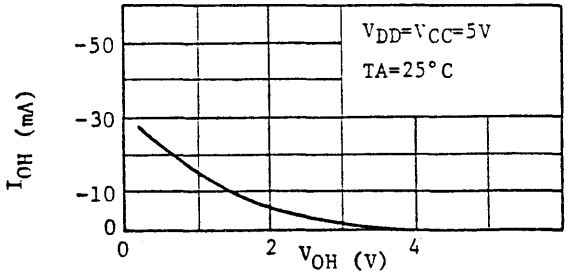
TYPICAL CHARACTERISTICS



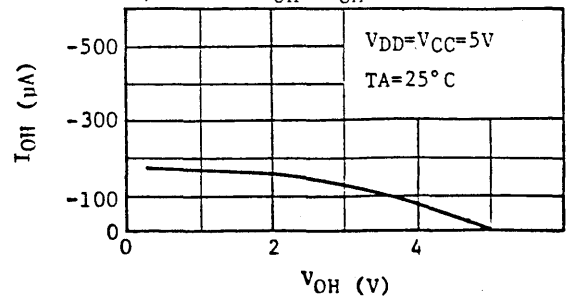
* Input Enabled State

TYPICAL CHARACTERISTICS

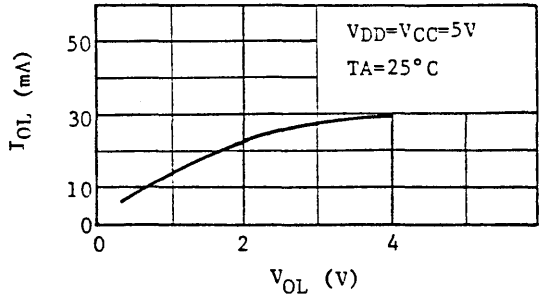
1) BUS: I_{OH} - V_{OH}



2) P1, P2: I_{OH} - V_{OH}



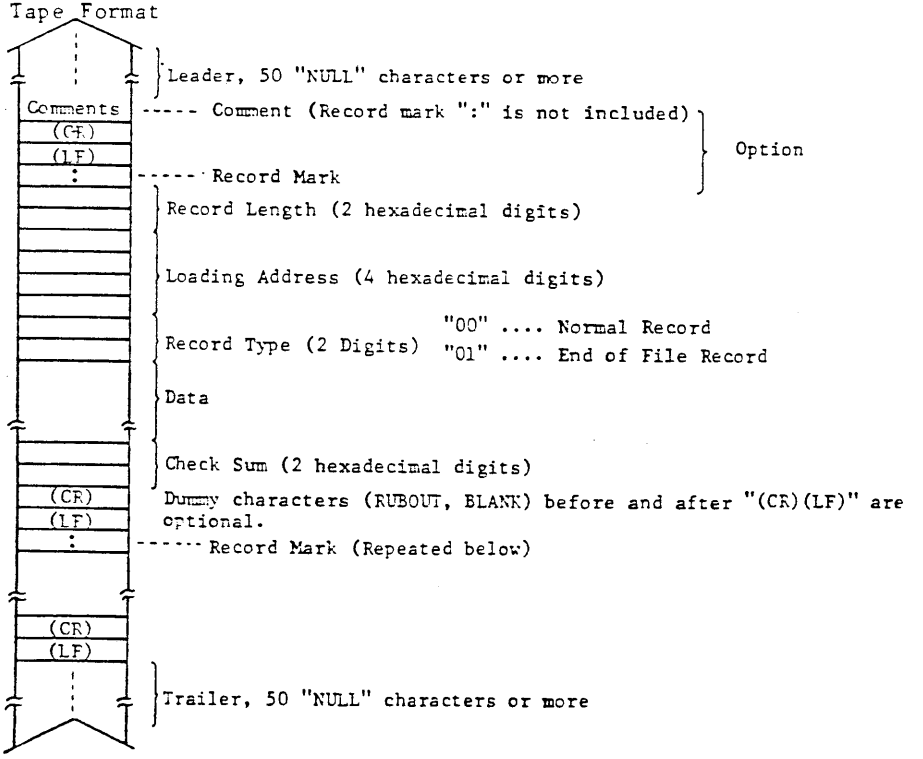
3) BUS, P1, P2: I_{OL} - V_{OL}



PROGRAM TAPE FORMAT

TMP8049 programs are delivered in the form of paper tape with the following format and it is required to attach the tape list. The format of paper tape is same as the Intel type object tape (hexadecimal tape output by Intel MDS system, PROMPT 48 Development Tool, etc.)

(1) Tape Format

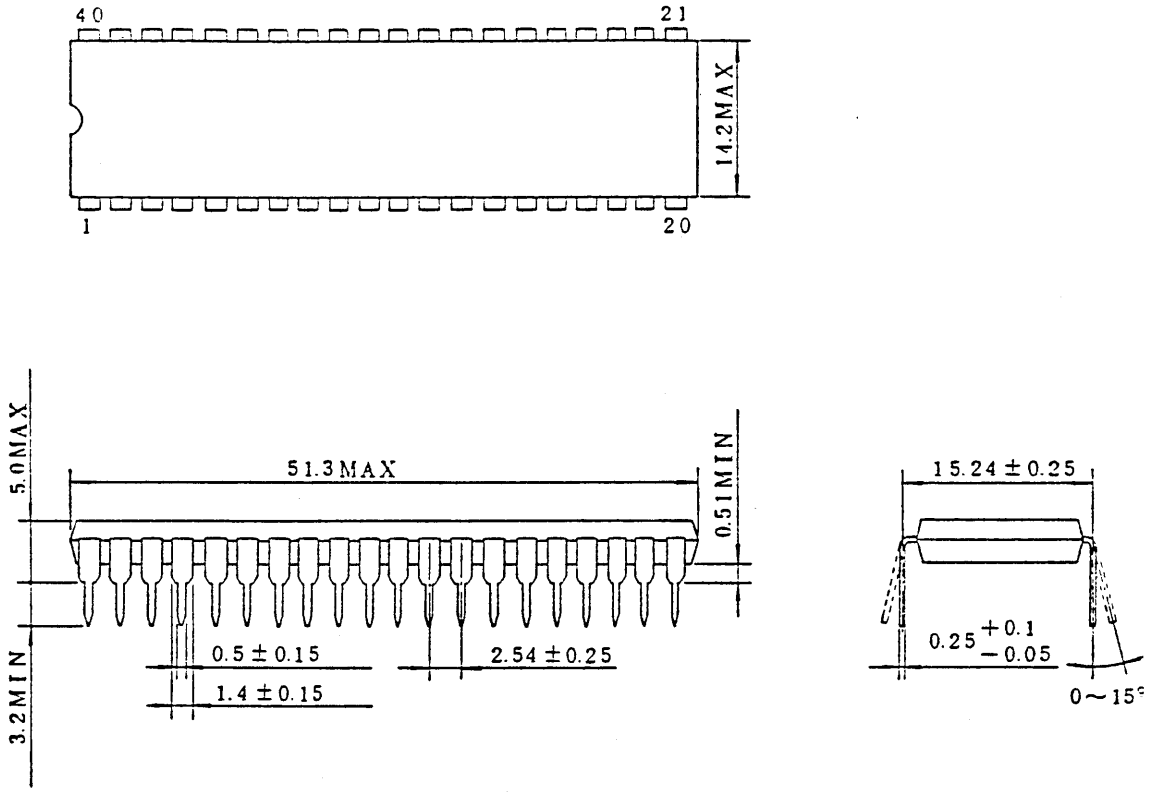


(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-48
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20VC372BF60BD6
:00000001FF
```

OUTLINE DRAWING

Unit in mm



Note: 1. This dimension is measured at the center of bending point of leads.
2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoritical positions with respect to No.1 and No.40 leads.

INPUT/OUTPUT EXPANDER

GENERAL DESCRIPTION

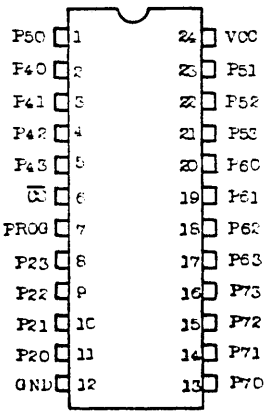
The TMP8243P is an input/output expander designed specifically to provide a low cost means of I/O expansion for the TLCS-84 family.

The I/O ports of the TMP8243P serve as a direct extension of the resident I/O facilities of the TLCS-84 microcomputers and are accessed by their own MOVD, ANLD, and ORLD instructions.

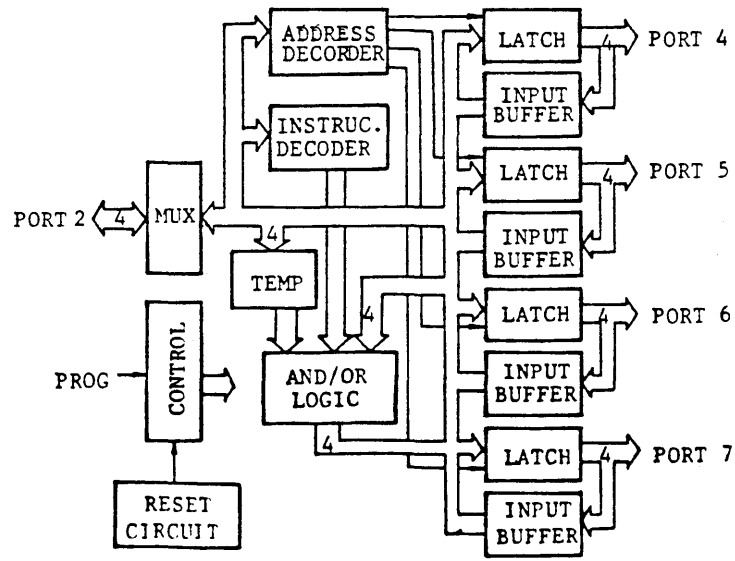
FEATURES

- o Low cost
- o Simple interface to TLCS-84 microcomputers
- o Four 4-bit I/O ports
- o AND and OR directly to ports
- o Single 5V supply
- o High output drive
- o Direct extension of resident TMP8048P/TMP8049P I/O ports.
- o Compatible with intel's 8243
- o -40°C to +85°C Operation (TMP8243PI: Industrial Specification)

PIN CONNECTION (TOP VIEW)



BLOCK DIAGRAM



PIN NAMES AND PIN DESCRIPTION

PROG (Input)
Clock Input. A high to low transistion on PROG signifies that address and control are available on P20-23, and a low to high transition signifies that data is available on P20-23.

$\overline{CS}$ (Input)
Chip Select Input. A high on CS inhibits any change of output or internal status.

P20-23 (Input/Output, 3-state)
Four (4) bit bi-directional port contains the address and control bits on a high to low transition of PROG. During a low to high transition contains the data for a selected output port if a write oepration, or the data from a selected port before the low to high transition if a read operation.

P40-43, P50-53, P60-63, P70-73 (Input/Output, 3-state)
Four (4) bit bi-directional I/O ports. May be programmed to be input (during read), low impedance latched output (after write) or a 3-state (after read). Data on pins P20-23 may be directly written, ANDed or ORed with previous data.

V_{CC} (Power)
+5 volt supply

GND (Power)
0 volt supply

FUNCTIONAL DESCRIPTION

General Operation
The TMP8243P contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports 4-7. The following operations may be performed on these ports.

- o Transfer accumulator to port
- o Transfer port to accumulator
- o AND accumulator to port
- o OR accumulator to port

All communication between the TMP8048P and the TMP8243P occurs over Port 2 (P20-23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles.

A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional TMP8243P'S may be added to the 4-bit bus and chip selected using additional output lines from the TMP8048P/8035P.

Power On Initialization

Initial application of power to the device forces input/output ports 4, 5, 6, and 7 to the tri-state and port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

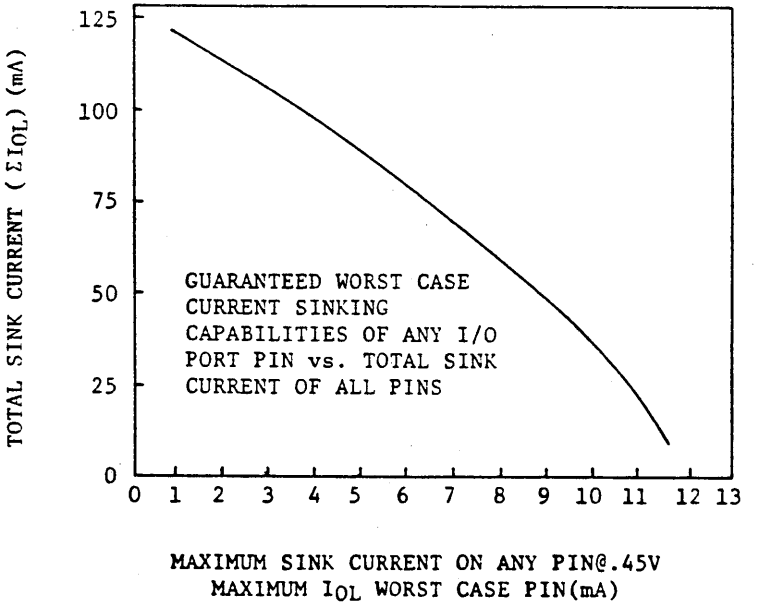
The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi, A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi, A takes new data AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are 3-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the 3-stated mode while port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the TMP8243P output. A read of any port will leave that port in a high impedance state.



Sink Capability

The TMP8243P can sink 5 mA@.45V on each of its 16 I/O lines simultaneously. If, however, all lines are not sinking simultaneously or all lines are not fully loaded, the drive capability of any individual line increases as is shown by the accompanying curve.

For example, if only 5 of the 16 lines are to sink current at one time, the curve shows that each of those 5 lines is capable of sinking 9 mA@.45V (if any lines are to sink 9 mA the total I_{OL} must not exceed 45 mA or five 9 mA loads).

Example: How many pins can drive 5 TTL loads (1.6 mA) assuming remaining pins are unloaded?

I_{OL} = 5 x 1.6 mA = 8 mA
εI_{OL} = 60 mA from curve
#pins = 60 mA ÷ 8 mA/pin = 7.5 = 7

In this case, 7 lines can sink 8 mA for a total of 56 mA. This leaves 4 mA sink current capability which can be divided in any way among the remaining 9 I/O lines of the TMP8243P.

Example: This examples shows now the use of the 20 mA sink capability of port 7 affects the sinking capability of the other I/O lines.

An TMP8243P will drive the following loads simultaneously.

2 loads - 20 mA@1V (port 7 only)

8 loads - 4 mA@.45V

6 loads - 3.2 mA@.45V

Is this within the specified limits?

εI_{OL} = (2 x 20) + (8 x 4) + (6 x 3.2) = 91.2 mA. From the curve: for I_{OL} = 4 mA, εI_{OL} = 93 mA since 91.2 mA < 93 mA the loads are within specified limits.

Although the 20 mA@1V. load are used in calculating εI_{OL}, it is the largest current required @.45V which determines the maximum allowable εI_{OL}.

TMP8243P

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{CC}	V_{CC} Supply Voltage with Respect to GND	-0.5V to +7.0V
V_{IN}	Input Voltage with Respect to GND	-0.5V to +7.0V
V_{OUT}	Output Voltage with Respect to GND	-0.5V to +7.0V
P_D	Power Dissipation	800mW
T_{SOLDER}	Soldering Temperature (Soldering Time 10 sec.)	260°C
T_{STG}	Storage Temperature	-55°C to +150°C
T_{OPR}	Operating Temperature	0°C to +70°C

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.0		$V_{CC}+0.5$	V
V_{OL1}	Output Low Voltage Ports 4-7	I_{OL} = 5mA*			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL} = 20mA			1	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.6mA			0.45	V
V_{OH1}	Output High Voltage Ports 4-7	I_{OH}=-240μA	2.4			V
V_{OH2}	Output High Voltage Port 2	I_{OH}=-100μA	2.4			
I_{IL1}	Input Leakage Port 4-7	$0V \leq V_{IN} \leq V_{CC}$	-10		20	μA
I_{IL2}	Input Leadage Port 2, $\overline{CS}$, PROG	$0V \leq V_{IN} \leq V_{CC}$	-10		10	μA
I_{CC}	V_{CC} Supply Current			10	20	mA
I_{OL}	Sum of all IO_L of 16 Outputs	5 mA Each Pin			80	mA

* See following graph for additional sink current capability

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
t_A	Code Valid Before PROG	C_L = 80pF	100			ns
t_B	Code Valid After PROG	C_L = 20pF	60			ns
t_C	Data Valid Before PROG	C_L = 80pF	200			ns
t_D	Data Valid After PROG	C_L = 20pF	20			ns
t_H	Floating After PROG	C_L = 20pF	0		150	ns
t_K	PROG Negative Pulse Width		700			ns
t_{CS}	$\overline{CS}$ Valid Before/After PROG		50			ns
t_{PO}	Ports 4-7 Valid After PROG	C_L = 100pF			700	ns
t_{LPI}	Ports 4-7 Valid Before/After PROG		100			ns
t_{ACC}	Port 2 Valid After PROG	C_L = 80pF			650	ns

TMP8243PI : INDUSTRIAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{CC}	V_{CC} Supply Voltage with Respect to GND	-0.5V to +7.0V
V_{IN}	Input Voltage with Respect to GND	-0.5V to +7.0V
V_{OUT}	Output Voltage with Respect to GND	-0.5V to +7.0V
P_D	Power Dissipation	800mW
T_{SOLDER}	Soldering Temperature (Soldering Time 10 sec.)	260°C
T_{STG}	Storage Temperature	-55°C to +150°C
T_{OPR}	Operating Temperature	-40°C to +85°C

D.C. CHARACTERISTICS TA=-40°C to 85°C, V_{CC}=5V±10%

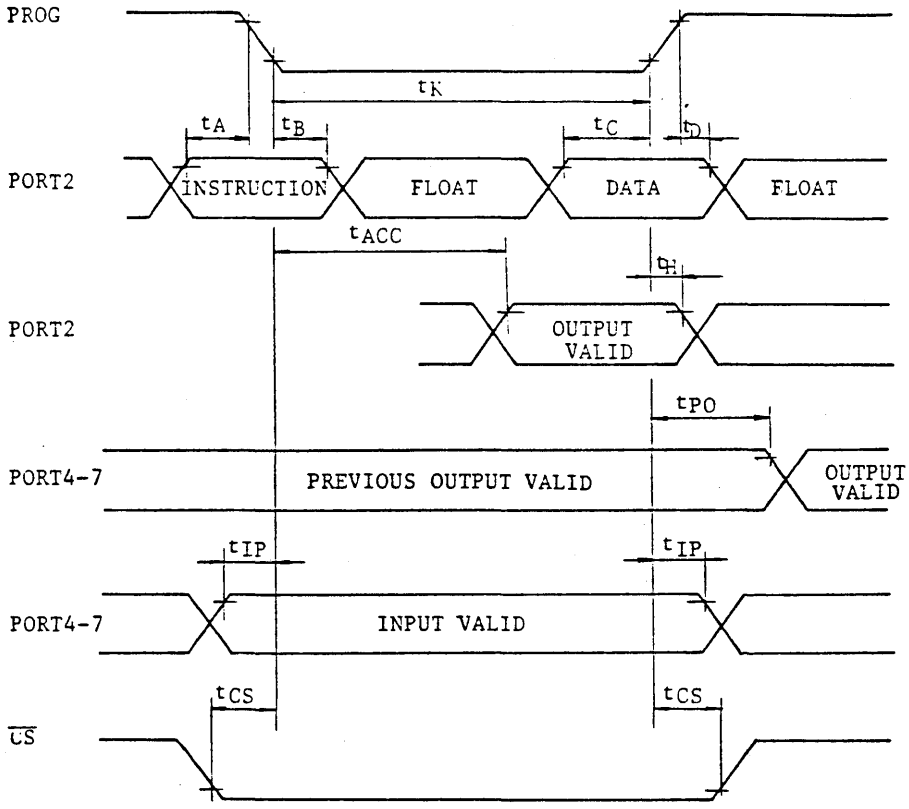
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
V_{OL1}	Outptu Low Voltage Ports 4-7	I_{OL}=4.5mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL}=20mA			1	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.6mA			0.45	V
V_{OH1}	Output High Voltage Ports 4-7	I_{OH}=-240μA	2.4			V
V_{OH2}	Output High Voltage Port 2	I_{OH}=-100μA	2.4			
I_{IL1}	Input Leakage Ports 4-7	$0V \leq V_{IN} \leq V_{CC}$	-10		20	μA
I_{IL2}	Input Leakage Port 2, $\overline{CS}$, PROG	$0V \leq V_{IN} \leq V_{CC}$	-10		10	μA
I_{CC}	V_{CC} Supply Current			10	20	mA
I_{OL}	Sum of all I_{OL} of 16 outputs	4.5mA each pin			72	mA

* See following graph for additional sink current capability

A.C. CHARACTERISTICS TA=-40°C to 85°C, V_{CC}=5V±10%

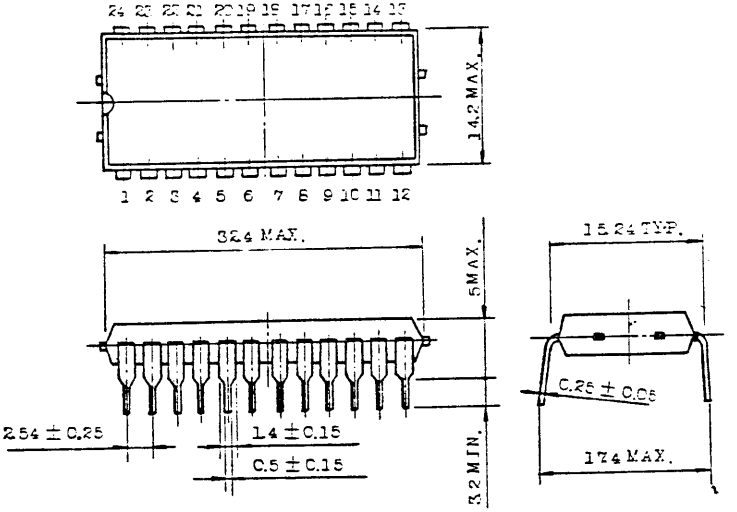
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
t_A	Code Valid before PROG	C_L = 80pF	100			ns
t_B	Code Valid after PROG	C_L = 20pF	60			ns
t_C	Data Valid before PROG	C_L = 80pF	200			ns
t_D	Data Valid after PROG	C_L = 20pF	20			ns
t_H	Floating after PROG	C_L = 20pF	0		150	ns
t_K	PROG Negative Pulse Width		700			ns
t_{CS}	$\overline{CS}$ Valid before/after PROG		50			ns
t_{PO}	Ports 4-7 Valid after PROG	C_L = 100pF			700	ns
t_{LPI}	Ports 4-7 Valid before/after PROG		100			ns
t_{ACC}	Port 2 Valid after PROG	C_L = 80pF			650	ns

TIMING WAVEFORM



OUTLINE DRAWINGS

Unit in mm



Note: Each lead pitch is 2.54mm. All leads are located within 0.25mm of their true longitudinal position with respect to No.1 and No.24 leads.
All dimensions are in millimeters.

INPUT/OUTPUT EXPANDER

GENERAL DESCRIPTION

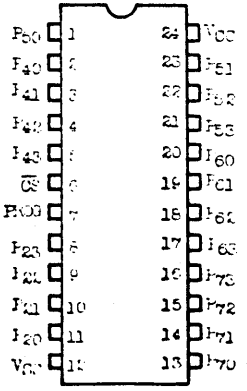
The TMP82C43P is an input/output expander designed specifically to provide a low cost means of I/O expansion for the TLCS-84C family.

The I/O ports of the TMP82C43P serve as a direct extension of the resident I/O facilities of the TLCS-84C microcomputers and are accessed by their own MOVD, ANLD, and ORLD instructions.

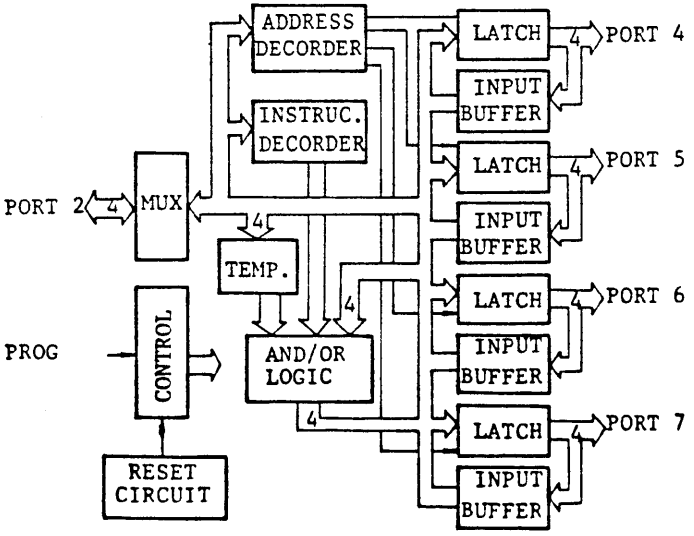
FEATURES

- CMOS LSI for low power dissipation
- Low cost
- Simple interface to TLCS-84C microcomputers
- Four 4-bit I/O ports
- AND and OR directly to ports
- Single 5V supply
- High output drive
- Direct extension of resident TMP80C49P-6 I/O ports.
- PIN compatible with intel's 8243
- Extended operation temperature range -40°C to 85°C

PIN CONNECTION (TOP VIEW)



BLOCK DIAGRAM



PIN NAMES AND PIN DESCRIPTION

PROG (Input)

Clock input. A high to low transistion on PROG signifies that address and control are available on P20-23, and a low to high transition signifies that data is available on P20-23.

$\overline{CS}$ (Input)

Chip Select Input. A high on $\overline{CS}$ inhibits any change of output or internal status.

P20-23 (Input/Output, 3-state)

Four (4) bit bi-directional port contains the address and control bits on a high to low transition of PROG. During a low to high transition contains the data for a selected output port if a write operation, or the data from a selected port before the low to high transition if a read operation.

P40-43, P50-53, P60-63, P70-73 (Input/Output, 3-state)

Four (4) bit bi-directional I/O ports. May be programmed to be input (during read), low impedance latched output (after write) or a 3-state (after read). Data on pins P20-23 may be directly written, ANDed or ORed with previous data.

V_{CC} (Power)

+5 volt supply

GND (Power)

0 volt supply

FUNCTIONAL DESCRIPTION

General Operation

The TMP82C43P contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports 4-7. The following operations may be performed on these ports.

- Transfer accumulator to port
- Transfer port to accumulator
- AND accumulator to port
- OR accumulator to port

All communication between the microcomputer (TMP80C49P-6) and the TMP82C43P occurs over Port 2 (P20-23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles.

A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional TMP82C43P's may be added to the 4-bit bus and chip selected using additional output lines from the microcomputer.

Power On Initialization

Initial application of power to the device forces input/output ports 4, 5, 6, and 7 to the tri-state and port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi, A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi, A takes new data AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are 3-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the 3-stated mode while port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read·mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the TMP82C43P output. A read of any port will leave that port in a high impedance state.

TMP8243P
ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage with Respect to GND	-0.5V to +7.0V
V_{IN}	Input Voltage with Respect to GND	-0.5V to V_{CC}+0.5V
V_{OUT}	Output Voltage with Respect to GND	-0.5V to V_{CC}+0.5V
P_D	Power Dissipation	250mW
T_{SOLDER}	Soldering Temperature (soldering Time 10 sec.)	260°C
T_{STG}	Storage Temperature	-65°C to +150°C
T_{OPR}	Operating Temperature	-40°C to +85°C

D.C. CHARACTERISTICS (I) T_{OPR}=-40°C ~ 85°C, V_{CC}=5V±10%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNITS
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.2			V
V_{OL1}	Output Low Voltage Ports 4-7	I_{OL}=5mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL}=20mA			1.0	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.8mA			0.45	V
V_{OH11}	Output High Voltage Ports 4-7	I_{OH}=-1.2mA	2.4			V
V_{OH21}	Output High Voltage Port 2	I_{OH}=-0.6mA	2.4			V
V_{OH12}	Output High Voltage Ports 4-7	I_{OH}=-0.6mA	V_{CC}-0.8			V
V_{OH22}	Output High Voltage Port 2	I_{OH}=-0.3mA	V_{CC}-0.8			V
I_{IL1}	Input Leakage Port 4-7	$V_{SS} \leq V_{IN} \leq V_{CC}$			±10	μA
I_{IL2}	Input Leakage Port 2, $\overline{CS}$,PROG	$V_{SS} \leq V_{IN} \leq V_{CC}$			±10	μA
I_{CC1}	Power Supply Current (1)	V_{CC}=5V, V_{IL}=0.2V V_{IH}=V_{CC}-0.2V PROG PERIOD=5μS			2	mA
I_{CC2}	Power Supply Current (2)	V_{CC}=5V, V_{IL}=0.2V V_{IH}=V_{CC}-0.2V PROG=V_{CC}-0.2V			10	μA
I_{OL}	Sum of all I_{OL} of 16 Outputs	5mA Each pin			80	mA

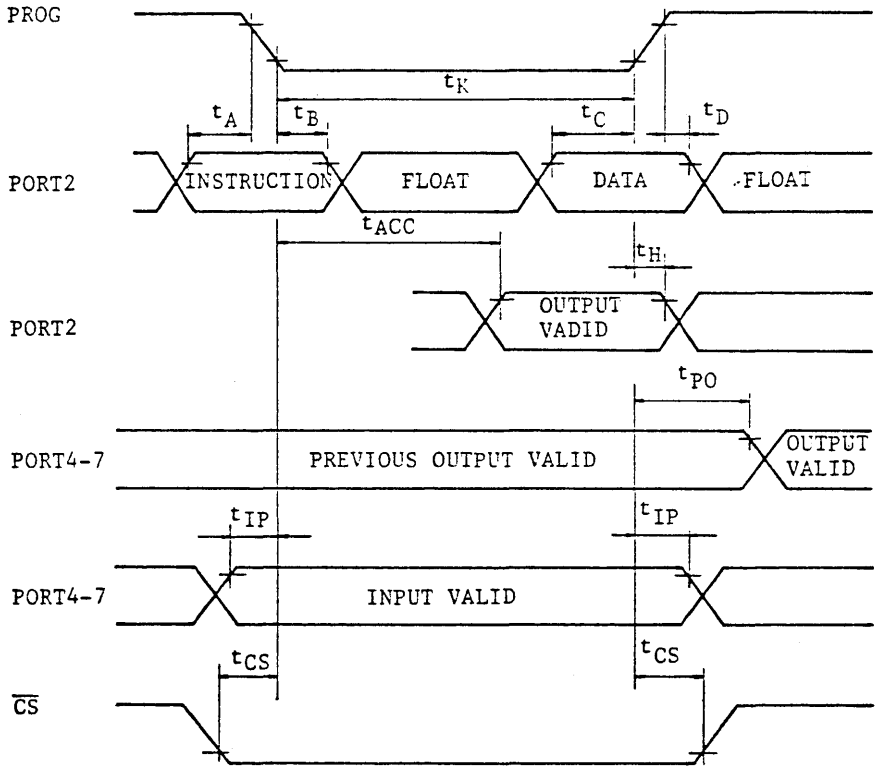
.C. CHARACTERISTICS (Ⅱ) T_{OPR}=-40°C to 85°C, V_{CC}=5V±20%, V_{SS}=0V

YMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNITS
V_{IL}	Input Low Voltage	4.0V≤V_{CC}≤4.5V	-0.5		0.15V_{CC}	V
V_{IH}	Input High Voltage	5.5V≤V_{CC}≤6.0V	0.5V_{CC}		V_{CC}	V
V_{OL1}	Output Low Voltage Ports 4-7	I_{OL}=4mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL}=15mA			1.0	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.6mA			0.45	V
V_{OH12}	Output High Voltage Ports 4-7	I_{OH}=-200μA	V_{CC}-0.8			V
V_{OH22}	Output High Voltage Port 2	I_{OH}=-100μA	V_{CC}-0.8			V
I_{OL}	Sum of all I_{OL} of 16 outputs	4mA Each Pin			64	mA

.C. CHARACTERISTICS T_{OPR}=-40°C to 80°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNITS
t_A	Code Valid Before PROG	C_L=80pF	100			ns
t_B	Code Valid After PROG	C_L=20pF	60			ns
t_C	Data Valid Before PROG	C_L=80pF	200			ns
t_D	Data Valid After PROG	C_L=20pF	20			ns
t_H	Floating After PROG	C_L=20pF	0		150	ns
t_K	PROG Negative Pulse Width		700			ns
t_{CS}	$\overline{CS}$ Valid Before/After PROG		50			ns
t_{PO}	Ports 4-7 Valid After PROG	C_L=100pF			700	ns
t_{IP}	Ports 4-7 Valid Before/After PROG		100			ns
t_{ACC}	Port 2 Valid After PROG	C_L=80pF			650	ns

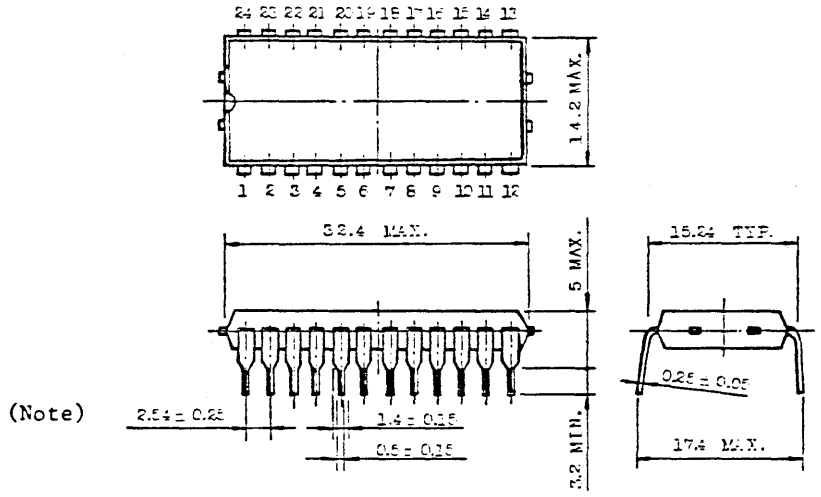
TIMING WAVEFORM



OUTLINE DRAWINGS

PLASTIC PACKAGE

Unit in mm



(Note)

Note : Each lead pitch is 2.54mm. All leads are located within 0.25mm of their true longitudinal position with respect to No.1 and No.24 leads.

PART 2

TLCS-90

LSI DEVICES

CONTENTS

CMOS 8-BIT MICROCONTROLLERS
TMP90C840N/TMP90C841N
TMP90C840F

1. OUTLINE AND CHARACTERISTICS

The TMP90C840 is a high-speed advanced 8-bit micro controller applicable to a variety of equipment.
With its 8-bit CPU, ROM, RAM, A/D converter, multi-function timer/event counter and general-purpose serial interface integrated into a single CMOS chip, the TMP90C840 allows the expansion of external memories for programs (up to 56K bytes) and data (1M bytes).

The characteristics of the TMP90C840 include:

(1) Efficient instruction system
  163 basic instructions
  Multiplication, division, 16-bit arithmetic operations, bit manipulation instructions
(2) Minimum instruction executing time: 400 ns (at 10 MHz oscillation frequency)
(3) Internal ROM: 8K bytes
(4) Internal RAM: 256 bytes
(5) Memory expansion
  External program memory: 56K bytes
  External data memory: 1M bytes
(6) Super-precision 8-bit A/D converter (6 channels)
(7) General-purpose serial interface (1 channel)
  Asynchronous mode, I/O interface mode
(8) Multi-function 16-bit timer/event counter
(9) Four 8-bit timers
(10) 2-channel stepping motor control port
(11) Input/Output ports (54 pins)
(12) Interrupt function: 10 internal interrupts and 4 external interrupts
(13) Micro Direct Memory Access (DMA) function (11 channels)
(14) Watchdog timer
(15) Standby function (4 HALT modes)
(16) Complementary metal oxide semiconductor (CMOS)
(17) Single power source

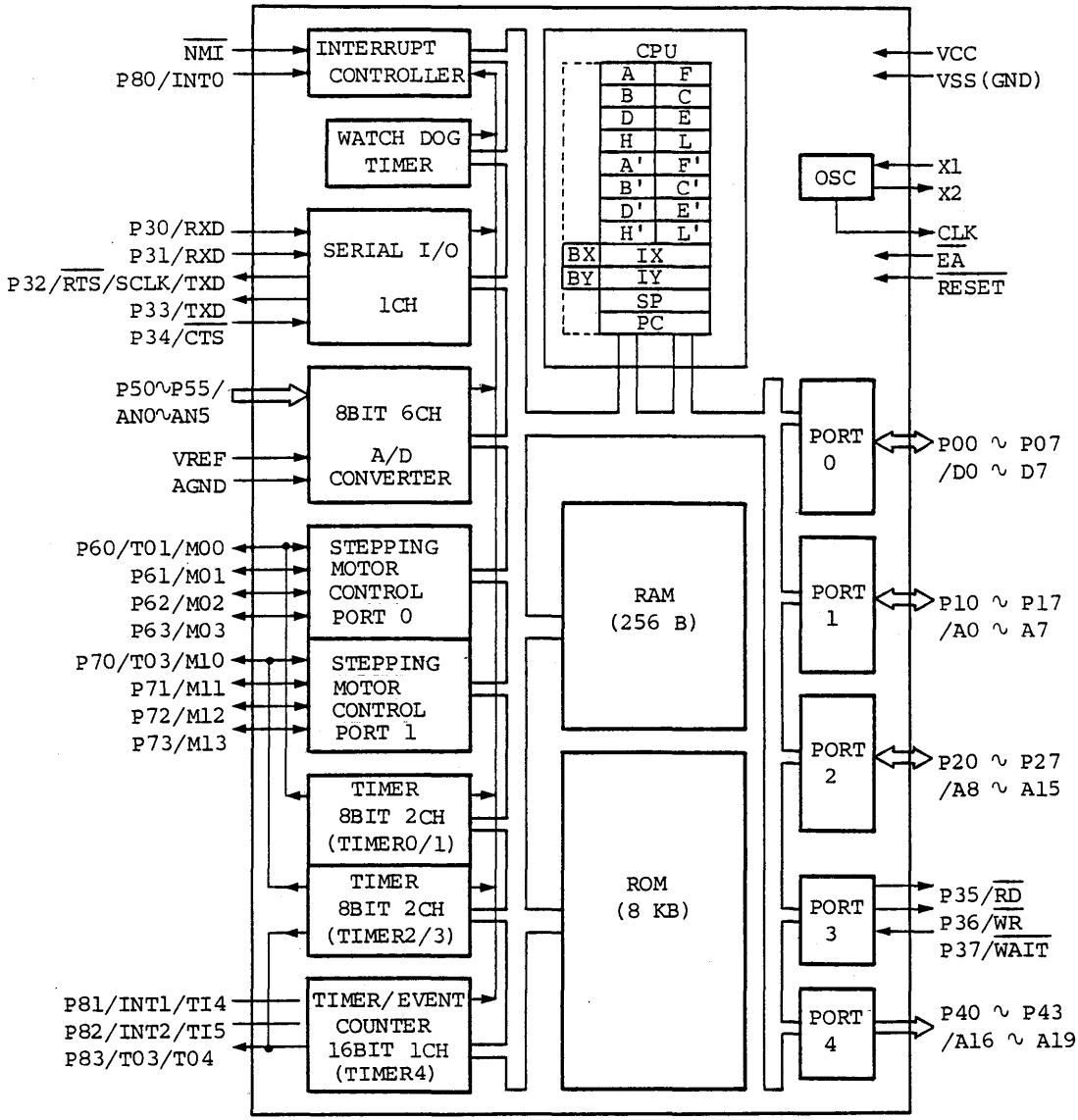


Fig. 1 TMP90C840 Block Diagram

2. PIN ARRANGEMENT AND FUNCTIONS

The arrangement of input/output pins, their names and functions are described below.

2.1 Pin Arrangement

Fig. 2.1 shows where the input/output pins are located in the TMP90C840.

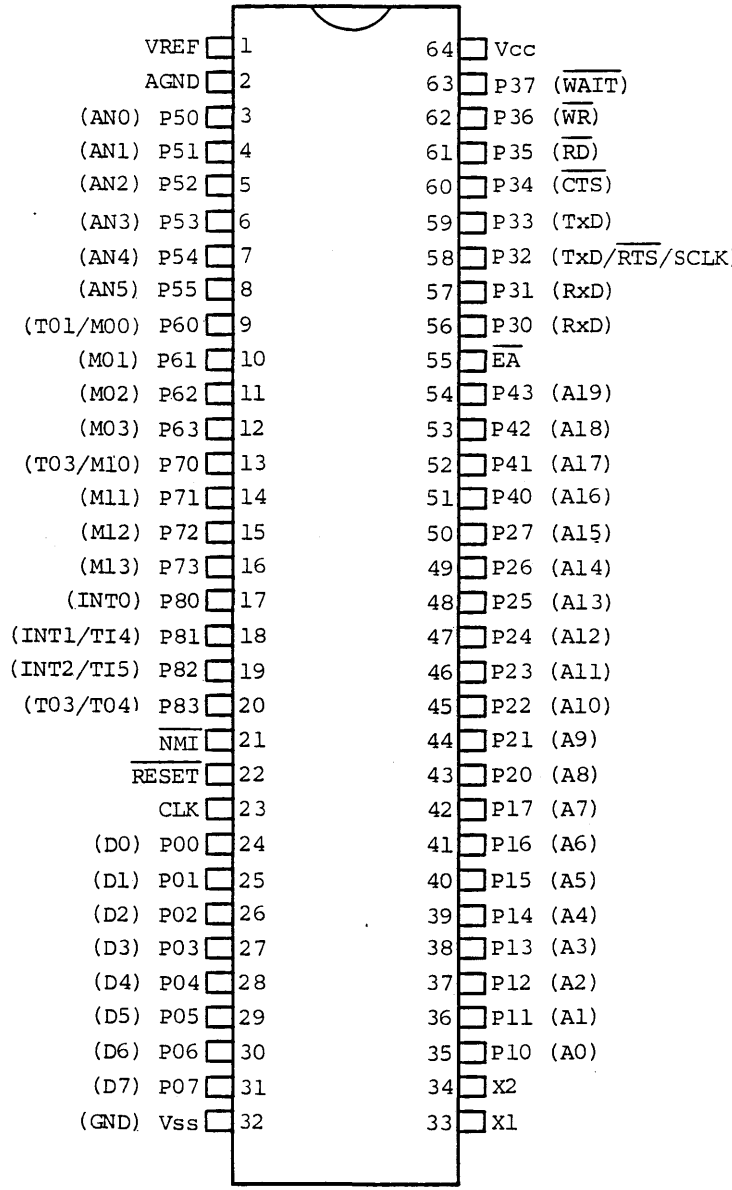


Fig. 2.1-(1) Pin Arrangement (Shrink Dual Inline Package)

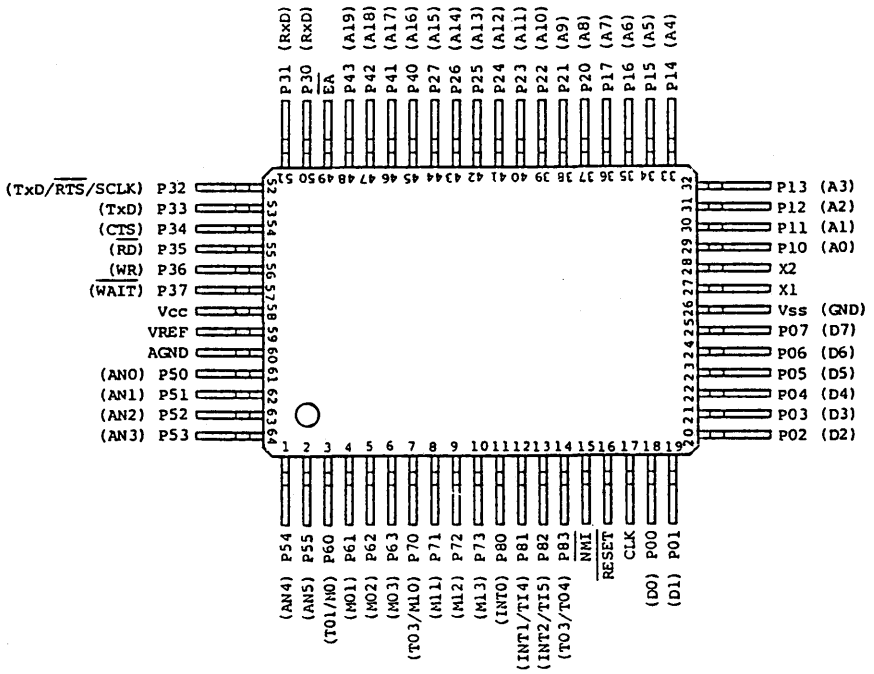


Fig. 2.1-(2) Pin Arrangement (Flat Package)

2.2 Pin Names and Functions

The names of input/output pins and their functions are summarized in Table 2.2.

Table 2.2 Pin Names and Functions

Pin Name	No. of pins	I/O 3 states	Function
P00 - P07 /D0 - D7	8	I/O 3 states	Port 0: 8-bit I/O port that allows selection of input/output on byte basis Data bus: Also functions as 8-bit bidirectional data bus for external memory
P10 - P17 /A0 - A7	8	I/O /Output	Port 1: 8-bit I/O port that allows selection on byte basis Address bus: The lower 8 bits function as address bus for external memory
P20 - P27 /A8 - A15	8	I/O /Output	Port 2: 8-bit I/O port that allows selection on bit basis Address bus: The upper 8 bits function as address bus for external memory
P30 /RxD	1	Input	Port 30: 1-bit input port Serial data receiving
P31 /RxD	1	Input	Port 31: 1-bit input port Serial data receiving
P32 /TxD /PTS /SCLK	1	Output	Port 32: 1-bit output port Serial data transmission Request serial data transmission Serial clock output
P33 /TxD	1	Output	Port 33: 1-bit output port Serial data transmission
P34 /CTS	1	Input	Port 34: 1-bit input port Capable of serial data transmission
P35 /RD	1	Output	Port 35: 1-bit output port Read: Generates strobe signal for reading external memory
P36 /WR	1	Output	Port 36: 1-bit output port Write: Generates strobe signal for writing into external memory
P37 /WAIT	1	Input	Port 37: 1-bit input port Wait: Input pin for connecting slow access memory or peripheral LSI

Pin Name	No. of pins	I/O 3 states	Function
P40 - P43 /A16 - A19	4	Output	Port 4: 4-bit output port that allows selection of port/address bus on bit basis Address bus: Also functions as address bus for external memory (4 bits of bank address)
P50 - P55 /AN0 - AN5	6	Input	Port 5: 6-bit input port Analog input: 6 points for analog input to A/D converter
VREF	1		Input of reference voltage to A/D converter
AGND	1		Ground pin for A/D converter
P60 - P63 /M00 - M03	4	I/O	Port 6: 4-bit I/O port that allows I/O selection on bit basis
/T01		/Output	Stepping motor control port 0
		/Output	Timer output 1: Output of Timer 0 or 1
P70 - P73 /M10 - M13	4	I/O	Port 7: 4-bit I/O port that allows I/O selection on bit basis
/T03		/Output	Stepping motor control port 1
		/Output	Timer output 3: Output of Timer 2 or 3
P80 /INT0	1	Input	Port 80: 1-bit input port Interrupt request pin 0: interrupt request pin (Level/rising edge is programmable)
P81 /INT1 /T14	1	Input	Port 81: 1-bit input port Interrupt request pin 1: interrupt request pin (Level/falling edge is programmable) Timer input 4: Counter/capture trigger signal for Timer 4
P82 /INT2 /T15	1	Input	Port 82: 1-bit input port Interrupt request pin 2: rising edge interrupt request pin Timer input 5: capture trigger signal for Timer 4
P83 /T03/T04	1	Output	Port 83: 1-bit output port Timer output 3/4: Output of Timer 3/4
NMI	1	Input	Non-maskable interrupt request pin: Falling edge interrupt request pin

Pin Name	No. of pins	I/O 3 states	Function
CLK	1	Output	Clock output: Generates clock pulse at 1/4 frequency of clock oscillation. It is a high-level while resetting.
EA	1	Input	External access: Connects with Vcc pin in the TMP90C840 using internal ROM, and with GND pin in the TMP90C841 with no internal ROM.
RESET	1	Input	Reset
X1/X2	2	Input/ Output	Pin for quartz crystal oscillator
Vcc	1		Power supply pin (+5V)
Vss(GND)	1		Ground pin (0V)

3. OPERATION

This chapter describes the functions and the basic operations of the TMP90C840 in every block.

3.1 CPU

The TMP90C840 incorporates a high-performance 8-bit CPU. This CPU improves its speed of processing, addressing and executing instructions compared to the conventional 8-bit versions.
This section describes the CPU functions available to the programmer.

3.1.1 Memory map

The TMP90C840 supports a program memory of up to 64K bytes and a data memory of maximum 1M bytes.
The program memory may be assigned to the address space from 00000H to 0FFFFH, while the data memory can be allocated to any address from 00000H to FFFFFH.

(1) Internal ROM
The TMP90C840 internally contains an 8K-byte ROM. The address space from 0000H to 1FFFH is provided to the ROM. The CPU starts executing a program from 0000H by resetting.
The addresses 0010H to 007FH in this internal ROM area are used for the entry area for the interrupt processing.

(2) Internal RAM
The TMP90C840 also contains a 256 byte RAM, which is allocated to the address space from FEC0H to FFBFH. The CPU allows the access to a certain RAM area (FF00H to FFBFH, 192 bytes) by a short operation code (opcode) in a "direct addressing mode".
The addresses from FF10H to FF7FH in this RAM area can be used as parameter area for micro DMA processing (and for any other purposes when the micro DMA function is not used).

(3) Internal I/O
The TMP90C840 provides a 48-byte address space as an internal I/O area, whose addresses range from FFC0H to FFEFH. This I/O area can be accessed by the CPU using a short opcode in the "direct addressing mode".

Fig. 3.1 (1) is a memory map indicating the areas accessible by the CPU in the respective addressing mode.

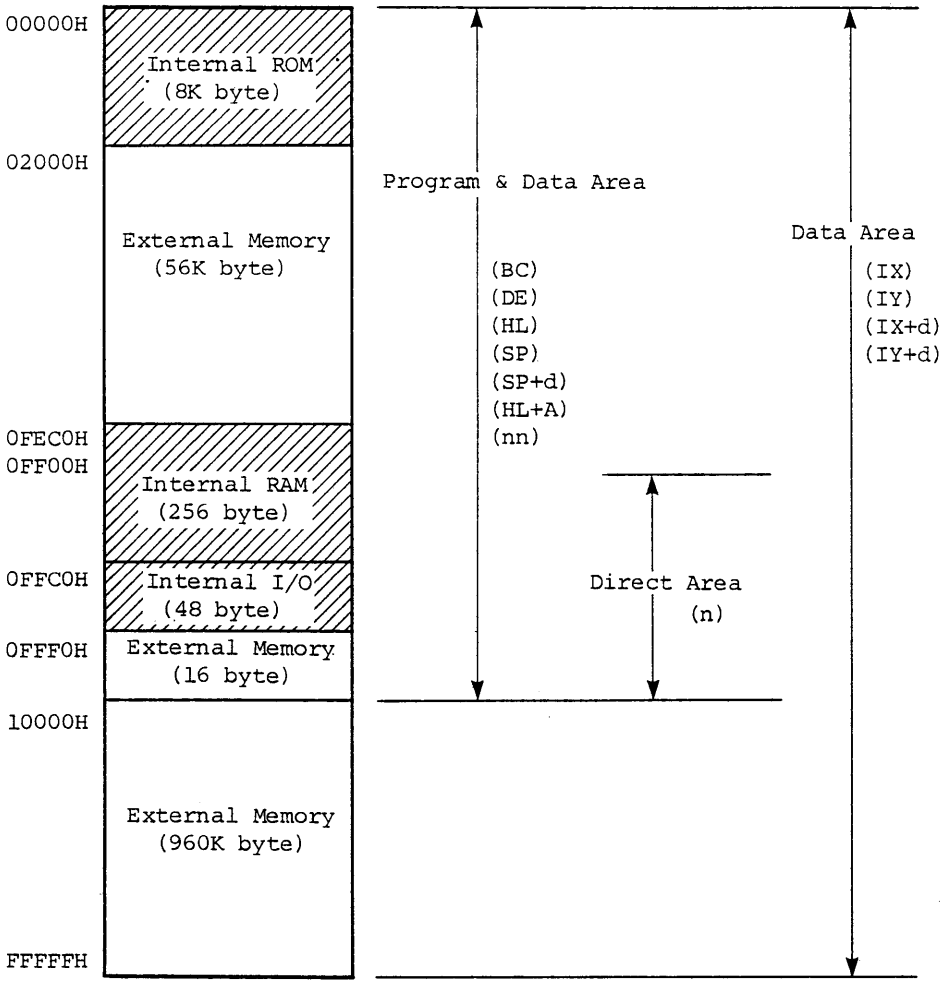


Fig. 3.1 (1) Memory Map

3.1.2 Registers

Fig. 3.1 (2) shows the configuration of registers.

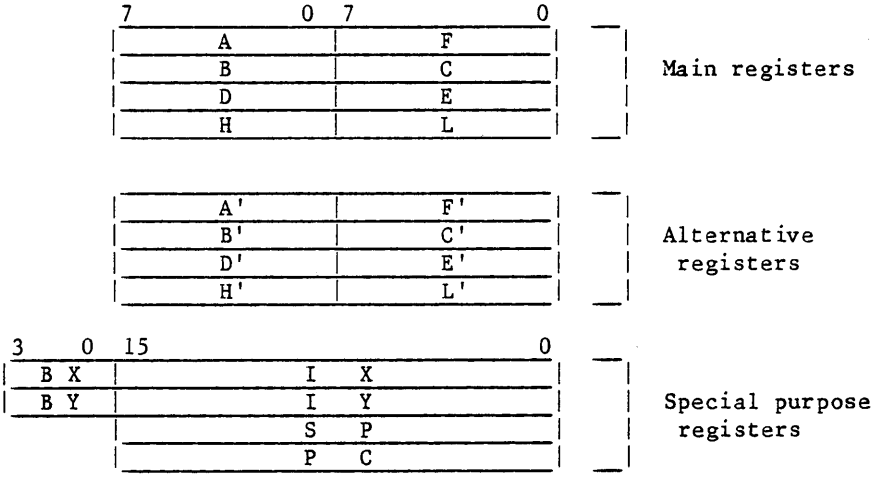


Fig. 3.1 (2) Configuration of Registers

The TMP90C840 uses main registers, alternative registers and dedicated registers. The main registers and the alternative registers are allowed to be exchanged of their contents by a register exchange instruction.

(1) Register A
This is an 8-bit register used mainly for 8-bit arithmetic and logic operations.

(2) Register F
This is an 8-bit register that stores the status of operation results. Configuration of register F is shown in Fig. 3.1 (3).

	7	6	5	4	3	2	1	0	
MSB	S	Z	IFF	H	X	P/V	N	C	LSB

Fig. 3.1 (3) Configuration of Register F

- o Sign flag (S)
  The sign flag is set to "1" when the arithmetic result is negative. It stores the contents of the most significant bit (MSB) of the arithmetic and logic unit (ALU).
- o Zero flag (Z)
  Z flag is set to "1" when the all bits of the ALU after operation are "0".

- o Parity/Overflow flag (P/V)
  This flag has two functions. One is to indicate the parity (P) resulted from a logical opepration (AND, OR, or XOR). It is set to "1" when the result is even, and "0" for odd parity.
  The other is to indicates the overflow (V) in an arithmetic operation (ADD, ADC, SUB, SBC, or CP). The flag is set to "1" when the result cannot be expressed by a signed integral number.
  The P/V flag selects either function according to the instruction.
- o Carry flag (C)
  The flag is set to "1" if a carry or borrow has occurred on the MSB of the ALU.
- o Expansion carry flag (X)
  Like the carry flag (C), it is set to "1" when the MSB of the ALU involves a carry or borrow as a result of an operation except that it applies to a wider range of instructions (e.g., INC rr).
- o Half carry flag (H)
  It is set to "1" when a carry or borrow has occurred on the 4th bit of the lower side in the ALU.
- o Addition/Subtraction flag (N)
  This flag is set to "1" if the executed operation is a subtraction (SUB, SBC, CP, or DEC).
- o Interrupt enable flag (IFF)
  A maskable interrupt is enabled or disabled by this flag. This flag is set to "1" by an EI instruction and "0" by an DI instruction.

(Note) This flag is shared with the alternative register F'.

(3) Registers B, C, D, E, H and L
All these registers have an 8-bit configuration. They function as 16-bit register pairs (concatenated BC, DE and HL) as well as independent 8-bit register. Registers B or register pair BC is also used as a counter for the loop instruction (DJNZ). Register pair HL is used for 16-bit data processing including 16-bit arithmetic/logic operations.

(4) Registers A', F', B', C', D', E', H' and L'
These registers have the same structure as the main registers (A, F, B, C, D, E, H and L). They are called alternative registers. There is no instruction that directly accesses these alternative registers, but its data can be processed by a register exchange instruction.
Following are examples of register exchange insturctions that allow the exchange of data between a main register and an alternative register:

```
EX      AF,AF'
EXX
```

(5) Registers IX, IY, BX and BY
IX and IY are 16-bit independent registers called index registers. BX and BY are 4-bit independent registers and referred to as bank registers.
These registers are used mainly for specifying memory addresses, and generate 20-bit addresses. IX and IY registers are also used for 16-bit additions.

BX and BY registers are allocated to the memory addresses FFECH (BX register) and FFEDH (BY register) in the internal I/O address spaces. Only their lower four bits are effective, with the upper four bits being undefined. These undefined bits are always set to "1" when reat out. By resetting, the lower four bits of BX and BY registers are initialized to "0".

BX (FFECH)	-	-	-	-	BX3	BX2	BX1	BX0	R/W
BY (FFECH)	-	-	-	-	BY3	BY2	BY1	BY0	R/W

(6) SP register

SP register is a 16-bit register called a stack pointer (SP), that stores the start address of the memory stack area (Last in, first out basis). It is decremented when a CALL or PUSH instruction is executed or an interrupt is accepted. It is incremented by execution of RET instruction or a POP instruction.

(7) PC register

This is a 16-bit register called a program counter, and stores the memory address of the next instruction to be executed.
It is initialized to 0000H when the RESET pin becomes low.

(8) Other

By executing the data exchange instruction [EXX] between a main register and an alternative register, the EXF bit (exchange flag: Bit 1 of memory address FFD2H) of the internal I/O register is inverted. This is a read-only bit, and is not initialized by resetting.

3.1.3 Addressing modes

Eight addressing modes are available for the TMP90C840. They are used in combination with various instructions to enhance the CPU's processing capabilities.
They are: Register mode, immediate mode, register indirect mode, index mode, register index mode, extend mode, direct mode and relative mode. The first seven addressing modes are used most frequently. The relative addressing mode is only applicable to specific instructions.

(1) Register addressing mode

In the register addressing mode, the operand represents a specified register.

