TOSHIBA

INTEGRATED CIRCUIT TECHNICAL DATA

MICROCOMPUTER

LSI DATA BOOK

July 1984

TOSHIBA CORPORATION

Thank you very much for your interest in Toshiba microcomputer LSI's. Since a microprocessor LSI family manufactured first in Japan by Toshiba was put on the market in 1973, Toshiba has devoted efforts in strengthening the microcomputer LSI devices. Toshiba has various highly efficient and low power consumption type LSIs for microcomputers to more and more diversifying application fields, and supplying numerous kinds of new products to customers. This manual covers the technical description of microcomputer LSIs supplied by Toshiba, including TLCS-42 Series (NMOS and CMOS) and TLCS-47 Series (NMOS and CMOS) 4-bit single chip microcontrollers, TLCS-48 Series (NMOS and CMOS) 8-bit single chip microcontrollers, TLCS-Z80 Family (CMOS) and TLCS-85 Family (NMOS) 8-bit microprocessors and microperipherals.

Toshiba has provided following product literature for microcomputers and development tools:

1. PRODUCTS LIST : MICROCOMPUTER LSI AND TOOL
2. PRODUCTS BRIEF: MICROCOMPUTER LSI AND TOOL
3. MICROCOMPUTER LSI DATA BOOK [MCU, MPU & MPR Device Spec.] (This manual)
4. TLCS-42 SYSTEM MANUAL [MCU spec., ASM42 & RTE42]
5. TLCS-47 SYSTEM MANUAL [MCU spec., ASM47, PL47 & App.Guide]
6. TLCS-48 SYSTEM MANUAL [MCU spec., ASM48 & App.Guide]
7. TLCS-Z80 SYSTEM MANUAL [MPU + MPR spec. & RMAC80]
8. TLCS-85 SYSTEM MANUAL [MPU + MPR spec. & ASM85]
9. TDS800A/RTE OPERATION MANUAL [TDS800A, RTE47, RTE48 & RTE80]
10. RTE OPERATION MANUAL [RTE42, RTE47 & RTE80] (Planned)

Note) TLCS-85 ≡ TLCS-85A

TABLE OF CONTENTS

MICROCOMPUTER PRODUCTS LIST

o 4bit Single Chip Microcntroller

TLCS-42 Series (NMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Pin	Remarks
TMP4240P	512 x 8	32 x 4	11	16	
*TMP4250N			23	28	
*TMP4260P	1,024 x 8		11	16	Freq. Divider
*TMP4270N			23	28	Freq. Divider

Note) * : under development

TLCS-42 Series (CMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Pin	Remarks
*TMP42C00Y	External 1,024 x 8	32 x 4	23	64	Evaluator
TMP42C40P	512 x 8		11	16	Freq. Divider
*TMP42C50N			23	28	Freq. Divider
*TMP42C60P	1,024 x 8		11	16	Freq. Divider
*TMP42C70N			23	28	Freq. Divider

Note) * : under development

TLCS-47 Series (NMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Pin	Remarks
TMP4700AC	External 4,096 x 8	256 x 4	35	80	Evaluator
TMP4720P TMP4720N	2,048 x 8	128 x 4		42	LED drivers
TMP4740P TMP4740N	4,096 x 8				LED drivers
*TMP4746N		256 x 4	57	64	
TMP4799C TMP4799E	External 4,096 x 8		35	42	Piggy Back

Note) * : under development

TLCS-47 Series (CMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Pin	Inst.Cycle Time (us)	Remarks
TMP47C20P TMP47C20N	2,048 x 8	128 x 4	35	42	4	
TMP47C21P TMP47C21N					4	VFT drivers
TMP47C22F		192 x 4	27	67	4	LCD drivers
TMP47C40P TMP47C40N	4,096 x 8	256 x 4	35	42	4	
TMP47C41P TMP47C41N					4	VFT drivers
TMP47C46N			57	64	4	
TMP47C47N					4	VFT drivers
*TMP47C200N	2,048 x 8	128 x 4	35	42	2	
*TMP47C210N					2	VFT drivers
*TMP47C220F		192 x 4	27	67	2	LCD drivers
TMP47C400N	4,096 x 8	256 x 4	35	42	2	
TMP47C410N					2	VFT drivers
TMP47C420F			27	67	2	LCD drivers
TMP47C432N			35	42	2	PWM

Note) * : under development

o 8bit Single Chip Microcontroller

TLCS-48 Series (NMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Operating Temp. (C)	Pin	Compatible Component
TMP8022P	2,048 x 8		28			8022
TMP8048P (TMP8035P)	1,024 x 8	64 x 8		0 to 70		8048 (8035L)
*TMP8048N *(TMP8035N)	(External					- (-)
TMP8048PI (TMP8035PI)	4,096x8)			-40 to 85		i8048(i8035L)
TMP8049P (TMP8039P)			27		40	8049 (8039)
*TMP8049N (TMP8039N)	2,048 x 8		(19)	0 to 70		- (-)
TMP8049P-6 (TMP8039P-6)	(External	128x 8				8049-6
	4,096x8)					(8039-6)
*TMP8049N-6 (TMP8039N-6)						- (-)
TMP8049PI-6(TMP8039PI-6)				-40 to 85		i8049(i8039)
TMP8043P	I/O Expander			0 to 70	24	8243
TMP8043PI				-40 to 85		i8243

Note) * : under development

TLCS-48 Series (CMOS Version)

Type Number	ROM (bit)	RAM (bit)	I/O (bit)	Operat. Temp.(C)	Pin	Remarks
	1,024 x 8					
TMP80C48AP (TMP80C35AP)	(External	64 x 8		-40to 85		
	4,096x8)					
TMP80C49AP (TMP80C39AP)	2,048 x 8		27	0 to 70	40	
TMP80C49AP-6	(External	128x 8	(19)	-40to 85		
(TMP80C39AP-6)	4,096x8)					
TMP80C50AP (TMP80C40AP)	2,048 x 8			0 to 70		
TMP80C50AP-6	(External	256x 8				
(TMP80C39P-6)	4,096x8)			-40to 85		
TMP80C43P	I/O Expander				24	

o 8bit Microprocessor

TLCS-Z80 Family (CMOS Version)

Type Number	Function	Inst. Cycle Time (us)	Supply Current(mA)	Pin	Remarks
TMPZ84C00P	Z80 MPU (4MHz)	1.0	15	40	MPU
T6497	Clock Generator/Controller		2	16	CLK
*TMPZ84C10P	DMA Controllor		TBD	40	DMA
TMPZ84C20P	Parallel I/O Controller		2		PIO
TMPZ84C30P	Counter/Timer Circuit		3	28	CTC
*TMPZ84C40P/41P/42P	Serial I/O Cntroller		TBD	40	SIO

Note) * : under development

TLCS-85 Family (NMOS Version)

Type Number	Function	Inst. Cycle Time (us)	Supply Current(mA)	Pin	Compatible Component
TMP80C85AP *(TMP8085AN)	8bit MPU	1.3	170	40	8085AP (-)
TMP8155/56P	2KB RAM with I/O and Timer		180		8155/56
TMP8251AP	Prog. Communication Interface		100	28	8251A
TMP8253P-5	Prog. Interval Timer		140	24	8253-5
TMP8255AP-5	Prog. Peripheral Interface		120	40	8255A-5
TMP8259AP	Prog. Interrupt Controller		85	28	8259A
TMP8279P-5	Prog. Keyboard/Display Interface		120		8279-5
TMP8355P*(TMP8355N)	16KB ROM with I/O Ports		180	40	8355 (-)
TMP8755AC	16KB EPROM with I/O Ports				8755A

Note) * : under development ,() : Shrunk DIP

PART 1

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-42

LSI DEVICES

July. 1 9 8 4

PREFACE

This part describes the detail functions and specifications of LSI devices of a 4-bit microcontroller TLCS-42 series. The TLCS-42 series is designed to replace small scale logic composed of several gates with a single chip microcontroller and to reduce the cost of application system. The TLCS-42 enhances versatility of the application systems without additional expense. Consequently, the TLCS-42 series has superior performance per cost in low end microcontroller application field.

CONTENTS

1. General Description

The TLCS-42 series is a 4-bit microcontroller series developed aiming at cost performance so that relatively small scale logic may be integrated in the microcomputers, and is available in NMOS version (TLCS- 42N) and CMOS version (TLCS-42C) provided with software compatibility.

1.1 Features

- o ROM 512 x 8 bits (0.5K byte)
  1024 x 8 bits (1K byte)
- o RAM 32 x 4 bits
- o Basic instructions 42 (NMOS)
  44 (CMOS)
- o Instruction execution time 2.5 us (at 2MHz clock)
- o Subroutine nesting 1 level
- o Table search function by programs
- o Built-in frequency divider for timer
  (NMOS version is available in ROM 1K byte products only.)
- o With hold function
  (CMOS version only)
- o Input/Output port 11 lines (16-pin products)
  23 lines (28-pin products)
- o Package 16-pin plastic DIL
  28-pin plastic shrink type DIL

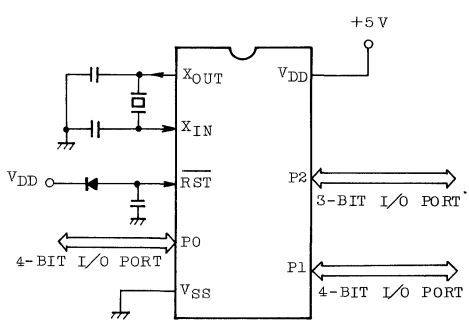


Fig. 1.1 Application example of TMP4240P (NMOS 0.5K ROM 16 PIN)

1.2 Series configuration

Version			TLCS-42N (NMOS)				TLCS-42C (CMOS)				
Item	Unit	Product No.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
ROM Capacity	Byte		512	512	1024	1024	512	512	1024	1024	(***) 1024
RAM Capacity	Word		32	32	32	32	32	32	32	32	32
Instruction execution time	us		2.5 to 25								
No. of basic instruction	Kind		42				44				
Subroutine	Level		1								
Divider for timer	Stage		11 (2048 divided frequencyes)								
Hold function			-				Available				
Port	I/O	Bit	11	23	11	23	11	19	11	19	23
	Output	Bit	-	-	-	-	-	4	-	4	-
	Total	Bit	11	23	11	23	11	23	11	23	23
Large current output	Bit		4	4	4	4	-	-	-	-	-
Clock oscillation circuit			Built-in								
Power supply	V		+5V								
Package (*)			DIP16	DIP28	DIP16	DIP28	DIP16	DIP28	DIP16	DIP28	PGA64
Process			Nch Si-Gate E/D MOS LSI				CMOS Si-Gate MOS LSI				
Piggy board			BM4210								
(**) RTE42			BM4220 or BM1020 + BM4221								

(*): Package DIP16 16-pin plastic DIL DIP28 28-pin plastic shrink DIL
PGA64 64-pin ceramic Pin Grid Array package
(**): Development tool
(***): external

(Note): (a) TMP4240P (b) TMP4250N (c) TMP4260P (d) TMP4270N
(e) TMP42C40P (f) TMP42C50N (g) TMP42C60P (h) TMP42C70N
(i) TMP42C00Y

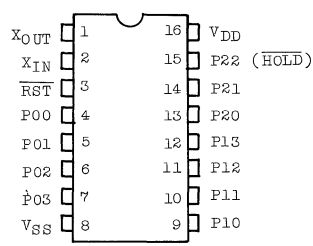
2. Pin Name and Pin Description

2.1 Pin connections

The TLCS-42 series is available in the products packed into a 16-pin plastic mold DIL package shown in Fig. 2.1 (1) and in the products packed into a shrink type 28-pin plastic mold DIL package shown in Fig. 2.1 (2). The $\overline{HOLD}$ pin which serves as port P22 is available for the products only of CMOS version, but not for those of NMOS version.

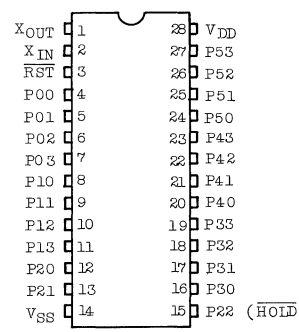
(1) For a 16-pin product, the ports are all I/O ports and 11 lines in total. In NMOS version, the pull-up resistor can be specified to all the ports in the unit of bit as mask option. In CMOS version, all the ports are programmable I/O ports, and input/output can be specified, in 4-bit unit for P0 and P1, in 3-bit unit for P2.

(2) For 28-pin products, the function of 28-pin products are equal to those of 16-pin products. The ports of P3, P4 and P5 are increased, resulting in 23 I/O lines.
For NMOS version, ports of P3 to P5 are of the same configuration as ports of P0 to P2. For CMOS version, Ports of P3 to P4 can specify I/O by mask option. Port P5 serves as CMOS output.



($\overline{HOLD}$) pin function is available for the products only of CMOS version.
TMP4240P, TMP4260P
TMP42C40P, TMP42C60P

Fig. 2.1(1) Pin connection of 16-pin Products



($\overline{HOLD}$) pin function is available for the products only of CMOS version.
TMP4250N, TMP4270N
TMP42C50N, TMP42C70N

Fig. 2.1(2) Pin Connection of 28-pin Products

2.2 Pin function

(1) NMOS Version (TLCS-42N)

Pin Symbols	Pin Name	I/O	Functional Description
P00 - P03	Port P0	I/O	4-bit I/O port (with pull-up resistor or open-drain)
P10 - P13	Port P1	I/O	4-bit I/O port (with pull-up resistor or open-drain)
P20 - P22	Port P2	I/O	3-bit I/O port (with pull-up resistor or open-drain)
P30 - P33	Port P3	I/O	4-bit I/O port (with pull-up resistor or open-drain) — 28-pin products only
P40 - P43	Port P4	I/O	4-bit I/O port (with pull-up resistor or open-drain) — 28-pin products only
P50 - P53	Port P5	I/O	4-bit I/O port (with pull-up resistor or open-drain) — 28-pin products only
XIN	Clock input	Input	Ceramic oscillator connecting pin. If external clock input is provided for XIN, XOUT should be open.
XOUT	" output	Output	
$\overline{RST}$	Reset input		When "L" is set over 3 instruction cycle, the CPU is initialized.
VDD	Power	-	Power supply +5V
VSS	GND	-	Power supply 0V

(1) CMOS Version (TLCS-42C)

Pin Symbols	Pin Name	I/O	Functional Description
P00 - P03	Port P0	I/O	4-bit I/O port (Programmable I/O of 4-bit unit)
P10 - P13	Port P1	I/O	4-bit I/O port (Programmable I/O of 4-bit unit)
P20 - P22	Port P2	I/O	3-bit I/O port (Programmable I/O of bit unit) P22 is served as hold release pin ($\overline{HOLD}$) too.
P30 - P33	Port P3	I/O	4-bit I/O port (Note 1)
P40 - P43	Port P4	I/O	4-bit I/O port (Note 1)
P50 - P53	Port P5	Output	4-bit output port (CMOS output) 28-pin products only
XIN	Clock input	Input	Ceramic oscillator connecting pin. If external clock input is provided for XIN, XOUT should be open.
XOUT	" output	Output	
$\overline{RST}$	Reset input		When "L" is set over 3 instruction cycle, the CPU is initialized.
VDD	Power	-	Power supply +5V
VSS	GND	-	Power supply 0V

(Note 1) One of three types of I/O circuit, CMOS output, CMOS input and NMOS input/output, can be chosen by mask option.

3. Operational Description

3.1 Configuration

The block diagram in Fig. 3.1 (1) shows the configuration of TLCS-42 series.

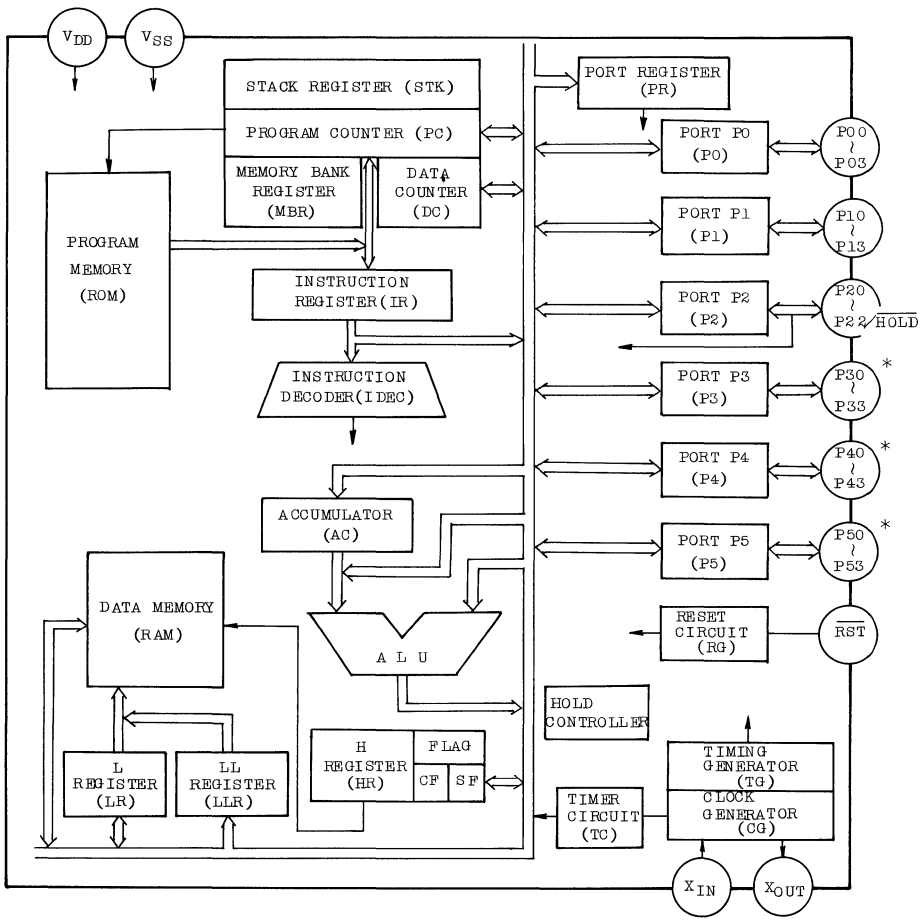


Fig. 3.1 (1) TLCS-42 Block Diagram

Notes:
- o The ports with * mark are 28-pin package products only.
- o ROM 0.5K byte products of NMOS version are not provided with timer circuit.
- o Port register, input function of $\overline{\text{HOLD}}$ pin and hold controlling circuit are available for the products only of CMOS version.

3.2 Description of each block

(1) Program counter (PC)

This is a 10-bit binary counter which specifies the address of ROM. Generally the program counter gains one increment at every instruction fetch. However, when executing the branch and subroutine instructions, the values specified by these instructions and operations are set as shown in Fig. 3.2 (1).

Execution Instruction	Condition	Program counter PC9 PC8 PC7 PC6 PC5 PC4 PC3 PC2 PC1 PC0
Normal instruction	—	Increment
(BSSa)	SF=0	Increment
	SF=1. PC0 to PC5 are not all "1".	Hold / The value specified by BSS
	SF=1. PC0 to PC5 are all "1".	Increment / The value specified by BSS
(BSSa) immediately after (LD MBR, #k)	SF=0	Increment
	SF=1	Contents of MBR / The value specified by BSS
(CALLSa)	—	0 / The value specified by CALLS / 0

Fig. 3.2 (1) PC Operation

1] Branch instruction (BSSa)

At the execution of (BSSa) instruction, after the PC has gained one increment, the value specified by the instruction is set in the lower 6 bits of PC0 to PC5 only when the branch condition is set. Therefore, if the (BSSa) instruction is stored in the last address of the page, it becomes a branch instruction in the next page.

Branching to all the program areas can be made by executing the (BSSa) instruction immediately after (LD MBR,#k) instruction under the condition of SF=1.

At this time, the contents of MBR is set in the higher 4 bits of PC6 to PC9.

2] Subroutine call instruction (CALLSa)

At the execution of (CALLSa) instruction, the value specified by the instruction is set in PC1 to PC4 and other values than it are set in "0". Therefore, the start address of the subroutine must be in the address 2n ($1 < n < 15$).

(2) Stack register (STK)

This is a 10-bit register in which the return address from the subroutine is stored.

When (CALLSa) instruction is executed by address PCi, the address PCi + 1, which has gained one increment, is stored in the STK, and the contents are returned to the PC by (RET) instruction.

The subroutine can be used by one level only.

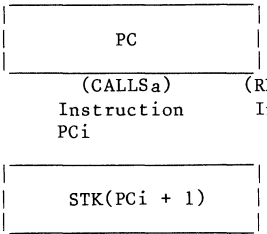


Fig. 3.2 (2) STK Operation

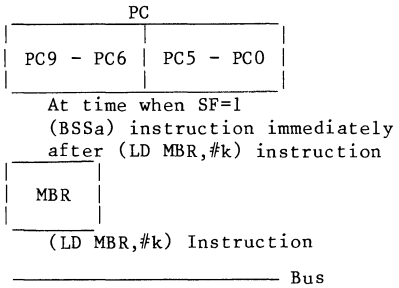


Fig. 3.2 (3) MBR Operation

(3) Memory bank register (MBR)

This is a 4-bit register, and when the branch is implemented to all program areas, the contents of this MBR are set in the higher 4 bits of the PC.

The values specified by (LD MBR,#k) instruction are set in the MBR. Only when the (BSSa) instruction has been executed immediately after this instruction under the condition of SF=1, the contents of MBR are set in the higher 4 bits of the PC.

When another instruction subsequent to (LD MBR,#k) has been executed, and then the (BSSa) instruction is executed, the contents of MBR are not set in the higher 4 bits of the PC.

(4) Data counter (DC)

The data counter, a 4-bit register, specifies the ROM address when the ROM contents are used as fixed data.

When the ROM data referring instruction (LDL A, @DC) or (LDH A, @DC) instruction is executed, the lower 4 bits of ROM address form the contents of DC and all the higher 6 bits are forced to be at "1".

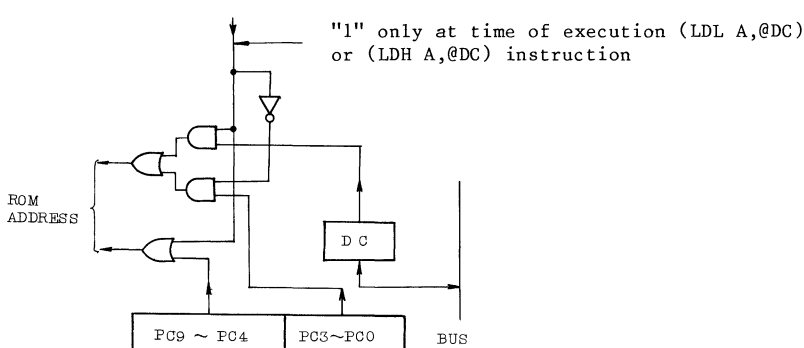


Fig. 3.2 (4) DC Operation

Therefore, the last 16 addresses of ROM are specified by the DC, and will be able to be used as the fixed data.
Since the DC is provided with the functions of data transfer to or from the accumulator, increment, and decrement.

(5) Program memory (ROM)
The program memory is a mask ROM, where the programs and the fixed data are stored.
The ROM capacity is available in two types, 0.5k (512 x 8 bits) capacity and 1k (1024 x 8 bits) capacity.

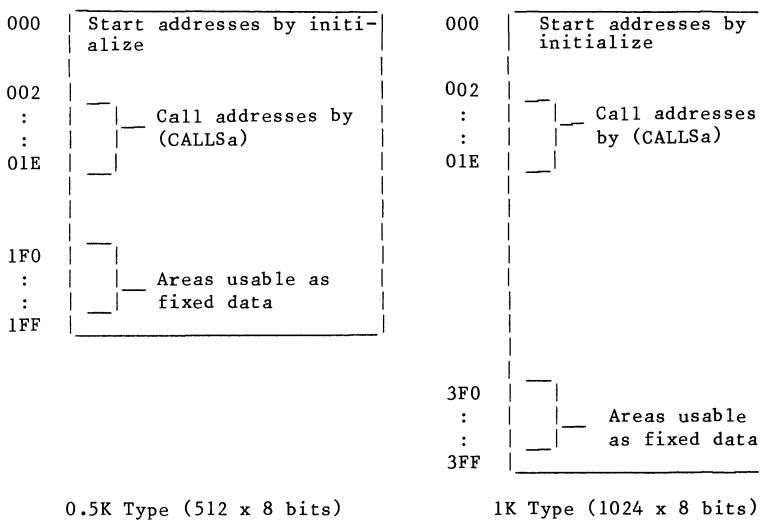


Fig. 3.2 (5) Configuration of ROM

(6) H register (HR), L register (LR), LL register (LLR) and Data memory (RAM)
The RAM has the capacity of 128 bits (32 x 4 bits) with the page structure based on 16 words per page.
The HR is a 1-bit register, and is used to specify the pages of RAM. The LR is a 4-bit register, and is used to specify the addresses in pages of RAM. The LLR is a 3-bit register, and is used as an address pointer of RAM instead of the HR or LR only at time of the execution of (LD A,x) or (ST A,x) instruction.
The addresses 0 to 7 in page 0 are the RAM areas where the LLR can directly use by an instruction.

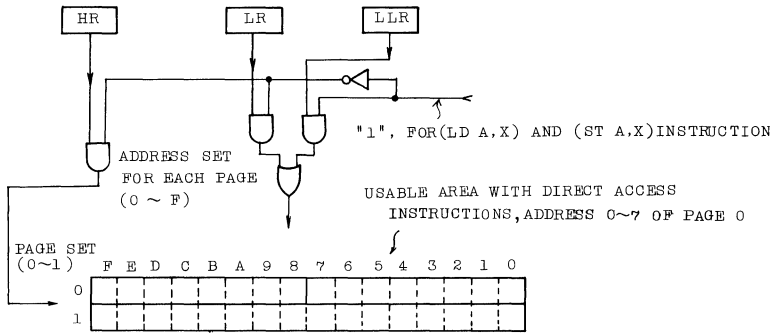


Fig. 3.2 (6) Configuration of RAM

(7) Arithmetic and logical unit (ALU), accumulator (AC)

The ALU is used for arithmetic and logical operation for 4-bit binary data.

In the operation instruction, the ALU performs the specified operation and outputs the 4-bit result and carry (C). Further it is the central circuit for the transfer, input/output, logical, and bit manipulation instructions, and in the input/output and logical instructions it outputs zero detection signal (Z) detecting the data to be transferred to the accumulator or memory.

The accumulator AC is a 4-bit register central to operation, logical transfer and input/output.

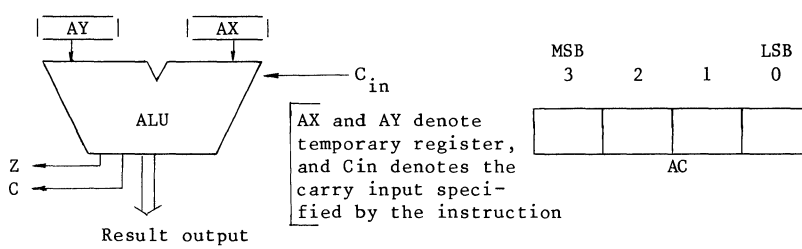


Fig. 3.2 (7) ALU and AC

Carry (C) and zero detection signal (Z)

Output C from the ALU indicates the carry output from the most significant position in the addition operation. However, the subtraction is processed as the addition of the 2's complement, so that the output C in the subtraction operation indicates the "non-borrow" from the most significant position. Z is set to "1" when the data transferred to accumulator or memory by the input instruction is "0" or when the data transferred to accumulator by the logical instruction is "0".

(8) Flags (CF, SF)

Each of carry flag (CF) and status flag (SF) is a 1-bit flag set/reset or held according to the conditions specified by an instruction.

Carry flag (CF)

This flag is used not only for an input to the ALU, but to hold the carry or the non-borrow (the carry in the binary addition of the 2's complement) in the operation result of (ADDC A, @HL) and (SUBRC A, @HL) instruction.

The carry flag is set by (SET CF) instruction and reset by (CLR CF) instruction.

Status flag (SF)

The status flag is referred to as branch condition in a branch instruction. When SF=1, the memory location is branched; normally the branch instruction can be regarded as "unconditional jump instruction". However, the instruction becomes a "conditional jump instruction" if it is executed immediately after the instruction to set/reset the status flag according to the condition specified by an instruction.

The status flag is initialized to "1" at initialization.

(9) Port (P0, P1, P2, P3, P4, P5)

NMOS version (TLCS-42N) and CMOS version (TLCS-42C) are different in the circuit configuration of port.

P2 port only is a 3-bit port, and the others are 4-bit ports.

The 16-pin package products contain 11 I/O lines of P0 to P2, and the 28-pin package products contain 23 I/O lines of P0 to P5.

1] NMOS version

All ports are of the same circuit configuration as shown in fig. 3.2 (8).

The data of accumulator are output by (OUT A,%P) instruction, and those of RAM by (OUT @HL, %P) instruction, respectively.

The data input from ports are transferred to accumulator by (IN %P,A) instruction, and are transferred to RAM by (IN %P,@HL).

At time of initialization, the output latch is initialized to "1", then the port is forced to the "1" state.

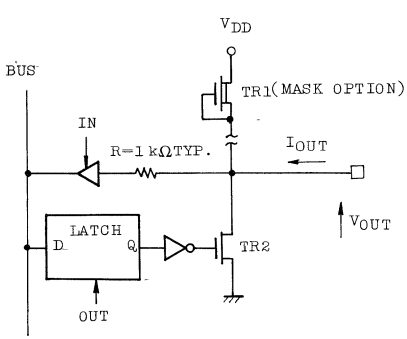


Fig. 3.2 (8) NMOS Version Circuit Configuration of Port

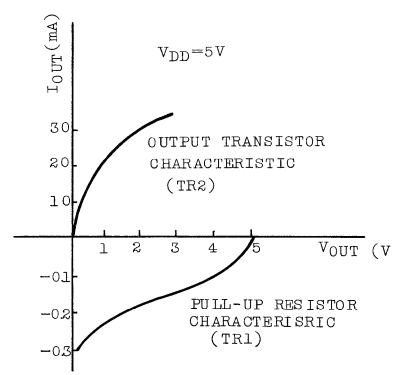


Fig. 3.2 (9) NMOS Version Characteristic of Port

Option:
In Fig 3.2 (8), TR1 is a pull-up resistor by mask option, and all the ports can be specified in the bit unit. Fig. 3.2 (9) shows the standard characteristics of a port.
The output transistor TR2 is large in current driving performance and is low in impedance, so that if it is used as input port, it is necessary to use after having turned off the output transistor TR2 to force it to the "1" level. In this case, if the pull-up resistor TR1 is optionally provided, the input impedance will become the characteristics of TR1.

2] CMOS version
Ports P0 to P2 are programmable I/O ports which input or output can be specified by a program.
Ports P3 to P4 are optionally hardware-selectable I/O ports, and are so designed that CMOS output, NMOS input/output or CMOS input with pull-up resistor may be selected. P5 is a CMOS output port.

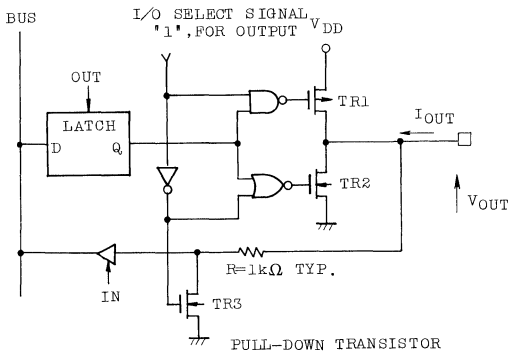
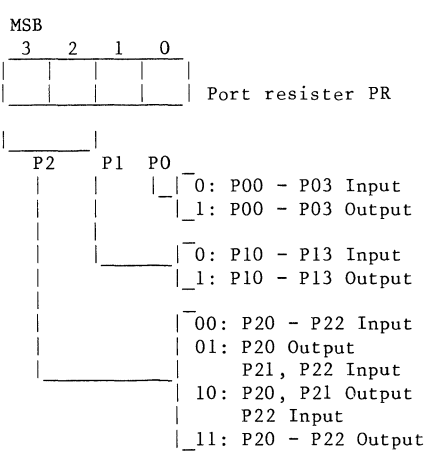


Fig. 3.2 (10) CMOS Verson Port



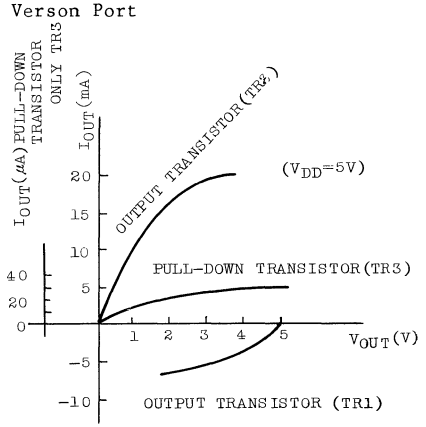


Fig. 3.2 (11) CMOS Version Characteristics of Port P0 to P2

Port P0 to P2

The circuit configuration is shown in Fig. 3.2 (10), and the standard characteristics are shown in Fig. 3.2 (11).

Input/output can be controlled by the data set to the port register (PR) by (MOV A,P) instruction.

Independently of the port register, the data of accumulator or RAM are transferred to the output latch by (OUT A,%P) or (OUT @HL,%P) instruction, and the port data are transferred to accumulator or RAM by (IN %P,A) or (IN %P,@HL) input instruction.

Since at time of initialization all the bits of the port register PR is initialized to "0", ports P0 to P2, are forced to the input mode. A port interfaced, as output, with an external circuit is forced to the input mode. When the output transistors, TR1 and TR2 are turned OFF, the level of port becomes unstable, which is likely to lead to the flow of excessive current. Therefore, CMOS version is so designed that the port impedance may not become high by the pull-down transistor TR3 being turned ON. The output latch is initialized to "1" at time of initialization.

Mask Option Code	P30 - P33	P40 - P43
(A)	CMOS Output	CMOS Output
(B)	NMOS I/O	CMOS Output
(C)	NMOS I/O	NMOS I/O
(D)	CMOS Input	CMOS Input

Fig. 3.2 (12) CMOS Version Mask Option Table for P3, P4

Hold terminal ($\overline{\text{HOLD}}$)

The port of P22 is provided with the ($\overline{\text{HOLD}}$ pin) function as a hold release signal input in addition to the function of a general pin. When the CMOS version is at a hold state, if the $\overline{\text{HOLD}}$ terminal is forced to a "H" level, the hold state is released.

Ports, P3 to P5

As for the ports of P3 and P4, one of four kinds of options of (A), (B), (C) and (D) as shown in Fig. 3.2 (12) can be chosen. There are three types of port configuration of CMOS output, NMOS input/output, and CMOS input. P5 is fixed to CMOS output.

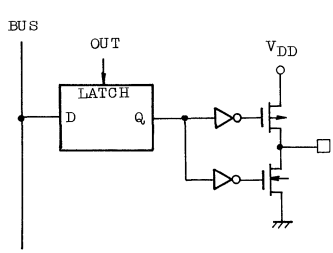


Fig. 3.2 (13) CMOS Version Circuit Configuration of Port P3 to P5 CMOS Output

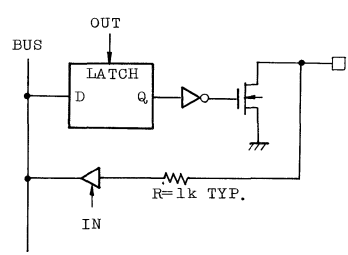


Fig. 3.2 (14) CMOS Version Circuit Configuration of Port P3 and P4 NMOS I/O

CMOS Output

The circuit configuration is shown in Fig. 3.2 (13).
The data of accumulator or RAM are transferred to the output latch by the output instruction of (OUT A,%P) or (OUT @HL,%P), and are output as they are.
The characteristics of output transistor TR1 and TR2 are the same as those in Fig. 3.2.(11).
The output latch is initialized to "1" at time of initialization.

NMOS Input/Output

The circuit configuration is shown in Fig. 3.2 (14).
The configuration is the same as that of NMOS version except the mask option of pull-up resistor.
The shift of the output level can be made by the external pull-up resistor. Being used as the input port, this is an input of high impedance. The data are transferred to the output latch by the output instruction. When the output transistor is OFF ("1" output state), the port data are transferred to the inside by the input instruction. The characteristics of output transistor TR2 are the same as those of TR2 in Fig. 3.2 (11).
At time of initializatoin, the output latch is initialized to "1".

CMOS Input

The circuit configuration is shown in Fig. 3.2 (15).
When ports, P0 to P2, are in the input mode, these ports serve as the CMOS input with pull-up resistor.
The port data are transferred to the inside by the input instruction.
The standard chracteristics are shown in Fig. 3.2 (16).

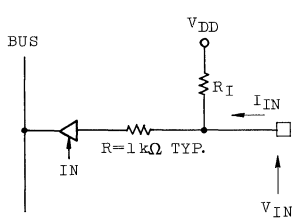


Fig. 3.2 (15) CMOS Version Circuit Configuration of CMOS input with Pull-up Resistor (P3, P4)

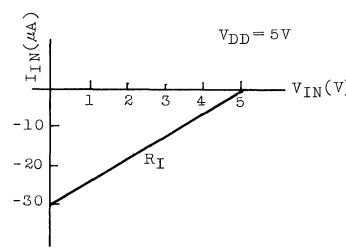


Fig. 3.2 (16) Characteristics of CMOS Input with Pull-up Resistor

(10) Reset circuit (RG)

When the RST terminal is as the "L" level during three instruction cycles at the least (15 clock cycles) under the condition that the power voltage VDD is within the range of the regular voltage, and that oscillation is stable, the CPU is initialized.
The reset input circuit has become the Schmitt circuit with a pull-up resistor, so that a power-on reset becomes possible by externally connecting the capacitor for reset with the diodes for electric discharge.

The Circuit configuration of the $\overline{RST}$ input is shown in Fig. 3.2 (17) and (18), and the standard characteristics are shown in fig. 3.2 (19) and (20), respectively.

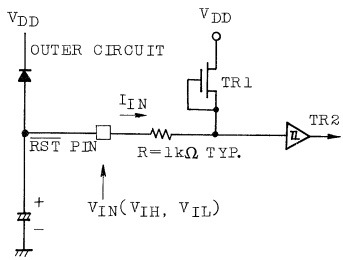


Fig. 3.2 (17) NMOS Version

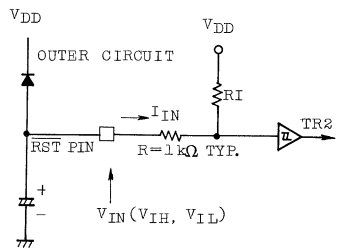


Fig. 3.2 (18) CMOS Version

$\overline{RST}$ Input circuit

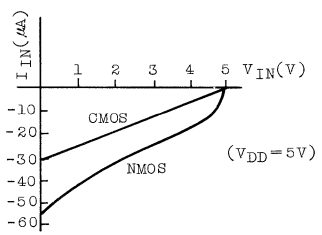


Fig. 3.2 (19) $\overline{RST}$ Input Circuit Pull-up Resistor Characteristic (TR1)

$\overline{RST}$ Input Circuit

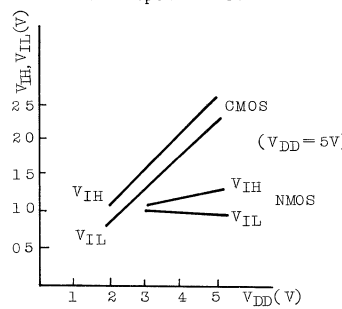


Fig. 3.2 (20) $\overline{RST}$ Input Circuit Schmitt Characteristic (TR2)

(11) Clock generation circuit (CG), Timer circuit (TC)

Oscillation is effected by a ceramic oscillator and the oscillation clock pulses form clock frequencies. Since the range of clock frequencies is from 0.2 MHz to 2 MHz, the ceramic oscillation is required to have a capacity of 1 MHz or 2 MHz as the standard. Fig. 3.2 (21) shows the clock generation circuit for NMOS version and Fig. 3.2. (22) the clock generation circuit for CMOS version. With reference to Fig. 3.2 (22), when the status of "hold" is assumed by a (HOLD) instruction, the HOLD A signal changes to "0" and the oscillation is completely stopped. As soon as the action for releasing the status of "hold" is initiated, the signal is changed to "1" and the oscillation is restarted.

During the reception of an input from an external clock, XOUT is set to an open status by an input from XIN.

The timer circuit is not found in the ROM 0.5K byte grade of NMOS version. Fig. 3.2 (23) shows the configuration of the timer circuit. The timer circuit is formed of 2 binary counters TC which divide the oscilaltion clock pulse into 2048 equal parts.

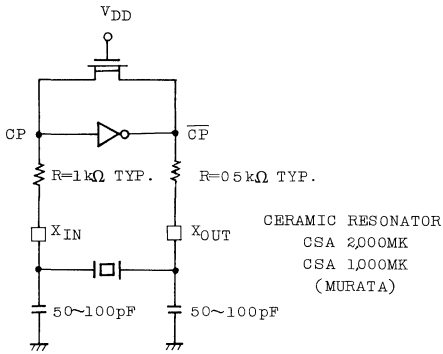


Fig. 3.2 (21) NMOS Version Clock Generation Circuit

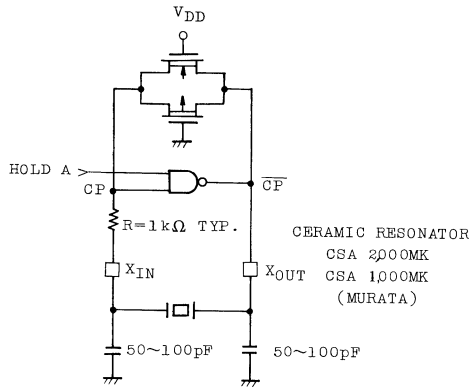


Fig. 3.2 (22) CMOS Version Clock Generation Circuit

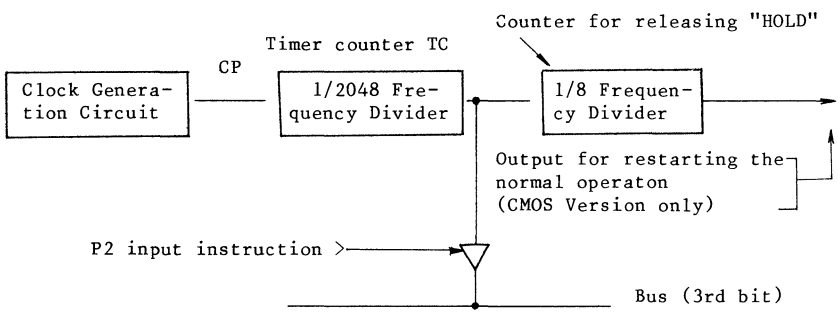


Fig. 3.2 (23) Configuration of Timer Cicuit

When the oscillation frequency (fc) is 2 MHz, the timer circuit functions as a timer having a time cycle of about 1ms. By the input instruction of (IN %P,A) or (IN %P,@HL), the output from the timer is transferred as the data of the third bit of P2 to the accumulator or RAM. In CMOS version, the 1/2048 frequency divider is used additionally for the purpose of interjecting a warming-up time between the time the status of "HOLD" is released and the time the oscillation is stabilized.
Since the warming-up time is required to be ample enough to permit further division of the output of TC into 8 equal parts and issuance of an output for restarting the normal operation, its duration is $2^{13} \times 1000$ /fc (msec). Both the TC and the 1/8 frequency divider are initialized to "0" when the status of "initialization" and the status of "HOLD" are assumed.

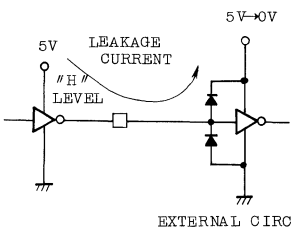
(12) Hold control circuit
The hold function is the function that holds the inside status at lower power consumption immediately before the system operation is stopped by the function provided to the CMOS version only.

The CPU is forced into the hold state by (HOLD) instruction, and the hold state is kept held as long as the hold terminal input HOLD is at the "L" level. The hold state is released when the hold terminal input HOLD goes to the "H" level, and the CPU retunrs to the normal operation.

The hold operation holds the following state:

1] The oscillation is stopped, and all internal operations are stopped.
2] The timer counter TC (1/2048 frequency divider) and counter for releasing hold (1/8 frequency divider) are reset to "0".
3] The conditions of data memory, register, output latch and so on just before hold mode are kept.
4] The program counter holds the address of the instruction following (HOLD) instruction.

a) Output Circuit



b) CMOS Input with Pull-up

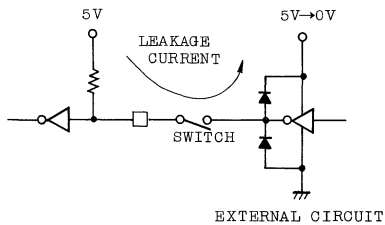


Fig. 3.2 (24) HOLD Operation Interface with The External Circuit

In the hold operation, care should be taken to see that no current flows in the interface with an external circuit. In case where the power supply for external circuit is dropped to 0V, it is required that the output terminals are forced to the "L" level before they start the hold operation, because if the output is at the "H" level as shown in Fig. 3.2 (24) a) "Output Circuit", sometimes the current flows through parasitic diodes, etc. of external circuit. Special attention should be paid to the CMOS input circuit with pull-up shown in b), and it is necessary to use the input circuit by such a method as the current pass is cut off by means of a switch or the like.

4. Basic Operations

4.1 Instruction cycle

The execution of instructions and the operation of the internal hardware are synchronized with the basic clocks (CP, fc Hz).

The minimum unit of the execution of instructions is called a "instruction cycle." All the instructions are executed in one instruction cycle or in two instruction cycles. The instructions thus executed are respectively called "one-cycle instruction" or "two-cycle instruction."

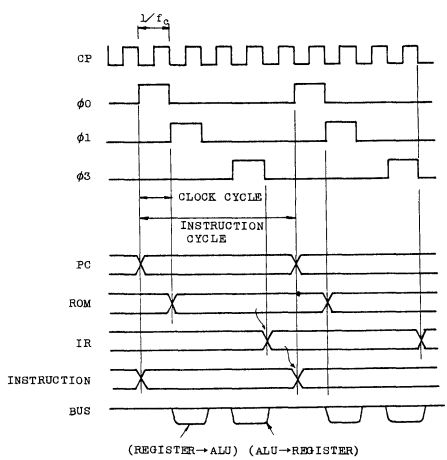


Fig. 4.1 Instruction Cycle

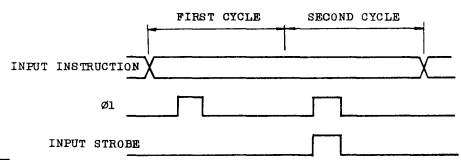


Fig. 4.2(1) Input Timing

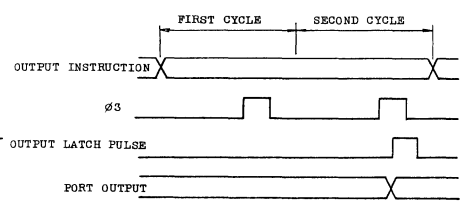


Fig. 4.2 (2) Output Timing

The instruction cycles occur in the three basic states of ϕ0, ϕ1, and ϕ3, each composed of five clock cycles.

The instructions are executed between one ϕ0 and next ϕ0. In the state of ϕ1, the source data from the RAM or the register are fed out on the bus and stored in the temporary register of ALU. In the state of ϕ3, the output data of ALU are fed out on the bus and stored in the RAM or the register.

4.2 I/O timing

(1) The external data from the port are fed in at the state ϕ1 of the second cycle of the input instruction (two-cycle instruction).

The input data from the port are fed out on the bus during the state Ø1 of the second cycle, latched to the temporary register of the accumulator at the fall of the state Ø1, and stored during the state Ø3 in the accumulator or the RAM.

(2) Output timing
The data are fed out to the port at the state Ø3 of the second cycle of the output instruction (two-cycle instruction). The output data are fed out during the state Ø3 of the second cycle onto the bus and latched by the pulse (the pulse shifted by one half time cycle of CP from the state Ø3 of the second cycle during the output instruction).
The data from the port undergo a change at the rising edge of the output latch pulse.

4.3 Initialization operation
If 1] the supply voltage is within the regular voltage, 2] the oscillation is stable, and 3] the $\overline{RST}$ terminal is held at the "L" level in three instruction cycles at least, the CPU is initialized.
The reset circuit contains a delay circuit synchronized with the internal timing behind the Schmitt circuit in order to avoid transfer at unstable level. Therefore, the input signal of $\overline{RST}$ terminal required for producing the internal reset signal is related to internal timing. This relation is shown in Fig. 4.3 (1). If the $\overline{RST}$ terminal is at the "L" level of 12 clock cycle or more, the CPU is sure to be initialized.

In the initialization operation, the internal hardware are initialized as follows:
1] Reset the program counter to "0".
2] Set the status flag SF to "1". (Carry flag CF is indefinite.)
3] Reset the timer counter TC and the hold releasing counter to "0".
4] Initialize the output latch to "1'.
5] Initialize the port register PR to "0". (CMOS version only)

$\overline{RST}$ Pin Level		Initialization Operation
"L" Level	0 to 2 clock cycle	Not initialized
	3 to 11 clock cycle	Indefinite
	12 clock cycle or more	Initialized
"H" Level	0 to 3 clock cycle	Indefinite
	4 clock cycle or more	Initialization release

Fig. 4.3 (1) Initialization Operation

5. Instructions

5.1 Description of symbols

The following symbols are used for describing the instructions in the following explanations.

Symbol	Description
AC	Accumulator
M[x]	Data memory (Address x)
HR	H register
LR	L register
PR	Port register
FLAG	Flag
CF	Carry flag
SF	Status flag
PC	Program counter
STACK	Stack
MBR	Memory bank register
DC	Data counter
ROM[x]	Program memory (Address x)
(ROMH,ROML)	High-order 4 bits or low-order 4 bits are expressed by suffix H/L
←	Transfer
+	Addition
−	Subtraction
∧	Logical AND of the corresponding bits
∨	Logical OR of the corresponding bits
∀	Logical exclusive OR of the corresponding bits
$(\overline{CF})$	Inversion of carry flag contents
null	Processed result is transferred nowhere or nothing is executed.
(AC)	Contents of accumulator
(H.L)	Contents of 5 bits coupling H register with L register
M[(H.L)]	Contents of data memory for which the contents of 5 bits coupling H register with L register is used as address.
P[p]	Contents of port register for which p is used address or contents of terminal
(AC)<b>	Contents of bit assigned by b of accumulator
(LR)<3:2>	Values of bit 3 to bit 2 of L register (4 bit)
(PC)<10:6>	Values of bit 10 to bit 6 of program counter (10 bit)

5.2 Description of instructions (*): Execution cycle

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
Data transfer instruction	LD A,@HL	0000 0110	06	(AC) <-- M[(H.L)] Loads the contents of the data memory specified by the H and L registers in the accumulator.	-	1	1
	LD A,#k	0001 kkkk	1k	(AC) <-- k Loads the immediate value k of the instruction field in the accumulator. Serves as the clear instruction when k=0.	-	1	1
	LD L,#k	0010 kkkk	2k	(LR) <-- K Loads the immediate value k of the instruction field in the L register. Serves as the clear instruction when k=0.	-	1	1
	LD A,x	1001 0xxx	9x	(AC) <-- M[(x)] Loads the contents of the data memory specified by the x of the instruction field in the accumulator.	-	1	2
	LD MBR,#k	1011 kkkk	Bk	(MBR) <-- k Stores the immediate value k of the instruction field in the memory bank register. Serves as the jump instruction to the whole address area when combines with the BSS instruction.	-	-	1
	LDL A,@DC	0110 0111	67	(AC) <-- ROML [(DC)] Loads the lower-order 4 bits of the data table of the program memory specified by the data counter in the accumulator.	-	1	2
	LDH A,@DC	0110 0110	66	(AC) <-- ROMH [(DC)] Loads the higher-order 4 bits of the data table of the program memory specified by the data counter in the accumulator.	-	1	2
	ST A,@HL	0111 0110	76	M[(H.L)] <-- (AC) Stores the contents of the accumulator in the data memory specified by the H and L registers.	-	1	1

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
Data transfer instruction	ST #k,@HL	0011 kkkk	3k	M[(H.L)] <-- k Stores the immediate value k of the instruction field in the data memory specified by the H and L registers. Serves as the clear instruction when k=0.	-	1	1
	ST A,x	1001 1xxx	98+x	M[x] <-- (AC) Stores the contents of accumulator in the data memory specified by x of the instruction field.	-	1	2
	MOV A,L	0000 1100	0C	(LR) <-- (AC) Stores the contents of accumulator in the L register.	-	1	1
	MOV L,A	0000 1111	0F	(AC) <-- (LR) Loads the contents of the L register in the accumulator.	-	1	1
	MOV D,A	0000 1110	0E	(AC) <-- (DC) Loads the contents of the data counter in the accumulator	-	1	1
	MOV A,D	0000 1101	0D	(DC) <-- (AC) Stores the contents of the accumulator in the data counter.	-	1	1
	MOV A,P	0111 1110	7E	(PR) <-- (AC) Stores the contents of the accumulator in the port register.	-	1	1
I/O instruction	IN %P,A	0110 0PPP	6P	(AC) <-- P[p] Loads the input data from the port specified by the P of the instruction field in the accumulator.	-	$\overline{Z}$	2
	IN %P,@HL	0110 1PPP	68+P	M[(H.L)] <-- P[p] Stores the input data from the port specified by the P of the instruction field in the data memory specified by the H and L registers.	-	$\overline{Z}$	2
	OUT A,%P	0111 0PPP	7P	P[p] <-- (AC) Outputs the contents of the accumulator to the port specified by the p of the instruction field.	-	1	2

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
I/O	OUT @HL,%P	0111 1PPP	78+P	P[p] <-- M[(H.L)] Outputs the contents of the data memory specified by the H and L registers to the port specified by the P of the instruction field.	-	1	2
Operation instruction	ADD A,@HL	0000 0011	03	(AC) <-- (AC)+M[(H.L)] Adds the contents of the data memory specified by the H and L registers to those of the accumulator, and stores the result in the accumulator.	-	$\overline{C}$	1
	ADDC A,@HL	0000 0100	04	(AC) <-- (AC)+M[(H.L)]+(CF) Adds the contents of carry flag to the ADD instruction, and stores the result in the accumulator.	C	$\overline{C}$	1
	ADD A,#k	0100 kkkk	4k	(AC) <-- (AC)+k Adds the immediate value k of the instruction field to the contents of the accumulator, and stores the result in the accumulator. Serves as the correction instruction for the decimal addition and subtraction when k=6 or A.	-	$\overline{C}$	1
	ADD L,#k	0101 kkkk	5k	(LR) <-- (LR)+k Adds the immediate value k of the instruction field to the contents of the L register, and stores the result in the L register.	-	$\overline{C}$	2
	SUBRC A, @HL	0000 0101	05	(AC) <-- M[(H.L)]-(AC)-($\overline{CF}$) Subtracts the contents of the accumulator and the inverse contents of the data carry flag from the contents of the data memory specified by the H and L register and stores the result in the accumulator.	C	C	1
	INC D	0000 1011	0B	(DC) <-- (DC)+1 Increments the contents of data counter.	-	$\overline{C}$	1

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
Operation instruction	INC @HL	0000 1001	09	M[(H.L)] <-- M[(H.L)]+1 Increments the contents of data memory specified by the H and L registers.	-	$\overline{C}$	1
	DEC D	0000 1010	0A	(DC) <-- (DC)-1 Decrements the contents of data counter.	-	C	1
	DEC @HL	0000 1000	08	M[(H.L)] <-- M[(H.L)]-1 Decrements the contents of data memory specified by the H and L registers.	-	C	1
Logical Operation instruction	AND A,@HL	0000 0000	00	(AC) <-- (AC) M[(H.L)] Carries out the logical AND of the corresponding bits with the contents of the accumulator and those of the data memory spcfi-fied by the H and L registers, and stores the result in the accumulator.	-	$\overline{Z}$	1
	OR A,@HL	0000 0001	01	(AC) <-- (AC) M[(H.L)] Carries out the logical OR of the corresponding bits with the contents of the accumulator and those of the data memory spcfi-fied by the H and L registers, and stores the result in the accumulator.	-	$\overline{Z}$	1
	XOR A,@HL	0000 0010	02	(AC) <-- (AC) M[(H.L)] Carries out the logical exclusive OR of the corresponding bits with the contens of the accumulator and those of data memory speci-fied by the H and L registers, and stores the result in the ac-cumulator	-	$\overline{Z}$	1
Bit Processing	SET H	1000 1000	88	(HR) <-- 1 Sets the H register to "1".	-	1	2
	SET CF	1000 1001	89	(CF) <-- 1 Sets the carry flag to "1".	1	1	2

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
Bit Processing instruction	SET @HL,b	1000 00bb	80+b	M[(H.L)]<b> <-- 1 Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, to "1".	-	1	2
	CLR H	1000 1010	8A	(HR) <-- 0 Sets the H register to "0".	-	1	2
	CLR CF	1000 1011	8B	(CF) <-- 0 Sets the carry flag to "0".	0	1	2
	CLR @HL,b	1000 01bb	84+b	M[(H.L)]<b> <-- 0 Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, to "0".	-	1	2
	TEST @HL,b	1000 11bb	8C+b	(SF) <-- $\overline{M[(H.L)]<b>}$ Stores the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, in the status flag.	-	*	2
	TESTP CF	0111 0111	77	(SF) <-- (CF) Stores the contents of carry flag in the status flag.	-	*	1
Branch	BSS d	11dd dddd	C0+d	IF SF=1 then (PC) <-- a else null, a=PC<9:6>•d Carries out the branch within a page if the status flag is at "1"; brings the immediate value d of the instruction field into the lower-order 6 bits of the program counter. If this instruction is specified in the last address in the page, branching is carried out to the next page. If the status flag is "0", sets it to "1", and moves to the execution of the next address instruction.	-	1	2

Item Class	Assembler Mnemonic	Object Code Binary	Hex.	Function	Flag CF	Flag SF	(*)
Subroutine	CALLS a	1010 nnnn	An	STACK <-- (PC), (PC) <-- a, a=2n $15 \geqq n > 0$ Carries out the subroutine call. Saves the contents of the program counter in the stack, and doubles the immediate value n of the instruction field, then stores it in the program counter.	-	-	2
	RET	0110 1110	6E	(PC) <-- STACK Returns from subroutine. Restores the return address from stack to the program counter.	-	-	2
Others	HOLD	0000 0111	07	Moves to the hold mode.	-	-	1
	NOP	0111 1111	7F	No operation	-	-	1

5.3 List of Instructions

*: Note
**: Cycle

Class	Assembler Mnemonic	Object code Binary	Object code Hex.	Functions	Flag*1 CF	Flag*1 SF	**	
Data transfer instruction	LD A, @HL	00000110	06	(AC)←M[(H.L)]	-	1	1	
	LD A, #K	0001KKKK	1K	(AC)←K	-	1	1	
	LD L, #K	0010KKKK	2K	(LR)←K	-	1	1	
	LD A, X	10010XXX	90+X	(AC)←M[X]	-	1	2	
	LD MBR, #K	1011KKKK	BK	(MBR)←K	-	-	1	
	LDL A, @DC	01100111	67	(AC)←ROML[(DC)]	-	1	2	
	LDH A, @DC	01100110	'66	(AC)←ROMH[(DC)]	-	1	2	
	ST A, @HL	01110110	76	M[(H·L)]←(AC)	-	1	1	
	ST #K, @HL	0011KKKK	3K	M[(H·L)]←K	-	1	1	
	ST A, X	10011XXX	98+X	M[X]←(AC)	-	1	2	
	MOV A, L	00001100	OC	(LR)←(AC)	-	1	1	
	MOV L, A	00001111	OF	(AC)←(LR)	-	1	1	
	MOV D, A	00001110	OE	(AC)←(DC)	-	1	1	
	MOV A, D	01111110	OD	(DC)←(AC)	-	1	1	
	MOV A, P	01111110	7E	(PR)←(AC)	-	1	1	*3
I/O	IN %p, A	01100PPP	60+P	(AC)←P[p]	-	$\overline{Z}$	2	
	IN %p, @HL	01101PPP	68+P	M[(H·L)]←P[p]	-	$\overline{Z}$	2	
	OUT A, %p	01110PPP	70+P	P[p]←(AC)	-	1	2	
	OUT @HL, %P	01111ppp	78+P	P[p]← M[(H·L)]	-	1	2	
Operation instruction	ADD A, @HL	00000011	03	(AC)←(AC)+M[(H·L)]	-	$\overline{C}$	1	
	ADDC A, @HL	00000100	04	(AC)←(AC)+M[(H·L)]+(CF)	C	$\overline{C}$	1	
	ADD A, #K	0100KKKK	4K	(AC)←(AC)+K	-	$\overline{C}$	1	
	ADD L, #K	0101KKKK	5K	(LR)←(LR)+K	-	$\overline{C}$	2	
	SUBRC A, @HL	00000101	05	(AC)←M[(H·L)]-(AC)-($\overline{CF}$)	C	C	1	
	INC D	00001011	0B	(DC)←(DC)+1	-	C	1	
	INC @HL	00001001	09	M[(H·L)]←M[(H·L)]+1	-	C	1	
	DEC D	00001010	0A	(DC)←(DC)-1	-	C	1	
	DEC @HL	00001000	08	M[(H·L)]←M[(H·L)]-1	-	C	1	
Logic.	AND A, @HL	00000000	00	(AC)←(AC)∧M[(H·L)]	-	$\overline{Z}$	1	
	OR A, @HL	00000001	01	(AC)←(AC)∨M[(H·L)]	-	$\overline{Z}$	1	
	XOR A, @HL	00000010	02	(AC)←(AC)∀M[(H·L)]	-	$\overline{Z}$	1	
Bit processing	SET H	10001000	88	(HR)←1	-	1	2	
	SET CF	10001001	89	(CF)←1	1	1	2	
	SET @HL, b	100000bb	80+b	M[(H·L)]<b>←1	-	1	2	
	CLR H	10001010	8A	(HR)←0	-	1	2	
	CLR CF	10001011	8B	(CF)←0	0	1	2	
	CLR @HL, b	100001bb	84+b	M[(H·L)]<b>←0	-	1	2	
	TEST @HL, b	100011bb	8C+b	(SF)←$\overline{M[(H·L)]<b>}$	-	*	2	
	TESTP CF	01110111	77	(SF)←(CF)	-	*	1	

5.3 List of Instructions

*: Note
**: Cycle

Class \ Item	Assembler Mnemonic	Object code		Functions	Flag*1		**	
		Binary	Hex.		CF	SF		
Branch & Subroutine	BSS a	11dddddd	C0+d	If SF=1 then (PC)←a else null, a=(PC)<9:6>•d	-	1	2	*2
	CALLS a	1010nnnn	An	STACK←(PC), (PC)←a, a=2n, 15≧n>0	-	-	2	
	RET	01101110	6E	(PC)←STACK	-	-	2	
Others	HOLD	00000111	07	HOLD	-	-	1	*3
	NOP	01111111	7F	no operation	-	-	1	

Note 1) The contents of the program counter indicate the next address of the instruction to be executed.

Note 2) The setting condition "C" of flag indicates the carry output from the most significant position in the addition operation, and the no borrow output from the significant position in the subtraction operation. "Z" indicates the zero detection signal to which "1" is applied only when the operation, processing result or all four bits of the data transferred to the accumulator are zero.
The flag is det to "C","$\overline{C}$","Z","1" or "0" according to the data processing result. The value specified by the function is set to the flag with mark "*", and the "-" denotes no change in the state of the flag.

Note 3) MOV A, P and HOLD are equivalent operations to NOP in NMOS version.

L : Lower order H : Higher order

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	AND A, @HL	OR A,@HL	XOR A,@HL	ADD A,@HL	ADDC A,@HL	SUBRC A,@HL	LD A,@HL	HOLD	DEC @HL	INC @HL	DEC D	INC D	MOV A,L	MOV A,D	MOV D,A	MOV L,A
1	LD A, #K															
2	LD L, #K															
3	ST #K, @HL															
4	ADD A, #K															
5	ADD L, #K															
6	IN %p, A						LDH A,@DC	LDL A,@DC	IN %p, @HL						RET	
7	OUT A, %p						ST A,@HL	TESTP CF	OUT @HL, %p						MOV A,P	NOP
8	SET @HL, b				CLR @HL, b				SET H	SET CF	CLR H	CLR CF	TEST @HL, b			
9	LD A, x								ST A, x							
A	CALLS a															
B	LD MBR, #K															
C	B S S a															
D																
E																
F																

(Note) 1. Blank code is undefined.
2. The block with ☐ are 2 sycle instruction.

6. Electrical Characteristics. Outline Drawing

6.1 Electrical characteristics

(1) Electrical characteristics NMOS version (TMP4240P, TMP4250N, TMP4260P, TMP4270N)

ABSOLUTE MAXIMUM RATINGS (VSS=0V)

SYMBOL	ITEM	RATING	UNIT
VDD	Supply Voltage	-0.5 - 7	V
VIN	Input Voltage	-0.5 - 7	V
VOUT1	Output Voltage (except open drain pin)	-0.5 - 10	V
VOUT2	Output Voltage (open drain pin)	-0.5 - 15	V
PD	Power Dissipation (Topr=85°C)	300	mW
Tsld	Soldering Temperature. Time	260 (10 sec)	°C
Tstg	Storage Temperature	-55 - 125	°C
Topr	Operating Temperature	-40 - 85	°C

RECOMMENDED OPERATING CONDITIONS (VSS=0V, VDD=5V±10%)

SYMBOL	PARAMTER	CONDITION	MIN.	MAX.	UNIT
Topr	Operating Temperature		-40	85	°C
VDD	Supply Voltage		4.5	5.5	V
VIH1	Input High Voltage(exc.$\overline{RST}$ input)		2.2	VDD	V
VIH2	Input High Voltage($\overline{RST}$ input)		3	VDD	V
VIL1	Input Low Voltage (exc.$\overline{RST}$ input)		0	0.8	V
VIL2	Input Low Voltage ($\overline{RST}$ input)		0	0.6	V
fc	Clock Frequency		0.2	2	MHz
tWCH	High Clock Pulse Width (Note 1)	VIN=VIH	100	-	ns
tWCL	Low Clock Pulse Width (Note 1)	VIN=VIL	100	-	ns

(Note 1) Under external clock operation

D.C CHARACTERISTICS (VSS=0V, VDD=5V±10%, Topr=-40 to 85°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
VHS	Hysteresis Voltage($\overline{RST}$ input)	Ta=25°C	-	0.3	-	V
IIN1	Input Current ($\overline{RST}$ input)	VDD=5.5V,VIN=0.6V	-	-50	-100	μA
IIN2	Input Current (*)	VDD=5.5V,VIN=0.4V	-	-0.1	-2	μA
IIL	Input Low Current (**)	VDD=5.5V,VIN=0.6V	-	-	-0.36	mA
ILO	Output Leakage Current (*)	VDD=5.5V,VOUT=5.5V	-	0.1	2	μA
IOH	Output High Current	VDD=4.5V,VOH=2.4V	-50	-	-	μA
IOL1	Output Low Current 1	VDD=4.5V,VOL=0.4V	1.6	6	-	mA
IOL2	Output Low Current 2 (Note 2)	VDD=4.5V,VOL=1.2V	10	16	-	mA
VOH	Output High Voltage	VDD=5.0V,IOH=-5μA	4.7	4.9	-	V
IDD	Supply Current	VDD=5.5V	-	13	28	mA

(Note 2) P00 to P03 only is possible. IOL is possible up to 30mA in the sum total. (*): With pull-up, Ta=25°C. (**): Open drain

A.C CHARACTERISTICS (VSS=0V, VDD=5V±10%, Topr=-40 to 85°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	Instruction Cycle Time		2.5		25	μs

(2) Electrical characteristics CMOS version TMP42C40P, TMP42C60P

ABSOLUTE MAXIMUM RATINGS (VSS=0V)

SYMBOL	ITEM	RATING	UNIT
VDD	Supply Voltage	-0.5 - 7	V
VIN	Input Voltage	-0.5 - VDD+0.5	V
VOUT	Output Voltage	-0.5 - VDD+0.5	V
PD	Power Dissipation (Topr=85°C)	300	mW
Tsld	Soldering Temperature. Time	260 (10 sec)	°C
Tstg	Storage Temperature	-55 - 125	°C
Topr	Operating Temperature	-40 - 85	°C

RECOMMENDED OPERATING CONDITIONS (VSS=0V, VDD=5V±20%)

SYMBOL	PARAMTER	CONDITION	MIN.	MAX.	UNIT
Topr	Operating Temperature		-40	85	°C
VDD	Supply Voltage		4.0	6.0	V
VIH1	Input High Voltage(exc.$\overline{\text{RST}}$ input)		VDDx0.7	VDD	V
VIH2	Input High Voltage($\overline{\text{RST}}$ input)		VDDx0.8	VDD	V
VIL1	Input Low Voltage (exc.$\overline{\text{RST}}$ input)		0	VDDx0.3	V
VIL2	Input Low Voltage ($\overline{\text{RST}}$ input)		0	VDDx0.2	V
fc	Clock Frequency		0.2	2	MHz
tWCH	High Clock Pulse Width (Note 1)	VIN=VIH	100	-	ns
tWCH	Low Clock Pulse Width (Note 1)	VIN=VIL	100	-	ns

(Note 1) Under external clock operation

D.C CHARACTERISTICS (VSS=0V, VDD=5V±20%, Topr=-40 to 85°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
VHS	Hysteresis Voltage($\overline{\text{RST}}$ input)	Ta=25°C	-	0.3	-	V
IIH	Input High Current (PORT)	VDD=6.0V,VIN=6.0V	20	40	100	μA
IIL	Input Low Current ($\overline{\text{RST}}$)	VDD=5.0V,VIN=0V	-	-30	-100	μA
VOH	Output High Voltage	VDD=5.0V,IOH=-5μA	4.7	4.9	-	V
IOH	Output High Current	VDD=4.0V,VOH=2.4V	-1.0	-2.5	-	mA
IOL	Output Low Current	VDD=4.5V,VOL=0.4V	1.6	3	-	mA
IDDO	Operating Supply Current (*)	VDD=6.0V,fc=2MHz	-	0.8	3	mA
IDDH	Holding Supply Current	VDD=5.0V, (**)	-	0.1	5	μA

(*): The $\overline{\text{RST}}$ pin is 0V, XOUT under external clock operation and port is released for supply current. (**): Releases except for supply pin.

A.C CHARACTERISTICS (VSS=0V, VDD=5.0V±20%, Topr=-40 to 85 °C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	Instruction Cycle Time		2.5		25	μs

(3) Electrical characteristics CMOS version (TMP42C50N, TMP42C70N)

ABSOLUTE MAXIMUM RATINGS (VSS=0V)

SYMBOL	ITEM	RATING	UNIT
VDD	Supply Voltage	-0.5 - 7	V
VIN	Input Voltage	-0.5 - VDD+0.5	V
VOUT1	Output voltage (except open drain pin)	-0.5 - VDD+0.5	V
VOUT2	Output Voltatge (open drain pin)	-0.5 - 12	V
PD	Power Dissipation (Topr=85°C)	300	mW
Tsld	Soldering Temperature. Time	260 (10 sec)	°C
Tstg	Storage Temperature	-55 - 125	°C
Topr	Operating Temperature	-40 - 85	°C

RECOMMENDED OPERATING CONDITIONS (VSS=0V, VDD=5V±20%)

SYMBOL	PARAMTER	CONDITION	MIN.	MAX.	UNIT
Topr	Operating Temperature		-40	85	°C
VDD	Supply Voltage		4.0	6.0	V
VIH1	Input High Voltage(exc.$\overline{\text{RST}}$ input)		VDDx0.7	VDD	V
VIH2	Input high Voltage($\overline{\text{RST}}$ input)		VDDx0.8	VDD	V
VIL1	Input Low Voltage (exc.$\overline{\text{RST}}$ input)		0	VDDx0.3	V
VIL2	Input Low Voltage ($\overline{\text{RST}}$ input)		0	VDDx0.2	V
fc	Clock Frequency		0.2	2	MHz
tWCH	High Clock Pulse Width (Note 1)	VIN=VIH	100	-	ns
tWCL	Low Clock Pulse Width (Note 1)	VIN=VIL	100	-	ns

(Note 1) Under external clock operation

D.C CHARACTERISTICS (VSS=0V, VDD=5V±20%, Topr=-40 to 85°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
VHS	Hysteresis Voltage($\overline{\text{RST}}$ input)	Ta=25°C	-	0.3	-	V
IIH	Input High Current (P0,P1,P2)	VDD=6.0V,VIN=6.0V	20	40	100	μA
IIN1	Input Low Current ($\overline{\text{RST}}$)	VDD=5.0V,VIN=0V	-	-30	-100	μA
IIL	Input Low Current (a)	VDD=6.0V,VIN=0.6V	-	-30	-100	μA
VOH	Output High Voltage(b)	VDD=5.0V,IOH=-5μA	4.7	4.9	-	V
IOH	Output High Current	VDD=4.0V,VOH=2.4V	-1.0	-2.5	-	mA
IOL	Output Low Current	VDD=4.5V,VOL=0.4V	1.6	3	-	mA
IIN2	Input Current (c)	VDD=6.0V,VIN=0.4V	-	-0.1	-2	μA
ILO	Output Leakage Current (c)	VDD=6.0V,VIN=6.0V	-	0.1	2	μA
IDDO	Operating Supply Current (d)	VDD=6.0V,fc=2MHz	-	0.8	3	mA
IDDH	Holding Supply Current	VDD=5.0V. (e)	-	0.1	5	μA

(a): With pull-up P3, P4 (b): CMOS Output (c): N-ch open drain drain P3,P4
(d): The $\overline{RST}$ pin is 0V, XOUT under external clock operation and port is released for supply current. (e): Releases except for supply pin. When NMOS I/O port is specified under mask option, this port is fixed at 0V.

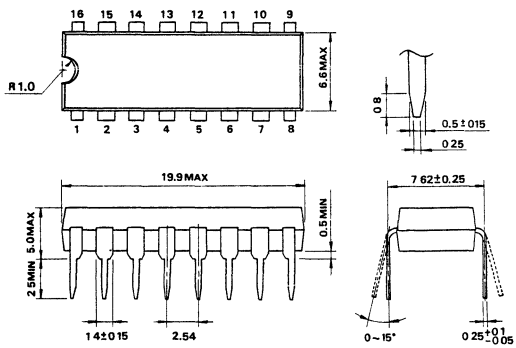
A.C CHARACTERISTICS (VSS=0V, VDD=5V±20%, Topr=-40 to 85°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	Instruction Cycle Time		2.5		25	μs

6.2 Outline Drawing

(1) 16 PIN Type

Unit: mm

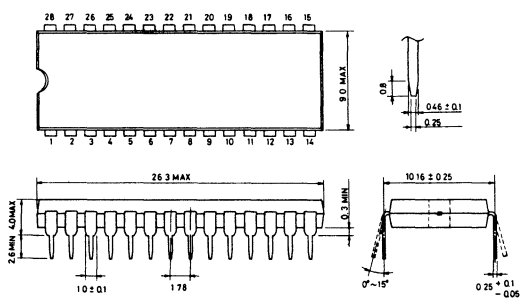


EIAJ SC-503-16A
JEDEC MO-C01AC
TOSHIBA 5-22E

(Note) Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No. 1 and No. 16.

(2) 28 PIN Type

Unit: mm



(Note) Each lead pitch is 1.78mm, and the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No. 28.

7. Designation of Formats of Program Tape and I/O Circuit

The user of TLCS-42N and TLCS-42C is requested to designate his program data and I/O circuit forms with a paper tape. We will draw up evaluation samples based on the information.
The format of this paper tape is required to conform to the Hex Format of Intel (I Format).
Of course, the program data must be designated within the address space proper for the capacity of the ROM incorporated. The address space covers the 000 - 1FF locations in the case of ROM 0.5K version.

(1) Designation of I/O circuit formats
The paper tape of I Format starts recording the program data by the record mark ":". The designation of an I/O circuit code is effected by the parenthesized data contents immediately preceding the first record mark.
The designation of the I/O circuit code is effected with the name of the pin of the relevant port (NMOS) or with the option code indicated by the alphabetic characters, A - D (CMOS).

(Note) Where the I/O circuit code is not designated, TLCS-42N processes the data on the assumption that no pull-up resistance is involved and the 28-pin grades (TMP42C50N and TMP42C70N) of TLCS-42C are not allowed to accept any program tape because the I/O circuit format is not definite. The acceptance of a program tape is made impossible when the designated format is not correct or when an illegal I/O circuit code is designated.

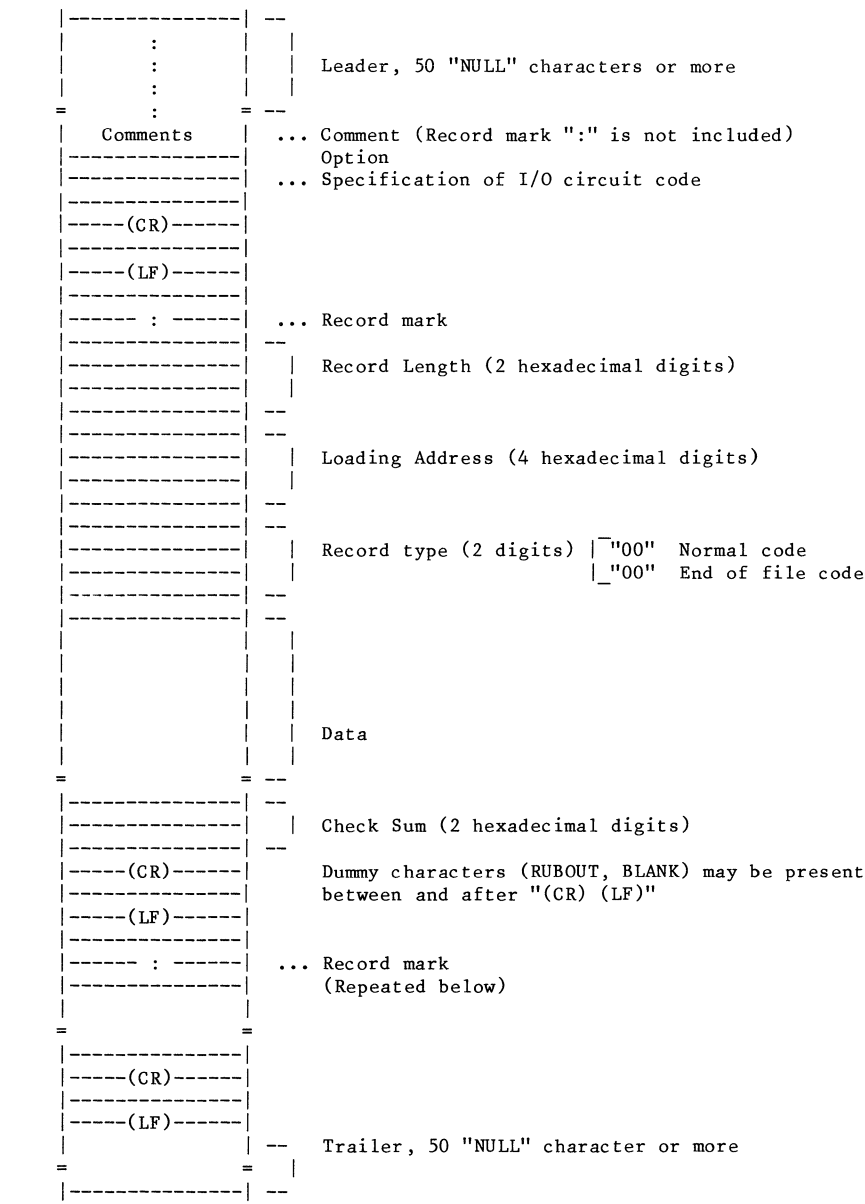
(2) Example of port mask OPTION Format (contents of paper tape)
Example of NMOS version

(P00)
(P01)
(P02)
(P03)
(P20)
(P21)
(P40)
(P41)
(P42)
(P43)

Only the port which has the pin name designated within the parentheses in the paper tape incorporates a built-in pull-up resistor. The port which has no such designation has no built-in pull-up resistor. Any two successive sets of parentheses must be served from each other by insertion of <CR> <LF> codes without fail.

Example of CMOS version
(A) ..... Any one of the alphabetic characters, A - D, must be select without fail. (cf. Fig. 3.2 (12))

(3) Tape format



(4) Example of tape list (TMP4270N)

```
TOSHIBA MICRO COMPUTER TLCS-42N
(P00)
(P01)
(P32)
(P33)
(P50)
(P52)
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FFIFB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5ADIE41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E3A5B6138060B20BC372BF60BD6
:00000001FF
```

(5) Example of tape list (TMP42C70N)

```
TOSHIBA MICRO COMPUTER TLCS-42C
(A)

:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FFIFB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5ADIE41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E3A5B6138060B20BC372BF60BD6
:00000001FF
```

POSTSCRIPT

This manual is a reference for the customer applying the TLCS-42 series. It contains the function and specification of each LSI device of the TLCS-42. The examples of application circuits described here are shown as the reference for explanation. Toshiba assumes no responsibility for any problems caused by using these examples. The information herein is subject to change without prior notice.

This manual has been prepared by the following section.

Microcomputer LSI Application Engineering Section
Integrated Circuit Division, Toshiba Corporation

1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 210, Japan
Phone: Japan (81)44-511-3111

PART 2

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-47

LSI DEVICES

July. 1 9 8 4

PREFACE

This part describes the detail functions and specifications of the LSI devices of the single chip microcontroller TLCS-47 series. The TLCS-47 series consists of NMOS and CMOS devices. These are pin and software compatible with each others. The TLCS-47 has an improved system architecture, highly efficient instruction set and variety in I/O characteristics. There are NMOS devices of high-speed and high-current driving output and CMOS devices of low and high-breakdown voltage out put or build-in liquid crystal driving circuit. The TLCS-47 is a new-generation high performance microcontroller series with a multiple interrupt processing mechanism, sufficient subroutine nesting stack, two timer/event counters, sireal I/O port with buffer, etc. Toshiba has further plans to develop optimum chips for each application field by attaching, eliminating, or modifying additional circuits and input-output functions.

CONTENTS

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series suitable for microcontroller. Various powerful functions have been integrated on the TLCS-47 chips in order to meet with the advanced and complicated applications, which will be made in near future. The TLCS-47 series consists of software compatible NMOS devices and CMOS devices.

FEATURES

(1) 4-bit single chip microcomputer with built-in
ROM, RAM, I/O ports, divider, timer/counters, and serial port.
(2) Memory capacity
ROM: Max. 4,096 x 8 bits and RAM: Max. 256 x 4 bits
(3) Instruction execution time
NMOS: 2us (at 4MHz clock) and CMOS: 4us (at 4MHz clock)
(4) Efficient instruction set
90 instructions, Software compatible in the series
(5) Subroutine nesting: Max. 15 levels
(6) 6 interrupts (External: 2, Internal: 4)
Independently latched, multiple levels of interrupts
(7) Input/Output (Standard products have 35 I/O pins)

. Input	1 port	4 pins
. Output (corresponding to PLA)	2 ports	8 pins
. Input and Output	4 ports	16 pins
. Input and Output (Note)	2 ports	7 pins

Note: These I/O ports are also used for the interrupt inputs, timer/counter inputs, and serial ports, respectively, and programmably selectable for each application.
(8) Data conversion instruction: from 5 bits (C,AC) to 8 bit output port.
Equivalent to PLA function
(9) Data counter and Read ROM instruction
Table of constant data can be set up in the whole ROM area.
(10) Two 12-bit timer/counters
(11) Serial port with 4-bit buffer
(12) 18-stage divider (with 4-stage prescaler)
(13) Built-in high current outputs (NMOS devices)
Typ. 20mA x 8 bits, directly driving LED
(14) Built-in high breakdown voltage outputs (CMOS devices)
Max. 42V breakdown voltage, directly driving vacuum fluorescent tube
(15) Built-in LCD drive circuit (automatically display) (CMOS devices)
Directly driving liquid crystal display (Max. 12-digits at 1/4 duty)
1/4, 1/3, 1/2 duties or static LCD drive is programmably selectable.
(16) Stand-by operation (NMOS/CMOS)
Battery back-up, battery operation and condenser back-up are available.
(17) On chip oscillator
(18) TTL/CMOS compatible
(19) +5V single power supply

LIST OF TLCS-47 LSI DEVICES (1/2)

Series			TLCS-47 NMOS				
Item		Unit	TMP47 00AC	TMP47 99C	TMP47 20P/N	TMP47 40P/N	*TMP47 46N
ROM Capacity		Bytes	External 4,096	4,096	2,048	4,096	4,096
RAM Capacity		Nibbles	256	256	128	256	256
Inst. Execution Time		usec.	2				
No. of Instructions			90				
Subroutine Nesting		Levels	Max.15				
Interrupts	External		2				
	Internal		4 (Serial I/O, timer/counter (2), and divider)				
Timer/counter		Ch.	2				
	(Bit length)	Bits	12				
	(Mode)		Event counter, timer or pulse widch measurement mode is programmably selectable.				
Serial port		Bits	4 (With buffer)				
	(Mode)		Receive/transmit mode is programmably selectable.				
	(Clock)		External/internal, and leading/trailing edge mode are programmably selectable.				
Divider		Stages	18 (With 4-stage prescaler)				
Input/ Output	Input	Bits	4		4		4
	Output (equivalent to PLA)		8		8		8
Ports	Output		-		-		8
	I/O		16		16		30
	I/O(Combined use)		7		7		7
	Total		35		35		57
With built-in high current outputs			8		8		8
With built-in high breakdown voltage outputs			-		-		-
With built-in LCD driver			-		-		-
Memory Standby operation			YES				
Hold operation			-				
Clock oscillator			on chip				
Power supply		V	+ 5.0				
Process			Si Gate Nch E/D MOS				
Package			QIC-80	DIC-42	P: DIP-42 N: Shrunk DIP-42		Shrunk DIP-64
Emulator for debugging			(Evaluator Chip) (Piggy back type)		BM4717 +BM4710		BM4717 +BM4710 +BM4714

* : Under Development

LIST OF TLCS-47 LSI DEVICES (2/2)

Series		TLCS-47CMOS					
Item	Unit	TMP47 C20P/N	TMP47 C40P/N	TMP47 C21P/N	TMP47 C41P/N	TMP47 C22F	TMP47 C46N
ROM Capacity	Bytes	2,048	4,096	2,048	4,096	2,048	4,096
RAM Capacity	Nibbles	128	256	128	256	192	256
Inst. Execution Time	usec.	4					
No. of Instructions		90					
Subroutine Nesting	Levels	Max.15					
Interrupts External		2					
Interrupts Internal		4 (Serial I/O, timer/counter (2), and divider)					
Timer/counter	Ch.	2					
Timer/counter (Bit length)	Bits	12					
Timer/counter (Mode)		Event counter, timer or pulse widch measurement mode is programmably selectable.					
Serial port	Bits	4 (With buffer)					
Serial port (Mode)		Receive/transmit mode is programmably selectable.					
Serial port (Clock)		External/internal, and leading/ trailing edge mode are programmably selectable.					
Divider	Stages	18 (With 4-stage prescaler)					
Input/Output Ports Input	Bits	4		4		4	4
Input/Output Ports Output (equivalent to PLA)		8		8		-	8
Input/Output Ports Output		-		12		-	8
Input/Output Ports I/O		16		4		16	30
Input/Output Ports I/O(Combined use)		7		7		7	7
Input/Output Ports Total		35		35		27	57
With built-in high current outputs	Bits	-		-		-	-
With built-in high breakdown voltage outputs		-		8 + 12		-	-
With built-in LCD driver		-		-		24 + 4	-
Memory Standby operation		-					
Hold operation		YES					
Clock oscillator		on chip					
Power supply	V	+ 5.0					
Process		Si Gate CMOS					
Package		P:DIP-42 N:Shrunk DIP-42				FP-67	Shrunk DIP-64
Emulator for debugging		BM4717 + BM4711A				BM4717 +BM4711A +BM4712A	BM4717 +BM4711A +BM4714

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-47

NMOS DEVICES

July. 1 9 8 4

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4740P TMP4720P

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47N) TMP4740P, TMP4720P

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series designed for the general purpose use.

The TLCS-47 has variously powerful functions in order to meet with the advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

The TMP4740P and TMP4720P are the standard chips for the TLCS-47N. These chips are similar to each other, except memory capacity.
The TMP4700AC is an evaluator chip used for the system development.

Part No.	ROM (Bit)	RAM (Bit)
TMP4740P	4,096 × 8	256 × 4
TMP4720P	2,048 × 8	128 × 4
TMP4700AC	Externally provided (4,096 × 8)	256 × 4
TMP4799C	Externally provided (4,096 × 8)	256 × 4

FEATURES

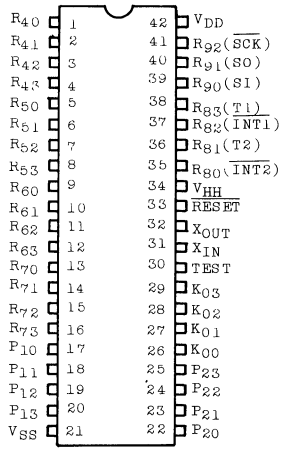
- 4-bit single chip microcomputer with built-in ROM, RAM, input/output port, divider, timer/counter, and serial port.
- Instruction execution time : 2 μs (at 4 MHz clock)
- Effective instruction set
  90 instructions, Software compatible in the seires
- Subroutine nesting : Maximum 15 levels
- 6 interrupts (External : 2, Internal : 4)
  Independently latched control and multiple interrupt control
- Input/Output port (35 pins)

Input	1 port	4 pins
tput (corresponding to PLA)	2 ports	8 pins
	4 ports	16 pins
l/O (Note)	2 ports	7 pins

  Note : These I/O ports are also used for the interrupt input, timer/counter input, and serial port; therefore, it is programmably selectable for each application.

- PLA data converting function (Instruction)
  Output of data to output port (8-bit)
- Table look-up and table search function (Instruction)
  Table can be set up in the whole ROM area.
- 12-bit timer/counter (2 channels)
  Event counter, timer, and pulse width measurement mode is programmably selectable.
- Serial port with 4-bit buffer
  Receive/Transfer mode is programmably selectable.
  External/Internal clock and Leading/Trailing edge mode are programmably selectable.
- 18-stage divider (with 4-stage precaler)
  Frequency applied for timer interrupt of divider is programmably selectable.
- High output current (Output ports)
  TYP. 20mA × 8 bits, LED direct drive is available.
- Memory stand-by function : Battery backup is available.
- On chip oscillator
- TTL/CMOS Compatible
- +5V single power supply
- 42-pin DIL plastic package
- N-channel Si gate E/D MOS LSI

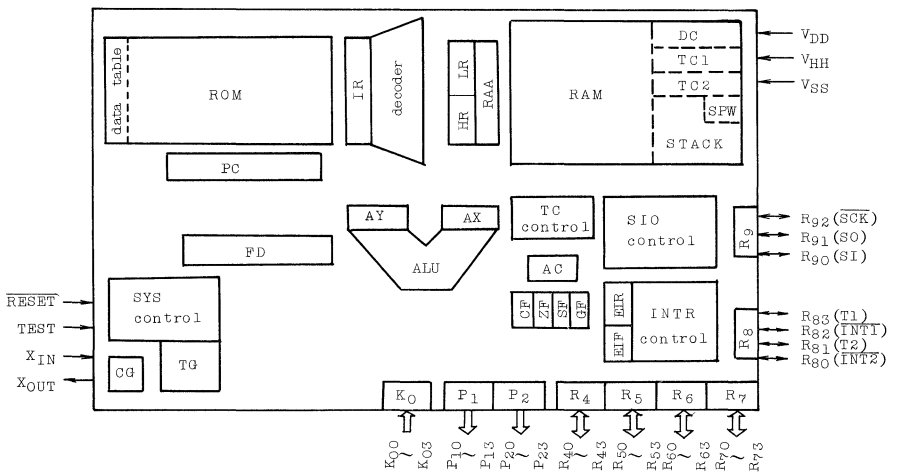
PIN CONNECTIONS (TOP VIEW)



PIN NAMES AND PIN DESCRIPTION

Pin Name	No.of pins	Input/Output	Function
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (Corresponding to PLA)
$P_{23} \sim P_{20}$	4	Output	" (")
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
R_{83} (T1)	1	I/O	I/O port or timer/counter input
R_{82} ($\overline{INT1}$)	1	I/O	I/O port or interrupt input
R_{81} (T2)	1	I/O	I/O port or timer/counter input
R_{80} ($\overline{INT2}$)	1	I/O	I/O port or interrupt input
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	I/O port or serial output
R_{90} (SI)	1	I/O	I/O port or serial input
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminals
$\overline{RESET}$	1	Input	Initialize signal input
TEST	1	Input	(Low level is input.)
V_{DD}	1	Power supply	+5V
V_{HH}	1	Power supply	+5V (Memory power supply)
V_{SS}	1	Power supply	0V

BLOCK DIAGRAM



BLOCK NAMES AND DESCRIPTION

Block Name	Function
PC	Program counter (12 bits)
ROM	Program memory (including fixed data)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Data memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC, data table	Data counter (12 bits, RAM area), Data table (ROM area).
AX, AY	Temporary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG (CF,ZF,SF,GF)	Flags
K, P, R	Ports
INTR Control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2 channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
SYS control	Generation of various internal control signals
CG, TG	Clock generator, Timing generator

FUNCTIONAL DESCRIPTION

1. System Configuration

1. Program Counter (PC)
2. Program Memory (ROM)
3. H Register (HR), L Register (LR), RAM Address Buffer Register (RAA)
4. Data Memory (RAM)
   (1) Stack (STACK)
   (2) Stack Pointer Word (SPW)
   (3) Data Counter (DC)
5. ALU, Accumulator (AC)
6. Flags (FLAC)
7. Ports (PORT)
8. Interrupt Control Circuit (INTR)
9. Frequency Divider (FD)
10. Timer/Counter (TC_1, TC_2)
11. Serial Port (SIO)

Concerning the above component parts, the configuration and functions of hardwares are described :

Hexadecimal notation is used for the description, charts, and tables in order to indicate the address and the like, without assigning identification symbols as far as it does not give rise to confusion.

The following names and symbols are used unconsciously.

(a)	CPU	Control Processing Unit except for the built-in peripheral circuitry, such as interrupt control circuit, timer/counter, and serial port.
(b)	CP	Clock pulse generated in the clock oscillator. It is called the "basic clock" or merely "clock".
(c)	fc	Indicates the frequency of the clock oscillator, namely, the frequency of the basic clock.
(d)	MSB/LSB	Indicates Most/Least Significant Bit.
(e)	F/F	Indicates Flip/Flop.

1.1 Program Counter (PC)

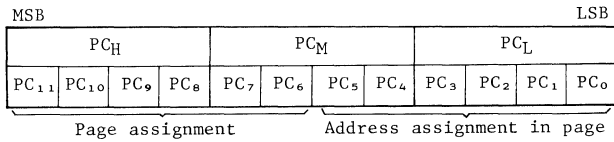
It is a 12-bit binary counter, and the contents of the program counter indicate the address of program memory in which the next instruction to be executed is stored.

The program counter generally gains increment at every instruction fetch by the number of bytes assigned to the instruction. However, when executing the branch and subroutine instructions or receiving the interrupt, the values specified by these instructions and operation are set.
Value "0" is specified by initializing the program counter.

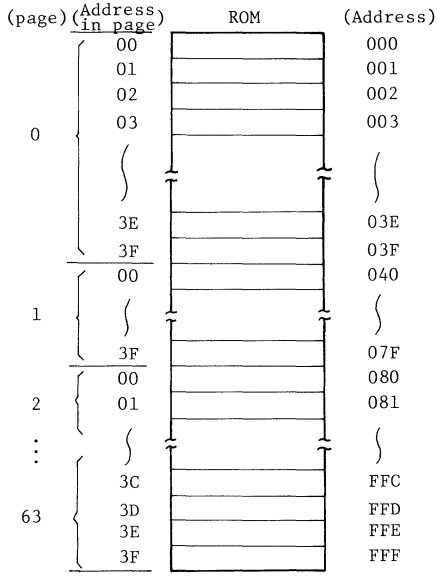
The page structure of program memory is made with 64 words per page. The TMP4740P has 64 pages and the TMP4720P 32 pages.

At the execution of (BSS a) instruction, the value assigned by the instruction is set in the lower 6 bits of the program counter when the branch condition is met. That is, the (BSS a) instruction is used as a branch or jump instruction within a page. If the (BSS a) instruction is stored in the last address of the page, the value in the higher 6 bits of the program counter indicates that the branch or jump instruction to the next page is executed.

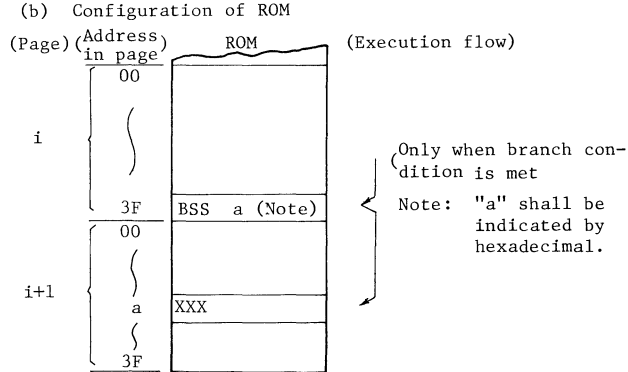
At the execution of (CALL a) instruction, the value specified by the instruction is set in the program counter after the previous contents of the program counter has been saved in the stack. Since 11 bits are of the address bit length which can be assigned by the instruction, the call address of subroutine should be in the range of addresses 000 - 7FF.



(a) Configuration of Program Counter



(b) Configuration of ROM



(c) Special example of branch caused by (BSS a)instrustion.

Fig.1.1.1 Program Counter and Program Memory (ROM)

1.2 Program Memory (ROM)

Processing programs and fixed data are stored in the program memory. The next instruction to be executed is read out from the address indicated by the contents of the program counter.

The fixed data stored in the program memory can be read by using the ROM data referring instruction or the PLA referring instruction. The ROM data referring instruction reads out the higher or lower 4-bit data of the fixed data stored in the address decided by the data counter [(LDH A, @DC+) and (LDL A, @DC) instruction respectively], and stores the data in the accululator. The PLA referring instruction (OUTB @HL) reads out the fixed data (8-bit) stored in the address decided by the contents of the data memory indicated by the contents of H and L registers as well as contents of the carry flag, and outputs the data to output ports (P2 · P1).

Addresses are individually assignged to the program memory and data memory, so that the fixed data in the ROM area cannot be directly read out bv the address of the data memory.

Specific Addresses of Program Memory

The following addresses of the program memory are used for specific purposes. When not used for these purposes, the specific addresses can be used to store the processing programs and fixed data.

Specific Address	Specific Purposes
000 (001)	Start address by initialization
002 (003)	INT1 Interrupt vector address
004 (005)	ISIO Interrupt vector address
006 (007)	IOVF1 Interrupt vector address
008 (009)	IOVF2 Interrupt vector address
00A (00B)	ITMR Interrupt vector address
00C (00D)	INT2 Interrupt vector address
8n + 6 (n = 1 ∿ 15) 086 (Note)	Call address by instruction (CALLS a)
FE0 ∿ FFF	PLA data conversion table

Note : 086 (hexadecimal) = 134 (decimal)

Table 1.2.1 Specific Address of Program Memory

ROM CAPACITY

The TMP4740P and TMP4720P contain a program memory with 4,096 x 8-bit (addresses 000 - FFF) capacity and 2,048 x 8-bit (addresses 000 - 7FF) capacity, respectively. But the TMP4720P contains a program counter with 12-bit length. Therefore, when one of addresses 800 - FFF is accessed in a program, the ROM data corresponding to addresses 000 - 7FF read out. It is because there is no physical ROM in addresses 800-FFF, but the MSB in the program counter is not decoded. For example, when the data located in address FF3 is output to a port by the PLA referring instruction on a program, the data located in address 7F3 is read out. In the TMP4720P, the PLA data conversion table (addresses FE0 - FFF) is, therefore, located in addresses 7E0 - 7FF.

"0" [(NOP) instruction] is read out for the ROM data within the range of the built-in ROM capacity, if it is not specified by the user.

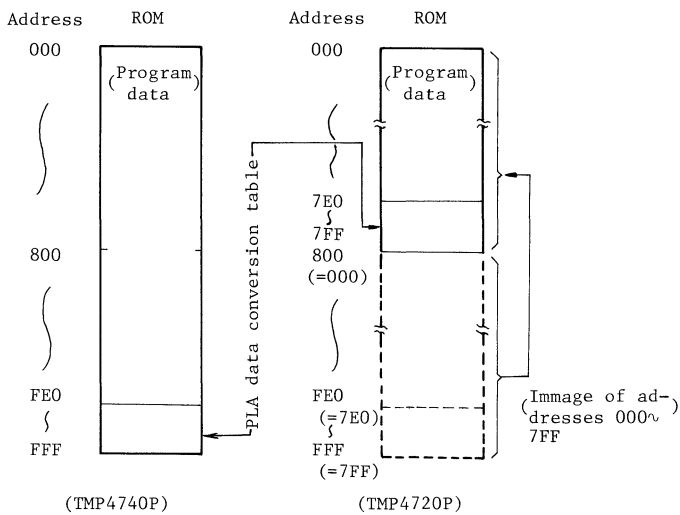


Fig. 1.2.1 ROM Capacity and Address

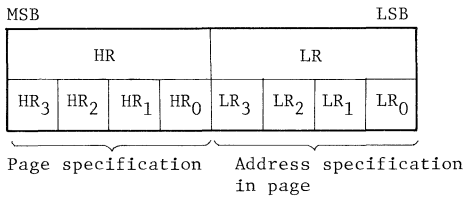
1.3 H Register (HR), L Register (LR), and RAM Address Buffer Register (RAA)

The H and L registers are 4-bit registers used as the data memory address pointers or general purpose registers.

The page structure of the data memory is based on 16 words per page. Pages are specified by H register, and addresses in page are done by L register, respectively. TMP4740P has 16 pages and TMP4720P 8 pages.

The L register is also used to specify the bits corresponding to pins $R_{73} \sim R_{40}$ of the I/O port when instructions (SET @L), (CLR @L), and (TEST @L), are executed.

The RAM address buffer register is a temporary register used to specify the address in the data memory, and serves as an input of the RAM address decoder. Normally, the data specified by the contents of the H and L registers or immediate data of an instruction is fed into the RAM address buffer register.



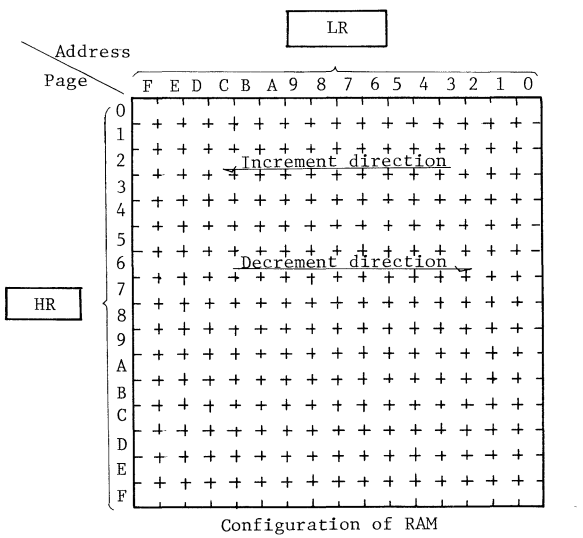


Fig. 1.3.1 H Register, L Register and Data Memory (RAM)

1.4 Data Memory (RAM)

The processing data of user are stored in the data memory. The data is read out or written in according to the address indicated by the contents of the RAM address buffer register.

Specific addresses of data memory

The data memory is also used for the following specific purposes. When it is not used for the respective purposes, the RAM of the corresponding address can be used to store the user processing data.

(1) Stack (STACK)
(2) Stack pointer word (SPW)
(3) Data counter (DC)
(4) Timer/Counter (TC1, TC2)

(1) Stack (STACK)

The stack, which is contained in the data memory (one level of the stack consists of 4-word RAM), is area to save the contents of the program counter (return address) and flag prior to jumping to the processing program at time of subroutine call or interrupt acceptance. To return from the processing program, (RET) instruction is used to restore the contents saved in the stack to the program counter, and (RETI) instruction is used to restore the contents saved in the stack to the program counter and flags.

The location of the stack to save/restore the contents is determined by the stack pointer word, which is automatically decremented after the saving operation, and incremented prior to the restoring operation.

(2) Stack Pointer Word (SPW)

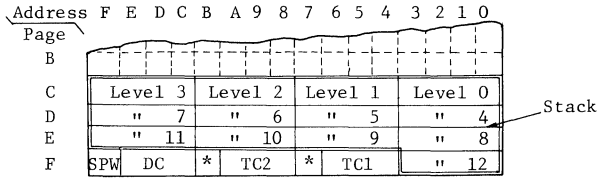
The address FF in the data memory is called a stack pointer word and decides the stack pointer. The stack is contained in the RAM, and accessed by the stack pointer.

The stack pointer is decided with the format shown in Fig. 1.4.1, but this address indicates the lower RAM address in each level of the stack.

Values "E" - "0" can be assigned for the stack pointer word, so that the maximum of 15 nesting levels are available for the stack. However, when the timer/counter mentioned following is used, the level containing the RAM address corresponding to the timer/counter cannot be used for the stack (value "F" is not assigned to the stack pointer word, because the stack contains the RAM address corresponding to the stack pointer word). The stack pointer word is automatically updated by the subroutine call or interrupt acceptance; however, it cannot exceed the allowable size of the stack for the system configuration.

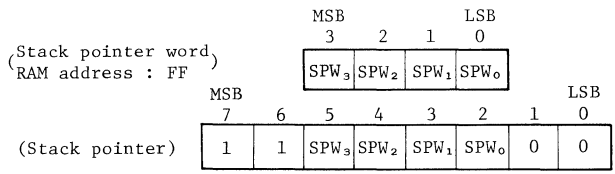
Since the stack pointer word is never initialized in terms of hardware, it is necessary to set it to the highest possible level of the stack in the user's initialization prpgram. For instance, it is set to "C" level when the two channels of timer/counter are used.

Note: The "level" indicates the depth of the nesting in the stack as well as the location of the next available stack. That is, it represents the contents of the stack pointer word.

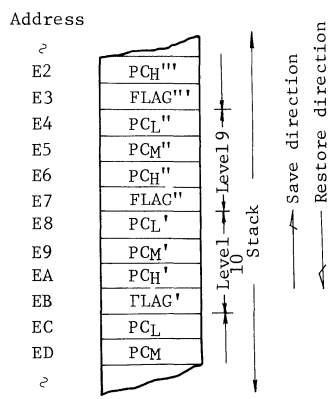


* : Can be used to store the user processing data

(a) Specific purposive map of RAM



(b) Stack pointer and stack pointer word



(c) Structure of stack

(3) Data Counter (DC)

Data counter is a 12-bit binary counter used to specify the address when the data table in the ROM area is referred (ROM data referring instruction).

The RAM address with 4-bit unit is allocated to the data counter, so that the initial value setting and the content reading of the data counter can be executed by the RAM manipulative instructions.

	MSB		LSB
	D		C
(Data Counter)	DC_H	DC_M	DC_L
(RAM Address)	(FE)	(FD)	(FC)

Fig. 1.4.2 Data Counter and RAM Address

(4) Timer/Counter (TC1, TC2)

The two channels of 12-bit timer/counter are built-in, and the RAM address with 4-bit unit is allocated to the timer/counter, so that the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulative instructions.

When the timer/counter 1 is not used, the stack lower from level 13 can be used. When both of the timer/counter 1 and 2 are not used, the stack lower from level 14 can be used.

	MSB		LSB
	TC1		
(Timer/Counter 1)	$TC1_H$	$TC1_M$	$TC1_L$
(RAM Address)	(F6)	(F5)	(F4)

	MSB		LSB
	TC2		
(Timer/Counter 2)	$TC2_H$	$TC2_M$	$TC2_L$
(RAM Address)	(FA)	(F9)	(F8)

Fig. 1.4.3 Timer/Counter and RAM Address

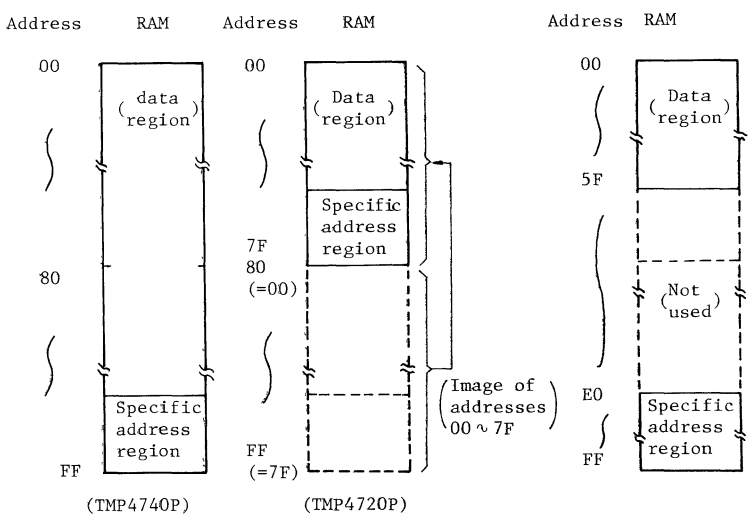
(5) Page 0 in Data Memory

Page 0 in the data memory (addresses 00 - 0F) is effectively used as a flag or pointer in a user's program.

RAM Capacity

Data memory contained in TMP4740P has a 256 x 4-bit (addresses 00 - FF) capacity, and that contained in TMP4720P has a 128 x 4-bit (addresses 00 - 7F) capacity.

Since the TMP4720P also has the RAM address buffer register of 8-bit length, there is no physical RAM in addresses 80 - FF in the TMP4720P. However, the RAM equivalent to addresses 00 - 7F are referred when addresses 80 - FF are accessed in a program, because the MSB of RAM address buffer register is not decoded. That is, the specific RAM address is distributed to C0 - FF in a program, but the RAM equivalent to addresses 40 - 7F are assigned in the TMP4720P.



(a) RAM Capacity and Address

(b) RAM Map example of TMP4720P (TC_1, TC_2 and stack 5 level are used.)

Fig. 1.4.4 RAM Capacity and Address

1.5 ALU, Accumulator (AC)

The ALU is a circuit used for various arithmetic and logical operation for 4-bit binary data. It performs the operation designated by the instruction, and outputs the 4-bit result, carry (C), and zero detection signal (Z).

The accumulator is a 4-bit register to use a source operand for the arithmetic operation, and in which the result is stored.

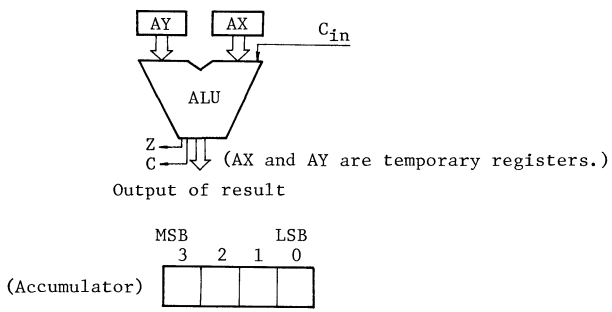


Fig. 1.5.1 ALU, Accumulator

Detection of operating condition

Output C from the ALU indicates the carry output from the most significant position in the addition operation.
However, the subtraction is executed with the addition of the 2's complement, so that output C in the subraction operation indicates the "non-borrow" from the most significant position (i.e., in case of non-borrow, C = "1"). Accordingly, borrow (B) can be represented with "$\overline{C}$".

Output Z indicates the zero detection signal to which "1" is applied when all of the 4-bit data transferred to accumulator or output of the ALU are cleared to zero.

Example (4-bit operation)

(a)	4 + 5 = 9	(C = 0, Z = 0)
(b)	7 + 9 = 0	(C = 1, Z = 1)
(c)	3 - 1 = 2	(B = 0, Z = 0)
(d)	2 - 2 = 0	(B = 0, Z = 1)
(e)	6 - 8 = -2 or E	(B = 1, Z = 0)

Note : B = $\overline{C}$ is indicated.

1.6 Flag (FLAG)

Flag is a 4-bit register used to store the condition of arithmetic operation, and of which the set/reset conditions are specified by the instruction. The flag consisting of CF, ZF, SF, and GF is saved in the stack when the interrupt is accepted. By executing the (RETI) instruction, it is restored from the stack to the conditions immediately before the interrupt is accepted.

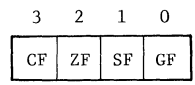


Fig. 1.6.1 Flag

(1) Carry Flag (CF)

This flag is used to hold the carry in the addition operation as an input to the ALU by the (ADDC A, @HL) instruction as well as to hold the non-borrow in the subtraction operation (the carry in the addition of the 2's complement) as an input to the ALU by the (SUBRC A, @HL) instruction. The rotate instruction makes the flag hold the data shifted out of the accumulator.

(2) Zero Flag (ZF)

This flag is stored the zero detection signal (Z) when the instruction designate to change. "1" is set if all 4 bits are cleared to zero by an arithmetic operation or data processing.

(3) Status Flag (SF)

This flag is set or reset according to the condition specified by the instruction. With the exception of particular cases, it is usually presented at every execution of an instruction, and holds the contents of the result during execution of the next instruction. It is normally set to "1", but is reset to "0" for a time under the certain condition (it varies according to the instruction, for examples, when the result is zero, when carry occurs in the addition, or when borrow occurs in the subtraction, the flag is reset).

The status flag is referred to as branch condition in a branch instruction. The memory location is branched when this flag is set to "1"; therefore, normally the branch instruction can be required as "unconditional jump instruction". On the contrary, the instruction becomes a "conditional instruction" if it is executed immediately after loading the instruction to set/reset the status flag according to the condition determined by some previous instruction.

The status flag is initialized to "1" at initialization, and is also set to "1" after the contents have been saved in the stack when the interrupt is accepted. The contents saved in the stack is restored by the (RETI) instruction.

(4) General Flag (GF)

This is a single-bit general purpose flag, being set or reset, and also used in a test by a program. This can be used for any purpose in the user program.

1.7 Port (PORT)

Data transfer to/from the external circuitry, and command/status/data transfer between the built-in periferal circuitry are carried out by the input/output instructions.

(a) Input/Output port : Data transfer to/from external circuitry.

(b) Command/data output : Control of circuitry of built-in peripheral circuitry, and output of data.

(c) Status/data input : Input of status signal(Note) and data from the built-in peripheral circuitry.

Note : Status signal is provided from serial port and is different from the status flag (SF).

To transfer the data or to control the circuitry, each port or register is selected by designating the address (Port address) by input/output operational instructions (13 instructions) in the same way as the memory.
The port address is composed of 5 bits (addresses 0 - 31).
The address to be accessed differs according to a instruction.
By way of caution, the port address space is independent of the program memory address space and the data memory address space.

Every output port contains a latch in order to hold the output data. Since every input port is operated without latching, it is desired to externally hold the data to be input from the external devices till the data is completely read out, or to read the data several times to confirm the contents.

The details to specify the input/output circuit format of ports and initialization of the output latch are 3.6 (2) Input/Output Circuit Format.

Port address	Symbol (Input/ Output)	Port, Register (Input/Output)	Input/Output Instructions						
			IN %P, A IN %P,@HL	OUT A ,%P OUT@HL,%P	OUT#K,%P	OUTB @HL	SET%P,b CLR%P,b	TEST %P,b TESTP%P,b	SET @L CLR @L TEST @L
00	IP00/OP00	K_0 Input port / P_1 Output	0					0	
01	IP01/OP01	P_1 Output latch/ port	0	0	0		0	0	
02	IP02/OP02	P_2 " /P_2 "	0	0	0		0	0	
03	IP03/OP03	———							
04	IP04/OP04	R_4 I/O port	0	0	0		0	0	0
05	IP05/OP05	R_5 "	0	0	0		0	0	0
06	IP06/OP06	R_6 "	0	0	0		0	0	0
07	IP07/OP07	R_7 "	0	0	0		0	0	0
08	IP08/OP08	R_8 "	0	0	0		0	0	
09	IP09/OP09	R_9 "	0	0	0		0	0	
0A	IP0A/OP0A	———							
0B	IP0B/OP0B	———	(*) Serial buffer register (Reception)						
0C	IP0C/OP0C	———	(**) Serial buffer register (Transmission)						
0D	IP0D/OP0D	———							
0E	IP0E/OP0E	Status input/ ———	0					0	
0F	IP0F/OP0F	(*) / (**)	0	0	0				
10	/OP10	/ ———							
11	/OP11	$/P_2 \cdot P_1$ output port (8-bit output)				0			
12	/OP12	/ ———							
13	/OP13	/ ———							
14	/OP14	/ ———							
15	/OP15	/ ———							
16	/OP16	/ ———							
17	/OP17	/ ———	(a) Control with timer interrupt of divider						
18	/OP18	/ ———	(b) Timer/Counter 1 control						
19	/OP19	/ (a)		0					
1A	/OP1A	/ ———	(c) Timer/Counter 2 control						
1B	/OP1B	/ ———	(d) Serial port control						
1C	/OP1C	/ (b)		0					
1D	/OP1D	/ (c)		0					
1E	/OP1E	/ ———							
1F	/OP1F	/ (d)		0					

Note 1: Inputs (IP10 - IP1F) of port addresses 10 - 1F remain undefined.
Note 2: Port addresses with "——" mark are reserved addresses and cannot be used at user's program.
Note 3: OP11 is automatically accessed by (OUTB @HL) instruction, but cannot be done by the instructions other than this one.

Table 1.7.1 Port Address Allocation and Input/Output Instructions

(1) K_0 ($K_{03} \sim K_{00}$) Port

This is a 4-bit port used for input.

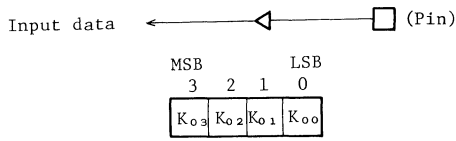


Fig. 1.7.1 K_0 Port

(2) P_1 ($P_{13} \sim P_{10}$), P_2 ($P_{23} \sim P_{20}$) Port

These ports are 4-bit ports with a latch used for output. The latch data can be read by the instruction.

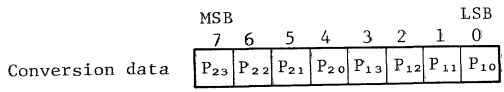
These two ports can independently access by specifying port addresses IP01/OP01, and IP02/OP02. In addition, they can output 8-bit data by the (OUTB @HL) instruction.

PLA data conversion

A hardware PLA is not contained in the system; however, the function equivalent to it can be performed by access to the PLA data conversion table provided in the ROM by use of the (OUTB @HL) instruction.

The PLA referring instruction (OUTB @HL) : This instruction reads out the 8-bit data stored in the program memory, whose address is determined by the contents of the data memory indicated by the contents of the H and L registers as well as the contents of the carry flag, and outputs the data to 8-bit ports P2 and P1. At this time OP11 is automatically selected as the port address.

Ports P1 and P2 are capable of reading the latch data by the instruction, so that the data output by the PLA referring instruction can be qualified or modified; that is, the convert pattern can be changed or the numbers of pattern will be increased.



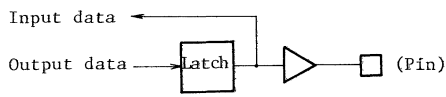


Fig. 1.7.2 P_1 and P_2 Ports

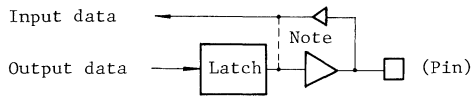
(3) $R_4(R_{43} \sim R_{40})$, $R_5(R_{53} \sim R_{50})$, $R_6(R_{63} \sim R_{60})$, $R_7(R_{73} \sim R_{70})$ Port

Each of these ports is a 4-bit I/O port with a latch. The latch should be set to "1" when the port is used as an input port.

Pins $R_{73} - R_{40}$ can be used for bit scanning for set/reset and test according to the contents of the L register by executing the (SET @L), (CLR @L) and (TEST @L) instructions. Table 1.7.2 shows the pins corresponding to the contents of the L register.

L register 3 2 1 0	Corresponding Pin	L register 3 2 1 0	Corresponding Pin
0 0 0 0	R_{40}	1 0 0 0	R_{60}
0 0 0 1	R_{41}	1 0 0 1	R_{61}
0 0 1 0	R_{42}	1 0 1 0	R_{62}
0 0 1 1	R_{43}	1 0 1 1	R_{63}
0 1 0 0	R_{50}	1 1 0 0	R_{70}
0 1 0 1	R_{51}	1 1 0 1	R_{71}
0 1 1 0	R_{52}	1 1 1 0	R_{72}
0 1 1 1	R_{53}	1 1 1 1	R_{73}

Table 1.7.2 Correspondence of Individual Bits of L Register and I/O Port



Note : For bit set/reset of port,
latch output serves as input data.

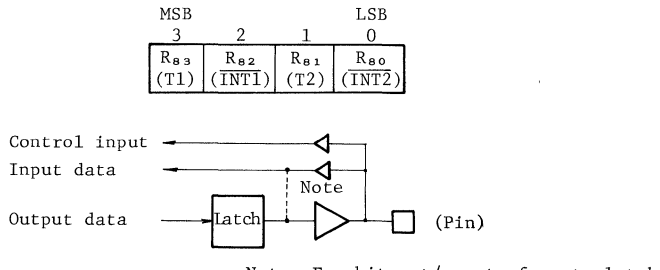
Fig. 1.7.3 $R_4 \sim R_7$ Ports

(4) R_8 ($R_{83} \sim R_{80}$) Port

This is a 4-bit I/O port with a latch. The latch should be set to "1" when the port is used as an input port.

It is a port common to external interrupt input or external timer/counter input. When it is driven by the external circuitry, such as external interrupt input or external timer/counter input, the latch must be set to "1". When it is used as normal I/O port, some measures, such as inhibition of the external interrupt input acceptance or disable of the mode depending on the external input of the timer/counter should be taken in a program.

(Note) When pin R_{82} ($\overline{INT1}$) is used as a port, INT1 interrupt request takes place because the falling edge of the pin input/output is detected (interrupt enabling master F/F is normally set to "1"). This causes the CPU to process a dummy interrupt acceptance [e.g. the (RETI) instruction only is executed]. When pin R_{80} ($\overline{INT2}$) is used, INT2 interrupt request also takes place in the same manner as the case of pin R_{82}, but the interrupt request is not accepted by merely resetting the LSB (EIR_0) of the enable interrupt register to "0" in advance. Therefore, the above processing is not required.

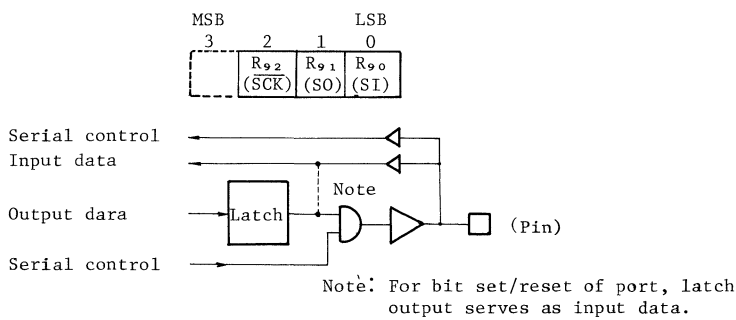


Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.4 R_8 Port

(5) R9($R_{92} \sim R_{90}$) Port

This is a 3-bit I/O port with a latch, and the latch must be set to "1" when it is used as input port.
The R9 port is also used as serial port. The latch must be set to "1" when R9 port is used as serial port. The port used as normal I/O port is not entirely influenced by disabling the serial port.
Pin R93 is not mounted in the port, but "1" is read by accessing to pin R93 in a program.



Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.5 R_9 Port

1.8 Interrupt control circuit (INTR)

Interrupt factors are composed of two from the external circuitry, and four from the internal circuitry. By setting the interrupt latch provided for each factor, an interrupt request is generated to the CPU. The interrupt latch is set when the edge of the input signal is detected.

The interrupt request is not always accepted by the CPU if generated. It is not accepted till the priority in the six factors determined according to the hardware and the enabling/disabling control by the program become all affirmative.

In order to control enabling/disabling of interrupt by the program, an F/F (EIF) and a 4-bit register (EIR) are provided. By using these means, preferential acceptance of the interrupt factors by the program, and multiple interrupt control can be realized.

Factor			Priority according to hardware	Interrupt Latch	Enable condition according to program	Vector Address
External interrupt 1 (INT1)			(Higher) 1	$INTL_5$	(Note 1) EIF = 1	002
Internal interrupt	Serial Input/Output interrupt	(ISIO)	2	$INTL_4$	EIF·EIR_3 = 1	004
	Timer counter 1 Overflow interrupt	(IOVF1)	3	$INTL_3$	EIF·EIR_2 = 1	006
	Timer counter 2 Overflow interrupt	(IOVF2)	4	$INTL_2$	(Note 2) EIF·EIR_1 = 1	008
	Timer interrupt of divider	(ITMR)	5	$INTL_1$	(Note 2) EIF·EIR_1 = 1	00A
External interrupt 2 (INT2)			6 (Lower)	$INTL_0$	EIF·EIR_0 = 1	00C

Interrupt enabling master F/F

EIF

Interrupt enabling register (EIR)

MSB 3	2	1	LSB 0
EIR_3	EIR_2	EIR_1	EIR_0

(Note 1) Since EIR register cannot make disabling of the INT1 interrupt, this interrupt is always accepted under the interrupt enabled condition (EIF = 1). Therefore, this should be used for the interrupt requiring the first priority such as emregency interrupt.

(Note 2) The given acceptance condition by the program is the same in IOVF2 and ITMR; accordingly, the action of these interrupts to the acceptance/inhibition control is the same.

Table 1.8.1 Interrupt Factors

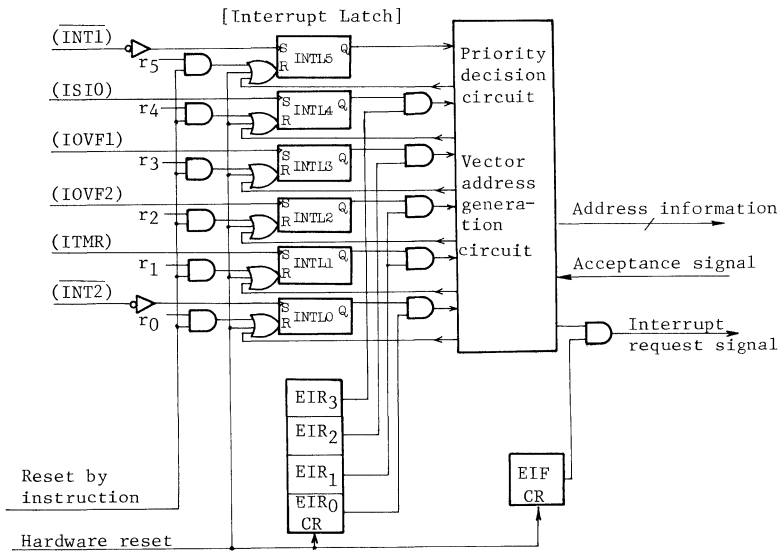
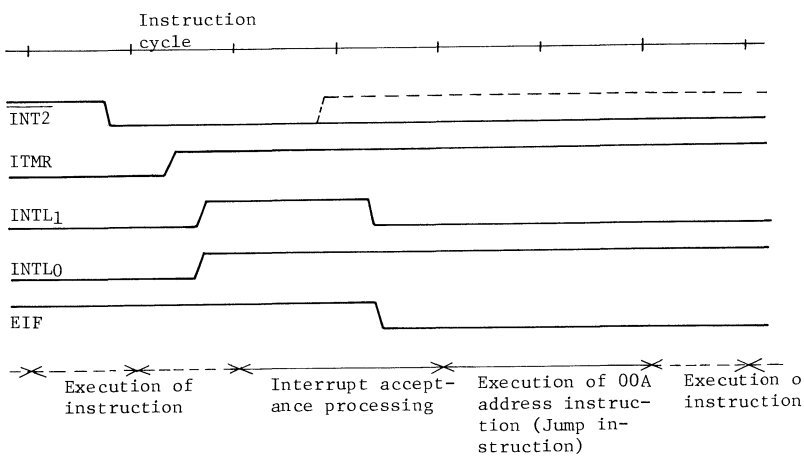


Fig. 1.8.1 Interrupt Control Circuit



Note: On the assumption that EIR_1 = 1, without other interrupt requests

Fig. 1.8.2 Interrupt Acceptance Timing Chart (Example)

(1) Interrupt processing

The interrupt request signal to be sent to the CPU is held by the interrupt latch till the request is accpeted or the latch is reset by the initialization operation or instruction.

The processing for the interrupt acceptance is performed within two instruction cycle time after the completion of the execution of instruction (after the completion of the timer/counter processing if it is required).

The following operations are performed by the interrupt service program.

① The contents of the program counter and flag are saved in the stack.

② The vector address is set to the program counter according to the interrupt factor.
(A jump instruction to each interrupt service program is usually stored in the program memory corresponding to the vector address.)

③ The status flag is set to "1".

④ The interrupt enabling master F/F is reset to "0" to inhibit the subsequent interrupt acceptance for a time.

⑤ The interrupt latch of the accepted interrupt factor is reset to "0".

⑥ The instruction stored in the vector address is executed.

The interrupt service program terminates after the execution of the (RETI) instruction.

The following operations are performed by the (RETI) instruction.

① The contents of the program counter and flag are restored out of the stack.

② The interrupt enabling master F/F is set to "1".

When the multiple interrupt is accepted, the interrupt enabling master F/F should be set by the instruction. At this time, the enabling/disabling for each interrupt factor can be changed by updating the interrupt enabling register by the (XCH A, EIR) instruction.

The program counter and flag are automatically saved/restored in the interrupt processing. However, if saving/restoring of the accumulator and other registers is necessary, it should be designated by a program.

(2) Interrupt control by program

EIF

This is an enabling interrupt master F/F. Interrupt is put in the interrupt acceptance enabling state by setting the EIF to "1". It is reset to "0" immediately after having accepted an interrupt to inhibit the subsequent interrupt acceptance for a time, but is set to "1" again by the (RETI) instruction after the completion of the interrupt service program to return the enable state again. And then the other interrupt can be received.

The EIF can be set/reset in a program by using the (EICLR IL,r) and (DICLR IL, r) instructions. It is reset to "0" at initialization operation.

EIR register

This is a 4-bit register used for selection/control of enabling/disabling of the interrupt acceptance in a program.

Read/write operation is performed by use of the (XCH A, EIR) instruction. It is set to "0" at the initialization operation.

Interrupt latch

The interrupt latches ($INTL_5$ - $INTL_0$) provided for each interrupt factor are set by the rising edge of the input signal if the interrupt is caused by the internal factors, and are set by the falling edge of the input pin if it is caused by the external factors. Then, interrupt request signal is sent to the CPU. The interrupt latch holds the signal till the interrupt request is accepted, and is reset to "0" immediately after the interrupt has been accepted.

Since the interrupt latch can be reset to "0" by the (EICLR IL, r), (DICLR IL, r) and (CLR IL, r) instructions, the interrupt request signal can be initialized by a program. The latch is reset to "0" at the initialization operation.

1.9 Frequency divider (FD)

The divider (FD_1 - FD_{18}) is made up 18-stage binary counter, and its output is used to generate various internal timing.

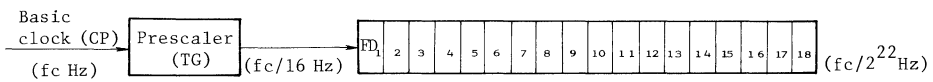
The basic clock (fc Hz) is divided into sixteen by the timing generator and input to the divider; therefore, the output frequency at the last stage is $fc/2^{22}$ Hz.
It is reset to "0" at the initialization operation.

Timer Interrupt of divider (ITMR)

The divider is capable of sending the interrupt request for a certain frequency. Four different frequencies can be selected for timer interrupt by instructions.

The command register is accessed as port address OP19, and is reset to "0" at time of the initialization.

The timer interrupt of divider is caused from the rising edge of the first output of the divider after the data has been written in the command resister.



(a) Structure of frequency divider

(Port address OP19)

MSB 3	2	1	LSB 0	
				(*: don't care)

*	0	*	*	:	Disable	
*	1	0	0	:	Interrupt frequency	$fc/2^{10}$Hz
*	1	0	1	:	"	$fc/2^{11}$Hz
*	1	1	0	:	"	$fc/2^{12}$Hz
*	1	1	1	:	"	$fc/2^{13}$Hz

Interrupt frequency (Hz)	For example, fc=4.194304MHz
$fc/2^{10}$	4,096 Hz
$fc/2^{11}$	2,048 Hz
$fc/2^{12}$	1,024 Hz
$fc/2^{13}$	512 Hz

(b) Command register

Fig. 1.9.1 Frequency Divider

1.10 Timer/Counter (TC_1, TC_2)

Two channels of 12-bit binary counter is contained to count time or event.

Since the RAM address with 4-bit unit is allocated to the timer/counter, the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulated instructions.

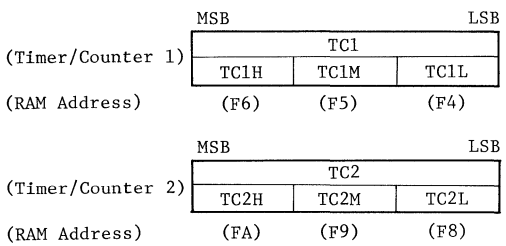
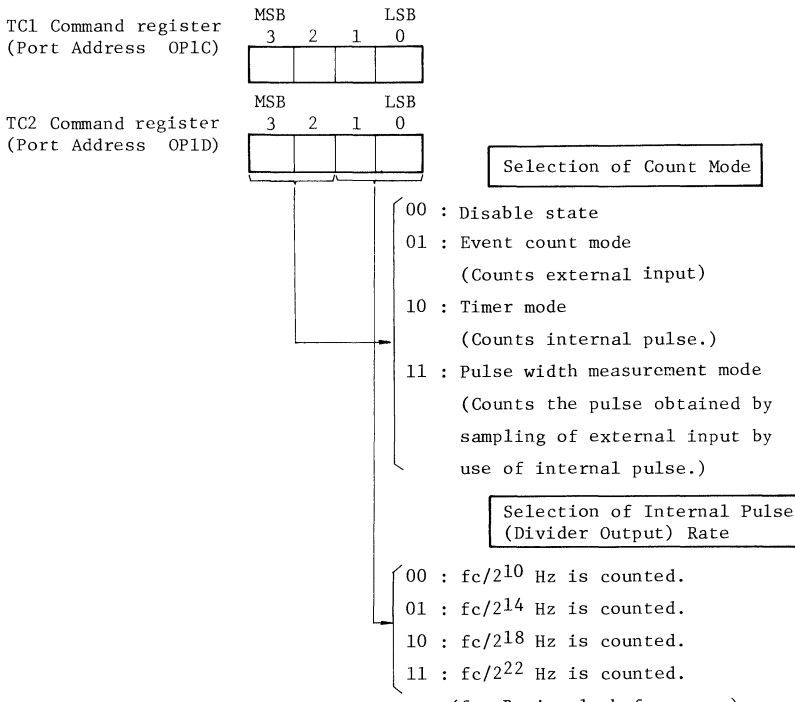


Fig. 1.10.1 Timer/Counter

(1) Timer/Counter Control

The timer/counter is controlled by the command specifying the operation mode. The command register for the timer/counter 1 and timer/counter 2 is accessed as port addresses OP1C and OP1D, respectively. It is reset to "0" at the initialization operation. The count operation is started from the first rising edge of the count pulse applied by setting the value (mode) to the command register.

When the timer/counter is not used, the RAM addresses corresponding to the timer/counter can be used to store the user processing data by selecting the "disable" state. In the timer mode, the external input pins can be used as I/O ports [R_{83} (T1), R_{81} (T2)].



(a) Command register

Internal Pulse Rate (Hz)	Max. Setting Time (SEC)	For example, fc=4.194304 MHz	
		Internal Pulse Rate (Hz)	Max. Setting Time (SEC)
$fc/2^{10}$	$2^{22}/fc$	4,096	1
$fc/2^{14}$	$2^{26}/fc$	256	16
$fc/2^{18}$	$2^{30}/fc$	16	256
$fc/2^{22}$	$2^{34}/fc$	1	4,096

(b) Selection of timer rate

Fig. 1.10.2 Control of Timer/Counter

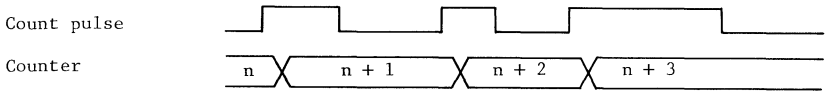
(2) Count Operation

When the rising edge of the count pulse is detected, the count latch is set to send a count request to the CPU.

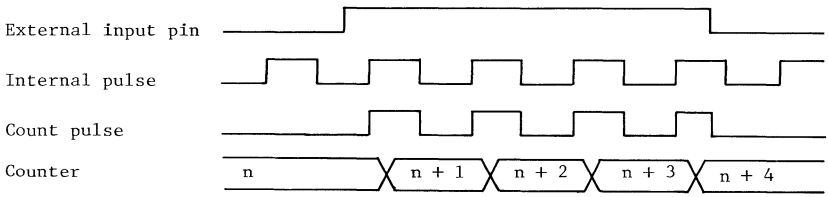
The count operation of the timer/counter is performed requiring one instruction cycle time after completion of the instruction execution. The execution of the next instruction and the acceptance of the interrupt request are kept waiting during the operation. When the count request is sent from the timer/counter 1 and 2, at the same time, the count request of the timer/counter 1 is preferentially executed.

The maximum frequency applied to the external input pin under the event counter mode is fc/32 Hz if one channel is used. When two channels are used, fc/32 Hz is applied to the timer/counter 1, and fc/40 Hz to the timer/counter 2.

In the timer mode, the maximum frequency is determined by a command.
The maximum frequency applied to the external input pin in the pulse width measurement mode should be the frequency level available for analyzing the count value in the program. Normally, the frequency sufficiently slower than the designated internal pulse rate is applied to the external input pin.



(a) Event counter/timer mode



(b) Pulse width measurement mode

Fig. 1.10.3 Mode and Count Value of Timer/Counter

Decrease in execution speed of instruction due to count operation

The CPU carries out the count operation requiring one instruction cycle time for the count request. Therefore, this causes the decrease in the apparent speed of instruction execution. Some examples are shown below :

(a) In the timer mode with count pulse rate of $fc/2^{10}$ Hz :

The count operation is inserted once every 128-instruction cycle time, so that the apparent speed is decreased by $1/127 \fallingdotseq 0.8\%$ instruction execution speed. For example, the apparent speed is 2.016μs to 2μs instruction execution speed.

(b) In the event count mode :

It depends on the count pulse rate applied to the external input pin. In the worst case, when the timer/counter 1 and 2 are operated at the same time with the maximum count pulse rate, the count operation is inserted once every 4-instruction cycle time for the timer/counter 1, and once every 5-instruction cycle time for the timer/counter 2.

The apparent speed of the instruction execution, therefore, decreases by 9/11 ≒ 82%. The apparent speed is 3.64μs to 2μs instruction execution speed.

(3) Interrupt by overflow (IOVF1, IOVF2)

At the time when the overflow occurs, the timer/counter generates the interrupt request.
That is, the interrupt request is generated when the count value of FFF is changed to 000. The counting is continued after the interrupt request signal is generated. Assuming that the CPU provides the interrupt enabling state, and that the interrupt is accepted as soon as the overflow interrupt has been generated, the interrupt processing can be performed in the sequence illustrated in Fig. 1.10.4.

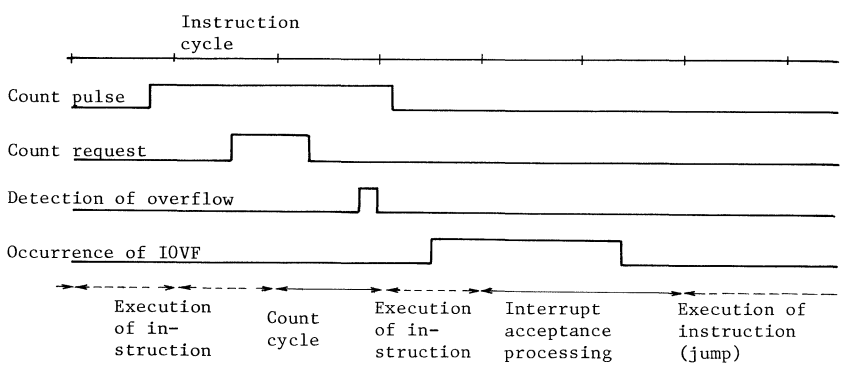


Fig. 1.10.4 Timing Chart of Timer/Counter in Interrupt by Overflow

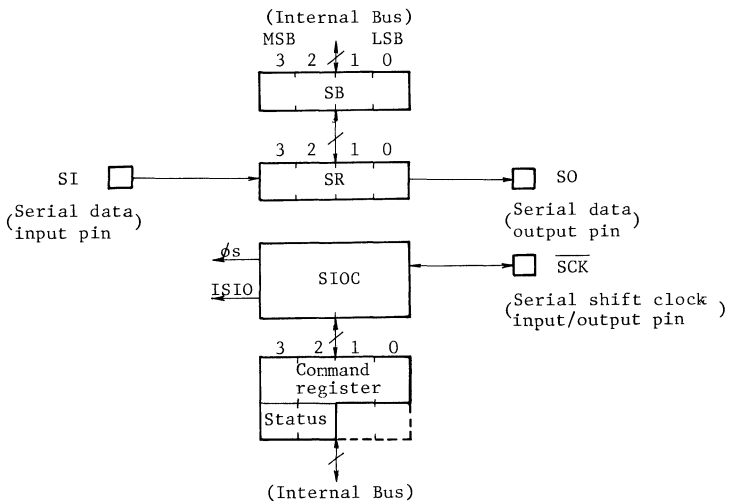
1.11 Serial Port (SIO)

A 4-bit serial port with a buffer is provided to transfer the serial data from/to the external circuitry. It is connected to the external circuitry through three pins [$R92$ ($\overline{SCK}$), $R91$ (SO), $R90$ (SI)]. Since these pins are also used as port R9, the output latch of the R9 port should be set to "1" when the serial port is used. When it is not used, the pins can be used as I/O port R9.

Pin $R90$ in the transmit mode and pin $R91$ in the receive mode are also available as I/O port pin.

(1) Circuit configuration

The serial port consists of a 4-bit shift register, a 4-bit buffer register, and its control circuit.



SR : 4-bit shift register
SB : 4-bit buffer register
SIOC : Serial port control circuit
φs : Internal shift clock
ISIO : Interrupt request

Fig. 1.11.1 Circuit Configuration of Serial Port

(2) Serial port control

The serial port operation is controlled by the command. The command register is accessed with port address OP1F, and reset to "0" at the initialization operation. The operation status can be informed through the status input, which is accessed with port address IPOE.