```
Example:    LD  A, B
```

The contents of Register B are loaded into Register A.

(2) Immediate addressing mode

In this mode, the operand is in the instruction.

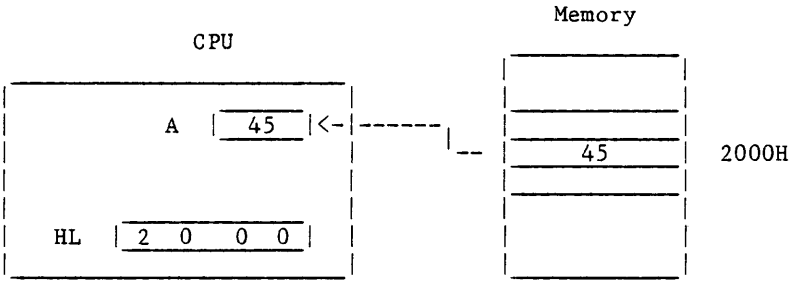
```
Example:    LD  A, 12H
```

Immediate data "12H" are loaded into Register A.

(3) Register indirect addressing mode

In the register indirect addressing mode, the operand is located in a memory address indicated by a register pair (BC, DE, HL, IX, IY or SP).

Example: LD A, (HL)

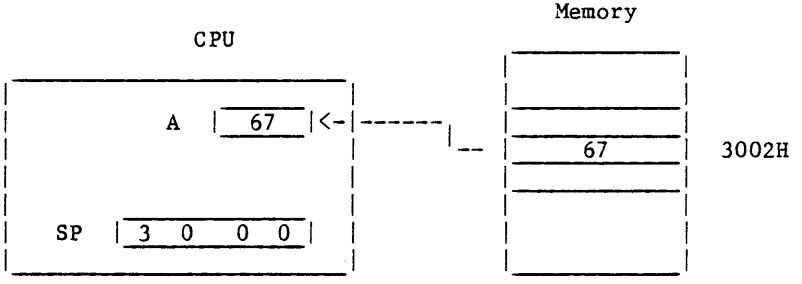


"45H" in the memory address 2000H is loaded into Register A.

(4) Index addressing mode

In the index addressing mode, the operand is located in a memory address specified by adding an 8-bit displacement value in the opcode to the contents of a specified register pair (IX, IY or SP).

Example: LD A, (SP+2)

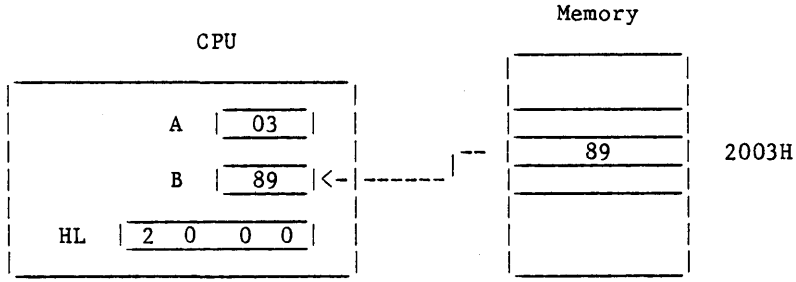


"67H" in the memory address 3002H is loaded into Register A. The displacement value ranges from -128 to +127.

(5) Register index addressing mode

In this mode, the operand is located in a memory address specified by adding the displacement value of Register A to the contents of register pair HL.

Example: LD B, (HL+A)

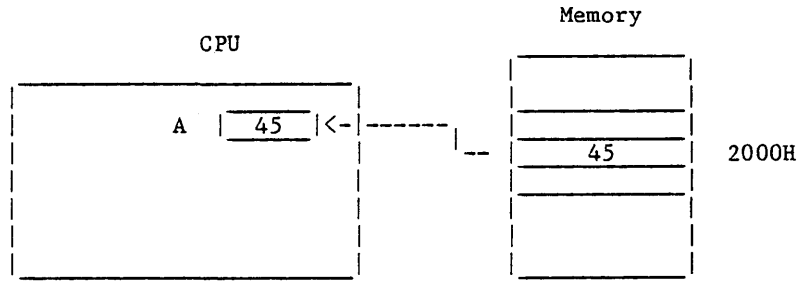


"89H" in the memory address 2003H is loaded into Register B. In this mode, the data in Register A are considered as 8-bit signed number, and the displacement value ranges from -128 to +127.

(6) Extended addressing mode

In this mode, the operand is accessed by 2-byte (16-bit) data in the opcode.

Example: LD A, (2000H)



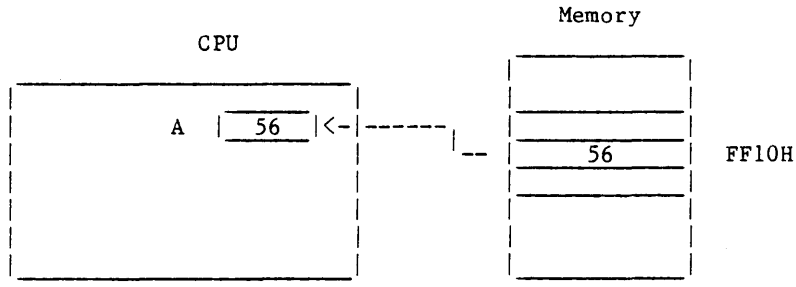
"45H" in the memory address 2000H is loaded into Register A.

(7) Direct addressing mode

The operand in this mode is located in a memory address from FF00H to FFFFH specified by 1-byte (8 bits) data in the opcode. Compared with the extended addressing mode, it saves both program memory and executing time. This mode allows the access to 256-byte addresses from FF00H to FFFFH.

For the TMP90C840, this direct area is divided into the internal RAM (192 bytes from FF00H to FFBFH) and the internal I/O area (48 bytes from FFC0H to FFEFH).

Example: LD A, (FF10H)

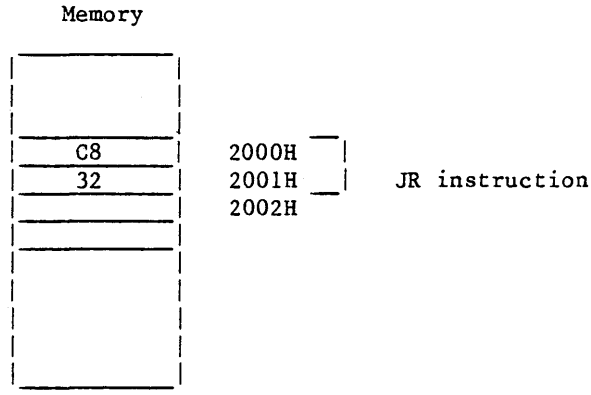


"56H" in the memory address FF10H is loaded into Register A.

(8) Relative addressing mode

In the relative addressing mode, the operand is found at the address relative to the current instruction. This mode is applicable to instructions involving an 8-bit displacement value (JR and DJNZ) and a 16-bit displacement value (LDAR, JRL and CALR).

Example: JR 2034H



In this example, the program execution jumps to the address 2034H. Since the program counter is already incremented by 2 at the time of address computation, the displacement is obtained by the following formula based on the "memory address of the JR instruction + 2":

Destination address - (address of instruction being executed + 2)

In the example, the displacement "32H" is obtained by:

2034H - (2000H + 2)

In any other instructions using the relative addressing mode (DJNZ, LDAR, JRL and CALR), the displacement is always calculated based on the "address of the current instruction + 2".

(9) Addressing modes for extended data area

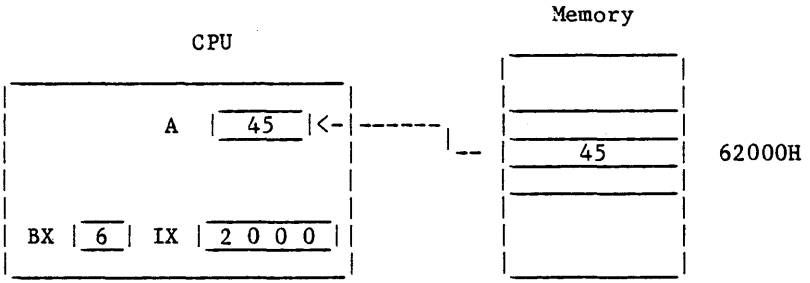
The TMP90C840 provides up to 1M bytes of data.
The addresses 00000H to 0FFFFH can be accessed in a normal addressing mode.
However, the addresses from 10000H to FFFFFH called an "extended data area" require a special addressing mode for access.
Accessing the extended data area requires to select the addressing mode which uses the index register IX or IY for obtaining the address of the operand (the register indirect addressing mode or the index addressing mode).

The following four special modes are available:

(IX)
(IY)
(IX+d)
(IY+d)

In these modes, the extended data area is accessed by using a 20-bit address consisting of a 16-bit offset address (address bus A0 to A15) and a bank address (address bus A16 to A19).
The 16-bit offset address is obtained by the same way as in a normal address computation. The 4-bit bank address is specified by the bank register BX or BY. The register pair BX is selected when the index register IX is used, and BY is selected when IY is used.

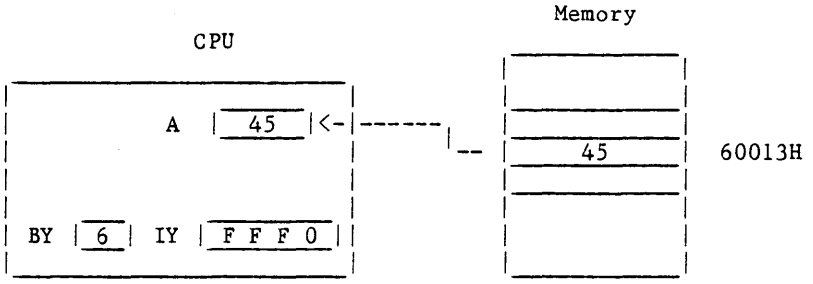
Example: LD A, (IX)



"45H" in the address 62000H is loaded into Register A.

In the index addressing mode, a carry resulted from calculating the 16-bit offset address is ignored; i.e., it is not added to the bank address.

Example: LD A, (IY+23H)



In this example, "45H" in the address 60013H is loaded into Register A.
In any other addressing mode that accesses a non-extended data area (the index register IX or IY is not used for address computing of the operand), the 4-bit bank address (address bus A16 to A19) becomes "0", indicating that the access range is from 00000H to 0FFFFH.

(Note) Given "FFECH" to the IX value of (BX address) an instruction "LD (IX), x", the normal write cycle is not performed, making the result indefinite.

3.1.4 Instructions

The TMP90C840 supports a rich variety of addressing modes as well as powerful instruction sets. There are 163 basic instructions as categorized into the following nine groups:

- o 8-bit transfer instruciton
- o 16-bit transfer instruciton
- o Exchange, block transfer and search instructions
- o 8-bit arithmetic and logical operation instruction
- o Special operation and CPU control instructions
- o 16-bit arithmetic and logical operation instruciton
- o Rotate and shift instructions
- o Bit manipulation instruction
- o Jump, call and return instruction

Table 3.1 (1) lists the 163 basic instructions. Table 3.1. (2) describes the mnemonics and their meaning.

(1) 8-bit transfer instruction
The 8-bit transfer instructions inclued those for transferring 8-bit data between registers, register and immediate address, register and memory, or memory and immediate address.

(2) 16-bit transfer instruction

The 16-bit transfer instructions include those for transferring 16-bit data between registers, register and immediate address, register and memory, and memory and immediate address, PUSH and POP instructions using the stack, and LDA (Load Address) instruction that calculates an effective address and loads its value into a register.

(3) Exchange, block transfer and search instructions

The data exchange instructions are executed to exchange 16-bit data between registers, between memory and register, or between a main register and an alternative register.

The block transfer instructions can transfer data in any memory block to other memory area.

The block search instructions are executed to find out a particular 8-bit character in a given memory block. LDIR, LDDR, CPIR and CPDR included in the block transfer and search instructions read the current instruction each time a 1-byte memory is transferred or compared, thus making it possible to acknowledge an interrupt before reaching to the end of the block.

(4) 8-bit arithmetic and logical operation instruction

8-bit arithmetic and logical operation instructions perform 8-bit arithmetic and logical operations between Register A and another register, Register A and immediate address, Register A and memory, register and immediate address, and memory and immediate address (ADD, ADC, SUB, SBC, AND, OR XOR and CP), or increment/decrement the contents of register or memory by 1 (INC, DEC, INCX and DECX).

The INCX (Increment if X) instruction increments the contents of a memory specified by the operand if the X flag is "1", and does nothing if not. The DECX instruction performs the same operation except that it decrements the data. These instructions are executed to increment or decrement the data with 20-bit width in the 20-bit address pointers (registers BX and IX or BY and IY) mainly to access an extended data area.

Examples:

```
INC IX          :   Increment Register IX
INCX (FFECH)    :   Increment Register BX if X
LD A,(IX)       :   Load contents of memory into Register A
```

Table 3.1 (1) TMP90C840 Basic Instructions (163 types)

LD r,r	SBC r,n	MUL HL,r	RRCA
LD r,n	SBC mem,n	MUL HL,n	RRC r
LD r,mem	AND A,r	MUL HL,mem	RRC mem
LD mem,r	AND A,n	DIV HL,r	RLA
LD mem,n	AND A,mem	DIV HL,n	RL r
LD rr,rr	AND r,n	DIV HL,mem	RL mem
LD rr,nn	AND mem,n	ADD HL,rr	RRA
LD rr,mem	OR A,r	ADD HL,nn	RR r
LD mem,rr	OR A,n	ADD HL,mem	RR mem
LDW mem,nn	OR A,mem	ADC HL,rr	SLAA
PUSH qq	OR r,n	ADC HL,nn	SLA r
POP qq	OR mem,n	ADC HL,mem	SLA mem
LDA rr,mem	XOR A,r	SUB HL,rr	SRAA
EX DE,HL	XOR A,n	SUB HL,nn	SRA r
EX AF,AF'	XOR A,mem	SUB HL,mem	SRA mem
EXX	XOR r,n	SBC HL,rr	SLLA
EX mem,rr	XOR mem,n	SBC HL,nn	SLL r
LDI	CP A,r	SBC HL,mem	SLL mem
LDIR	CP A,n	AND HL,rr	SRLA
LDD	CP A,mem	AND HL,nn	SRL r
LDDR	CP r,n	AND HL,mem	SRL mem
CPI	CP mem,n	OR HL,rr	RLD mem
CPIR	INC r	OR HL,nn	RRD mem
CPD	INC mem	OR HL,mem	BIT b,r
CPDR	DEC r	XOR HL,rr	BIT b,mem
ADD A,r	DEC mem	XOR HL,nn	RES b,r
ADD A,mem	INCX (n)	XOR HL,mem	RES b,mem
ADD r,n	DECX (n)	CP HL,rr	SET/TSET b,r
ADD mem,n	DAA A	CP HL,nn	SET/TSET b,mem
ADC A,r	CPL A	CP HL,mem	JP cc,mem
ADC A,mem	NEG A	ADD ix,rr	JR cc,PC+d
ADC r,n	LDAR HL,PC+dd	ADD ix,nn	JRL PC+dd
ADC mem,n	CCF	ADD ix,mem	CALL cc,mem
SUB A,r	SCF	INC rr	CALR PC+dd
SUB A,mem	RCF	INCW mem	DJNZ [BC,]PC+d
SUB r,n	NOP	DEC rr	RET cc
SUB mem,n	HALT	DECW mem	RETI
SBC A,r	DI	RLCA	
SBC A,n	EI	RLC r	
SBC A,mem	SWI	RLC mem	

Table 3.1 (2) TMP90C840 Mnemonics and Their Meaning

Mnemonic	Meaning	Mnemonic	Meaning
LD	Load	MUL	Multiply
LDW	Load Word	DIV	Divide
PUSH	Push	INCW	Increment Word
POP	Pop	DECW	Decrement Word
LDA	Load Address	RLCA	Rotate Left Circular Accumulator
EX	Exchange		
EXX	Exchange X	RLC	Rotate Left Circular
LDI	Load and Increment	RRCA	Rotate Right Circular Accumulator
LDIR	Load, Increment and Repeat		
LDD	Load and Decrement	RRC	Rotate Right Circular
LDDR	Load, Decrement and Repeat	RLA	Rotate Left Accumulator
CPI	Compare and Increment	RL	Rotate Left
CPIR	Compare, Increment and Repeat	RRA	Rotate Right Accumulator
		RR	Rotate Right
CPD	Compare and Decrement	SLAA	Shift Left Arithmetic Accumulator
CPDR	Compare, Decrement and Repeat		
		SLA	Shift Left Arithmetic
ADD	Add	SRAA	Shift Right Arithmetic Accumulator
ADC	Add with Carry		
SUB	Subtract	SRA	Shift Right Arithmetic
SBC	Subtract with Carry	SLLA	Shift left Logical Accumulator
AND	And		
OR	Or	SLL	Shift Left Logical
XOR	Exclusive Or	SRLA	Shift Right Logical Accumulator
CP	Compare		
INC	Increment	SRL	Shift Right Logical
DEC	Decrement	RLD	Rotate Left Digit
INCX	Increment if X	RRD	Rotate Right Digit
DECX	Decrement if X	BIT	Bit Test
DAA	Decimal Adjust Accumulator	RES	Reset Bit
CPL	Complement	SET	Set Bit
NEG	Negate	TSET	Test and Set
LDAR	Load Address Relative	JP	Jump
CCF	Complement Carry Flag	JR	Jump Relative
SCF	Set Carry Flag	JRL	Jump Relative Long
RCF	Reset Carry Flag	CALL	Call
NOP	No Operation	CALR	Call Relative
HALT	Halt	DJNZ	Decrement and Jump if Non Zero
DI	Disable Interrupt		
EI	Enable Interrupt	RET	Return
SWI	Software Interrupt	RETI	Return from Interrupt

(5) Special operation and CPU control instruction
Special operations are used to control Register A (DAA, CPL and NEG), load the data in a relative address into a register (LDAR), control the carry flag (CCF, SCF and RCF), multiply an 8-bit data by an 8-bit data and convert the result into a 16-bit representation (MUL), divide a 16-bit data by an 8-bit divisor and obtain an 8-bit quotient with an 8-bit residual (DIV), and to nothing (NOP).
If a quotient cannot be represented by an 8-bit number (outside the 0-255 range) or if the divisor is 0 (e.g., "divide 1,000 by 0" or "divide 5,000 by 0"), the overflow flag is set to "1".
CPU control instructions are executed to suspend the CPU operation (HALT), enable/disable a maskable interrupt (EI/DI), and to execute a software interrupt (SWI).

(6) 16-bit arithmetic and logical operation instruction
The 16-bit arithmetic and logical operation instructions perform 16-bit arithmetic logical operations between the register pair HL and another register pair, the register pair HL and immediate address, and the register pair HL and memory (ADD, ADC, SUB, SBC, AND, OR, XOR and CP), perform an addition between the index registers IX and IY or Stack Pointer SP and a register pair, or immediate address and memory (ADD), or increment/decrement the contents of a register pair or memory by 1 (INC, INCW, DEC, and DECW).
Note that "ADD HL,rr", "ADD ix,gg", "INC rr" and "DEC rr" result in a different flag status.

(7) Rotate and shift instructions
The rotate and shift instructions use 8-bit data (RLC, RRC, RL, RR, SLA, SRA, SLL and SRL) or binary-coded decimal (BCD) data (RLD and RRD).

(8) Bit manipulation instruction
The bit manipulation instructions perform testing, setting and resetting a particular bit in a register or memory (BIT, SET and RES). A test and reset instruction (TSET) is also available for executing multiple tasks.

(9) Jump, call and return instructions
A jump instruction can be used, in addition to the register indirect, index, register index, and extended addressing modes, in the 8-bit and 16-bit relative addressing modes. Note, however, that the 16-bit relative addressing mode can be used only for an unconditional jump instruction.
A call instruction uses the 16-bit relative addressing mode, as well as the register indirect, index, register index, and extended addressing modes. Again, 16-bit relative addressing mode is only applicable to an unconditional call instruction.
Return instructions contain unconditional return instruction, conditional return instruction and RETI instruction for return back from the interrupt processing.
These instructions pop up the program counter PC and the register pair AF from the stack. 16 condition codes and over 16 mnemonics are used in these instructions, because certain flags have more than one meaning (e.g. Z and EQ, NZ and NE, PE and OV, PO and NOV).

In addition to the above instructions, "DJNZ PC+d" and "DJNZ BC,PC+d" may be used to control program loops.
"DJNZ PC+d" decrements the contents of the Register B (8-bit) each time the instruction is executed, and executes a relative jump until it becomes zero. "DJNZ BC,PC+d" decrements the contents of the register pair BC (16-bit), and executes a relative jump until it becomes zero.
Appendix A lists the TMP90C840 machine instructions. The table includes the instruction groups, mnemonics, codes, functions, flag status and executing time.
The executing time can be calculated using the value in the "T" column, which denotes the number of states. Time for one state is equivalent to a time twice as long as the clock oscillation cycle. For example, if the clock oscillation frequency is 10MHz, the time for one state is 200 ns.
Executing "LD A,r" at the clock frequency of 10MHz requires two states, and thus takes 200ns × 2 = 400ns for the execution.
Appendix B contains code maps. The TMP90C840 supports 1-byte opcode instructions and 2-byte opcode instructions. The 1-byte opcode instruction is formatted as follows:

LD	A,B	Opcode
LD	A,n	Opcode | n
LD	HL,mn	Opcode | n | m
LD	(w),n	Opcode | w | n
LDW	(w),mn	Opcode | w | n | m
JR	PC+d	Opcode | d
CALL	mn	Opcode | n | m

As shown in the code format, a 1-byte opcode instruction has an opcode in the first byte and operand codes in the subsequent bytes. If there are two 1-byte operand codes, the lower operand is placed before the upper operand. If both a source and a destination are included as operand codes, the destination is placed first.
A 2-byte opcode instruction begins with the first opcode, followed by an operand code specified by the first opcode, then the second opcode and its operand code. For example,

LD B,C	1st Op Code	2nd Op Code			
LD B,(n)	1st Op Code	n	2nd Op Code		
LD B,(IX+d)	1st Op Code	d	2nd Op Code		
LD B,(mn)	1st Op Code	n	m	2nd Op Code	
ADD B,n	1st Op Code	2nd Op Code	n		
ADD (VW),n	1st Op Code	w	v	2nd Op Code	n

In a 2-byte opcode instruction, the position of the second opcode is determined by the first opcode. Basically, the first opcode in a 2-byte opcode instruction provides data to select the mode of addressing the operand in the range of E0H to FEH. The first operand code that follows the first opcode serves to specify the memory addressing mode. The second operand code that follows the second opcode specifies the immediate addressing mode. Their roles can be summarized as follows:

1st opcode	1st operand code	2nd opcode	2nd operand code
Specify addressing mode	Specify memory addressing	Specify instruction	Specify immediate addressing

3.2 Basic Timing

Each instruction of the TMP90C840 is executed by combination of read, write and dummy cycles. These are basic cycles that synchronize with the system clock. 1/2 of the frequency of the clock oscillation is used as the system clock; e.g., if the clock frequency is 10 MHz, the frequency of the system clock is 5 MHz. The system clock cycle is also called a "state".
The TMP90C840 bus operation are basically synchronous, and each of memory read, memory write and dummy cycles is completed in two states, unless they are not requested to wait.
The "CLK" pin generates a pulse at a frequency that further halves the frequency of the system clock. This CLK signal synchronizes with the bus cycles with no wait request.

3.2.1 Read/Write cycles

Fig. 3.2.(1) is a timing chart of external memory read/write cycles. The left side shows the bus operation timing with no wait request, and on the right side shows that with a 2-state wait request.
Each wait consists of a multiple of two states, making a bus cycle wait for two, four, six, eight states, etc. ...

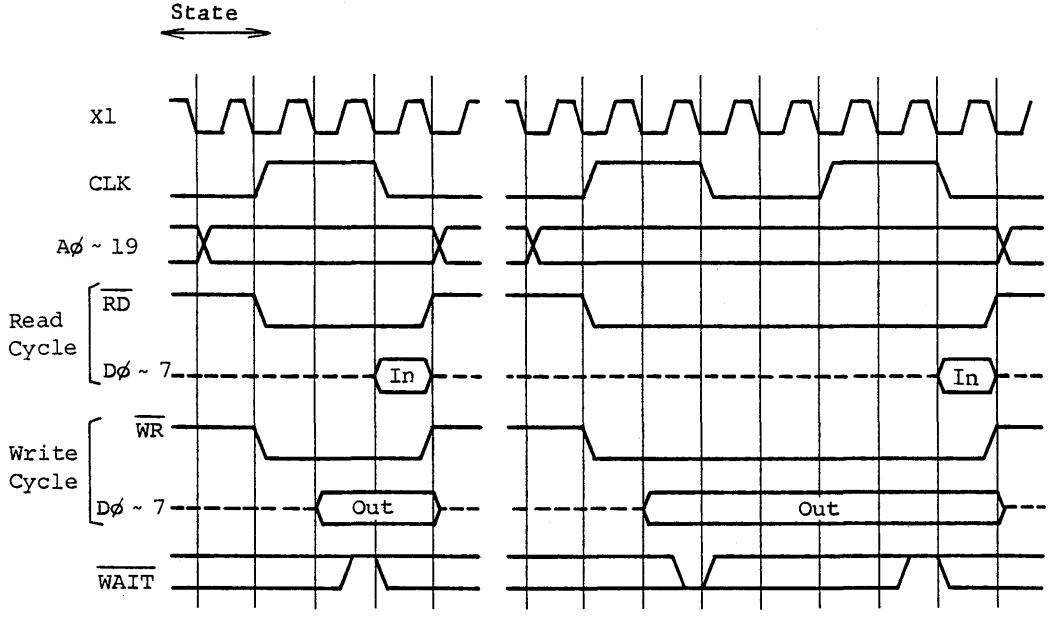


Fig. 3.2.(1) Timing of External Memory Read/Write Cycles

The TMP90C840 CPU has a wait control register (WAITC) that controls waits by using software. The configuration of this register is shown in Fig. 3.2 (2).

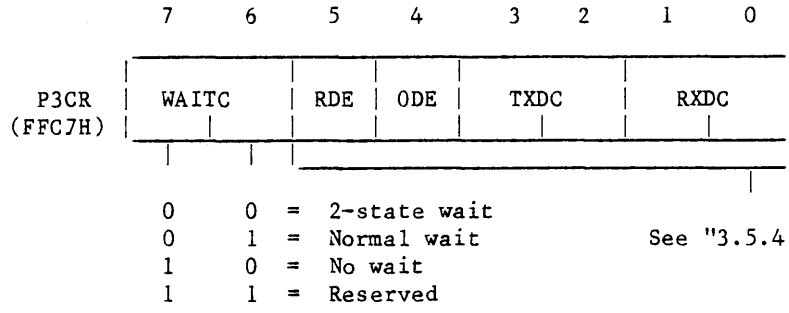


Fig. 3.2 (2) Wait Control Register

This register is assigned to the bits 6 and 7 of the memory address FFC7H in the internal I/O register area (the other bits are used for controlling other functions). It is reset to "00", whereby the register is placed in the 2-state wait mode.
In the "2-state wait mode", only the first wait in a bus cycle is sampled, and all subsequent waits are ignored.
In the "normal wait mode", all wait requests are sampled. The "no wait mode" ignores all waits.

3.2.2 Dummy cycles

The timing of dummy cycles is shown in Fig. 3.2 (3). All through the dummy cycles, the level of both the $\overline{RD}$ and $\overline{WR}$ signals remains at "1" and wait requests are ignored, with the address bus being undefined.
A dummy cycle is also called an "internal operating cycle". The bus cycle becomes the dummy cycle when the CPU reads or write the data from/to the internal memory or internal I/O area.

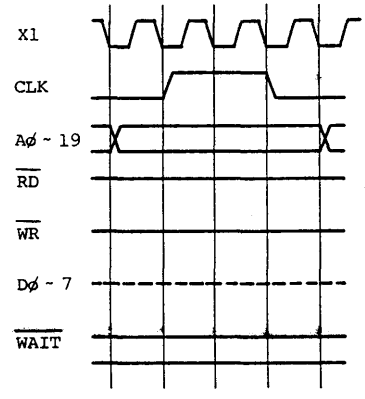


Fig. 3.2 (3) Timing of Dummy Cycles

3.2.3 Interrupt Acknowledge Timing

Fig. 3.2 (4) shows the basic timing of interrupts being acknowledged. An interrupt request may be sampled by the CPU at the falling edge of the CLK signal in the last bus cycle of each instruction. Note, however, that the sampling of a non-maskable interrupt ($\overline{NMI}$) is delayed a half the system clock cycle.

When an interrupt request is acknowledged, the CPU starts an interrupt response sequence that proceeds as follows: 1) A read cycle (In this cycle, the read data is not used in the CPU because the pipeline processing prefetches instructions. The pipeline processing is described in "3.2.6 Bus Operation for Executing Instructions".), 2) two dummy cycles (the CPU receives an interrupt vector from an internal interrupt controller), 3) output of the interrupt vector (000H for the upper address locations A8 to A19) and read out of the dummy cycle, 4) one dummy cycle, 5) saving the contents of the program counter PC and those of the register pair AF into the stack (four write cycles), and 6) the CPU resets the interrupt enable flag IFF to "0" (to disable interrupts) and jumps to the interrupt processing routine.

If a "micro DMA processing" is specified as the interrupt, the CPU follows the sequence to be described in "3.3.2 Micro DMA processing".

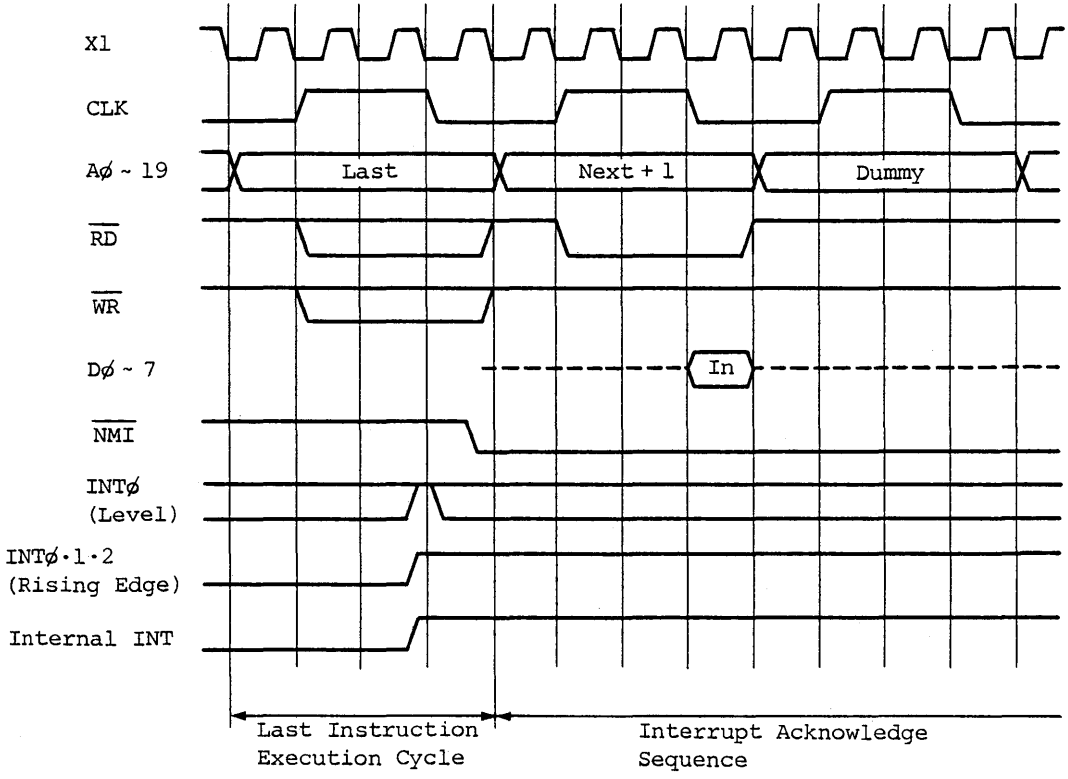


Fig. 3.2 (4) Interrupt Acknoledge Timing

3.2.4 Reset

The basic timing of the reset operation is indicated in Fig. 3.2 (5). In order to reset the TMP90C840, the $\overline{\text{RESET}}$ input must be maintained at the "0" level for at least ten system clock cycles (10 states). When a reset request is accepted, all I/O ports (Port 0/data bus D0 to D7, Port 1/address bus A0 to A7, Port 2/address bus A8 to A15, Port 6 and Port 7) function as input ports (high impedance state). The RD, WR and CLK pins that always function as output ports turn to the "1" level, and the other input ports (P32, P33, Port 4/address bus A16 to A19 and P83) turn to the "0" level. The dedicated input ports remain unchanged. The registers and external memory of the CPU also remain unchanged. Note, however, that the program counter PC, the interrupt enable flag IFF and the bank registers BX and BY are cleared to "0". Register A shows an undefined status.
When the reset is cleared, the CPU starts executing instructions from the address 0000H.

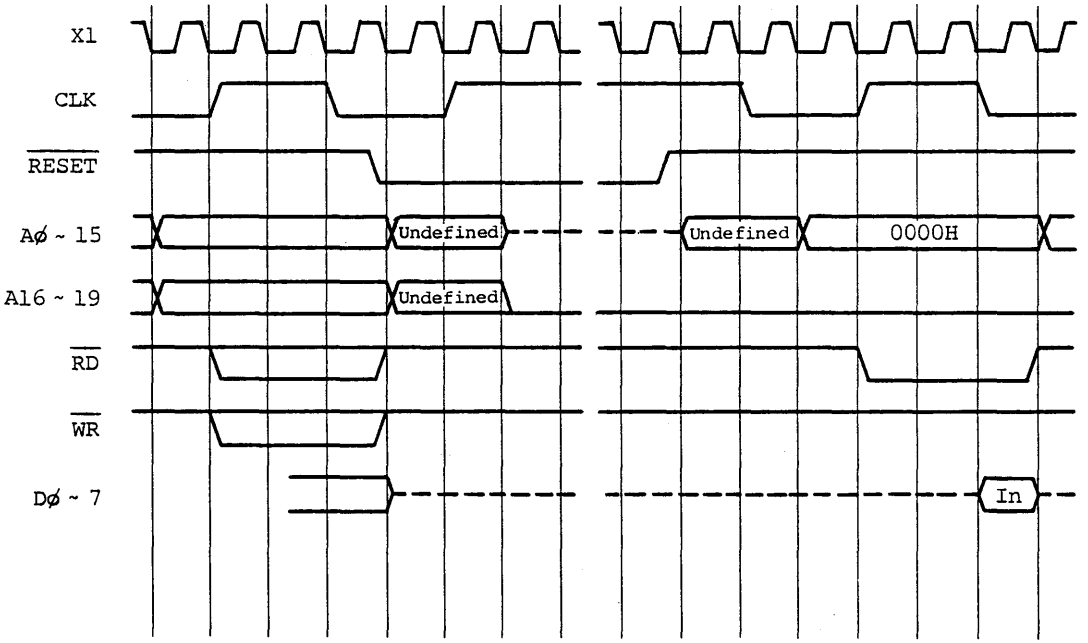


Fig. 3.2 (5) Reset Timing

3.2.5 System flowchart

Fig. 3.2 (6) is a system flowchart of the TMP90C840. A normal operation repeats a loop between "Fetch instruction" and "Execute instruction".
When an interrupt is acknowledged, the CPU proceeds to "Interrupt processing". Executing the return instruction RETI makes the CPU return to the address that follows the address of the SWI instruction in the same way as the SWI instruction is executed under software control.
When a HALT instruction is executed, the CPU suspends the operation until an interrupt is requested. When the interrupt is acknowledged, the CPU starts the interrupt processing. However, when a maskable interrupt is requested with the interrupt enable flag at "0" (interrupts are disabled), the CPU only releases the HALT state and starts executing an instruction that follows the HALT instruction.
For details of the interrupt processing, refer to "3.3 Interrupt Function". The timing of releasing the HALT state is described in "3.4 Standby Function".
By setting the $\overline{\text{RESET}}$ input level to "0", the CPU always returns to "$\overline{\text{RESET}}$" start position without regard to its current position in the flowchart.

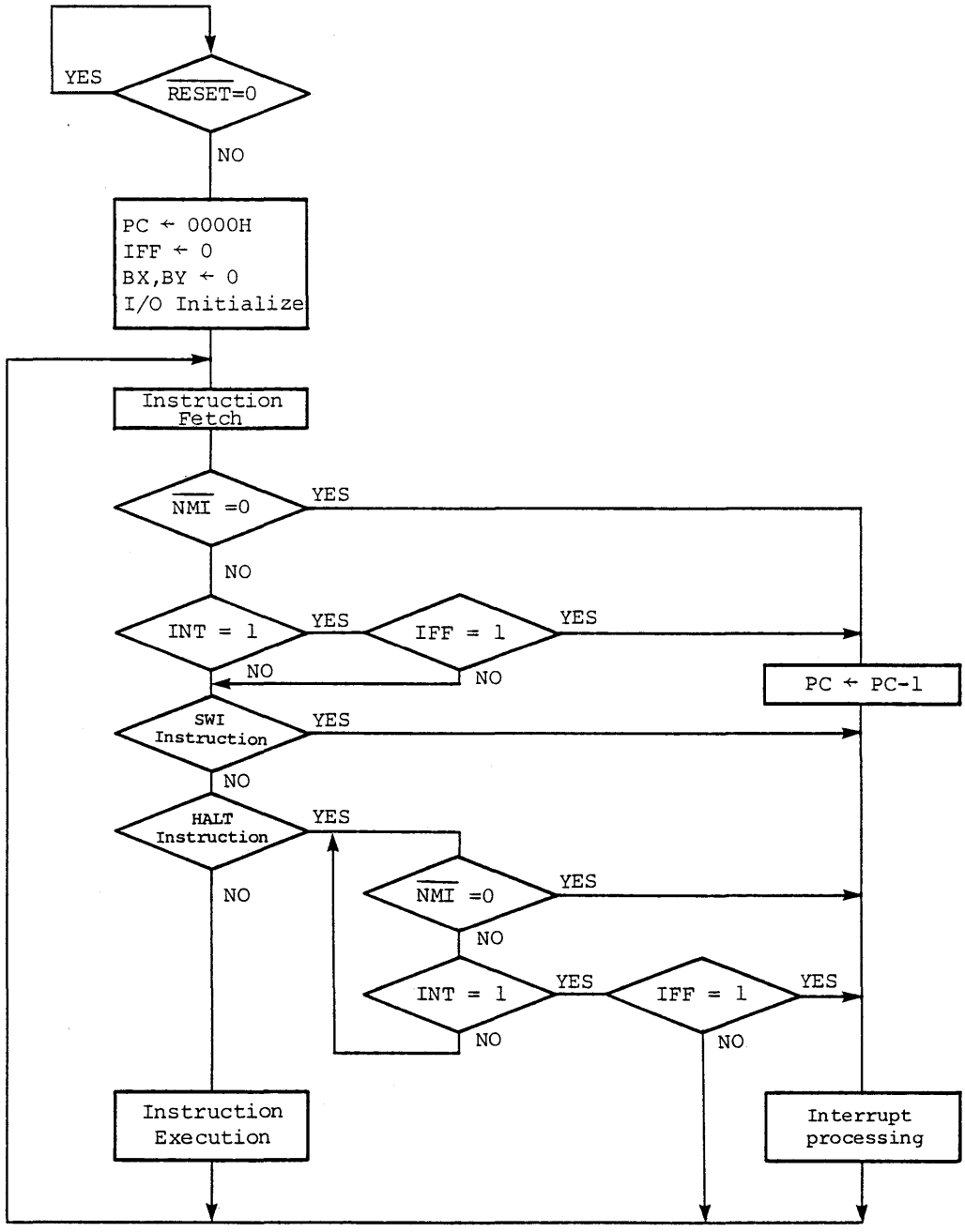


Fig. 3.2 (6) TMP90C840 System Flowchart

3.2.6 Bus operation for executing instructions

The TMP90C840 adopts a pipeline processing method in which it executes an instruction simultaneous with the next instruction fetch. The concept of this processing is illustrated in Fig. 3.2 (7).

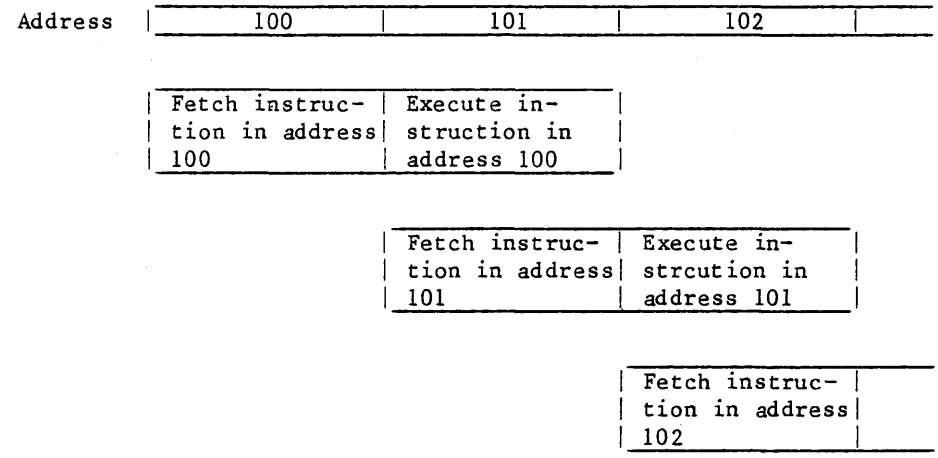


Fig. 3.2 (7) Pipeline Processing

This pipeline processing allows the TMP90C840 to obtain a higher executing speed than the conventional method that fetches the next instruction after the previous instruction is executed. The bus operation for each instruction begins with "fetching a code in the address that follows the first instruction code", and not with "fetching the first instruction code". The first instruction code is fetched when the CPU is executing the previous instruction. An example of this processing is shown in Fig. 3.2 (8).

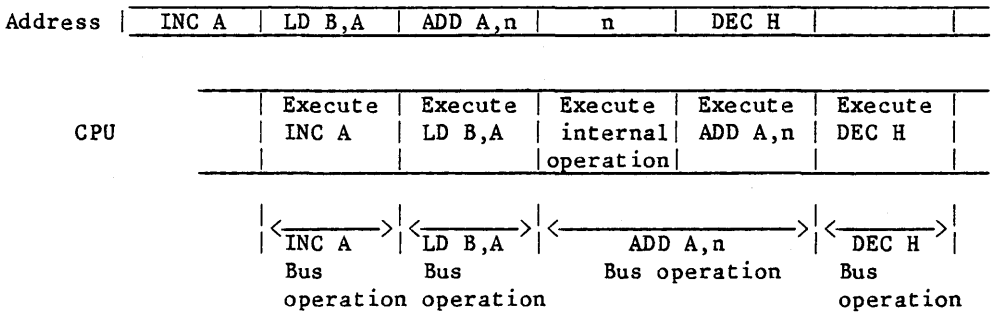


Fig. 3.2 (8) Pipeline Processing and Bus Operation

Table 3.2 is a list of the bus operations for each instruction. Their cycles (read, write or dummy) are indicated by symbols in the table. Each bus cycle is represented by a single symbol (a character string with one to three characters length) and delimited by a colon ":". The bus operations should be read from the left to right.
A capital letter denotes an effective bus cycle, and a small letter (n or d) denotes an invalid bus cycle. For example, data read out in the read cycle of the "n" bus operation have no effect on the CPU operation, and are ignored. The symbol "d" represents an internal operation cycle that involves no read or write of memory.

Table 3.2 Bus Operations for Executing Instructions

Meaning of symbols

Symbol	Data Bus	Address Bus
N	Next Code Read	Next Op Code Address
n	Next Code Read(dummy)	Next Op Code Address
1	1st Code Read	Jump/Call/Return Address
2	2nd Code Read	Op Code Address+1
3	3rd Code Read	Op Code Address+2
4	4th Code Read	Op Code Address+3
5	5th Code Read	Op Code Address+4
6	6th Code Read	Op Code Address+5
d	Dummy (No Read/Write)	Undefined
R	1st Data Read	1st Data Address
R+1	2nd Data Read	2nd Data Address
W	1st Data Write	1st Data Address
W+1	2nd Data Write	2nd Data Address

1. 8-BIT LOADS

Mnemonic		Bus Operation
LD	A,r	N
	r,A	N
	r,g	2:N
	r,n	2:N
	A,(n)	2:d:R:N
	r,(gg)	2:R:N
	r,(ix+d)	2:d:3:R:N
	r,(HL+A)	2:d:d:d:d:R:N
	r,(mn)	2:3:4:R:N
	r,(n)	2:3:R:N
	(n),A	2:d:N:W
	(gg),r	2:N:W
	(ix+d),r	2:d:3:N:W
	(HL+A),r	2:d:d:d:d:N:W
	(mn),r	2:3:4:N:W
	(n),r	2:3:N:W
	(gg),n	2:3:W:N
	(ix+d),n	2:d:3:4:W:N
	(HL+A),n	2:d:d:d:d:3:W:N
	(vw),n	2:3:4:5:W:N
	(w),n	2:d:3:W:N

2. 16-BIT LOADS

Mnemonic		Bus Operation
LD	HL,rr	N:d
	rr,HL	N:d
	rr,gg	2:N:d
	rr,mn	2:3:N
	HL,(n)	2:d:R:R+1:N
	rr,(gg)	2:R:R+1:N
	rr,(ix+d)	2:d:3:R:R+1:N
	rr,(HL+A)	2:d:d:d:d:R:R+1:N
	rr,(mn)	2:3:4:R:R+1:N
	rr,(n)	2:3:R:R+1:N
	(n),HL	2:d:N:W:W+1
	(gg),rr	2:N:W:W+1
	(ix+d),rr	2:d:3:N:W:W+1
	(HL+A),rr	2:d:d:d:d:N:W:W+1
	(mn),rr	2:3:4:N:W:W+1
	(n),rr	2:3:N:W:W+1

16-BIT LOADS(Continued)

Mnemonic		Bus Operation
LDW	(gg),mn	2:3:4:N:W:W+1
	(ix+d),mn	2:d:3:4:5:N:W:W+1
	(HL+A),mn	2:d:d:d:d:3:4:N: W:W+1
	(vw),mn	2:3:4:5:6:N:W:W+1
	(w),mn	2:d:3:4:N:W:W+1
PUSH	qq	N:d:(SP-1)←qqH: (SP-2)←qqL
POP	qq	n: qqL←(SP): qqH←(SP+1): d:N
LDA	rr,ix+d	2:d:3:N:d
	rr,HL+A	2:d:d:d:d:N:d

3. EXCHANGES, BLOCK TRANSFERS AND SEARCHES

Mnemonic		Bus Operation
EX	DE,HL	N
EX	AF,AF'	N
EXX		N
EX	(gg),rr	2:R:R+1: d:W:W+1:N
	(ix+d),rr	2:d:3:R:R+1: d:W:W+1:N
	(HL+A),rr	2:d:d:d:d:R:R+1: d:W:W+1:N
	(mn),rr	2:3:4:R:R+1: d:W:W+1:N
	(n),rr	2:3:R:R+1: d:W:W+1:N
LDI/LDD		2:n:R:W:d:d:N
LDIR/LDDR	repeat	2:n:R:W:d:d:d:d:1
	end	2:n:R:W:d:d:N
CPI/CPD		2:n:R:d:d:d:N
CPIR/CPDR	repeat	2:n:R:d:d:d:d:d:1
	end	2:n:R:d:d:d:N

4. 8-BIT ARITHMETIC AND LOGIC OPERATIONS

Mnemonic		Bus Operation
ADD/ADC/SUB/SBC/ AND/OR/XOR		
	A, g	2:N
	A, n	2:N
	A, (gg)	2:R:N
	A, (ix+d)	2:d:3:R:N
	A, (HL+A)	2:d:d:d:d:R:N
	A, (mn)	2:3:4:R:N
	A, (n)	2:d:R:N
	g, n	2:3:N
	(gg), n	2:3:R:N:W
	(ix+d), n	2:d:3:4:R:N:W
	(HL+A), n	2:d:d:d:d:3:R:N:W
	(vw), n	2:3:4:5:R:N:W
	(w), n	2:3:4:R:N:W
CP	A, g	2:N
	A, n	2:N
	A, (gg)	2:R:N
	A, (ix+d)	2:d:3:R:N
	A, (HL+A)	2:d:d:d:d:R:N
	A, (mn)	2:3:4:R:N
	A, (n)	2:d:R:N
	g, n	2:3:N
	(gg), n	2:3:R:N
	(ix+d), n	2:d:3:4:R:N
	(HL+A), n	2:d:d:d:d:3:R:N
	(vw), n	2:3:4:5:R:N
	(w), n	2:3:4:R:N
INC/DEC	r	N
	(gg)	2:R:N:W
	(ix+d)	2:d:3:R:N:W
	(HL+A)	2:d:d:d:d:R:N:W
	(mn)	2:3:4:R:N:W
	(n)	2:d:R:N:W
INCX/DECX	(n)true	2:d:R:N:W
	false	2:d:N

5. SPECIAL FUNCTIONS

Mnemonic		Bus Operation
DAA	A	N:d
CPL/NEG	A	N
LDAR	HL, PC+cd	2:3:N:d
CCF/SCF/RCF		N
NOP		N
HALT		N:d
DI/EI		N
SWI		n:d:d:1:d: (SP-1)←PCH: (SP-2)←PCL: (SP-3)←A: (SP-4)←F: 1
MUL/DIV	HL, g	2:n:d:d:d:d:d:d:N
	HL, n	2:d:d:d:d:d:d:N
	HL, (gg)	2:R:d:d:d: d:d:d:N
	HL, (ix+d)	2:d:3:R:d:d:d: d:d:d:N
	HL, (HL+A)	2:d:d:d:d:R:d:d: d:d:d:d:N
	HL, (mn)	2:3:4:R:d:d:d: d:d:d:N
	HL, (n)	2:3:R:d:d:d: d:d:d:N

6. 16-BIT ARITHMETIC AND LOGIC OPERATIONS

Mnemonic		Bus Operation
ADD/ADC/SUB/SBC/ AND/OR/XOR/CP		
	HL,gg	2:N:d:d
	HL,mn	2:3:N
	HL,(gg)	2:R:R+1:N
	HL,(ix+d)	2:d:3:R:R+1:N
	HL,(HL+A)	2:d:d:d:d:R:R+1:N
	HL,(mn)	2:3:4:R:R+1:N
	HL,(n)	2:d:R:R+1:N
ADD	ix,gg	2:N:d:d
	ix,mn	2:3:N
	ix,(gg)	2:R:R+1:N
	ix,(jx+d)	2:d:3:R:R+1:N
	ix,(HL+A)	2:d:d:d:d:R:R+1:N
	ix,(mn)	2:3:4:R:R+1:N
	ix,(n)	2:3:R:R+1:N
INC/DEC	rr	N:d
INCW/DECW	(gg)	2:R:R+1: N:W:W+1
	(ix+d)	2:d:3:R:R+1: N:W:W+1
	(HL+A)	2:d:d:d:d:R:R+1: N:W:W+1
	(mn)	2:3:4:R:R+1: N:W:W+1
	(n)	2:d:R:R+1: N:W:W+1

7. ROTATES AND SHIFTS

Mnemonic		Bus Operation
RLC/RRC/RL/RR/ SLA/SRA/SLL/SRL		
	A	N
	g	2:N
	(gg)	2:R:N:W
	(ix+d)	2:d:3:R:N:W
	(HL+A)	2:d:d:d:d:R:N:W
	(mn)	2:3:4:R:N:W
	(n)	2:3:R:N:W
RLD/RRD	(gg)	2:R:d:d:N:W
	(ix+d)	2:d:3:R:d:d:N:W
	(HL+A)	2:d:d:d:d: R:d:d:N:W
	(mn)	2:3:4:R:d:d:N:W
	(n)	2:3:R:d:d:N:W

8. BIT OPERATIONS

Mnemonic		Bus Operation
BIT	b,g	2:N
	b,(gg)	2:R:N
	b,(ix+d)	2:d:3:R:N
	b,(HL+A)	2:d:d:d:d:R:N
	b,(mn)	2:3:4:R:N
	b,(n)	2:d:R:N
SET/RES	b,g	2:N
	b,(gg)	2:R:N:d:W
	b,(ix+d)	2:d:3:R:N:d:W
	b,(HL+A)	2:d:d:d:d: R:N:d:W
	b,(mn)	2:3:4:R:N:d:W
	b,(n)	2:d:R:N:d:W
TSET	b,g	2:N:d:d
	b,(gg)	2:R:N:d:d:W
	b,(ix+d)	2:d:3:R:N:d:d:W
	b,(HL+A)	2:d:d:d:d: R:N:d:d:W
	b,(mn)	2:3:4:R:N:d:d:W
	b,(n)	2:3:R:N:d:d:W

9. JUMPS, CALLS AND RETURNS

Mnemonic		Bus Operation
JP cc, gg	true	2:n:d:1
	false	2:N:d
cc, ix+d	true	2:d:3:n:d:1
	false	2:d:3:N:d
cc, HL+A	true	2:d:d:d:d:n:d:1
	false	2:d:d:d:d:N:d
cc, mn	true	2:3:4:n:d:1
	false	2:3:4:N:d
JR cc, PC+d	true	2:n:d:1
	false	2:N
JP mn		2:3:d:1
JRL PC+cd		2:3:d:d:1
CALL cc, gg	true	2:n:d:d: (SP-1)←PCH: (SP-2)←PCL: 1
	false	2:N:d
cc, ix+d	true	2:d:3:n:d:d: (SP-1)←PCH: (SP-2)←PCL: 1
	false	2:d:3:N:d
cc, HL+A	true	2:d:d:d:d:n:d:d: (SP-1)←PCH: (SP-2)←PCL: 1
	false	2:d:d:d:d:N:d
cc, mn	true	2:3:4:n:d:d: (SP-1)←PCH: (SP-2)←PCL: 1
	false	2:3:4:N:d
CALL mn		2:3:d:d: (SP-1)←PCH (SP-2)←PCL 1
CALR PC+cd		2:3:d:d:d: (SP-1)←PCH (SP-2)←PCL 1
DJNZ PC+d	true	2:d:d:d:1
	false	2:d:d:d:N
BC, PC+d	true	2:d:d:d:1
	false	2:d:d:d:N
RET		n: PCL←(SP): PCH←(SP+1): d:1
RET cc	true	2:n:d: PCL←(SP): PCH←(SP+1): d:1
	false	2:N:d
RETI		n: F←(SP): A←(SP+1): PCL←(SP+2): PCH←(SP+3): d:1

10. INTERRUPT

MODE	Bus Operation
NORMAL INTERRUPT	n+1:d:d:1:d: (SP-1)←PCH: (SP-2)←PCL: (SP-3)←A: (SP-4)←F: 1
MICRO DMA	n+1:d:d:d: DSTL←(FF00H+V+1): DSTH←(FF00H+V+2): SRCL←(FF00H+V+3): SRCH←(FF00H+V+4): CMD ←(FF00H+V+5): TEMP←(SRC): (DST) ←TEMP: TEMP←(SRC+1) / d: (DST+1) ←TEMP / d: d: d: (FF00H+V+4) ←SRCH': (FF00H+V+3) ←SRCL': (FF00H+V+2) ←DSTH': (FF00H+V+1) ←DSTL': COUNT ←(FF00H+V): d: (FF00H+V) ←COUNT': N
if COUNT'=0 then execute	d:d:d: (SP-1)←PCH: (SP-2)←PCL: (SP-3)←A: (SP-4)←F: 1

3.3 Interrupt Functions

The TMP90C840 supports a general purpose interrupt processing mode to acknowledge internal and external interrupt requests, as well as a micro DMA processing mode that enables automatic data transfer by the CPU.
Immediately after the reset state is released, all interrupt requests are processed in the general purpose interrupt processing mode. However, they can be processed in the micro DMA processing mode by using a DMA enable register to be described later.
Fig. 3.3 (1) is a flowchart of the interrupt response sequence.

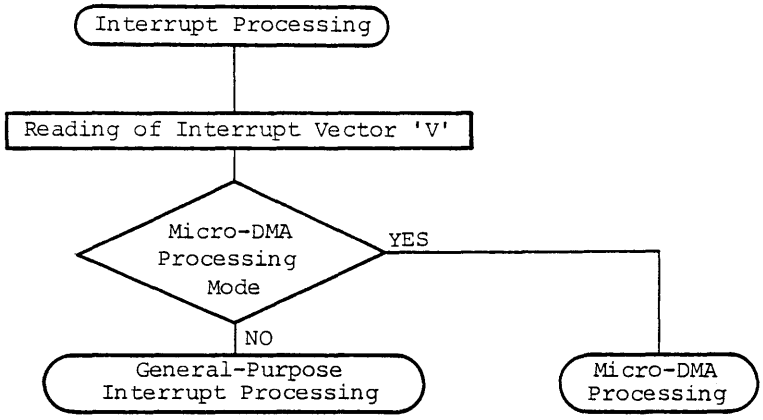


Fig. 3.3 (1) Interrupt Response Flowchart

When an interrupt is requested, the source of the interrupt transmits the request to the CPU via an internal interrupt controller. The CPU starts processing the interrupt if it is a non-maskable or maskable interrupt requested in the EI state. However, a maskable interrupt requested in the DI state is ignored and acknowledged.
Having acknowledged an interrupt, the CPU reads out the interrupt vector from the internal interrupt controller to find out the interrupt source. Then, the CPU checks if the interrupt requests the general purpose interrupt processing or the micro DMA processing, and proceeds to each processing..
As the reading of an interrupt vecotors is performed in the internal operating cycles, the bus cycle at that time results in dummy cycles.

3.3.1 Normal interrupt processing

A normal interrupt is processed as shown in Fig. 3.3. (2).
The CPU stores the contents of the program counter PC and the register pair AF into the stack, and resets the interrupt enable flag IFF to "0" (disable interrupts). It then transfers the value of the interrupt vector "V" to the program counter, and the processing jumps to an interrupt processing program.

The overhead for the entire process from accepting an interrupt to jumping to an interrupt processing program is 20 states.

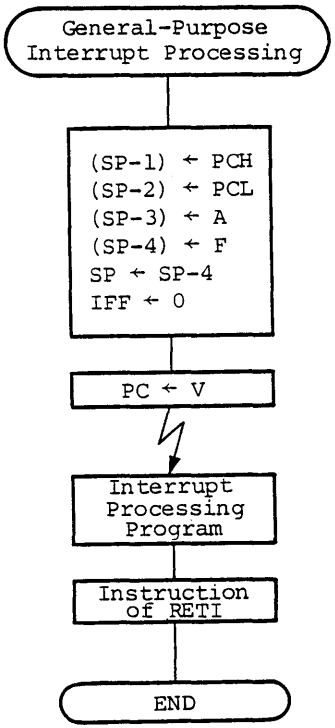


Fig. 3.3 (2) General Purpose Interrupt Processing Flowchart

An interrupt processing program ends with a RETI instruction.
When this instruction is executed, the data previously stacked from the program counter PC and the register pair AF are restored.
After the CPU reads out the interrupt vector, the source of an interrupt requested acknowledges that the CPU accepts the request, and clears the request.
A non-maskable interrupt cannot be disabled by programming. A maskable interrupt, on the other hand, can be enabled or disabled by programming. An interrupt enable flip flop (IFF) is provided on the bit 5 of Register F in the CPU. The interrupt is enabled or disabled by setting IFF to "1" by the EI instruction or to "0" by the DI instruction, respectively. IFF is reset to "0" by the reset operation or the acceptance of any interrupt (including non-maskable interrupt). The EI instruction is executed after the subsecuent instructions is executed.

Table 3.3 (1) lists the possible interrupt sources.

Table 3.3 (1) Interrupt Sources

Priority order	Type	Interrupt source	Vector value	Start address of general purpose interrupt processing	Start address of Mico DMA processing parameter
1	Non maskable	SWI instruction	10H	0010H	-
2		NMI(Input from NMI pin)	18H	0018H	-
3		INTWD (watchdog)	20H	0020H	-
4	Maskable	INT0 (External input 0)	28H	0028H	FF28H
5		INTT0 (Timer 0)	30H	0030H	FF30H
6		INTT1 (Timer 1)	38H	0038H	FF38H
7		INTT2 (Timer 2)	40H	0040H	FF40H
7		INTAD (A/D Converter)	40H	0040H	FF40H
8		INTT3 (Timer 3)	48H	0048H	FF48H
9		INTT4 (Timer 4)	50H	0050H	FF50H
10		INT1 (External input 1)	58H	0058H	FF58H
11		INTT5 (Timer 5)	60H	0060H	FF60H
12		INT2 (External input 2)	68H	0068H	FF68H
13		INTRX (End of serial receiving)	70H	0070H	FF70H
14		INTTX (End of serial transmission)	78H	0078H	FF78H

(Note) Either INTT2 or INTAD is selected by software.

The "priority order" in the table is the order of the interrupt source used by the CPU for accepting more than one interrupt requested at one time.
If interrupt of fourth and fifth orders are requested simultaneously, for example, an interrupt of the "5th" priority is acknowledged after a "4th" priority interrupt processing has been completed by a RETI instruction. However, a lower priority interrupt can be acknowledged immediately by executing an EI instruction in a program that processes a higher priority interrupt.
The internal interrupt controller merely determines the priority of the sources of interrupts to be acknowledged by the CPU when more than one interrupt are requested at a time. It is, therefore, unable to compare the priority of interrupt being executed with the one being requested.

3.3.2 Micro DMA processing

Fig. 3.3 (3) is a flowchart of the micro DMA processing. Parameters (addresses of source and destination, and transfer mode) for the data transfer between memories are loaded by the CPU from an address modified by an interrupt vector value. After the data transfer between memories accroding to these parameter, these parameters are

updated and saved by the CPU into the original locations. The CPU then decrements the number of transfers, and completes the micro DMA processing unless the result is "0".
If the number of transfer becomes "0", the CPU proceeds to the general purpose interrupt handling described in the previous chapter.

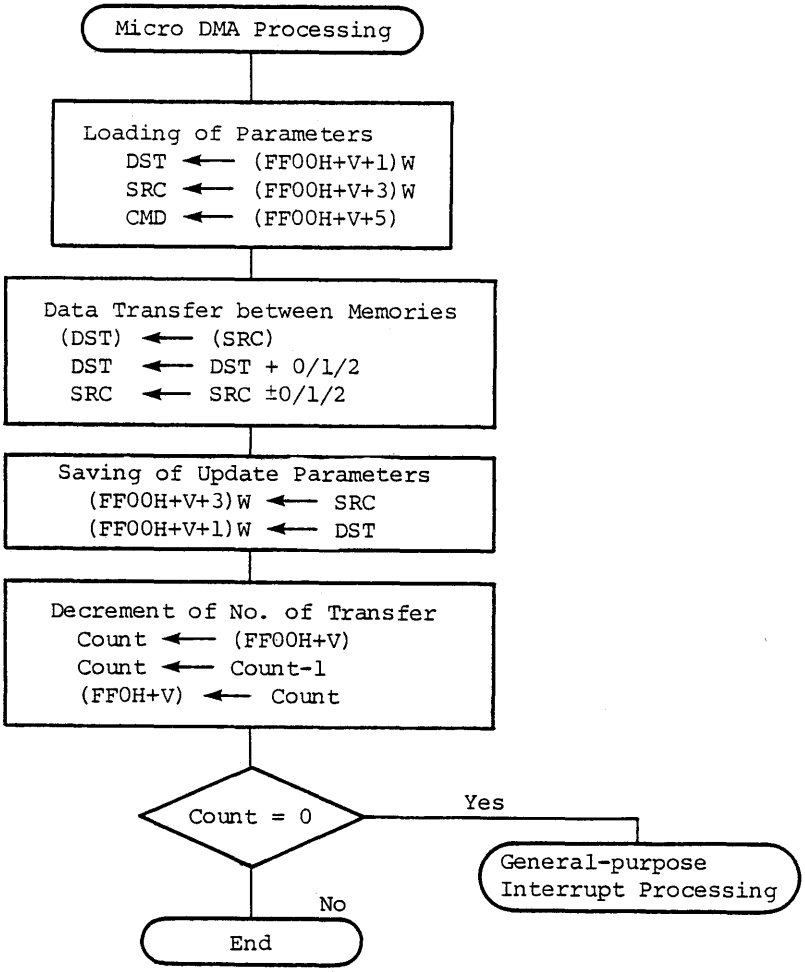


Fig. 3.3 (3) Micro DMA Processing Flowchart

The micro DMA processing is performed by using only hardware to process interrupts mostly completed by simple data transfer. The use of hardware allows the micro DMA processing to handle the interrupt in a higher speed than the conventional methods using software. The CPU registers are not affected by the micro DMA processing.
Fig. 3.3.(4) shows the functions of parameters used in the micro DMA processing.

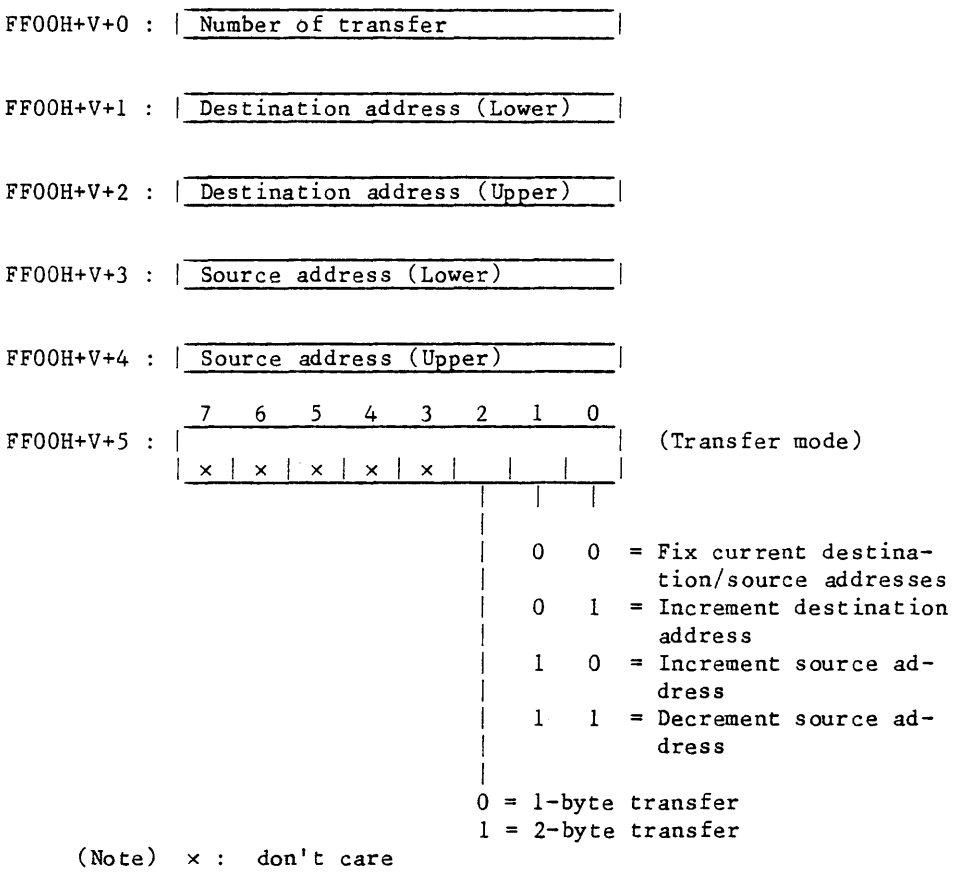


Fig. 3.3 (4) Parameters for Micro DMA Processing

Parameters for the micro DMA processing are located in the internal RAM area (See Table 3.3 (1) Interrupt Sources). The start address of each parameter is "FF00HH + interrupt vector value", from which a six bytes' space is used for the parameter. This space can be used for any other memory purposes if the micro DMA processing is not used.
The parameters normally consist of the number of transfer, addresses of destination and source, and transfer mode. The number of transfer indicates the number of data transfer accepted in the micro DMA processing.
The amount of data transferred by a single micro DMA processing is limited to one or two bytes. Both the destination and source addresses are specified by 2-byte data. The address space available for the micro DMA processing ranges from 0000H to FFFFH.
Bits 0 and 1 of the transfer mode indicates the mode updating the source and/or destination, and the bit 2 indicates the data length (one byte or two bytes).

Table 3.3 (2) shows the relation between the transfer mode and the result of updating the destination/source addresses.

Table 3.3 (2) Addresses Updated by Micro DMA Processing

Transfer mode	Function	Destination address	Source address
000	1-byte transfer: Fix the current source/ destination addresses	0	0
001	1-byte transfer: Increment the destination address	+ 1	0
010	1-byte transfer: Increment the source address	0	+ 1
011	1-byte transfer: Decrement the source address	0	- 1
100	2-byte transfer: Fix the current source/ destination addresses	0	0
101	2-byte transfer: Increment the destination address	+ 2	0
110	2-byte transfer: Increment the source address	0	+ 2
111	2-byte transfer: Decrement the source address	0	- 2

In the 2-byte transfer mode, data are transferred as follows:

(Destination address) <- (Source address)
(Destination address+1) <- (Source address+1)

Similar data transfers are made in the modes that "decrement the source address", but the updated results are different as shown in the table 3.3 (2).

Fig. 3.3 (5) shows an example of the micro DMA processing that handles data receiving of internal serial I/O.
This is an example of executing "an interrupt processing program after serial data receiving" after receiving 7-frame data (Assume 1 frame = 1 byte for this example) and saving them into the memory addresses from FF00H to FF06H.

```
; Initial setting for serial receiving

      CALL  SIOINIT
      SET   1,(FFE6H)      ;  Enable an interrupt for serial data
                              receiving.
      SET   1,(FFE8H)      ;  Set the micro DMA processing mode for the
                              interrupt.
      LD    (FF70H),7      ;  Set the number of transfer = 7
      LDW   (FF71H),FF00H  ;  Set FF00H for the destination address.
      LDW   (FF73H),FFEBH  ;  Set FFEBH for the source (serial
                              receiving buffer) address.
      LD    (FF75H),1      ;  Set the transfer mode (1-byte transfer:
                              Increment destination address).
      EI
       :
       :
      ORG   0070H

     ┌──────────────────────────────────┐
     │                                  │
     │ Interrupt processing program     │
     │ after serial data receiving      │
     │                                  │
     └──────────────────────────────────┘

      REETI
```

Fig. 3.3 (5) Example of Micro DMA Processing

The bus operation in the general purpose interrupt processing and the micro DMA processing is included in "Table 3.2 Bus Operation for Executing Instructions" in the previous section.
The micro DMA processing time (when the number of transfer is not decremented to 0) is 46 states without regard to the 1-byte/2-byte transfer mode.
Figure 3.3 (6) shows the interrupt processing flowchart.

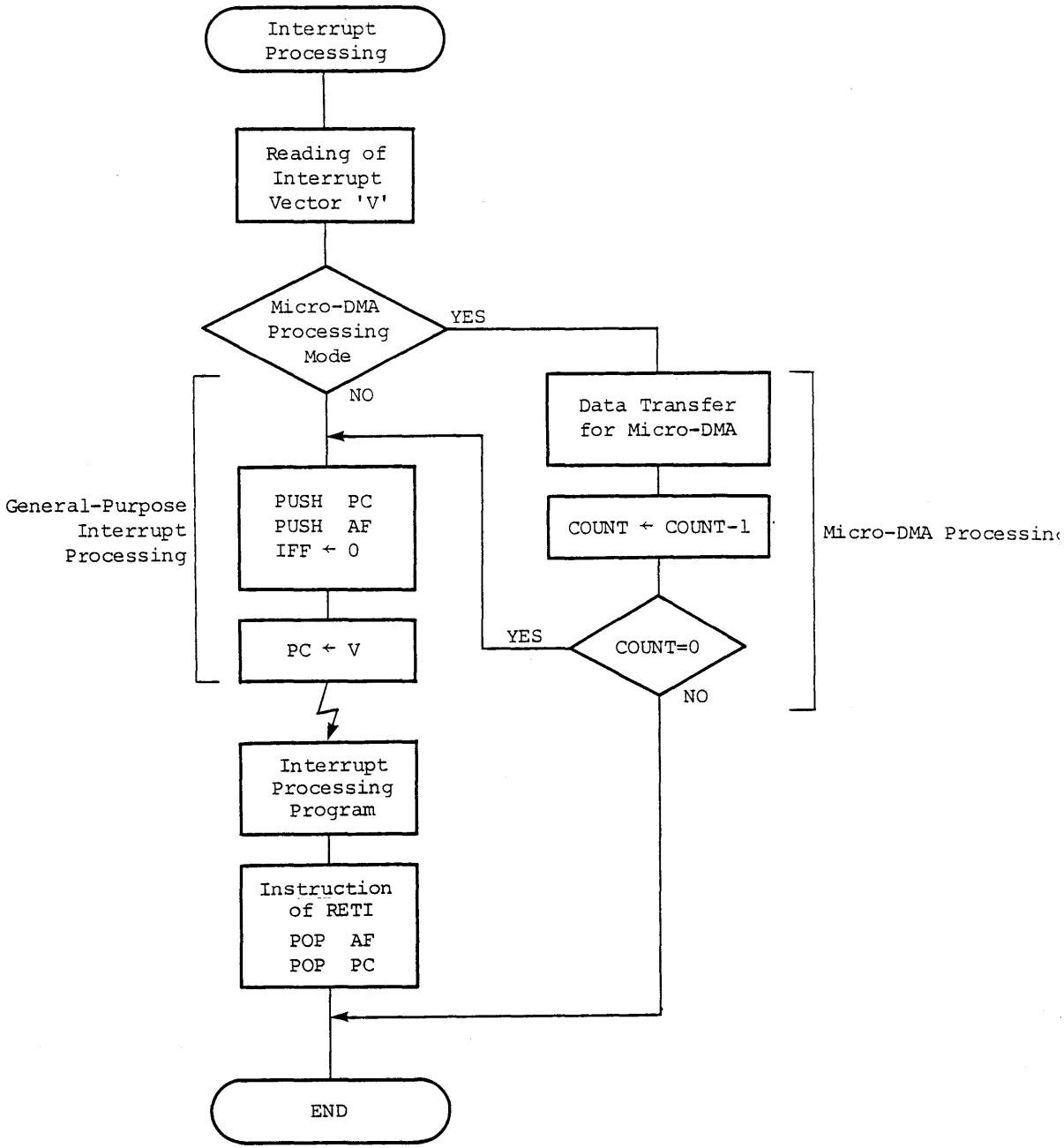


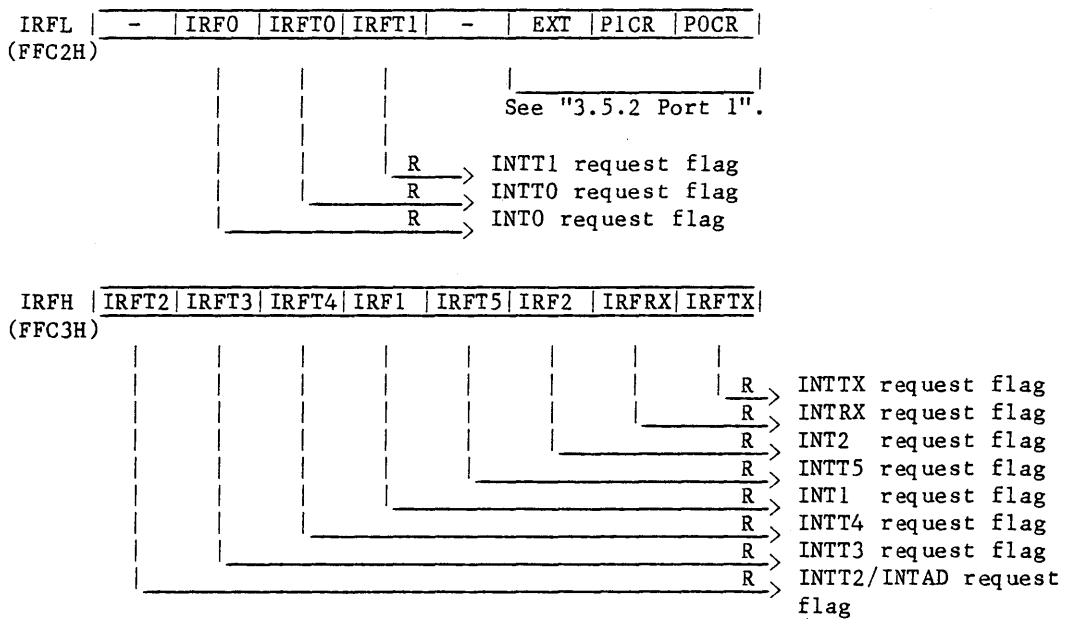
Fig. 3.3 (6) Interrupt Processing Flowchart

3.3.3 Interrupt controller

Fig. 3.3 (8) outlines the interrupt circuit. The left side of this figure represents an interrupt controller, and the right side comprises the CPU's interrupt request signal circuit and HALT release signal circuit (See "3.4 Standby Function" for the HALT operation).
The interrupt controller consists of Interrupt Request Flip-flops, interrupt enable flags, and micro DMA enable flags allocated to each of 14 channels. The Interrupt Request Flip-flops serve to latch interrupt requests from peripherals. Each flip-flop is reset to "0" when a reset or interrupt is acknowledged by the CPU and the vector of the interrupt channel is read into the CPU, or when the CPU executes an instruction that clears a request to interrupt that channel (write "vector divided by 8" into the memory address FFC3H). For example, by executing

LD (FFC3H), 58H/8,

the Interrupt Request Flip-flops for the interrupt channel "INT1" whose vector is 58H is reset to "0".
The status of an Interrupt Request Flip-flops is found out by reading the memory address FFC2H or FFC3H. "0" denotes there is no interrupt request, and "1" denotes that an interrupt is requested. Fig. 3.3 (7) illustrates the bit configuration indicating the status of Interrupt Request Flip-flops.



(Caution) Writing "vector divided by 8" into the memory address FFC3H clears the Flip-flop for the specified interrupt request.

Fig. 3.3 (7) Configuration of Interrupt Request Flip-flops

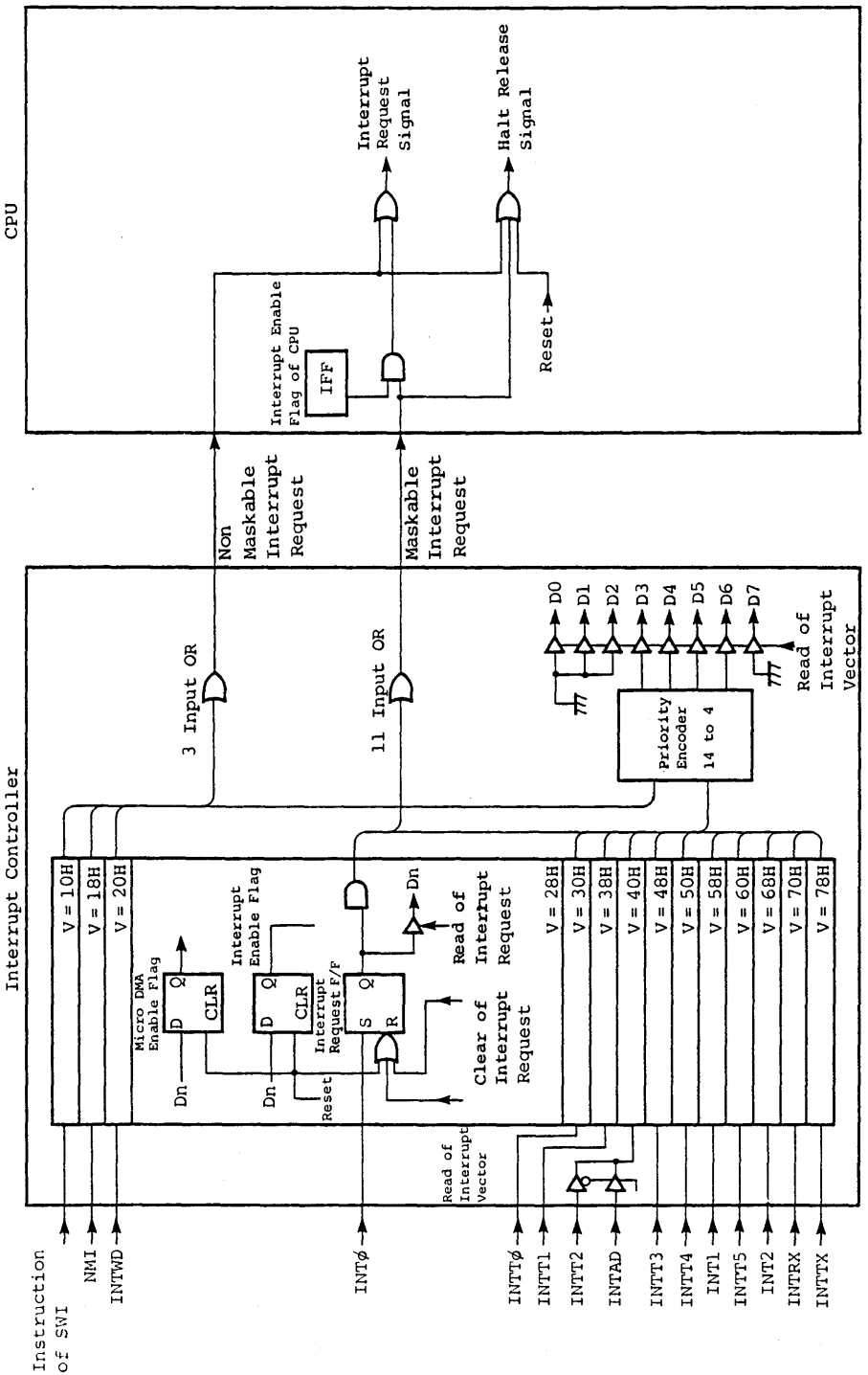


Fig. 3.3 (8) Interrupt Circuit

The interrupt enable flags provided for all interrupt request channels are assigned to the memory address FFE7H or FFE8H. Setting any of these flags to "1" enables an interrupt of the respective channel. These flags are initialized to "0" by resetting.
The micro DMA enable flag also provided for each interrupt request channel is assigned to the memory address FFE6H or FFE7H. The interrupt request for each channel is placed in the micro DMA processing mode by setting this flag to "1". This flag is initialized to "0" (general purpose interrupt processing mode) by resetting.
Fig. 3.3. (9) shows the bit configuration of the interrupt enable flags and micro DMA enable flags.
Interrupt by Timer 2 (INTT2) and that by A/D converter (INTAD) use a common interrupt request channel. The interrupt controller first accepts INTT2 after a reset. INTAD can be used by setting the "INTT2/INTAD selection bit" (ADIS: Bit 3 of memory address FFE7H) to "1".

Attention should be paid to the following three modes having special circuits:

INT0 Level mode	If INT0 is not an edge-based interrupt, the function of Interrupt Request Flip-flop is cancelled. Therefore the interrupt request signal must be held until the interrupt request is acknowledged by the CPU. A change in the mode (between edge and level) automatically clears the interrupt request flag.
INTAD level mode	The Interrupt Request Flip-flop can be cleared only by resetting or reading the register that stores A/D conversion value, and cannot be cleared by an instruction. A change in the interrupt source (between INTAD and INTT2) automatically clears the interrupt request flag.
INTRX level mode	The Interrupt Request Flip-flop is cleared only by resetting or reading the serial channel receiving buffer, and not by an instruction.

Interrupt enable flags

	IET2	IET3	IET4	IE1	IET5	IE2	IERX	IETX	R/W
INTEL (FFE6H)									

- IETX → INTTX interrupt enable flag
- IERX → INTRX interrupt enable flag
- IE2 → INT2 interrupt enable flag
- IET5 → INTT5 interrupt enable flag
- IE1 → INT1 interrupt enable flag
- IET4 → INTT4 interrupt enable flag
- IET3 → INTT3 interrupt enable flag
- IET2 → INTT2 interrupt enable flag

Interrupt and micro DMA enable flags

	"0"	DE0	DET0	DET1	ADIS	IE0	IET0	IET1	R/W
INTEH (FFE7H)									

- IET1 → INTT1 interrupt enable flag
- IET0 → INTT0 interrupt enable flag
- IE0 → INT0 interrupt enable flag
- ADIS → INTT2/INTAD selection
  - * 0 : INTT2
  - 1 : INTAD
- DET1 → INTT1 DMA enable flag
- DET0 → INTT0 DMA enable flag
- DE0 → INT0 DMA enable flag
- "0" → Write "0"

* indicates the initial value after reset.

Micro DMA enable flags

	DET2	DET3	DET4	DE1	DET5	DE2	DERX	DETX	R/W
DMAEH (FFE8H)									

- DETX → INTTX DMA enable flag
- DERX → INTRX DMA enable flag
- DE2 → INT2 DMA enable flag
- DET5 → INTT5 DMA enable flag
- DE1 → INT1 DMA enable flag
- DET4 → INTT4 DMA enable flag
- DET3 → INTT3 DMA enable flag
- DET2 → INTT2 DMA enable flag

Fig. 3.3 (9) Interrupt/Micro DMA Enable Flags

3.4 Standby Function

When a HALT instruction is executed, the TMP90C840 selects one of the following modes as determined by the halt mode set register:

(1) RUN : Suspends only the CPU operation. The power consumption remains unchanged.

(2) IDLE1 : Suspends all internal circuits except the internal oscillator. In this mode, the power consumption is less than 1/10 of that in the normal operation.

(3) IDLE2 : Operate only the internal oscillator and specific internal I/O devices. The power consumption is less than 1/3 of that in the normal operation.

(4) STOP : Suspends all internal circuits including the internal oscillator. In this mode, the power consumption is considerably reduced.

The HALT mode set register (HALTM) is assigned to the bits 2 and 3 of the memory address FFD2H in the internal I/O register area (other bits are used to control other functions). The register is reset to "00" (RUN mode) by resetting.
These HALT state can be released by resetting or requesting an interrupt. Either a non-maskable or maskable interrupt is acknowledged and processed if the CPU executes the EI (enable interrupt) instruction. However the CPU executes the DI (disable interrupt) instruction, a maskable interrupt may be accepted, and the CPU starts executing the instruction following the HALT instruction.

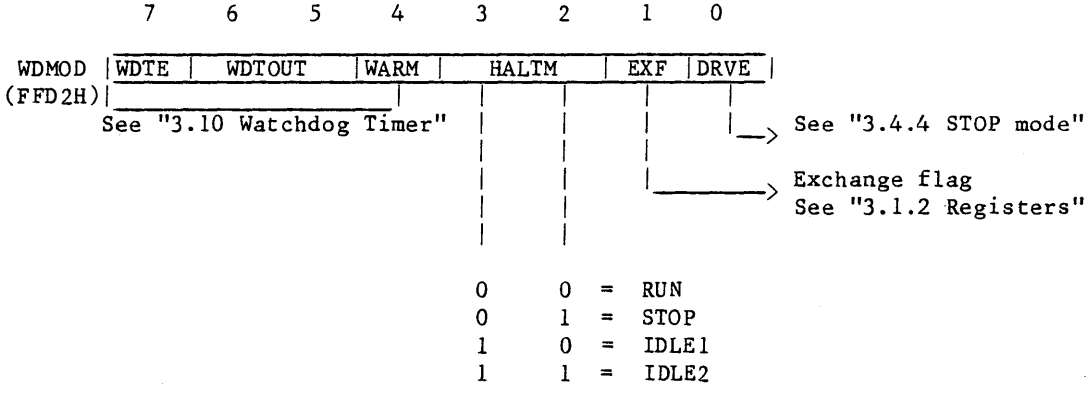


Fig. 3.4 (1) HALT Mode Set Register

3.4.1 RUN Mode

Fig. 3.4 (2) shows the timing for releasing the HALT state by interrupts in the RUN/IDLE 2 mode.
In the RUN mode, the system clock in the MCU continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction. Until the HALT state is released, the CPU repeats dummy cycles. In the HALT state, an interrupt request is sampled with the rising edge of the "CLK" signal.

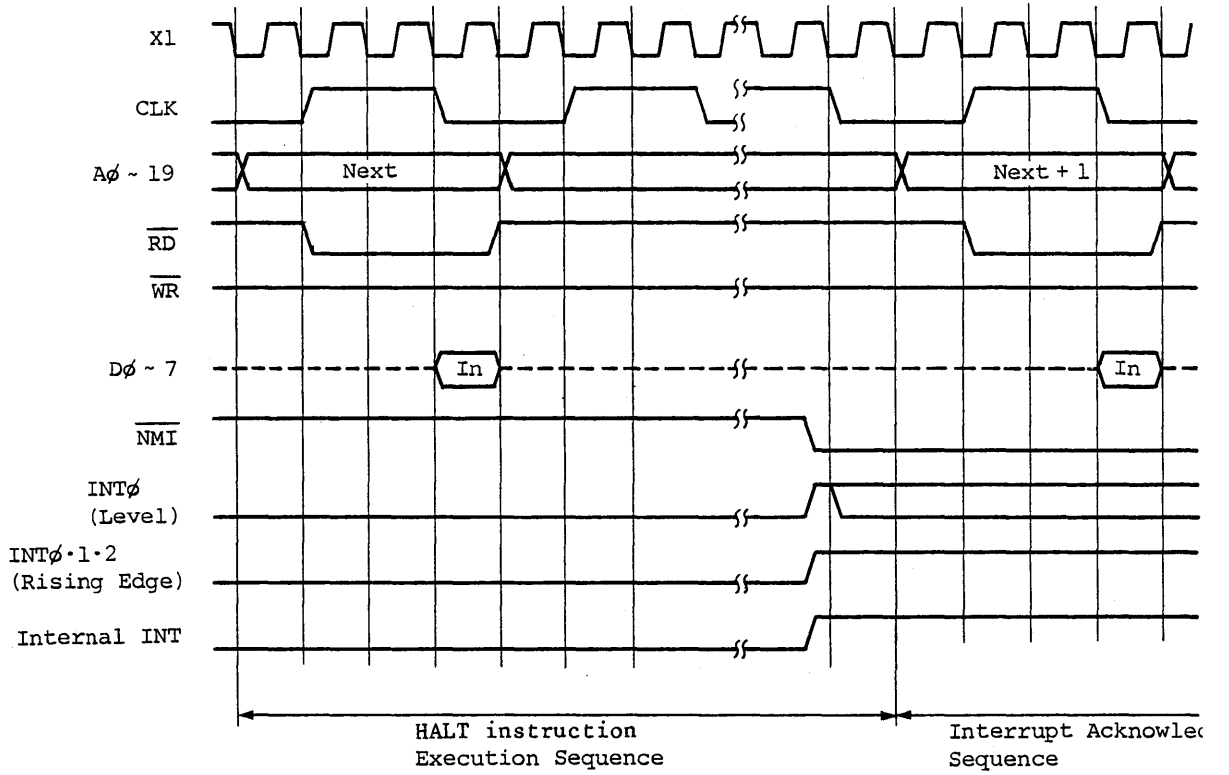


Fig. 3.4 (2) Timing Chart for Releasing the HALT State by Interrupts in RUN/IDLE 2 Modes

3.4.2 IDLE 1 mode

Fig. 3.4 (3) illustrates the timing for releasing the HALT state by interrupts in the IDLE 1 mode.
In the IDLE 1 mode, only the internal oscillator and the watchdog timer counter operate. The system clock in the MCU stops, and the CLK signal is fixed at the "1" level.
In the HALT state, an instruction request is sampled asynchronously with the system clock, however the HALT release (restart of operation) is performed synchronously with it.

(Note) An interrupt requested by the watchdog timer is prohibited through the HALT period in this mode.

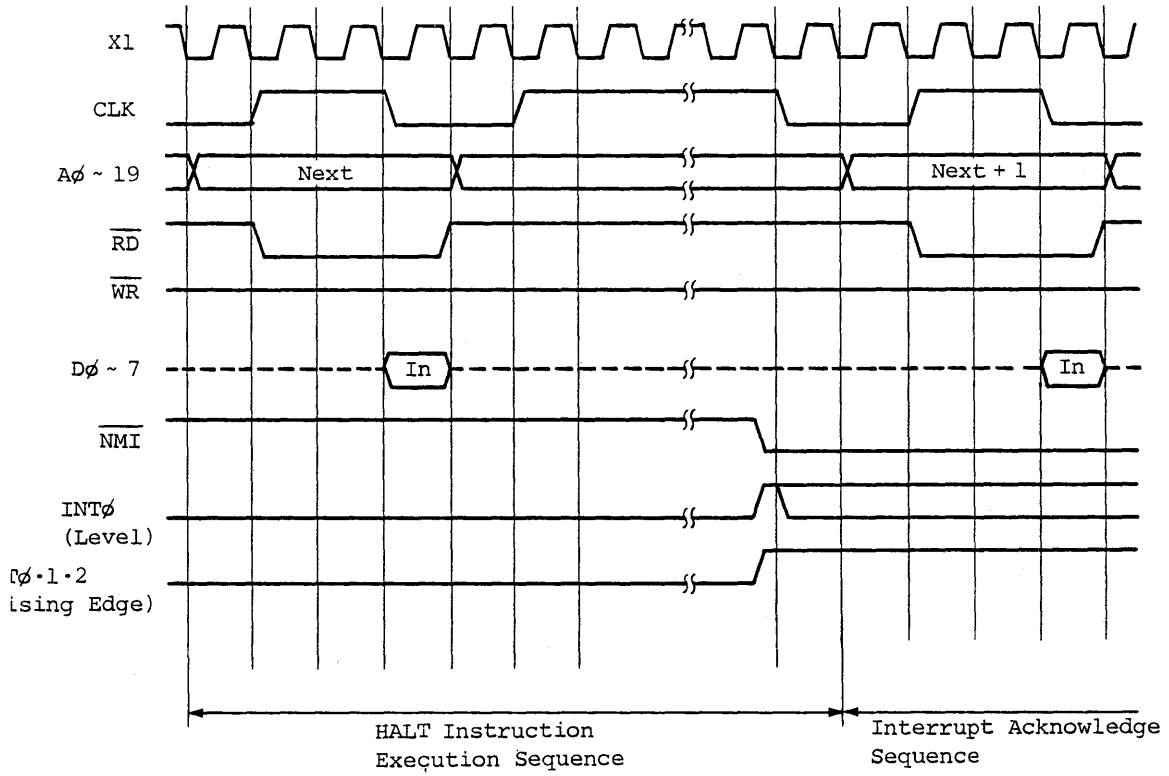


Fig. 3.4 (3) Timing Chart of HALT Released by Interrupts in IDLE1 Mode

3.4.3 IDLE 2 mode

Fig. 3.4 (2) shows the timing of HALT release caused by interupts in the RUN/IDLE 2 mode.

In the IDLE2 mode, the HALT state is released by an interrupt with the same timing as in the RUN mode, except the internal operation of the MCU. In the RUN mode, only the CPU stops executing the current instruction, and the system clock is supplied to all internal devices. In the IDLE 2 mode, however, the system clock is supplied to only specific internal I/O devices. As a result, the HALT state in the IDLE 2 mode requires only a 1/3 of the power consumed in the RUN mode. In the IDLE 2 mode, the system clock is supplied to the following I/O devices:

- o 8-bit timer
- o 16-bit timer
- o Serial interface
- o Watchdog timer

3.4.4 STOP mode

Fig. 3.4 (4) is a timing chart for releasing the HALT state by interrupts in the STOP mode.
The STOP mode is selected to stop all internal circuits including the internal oscillator. In this mode, all pins except special ones are put in the high-impedance state, independent of the internal operation of the MCU. Table 3.4 summarizes the state of these pins in the STOP mode. Note, however, that the pre-halt state can be retained by setting the internal I/O register DRVE (Drive enable: Bit 0 of memory address FFD2H) to "1". The content of this register is initialized to "0" by resetting.
When the CPU accepts an interrupt request, the internal oscillator is restarted immediately. However, to stabilize the oscillation, the system clock starts its output after the time set by the warming up counter WARM (Warming up: Bit 4 of memory address FFD2H) has passed. A warming-up time of either the clock oscillation time $\times 2^{14}$ or 2^{16} can be set by setting this bit to either "0" or "1". This bit is initialized to "0" by resetting.

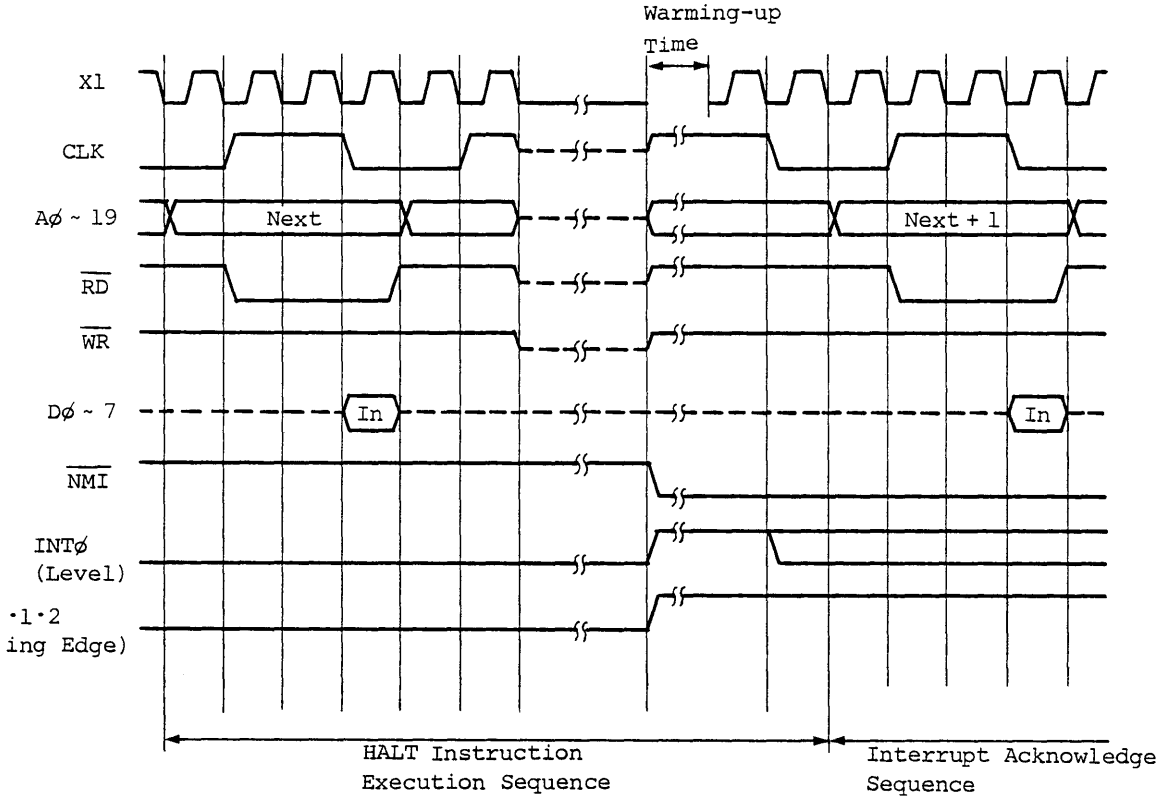


Fig. 3.4 (4) Timing Chart of HALT Released by Interrupt in STOP 2 Mode

The internal oscillator can be also restarted by the input of the RESET signal at "0" to the CPU. In this case, however, the warming-up counter remains inactive because the power is turned on too quickly. As a result, the normal clock operation may not be performed due to the unstable clock supplied immediately after restarting the internal oscillator. To avoid this, it is necessary to keep the RESET signal at "0' long enough to release the HALT state in the STOP mode.

Table 3.4 State of Pins in STOP Mode
(DRVE bit is set to "0")

Pin Name	State
P00 - P07/D0 - 7	High impedance
P10 - P17/A0 - 7	High impedance
P20 - P27/A8 - 15	High impedance
P30 - P37	High impedance
P40 - P43/A16 - 19	High impedance
P50 - P55	High impedance
P60 - P63	Pre-HALT state
P70 - P73	Pre-HALT state
P80	Ready for input
P81	High impedance (Note)
P82	High impedance (Note)
P83	High impedance
NMI	Ready for input
CLK	High impedance
EA	Ready for input
RESET	Ready for input
X1	High impedance
X2	"1"

(Note) P81 and P82 are pulled up slightly in the zero-cross detection mode.

3.5 Functions of Ports

The TMP90C840 contains total 54 bits input/output ports. These ports function not only for the general-purpose I/O but also for the input/output of the internal CPU and I/O devices. Table 3.5 describes the functions of these ports.

Table 3.5 Functions of Ports

Port name	Pin name	No. of pins	Direction	Direction set unit	Pin name for internal function
Port 0	P00 - P07	8	I/O	Byte	D0 - D7
Port 1	P10 - P17	8	I/O	Byte	A0 - A7
Port 2	P20 - P27	8	I/O	Bit	A8 - A15
Port 3	P30	1	Input	-	RxD
	P31	1	Input	-	RxD
	P32	1	Output	-	TxD/RTS/SCLK
	P33	1	Output	-	TxD
	P34	1	Input	-	CTS
	P35	1	Output	-	RD
	P36	1	Output	-	WR
	P37	1	Input	-	WAIT
Port 4	P40 - P43	4	Output	-	A16 - A19
Port 5	P50 - P55	6	Input	-	AN0 - AN5
Port 6	P60 - P63	4	I/O	Bit	M00 - M03/T01
Port 7	P70 - P73	4	I/O	Bit	M10 - M13/T03
Port 8	P80	1	Input	-	INT0
	P81	1	Input	-	INT1/TI4
	P82	1	Input	-	INT2/TI5
	P83	1	Output	-	T03/T04

These port pins function as the general-purpose input/output ports by resetting. The port pins, for which input or output is programmably selectable, function as input ports by resetting. A separate program is required to use them for an internal function.
The TMP90C841 functions in the same way as the TMP90C840 except:

- Port 0 always functions as a data bus (D0 to D7).
- Port 1 always functions as Address bus (A0 to A7).
- Port 2 always functions as Address bus (A8 to A15).
- P35, P36 and P37 of Port 3 always function as $\overline{RD}$, $\overline{WR}$ and $\overline{WAIT}$ pins, respectively.

3.5.1 Port 0 (P00 - P07)

Port 0 is an 8-bit general-purpose I/O port (P0: memory address FFC0H) whose I/O function is specified by the control register (P0C: bit 0 of memory address FFC2H) for each byte. By resetting all bits of the control register are initialized to "0", whereby Port 0 turns to the input mode, and the contents of the output latch register are undefined.

In addition to the general-purpose I/O port function, it functions as a data bus (D0 - 07). Access of an external memory makes it automatically function as a data bus.

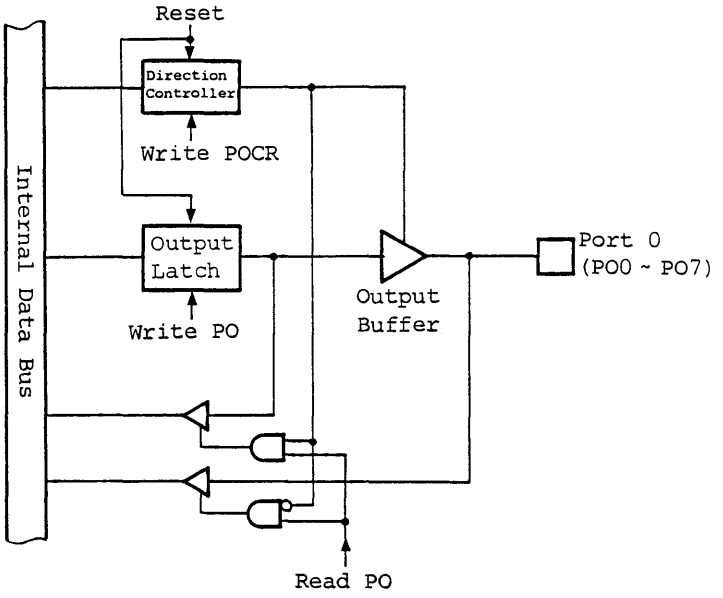


Fig. 3.5 (1) Port 0

3.5.2 Port 1 (P10 - P17)

Port 1 is an 8-bit general-purpose I/O port (PL: memory address FFC1H) whose I/O function is specified by the control register (P1C: bit 1 of memory address FFC2H) for each byte. All bits of the output latch and the control register are initialized to "0" by resetting, whereby Port 1 is put in the input mode.
In addition to the general-purpose I/O port function, it functions as an address bus (A0 - A7). The address bus function can be selected by setting the external extension control register (EXT: bit 2 of memory address FFC2H) to "1" regardless of the status of the above control register (P1C). The EXT register is reset to "0" whereby Port 1 and Port 2 turn to the general-purpose I/O mode.
The EXT register of the TMP90C841 is always set to "1" so that Port 1 functions as an address bus (A0 - A7).

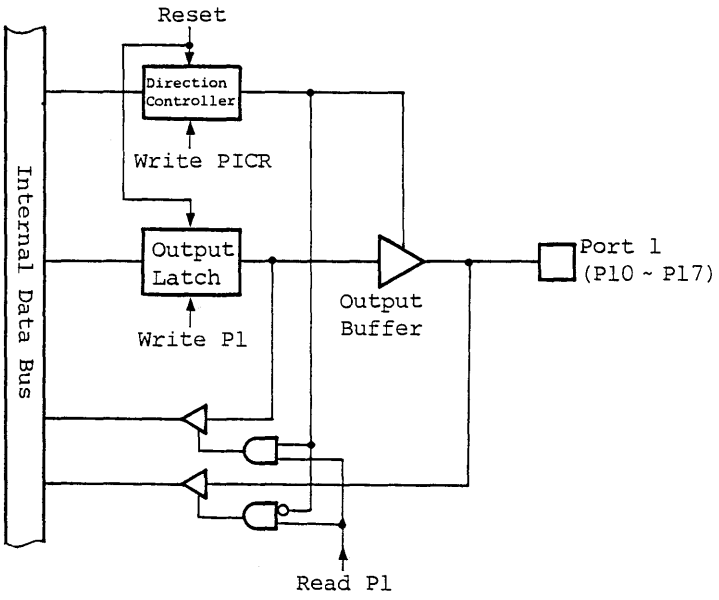


Fig. 3.5 (2) Port 1

Port 0 Register

	P07	P06	P05	P04	P03	P02	P01	P00	
P0 (FFC0H)									R/W

Port 1 Register

	P17	P16	P15	P14	P13	P12	P11	P10	
P1 (FFC1H)									R/W

Port 0/1 Control Register

	-	IRF0	IRFT0	IRFT1	-	EXT	P1C	P0C	
P01CR (FFC2H)		R						W	

See "3.3.3 Interrupt controller".

P0C → Set I/O of Port 0

* 0	Input
1	Output

P1C → Set I/O of Port 1

* 0	Input
1	Output

EXT → Set general-purpose port/address bus of Ports 1 and 2

* 0	General-purpose port
1	Address bus

* indicates initial value after reset.

Fig. 3.5 (3) Registers for Ports 0 and 1

3.5.3 Port 2 (P20 - P27)

Port 2 is an 8-bit general-purpose I/O port (P2: memory address FFC4H) whose I/O functionsis specified by the control register (P2CR : bit 1 of memory address FFC5H) for each bit. All bits of the output latch and the control register are initialized to "0" by resetting , whereby Port 2 turns to the input mode.

In addition to the general-purpose I/O port function, it functions as an address bus (A8 - A15). The address bus function can be selected by setting the EXT register (EXT: bit 2 of memory address FFC2H, shared with port 1) to "1" and setting the Port 2 control register (P2CR) to the output mode. When the Port 2 control register is set to "0", Port 2 functions as an input port, regardless of the status of the EXT register.

For the TMP90C841, all bits of the EXT register and the control register are always set to "1", and Port 2 functions as an address bus (A8 to A15).

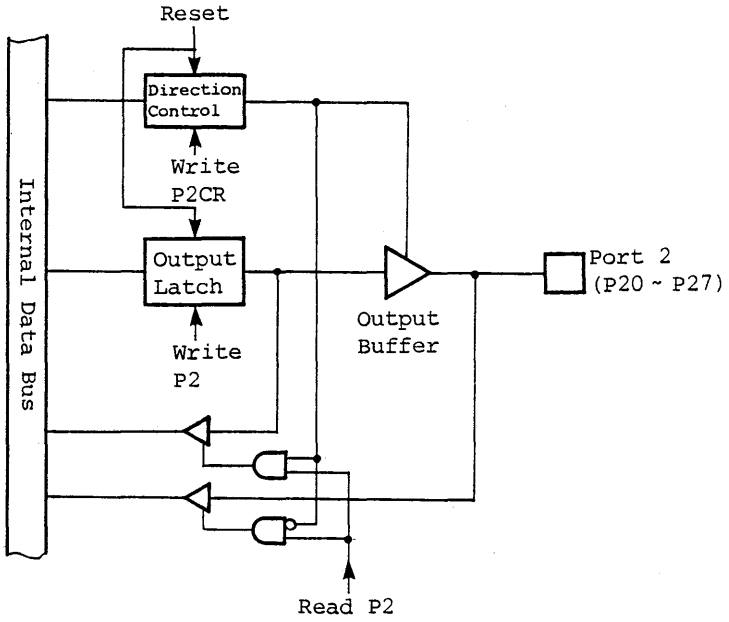


Fig. 3.5 (4) Port 2

Port 2 Register

P2 (FFC4H)	P27	P26	P25	P24	P23	P22	P21	P20	R/W

Port 2 Control Register

	P2CR (FFC5H)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C	W
*	EXT=0	0	Input	Input	Input	Input	Input	Input	Input	Input
	EXT=0	1	Output	Output	Output	Output	Output	Output	Output	Output
	EXT=1	0	Input	Input	Input	Input	Input	Input	Input	Input
	EXT=1	1	A15	A14	A13	A12	A11	A10	A9	A8

* indicates initial value after reset.

Fig. 3.5 (5) Registers for Port 2

3.5.4 Port 3 (P30 - P37)

Port 3 is an 8-bit general-purpose I/O port (P3: memory address FFC6H) with fixed I/O function. All bits of the output latch are set to "1", and "1" is generated to the output port.