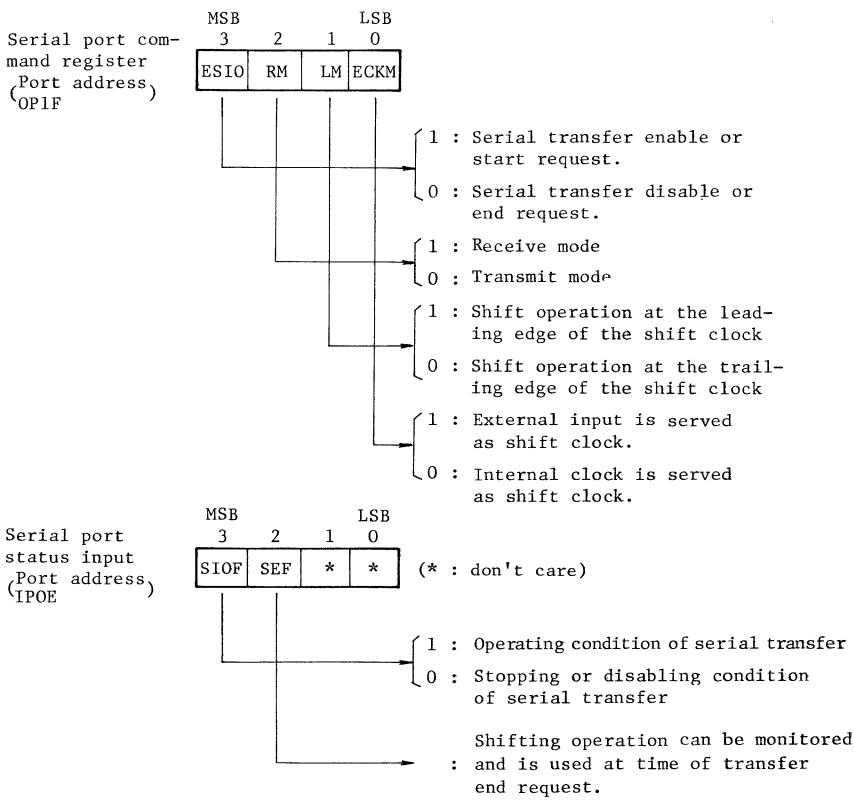


Fig. 1.11.2 Command Register, Status Input

(3) Shift clock (SCK)

The following shift clock modes can be selected by the contents of the command register.

(a) Clock source (External/internal mode)
(b) Shift edge of clock (Leading edge/trailing edge mode)

Internal clock mode

$fc/2^7$ Hz is used for the shift clock (when the basic clock frequency fc is 4.194304 MHz, the shift clock frequency is 32.768 kHz.). At this time, the clock is supplied to the external devices through the $\overline{SCK}$ pin. If the data setting (transmit mode) or the data reading (receive mode) rate by the program cannot follow the clock rate, the shift clock is automatically stopped and the next shift operation is suspended until the data processing is completed ("Wait"operation).

External clock mode

The shift operation is performed by the clock provided from the external circuitry since the $\overline{SCK}$ pin serves as an input.

Leading edge shift mode

Data is transmitted (transmit mode) or received (receive mode) at the leading edge of the $\overline{SCK}$ pin signal.

Trailing edge shift mode

Data is received (receive mode) at the trailing edge of the $\overline{SCK}$ pin signal.

The $\overline{SCK}$ pin must be set to the "high" level when the serial transfer is started. In the internal clock mode, the $\overline{SCK}$ pin is automatically set to the "high" level because it serves as an output.

(4) Operation mode

Selection of the following three transfer modes is available by changing the combination of the RM bit and LM bit of the command register.

RM (Bit 2)	LM (Bit 1)	ECKM (Bit 0)	Operation Mode
0	0	1/0	Can not be used
0	1	1/0	Transmit mode (Note) (External/Internal clock)
1	0	1/0	Receive(Trailing edge shift) mode (External/Internal clock)
1	1	1/0	Receive(Leading edge shift) mode (External/Internal clock)

(Note) Leading edge shift operation is performed.

Table 1.11.1 Operation Mode of Serial Port

In the transmit mode, the 4-bit data written to the buffer register from the CPU is shifted out by the shift register, and is output in the SO pin from the data of the LSB in sequence. The buffer register is accessed as the port address OPOF.

In the receive mode, the data to be input to the SI pin is shifted toward the LSB by the shift register in sequence, and is set in the buffer register after the 4-bit data has been received.

The CPU reads the contents of the buffer register, which is accessed as the port address IPOF.

Transmit mode

After this mode is set in the command register, the first transmit data (4-bit) is written in the buffer register (the data cannot be written in the buffer register, if the transmit mode is not set). Then the data can be transmitted by setting the ESIO (MSB of command register) to "1". The content of the buffer register is transferred to the shift register by the first shift clock, and the data in the LSB (D_0) is output to the SO pin. The buffer register then becomes empty, so that the interrupt (ISIO) requesting the next data takes place (buffer empty). After that, the remaining data (D_1 - D_3) is automatically shifted out by the shift register by one data at a shift clock. The control by use of a program is not necessary in this operation.

Data is written in the buffer register by outputting the next transmit data (4-bit) to the port address OPOF in the interrupt service program, and at the same time the interrupt request is reset to "0".

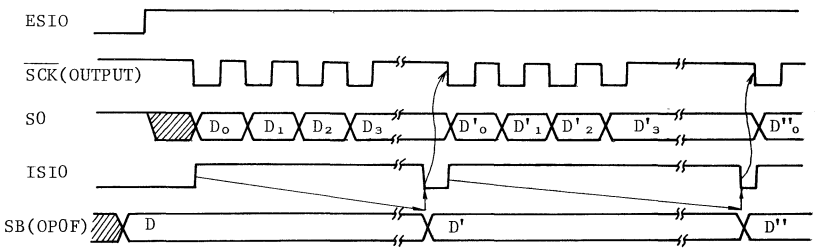
Internal clock operation

In case of $fc/2^7$ Hz internal clock operation, if the next data is not set in the buffer register (OPOF has not been accessed by the program) though the 4-bit data has been entirely shifted out, the shift clock automatically stops, and the wait operation is taking place until the data is set.

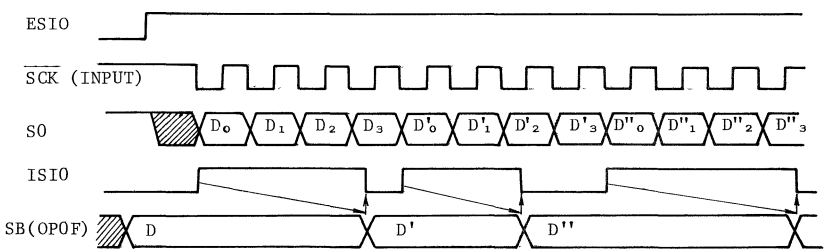
The maximum transmission rate is 31250 bit/sec. at the 4 MHz basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock provided from the external circuitry, the data should have been written in the buffer register before the next 4-bit data is shifted out. Therefore, the transfer rate is determined by the maximum time lag from the receipt of interrupt request (ISIO) to the writing of data in the buffer register by the interrupt service program.



(a) Internal clock operation (with wait operation)



(b) External clock operation

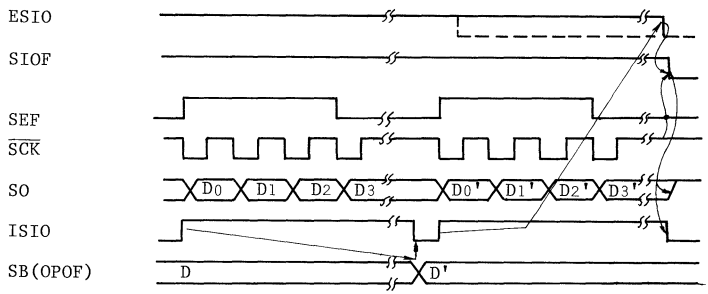
Fig. 1.11.3 Transmit Mode

Completion of transmission

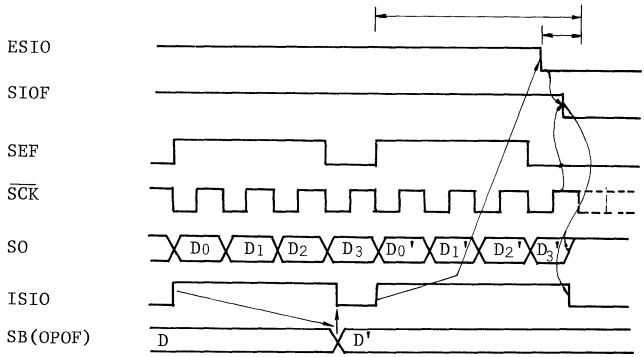
When the buffer register becomes empty, the interrupt occurs to request the next data. In case where the transmission is desired to be completed after the data is entirely transferred, the transmit operation can be stopped upon completion of transferring the current data shifted out, by resetting the ESIO to "0" without outputting the data. Whether or not the transfer operation is completed can be sensed in a program by the SIOF (MSB of the status input).

In the external clock operation, the ESIO must be reset to "0" before the next data is shifted out as in the data updating operation (however, the data is not updated when the operation is completed). When the wait operation have been already performed in the internal clock operation, the data transfer is terminated immediately after ESIO = 0.

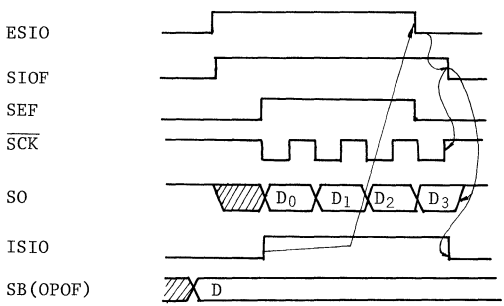
One word transfer can be terminated by ESIO = 0 in the interrupt service program on receipt of the interrupt caused by the buffer empty.



(a) Internal clock operation (with wait operation)



(b) External clock operation



(c) Completion at one-word transfer

Fig. 1.11.4 Completion of Transmission

Receive (trailing edge shift) mode

Data can be received by setting the receive mode in the command register as well as by setting the ESIO (MSB of command register) to "1". When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, interrupt (ISIO) takes place to request the data reading (buffer full). Since the shift register has been transferring the data to the buffer register, the shift operation is continued without waiting for the data being read.

When the data received from the port address IPOF is read in the interrupt service program, the interrupt request is reset. And then the next 4-bit data is transferred from the shift register to the buffer register if the buffer register has been full.

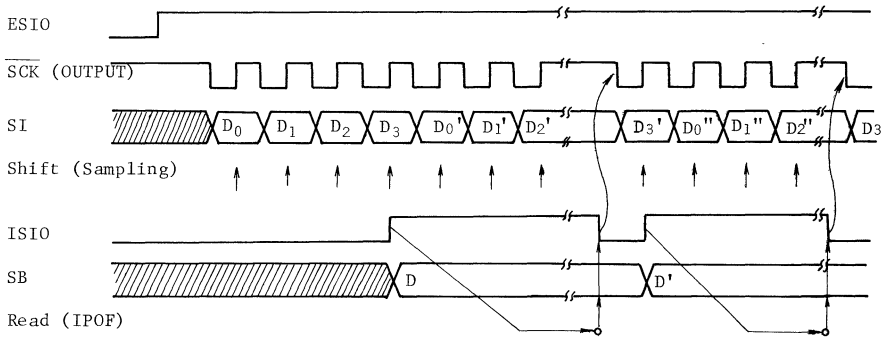
Internal clock operation

During the operation of the internal clock of $fc/2^7$Hz, if the next 4-bit data is not read out of the buffer register (the IPOF has not been accessed) in the program though the 4-bit data has been entirely input, the shift clock automatically stops, and the wait operation is taking place until the data is read out.

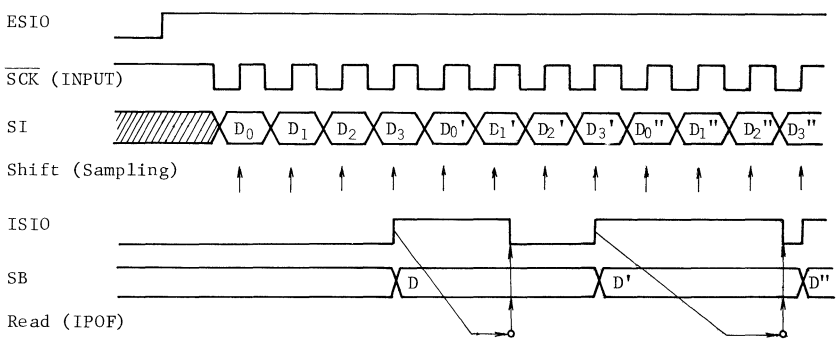
The maximum receiving rate is 31250 bit/sec at the 4 MHz basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock provided from the external circuitry, the current data should have been read by the instruction before the next 4-bit data is transferred to the buffer register. The transfer rate is, therefore, determined by the maximum time lag from the receipt of interrupt request (ISIO) to the read of the data in the buffer register by the interrupt service program.



(a) Internal clock operation (with wait operation)



(b) External clock operation

Fig. 1.11.5 Receive (trailing edge shift) Mode

Completion of receiving

When all of the data are read, the receiving of data can be completed upon termination of the current data transfer, by resetting the ESIO to "0".

Whether or not the data transmission is terminated can be sensed in a program by the SIOF (MSB of status input).

To complete the receive operation when the synchronization is desired between the serial transfer and interrupt service program (indicates data reading or completion of receiving), there are two ways according to the speed of shift clock.

The receive/transmit mode must be maintained without switching the mode until the last data is read out even if the completion of the data transfer is indicated; otherwise the contents of the buffer register will be lost.

(a) Sufficiently slow data transfer rate (external clock operation)

If the timing, operated by the external clock, is slow enough to reset the ESIO to "0" prior to the generation of the next shift clock, the ESIO can be reset to "0" in the interrupt service program which is loaded to read out the last data. Thereafter the last data is read.

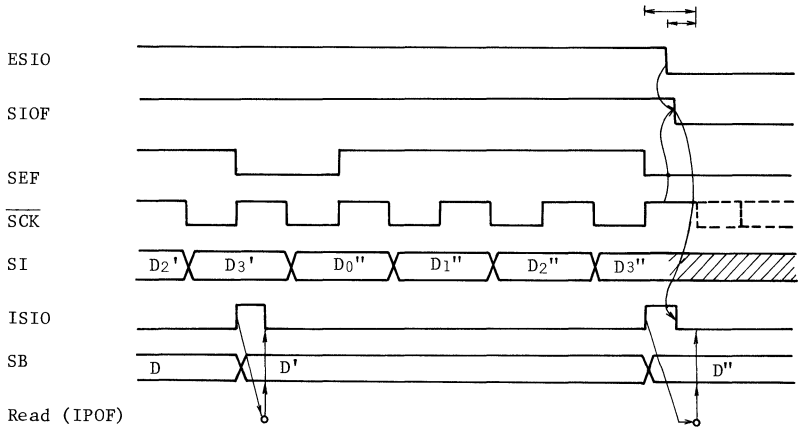


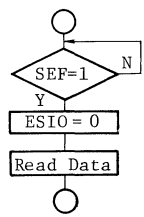
Fig. 1.11.6 Completion of Receiving (at slow transfer rate)

(b) Fast transfer rate

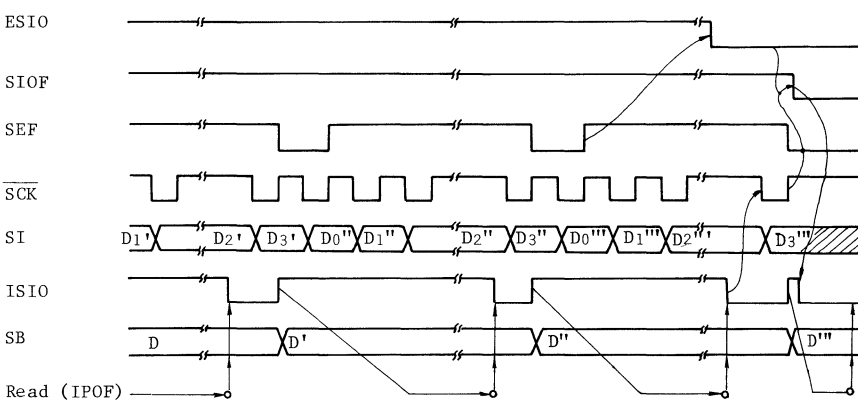
If the shift operation for the next data may start before the current data is read out by receipt of the interrupt request becuase the transfer rate is too fast, the interrupt service program which is loaded to read out the last data but one should be used to reset the ESIO to "0" after confirming that the SEF (bit 2 of status input) has been set to "1".

Thereafter, the data should be read. No operation is required to complete the data transfer in the interrupt service program for reading the last data.

The method mentioned above is usually taken for the internal clock operation. In the external clock operation, however, the reset of the ESIO and the read of data must be completed before the last data is transferred to the buffer register.



(a) Program sequence of receive end indication

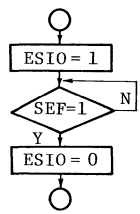


(b) Timing Chart
(in case of internal clock operation with wait operation)

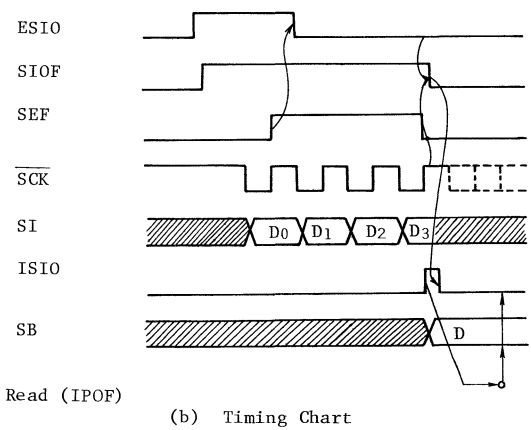
Fig. 1.11.7 Completion of Receiving (at fast transfer rate)

(c) One word transfer

The data receive operation starts after the ESIO is set to "1". Then, the ESIO is reset to "0" after confirming that the SEF status is set to "1". In this sequence, one interrupt casued by the buffer full takes place; therefore, the data should be read out by the service program.



(a) Program sequence of receiving start/end indication



(b) Timing Chart

Fig. 1.11.8 Receiving Start/Completion (at one word transfer)

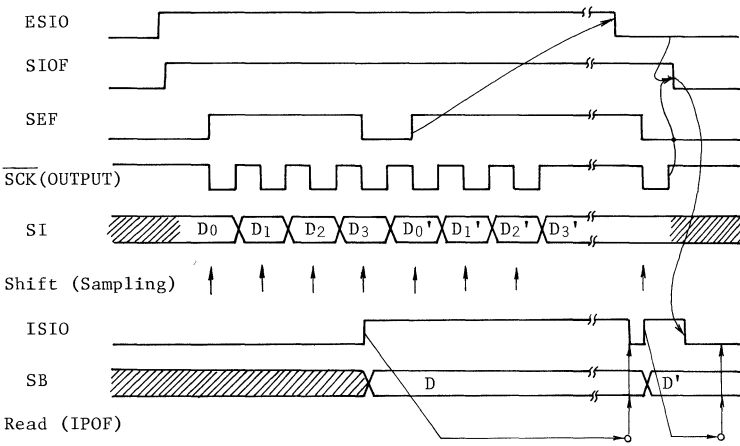
Receive (leading edge shift) mode

With this mode set in the command register, the data can be received by setting the ESIO (MSB of command register) to "1".

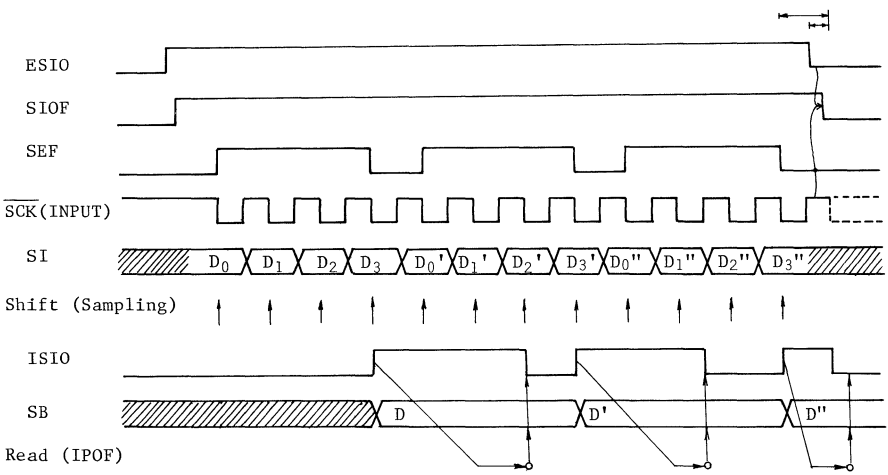
When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, the interrupt (ISIO) occurs to request the data reading (buffer full). Since the shift register is transferring the data to the buffer register, the shift operation has been continued without waiting for the data being read.

When the data received from the port address IPOF is read in the interrupt service program, the interrupt request is reset. And then the next 4-bit data is transferred from the shift register to the buffer register if the buffer register has been full.

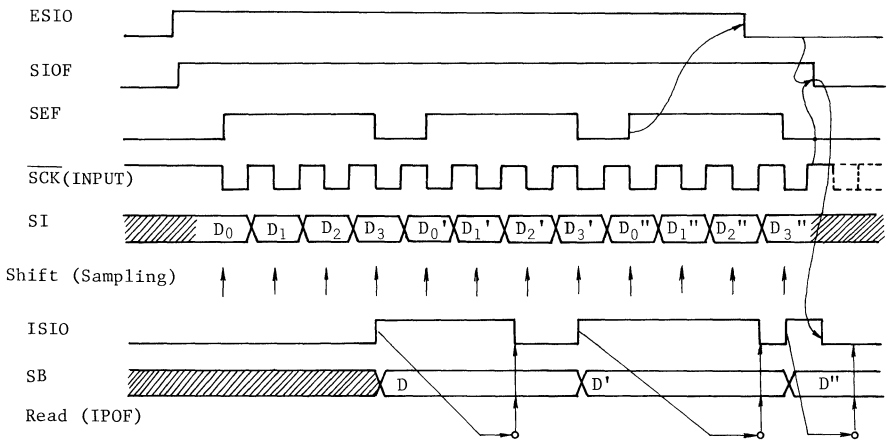
The basic operation in the receive (leading edge shift) mode is equivalent to that in the receive (trailing edge shift) mode except that the edge for the shift clock is different, and that at time of the transfer start, the first shifted data has been already input from the external circuitry before the first shift clock is applied to the data receipt. Timing charts are shown below.



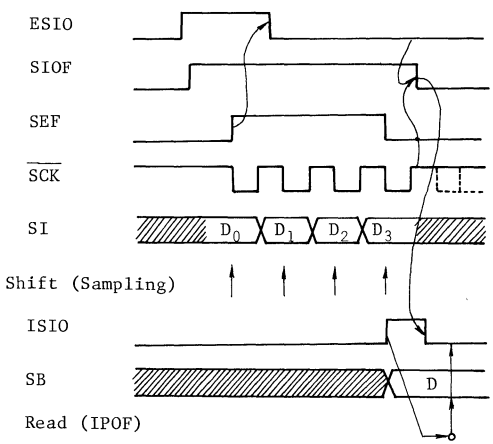
(a) Internal clock operation (with wait operation)



(b) External clock operation (at slow transfer rate)



(c) External clock operation (at fast transfer rate)



(d) One-word transfer

Fig. 1.11.9 Receive(Leading Edge Shift) Mode

2. Instructions

The TLCS-47 series microcomputer is provided with 90 instructions, which are software compatible within the series. The instructions of the TLCS-47 series is consist of 1-byte instructions or 2-byte instructions. To classify them in terms of the execution time, there are 1-cycle instructions and 2-cycle instructions.

1-byte, 1-cycle instructions are mainly used in this series, and are arranged so as to improve the program efficiency.

1-byte	1-cycle instruction	40
1-byte	2-cycle instruction	11
2-byte	2-cycle instruction	39
	Total	90

(a) Classification by byte/cycle

Move instruction (Note 1)	22
Compare instruction	6
Arithmetic instruction	16
Logical instruction	9
Bit manipulation instruction	24
Input/Output instruction (Note 2)	6
Branch, subroutine instruction	6
Other instruction	1
Total	90

(Note 1) : Including ROM data referring instructions

(Note 2) : Including PLA referring instruction.

(b) Classification by function

Table 2.0.1 Classification of Instructions.

2.1 Description of symbols

The following symbols are used for describing the instructions in the followirg explanations.

Symbol	Description
AC	Accumulator
M[x]	Data memory (Address x)
HR	H register
LR	L register
P[p]	Port (Address p)
FLAG	Flag
CF	Carry flag
ZF	Zero flag
SF	Status flag
GF	General flag
PC	Program counter
STACK[(SPW)]	Stack (Stack level is indicated by the contents of stack pointer word.)
SPW	Stack pointer word
EIF	Enable interrupt master F/F
EIR	Enable interrupt register
INTLj	Interrupt latch (j = 5 - 0)
DC	Data counter
ROM[x]	Program memory (Address x)
(ROM_H,ROM_L)	(High-order 4 bits or low-order 4 bits are expressed by suffix H/L.)
$\leftarrow$	Transfer
$\leftrightarrows$	Exchange
+	Addition
-	Substraction
$\wedge$	Logical AND of the corresponding bits
$\vee$	Logical OR of the corresponding bits
$\forall$	Exclusive OR of the corresponding bits

Symbol	Description
$(\overline{CF})$	Inversion of carry flag contents
null	Processed result is transferred nowhere
(AC)	Contents of accumulator
(H.L)	Contents of 8 bits coupling H register with L register
M[(H.L)]	Contents of data memory for which the contents of 8 bits coupling H register with L register is used as address.
(AC)<b>	Contents of bit assigned by b of accumulator.
(LR)<3:2>	Contents of bit 3 to bit 2 of L register
(PC)<11:6>	Contents of bit 11 to bit 6 of program counter

2.2 Description of instructions (*): Note 1 (**): Exec.cycle ($\overset{*}{**}$): Hexadecimal

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	($\overset{*}{**}$)	Functional Description		
Move Instruction	LD A, @HL	0 0 0 0 1 1 0 0	0 C	(AC)←M[(H·L)]	- Z 1	1
				Loads the contents of the data memory specified by the H and L registers in the accumulator.		
	LD A, x	0 0 1 1 1 1 0 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 C x_Hx_L	(AC)←M[x]	- Z 1	2
				Loads the contents of the data memory specified by the x of the instruction field in the accumulator.		
	LD HL, x	0 0 1 0 1 0 0 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 8 x_Hx_L	(LR)←M[x'], (HR)←M[x'+1] $x' = x_7x_6x_5x_4x_3x_200$	- - 1	2
				Loads the consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field in the H and L registers.		
	LD A, #k	0 1 0 0 $k_3k_2k_1k_0$	4 k	(AC)←k	- Z 1	1
				Loads the immediate data k of the instruction field in the accumulator. Serves as the clear instruction when k = 0.		

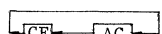
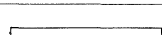
Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(※※)	Functional Description		
Move Instruction	LD H, #k	1 1 0 0 $k_3k_2k_1k_0$	C k	(HR)←k	- - 1	1
				Loads the immediate data k of the instruction field in the H register. Serves as the clear instruction when k = 0.		
	LD L, #k	1 1 1 0 $k_3k_2k_1k_0$	E k	(LR)←k	- - 1	1
				Loads the immediate data k of the instruction field in the L register. Serves as the clear instruction when k = 0.		
	LDL A, @DC	0 0 1 1 0 0 1 1	3 3	(AC)←ROM_L[(DC)]	- Z 1	2
				Loads the lower-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator.		
	LDH A,@DC+	0 0 1 1 0 0 1 0	3 2	(AC)←ROM_H[(DC)], (DC)←(DC)+1	- Z 1	2
				Loads the higher-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator, and then increments the contents of the data counter. [Note 2]		
	ST A, @HL	0 0 0 0 1 1 1 1	0 F	M[(H·L)]←(AC)	- - 1	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers.		
	ST A,@HL+	0 0 0 1 1 0 1 0	1 A	M[(H·L)]←(AC), (LR)←(LR)+1	- Z $\overline{C}$	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]		
	ST A,@HL-	0 0 0 1 1 0 1 1	1 B	M[(H.L)]←(AC), (LR)←(LR)-1	- Z $\overline{B}$	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers, and then decrements the contents of the L register. [Note 3]		
	ST A, x	0 0 1 1 1 1 1 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 F x_Hx_L	M[x]←(AC)	- - 1	2
				Stores the contents of the accumulator in the data memory specified by the x of the instruction field.		
	ST #k,@HL+	1 1 1 1 $k_3k_2k_1k_0$	F k	M[(H·L)]←k,(LR)←(LR)+1	- Z $\overline{C}$	1
				Stores the immediate data k of the instruction field in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]		

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (*)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Move Instruction	ST #k, y	0 0 1 0 1 1 0 1 $k_3k_2k_1k_o$ $y_3y_2y_1y_o$	2 D k y	M[y]←k Stores the immediate value k of the instruction field in the data memory specified by y (page 0) of the instruction field. Serves as the clear instruction when k = 0.	- - 1	2
	$\overline{\text{MOV}}$ H, A	0 0 0 1 0 0 0 0	1 0	(AC)←(HR) Loads the contents of the H register in the accumulator.	- Z 1	1
	$\overline{\text{MOV}}$ L, A	0 0 0 1 0 0 0 1	1 1	(AC)←(LR) Loads the contents of the L register in the accumulator.	- Z 1	1
	XCH A, H	0 0 1 1 0 0 0 0	3 0	(HR)⇆(AC) Exchanges the contents of the accumulator for those of the H register. [Note 2]	- Z 1	2
	XCH A, L	0 0 1 1 0 0 0 1	3 1	(LR)⇆(AC) Exchanges the contents of the accumulator for those of the L register. [Note 2]	- Z 1	2
	XCH A,EIR	0 0 0 1 0 0 1 1	1 3	(EIR)⇆(AC) Exchanges the contents of the accumulator for those of the interrupt enable register.	- - 1	1
	XCH A,@HL	0 0 0 0 1 1 0 1	0 D	M[(H·L)]⇆(AC) Exchanges the contents of the accumulator for those of the data memory specified by the H and L registers. [Note 2]	- Z 1	1
	XCH A, x	0 0 1 1 1 1 0 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_o$	3 D x_Hx_L	M[x]⇆(AC) Exchanges the contents of the accumulator for those of the data memory specified by the x of the instruction field. [Note 2]	- Z 1	2
	XCH HL, x	0 0 1 0 1 0 0 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_o$	2 9 x_Hx_L	M[x']⇆(LR), M[x'+1]⇆(HR) $x' = x_7x_6x_5x_4x_3x_2 00$ Exchanges the contents of the H and L registers for consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field.	- - 1	2

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (※※)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Compare Instruction	CMPR A,@HL	0 0 0 1 0 1 1 0	1 6	null←M[(H·L)]-(AC) Compares the contents of the data memory specified by the H and L registers with those of the accumulator.	$\overline{B}$ Z $\overline{Z}$	1
	CMPR A, x	0 0 1 1 1 1 1 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 E x_Hx_L	null←M[x]-(AC) Compares the contents of the data memory specified by the x of the instruction field with those of the accumulator.	$\overline{B}$ Z $\overline{Z}$	2
	CMPR A,#k	1 1 0 1 $k_3k_2k_1k_0$	D k	null←k-(AC) Compares the immediate data k of the instruction field with the contents of the accumulator. Serves as the accumulator test instruction when k = 0.	$\overline{B}$ Z $\overline{Z}$	1
	CMPR H,#k	0 0 1 1 1 0 0 0 1 1 0 1 $k_3k_2k_1k_0$	3 8 D k	null←k-(HR) Compares the immediate data k of the instruction field with the contents of the H register. Serves as the H register test instruction when k = 0.	- Z $\overline{B}$	2
	CMPR L,#k	0 0 1 1 1 0 0 0 1 0 0 1 $k_3k_2k_1k_0$	3 8 9 k	null←k-(LR) Compares the immediate data k of the instruction field with the contents of the L register. Serves as the L register test instruction when k = 0.	- Z $\overline{B}$	2
	CMPR y,#k	0 0 1 0 1 1 1 0 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 E k y	null←k-M[y] Compares the immediate data k of the instruction field with the contents of the data memory specified by the y (page 0) of the instruction field. Serves as the data memory test instruction when k = 0.	$\overline{B}$ Z $\overline{Z}$	2
Arithmetic Instruction	INC A	0 0 0 0 1 0 0 0	0 8	(AC)←(AC)+1 Increments the contents of the accumulator.	- Z $\overline{C}$	1
	INC L	0 0 0 1 1 0 0 0	1 8	(LR)←(LR)+1 Increments the contents of the L register.	- Z $\overline{C}$	1

Items / Class	Assembler Mnemonic	Object Code Binary	(*)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	INC @HL	0 0 0 0 1 0 1 0	0 A	M[(H·L)]←M[(H·L)]+1 Increments the contents of the data memory specified by the H and L registers.	- Z $\overline{C}$	1
	DEC A	0 0 0 0 1 0 0 1	0 9	(AC)←(AC)-1 Decrements the contents of the accumulator.	- Z $\overline{B}$	1
	DEC L	0 0 0 1 1 0 0 1	1 9	(LR)←(LR)-1 Decrements the contents of the L register.	- Z $\overline{B}$	1
	DEC @HL	0 0 0 0 1 0 1 1	0 B	M[(H·L)]←M[(H·L)]-1 Decrements the contents of the data memory specified by the H and L registers.	- Z $\overline{B}$	1
	ADDC A,@HL	0 0 0 1 0 1 0 1	1 5	(AC)←(AC)+M[(H·L)]+(CF) Adds the contents of the data memory specified by the H and L registers as well as those of the carry flag to those of the accumulator, and places the result in the accumulator.	C Z $\overline{C}$	1
	ADD A,@HL	0 0 0 1 0 1 1 1	1 7	(AC)←(AC)+M[(H·L)] Adds the contents of the data memory specified by the H and L registers to those of the accumulator, and places the result in the accumulator.	- Z $\overline{C}$	1
	ADD A, #k	0 0 1 1 1 0 0 0 0 0 0 0 $k_3k_2k_1k_0$	3 8 0 k	(AC)←(AC)+k Adds the immediate data k of the instruction field to the contents of the accumulator, and places the result in the accumulator. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	ADD H, #k	0 0 1 1 1 0 0 0 1 1 0 0 $k_3k_2k_1k_0$	3 8 C k	(HR)←(HR)+k Adds the immediate data k of the instruction field to the contents of the H register, and places the result in the H register. Serves as the H register increment instruction or the decrement instruction when k = 1 or F, respectively.	- Z $\overline{C}$	2

Items / Class	Assembler Mnemonic	Object Code (Binary)	(*) (**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	ADD L,#k	0 0 1 1 1 0 0 0 1 0 0 0 $k_3k_2k_1k_0$	3 8 8 k	(LR)←(LR)+k Adds the immediate data k of the instruction field to the contents of the L register, and places the result in the L register.	- Z $\overline{C}$	2
	ADD @HL,#k	0 0 1 1 1 0 0 0 0 1 0 0 $k_3k_2k_1k_0$	3 8 4 k	M[(H·L)]←M[(H·L)]+k Adds the immediate data k of the instruction field to the contents of the data memory specified by the H and L register, and places the result in the data memory. Serves as the correction instruction for the decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	ADD y,#k	0 0 1 0 1 1 1 1 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 F k y	M[y]←M[y]+k Adds the immediate data k of the instruction field to contents of the data memory specified by the y (page 0) of the instruction field, and places the result in the data memory. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	SUBRC A,@HL	0 0 0 1 0 1 0 0	1 4	(AC)←M[(H·L)]-(AC)-($\overline{CF}$) Subtracts the contents of the accumulator and the inverse contents of the carry flag from the contents of the data memory specified by the H and L registers, and places the result in the accumulator.	$\overline{B}$ Z $\overline{B}$	1
	SUBR A,#k	0 0 1 1 1 0 0 0 0 0 0 1 $k_3k_2k_1k_0$	3 8 1 k	(AC)←k-(AC) Subtracts the contents of the accumulator from the immediate data k of the instruction field, and places the result in the accumulator. Serves as the accumulator 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	- Z $\overline{B}$	2

Items / Class	Assembler Mnemonic	Object Code (Binary)	(*) (**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	SUBR@HL,#k	0 0 1 1 1 0 0 0 0 1 0 1 $k_3k_2k_1k_0$	3 8 5 k	M[(H·L)]←k-M[(H·L)] Subtracts the contents of the data memory specified by the H and L registers from the immediate data k of the instruction field, and places the result in the data memory. Serves as the data memory 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	- Z $\overline{B}$	2
Logical Instruction	ROLC A	0 0 0 0 0 1 0 1	0 5	 (rotate left by 1 bit) Rotates the contents of the accumulator and carry flag to the left by one bit. [Note 4]	C Z $\overline{C}$	1
	RORC A	0 0 0 0 0 1 1 1	0 7	 (rotate right by 1 bit) Rotates the contents of the accumulator and carry flag to the right by one bit. [Note 4]	C Z $\overline{C}$	1
	AND A,@HL	0 0 0 1 1 1 1 0	1 E	(AC)←(AC)∧M[(H·L)] Carries out the logical AND of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L register, and places the result in the accumulator.	- Z $\overline{Z}$	1
	AND A,#k	0 0 1 1 1 0 0 0 0 0 1 1 $k_3k_2k_1k_0$	3 8 3 k	(AC)←(AC)∧k Carries out the logical AND of the corresponding bits with the contents of the accumulator and the immediate data k of the instruction field, and places the result in the accumulator.	- Z $\overline{Z}$	2
	AND @HL,#k	0 0 1 1 1 0 0 0 0 1 1 1 $k_3k_2k_1k_0$	3 8 7 k	M[(H·L)]←M[(H·L)]∧k Carries out the logical AND of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.	- Z $\overline{Z}$	2

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (※※)	Function / Functional Description	Flag(*): CF	ZF	SF	(**)
Logical Instruction	OR A, @HL	0 0 0 1 1 1 0 1	1 D	(AC)←(AC)∨M[(H·L)]	-	Z	$\overline{Z}$	1
				Carries out the logical OR of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L registers, and places the result in the accumulator.				
	OR A, #k	0 0 1 1 1 0 0 0 0 0 1 0 $k_3k_2k_1k_0$	3 8 2 k	(AC)←(AC)∨k	-	Z	$\overline{Z}$	2
				Carries out the logical OR of the corresponding bits with the contents of the accumulator and the immediate data k of the istruction field, and places the result in the accumulator.				
	OR @HL,#k	0 0 1 1 1 0 0 0 0 1 1 0 $k_3k_2k_1k_0$	3 8 6 k	M[(H·L)]←M[(H·L)]∨k	-	Z	$\overline{Z}$	2
				Carries out the logical OR of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.				
	XOR A, @HL	0 0 0 1 1 1 1 1	1 F	(AC)←(AC)∀M[(H·L)]	-	Z	$\overline{Z}$	1
				Carries out the logical exclusive OR of the corresponding bits with the contents of the accumulator and those of data memory specified by the H and L registers, and places the result in the accumulator.				
Bit Manipulation Instruction	TEST CF	0 0 0 0 0 1 1 0	0 6	(SF)←($\overline{CF}$), (CF)←0	0	-	*	1
				Places the inverse contents of the carry flag in the status flag, and then resets the carry flag to "0".				
	TEST A, b	0 1 0 1 1 1 b_1b_0	5 C+b	(SF)←$\overline{(AC)<b>}$	-	-	*	1
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the accumulator, in the status flag.				
	TEST @HL,b	0 1 0 1 1 0 b_1b_0	5 8+b	(SF)←$\overline{M[(H·L)]<b>}$	-	-	*	1
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, in the status flag.				

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(*,**)	Functional Description		
Bit Manipulation Instruction	TEST y, b	0 0 1 1 1 0 0 1 1 0 b_1b_0 $y_3y_2y_1y_0$	3 9 8+b y	(SF)←$\overline{M[y]<b>}$	- - *	2
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.		
	TEST %P,b	0 0 1 1 1 0 1 1 1 0 b_1b_0 $p_3p_2p_1p_0$	3 B 8+b P	(SF)←$\overline{P[p]<b>}$	- - *	2
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the port (port register in the output port, and pin input in the input and I/O port) specified by the p of the instruction field, in the status flag.		
	TEST @L	0 0 1 1 0 1 1 1	3 7	(SF)←$\overline{P[(LR)<3:2>+4]<(LR)<1:0>>}$	- - *	2
				Places the inverse contents of the bit, which is specified by the lower-order two bits of the L register, of the ports R_4 - R_7 (pin input) specified by the higher two bits of the L register, in the status flag.		
	TESTP CF	0 0 0 0 0 1 0 0	0 4	(SF)←(CF), (CF)←1	1 - *	1
				Places the contents of the carry flag in the status flag, and then sets the carry flag to "1".		
	TESTP ZF	0 0 0 0 1 1 1 0	0 E	(SF)←(ZF)	- - *	1
				Places the contents of the zero flag in the status flag.		
	TESTP GF	0 0 0 0 0 0 0 1	0 1	(SF)←(GF)	- - *	1
				Places the contents of the general flag in the status flag.		
	TESTP y,b	0 0 1 1 1 0 0 1 1 1 b_1b_0 $y_3y_2y_1y_0$	3 9 C+b y	(SF)←M[y]<b>	- - *	2
				Places the contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.		

Items / Class	Assembler Mnemonic	Object Code Binary	(*)	Function / Functional Description	Flag(*) CF	ZF	SF	(**)
Bit Manipulation Instruction	TESTP %P,b	0 0 1 1 1 0 1 1 1 1 b_1b_0 $p_3p_2p_1p_0$	3 B C+b P	(SF)←P[p]<b> Places the contents of the bit, which is specified by the b of the instruction field, of the port (port register for the output port, and pin input for the input or I/O ports), which is specified by the p of the instruction field, in the status flag.	-	-	*	2
	SET GF	0 0 0 0 0 0 1 1	0 3	(GF)←1 Sets the general flag to "1".	-	-	1	1
	SET @HL, b	0 1 0 1 0 0 b_1b_0	5 b	M[(H·L)]<b>←1 Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, to "1".	-	-	1	1
	SET y, b	0 0 1 1 1 0 0 1 0 0 b_1b_0 $y_3y_2y_1y_0$	3 9 b y	M[y]<b>←1 Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "1".	-	-	1	2
	SET %p, b	0 0 1 1 1 0 1 1 0 0 b_1b_0 $p_3p_2p_1p_0$	3 B b y	P[p]<b>←1 Sets the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "1".	-	-	1	2
	SET @L	0 0 1 1 0 1 0 0	3 4	P[(LR)<3:2>+4]<(LR)<1:0>>←1 Sets the bit, which is specified by the lower-order two bits of the L register, of the ports R4 - R7 specified by the higher-order two bits of the L register, to "1".	-	-	1	2
	CLR GF	0 0 0 0 0 0 1 0	0 2	(GF)←0 Clears the general flag to "0".	-	-	1	1
	CLR @HL, b	0 1 0 1 0 1 b_1b_0	5 4+b	M[(H·L)]<b>←0 Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L register, to "0".	-	-	1	1

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF	ZF	SF	(**)
		Binary	(*,**)	Functional Description				
Bit Manipulation Instruction	CLR y, b	0 0 1 1 1 0 0 1 0 1 b_1b_0 $y_3y_2y_1y_0$	3 9 4+b y	M[y]<b> ← 0 Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "0".	-	-	1	2
	CLR %P, b	0 0 1 1 1 0 1 1 0 1 b_1b_0 $p_3p_2p_1p_0$	3 B 4+b p	P[p]<b> ← 0 Clears the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "0".	-	-	1	2
	CLR @L	0 0 1 1 0 1 0 1	3 5	P[(LR)<3:2>+4]<(LR)<1:0>> Clears the bit, which is specified by the lower-order two bits of the L register, of the ports R_4 - R_7 specified by the higher-order two bits of the L register, to "0".	-	-	1	2
	CLR IL, r	0 0 1 1 0 1 1 0 1 1 r_5r_4 $r_3r_2r_1r_0$	3 6 C+r_Hr_L	(INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Resets the interrupt latch INTLj when the rj of the instruction field is "0". (j = 5 - 0)	-	-	1	2
	EICLR IL,r	0 0 1 1 0 1 1 0 0 1 r_5r_4 $r_3r_2r_1r_0$	3 6 4+r_Hr_L	(EIF) ← 1, (INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Sets the interrupt enable master F/F to "1". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 - 0)	-	-	1	2
	DICLR IL,r	0 0 1 1 0 1 1 0 1 0 r_5r_4 $r_3r_2r_1r_0$	3 6 8+r_Hr_L	(EIF) ← 0, (INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Resets the interrupt enable master F/F to "0". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 - 0)	-	-	1	2
Input Instruction	IN %P, A	0 0 1 1 1 0 1 0 0 0 1 0 $p_3p_2p_1p_0$	3 A 2 P	(AC) ← P[p] Places the input data from the port specified by the p of the instruction field in the accumulator.	-	Z	$\overline{Z}$	2

Items / Class	Assembler Mnemonic	Object Code (Binary)	Object Code ($\frac{*}{**}$)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Input/Output Instruction	IN %P, @HL	0 0 1 1 1 0 1 0 0 1 1 0 $p_3p_2p_1p_0$	3 A 6 P	M[(H·L)]←P[p] Places the input data from the port specified by the p of the instruction field in the data memory specified by the H and L registers.	- - $\overline{Z}$	2
	OUT A, %P	0 0 1 1 1 0 1 0 1 0 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $8+2\overline{p}_4p$	P[p]←(AC), P=$P_4P_3P_2P_1P_0$ Outputs the contents of the accumulator to the port specified by the p of the instruction field. ($0 \leq p \leq 31$)	- - 1	2
	OUT @HL,%P	0 0 1 1 1 0 1 0 1 1 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $C+2\overline{p}_4p$	P[p]←M[(H·L)], P=$P_4P_3P_2P_1P_0$ Outputs the contents of the data memory specified by the H and L registers to the port specified by the p of the instruction field. ($0 \leq p \leq 31$)	- - 1	2
	OUT #k,%P	0 0 1 0 1 1 0 0 $k_3k_2k_1k_0$ $p_3p_2p_1p_0$	2 C k P	P[p]←k Outputs the immediate data k of the instruction field to the port specified by the p of the instruction field. Serves as the clear instruction when k = 0.	- - 1	2
	OUTB @HL	0 0 0 1 0 0 1 0	1 2	P[2]·P[1]←ROM[F·(F+(CF))·M[(H·L)]] Outputs the data (eight bits) of the program memory located in addresses FE0 - FFF, which use a five-bit data connecting the contents of the data memory specified by the H and L registers and those of the carry flag, as lower-order five-bit addresses, to the P_2 - P_1 ports.	- - 1	2
Branch· Subroutine Instruction	BS a	0 1 1 0 $a_{11}a_{10}a_9a_8$ $a_7a_6a_5a_4$ $a_3a_2a_1a_0$	6 a_H a_M a_L	If SF=1 then (PC)←a else null. Places the immediate data a of the instruction field in the program counter if the status flag is at "1". If the status flag is at "0", sets the status flag only to "1", and moves to the next address.	- - 1	2

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF	ZF	SF	(**)
		Binary	(**)	Functional Description				
Branch·Subroutine Instruction	BSS a	1 0 d_5d_4 $d_3d_2d_1d_0$	8+d_H d_L	If SF=1 then (PC)←a else null, a=(PC)<11·6 > ·d	-	-	1	1
				Carries out the branch within a page (64-byte) if the status flag is at "1"; brings the immediate value d of the instruction field into the lower-order six bits of the program counter. Since the updated value remains in the higher-order six bits, if this instruction is specified in the last address in the page, branching is carried out to the next page. If the status flag is at "0", it sets the status flag only to "1", and moves to the next address. [Note 5]				
	CALL a	0 0 1 0 0 $a_{10}a_9a_8$ $a_7a_6a_5a_4$ $a_3a_2a_1a_0$	2 aH aM aL	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, 0 ≤ a ≤ 2,047	-	-	-	2
				Carries out the subroutine call; saves the contents of the program counter in the stack, and decrements the stack pointer word, and then places the immediate data a of the instruction field in the program counter. However, the call address of the subroutine must be in the addresses 000 -7FF. [Note 5]				
	CALLS a	0 1 1 1 $n_3n_2n_1n_0$	7 n	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, a=8n+6(n≠0), 134(n=0)	-	-	-	2
				Carries out the short form subroutine call. The operation is the same as that of the "CALL" instruction except that the value to be set in the program counter is automatically defined by the n of the instruction field. [Note 5]				
	RET	0 0 1 0 1 0 1 1	2 A	(SPW)←(SPW)+1,(PC)←STACK[(SPW)]	-	-	-	2
				Returns from the subroutine to the previous program; increments the stack pointer word, and restores the data of the return address from the stack to the program counter.				

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binaty	(**)	Functional Description		
Branch·Subroutine Instruction	RETI	0 0 1 0 1 0 1 1	2 B	(SPW)←(SPW)+1, (FLAG·PC)←STACK[(SPW)], (EIF)←1	* * *	2
				Returns from the interrupt processing routine; increments the stack pointer word, and restores the data of the return address from the stack and the data of the flag, to the program counter and the flag, respectively. And then, it sets the interrupt enable master F/F to "1".		
Other Instruction	$\overline{\text{NOP}}$	0 0 0 0 0 0 0 0	0 0	no operation	- - -	1
				Moves to the next instruction without performing any operation.		

Note 1. Setting Condition of Flag.

"C" indicates the carry output from the most significant position in the addition operation, and "B" indicates the borrow output from the most significant position in the subtraction operation.

"Z" indicates the zero detection signal to which "1" is applied only when either the ALU output of the processing result or all four bits of the data transferred to the accumulator are zero.

The flag is set to "C", "$\overline{C}$", "$\overline{B}$", "Z", "$\overline{Z}$", "1", or "0" according to the data processing rusult. The value specified by the function is set to the flag with the mark "*", and the mark "-" denotes no change in the state of the flag.

Note 2. The zero flag is set according to the data set in the accumulator.

Note 3. The flags (ZF, SF) are set according to the result of increment or decrement of the L register.

Note 4. The carry is the data shifted out from the accumulator.

Note 5. The contents of the program counter indicate the next address of the instruction to be executed.

3. Basic operation and pin operation

1. Instruction cycle
2. Basic clock (CP) generation
3. Initialization operation
4. Memory stand-by function
5. Interrupt input
6. Input/output port
7. Other pins

The timing in each basic operation, and the configuration, function, and timing of the pins are described according to the above items.

The operation and timing with each component of the hardware are covered in detail in the description of each item of the components.

Different input/output port circuit system can be specified according to the port. The details to specify the type of input/output port circuit are given in the descreption covering the program tape format.

3.1 Instruction cycle

The instruction execution and the internal hardware control are synchronized with the basic clock (CP, fc Hz).

The minimum unit of the instruction execution is called the "instruction cycle", and all instructions are executed by one or two instruction cycles, each of which is called one-cycle instruction or two-cycle instruction.

An instruction cycle consists of four machine cycles ($M_1 \sim M_4$), and each machine cycle requires two basic clock times.

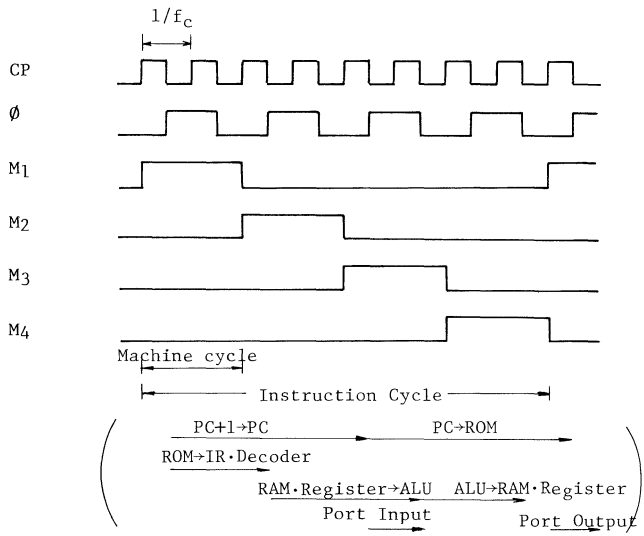


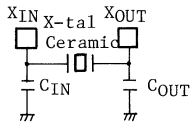
Fig. 3.1.1 Instruction Cycle

3.2 Basic clock (CP) generation

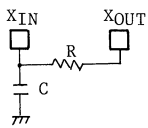
An oscillation circuit is contained, and the necessary clock is easily generated by connecting the resonator to external pins (X_{IN}, X_{OUT}). By the way, the oscillation circuit serves as schmitt circuit.

The clock generated in the oscillation circuit is called the "basic clock" with which the internal control is synchronized. The basic clock is applied to the timing generator and the control circuit of system to provide various control signals.

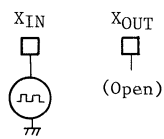
The following are the examples of the resonator connection.



(a) For X-tal or ceramic resonator



(b) For RC



(c) For external oscillator

Fig. 3.2.1 Resonator Connections

3.3 Initialization operation

Initialization operation is performed by keeping the $\overline{\text{RESET}}$ pin to the low level. However, the following conditions are required to put the initialization operation into practice with certainty

① The supply voltage is within the operating voltage.

② The oscillation circuit operates stably.

③ The $\overline{\text{RESET}}$ is held at the low level in at least three instruction cycle time.

The following processings are performed by the initialization operation.

① Reset the program counter to "0".

② Set the status flag to "1".

③ Reset the interrupt enabling master F/F and the interrupt enabling register to "0", and also reset the interrupt latch to "0".

④ Reset the divider to "0".

⑤ Initialize the input/output port and command register to the fixed level.

The initialization operation is released due to the rise of the $\overline{\text{RESET}}$ pin to the high level, and the program can be executed from address 0 in sequence.

The $\overline{\text{RESET}}$ pin serves as Schmitt circuit input, and is connected with pull-up resister ($\approx$300kΩ TYP., MOS-load resister).

3.4 Memory Stand-by function

Even during the cut off of the main power supply (V_{DD}), the RAM data can be held with low power dissipation by connecting the back-up power supply to the V_{HH} pin. This memory stand-by operation is performed by the following procedure.

① Keep the $\overline{\text{RESET}}$ pin at the low level in at least three instruction cycle time before the V_{DD} power goes to the minimum operating voltage.

② Hold the low level of the $\overline{\text{RESET}}$ pin. At the same time, the level of the V_{HH} voltage should be kept at that of more than mimimum stand-by voltage.

The operation should be started from initialization operation after the main power supply is resumed.

The power dissipation at the stand-by time can be minimized by this function.

3.5 Interrupt input

Two pins ($\overline{INT_1}$, $\overline{INT_2}$) are provided for the external interrupt input. Since these pins are common pins with R_8 port, they can be used as I/O pins (R_{82}, R_{80}) respectively, if not used as the interrupt input pins.

The interrupt via INT_2 can be inhibited at any time by the program, but the interrupt via INT_1 is not inhibited by it independently. Therefore, when this pin is used for the R_{82} port, the interrupt will always take place due to the detection of the falling edge of the signal. It is necessary to set a dummy interrrupt service program including the (RETI) instruction only, even if the INT_1 is not used.

The interrupt latch is set by the falling edge of the external inputs (INT_1, INT_2), and an interrupt request is made to the CPU. To assure that the interrupt latch is positively set or reset, and that the next interrupt request is set, both of the high and low levels should be kept for more than two instruction cycle time.

The external interrupt input is the Schmitt circuit input.

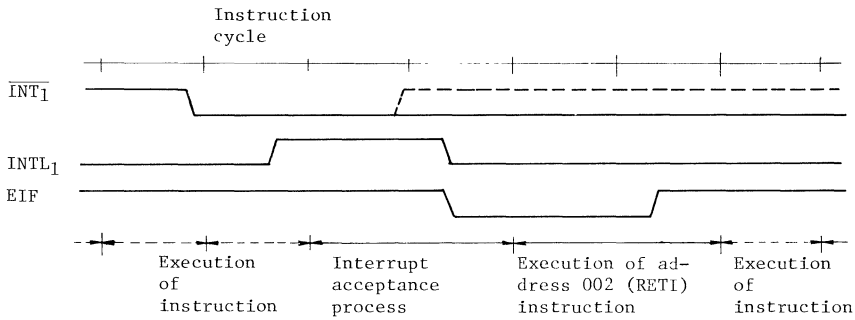


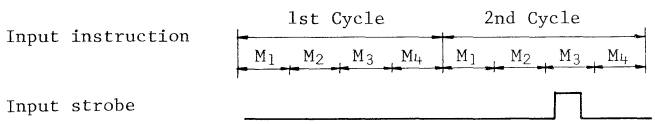
Fig. 3.5.1 Interrupt Timing (Dummy process of INT_1 interrupt)

3.6 Input/output port

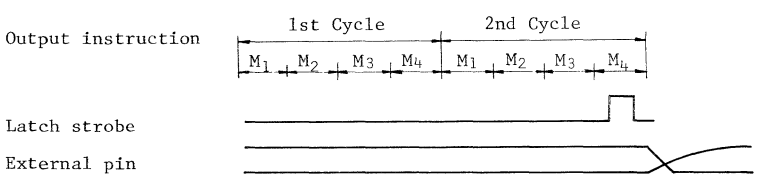
(1) Input/output timing

The timing to read the external data from the input port or I/O port is in M_3 machine cycle in the second cycle of the input instruction (two-cycle instruction). Since this timing cannot be externally recognized, the transient input data should be processed by a program.

The timing to output the data to the output port or I/O port is in M_4 machine cycle in the second cycle of the output instruction (two-cycle instruction), but this timing cannot be externally recognized.



(a) Input Timing



(b) Output Timing

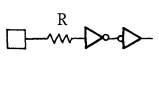
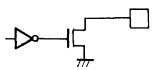
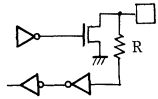
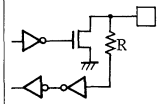
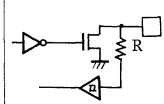
Fig. 3.6.1 Input/Output Timing

(2) Input/Output circuit format

The input/output circuit format of the input/output port is shown following.

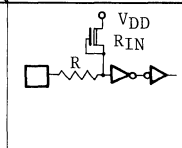
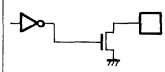
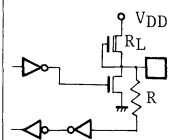
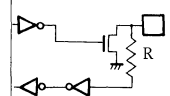
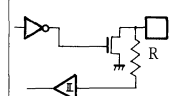
For the TMP4740P and the TMP4720P, any of the input/output circuit systems shown in the following tables can be selected. You can specify your input/output circuit system when requesting the program tape.

"IOCODE AA" is employed if not specified.

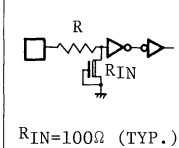
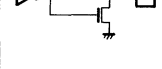
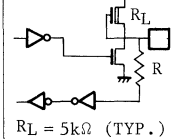
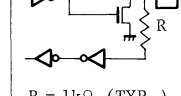
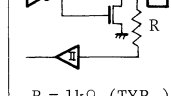
Input/Output Circuit Code (IOCODE) AA					
Port / Circuit	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
I/O equivalent Circuit	 R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	∘ High threshold input. ∘ No resistor is contained.	∘ Sink open drain output. ∘ High output current. ∘ Output latch is initialized to the high level.	∘ Sink open drain output. ∘ Output latch is initialized to the high level.	∘ Sink open drain output. ∘ Output latch is initialized to the high level.	∘ Schmitt circuit input. ∘ Sink open drain output. ∘ Output latch is initialized to the high level.

Note: The input/output port of the evaluator chip TMP4700AC is made up with the circuit system equivalent to this input/output circuit system; therefore, the system of the TMP4700AC can become equivalent to that of the TMP4740P or the TMP4720P by externally installing EPROM (program memory) on the TMP4700AC (but TMP4700AC is not contained the pull-up resistor with $\overline{\text{RESET}}$ pin and the pull-down resistor with TEST pin.).

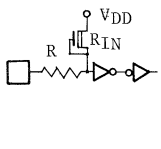
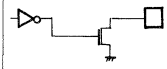
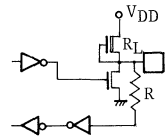
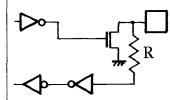
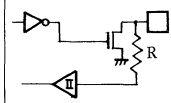
Input/Output Circuit Code ($I\overline{OCO}DE$) AE

Port / Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN}=100kΩ(TYP.) R = 1kΩ		 R_L = 5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-up resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the high level.	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

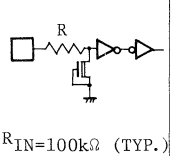
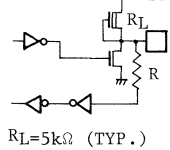
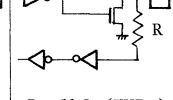
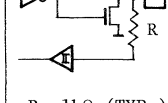
Input/Output Circuit Code ($I\overline{OCO}DE$) AF

Port / Circuit	Input (K_0)	Output (P_1.P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN}=100Ω (TYP.) R = 1kΩ (TYP.)		 R_L = 5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-down resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor contained. o Output latch is initialized to the high level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) AH

Port / Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN}=100kΩ (TYP.) R = 1kΩ (TYP.)		 R_L=5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-up resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the low level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) AI

Port / Circuit	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN}=100kΩ (TYP.) R = 1kΩ (TYP.)		 R_L=5kΩ (TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o High threshold input o Pull-down resistor is contained.	o Sink open drain output. o High output current. o Output latch is initialized to the high level	o Sink open drain output. o Pull-up resistor is contained. o Output latch is initialized to the low level	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level

3.7 Other pins

Timer/Counter input

Two pins (T_1, T_2) are provided for the external timer/counter inputs. Since these pins are common pins with R_8 port, they can be also used as I/O pins (R_{83}, R_{81}), respectively, if not used as the timer/counter inputs.

The count latch is set by the rising edge of the external input (T_1, T_2), and a count request is made to the CPU. To assure that the count latch is positively set or reset, both of the high and low levels should be kept for more tham two instruction cycle times.

The external timer/counter input is the Schmitt circuit input.

Serial port

This port is connected to the external circuitry via three pins ($\overline{SCK}$, SO, SI), which are also used for the R_9 port. These pins can be used as the pins of the R_9 port (R_{92}, R_{91}, R_{90}), if not used for the serial port.

To assure that the shift operation is positively performed in the external clock mode, both of the high and low levels should be kept for more than two instruction cycle times.

The $\overline{SCK}$ input in the external clock mode and the SI input in the receive mode are Schmitt circuit inputs.

TEST pin

This pin is used for the shipment test. To operate the user system with this pin, the input should be surely set to the low level. By the way, TEST pin is connected with pull-down resistor ($\approx$ 70kΩ TYP., MOS-load resistor).

TENTATIVE

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS}=0V)

SYMBOL	ITEM	RATING	UNITS
V_{DD}	Supply Voltage	-0.5 ∿ 7	V
V_{HH}			
V_{IN}	Input Voltage	-0.5 ∿ 7	V
V_{OUT1}	Output Voltage (Except Open Drain Port)	-0.5 ∿ 7	V
V_{OUT2}	Output Voltage (Open Drain Port)	-0.5 ∿ 10	
I_{OUT}	Output Current (P_1, P_2)	30	mA
P_D	Power Dissipation (T_{opr}=70°C)	850	mW
T_{sol}	Soldering Temperature · Time	260 (10 sec)	°C
T_{stg}	Storage Temperature	-55 ∿ 125	
T_{opr}	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS}=0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNITS
T_{opr}	Operating Temperature		-30	70	°C
V_{DD}	Supply Voltage		4.5	5.5	V
V_{HH}					
V_{HH1}	Supply Voltage (Memory Stand-by)		3.5	5.5	
V_{IH1}	High Level Input Voltage (R_4 ∿ R_7)		2.2	V_{DD}	V
V_{IH2}	High Level Input Voltage(Except R_4∿ R_7)		3	V_{DD}	
V_{IL1}	Low Level Input Voltage (Except K_0)		0	0.8	
V_{IL2}	Low Level Input Voltage (K_0)		0	1.2	
f_C	Clock Frequency		0.4	4.2	MHz
t_{WCH}	High Level Clock Pulse Width(Note 1)	V_{IN}=V_{IH}	80	-	nS
t_{WCL}	Low Level Clock Pulse Width (Note 1)	V_{IN}=V_{IL}	80	-	

(Note 1) For external clock operation.

DC CHARACTERISTICS (V_{SS}=0V, $V_{DD}=V_{HH}$=5V±10%, T_{opr}=-30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	(Note1) TYP.	MAX.	UNITS
V_{HS}	Hysteresis Voltage (Schmitt Circuit Input)		-	0.5	-	V
I_{IN1}	Input Current (Note 2) (K_0)	$V_{DD}=V_{HH}$=5.5V, V_{IN}=5.5V	-	-	20	μA
I_{IN2}	Input Current (Open Drain R Port)	V_{DD}=5.5V, V_{IN}=5.5V	-	-	20	μA
I_{IL}	Low Level Input Current (R Port with Pull-up Resistor)	V_{DD}=5.5V, V_{IN}=0.4V	-	-	-2	mA
I_{LO}	Output Leak Current (Open Drain P, R Port)	V_{DD}=5.5V, V_{OUT}=5.5V	-	-	20	μA
V_{OH}	High Level Output Voltage (R Port with Full-up Resistor)	V_{DD}=4.5V, I_{OH}=-200μA	2.4	-	-	V
V_{OL}	Low Level Output Voltage (Except X_{OUT})	V_{DD}=4.5V, I_{OL}=1.6mA	-	-	0.4	V
I_{OL}	Low Level Output Current (P_1, P_2)	V_{DD}=5V, V_{OL}=1V	-	20	-	mA
$I_{DD}+I_{HH}$	Supply Current	$V_{DD}=V_{HH}$=5.5V	-	50	120	mA
I_{HH1}	Supply Current(Memory Stand-by)	$V_{DD}=V_{SS}$, V_{HH}=3.5V	-	5	10	mA

(Note 1) Typical values are at T_{opr}=25°C, $V_{DD}=V_{HH}$=5V.

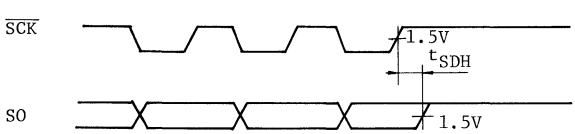
(Note 2) When an input resistor is built in the device, the input current through the resistor is eliminated.

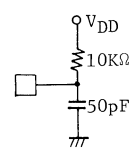
AC CHARACTERISTICS (V_{SS}=0V, $V_{DD}=V_{HH}$=5V±10%, T_{opr}=-30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
t_{cy}	Instruction Cycle Time		1.9	-	20	μS
t_{SDH}	Shift data hold time	(Note 1)	0.5tcy-300	-	-	nS

AC TIMING CHART

• Serial Port (Completion of transmission)

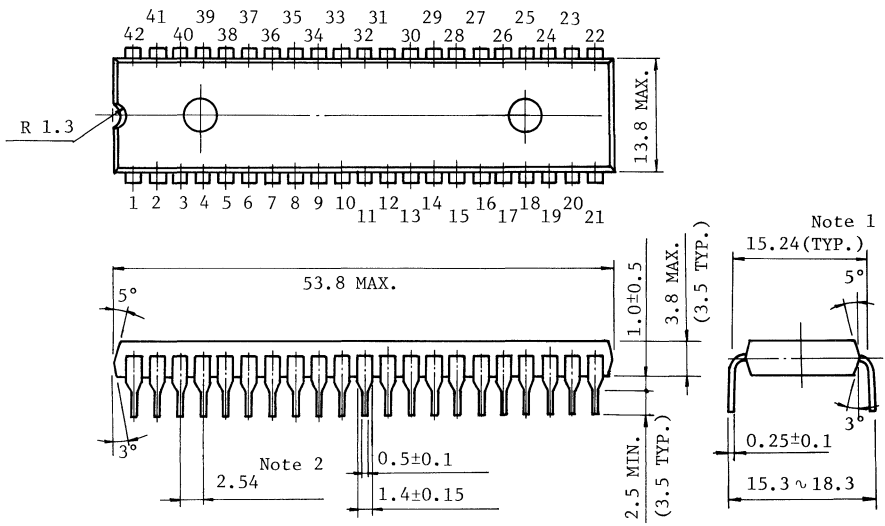




(Note 1) External circuit for serial ports SCK and SO

EXTERNAL DIMENSION VIEW

Unit in mm



Weight 5.7g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.42 leads.

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4740N, TMP4720N

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47N)

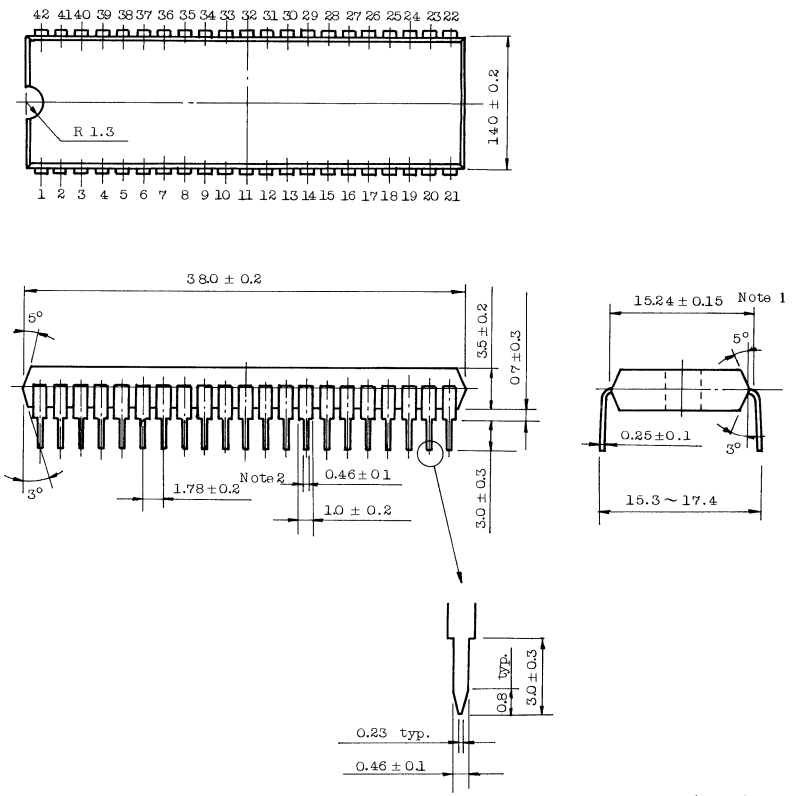
TMP4740N, TMP4720N

GENERAL DESCRIPTION

TMP4740N and TMP4720N are the shrunk package versions of TMP4740P and TMP4720N, respectively. Their function, instruction, pin description and electrical characteristics are compatible. The package area is reduced to around 70 percent in comparison with the standard package.

EXTERNAL DIMENSION VIEW

Unit in mm



Weight 4.0g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 1.78mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.42 leads.

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4700AC

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER TLCS-47(N) TMP4700AC

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series designed for general purpose use.

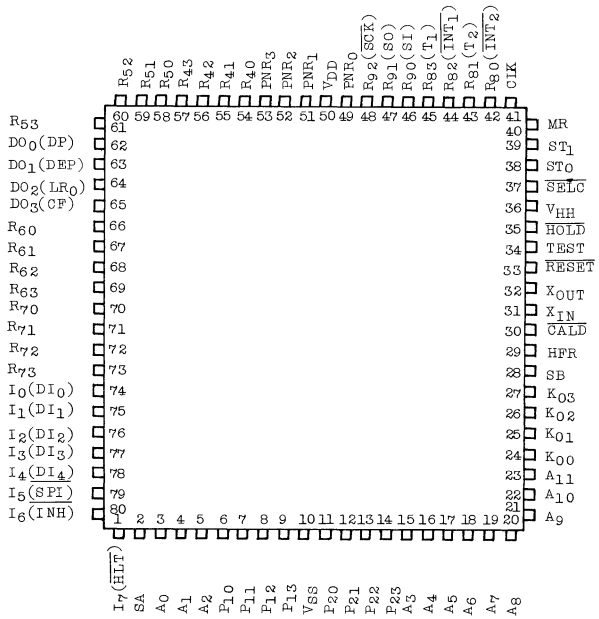
The TLCS-47 has variously powerful functions in order to meet with the advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

The TMP4700AC is the system development evaluator chip used for developmental and operational check of the TLCS-47 application systems (programs).

Although the TLCS-47N and the TLCS-47C have different electric characteristics and some functions, the individual configuration of a functionally equivalent system is possible by using the TMP4700AC.

Further, when the TMP4700AC is used, the evaluation boards equivalent to respective versions of the TLCS-47 should be used.

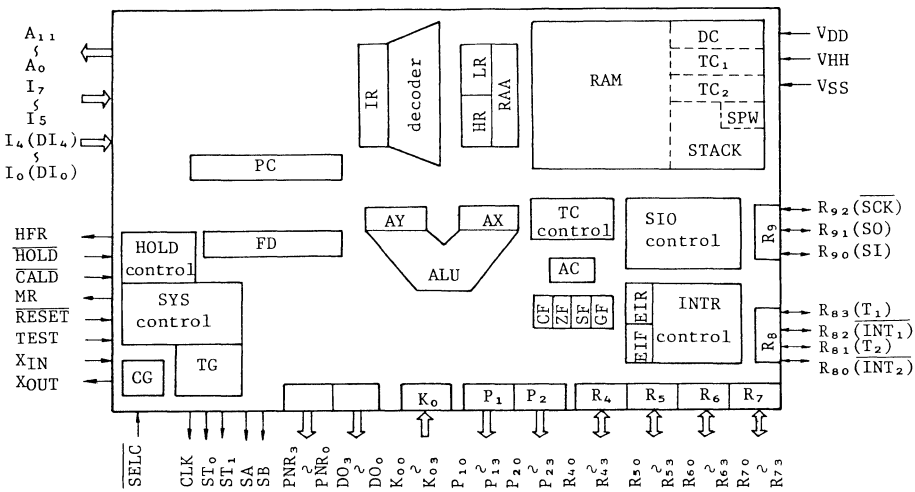
PIN CONNECTIONS (Top View)



PIN NAMES AND PIN DESCRIPTIONS

Pin Names	No.of Pins	I/O	Functions
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (corresponding to PLA)
$P_{23} \sim P_{20}$	4	Output	" (")
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
R_{83} (T_1)	1	I/O	I/O port or timer/counter input
R_{82} ($\overline{INT_1}$)	1	I/O	" or interrupt input
R_{81} (T_2)	1	I/O	" or timer/counter input
R_{80} ($\overline{INT_2}$)	1	I/O	" or interrupt input
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	" or serial output
R_{90} (SI)	1	I/O	" or serial input
$A_{11} \sim A_0$	12	Output	Program memory address
I_7 ($\overline{HLT}$)	1	Input	Program data input (Holt request signal input)
I_6 ($\overline{INH}$)	1	Input	" (Inhibit control signal input)
I_5 ($\overline{SPI}$)	1	Input	" (Port control signal input)
$L_4(DI_4) \sim I_0(DI_0)$	5	Input	" (Data input)
DO_3 (CF)	1	Output	Data Output (Carry flag monitor)
DO_2 (LR_0)	1	Output	" (L register monitor)
DO_1 (DEP)	1	Output	" (Port control signal output)
DO_0 (DP)	1	Output	" (")
$PNR_3 \sim PNR_0$	4	Output	Port address output
CLK	1	Output	Strobe signal
ST_0, ST_1	2	Output	State signal
SA, SB	2	Output	Status signal
MR	1	Output	Master reset signal output
HFR	1	Output	Hold monitor output
$\overline{CALD}$	1	Input	Data fetch cycle request signal input
$\overline{SELC}$	1	Input	Clock select input
$\overline{HOLD}$	1	Input	Hold signal input
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminal
$\overline{RESET}$	1	Input	Initialize signal input
TEST	1	Input	(Low level is input.)
V_{DD}	1	Power supply	+5V
V_{HH}	1	Power supply	+5V (Memory power supply)
V_{SS}	1	Power supply	0V

BLOCK DIAGRAM



BLOCK NAMES AND DESCRIPTIONS

Block Names	Functions
PC	Program counter (12 bits)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Deta memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC	Data counter (12 bits, RAM area)
AX, AY	Temporary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG(CF,ZF,SF,GF)	Flags
K, P, R	Ports
INTR control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2-channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
HOLD control	Control of hold function
SYS control	Generation of various internal control signals
CG, TG,	Clock generator, timing generator

FUNCTIONAL DESCRIPTION

The TMP4700AC is the system development evaluator chip for the TLCS-47. When a program memory (equivalent to TMM2732D, TMM323D-1) is externally mounted, it is possible to configurate a system equivalent to the TMP4740P or the TMP4720P (the input/output circuit format, however, must be equivalent to ($\overline{\text{IOCODE}}$ AA) and in the case of ($\overline{\text{IOCODE}}$ AE) and ($\overline{\text{IOCODE}}$ AF), externally mounted resistors are required).

In the case of other input/output circuit formats of the TMP4740P and TMP4720P, or in the case of other NMOS family or CMOS family, it is also possible to configurate an equivalent system by adding an external circuit using an evaluator chip dedicated terminal. Therefore, in application systems of these models, the evaluation boards equivalent to respective versions shall be used.

Further, when the TMP4700AC is used, the technical descriptions for respective versions and the instruction manuals for equivalent evaluation boards, debugging tools and the like shall also be read.

The operation of the TMP4700AC is described in the following on the basis of the terminal functions.

1. TLCS-47N standard chip equivalent terminals

   The terminals shown in Fig. 1.1 have the functions and characteristics equivalent to the input/output circuit format ($\overline{\text{IOCODE}}$ AA) of the standard chips (TMP4740P, TMP4720P) of the TLCS-47N. Therefore, in this case it is possible to configurate an equivalent system by externally mounting a program memory.

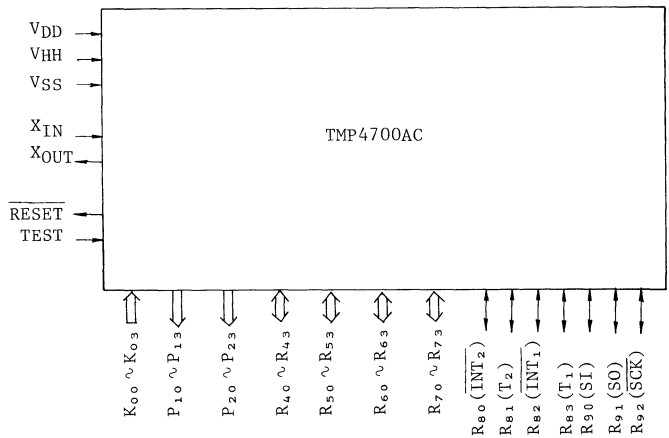
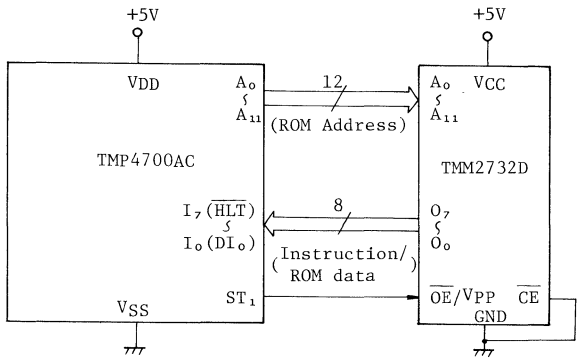


Fig.1.1 TLCS-47N Standard Chip ($\overline{\text{IOCODEAA}}$) Equivalent Terminals

2. Connection of Program Memory

As an externally mounted program memory, a programable ROM equivalent to the TMM2732D (4K x 8 bits) or TMM323D-1 (2K x 8 bits) is used.

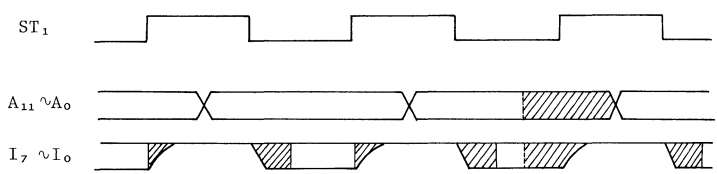
The connecting method of a program memory and the timing chart are shown in Fig. 2.1.
Further, A_{11} and I_7 ($\overline{\text{HLT}}$) terminals in the diagram are MSB, respectively.



Note 1. When the TMM323D-1 is used, the TMP4700AC output terminal A_{11} should be opened.

Note 2. The instruction/ROM data input terminal has a built-in pull-up resistors.

(a) Connection of Program Memory



(b) Program Memory Access Timing Chart

Fig. 2.1 Connection of Program Memory

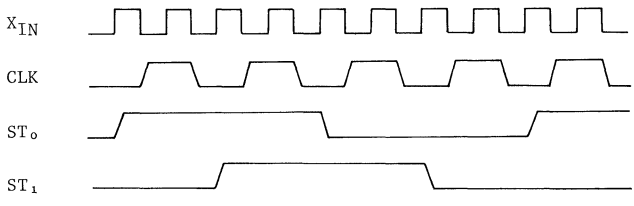
3. Control Terminals for External Circuits

(1) Timing signals (CLD, ST_0, ST_1, $\overline{SELC}$)

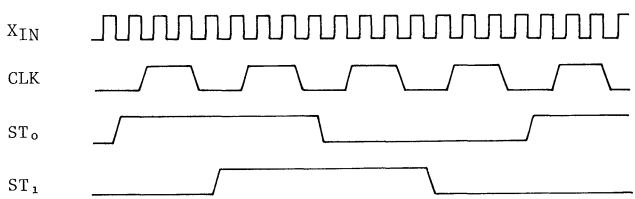
In order for the timing control of the external circuits, 3 types of signals are transmitted from the timing generator of the TMP4700AC.

The TMP4700AC is capable of supporting either system of the TLCS-47N and the TLCS-47C. For selecting these systems, the $\overline{SELC}$ signal input is used.

The timing chart of these signals is shown in Fig. 3.1. Further, the $\overline{SELC}$ terminal has a built-in pull-up resistor.



(a) TLCS-47N Support ($\overline{SELC}$ = 1)



Note: These are somewhat different from the operating timings of CMOS family.

(b) TLCS-47C Support ($\overline{SELC}$ = 0)

Fig. 3.1 Clock Timing Chart

(2) System control signal inputs

I_7 ($\overline{HLT}$)

$\overline{HLT}$ signal is the halt request signal input to the TMP4700AC at time of the system debugging. $\overline{HLT}$ signal input is multiplexed with data input from the external ROM and a signal is input when ST_1 signal is at high level.

When a low level signal is input into $\overline{HLT}$ signal input and accepted, the TMP4700AC starts the half operation. At this time, CPU executes no operation cycle, but as long as HLT request is being accepted, it stops the divider to operate (therefore, the counting for the timer interruption of divider, the internal clock to the timer/counter and the internal shift clock for serial transfer are also stopped, accordingly), and furthermore, it inhibits the timer/counter operation and acceptance of interrupt requests.
However, a request for LCD data fetch cycle, which is used on LCD driver built-in version is accepted (one instruction cycle).
Further, the interrupt latch and the count latch for the timer/counter are set/reset independently of HLT operation and subsequent INH operations.

Further, I_7 ($\overline{HLT}$) terminal has a built-in pull-up resistor.

I_6 ($\overline{INH}$)

$\overline{INH}$ signal is the control signal input for temporarily inhibiting the divider operation, timer/counter operation and interrupt request acceptance at time of the system debugging. $\overline{INH}$ signal input is multiplexed with data input from the external ROM and a signal is input when ST_1 is at high level.

As long as a low level signal is input into $\overline{INH}$ input and is being accepted, the TMP4700AC stops the divider to operate and inhibits the timer/counter operation and acceptance of interrupt requests. However, a request for LCD data fetch cycle, which is used on LCD driver built-in version is accepted (one instruction cycle).

Since this INH operation can be controlled independently of HLT operation, it can be used in normal system program operation. Furthermore, it also can be used for controlling the internal monitor at time of the system debugging.

Further, I_6 ($\overline{INH}$) terminal has a built-in pull-up resistor.

$\overline{HOLD}$

This input is equivalent to the $\overline{HOLD}$ terminal provided in the TLCS-47C.

As the system operation for the hold function, operation of this input is similar to that of each version of the TLCS-47C for $\overline{HOLD}$ terminal input except the followings:

(a) The oscillator is not stopped (normal oscillation is continued).

(b) Supply current don't decrease from the value of the TMP4700AC operating current.

Further, this $\overline{HOLD}$ terminal has a built-in pull-up resistor.

$\overline{CALD}$

This is a request signal input for the data fetch cycle, which is used on LCD driver built-in version.

When a low level signal is input into the $\overline{CALD}$ input and accepted, the TMP4700AC executes the LCD data fetch cycle (one instruction cycle).

Further, this $\overline{CALD}$ terminal has a built-in pull-up resistor.

(3) System control signal outputs

SA, SB

SA and SB signal outputs are signals for monitoring the internal operation of the TMP4700AC(See Table 3.1). These signals are switched for every instruction cycle.

SA	SB	
0	0	Executes the first cycle of an instruction
0	1	Executes the LCD data fetch cycle by a CALD request
1	0	Executes the halt operation by a HLT request
1	1	Executes other operations

Table 3.1 SA, SB Signal Outputs

MR

This is a response signal to $\overline{\text{RESET}}$ signal input, and is the system reset signal.

HFR

This signal is a monitor signal relative to the hold operation and is also used for an external circuit control.

(4) Port control

In order to support the versions of TLCS-47 series commonly, the TMP4700AC is able to input data from an external circuit or to output data to a register created in an external circuit.

(a) Control signals

$PNR_3 \sim PNR_0$

4 bit outputs indicating port addresses.

DO_0 (DP), DO_1 (DEP)

These signals (DP, DEP) control the port write/read by the external circuits. They are multiplexed with data output (DO) and are transmitted when ST_1 signal is at high level.

I_5 ($\overline{SPI}$)

$\overline{SPI}$ signal controls the port read by the external circuits. This signal is multiplexed with data input from an external ROM and is input when ST_1 signal is at high level.

(b) Data inputs

I_4 (DI_4) ∿ I_0 (DI_0)

These (DI_4 ∿ DI_0) are the data input terminals at time of the read operation from the external circuits. They are multiplexed with data inputs from the external ROM and are input when ST_1 signal is at high level.

(c) Data outputs

DO_3 (CF), DO_2 (LR_0), DO_1 (DEP), DO_0 (DP)

These (DO_3 ∿ DO_0) are the data output terminals at time of the write operation to the external circuits. These outputs are multiplexed with other outputs and are transmitted out when ST_1 signal is at low level.

Note: The port output timing on each versions of the TLCS-47 series and that on the TMP4700AC external circuit somewhat differ each other.

DO_3 (CF)

Contents of the carry flag are transmitted. This CF output is multiplexed with the data output (DO) and is sent out when ST_1 signal is at high level.

DO_2 (LR_0)

Contents of LSB of L register is transmitted. This LR_0 output is multiplexed with the data output (DO) and is sent out when ST_1 signal is at high level.

TENTATIVE

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATING (V_{SS} = 0V)

SYMBOL	ITEM	RATING	UNIT
V_{DD} / V_{HH}	Supply Voltage	-0.5 ∿ 7	V
V_{IN}	Input Voltage	-0.5 ∿ 7	V
V_{OUT1}	Output Voltage (Except Open Drain Port)	-0.5 ∿ 7	V
V_{OUT2}	Output Voltage (Open Drain Port)	-0.5 ∿ 10	
I_{OUT}	Output Current (P_1, P_2)	30	mA
P_D	Power Dissipation (Topr = 70°C)	1	W
Tsol	Soldering Temperature · Time	260(10sec.)	°C
Tstg	Storage Temperature	-55 ∿ 125	
Topr	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNIT
Topr	Operating Temperature		-30	70	°C
V_{DD} / V_{HH}	Supply Voltage		4.5	5.5	V
V_{HH1}	Supply Voltage (Memory Stand-by)		3.5	5.5	
V_{IH1}	High Level Input Voltage (Note 1)		2.2	VDD	V
V_{IH2}	High Level Input Voltage (Note 2)		3	VDD	
V_{IL1}	Low Level Input Voltage (Except K_Q)		0	0.8	
V_{IL2}	Low Level Input Voltage (K_Q)		0	1.2	
f_C	Clock Frequency		0.4	4.2	MHz
t_{WCH}	High Level Clock Pulse Width (Note 3)	VIN = VIH	80	-	nS
t_{WCL}	Low Level Clock Pulse Width (Note 3)	VIN = VIL	80	-	

(Note 1) Application terminals: R_4 ∿ R_7, $I_7(\overline{HLT})$ ∿ $I_O(DI_O)$

(Note 2) Application terminals: Inputs other than application termianl (Note 1)

(Note 3) For external clock operation

D.C. CHARACTERISTICS (V_{SS} = 0V, V_{DD} = V_{HH} = 5V±10%, Topr = −30 ∼ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP. (*)	MAX.	UNIT
V_{HS}	Hysteresis Voltage (Schmitt Circuit Input)		-	0.5	-	V
I_{IN1}	Input Current (K_0, $\overline{RESET}$, TEST)	V_{DD}=V_{HH}=5.5V, V_{IN}=5.5V	-	-	20	μA
I_{IN2}	Input Current (R Port)	V_{DD}=5.5V, V_{IN}=5.5V	-	-	20	
I_{IL}	Current (**)	V_{DD}=5.5V, V_{IN}=0.4V	-	-	-2	mA
I_{LO}	Output Leakage Current (P, R Port)	V_{DD}=5.5V, V_{OUT}=5.5V	-	-	20	μA
V_{OH}	High Level Output Voltage (***)	V_{DD}=4.5V, I_{OH}=−400μA	2.4	-	-	V
V_{OL}	Low Level Output Voltage (Except X_{OUT})	V_{DD}=4.5V, I_{OL}=1.6mA	-	-	0.4	
I_{OL}	Low Level Output Current (P_1, P_2)	V_{DD}=5V, V_{OL}=1V	-	20	-	mA
I_{DD+IHH}	Supply Current	V_{DD}=V_{HH}=5.5V	-	70	150	mA
I_{HH1}	Supply Current (Memory stand-by)	V_{DD}=V_{SS}, V_{HH}=3.5V	-	5	10	

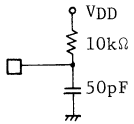
(*) TYP. values are at Topr=25°C, V_{DD}=V_{HH}=5V.
(**) Application terminals: $\overline{HOLD}$, $\overline{CALD}$, $\overline{SELC}$, I_7($\overline{HLT}$) ∼ I_0(DI_0).
(***) Application terminals: Control output terminal specific to evaluation.

A.C. CHARACTERISTICS (V_{SS} = 0V, V_{DD} = V_{HH} = 5V±10%, Topr = −30 ∼ 70°C)

(1)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	Unit
tcy	Instruction Cycle Time		1.9	-	40	μS
t_{SDH}	Shift Data Holding Time	(Note 1)	0.5tcy - 300	-	-	nS

(Note 1) $\overline{SCK}$, SO Terminal External Circuit

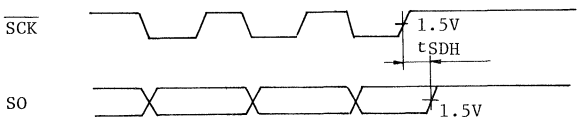


(2)

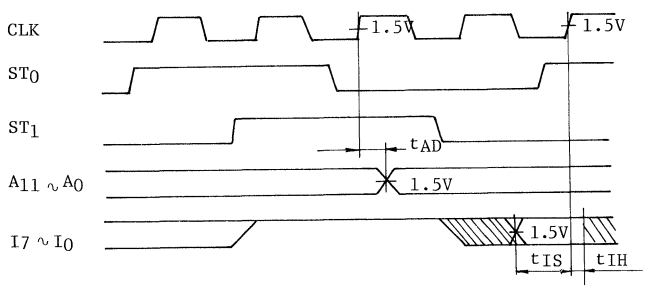
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tAD	Address Delay Time	C_L = 100pF	-	-	270	nS
tIS	Data Set-up Time	"	150	-	-	
tIH	Data Hold Time	"	50	-	-	

A.C. Timing Chart

(1) Serial Port (Completion of transmission)

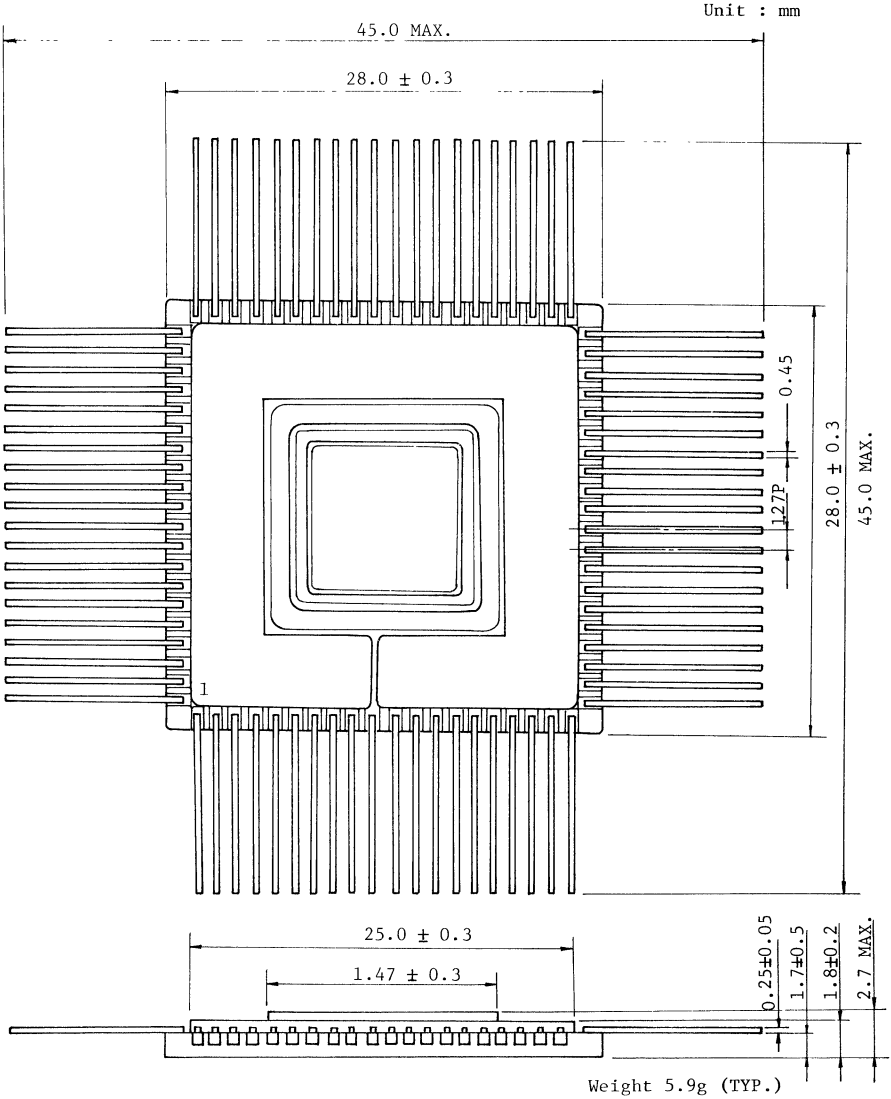


(2)



EXTERNAL DIMENSIONS

Unit : mm



Weight 5.9g (TYP.)

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4799C

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER TLCS-47(N) TMP4799C

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series designed for the general purpose use.

The TLCS-47 has veriously powerful functions in order to meet with the advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

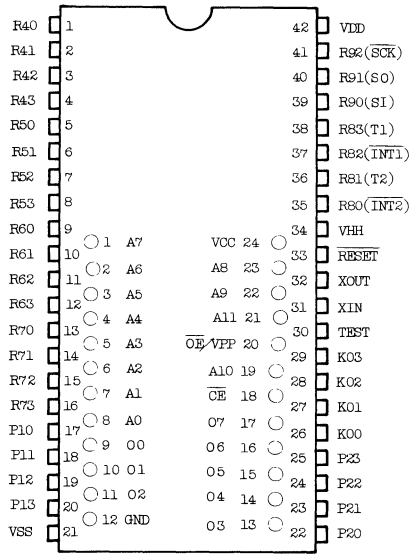
TMP4799C is the system development evaluator chip, which is equipped with a 24-pin socket which may directly mount the general purpose 32K EPROM (TMM2732D) on the top of the package. Therefore, when the program written in the 32K EPROM is mounted on the package, TMP4799C becomes pin compatible with TMP4740P, TMP4720P and can be used for developmental and operational check of the TLCS-47N application systems and programs. The former operates the same as the latter.

TMP4799C can be used within the range of a microcomputer for the TLCS-47N system as well as for mounting an equipment made on an experimental basis.

FEATURES

- General purpose 32K EPROM TMM2732D (equivalent to INTEL 2732) can be used.
- Compatible with TLCS-47N single chip microcomputer family TMP4740P/TMP4720P in pin.
- Compatible with TLCS-47 series in software.
- ROM 4,096 × 8 BIT (external), RAM 256 × 4 BIT (internal)

PIN CONNECTIONS (Top View)

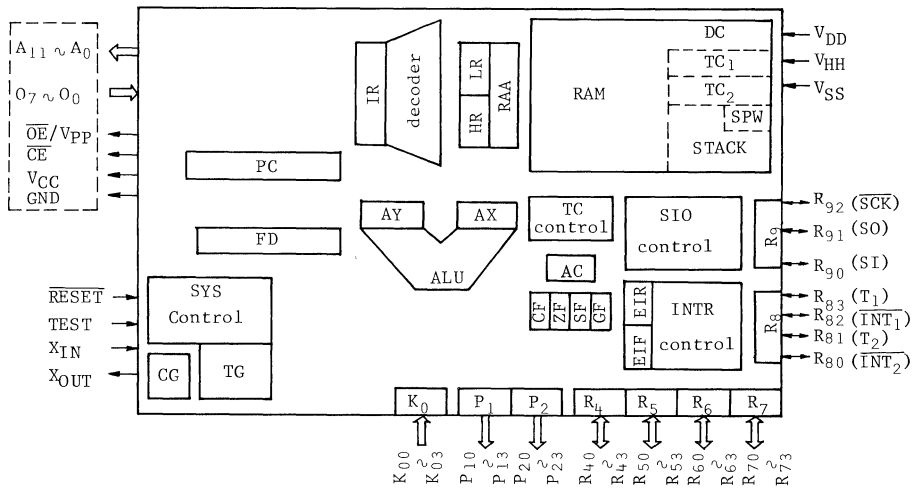


(NOTE) ○ Mark : Socket for TMM2732D

PIN NAMES AND PIN DESCRIPTIONS

Pin Names	No. of pins	Input/ Output	Functions	
$K_{03} \sim K_{00}$	4	Input	Input port	
$P_{13} \sim P_{10}$	4	Output	Output port (corresponding to PLA)	
$P_{23} \sim P_{20}$	4	Output	" (")	
$R_{43} \sim R_{40}$	4	I/O	I/O port	
$R_{53} \sim R_{50}$	4	I/O	"	
$R_{63} \sim R_{60}$	4	I/O	"	
$R_{73} \sim R_{70}$	4	I/O	"	
R_{83} (T_1)	1	I/O	I/O port or timer/counter input	
R_{82} ($\overline{INT_1}$)	1	I/O	" or interrupt input	
R_{81} (T_2)	1	I/O	" or timer/counter input	
R_{80} ($\overline{INT_2}$)	1	I/O	" or interrupt input	
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port	
R_{91} (SO)	1	I/O	" or serial output	
R_{90} (SI)	1	I/O	" or serial input	
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminals	
$\overline{RESET}$	1	Input	Initialize signal input	
TEST	1	Input	(Low level is input.)	
V_{DD}	1	Power supply	+5V	
V_{HH}	1	Power supply	+5V (Memory power supply)	
V_{SS}	1	Power supply	0V	
$A_{11} \sim A_0$	12	Output	Program memory address	Socket for TMM2732D
$O_7 \sim O_0$	8	Input	Program data input	
$\overline{OE}/V_{PP}$	1	Output	Output buffer control	
$\overline{CE}$	1	Output	Chip Enable (connected with V_{SS})	
V_{CC}	1	Power supply	+5V (connected with V_{DD})	
GND	1	Power supply	0V (connected with V_{SS})	

Note : $\overline{RESET}$ terminal has no built-in pull-up resistor as well as TEST terminal has no built-in pull-down resistor.



BLOCK NAMES AND DESCRIPTIONS

Block Names	Functions
PC	Program counter (12 bits)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Deta memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC	Data counter (12 bits, RAM area)
AX, AY	Temporary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG(CF,ZF,SF,GF)	Flags
K, P, R	Ports
INTR control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2-channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
SYS control	Generation of various internal control signals
CG, TG	Clock generator, timing generator

FUNCTIONAL DESCRIPTION

TMP4799C is the system development evaluator chip for the TLCS-47N. When the 32K EPROM (TMM2732D) in which the program is written is mounted on the package, it is possible to configurate a system equivalent to TMP4740P or TMP4720P.

The precautions for using TMP4799C are described.

1. Program Memory (ROM) and ROM address

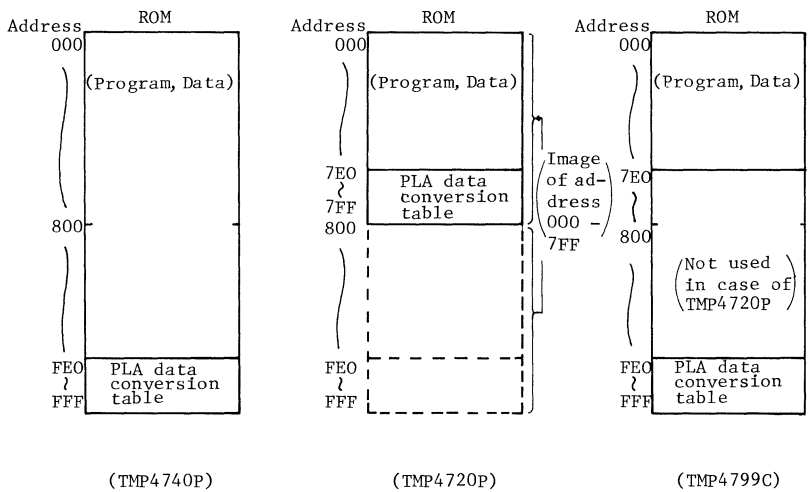
The precautions for using TMP4799C as an evaluator chip for TMP4720P are described.

TMP4720P contains a program memory with 2,048 × 8-bit (addresses 000 - 7FF) capacity. In case of TMP4720P, the PLA data conversion table must be located in addresses 7E0 - 7FF, because the MSB in the program counter is not decoded and there is no physical ROM in addresses 800 - FFF.

When TMP4799C is used with 32K EPROM, the program counter with 12-bit length is decoded and there is a program memory with 4,096 × 8-bit (addresses 000 - FFF) capacity. In case of TMP4799C, the PLA data conversion table is, therefore, located in addresses FE0 - FFF.

No precaution is required, when TMP4799C is used as an evaluator chip for TMP4740P. It is the former has the same address space as the latter.

Fig. 1.1 shows the ROM address space of TMP4740P, TMP4720P and TMP4799C.



2. Data Memory (RAM) and RAM address

The precautions for using TMP4799C as an evaluator chip for TMP4720P are described.

Data memory contained in TMP4720P has a 128 × 4-bit (addresses 00 - 7F) capacity, and the specific RAM address, which is used for the stack area, the data counter, etc., is located in addresses 40 - 7F. It is because the MSB of RAM address buffer register is not decoded and there is no physical RAM in addresses 80 - FF in TMP4720P.

In case of TMP4799C, the RAM address buffer register with 8-bit length is decoded and there is data memory with 256 × 4-bit (addresses 00 - FF) capacity. Then the specific RAM address area is located in addresses C0 - FF in TMP4799C, while it located in addresses 40 - 7F in TMP4720P. Further, it is necessary to pay attention to the addresses of the data memory in case of accessing the data in the specific RAM address area.

Fig. 2.1 shows the RAM address space of TMP4740P, TMP4720P and TMP4799C.

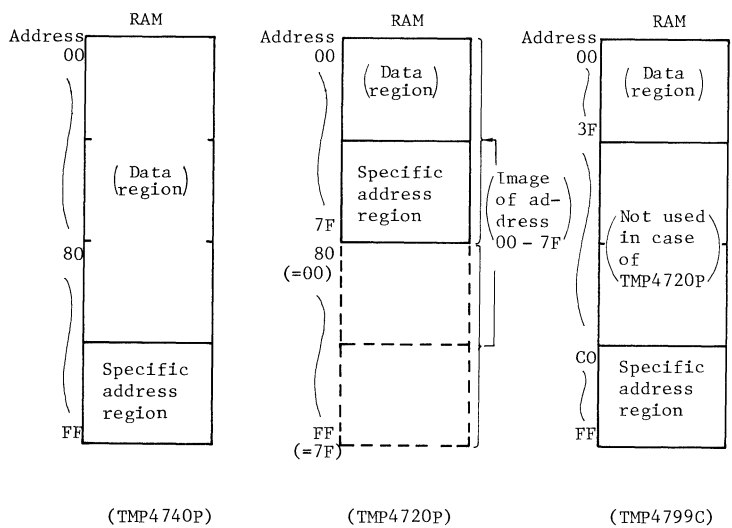
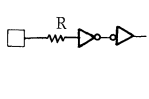
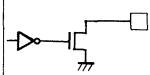
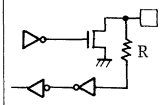
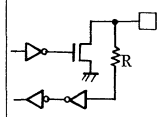
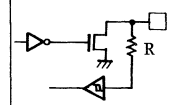


Fig. 2.1 RAM Capacity and Address

3. Input/Output circuit format

Fig. 3.1 shows the input/output circuit format of TMP4799C which is equivalent to "IOCODE AA" of TMP4740P and TMP4720P.

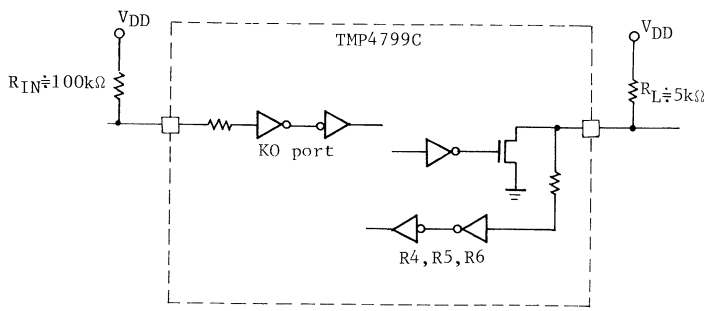
port / Circuit	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
I/O equivalent Circuit	 R=1kΩ (TYP.)		 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)
Remark	∘ High threshold input. ∘ No resistor is contained.	∘ Sink open-drain output. ∘ High output current. ∘ Output latch is initialized to the high level.	∘ Sink open-drain output. ∘ Output latch is initialized to the high level.	∘ Sink open-drain output. ∘ Output latch is initialized to the high level.	∘ Schmitt circuit input. ∘ Sink open-drain output. ∘ Output latch is initialized to the high level.

Note : TMP4799C does not contain the pull-up resister with $\overline{\text{RESET}}$ pin and does not contain the pull-down resister with TEST pin. It is necessary to provide $\overline{\text{RESET}}$ pin with the pull-up resister (≈ 300kΩ TYP.) and to provide TEST pin with the pull-down resister (≈ 70kΩ TYP.), respectively.

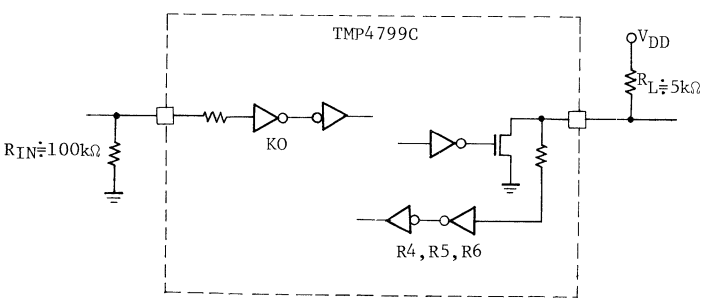
Fig. 3.1 Input/Output circuit format of TMP4799C

TMP4799C cannot be used as an evaluator chip for TMP4740P or TMP4720P which employs "IOCODE AH" or "IOCODE AI", because the output latches of R4, R5, R6 are initialized to the high level in the former and to the low level in the latter.

It is necessary to provide the pull-up or pull-down resisters with KO port and to provide the pull-up resisters with R4, R5, R6 ports when TMP4799C is used as an evaluator chip for TMP4740P or TMP4720P which employs "IOCODE AE" or "IOCODE AF", respectively. Fig. 3.2 shows the examples of the external circuitries.



(1) The external circuitry for TMP4799C
(equivalent to "IOCODE AE")



(2) The external circuitry for TMP4799C
(equivalent to "IOCODE AF")

Fig. 3.2 Example of external circuitry for TMP4799C

TENTATIVE

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS}=0V)

SYMBOL	ITEM	RATING	UNITS
V_{DD}	Supply Voltage	-0.5 ∿ 7	V
V_{HH}			
V_{IN}	Input Voltage	-0.5 ∿ 7	V
V_{OUT1}	Output Voltage(Except Open Drain Port)	-0.5 ∿ 7	V
V_{OUT2}	Output Voltage (Open Drain Port)	-0.5 ∿ 10	
I_{OUT}	Output Current (P_1, P_2)	30	mA
P_D	Power Dissipation (T_{opr}=70°C)	1	W
T_{sol}	Soldering Temperature • Time	260 (10 sec)	°C
T_{stg}	Storage Temperature	-55 ∿ 125	
T_{opr}	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS}=0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNITS
T_{opr}	Operating Temperature		-30	70	°C
V_{DD}	Supply Voltage		4.5	5.5	V
V_{HH}					
V_{HH1}	Supply Voltage (Memory Stand-by)		3.5	5.5	
V_{IH1}	High Level Input Voltage ($R_4 \sim R_7$)		2.2	V_{DD}	V
V_{IH2}	High Level Input Voltage (Except $R_4 \sim R_7$)		3	V_{DD}	
V_{IL1}	Low Level Input Voltage (Except K_0)		0	0.8	
V_{IL2}	Low Level Input Voltage (K_0)		0	1.2	
f_C	Clock Frequency		0.4	4.2	MHz
t_{WCH}	High Level Clock Pulse Width (Note 1)	V_{IN}=V_{IH}	80	-	nS
t_{WCL}	Low Level Clock Pulse Width (Note 1)	V_{IN}=V_{IL}	80	-	

(Note 1) For external clock operation.

D.C. CHARACTERISTICS (V_{SS}=0V, $V_{DD}=V_{HH}$=5V±10%, T_{opr}=-30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP. (*)	MAX.	UNIT
V_{HS}	Hysteresis Voltage (schmitt Circuit Input)		-	0.5	-	V
I_{IN1}	Input Current (K_0, $\overline{RESET}$, TEST)	$V_{DD}=V_{HH}$=5.5V, V_{IN}=5.5V	-	-	20	μA
I_{IN2}	Input Current (R Port)	V_{DD}=5.5V, V_{IN}=5.5V	-	-	20	μA
I_{IL}	Current (**)	V_{DD}=5.5V, V_{IN}=0.4V	-	-	-2	mA
I_{LO}	Output Leakage Current (P, R Port)	V_{DD}=5.5V, V_{OUT}=5.5V	-	-	20	μA
V_{OH}	High Level Output Voltage (***)	V_{DD}=4.5V, I_{OH}=-400μA	2.4	-	-	V
V_{OL}	Low Level Output Voltage (Except X_{OUT})	V_{DD}=4.5V, I_{OL}=1.6mA	-	-	0.4	V
I_{OL}	Low Level Output Current (P_1, P_2)	V_{DD}=5V, V_{OL}=1V	-	20	-	mA
$I_{DD}+I_{HH}$	Supply Current	$V_{DD}=V_{HH}$=5.5V	-	70	150	mA
I_{HH1}	Supply Current (Memory stand-by)	$V_{DD}=V_{SS}$, V_{HH}=3.5V	-	5	10	mA

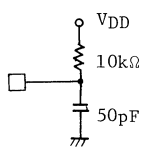
(*) TYP. values are at T_{opr}=25°C, $V_{DD}=V_{HH}$=5V.
(**) Application terminals : O_7 ∿ O_0
(***) Application terminals : A_{11} ∿ A_0, $\overline{OE}/V_{PP}$

A.C. CHARACTERISTICS (V_{SS}=0V, $V_{DD}=V_{HH}$=5V±10%, T_{opr}=-30 ∿ 70°C)

(1)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
t_{cy}	Instruction Cycle Time		1.9	-	20	μS
t_{SDH}	Shift Data Holding Time	(Note 1)	0.5tcy-300	-	-	nS

(Note 1) $\overline{SCK}$, SO Terminal External Circuit

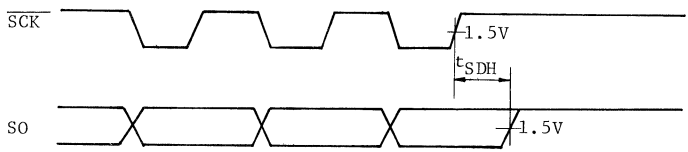


(2)

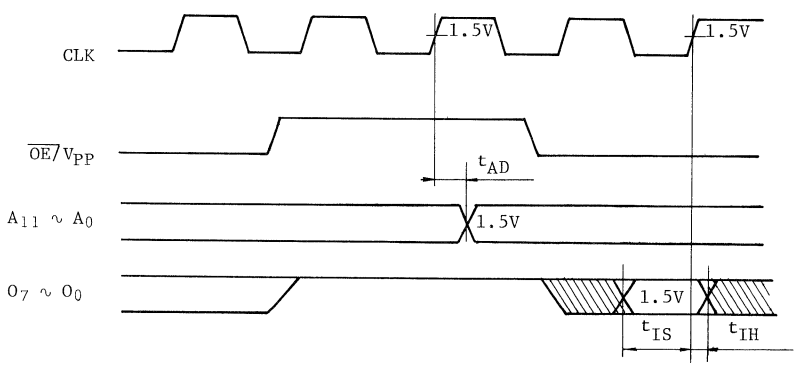
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
t_{AD}	Address Delay Time	C_L=100pF	-	-	270	nS
t_{IS}	Data Set-up Time	"	150	-	-	
t_{IH}	Data Hold Time	"	50	-	-	

A.C. Timing Chart

(1) Serial Port (Completion of transmission)

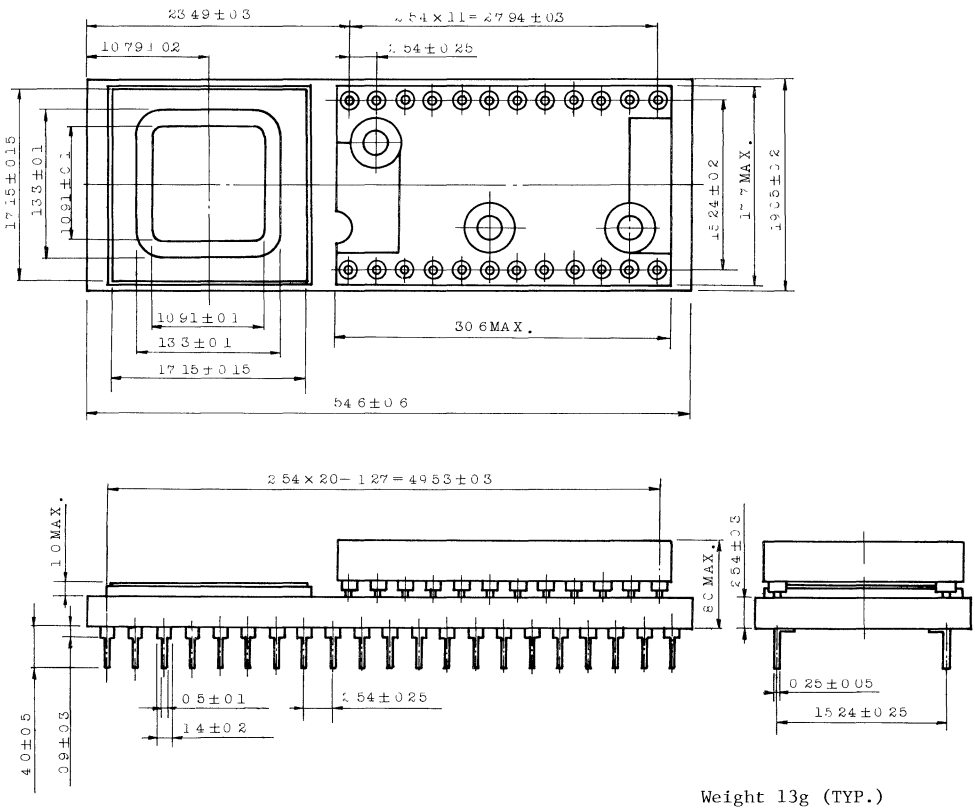


(2)



EXTERNAL DIMENSION VIEW

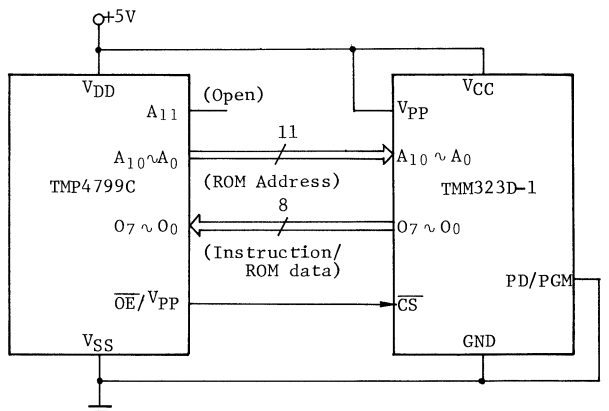
Unit in mm



Weight 13g (TYP.)

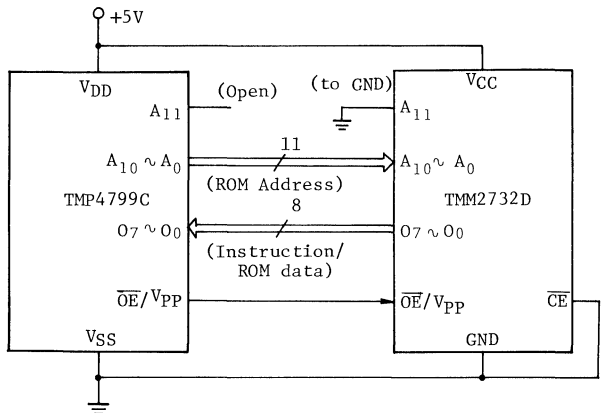
CONNECTION OF PROGRAM MEMORY

When TMP4799C operates as the evaluator chip for TMP4720P, TMM323D-1 (2,048 × 8 bit) can be used as the program memory. The connecting method of a program memory is shown below.



Pin Names of TMM323D-1	Pin Names of TMM2732D	Connection
PD/PGM	$\overline{CE}$	No change
$\overline{CS}$	$\overline{OE}/V_{PP}$	No change
V_{PP}	A_{11}	A_{11} is open. V_{PP} is connected to V_{DD}.

TMP4799C used with TMM2732D, in which the program is written in the range of addresses 000 - 7FF, operates the same as TMP4720P when the connecting method shown below is adopted.



A_{11} of TMP4799C is open.

A_{11} of TMM2732D is connected to V_{SS}.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4746N

UNDER DEVELOPMENT

TMP4746N

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER

The TMP4746N is a 4-bit single chip microcomputer with built-in ROM, RAM, input/output port, divider, timer/counter, and serial port.

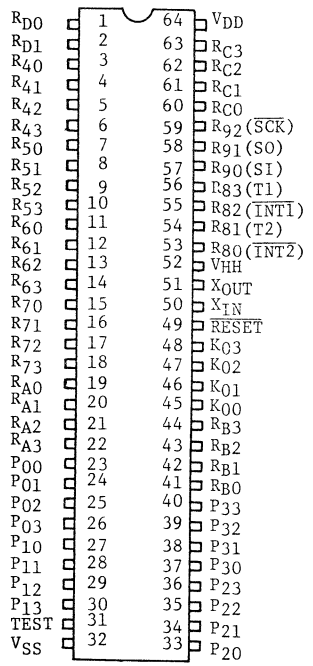
FEATURES

- ROM: 4,096 x 8 Bit
- RAM: 256 x 4 Bit
- Instruction execution time: 2 μs (at 4 MHz clock)
- Effective instruction set
  90 instructions, Software compatible in the series
- Subroutine nesting: Maximum 15 levels
- 6 interrupts (External: 2, Internal: 4)
  Independently latched control and multiple interrupt control
- Input/Output port (57 pins)

Input	1 port	4 pins
Output (corresponding to PLA)	2 ports	8 pins
Output	2 ports	8 pins
I/O	8 ports	30 pins
I/O (Note)	2 ports	7 pins

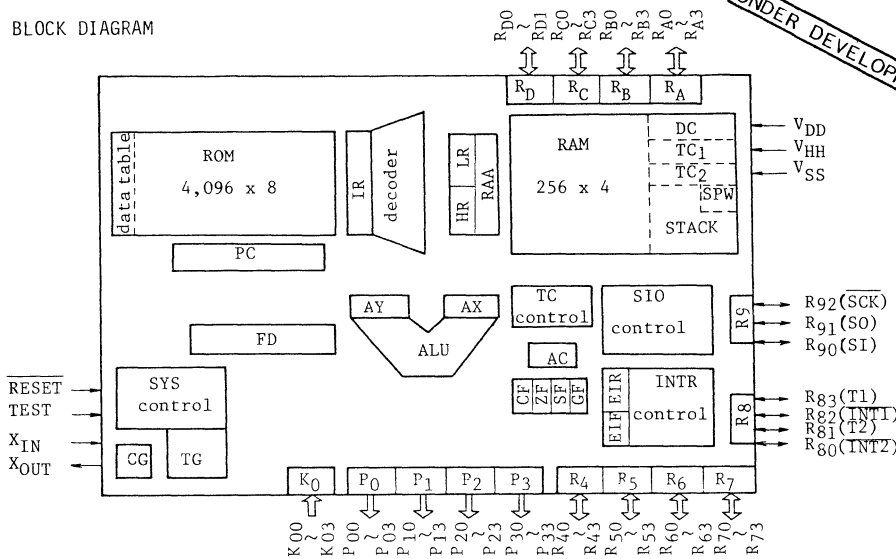
  Note: These I/O ports are also used for the interrupt input, timer/counter input, and serial port; therefore, it is programmably selectable for each application.
- PLA data converting function (Instruction)
  Output of data to output port (8-bit)
- Table look-up and table search function (Instruction)
  Table can be set up in the whole ROM area.
- 12-bit timer/counter (2 channels)
  Event counter, timer, and pulse width measurement mode is programmably selectable.
- Serial port with 4-bit buffer
  Receive/Transfer mode is programmably selectable.
  External/Internal clock and Leading/Trailing edge mode are programmably selectable.
- 18-stage divider (with 4-stage precaler)
  Frequency applied for timer interrupt of divider is programmably selectable.
- High output current (Output ports)
  TYP. 20mA x 8 bits, LED direct drive is available.
- Memory stand-by function: Battery backup is available.
- On chip oscillator
- TTL/CMOS Compatible
- +5V single power supply
- 64-pin DIL plastic shrunk package
- N-channel Si gate E/D MOS LSI

PIN CONNECTIONS (Top View)



TMP4746N BLOCK DIAGRAM

UNDER DEVELOPMENT



PIN NAMES AND PIN DESCRIPTION

Pin Name	No. of Pins	Input/Output	Functions
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (Corresponding to PLA, High current)
$P_{23} \sim P_{20}$	4	Output	" (" ")
$P_{03} \sim P_{00}$	4	Output	"
$P_{33} \sim P_{30}$	4	Output	"
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
$R_{A3} \sim R_{A0}$	4	I/O	"
$R_{B3} \sim R_{B0}$	4	I/O	"
$R_{C3} \sim R_{C0}$	4	I/O	"
$R_{D1} \sim R_{D0}$	2	I/O	"
R_{83} (T1)	1	I/O	I/O port or timer/counter input
R_{82} (INT1)	1	I/O	I/O port or interrupt input
R_{81} (T2)	1	I/O	I/O port or timer/counter input
R_{80} (INT2)	1	I/O	I/O port or interrupt input
R_{92} (SCK)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	I/O port or serial output
R_{90} (SI)	1	I/O	I/O port or serial input
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminals
RESET	1	Input	Initialize signal input
TEST	1	Input	(Low level is input.)
V_{DD}	1	Power supply	+5V
V_{HH}	1	Power supply	+5V (Memory power supply)
V_{SS}	1	Power supply	0V

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-47

CMOS DEVICES

July. 1 9 8 4

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP47C40P, TMP47C41P

TMP47C20P, TMP47C21P

SILICON MONOLITHIC SILICON GATE CMOS

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47C)
TMP47C40P, TMP47C20P, TMP47C41P, TMP47C21P

GENERAL DESCRIPTION

The TLCS-47 is the high speed and high performance, 4-bit single chip microcomputer series designed for the general purpose use.

The TLCS-47 has variously powerful functions in order to meet with advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

The TMP47C40P and TMP47C20P are the standard chips for the TLCS-47C. These chips are similar to each other, except memory capacity. And in the case of high breakdown voltage output type, production part's number is TMP47C41P or TMP47C21P. The TMP4700AC(NMOS) is an evaluator chip used for the system development.

Part No.	ROM (Bit)	RAM (Bit)
TMP47C40/41P	4,096 × 8	256 × 4
TMP47C20/21P	2,048 × 8	128 × 4
TMP4700AC	Externally provided (4,096 × 8)	256 × 4

FEATURES

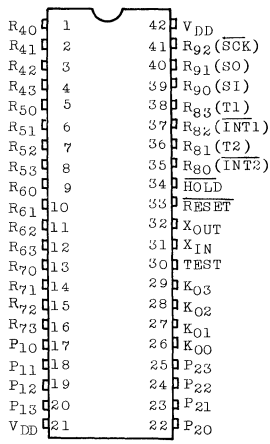
- 4-bit single chip microcomputer with built-in
  ROM, RAM, input/output port, divider, timer/counter, and serial port.
- Instruction execution time: 4μs (at 4 MHz clock)
- Effective instruction set
  90 instructions, software compatible in the series
- Subroutine nesting: Maximum 15 levels
- 6 interrupts (External: 2, Internal: 4)
  Independently latched control and multiple interrupt control
- Input/output port (35 pins)

Input	1 port	4 pins
Output (corresponding to PLA)	2 ports	8 pins
I/O	4 ports	16 pins
I/O (Note)	2 ports	7 pins

  Note: These I/O ports are also used for the interrupt input, timer/counter input, and serial port; therefore, it is programmably selectable for each application.

- PLA data converting function (Instruction)
  Output of data to output port (8-bit)
- Table look-up and table search function (Instruction)
  Table can be set up in the whole ROM area.
- 12-bit timer/counter (2 channels)
  Event counter, timer, and pulse width measurement mode is programmably selectable.
- Serial port with 4-bit buffer
  Receive/transfer mode is programmably selectable.
  External/internal clock and leading/trailing edge mode are programmably selectable.
- 18-stage divider (with 4-stage prescaler)
  Frequency applied for timer interrupt of divider is programmably selectable.
- High breakdown voltage output (20 pins)
  Maximum rating 42V, FL tube direct drive is available.
- Hold function
  Battery operation/condenser backup is available.
- On chip oscillator
- TTL/CMOS compatible
- +5V single power supply
- 42-pin DIL plastic package
- Si-gate CMOS LSI

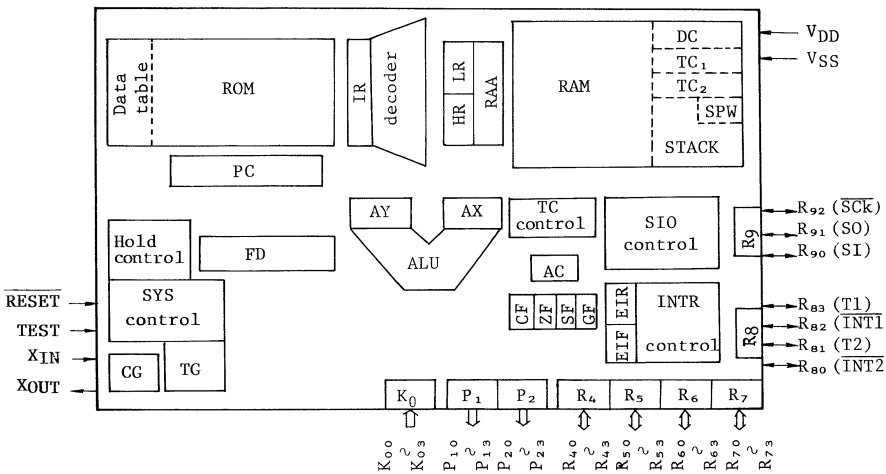
PIN CONNECTIONS (Top View)



PIN NAMES AND PIN DESCRIPTION

Pin Name	No. of pins	Input/Output	Function
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (Corresponding to PLA)
$P_{23} \sim P_{20}$	4	Output	" (")
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
R_{83} (T1)	1	I/O	I/O port or timer/counter input
R_{82} ($\overline{INT1}$)	1	I/O	I/O port or interrupt input
R_{81} (T2)	1	I/O	I/O port or timer/counter input
R_{80} ($\overline{INT2}$)	1	I/O	I/O port or interrupt input
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	I/O port or serial output
R_{90} (SI)	1	I/O	I/O port or serial input
X_{IN}, X_{OUT}	2	Input,Output	Resonator connection terminals
$\overline{RESET}$	1	Input	Initialize signal input
$\overline{HOLD}$	1	Input	Hold signal input
TEST	1	Input	(Low level is input.)
VDD	1	Power Supply	+5V
VSS	1	Power Supply	0V

BLOCK DIAGRAM



BLOCK NAMES AND DESCRIPTION

Block Name	Function
PC	Program counter (12 bits)
ROM	Program memory (including fixed data)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Data memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC, data table	Data counter (12 bits, RAM area), Data table (ROM area).
AX, AY	Temporary register of ALU input
ALU	Arighmetic and logic unit
AC	Accumulator
FLAG (CF,ZF,SF, GF)	Flags
K, P, R	Ports
INTR control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2 channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
HOLD control	Control for hold function
SYS control	Generation of various internal control signals
CG, TG	Clock generator, Timing generator

FUNCTIONAL DESCRIPTION

1. System Configuration

1. Program Counter (PC)
2. Program Memory (ROM)
3. H Register (HR), L Register (LR), RAM Address Buffer Register (RAA)
4. Data Memory (RAM)
   (1) Stack (STACK)
   (2) Stack Pointer Word (SPW)
   (3) Data Counter (DC)
5. ALU, Accumulator (AC)
6. Flags (FLAG)
7. Ports (PORT)
8. Interrupt Control Circuit (INTR)
9. Frequency Divider (FD)
10. Timer/Counter (TC_1, TC_2)
11. Serial Port (SIO)
12. Hold Control Circuit (HOLDC)

Concerning the above component parts, the configuration and functions of hardwares are described :

Hexadecimal notation is used for the description, charts, and tables in order to indicate the address and the like, without assigning identification symbols as far as it does not give rise to confusion.

The following names and symbols are used unconsciously.

(a)	CPU	Control Processing Unit except for the built-in peripheral circuitry, such as interrupt control circuit, timer/counter, and serial port.
(b)	CP	Clock pulse generated in the clock oscillator. It is called the "basic clock" or merely "clock".
(c)	fc	Indicates the frequency of the clock oscillator, namely, the frequency of the basic clock.
(d)	MSB/LSB	Indicates Most/Least Significant Bit.
(e)	F/F	Indicates Flip/Flop.

1.1 Program Counter (PC)

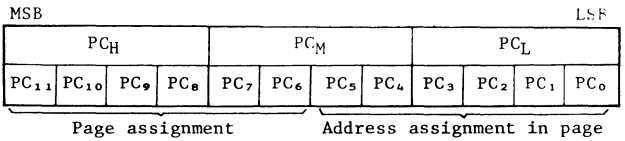
It is a 12-bit binary counter, and the contents of the program counter indicate the address of program memory in which the next instruction to be executed is stored.

The program counter generally gains increment at every instruction fetch by the number of tytes assigned to the instruction. However, when executing the branch and subroutine instructions or receiving the interrupt, the values specified by these instructions and operation are set.
Value "0" is specified by initializing the program counter.

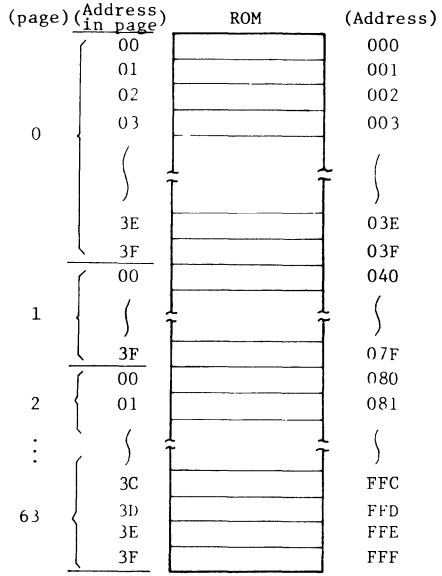
The page structure of program memory is made with 64 words per page. The TMP47C40P has 64 pages and the TMP47C20P 32 pages.

At the execution of (BSS a) instruction, the value assigned by the instruction is set in the lower 6 bits of the program counter when the branch condition is met. That is, the (BSS a) instruction is used as a branch of jump instruction within a page. If the (BSS a) instruction is stored in the last address of the page, the value in the higher 6 bits of the program counter indicates that the branch or jump instruction to the next page is executed.

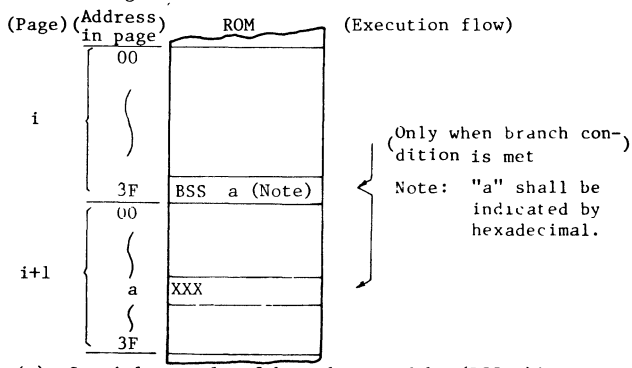
At the execution of (CALL a) instruction, the value specified by the instruction is set in the program counter after the previous contents of the program counter has been saved in the stack. Since 11 bits are of the address bit length which can be assigned by the instruction, the call address of subroutine should be in the range of addresses 000 - 7FF.



(a) Configuration of Program Counter



(b) Configuration of ROM



(c) Special example of branch caused by (BSS a)instrustion.

Fig.1.1.1 Program Counter and Program Memory (ROM)

1.2 Program Memory (ROM)

Processing programs and fixed data are stored in the program memory. The next instruction to be executed is read out from the address indicated by the contents of the program counter.

The fixed data stored in the program memory can be read by using the ROM data referring instruction or the PLA referring instruction. The ROM data referring instruction reads out the higher or lower 4-bit data of the fixed data stored in the address decided by the data counter [(LDH A, @DC+) and (LDL A, @DC) instruction respectively], and stores the data in the accululator. The PLA referring instruction (OUTB @HL) reads out the fixed data (8-bit) stored in the address decided by the contents of the data memory indicated by the contents of H and L registers as well as contents of the carry flag, and outputs the data to output ports (P2 · P1).

Addresses are individually assignged to the program memory and data memory, so that the fixed data in the ROM area cannot be directly read out by the address of the data memory.

Specific Addresses of Program Memory

The following addresses of the program memory are used for specific purposes. When not used for these purposes, the specific addresses can be used to store the processing programs and fixed data.

Specific Address	Specific Purposes
000 (001)	Start address by initialization
002 (003)	INT1 Interrupt vector address
004 (005)	ISIO Interrupt vector address
006 (007)	IOVF1 Interrupt vector address
008 (009)	IOVF2 Interrupt vector address
00A (00B)	ITMR Interrupt vector address
00C (00D)	INT2 Interrupt vector address
8n + 6 (n = 1 ∿15) 086 (Note)	Call address by instruction (CALLS a)
FE0 ∿ FFF	PLA data conversion table

Note : 086 (hexadecimal) = 134 (decimal)

Table 1.2.1 Specific Address of Program Memory

ROM CAPACITY

The TMP47C40P and TMP47C20P contain a program memory with 4,096 × 8-bit (addresses 000 - FFF) capacity and 2,048 x 8-bit (addresses 000 - 7FF) capacity, respectively. But the TMP47C20P contains a program counter with 12-bit length. Therefore, when one of addresses 800 - FFF is accessed in a program, the ROM data corresponding to addresses 000 - 7FF read out. It is because there is no physical ROM in addresses 800-FFF, but the MSB in the program counter is not decoded. For example, when the data located in address FF3 is output to a port by the PLA referring instruction on a program, the data located in address 7F3 is read out. In the TMP47C20P, the PLA data conversion table (addresses FE0 - FFF) is, therefore, located in addresses 7E0 - 7FF.

"0" [(NOP) instruction] is read out for the ROM data within the range of the built-in ROM capacity, if it is not specified by the user.

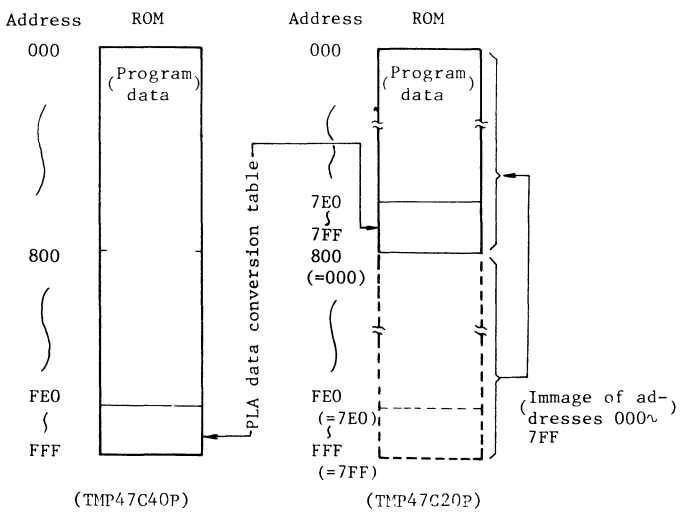


Fig. 1.2.1 ROM Capacity and Address

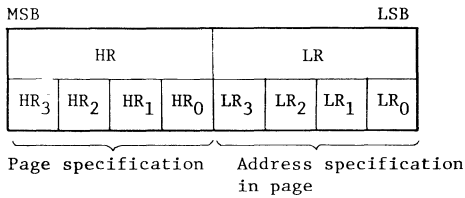
1.3 H Register (HR), L Register (LR), and RAM Address Buffer Register (RAA)

The H and L registers are 4-bit registers used as the data memory address pointers or general purpose registers.

The page structure of the data memory is based on 16 words per page. Pages are specified by H register, and addresses in page are done by L register, respectively. TMP47C40P has 16 pages and TMP47C20P 8 pages.

The L register is also used to specify the bits corresponding to pins $R_{73} \sim R_{40}$ of the I/O port when instructions (SET @L), (CLR @L), and (TEST @L), are executed.

The RAM address buffer register is a temporary register used to specify the address in the data memory, and serves as an input of the RAM address decoder. Normally, the data specified by the contents of the H and L registers or immediate data of an instruction is fed into the RAM address buffer register.



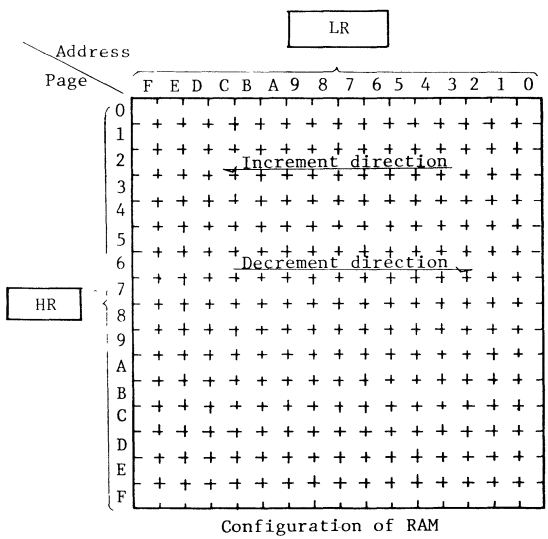


Fig. 1.3.1 H Register, L Register and Data Memory (RAM)

1.4 Data Memory (RAM)

The processing data of user are stored in the data memory. The data is read out or written in according to the address indicated by the contents of the RAM address buffer register.

Specific addresses of data memory

The data memory is also used for the following specific purposes. When it is not used for the respective purposes, the RAM of the corresponding address can be used to store the user processing data.

(1) Stack (STACK)
(2) Stack pointer word (SPW)
(3) Data counter (DC)
(4) Timer/Counter (TC1, TC2)

(1) Stack (STACK)

The stack, which is contained in the data memory (one level of the stack consists of 4-word RAM), is area to save the contents of the program counter (return address) and flag prior to jumping to the processing program at time of subroutine call or interrupt acceptance. To return from the processing program, (RET) instruction is used to restore the contents saved in the stack to the program counter, and (RETI) instruction is used to restore the contents saved in the stack to the program counter and flags.

The location of the stack to save/restore the contents is determined by the stack pointer word, which is automatically decremented after the saving operation, and incremented prior to the restoring operation.

(2) Stack Pointer Word (SPW)

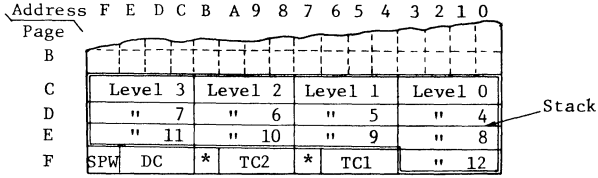
The address FF in the data memory is called a stack pointer word and decides the stack pointer. The stack is contained in the RAM, and accessed by the stack pointer.

The stack pointer is decided with the format shown in Fig. 1.4.1, but this address indicates the lower RAM address in each level of the stack.

Values "E" - "0" can be assigned for the stack pointer word, so that the maximum of 15 nesting levels are available for the stack. However, when the timer/counter mentioned following is used, the level containing the RAM address corresponding to the timer/counter cannot be used for the stack (value "F" is not assigned to the stack pointer word, because the stack contains the RAM address corresponding to the stack pointer word). The stack pointer word is automatically updated by the subroutine call or interrupt acceptance; however, it cannot exceed the allowable size of the stack for the system configuration.

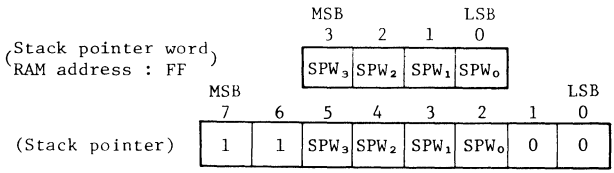
Since the stack pointer word is never initialized in terms of hardware, it is necessary to set it to the highest possible level of the stack in the user's initialization prpgram. For instance, it is set to "C" level when the two channels of timer/counter are used.

Note: The "level" indicates the depth of the nesting in the stack as well as the location of the next available stack. That is, it represents the contents of the stack pointer word.

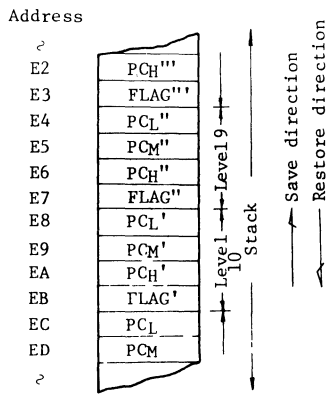


* : Can be used to store the user processing data

(a) Specific purposive map of RAM



(b) Stack pointer and stack pointer word



(c) Structure of stack

Fig. 1.4.1 Specific Address and Stack of Data Memory

(3) Data Counter (DC)

Data counter is a 12-bit binary counter used to specify the address when the data table in the ROM area is referred (ROM data referring instruction).