In addition to the I/O port function, P30 - P34 have the I/O function for the internal serial interface, while P35 - P37 have the external memory control function. The additional functions can be selected by the control register (P3CR: memory address FFC7H). All bits of the control register are initialized to "0" by resetting, by which the port turns to the general-purpose I/O port mode.
However, access of an external memory makes P35 - P37 automatically function as the memory control pins ($\overline{RD}$, $\overline{WR}$ and $\overline{WAIT}$), and access of an internal memory makes them function as general-purpose I/O ports.
When an external memory is accessed , therefore, the output latch registers P35 ($\overline{RD}$) and P36 ($\overline{WR}$) should be kept at "1" which is the initial value after the reset.
The bit 5 (RDE) of the control register is intended for a pseudostatic RAM. When set to "1", it always functions as an $\overline{RD}$ pin. It is set to "0" by reading the internal I/O unit.

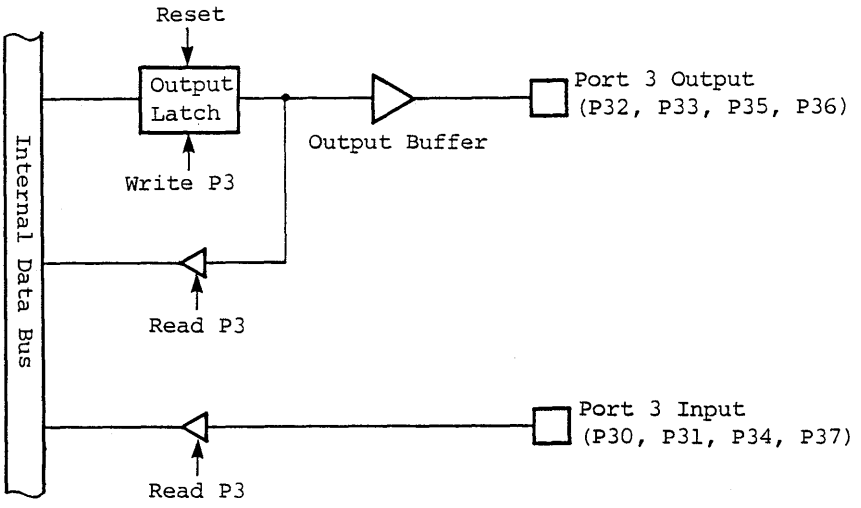


Fig. 3.5 (6) Port 3

Port 3 Register

P3	P37	P36	P35	P34	P33	P32	P31	P30
(FFC6H)	R	R/W	R/W	R	R/W	R/W	R	R

Port 3 Control Register

P3CR (FFC7H)	WAITC	RDE	ODE	TXDC	RXDC
	R/W			R/W	

ODE: See "3.8 Serial Channel".

RDE: R/W — Set port P33 to open drain output

*	0	CMOS output
	1	Open drain output

WAITC: R/W — Set port P35 to fixed $\overline{RD}$ mode

*	0	General-purpose I/O port
	1	Fixed as $\overline{RD}$ pin

TXDC: R/W — WAIT control

*	00	2-state wait
*	01	Normal wait
*	10	No wait
	1	Reserve

* indicates initial value after reset.

Fig. 3.5 (7) Register for Ports 3

3.5.5 Port 4 (P40 - P43)

Port 4 is a 4-bit port (P4: memory address FFC8H) intended only for the output. All bits of the output latch are initialized to "0" by resetting, and "0" is generated from the port.
In addition to the output port function, it works as an address bus (A16 - A19). The selection of the address bus function is made by the control register (P4CR: memory address FFC9H). The output port or address bus function can be selected for each bit. All bits of the control register are initialized to "0" by resetting, by which the port turns to the output mode.

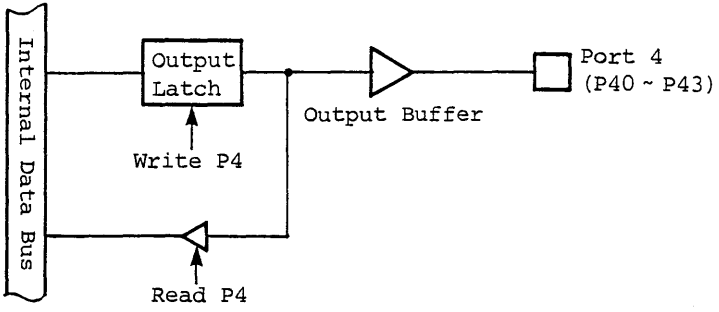


Fig. 3.5 (8) Port 4

Port 4 Register

	7	6	5	4	3	2	1	0	
P4 (FFC8H)	-	-	-	-	P43	P42	P41	P40	R/W

Port 4 Control Register

	7	6	5	4	3	2	1	0	
P4CR (FFC9H)	-	-	-	-	P43C	P42C	P41C	P40C	W

Set output port/ address bus

*	0	Output port
	1	Address bus

* indicates initial value after reset.

Bank register BX

	7	6	5	4	3	2	1	0	
BX (FFECH)	-	-	-	-	BX3	BX2	BX1	BX0	R/W

Address bus A16 - A19

Bank register BY

	7	6	5	4	3	2	1	0	
BY (FFEDH)	-	-	-	-	BY3	BY2	BY1	BY0	R/W

Address bus A16 - A19

Fig. 3.5 (9) Registers for Port 4

3.5.6 Port 5 (P50 - P55)

Port 5 is a 6-bit input port (P5: bit 0 - 5 of memory address FFCAH) and also used as an analog input pin (AN0 - AN5).
Writing data into Port 5 register is prohibited.

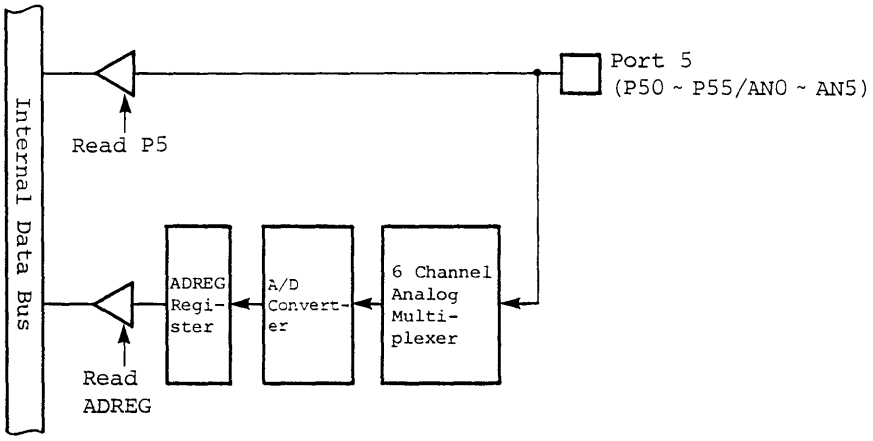


Fig. 3.5 (10) Port 5

Port 5 Register

P5 (FFCAH)	0	1	P55	P54	P53	P52	P51	P50	R

Fig. 3.5 (11) Register for Port 5

3.5.7 Port 6 (P60 - P63)

Port 6 is a 4-bit general-purpose I/O port (P6: memory address FFCCH) whose function is specified by the control register (P67CR: bits 0 - 3 of memory address FFCEH) for each bit. The control register is initialized to "0" by resetting, and Port 6 enters in the input mode. This port is also serviceable as a stepping motor control port channel 0 (M00 - M03), so either the general-purpose I/O or the stepping motor control port can be selected by the control register (SMMOD: bits 0 and 1 of memory address FFCBH). This port is served as the general-purpose I/O port by resetting.

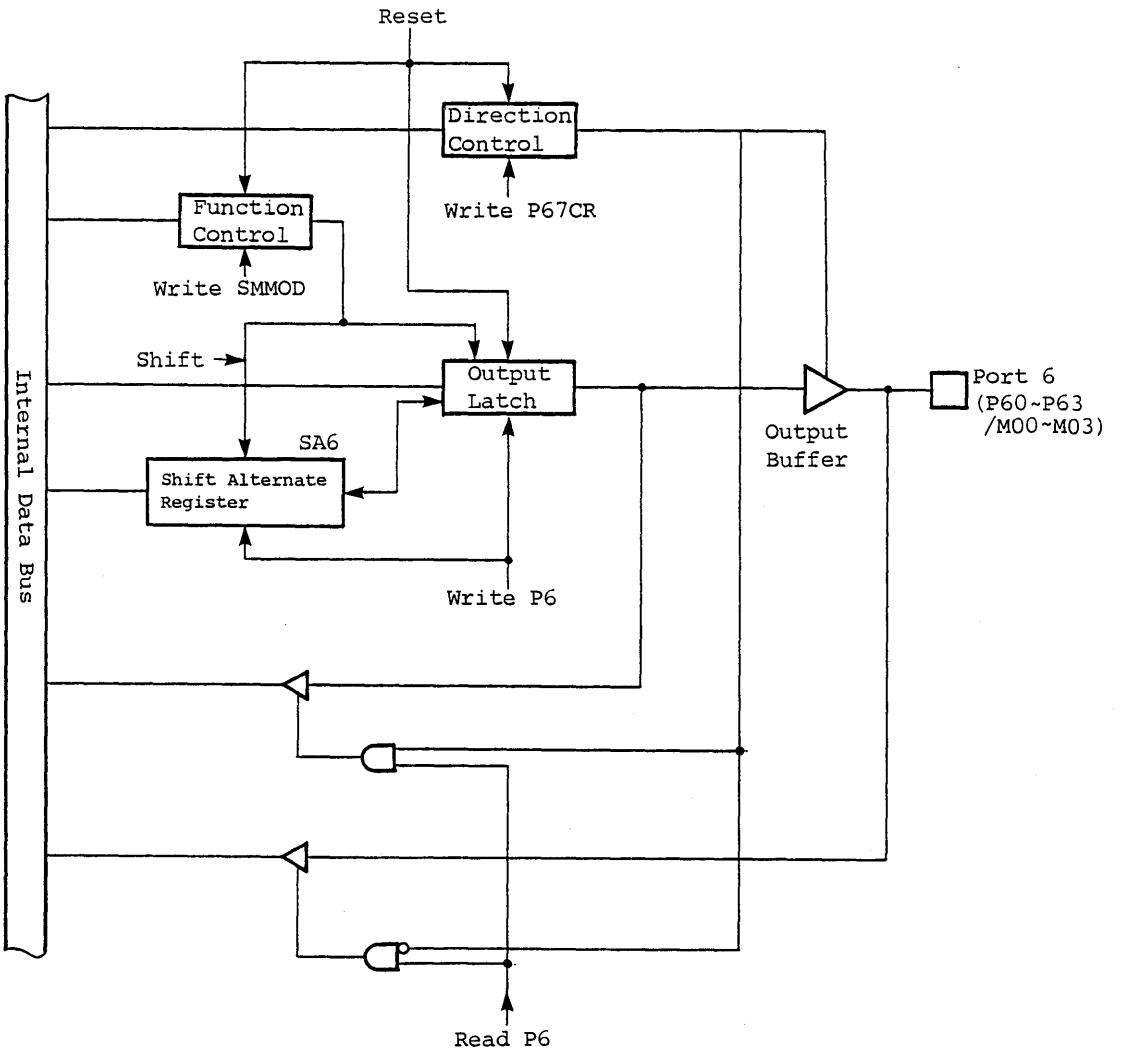
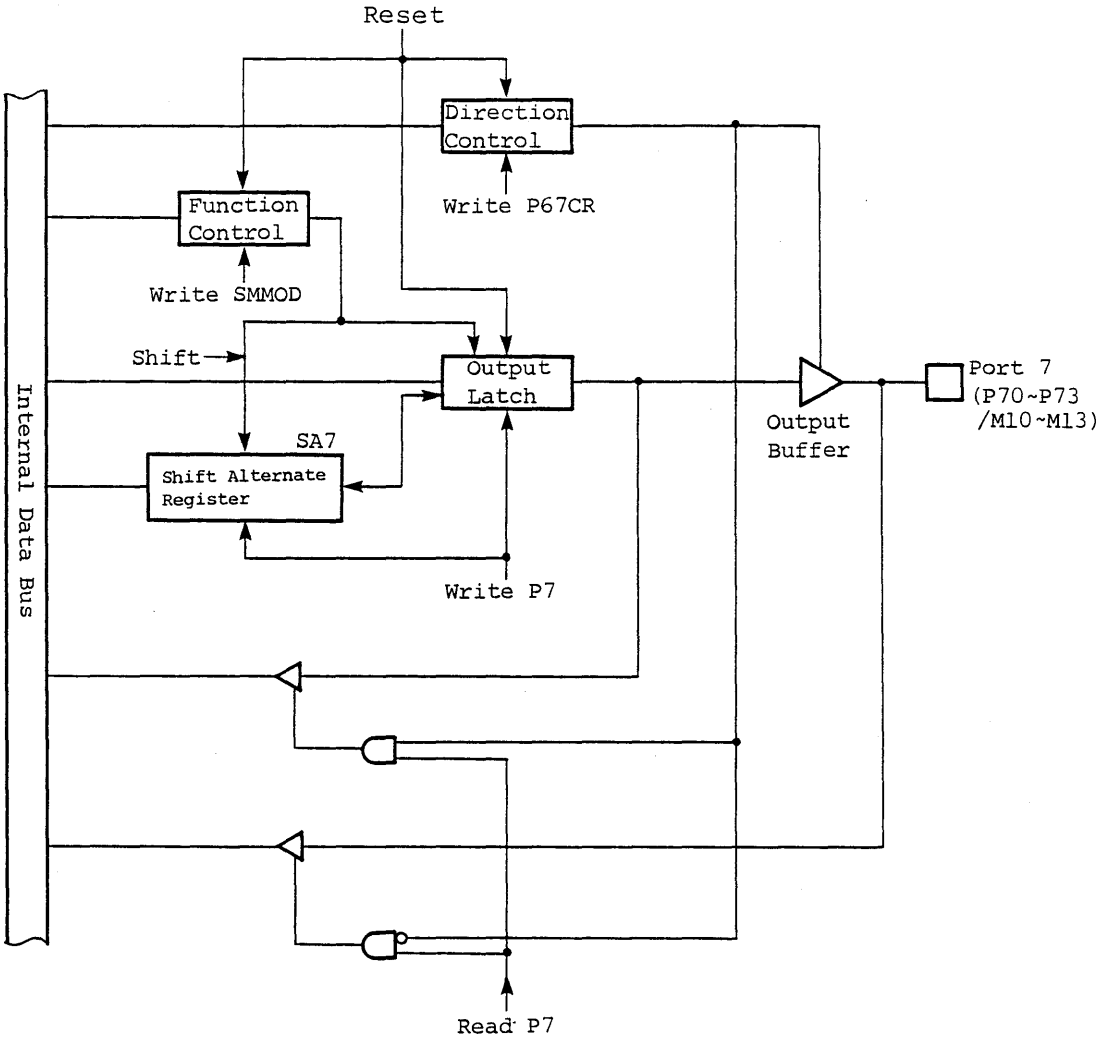


Fig. 3.5 (12) Port 6

3.5.8 Port 7 (P70 - P73)

Port 7 is a 4-bit general-purpose I/O port (P7: memory address FFCDH) whose I/O function is specified by the control register (P67CR: bits 4 - 7 of memory address FFCEH). The control register is initialized to "0" by resetting, whereby Port 7 turns to the input mode.
This port is also serviceable as a stepping motor control port channel 1 (M10 - M13), so either the general-purpose I/O or the stepping motor control port can be selected by the control register (SMMOD: bits 4 and 5 of memory address FFCBH). It is served as the general-purpose I/O port by resetting.



Figm 3.5 (13) Port 7

Port 6 Register

P6 (FFCCH)	SA63	SA62	SA61	SA60	P63	P62	P61	P60
	W				R/W			

Stepping motor control port
Channel 0 shifter alternate register

Port 6

Port 7 Register

P7 (FFCDH)	SA73	SA72	SA71	SA70	P73	P72	P71	P70
	W				R/W			

Stepping motor control port
Channel 1 shifter alternate register

Port 7

Port 6 and 7 Register

P67CR (FFCEH)	P73C	P72C	P71C	P70C	P63C	P62C	P61C	P60C

W → Select input/output of port 6

*	0	Input
	1	Output

W → Select input/output of port 7

*	0	Input
	1	Output

* indicates initial value after reset.

Fig. 3.5 (14a) Registers for Ports 6 and 7

Stepping motor mode register

SMM0D (FFC8H)	SM7M	P70C	SM6M	P60C

See "3.7 Stepping motor control port".

R/W → Select function of Port 6. (Port 6 Output Control)

		P63	P62	P61	P60	Shift trigger
*	00	IN/OUT	IN/OUT	IN/OUT	IN/OUT	-
	01	IN/OUT	IN/OUT	IN/OUT	IN/TO1	-
	10	IN/M03	IN/M02	IN/M01	IN/M00	Timer 0,1
	11					

Shift trigger signal output timer for stepping motor control

R/W → Select function of Port 7. (Port 7 Output Control)

		P73	P72	P71	P70	Shift trigger
*	00	IN/OUT	IN/OUT	IN/OUT	IN/OUT	-
	01	IN/OUT	IN/OUT	IN/OUT	IN/TO3	-
	10	IN/M13	IN/M12	IN/M11	IN/M10	Timer 2,3
	11					Timer 4

Shift trigger signal output timer for stepping motor control

* indicates initial value after reset.

(Note) SM6M and SM7M are included in Fig. 3.7.

Fig. 3.5 (14b) Register for Port 6 and 7

3.5.9 Port 8 (P80 - P83)

Port 8 is a 4-bit general-purpose I/O port (P8: memory address FFD0H), with P80 to P82 intended only for the input and P83 only for the output. The output latch of P83 is reset to "0", whereby the output level turns to "L".
Port 8 also has the functions of interrupt request input, clock input for a timer/event counter, and timer output.

(1) P80/INT0
P80 is a general-purpose input port, also used as the external interrupt request input pin INT0. INT0 allows the selection of either an "H" level interrupt or rising edge interrupt by using the control register (P8CR: Bit 0 of memory address FFD1H).

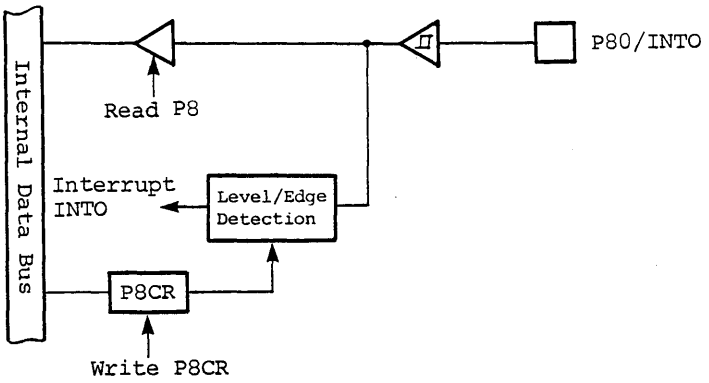


Fig. 3.5 (15) Port P80/INT0

(2) P81/INT1/TI4
P81 is a general-purpose input port, also used as the external interrupt request input pin INT1 and the clock input pin TI4 for the timer/event counter.
This port incorporates a zero-cross detection circuit, and enables zero-cross detection by connecting an external capacitor. The zero-cross detection can be disabled/enabled by using the control register (P8CR: Bit 1 of memory address FFD1H). This control register is reset to "0", making the zero-cross detection disabled.

(3) P82/INT2/TI5
Like P81, P82 is a general-purpose input port, also used as the external interrupt request input pin INT2 and the clock input pin TI5 for the timer/event counter. This port also contains a zero-cross detection circuit, and disables/enables the zero-cross detection by using the control register (P8CR: Bit 2 of memory address FFD1H). When this control register is reset, the zero-cross detection is disabled.

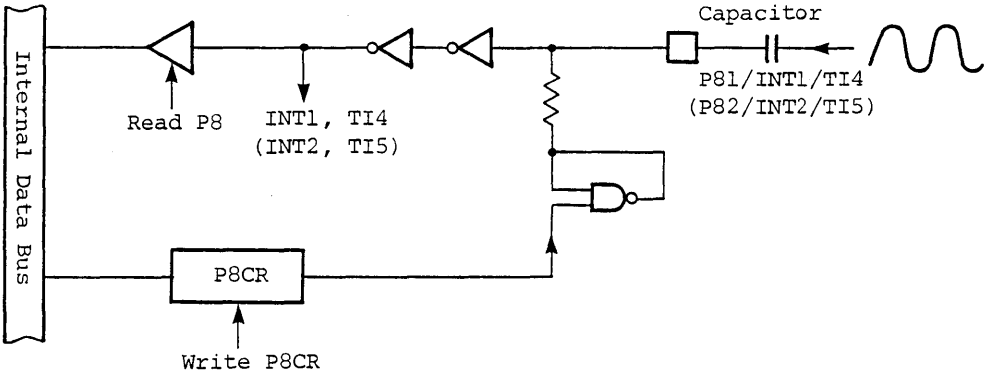


Fig. 3.5 (16) Port P81/INT1/TI4 and P82/INT2/TI5

(4) P83/T03/T04

P83 is a general-purpose output port, also used as the timer output pins T03/T04. Either function can be selected by using two control registers (P83C: bit 3 of memory address FFD1H; SMMODs 5 and 4: bits 5 and 4 of FFCBH).

	P830C	SMMOD5,4	Function
*	0	XX	P83 output port
	1	0X	T04
	1	10	
	1	11	T03

* indicates initial value after reset.

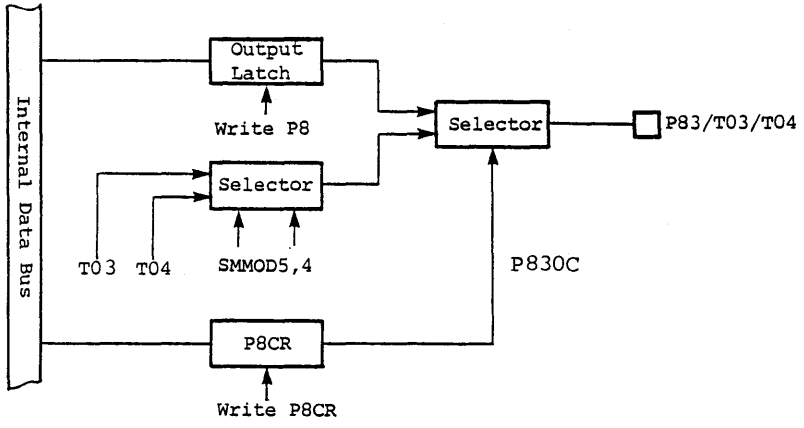


Fig. 3.5 (17) Port P83/T03/T04

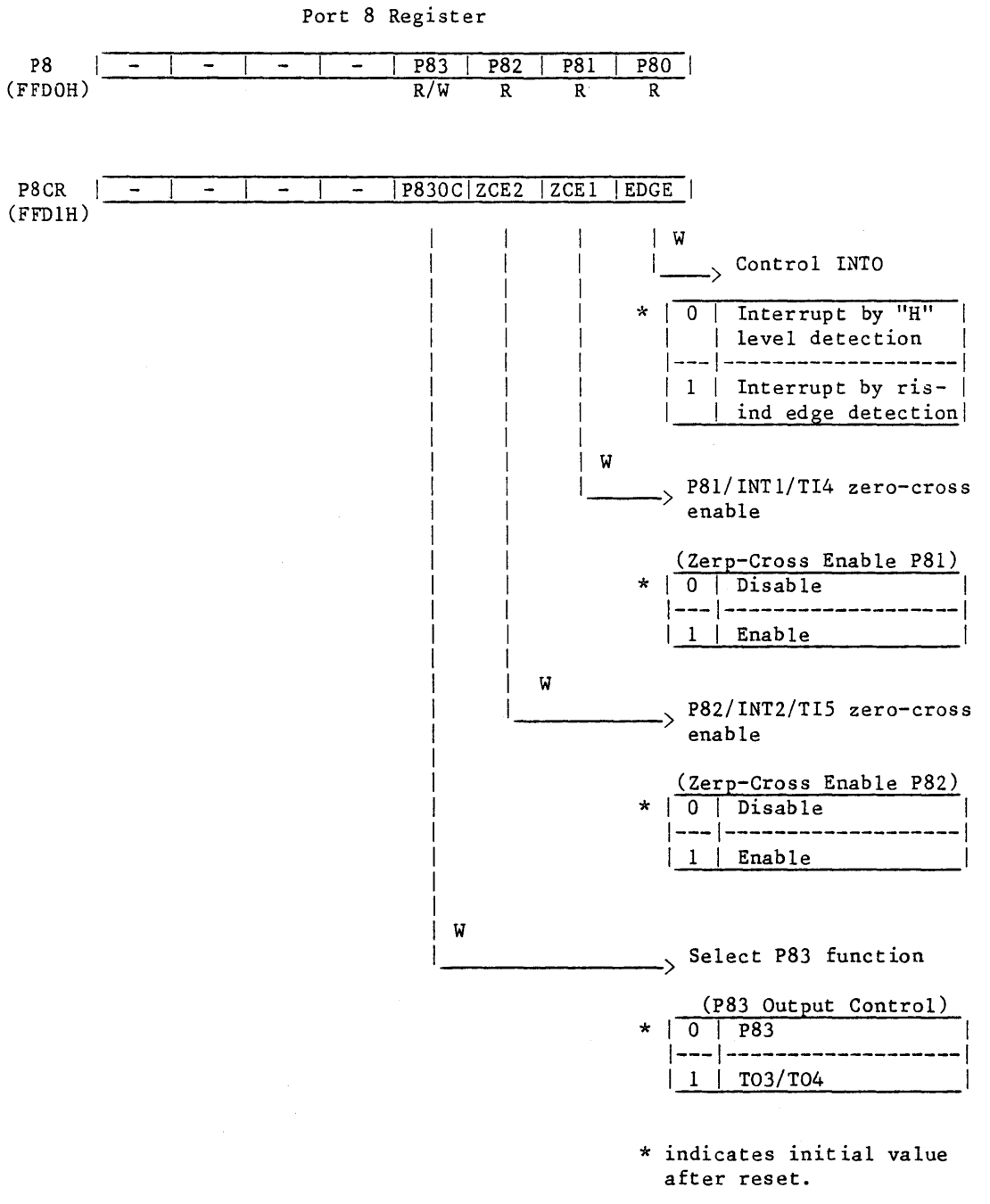


FIG. 3.5 (18) Registers for Port 8

3.6 Timers

The TMP90C840 incorporates four 8-bit timers and a 16-bit multi-function timer/event counter.

The four 8-bit timers can operate independently, and also function as two 16-bit timers through cascade connection. The following four operating modes are provided for the 8-bit timers:

- o 8-bit interval timer mode (4 timers)
- o 16-bit interval timer mode (2 timers)
  Possible arrangements: 8-bit × 2 and 16-bit × 1, or 16-bit × 2
- o 8-bit programmable square wave (pulse) generation (PPG) mode (2 timers)
- o 8-bit pulse width modulation (PWM) mode (2 timers)

The 16-bit multi-function timer/event counter can operate in the following six modes:

- o 16-bit interval timer mode
- o 16-bit event counter mode
- o 16-bit programmable square wave (pulse) generation (PPG) mode
- o Frequency measurement mode
- o Pulse width measurement mode
- o Timer deviation measurement mode

3.6.1 8-bit timers

The TMP90C840 incorporates four 8-bit interval timers (Timers 0, 1, 2 and 3), each of which can be operated independently. The cascade connection of Timer 0 and 1, or Timer 2 and 3 allows these timers used as 16-bit internal timers.
Fig. 3.6 (1) is a block diagram of the 8-bit timers (Timer 0 and Timer 1).
Timer 2 and 3 have the same circuit configuration as Timer 0 and Timer 1 respectively.
Each interval timer is composed of an 8-bit up-counter, an 8-bit comparator and an 8-bit timer/register, with a Timer Flip-flop (TFF1/TFF3) provided to each pair of Timer 0 and Timer 1 and Timer 2 and Timer 3.
Internal clocks (øT1, øT16 and øT256), some of the input clock sources for the interval timers, are generated by the 9-bit prescaler shown in Fig. 3.6 (2).
Their operating modes of the 8-bit timers and flip-flops are controlled by four control registers (TCLK, TFFCR, TMOD and TRUN).

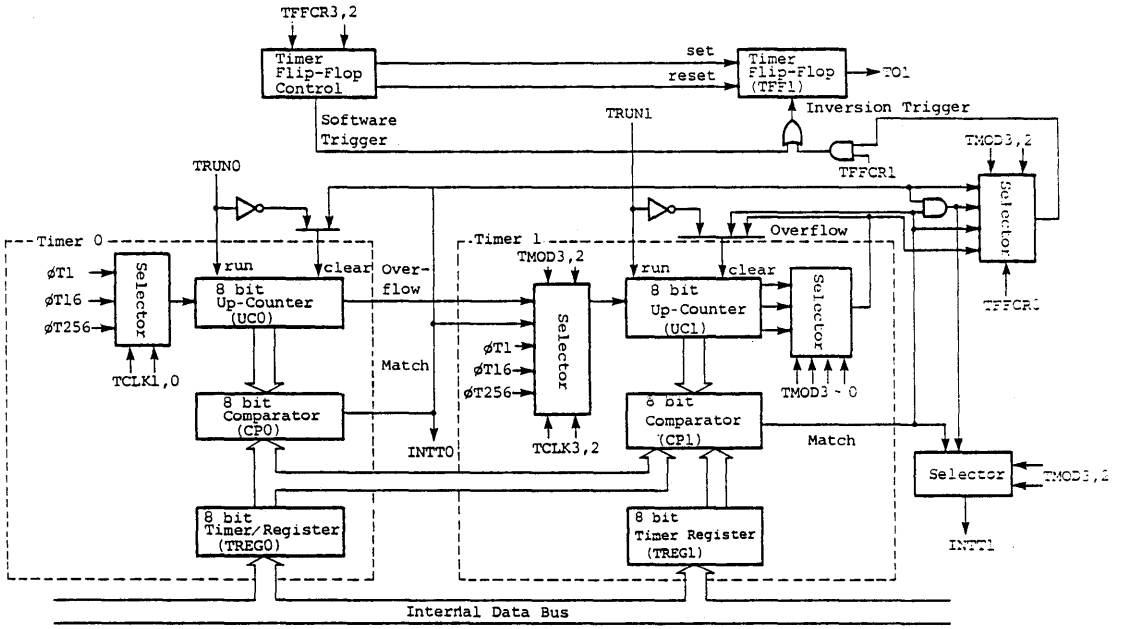


Fig. 3.6 (1) Block Diagram of 8-bit Timers (Timers 0 and 1)

① Prescaler

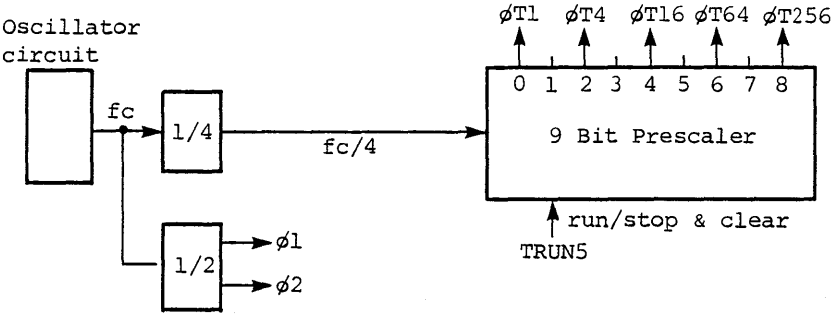
An 9-bit prescaler is provided to further devide the clock frequency already divided to a 1/4 of the frequency of the source clock (fc). It generates a input clock pulse for the 8-bit timers, 16-bit timer/event counter, the baud-rate generator, etc.

For the 8-bit timers, three types of clock are generated (oT1, oT16 and oT256).

The prescaler can be run or stopped by using the 5th bit TRUN5 (PRRUN) of the timer control register (TRUN). Setting TRUN5 to "1" makes the prescaler count, and setting it to "0" clears the prescaler to stop. By resetting, TRUN5 is initialized to "0", making the prescaler cleare and stop.

Cycle

fc / Input clock	8MHz	10MHz
oT1	1.0μs	0.8μs
oT16	16μs	12.8μs
oT356	256μs	204.8μs



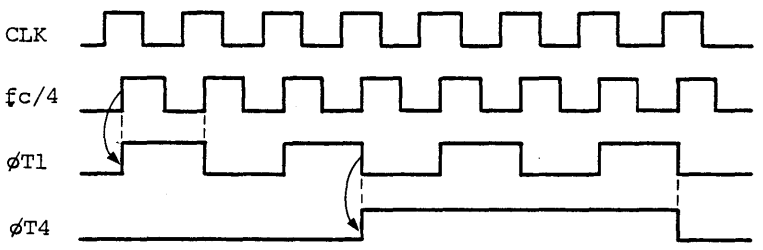


Fig. 3.6 (2) Prescaler

② Up-counter

This is an 8-bit binary counter that counts up by an input clock pulse specified by an 8-bit timer clock control register (TCLK) and an 8-bit timer mode register (TMOD).

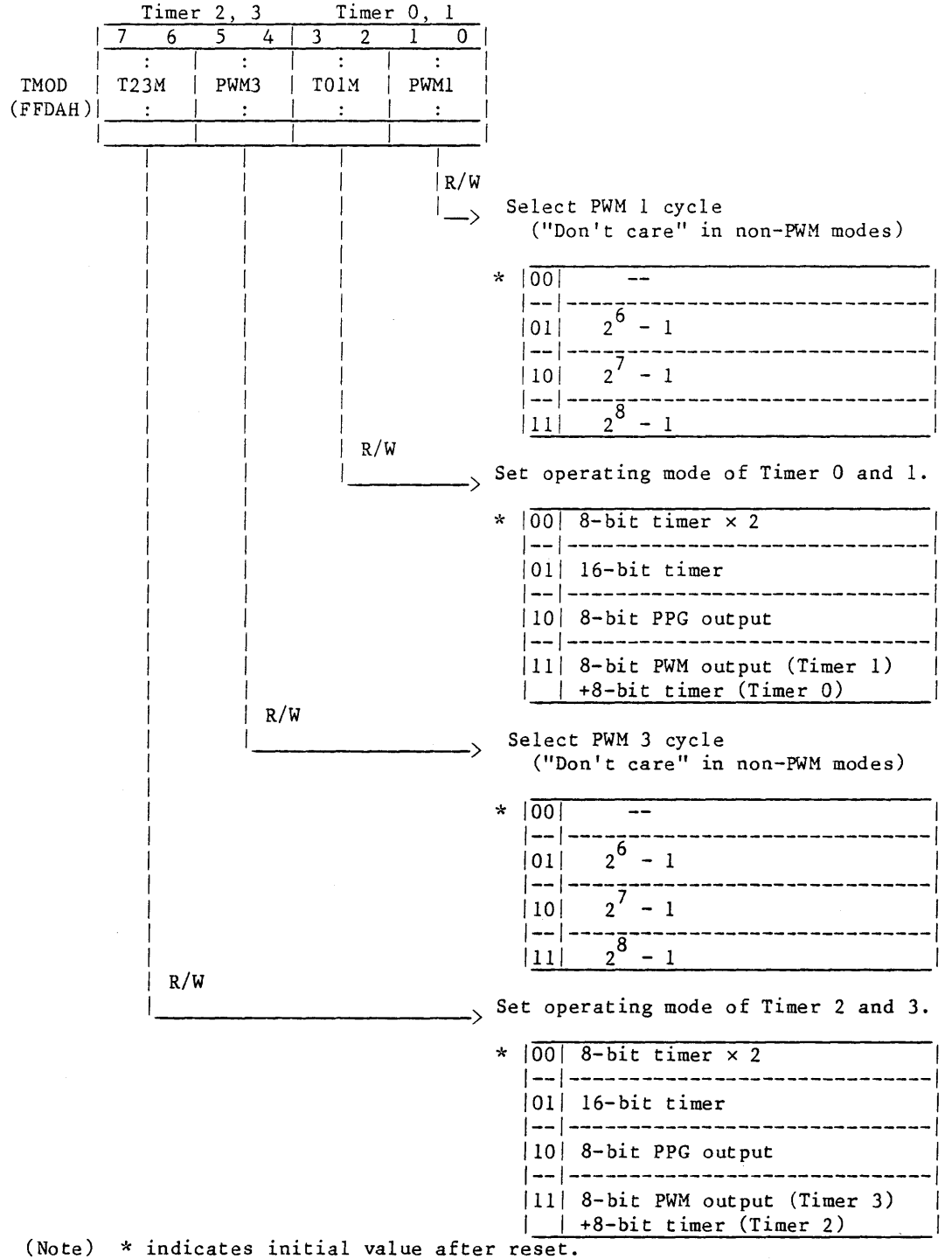
The input clock pulse for Timer 0 and 2 is selected from ϕT1, ϕT16 and ϕT256 according to the setting of the TCLK register.
However, ϕT256 cannot be selected as the input clock pulse for these timers in the 16-bit timer mode (TMOD3,2=01/TMOD7,6=01).

Example: When setting TCLK1, 0 = "01", ϕT1 is selected as the input clock pulse for Timer 0.

The input clock pulse to Timer 1 and 3 is selected according to the operating mode. In the 16-bit timer mode, the overflow output of Timer 0 and 2 is automatically selected, regardless of the setting of the TCLK register.
In the other operating modes, the clock pulse is selected among the internal clocks ϕT1, ϕT16 and ϕT256, and the output of the Timer 0 and 2 comparator (match signal).

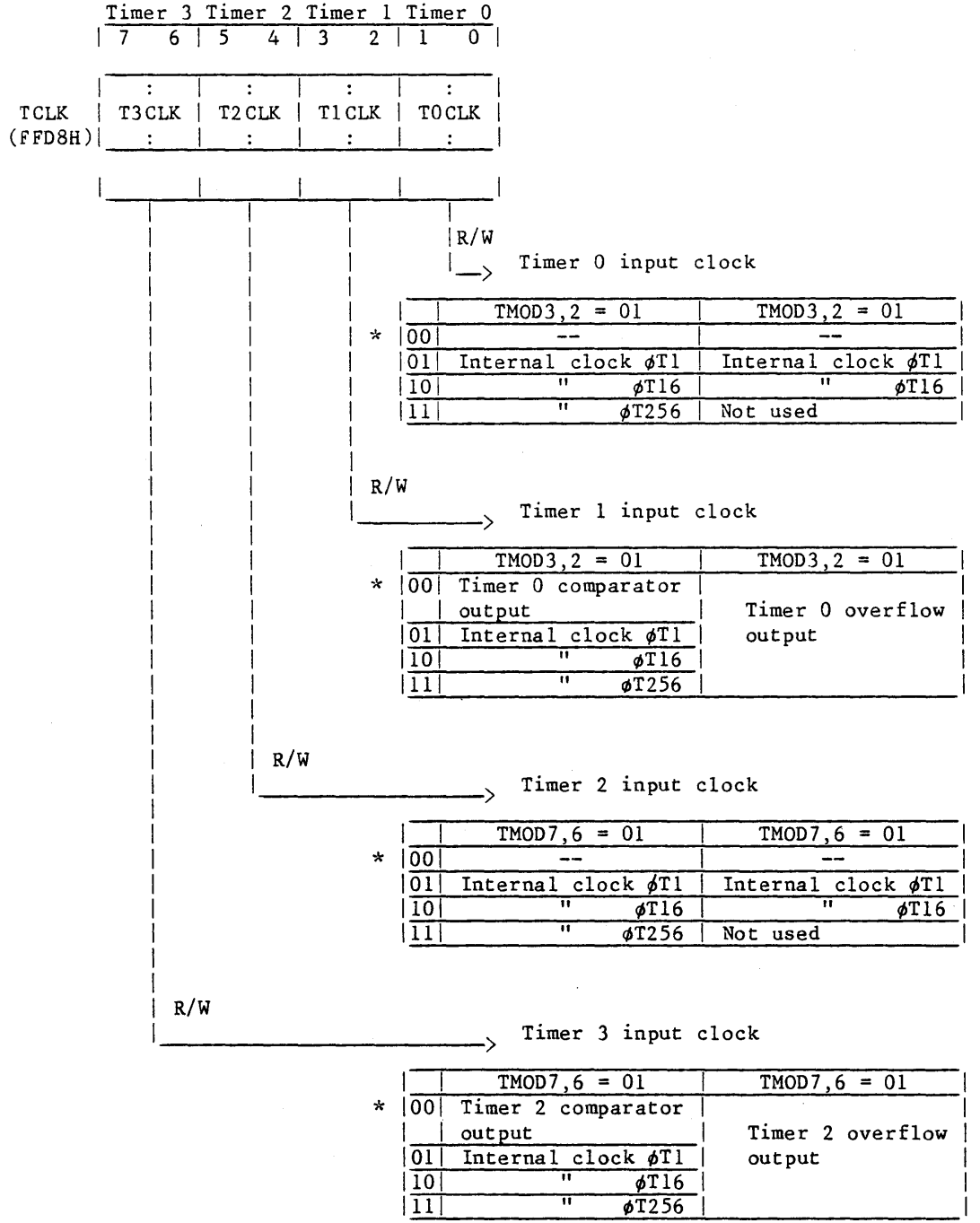
Example: If TMOD3,2 = 01, the overflow output of Timer 0 is selected as the input clock to Timer 1.
If TMOD3,2 = 00 and TCLK3,2 = 01, ϕT1 is selected as the input clock to Timer 1.

The operating mode is selected by the TMOD register. This register is initialized to TMOD3,2=00/TMOD7,6=00 by resetting, whereby the up-counter is placed in the 8-bit timer mode.
The up-counter can control its functions, count, stop or clear, for each interval timer as selected by the timer control register TRUN.
By resetting, all up-counters are cleared to stop the timers.



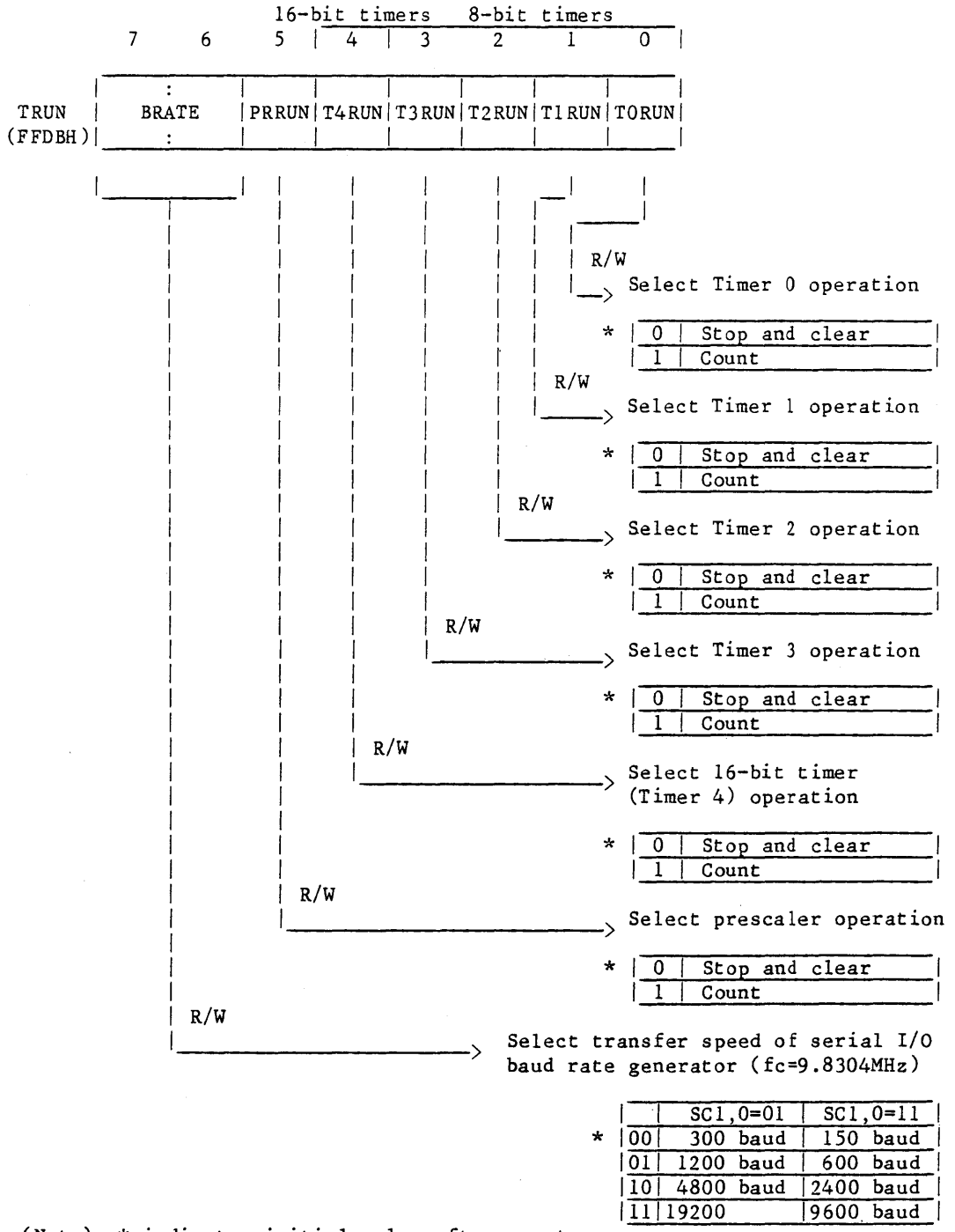
(Note) * indicates initial value after reset.
R/W denotes that read or write is possible.

Fig. 3.5 (3) 8-bit Timer Mode register (TMOD)



(Note) * indicates initial value after reset.

Fig. 3.6 (4) 8-bit Timer Clock Control Register (TCLK)



(Note) * indicates initial value after reset.
R/W denotes that read or write is possible

Fig. 3.6 (5) Timer/Serial Channel Control Registers (TRUN)

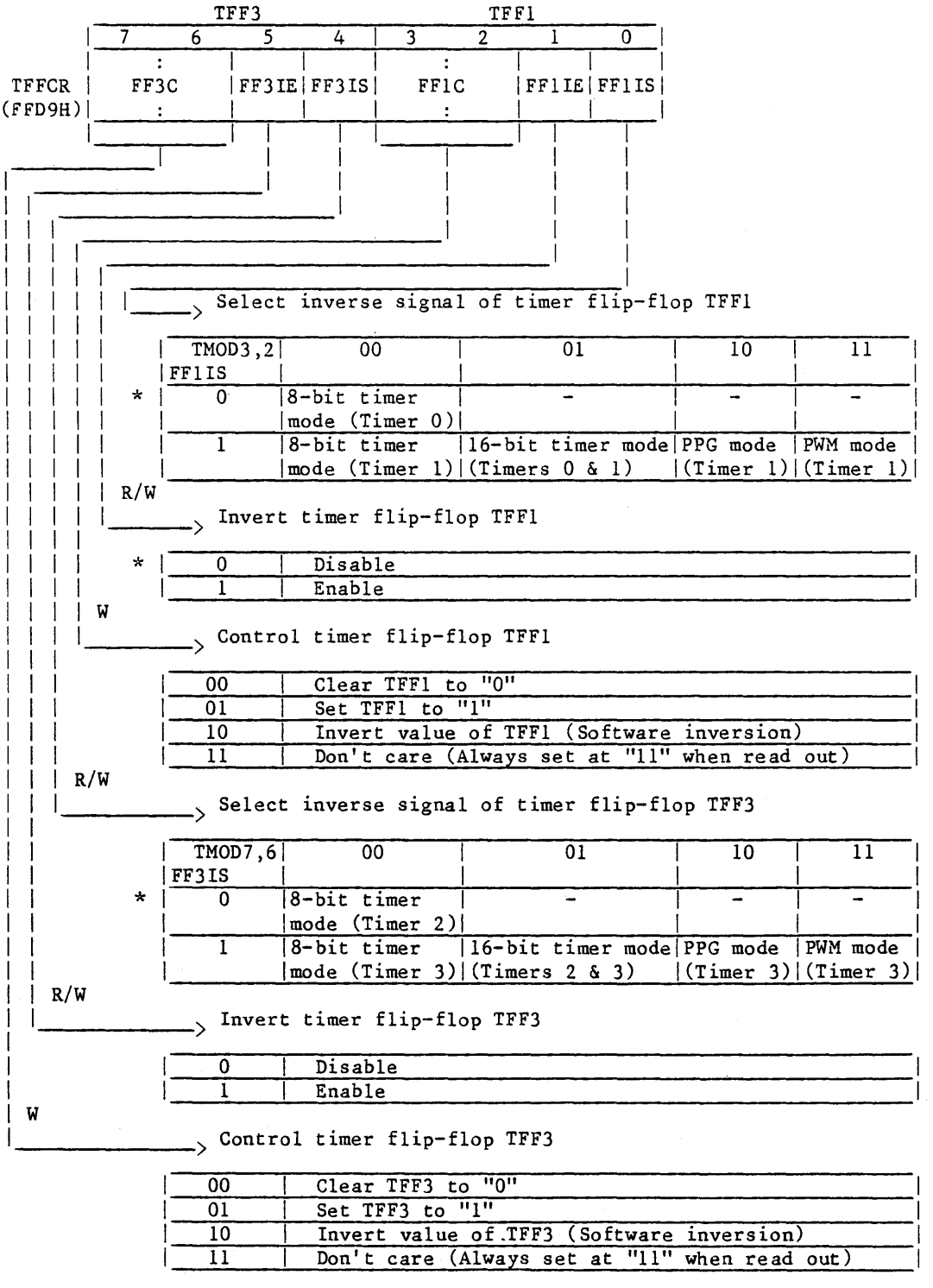


Fig. 3.6 (6) 8-Bit Timer Flip-Flop Control Register (TFFCR)

③ Timer registers
8-bit registers are provided to set the interval time. When the set value of a timer register matches that of an up-counter, the match signal of their comparators turn to the active mode. If "00H" is set, this signal becomes active when the up-counter overflows.

④ Comparators
A comparator compares the values in an up-counter and a timer register. When they matches, the comparator clears the up-counter to "0", and generates an interrupt signal (INTTn). If the timer flip-flop inversion is enabled by the Timer Flip-Flop control register, the comparator inverts the Timer Flip-flop.

⑤ Timer flip-flops (Timer F/Fs)
The status of the Timer Flip-flop is inverted by the match signal (output by comparator). Its status can be output to the timer output pin TO1 (also used as P60) and TO3 (used as P70 or P80).
A Timer F/F is provided to each of the timer pairs Timer 0 - Timer 1 and Timern 2 - Timer 3, and is called TFF1 and TFF3, respectively. The status of TFF1 is output to TO1, and that of TFF3 to TO3. TO3 has 2 pins (P70 and P83). P83 is selected only when the port 7 is used as a stepping motor control port.
The Timer F/Fs are controlled by a timer flip-flop control register (TFFCR).

In the case of TFF1 (timer F/F for the Timer 0 and Timer 1), the flip-flop operation is described as follows:
TFFCR0 (FF1IS) selects the signal for inversion of TFF1. In the 8-bit timer mode, inversion is enabled by the match signal from Timer 0 if this bit is set to "1", or by the signal from Timer 1 if set to "0". In any other mode, FF1IS must be always set to "1". It is initialized to "0" by resetting.
TFFCR1 (FF1IE) controls the inversion of TFF1. Setting this bit to "1" enables the inversion and setting it to "0" disable.
FF1IE is initialized to "0" by resetting.
The bits TFFCR3 and TFFCR2 (FF1C) are used to set/reset TFF1 or enable its inversion by software. TFF1 is reset by writing "00", set by "01" and inverted by "10".
Similarly, TFF3 is controlled by TFFCR7 - 4.

The 8-bit timers operate as follows:

(1) 8-bit timer mode
The four interval timers 0, 1, 2 and 3 can operate independently as an 8-bit interval timer. Only the operation of Timer 1 is described because their operations are the same.

① Generating interrupts at specified intervals
Periodic interrupts can be generated by using Timer 1 (INTT1) in the following procedure: Stop Timer 1, set the desired operating mode, input clock and cycle time in, the registers TMOD, TCLK and TREG1 enable INTT1, and start the counting of Timer 1.

Example: To generate Timer 1 interrupt every 4.0μs at fc=10 MHz, the registers should be set as follows:

```
          MSB              LSB
            7 6 5 4 3 2 1 0
 TRUN   <- - - - - - - - 0 -   Stop Timer 1, and clear it to "0".
 TMOD   <- - - - - 0 0 x x   Set the 8-bit timer mode.
 TCLK   <- - - - - 0 1 - -   Select 0T1 (0.8μs @fc=10 MHz) as the
                             input clock.
 TREG1  <- 0 0 1 1 0 0 1 0   Set the timer register at 40μs/T1 = 50.
 INTEH  <- - - - - - - - 1   Enable INTT1.
 IRUN   <- - - 1 - - - 1 -   Start Timer 1.
```

(Note) x: Don't care - : No change

Use the following table for selecting the input clock:

Table 3.6 (1) 8-bit timer interrupt cycle and input clock

Interrupt cycle @fc=10 MHz	Resolution	Input clock
0.8μs - 204μs	0.8μs	øT1
12.8μs - 3.264ms	12.8μs	øT16
204.8μs - 52.429ms	204.8μs	øT256

Caution for using Timer 2

Interrupts generated by Timer 2 (INTT2) uses the same interrupt mask flag (INTEL 7) as used for those of the A/D converter (INTAD). To select either interrupt, another flag INTEH3 is provided. Setting this flag to "0" enables interrupts by Timer 2 and disables those by the A/D converter.

[2] Generating pulse at 50% duty

The Timer Flip-flop is inverted at specified intervals, and its status is output to a timer output pin (TO1 or TO3).

Example: To output pulse from TO1 at fc=10 MHz every 4.8μs, the registers should be set as follows:
This example uses Timer 1, but the same operation can be effected by using Timer 0.