The RAM address with 4-bit unit is allocated to the data counter, so that the initial value setting and the content reading of the data counter can be executed by the RAM manipulative instructions.

	MSB		LSB
	D		C
(Data Counter)	DC_H	DC_M	DC_L
(RAM Address)	(FE)	(FD)	(FC)

Fig. 1.4.2 Data Counter and RAM Address

(4) Timer/Counter (TC1, TC2)

The two channels of 12-bit timer/counter are built-in, and the RAM address with 4-bit unit is allocated to the timer/counter, so that the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulative instructions.

When the timer/counter 1 is not used, the stack lower from level 13 can be used. When both of the timer/counter 1 and 2 are not used, the stack lower from level 14 can be used.

	MSB		LSB
	TC1		
(Timer/Counter 1)	$TC1_H$	$TC1_M$	$TC1_L$
(RAM Address)	(F6)	(F5)	(F4)

	MSB		LSB
	TC2		
(Timer/Counter 2)	$TC2_H$	$TC2_M$	$TC2_L$
(RAM Address)	(FA)	(F9)	(F8)

Fig. 1.4.3 Timer/Counter and RAM Address

(5) Page 0 in Data Memory

Page 0 in the data memory (addresses 00 - 0F) is effectively used as a flag or pointer in a user's program.

RAM Capacity

Data memory contained in TMP47C40P has a 256 × 4-bit (addresses 00 - FF) capacity, and that contained in TMP47C20P has a 128 × 4-bit (addresses 00 - 7F) capacity.

Since the TMP47C20P also has the RAM address buffer register of 8-bit length, there is no physical RAM in addresses 80 - FF in the TMP47C20P. However, the RAM equivalent to addresses 00 - 7F are referred when addresses 80 - FF are accessed in a program, because the MSB of RAM address buffer register is not decoded. That is, the specific RAM address is distributed to C0 - FF in a program, but the RAM equivalent to addresses 40 - 7F are assigned in the TMP47C20P.

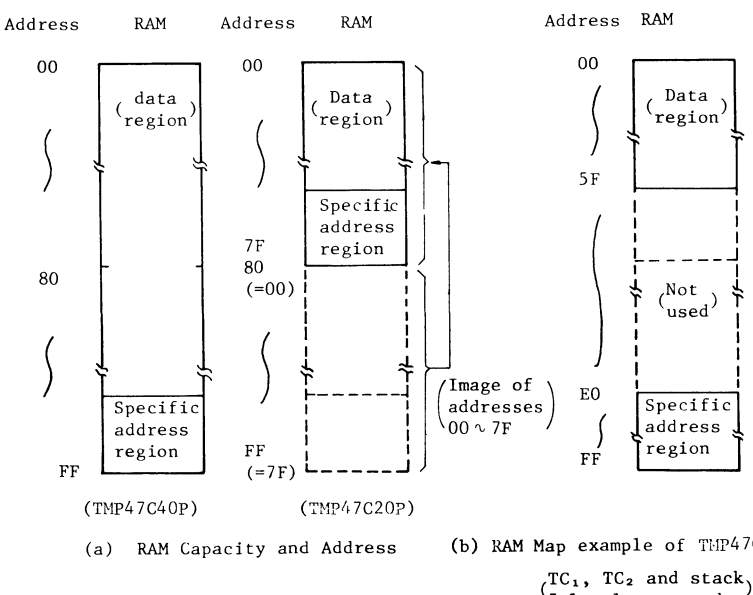


Fig. 1.4.4 RAM Capacity and Address

1.5 ALU, Accumulator (AC)

The ALU is a circuit used for various arithmetic and logical operation for 4-bit binary data. It performs the operation designated by the instruction, and outputs the 4-bit result, carry (C), and zero detection signal (Z).

The accumulator is a 4-bit register to use a source operand for the arithmetic operation, and in which the result is stored.

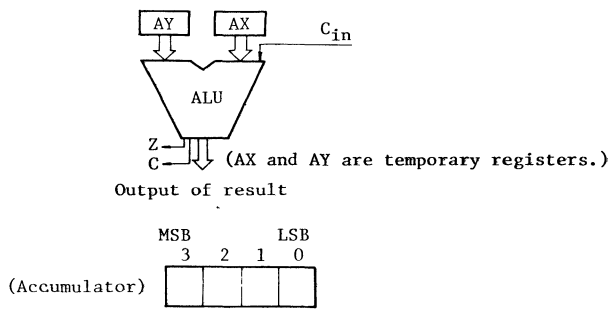


Fig. 1.5.1 ALU, Accumulator

Detection of operating condition

Output C from the ALU indicates the carry output from the most significant position in the addition operation.
However, the subtraction is executed with the addition of the 2's complement, so that output C in the subraction operation indicates the "non-borrow" from the most significant position (i.e., in case of non-borrow, C = "1"). Accordingly, borrow (B) can be represented with "$\overline{C}$".

Output Z indicates the zero detection signal to which "1" is applied when all of the 4-bit data transferred to accumulator or output of the ALU are cleared to zero.

Example (4-bit operation)

(a)	4 + 5 = 9	(C = 0, Z = 0)
(b)	7 + 9 = 0	(C = 1, Z = 1)
(c)	3 - 1 = 2	(B = 0, Z = 0)
(d)	2 - 2 = 0	(B = 0, Z = 1)
(e)	6 - 8 = -2 or E	(B = 1, Z = 0)

Note : B = $\overline{C}$ is indicated.

1.6 Flag (FLAG)

Flag is a 4-bit register used to store the condition of arithmetic operation, and of which the set/reset conditions are specified by the instruction. The flag consisting of CF, ZF, SF, and GF is saved in the stack when the interrupt is accepted. By executing the (RETI) instruction, it is restored from the stack to the conditions immediately before the interrupt is accepted.

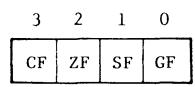


Fig. 1.6.1 Flag

(1) Carry Flag (CF)

This flag is used to hold the carry in the addition operation as an input to the ALU by the (ADDC A, @HL) instruction as well as to hold the non-borrow in the subtraction operation (the carry in the addition of the 2's complement) as an input to the ALU by the (SUBRC A, @HL) instruction. The rotate instruction makes the flag hold the data shifted out of the accumulator.

(2) Zero Flag (ZF)

This flag is stored the zero detection signal (Z) when the instruction designate to change. "1" is set if all 4 bits are cleared to zero by an arithmetic operation or data processing.

(3) Status Flag (SF)

This flag is set or reset according to the condition specified by the instruction. With the exception of particular cases, it is usually presented at every execution of an instruction, and holds the contents of the result during execution of the next instruction. It is normally set to "1", but is reset to "0" for a time under the certain condition (it varies according to the instruction, for examples, when the result is zero, when carry occurs in the addition, or when borrow occurs in the subtraction, the flag is reset).

The status flag is referred to as branch condition in a branch instruction. The memory location is branched when this flag is set to "1"; therefore, normally the branch instruction can be required as "unconditional jump instruction". On the contrary, the instruction becomes a "conditional instruction" if it is executed immediately after loading the instruction to set/reset the status flag according to the condition determined by some previous instruction.

The status flag is initialized to "1" at initialization, and is also set to "1" after the contents have been saved in the stack when the interrupt is accepted. The contents saved in the stack is restored by the (RETI) instruction.

(4) General Flag (GF)

This is a single-bit general purpose flag, being set or reset, and also used in a test by a program. This can be used for any purpose in the user program.

1.7 Port (PORT)

Data transfer to/from the external circuitry, and command/status/data transfer between the built-in periferal circuitry are carried out by the input/output instructions.

(a) Input/Output port : Data transfer to/from external circuitry.

(b) Command/data output : Control of circuitry of built-in peripheral circuitry, and output of data.

(c) Status/data input : Input of status signal(Note) and data from the built-in peripheral circuitry.

Note : Status signal is provided from serial port and hold control circuit, and is different from the status flag (SF).

To transfer the data or to control the circuitry, each port or register is selected by designating the address (Port address) by input/output operational instructions (13 instructions) in the same way as the memory.
The port address is composed of 5 bits (addresses 0 - 31).
The address to be accessed differs according to a instruction.
By way of caution, the port address space is independent of the program memory address space and the data memory address space.

Every output port contains a latch in order to hold the output data. Since every input port is operated without latching, it is desired to externally hold the data to be input from the external devices till the data is completely read out, or to read the data several times to confirm the contents.

The details to specify the input/output circuit format of ports and initialization of the output latch are 3.6 (2) Input/Output Circuit Format.

Port address	Symbol (Input/ Output)	Port, Register (Input/Output)	Input/Output Instructions						
			IN %P, A IN %P,@HL	OUT A ,%P OUT@HL,%P	OUT#K,%P	OUTB @HL	SET%P,b CLR%P,b	TEST %P,b TESTP%P,b	SET @L CLR @L TEST @L
00	IP00/OP00	K_0 Input port / P_1 Output	0					0	
01	IP01/OP01	P_1 Output latch/ port	0	0	0		0	0	
02	IP02/OP02	P_2 " /P_2 "	0	0	0		0	0	
03	IP03/OP03	———							
04	IP04/OP04	R_4 I/O port	0	0	0		0	0	0
05	IP05/OP05	R_5 "	0	0	0		0	0	0
06	IP06/OP06	R_6 "	0	0	0		0	0	0
07	IP07/OP07	R_7 "	0	0	0		0	0	0
08	IP08/OP08	R_8 "	0	0	0		0	0	
09	IP09/OP09	R_9 "	0	0	0		0	0	
0A	IP0A/OP0A	———							
0B	IP0B/OP0B	———	(*) Serial buffer register (Reception)						
0C	IP0C/OP0C	———	(**) Serial buffer register (Transmission)						
0D	IP0D/OP0D	———							
0E	IP0E/OP0E	Status input/ ———	0					0	
0F	IP0F/OP0F	(*) / (**)	0	0	0				
10	/OP10	/ (a)							
11	/OP11	$/P_2 \cdot P_1$ output port (8-bit output)				0			
12	/OP12	/ ———							
13	/OP13	/ ———							
14	/OP14	/ ———							
15	/OP15	/ ———	(a) Hold control						
16	/OP16	/ ———	(b) Control with timer interrupt of divider						
17	/OP17	/ ———	(c) Timer/Counter 1 control						
18	/OP18	/ ———							
19	/OP19	/ (b)	0						
1A	/OP1A	/ ———	(d) Timer/Counter 2 control						
1B	/OP1B	/ ———	(e) Serial port control						
1C	/OP1C	/ (c)	0						
1D	/OP1D	/ (d)	0						
1E	/OP1E	/ ———							
1F	/OP1F	/ (e)	0						

Note 1: Inputs (IP10 - IP1F) of port addresses 10 - 1F remain undefined.
Note 2: Port addresses with "——" mark are reserved addresses and cannot be used at user's program.
Note 3: OP11 is automatically accessed by (OUTB @HL) instruction, but cannot be done by the instructions other than this one.

Table 1.7.1 Port Address Allocation and Input/Output Instructions

(1) K_0 ($K_{03} \sim K_{00}$) Port

This is a 4-bit port used for input.

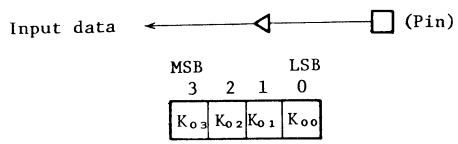


Fig. 1.7.1 K_0 Port

(2) P_1 ($P_{13} \sim P_{10}$), P_2 ($P_{23} \sim P_{20}$) Port

These ports are 4-bit ports with a latch used for output. The latch data can be read by the instruction.

These two ports can independently access by specifying port addresses IPO1/OPO1, and IPO2/OPO2. In addition, they can output 8-bit data by the (OUTB @HL) instruction.

PLA data conversion

A hardware PLA is not contained in the system; however, the function equivalent to it can be performed by access to the PLA data conversion table provided in the RCM by use of the (CUTB @HL) instruction.

The PLA referring instruction (OUTB @HL) : This instruction reads out the 8-bit data stored in the program memory, whose address is determined by the contents of the data memory indicated by the contents of the H and L registers as well as the contents of the carry flag, and outputs the data to 8-bit ports P2 and P1. At this time OP11 is automatically selected as the port address.

Ports P1 and P2 are capable of reading the latch data by the instruction, so that the data output by the PLA referring instruction can be qualified or modified; that is, the convert pattern can be changed or the numbers of pattern will be increased.

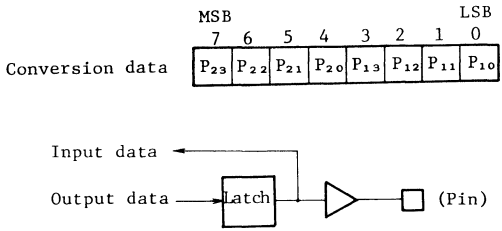


Fig. 1.7.2 P_1 and P_2 Ports

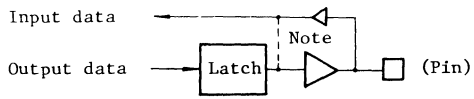
(3) $R_4(R_{43} \sim R_{40})$, $R_5(R_{53} \sim R_{50})$, $R_6(R_{63} \sim R_{60})$, $R_7(R_{73} \sim R_{70})$ Port

Each of these ports is a 4-bit I/O port with a latch. The latch should be set to "1" when the port is used as an input port. (But, these ports are only used to output ports with some input/output circuits.)

Pins $R_{73} - R_{40}$ can be used for bit scanning for set/reset and test according to the contents of the L register by executing the (SET @L), (CLR @L) and (TEST @L) instructions. Table 1.7.2 shows the pins corresponding to the contents of the L register.

L register 3 2 1 0	Corresponding Pin	L register 3 2 1 0	Corresponding Pin
0 0 0 0	R_{40}	1 0 0 0	R_{60}
0 0 0 1	R_{41}	1 0 0 1	R_{61}
0 0 1 0	R_{42}	1 0 1 0	R_{62}
0 0 1 1	R_{43}	1 0 1 1	R_{63}
0 1 0 0	R_{50}	1 1 0 0	R_{70}
0 1 0 1	R_{51}	1 1 0 1	R_{71}
0 1 1 0	R_{52}	1 1 1 0	R_{72}
0 1 1 1	R_{53}	1 1 1 1	R_{73}

Table 1.7.2 Correspondence of Individual Bits of L Register and I/O Port



Note : For bit set/reset of port, latch output serves as input data.

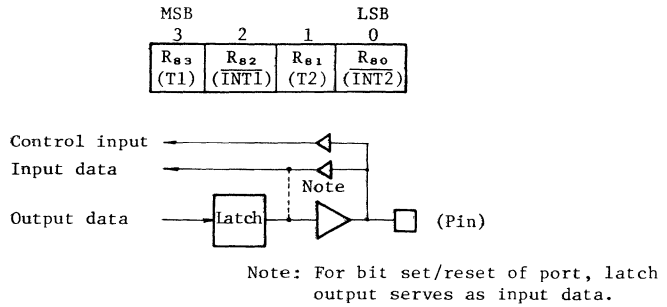
Fig. 1.7.3 $R_4 \sim R_7$ Ports

(4) R_8 ($R_{83} \sim R_{80}$) Port

This is a 4-bit I/O port with a latch. The latch should be set to "1" when the port is used as an input port.

It is a port common to external interrupt input or external timer/counter input. When it is driven by the external circuitry, such as external interrupt input or external timer/counter input, the latch must be set to "1". When it is used as normal I/O port, some measures, such as inhibition of the external interrupt input acceptance or disable of the mode depending on the external input of the timer/counter should be taken in a program.

(Note) When pin R_{82} ($\overline{INT1}$) is used as a port, INT1 interrupt request takes place because the falling edge of the pin input/output is detected (interrupt enabling master F/F is normally set to "1"). This causes the CPU to process a dummy interrupt acceptance [e.g. the (RET1) instruction only is executed]. When pin R_{80} ($\overline{INT2}$) is used, INT2 interrupt request also takes place in the same manner as the case of pin R_{82}, but the interrupt request is not accepted by merely resetting the LSB (EIR_0) of the enable interrupt register to "0" in advance. Therefore, the above processing is not required.

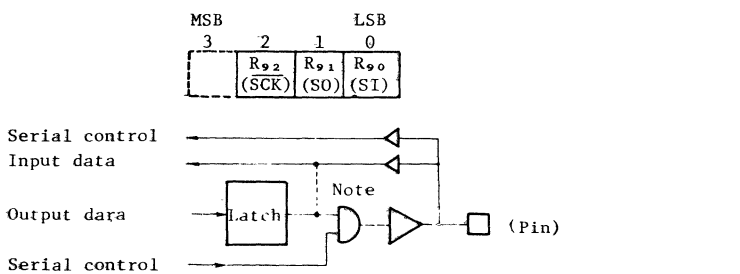


Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.4 R_8 Port

(5) R9($R_{92} \sim R_{90}$) Port

This is a 3-bit I/O port with a latch, and the latch must be set to "1" when it is used as input port.
The R9 port is also used as serial port. The latch must be set to "1" when R9 port is used as serial port. The port used as normal I/O port is not entirely influenced by disabling the serial port.
Pin R_{93} is not mounted in the port, but "1" is read by accessing to pin R_{93} in a program.



Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.5 R_9 Port

1.8 Interrupt control circuit (INTR)

Interrupt factors are composed of two from the external circuitry, and four from the internal circuitry. By setting the interrupt latch provided for each factor, an interrupt request is generated to the CPU. The interrupt latch is set when the edge of the input signal is detected.

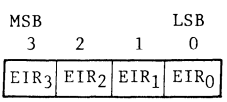
The interrupt request is not always accepted by the CPU if generated. It is not accepted till the priority in the six factors determined according to the hardware and the enabling/disabling control by the program become all affirmative.

In order to control enabling/disabling of interrupt by the program, an F/F (EIF) and a 4-bit register (EIR) are provided. By using these means, preferential acceptance of the interrupt factors by the program, and multiple interrupt control can be realized.

Factor		Priority according to hardware	Interrupt Latch	Enable condition according to program	Vector Address
External interrupt 1 (INT1)		(Higher) 1	$INTL_5$	(Note 1) EIF = 1	002
Internal interrupt	Serial Input/Output interrupt (ISIO)	2	$INTL_4$	$EIF \cdot EIR_3 = 1$	004
	Timer counter 1 Overflow interrupt (IOVF1)	3	$INTL_3$	$EIF \cdot EIR_2 = 1$	006
	Timer counter 2 Overflow interrupt (IOVF2)	4	$INTL_2$	(Note 2) $EIF \cdot EIR_1 = 1$	008
	Timer interrupt of divider (ITMR)	5	$INTL_1$	(Note 2) $EIF \cdot EIR_1 = 1$	00A
External interrupt 2 (INT2)		6 (Lower)	$INTL_0$	$EIF \cdot EIR_0 = 1$	00C

Interrupt enabling master F/F

Interrupt enabling register (EIR)



(Note 1) Since EIR register cannot make disabling of the INT1 interrupt, this interrupt is always accepted under the interrupt enabled condition (EIF = 1). Therefore, this should be used for the interrupt requiring the first priority such as emregency interrupt.

(Note 2) The given acceptance condition by the program is the same in IOVF2 and ITMR; accordingly, the action of these interrupts to the acceptance/inhibition control is the same.

Table 1.8.1 Interrupt Factors

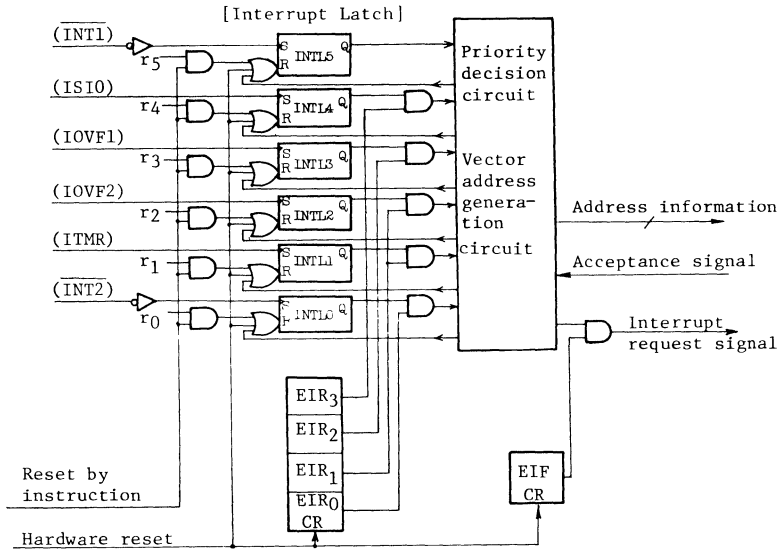
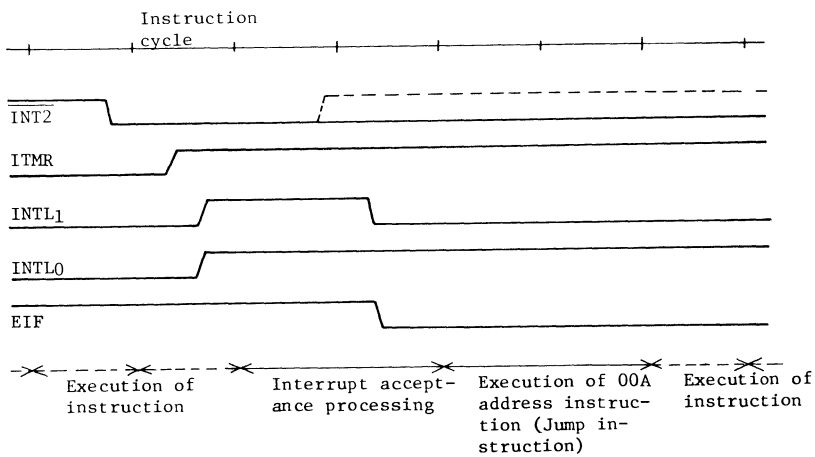


Fig. 1.8.1 Interrupt Control Circuit



Note: On the assumption that EIR_1 = 1, without other interrupt requests

Fig. 1.8.2 Interrupt Acceptance Timing Chart (Example)

(1) Interrupt processing

The interrupt request signal to be sent to the CPU is held by the interrupt latch till the request is accpeted or the latch is reset by the initialization operation or instruction.

The processing for the interrupt acceptance is performed within two instruction cycle time after the completion of the execution of instruction (after the completion of the timer/counter processing if it is required).

The following operations are performed by the interrupt service program.

① The contents of the program counter and flag are saved in the stack.

② The vector address is set to the program counter according to the interrupt factor.
(A jump instruction to each interrupt service program is usually stored in the program memory corresponding to the vector address.)

③ The status flag is set to "1".

④ The interrupt enabling master F/F is reset to "0" to inhibit the subsequent interrupt acceptance for a time.

⑤ The interrupt latch of the accepted interrupt factor is reset to "0".

⑥ The instruction stored in the vector address is executed.

The interrupt service program terminates after the execution of the (RETI) instruction.

The following operations are performed by the (RETI) instruction.

① The contents of the program counter and flag are restored out of the stack.

② The interrupt enabling master F/F is set to "1".

When the multiple interrupt is accepted, the interrupt enabling master F/F should be set by the instruction. At this time, the enabling/disabling for each interrupt factor can be changed by updating the interrupt enabling register by the (XCH A, EIR) instruction.

The program counter and flag are automatically saved/restored in the interrupt processing. However, if saving/restoring of the accumulator and other registers is necessary, it should be designated by a program.

(2) Interrupt control by program

EIF

This is an enabling interrupt master F/F. Interrupt is put in the interrupt acceptance enabling state by setting the EIF to "1". It is reset to "0" immediately after having accepted an interrupt to inhibit the subsequent interrupt acceptance for a time, but is set to "1" again by the (RETI) instruction after the completion of the interrupt service program to return the enable state again. And then the other interrupt can be received.

The EIF can be set/reset in a program by using the (EICLR IL,r) and (DICLR IL, r) instructions. It is reset to "0" at initialization operation.

EIR register

This is a 4-bit register used for selection/control of enabling/disabling of the interrupt acceptance in a program.

Read/write operation is performed by use of the (XCH A, EIR) instruction. It is set to "0" at the initialization operation.

Interrupt latch

The interrupt latches ($INTL_5$ - $INTL_0$) provided for each interrupt factor are set by the rising edge of the input signal if the interrupt is caused by the internal factors, and are set by the falling edge of the input pin if it is caused by the external factors. Then, interrupt request signal is sent to the CPU. The interrupt latch holds the signal till the interrupt request is accepted, and is reset to "0" immediately after the interrupt has been accepted.

Since the interrupt latch can be reset to "0" by the (EICLR IL, r), (DICLR IL, r) and (CLR IL, r) instructions, the interrupt request signal can be initialized by a program. The latch is reset to "0" at the initialization operation.

1.9 Frequency divider (FD)

The divider (FD_1 - FD_{18}) is made up 18-stage binary counter, and its output is used to generate various internal timing.

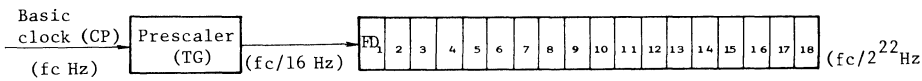
The basic clock (fc Hz) is divided into sixteen by the timing generator and input to the divider; therefore, the output frequency at the last stage is $fc/2^{22}$ Hz.
It is reset to "0" at the initialization operation.

Timer Interrupt of divider (ITMR)

The divider is capable of sending the interrupt request for a certain frequency. Four different frequencies can be selected for timer interrupt by instructions.

The command register is accessed as port address OP19, and is reset to "0" at time of the initialization.

The timer interrupt of divider is caused from the rising edge of the first output of the divider after the data has been written in the command resister.



(a) Structure of frequency divider

(Port address OP19)

MSB 3	2	1	LSB 0	
				(*: don't care)
*	0	*	*	: Disable
*	1	0	0	: Interrupt frequency fc/2^{10}Hz
*	1	0	1	: " fc/2^{11}Hz
*	1	1	0	: " fc/2^{12}Hz
*	1	1	1	: " fc/2^{13}Hz

Interrupt frequency (Hz)	For example, fc=4.194304MHz
fc/2^{10}	4,096 Hz
fc/2^{11}	2,048 Hz
fc/2^{12}	1,024 Hz
fc/2^{13}	512 Hz

(b) Command register

Fig. 1.9.1 Frequency Divider

1.10 Timer/Counter (TC_1, TC_2)

Two channels of 12-bit binary counter is contained to count time or event.

Since the RAM address with 4-bit unit is allocated to the timer/counter, the initial value setting and the content reading of the timer/counter can be executed by the RAM manipulated instructions.

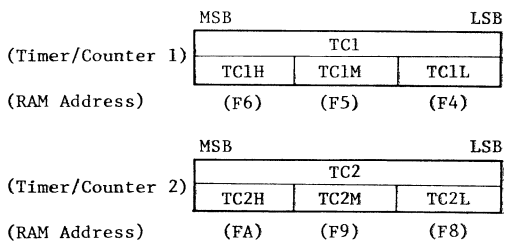
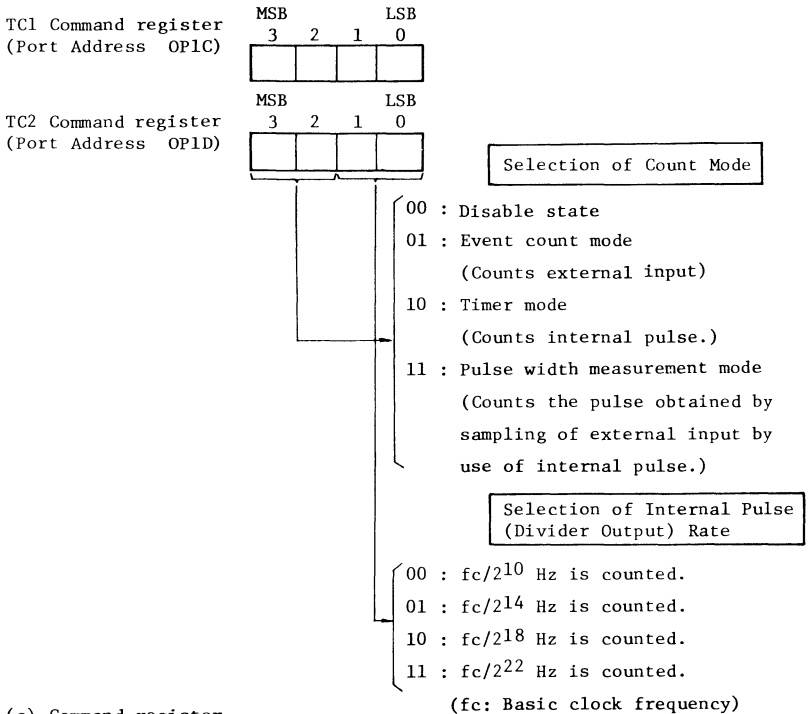


Fig. 1.10.1 Timer/Counter

(1) Timer/Counter Control

The timer/counter is controlled by the command specifying the operation mode. The command register for the timer/counter 1 and timer/counter 2 is accessed as port addresses OP1C and OP1D, respectively. It is reset to "0" at the initialization operation. The count operation is started from the first rising edge of the count pulse applied by setting the value (mode) to the command register.

When the timer/counter is not used, the RAM addresses corresponding to the timer/counter can be used to store the user processing data by selecting the "disable" state. In the timer mode, the external input pins can be used as I/O ports [R_{83} (T1), R_{81} (T2)].



(a) Command register

Internal Pulse Rate (Hz)	Max. Setting Time (SEC)	For example, fc=4.194304 MHz	
		Internal Pulse Rate (Hz)	Max. Setting Time (SEC)
$fc/2^{10}$	$2^{22}/fc$	4,096	1
$fc/2^{14}$	$2^{26}/fc$	256	16
$fc/2^{18}$	$2^{30}/fc$	16	256
$fc/2^{22}$	$2^{34}/fc$	1	4,096

(b) Selection of timer rate

Fig. 1.10.2 Control of Timer/Counter

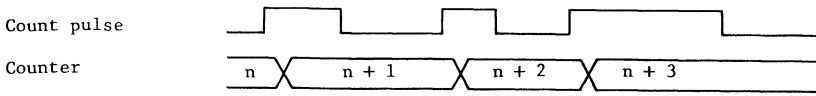
(2) Count Operation

When the rising edge of the count pulse is detected, the count latch is set to send a count request to the CPU.

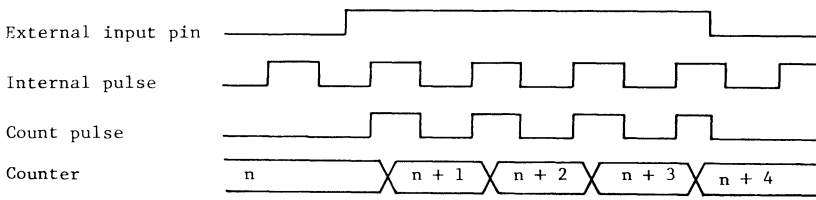
The count operation of the timer/counter is performed requiring one instruction cycle time after completion of the instruction execution. The execution of the next instruction and the acceptance of the interrupt request are kept waiting during the operation. When the count request is sent from the timer/counter 1 and 2, at the same time, the count request of the timer/counter 1 is preferentially executed.

The maximum frequency applied to the external input pin under the event counter mode is fc/64 Hz if one channel is used. When two channels are used, fc/64 Hz is applied to the timer/counter 1, and fc/80 Hz to the timer/counter 2.

In the timer mode, the maximum frequency is determined by a command.
The maximum frequency applied to the external input pin in the pulse width measurement mode should be the frequency level available for analyzing the count value in the program. Normally, the frequency sufficiently slower than the designated internal pulse rate is applied to the external input pin.



(a) Event counter/timer mode



(b) Pulse width measurement mode

Fig. 1.10.3 Mode and Count Value of Timer/Counter

Decrease in execution speed of instruction due to count operation

The CPU carries out the count operation requiring one instruction cycle time for the count request. Therefore, this causes the decrease in the apparent speed of instruction execution. Some examples are shown below :

(a) In the timer mode with count pulse rate of $fc/2^{10}$ Hz :

The count operation is inserted once every 64-instruction cycle time, so that the apparent speed is decreased by $1/63 \fallingdotseq 1.6\%$ instruction execution speed. For example, the apparent speed is 4.063μs to 4μs instruction execution speed.

(b) In the event count mode :

It depends on the count pulse rate applied to the external input pin. In the worst case, when the timer/counter 1 and 2 are operated at the same time with the maximum count pulse rate, the count operation is inserted once every 4-instruction cycle time for the timer/counter 1, and once every 5-instruction cycle time for the timer/counter 2.

The apparent speed of the instruction execution, therefore, decreases by 9/11≒.82%. The apparent speed is 7.28μs to 4μs instruction execution speed.

(3) Interrupt by overflow (IOVF1, IOVF2)

At the time when the overflow occurs, the timer/counter generates the interrupt request.
That is, the interrupt request is generated when the count value of FFF is changed to 000. The counting is continued after the interrupt request signal is generated. Assuming that the CPU provides the interrupt enabling state, and that the interrupt is accepted as soon as the overflow interrupt has been generated, the interrupt processing can be performed in the sequence illustrated in Fig. 1.10.4.

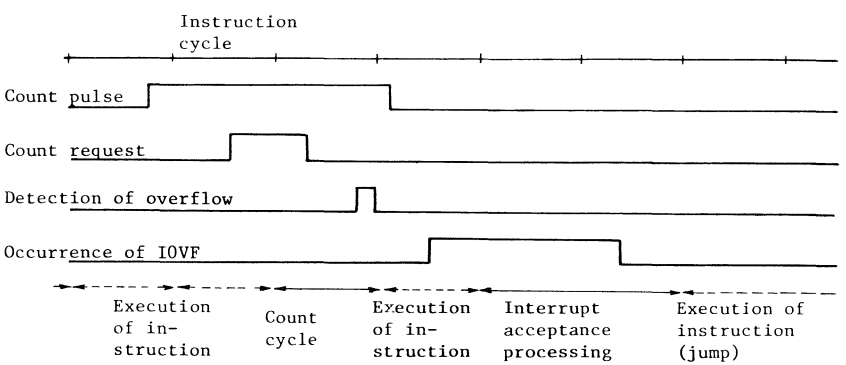


Fig. 1.10.4 Timing Chart of Timer/Counter in Interrupt by Overflow

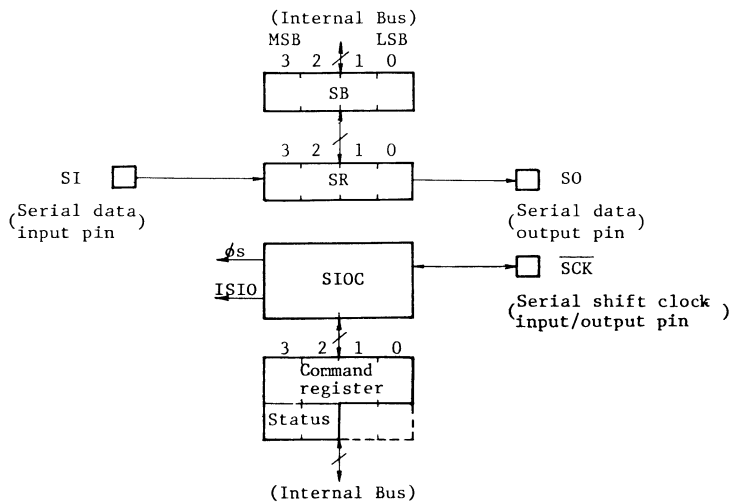
1.11 Serial Port (SIO)

A 4-bit serial port with a buffer is provided to transfer the serial data from/to the external circuitry. It is connected to the external circuitry through three pins [R92 ($\overline{SCK}$), R91 (SO), R90 (SI)]. Since these pins are also used as port R9, the output latch of the R9 port should be set to "1" when the serial port is used. When it is not used, the pins can be used as I/O port R9.

Pin R90 in the transmit mode and pin R91 in the receive mode are also available as I/O port pin.

(1) Circuit configuration

The serial port consists of a 4-bit shift register, a 4-bit buffer register, and its control circuit.



SR : 4-bit shift register
SB : 4-bit buffer register
SIOC : Serial port control circuit
φs : Internal shift clock
ISIO : Interrupt request

Fig. 1.11.1 Circuit Configuration of Serial Port

(2) Serial port control

The serial port operation is controlled by the command. The command register is accessed with port address OP1F, and reset to "0" at the initialization operation. The operation status can be informed through the status input, which is accessed with port address IPOE.

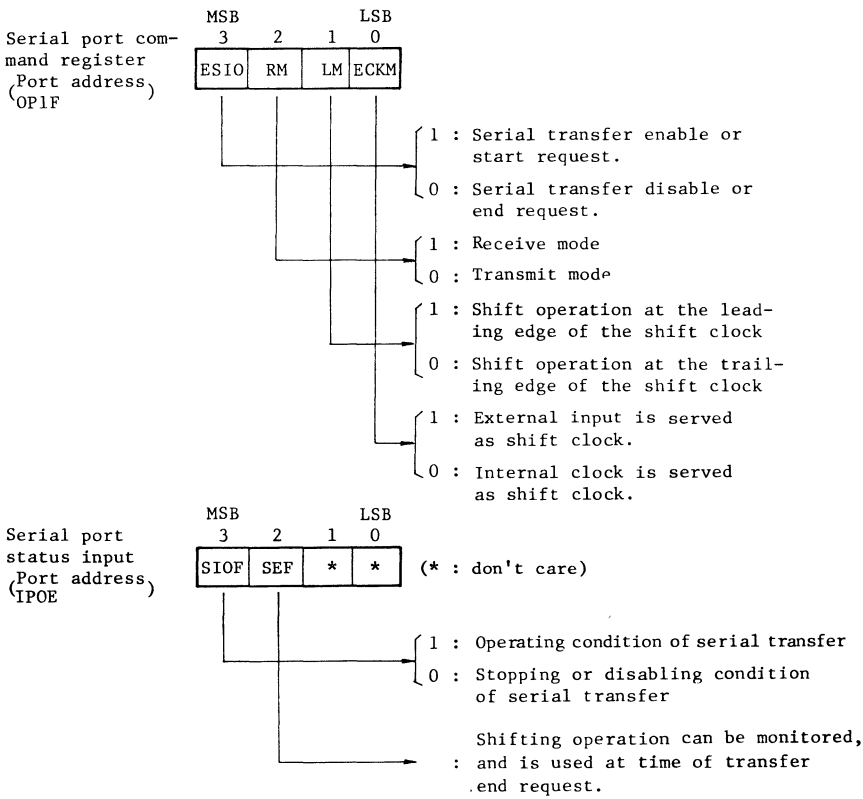


Fig. 1.11.2 Command Register, Status Input

(3) Shift clock (SCK)

The following shift clock modes can be selected by the contents of the command register.

(a) Clock source (External/internal mode)
(b) Shift edge of clock (Leading edge/trailing edge mode)

Internal clock mode

$fc/2^7$ Hz is used for the shift clock (when the basic clock frequency fc is 4.194304 MHz, the shift clock frequency is 32.768 kHz.). At this time, the clock is supplied to the external devices through the $\overline{SCK}$ pin. If the data setting (transmit mode) or the data reading (receive mode) rate by the program cannot follow the clock rate, the shift clock is automatically stopped and the next shift operation is suspended until the data processing is completed ("Wait"operation).

External clock mode

The shift operation is performed by the clock provided from the external circuitry since the $\overline{SCK}$ pin serves as an input.

Leading edge shift mode

Data is transmitted (transmit mode) or received (receive mode) at the leading edge of the $\overline{SCK}$ pin signal.

Trailing edge shift mode

Data is received (receive mode) at the trailing edge of the $\overline{SCK}$ pin signal.

The $\overline{SCK}$ pin must be set to the "high" level when the serial transfer is started. In the internal clock mode, the $\overline{SCK}$ pin is automatically set to the "high" level because it serves as an output.

(4) Operation mode

Selection of the following three transfer modes is available by changing the combination of the RM bit and LM bit of the command register.

RM (Bit 2)	LM (Bit 1)	ECKM (Bit 0)	Operation Mode
0	0	1/0	Can not be used
0	1	1/0	Transmit mode (Note) (External/Internal clock)
1	0	1/0	Receive(Trailing edge shift) mode (External/Internal clock)
1	1	1/0	Receive(Leading edge shift) mode (External/Internal clock)

(Note) Leading edge shift operation is performed.

Table 1.11.1 Operation Mode of Serial Port

In the transmit mode, the 4-bit data written to the buffer register from the CPU is shifted out by the shift register, and is output in the SO pin from the data of the LSB in sequence. The buffer register is accessed as the port address OPOF.

In the receive mode, the data to be input to the SI pin is shifted toward the LSB by the shift register in sequence, and is set in the buffer register after the 4-bit data has been received.

The CPU reads the contents of the buffer register, which is accessed as the port address IPOF.

Transmit mode

After this mode is set in the command register, the first transmit data (4-bit) is written in the buffer register (the data cannot be written in the buffer register, if the transmit mode is not set). Then the data can be transmitted by setting the ESIO (MSB of command register) to "1". The content of the buffer register is transferred to the shift register by the first shift clock, and the data in the LSB (D_0) is output to the SO pin. The buffer register then becomes empty, so that the interrupt (ISIO) requesting the next data takes place (buffer empty). After that, the remaining data (D_1 - D_3) is automatically shifted out by the shift register by one data at a shift clock. The control by use of a program is not necessary in this operation.

Data is written in the buffer register by outputting the next transmit data (4-bit) to the port address OPOF in the interrupt service program, and at the same time the interrupt request is reset to "0".

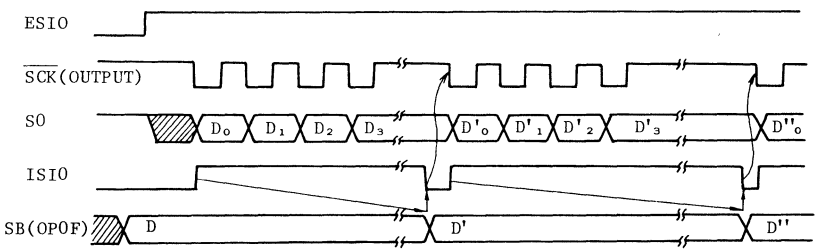
Internal clock operation

In case of $fc/2^7$ Hz internal clock operation, if the next data is not set in the buffer register (OPOF has not been accessed by the program) though the 4-bit data has been entirely shifted out, the shift clock automatically stops, and the wait operation is taking place until the data is set.

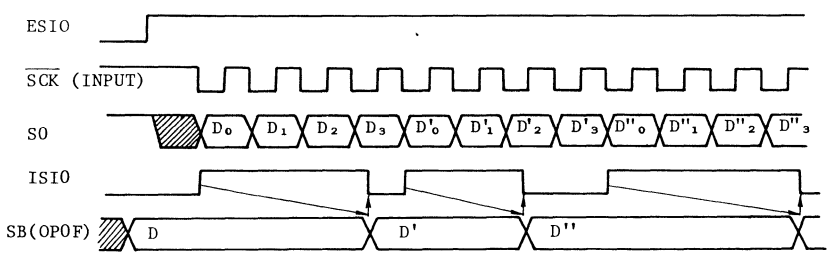
The maximum transmission rate is 31250 bit/sec. at the 4 MHz basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock provided from the external circuitry, the data should have been written in the buffer register before the next 4-bit data is shifted out. Therefore, the transfer rate is determined by the maximum time lag from the receipt of interrupt request (ISIO) to the writing of data in the buffer register by the interrupt service program.



(a) Internal clock operation (with wait operation)



(b) External clock operation

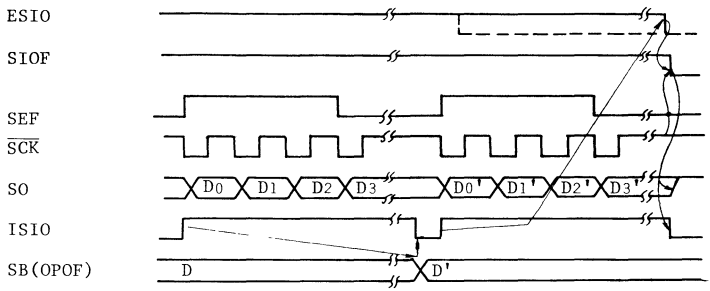
Fig. 1.11.3 Transmit Mode

Completion of transmission

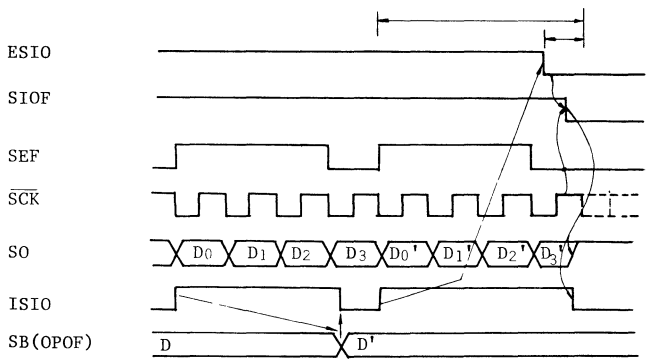
When the buffer register becomes empty, the interrupt occurs to request the next data. In case where the transmission is desired to be completed after the data is entirely transferred, the transmit operation can be stopped upon completion of transferring the current data shifted out, by resetting the ESIO to "0" without outputting the data. Whether or not the transfer operation is completed can be sensed in a program by the SIOF (MSB of the status input).

In the external clock operation, the ESIO must be reset to "0" before the next data is shifted out as in the data updating operation (however, the data is not updated when the operation is completed). When the wait operation have been already performed in the internal clock operation, the data transfer is terminated immediately after ESIO = 0.

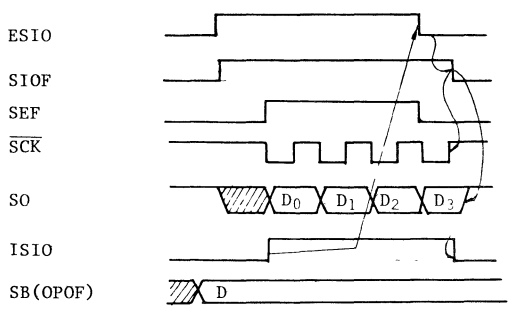
One word transfer can be terminated by ESIO = 0 in the interrupt service program on receipt of the interrupt caused by the buffer empty.



(a) Internal clock operation (with wait operation)



(b) External clock operation



(c) Completion at one-word transfer

Fig. 1.11.4 Completion of Transmission

Receive (trailing edge shift) mode

Data can be received by setting the receive mode in the command register as well as by setting the ESIO (MSB of command register) to "1". When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, interrupt (ISIO) takes place to request the data reading (buffer full). Since the shift register has been transferring the data to the buffer register, the shift operation is continued without waiting for the data being read.

When the data received from the port address IPOF is read in the interrupt service program, the interrupt request is reset. And then the next 4-bit data is transferred from the shift register to the buffer register if the buffer register has been full.

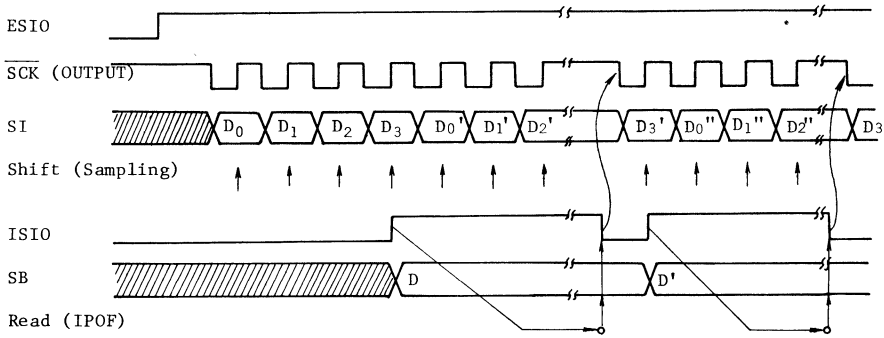
Internal clock operation

During the operation of the internal clock of $fc/2^7$Hz, if the next 4-bit data is not read out of the buffer register (the IPOF has not been accessed) in the program though the 4-bit data has been entirely input, the shift clock automatically stops, and the wait operation is taking place until the data is read out.

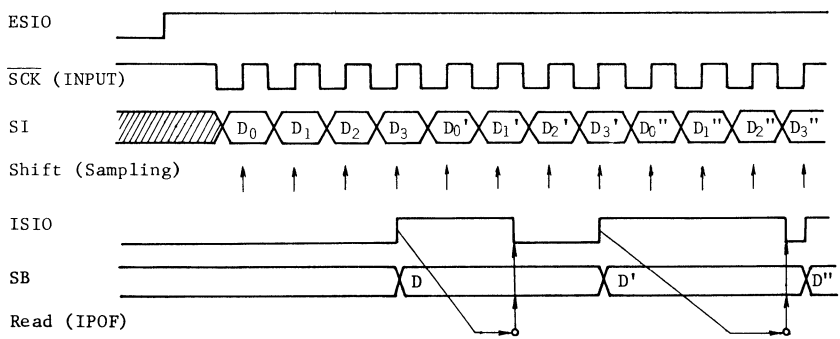
The maximum receiving rate is 31250 bit/sec at the 4 MHz basic clock.

External clock operation

Since the shift operation synchronizes entirely with the clock provided from the external circuitry, the current data should have been read by the instruction before the next 4-bit data is transferred to the buffer register. The transfer rate is, therefore, determined by the maximum time lag from the receipt of interrupt request (ISIO) to the read of the data in the buffer register by the interrupt service program.



(a) Internal clock operation (with wait operation)



(b) External clock operation

Fig. 1.11.5 Receive (trailing edge shift) Mode

Completion of receiving

When all of the data are read, the receiving of data can be completed upon termination of the current data transfer, by resetting the ESIO to "0".

Whether or not the data transmission is terminated can be sensed in a program by the SIOF (MSB of status input).

To complete the receive operation when the synchronization is desired between the serial transfer and interrupt service program (indicates data reading or completion of receiving), there are two ways according to the speed of shift clock.

The receive/transmit mode must be maintained without switching the mode until the last data is read out even if the completion of the data transfer is indicated; otherwise the contents of the buffer register will be lost.

(a) Sufficiently slow data transfer rate (external clock operation)

If the timing, operated by the external clock, is slow enough to reset the ESIO to "0" prior to the generation of the next shift clock, the ESIO can be reset to "0" in the interrupt service program which is loaded to read out the last data. Thereafter the last data is read.

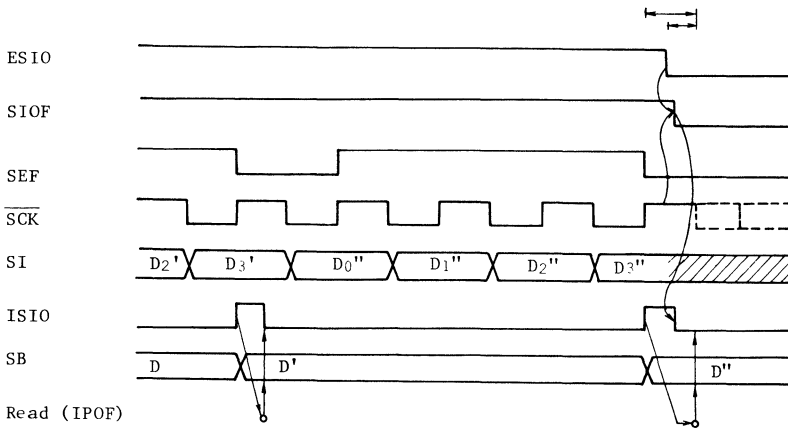


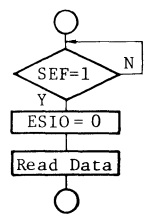
Fig. 1.11.6 Completion of Receiving (at slow transfer rate)

(b) Fast transfer rate

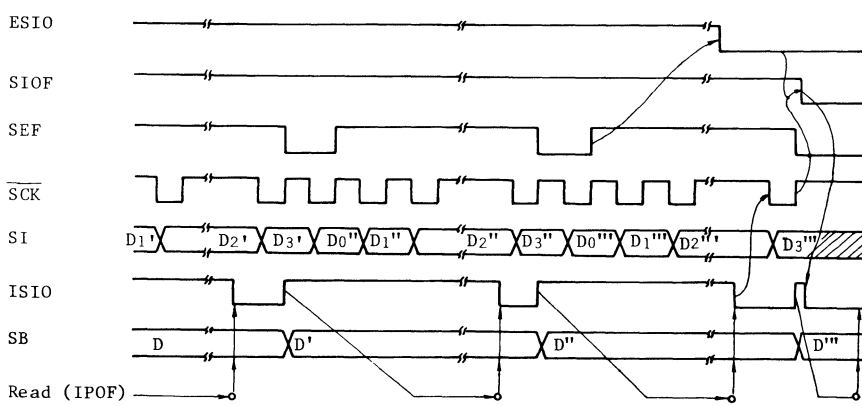
If the shift operation for the next data may start before the current data is read out by receipt of the interrupt request becuase the transfer rate is too fast, the interrupt service program which is loaded to read out the last data but one should be used to reset the ESIO to "0" after confirming that the SEF (bit 2 of status input) has been set to "1".

Thereafter, the data should be read. No operation is required to complete the data transfer in the interrupt service program for reading the last data.

The method mentioned above is usually taken for the internal clock operation. In the external clock operation, however, the reset of the ESIO and the read of data must be completed before the last data is transferred to the buffer register.



(a) Program sequence of receive end indication

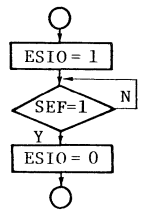


(b) Timing Chart
(in case of internal clock operation with wait operation)

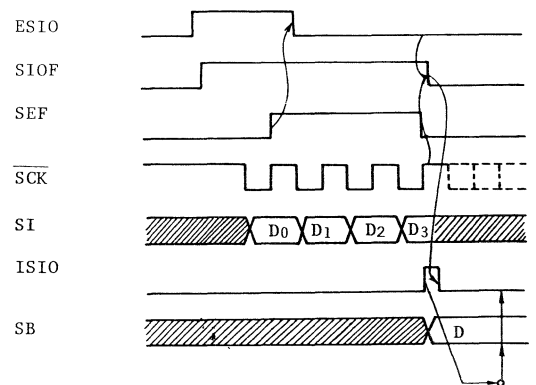
Fig. 1.11.7 Completion of Receiving (at fast transfer rate)

(c) One word transfer

The data receive operation starts after the ESIO is set to "1". Then, the ESIO is reset to "0" after confirming that the SEF status is set to "1". In this sequence, one interrupt casued by the buffer full takes place; therefore, the data should be read out by the service program.



(a) Program sequence of receiving start/end indication



(b) Timing Chart

Fig. 1.11.8 Receiving Start/Completion (at one word transfer)

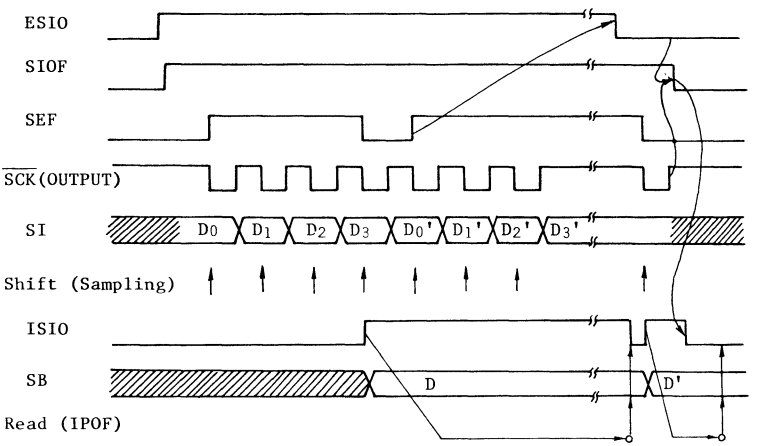
Receive (leading edge shift) mode

With this mode set in the command register, the data can be received by setting the ESIO (MSB of command register) to "1".

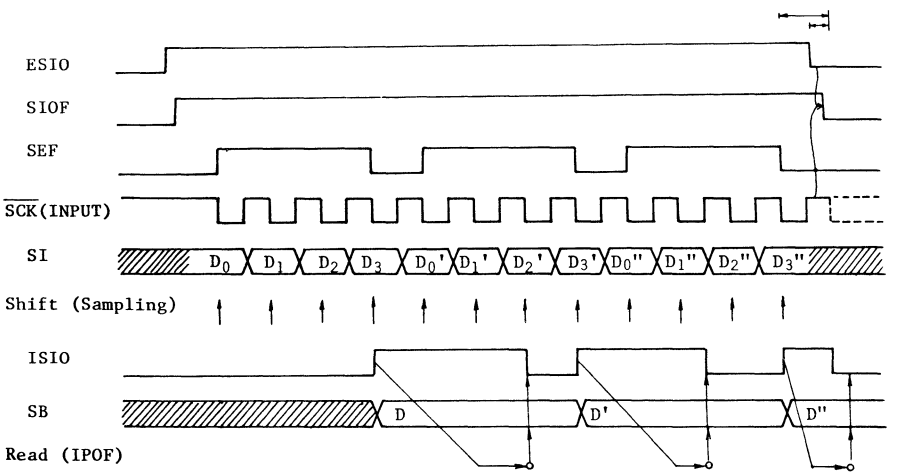
When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, the interrupt (ISIO) occurs to request the data reading (buffer full). Since the shift register is transferring the data to the buffer register, the shift operation has been continued without waiting for the data being read.

When the data received from the port address IPOF is read in the interrupt service program, the interrupt request is reset. And then the next 4-bit data is transferred from the shift register to the buffer register if the buffer register has been full.

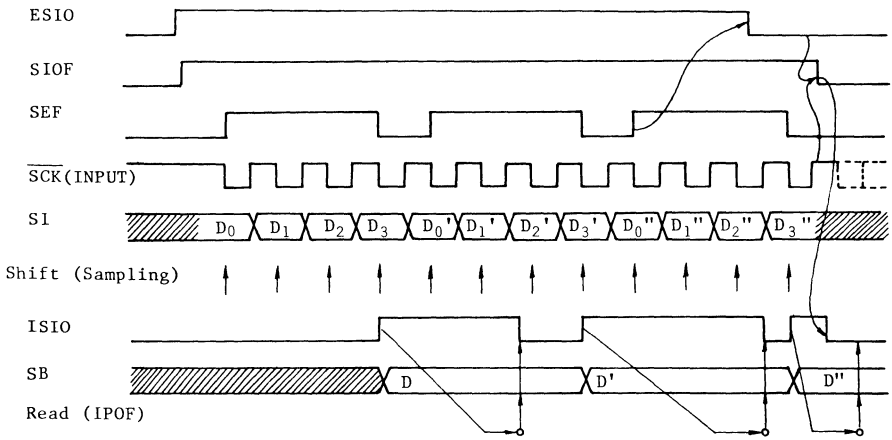
The basic operation in the receive (leading edge shift) mode is equivalent to that in the receive (trailing edge shift) mode except that the edge for the shift clock is different, and that at time of the transfer start, the first shifted data has been already input from the external circuitry before the first shift clock is applied to the data receipt. Timing charts are shown below.



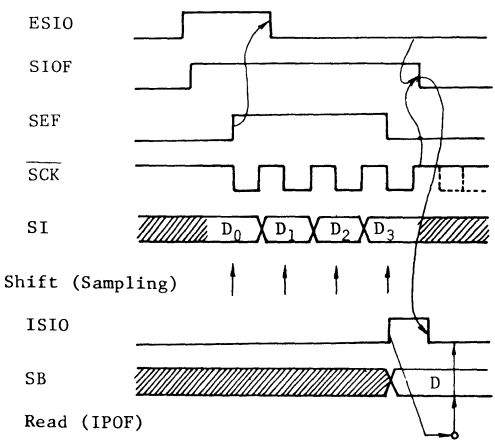
(a) Internal clock operation (with wait operation)



(b) External clock operation (at slow transfer rate)



(c) External clock operation (at fast transfer rate)



(d) One-word transfer

Fig. 1.11.9 Receive(Leading Edge Shift) Mode

1.12 Hold control circuit (HOLDC)

The hold function is the function that holds the status (contents of the data memory, program counter and other registers) immediately before the system operation is stopped at lower power consumption making the most of the features of CMOS. The hold function is controlled by the $\overline{\text{HOLD}}$ terminal and the command register.

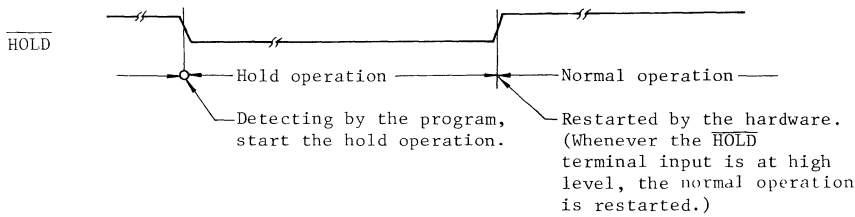
There are two operation modes for the hold operation as shown below. The designation for the hold operation start is made by the command in either mode.

(a) Backup mode

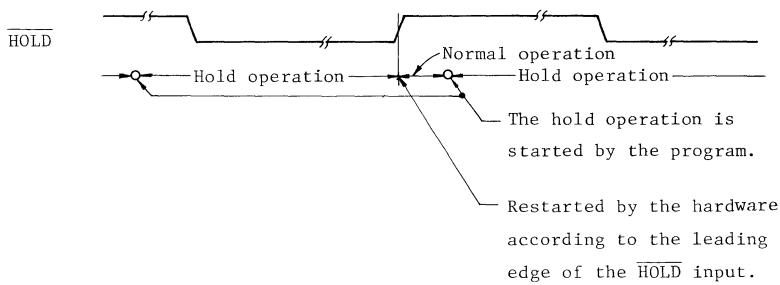
The $\overline{\text{HOLD}}$ terminal input controls the request/release of the hold operation. Namely, it is the state of the hold that the $\overline{\text{HOLD}}$ input is at the low level, and it is the state of the normal operations that the $\overline{\text{HOLD}}$ input is at the high level.
This mode is used for backup of the capacitor when the main power supply is cut off, backup of the battery for a long time, etc.

(b) Clock mode

When the hold operation is started by the command even when the $\overline{\text{HOLD}}$ terminal input is at the either levels, the hold operation is continued till the leading edge of the $\overline{\text{HOLD}}$ terminal input is detected. This mode is used when signals in constant cycle are applied to the $\overline{\text{HOLD}}$ terminal input in applications, for example, the clock or timer applications, where relatively short period program processings are repeatedly executed in constant cycle. For instance, the signal is applied to the $\overline{\text{HOLD}}$ input from a source of oscillation at low power consumption.



(a) Backup Mode



(b) Clock Mode

Fig. 1.12.1 Hold Modes

(1) Control of hold function

The hold operation is started by a command. The command register is accessed as the port address OP10, and operation mode selection, start control and set-up of warming-up time at time of restart are designated. Furthermore, it is possible to read the status of the $\overline{HOLD}$ terminal input from the status input. The status input is accessed as the port address IPOE.

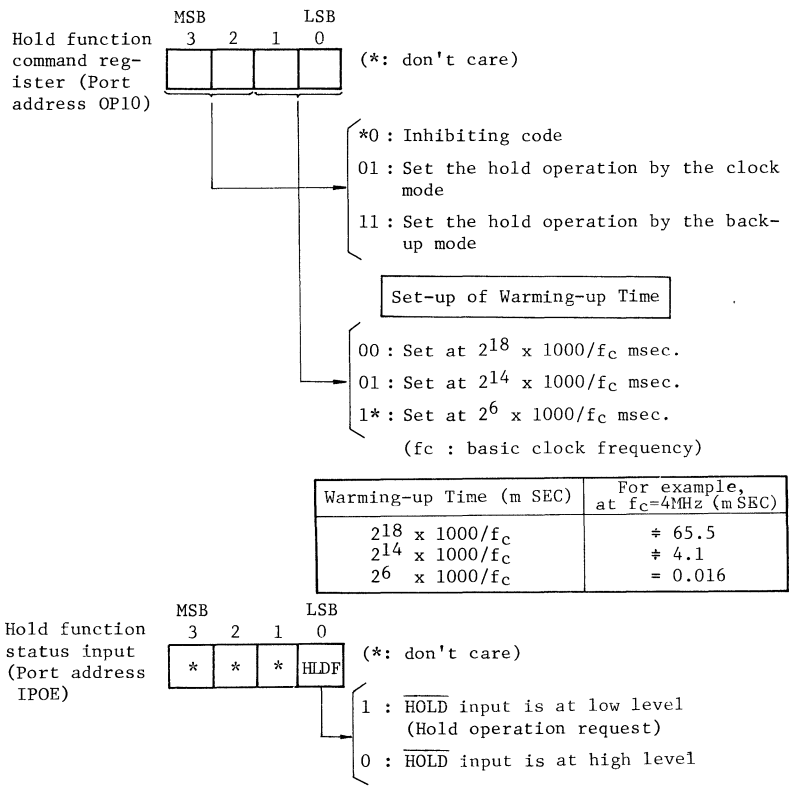


Fig. 1.12.2 Hold Function Control

(2) Hold operation

The hold operation holds the following state:

① The oscillation is stopped, and all internal operation are stopped.

② The frequency divider is reset to "0".

③ The data memory, registers and port latches are held at the state immediately before the hold operation is started. It is therefore necessary to program in advance for the processing of uninterruptable program or status of the output terminal.

④ The program counter holds address of the instruction following the instruction directing start of the hold operation.

The hold operation is started under either mode when data is set in a command register. In the case of the backup mode, it is therefore necessary to recognize the status of the $\overline{HOLD}$ input (the hold operation request) on the program. To do this, the following two methods are considered available:

(a) Test HLDF of the status input

(b) Apply the $\overline{HOLD}$ input to the $\overline{INT_1}$ input as an interrupt request.

The hold operation is released when the $\overline{HOLD}$ terminal input becomes the high level. That is, under the backup mode, the hold status is kept held as long as the $\overline{HOLD}$ input is at the low level. However, if the $\overline{HOLD}$ input is already at the high level when a command directing start of the hold operation is executed, the hold operation is not started but the restarting sequence is started. Under the clock mode, the hold operation is continued till the leading edge of the $\overline{HOLD}$ input is detected.

Further, in the hold operation, current consumption based on the oscillator and internal hardware is reduced, but current consumption based on the terminal interface (depending upon the external circuit and program) is not directly concerned with the hardware operation of the hold function, and it is therefore necessary to pay attention in designing system as well as interface circuits. When the input level is stable at the V_{DD}/V_{SS} level, current flows scarcely through the CMOS circuit. On the other hand, when the input level is floating from the V_{DD}/V_{SS} level (by about 0.3 ∿ 0.5V), current will flow through the CMOS circuit. Therefore, in a case where the signal level at the I/O port (the open drain output with an input port circuit connected) becomes the 3-state status when the output transistor is cut off, current flows through the input port resistor and it is therefore necessary to fix the signal level by pulling up, etc.

Restart from Hold

The restart from hold is performed in the following sequence.

① Oscillation is started.

② The internal operation is kept stopped for a period of warming-up time assigned by the hold setting command to prevent the malfunction due to unstable oscillation.

③ After the warming-up time has passed(Note), the normal operation is restarted by the instruction following the instruction directing the hold setting. Further, the divider starts to operate from the state where it has been reset to "0".

Note : Since the warming-up time is obtained from the value counted the basic clock by the divider, if oscillation frequency fluctuates at time of the restarting from the hold operation, the warming-up time shown in Fig. 1.12.2 may include errors. It is therefore necessary to regard the warming-up time as an approximate value.

The hold operation is released when the $\overline{\text{RESET}}$ terminal is set at the low level and the normal operation (the initialization operation) is immediately executed.

2. Instructions

The TLCS-47 series microcomputer is provided with 90 instructions, which are software compatible within the series. The instructions of the TLCS-47 series is consist of 1-byte instructions or 2-byte instructions. To classify them in terms of the execution time, there are 1-cycle instructions and 2-cycle instructions.

1-byte, 1-cycle instructions are mainly used in this series, and are arranged so as to improve the program efficiency.

1-byte	1-cycle instruction	40
1-byte	2-cycle instruction	11
2-byte	2-cycle instruction	39
	Total	90

(a) Classification by byte/cycle

Move instruction (Note 1)	22
Compare instruction	6
Arithmetic instruction	16
Logical instruction	9
Bit manipulation instruction	24
Input/Output instruction (Note 2)	6
Branch, subroutine instruction	6
Other instruction	1
Total	90

(Note 1) : Including ROM data referring instructions

(Note 2) : Including PLA referring instruction.

(b) Classification by function

Table 2.0.1 Classification of Instructions.

2.1 Description of symbols

The following symbols are used for describing the instructions in the following explanations.

Symbol	Description
AC	Accumulator
M[x]	Data memory (Address x)
HR	H register
LR	L register
P[p]	Port (Address p)
FLAG	Flag
CF	Carry flag
ZF	Zero flag
SF	Status flag
GF	General flag
PC	Program counter
STACK[(SPW)]	Stack (Stack level is indicated by the contents of stack pointer word.)
SPW	Stack pointer word
EIF	Enable interrupt master F/F
EIR	Enable interrupt register
INTLj	Interrupt latch (j = 5 - 0)
DC	Data counter
ROM[x]	Program memory (Address x)
(ROM_H,ROM_L)	(High-order 4 bits or low-order 4 bits are expressed by suffix H/L.)
←	Transfer
⇄	Exchange
+	Addition
−	Substraction
∧	Logical AND of the corresponding bits
∨	Logical OR of the corresponding bits
∀	Exclusive OR of the corresponding bits

Symbol	Description
$(\overline{CF})$	Inversion of carry flag contents
null	Processed result is transferred nowhere
(AC)	Contents of accumulator
(H.L)	Contents of 8 bits coupling H register with L register
M[(H.L)]	Contents of data memory for which the contents of 8 bits coupling H register with L register is used as address.
(AC)<b>	Contents of bit assigned by b of accumulator.
(LR)<3:2>	Contents of bit 3 to bit 2 of L register
(PC)<11:6>	Contents of bit 11 to bit 6 of program counter

2.2 Description of instructions (*): Note 1 (**): Exec.cycle ($\overset{*}{**}$): Hexadecimal

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	($\overset{*}{**}$)	Functional Description		
Move Instruction	LD A, @HL	0 0 0 0 1 1 0 0	0 C	(AC)←M[(H·L)]	- Z 1	1
				Loads the contents of the data memory specified by the H and L registers in the accumulator.		
	LD A, x	0 0 1 1 1 1 0 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 C x_Hx_L	(AC)←M[x]	- Z 1	2
				Loads the contents of the data memory specified by the x of the instruction field in the accumulator.		
	LD HL, x	0 0 1 0 1 0 0 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 8 x_Hx_L	(LR)←M[x'], (HR)←M[x'+1] $x' = x_7x_6x_5x_4x_3x_200$	- - 1	2
				Loads the consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field in the H and L registers.		
	LD A, #k	0 1 0 0 $k_3k_2k_1k_0$	4 k	(AC)←k	- Z 1	1
				Loads the immediate data k of the instruction field in the accumulator. Serves as the clear instruction when k = 0.		

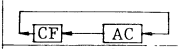
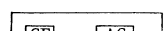
Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(⁂)	Functional Description		
Move Instruction	LD H, #k	1 1 0 0 $k_3k_2k_1k_0$	C k	(HR)←k	- - 1	1
				Loads the immediate data k of the instruction field in the H register. Serves as the clear instruction when k = 0.		
	LD L, #k	1 1 1 0 $k_3k_2k_1k_0$	E k	(LR)←k	- - 1	1
				Loads the immediate data k of the instruction field in the L register. Serves as the clear instruction when k = 0.		
	LDL A, @DC	0 0 1 1 0 0 1 1	3 3	(AC)←ROM_L[(DC)]	- Z 1	2
				Loads the lower-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator.		
	LDH A,@DC+	0 0 1 1 0 0 1 0	3 2	(AC)←ROM_H[(DC)], (DC)←(DC)+1	- Z 1	2
				Loads the higher-order 4 bits of the data read out of the data table of the program memory specified by the data counter, in the accumulator, and then increments the contents of the data counter. [Note 2]		
	ST A, @HL	0 0 0 0 1 1 1 1	0 F	M[(H·L)]←AC	- - 1	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers.		
	ST A,@HL+	0 0 0 1 1 0 1 0	1 A	M[(H·L)]←(AC), (LR)←(LR)+1	- Z $\overline{C}$	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]		
	ST A,@HL-	0 0 0 1 1 0 1 1	1 B	M[(H·L)]←(AC), (LR)←(LR)-1	- Z $\overline{B}$	1
				Stores the contents of the accumulator in the data memory specified by the H and L registers, and then decrements the contents of the L register. [Note 3]		
	ST A, x	0 0 1 1 1 1 1 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 F x_Hx_L	M[x]←(AC)	- - 1	2
				Stores the contents of the accumulator in the data memory specified by the x of the instruction field.		
	ST #k,@HL+	1 1 1 1 $k_3k_2k_1k_0$	F k	M[(H·L)]←k, (LR)←(LR)+1	- Z $\overline{C}$	1
				Stores the immediate data k of the instruction field in the data memory specified by the H and L registers, and then increments the contents of the L register. [Note 3]		

Items / Class	Assembler Mnemonic	Object Code Binary	(*,**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Move Instruction	ST #k, y	0 0 1 0 1 1 0 1 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 D k y	M[y]←k Stores the immediate value k of the instruction field in the data memory specified by y (page 0) of the instruction field. Serves as the clear instruction when k = 0.	- - 1	2
	$\overline{\text{MOV}}$ H, A	0 0 0 1 0 0 0 0	1 0	(AC)←(HR) Loads the contents of the H register in the accumulator.	- Z 1	1
	$\overline{\text{MOV}}$ L, A	0 0 0 1 0 0 0 1	1 1	(AC)←(LR) Loads the contents of the L register in the accumulator.	- Z 1	1
	XCH A, H	0 0 1 1 0 0 0 0	3 0	(HR)↔(AC) Exchanges the contents of the accumulator for those of the H register. [Note 2]	- Z 1	2
	XCH A, L	0 0 1 1 0 0 0 1	3 1	(LR)↔(AC) Exchanges the contents of the accumulator for those of the L register. [Note 2]	- Z 1	2
	XCH A,EIR	0 0 0 1 0 0 1 1	1 3	(EIR)↔(AC) Exchanges the contents of the accumulator for those of the interrupt enable register.	- - 1	1
	XCH A,@HL	0 0 0 0 1 1 0 1	0 D	M[(H·L)]↔(AC) Exchanges the contents of the accumulator for those of the data memory specified by the H and L registers. [Note 2]	- Z 1	1
	XCH A, x	0 0 1 1 1 1 0 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 D x_Hx_L	M[x]↔(AC) Exchanges the contents of the accumulator for those of the data memory specified by the x of the instruction field. [Note 2]	- Z 1	2
	XCH HL, x	0 0 1 0 1 0 0 1 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 9 x_Hx_L	M[x']↔(LR), M[x'+1]↔(HR) $x' = x_7x_6x_5x_4x_3x_200$ Exchanges the contents of the H and L registers for consecutive two-word contents of the data memory specified by the x' (modified x) of the instruction field.	- - 1	2

Items / Class	Assembler Mnemonic	Object Code Binary	(※※)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Compare Instruction	CMPR A,@HL	0 0 0 1 0 1 1 0	1 6	null←M[(H·L)]-(AC) Compares the contents of the data memory specified by the H and L registers with those of the accumulator.	$\overline{B}$ Z $\overline{Z}$	1
	CMPR A, x	0 0 1 1 1 1 1 0 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	3 E x_Hx_L	nill←M[x]-(AC) Compares the contents of the data memory specified by the x of the instruction field with those of the accumulator.	$\overline{B}$ Z $\overline{Z}$	2
	CMPR A,#k	1 1 0 1 $k_3k_2k_1k_0$	D k	null←k-(AC) Compares the immediate data k of the instruction field with the contents of the accumulator. Serves as the accumulator test instruction when k = 0.	$\overline{B}$ Z $\overline{Z}$	1
	CMPR H,#k	0 0 1 1 1 0 0 0 1 1 0 1 $k_3k_2k_1k_0$	3 8 D k	null←k-(HR) Compares the immediate data k of the instruction field with the contents of the H register. Serves as the H register test instruction when k = 0.	- Z $\overline{B}$	2
	CMPR L,#k	0 0 1 1 1 0 0 0 1 0 0 1 $k_3k_2k_1k_0$	3 8 9 k	null←k-(LR) Compares the immediate data k of the instruction field with the contents of the L register. Serves as the L register test instruction when k = 0.	- Z $\overline{B}$	2
	CMPR y,#k	0 0 1 0 1 1 1 0 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 E k y	null←k-M[y] Compares the immediate data k of the instruction field with the contents of the data memory specified by the y (page 0) of the instruction field. Serves as the data memory test instruction when k = 0.	$\overline{B}$ Z $\overline{Z}$	2
Arithmetic Instruction	INC A	0 0 0 0 1 0 0 0	0 8	(AC)←(AC)+1 Increments the contents of the accumulator.	- Z $\overline{C}$	1
	INC L	0 0 0 1 1 0 0 0	1 8	(LR)←(LR)+1 Increments the contents of the L register.	- Z $\overline{C}$	1

Items / Class	Assembler Mnemonic	Object Code Binary	(*)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	INC @HL	0 0 0 0 1 0 1 0	0 A	M[(H·L)]←M[(H·L)]+1 Increments the contents of the data memory specified by the H and L registers.	- Z $\overline{C}$	1
	DEC A	0 0 0 0 1 0 0 1	0 9	(AC)←(AC)-1 Decrements the contents of the accumulator.	- Z $\overline{B}$	1
	DEC L	0 0 0 1 1 0 0 1	1 9	(LR)←(LR)-1 Decrements the contents of the L register.	- Z $\overline{B}$	1
	DEC @HL	0 0 0 0 1 0 1 1	0 B	M[(H·L)]←M[(H·L)]-1 Decrements the contents of the data memory specified by the H and L registers.	- Z $\overline{B}$	1
	ADDC A,@HL	0 0 0 1 0 1 0 1	1 5	(AC)←(AC)+M[(H·L)]+(CF) Adds the contents of the data memory specified by the H and L registers as well as those of the carry flag to those of the accumulator, and places the result in the accumulator.	C Z $\overline{C}$	1
	ADD A,@HL	0 0 0 1 0 1 1 1	1 7	(AC)←(AC)+M[(H·L)] Adds the contents of the data memory specified by the H and L registers to those of the accumulator, and places the result in the accumulator.	- Z $\overline{C}$	1
	ADD A, #k	0 0 1 1 1 0 0 0 0 0 0 0 $k_3k_2k_1k_0$	3 8 0 k	(AC)←(AC)+k Adds the immediate data k of the instruction field to the contents of the accumulator, and places the result in the accumulator. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	ADD H, #k	0 0 1 1 1 0 0 0 1 1 0 0 $k_3k_2k_1k_0$	3 8 C k	(HR)←(HR)+k Adds the immediate data k of the instruction field to the contents of the H register, and places the result in the H register. Serves as the H register increment instruction or the decrement instruction when k = 1 or F, respectively.	- Z $\overline{C}$	2

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (*)	Function / Functional Description	Flag(*): CF ZF SF	(**)
Arithmetic Instruction	ADD L,#k	0 0 1 1 1 0 0 0 1 0 0 0 $k_3k_2k_1k_0$	3 8 8 k	(LR)←(LR)+k Adds the immediate data k of the instruction field to the contents of the L register, and places the result in the L register.	- Z $\overline{C}$	2
	ADD @HL,#k	0 0 1 1 1 0 0 0 0 1 0 0 $k_3k_2k_1k_0$	3 8 4 k	M[(H·L)]←M[(H·L)]+k Adds the immediate data k of the instruction field to the contents of the data memory specified by the H and L register, and places the result in the data memory. Serves as the correction instruction for the decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	ADD y,#k	0 0 1 0 1 1 1 1 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	2 F k y	M[y]←M[y]+k Adds the immediate data k of the instruction field to contents of the data memory specified by the y (page 0) of the instruction field, and places the result in the data memory. Serves as the correction instruction for decimal addition and subtraction when k = 6 or A.	- Z $\overline{C}$	2
	SUBRC A,@HL	0 0 0 1 0 1 0 0	1 4	(AC)←M[(H·L)]-(AC)-($\overline{CF}$) Subtracts the contents of the accumulator and the inverse contents of the carry flag from the contents of the data memory specified by the H and L registers, and places the result in the accumulator.	$\overline{B}$ Z $\overline{B}$	1
	SUBR A,#k	0 0 1 1 1 0 0 0 0 0 0 1 $k_3k_2k_1k_0$	3 8 1 k	(AC)←k-(AC) Subtracts the contents of the accumulator from the immediate data k of the instruction field, and places the result in the accumulator. Serves as the accumulator 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	- Z $\overline{B}$	2

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (*,**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Arithmetic Instruction	SUBR @HL,#k	0 0 1 1 1 0 0 0 0 1 0 1 $k_3k_2k_1k_0$	3 8 5 k	M[(H·L)]←k−M[(H·L)] Subtracts the contents of the data memory specified by the H and L registers from the immediate data k of the instruction field, and places the result in the data memory. Serves as the data memory 2's complement instruction or the data inversion (1's complement) instruction when k = 0 or F, respectively.	− Z $\overline{B}$	2
Logical Instruction	ROLC A	0 0 0 0 0 1 0 1	0 5	 (rotate left by 1 bit) Rotates the contents of the accumulator and carry flag to the left by one bit. [Note 4]	C Z $\overline{C}$	1
	RORC A	0 0 0 0 0 1 1 1	0 7	 (rotate right by 1 bit) Rotates the contents of the accumulator and carry flag to the right by one bit. [Note 4]	C Z $\overline{C}$	1
	AND A,@HL	0 0 0 1 1 1 1 0	1 E	(AC)←(AC)∧M[(H·L)] Carries out the logical AND of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L register, and places the result in the accumulator.	− Z $\overline{Z}$	1
	AND A,#k	0 0 1 1 1 0 0 0 0 0 1 1 $k_3k_2k_1k_0$	3 8 3 k	(AC)←(AC)∧k Carries out the logical AND of the corresponding bits with the contents of the accumulator and the immediate data k of the instruction field, and places the result in the accumulator.	− Z $\overline{Z}$	2
	AND @HL,#k	0 0 1 1 1 0 0 0 0 1 1 1 $k_3k_2k_1k_0$	3 8 7 k	M[(H·L)]←M[(H·L)]∧k Carries out the logical AND of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.	− Z $\overline{Z}$	2

Items / Class	Assembler Mnemonic	Object Code Binary	(*) (**)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Logical Instruction	OR A, @HL	0 0 0 1 1 1 0 1	1 D	(AC)←(AC)∨M[(H·L)]	- Z $\overline{Z}$	1
				Carries out the logical OR of the corresponding bits with the contents of the accumulator and those of the data memory specified by the H and L registers, and places the result in the accumulator.		
	OR A, #k	0 0 1 1 1 0 0 0 0 0 1 0 $k_3k_2k_1k_0$	3 8 2 k	(AC)←(AC)∨k	- Z $\overline{Z}$	2
				Carries out the logical OR of the corresponding bits with the contents of the accumulator and the immediate data k of the istruction field, and places the result in the accumulator.		
	OR @HL,#k	0 0 1 1 1 0 0 0 0 1 1 0 $k_3k_2k_1k_0$	3 8 6 k	M[(H·L)]←M[(H·L)]∨k	- Z $\overline{Z}$	2
				Carries out the logical OR of the corresponding bits with the contents of the data memory specified by the H and L registers and the immediate data k of the instruction field, and places the result in the data memory.		
	XOR A, @HL	0 0 0 1 1 1 1 1	1 F	(AC)←(AC)∀M[(H·L)]	- Z $\overline{Z}$	1
				Carries out the logical exclusive OR of the corresponding bits with the contents of the accumulator and those of data memory specified by the H and L registers. and places the result in the accumulator.		
Bit Manipulation Instruction	TEST CF	0 0 0 0 0 1 1 0	0 6	(SF)←($\overline{CF}$), (CF)←0	0 - *	1
				Places the inverse contents of the carry flag in the status flag, and then resets the carry flag to "0".		
	TEST A, b	0 1 0 1 1 1 b_1b_0	5 C+b	(SF)←$\overline{(AC)<b>}$	- - *	1
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the accumulator, in the status flag.		
	TEST @HL,b	0 1 0 1 1 0 b_1b_0	5 8+b	(SF)←$\overline{M[(H·L)]<b>}$	- - *	1
				Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, in the status flag.		

Items / Class	Assembler Mnemonic	Object Code Binary	Object Code (*)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Bit Manipulation Instruction	TEST y, b	0 0 1 1 1 0 0 1 1 0 b_1b_0 $y_3y_2y_1y_0$	3 9 8+b y	(SF)←$\overline{M[y]<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.	- - *	2
	TEST %P,b	0 0 1 1 1 0 1 1 1 0 b_1b_0 $p_3p_2p_1p_0$	3 B 8+b P	(SF)←$\overline{P[p]<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the port (port register in the output port, and pin input in the input and I/O port) specified by the p of the instruction field, in the status flag.	- - *	2
	TEST @L	0 0 1 1 0 1 1 1	3 7	(SF)←$\overline{P[(LR)<3:2>+4]<(LR)<1:0>>}$ Places the inverse contents of the bit, which is specified by the lower-order two bits of the L register, of the ports R_4 - R_7 (pin input) specified by the higher two bits of the L register, in the status flag.	- - *	2
	TESTP CF	0 0 0 0 0 1 0 0	0 4	(SF)←(CF), (CF)←1 Places the contents of the carry flag in the status flag, and then sets the carry flag to "1".	1 - *	1
	TESTP ZF	0 0 0 0 1 1 1 0	0 E	(SF)←(ZF) Places the contents of the zero flag in the status flag.	- - *	1
	TESTP GF	0 0 0 0 0 0 0 1	0 1	(SF)←(GF) Places the contents of the general flag in the status flag.	- - *	1
	TESTP y,b	0 0 1 1 1 0 0 1 1 1 b_1b_0 $y_3y_2y_1y_0$	3 9 C+b y	(SF)←M[y]<b> Places the contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, in the status flag.	- - *	2

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF	ZF	SF	(**)
		Binary	(⁂)	Functional Description				
Bit Manipulation Instruction	TESTP %P,b	0 0 1 1 1 0 1 1 1 1 b_1b_0 $p_3p_2p_1p_0$	3 B C+b P	(SF)←P[p]<b>	-	-	*	2
				Places the contents of the bit, which is specified by the b of the instruction field, of the port (port register for the output port, and pin input for the input or I/O ports), which is specified by the p of the instruction field, in the status flag.				
	SET GF	0 0 0 0 0 0 1 1	0 3	(GF)←1	-	-	1	1
				Sets the general flag to "1".				
	SET @HL, b	0 1 0 1 0 0 b_1b_0	5 b	M[(H·L)]<b>←1	-	-	1	1
				Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, to "1".				
	SET y, b	0 0 1 1 1 0 0 1 0 0 b_1b_0 $y_3y_2y_1y_0$	3 9 b y	M[y]<b>←1	-	-	1	2
				Sets the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "1".				
	SET %p, b	0 0 1 1 1 0 1 1 0 0 b_1b_0 $p_3p_2p_1p_0$	3 B b y	P[p]<b>←1	-	-	1	2
				Sets the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "1".				
	SET @L	0 0 1 1 0 1 0 0	3 4	P[(LR)<3:2>+4]<(LR)<1:0>>←1	-	-	1	2
				Sets the bit, which is specified by the lower-order two bits of the L register, of the ports R_4 - R_7 specified by the higher-order two bits of the L register, to "1".				
	CLR GF	0 0 0 0 0 0 1 0	0 2	(GF)←0	-	-	1	1
				Clears the general flag to "0".				
	CLR @HL, b	0 1 0 1 0 1 b_1b_0	5 4+b	M[(H·L)]<b>←0	-	-	1	1
				Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L register, to "0".				

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (*)	Function / Functional Description	Flag(*) CF ZF SF	(**)
Bit Manipulation Instruction	CLR y, b	0 0 1 1 1 0 0 1 0 1 b_1b_0 $y_3y_2y_1y_0$	3 9 4+b y	M[y]<b> ← 0 Clears the bit, which is specified by the b of the instruction field, of the data memory specified by the y (page 0) of the instruction field, to "0".	- - 1	2
	CLR %P, b	0 0 1 1 1 0 1 1 0 1 b_1b_0 $p_3p_2p_1p_0$	3 B 4+b p	P[p]<b> ← 0 Clears the bit, which is specified by the b of the instruction field, of the port specified by the p of the instruction field, to "0".	- - 1	2
	CLR @L	0 0 1 1 0 1 0 1	3 5	P[(LR)<3:2>+4]<(LR)<1:0>> ← 0 Clears the bit, which is specified by the lower-order two bits of the L register, of the ports R_4 - R_7 specified by the higher-order two bits of the L register, to "0".	- - 1	2
	CLR IL, r	0 0 1 1 0 1 1 0 1 1 r_5r_4 $r_3r_2r_1r_0$	3 6 C+r_Hr_L	(INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Resets the interrupt latch INTLj when the rj of the instruction field is "0". (j = 5 - 0)	- - 1	2
	EICLR IL,r	0 0 1 1 0 1 1 0 0 1 r_5r_4 $r_3r_2r_1r_0$	3 6 4+r_Hr_L	(EIF) ← 1, (INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Sets the interrupt enable master F/F to "1". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 - 0)	- - 1	2
	DICLR IL,r	0 0 1 1 0 1 1 0 1 0 r_5r_4 $r_3r_2r_1r_0$	3 6 8+r_Hr_L	(EIF) ← 0, (INTL)<5:0> ← (INTL)<5:0> ∧ r<5:0> Resets the interrupt enable master F/F to "0". Interrupt latch INTLj is reset when the rj of the instruction field is "0". (j = 5 - 0)	- - 1	2
Input Instruction	IN %P, A	0 0 1 1 1 0 1 0 0 0 1 0 $p_3p_2p_1p_0$	3 A 2 P	(AC) ← P[p] Places the input data from the port specified by the p of the instruction field in the accumulator.	- Z $\overline{Z}$	2

Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: ($\overset{*}{**}$)	Function: Functional Description	Flag(*) CF ZF SF	(**)
Input/Output Instruction	IN %P, @HL	0 0 1 1 1 0 1 0 0 1 1 0 $p_3p_2p_1p_0$	3 A 6 P	M[(H·L)]←P[p] Places the input data from the port specified by the p of the instruction field in the data memory specified by the H and L registers.	- - $\overline{Z}$	2
	OUT A, %P	0 0 1 1 1 0 1 0 1 0 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $8+2\overline{p}_4p$	P[p]←(AC), P=$P_4P_3P_2P_1P_0$ Outputs the contents of the accumulator to the port specified by the p of the instruction field. ($0 \leq p \leq 31$)	- - 1	2
	OUT @HL,%P	0 0 1 1 1 0 1 0 1 1 $\overline{p_4}$ 0 $p_3p_2p_1p_0$	3 A $C+2\overline{p}_4p$	P[p]←M[(H·L)], P=$P_4P_3P_2P_1P_0$ Outputs the contents of the data memory specified by the H and L registers to the port specified by the p of the instruction field. ($0 \leq p \leq 31$)	- - 1	2
	OUT #k,%P	0 0 1 0 1 1 0 0 $k_3k_2k_1k_0$ $p_3p_2p_1p_0$	2 C k P	P[p]←k Outputs the immediate data k of the instruction field to the port specified by the p of the instruction field. Serves as the clear instruction when k = 0.	- - 1	2
	OUTB @HL	0 0 0 1 0 0 1 0	1 2	P[2]·P[1]←ROM[F·(E+(CF))·M[(H·L)]] Outputs the data (eight bits) of the program memory located in addresses FE0 - FFF, which use a five-bit data connecting the contents of the data memory specified by the H and L registers and those of the carry flag, as lower-order five-bit addresses, to the P_2 - P_1 ports.	- - 1	2
Branch·Subroutine Instruction	BS a	0 1 1 0 $a_{11}a_{10}a_9a_8$ $a_7a_6a_5a_4$ $a_3a_2a_1a_0$	6 a_H a_M a_L	If SF=1 then (PC)←a else null. Places the immediate data a of the instruction field in the program counter if the status flag is at "1". If the status flag is at "0", sets the status flag only to "1", and moves to the next address.	- - 1	2

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binary	(*)	Functional Description		
Branch·Subroutine Instruction	BSS a	1 0 d_5d_4 $d_3d_2d_1d_0$	$8+d_Hd_L$	If SF=1 then (PC)←a else null, a=(PC)<11·6>·d	- - 1	1
				Carries out the branch within a page (64-byte) if the status flag is at "1"; brings the immediate value d of the instruction field into the lower-order six bits of the program counter. Since the updated value remains in the higher-order six bits, if this instruction is specified in the last address in the page, branching is carried out to the next page. If the status flag is at "0", it sets the status flag only to "1", and moves to the next address. [Note 5]		
	CALL a	0 0 1 0 0 $a_{10}a_9a_8$ $a_7a_6a_5a_4$ $a_3a_2a_1a_0$	2 aH a_M a_L	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, 0 ≤ a ≤ 2,047	- - -	2
				Carries out the subroutine call; saves the contents of the program counter in the stack, and decrements the stack pointer word, and then places the immediate data a of the instruction field in the program counter. However, the call address of the subroutine must be in the addresses 000 -7FF. [Note 5]		
	CALLS a	0 1 1 1 $n_3n_2n_1n_0$	7 n	STACK[(SPW)]←(PC),(SPW)←(SPW)-1 (PC)←a, a=8n+6(n≠0), 134(n=0)	- - -	2
				Carries out the short form subroutine call. The operation is the same as that of the "CALL" instruction except that the value to be set in the program counter is automatically defined by the n of the instruction field. [Note 5]		
	RET	0 0 1 0 1 0 1 1	2 A	(SPW)←(SPW)+1,(PC)←STACK[(SPW)]	- - -	2
				Returns from the subroutine to the previous program; increments the stack pointer word, and restores the data of the return address from the stack to the program counter.		

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*) CF ZF SF	(**)
		Binaty	(⁂)	Functional Description		
Branch·Subroutine Instruction	RETI	0 0 1 0 1 0 1 1	2 B	(SPW)←(SPW)+1, (FLAG·PC)←STACK[(SPW)], (EIF)←1	* * *	2
				Returns from the interrupt processing routine; increments the stack pointer word, and restores the data of the return address from the stack and the data of the flag, to the program counter and the flag, respectively. And then, it sets the interrupt enable master F/F to "1".		
Other Instruction	$\overline{\text{NOP}}$	0 0 0 0 0 0 0 0	0 0	no operation	- - -	1
				Moves to the next instruction without performing any operation.		

Note 1. Setting Condition of Flag.

"C" indicates the carry output from the most significant position in the addition operation, and "B" indicates the borrow output from the most significant position in the subtraction operation.

"Z" indicates the zero detection signal to which "1" is applied only when either the ALU output of the processing result or all four bits of the data transferred to the accumulator are zero.

The flag is set to "C", "$\overline{C}$", "$\overline{B}$", "Z", "$\overline{Z}$", "1", or "0" according to the data processing rusult. The value specified by the function is set to the flag with the mark "*", and the mark "-" denotes no change in the state of the flag.

Note 2. The zero flag is set according to the data set in the accumulator.

Note 3. The flags (ZF, SF) are set according to the result of increment or decrement of the L register.

Note 4. The carry is the data shifted out from the accumulator.

Note 5. The contents of the program counter indicate the next address of the instruction to be executed.

3. Basic operation and pin operation

1. Instruction cycle
2. Basic clock (CP) generation
3. Initialization operation
4. Hold Input
5. Interrupt input
6. Input/output port
7. Other pins

The timing in each basic operation, and the configuration, function, and timing of the pins are described according to the above items.

The operation and timing with each component of the hardware are covered in detail in the description of each item of the components.

Different input/output port circuit system can be specified according to the port. The details to specify the type of input/output port circuit are given in the descreption covering the program tape format.

3.1 Instruction cycle

The instruction execution and the internal hardware control are synchronized with the basic clock (CP, fc Hz).

The minimum unit of the instruction execution is called the "instruction cycle", and all instructions are executed by one or two instruction cycles, each of which is called one-cycle instruction or two-cycle instruction.

An instruction cycle consists of four machine cycles ($M_1 \sim M_4$). and each machine cycle requires four basic clock times.

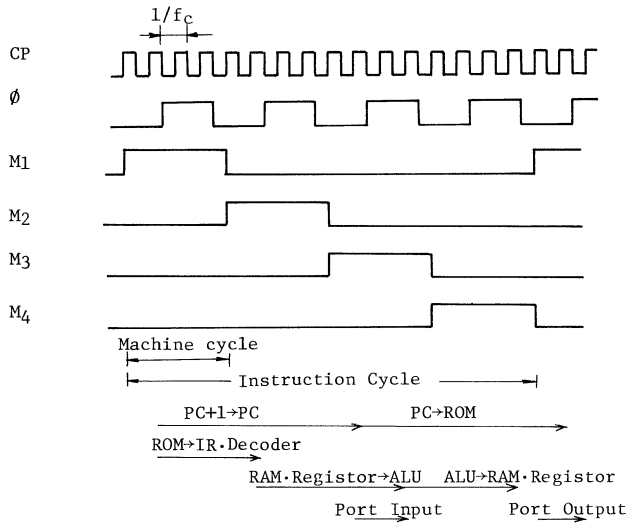


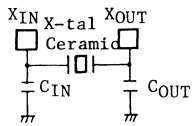
Fig. 3.1.1 Instruction Cycle

3.2 Basic clock (CP) generation

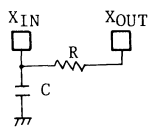
An oscillation circuit is contained, and the necessary clock is easily generated by connecting the resonator to external pins (X_{IN}, X_{OUT}). By the way, the oscillation circuit serves as Schmitt circuit.

The clock generated in the oscillation circuit is called the "basic clock" with which the internal control is synchronized. The basic clock is applied to the timing generator and the control circuit of system to provide various control signals.

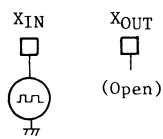
The following are the examples of the resonator connection.



(a) For X-tal or ceramic resonator



(b) For RC



(c) For external oscillator

Fig. 3.2.1 Resonator Connections

3.3 Initialization operation

Initialization operation is performed by keeping the $\overline{\text{RESET}}$ pin to the low level. However, the following conditions are required to put the initialization operation into practice with certainty

① The supply voltage is within the operating voltage.

② The oscillation circuit operates stably.

③ The $\overline{\text{RESET}}$ is held at the low level in at least three instruction cycle time.

The following processing are performed by the initialization operation.

① Reset the program counter to "0"

② Set the status flag to "1".

③ Reset the interrupt enabling master F/F and the interrupt enabling register to "0", and also reset the interrupt latch to "0".

④ Reset the divider to "0".

⑤ Initialize the input/output port and command register to the fixed level.

The initialization operation is released due to the rise of the $\overline{\text{RESET}}$ pin to the high level, and the program can be executed from address 0 in sequence.

The $\overline{\text{RESET}}$ pin serves as Schmitt circuit input, and is connected with pull-up resistor ($\fallingdotseq$300kΩ TYP., MOS-load resistor).

3.4 Hold input

The hold function is the function that holds the status immediately before the system operation is stopped at low power consumption. The $\overline{\text{HOLD}}$ terminal serves as Schmitt circuit input and is used to the signal input requesting or releasing of the hold operation. Further, for details of the hold operation, refer to the description of the hold control circuit.

Caution: To restart the system operation from the hold operation at low holding voltage, the following precaution is required. When supply voltage rises from holding voltage to operating voltage, the $\overline{\text{RESET}}$ input is also at the high level and rises together with supply voltage.

If a time constant circuit, etc. are externally added in this case, voltage build-up at the $\overline{\text{RESET}}$ input will be slower than that of supply voltage. Therefore, if voltage level at the $\overline{\text{RESET}}$ input drops below the non-reversible high level (Schmitt circuit) at the $\overline{\text{RESET}}$ terminal input at this time, the initialization operation may possibly be executed.

3.5 Interrupt input

Two pins ($\overline{\text{INT}_1}$, $\overline{\text{INT}_2}$) are provided for the external interrupt input. Since these pins are common pins with R8 port, they can be used as I/O pins ($R8_2$, $R8_0$) respectively, if not used as the interrupt input pins.

The interrupt via INT_2 can be inhibited at any time by the program, but the interrupt via INT_1 is not inhibited by it independently. Therefore, when this pin is used for the $R8_2$ port, the interrupt will always take place due to the detection of the falling edge of the signal. It is necessary to set a dummy interrupt service program including the (RETI) instruction only, even if the INT_1 is not used.

The interrupt latch is set by the falling edge of the external inputs ($\overline{\text{INT}_1}$, $\overline{\text{INT}_2}$), and an interrupt request is made to the CPU. To assure that the interrupt latch is positively set or reset, and that the next interrupt request is set, both of the high and low levels should be kept for more than two instruction cycle time.

The external interrupt input is the Schmitt circuit input.

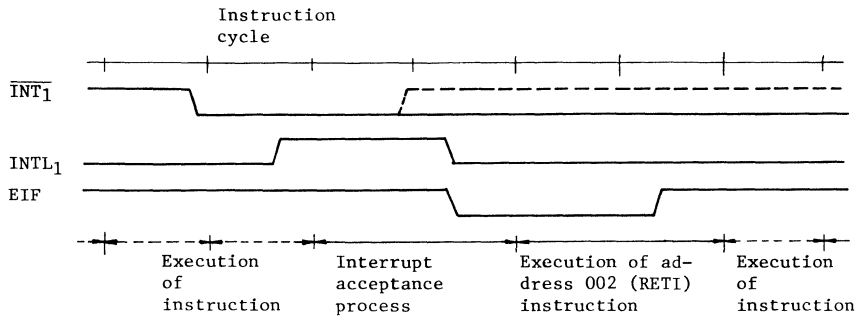


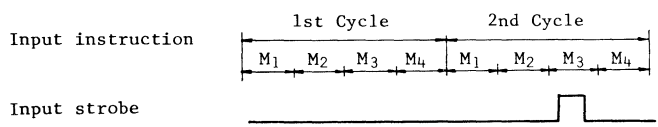
Fig. 3.5.1 Interrupt Timing (Dummy process of INT_1 interrupt)

3.6 Input/output port

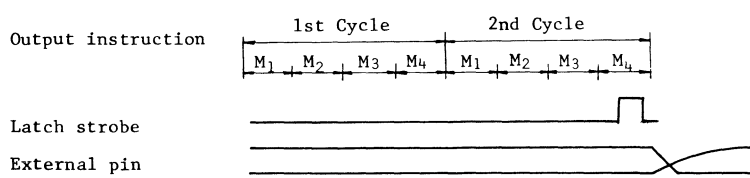
(1) Input/output timing

The timing to read the external data from the input port or I/O port is in M_3 machine cycle in the second cycle of the input instruction (two-cycle instruction). Since this timing cannot be externally recognized, the transient input data should be processed by a program.

The timing to output the data to the output port or I/O port is in M_4 machine cycle in the second cycle of the output instruction (two-cycle instruction), but this timing cannot be externally recognized.



(a) Input Timing



(b) Output Timing

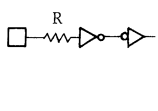
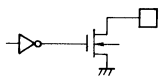
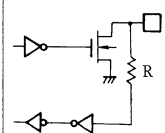
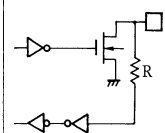
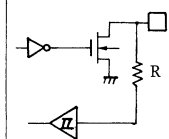
Fig. 3.6.1 Input/Output Timing

(2) Input/output circuit format

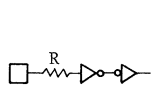
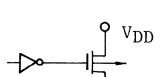
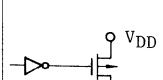
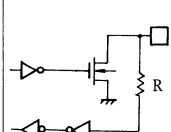
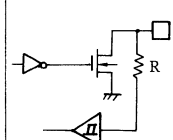
The input output circuit format of the input/output port is shown following. For the TMP47C40P, TMP47C20P, TMP47C41P and TMP47C21P, any of the input/output circuit systems shown in the following tables can be selected. You can specify your input/output circuit system when requesting the program tape. (In the case of high breakdown voltage output type, production part's number is TMP47C41P or TMP47C21P.)

*: Port **: Circuit

Input/Output Circuit Code ($\overline{\text{IOCODE}}$) FA

* / **	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
Input/Output Circuit Format	 R = 1kΩ (TYP.)		 R = 1kΩ(TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained.	o Sink open drain output. o Output latch is initialized to the high level.	o Sink open drain output. o Output latch is initialized to the high level.	o Sink open drain output. o Output latch is initialized to the high level.	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level.

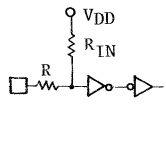
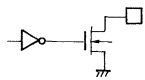
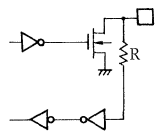
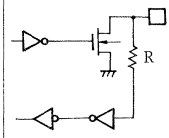
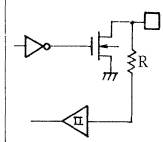
Input/Output Circuit Code ($\overline{\text{IOCODE}}$) HA

* / **	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
Input/Output Circuit Format	 R = 1kΩ (TYP.)			 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained.	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the low level	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the low level o Only for output	o Sink open drain output o Output latch is initialized to the high level.	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level.

(Note: In this case, production part's number is TMP47C41P or TMP47C21P.)

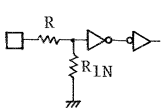
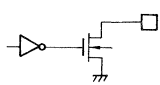
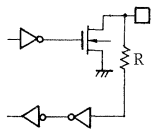
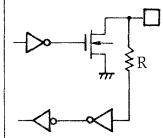
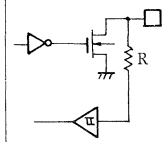
*: Port **: Circuit

Input/Output Circuit Code ($\overline{IOCODE}$) FB

***	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
Input/Output Circuit Format	 R_{IN}=70kΩ(TYP.) R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-up resistor is contained.	o Sink open drain output o Output latch is initialized to the high level.	o Sink open drain output o Output latch is initialized to the high level.	o Sink open drain output o Output latch is initialized to the high level.	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level.

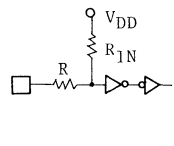
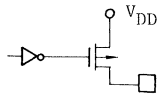
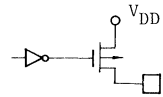
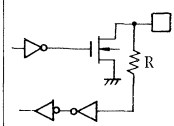
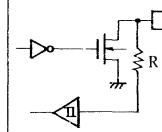
*: Port **: Circuit

Input/Output Circuit Code ($\overline{IOCODE}$) FC

***	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
Input/Output Circuit Format	 R_{IN}=70kΩ (TYP.) R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-down resistor is contained.	o Sink open drain output o Output latch is initialized to the high level.	o Sink open drain output o Output latch is initialized to the high level.	o Sink open drain output o Output latch is initialized to the high level.	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level.

*: Port **: Circuit

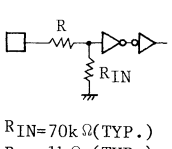
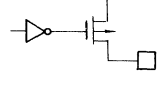
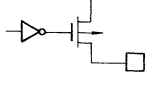
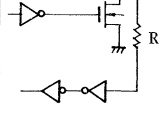
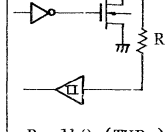
Input/Output Circuit Code ($\overline{IOCODE}$) HB

** \ *	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
Input/Output Circuit Format	 R_{IN}=70kΩ(TYP.) R = 1kΩ (TYP.)			 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-up resistor is contained.	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the low level.	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the low level. o Only for output.	o Sink open drain output. o Output latch is initialized to the high level	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level.

(Note: In this case, production part's number is TMP47C41P or TMP47C21P.)

*: Port **: Circuit

Input/Output Circuit Code ($\overline{IOCODE}$) HC

** \ *	Input (K_0)	Output (P_1, P_2)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
Input/Output Circuit Format	 R_{IN}=70kΩ(TYP.) R = 1kΩ (TYP.)			 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-down resistor is contained.	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the low level.	o Source open drain output. o High breakdown voltage output o Output latch is initialized to the high level. o Only for output.	o Sink open drain output. o Output latch is initialized to the high level.	o Schmitt circuit input. o Sink open drain output. o Output latch is initialized to the high level.

(Note: In this case, production part's number is TMP47C41P or TMP47C21P.)

3.7 Other pins

Timer/Counter input

Two pins (T_1, T_2) are provided for the external timer/counter inputs. Since these pins are common pins with R_8 port, they can be also used as I/O pins (R_{83}, R_{81}), respectively, if not used as the timer/counter inputs.

The count latch is set by the rising edge of the external input (T_1, T_2), and a count request is made to the CPU. To assure that the count latch is positively set or reset, both of the high and low levels should be kept for more tham two instruction cycle times.

The external timer/counter input is the Schmitt circuit input.

Serial port

This port is connected to the external circuitry via three pins ($\overline{SCK}$, SO, SI), which are also used for the R_9 port. These pins can be used as the pins of the R_9 port (R_{92}, R_{91}, R_{90}), if not used for the serial port.

To assure that the shift operation is positively performed in the external clock mode, both of the high and low levels should be kept for more than two instruction cycle times.

The $\overline{SCK}$ input in the external clock mode and the SI input in the receive mode are Schmitt circuit inputs.

TEST pin

This pin is used for the shipment test. To operate the user system with this pin, the input should be surely set to the low level. By the way, TEST pin is connected with pull-down resistor ($\approx$ 70kΩ TYP.).

TENTATIVE

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS} = 0V)

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.5 ∿ 7	V
V_{IN}	Input Voltage	-0.5 ∿ V_{DD} + 0.5	V
V_{OUT1}	Output Voltage (Except open drain terminal)	-0.5 ∿ V_{DD} + 0.5	V
V_{OUT2}	Output Voltage (Sink open drain terminal)	-0.5 ∿ 10	
V_{OUT3}	Output Voltage (Source open drain terminal)	-35 ∿ V_{DD} + 0.5	
P_D	Power Dissipation (Topr=70°C)	600	mW
Tsld	Soldering Temperature · Time	260 (10 sec.)	°C
Tstg	Storage Temperature	-55 ∿ 125	
Topr	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNIT
T_{opr}	Operating Temperature		-30	70	°C
V_{DD}	Supply Voltage		4.5	6	V
V_{DDH}	Supply Voltage (Hold)		2	6	
V_{IH1}	Input High Voltage (Except Schmitt circuit input) [Note 1]	$V_{DD} \geq 4.5V$	V_{DD} x 0.7	V_{DD}	V
V_{IH2}	Input High Voltage (Schmitt circuit input)		V_{DD}x0.75	V_{DD}	
V_{IH3}	Input High Voltage	$V_{DD} < 4.5$	V_{DD} x 0.9	V_{DD}	
V_{IL1}	Input Low Voltage (Except Schmitt circuit input) [Note 1]	$V_{DD} \geq 4.5V$	0	V_{DD} x 0.3	
V_{IL2}	Input Low Voltage (Schmitt circuit input)		0	V_{DD}x0.25	
V_{IL3}	Input Low Voltage	$V_{DD} < 4.5V$	0	V_{DD} x 0.1	
V_{OUT}	Output Voltage (Source open drain P_1, P_2, P_4 ∿ P_6)		V_{DD}-35	V_{DD}	V
f_C	Clock Frequency		0.4	4.2	MHz
t_{WCH}	Clock High Pulse Width [Note 2]	$V_{IN} = V_{IH}$	80	-	ns
t_{WCL}	Clock Low Pulse Width [Note 2]	$V_{IN} = V_{IL}$	80	-	

(Note 1) R_4 ∿ R_6 ports are exclusively used for output except the sink open drain output.

(Note 2) In case of the external clock operation.

D.C. CHARACTERISTICS (V_{SS}=0V, V_{DD}=5V±10%, T_{opr}=-30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	Note 1 TYP.	MAX.	UNIT
V_{HS}	Hysteresis Voltage (Schmitt circuit input)		-	0.7	-	V
I_{IN1}	Input Current (K_0, $\overline{HOLD}$) [Note 2]	V_{DD}=5.5V, V_{IN}=5.5/0V	-	-	±20	µA
I_{IN2}	Input Current (Sink open drain R port)		-	-	±20	
I_{IL}	Input Low Current (Push-pull R_7 ∿ R_9)	V_{DD}=5.5V, V_{IN}=0.4V	-	-	-2	mA
R_{IN}	Input resistor (K_0 with input resistor)		30	70	150	kΩ
I_{LO1}	Output Leak Current (Sink open drain P, R port)	V_{DD}=5.5V, V_{OUT}=5.5V	-	-	20	µA
I_{LO2}	Output Leak Current (Source open drain P_1,P_2,R_4 ∿ R_6)	V_{DD}=5.5V, V_{OUT}=-32V	-	-	-20	
V_{OH1}	Output High Voltage (Push pull R_7 ∿ R_9)	V_{DD}=4.5V, I_{OH}=-200µA	2.4	-	-	V
V_{OH2}	Output High Voltage (Source open drain P_1, P_2)	V_{DD}=4.5V, I_{OH}=-1.6mA	2.4	-	-	
V_{OH3}	Output High Voltage (Source open drain R_4 ∿ R_6)	V_{DD}=4.5V, I_{OH}=-10mA	2.4	-	-	
V_{OL}	Output Low Voltage (P, R port except source open drain)	V_{DD}=4.5V, I_{OL}=1.6mA	-	-	0.4	
I_{DDO}	Operating Supply Current [Note 3]	V_{DD}(V_{DDH})=5.5V, fc=4MHz, V_{IN}=5.3/0.2V (all valid) C_L=50pF, C_{XIN}=C_{XOUT}=10pF	-	3	6	mA
I_{DDH}	Holding Supply Current [Note 3]		-	0.5	20	µA

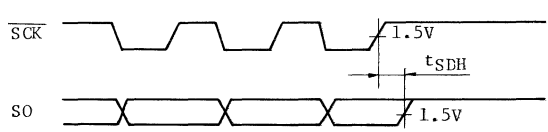
(Note 1) TYP. values show those when T_{opr}=25°C, V_{DD}=5V.
(Note 2) When the K_0 port has a built-in input resistor, current by resistor is excluded.
(Note 3) When K_0 port has a built-in input resistor, current value is that at time of open. Further, voltage level at R port is valid.

A.C. CHARACTERISTICS (V_{SS}=0V, V_{DD}=5V±10%, T_{opr}=-30 ∿ 70°C)

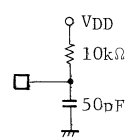
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
t_{cy}	Instruction Cycle Time		3.8	-	40	µs
t_{SDH}	Shift Data Hold Time	(Note 1)	0.5tcy-300	-	-	ns

A.C. TIMING CHART

Serial Port (Completion of Transmission)

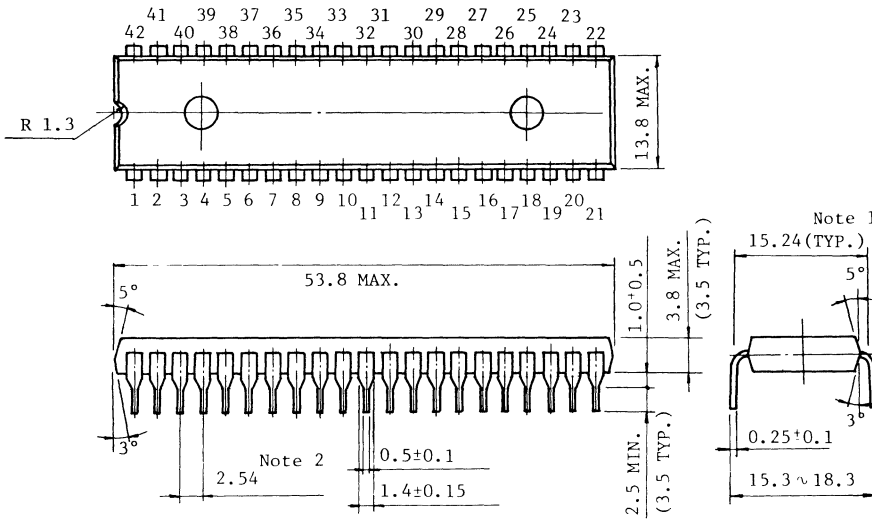


(Note 1) $\overline{SCK}$, SO terminal external circuit



EXTERNAL DIMENSION VIEW

Unit in mm



Weight 5.7g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.42 leads.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP47C40N, TMP47C41N
TMP47C20N, TMP47C21N

SILICON MONOLITHIC SILICON GATE MOS

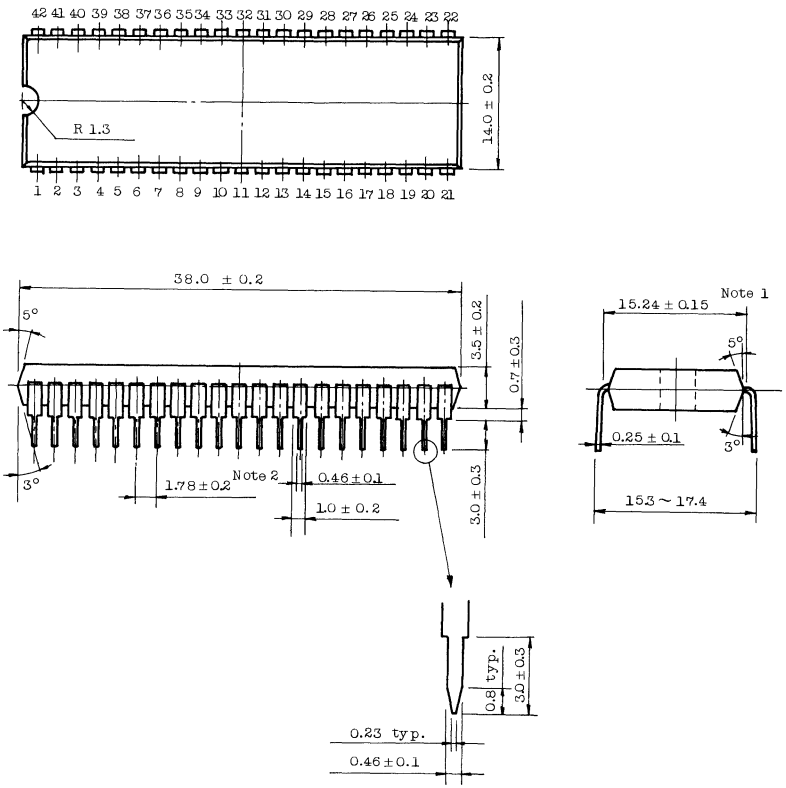
CMOS 4-BIT SINGLE CHIP MICROCMPUTER (TLCS-47C)
TMP47C40N, TMP47C20N, TMP47C41N, TMP47C21N

GENERAL DESCRIPTION

TMP47C40N, TMP47C20N, TMP47C41N and TMP47C21N are the shrunk package versions of TMP47C40P, TMP47C20P, TMP47C41P, TMP47C21P, respectively. Their function, instruction, pin description and electrical characteristics are compatible. The package area is reduced to around 70 percent in comparison with the standard package.

EXTERNAL DIMENSION VIEW

Unit in mm



Weight 4.0g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 1.78mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.42 leads.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP47C22F

SILICON MONOLITHIC SILICON GATE CMOS

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER TLCS-47(C) TMP47C22F

GENERAL DESCRIPTION

The TMP47C22F is a chip countaining LCD driver for the TLCS-47C.

The memory capacity consists of ROM 2,048 × 8 bits and RAM 192 × 4 bits.

The TMP4700AC (NMOS) is an evaluator chip used for the system development.

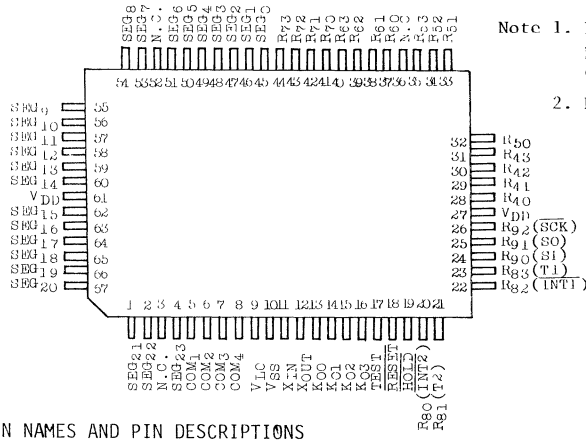
FEATURES

- 4-bit single chip microcomputer with built-in
  ROM, RAM, input/output port, divider, timer/counter, and serial port.
- Instruction execution time: 4 μs (at 4 MHz clock)
- Effective instruction set
  90 instructions, software compatible in the series
- Subroutine nesting: Maximum 15 levels
- 6 interrupts (External: 2, Internal: 4)
  Independently latched control and multiple interrupt control
- Input/output port (27 pins)

Input	1 port	4 pins
I/O	4 ports	16 pins
I/O (Note)	2 ports	7 pins

  Note: These I/O ports are also used for the interrupt input, timer/counter input, and serial port; therefore, it is programmably selectable for each application.
- Table look-up and table search function (Instruction)
  Table can be set up in the whole ROM area.
- 12-bit timer/counter (2 channels)
  Event counter, timer, and pulse width measurement mode is programmably selectable.
- Serial port with 4-bit buffer
  Receive/Transfer mode is programmably selectable.
  External/internal clock and leading/trailing edge mode are programmably selectable.
- 18-stage divider (with 4-stage prescaler)
  Frequency applied for timer interrupt of divider is programmably selectable.
- LCD drive circuit (automatic display) built-in
  - LCD direct drive is available (Max. 12-digit display at 1/4 duty LCD)
  - 1/4, 1/3, 1/2 duties or static drive are programmably selectable.
- Hold function
  Battery operation/condenser backup is avilable.
- On Chip oscillator
- TTL/CMOS compatible
- +5V single power supply
- 67-pin flat package
- Si-gate CMOS LSI

PIN CONNECTIONS (Top View)



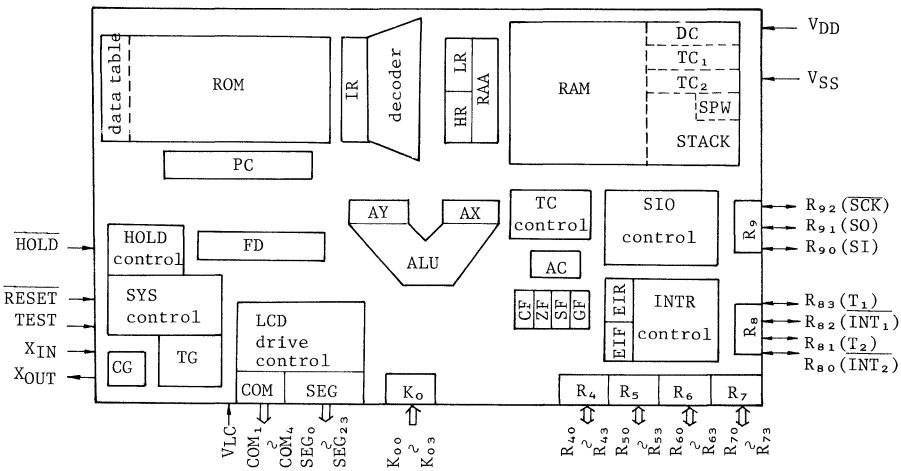
Note 1. Pin 27 is connected to pin 61 through external circuit.

2. N.C. No connection

PIN NAMES AND PIN DESCRIPTIONS

Pin Names	No.of Pins	I/O	Functions
$K_{03} \sim K_{00}$	4	Input	Input port
$R_{43} \sim R_{40}$	4	I/O	I/O port
$R_{53} \sim R_{50}$	4	I/O	"
$R_{63} \sim R_{60}$	4	I/O	"
$R_{73} \sim R_{70}$	4	I/O	"
R_{83} (T1)	1	I/O	I/O port or timer/counter input
R_{82} ($\overline{INT1}$)	1	I/O	I/O port or interrupt input
R_{81} (T2)	1	I/O	I/O port or timer/counter input
R_{80} ($\overline{INT2}$)	1	I/O	I/O port or interrupt input
R_{92} ($\overline{SCK}$)	1	I/O	I/O port or shift clock for serial port
R_{91} (SO)	1	I/O	" or serial output
R_{90} (SI)	1	I/O	" or serial input
$SEG_{23} \sim SEG_0$	24	Output	LCD Segment driver output
$COM_4 \sim COM_1$	4	Output	LCD Common driver output
X_{IN}, X_{OUT}	2	Input, Output	Resonator connection terminal
$\overline{RESET}$	1	Input	Initialize signal input
HOLD	1	Input	Hold signal input
TEST	1	Input	(Low level is input.)
V_{DD}	1	Power supply	+5V
V_{SS}	1	Power supply	0V
V_{LC}	1	Power supply	LCD drive power supply

BLOCK DIAGRAM



BLOCK NAME AND DESCRIPTION

Block Names	Functions
PC	Program counter (12 bits)
ROM	Program memory
IR, decoder	Instruction register, Decoder
HR, LR	H register (Page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register).
RAA	RAM address buffer register (8 bits)
RAM	Data memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC, data table	Data counter (12 bits, RAM area), Data table (ROM area)
AX, AY	Tempoerary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG(CF,ZF,SF,GF)	Flags
K, R	Ports
INTR control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC_1, TC_2	12-bit timer/counter 2 channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
LCD drive control (COM,SEG)	LCD drive control
HOLD control	Control of hold function
SYS CONTROL	Generation of various internal control signals
CG, TG	Clock generator, timing generator

FUNCTIONAL DESCRIPTION

Concerning the TMP47C22F, the configuration and functions of hardwares are described.

As the description has been provided with priority on those parts differing from the TMP47C20P (The TLCS-47C standard chip), the technical material for the TMP47C20P shall also be referred to.

1. System Configuration

1.1. Program Memory (ROM)

The TMP47C22F is in 32 page configuration in a unit of 64 words per page with the built-in 12 bit program counter and 2,048 × 8 bits (000 ∿ 7FF addresses) program memory.

Further, as the TMP47C22F has no built-in output ports P_1 and P_2, the instruction (OUTB @HL) and PLA data conversion table cannot be used.

The relationship between ROM capacity and addresses is shown in Fig. 1.2.1.

1.2 Data Memory (RAM)

The TMP47C22F contains a data memory with 192 x 4-bit (addresses 00 ∿ BF) and is in 12 pages configuration in a unit of 16 words per page.

On the other hand, since RAM address buffer register (RAA) has 8-bit length, addresses C0 ∿ FF have no physical RAM, but the higher order 2 bits (RAA_7 and RAA_6) are decoded to [(00), (01) and (1*). * denotes "don't care."]; therefore, when addresses C0 ∿ FF are accessed on a program, RAM equivalent to addresses 80 ∿ BF is accessed. In other words, on a program a specific address of RAM is addressed to addresses C0 ∿ FF, while on the TMP 47C22F, RAM equivalent to addresses 80 ∿ BF is allocated.

The relationship between RAM capacity and addresses is shown in Fig. 1.2.1.

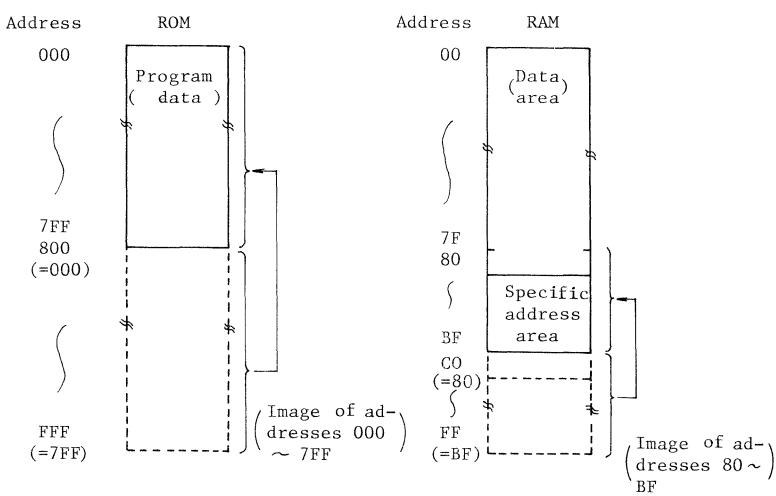


Fig. 1.2.1 ROM/RAM Capacity and Addresses

1.3 Port (PORT)

Data transfer to/from the external circuitry, and command/status/data transfer between of the built-in peripheral circuitry are carried out by the input/output instructions.

The details to specify the input/output circuit format of ports and initialization of the output latches are 2.3 Input/Output Port (Input/Output Circuit Format).

1.4 Timer/counter (TC_1, TC_2)

Two channels of 12-bit binary counter is contained to count time or event.

The maximum frequency applied to the external input pin under the event counter mode is dependent upon the operating state of the LCD drive circuit.

(a) At time of blanking operation

Frequency applied at time of a single channel operation is fc/64Hz. When 2 channels are operated simultaneously, timer/counter 1 is fc/64Hz and timer/counter 2 is fc/80Hz.

(b) When LCD display is enabled

Frequency applied at time of a single channel operation is fc/128Hz. When 2 channels are operated simultaneously, both timer/counter 1 and timer/counter 2 are fc/144Hz.

1.5 LCD Drive Circuit (LCDC)

The TMP47C22F has the built-in circuit that directly drives the liquid crystal display (LCD) and its control circuit.
The TMP47C22F has the following connecting terminals with LCD:

(a) Common output terminals (COM_1 - COM_4)

(b) Segment output terminals (SEG_0 - SEG_{23})

In addition, V_{LC} terminal is provided as the drive power terminal.

As display data transfer operations to the drive circuit are entirely executed by the hardware automatically on the TMP47C22F, it is possible to illuminate LCD if only display data is stored in the data memory.

The devices that can be directly driven is selectable from LCD devices of following drive methods:

(a) 1/4 duty (1/3 bias) LCD
Max. 96 segments (12 digits x 8 segments) can be driven.

(b) 1/3 duty (1/3 bias) LCD
Max. 72 ssgments (8 digits x 9 segments) can be driven.

(c) 1/2 duty (1/2 bias) LCD
Max. 48 segments (6 digits x 8 segments) can be driven.

(d) Static LCD
Max. 24 segments (3 digits x 8 segments) can be driven.

(1) Circuit configuration

The LCD drive circuit consists of the function blocks shown in Fig. 1.5.1.

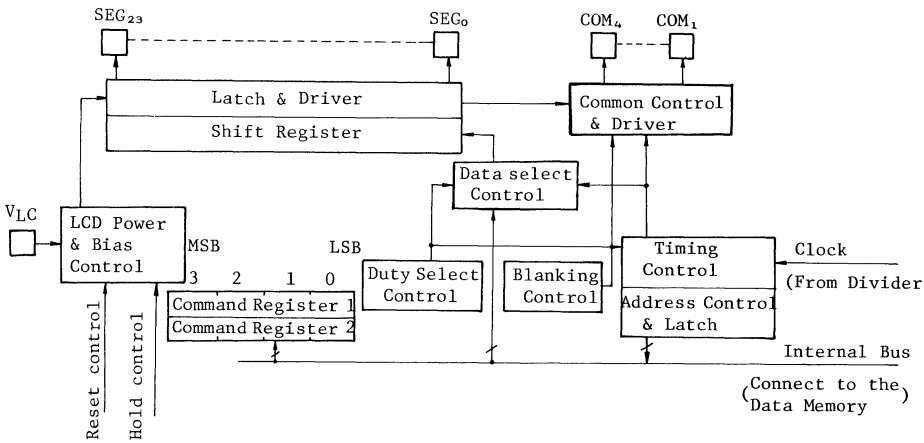


Fig. 1.5.1 LCD Drive Circuit

(2) Control of drive circuit

The operation of LCD drive circuit is controlled by the command. The command registers are accessed as port addresses OP1A and OP1B, and are reset to "8" and "0" at initialization, respectively.

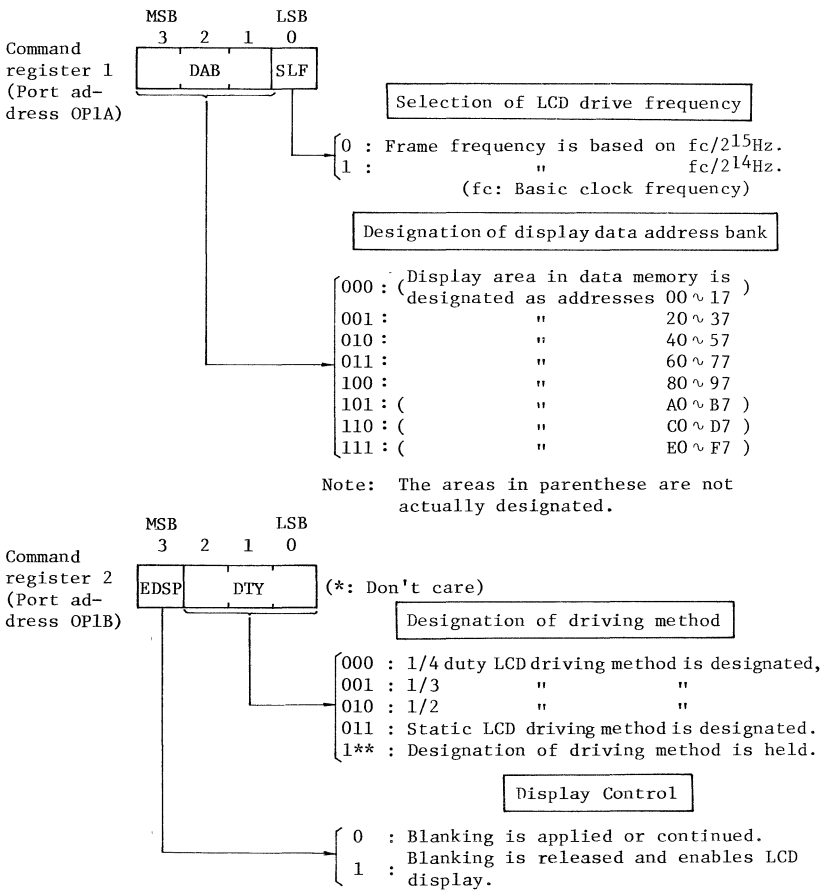


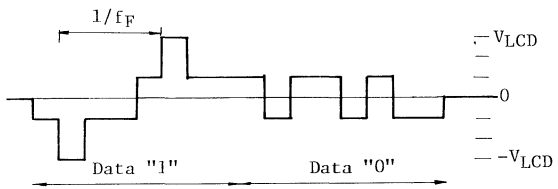
Fig. 1.5.2 Control of Drive Circuit

Drive waveform of LCD

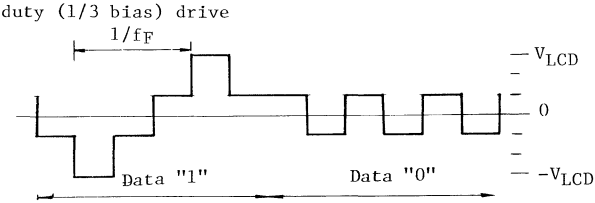
The LCD drive method is selected according to DTY of command register 2. DTY is reset to "0" at initialization.

The drive method is initialized according to a LCD used in the initial program. (In the case of a 1/4 duty LCD, it is set at initialization.) Thereafter, DTY sets disable code only.

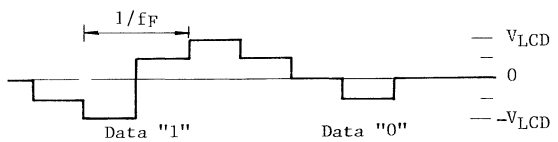
Examples of LCDs and their drive waveforms are shown in Fig. 1.5.3.



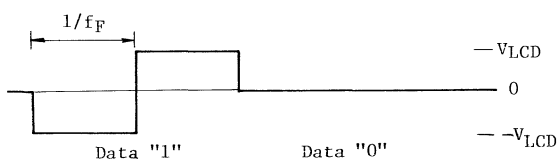
(a) 1/4 duty (1/3 bias) drive



(b) 1/3 duty (1/3 bias) drive



(c) 1/2 duty (1/2 bias) drive



(d) Static drive

(Note) f_F : LCD Frame frequency, $V_{LCD}=V_{DD}-V_{LC}$

Fig. 1.5.3 LCD Drive Waveform (COM-SEG Terminals)

LCD Frame frequency

Frame frequency (LCD drive frequency) is given by the built-in frequency divider. It is possible to select base frequency (either one of 2 kind frequencies obtained from the divider) by SLF of command register 1. SLF is reset to "0" at the initialization.

Frame frequency (f_F) is set according to the drive method and base frequency as shown in the following table:

SLF	Base frequency(Hz)	Frame Frequency (Hz)			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
0	$\frac{f_c}{2^{15}}$	$\frac{f_c}{2^{15}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{15}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{15}}$	$\frac{f_c}{2^{15}}$
	(f_c=4 MHz)	≒ 122	≒ 163	≒ 244	≒ 122
1	$\frac{f_c}{2^{14}}$	$\frac{f_c}{2^{14}}$	$\frac{4}{3} \cdot \frac{f_c}{2^{14}}$	$\frac{4}{2} \cdot \frac{f_c}{2^{14}}$	$\frac{f_c}{2^{14}}$
	(f_c=2 MHz)	≒ 122	≒ 163	≒ 244	≒ 122

(f_c: Basic clock frequency)

Table 1.5.1 LCD Frame Frequency Setting

LCD drive voltage

The V_{LC} terminal is the LCD drive power terminal. LCD drive voltage (V_{LCD}) is given by $V_{DD} - V_{LC}$. Therefore, if CPU operating voltage and LCD drive voltage are same, connect the V_{LC} terminal to the V_{SS} terminal.

Drive voltage applied to the LCD drive circuit is internally turned ON/OFF according to the operating state of CPU. That is, at the time of initialize operation and hold operation, the built-in power switch is automatically turned off to cut off drive voltage.

The LCD power switch turned off by the initialize operation is automatically turned on when EDSP (MSB of command register 2) is set at "1" and voltage is applied to the drive circuit. Thereafter, as the power switch is not turned off by the blanking control by means of a program, drive voltage is kept applied to the drive circuit.

On the other hand, the power switch is also turned off at the time of the hold operation, LCD display is turned off and the hold operation is executed at low power consumption. After the hold is released, the TMP47C22F is automatically returned to the state immediately before the hold operation was started.

Further, when the built-in power switch is OFF, V_{DD} level voltage is generally at either COM terminals or SEG terminals.

Display data setting

Display data is stored in the display area (max. 24 words) in the data memory. The conversion process of ordinary data into LCD display data is executed by instructions (ROM data referring instruction is mainly used.).

Display data converted and stored in the display area is automatically transfered to the LCD drive circuit and displayed by the hardware without any participation by a program. Therefore, change of display pattern is possible by changing only data in the display area in the data memory by a program.

The LCD segment (dot) corresponds to each bit in the display area in the data memory on the one-for-one basis. This relation is shown in Fig. 1.5.4.

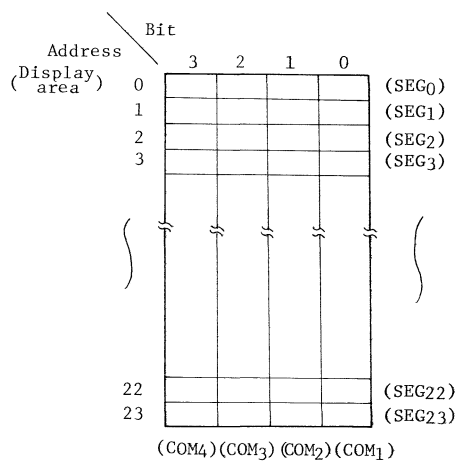


Fig. 1.5.4 LCD Diaply Data Area (Data Memory)

Where, each bit of the display data memory shows data of segment (dot) equivalent to SEGi, COMj ($0 \leq i \leq 23$, $1 \leq j \leq 4$), and when data is "1", the LCD illuminates.

Number of segments that can be driven varies depending upon the LCD drive method. This denotes that even in the display area of the data memory, number of bits used for storing display data varies.

(a) 1/4 duty LCD (COM_3 - COM_1 are used)
All bits in the display area becomes display data.

(b) 1/3 duty LCD (COM_3 - COM_1 are used)
Bit 2 - Bit 0 only become display data.

(c) 1/2 duty LCD (COM_2 - COM_1 are used)
Bit 1 and Bit 0 only become display data.

(d) Static LCD (COM_1 only is used)
Bit 0 only becomes display data.

Therefore, the data memory bits that are not used for storing display data or are equivalent to addresses to which no LCD is connected in the display area can be used for storing ordinary user's processing data.

As stated above, the data memory is used for storing display data (max. 24 words), and it is possible to set an address space in the data memory, to which this display area is to be set, by DAB of command register 1 (See Fig. 1.5.2.).

As the command register 1 is reset at "8" at initialization, the display area is initialized to 80 - 97 addresses.

Transfer of display data

Display data that has been set in the display area of the data memory is automatically transfered to the drive circuit. This operation is executed in the following sequence.

A display data transfer request is sent from the LCD drive circuit to CPU. Upon completion of an instruction under execution (if the timer/counter processing and the interrupt acceptance processing exist, after they are executed), CPU sends segment (dot) data in the display data area to the drive circuit in one instruction cycle.

This data sending cycle is taken place when drive voltage is kept applied to the LCD display drive circuit. Therefore, after intialize operation, this cycle is not taken place until EDSP is set to "1". Frequency of data sending cycle insertion is as follows:

(a) In case of other than static drive at SLF=0, 24 times in 512 instruction cycles.

(b) In case of static drive at SLF=0, 24 times in 2,048 instruction cycles.

(c) In case of other than static drive at SLF=1, 24 times in 256 instruction cycles.

(d) In case of static drive at SLF=1, 24 times in 1,024 instruction cycles.

Therefore, when LCD display is enable, the apparent speeds in above cases are decreased by 4.9, 1.2, 10.3 and 2.4%, respectively. For instance, in case of other than the static drive at SLF=0. The apparent speed is 4.2 μs to 4 μs instruction execution speed.

Blanking Operation

When EDSP (MSB of command register 2) is reset to "0", the LCD display becomes blank. EDSP is reset to "0" at initialization.

The blanking operation turns off the LCD by conditioning non-lighting operation level voltage to COM terminals. On the other hand, the SEG terminals are kept continued at normal operating state. (In the case of static drive, no voltage is applied to COM-SEG terminals when the LCD is turned off by data, however, as the blanking operation keeps the COM terminal at constant $V_{LCD}/2$ level, the LCD is turned off and the state between COM-SEG terminals where the LCD is driven by $V_{LCD}/2$. Therefore, note that the display state is somewhat different in these cases.) For drive waveforms, refer to Fig. 1.5.6 - Fig. 1.5.9.

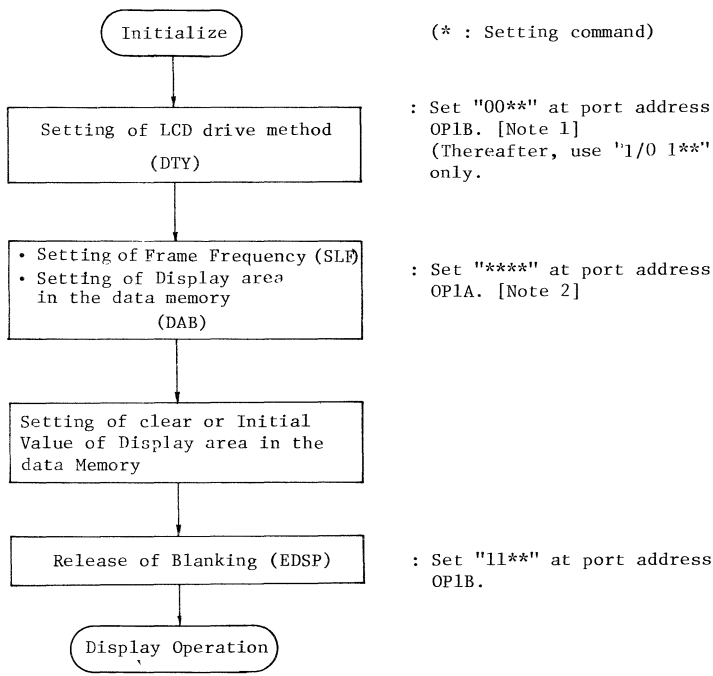
When EDSP is set at "1", the LCD display is enabled and the LCD display is made according to data stored in the display area of the data memory.