```
            7 6 5 4 3 2 1 0
 TRUN   <- - - - - - - - 0 -   Stop Timer 1, and clear it to "0".
 TMOD   <- - - - - 0 0 x x   Set the 8-bit timer mode.
 TCLK   <- - - - - 0 1 - -   Select øT1 as the input clock.
 TREG1  <- 0 0 0 0 0 0 1 0   Set the timer register at 4.8μs/øT1/2=3.
 TFFCR  <- - - - - 0 0 1 1   Clear TFF1 to "0", and set to invert by
                             the match signal from Timer 1.
 SMMOD  <- - - - - x x 0 1   Select P60 as TO1 pin.
 P67CR  <- - - - - - - - 1   Select P60 as TO1 pin.
 TRUN   <- - - 1 - - - 1 -   Start Timer 1.
```

(Note) x: Don't care - : No change

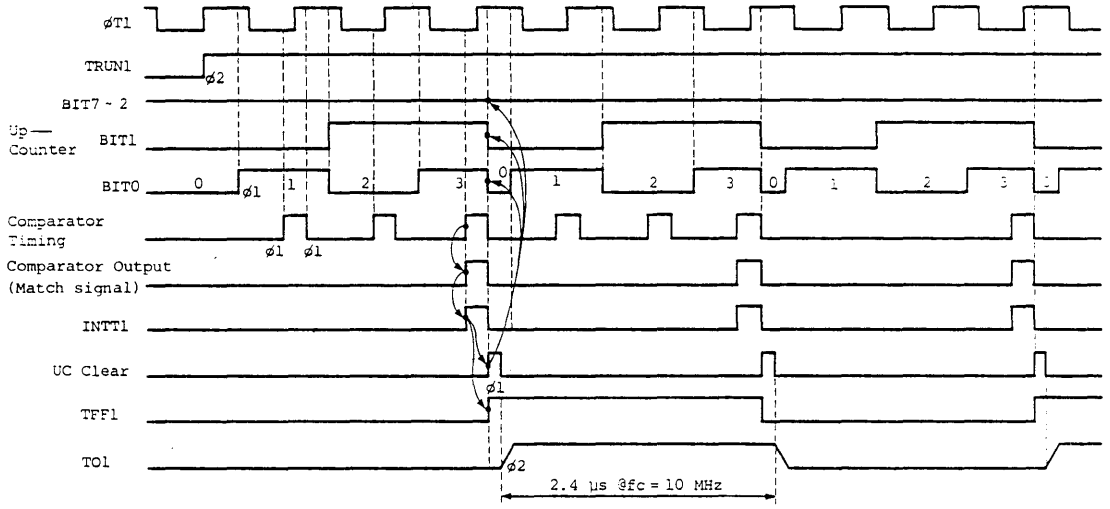


Fig. 3.6 (7) Pulse Output (50% duty) Timing Chart

③ Making Timer 1 count up by match signal from Timer 0 comparator.
Select the 8-bit timer mode, and the set output of Timer 0 comparator as the input clock to Timer 1.

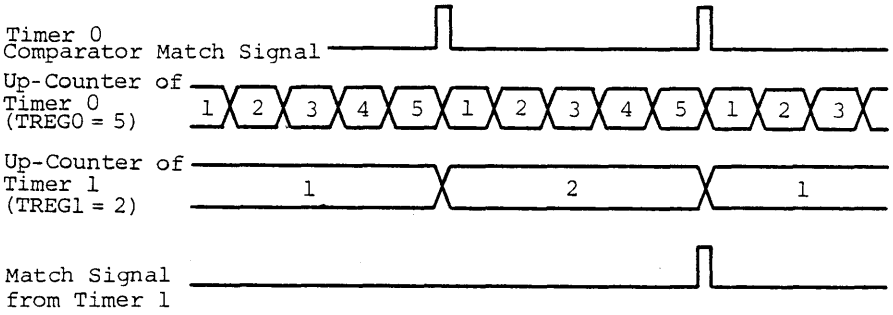


Fig. 3.6 (8)

④ Software inversion
The timer flip-flops can be inverted by software independent of the timer operation.
Writing "10" into the bits TFFCR3 and TFFCR2 inverts TFF1, and writing the same into TFFCR7 and TFFCR6 inverts TFF3.

⑤ Initial setting of Timer Flip-Flops
The Timer Flip-flops can be initialized to either "0" or "1" without regard to the timer operation.
TFF1 is initialized to "0" by writing "00" into TFFCR3 and TFFCR2, and "1" by writing "01" into these bits.

(Note) Reading the data from the Timer Flip-flops and timer registers is prohibited.

(2) 16-bit timer mode

The Timer 0 and Timer 1 or Timer 2 and Timer 3 can be used as one 16-bit interval timer.
As both timer pairs have the same function, only the timer pair Timer 0 and Timer 1 is discussed.
Cascade connection of Timer 0 and Timer 1 to use them as a 16-bit interval timer requires to set the bits 3 and 2 of the mode register TMOD to "01".
By selecting the 16-bit timer mode, the overflow output of Timer 0 is automatically selected as the input clock to Timer 1, regardless of the set value of the clock control register TCLK. The input clock to Timer 0 is selected by TCLK. Note, however, that øT256 cannot be selected in the 16-bit timer mode. Table 3.6 (2) shows the combinations of timer (interrupt) cycle and input clock.

Table 3.6 (2) 16-bit timer (interrupt) cycle and input clock

Timer (interrupt) cycle @fc=10 MHz	Resolution	Input clock to Timer 0
12.8μs - 52.43ms	0.8μs	øT1
12.8μs - 838.86ms	12.8μs	øT16

The timer (interrupt) cycle is selected by the lower eight bits set by TREG0 and the upper eight bits set in TREG1. Note that TREG0 must be always set first (Writing data into TREG0 disables the comparator temporarily, which is restarted by writing data into TREG1).

Example: To generate interrupts INTT1 at fc=8MHz every 1 second, the timer registers TREG0 and TREG1 should be set as follows:
As øT16 (=16μs @8MHz) is selected as the input clock,
1 sec/16μs = 62500 = F424H
Therefore,
TREG1 = F4H
TREG0 = 24H

The match signal is generated by Timer 0 comparator each time the up-counter UC0 matches TREG0. In this case, the up-counter UC0 is not cleared, and the interrupt INTT0 is not generated.
Timer 1 comparator also generates the match signal each time the up-counter UC1 match TREG1. When the match signal is generated simultaneously from comparators of Timer 0 and Timer 1, the up-counters UC0 and UC1 are cleared to "0", and the interrupt INTT1 is generated. If the Timer Flip-flop inversion is enabled by the Timer Flip-flop control register, the timer flip-flop TFF1 is inverted at the same time.

Example: Given TREG1 = 04H and TREG0 = 80H,

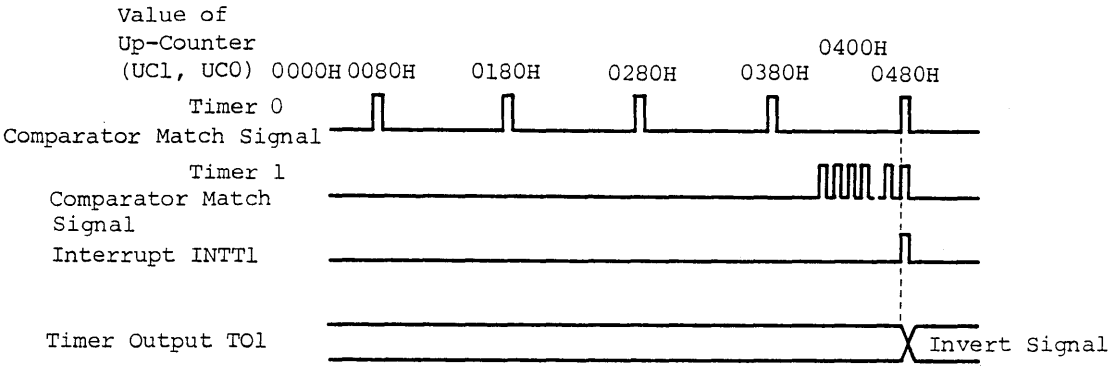
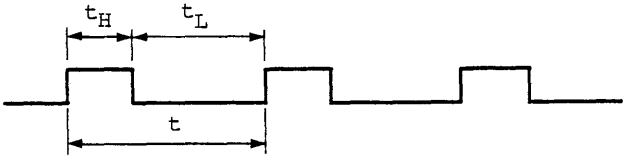


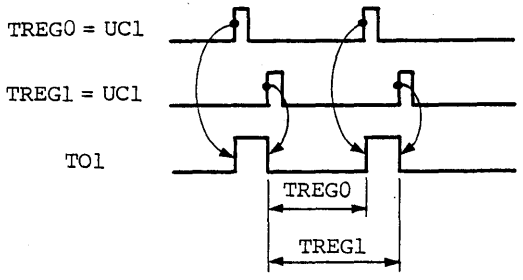
Fig. 3.6 (9)

(3) 8-bit PPG (Programmable pulse generation) mode

Pulse can be generated at any frequency and duty rate by Timer 1 or Timer 3. The output pulse may be either low- or high-active.
In this mode, Timers 0 and Timer 2 are not be used.
If Timer 1 is used, pulse is output to TO1 (also used as P60), and the use of Timer 3 results in the output to TO3 (also used as P70/P83).



Following is the timing of Timer 1 (The operation is the same as when Timer 3 is selected):



In the 8-bit PPG mode, programmable pulse is generated by the inversion of the timer output each time the 8-bit up-counter 1 (UC1) matches the timer register TREG0 or TREG1.

Note that the set value of TREG0 must be smaller than that of TREG1. In this mode, the up-counter UC0 of Timer 0 is disabled (Set TRUN 0 = 0, and stop and clear Timer 0).
The PPG mode can be illustrated as follows:

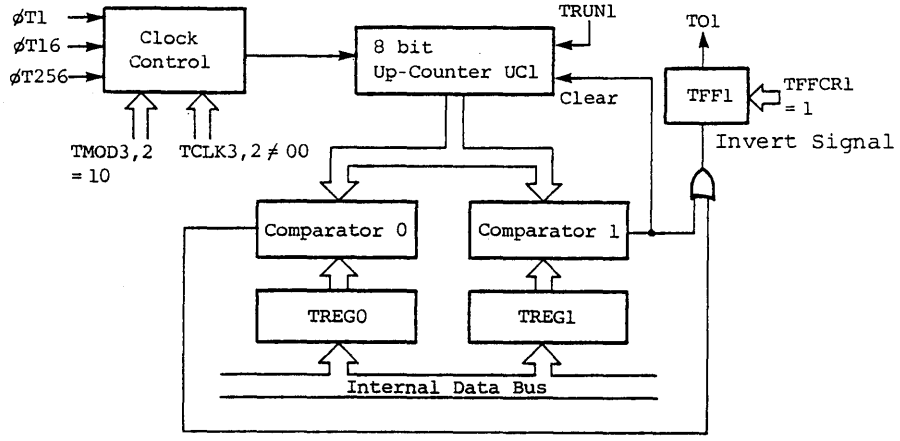
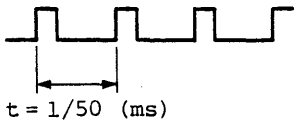


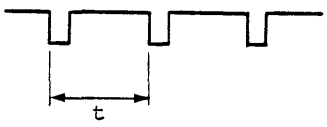
Fig. 3.6 (10) Block Diagram of 8-bit PPG Mode

Example: Generate pulse at 50kHz and 1/4 duty rate (@fc = 8MHz)



o Calculate the set values of the timer registers.
To obtain the frequency of 50kHz, the pulse cycle t should be:
1/50kHz = 20μs.
Given oT1 = 1μs (@8MHz),
20μs/1μs = 20
Consequently, the timer register 1 (TREG1) should be set to 20 = 14H.
Given a 1/4 duty, t × 1/4 = 20 × 1/4 = 5 μs
5 μs / 1 μs = 5
As a result, the timer register 0 (TREG0) should be set to 5 = 05H.

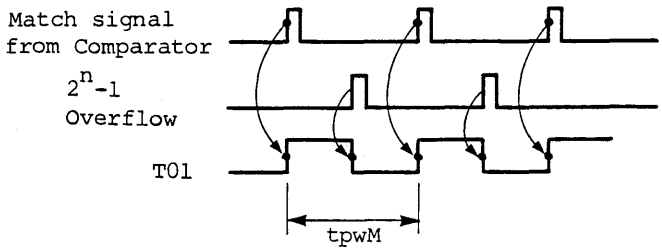
```
TRUN  <- - - - - - - 0 0   Stop Timer 0 and Timer 1, and clear them
                           to "0".
TCLK  <- - - - - 0 1 x x   Select øT1 as the input clock.
TMOD  <- - - - - 1 0 x x   Set 8-bit PPG mode.
TFFCR <- - - - - 0 0 1 1   Clear the output, and enable the inver-
                  |_|      sion by Timer 1.
                    Writing "01" provides negative logic pulse ---
TREG0 <- 0 0 0 0 0 1 0 1   Write "5H".                              |
TREG1 <- 0 0 0 1 0 1 0 0   Write "14H".                             |
SMMOD <- - - - - x x 0 1   Select P60 as the T01 pin.               |
P67CR <- - - - - - - - 1                                            |
TRUN  <- - - 1 - - - 1 0   Start Timer 1.                           |
                                                                    |
(Note) x:  Don't care    - :  No change                             |
```



(4) 8-bit PWM (pulse width modulation) mode

This mode is only available for Timer 1 and Timer 3, and generates two types of 8-bit resolution PWM (PWM1 and PWM3).
Pulse width modulation is output to T01 pin (also used as P60) when using Timer 1, and to T03 pin (also used as P70 or P83) when using Timer 3.
Timer 0 and Timer 2 can be also used as 8-bit timers.
Following is the timing of Timer 1 (PWM1) (The operation is the same as when Timer 3 is selected):
The inversion of the timer output occurs when the up-counter (UC1) matches the set value of the timer register TREG1, as well as when an overflow of 2n - 1 (n = 6, 7 or 8 selected by TMOD1 or TMOD0) occurred at the counter. The up-counter UC1 is cleared by the occurrence of an overflow of 2n - 1.
The following condition must be obtained in this PWM mode:

(Set value of timer register) < (set overflow value of 2^n - 1 counter)



The PWM mode can be illustrated as follows:

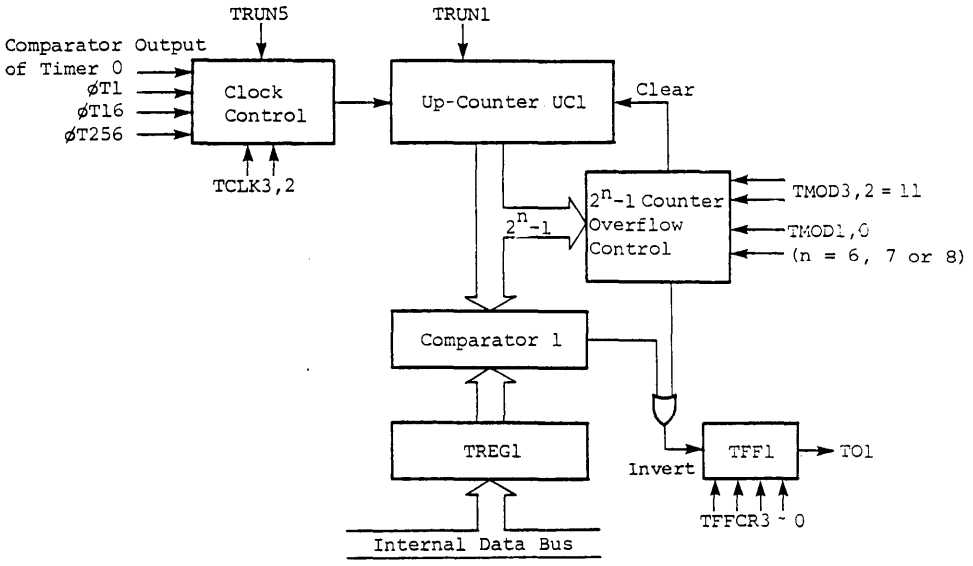
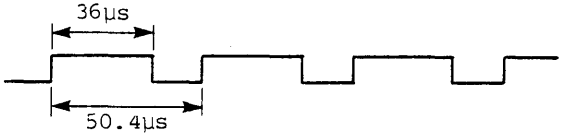


Fig. 3.6 (11) Block Diagram of 8-bit PWM Mode

Example: Generate the following PWM to the TO1 pin at fc=10MHz.



Assuming the PWM cycle is 50.4 µs when øT1 = 0.8 µs and @fc = 10MHz,

$50.4\mu s/0.8\mu s = 63 = 2^6 - 1$

Consequently, n should be set at 6 (TMOD1,0 = 01).

Given the "H" level period of 36µs, setting øT1=0.8µs results:

$36\mu s/0.8\mu s = 45 = 2DH$

As a result, TREG1 should be set at 2DH.

```
TRUN   <- - - - - - - 0 -   Stop Timer 1.
TCLK   <- - - - - 0 1 - -   Select øT1 as the input clock.
TMOD   <- - - - - 1 1 0 1   Set the 2^6 - 1 cycle in the PWM mode.
TFFCR  <- - - - - 0 0 1 1   Set the initial output to 0 ("L" level).
TREG1  <- 0 0 1 0 1 1 0 1   Write "2DH".
SMMOD  <- - - - - × × 0 1   Select P60 as the TO1 pin.
P67CR  <- - - - - - - - 1
TRUN   <- - - 1 - - - 1 -   Start Timer 1.
```

(Note) ×: Don't care - : No change

Table 3.6 (3) PWM Cycle and Selection of 2^n- 1 counter

	PWM cycle (@fc = 10MHz)		
	øT1	øT16	øT256
2^6- 1	50.4µs	806.4µs	12.9ms
2^7- 1	101.6µs	1625.6µs	26.0ms
2^8- 1	204.0µs	3264.0µs	52.2ms

3.6.2 Multi-function 16-bit timer/event counter (Timer 4)

The TMP90C840 incorporates a multi-function 16-bit timer/event counter that functions in the following operating modes:

- o 16-bit timer
- o 16-bit event counter
- o 16-bit PPG mode
- o Frequency measurement
- o Pulse width measurement
- o Time deviation measurement

Fig. 3.6 (12) is a block diagram of the 16-bit timer/event counter.

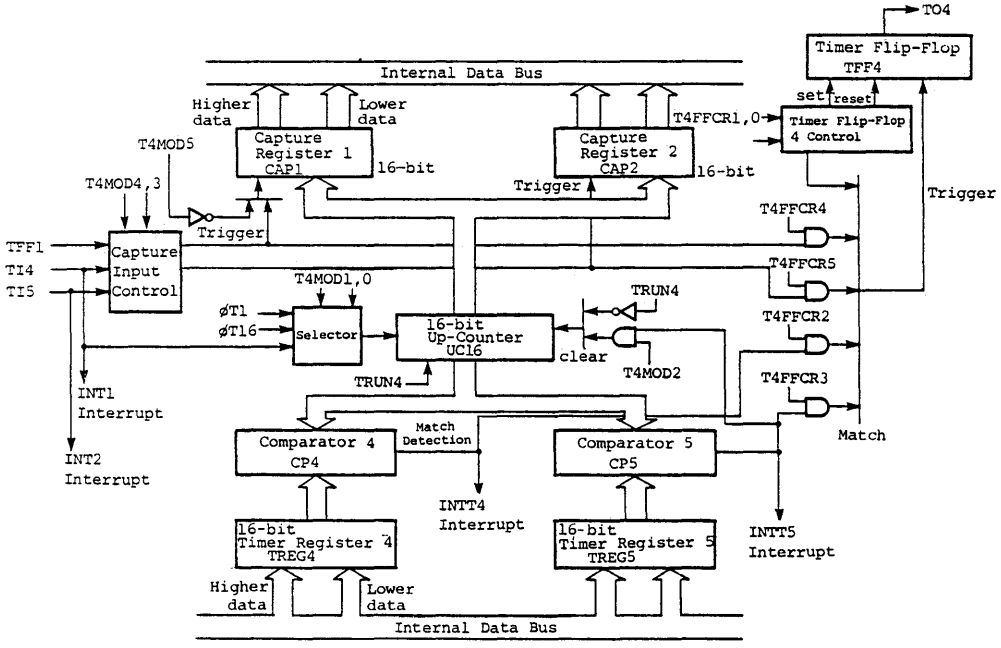
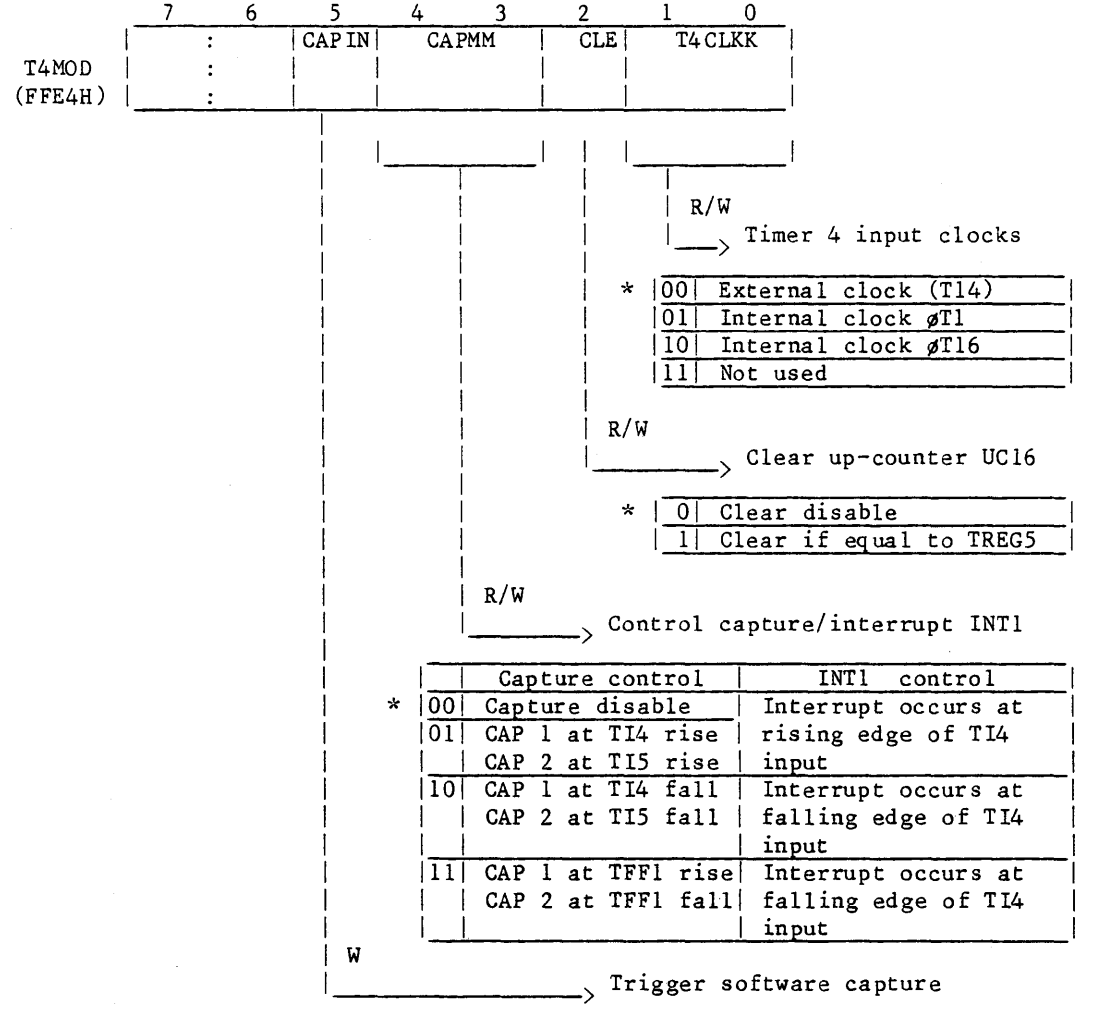


Fig. 3.6 (12) Block Diagram of 16-bit Timer/Event counter (Timer 4)

The timer/event counter is composed of a 16-bit up-counter, two 16-bit timer registers, two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its control circuit.

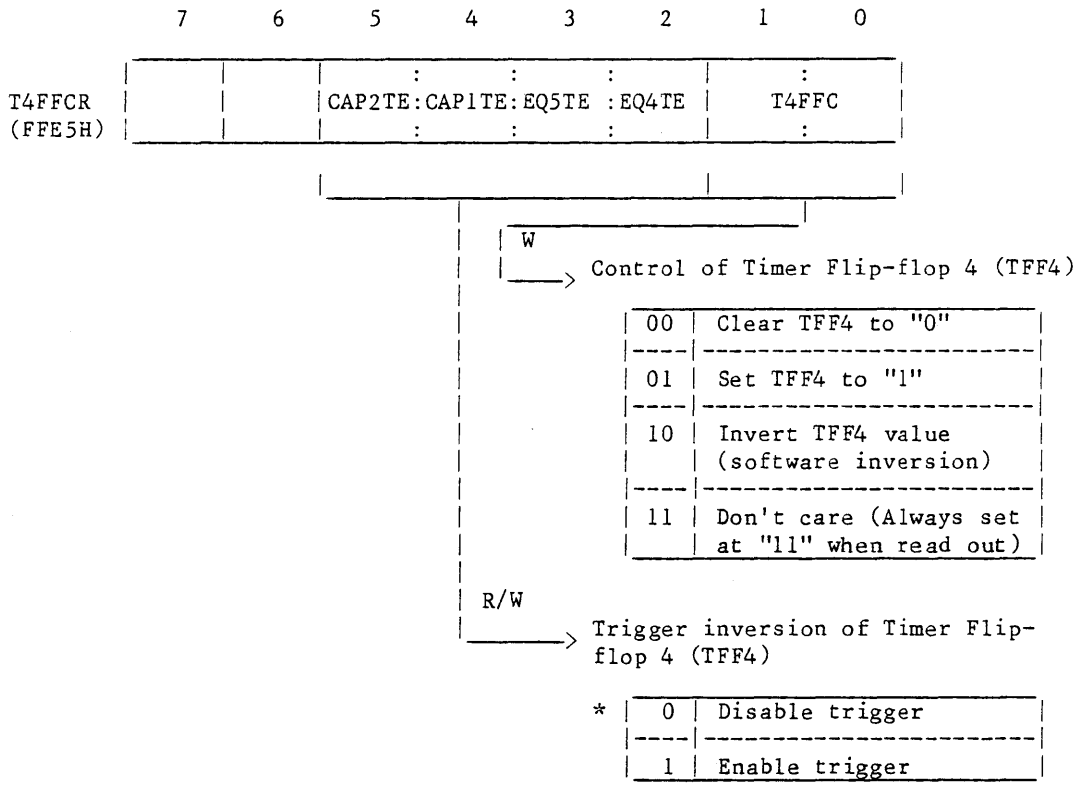
The timer/event counter is controlled by four control registers (T4MOD, T4FFCR, TRUN, and P8CR). The TRUN register also controls the 8-bit timers. For details, refer to Fig. 3.6 (5).
The P8CR register is the control register for the port P8.



(Note) * indicates initial value after reset.

o R/W denotes either reading or writing is possible.

Fig. 3.6 (13) 16-Bit Timer/Event counter (Timer 4) Control Mode registers



CAP2TE: When capturing up-counter value into CAP2
CAP1TE: When capturing up-counter value into CAP1
EQ5TE : When equalizing up-counter with TREG5
EQ4TE : When equalizing up-counter with TREG4

Fig. 3.6 (14) 16-bit Timer/Event Counter Timer Flip-Flop 4 Control Registers

1 Up-counter (UC16)
UC16 is a 16-bit binary counter that counts up by the input clock specified by the register T4MOD (T4CLK) (bits 1 ans 0).
Its input clock is selected from the internal clocks øT1 and øT16 generated by the 9-bit prescaler (also used as the 8-bit timer), or the external clock from the T14 pin (also used as P81/INT1). By resetting, the bits 0 and 1 of T4MOD are initialized to "00", whereby T14 is selected as the input clock to this up-counter.
The Timer Control Register TRUN 4 (T4RUN) controls the counter to start, stop or clear.
Clearing the up-counter UC16 is controlled by the bit 2 of T4MOD2 register (CLE). The up-counter UC16 is cleared to "0" when matching the timer register TREG5.

2 Timer registers (TREG4 and TREG5)
Two 16-bit registers are provided to set the counter value. When the values in these timer registers equal with the value of the up-counter UC16, the match signal of the comparator becomes active.

Data for the timer registers TREG4 and TREG5 are provided by a 2-byte data load instruction or two 1-byte data load instructions, from the lower eight bits followed by the upper eight bits.

TREG4		TREG5	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
FFE1H	FFE0H	FFE3H	FFE2H

3 Capture registers (CAP1 and CAP2)

CAP1 and CAP2 are 16-bit registers that hold the values of the up-counter UC16.

Data in the capture registers are read by a 2-byte data load instruction or two 1-byte data load instructions, from the lower eight bits followed by the upper eight bits.

CAP1		CAP2	
Upper 8 bits	Lower 8 bits	Upper 8 bits	Lower 8 bits
FFDDH	FFDCH	FFDFH	FFDEH

4 Capture input control circuit

This circuit controls the timing that the capture registers (CAP1 and CAP2) latch the UC16 up-counter value.

The latch timing is set by the register T4MOD4,3 (CAPM).

- o If T4MOD4,3 = 00,
  The capture function is disabled. Theese bits are initialized to this mode by resetting.
- o If T4MOD4,3 = 01,
  The up-counter value is latched into CAP1 at the rising edge of the TI4 (also used as P81/INT1) input, and into CAP2 at the rising edge of TI5 (also used as P82/INT2) input.
- o If T4MOD4,3 = 10,
  The up-counter value is latched into CAP1 at the rising edge of the TI4 input, and by CAP2 at its falling edge. Only in this mode, the interrupt INT1 occurs at the falling edge.
- o If T4MOD4,3 = 11,
  The up-counter value is latched into CAP1 at the rising edge of the timer flip-flop TFF1, and into CAP2 at its falling edge.

The up-counter value can be also latched into the capture registers by using software. In this case, the current up-counter value is latched into CAP1 each time "0" is written into the T4MOD5 (CAPIN) register.

⑤ Comparators (CP4 and CP5)

The 16-bit comparators detect the match of the up-counter UC16 and the timer register TREG4 or TREG5.

When the up-counter matches registers TREG4 or GREG5, the interrupt INTT4 or INTT5 occurs, respectively. The up-counter is cleared only when it matches TREG5 (the clear operation can be disabled if necessary).

⑥ Timer Flip-flop (TFF4)

This Timer Flip-flop is inverted by the match signal from the comparators (CP4 and CP5) and the latch signal to the capture registers (CAP1 and CAP2).

It is possible to enable/disable the inversion for each of these sources by using TRFFCR5-2.

Also TFF4 can be inverted, cleared to "0" and set to "1" under software control.

Its value can be output to the timer output pin TO4 also used as P83 or TO3. Either pin function may be selected by the registers P8CR and SMMOD. TO4 is selected by setting P8CR3 = 1 and SMMOD5,4 = 11.

(1) 16-bit timer mode

In this mode, the interval time is set in the timer register TREG5 to generate the interrupt INTT5.

```
| TRUN  <- - - - 0 - - - -     Stop Timer 4.
| INTEL <- - - 0 - 1 - - -     Enable INTT5 and disable INTT4.
| T4FFCR<- x x 0 0 0 0 1 1     Disable trigger.
| T4MOD <- x x 1 0 0 1 * *     Select internal clock for input, and
|                (**=01,10,11) disable the capture function.
| TREG5 <- **** **** **** **** Set the interval time (16 bits).
| TRUN  <- - - 1 1 - - - -     Start Timer 4.
```

(Note) x: Don't care - : No change

(2) 16-bit event counter mode

This timer can be used as a 16-bit event counter by selecting the external clock (TI4) as the input clock in the above timer mode (1).

The counter counts up at the rising edge of the TI4 input clock.

The TI4 pin is also used as P81 or INT1.

```
| TRUN  <- - - - 0 - - - -     Stop Timer 4.
| P8CR  <- - - - - - - * -     *=0; TI4 is square wave.
|                              *=1; TI4 is sign (zero-cross) wave
| INTEL <- - - 0 0 1 - - -     Enable INTT5 and disable INTT4 and
|                              INT1.
| T4FFCR<- x x 0 0 0 0 1 1     Disable trigger.
| T4MOD <- x x 1 0 0 1 0 0     Select TI4 as the input clock.
| TREG5 <- **** **** **** **** Set the number of counts (16 bits).
| TRUN  <- - - 1 1 - - - -     Start Timer 4.
```

(3) 16-bit programmable pulse generation (PPG) mode

The PPG mode is obtained by inversion of the Timer Flip-flop TFF4 to be enabled by match of the up-counter UC16 with the timer register TREG 4 or 5 and output it to TO4 (also used as P83/TO3). In this mode the following condition must be satisfied.

(Set value of TREG4) < (Set value of TREG5)

```
TRUN   <- - - - 0 - - - -      Stop Timer 4.
TREG4  <- **** **** **** ****  Set the duty rate.
TREG5  <- **** **** **** ****  Set the cycle.
T4FFCR <- x x 0 0 1 1 0 0      Set the TFF4 inversion to be effected
                               by match with TREG4 or 5.
                               Initialize TFF4 to "0".
T4MOD  <- x x 1 0 0 1 **       Select the internal clock for the
                  (**=01,10,11) input,and disable the capture function.
SMMOD  <- - - * * - - - -
                  (**=00,01,10) Select P83 as the TO4 pin.
P8CR   <- - - - - 1 - - -
TRUN   <- - - 1 1 - - - -      Start Timer 4.
```

(Note) x: Don't care - : No change

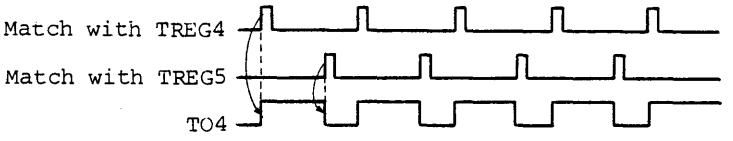


Fig. 3.6 (15) Programmable Pulse Output

(4) Application examples of capture function

Loading the up-counter (UC16) value into CAP1 or CAP2, the Timer Flip-flop TFF4 inversion by the match signal from CP4 or CP5, and the output of the TFF4 status to TO4 pin can be enabled or disabled. Various applications are available by combining with the interrupt function:

① One-shot pulse output by using external trigger pulse.
② Frequency measurement.
③ Pulse width measurement.
④ Time deviation measurement.

① One-shot pulse output from the rising edge of external trigger pulse. Keep the up-counter (UC16) counting (free-running) with by the internal clock input by setting T4RUN of the TRUN register to "1". The input an external trigger pulse through the TI4 pin, and also load the up-counter value into the capture register CAP1 at the rising edge of TI4 (T4MOD4,3 = 01).

When the interrupt INT1 is generated at the rising edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4, and set the above set value (c+d) plus a one-shot pulse width (p) to TREG5. When the interrupt INT1 occurs the register T4TRG is selected that the TFF4 inversion is enabled only when the up-counter value matches TREG4 or 5. When interrupt INTT5 occurs, the inversion is disabled.

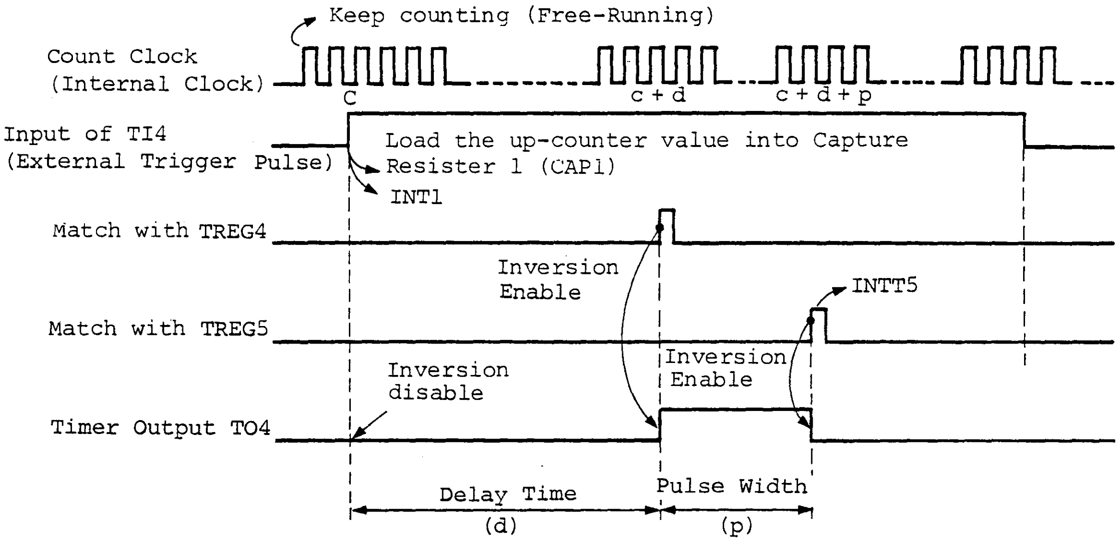


Fig. 3.6 (16) One-Shot Pulse Output (with delay)

Example: Generate a 2ms one-shot pulse with a 3ms delay from the rising edge of an external trigger pulse ("H" level width: 10ms - 30ns)

<u>Main setting</u>

```
                           ______> Keep counting (Free-running)
                          |  __> Count with oT1.
                          | |
                          | |‾|
 |‾T4MOD <- x x 1 0 1 0 0 1
 |                |_|
 |                 |______> Load the up-counter value into CAP1 at
 |                          the rising edge of TI4 input.
 | T4FFCR<- x x 0 0 0 0 0 0
 |              |_____| |_|
 |                 |      |__> Clear TFF4 to "0".
 |                 |______> Disable TFF4 inversion.
 | SMMOD <- - - * * - - - - \
 |          (**=00,01,10)    > Select P83 as the TO4 pin.
 | P8CR  <- - - - - 1 - - - /
 | INTEL <- - - 0 1 0 - - -   Enable INT1, and disable INTT4 and INTT5.
 |_TRUN  <- - - 1 1 - - - -   Start Timer 4.
```

Setting of INT1

```
TREG4 <- CAP1+3ms/øT1
TREG5 <- TREG4+2ms/øT1
T4FFCR<- × × - - 1 1 - -
                  |_|
                   |_____> Enable TFF4 inversion when the up-counter
                            value matches TREG4 or 5.
INTEL <- - - - - 1 - - -   Enable INTT5.
```

Setting of INT5