Further, when EDSP is initially set at "1" after the initialization, the LCD power switch is also turned ON and drive voltage is applied to the drive circuit.

LCD Display Control by Program

Provided that EDSP has been set at "1", the LCD is automatically turned ON according to data stored in the display area of the data memory. However, prior to actual display operation it is normally necessary to initialize as shown in Fig. 1.5.5.

To drive the 1/4 duty LCD, 80 - 97 addresses in the display area of the data memory are used, and to operate it at SLF = 0 (low speed operation), when EDSP is set to "1" after initialization of data in the display area, the display operation is started.



[Note 1] Classification of commands for port address OP1B.

"0000" ∿ "0011" : Setting of LCD drive method
"01**" : Blanking by program
"11**" : Releasing of blanking (display enable)
"10**" : Cannot be used

[Note 2] Normally, only one time of setting is required at the time of initialization, but as an exception, commands should be set at port address OP1A under the blanking state whenever the display area are switched.

Fig. 1.5.5 Initialization of LCD Drive by Program

Examples of display data when a numeral display is made by using the 1/4 duty LCD are shown in Table 1.5.2. For the connecting method of COM terminals and SEG terminals, the example shown in Fig. 1.5.6 is used.

Numeral	Display	Display data memory		Numeral	Display	Display data memory	
		High order address	Low order address			High order address	Low order address
0		1 1 0 1	1 1 1 1	5		1 0 1 1	0 1 0 1
1		0 0 0 0	0 1 1 0	6		1 1 1 1	0 1 0 1
2		1 1 1 0	0 0 1 1	7		0 0 0 1	0 1 1 1
3		1 0 1 0	0 1 1 1	8		1 1 1 1	0 1 1 1
4		0 0 1 1	0 1 1 0	9		1 0 1 1	0 1 1 1

Table 1.5.2 Examples of Display Data (1/4 Duty LCD)

Further, examples of display data when a numeral display similar to Table 1.5.2 is made by using the 1/3 duty LCD are shown in Table 1.5.3. For the connecting method of COM terminals and SEG terminals, the example shown in Fig. 1.5.7 is used.

Numeral	Display data memory			Numeral	Display data memory		
	High order address	Middle order address	Low order address		High order address	Middle order address	Low order address
0	* * 1 1	* 1 0 1	* 1 1 1	5	* * 0 1	* 1 1 1	* 0 1 0
1	* * 0 0	* 0 0 0	* 0 1 1	6	* * 1 1	* 1 1 1	* 0 1 0
2	* * 1 0	* 1 1 1	* 0 0 1	7	* * 0 1	* 0 0 1	* 0 1 1
3	* * 0 0	* 1 1 1	* 0 1 1	8	* * 1 1	* 1 1 1	* 0 1 1
4	* * 0 1	* 0 1 0	* 0 1 1	9	* * 0 1	* 1 1 1	* 0 1 1

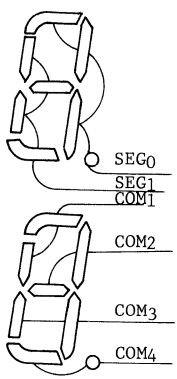
(* : don't care)

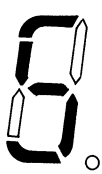
Table 1.5.3 Examples of Display Data (1/3 Duty LCD)

Display Output

The following are the examples of display output from LCD drive circuit according to each drive method.

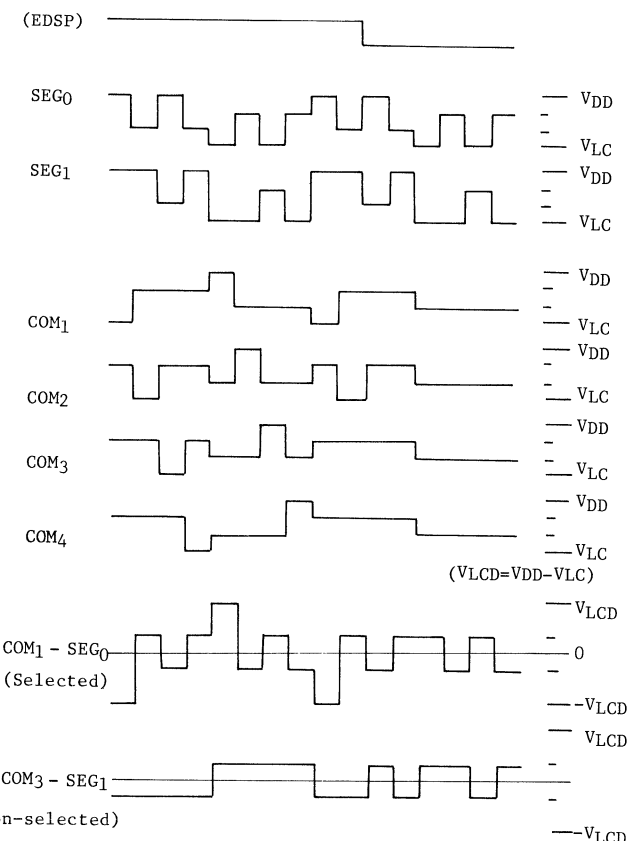
1/4 Duty (1/3 Bias) Drive





(Display data memory)

Address	Data
0	0 1 0 1
1	1 0 1 1

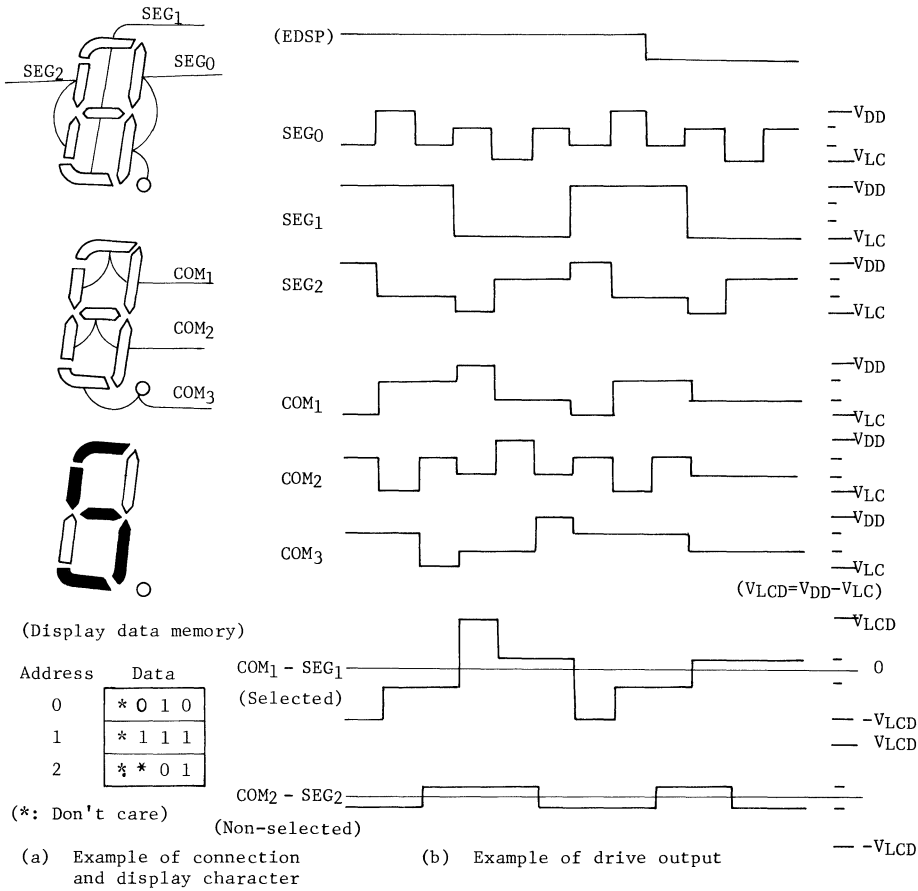


(a) Example of connection and display character

(b) Example of drive output

Fig. 1.5.6 Example of 1/4 Duty LCD Display Output

1/3 Duty (1/3 Bias) Drive



(Display data memory)

Address	Data
0	* 0 1 0
1	* 1 1 1
2	* * 0 1

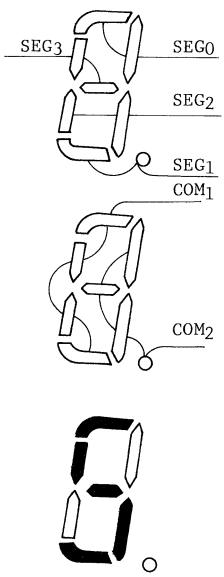
(*: Don't care)

(a) Example of connection and display character

(b) Example of drive output

Fig. 1.5.7 Example of 1/3 Duty LCD Display Output

1/2 Duty (1/2 Bias) Drive

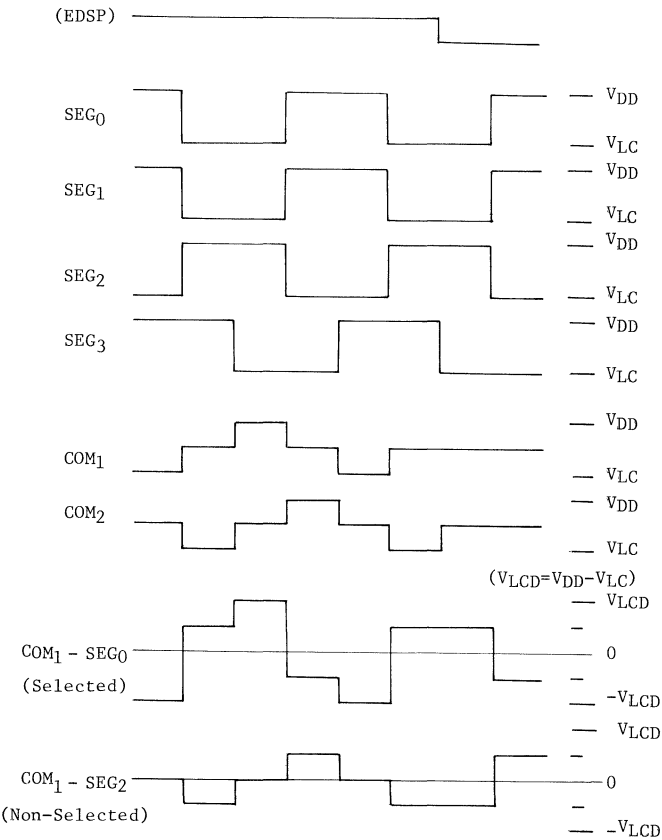


(Display data memory)

Address	Data
0	* * 0 1
1	* * 0 1
2	* * 1 0
3	* * 1 1

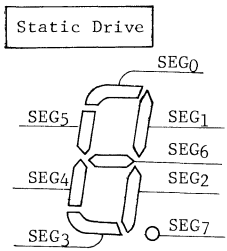
(*: Don't care)

(a) Example of connection and display character



(b) Example of drive output

Fig. 1.5.8 Example of 1/2 Duty LCD Display Output

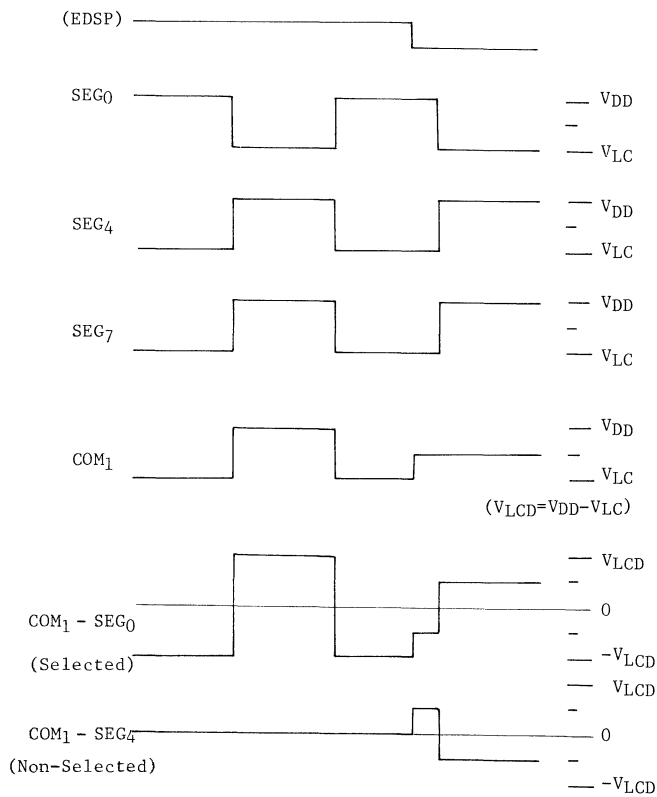


(Display data memory)

Address	Data
0	* * * 1
1	* * * 0
2	* * * 1
3	* * * 1
4	* * * 0
5	* * * 1
6	* * * 1
7	* * * 0

(*: Don't care)

(a) Example of connection and display character



(b) Example of drive output

Fig. 1.5.9 Example of Static LCD Display Output

2. Basic operation and pin operation

2.1 Instruction cycle, basic clock generation

As the oscillation circuit has been built in, when the external (X_{IN}, X_{OUT}) are connected to the oscillator, required clocks can be easily obtained. Further, this oscillation circuit is the Schmitt circuit. The clocks obtainable from the oscillation circuit are called the basic clock (CP, fc Hz). The basic clock is input into the timing generator and system control circuit from where various control signals are generated.

The instruction execution and the internal hardware control are synchronized with the basic clock. An instruction cycle consists of four machine cycles ($M_1 \sim M_4$), and each machine cycle requires four basic clock times.

2.2 Initialization operation, Hold function, interrupt input and others

Initialization operation is performed by keeping the $\overline{\text{RESET}}$ pin to the low level. By this initialize operation, the internal registers are initialized and at the same time, the LCD power switch is turned OFF. Further, no pull-up resistor is built in the $\overline{\text{RESET}}$ termianl of the TMP47C22F.

The hold function is the function to hold the status just before the system operation is stopped at low power consumption by making the most of the features of CMOS. The $\overline{\text{HOLD}}$ terminal is the signal input for the hold operation request and hold operation release request.

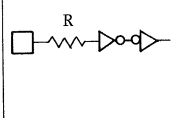
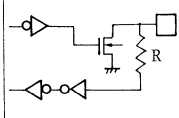
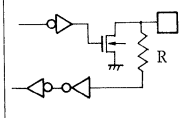
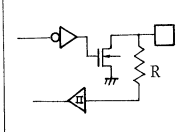
Two pins ($\overline{INT_1}$, $\overline{INT_2}$) are provided for the external interrupt input. Since these pins are common pins with K_8 port, they can be used as I/O pins respectively, if not used as the interrupt input pins. The interrupt latch is set by the falling edge of the external interrupt inputs.

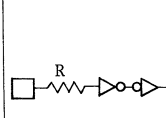
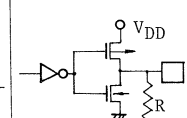
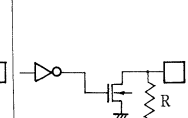
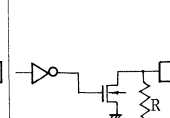
The TEST terminal is used at time of the shippint test. When a user's system is to be operated, low level voltage should be positively applied. Further, the TEST terminal of the TMP 47C22F has no built-in pull-down resistor.

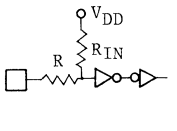
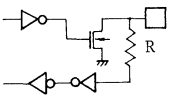
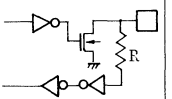
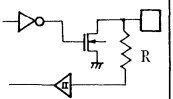
2.3 Input/Output port

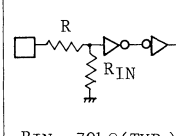
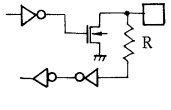
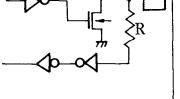
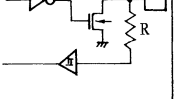
Input/Output Circuit Format

The input/output circuit format of the input/output port is shown following. For the TMP47C22F, any of the input/output circuit systems shown in the following tables can be selected. You can specify your input/output circuit system when requesting the program tape.

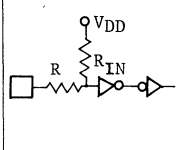
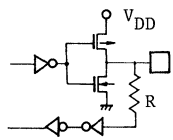
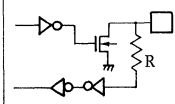
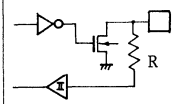
Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) GA				
Cir-cuit \ Port	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equiv-alent cir-cuit	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained	o Sink open drain output o Output latch is ini-tialized to the high level	o Sink open drain output o Output latch is ini-tialized to the high level	o Schmitt cir-cuit input o Sink open drain output o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) GD				
Cir-cuit \ Port	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equiv-alent cir-cuit	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained	o Push-pull output o Output latch is ini-tialized to the high level	o Sink open drain output o Output latch is ini-tialized to the high level	o Schmitt cir-cuit input o Sink open drain output o Output latch is initialized to the high level

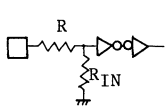
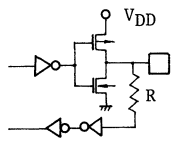
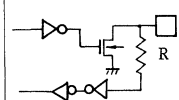
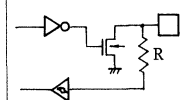
Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) GB				
Port / Circuit	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN} = 70kΩ(TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-up resistor is contained	o Sink open drain output o Output latch is initialized to the high level	o Sink open drain output o Output latch is initialized to the high level	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{O}C\overline{O}DE$) GC				
Port / Circuit	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/P (R_8, R_9)
I/O equivalent circuit	 R_{IN} = 70kΩ(TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-down resistor is contained	o Sink open drain output o Output latch is initialized to the high level	o Sink open drain output o Output latch is initialized to the high level	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{OC}\overline{ODE}$) GE

Port / Circuit	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN} = 70kΩ(TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-up resistor is contained	o Push-pull output o Output latch is initialized to the high level	o Sink open drain output o Output latch is initialized to the high level	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level

Input/Output Circuit Code ($I\overline{OC}\overline{ODE}$) GF

Port / Circuit	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R_{IN}=70kΩ(TYP.) R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-down resistor is contained	o Push-pull output o Output latch is initialized to the high level	o Sink open drain output o Output latch is initialized to the high level	o Schmitt circuit input o Sink open drain output o Output latch is initialized to the high level

TENTATIVE

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS}=0V)

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.5 ∿ 7	
V_{LC}	Supply Voltage (LCD Drive)	-0.5 ∿ V_{DD}+0.5	
V_{IN}	Input Voltage	-0.5 ∿ V_{DD}+0.5	V
V_{OUT1}	Output Voltage (Except open drain terminal)	-0.5 ∿ V_{DD}+0.5	V
V_{OUT2}	Output Voltage (Open drain terminal)	-0.5 ∿ 10	
P_D	Power Dissipation (T_{opr}=70°C)	400	mW
T_{sld}	Soldering Temperature · Time	260 (10 sec.)	°C
T_{stg}	Storage Temperature	-55 ∿ 125	
T_{opr}	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS}=0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNIT
T_{opr}	Operating Temperature		-30	70	°C
V_{DD}	Supply Voltage		4.5	6	V
V_{DDH}	Supply Voltage (Hold)		2	6	
V_{LC}	Supply Voltage (LCD Drive)		0	V_{DD}-2.7	
V_{IH1}	High Level Input Voltage (Except Schmitt circuit input)	$V_{DD} \geq 4.5V$	V_{DD}×0.7	V_{DD}	V
V_{IH2}	High Level Input Voltage (Schmitt circuit input)		V_{DD}×0.75	V_{DD}	
V_{IH3}	High Level Input Voltage	$V_{DD} < 4.5V$	V_{DD}×0.9	V_{DD}	
V_{IL1}	Low Level Input Voltage (Except Schmitt circuit input)	$V_{DD} \geq 4.5V$	0	V_{DD}×0.3	
V_{IL2}	Low Level Input Voltage (Schmitt circuit input)		0	V_{DD}×0.25	
V_{IL3}	Low Level Input Voltage	$V_{DD} < 4.5V$	0	V_{DD}×0.1	
f_C	Clock Frequency		0.4(Note2)	4.2	MHz
t_{WCH}	High Level Clock Pulse Width(Note 1)	V_{IN}=V_{IH}	80	-	nS
t_{WCL}	Low Level Clock Pulse Width (Note 1)	V_{IN}=V_{IL}	80	-	

(Note 1) For external clock operation

(Note 2) 1MHz is recommended as minimum frequency when SLF=1.
And 2MHz is when SLF=0.

D.C. CHARACTERISTICS (V_{SS}=0V, V_{DD}=5V±10%, T_{opr}=-30 ∿ 70°C)

SYMBOL	PARAMETER		CONDITION	MIN.	TYP.(NOTE.1)	MAX.	UNIT
V_{HS}	HYSTERESIS VOLTAGE (SCHMITT CIRCUIT INPUT)			—	0.7	—	V
I_{IN1}	INPUT CURRENT (KO, $\overline{RESET}$, $\overline{HOLD}$, TEST) (NOTE2)		V_{DD}=5.5V, V_{IN}=5.5/0V	—	—	±20	μA
I_{IN2}	INPUT CURRENT (OPEN DRAIN R PORT)			—	—	±20	
I_{IL}	LOW LEVEL INPUT CURRENT (PUSH-PULL R PORT)		V_{DD}=5.5V, V_{IN}=0.4V	—	—	-2	mA
R_{IN}	INPUT RESISTANCE (KO WITH INPUT RESISTOR)			30	70	150	kΩ
I_{LO}	OUTPUT LEAKAGE CURRENT (OPEN DRAIN R PORT)		V_{DD}=5.5V, V_{OUT}=5.5V	—	—	20	μA
V_{OH}	OUTPUT VOLTAGE	HIGH LEVEL (PUSH-PULL R PORT)	V_{DD}=4.5V, I_{OH}=-200μA	2.4	—	—	V
V_{OL}		LOW LEVEL (R PORT)	V_{DD}=4.5V, I_{OL}=1.6mA	—	—	0.4	
R_{OS4}, R_{OS0}	OUTPUT IMPEDANCE	HIGH·LOW LEVEL (SEG)(NOTE.4,5)	V_{DD}=5V, V_{LCD}(NOTE.3)=3V	—	10	--	kΩ
R_{OC4}, R_{OC0}		HIGH·LOW LEVEL (COM)(NOTE.4,5)	V_{OUT}=V_{DD}-0.5V/V_{LC}+0.5V	—	10	--	
R_{OS3}, R_{OS1}		2/3,1/3 LEVEL (SEG)(NOTE.4)	V_{DD}=5V, V_{LCD}=3V	—	10	—	
R_{OC3}, R_{OC1}		2/3,1/3 LEVEL (COM)(NOTE.4)	V_{OUT}=4-0.5V/3+0.5V	—	10	--	
R_{OS2}		1/2 LEVEL (SEG)(NOTE.5)	V_{DD}=5V, V_{LCD}=3V	—	10	—	
R_{OC2}		1/2 LEVEL (COM)(NOTE.5)	V_{OUT}=3.5±0.5V	—	10	--	
V_{O3}	OUTPUT VOLTAGE	2/3 LEVEL (SEG,COM)(NOTE.4)		4-0.2	4	4+0.2	V
V_{O2}		1/2 LEVEL (SEG,COM)(NOTE.5)	V_{DD}=5V, V_{LCD}=3V	3.5-0.2	3.5	3.5+0.2	
V_{O1}		1/3 LEVEL (SEG,COM)(NOTE.4)		3-0.2	3	3+0.2	
I_{DDO}	SUPPLY CURRENT (AT OPERATING) (NOTE.6)		V_{DD}=(V_{DDH})=5.5V, V_{LC}=V_{SS} f_C=4MHz	—	3	6	mA
I_{DDH}	SUPPLY CURRENT (AT HOLDING) (NOTE.6)		V_{IN}=5.3/0.2V(all valid) C_L=50pF, C_{XIN}=C_{XOUT}=10pF	—	0.5	20	μA

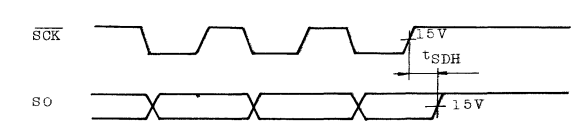
(NOTE.1) TYP.VALUES SHOW THOSE WHEN T_{OPR}=25°C, V_{DD}=5V.
(NOTE.2) WHEN THE KO PORT HAS A BUILT-IN INPUT RESISTER, CURRENT BY RESISTER IS EXCLUDED.
(NOTE.3) V_{LCD}=V_{DD}-V_{LC}.
(NOTE.4) SHOWS ON-RESISTANCE AT TIME OF LEVEL SWITCHING WHEN THE 1/4 OR 1/3 DUTY LCD IS USED.
(NOTE.5) SHOWS ON-RESISTANCE AT TIME OF LEVEL SWITCHING WHEN THE 1/2 DUTY OR STATIC LCD IS USED.
(NOTE.6) WHEN KO PORT HAS A BUILT-IN INPUT RESISTER, CURRENT VALUE IS THAT AT TIME OF OPEN. FURTHER, VOLTAGE LEVEL AT R PORT IS VALID.

A.C. CHARACTERISTICS (V_{SS}=0V, V_{DD}=5V±10%, T_{opr}=-30 ∿ 70°C)

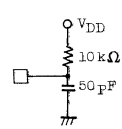
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	INSTRUCTION CYCLE TIME		3.8	—	4.0	μS
t_{SDH}	SHIFT DATA HOLD TIME	(NOTE 1)	0.5tcy-300	—	—	nS

A.C. TIMING CHART

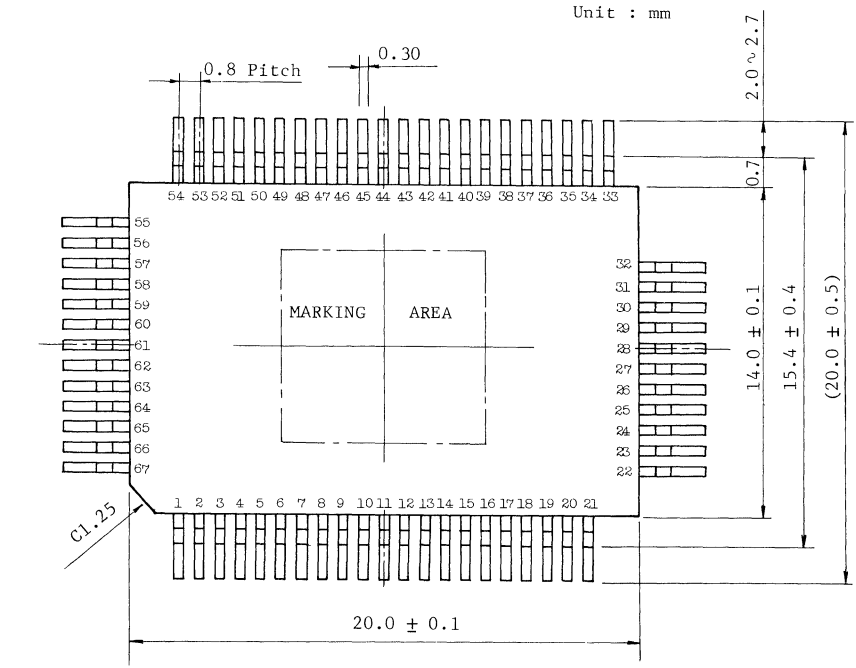
• SERIAL PORT (Completion of Transmission)

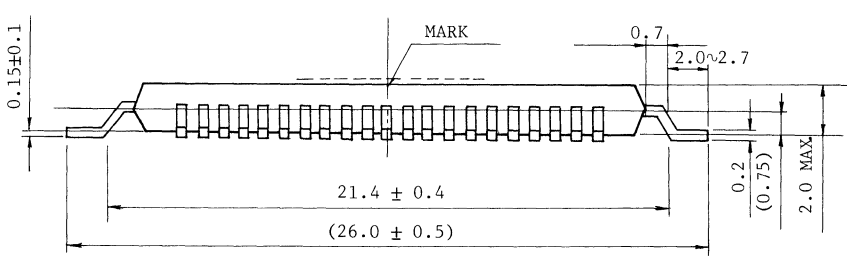


(NOTE.1) $\overline{SCK}$, SO TERMINAL EXTERNAL CIRCUIT



EXTERNAL DIMENSIONS





Weight 1.3g (TYP.)

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP47C46N

SILICON MONOLITHIC SILICON GATE CMOS

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER TLCS-47(C)

GENERAL DESCRIPTION

The TMP47C46N is the chip with built-in multiple I/O ports for which CMOS process have been employed.

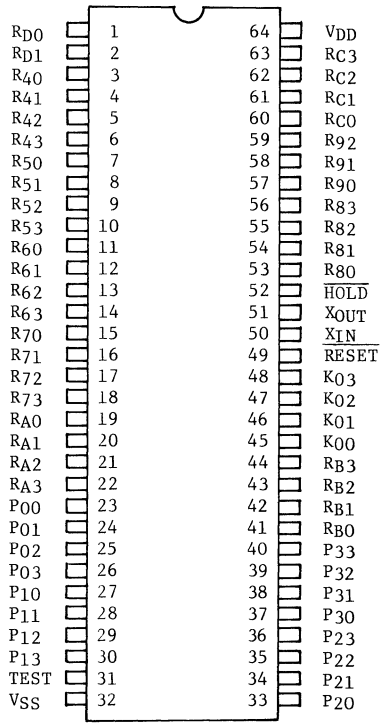
FEATURES

- TLCS-47 Family
- ROM 4,096 × 8 bits, RAM 256 × 4 bits
- Input/Output port (57 pins)

• Input	1 port	4 pins
• Output	2 ports	8 pins
• Output (corresponding to PLA)	2 ports	8 pins
• Input/output	8 ports	30 pins
• I/O (combined use)	2 ports	7 pins

- TTL/CMOS Compatible
- +5V single power supply
- 64-pin DIL plastic (Shrink-type) package

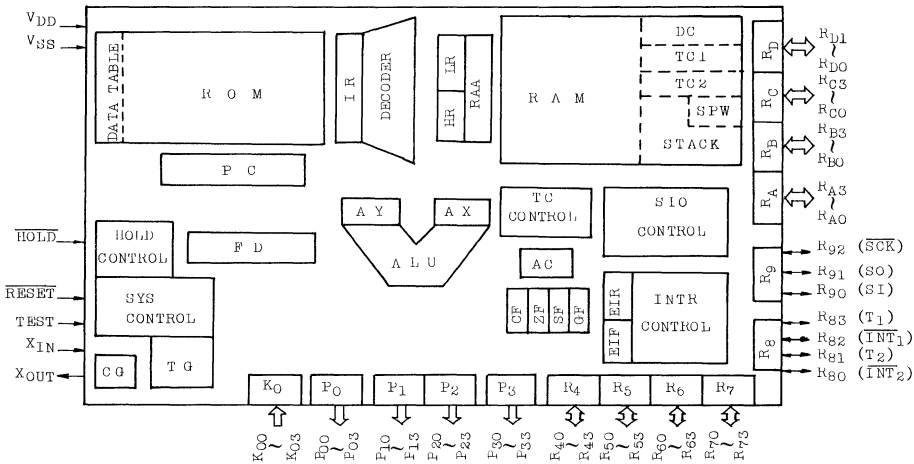
PIN CONNECTIONS (Top View)



Pin Name and Function

Pin Name	No.of pins	I/O	Function
$K_{03} \sim K_{00}$	4	Input	Input port
$P_{13} \sim P_{10}$	4	Output	Output port (corresponding to PLA)
$P_{23} \sim P_{20}$	4	Output	" (")
$P_{03} \sim P_{00}$	4	Output	Output port
$P_{33} \sim P_{30}$	4	Output	Output port
$P_{43} \sim P_{40}$	4	I/O	I/O port
$P_{53} \sim P_{50}$	4	I/O	"
$P_{63} \sim P_{60}$	4	I/O	"
$P_{73} \sim P_{70}$	4	I/O	"
$R_{A3} \sim R_{A0}$	4	I/O	"
$R_{B3} \sim R_{B0}$	4	I/O	"
$R_{C3} \sim R_{C0}$	4	I/O	"
$R_{D1} \sim R_{D0}$	2	I/O	"
$R_{83}(\overline{T1})$	1	I/O	I/O port or timer/counter input
$R_{82}(\overline{INT_1})$	1	I/O	I/O port or interrupt input
$R_{81}(\overline{T2})$	1	I/O	I/O port or timer/counter input
$R_{80}(\overline{INT_2})$	1	I/O	I/O port or interrupt input
$R_{92}(\overline{SCK})$	1	I/O	I/O port or shift clock I/O for serial port
$R_{91}(SO)$	1	I/O	I/O port or serial output
$R_{90}(SI)$	1	I/O	I/O port or serial input
XIN, XOUT	2	Input, Output	Oscillator connecting pin
$\overline{RESET}$	1	Input	Initialize signal input
$\overline{HOLD}$	1	Input	Hold signal input
TEST	1	Input	(Inputs low level or opened.)
V_{DD}	1	Power supply	+5V
V_{SS}	1		0V

BLOCK DIAGRAM



BLOCK NAMES AND DESCRIPTION

Block Name	Function
PC	Program counter (12 bits)
ROM	Program memory (including fixed data)
IR, decoder	Instruction register, Decoder
HR, LR	H register (page assignment of RAM), L register (address assignment in RAM page), (each 4-bit register)
RAA	RAM address buffer register (8 bits)
RAM	Data memory
STACK	Save area of program counter and flags (RAM area)
SPW	Stack pointer word (RAM area)
DC, data table	Data counter (12 bits, RAM area), Data table (ROM area).
AX, AY	Temporary register of ALU input
ALU	Arithmetic and logic unit
AC	Accumulator
FLAG (CF, ZF, SF, GF)	Flags
K, P, R	Ports
INTR control	Interrupt control (EIF: Enable interrupt master F/F, EIR: Enable interrupt register)
FD	Frequency divider (4-stage prescaler + 18 stages)
TC1, TC2	12-bit timer/counter 2 channels (RAM area)
TC control	Timer/counter control
SIO control	Serial port control
HOLD control	Control for hold function
SYS control	Generation of various internal control signals
CG, TG	Clock generator, Timing generator

PORT

The ports added to the TMP47C40P are explained as follows:

(1) $P_0(P_{03} \sim P_{00})$ Port

4-bit port exclusively used for output.
Latch data can not be read by instructions.

(2) $P_3(P_{33} \sim P_{30})$

4-bit port exclusively used for output.
Latch data can be read by instruction.

(3) $R_A(R_{A3} \sim R_{A0})$ Port
$R_B(R_{B3} \sim R_{B0})$ "
$R_C(R_{C3} \sim R_{C0})$ "
$R_D(R_{D1} \sim R_{D0})$ "

Each of $R_A \sim R_C$ ports is a 4-bit I/O port with a latch and R_D port is a 2-bit I/O port with a latch.
The latch should be set to "1" when the port is used as an input port.

Port Address	Symbol (Input/Output)	Port, Register (Input / Output)	Input / Output Instructions						
			IN %P, A IN %P, @HL	OUT A, %P OUT @HL, %P	OUT #K, %P	OUT B @HL	SET %P, b CLR %P, b	TEST %P, b TESTP %P, b	SET @L CLR @L TEST @L
0 0	IP00/OP00	K_0 Input port / P_0 Output port (Note)	0	0	0			0	
0 1	IP01/OP01	P_1 Output latch / P_1 Output port	0	0	0		0	0	
0 2	IP02/OP02	P_2 〃 / P_2 〃	0	0	0		0	0	
0 3	IP03/OP03	P_3 〃 / P_3 〃	0	0	0		0	0	
0 4	IP04/OP04	R_4 I/O Port	0	0	0		0	0	0
0 5	IP05/OP05	R_5 〃	0	0	0		0	0	0
0 6	IP06/OP06	R_6 〃	0	0	0		0	0	0
0 7	IP07/OP07	R_7 〃	0	0	0		0	0	0
0 8	IP08/OP08	R_8 〃	0	0	0		0	0	
0 9	IP09/OP09	R_9 〃	0	0	0		0	0	
0 A	IP0A/OP0A	R_A 〃	0	0	0		0	0	
0 B	IP0B/OP0B	R_B 〃	0	0	0		0	0	
0 C	IP0C/OP0C	R_C 〃	0	0	0		0	0	
0 D	IP0D/OP0D	R_D 〃	0	0	0		0	0	
0 E	IP0E/OP0E	Status input / —	0					0	
0 F	IP0F/OP0F	Serial buffer register (Reception) / Serial buffer register (Transmission)	0	0	0				

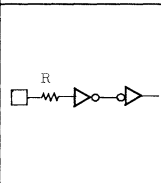
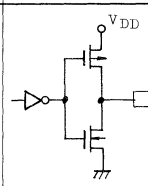
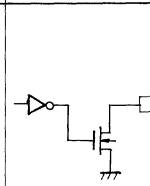
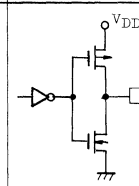
Note: IN instruction and TSET instruction operate K_0 port, OUT instruction operate P_0 port.

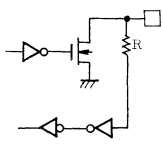
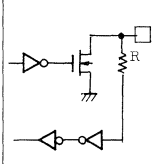
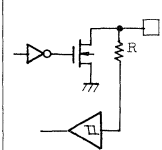
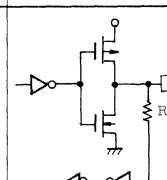
Input/Output circuit format

The input/output circuit format of the input/output port is shown following. For the TMP47C46N, any of the input/output circuit systems shown in the following tables can be selected. You can specify your input/output circuit system when requesting the program tape.

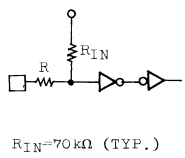
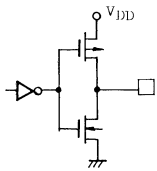
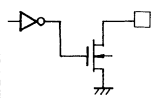
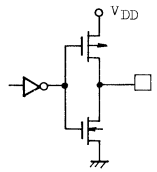
*: Port **. Circuit

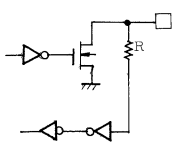
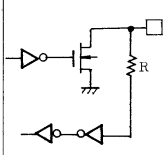
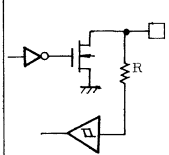
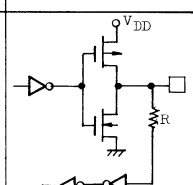
Input/Output Circuit Code (IOCODE) IA

* / **	Input (K_0)	Output (P_0)	Output (P_1, P_2)	Output (P_3)
Input/Output Circuit Format	 R=1kΩ (TYP.)			
Intialized Value of latch		low level	high level	low level
Remark	○ No resistor is contained	○ Push-pull output	○ Sink open drain output	○ Push-pull output

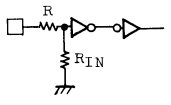
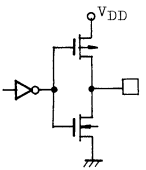
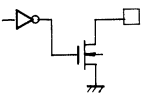
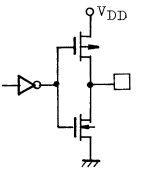
* / **	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)	I/O (R_A, R_B, R_C, R_D)
Input/Output Circuit Format	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)
Intialized Value of latch	high level	high level	high level	high level
Remark	○ Sink open drain output	○ Sink open drain output	○ Schmitt circuit input ○ Sink open drain output	○ Push-pull output

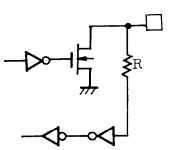
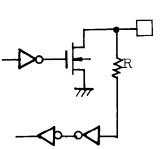
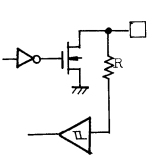
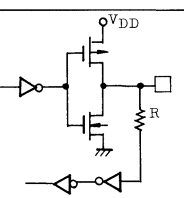
Input/Output Circuit Code (IOCODE) IB

* / **	Input (K_O)	Output (P_O)	Output (P_1, P_2)	Output (P_3)
Input/Output Circuit Format	 R_{IN}=70kΩ (TYP.) R = 1kΩ (TYP.)			
Intialized value of latch		low level	high level	low level
Remark	◦ Pull-up resistor is contained	◦ Push-pull output	◦ Sink open drain output	◦ Push-pull output

* / **	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)	I/O (R_A, R_B, R_C, R_D)
Input/Output Circuit Format	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)
Intialized value of latch	high level	high level	high level	high level
Remark	◦ Sink open drain output	◦ Sink open drain output	◦ Schmitt circuit input ◦ Sink open drain output	◦ Push-pull output

Input/Output Circuit Code (IOCODE) IC

** \ *	Input (K_O)	Output (P_O)	Output (P_1, P_2)	Output (P_3)
Input/Output Circuit Format	 R_{IN}=70kΩ (TYP.) R = 1kΩ (TYP.)			
Intialized Value of latch		low level	high level	low level
Remark	∘ Pull-down resistor is contained	∘ Push-pull output	∘ Sink open drain output	∘ Push-pull output

** \ *	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)	I/O (R_A, R_B, R_C, R_D)
Input/Output Circuit Format	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)
Intialized Value of latch	high level	high level	high level	high level
Remark	∘ Sink open drain output	∘ Sink open drain output	∘ Schmitt circuit input ∘ Sink open drain output	∘ Push-pull output

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS} = 0V)

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Power Supply Voltage	-0.5 ∿ 7	V
V_{IN}	Input Voltage	-0.5 ∿ V_{DD} +0.5	V
V_{OUT1}	Output Voltage (Except open drain pin)	-0.5 ∿ V_{DD} +0.5	V
V_{OUT2}	Output Voltage (Open drain pin)	-0.5 ∿ 10	
P_D	Power Dissipation (Topr = 70°C)	600	mW
Tsld	Soldering Temperature · Time	260 (10 sec.)	°C
Tstg	Storage Temperature	-55 ∿ 125	
Topr	Operating Temperature	-30 ∿ 70	

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V)

SYMBOL	PARAMETER	CONDITION	MIN.	MAX.	UNIT
Topr	Operating Temperature		-30	70	°C
V_{DD}	Power Supply Voltage		4.5	6	V
V_{DDH}	Power Supply Voltage (Hold)		2	6	
V_{IH1}	Input High Voltage (Except Schmitt circuit input)	$V_{DD} \geq 4.5V$	V_{DD}x0.7	V_{DD}	V
V_{IH2}	Input high Voltage (Schmitt circuit input)		V_{DD}x0.75	V_{DD}	
V_{IH3}	Input High Voltage	$V_{DD} < 4.5V$	V_{DD} x 0.9	V_{DD}	
V_{IL1}	Input Low Voltage (Except Schmitt circuit input)	$V_{DD} \geq 4.5V$	0	V_{DD}x0.3	
V_{IL2}	Input Low Voltage (Schmitt circuit input)		0	V_{DD}x0.25	
V_{IL3}	Input Low Voltage	$V_{DD} < 4.5V$	0	V_{DD}x0.1	
f_C	Clock Frequency		0.4	4.2	NHz
tWCH	Clock High Pulse Width (*)	$V_{IN} = V_{IH}$	80	-	nS
tWCL	Clock Low Pulse Width (*)	$V_{IN} = V_{IL}$	80	-	

(*) In case of external clock operation

D.C. CHARACTERISTICS (V_{SS}=0V, V_{DD}=5V±10%, T_{opr}=-30 ~ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	Note 1 TYP.	MAX.	UNIT
V_{HS}	Hysteresis Voltage (Schmitt circuit input)		-	0.7	-	V
I_{IN1}	Input Current (K_0, $\overline{HOLD}$) [Note 2]	V_{DD}=5.5V, V_{IN}=5.5/0V	-	-	±20	µA
I_{IN2}	Input Current (Sink open drain R port)		-	-	±20	
I_{IL}	Input Low Current (Push-pull R port)	V_{DD}=5.5V, V_{IN}=0.4V	-	-	-2	mA
R_{IN}	Input resistor (K_0 with input resistor)		30	70	150	kΩ
I_{LO}	Output Leak Current (Sink open drain P, R port)	V_{DD}=5.5V, V_{OUT}=5.5V	-	-	20	µA
V_{OH}	Output High Voltage (Push pull P, R port)	V_{DD}=4.5V, I_{OH}=-200µA	2.4	-	-	V
V_{OL}	Output Low Voltage (P, R port)	V_{DD}=4.5V, I_{OL}=1.6mA	-	-	0.4	
I_{DDO}	Operating Supply Current [Note 3]	V_{DD}(V_{DDH})=5.5V, f_C=4MHz, V_{IN}=5.3/0.2V (all valid), C_L=50pF, C_{XIN}=C_{XOUT}=10pF	-	3	6	mA
I_{DDH}	Holding Supply Current [Note 3]		-	0.5	20	µA

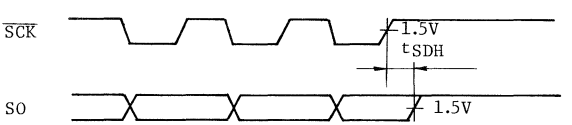
(Note 1) TYP. values show those when Topr=25°C, V_{DD}=5V.
(Note 2) When the K_0 port has a built-in input resistor, current by resistor is excluded.
(Note 3) When K_0 port has a built-in input resistor, current value is that at time of open. Further, voltage level at R port is valid.

A.C. CHARACTERISTICS (VSS = 0V, VDD = 5V±10%, Topr = -30 ~ 70°C)

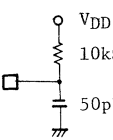
SYMBOL	Parameter	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	Instruction Cycle Time		3.8	-	40	µs
tSDH	Shift Data Hold Time	(Note 1)	0.5tcy-300	-	-	ns

A.C. TIMING CHART

• Serial Port (Completion of Transmission)

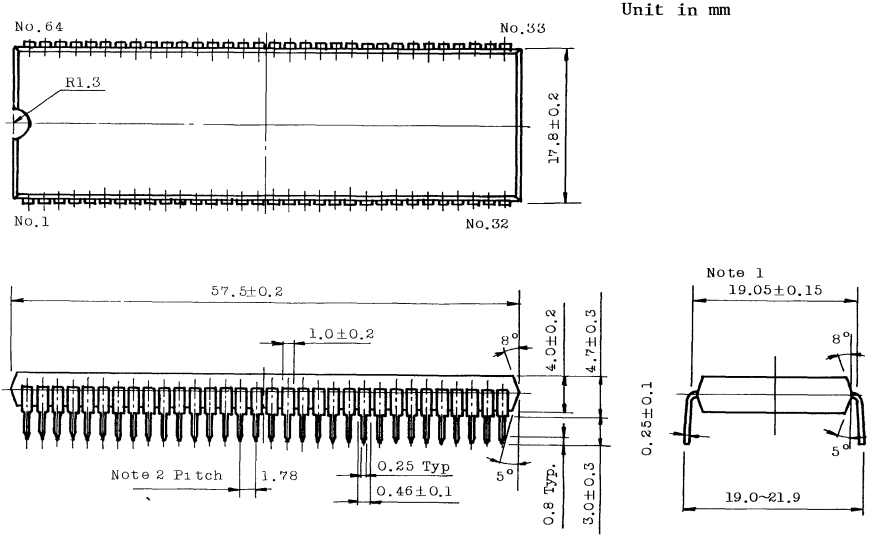


(Note 1) $\overline{SCK}$, SO terminal external circuit



EXTERNAL DIMENSION VIEW

Unit in mm



Weight 9.0g (TYP.)

Note 1. This dimension is measured at the center of bending point of leads.

Note 2. Each lead pitch is 1.78mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No.1 and No.64 leads.

POSTSCRIPT

This manual is a reference for the customer applying the TLCS-47 series. It contains the functions and specifications of each LSI device of the TLCS-47. The examples of application circuits described here are shown as the reference for explanation. Toshiba assumes no responsibility for any problems caused by using these examples. The information herein is subject to change without prior notice.

This manual has been prepared by the following section.

Microcomputer LSI Application Engineering Section
Integrated Circuit Division, Toshiba Corporation

1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 210, Japan
Phone: Japan (81)44-511-3111

PART 3

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-48

LSI DEVICES

July. 1 9 8 4

PREFACE

This part describes the detail functions and specifications of the LSI devices of the single chip microcontroller TLCS-48 series. The TLCS-48 series consists of NMOS and CMOS devices. These are pin and software compatible with each others. Basic devices of this series are the TMP8048P/49P and TMP8243P of NMOS and the TMP80C48AP/C49AP/C50AP and TMP82C43P of CMOS. Several versions are available for the conditions of operating temperature range, operating speed range, and external program ROM application.

CONTENTS

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-48

NMOS DEVICES

July. 1 9 8 4

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8048P/8048PI

TMP8035P/8035PI

N-CHANNEL SILICON GATE MOS

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP8048P, 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 × 8 RAM data memory, 1K × 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

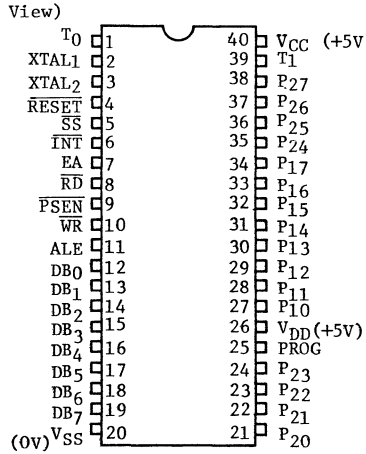
The TMP8048 is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

The TMP8035P 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.

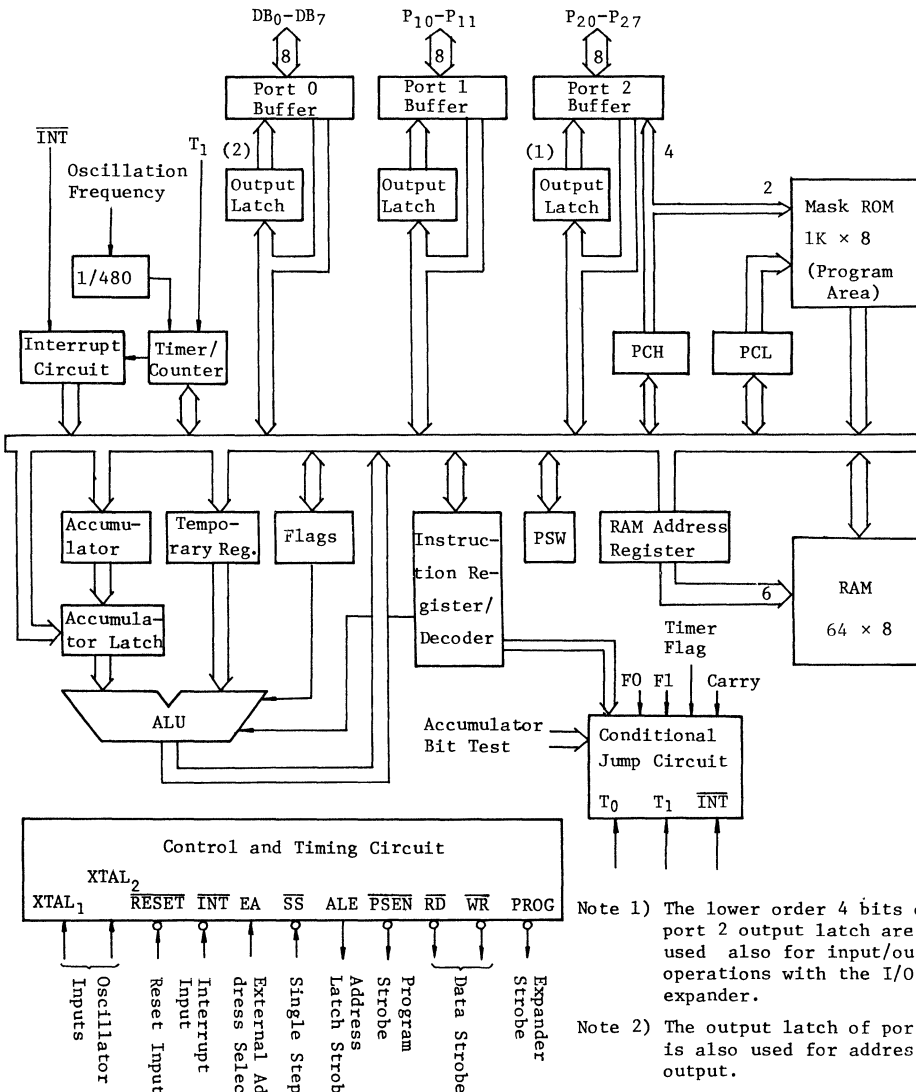
FEATURES

- Compatible with Intel's 8048
- 2.5 μS Instruction Cycle
- All instruction 1 or 2 cycles
- Over 90 instructions; 70% single byte
- Easy expandable memory and I/O
- 1K × 8 masked ROM
- 64 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Single level interrupt
- Single 5V supply
- -40°C to +85°C Operation (TMP8048PI/TMP8035PI : Industrial Specification)

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

V_{SS} (Power Supply)

Circuit GND potential

V_{DD} (Power Supply)

+5V during operation Low power standby pin for TMP8048 RAM

V_{CC} (Main Power Supply)

+5V during operation

PROG (Output)

Output strobe for the TMP8243P I/O expander

P_{10}-P_{17} (Input/Output) Port 1

8-bit quasi-bidirectional port (Internal Pullup $\cong$ 50KΩ).

P_{20}-P_{27} (Input/Output) Port 2

8-bit quasi-bidirectional port (Internal Pullup $\cong$ 50KΩ).

P_{20}-P_{23} 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.

DB_0 -DB_7 (Input/Output, 3 State)

Ture 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 $\overline{WR}$.

T_0 (Input/Output)

Input pin testable using the conditional transfer instructions JT0 and JNT0. T_0 can be designated as a clock output using ENT0 CLK instruction.

T_1 (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 executed.

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

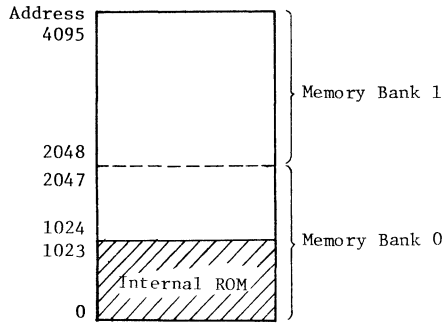
The following system functions of the TMP8048 are described in detail.

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

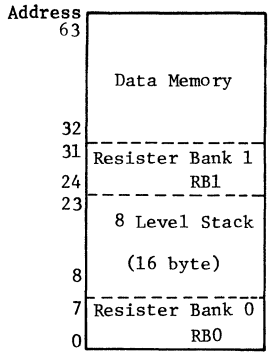
  A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

- Program addresses 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.

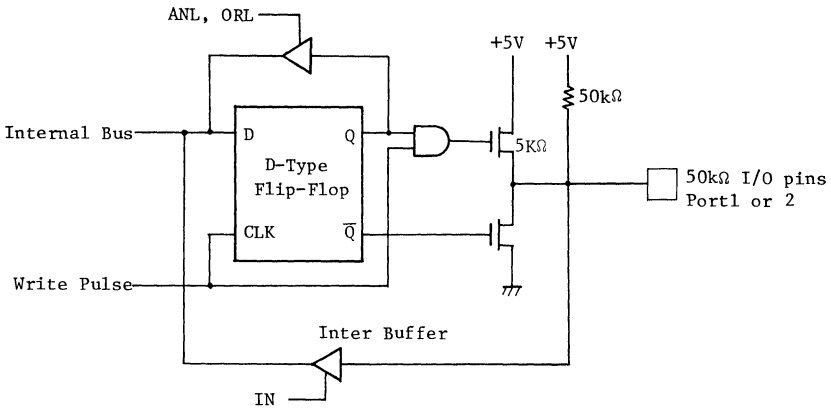


Fig.1 Input/Output Circuit of Port 1, Port 2

- Reset initializes all lines to a high impedance "1" state.
- When external data memory area is not addressed during 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 $\overline{WR}$ $\overline{RD}$ 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 × 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 executing a JTF or by RESET. Figure 2 illustrates the concept of the timer circuit.

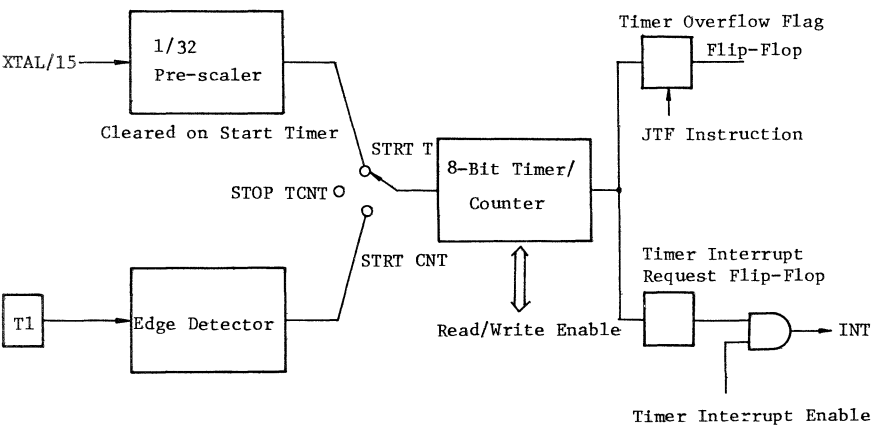


Fig.2 Concept of Timer Circuit

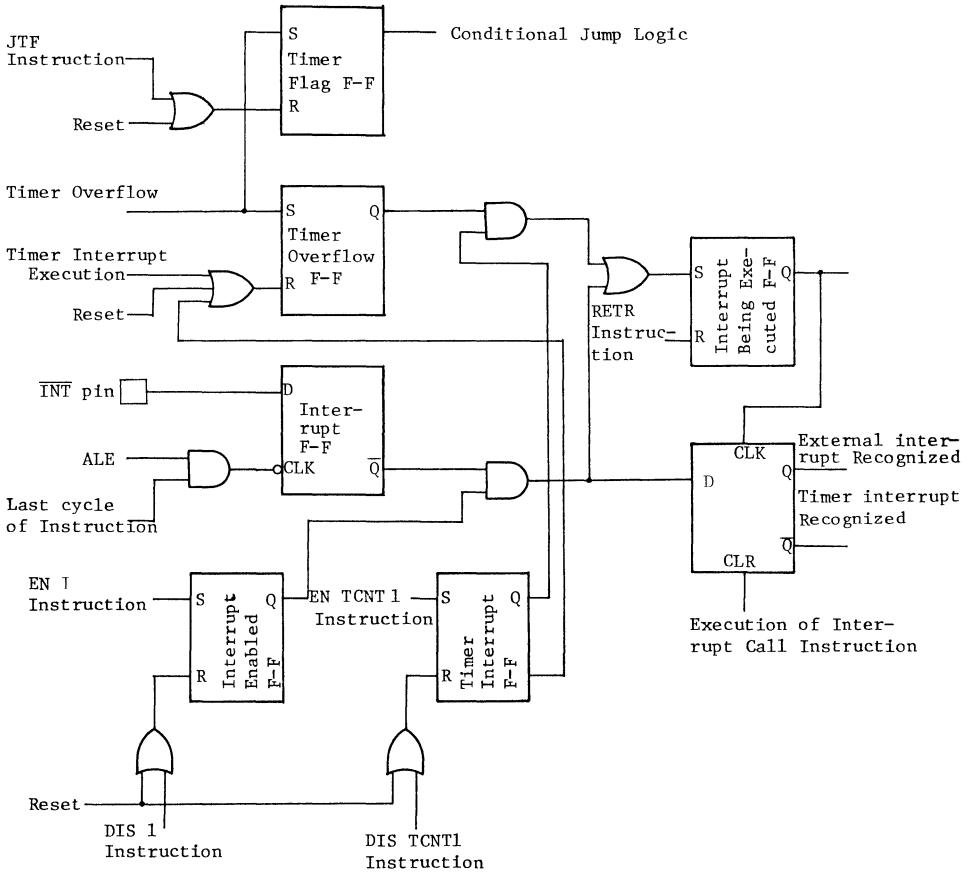


Fig.3 Concept of Interrupt Control Circuit

(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

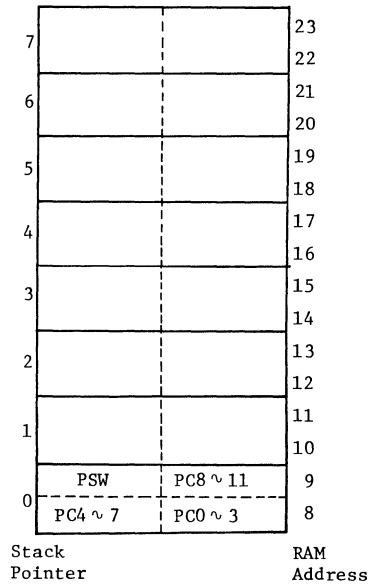
The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR (which should occur at the end of an interrupt service routine) reenables the interrupt input logic.

- 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 a 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 Word (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 locations 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.

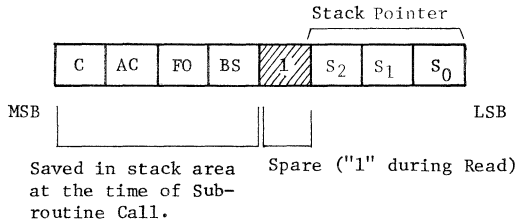


(7) Flag 0, Flag 1, (F0, F1)

- 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 accumulator exsists 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(S_0, S_1, S_2)

Bit 3 : Not used ("1" level when read.)

Bit 4 : Working Register Bank Switch Bit (BS)

0 = Bank 0

1 = Bank 1

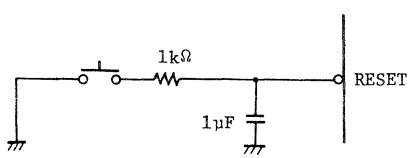
Bit 5 : Flag 0 (FO)

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

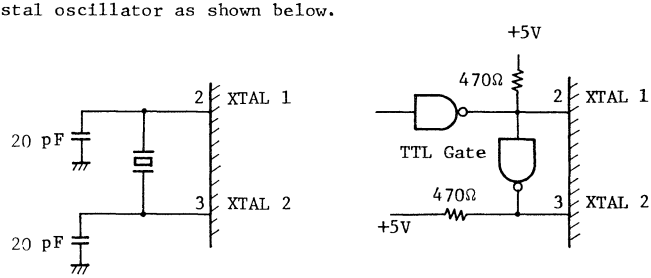
- The reset input provides a means for initialization of the processor. This Schmitt trigger input has an internal pullup resistor which in combination with an external 1μF capacitor provides an internal reset pulse sufficient length to guarantee that all internal logic is initialized.



- If the pulse is generated externally the reset pin must be held at ground (≤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 - 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

- 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 consists 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 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 (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.

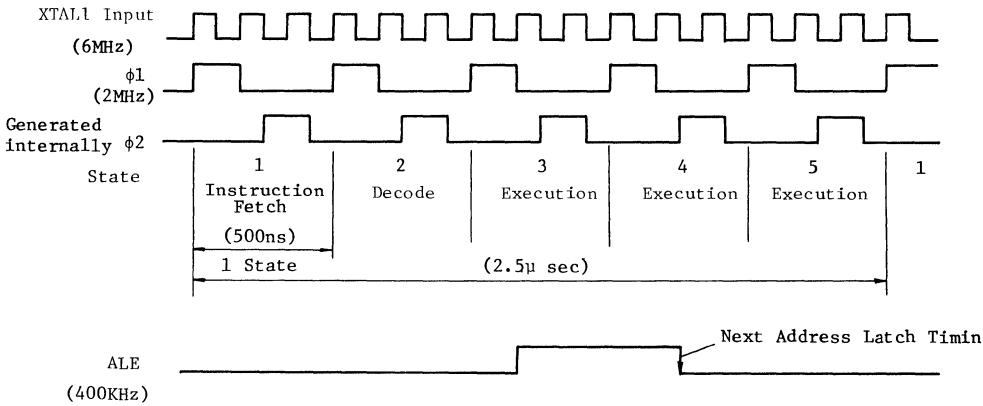


Fig.4 Instruction Cycle Timing

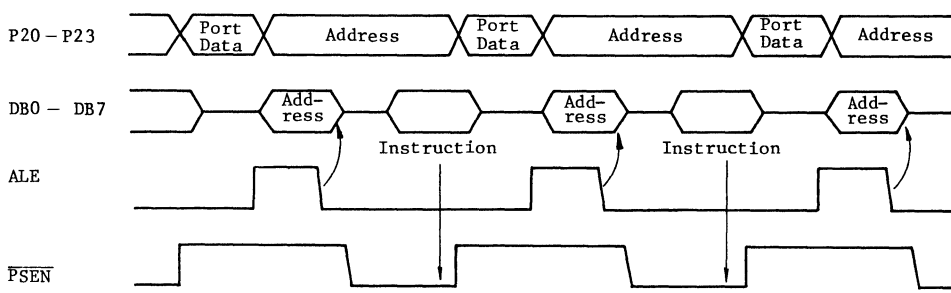
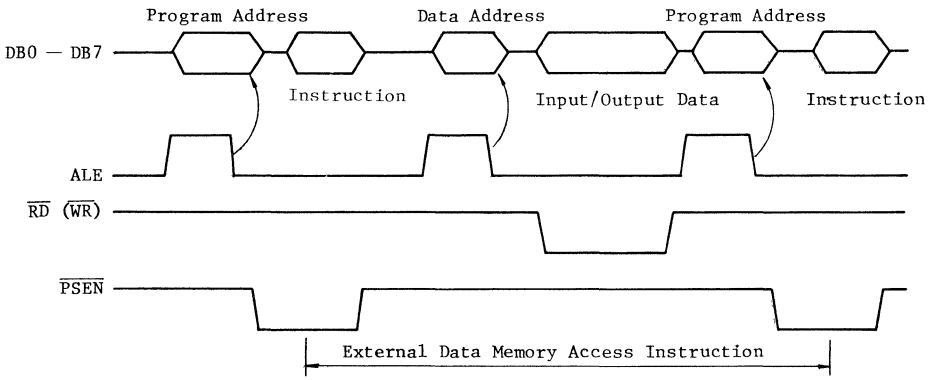


Fig.5 Timing of External Program Memory Access



Suggest we have two diagrams

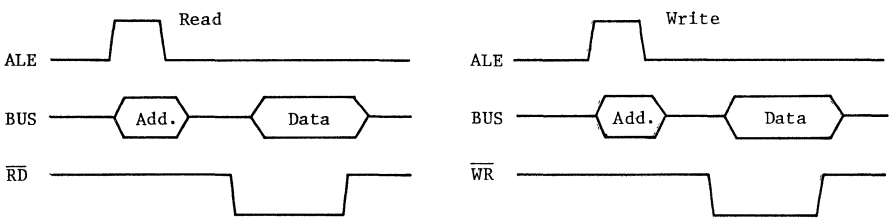


Fig.6 Timing of Accessing External Data Memory

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

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

The 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 communication 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.

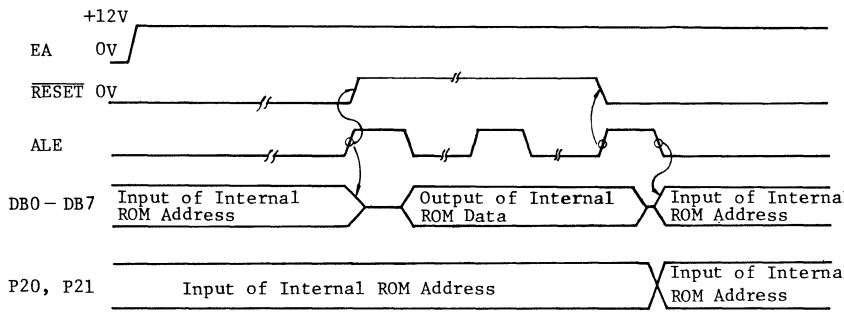


Fig.7 Timing of Rading Internal Program Memory

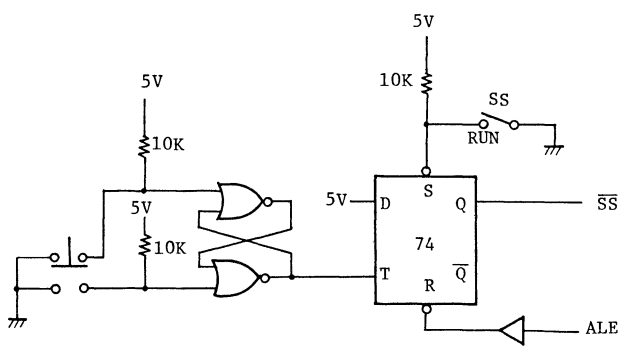


Fig.8(a) Single Step Circuit

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 $\overline{SS}$ 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 timeing deagram in this case is as shown in Figure 8 (b). (EA = 5V).

(6) Lower Power Stand-by Mode.

- The TMP8048 has been organized to allow power to be removed from all but the volatile, 64 × 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 V_{DD} 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.