```
T4FFCR<- × × - - 0 0 - -
                  |_|
                   |_____> Disable TFF4 inversion when the up-
                            counter value matches TREG4 or 5.
INTEL <- - - - - 0 - - -   Disable INTT5.
```

(Note) ×: Don't care - : No change

When no delay time is required, invert the TFF4 when the up-counter value is loaded into CAP1, and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt INT1 occurs. The TFF4 inversion should be enabled when the up-counter (UC16) value matches TREG5, and disabled when generating the interrupt INTT5.

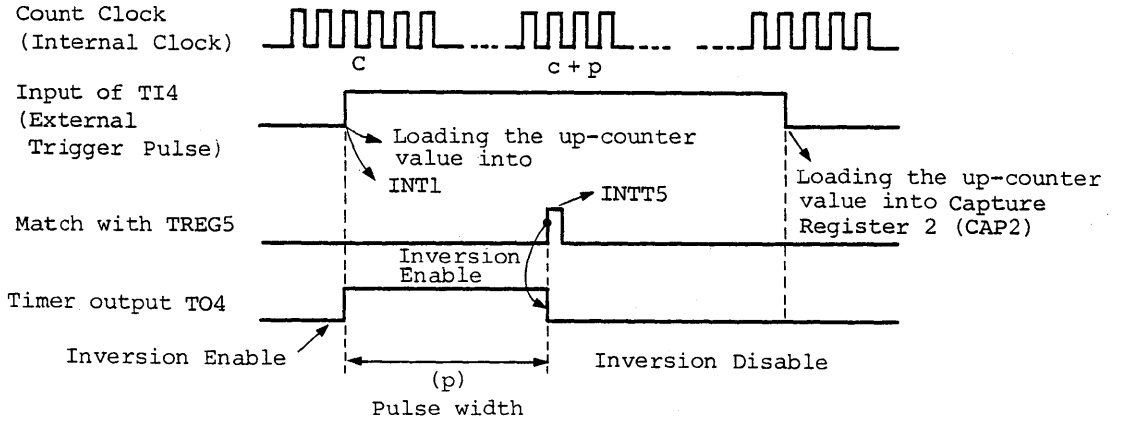


Fig. 3.6 (17) One-Shot Pulse Output (with no delay)

② Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TI4 pin, and its frequency is calculated by using the 8-bit timers (Timer 0 and Timer 1) and the 16-bit timer/event counter (Timer 4);

The frequency of the external clock is calculated by using the frequency of the input clock to Timer 4 (TI4). The value of the up-counter is loaded into the capture register CAP1 at the rising edge of the Timer Flip-flop TFF1 of 8-bit timers (Timer 0 and Timer 1), and into CAP2 at its falling edge.

The frequency is calculated by using the number of input clock in a certain period (the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.

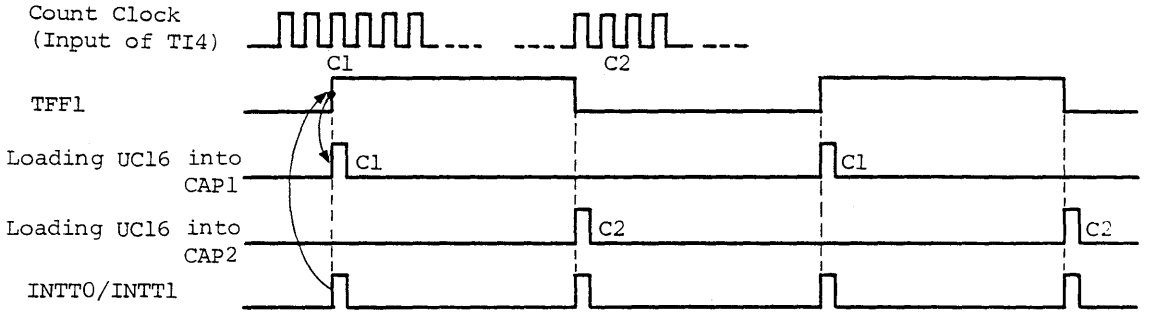


Fig. 3.6 (18) Frequency Measurement

③ Pulse width measuremnt

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer/event counter counting (free-running) with the internal clock input, the external pulse is input through the TI4 pin. Then the capture function is used to load the UC16 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT1 occurs at the falling edge of TI4.

The pulse width is obtained from the difference between the values of CAP1 and CAP2 and the internal clock cycle.

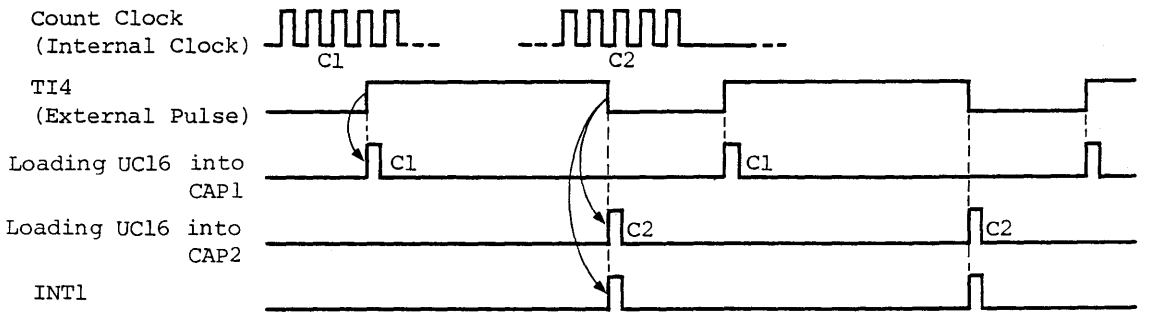


Fig. 3.6 (19) Pulse Width Measurement

(Note) The external interrupt INT1 occurs at the falling edge of TI4 only in this pulse width measurement mode (T4MOD4,3 = 10). In any other mode, the interrupt occurs at its rising edge.

The width of "L" level can be obtained from the difference between the first C2 and the second C1 when the second INT1 interrupt is generated.

④ Time difference measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through TI4 and TI5.

Keep the 16-bit timer/event counter (Timer 4) counting (free-running) with the internal clock, and load the UC16 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT1 is generated.

Similarly, the UC16 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT2.

The time difference between these pulses can be obtained from the difference between the time loading the up-counter value into CAP1 and CAP2.

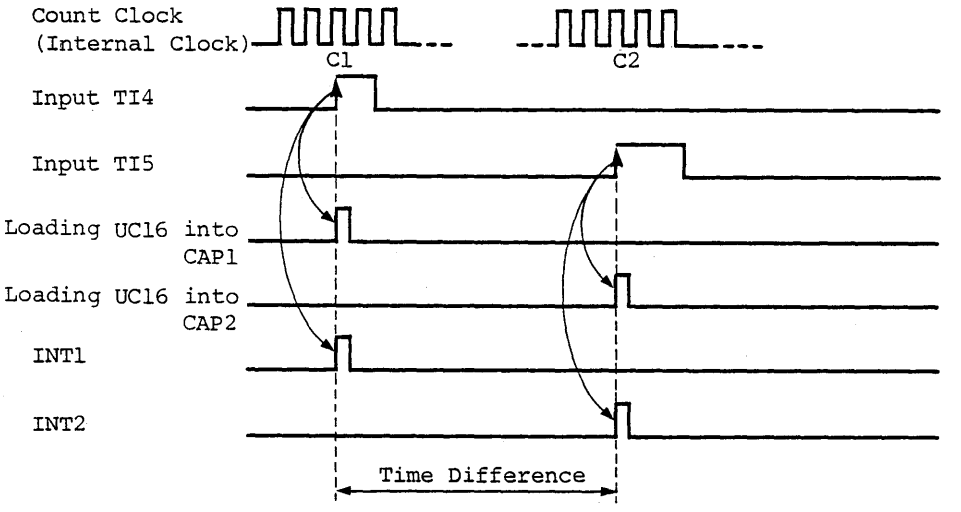


Fig. 3.6 (20) Time Difference measurement

3.7 Stepping Motor control ports (P6 and P7)

The TMP90C840 has two 4-bit hardware stepping motor control ports (SMC ports M0 and M1) that synchronize with the (8-bit/16-bit) timers. These SMC ports M0 and M1 are also used as the I/O ports P6 and P7.

The channel 0 (M0) synchronizes with the trigger signal of the Timer Flip-flop TFF1, and the channel 1 (M1) is synchronous with that of the Timer Flip-flop TFF3 or TFF4, for output.

The SMC ports are controlled by three control registers (P67CR, SMMOD, and SMCR), and allow the selection of driving method: 1) 4-phase 1-step/2-step excitation and 2) 4-phase 1-2 step excitation.

3.7.1 Control registers

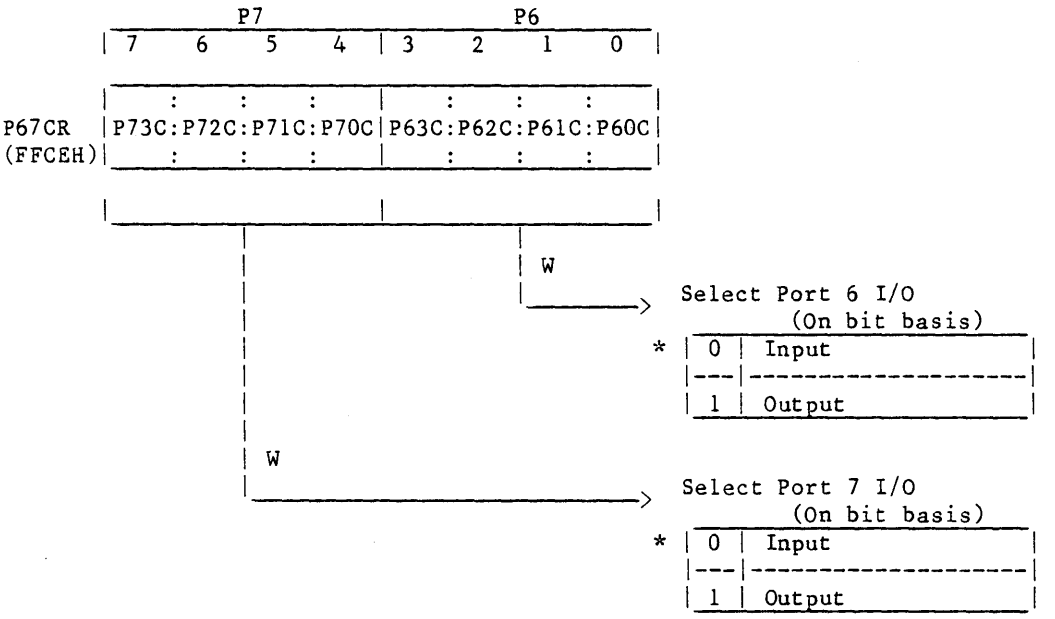
(1) Port 6 and Port 7 I/O selection register (P67CR)
This register specifies either input or output for each bit of the 4-bit I/O ports 6 and 7.
When all bits of P67CR are initialized to "0" by resetting, Ports 6 and 7 function as input ports.
P67CR is a write-only register (not for readout).

(2) Stepping motor control port mode register (SMMOD)
Ports 6 and 7 also function as SMC Ports (M0 and M1) or Timer Output Ports (T01 and T03), as selected by SMMOD1, 0 or SMMOD5, 4.
When Port 6 is used as SMC Port (M0), SMMOD1, 0 should be set to 10 or 11.
When Port 7 is used as SMC Port (M1), SMMOD5, 4 should be set to 10 to make it synchronize with the trigger pulse of the timer flip-flop TFF3 and set at 11 to make it synchronize with the trigger pulse of the Timer Flip-flop TFF4. P83 (also used as T03 and T04) can function as T03 only when SMMOD5, 4 are set at 11.
SMMOD2 and SMMOD6 serve to select the excitation method. With "0" the full-step (1-step/2-step) excitation is selected and with "1" the half-step (1-2 step) excitation is selected. When full-step excitation is selected, the selection of either 1 or 2-step excitation is determined by the initial output value.
SMMOD3 and SMMOD7 allow to set the number of phases.

(3) Stepping motor control port rotating direction control register (SMCR)
This register controls the rotating direction. The direction of the channel 0 (M0) is set by SMCR0 (CCW6) and that of the channel 1 (M1) is set by SMCR4 (CCW7).
"0" denotes normal rotation and "1" denotes reverse rotation.

(4) Port 6
This is a 4-bit I/O port allocated to the address FFCCH.
The lower four bits are assigned as Port 6, while the upper four bits function as the shifter alternate register (SA6) for driving the stepping motor with the half-step (1 - 2) excitation.

(5) Port 7
This is a 4-bit I/O port allocated to the address FFCDH.
The lower four bits are assigned as Port 7, while the upper four bits function as the shifter alternate register (SA7) for driving the stepping motor with the half-step (1 - 2) excitation.



(Note) *: initial value after reset.
w: write only.

Fig. 3.7 (1) Ports 6 and 7 I/O Selection Registers

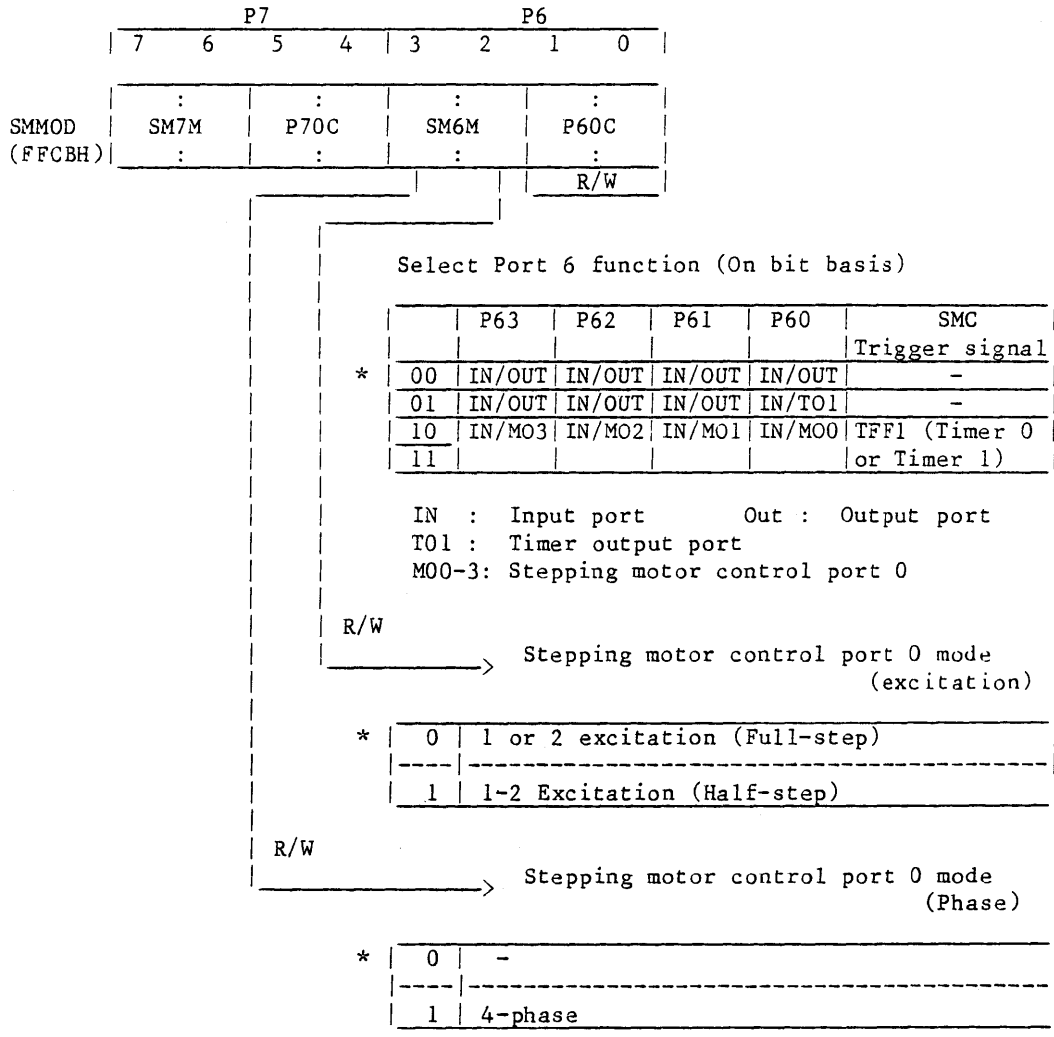


Fig. 3.7 (2a) Stepping Motor Control Port Mode Registers

	P7				P6			
	7	6	5	4	3	2	1	0
SMMOD (FFCBH)	: SM7M :		: P70C :		: SM6M :		: P60C :	

R/W

Select Port 6 function (On bit basis)

		P73	P72	P71	P70	P83	SMC Trigger signal
*	00	IN/OUT	IN/OUT	IN/OUT	IN/OUT		-
	01	IN/OUT	IN/OUT	IN/OUT	IN/TO3	OUT/TO4	
	10	IN/M13	IN/M12	IN/M11	IN/M10		TFF3 (Timer 2 and Timer 3
	11					OUT/TO3	TFF4 (Timer 4)

IN : Input port OUT : Output port
TO3,TO4 : Timer output port
M10-3: Stepping motor control port 1

R/W

Stepping motor control port 1 mode (excitation)

*	0	1 or 2 excitation (Full-step)
	1	1 - 2 excitation (Half-step)

R/W

Stepping motor control port 1 mode (Phase)

*	0	-
	1	4-phase

Fig. 3.7 (2b) Stepping Motor Control Port Mode Registers

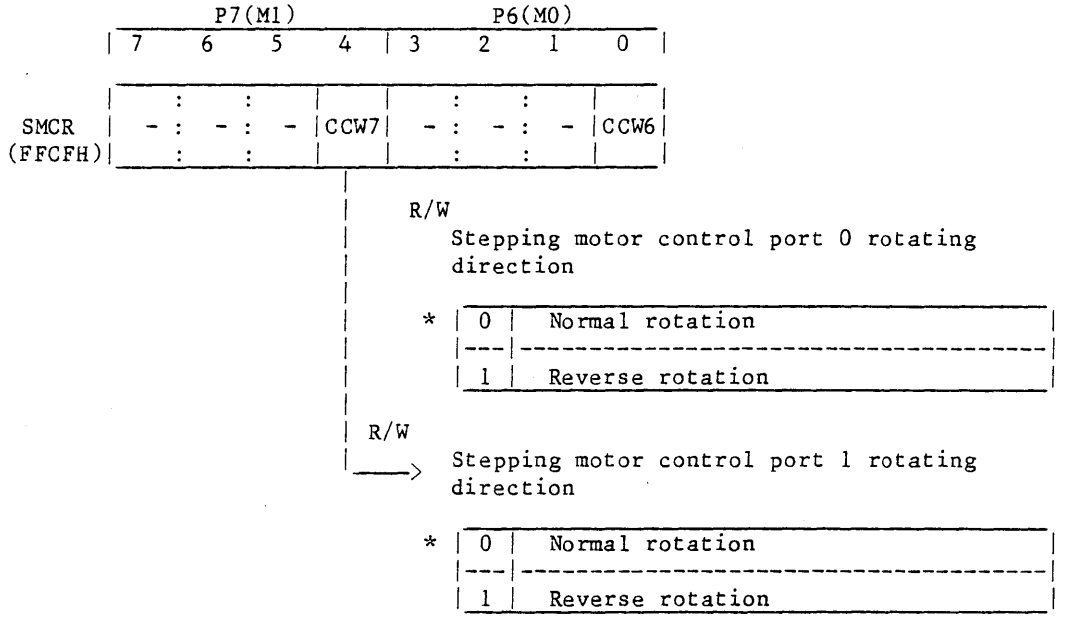


Fig. 3.7 (3) Stepping Motor Control Port Rotating Direction Control Registers

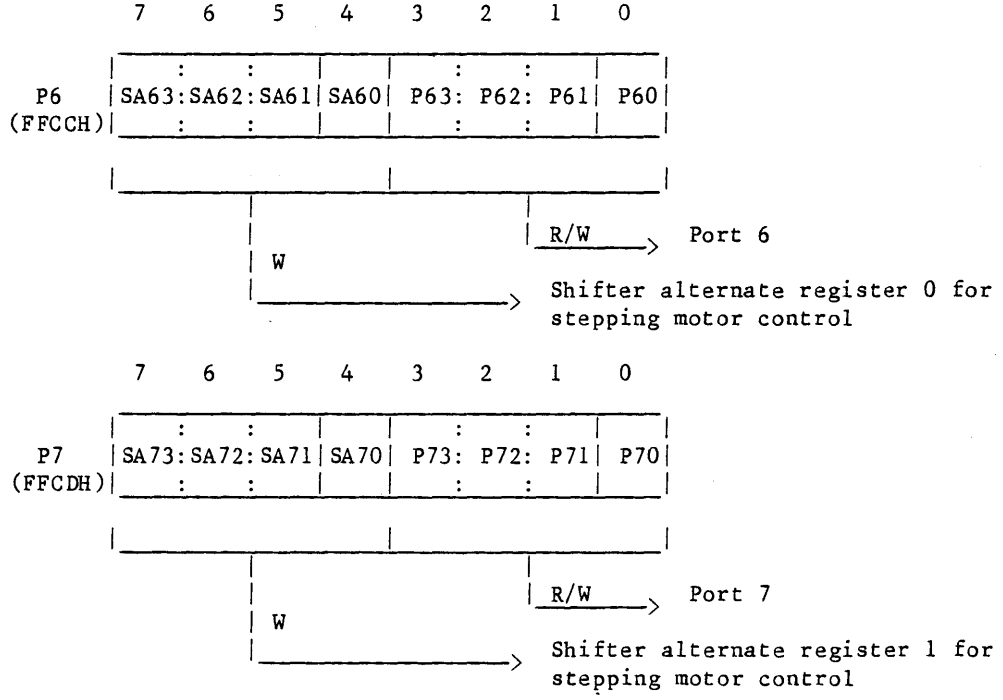
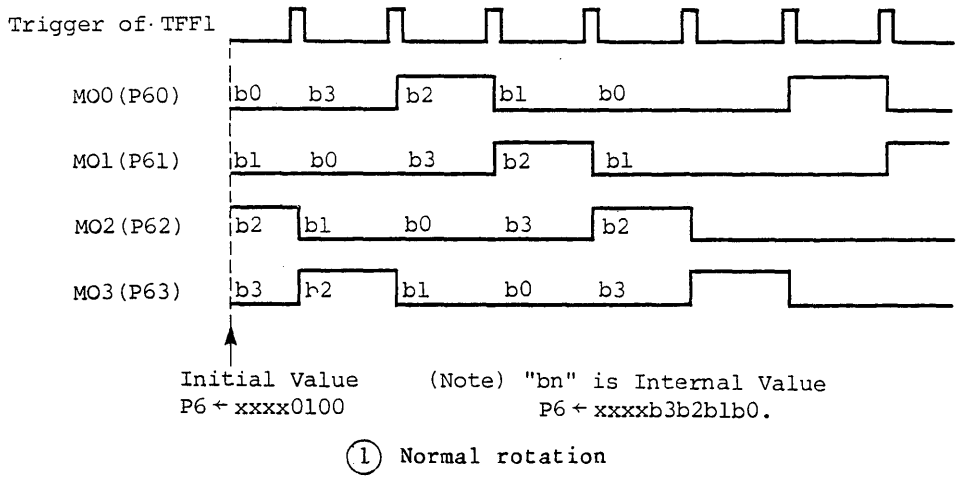


Fig. 3.7 (4) Ports 6 and 7

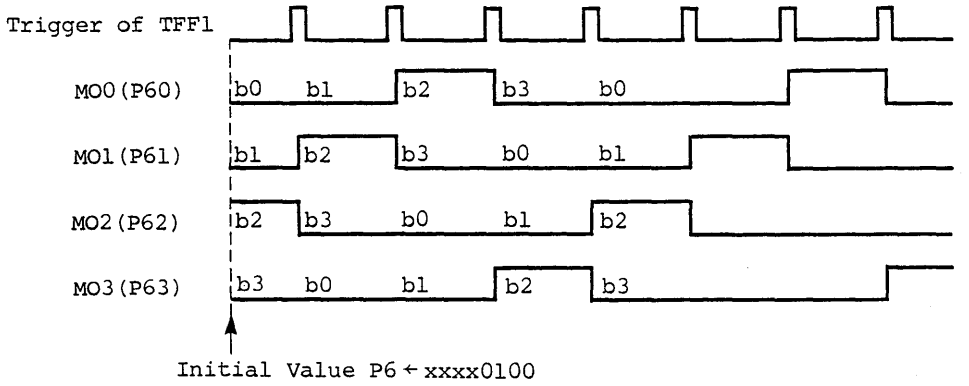
3.7.2 4-phase 1-step/2-step excitation

Figs. 3.7 (5) and (6) show the output waveforms of 4-phase 1-step excitation and 2-step excitation when Channel 0 is selected.

Sets circle by using TIMER-0 or TIMER-1



① Normal rotation



② Reverse rotation

Fig. 3.7 (5) Output Waveforms of 4-phase 1-step Excitation (Normal Rotation and Reverse Rotation)

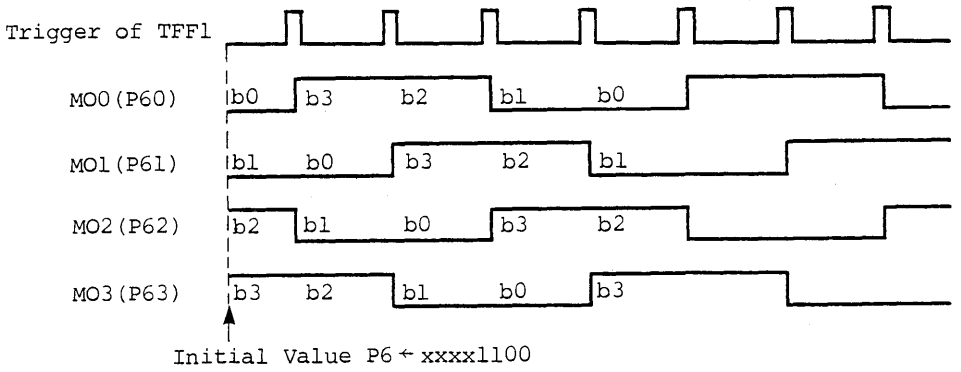


Fig. 3.7 (6) Output Waveforms of 4-phase 1-step Excitation (Normal Rotation)

When Channel 0 is selected, for example, the stepping motor control port is controlled as follows:
The output latch of M0 (also used as P6) rotates at the rising edge of the TFF1 trigger pulse (that inverts the value of TFF1) and is output to the port.
The rotating direction is selected by SMCR0 (CCW6). Setting SMCR0 to "0" selects the normal rotation (M00 -> M01 -> M02 -> M03), and setting it to "1" results in the reverse rotation (M00 <- M01 <- M02 <- M03). The 4-phase 1-step excitation can be obtained by initializing any one bit of Port 6 to "1", and setting two succeeding bits to "1" produces the 4-phase 2-step excitation.
Fig. 3.7 (7) is a block diagram of the two excitations.

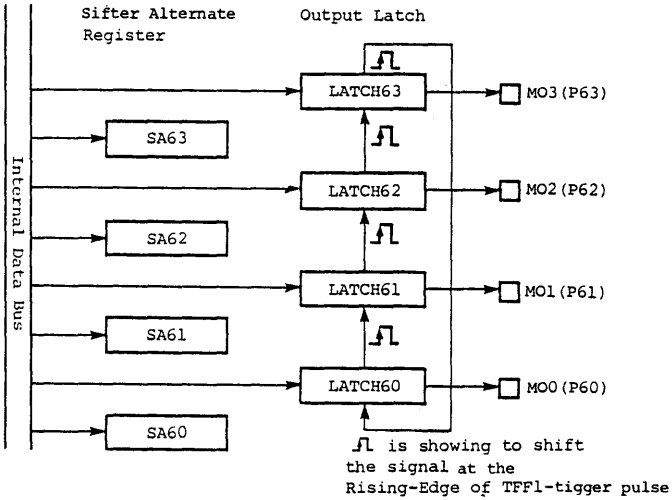
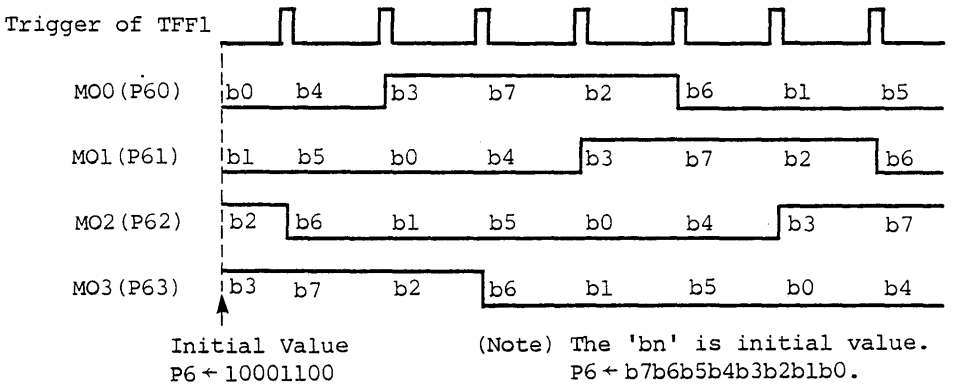


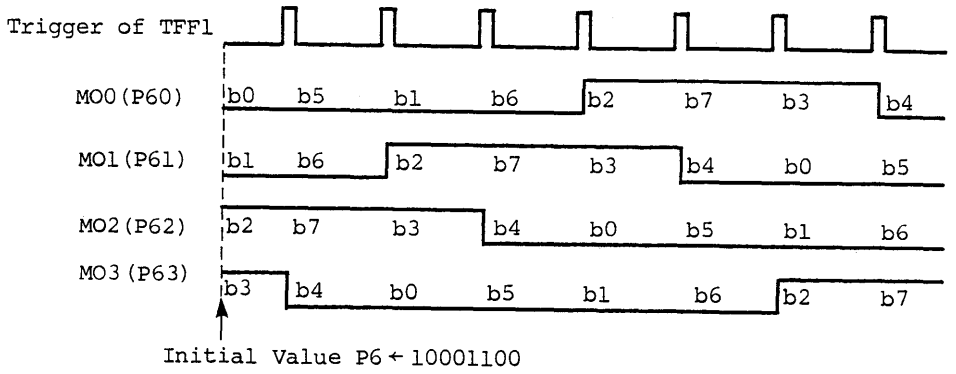
Fig. 3.7 (7) Block Diagram of 4-phase 1-step/2-step Excitation (Normal Rotation)

3.7.3 4-phase 1-2 step excitation

Figs. 3.7 (8) shows the output waveforms of 4-phase 1-2 step excitation when Channel 0 is selected.



① Normal rotation



② Reverse rotation

Fig. 3.7 (8) Output Waveforms of 4-phase 1-2 step Excitation (Normal Rotation and Reverse Rotation)

The 4-phase 1-2 step excitation is obtained as follows:

Given the Port bits | b7 b6 b5 b4 b3 b2 b1 b0 | initialially arranged as | b3 b7 b2 b6 b1 b5 b0 b2 |, set three succeeding bits to "1", and the other bits to "0" (Positive logic).
For example, initializing b3, b7 and b2 to "1" provides the initial value "10001100", which produces the waveforms in Fig. 3.7 (8).
To obtain the negative logic output, the above initial value of Port 6 should be inverted.
That is, the waveforms in Fig. 3.7 (8) can be converted to the negative logic output by initializing the P6 bits to "011110011".
When Channel 0 is selected, for example, the stepping morot control port is controlled as follows:
The output latch of M0 (also used as P6) and the stepping motor control shifter alternate register (SA6) rotate at the rising edge of the TFF1 trigger pulse and are output to the port. The rotating direction is selected by SMCR0 (CCW6).
Fig. 3.7 (9) is a block diagram of this excitation.

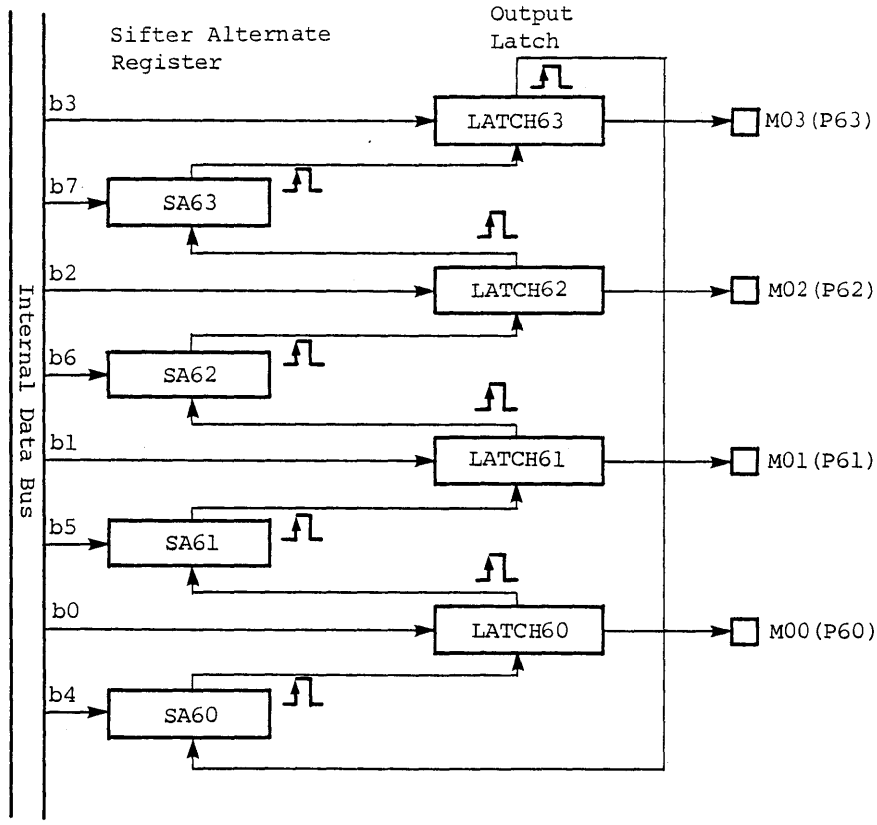


Fig. 3.7 (9) Block Diagram of 4-phase 1-2 step Excitation (Normal Rotation)

Example: The registers should be set as follows to drive Channel 0 (M0) with the 4-phase 1-2 step excitation (normal rotation) when Timer 0 is selected:

```
                7 6 5 4 3 2 1 0
 TRUN   <- - - - - - - - - 0    Stop Timer 0, and clear it to "0".
 TMOD   <- - - - - 0 0 x x      Set the 8-bit timer mode.
 TCLK   <- - - - - - - - 0 1    Select øT1 as the input clock.
 TREG0  <- * * * * * * * *      Set the cycle in Timer register.
 TFFCR  <- - - - - 0 0 1 0      Clear TFF1 to "0", and enable inversion
                                trigger by Timer 0
 SMMOD  <- - - - - 1 1 1 x      Select 4-phase 1-2 step excitation mode.
 SMCR   <- - - - - - - - - 0    Select normal rotation.
 P67CR  <- - - - - 1 1 1 1      Set all P6 bits to the output mode.
 P6     <- 1 0 0 0 1 1 0 0      Initialize the P6 bits.
 TRUN   <- - - 1 - - - - 1      Start Timer 0.
```

(Note) x: Don't care - : No change

3.8 Serial Channel

The TMP90C840 incorporates a serial I/O channel for full-duplex asynchronous transmission (UART) and I/O expansion.
The serial channel has the following operating modes:

- o I/O interface mode --- Mode 0: Transmit/receive I/O data for expand I/O and transmit its synchronous signals (SCLK)
- o Asynchronous transmission (UART) mode
  - Mode 1 : 7-bit data
  - Mode 2 : 8-bit data
  - Mode 3 : 9-bit data

The Mode 1 and Mode 2 allow the addition of a parity bit. The Mode 3 accommodates a wake-up function to start the slave controllers in a controller serial link (multi-controller system).
Fig. 3.8 (1) shows the data format (1-frame data) in each mode.

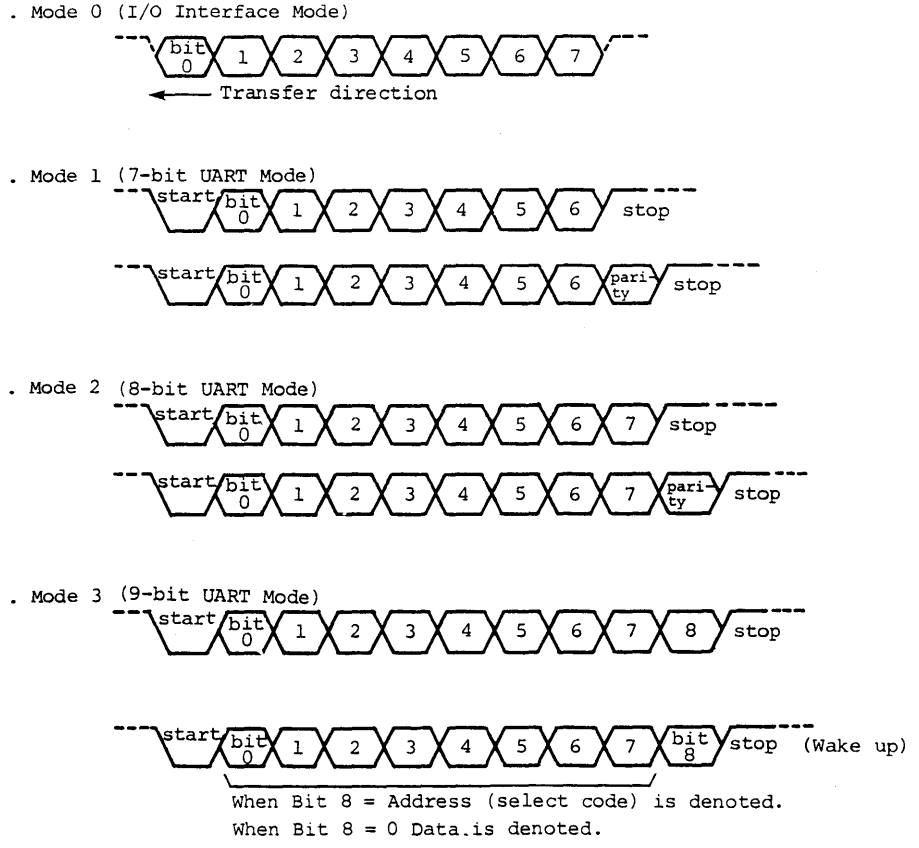


Fig. 3.8 (1) Data Formats

Data received and transmitted are stored temporarily into separate buffer registers to allow independent transmission and receiving (Full-duplex).

In the I/O interface mode, however, the data transfer is half-duplex due to the single SCLK (serial clock) pin is used for transmisison and receiving.

Two pins are provided for receiving (RxD, also used as P30 and P31) and transmission (TxD, also used as P32 and P33), which allows two serial transmisison/receiving using the time-sharing technique. The pin function is selected by the P3CR registers. For example, P30 can be used as the RxD pin by setting P3CR1, 0 to 01.

The receiving buffer register has a double-buffer structure to prevent overruns. The one buffer receives the next frame data while the other buffer stored the receive data is read by the CPU.

The use of $\overline{CTS}$ and $\overline{RTS}$ allows to halt the data transmission until the CPU completes reading of the receive data for each frame (Hand-shake function).

In the UART mode, a check function is added not to start the receiving operation by error start bits due to noise. The channel starts receiving data only when the start bit is detected to be normal at least twice in three samplings.

When an request is issued to the CPU to transmit data after the transmitting buffer becomes empty or to read data after the receiving buffer completed to store data, the interrupt INTTX or INTRX occurs respectively. In receiving data, the occurrence of an overrun error, parity error or framing error sets the flag (SCCR4, 3 or 2) accordingly.

3.8.1 Control Registers

The serial channel is controlled by four control registers (SCMOD, SCCR, TRUN, and P3CR). The received/transmitted data are stored into SCBUF.

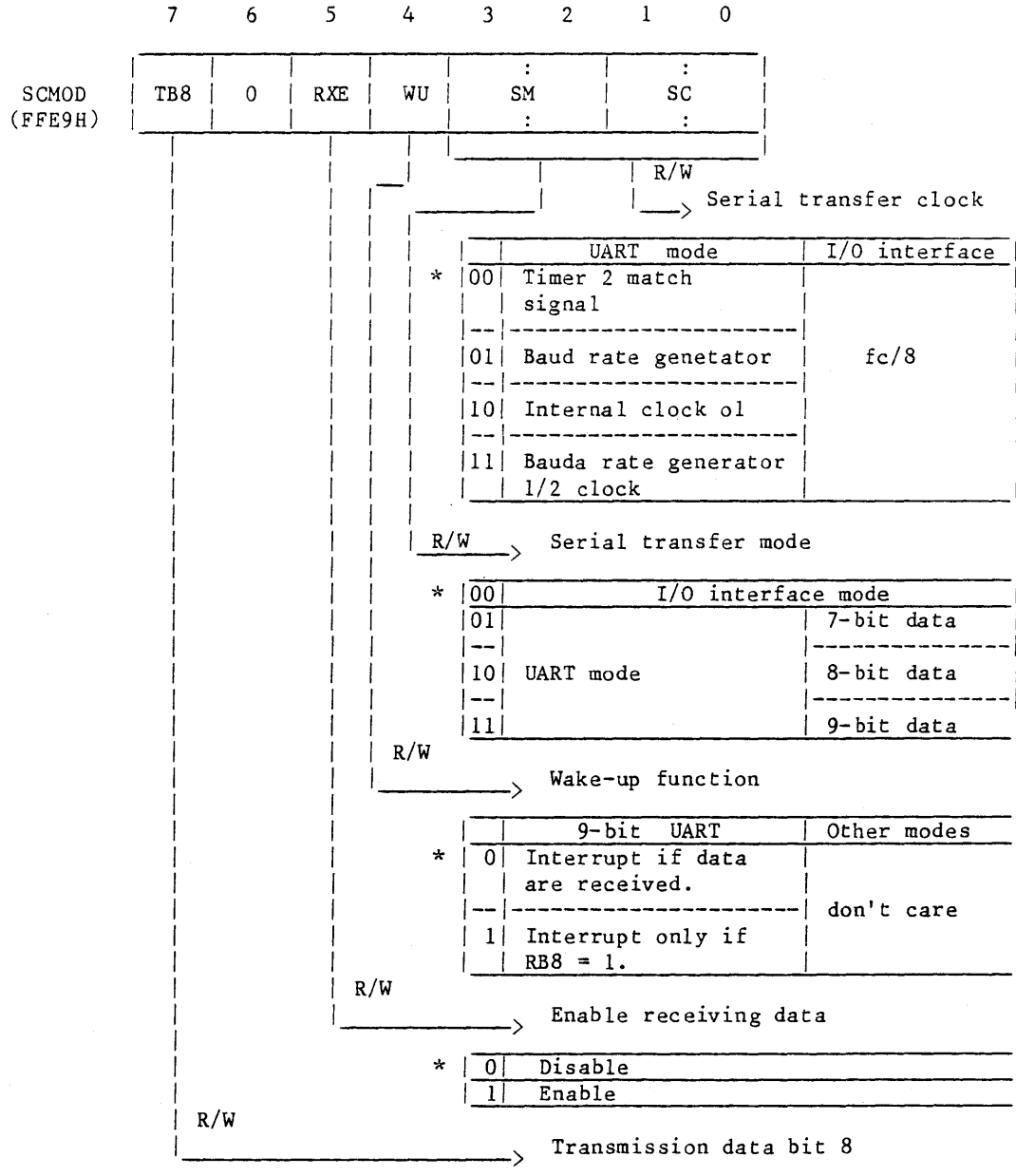
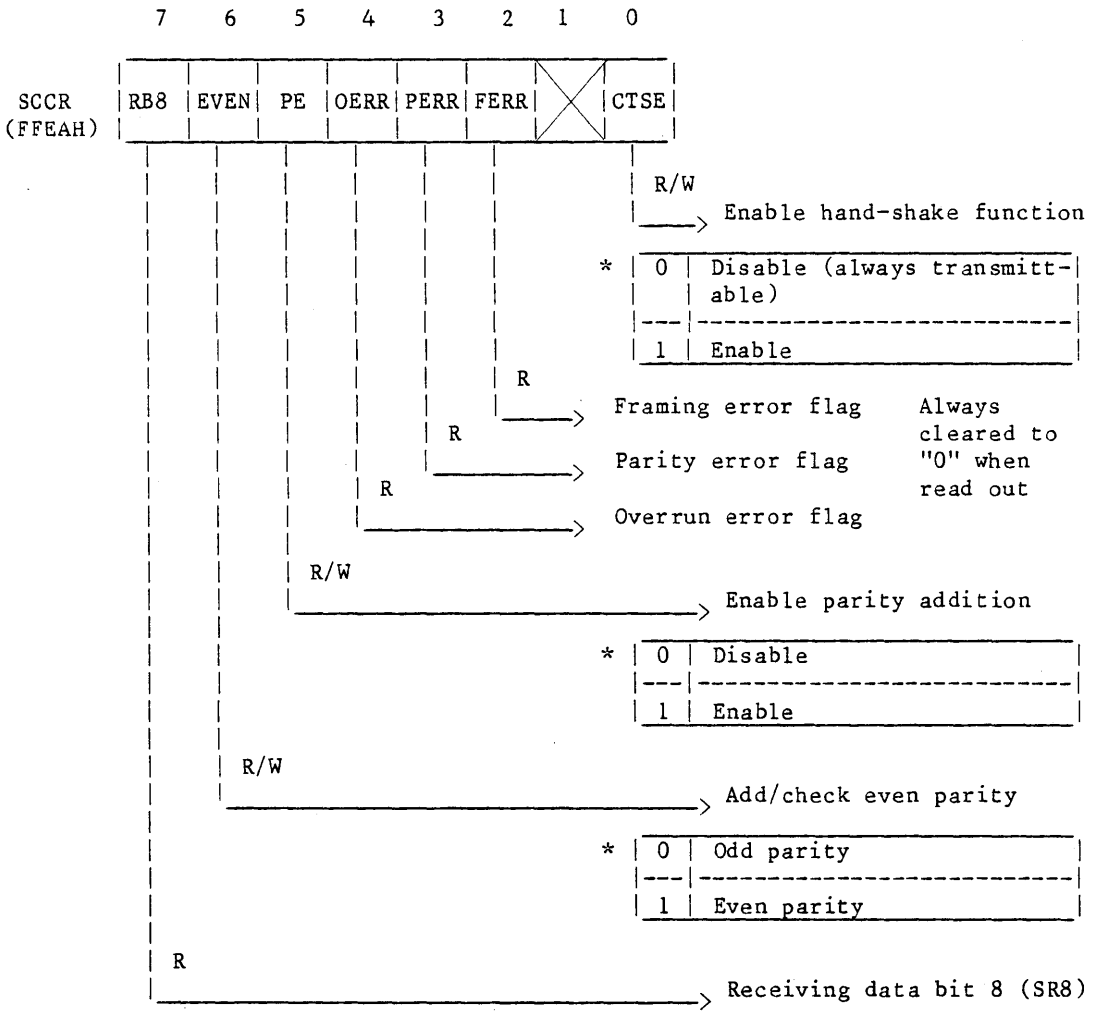


Fig. 3.8 (2) Serial Channel Mode Register



(Caution) Since all error flags are cleared after readout, avoid testing for only one bit by using a bit-testing instruction.

Fig. 3.8 (3) Serial Channel Control Register

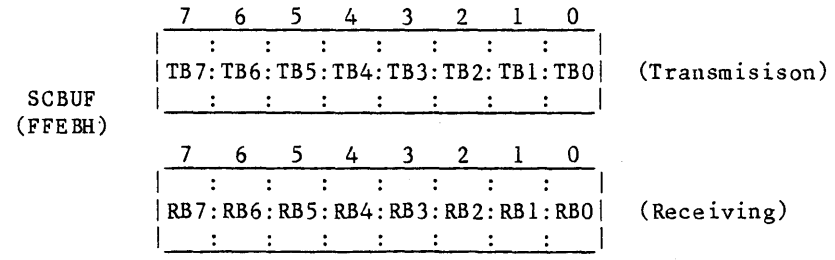
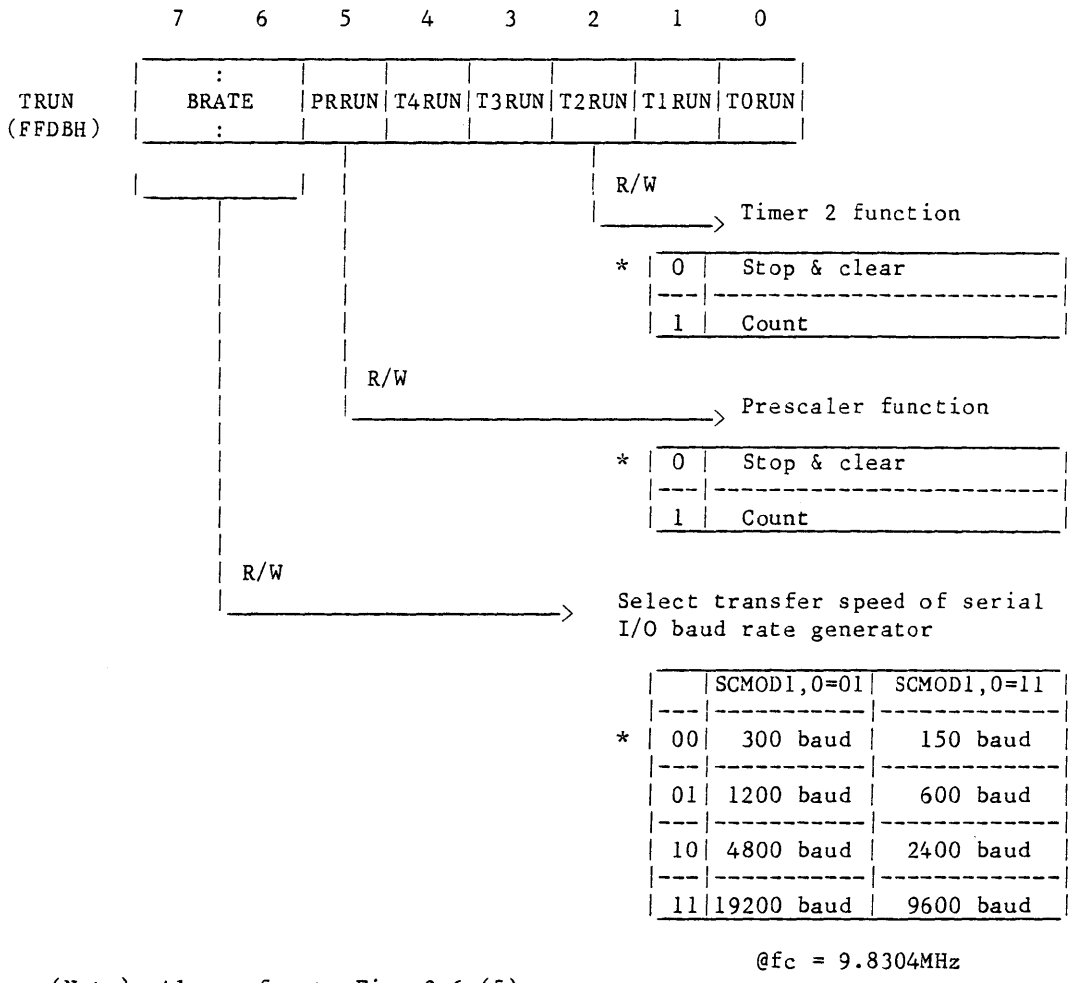


Fig. 3.8 (4) Serial Transmission/Receiving Buffer Registers



(Note) Also refer to Fig. 3.6 (5)

Fig. 3.8.(5) Timer/Serial Channel Operation Control Register

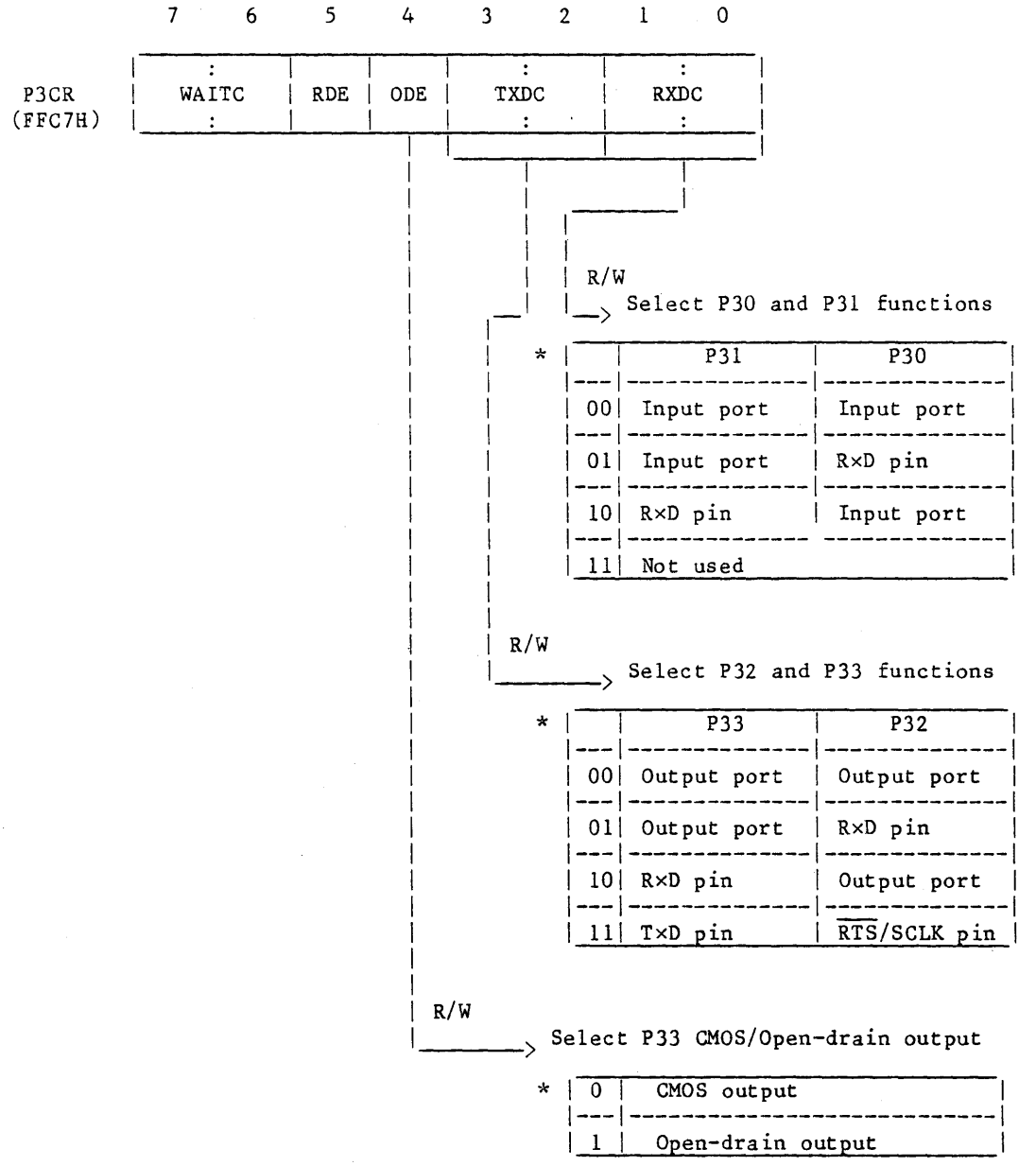


Fig. 3.8.(6) Port 3 Control Register

3.8.2 Architecture

Fig. 3.8 (7) is a block diagram of the serial channel.

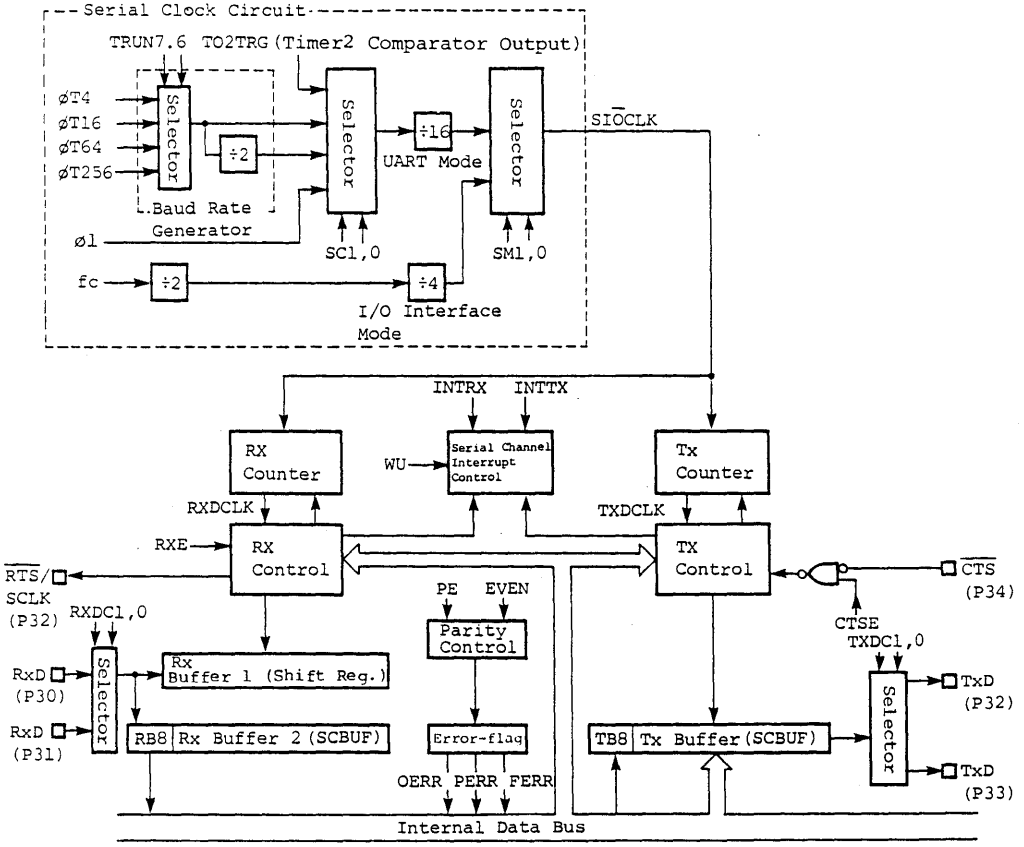


Fig. 3.8 (7) Block Diagram of Serial Channel

① Baud-rate generator

The baud-rate generator comprises a circuit that generates a clock pulse to determine the transfer speed for transmission/receiving in the asynchronous communication (UART) mode.

The input clock to the baud-rate generator øT4, øT16, øT64 or øT256 is generated by the 9-bit prescaler. One of these input clocks is selected by the timer/serial channel control register TRUN7, 6 (BRATE).

Also, either no frequency division or 1/2 division can be selected by the serial channel mode register SCMOD1, 0 (SC1, 0).

Table 3.8 (1) shows the baud-rate when fc = 9.8304 MHz.

Table 3.8 (1) Baud Rate Selection

BRATE	Input clock	No division(SC1,0=01)	1/2 division(SC1,0=11)
00	øT256	300 bps	150 bps
01	øT64	1200 bps	600 bps
10	øT16	4800 bps	2400 bps
11	øT4	19200 bps	9600 bps

@fc = 9.8304 MHz

② Serial clock generating circuit

This circuit generates the basic clock for transmitting and receiving data.

1) In case of I/O interface mode
   It generates a clock at a 1/8 frequency of the system clock (fc). This clock is output from the SCLK pin (also used as P32/RTS).
2) In case of asynchronous commumication (UART) mode
   A basic clock is generated based on the above baud rate generator clock, the internal clock o1, or the match signal from Timer 2, as selected by bits 1 and 0 of SCMOD register (SC).

③ Receiving counter

The receiving counter is a 4-bit binary counter used in the asynchronous communication (UART) mode and is counted by using SIOCLK. 16 pulses of SIOCLK is used for receiving 1 bit data. The data are sampled three times at 7th, 8th and 9th pulses and evaluated by the rule of majority. For example, if data sampled at the 7th, 8th and 9th clock are "1", "0" and "1", the recieved data is evaluated as "1". The sampled data "0", "0" and "1" is evaluated that the received data is "0".

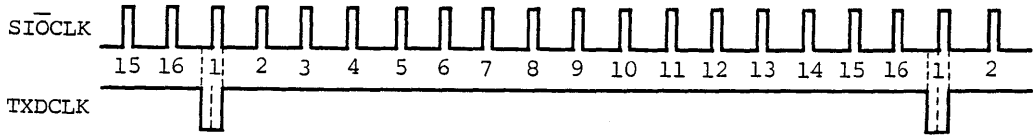
④ Receiving control

1) In case of I/O interface mode
   The RxD signal is sampled on the rising edge of the shift clock which is output to the SCLK pin.

2) In case of asynchronous communication (UART) mode
The receiving control features a circuit for detecting the start bit by the rule of majority. When two or more "0" are detected during 3 samples, it is recognized as normal start bit and the receiving operation is started.
Data being received are also evaluated by the rule of majority.

⑤ Receiving buffer
The receiving buffer has a double-buffer structure to prevent overruns. Received data are stored into the Receiving buffer 1 (shift register type) for each 1 bit. When 7 or 8 bits data are stored in the Receiving buffer 1, the stored data is transferred to the Receiving buffer 2 (SCBUF), and the interrupt INTRX occurs at the same time. The CPU reads out the Receiving buffer 2 (SCBUF). Data may be stored into the Receiving buffer 1 before the CPU reads out the Receiving buffer 2 (SCBUF). Note, however, that an overrun occurs unless the CPU reads out the Receiving buffer 2 (SCBUF) before the Receiving buffer 1 receives all bits of the next data. When an overrun occurred, the data in the buffer 2 and RB8 are not lost, however, that in the buffer 1 are lost.
SCCR7 (RB8) stores the parity bit in the case adding parity in the 8-bit UART mode and the MSB in the 9-bit UART mode.
In the 9-bit UART mode, setting SCMOD4 (WU) to "1" enables the wake-up function of the slave controllers, and the interrupt INTRX occurs only if RB8=1.

⑥ Transmission counter
This is a 4-bit binary counter used in the asynchronous communication (UART) mode. Like the receiving counter, it counts based on SIOCLK to generate a transmission clock TXDCLK for every 16 counts.



⑦ Transmission control
1) I/O interface mode
Data in the transmission buffer are output to the TxD pin for each bit at the rising edge of the shift clock output from the SCLK pin.
2) Asynchronous communication (UART) mode
When the CPU have written data into the transmission buffer, transmission is started with the next rising edge of TxDCLK, and a transmission shift clock TxDSFT is generated.

Hand-shake function

The TMP90C840 supports a hand-shake function, by which the connection of $\overline{CTS}$ of one TMP90C840 and $\overline{RTS}$ of the other TMP90C840 allows receiving/transmitting data on a frame basis to prevent overrun errors. This function is enabled or disabled by the control register SCCR0 (CTSE).

When the last bit (parity bit or MSB) of 1-frame data is received by the receiving unit, the $\overline{RTS}$ pin turns to the "H" level to request the transmission unit to halt transmission.
When the $\overline{CTS}$ pin turned to the "H" level, the transmission unit halts transmission, after completing the current data transmission, until the pin turns to the "L" level. At this time, the interrupt INTTX is generated, to request the CPU to transfer data. Then the data is written into the transmission buffer, and the CPU is placed in the standby mode.
When the received data are read by the receiving unit, the $\overline{RTS}$ pin returns to the "L" level, requesting that the transmission is restarted.

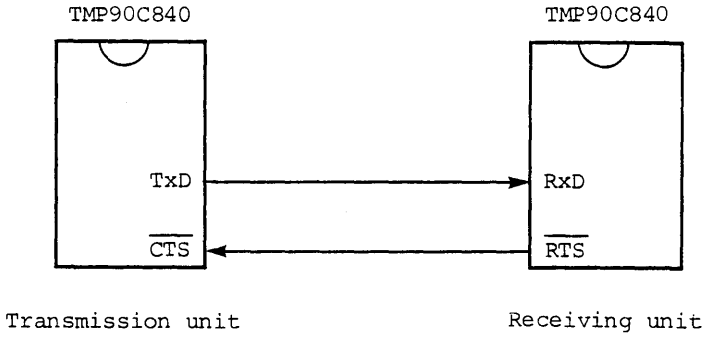
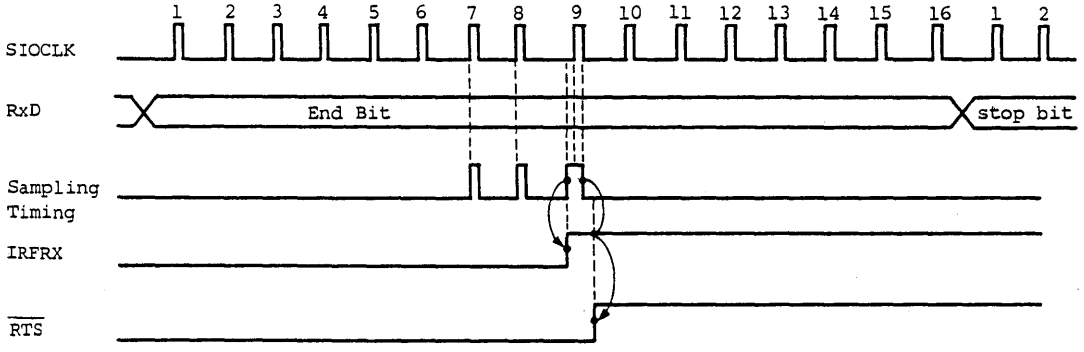
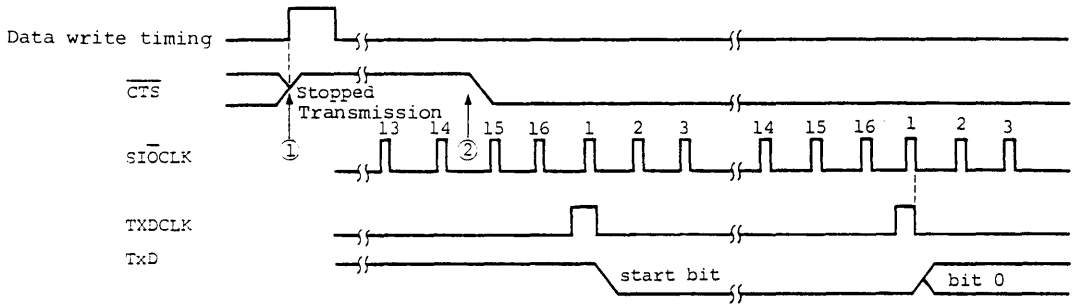


Fig. 3.8 (8) Hand-shake Function



(Note) In case of 8-bit asynchronous communication (UART), the last bit is the bit 7 in the non-parity mode, and the parity bit in the parity-added mode..

Fig. 3.8 (9) Timing Chart of $\overline{RTS}$ (request to send) Signal



(Note) 1) Keep the $\overline{CTS}$ signal at the "H" level until data are written into the transmission buffer (SCBUF).
A fall of the $\overline{CTS}$ signal halts the transmisison of the next data.
2) The transmission is restarted from the first fall of TXDCLK after a fall of the $\overline{CTS}$ signal.

Fig. 3.8 (10) Hand-shake by $\overline{CTS}$ (clear to send) Signal

⑧ Transmission buffer
The transmission buffer (SCBUF) shifts out the data written by the CPU from the LSB as based on the shift clock TXDSFT generated by the transmission control unit. When all bits are shifted out, the transmission buffer becomes empty, generating the interrupt INTTX.

⑨ Parity control circuit
Setting the serial channel control register SCCR5 (PE: Parity enable) to "1" allows the addition of a parity bit in transmitting/receiving data, only in the 7-bit UART or 8-bit UART mode. Either even or odd parity can be selected by the SCCR6 (EVEN) register.
In the transmission mode, the parity is automatically generated as based on the data written into the transmission buffer (SCBUF), storing into the 7th bit (TB7) of SCBUF in the 7-bit UART mode or into TB8 in the 8-bit UART mode for transmission. PE and EVEN should be designated before writing data into the transmission buffer.
In the receiving mode, the parity is generated from data shifted into the receiving buffer 1 and transferred to the receiving buffer 2 (SCBUF). A parity error is detected and the PERR flag is set if the parity status mismatches the 7th bit (RB7) of SCBUF in the 7-bit UART mode or RB8 in the 8-bit UART mode.

3.8.3 Operation
(1) Mode 0 (I/O interface mode)
This mode is used to increase the number of I/O pins of the TMP90C840 for transmitting/receiving data and supplying a synchronous clock (SCLK) to an external shift register.

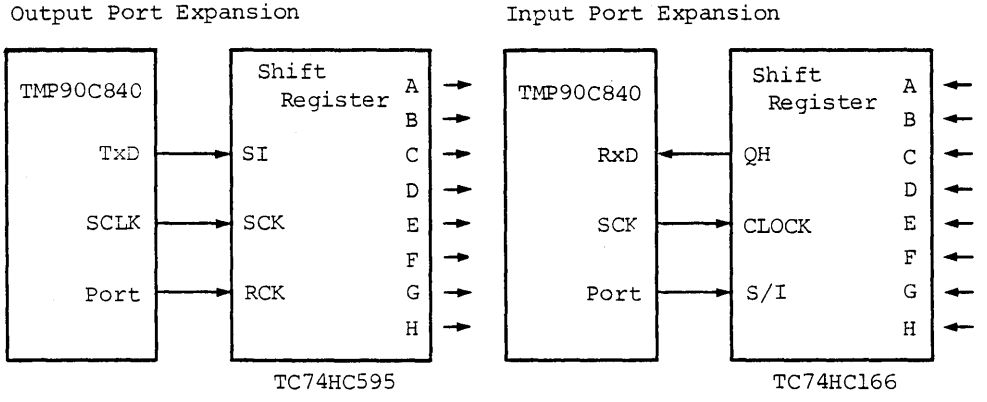


Fig. 3.8 (11) I/O Interface Mode

① Transmission

Each time the CPU writes data into the transmisison buffer, 8-bit data are output from TxD pin. When all data are output, IRFTX is set, and the interrupt INTTX occurs.

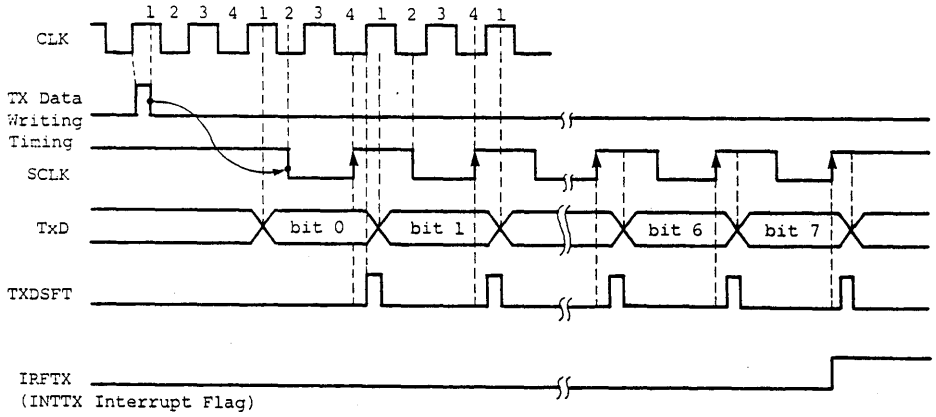


Fig 3.8 (12) Transmitting Operation (I/O Interface Mode)

② Receiving

Each time the CPU reads the receiving data and clears the receiving interrupt flag IRFRX, the next data are shifted into the receiving buffer 1. When 8-bit data are received, the data are transferred to the receiving buffer 2 (SCBUF), which sets IRFRX and generates interrupt INTRX.

For receiving data, the receiving enable state is previously set (RxE=1).

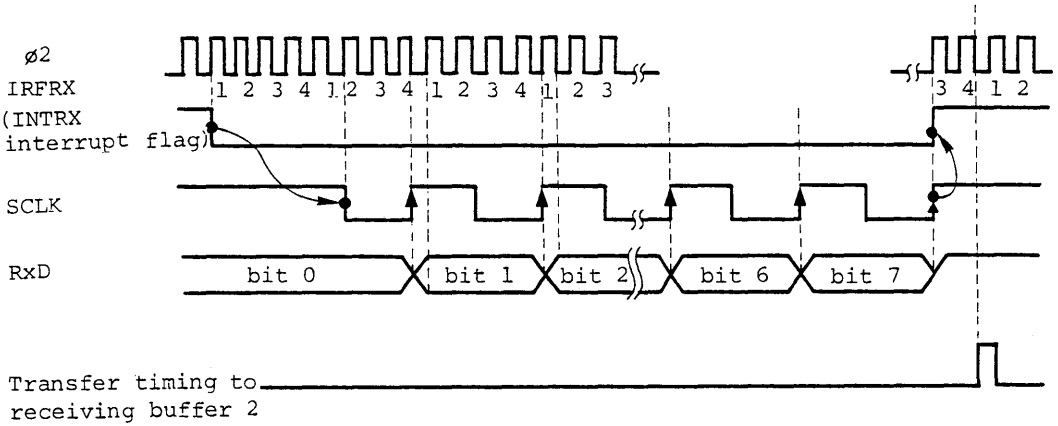


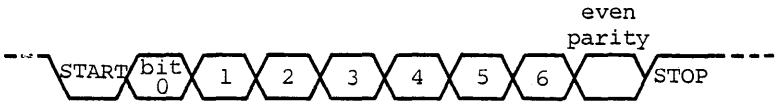
Fig 3.8 (13) Receiving Operation (I/O Interface Mode)

(2) Mode 1 (7-bit UART mode)

The 7-bit UART mode is obtained by setting the serial channel mode register SCMOD3 2 (SM) to "01".

This mode allows the addition of a parity bit, which is enabled or disabled by the serial channel control register SCCR5 (PE). When PE = 1 (enable), even or odd parity can be selected by SCCR6 (EVEN).

Example: When transmitting data with the following format, the control registers should be set as described below.



(Baud Rates 2400 baud @fc = 9.8304 MHz)

← Transfer direction

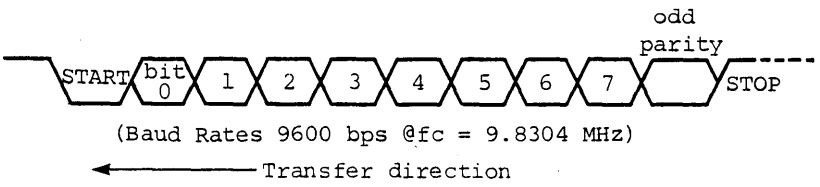
P3CR	<- - - - - 0 1 - -	Select P32 as the TxD pin.
SCMOD	<- x 0 - x 0 1 1 1	Set the transfer speed at 2,400 bps in
TRUN	<- 1 0 1 - - - - -	the 7-bit UART mode.
SCCR	<- x 1 1 x x x x 0	Add an even parity.
INTEL	<- - - - - - - - 1	Enable INTTX interrupt.
SCBUF	<- * * * * * * * *	Set data for transmission.

(Note) x: Don't care - : No change

(3) Mode 2 (8-bit UART mode)

The 8-bit UART mode is obtained by setting SCMOD3 and SCMOD2 (SM) to "10". This mode also allows the addition of a parity bit, as enabled or disabled by SCCR5 (PE). When PE = 1 (enable), even or odd parity can be selected by SCCR6 (EVEN).

Example: When receiving data with the following format, the control registers should be set as described below.



Main setting:

```
P3CR  <- - - - - - - 0 1   Select P30 as the RxD pin.
SCCR  <- x 0 1 x x x x 0   Add an odd parity.
TRUN  <- 1 1 1 - - - - -   Set the transfer speed at 9,600 bps in
SCMOD <- - 0 1 x 1 0 1 1   the 8-bit UART mode.
INTEL <- - - - - - - 1 -   Enable INTTX interrupt.
```

INTRX processing

```
Acc   <- SCCR ^ 00011100 Check errors.
If Acc = 0 then erro
Acc   <- SCBUF           Read out the received data.
```

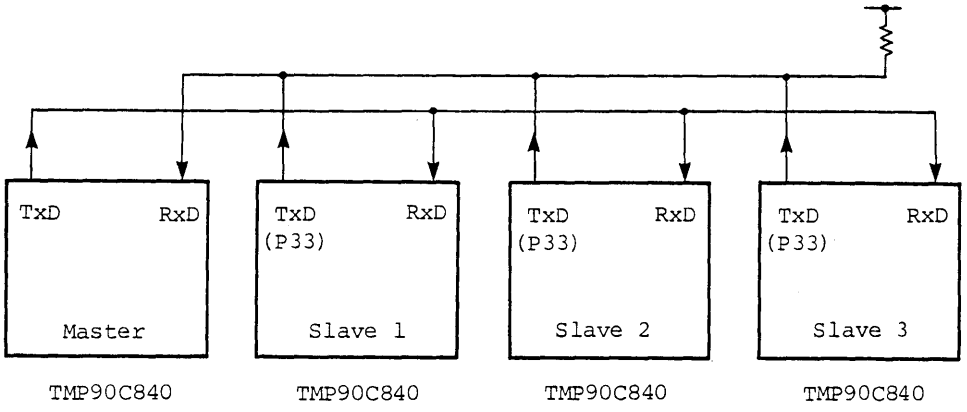
(Note) x; Don't care - ; No change

(4) Mode 3 (9-bit UART mode)

The 9-bit UART mode is obtained by setting SCMOD3, 2 (SM) = "11". The addition of a parity bit is disabled in this mode.
The MSB (9th bit) is written into TB8 for transmission, and into RB8 for receiving. Writing into or reading from the buffer must begin with the MSB followed by SCBUF.

Wake-up function

In the 9-bit UART mode, setting SCMOD4 (WU) to "1" allows the wake-up operation of the slave controllers. The interrupt INTRX occurs only when RB8 = 1.

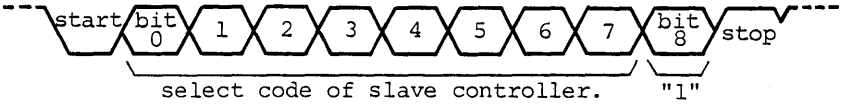


(Note) For the wake-up operation, P33 should be always selected as the TxD pin of the slave controllers, and put in the open drain output mode.

Fig. 3.8 (14) Serial Link Using Wake-up Function MPU90-137

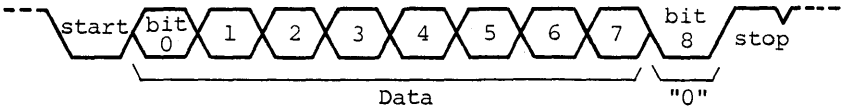
|Protocol|

① Select the 9-bit UART mode for the master and slave controllers.
② Set the WU bit of each slave controller to "1" to enable data receiving.
③ The master controller transmits 1-frame data including the 8-bit select code for the slave controllers. The MSB (bit 8) is set to "1".



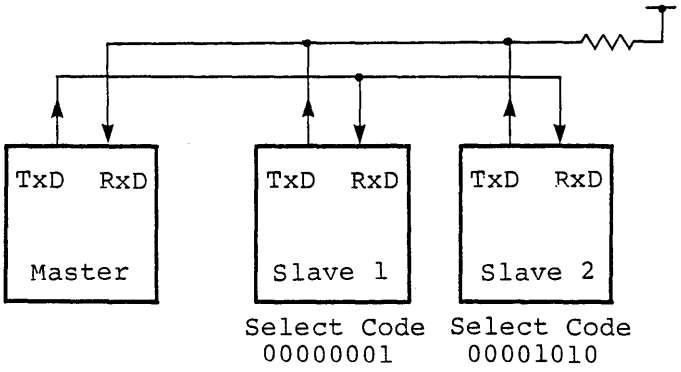
④ Each slave controller receives the above frame, and clears the WU bit to "0" if the above select code matches its own select code.

⑤ The master controller transmits data to the specified slave controller (whose WU bit is cleared to "0"). At the same time, the MSB (bit 8) is cleared to "0".



⑥ The other slave controllers (with the WU bit remaining at "1") ignore the receiving data because their MSBs (bit 8 or RB8) are set to "0" to disable the interrupt INTRX.
When the WU bit is cleared to "0", the interrupt INTRX occurs, making it possible to read the receiving data.
Only when WU = 0, the slave controllers can transmit data to the master controller, including those indicating the end of data receiving.

Example: Link two slave controllers serially with the master controller, and use the internal clock o1 as the transfer clock.



o Set the master control

Main

```
P3CR  <- - - - 0 0 1 1 0   Select P32 as TxD and P31 as RxD.
INTEL <- - - - - - - 1 1   Enable INTRX and INTTX.
SCCR  <- x x x x x x x 0   Disable the hand-shake function.
SCMOD <- 1 0 1 0 1 1 1 0   Select o1 as the transfer clock in the
                           9-bit UART mode.
SCBUF <- 0 0 0 0 0 0 0 1   Set the select code for the slave con-
                           troller 1.
```

INTTX interrupt

```
SCMOD <- 0 - - - - - - -   Set TB8 to "0".
SCBUF <- * * * * * * * *   Set data for transmission.
```

o Set the slave controller 2

Main

```
P3CR  <- - - - 1 1 0 1 0   Select P33 as TxD and P31 as RxD.
INTEL <- - - - - - - 1 1   Enable INTRX and INTTX.
SCCR  <- x x x x x x x 0   Disable the hand-shake function.
SCMOD <- 0 0 1 1 1 1 1 0   Set WU to 1 in the 9-bit UART mode
                            (transfer clock: o1) .
```

INTRX interrupt

```
┌ Acc   <- SCBUF
│ if Acc = Select code
└ then SCMOD  <- - - - 0 - - - -     Clear WU to "0".
```

(Note) ×: Don't care - : No change

3.9 Analog/Digital Converter

The TMP90C840 incorporates a high-speed and high-precision analog/digital converter (A/D converter) with six analog input channels that features f8-bit sequential comparison.
Fig. 3.9 (1) is a block diagram of A/D converter. The 6-channel analog input pin (AN5 - AN0) is also used as the input port P5.

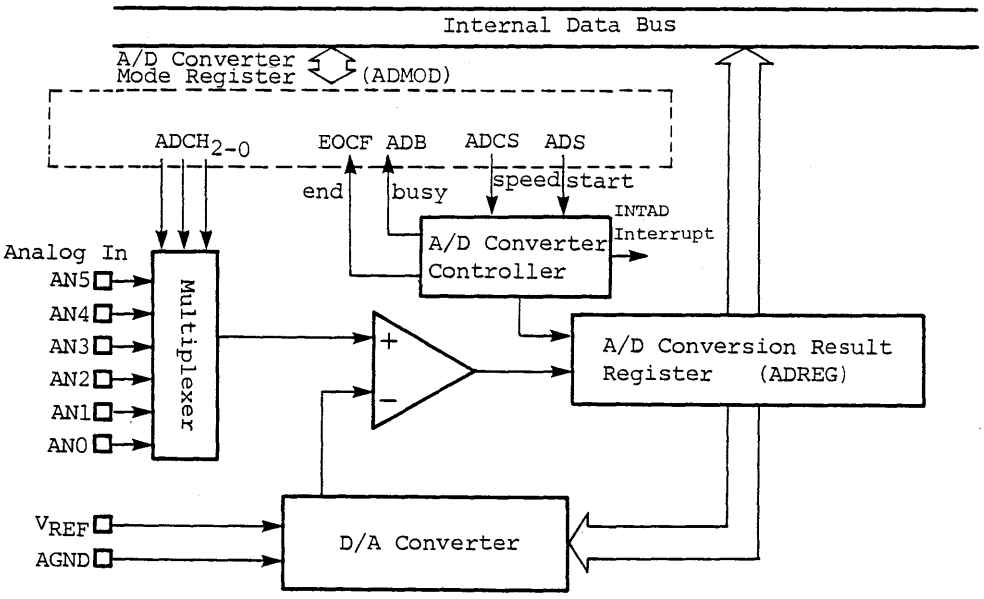


Fig. 3.9 (1) Block Diagram of A/D Converter

3.9.1 Control Registers

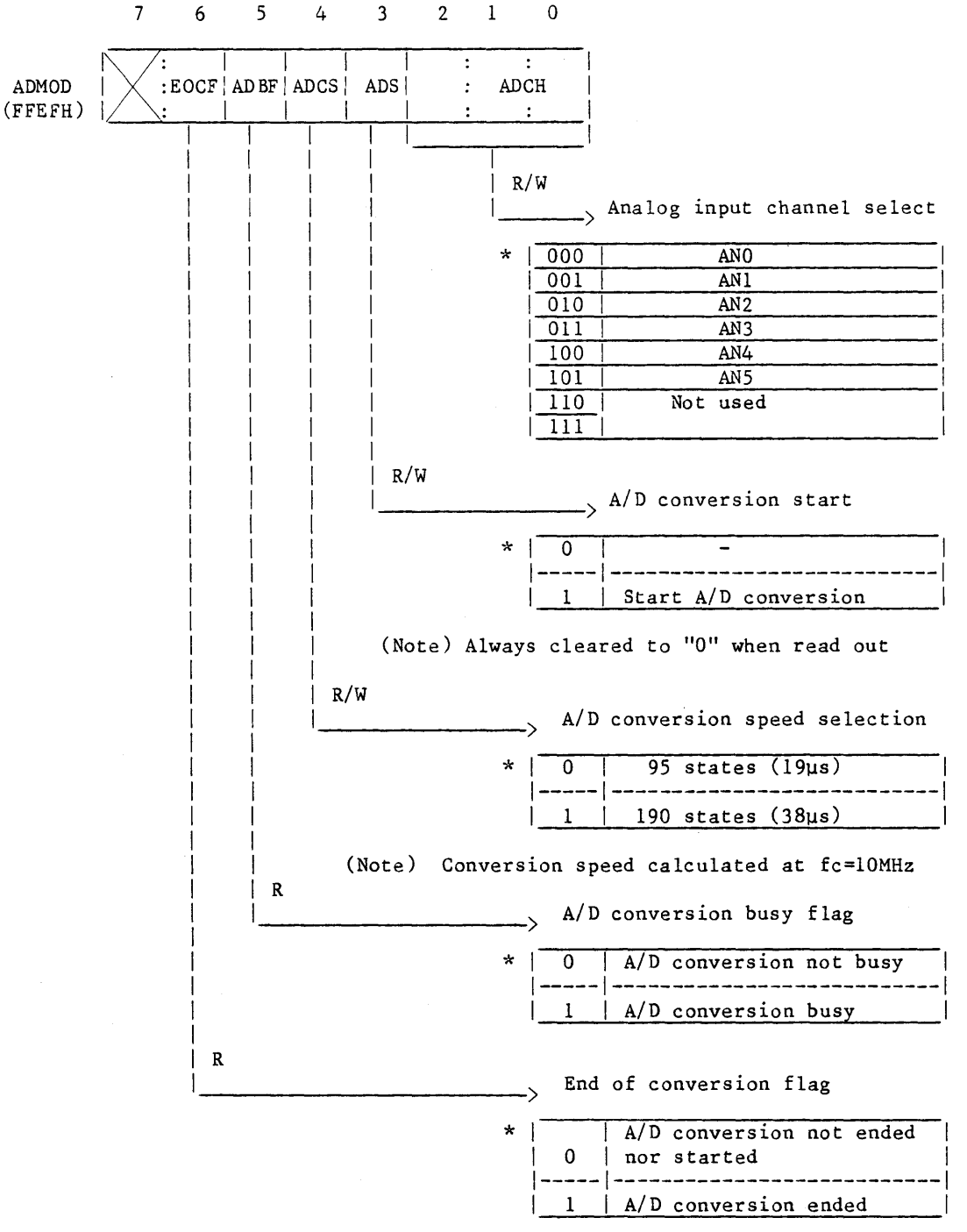


Fig. 3.9 (2) A/D Converter Mode Register

	MSB 7	6	5	4	3	2	1	LSB 0
ADREG (FFEEH)	:	:	:	:	:	:	:	:

A/D conversion result register (Read only)

Fig. 3.9 (3) A/D Conversion Result Register

3.9.2 Operation

(1) Analog reference voltage
The high analog reference voltage is applied to the VREF pin, and the low analog reference voltage is applied to the AGND pin.
The reference voltage between VREF and AGND is divided by 256 by connecting ladder resistance, and compared with the analog input voltage for A/D conversion.

(2) Analog input channels
For the A/D conversion, one of the six analog input channels (AN0 (P50) to AN5 (P55)) is selected by the register ADMOD2-0 (ADCH).
ADCH is initialized to "000" by resetting, whereby the AN0 pin is selected.
The pins not used for the analog input can be used for ordinary input (P5).

(3) Selection of A/D conversion speed
Normally, the A/D converter is used in the high-speed conversion mode that completes the operation in 95 states (19 μs @fc=10MHz).
When using an oscillation frequency of more than 10MHz, set the register ADMOD4 (ADCS) to "1" and obtain the conversion speed of 190 states (38 μs, @fc=10MHz).
ADCS is initialized to "0" by resetting, by which the A/D converter turns to the high-speed conversion mode.

(4) Starting A/D conversion
The A/D conversion is started by writing "1" into the register ADMOD3 (ADS). When the A/D conversion is started, the ADMOD5 (ADBF) flag is set to "1", indicating that the conversion is in progress.

(5) A/D conversion end and interrupt
When the A/D conversion is completed, the ADMOD6 (EOCF) flag is set to "1", indicating that the A/D conversion is completed. Then the ADBF flag is cleared to "0" to generate the interrupt INTAD.
INTAD (interrupt by the A/D converter) is controlled by the interrupt enable flag INTEL 7 (IET2), also used for INTT2 (interrupt by Timer 2). Either INTAD or INTT2 is selected by the INTEH3 (ADIS) flag. To generate INTAD interrupt, both IET2 and ADIS should be set to "1".
Both INTAD and INTT2 jump to the same vector address (0040H), but can be distinguished by the ADIS flag.
When INTAD is latched, the INTAD status cannot be cleared by software.

(6) Reading A/D conversion results

The results of A/D conversion are stored into the ADREG register. By reading the contents of the ADREG register, EOCF flag is cleared to "0".

When the contents of the ADREG register is read while the A/D conversion is performed, the A/D conversion results become undefined. A/D conversion is stopped by a HALT instruction except in the RUN mode, by which the A/D conversion results also become undefined. In the RUN mode, A/D conversion is not stopped by a HALT instruction.

Example: [1] Analog input voltage to the AN3 pin is converted to a digital value in the high-speed conversion mode (95 states), and the result is stored into the memory address FF10H by using A/D interrupt INTAD service routine.

Main setting

```
INTEH <- - - - - 1 - - -   Set INTADE to "1".
INTEL <- 1 - - - - - - -   Enable INTT2
ADMOD <- x x x 0 1 0 1 1   Select AN3 as the analog input channel,
                           and start the A/D conversion in the high-
                           speed conversion mode.
```

A/D interrupt service routine

```
A      <-     ADREG       Load the contents of ADREG into the ac-.
(FF10H)  <-     A         cumulator. Store the accumulator value
                          into the memory address FF10H.
```

Example: [2] Analog input voltage to the AN2 pin is converted to a digital value in the high-speed conversion mode (95 states), and the result is loaded into the accumulator when the end of conversion is detected by the EOCF flag.

```
ADMOD <- x x x 0 1 0 1 0   Select AN2 as the analog input channel,
                           and start the A/D conversion in the high-
                           speed conversion mode.
loop:
  if EOCF = 0 then loop
              else A <- ADREG
```

(Note) x: Don't care - : No change

3.10 Watchdog timers (Runaway detecting timer)

When the mulfunction (runaway) of the CPU occurs due to any cause such as noise, the watchdog timer (WDT) detects it to return to the normal sate. When WDT has detected malfunction, a non-maskable interrupt is generated to indicate it to the CPU.

3.10.1 Architecture

Fig. 3.10 (1) is a block diagram of the watchdog timer (WDT).
The watchdog timer consists of a 20-stage binary counter (input clock: o @fc/2), a flip-flop that disables/enables the selector, a selector that selects one of the four output clocks generated from the binary counter, and two control registers.

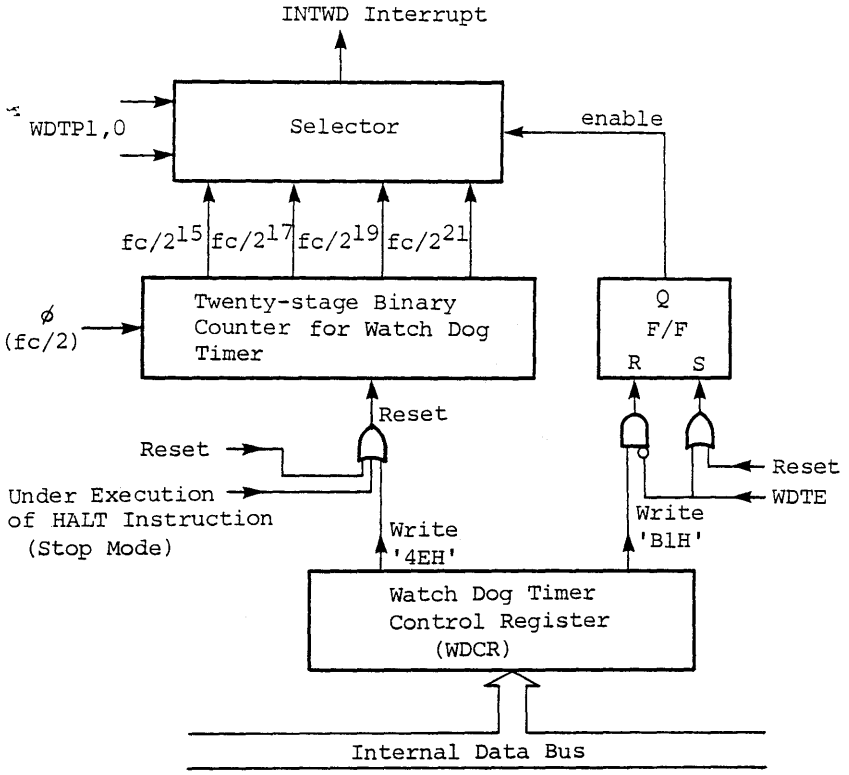


Fig. 3.10 (1) Block Diagram of Watchdog Timer

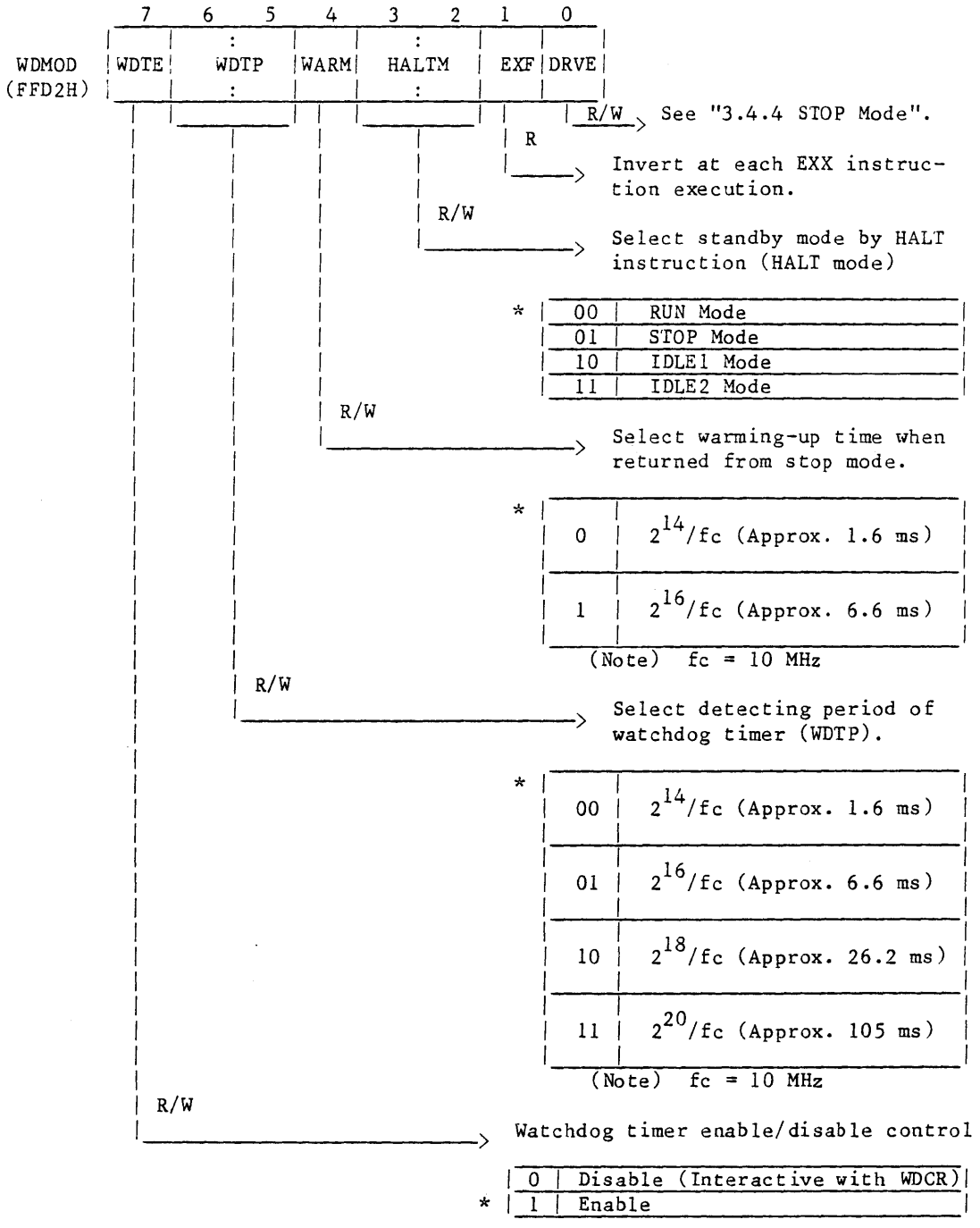
3.10.2 Control Registers

WDT is controlled by two control registers (WDMOD and WDCR).

(1) Watchdog timer mode register (WDMOD)

1 Set the detecting time of watchdog timer (WDTP)
The WDT interrupt period is set by this 2-bit flag. WDTP is initialized to "00" by resetting, providing the initial set value of 2^{14}/fc (sec.) (approx. 8,192 states).

② WDT enable/disable control (WDTE)
WDTE is initialized to "1" by resetting, which enables the watchdog timer function.
To disable the function, the bit should be cleared to "0" and the disable code "B1H" should be written into the WDCR register. By using this dual procedure, it becomes hard to disable the WDT even if the malfunction occurs.
The disable state can be returned to the enable state by setting the WDTE bit to "1".



(Note) To disable the WDT function, the disable code should be written into the WDCR register.

Fig. 3.10 (2) Watchdog Timer Mode Register

(2) Watchdog timer control register (WDCR)

This register is used to disable the watchdog timer function and clear the binary counters.

o Disable WDT

The watchdog timer can be disabled by, after clearing WDMOD7 (WDTE) to "0", writing the disable code (B1H) into this WDCR register.

```
WDMOD  <- 0-----XX    Clear WDTE to "0".
WDCR   <- 10110001    Write disable code (B1H).
```

o Clear binary counter

The binary counter can be cleared and resume counting by writing the clear code (4EH) into the WDCR register.

```
WDCR   <- 01001110    Write clear code (4EH).
```

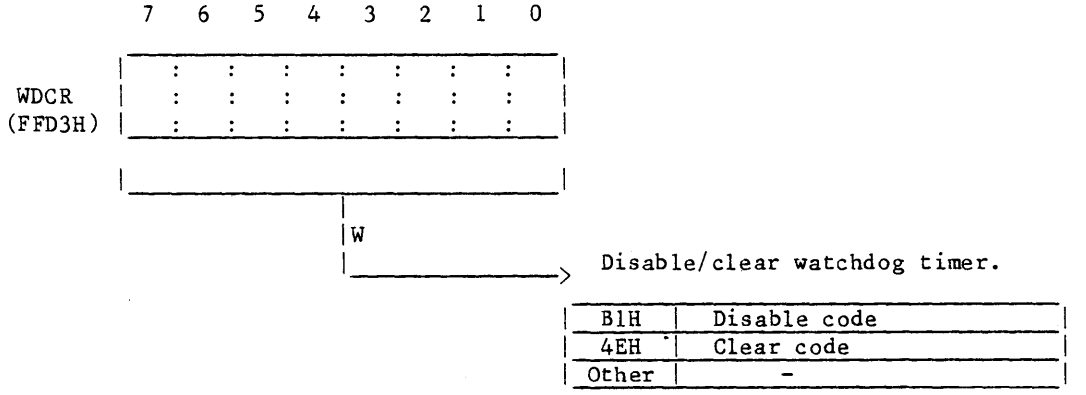


Fig. 3.10 (3) Watchdog Timer Control Register

3.10.3 Operation

The watchdog timer generates INTWD (watchdog timer interrupt) after a time specified by the register WDMOD6, 5 (WDTP). The binary counter for the watchdog timer is cleared to "0" by software (instruction) before the interrupt occurs. If the CPU caused a malfunction (runaway) for reason such as noise and fails to execute the instruction to clear the binary counter, the counter will overflow and the watchdog timer interrupt INTWD occurs. The CPU detects the malfunction (runaway) by this interrupt.

The watchdog timer starts its operation as soon as the reset state is cleared.

The watchdog timer stops its operation only in the STOP mode. When the STOP mode is released, the watchdog timer starts its operation after a specified warming-up time.

In the other standby mode (IDLE 1, IDLE 2 or RUN modes), the watchdog timer is enabled. However, the function can be disabled before selecting any of these modes.

Example: (1) Clear the binary counter.

```
WDCR  <- 01001110    Write clear code (4EH)
```

(2) Set 2^{16}/fc for the detecting time of watchdog timer.

```
WDMOD <- 101---XX
```

(3) Disable the watchdog timer.

```
WDMOD <- 0-----XX    Clear WDTE to "0"
WDCR  <- 10110001    Write disable code (B1H)
```

(4) Select the IDLE 2 mode.

```
WDMOD <- 0---11XX    Disable WDT and set IDLE 2 mode
WDCR  <- 10110001    Select the standby mode
Execute HALT instruction
```

(5) Select the STOP mode (Warming-up time: 2^{16}/fc)

```
WDMOD <- ---101XX    Select STOP mode
Execute HALT instruction
                     Select the standby mode
```

4. ELECTRICAL CHARACTERISTICS

TMP90C840N/TMP90C840F/TMP90841N

4.1 Absolute maximum ratings

Symbol	Parameter	Rating	Unit
VCC	Supply voltage	-0.5 - +7	V
VIN	Input voltage	-0.5 - Vcc +0.5	V
PD	Power dissipation (Ta=85°C)	250	mV
TSOLDER	Soldering temperature	260	°C
TSTG	Storage temperature	-65 - 150	°C
TOPR	Operating temperature	-40 - 85	°C

4.2 DC characteristics

TA=-40 ~ 85°C Vcc=5V±10%

Symbol	Parameter	Min.	Max.	Unit	Test Condition
VIL	Input Low Voltage (PO)	-0.3	0.2Vcc-0.1	V	
VIL1	P1,P2,P3,P4,P5,P6,P7,P8	-0.3	0.3Vcc	V	
VIL2	/RESET,INT0,/NMI,×1,×2	-0.3	0.25Vcc	V	
VIL3	/EA	-0.3	0.3	V	
VIH	Input High Voltage (PO)	0.2Vcc+1.1	Vcc+0.3	V	
VIH1	P1,P2,P3,P4,P5,P6,P7,P8	0.7Vcc	Vcc+0.3	V	
VIH2	/RESET,INT0,/NMI,×1,×2	0.75Vcc	Vcc+0.3	V	
VIH3	/EA	Vcc-0.3	Vcc+0.3	V	
VOL	Output Low Voltage		0.45	V	IOL=1.6mA
VOH	Output High Voltage	2.4		V	IOH=-400UA
VOH1		0.75Vcc		V	IOH=-100UA
VOH2		0.9Vcc		V	IOH=-20UA
IDAR	Darlington Drive Current (8I/O pins)	-1.0	-5.0	mA	VEXT=1.5V REXT=1.1kΩ
ILI	Input Leakage Current		±5	UA	0.2<Vin<Vcc -0.2
ILO	Output Leakage Current		±10	UA	0.2<Vin<Vcc -0.2
ICC	Operating Current (RUN)		40	mA	tosc=10MHz
	Idle 1		5	mA	
	Idle 2		15	mA	
	STOP		TBD	UA	0.2<Vin<Vcc -0.2
V STOP	Power Down Voltage	TBD	6	V	Vcc MIN: RAM BACK UP
R REST	/RESET Pull Up Register	50	150	Kohm	
CIO	Pin Capacitance		10	PF	testfreq= 1MHz
VTH	Schmitt width/RESET,/NMI, INT0	0.4		V	

4.3 AC power characteristics

TA=-40 - 85°C Vcc=5V±10%
CL=50PF

Symbol	Parameter	10MHz Clock		Variable		Units
		Min.	Max.	Min.	Max.	
tOSC	OSC. Period=x	100		100	1000	ns
tCYC	CLK Period	400		4x	4x	ns
tWL	CLK Low width	160		2x-40		ns
tWH	CLK High width	160		2x-40		ns
tAC	Address Setup to /RD, /WR	50		x-50		ns
tRR	/RD Low width	210		3x-90		ns
tCA	Address Hold Time After/RD./WR	30		30		ns
tAD	Address to Valid Data In		240		4x-160	ns
tRD	/RD to Valid Data In		170		3x-130	ns
tDR	Data Setup to/RD	40		40		ns
tHR	Input Data Hold After /RD	0		0		ns
tWW	/WR Low width	210		3x-90		ns
tDW	Data Setup to /WR	150		2x-50		ns
tWD	Data Hold After /WR	30	90	30	x-10	ns
tCWA	/RD,/WR to Valid /WAIT	50		2x-150		ns
tAWA	Address to Valid /WAIT	120		3x-180		ns
tWAS	/WAIT Setup to CLK	70		70		ns
tWAH	/WAIT Hold After CLK	0		0		ns
tRV	Recovery Time	90		x-10		ns
tCPW	CLK to Port Data Output		300		x+200	ns
tPRC	Port Data Setup to CLK	200		2x		ns
tCRP	Port Data Hold After CLK	100		x		ns

- o AC output level High 2.2V / Low 0.8V
- o AC input level High 2.4V / Low 0.45V (D0-0.7)
  High 0.8Vcc/ Low 0.2Vcc (excluding D0-D7)

4.4 A/D conversion characteristics

TA=-40 - 85°C Vcc=5V±10%

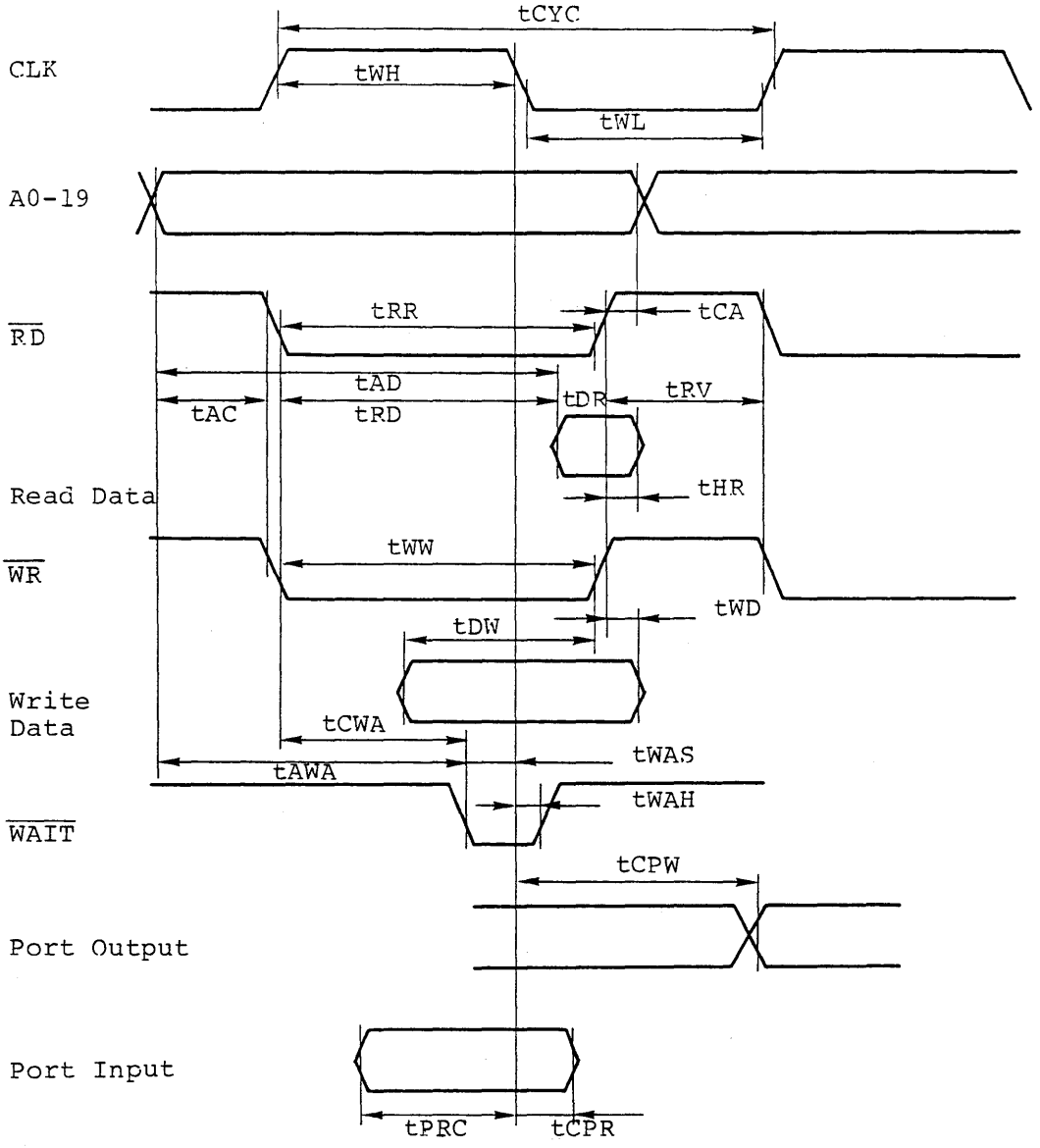
Symbol	Parameter	Min.	TYP.	Max.	Unit
VREF	Analog reference voltage	VCC-1.5	VCC	VCC	V
AGND	Analog reference voltage	VSS	VSS	0.5	
VREF-AGND	Allowable analog reference voltage	2.5			
VAIN	Allowable analog input voltage	VSS		VCC	
IREF	Supply current for analog reference voltge		0.5	1.0	mA
	Nonlinear error			TBD	LSB
	Zero error			TBD	
	Full-scale error			TBD	
	Total error			TBD	

4.5 Zero-cross characteristics

TA=-40 - 85°C Vcc=5V±10%

Symbol	Parameter	Condition	MIN.	Max.	Unit
Vzx	Zero-cross detection input	AC connection C=0.1μF	1	TBD	VAC p-p
Azx	Zero-cross accuracy	50/60Hz sine wave		TBD	mV
Fzx	Zero-cross detection input frequency		0.04	1	KHz

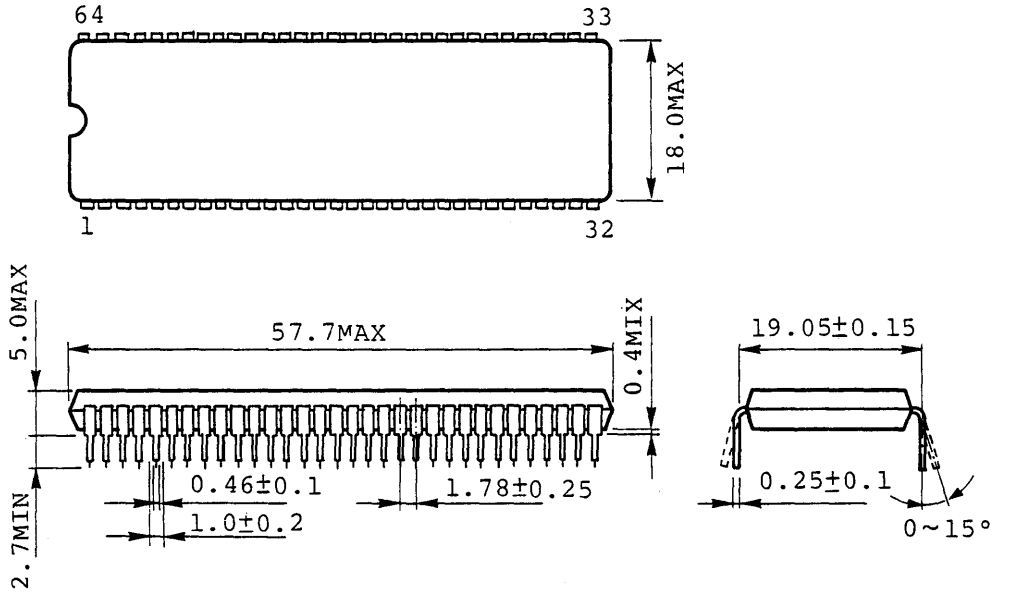
4.6 Timing chart



5. OUTSIDE DIMENSIONS

5.1 DIP package

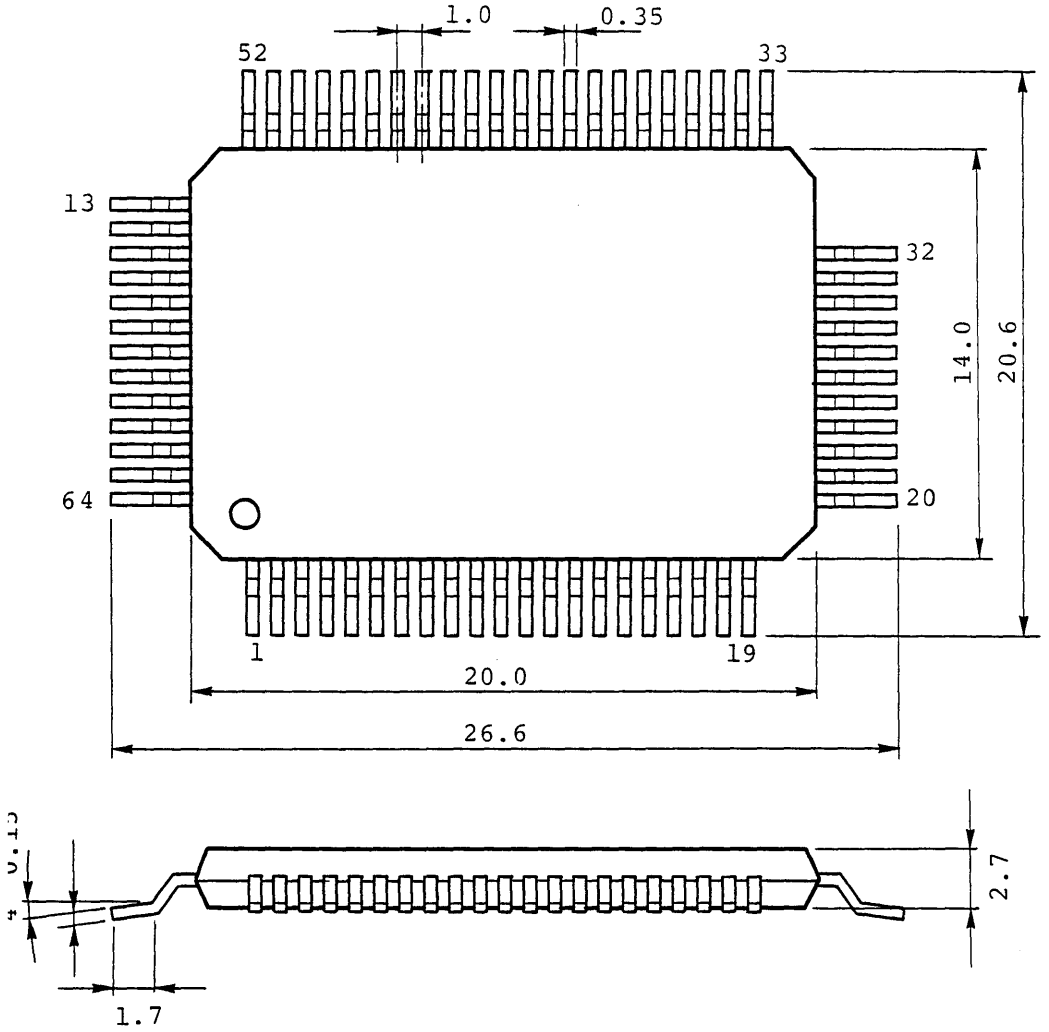
Unit: mm



Note) Lead pitch: 1.78
Tolerance: ±0.25 to the theoretical center of each lead obtained as based on the No. 1 and No. 64 pins.

5.2 Mini-flat package

Unit: mm



6. CAUTIONS

Followings are some important cautions for the user of the TMP90C840:

(1) An undefined or write-only bit of the internal I/O register is always set to "1" when read out.

(2) The write-only bit is always calculated as "1" when internal I/O register containing the write-only bit is accessed by using a read-modify-write instruction (operation between memory and immediate address, bit set/reset instruction, etc.).

(3) The system adopts the pipeline processing in which it prefetches a 1 byte instruction. Thus a jump instruction may cause the system to fetch the instruction that follows the jump instruction. This may also occur for call or return instruction.

(4) If an undefined code is executed, no subsequent operation will be guaranteed.

(5) The internal watchdog timer is returnd to the "enable" mode by resetting. If this function is not required, the watchdog timer should be placed in the "disable" mode.

(6) After the power is turned on, no internal device can be reset until the oscillator starts its operation, placing the output and I/O ports in the undefined state. In order to define their status, a separate external circuit is required.

APPENDIX

(TLCS-90)

Appendix A Table of Machine Instructions (1/11)

1. 8-bit transfer

Instruction	Mnemonic		Code	Function	SZIHXVNC	T
	LD	A,r	20+r	A<-r	--------	2
LD r,r	LD	r,A	28+r	r<-A	--------	2
	LD	r,g	F8+g:30+r	r<-g	--------	4
LD r,n	LD	r,n	30+r:n	r<-n	--------	4
	LD	A,(n)	27:n	A<-(n)	--------	8
	LD	r,(gg)	E0+gg:28+r	r<-(gg)	--------	6
LD r,mem	LD	r,(ix+d)	F0+ix:d:28+r	r<-(ix+d)	--------	10
	LD	r,(HL+A)	F3:28+r	r<-(HL+A)	--------	14
	LD	r,(mn)	E3:n:m:28+r	r<-(mn)	--------	10
	LD	r,(n)	EF:n:28+r	r<-(n)	--------	8
	LD	(n),A	2F:n	(n)<-A	--------	8
	LD	(gg),r	E8+gg:20+r	(gg)<-r	--------	6
LD mem,r	LD	(ix+d),r	F4+ix:d:20+r	(ix+d)<-r	--------	10
	LD	(HL+A),r	F7:20+r	(HL+A)<-r	--------	14
	LD	(mn),r	EB:n:m:20+r	(mn)<-r	--------	10
	LD	(n),r	EF:n:20+r	(n)<-r	--------	8
	LD	(gg),n	E8+gg:37:n	(gg)<-n	--------	8
	LD	(ix+d),n	F4+ix:d:37:n	(ix+d)<-n	--------	12
LD mem,n	LD	(HL+A),n	F7:37:n	(HL+A)<-n	--------	16
	LD	(vw),n	EB:w:v:37:n	(vw)<-n	--------	12
	LD	(w),r	37:w:n	(w)<-n	--------	10

2. 16-bit transfer

Instruction	Mnemonic		Code	Function	SZIHXVNC	T
	LD	HL,rr	40+rr	HL<-rr	--------	4
LD rr,rr	LD	rr,HL	48+rr	rr<-HL	--------	4
	LD	rr,gg	F8+gg:38+rr	rr<-gg	--------	6
LD rr,nn	LD	rr,mn	38+rr:n:m	rr<-mn	--------	6
	LD	HL,(n)	47:n	HL<-(n)W	--------	10
	LD	rr,(gg)	E0+gg:48+rr	rr<-(gg)W	--------	8
LD rr,mem	LD	rr,(ix+d)	F0+ix:d:48+rr	rr<-(ix+d)W	--------	12
	LD	rr,(HL+A)	F3:48+rr	rr<-(HL+A)W	--------	16
	LD	rr,(mn)	E3:n:m:48+rr	rr<-(mn)W	--------	12
	LD	rr,(n)	E7:n:48+rr	rr<-(n)W	--------	10
	LD	(n),HL	4F:n	(n)W<-HL	--------	10
	LD	(gg),rr	E8+gg:40+rr	(gg)W<-rr	--------	8
LD mem,rr	LD	(ix+d),rr	F4+ix:d:40+rr	(ix+d)W<-rr	--------	12
	LD	(HL+A),rr	F7:40+rr	(HL+A)W<-rr	--------	16
	LD	(mn),rr	EB:n:m:40+rr	(mn)W<-rr	--------	12
	LD	(n),rr	EF:n:40+rr	(n)W<-rr	--------	10
	LDW	(gg),mn	E8+gg:3F:n:m	(gg)W<-mn	--------	12
	LDW	(ix+d),mn	F4+ix:d:3F:n:m	(ix+d)W<-mn	--------	16
LDW mem,nn	LDW	(HL+A),mn	F7:3F:n:m	(HL+A)W<-mn	--------	20
	LDW	(vw),mn	EB:w:v:3F:n:m	(vw)W<-mn	--------	16
	LDW	(w),mn	3F:w:n:m	(w)W<-mn	--------	14
PUSH	PUSH	qq	50+qq	SP<-SP-2,(SP)W<-qq	--------	8
POP	POP	qq	58+qq	qq<-(SP)W,SP<-SP+2	--------	10

Appendix A Table of Machine Instructions (2/11)

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
LDA rr,mem	LDA rr,ix+d	F4+ix:d:38+rr	rr<-ix+d	--------	10
	LDA rr,HL+A	F7:38+rr	rr<-HL+A	--------	14

3. Exchange, block transfer and search

Instruction	Mnemonic	Code	Function	SZIHXVNC	T	
	EX DE,HL	08	DE<-->HL	--------	2	
EX rr,rr	EX AF,AF'	09	AF<-->AF'	--------	2	
	EXX	0A	BC/DE/HL<-->BC'/DE'/HL'	--------	2	
	EX (gg),rr	E0+gg:50+rr	(gg)W<-->rr	--------	14	
	EX (ix+d),rr	F0+ix:d:50+rr	(ix+d)W<-->rr	--------	18	
EX mem,rr	EX (HL+A),rr	F3:50+rr	(HL+A)W<-->rr	--------	22	
	EX (mn),rr	E3:n:m:50+rr	(mn)W<-->rr	--------	18	
	EX (n),rr	E7:n:50+rr	(n)W<-->rr	--------	16	
LDI	LDI	FE:58	(DE)<-(HL) DE<-DE+1 HL<-HL+1 BC<-BC-1	---0-M0-	14	
LDIR	LDIR	FE:59	(DE)<-(HL) DE<-DE+1 HL<-HL+1 BC<-BC-1, Repeat until BC=0.	---0-00-	18/14	<- repe /no rep
LDD	LDD	FE:5A	(DE)<-(HL) DE<-DE-1 HL<-HL-1 BC<-BC-1	---0-M0-	14	
LDDR	LDDR	FE:5B	(DE)<-(HL) DE<-DE-1 HL<-HL-1 BC<-BC-1, Repeat until BC=0.	---0-00-	18/14	<- repe /no rep
CPI	CPI	FE:5C	A-(HL) HL<-HL+1 BC<-BC-1	*N-**M1-	14	
CPIR	CPIR	FE:5D	A-(HL) HL<-HL+1 BC<-BC-1 Repeat until A=(HL) or BC=0.	*N-**M1-	18/14	<- repe /no rep
CPD	CPD	FE:5E	A-(HL) HL<-HL-1 BC<-BC-1	*N-**M1-	14	
CPDR	CPDR	FE:5F	A-(HL) HL<-HL-1 BC<-BC-1 Repeat until A=(HL) or BC=0.	*N-**M1-	18/14	<- repe /no rep

(Note) Flag M: If BC=0 after execution, P/V flag is 0, and otherwise 1.
Flag N: If A=(HL), Z flag is 1, and otherwise 0.

Appendix A Table of Machine Instructions (3/11)

4. 8-bit arithmetic operation

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
ADD A,r	ADD A,g	F8+g:60	A <-A+g	**-**V0*	4
ADD A,n	ADD A,n	68:n	A <-A+n	**-**V0*	4
	ADD A,(gg)	E0+gg:60	A <-A+(gg)	**-**V0*	6
	ADD A,(ix+d)	F0+ix:d:60	A <-A+(ix+d)	**-**V0*	10
ADD A,mem	ADD A,(HL+A)	F3:60	A <-A+(HL+A)	**-**V0*	14
	ADD A,(mn)	E3:n:m:60	A <-A+(mn)	**-**V0*	10
	ADD A,(n)	60:n	A <-A+(n)	**-**V0*	8
ADD r,n	ADD g,n	F8+g:68:n	g <-g+n	**-**V0*	6
	ADD (gg),n	E8+gg:68:n	(gg)<-(gg)+n	**-**V0*	10
	ADD (ix+d),n	F4+ix:d:68:n	(ix+d)<-(ix+d)+n	**-**V0*	14
ADD mem,n	ADD (HL+A),n	F7:68:n	(HL+A)<-(HL+A)+n	**-**V0*	18
	ADD (vw),n	EB:w:v:68:n	(vw)<-(vw)+n	**-**V0*	14
	ADD (w),n	EF:w:68:n	(w)<-(w)+n	**-**V0*	12
ADC A,r	ADC A,g	F8+g:61	A <-A+g+CY	**-**V0*	4
ADC A,n	ADC A,n	69:n	A <-A+n+CY	**-**V0*	4
	ADC A,(gg)	E0+gg:61	A <-A+(gg)+CY	**-**V0*	6
	ADC A,(ix+d)	F0+ix:d:61	A <-A+(ix+d)+CY	**-**V0*	10
ADC A,mem	ADC A,(HL+A)	F3:61	A <-A+(HL+A)+CY	**-**V0*	14
	ADC A,(mn)	E3:n:m:61	A <-A+(mn)+CY	**-**V0*	10
	ADD A,(n)	61:n	A <-A+(n)+CY	**-**V0*	8
ADC r,n	ADC g,n	F8+g:69:n	g <-g+n+CY	**-**V0*	6
	ADC (gg),n	E8+gg:69:n	(gg)<-(gg)+n+CY	**-**V0*	10
	ADC (ix+d),n	F4+ix:d:69:n	(ix+d)<-(ix+d)+n+CY	**-**V0*	14
ADC mem,n	ADC (HL+A),n	F7:69:n	(HL+A)<-(HL+A)+n+CY	**-**V0*	18
	ADC (vw),n	EB:w:v:69:n	(vw)<-(vw)+n+CY	**-**V0*	14
	ADC (w),n	EF:w:69:n	(w)<-(w)+n+CY	**-**V0*	12
SUB A,r	SUB A,g	F8+g:62	A <-A-g	**-**V1*	4
SUB A,n	SUB A,n	6A:n	A <-A-n	**-**V1*	4
	SUB A,(gg)	E0+gg:62	A <-A-(gg)	**-**V1*	6
	SUB A,(ix+d)	F0+ix:d:62	A <-A-(ix+d)	**-**V1*	10
SUB A,mem	SUB A,(HL+A)	F3:62	A <-A-(HL+A)	**-**V1*	14
	SUB A,(mn)	E3:n:m:62	A <-A-(mn)	**-**V1*	10
	SUB A,(n)	62:n	A <-A-(n)	**-**V1*	8
SUB r,n	SUB g,n	F8+g:6A:n	g <-g-n	**-**V1*	6
	SUB (gg),n	E8+gg:6A:n	(gg)<-(gg)-n	**-**V1*	10
	SUB (ix+d),n	F4+ix:d:6A:n	(ix+d)<-(ix+d)-n	**-**V1*	14
SUB mem,n	SUB (HL+A),n	F7:6A:n	(HL+A)<-(HL+A)-n	**-**V1*	18
	SUB (vw),n	EB:w:v:6A:n	(vw)<-(vw)-n	**-**V1*	14
	SUB (w),n	EF:w:6A:n	(w)<-(w)-n	**-**V1*	12
SBC A,r	SBC A,g	F8+g:63	A <-A-g-CY	**-**V1*	4
SBC A,n	SBC A,n	6B:n	A <-A-n-CY	**-**V1*	4
	SBC A,(gg)	E0+gg:63	A <-A-(gg)-CY	**-**V1*	6
	SBC A,(ix+d)	F0+ix:d:63	A <-A-(ix+d)-CY	**-**V1*	10
SBC A,mem	SBC A,(HL+A)	F3:63	A <-A-(HL+A)-CY	**-**V1*	14
	SBC A,(mn)	E3:n:m:63	A <-A-(mn)-CY	**-**V1*	10
	SBC A,(n)	63:n	A <-A-(n)-CY	**-**V1*	8
SBC r,n	SBC g,n	F8+g:6B:n	g <-g-n-CY	**-**V1*	6
	SBC (gg),n	E8+gg:6B:n	(gg)<-(gg)-n-CY	**-**V1*	10
	SBC (ix+d),n	F4+ix:d:6B:n	(ix+d)<-(ix+d)-n-CY	**-**V1*	14
SBC mem,n	SBC (HL+A),n	F7:6B:n	(HL+A)<-(HL+A)-n-CY	**-**V1*	18
	SBC (vw),n	EB:w:v:6B:n	(vw)<-(vw)-n-CY	**-**V1*	14
	SBC (w),n	EF:w:6B:n	(w)<-(w)-n-CY	**-**V1*	12

Appendix A Table of Machine Instructions (4/11)

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
AND A,r	AND A,g	F8+g:64	A <-A AND g	**-10P00	4
AND A,n	AND A,n	6C:n	A <-A AND n	**-10P00	4
	AND A,(gg)	E0+gg:64	A <-A AND (gg)	**-10P00	6
	AND A,(ix+d)	F0+ix:d:64	A <-A AND (ix+d)	**-10P00	10
AND A,mem	AND A,(HL+A)	F3:64	A <-A AND (HL+A)	**-10P00	14
	AND A,(mn)	E3:n:m:64	A <-A AND (mn)	**-10P00	10
	AND A,(n)	64:n	A <-A AND (n)	**-10P00	8
AND r,n	AND g,n	F8+g:6C:n	g <-g AND n	**-10P00	6
	AND (gg),n	E8+gg:6C:n	(gg)<-(gg) AND n	**-10P00	10
	AND (ix+d),n	F4+ix:d:6C:n	(ix+d)<-(ix+d) AND n	**-10P00	14
AND mem,n	AND (HL+A),n	F7:6C:n	(HL+A)<-(HL+A) AND n	**-10P00	18
	AND (vw),n	EB:w:v:6C:n	(vw)<-(vw) AND n	**-10P00	14
	AND (w),n	EF:w:6C:n	(w)<-(w) AND n	**-10P00	12
OR A,r	OR A,g	F8+g:66	A <-A OR g	**-00P00	4
OR A,n	OR A,n	6E:n	A <-A OR n	**-00P00	4
	OR A,(gg)	E0+gg:66	A <-A OR (gg)	**-00P00	6
	OR A,(ix+d)	F0+ix:d:66	A <-A OR (ix+d)	**-00P00	10
OR A,mem	OR A,(HL+A)	F3:66	A <-A OR (HL+A)	**-00P00	14
	OR A,(mn)	E3:n:m:66	A <-A OR (mn)	**-00P00	10
	OR A,(n)	66:n	A <-A OR (n)	**-00P00	8
OR r,n	OR g,n	F8+g:6E:n	g <-g OR n	**-00P00	6
	OR (gg),n	E8+gg:6E:n	(gg)<-(gg) OR n	**-00P00	10
	OR (ix+d),n	F4+ix:d:6E:n	(ix+d)<-(ix+d) OR n	**-00P00	14
OR mem,n	OR (HL+A),n	F7:6E:n	(HL+A)<-(HL+A) OR n	**-00P00	18
	OR (vw),n	EB:w:v:6E:n	(vw)<-(vw) OR n	**-00P00	14
	OR (w),n	EF:w:6E:n	(w)<-(w) OR n	**-00P00	12
XOR A,r	XOR A,g	F8+g:65	A <-A XOR g	**-00P00	4
XOR A,n	XOR A,n	6D:n	A <-A XOR n	**-00P00	4
	XOR A,(gg)	E0+gg:65	A <-A XOR (gg)	**-00P00	6
	XOR A,(ix+d)	F0+ix:d:65	A <-A XOR (ix+d)	**-00P00	10
XOR A,mem	XOR A,(HL+A)	F3:65	A <-A XOR (HL+A)	**-00P00	14
	XOR A,(mn)	E3:n:m:65	A <-A XOR (m)	**-00P00	10
	XOR A,(n)	65:n	A <-A XOR (n)	**-00P00	8
XOR r,n	XOR g,n	F8+g:6D:n	g <-g XOR n	**-00P00	6
	XOR (gg),n	E8+gg:6D:n	(gg)<-(gg) XOR n	**-00P00	10
	XOR (ix+d),n	F4+ix:d:6D:n	(ix+d)<-(ix+d) XOR n	**-00P00	14
XOR mem,n	XOR (HL+A),n	F7:6D:n	(HL+A)<-(HL+A) XOR n	**-00P00	18
	XOR (vw),n	EB:w:v:6D:n	(vw)<-(vw) XOR n	**-00P00	14
	XOR (w),n	EF:w:6D:n	(w)<-(w) XOR n	**-00P00	12
CP A,r	CP A,g	F8+g:67	A-g	**-**V1*	4
CP A,n	CP A,n	6F:n	A-n	**-**V1*	4
	CP A,(gg)	E0+gg:67	A-(gg)	**-**V1*	6
	CP A,(ix+d)	F0+ix:d:67	A-(ix+d)	**-**V1*	10
CP A,mem	CP A,(HL+A)	F3:67	A-(HL+A)	**-**V1*	14
	CP A,(mn)	E3:n:m:67	A-(mn)	**-**V1*	10
	CP A,(n)	67:n	A-(n)	**-**V1*	8
CP r,n	CP g,n	F8+g:6F:n	g-n	**-**V1*	6
	CP (gg),n	E8+gg:6F:n	(gg)-n	**-**V1*	8
	CP (ix+d),n	F4+ix:d:6F:n	(ix+d)-n	**-**V1*	12
CP mem,n	CP (HL+A),n	F7:6F:n	(HL+A)-n	**-**V1*	16
	CP (vw),n	EB:w:v:6F:n	(vw)-n	**-**V1*	12
	CP (w),n	EF:w:6F:n	(w)-n	**-**V1*	10

Appendix A Table of Machine Instructions (5/11)

Instruction	Mnemonic	Code	Function	SZIHXVNC	T	
INC r	INC r	80+r	r <-r+1	**-**V0-	2	
	INC (gg)	E0+gg:87	(gg)<-(gg)+1	**-**V0-	8	
	INC (ix+d)	F0+ix:d:87	(ix+d)<-(ix+d)+1	**-**V0-	12	
INC mem	INC (HL+A)	F3:87	(HL+A)<-(HL+A)+1	**-**V0-	16	
	INC (mn)	E3:n:m:87	(mn)<-(mn)+1	**-**V0-	12	
	INC (n)	87:n	(n)<-(n)+1	**-**V0-	10	
DEC r	DEC r	88+r	r <-r-1	**-**V1-	2	
	DEC (gg)	E0+gg:8F	(gg)<-(gg)-1	**-**V1-	8	
	DEC (ix+d)	F0+ix:d:8F	(ix+d)<-(ix+d)-1	**-**V1-	12	
DEC mem	DEC (HL+A)	F3:8F	(HL+A)<-(HL+A)-1	**-**V1-	16	
	DEC (mn)	E3:n:m:8F	(mn)<-(mn)-1	**-**V1-	12	
	DEC (n)	8F:n	(n)<-(n)-1	**-**V1-	10	
INCX	INCX (n)	07:n	If X=1, (n) <- (n)+1	**-**V0-	6/10	<- X=0/1
DECX	DECX (n)	0F:n	If X=1, (n) <- (n)-1	**-**V1-	6/10	<- X=0/1

(Note) If X=0 in INCX and DECX instructions, all flags remain unchanged.

5. Special operation and CPU control

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
A register	DAA A	0B	Decimal adjust accumulator	**-**P-*	4
operation	CPL A	10	A <- A	---1--1-	2
	NEG A	11	A <-0-A	**-**V1*	2
LDAR	LDAR HL,PC+cd	17:d:c	HL<-PC+cd		8
CY flag	CCF	0E	CY<- CY	---X*-0*	2
operation	SCF	0D	CY<-1	---01-01	2
	RCF	0C	CY<-0	---00-00	2
NOP	NOP	00	No operation		2
HALT	HALT	01	Halt CPU		4
Interrupt	DI	02	Interrupt disable (IFF <- 0)	--0-----	2
operation	EI	03	Interrupt enable (IFF <- 1)	--1-----	2
	SWI	FF	Software interrupt (PC <- 0010H)	--0-----	20
MUL HL,r	MUL HL,g	F8+g:12	HL<-L×g		18
MUL HL,n	MUL HL,n	12:n	HL<-L×n		16
	MUL HL,(gg)	E0+gg:12	HL<-L×(gg)		18
	MUL HL,(ix+d)	F0+ix:d:12	HL<-L×(ix+d)		22
MUL HL,mem	MUL HL,(HL+A)	F3:12	HL<-L×(HL+A)		26
	MUL HL,(mn)	E3:n:m:12	HL<-L×(mn)		22
	MUL HL,(n)	E7:n:12	HL<-L×(n)		20
DIV HL,r	DIV HL,g	F8+g:13	L<-HL-g ,H<- Remainder		18
DIV HL,n	DIV HL,n	13:n	L<-HL-n ,H<- Remainder		16
	DIV HL,(gg)	E0+gg:13	L<-HL-(gg) ,H<- Remainder	-----V--	18
	DIV HL,(ix+d)	F0+ix:d:13	L<-HL-(ix+d),H<- Remainder	-----V--	22
DIV HL,mem	DIV HL,(HL+A)	F3:13	L<-HL-(HL+d),H<- Remainder	-----V--	26
	DIV HL,(mn)	E3:n:m:13	L<-HL-(mn) ,H<- Remainder	-----V--	22
	DIV HL,(n)	E7:n:13	L<-HL-(n) ,H<- Remainder	-----V--	20

Appendix A Table of Machine Instructions (6/11)

6. 16-bit arithmetic operation

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
ADD HL,rr	ADD HL,gg	F8+gg:70	HL<-HL+gg	---X*-0*	8
ADD HL,nn	ADD HL,mn	78:n:m	HL<-HL+mn	**-X*V0*	6
	ADD HL,(gg)	ED+gg:70	HL<-HL+(gg)W	**-X*V0*	8
	ADD HL,(ix+d)	FD+ix:d:70	HL<-HL+(ix+d)W	**-X*V0*	12
ADD HL,mem	ADD HL,(HL+A)	F3:70	HL<-HL+(HL+A)W	**-X*V0*	16
	ADD HL,(mn)	ED:n:m:70	HL<-HL+(mn)W	**-X*V0*	12
	ADD HL,(n)	70:n	HL<-HL+(n)W	**-X*V0*	10
ADC HL,rr	ADC HL,gg	F8+gg:71	HL<-HL+gg+CY	**-X*V0*	8
ADC HL,nn	ADC HL,mn	79:n:m	HL<-HL+mn+CY	**-X*V0*	6
	ADC HL,(gg)	ED+gg:71	HL<-HL+(gg)W+CY	**-X*V0*	8
	ADC HL,(ix+d)	FD+ix:d:71	HL<-HL+(ix+d)W+CY	**-X*V0*	12
ADC HL,mem	ADC HL,(HL+A)	F3:71	HL<-HL+(HL+A)W+CY	**-X*V0*	16
	ADC HL,(mn)	ED:n:m:71	HL<-HL+(mn)W+CY	**-X*V0*	12
	ADC HL,(n)	71:n	HL<-HL+(n)W+CY	**-X*V0*	10
SUB HL,rr	SUB HL,gg	F8+gg:72	HL<-HL-gg	**-X*V1*	8
SUB HL,nn	SUB HL,mn	7A:n:m	HL<-HL-mn	**-X*V1*	6
	SUB HL,(gg)	ED+gg:72	HL<-HL-(gg)W	**-X*V1*	8
	SUB HL,(ix+d)	FD+ix:d:72	HL<-HL-(ix+d)W	**-X*V1*	12
SUB HL,mem	SUB HL,(HL+A)	F3:72	HL<-HL-(HL+A)W	**-X*V1*	16
	SUB HL,(mn)	ED:n:m:72	HL<-HL-(mn)W	**-X*V1*	12
	SUB HL,(n)	72:n	HL<-HL-(n)W	**-X*V1*	10
SBC HL,rr	SBC HL,gg	F8+gg:73	HL<-HL-gg-CY	**-X*V1*	8
SBC HL,nn	SBC HL,mn	7B:n:m	HL<-HL-mn-CY	**-X*V1*	6
	SBC HL,(gg)	ED+gg:73	HL<-HL-(gg)W-CY	**-X*V1*	8
	SBC HL,(ix+d)	FD+ix:d:73	HL<-HL-(ix+d)W-CY	**-X*V1*	12
SBC HL,mem	SBC HL,(HL+A)	F3:73	HL<-HL-(HL+A)W-CY	**-X*V1*	16
	SBC HL,(mn)	ED:n:m:73	HL<-HL-(mn)W-CY	**-X*V1*	12
	SBC HL,(n)	73:n	HL<-HL-(n)W-CY	**-X*V1*	10
AND HL,rr	AND HL,gg	F8+gg:74	HL<-HL AND gg	**-10X00	8
AND HL,nn	AND HL,mn	7C:n:m	HL<-HL AND mn	**-10X00	6
	AND HL,(gg)	ED+gg:74	HL<-HL AND (gg)W	**-10X00	8
	AND HL,(ix+d)	FD+ix:d:74	HL<-HL AND (ix+d)W	**-10X00	12
AND HL,mem	AND HL,(HL+A)	F3:74	HL<-HL AND (HL+A)W	**-10X00	16
	AND HL,(mn)	ED:n:m:74	HL<-HL AND (mn)W	**-10X00	12
	AND HL,(n)	74:n	HL<-HL AND (n)W	**-10X00	10
OR HL,rr	OR HL,gg	F8+gg:76	HL<-HL OR gg	**-00X00	8
OR HL,nn	OR HL,mn	7E:n:m	HL<-HL OR mn	**-00X00	6
	OR HL,(gg)	ED+gg:76	HL<-HL OR (gg)W	**-00X00	8
	OR HL,(ix+d)	FD+ix:d:76	HL<-HL OR (ix+d)W	**-00X00	12
OR HL,mem	OR HL,(HL+A)	F3:76	HL<-HL OR (HL+A)W	**-00X00	16
	OR HL,(mn)	ED:n:m:76	HL<-HL OR (mn)W	**-00X00	12
	OR HL,(n)	76:n	HL<-HL OR (n)W	**-00X00	10

Appendix A Table of Machine Instructions (7/11)

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
XOR HL,rr	XOR HL,gg	F8+gg:75	HL<-HL XOR gg	**-00X00	8
XOR HL,nn	XOR HL,mn	7D:n:m	HL<-HL XOR mn	**-00X00	6
XOR HL,mem	XOR HL,(gg)	E0+gg:75	HL<-HL XOR (gg)W	**-00X00	8
	XOR HL,(ix+d)	F0+ix:d:75	HL<-HL XOR (ix+d)W	**-00X00	12
	XOR HL,(HL+A)	F3:75	HL<-HL XOR (HL+A)W	**-00X00	16
	XOR HL,(mn)	E3:n:m:75	HL<-HL XOR (mn)W	**-00X00	12
	XOR HL,(n)	75:n	HL<-HL XOR (n)W	**-00X00	10
CP HL,rr	CP HL,gg	F8+gg:77	HL-gg	**-X*V1*	8
CP HL,nn	CP HL,mn	7F:n:m	HL-mn	**-X*V1*	6
CP HL,mem	CP HL,(gg)	E0+gg:77	HL-(gg)W	**-X*V1*	8
	CP HL,(ix+d)	F0+ix:d:77	HL-(ix+d)W	**-X*V1*	12
	CP HL,(HL+A)	F3:77	HL-(HL+A)W	**-X*V1*	16
	CP HL,(mn)	E3:n:m:77	HL-(mn)W	**-X*V1*	12
	CP HL,(n)	77:n	HL-(n)W	**-X*V1*	10
ADD ix,rr	ADD ix,gg	F8+gg:14+ix	ix<-ix+gg	---X*-0*	8
ADD ix,nn	ADD ix,mn	14+ix:n:m	ix<-ix+mn	**-X*X0*	6
ADD ix,mem	ADD ix,(gg)	E0+gg:14+ix	ix<-ix+(gg)W	**-X*X0*	8
	ADD ix,(jx+d)	F0+jx:d:14+ix	ix<-ix+(jx+d)W	**-X*X0*	12
	ADD ix,(HL+A)	F3:14+ix	ix<-ix+(HL+A)W	**-X*X0*	16
	ADD ix,(mn)	E3:n:m:14+ix	ix<-ix+(mn)W	**-X*X0*	12
	ADD ix,(n)	E7:n:14+ix	ix<-ix+(n)W	**-X*X0*	10
INC rr	INC rr	90+rr	rr<-rr+1	----*---	4
INCW mem	INCW (gg)	E0+gg:97	(gg)W <-gg)W+1	**-X*V0-	12
	INCW (ix+d)	F0+ix:d:97	(ix+d)W <-(ix+d)W+1	**-X*V0-	16
	INCW (HL+A)	F3:97	(HL+A)W <-(HL+A)W+1	**-X*V0-	20
	INCW (mn)	E3:n:m:97	(mn)W <-(mn)W+1	**-X*V0-	16
	INCW (n)	97:n	(n)W <-(n)W+1	**-X*V0-	14
DEC rr	DEC rr	98+rr	rr<-rr-1	----*---	4
DECW mem	DECW (gg)	E0+gg:9F	(gg)W <-gg)W-1	**-X*V1-	12
	DECW (ix+d)	F0+ix:d:9F	(ix+d)W <-(ix+d)W-1	**-X*V1-	16
	DECW (HL+A)	F3:9F	(HL+A)W <-(HL+A)W-1	**-X*V1-	20
	DECW (mn)	E3:n:m:9F	(mn)W <-(mn)W-1	**-X*V1-	16
	DECW (n)	9F:n	(n)W <-(n)W-1	**-X*V1-	14

Appendix A Table of Machine Instructions (8/11)

7. Rotate, Shift

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
RLC r	RLCA	A0		---0X-0*	2
	RLC g	F8+g:A0		**-0XP0*	4
	RLC (gg)	E0+gg:A0		**-0XP0*	8
	RLC (ix+d)	F0+ix:d:A0	| CY | <-- | 7 <-- 0 |--	**-0XP0*	12
RLC mem	RLC (HL+A)	F3:A0		**-0XP0*	16
	RLC (mn)	E3:n:m:A0		**-0XP0*	12
	RLC (n)	E7:n:A0		**-0XP0*	10
RRC r	RRCA	A1		---0X-0*	2
	RRC g	F8+g:A1		**-0XP0*	4
	RRC (gg)	E0+gg:A1		**-0XP0*	8
	RRC (ix+d)	F0+ix:d:A1	--| 7 --> 0 | --> | CY |	**-0XP0*	12
RRC mem	RRC (HL+A)	F3:A1		**-0XP0*	16
	RRC (mn)	E3:n:m:A1		**-0XP0*	12
	RRC (n)	E7:n:A1		**-0XP0*	10
RL r	RLA	A2		---0X-0*	2
	RL g	F8+g:A2		**-0XP0*	4
	RL (gg)	E0+gg:A2		**-0XP0*	8
	RL (ix+d)	F0+ix:d:A2	--| CY |<--| 7 <-- 0 |--	**-0XP0*	12
RL mem	RL (HL+A)	F3:A2		**-0XP0*	16
	RL (mn)	E3:n:m:A2		**-0XP0*	12
	RL (n)	E7:n:A2		**-0XP0*	10
RR r	RRA	A3		---0X-0*	2
	RR g	F8+g:A3		**-0XP0*	4
	RR (gg)	E0+gg:A3		**-0XP0*	8
	RR (ix+d)	F0+ix:d:A3	--| 7 --> 0 |-->| CY |--	**-0XP0*	12
RR mem	RR (HL+A)	F3:A3		**-0XP0*	16
	RR (mn)	E3:n:m:A3		**-0XP0*	12
	RR (n)	E7:n:A3		**-0XP0*	10
SLA r	SLAA	A4		---0X-0*	2
	SLA g	F8+g:A4		**-0XP0*	4
	SLA (gg)	E0+gg:A4		**-0XP0*	8
	SLA (ix+d)	F0+ix:d:A4	| CY | <- | 7 <- 0 | <- 0	**-0XP0*	12
SLA mem	SLA (HL+A)	F3:A4		**-0XP0*	16
	SLA (mn)	E3:n:m:A4		**-0XP0*	12
	SLA (n)	E7:n:A4		**-0XP0*	10
SRA r	SRAA	A5		---0X-0*	2
	SRA g	F8+g:A5		**-0XP0*	4
	SRA (gg)	E0+gg:A5		**-0XP0*	8
	SRA (ix+d)	F0+ix:d:A5	--| 7 --> 0 | --> | CY |	**-0XP0*	12
SRA mem	SRA (HL+A)	F3:A5		**-0XP0*	16
	SRA (mn)	E3:n:m:A5		**-0XP0*	12
	SRA (n)	E7:n:A5		**-0XP0*	10
SLL r	SLLA	A6		---0X-0*	2
	RLL g	F8+g:A6		**-0XP0*	4
	SLL (gg)	E0+gg:A6		**-0XP0*	8
	SLL (ix+d)	F0+ix:d:A6	| CY | <- | 7 <- 0 | <- 0	**-0XP0*	12
SLL mem	SLL (HL+A)	F3:A6		**-0XP0*	16
	SLL (mn)	E3:n:m:A6		**-0XP0*	12
	SLL (n)	E7:n:A6		**-0XP0*	10

Appendix A Table of Machine Instructions (9/11)

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
SRL r	SRLA	A7		---0X-0*	2
	SRL g	F8+g:A7		**-0XP0*	4
	SRL (gg)	E0+gg:A7		**-0XP0*	8
	SRL (ix+d)	F0+ix:d:A7	0 -> | 7 --> 0 | ->| CY |	**-0XP0*	12
SRL mem	SRL (HL+A)	F3:A7		**-0XP0*	16
	SRL (mn)	E3:n:m:A7		**-0XP0*	12
	SRL (n)	E7:n:A7		**-0XP0*	10
	RLD (gg)	E0+gg:10	Areg memory	**-00P0-	12
	RLD (ix+d)	F0+ix:d:10		**-00P0-	16
RLD mem	RLD (HL+A)	F3:10	|7 4|3 0| |7 4|3 0|	**-00P0-	20
	RLD (mn)	E3:n:m:10		**-00P0-	16
	RLD (n)	E7:n:10		**-00P0-	14
	RRD (gg)	E0+gg:11	Areg memory	**-00P0-	12
	RRD (ix+d)	F0+ix:d:11		**-00P0-	16
RRD mem	RRD (HL+A)	F3:11	|7 4|3 0| |7 4|3 0|	**-00P0-	20
	RRD (mn)	E3:n:m:11		**-00P0-	16
	RRD (n)	E7:n:11		**-00P0-	14

8. Bit manipulation

Instruction	Mnemonic	Code	Function	SZIHXVNC	T
BIT b,r	BIT b,g	F8+g:A8+b	Z <- ~ g.b	X*-1XX0-	4
	BIT b,(gg)	E0+gg:A8+b	Z <- ~ (gg).b	X*-1XX0-	6
	BIT b,(ix+d)	F0+ix:d:A8+b	Z <- ~ (ix+d).b	X*-1XX0-	10
BIT b,mem	BIT b,(HL+A)	F3+A8+b	Z <- ~ (HL+A).b	X*-1XX0-	14
	BIT b,(mn)	E3:n:m:A8+b	Z <- ~ (mn).b	X*-1XX0-	10
	BIT b,(n)	A8+b:n	Z <- ~ (n).b	X*-1XX0-	8
SET b,r	SET b,g	F8+g:B8+b	g.b <- 1	--------	4
	SET b,(gg)	E0+gg:B8+b	(gg).b <- 1	--------	10
	SET b,(ix+d)	F0+ix:d:B8+b	(ix+d).b <- 1	--------	14
SET b,mem	SET b,(HL+A)	F3+B8+b	(HL+A).b <- 1	--------	18
	SET b,(mn)	E3:n:m:B8+b	(mn).b <- 1	--------	14
	SET b,(n)	B8+b:n	(n).b <- 1	--------	12
RES b,r	RES b,g	F8+g:B0+b	g.b <- 0	--------	4
	RES b,(gg)	E0+gg:B0+b	(gg).b <- 0	--------	10
	RES b,(ix+d)	F0+ix:d:B0+b	(ix+d).b <- 0	--------	14
RES b,mem	RES b,(HL+A)	F3+B0+b	(HL+A).b <- 0	--------	18
	RES b,(mn)	E3:n:m:B0+b	(mn).b <- 0	--------	14
	RES b,(n)	B0+b:n	(n).b <- 0	--------	12
TEST b,r	TEST b,g	F8+g:18+b	Z <- ~ g.b :g.b <-1	X*-1XX0-	8
	TEST b,(gg)	E0+gg:18+b	Z <- ~ (gg).b :(gg).b <-1	X*-1XX0-	12
	TEST b,(ix+d)	F0+ix:d:18+b	Z <- ~ (ix+d).b:(ix+d).b <-1	X*-1XX0-	16
TEST b,mem	TEST b,(HL+A)	F3+18+b	Z <- ~ (HL+A).b:(HL+A).b <-1	X*-1XX0-	20
	TEST b,(mn)	E3:n:m:18+b	Z <- ~ (mn).b :(mn).b <-1	X*-1XX0-	16
	TEST b,(n)	E7:n:18+b	Z <- ~ (n).b :(n).b <-1	X*-1XX0-	14

Appendix A Table of Machine Instructions (10/11)

9. Jump, call and return

Instruction	Mnemonic	Code	Function	SZIHXVNC	T	
	JP cc,gg	E8+gg:C0+cc	If cc, PC <- gg	--------	6/8	<- F/T
	JP cc,ix+d	F4+ix:d:C0+cc	If cc, PC <- ix+d	--------	10/12	<- F/T
JP cc,mem	JP cc,HL+A	F7:C0+cc	If cc, PC <- HL+A	--------	14/16	<- F/T
	JP cc,mn	EB:n:m:C0+cc	If cc, PC <- mn	--------	10/12	<- F/T
	JR cc,PC+d	C0+cc:d	If cc, PC <- PC+d	--------	4/8	<- F/T
	JP mn	1A:n:m	PC <- mn	--------	8	
	JRL PC+cd	1B:d:c	PC <- PC+cd	--------	10	
	CALL cc,gg	E8+gg:D0+cc	If cc, PUSH PC,PC <- gg	--------	6/14	<- F/T
	CALL cc,ix+d	F4+ix:d:D0+cc	If cc, PUSH PC,PC <- ix+d	--------	10/18	<- F/T
CALL cc,mem	CALL cc,HL+A	F7:D0+cc	If cc, PUSH PC,PC <- HL+A	--------	14/22	<- F/T
	CALL cc,mn	EB:n:m:D0+cc	If cc, PUSH PC,PC <- mn	--------	10/18	<- F/T
	CALL mn	1C:n:m	PUSH PC,PC <- mn	--------	14	
	CALL PC+cd	1D:d:c	PUSH PC,PC <- PC+cd	--------	16	
DJNZ	DJNZ PC+d	18:d	B <-B-1, if B< >0, PC<-PC+d	--------	10	
	DJNZ BC,PC+d	19:d	BC<-BC-1, if BC< >0, PC<-PC+d	--------	10	
	RET	1E	POP PC	--------	10	
RET	RET cc	FE:D0+cc	If cc, POP PC	--------	6/14	<- F/T
	RETI	1F	POP AF, POP PC	--------	14	

Condition codes ⌜cc⌟

Symbol	Meaning	Flag value	Code
F	Always false	-	0
(None)	Always true	-	8
Z	Zero	Z = 1	6
NZ	Not Zero	Z = 0	E
C	Carry	C = 1	7
NC	No carry	C = 0	F
PL or P	Plus	S = 0	D
MI or M	Minus	S = 1	5
NE	Not equal	Z = 0	E
EQ	Equal	Z = 1	6
OV	Overflow	P/V = 1	4
NOV	No overflow	P/V = 0	C
PE	Parity is even	P/V = 1	4
PO	Parity is odd	P/V = 0	C
GE	Greater than or equal (signed)	(S XOR P/V) = 0	9
LT	Less than (signed)	(S XOR P/V) = 1	1
GT	Greater than (signed)	[Z OR (S XOR P/V)] = 0	A
LE	Less than or equal (signed)	[Z OR (S XOR P/V)] = 1	2
UGE	Unsigned greater than or equal	C = 0	F
ULT	Unsigned less than	C = 1	7
UGT	Unsigned greater than	(C OR Z) = 0	B
ULE	Unsigned less than or equal	(C OR Z) = 1	3

Appendix A Table of Machine Instructions (11/11)

10. Meaning of symbols

Category	Symbol	Meaning
Operand	r, g	Register : A,B,C,D,E,H,L
	rr, gg	Register pair : BC,DE,HL,IX,IY,SP
	gg	Register pair : BC,DE,HL,IX,IY,AF
	ix, jx	Register pair : IX,IY,SP
	n	8-bit constant : 00H - FFH
	mn	16-bit constant : 0000H - FFFFH
	d	8-bit displacement : -128 - +127
	cd	16-bit displacement : -32768 - +32767
	b	Bit number : 0 - 7
	cc	Condition code
	(gg)	Register indirect:(BC),(DE),(HL),(IX),(IY),(SP)
	(ix+d),(jx+d)	Index : (IX+d),(IY+d),(SP+d)
	(HL+A)	Register index : A=-128 - +127
	(mn),(vw)	Extend : (0000H) - (FFFFH)
	(n),(w)	Direct : (FF00H) - (FFFFH)
	(x)W	2-byte memory data
Flag	S	Signed flag. Set MSB according to operation result.
	Z	Zero flag. Set to 1 if operation result is zero.
	I	Interrupt enable flag "IFF"
	H	Half carry flag
	X	Expansion carry flag
	V	Parity/overflow flag (P/V)
	N	Addition/subtraction flag
	C	Carry flag
Flag Status	0	Reset to 0 according to operation
	1	Reset to 1 according to operation
	-	No change
	*	Subject to operation result
	X	Undefined
	P	Treated as parity flag
	V	Treated as overflow flag
Time	T	Number of states indicating executing time (1 state = 200 ns @10MHz)

(Note) The value of displacement object code in the relative addressing mode is calculated with respect to the address "The start address plus 2".

r,g	Code
A	6
B	0
C	1
D	2
E	3
H	4
L	5

rr,gg	Code
BC	0
DE	1
HL	2
IX	4
IY	5
SP	6

qq	Code
BC	0
DE	1
HL	2
IX	4
IY	5
AF	6

ix,jx	Code
IX	0
IY	1
SP	2

b	Code
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Appendix B Table of Machine Instruction Codes (1/4)

1-byte opcode instruction

H\L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	
0	NOP	HALT	DI	EI				INCX (n)	EX DE,HL	EX AF,AF'	EXX	DAA A	RCF	SCF	CCF	DE (
1	CPL A	NEG A	MUL HL,n	DIV HL,n	ADD IX,nn	ADD IY,nn	ADD SP,nn	LDAR HL,dd	DJNZ d	DJNZ BC,d	JP nn	JRL dd	CALL nn	CALR dd	RET	RE
2	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,A	LD A,(n)	LD B,A	LD C,A	LD D,A	LD E,A	LD H,A	LD L,A	LD A,A	L (n
3	LD B,n	LD C,n	LD D,n	LD E,n	LD H,n	LD L,n	LD A,n	LD (n)n	LD BC,nn	LD DE,nn	LD HL,nn		LD IX,nn	LD IY,nn	LD SP,nn	L (n
4	LD HL,BC	LD HL,DE	LD HL,HL		LD HL,IX	LD HL,IY	LD HL,SP	LD HL(n)	LD BC,HL	LD DE,HL	LD HL,HL		LD IX,HL	LD IY,HL	LD SP,HL	L (n
5	PUSH BC	PUSH DE	PUSH HL		PUSH IX	PUSH IY	PUSH AF		POP BC	POP DE	POP HL		POP IX	POP IY	POP AF	
6	ADD A,(n)	ADC A,(n)	SUB A,(n)	SBC A,(n)	AND A,(n)	XOR A,(n)	OR A,(n)	CP A,(n)	ADD A,n	ADC A,n	SUB A,n	SBC A,n	AND A,n	XOR A,n	OR A,n	C A
7	ADD HL(n)	ADC HL(n)	SUB HL(n)	SBC HL(n)	AND HL(n)	XOR HL(n)	OR HL(n)	CP HL(n)	ADD HL,nn	ADC HL,nn	SUB HL,nn	SBC HL,nn	AND HL,nn	XOR HL,nn	OR HL,nn	C HL
8	INC B	INC C	INC D	INC E	INC H	INC L	INC A	INC (n)	DEC B	DEC C	DEC D	DEC E	DEC H	DEC L	DEC A	D
9	INC BC	INC DE	INC HL		INC IX	INC IY	INC SP	INCW (n)	DEC BC	DEC DE	DEC HL		DEC IX	DEC IY	DEC SP	D
A	RLC A	RRC A	RL A	RR A	SLA A	SRA A	SLL A	SRL A	BIT 0,(n)	BIT 1,(n)	BIT 2,(n)	BIT 3,(n)	BIT 4,(n)	BIT 5,(n)	BIT 6,(n)	B 7,
B	RES 0,(n)	RES 1,(n)	RES 2,(n)	RES 3,(n)	RES 4,(n)	RES 5,(n)	RES 6,(n)	RES 7,(n)	SET 0,(n)	SET 1,(n)	SET 2,(n)	SET 3,(n)	SET 4,(n)	SET 5,(n)	SET 6,(n)	S 7,
C	JR F,d	JR LT,d	JR LE,d	JR ULE,d	JR PE,d	JR M,d	JR Z,d	JR C,d	JR d	JR GE,d	JR GT,d	JR UGT,d	JR PO,d	JR P,d	JR NZ,d	J NC
D																
E	src (BC)	src (DE)	src (HL)	src (nn)	src (IX)	src (IY)	src (SP)	src (n)	dst (BC)	dst (DE)	dst (HL)	dst (nn)	dst (IX)	dst (IY)	dst (SP)	d (
F	src (IX+d)	src (IY+d)	src (SP+d)	src (HL+A)	dst (IX+d)	dst (IY+d)	dst (SP+d)	dst (HL+A)	reg B/BC	reg C/DE	reg D/HL	reg E	reg H/IX	reg L/IY	reg A/SP	S

(Note) Codes located in "EOH to FEH" are part of 2-byte opcode instructions.

ndix B Table of Machine Instruction Codes (2/4)

byte opcode F8H - FEH: reg g/gg

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		MUL HL,g	DIV HL,g	ADD IX,gg	ADD IY,gg	ADD SP,gg		TSET 0,g	TSET 1,g	TSET 2,g	TSET 3,g	TSET 4,g	TSET 5,g	TSET 6,g	TSET 7,g
LD B,g	LD C,g	LD D,g	LD E,g	LD H,g	LD L,g	LD A,g		LD BC,gg	LD DE,gg	LD HL,gg		LD IX,gg	LD IY,gg	LD SP,gg	
								LDI	LDIR	LDD	LDDR	CPI	CPIR	CPD (NOTE)	CPDR
ADD A,g	ADC A,g	SUB A,g	SBC A,g	AND A,g	XOR A,g	OR A,g	CP A,g	ADD g,n	ADC g,n	SUB g,n	SBC g,n	AND g,n	XOR g,n	OR g,n	CP g,n
ADD HL,gg	ADC HL,gg	SUB HL,gg	SBC HL,gg	AND HL,gg	XOR HL,gg	OR HL,gg	CP HL,gg								
RLC g	RRC g	RL g	RR g	SLA g	SRA g	SLL g	SRL g	BIT 0,g	BIT 1,g	BIT 2,g	BIT 3,g	BIT 4,g	BIT 5,g	BIT 6,g	BIT 7,g
RES 0,g	RES 1,g	RES 2,g	RES 3,g	RES 4,g	RES 5,g	RES 6,g	RES 7,g	SET 0,g	SET 1,g	SET 2,g	SET 3,g	SET 4,g	SET 5,g	SET 6,g	SET 7,g
RET F	RET LT	RET LE	RET ULE	RET PE	RET M	RET Z	RET C	RET	RET GE	RET GT	RET UGT	RET PO	RET P	RET NZ (NOTE)	RET NC

te) The first byte of block transfer instructions (LDI, LDIR, LDD and LDDR), block search instructions (CPI, CPIR, CPD and CPDR) and a conditional return instruction (RET cc) must be "FEH".

Appendix B Table of Machine Instruction Codes (3/4)

1st byte code E0H - E7H/F0H - F3H: src (x)

H\L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	
0																
1	RLD (x)	RRD (x).	MUL HL(x)	DIV HL(x)	ADD IX(x)	ADD IY(x)	ADD SP(x)		TSET 0,(x)	TSET 1,(x)	TEST 2,(x)	TSET 3,(x)	TSET 4,(x)	TSET 5,(x)	TSET 6,(x)	TS 7,
2									LD B,(x)	LD C,(x)	LD D,(x)	LD E,(x)	LD H,(x)	LD L,(x)	LD A,(x)	
3																
4									LD BC(x)	LD DE(x)	LD HL(x)		LD IX(x)	LD IY(x)	LD SP(x)	
5	EX (x)BC	EX (x)DE	EX (x)HL		EX (x)IX	EX (x)IY	EX (x)SP									
6	ADD A,(x)	ADC A,(x)	SUB A,(x)	SBC A,(x)	AND A,(x)	XOR A,(x)	OR A,(x)	CP A,(x)								
7	ADD HL(x)	ADC HL(x)	SUB HL(x)	SBC HL(x)	AND HL(x)	XOR HL(x)	OR HL(x)	CP HL(x)								
8								INC (x)								D (
9								INCW (x)								DE (
A	RLC (x)	RRC (x)	RL (x)	RR (x)	SLA (x)	SRA (x)	SLL (x)	SRL (x)	BIT 0,(x)	BIT 1,(x)	BIT 2,(x)	BIT 3,(x)	BIT 4,(x)	BIT 5,(x)	BIT 6,(x)	B 7,
B	RES 0,(x)	RES 1,(x)	RES 2,(x)	RES 3,(x)	RES 4,(x)	RES 5,(x)	RES 6,(x)	RES 7,(x)	SET 0,(x)	SET 1,(x)	SET 2,(x)	SET 3,(x)	SET 4,(x)	SET 5,(x)	SET 6,(x)	S 7,
C																
D																
E																
F																

-endix B Table of Machine Instruction Codes (4/4)

byte code E8H - EFA/F4H - F7H: dst (×)

L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	LD (×),B	LD (×),C	LD (×),D	LD (×),E	LD (×),H	LD (×),L	LD (×),A									
								LD (×)n	LD BC,×	LD DE,×	LD HL,×		LD IX,×	LD IY,×	LD SP,×	LDW (×)nn
	LD (×)BC	LD (×)DE	LD (×)HL		LD (×)IX	LD (×)IY	LD (×)SP									
									ADD (×),n	ADC (×),n	SUB (×),n	SBC (×),n	AND (×),n	XOR (×),n	OR (×),n	CP (×),n
:	JP F,×	JP LT,×	JP LE,×	JP ULE,×	JP PE,×	JP M,×	JP Z,×	JP C,×	JP ×	JP GE,×	JP GT,×	JP UGT,×	JP PO,×	JP P,×	JP NZ,×	JP NC,×
)	CALL F,×	CALL LT,×	CALL LE,×	CALL ULE,×	CALL PE,×	CALL M,×	CALL Z,×	CALL C,×	CALL ×	CALL GE,×	CALL GT,×	CALL UGT,×	CALL PO,×	CALL P,×	CALL NZ,×	CALL NC,×
?																

Appendix C

Table of Special Function Registers

The special function registers include the I/O ports, peripheral control regisers and bank registers (BX and BY) allocated to the 48-byte addresses from FFC0H OFFEFH.

(1) I/O port

(2) I/O port control

(3) Stepping motor control port control

(4) Watchdog timer control

(5) Timer/event counter control

(6) Serial channel control

(7) A/D converter control

(8) Interrupt control

(9) Bank register

Format of table

Symbol	Name	Address	7 : 6 : : 1 : 0	
			: : : :	-> bit Symbol
			: : : :	-> Read/Write
			: : : :	-> Initial value after reset
			: : : :	-> Remarks

.) I/O Port

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
P0	Port 0	OFFC0H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Input mode							
P1	Port 1	OFFC1H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode							
P2	Port 2	OFFC4H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode							
P3	Port 3	OFFC6H	P37	P36	P35	P34	P33	P32	P31	P30
			R	R/W	R/W	R	R/W	R/W	R	R
			Input	1	1	Input	1	1	Input	Input
P4	Port 4	OFFC8H					P43	P42	P41	P40
							R/W			
							0	0	0	0
P5	Port 5	OFFCAH	0		P55	P54	P53	P52	P51	P50
					R					
					Input only					
					Shared with analog input pin (AN0 - AN5)					
P6	Port 6	OFFCCH	SA63	SA62	SA61	SA60	P63	P62	P61	P60
			W				R/W			
			Undefined				Input mode			
			Stepping motor control Port 0 shifter alternate reg.				Shared with stepping motor control port 0 (M0)			
P7	Port 7	OFFCDH	SA73	SA72	SA71	SA70	P73	P72	P71	P70
			W				R/W			
			Undefined				Input mode			
			Stepping motor control Port 1 shifter alternate reg.				Shared with stepping motor control port 1 (M1)			
P8	Port 8	OFFD0H					P83	P82	P81	P80
							R/W	R	R	R
							0	Input mode		

(Note) Read/Write
R/W : Either read or write is possible.
R : Only read is possible.
W : Only write is possible.

(2) I/O Port control

Symbol	Name	Address	7	6	5	4	3	2	1	0
P01CR (IRFL)	Port 0/1 Control Reg.	OFFC2H		IRF0	IRFT0	IRFT1		EXT	P1C	P0C
				R				W	W	W
				0	0	0		0	0	0
				Interrupt Request Flag 1. Interrupt being requested				P1-P2 control 0: I/O Port: 1: Address bus	P1 control 0: In 1: Out	P0 control 0: In 1: Out
P2CR	Port 2 Control Reg.	OFFC5H	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
							W			
			0	0	0	0	0	0	0	0
			0: In 1: Out (I/O selected bit by bit)							
P3CR	Port 3 Control Reg.	OFFC7H	WAITC1	WAITC0	RDE	ODE	TXDC1	TXDC0	RXDC1	RXDC0
			R/W		R/W	R/W	R/W		R/W	
			0	0	0	0	0	0	0	0
			Wait control 00: 2 state wait 01: normal wait 10: non wait 11: reserved		RD control 0: RD for only external access 1: Always RD	P33 control 0: CMOS Open drain	P33 00: Out 01: Out 10: TxD 11: TxD	P32 Out TxD Out RTS/ SCLK:	P31 00: In 01: In 10: RxD 11: Not used	P30 In RxD In
P4CR	Port 4 Control Reg.	OFFC9H					P43C	P42C	P41C	P40C
								W		
							0	0	0	0
							0: Out (Port) 1: Address output			
P67CR	Port 6/7 Control Reg.	OFFCEH	P73C	P72C	P71C	P70C	P63C	P62C	P61C	P60C
					W				W	
			0	0	0	0	0	0	0	0
			0: In		1: Out		0: In		1: Out	
P8CR	Port 8 Control Reg.	OFFD1H					P830C	ZCE2	ZCE1	EDGE
							W	W	W	W
							0	0	0	0
							P83 control 0: P83 1: TO3/TO4	INT2/TI5 control 1: ZCD enable	INT1/TI4 control 1: ZCD enable	INT0 control 0: level 1: edge

Symbol in ( ) denotes another name.

(3) Stepping motor control port control

Symbol	Name	Address	7	6	5	4	3	2	1	0
			SM7M1	SM7M0	P70C1	P70C0	SM6M1	SM6M0	P60C1	P60C0
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
SMMOD	Stepping Motor Mode Reg.	OFFCBH	0: 3-phase 1: 4-phase	0:1-step: /2-step: excita-tion 1:1-2 step excita-tion	00: IN/OUT 01: IN/OUT,TO3 10: IN/M1 (Timer 2, Timer 3) 11: " (Timer 4)		0: 3-phase 1: 4-phase	0:1-step: /2-step: excita-tion 1:1-2 step excita-tion	00: IN/OUT 01: IN/OUT,TO1 1X: IN/M0 (Timer 0, Timer 1)	
						CCW7				CCW6
						R/W				R/W
	Stepping Motor Control Reg.					0				0
SMCR		OFFCFH				0:Normal rotation 1: Re-verse rotation				0:Normal rotation 1: Re-verse rotation

Also refer to P67CR, P6 and P7 registers.

(4) Watchdog timer control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
			WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	EXF	DRVE
			R/W	R/W		R/W	R/W		R	R/W
			1	0	0	0	0	0	Un-defined:	0
WDMOD	Watchdog Timer Mode Reg.	OFFD2H	1: WDT Enable:	Detecting time 00: 2^{14}/fc 01: 2^{16}/fc 10: 2^{18}/fc 11: 2^{20}/fc		Warming-up time 0:2^{14}/fc 1:2^{16}/fc	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Invert each time EXF: instruction is executed:	1: to drive pin in STOP mode.
WDCR	Watchdog Timer Control Reg.	OFFD3H	-							
			W							
			-							
			B1H: WDT Disable code				4EH: WDT Clear code			

(5) Timer/event counter control

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG0	8-bit Timer Register 0	OFFD4H	-							
			W							
			Undefined							
TREG1	8-bit Timer Register 1	OFFD5H	-							
			W							
			Undefined							
TREG2	8-bit Timer Register 2	OFFD6H	-							
			W							
			Undefined							
TREG3	8-bit Timer Register 3	OFFD7H	-							
			W							
			Undefined							
TCLK	8-bit Timer Source Clock Control Reg.	OFFD8H	T3CLK1	T3CLK0	T2CLK1	T2CLK0	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			8-bit		00: -		8-bit		00: -	
			00: T02TRG		01: ϕT1		00: T00TRG		01: ϕT1	
			01: ϕT1		10: ϕT16		01: ϕT1		10: ϕT16	
			10: ϕT16		11: ϕT256		10: ϕT16		11: ϕT256	
			11: ϕT256		(8-bit mode only)		11: ϕT256		(8-bit mode only	
TFFCR	8-bit Timer Flip-Flop Control Reg.	OFFD9H	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			-		0	0	-		0	0
			00: Clear TFF3		1: TFF3	0:	00: Clear TFF1		1: TFF1	0:
			01: Set TFF3		Invert	Invert	01: Set TFF1		Invert	Invert
			10: Invert TFF3		Enable	by 8-bit	10: Invert TFF1		Enable	by 8-bi
			11: Don't care			timer 2	11: Don't care			timer C
TMOD	8-bit Timer Mode Reg.	OFFDAH	T23M1	T23M0	TWM21	PWM20	T10M1	T10M0	PWM01	PWM0C
			R/W		R/W		R/W		R/W	
			0	0	0	0	0	0	0	0
			00: 8-bit Timer		PWM Frequency		00: 8-bit Timer		PWM Frequency	
			01: 16-bit Timer		00: -		01: 16-bit Timer		00: -	
			10: 8-bit PPG		01: 2^6-1		10: 8-bit PPG		01: 2^6-1	
			11: 8-bit PWM		10: 2^7-1		11: 8-bit PWM		10: 2^7-1	
					11: 2^8-1				11: 2^8-1	

Symbol	Name	Address	7	6	5	4	3	2	1	0
			BRATE1	BRATE0	PRRUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
	8-bit Timer/		R/W				R/W			
	Serial		0	0	0	0	0	0	0	0
TRUN	Channel	OFFDBH	00: 300/150 baud: Prescaler & Timer Run/Stop Control							
	Baud Rate		01:1200/600 :							
	Control Reg.		10:4800/2400 : 0: Stop & Clear							
			11:19200/9600 : 1: Run (Count up)							
	16-bit Timer		-							
CAP1L	/Event	OFFDCH	R							
	Counter		Undefined							
	Capture		-							
CAP1H	Register 1	OFFDDH	R							
			Undefined							
	16-bit Timer		-							
CAP2L	/Event	OFFDEH	R							
	Counter		Undefined							
	Capture		-							
CAP2H	Register 2	OFFDFH	R							
			Undefined							
			-							
TREG4L	16-bit Timer	OFFE0H	W							
	/Event		Undefined							
	Counter		-							
TREG4H	Register 4	OFFE1H	W							
			Undefined							
			-							
TREG5L	16-bit Timer	OFFE2H	W							
	/Event		Undefined							
	Counter		-							
TREG5H	Register 5	OFFE3H	W							
			Undefined							

Symbol	Name	Address	7	6	5	4	3	2	1	0
					CAP1IN	CAPM1	CAPM0	CLE	T4CLK1	T4CLK0
					W	R/W		R/W	R/W	
					-	0	0	0	0	0
T4MOD	16-bit Timer /Event Counter Mode Reg.	OFFE4H			:0: Software: 1: Capture: 1: Don't care	Capture timing 00: Disable 01: TI4 ↑ TI5 ↑ 10: TI4 ↑ TI4 ↓ 11: TFF1 ↑ TFF1 ↓		1: UC16 Clear Enable:	Timer 4 source clock 00: TI4 01: øT1 10: øT16 11: -	
					CAP2TE	CAP1TE	EQ5TE	EQ4TE	TFF4C1	TFF4C0
					R/W				W	
T4FFCR	16-bit Timer Flip-Flop 4 Control Reg.	OFFE5H			0	0	0	0	-	
					TFF 4 inversion trigger 0: Disable trigger 1: Enable trigger				00: Clear TFF4 01: Set TFF4 10: Invert TFF4 11: Don't care	

(6) Serial channel control

Symbol	Name	Address	7	6	5	4	3	2	1	0
			TB8	Fixed at "0"	RXE	WU	SM1	SM0	SC1	SC0
							R/W			
SCMOD	Serial Channel Mode Reg.	OFFE9H	Undefined Transmission Bit-8 data in 9-bit UART	0	1: Receive Enable	1: Wake up Enable	00: I/O interface 01: UART 7-bit 10: UART 8-bit 11: UART 9-bit	0	00: T02TRG 01: BR 10: o1 11: BR 1/2	U A R T
			RB8	EVEN	PE	OERR	PERR	FERR		CTSE
			R	R/W		R(Cleared to "0" by reading)				R/W
SCCR	Serial Channel Control Register	OFFEAH	Undefined Receiving Bit-8 data	0 Parity 0: Odd 1: Even	1: Parity Enable	0	0 1: Error Overrun	0 Parity Flaming		0 1: CTS Enable
			RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
SCBUF	Serial Channel Buffer Register	OFFEBH	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
			R (Receiving)/W (Transmission) Undefined							

Also refer to P3CR, TRUN register. (Note) BR: Baud Rate Generator

7) A/D Converter control

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
				EOCF	ADBF	ADCS	ADS	ADCH2	ADCH1	ADCH0
	A/D			R		R/W	R/W		R/W	
ADMOD	Converter Mode Reg.	OFFEFH		0	0	0	0	0	0	0
				1: END	1: Busy	0: 95s 1:190s	1: Start	Analog Input Channel Select		
	A/D Result					-				
ADREG	Register	OFFEEH				R				
						-				

8) Interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
			*IET2	IET3	IET4	IE1	IET5	IE2	IERX	IETX
INTEL	Interrupt	OFFE6H				R/W				
	Enable		0	0	0	0	0	0	0	0
	Mask Reg.			1: Enable			0: Disable			
			-	DE0	DET0	DET1	ADIS	IE0	IET0	IET1
INTEH	.....	OFFE7H		R/W				R/W		
(DMA			0	0	0	0	0	0	0	0
EL)	Micro DMA		1: Enable		0: Disable		1: Enable		0: Disable	
	Enable		DET2	DET3	DET4	DE1	DET5	DE2	DERX	DETX
DMAEH	Register	OFFE8H				R/W				
			0	0	0	0	0	0	0	0
				1: Enable			0: Disable			
				IRF0	IRFT0	IRFT1		EXT	P1CR	P0CR
				R				W	W	
				0	0	0		0	0	0
IRFL		OFFC2H		Interrupt Request Flag				P1,P2	P1	P0
(P01—	Interrupt							Controls:	Control:	Control
CR)	Request			1: Interrupt being re-				0: I/O	0: In	0: In
	Flag &			quested				port	1: Out	1: Out
	IRF							1: Ad-		
	Clear							dress:		
								bus		
			IRFT2	IRFT3	IRFT4	IRF1	IRFT5	IRF2	IRFRX	IRFTX
			R (Only IRF Clear code can be used to write.)							
IRFH		OFFC3H	0	0	0	0		0	0	0
			1: Interrupt being requested (IRF is cleared to "0" by writing IRF Clear code.)							

Symbol in ( ) denotes another name. * Share with IEAD (If ADIS=1, use as INTAD Mask Reg.)

9) Bank register

Symbol	Name	Address	MSB 7	6	5	4	3	2	1	LSB 0
BX	Bank	OFFECH					BX3	BX2	BX1	BX0
	Register X						R/W			
							0	0	0	0
BY	Bank	OFFEDH					BY3	BY2	BY1	BY0
	Register Y						R/W			
							0	0	0	0