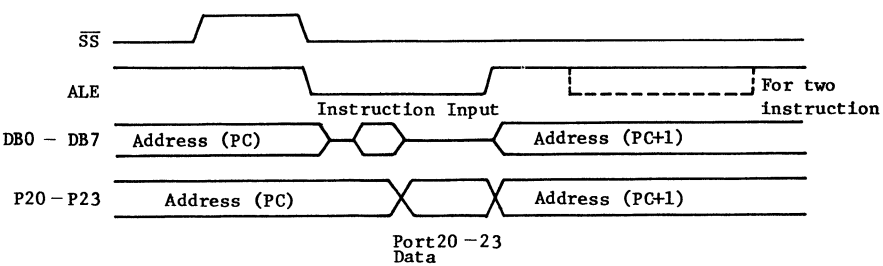


Fig.8(b) Single Step Operation Timing

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	○	○
ADD A,@Rr	0	1	1	0	0	0	0	r	(A)←(A)+(Rr) r = 0, 1	1	1	○	○
ADD A,#Data	0 d7	0 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data	2	2	○	○
ADDC A,Rr	0	1	1	1	1	r	r	r	(A)←(A)+(Rr)+(C) r = 0 – 7	1	1	○	○
ADDC A,@Rr	0	1	1	1	0	0	0	r	(A)←(A)+((Rr))+(C) r = 0, 1	1	1	○	○
ADDC A,#Data	0 d7	0 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data+(C)	2	2	○	○
ANL A,Rr	0	1	0	1	1	r	r	r	(A)←(A)∧(Rr) r = 0 – 7	1	1	–	–
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)←(A)∧((Rr)) r = 0, 1	1	1	–	–
ANL A,#Data	0 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∧Data	2	2	–	–
ORL A,Rr	0	1	0	0	1	r	r	r	(A)←(A)∨(Rr) r = 0 – 7	1	1	–	–
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)←(A)∨((Rr)) r = 0, 1	1	1	–	–
ORL A,#Data	0 d7	1 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∨Data	2	2	–	–
XRL A,Rr	1	1	0	1	1	r	r	r	(A)←(A)∀(Rr) r = 0 – 7	1	1	–	–
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)←(A)∀((Rr)) r = 0, 1	1	1	–	–
XRL A,#Data	1 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∀Data	2	2	–	–
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	○	–
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				A	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)∧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)∨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)∧Data	2	2	-	-
ORL BUS,#Data	1 d7	0 d6	0 d5	0 d4	1 d3	0 d2	0 d1	0 d0	(BUS)←(BUS)∨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	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
ANLD Pp,A	1	0	0	1	1	1	P	P	(Pp)←(Pp)∧(A0-3) P = 4 - 7	1	2	-	-
ORLD Pp,A	1	0	0	0	1	1	P	P	(Pp)←(Pp)∨(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 a7	a9 a6	a8 a5	1 a4	0 a3	1 a2	0 a1	0 a0	(PC0-7)←(a0-7) (PC8-10)←(a8-10) (PC11)←DBF	2	2	-	-
JMPP @A	1	0	1	1	0	0	1	1	(PC0-7)←((A))	1	2	-	-
DJNZ Rr, Address	1 a7	1 a6	1 a5	0 a4	1 a3	r a2	r a1	r a0	(Rr)←(Rr)-1 if Rr not 0 (PC0-7)←(a0-7)	2	2	-	-
JC Address	1 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if C = 1 (PC) = (PC)+2 if C = 0	2	2	-	-
JNC Address	1 a7	1 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if C = 0 (PC)←(PC)+2 if C = 1	2	2	-	-

Mnemonic	Instruction								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JZ Address	1 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if (A) = 0 (PC)←(PC)+2 if (A) ≠ 0	2	2	–	–
JNZ Address	1 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if (A) ≠ 0 (PC)←(PC)+2 if (A) = 0	2	2	–	–
JTO Address	0 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TO = 1 (PC)←(PC)+2 if TO = 0	2	2	–	–
JNTO Address	0 a7	0 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TO = 0 (PC)←(PC)+2 if TO = 1	2	2	–	–
JT1 Address	0 a7	1 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T1 = 1 (PC)←(PC)+2 if T1 = 0	2	2	–	–
JNT1 Address	0 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T1 = 0 (PC)←(PC)+2 if T1 = 1	2	2	–	–
JF0 Address	1 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if F0 = 1 (PC)←(PC)+2 if F0 = 0	2	2	–	–
JF1 Address	0 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if F1 = 1 (PC)←(PC)+2 if F1 = 0	2	2	–	–
JTF Address	0 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TF = 1 (PC)←(PC)+2 if TF = 0	2	2	–	–
JN1 Address	1 a7	0 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if INT = 0 (PC)←(PC)+2 if INT = 1	2	2	–	–
JBb Address	b2 a7	b1 a6	b0 a5	1 a4	0 a3	0 a2	1 a1	0 a0	(PC0-7)←(a0-7) if Bb = 1 (PC)←(PC)+2 if Bb = 0 (b = 0 – 7)	2	2	–	–

Subroutine Instruction

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
CALL Address	a10 a7	a9 a6	a8 a5	1 a4	0 a3	1 a2	0 a1	0 a0	((SP))← (PC),(PSW4-7) (SP)←(SP)+1 (PC8-10)←(a8-10) (PC0-7)←(a0-7) (PC11)←DBF	2	2	–	–
RET	1	0	0	0	0	0	1	1	(SP)←(SP)-1 (PC)←((SP))	1	2	–	–
RETR	1	0	0	1	0	0	1	1	(SP)←(SP)-1 (PC)←((SP)) (PSW4-7)←((SP))	1	2	–	–

Flag Manipulation Instruction

Mnemonics	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	○	–
CPL C	1	0	1	0	0	1	1	1	(C)←NOT(C)	1	1	○	–
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 Transter Instruction

Mnemonics	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	–	–

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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 d7	0 d6	1 d5	1 d4	0 d3	0 d2	0 d1	r d0	((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

Mnemonics	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	-	-

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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

Mnemonics	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	T_0 is enabled to act as the clock output	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-

TMP8048P/8035P

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	V_{DD} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to +7V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=70°C)	1.5W
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

DC CHARACTERISTICS Ta=0°C to 70°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		-0.5	-	0.8	V
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)		-0.5	-	0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		2.0	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8	-	V_{CC}	V
V_{OL}	Output Low Voltage (BUS)	$I_{OL}=2.0mA$	-	-	0.45	V
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OL}=1.8mA$	-	-	0.45	V
V_{OL2}	Output Low Voltage (PROG)	$I_{OL}=1.0mA$	-	-	0.45	V
V_{OL3}	Output Low Voltage (For other output pins)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OH}	Output High Voltage (BUS)	$I_{OH}=-400\mu A$	2.4	-	-	V
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OH}=-100\mu A$	2.4	-	-	V
V_{OH2}	Output High Voltage (For other output pins)	$I_{OH}=-40\mu A$	2.4	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$)	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current (P10-17, P20-P27, EA, $\overline{SS}$)	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, TO) (High impedance condition)	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	-	-	±10	μA
I_{DD}	V_{DD} Supply Current		-	-	15	mA
$I_{DD}+I_{CC}$	Total Supply Current		-	-	135	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	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		400	-	-	ns
t_{AL}	Address Setup Time (ALE)		120	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		700	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		500	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)		120	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)	C_L=20pF	0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	500	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	950	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		0	-	-	ns
t_{CA}	Internal between Control Pulse and ALE		10	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		110	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		100	-	-	ns
t_{PR}	Port 2 Input Data Set Time(PROG)		-	-	810	ns
t_{DP}	Output Data Setup Time (PROG)		250	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		65	-	-	ns
t_{PF}	Port2 Input Data Hold Time(PROG)		0	-	150	ns
t_{PP}	PROG Pulse Width		1200	-	-	ns
t_{PL}	Port2 I/O Data Setup Time		350	-	-	ns
t_{LP}	Port2 I/O Data Hold Time		150	-	-	ns

Note : t_{CY}=2.5μs, Control Output: C_L=80 pF, BUS Output: C_L=150pF, PORT20 - 23: C_L=80pF.

TMP8048PI/8035PI : INDUSTRIAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	V_{DD} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to +7V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=70°C)	1.5W
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

DC CHARACTERISTICS TA=-40°C to 85°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, Unless otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		-0.5	-	0.7	V
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)		-0.5	-	0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8	-	V_{CC}	V
V_{OL}	Output Low Voltage (BUS)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OL2}	Output Low Voltage (PROG)	$I_{OL}=0.8mA$	-	-	0.45	V
V_{OL3}	Output Low Voltage (For other output pins)	$I_{OL}=1.2mA$	-	-	0.45	V
V_{OH}	Output High Voltage (BUS)	$I_{OH}=-280\mu A$	2.4	-	-	V
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OH}=-80\mu A$	2.4	-	-	V
V_{OH2}	Output High Voltage (For other output pins)	$I_{OH}=-30\mu A$	2.4	-	-	V
I_{LI}	Input Leak Current(T1, $\overline{INT}$)	$V_{SS}\leq V_{IN}\leq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current (P10-17, P20-27, EA, $\overline{SS}$)	$V_{SS}+0.45\leq V_{IN}\leq V_{CC}$	-	-	-600	μA
I_{LO}	Output Leak Current(BUS,T0) (High impedance condition)	$V_{SS}+0.45\leq V_{IN}\leq V_{CC}$	-	-	±10	μA
I_{DD}	V_{DD} Supply Current		-	-	20	mA
$I_{DD}+I_{CC}$	Total Supply Current		-	-	145	mA

AC CHARACTERISTICS

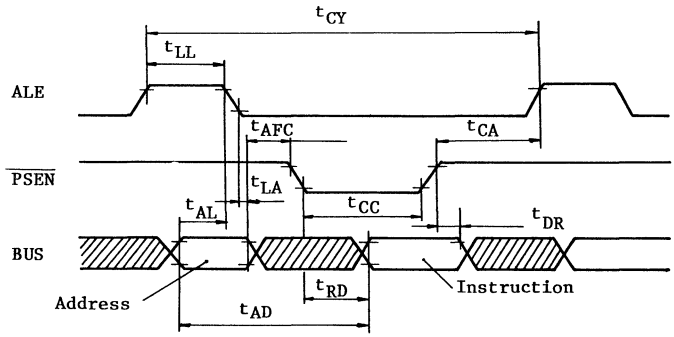
T_A=-40°C to 85°C, V_{CC}=V_{DD}=+5V±10%, V_{SS}=0V, Unless otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		200	-	-	ns
t_{AL}	Address Setup Time (ALE)		120	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		400	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		420	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	80	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	400	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	600	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-40	-	-	ns
t_{CA}	Internal between Control Pulse and ALE		10	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		115	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		65	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	860	ns
t_{DP}	Output Data Setup Time (PROG)		230	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		25	-	-	ns
t_{PF}	Port2 Input Data Hold Time (PROG)		0	-	160	ns
t_{PP}	PROG Pulse Width		920	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		300	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		120	-	-	ns

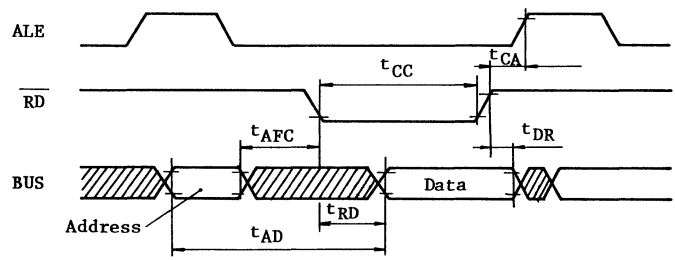
Note : tcy=2.5μs, Control Output: C_L=80pF, BUS Output: C_L=150pF, PORT 20 - 23: C_L=80pF.

TIMING WAVEFORM

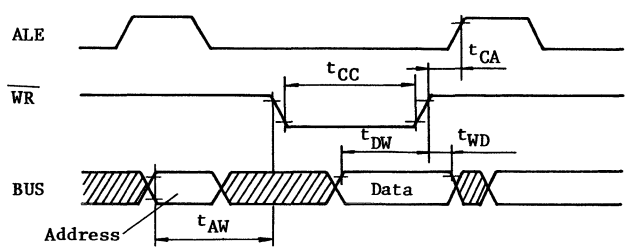
A. Instruction Fetch from External Program Memory



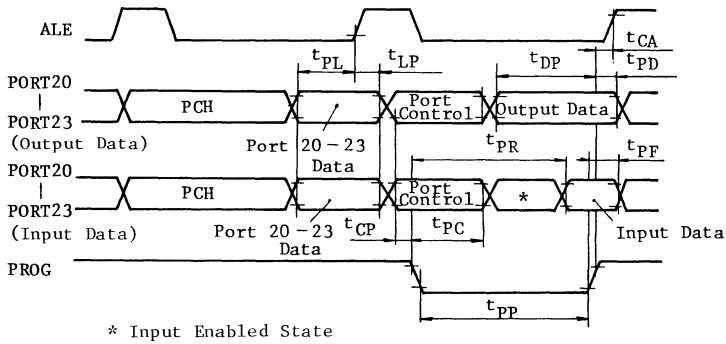
B. Read from External Data Memory



C. Write into External Data Memory



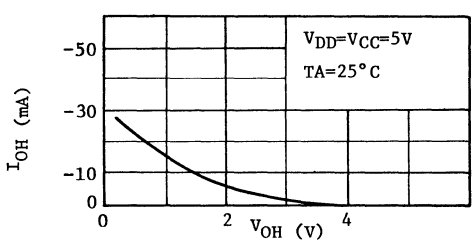
D. Timing of Port 2 during Expander Instruction Execution



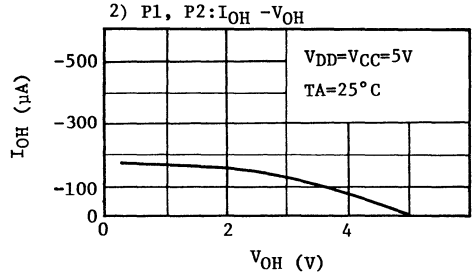
* Input Enabled State

TYPICAL CHARACTERISTICS

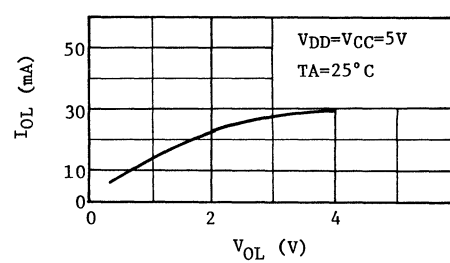
1) BUS: I_{OH} – V_{OH}



2) P1, P2: I_{OH} – V_{OH}



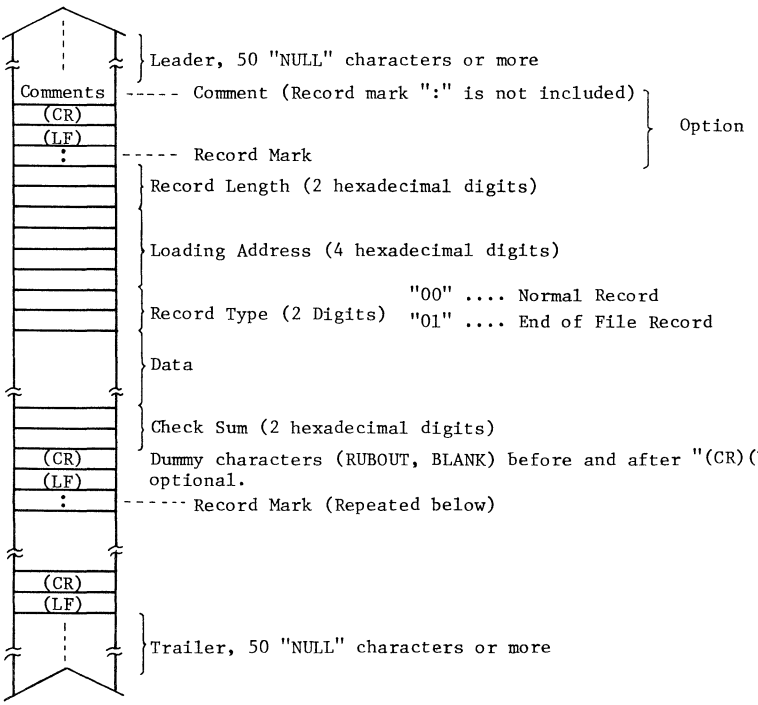
3) BUS, P1, P2: I_{OL} – V_{OL}



PROGRAM TAPE FORMAT

TMP8084 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

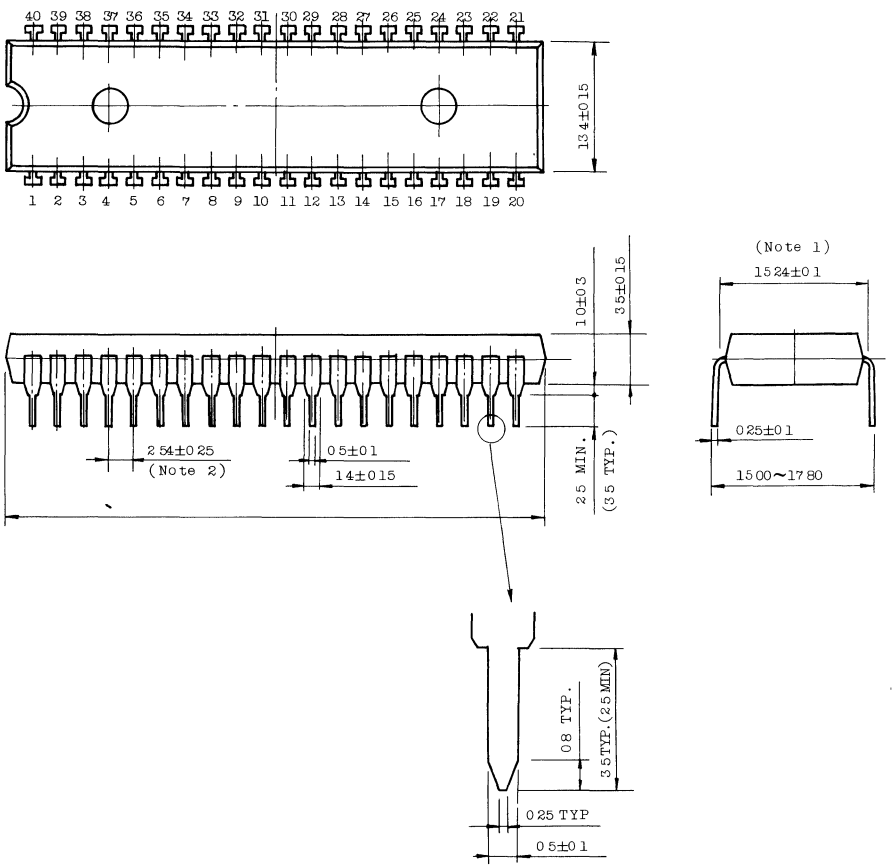


(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-84
: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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8049P/8049P-6/8049PI-6

TMP8039P/8039P-6/8039PI-6

N-CHANNEL SILICON GATE MOS

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP8049P, 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 × 8 RAM data memory, 2K × 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 handling capability as well as facilities for both binary and BCD arithmetic.

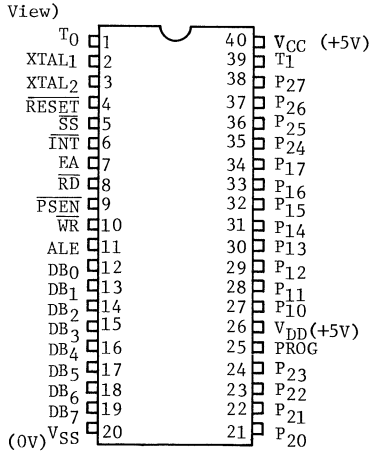
The TMP8039P is the equivalent of a TMP8049 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP8049P-6/TMP8039P-6 is a lower speed (6MHz) version of the TMP8049P/TMP8039P.

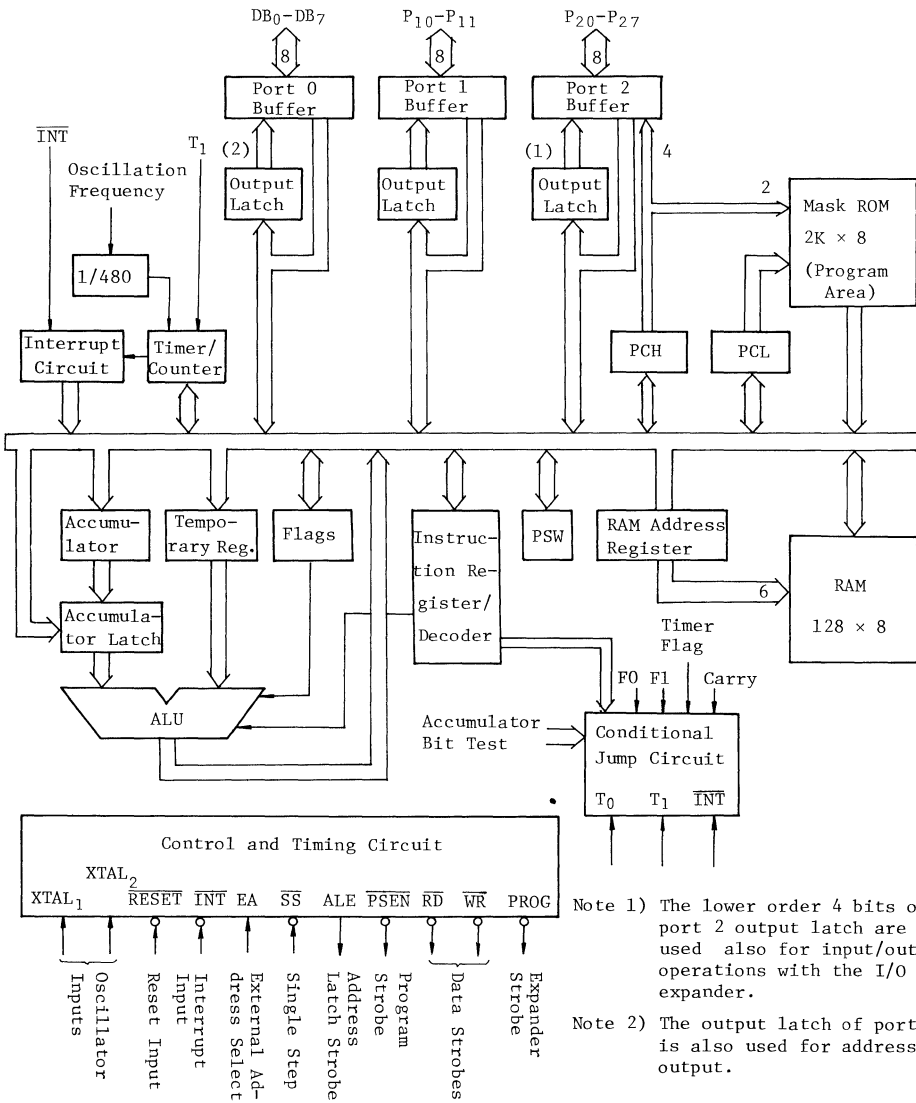
FEATURES

- Compatible with Intel's 8049
- 1.36μS Instruction Cycle
- All instruction 1 or 2 cycles
- Over 90 instructions; 70% single byte
- Easy expandable memory and I/O
- 2K × 8 masked ROM
- 128 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Single level interrupt
- Single 5V supply
- -40°C to +85°C Operation (TMP8049PI-6/ TMP8039PI-6 : Industrial Specification)

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

V_{SS} (Power Supply)

Circuit GND potential

V_{DD} (Power Supply)

+5V during operation Low power standby pin for TMP8049 RAM

V_{CC} (Main Power Supply)

+5V during operation

PROG (Output)

Output strobe for the TMP8243P I/O expander

P_{10}-P_{17} (Input/Output) Port 1

8-bit quasi-bidirectional port (Internal Pullup $\cong$ 50KΩ).

P_{20}-P_{27} (Input/Output) Port 2

8-bit quasi-bidirectional port (Internal Pullup $\cong$ 50KΩ).

P_{20}-P_{23} 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.

DB_0 -DB_7 (Input/Output, 3 State)

Ture 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 $\overline{WR}$.

T_0 (Input/Output)

Input pin testable using the conditional transfer instructions JTO and JNTO. T_0 can be designated as a clock output using ENTO CLK instruction.

T_1 (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 executed.

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

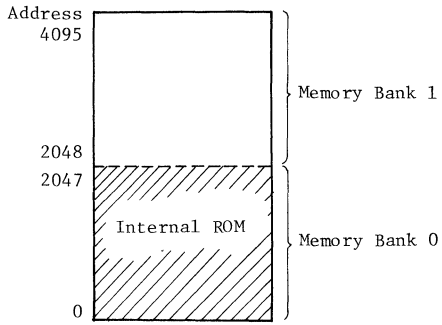
The following system functions of the TMP8049 are described in detail.

(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

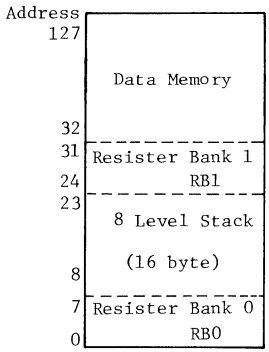
  A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

- Program addresses 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.

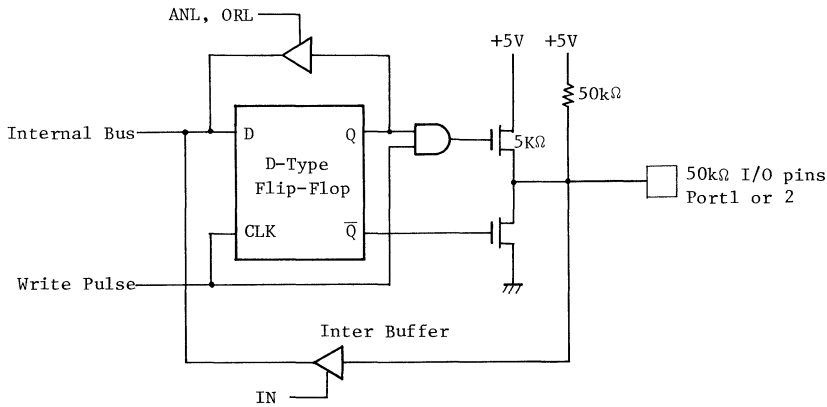


Fig.1 Input/Output Circuit of Port 1, Port 2

- Reset initializes all lines to a high impedance "1" state.
- When external data memory area is not addressed during 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 $\overline{WR}$ $\overline{RD}$ 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 × 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 executing a JTF or by RESET. Figure 2 illustrates the concept of the timer circuit.

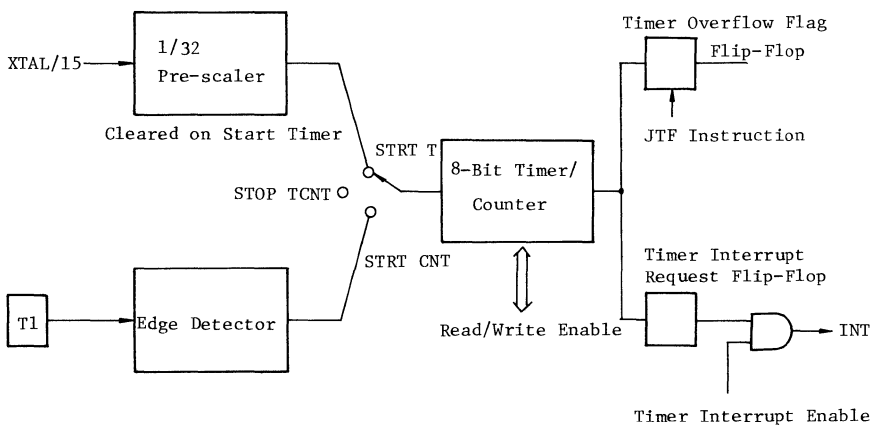


Fig.2 Concept of Timer Circuit

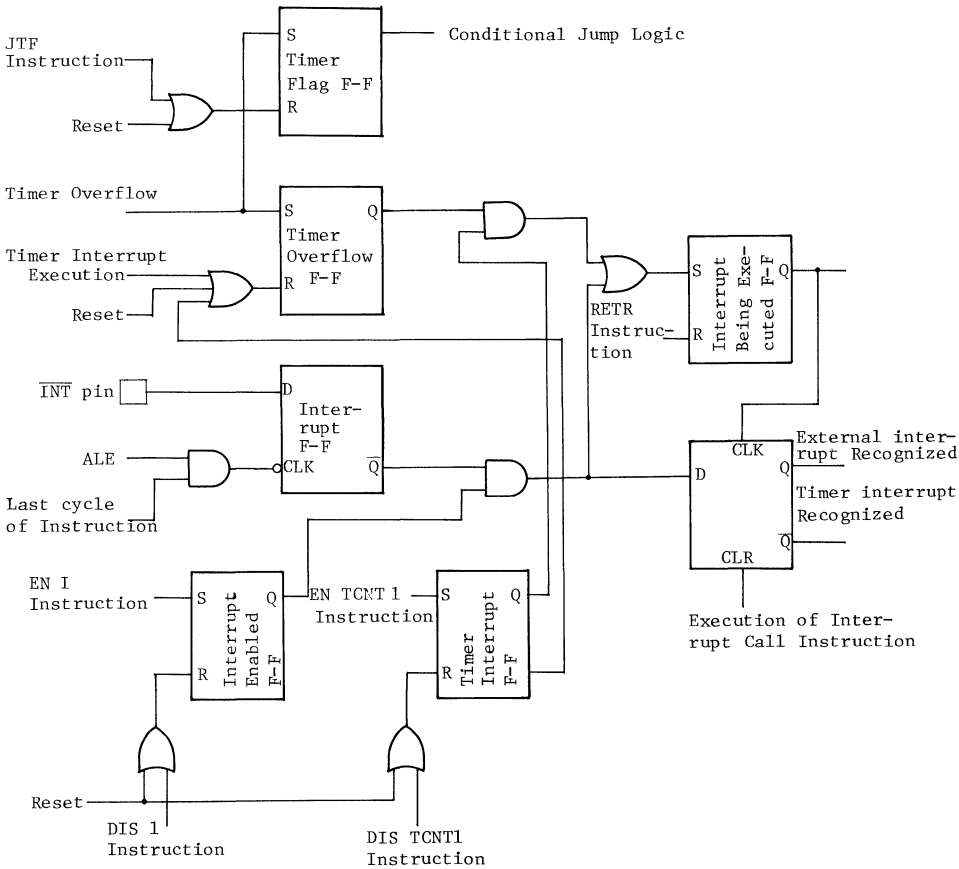


Fig. 3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

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

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

(2) Timer Interrupt caused by timer overflow

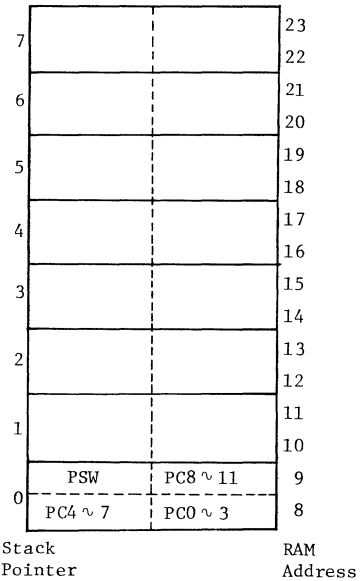
The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR (which should occur at the end of an interrupt service routine) reenables the interrupt input logic.

- 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 a 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 Word (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 locations 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.

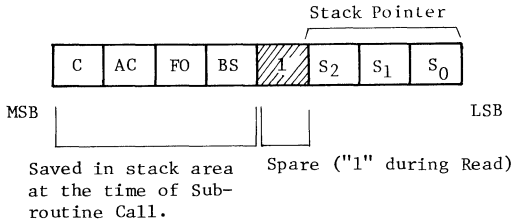


(7) Flag 0, Flag 1, (F0, F1)

- 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 accumulator exsists 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(S_0, S_1, S_2)

Bit 3 : Not used ("1" level when read.)

Bit 4 : Working Register Bank Switch Bit (BS)

0 = Bank 0

1 = Bank 1

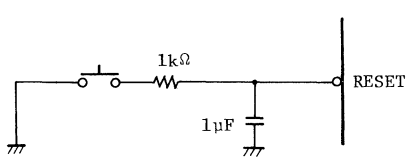
Bit 5 : Flag 0 (FO)

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 resistor 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 (≤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 - 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.

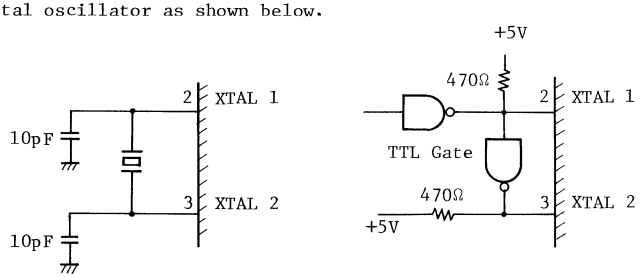
(ix) Clears Timer Flag.

(x) Clears F0 and F1.

(xi) Disables clock output 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 consists 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

- TMP8049 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 (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.

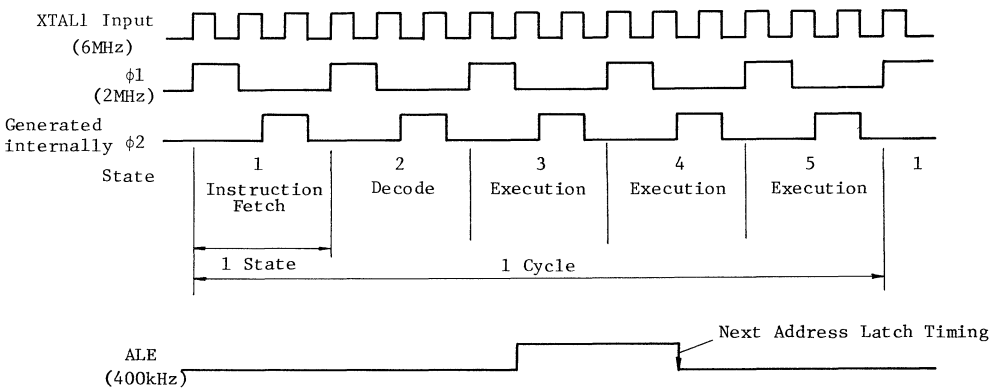


Fig.4 Instruction Cycle Timing

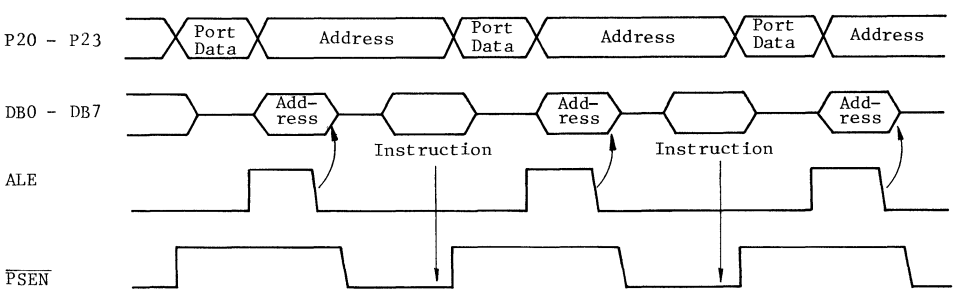
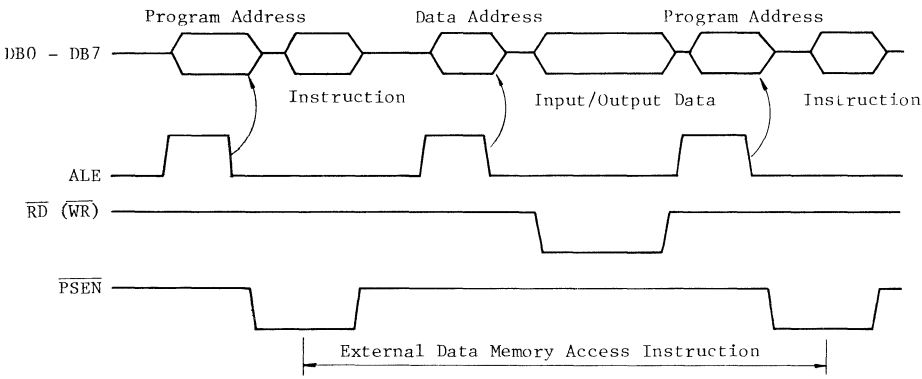


Fig.5 Timing of External Program Memory Access



Suggest we have two diagrams

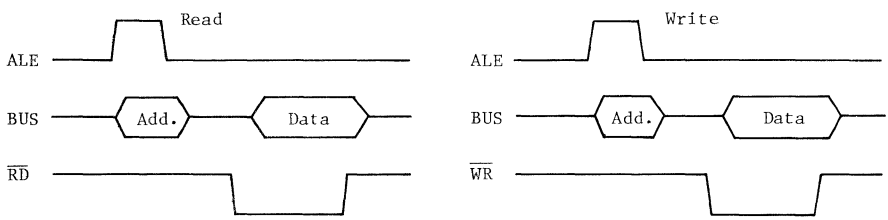


Fig.6 Timing of Accessing External Data Memory

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

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

- The 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 communication 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.

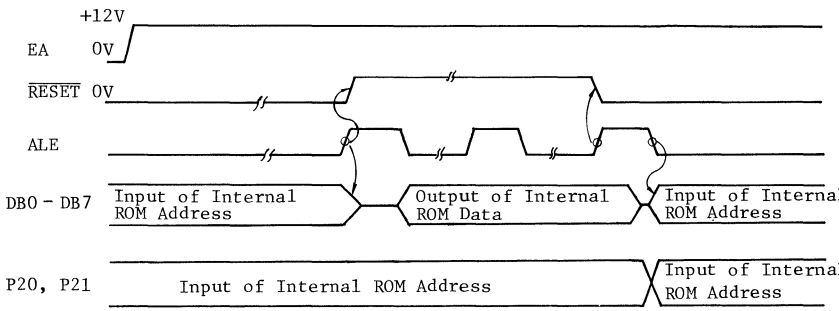


Fig.7 Timing of Rading Internal Program Memory

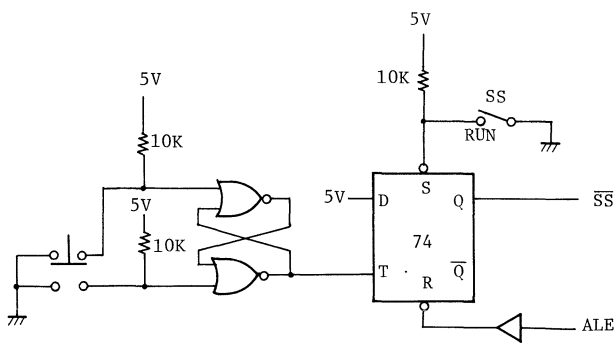


Fig.8(a) Single Step Circuit

(4) 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 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 timeing deagram in this case is as shown in Figure 8 (b). (EA = 5V).

(6) Lower Power Stand-by Mode.

- The TMP8049 has been organized to allow power to be removed from all but the volatile, 128 × 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 V_{DD} supplies only the RAM array. In standby mode VCC is reduced to 0V but V_{DD} 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.

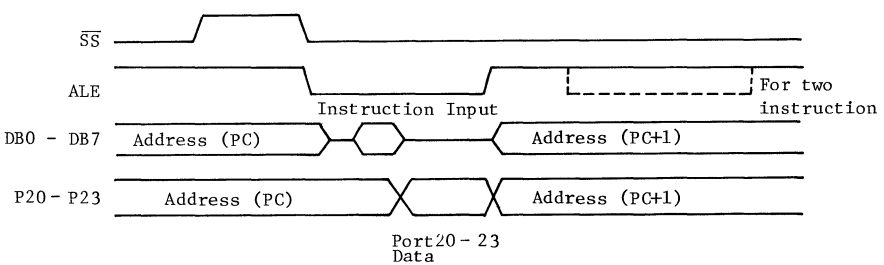


Fig.8(b) Single Step Operation Timing

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	○	○
ADD A,@Rr	0	1	1	0	0	0	0	r	(A)←(A)+(Rr) r = 0, 1	1	1	○	○
ADD A,#Data	0 d7	0 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data	2	2	○	○
ADDC A,Rr	0	1	1	1	1	r	r	r	(A)←(A)+(Rr)+(C) r = 0 - 7	1	1	○	○
ADDC A,@Rr	0	1	1	1	0	0	0	r	(A)←(A)+((Rr))+(C) r = 0, 1	1	1	○	○
ADDC A,#Data	0 d7	0 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data+(C)	2	2	○	○
ANL A,Rr	0	1	0	1	1	r	r	r	(A)←(A)∧(Rr) r = 0 - 7	1	1	-	-
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)←(A)∧((Rr)) r = 0, 1	1	1	-	-
ANL A,#Data	0 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∧Data	2	2	-	-
ORL A,Rr	0	1	0	0	1	r	r	r	(A)←(A)∨(Rr) r = 0 - 7	1	1	-	-
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)←(A)∨((Rr)) r = 0, 1	1	1	-	-
ORL A,#Data	0 d7	1 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∨Data	2	2	-	-
XRL A,Rr	1	1	0	1	1	r	r	r	(A)←(A)∀(Rr) r = 0 - 7	1	1	-	-
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)←(A)∀((Rr)) r = 0, 1	1	1	-	-
XRL A,#Data	1 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∀Data	2	2	-	-
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	○	-
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				A	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)∧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)∨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)∧Data	2	2	-	-
ORL BUS,#Data	1 d7	0 d6	0 d5	0 d4	1 d3	0 d2	0 d1	0 d0	(BUS)←(BUS)∨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	-	-

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
ANLD Pp,A	1	0	0	1	1	1	P	P	(Pp)←(Pp)∧(A0-3) P = 4 - 7	1	2	-	-
ORLD Pp,A	1	0	0	0	1	1	P	P	(Pp)←(Pp)∨(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 a7	a9 a6	a8 a5	1 a4	0 a3	1 a2	0 a1	0 a0	(PC0-7)←(a0-7) (PC8-10)←(a8-10) (PC11)←DBF	2	2	-	-
JMPP @A	1	0	1	1	0	0	1	1	(PC0-7)←((A))	1	2	-	-
DJNZ Rr, Address	1 a7	1 a6	1 a5	0 a4	1 a3	r a2	r a1	r a0	(Rr)←(Rr)-1 if Rr not 0 (PC0-7)←(a0-7)	2	2	-	-
JC Address	1 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if C = 1 (PC) = (PC)+2 if C = 0	2	2	-	-
JNC Address	1 a7	1 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if C = 0 (PC)←(PC)+2 if C = 1	2	2	-	-

Mnemonic	Instruction								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
JZ Address	1 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if (A) = 0 (PC)←(PC)+2 if (A) ≠ 0	2	2	-	-
JNZ Address	1 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if (A) ≠ 0 (PC)←(PC)+2 if (A) = 0	2	2	-	-
JTO Address	0 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TO = 1 (PC)←(PC)+2 if TO = 0	2	2	-	-
JNTO Address	0 a7	0 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TO = 0 (PC)←(PC)+2 if TO = 1	2	2	-	-
JT1 Address	0 a7	1 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T1 = 1 (PC)←(PC)+2 if T1 = 0	2	2	-	-
JNT1 Address	0 a7	1 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T1 = 0 (PC)←(PC)+2 if T1 = 1	2	2	-	-
JFO Address	1 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if FO = 1 (PC)←(PC)+2 if FO = 0	2	2	-	-
JF1 Address	0 a7	1 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if F1 = 1 (PC)←(PC)+2 if F1 = 0	2	2	-	-
JTF Address	0 a7	0 a6	0 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if TF = 1 (PC)←(PC)+2 if TF = 0	2	2	-	-
JN1 Address	1 a7	0 a6	0 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if INT = 0 (PC)←(PC)+2 if INT = 1	2	2	-	-
JBb Address	b2 a7	b1 a6	b0 a5	1 a4	0 a3	0 a2	1 a1	0 a0	(PC0-7)←(a0-7) if Bb = 1 (PC)←(PC)+2 if Bb = 0 (b = 0 - 7)	2	2	-	-

Subroutine Instruction

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
CALL Address	a10 a7	a9 a6	a8 a5	1 a4	0 a3	1 a2	0 a1	0 a0	((SP))← (PC),(PSW4-7) (SP)←(SP)+1 (PC8-10)←(a8-10) (PC0-7)←(a0-7) (PC11)←DBF	2	2	-	-
RET	1	0	0	0	0	0	1	1	(SP)←(SP)-1 (PC)←((SP))	1	2	-	-
RETR	1	0	0	1	0	0	1	1	(SP)←(SP)-1 (PC)←((SP)) (PSW4-7)←((SP))	1	2	-	-

Flag Manipulation Instruction

Mnemonics	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	○	-
CPL C	1	0	1	0	0	1	1	1	(C)←NOT(C)	1	1	○	-
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 Transter Instruction

Mnemonics	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	-	-

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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 d7	0 d6	1 d5	1 d4	0 d3	0 d2	0 d1	r d0	((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

Mnemonics	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	–	–

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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

Mnemonics	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	T_0 is enabled to act as the clock output	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-

TMP8049P/8039P/8049P-6/8039P-6

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	V_{DD} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to +7V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=25°C)	1.5W
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

DC CHARACTERISTCS

TA=0°C to 70°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, 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, $\overline{RESET}$)		2.0	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8	-	V_{CC}	V
V_{OL}	Output Low Voltage (BUS)	I_{OL}=2.0mA	-	-	0.45	V
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OL}=1.8mA	-	-	0.45	V
V_{OL2}	Output Low Voltage (PROG)	I_{OL}=1.0mA	-	-	0.45	V
V_{OL3}	Output Low Voltage (For other output pins)	I_{OL}=1.6mA	-	-	0.45	V
V_{OH}	Output High Voltage (BUS)	I_{OH}=-400μA	2.4	-	-	V
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OH}=-100μA	2.4	-	-	V
V_{OH2}	Output High Voltage (For tothe output pins)	I_{OH}=-40μA	2.4	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$)	$V_{SS}\leq V_{IN}\leq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current (P10-17, P20-P27, EA, $\overline{SS}$)	$V_{SS}+0.45\leq V_{IN}\leq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, TO) (High impedance condition)	$V_{SS}+0.45\leq V_{IN}\leq V_{CC}$	-	-	±10	μA
I_{DD}	V_{DD} Supply Current		-	-	50	mA
$I_{DD}+I_{CC}$	Total Supply Current		-	-	170	mA

AC CHARACTERISTICS

TA=0°C to 70°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITION	TMP8049P/ TMP8039P		TMP8049P-6/ TMP8039P-6		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{LL}	ALE Pulse Width		150	-	400	-	ns
t_{AL}	Address Setup Time (ALE)		70	-	150	-	ns
t_{LA}	Address Hold Time (ALE)		50	-	80	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		300	-	700	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		250	-	500	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	40	-	120	-	ns
t_{CY}	Cycle Time	11MHz XTAL (6MHz XTAL for -6)	1.36	15.0	2.5	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	100	0	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	200	-	500	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		200	-	230	-	ns
t_{AD}	Address Setup Time (Data Input)		-	400	-	950	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-10	-	0	-	ns
t_{CP}	Port Control Setup Time (PROG)		100	-	110	-	ns
t_{PC}	Port Control Hold Time (PROG)		60	-	130	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	650	-	810	ns
t_{DP}	Output Data Setup Time (PROG)		200	-	220	-	ns
t_{PD}	Output Data Hold Time (PROG)		20	-	65	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	150	0	150	ns
t_{PP}	PROG Pulse Width		700	-	1510	-	ns
t_{PL}	Port 2 I/O Data Setup Time		250	-	500	-	ns
t_{LP}	Port 2 I/O Data Hold Time		120	-	150	-	ns

Control Outputs : C_L=80pF, BUS Outputs : C_L=150pF

TMP8049PI-6/8039PI-6 : INDUSTRIAL SPECIFICATION.

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	V_{DD} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{CC}	V_{CC} Supply Voltage (with respect ro GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to +7V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=25°C)	1.5W
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

DC CHARACTERISTICS TA=-40°C to 85°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, Unless Otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage		-0.5	-	0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8	-	V_{CC}	V
V_{OL}	Output Low Voltage (BUS)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OL2}	Output Low Voltage (PROG)	$I_{OL}=0.8mA$	-	-	0.45	V
V_{OL3}	Output Low Voltage (For other output pins)	$I_{OL}=1.2mA$	-	-	0.45	V
V_{OH}	Output High Voltage (BUS)	$I_{OH}=-80\mu A$	2.4	-	-	V
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	$I_{OH}=-80\mu A$	2.4	-	-	V
V_{OH2}	Output High Voltage (For other output pins)	$I_{OH}=-30\mu A$	2.4	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$)	$V_{SS}\leqq V_{IN}\leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current (P10-17, P20-P27, EA, $\overline{SS}$)	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	-	-	-700	μA
I_{LO}	Output Leak Current (BUS, TO) (High impedance condition)	$V_{SS}+0.45\leqq V_{IN}\leqq V_{CC}$	-	-	±10	μA
I_{DD}	V_{DD} Supply Current		-	-	50	mA
$I_{DD}+I_{CC}$	Total Supply Current		-	-	170	mA

AC CHARACTERISTICS

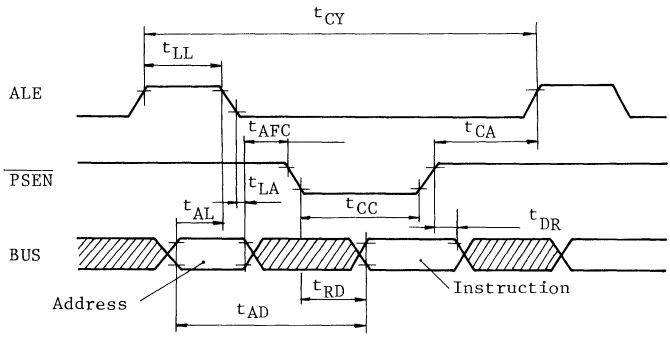
T_A=-40°C to 85°C, V_{CC}=V_{DD}=+5V±10%, V_{SS}=0V, Unless otherwise Noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		200	-	-	ns
t_{AL}	Address Setup Time (ALE)		120	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		400	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		420	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	80	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	400	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	600	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-40	-	-	ns
t_{CA}	Internal between Control Pulse and ALE		10	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		115	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		65	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	860	ns
t_{DP}	Output Data Setup Time (PROG)		230	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		25	-	-	ns
t_{PF}	Port2 Input Data Hold Time (PROG)		0	-	160	ns
t_{PP}	PROG Pulse Width		920	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		300	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		120	-	-	ns

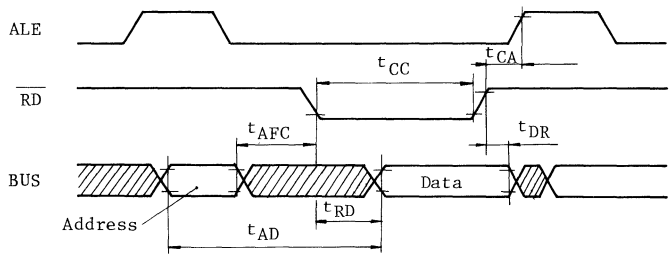
Note : tcy=2.5μs, Control Output: C_L=80pF, BUS Output: C_L=150pF, PORT 20-23: C_L=80pF.

TIMING WAVEFORM

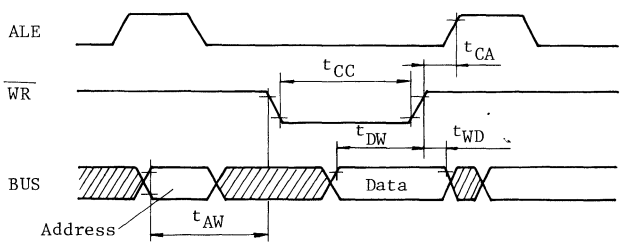
A. Instruction Fetch from External Program Memory



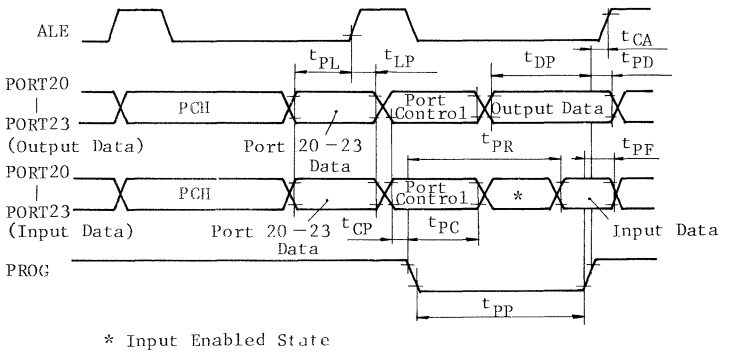
B. Read from External Data Memory



C. Write into External Data Memory



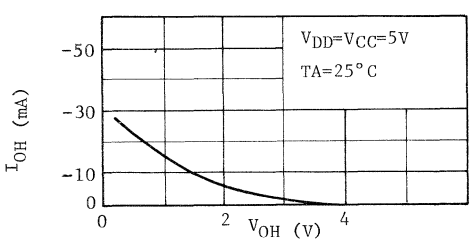
D. Timing of Port 2 during Expander Instruction Execution



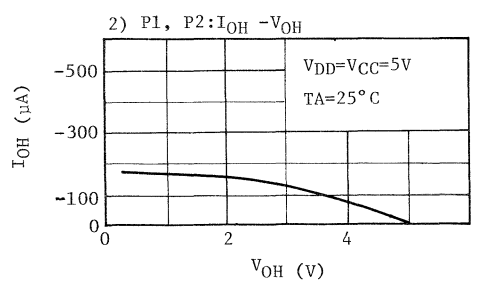
* Input Enabled State

TYPICAL CHARACTERISTICS

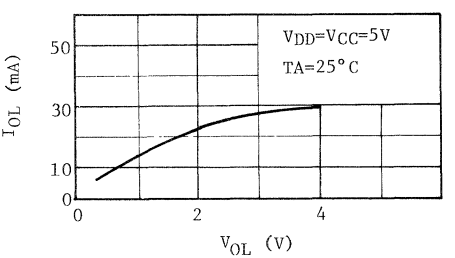
1) BUS: I_{OH} - V_{OH}



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



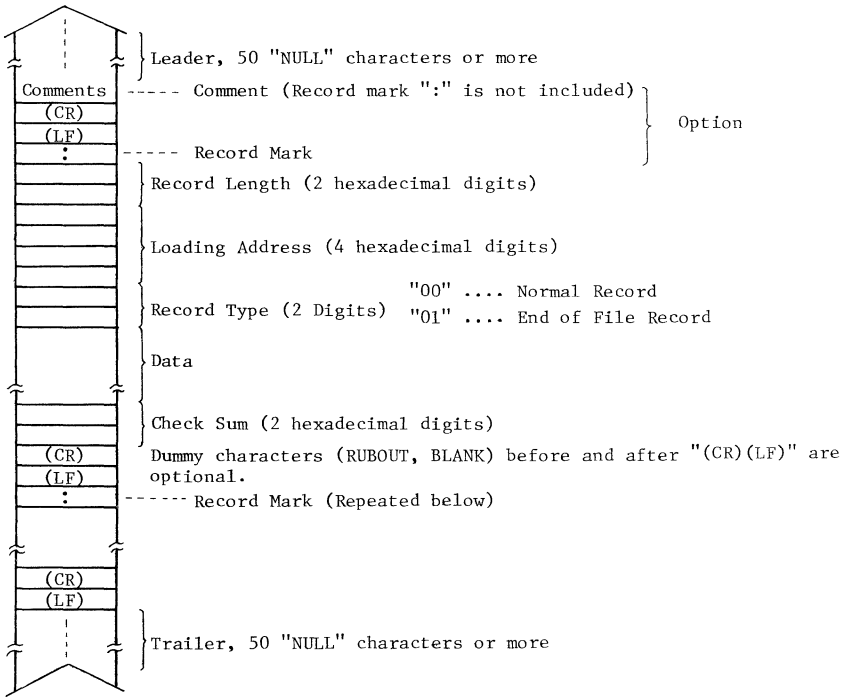
3) BUS, P1, P2: I_{OL} - V_{OL}



PROGRAM TAPE FORMAT

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

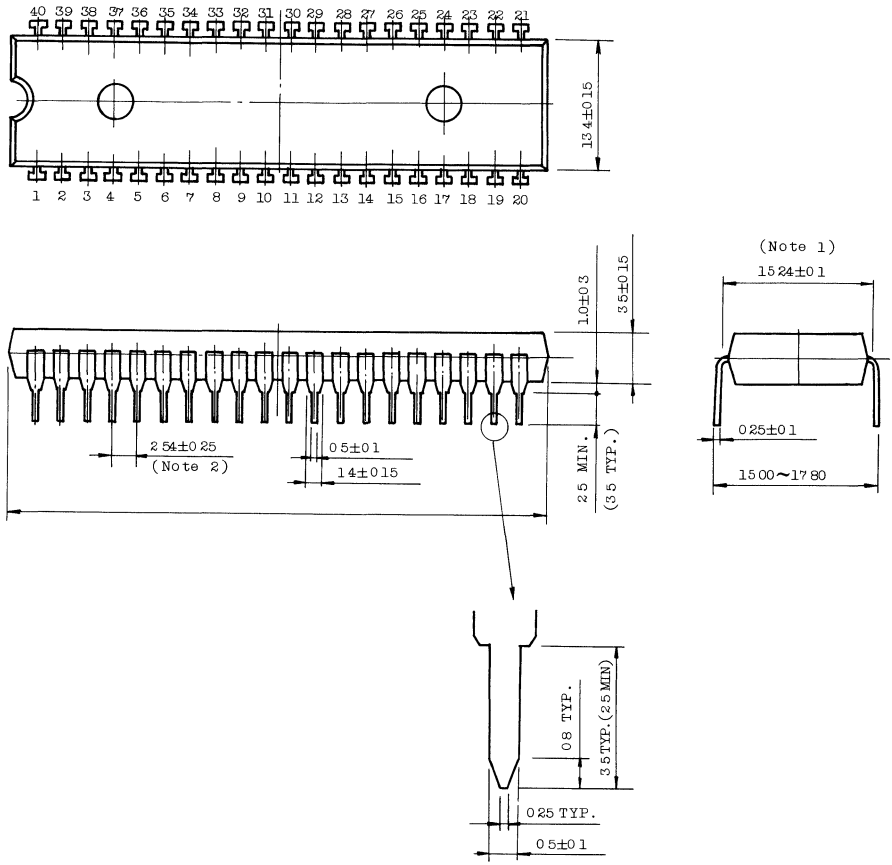


(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-84
: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.

GENERAL DESCRIPTION

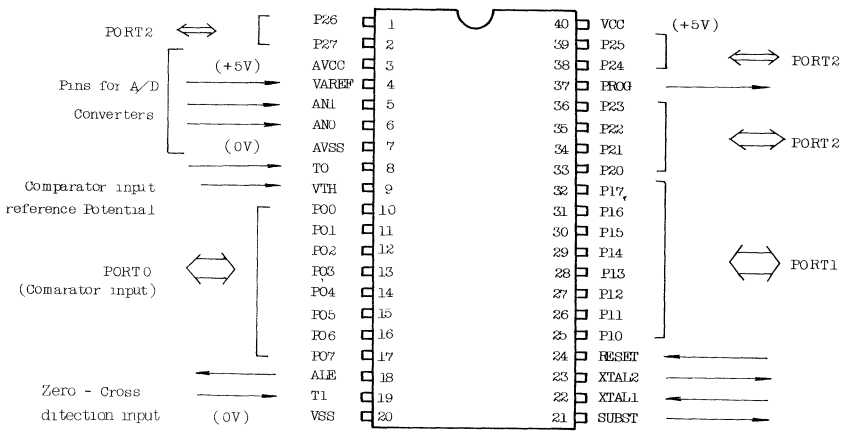
The TMP8022P is one version of the TLCS-84 family, which is an 8-bit single chip microcomputer containing A/D converter.

The CPU, data memory (RAM), program memory (ROM), I/O port, and timer, which are basic functions as a computer, and further, A/D converter, comparator input port, zero-cross ditection circuit, etc. are all integrated on single chip.

FEATURES

- Compatible with Intel's 8022
- 2K × 8 ROM, 64 × 8 RAM, 28 I/O Lines
- 8 Bit Interval Timer/Event Counter
- On-chip 8 Bit A/D Converter; Two Input Channels
- 8 Comparator Inputs (PORT0)
- Zero Cross Detection Capability
- High Current Drive Capability (V_{OL}<2.5V @I_{OL}=7mA)
- 8 Level Subroutine Nesting
- Two Interrupts - External and Timer
- Instructions - 8048 Subset
- 8.38 sec Cycles; All Instructions 1 or 2 Cycles
- Single 5V Supply (4.5V to 6.5V)

PIN CONNECTIONS (TOP VIEW)



PIN NAMES AND DESCRIPTION

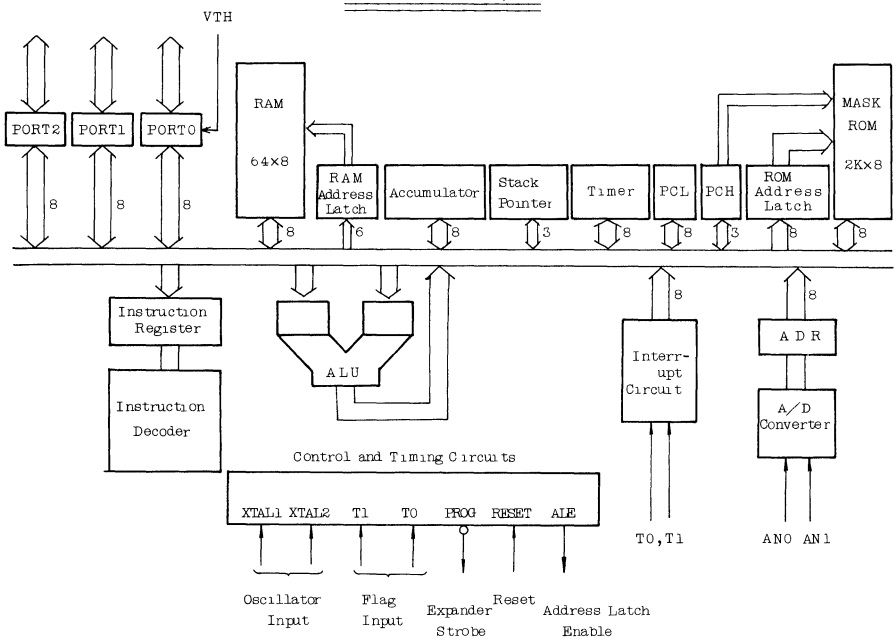
Pin Name	Pin No.	Input/ Output	Function
XTAL1 XTAL2	22 23	Input	A terminal for connecting the oscillator or an input terminal for the external clock.
RESET	24	Input	High active signal and initializes the chip. When a low level voltage is applied to this pin, a program starts from address 0.

Pin Name	Pin No.	Input/ Output	Function
T0	8	Input	Ecternal interrupt input. Since this pin is of a level interrupt type, it is required to be held at low level until an interrupt is accepted. Further, this pin serves as a test flag input for the conditional jump instructions (JT1 and JNT1).
T1	19	Input	This pin is an external clock input for the timer counter at the time of the event counter mode, and has a built-in zero-cross ditection circuit. Further, it serves as a test flag input for the conditional jump instructions (JT1 and JNT1).
ALE	18	Output	Address Latch Enable pin. This pin is a clock output that regards 1 machine cycle (clock cycle x 30) as a cycle. (It is also used for the address latch in the test mode 2.)
PROG	37	Output	Output strobe for the I/O expander 8243.
P00 ∿ P07 (PORT0)	10∿17	I/O	8-bit open drain port. Since this pin has a built-in comparator which regards the voltage applied to VTH pin as a comparison voltage, it can change the input inverse level. It contains a mask option with a pull-up resistor.

Pin Name	Pin No.	Input/ Output	Function
P10 ∿ P27 (PORT1)	25∿32	I/O	8-bit quasi-bidirectional port.
P20 ∿ P27 (PORT2)	33∿36 38,39 1,2	I/O	8-bit-bidirectional port. The lower order 4-bit pins P20 to P23 serve as lines connecting the 4-bit I/O expander 8243.
VTH	9	Input	PORT0 threshold reference pin.
AN0, AN1	6,5	Input	Analog input to A/D converter. This pin switches the channels by use of soft according to SEL AN0 and SEL AN1 instructions. (2 channels)
V_{AREF}	4	Input	The reference voltage of A/D Converter Establishes the upper limit of A/D conversion range.
AVCC	3	Power supply	+5V (For A/D converter section)
AVSS	7	"	+5V (For A/D converter section)
VCC	40	"	+5V
VSS	20	"	+0V
SUBST	21	Output	Substrate potential output pin. This pin is used for the purpose of connecting a bypass capacitor across the VSS pin for improving the accuracy of the A/D converter by stabilizing the substrate potential.

BLOCK DIAGRAM

8022 Block Diagram



DESCRIPTION OF INSTRUCTIONS

o The table of instructions for the TMP8022P is described by use of the following symbols and abbreviations.

A	:	Accumulator
AC	:	Auxiliary carry
addr	:	Lower order 8-bit address
C	:	Carry
CRR	:	A/D conversion result register
data	:	8-bit data
Pp	:	Port p=0, 1, 2 or P=4 ∿ 7
PC	:	Program counter
Rr	:	Register r=0, 1 or r=0 ∿ 7
SP	:	Stack pointer
T	:	Timer
TF	:	Timer flag
T0	:	Test 0
T1	:	Test 1
(x)	:	Contents of x
((x))	:	Contents of address indicated by x
∧	:	AND
∨	:	Logical OR
⊻	:	Exclusive OR

TMP8022 Instruction List (I)

Classification	Mnemonics	Functional description	Effected Flag C	Effected Flag AC	Bytes	Cycles	Operation Code (Hexadecimal)
Accumulator Instruction	ADD A, Rr	(A)←(A)+(Rr) r=0-7	○	○	1	1	68-6F
	ADD A, @Rr	(A)←(A)+((Rr)) r=0,1	○	○	1	1	60, 61
	ADD A, #data	(A)←(A)+data	○	○	2	2	03
	ADDC A, Rr	(A)←(A)+(Rr)+(C) r=0-7	○	○	1	1	78-7F
	ADDC A, @Rr	(A)←(A)+((Rr))+(C) r=0, 1	○	○	1	1	70-71
	ADDC A, #data	(A)←(A)+data+(C)	○	○	2	2	13
	ANL A, Rr	(A)←(A)∧(Rr) r=0-7	-	-	1	1	58-5F
	ANL A, @Rr	(A)←(A)∧((Rr)) r=0, 1	-	-	1	1	50-51
	ANL A, #data	(A)←(A)∧data	-	-	2	2	53
	ORL A, Rr	(A)←(A)∨(Rr) r=0-7	-	-	1	1	48-4F
	ORL A, @Rr	(A)←(A)∨((Rr)) r=0, 1	-	-	1	1	40-41
	ORL A, #data	(A)←(A)∨data	-	-	2	2	43
	XRL A, Rr	(A)←(A)⊻(Rr) r=0-7	-	-	1	1	D8-DF
	XRL A, @Rr	(A)←(A)⊻((Rr)) r=0, 1	-	-	1	1	D0-D1
	XRL A, #data	(A)←(A)⊻data	-	-	2	2	D3
	INC A	(A)←(A)+1	-	-	1	1	17
	DEC A	(A)←(A)-1	-	-	1	1	07
	CLR A	(A)←0	-	-	1	1	27
	CPL A	(A)←NOT(A)	-	-	1	1	37
	DA A	Decimal adjust A	○	-	1	1	57

Classification	Mnemonics	Functional Description	Effected Flag C	Effected Flag AC	Bytes	Cycles	Operation Code (Hexadecimal)
Accumulator Instruction	SWAP A	$(A_{4-7}) \rightleftarrows (A_{0-3})$	-	-	1	1	47
	RL A	$(A_{n+1}) \leftarrow (A_n), (A_0) \leftarrow (A_7)$ n=0-6	-	-	1	1	E7
	RLC A	$(A_{n+1}) \leftarrow (A_n), (C) \leftarrow (A_7)$ n=0-6, $(A_0) \leftarrow (C)$	○	-	1	1	F7
	RR A	$(A_n) \leftarrow (A_{n+1}), (A_7) \leftarrow (A_0)$	-	-	1	1	77
	RRC A	(A_n) $(A_{n+1}), (C) \leftarrow (A_0)$ n=0-6 $(A_7) \leftarrow (C)$	○	-	1	1	67
I/O Instruction	IN A,Pp	$(A) \leftarrow (Pp)$ p=0,1,2	-	-	1	2	08, 09, 0A
	OUTL Pp,A	$(Pp) \leftarrow (A)$ p=0,1,2	-	-	1	2	90, 39, 3A
	MOVD A, Pp	$(A_{0-3}) \leftarrow (Pp)$ p=4-7 $(A_{4-7}) \leftarrow 0$	-	-	1	2	0C-0F
	MOVD Pp,A	$(P_p) \leftarrow (A_{0-3})$ p=4-7	-	-	1	2	3C-3F
	ANLD Pp,A	$(Pp) \leftarrow (Pp) \wedge (A_{0-3})$ p=4-7	-	-	1	2	9C-9F
	ORLD Pp,A	$(Pp) \leftarrow (Pp) \vee (A_{0-3})$ p=4-7	-	-	1	2	8C-8F
Register Instruction	INC Rr	$(Rr) \leftarrow (Rr)+1$ r=0-7	-	-	1	1	18-1F
	INC @Rr	$((Rr)) \leftarrow ((Rr))+1$ r=0,1	-	-	1	1	10-11
Branch Instruction	JMP addr	$(PC_{0-7}) \leftarrow$ Upper 3-bit of Operation code	-	-	2	2	04,24,44,64 84,A4,C4,E4
	JMPP @A	$(PC_{0-7}) \leftarrow ((A))$	-	-	1	2	B3
	DJNZ Rr,addr	$(Rr) \leftarrow (Rr)-1$, If $(r) \neq 0$ $(PC_{0-7}) \leftarrow$ addr	-	-	2	2	E8-EF
	JC addr	IF (C)=1, $(PC_{0-7}) \leftarrow$ addr	-	-	2	2	F6

Classification	Mnemonics	Functional Description	Effected Flag C	Effected Flag AC	Bytes	Cycles	Operation Code (Hexadecimal)
Branch Instruction	JNC addr	If (C)=0, (PC_{0-7})←addr	-	-	2	2	E6
	JZ addr	If (A)=0, (PC_{0-7})←addr	-	-	2	2	C6
	JNZ addr	If (A)≠0, (PC_{0-7})←addr	-	-	2	2	96
	JT0 addr	If T0=1, (PC_{0-7}) ← addr	-	-	2	2	36
	JNT0 addr	If T0=0, (PC_{0-7}) ← addr	-	-	2	2	26
	JT1 addr	If T1=1, (PC_{0-7}) ← addr	-	-	2	2	56
	JNT1 addr	If T1=0, (PC_{0-7}) ← addr	-	-	2	2	46
	JTF addr	If TF=1, (PC_{0-7}) ← addr	-	-	2	2	16
Subroutine Instruction	CALL addr	((SP))←(PC),(SP)←(SP)+1 (PC_{0-7})←addr (PC_{8-10})←Upper 3-bit of operation code	-	-	1	2	14,34,54,74 94,B4,D4,F4
	RET	(SP) ← (SP)-1 (PC) ← ((SP))	-	-	1	2	83
Flags Instruction	CLR C	(C)←0	○	-	1	1	97
	CPL C	(C)←NOT(C)	○	-	1	1	A7
Data Moves Instruction	MOV A,Rr	(A) ← (Rr) r=0-7	-	-	1	1	F8-FF
	MOV A,@Rr	(A) ← ((Rr)) r=0,1	-	-	1	1	F0-F1
	MOV A,#data	(A) ← data	-	-	2	2	23
	MOV Rr,A	(Rr) ← (A) r=0-7	-	-	1	1	A8-AF
	MOV @Rr,A	((Rr)) ← (A) r=0,1	-	-	1	1	A0-A1
	MOV Rr,#data	(Rr) ← data r=0-7	-	-	2	2	B8-8F
	MOV @Rr,#data	((Rr)) ← data r=0,1	-	-	2	2	B0-B1

Classification	Mnemonics	Functional Description	Effected Flag C	Effected Flag AC	Bytes	Cycles	Operation Code (Hexadecimal)
Data Moves Instruction	XCH A,Rr	(A) $\rightleftarrows$ (Rr) r=0-7	-	-	1	1	28-2F
	XCH A,@Rr	(A) $\rightleftarrows$ ((Rr)) r=0,1	-	-	1	1	20-21
	XCHD a,@Rr	$(A_{0-3}) \rightleftarrows ((Rr))$ r=0,1	-	-	1	1	30-31
	MOVP A,@A	$(PC_{0-7}) \leftarrow$(A) [Note] (A) $\leftarrow$ ((PC))	-	-	1	2	A3
Timer/Counter Instruction	MOV A,T	(A) $\leftarrow$ (T)	-	-	1	1	42
	MOV T,A	(T) $\leftarrow$ (A)	-	-	1	1	62
	STRT T	Start timer	-	-	1	1	55
	STRT CNT	Start counter	-	-	1	1	45
	STOP TCNT	Stop timer/counter	-	-	1	1	65
A/D Converter Instruction	RAD	(A) $\leftarrow$ (CRR)	-	-	1	2	80
	SEL AN0	AN0 Selection, Conversion restart	-	-	1	1	85
	SNL AN1	AN1 " "	-	-	1	1	95
Interrupts Instruction	EI 1	Enable external interrupt	-	-	1	1	05
	DIS 1	Disable external interrupt	-	-	1	1	15
	EN TCNT1	Enable timer/counter interrupt	-	-	1	1	25
	DIS TCNT1	Disable timer/counter interrupt	-	-	1	1	35
	RETI	(SP) $\leftarrow$ (SP) - 1 (PC) $\leftarrow$ ((SP))	-	-	1	2	93
	NOP	No Operation	-	-	1	1	00

Note) MOVP A, @A loads the contents of address indicated by accumulator A in the page, into accumulatorA. After the execution, the contents of PC indicate the next address.

TMP8022P Instruction List (I)

IR0-3 / IR4-7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP			ADD A,#	JMP	ENI		DEC A	IN A,P0	IN A,P1	IN A,P2		MOVD A,P4	MOVD A,P5	MOVD A,P6	MOVD A,P7
1	INC @R0	INC @R1		ADDC A,#	CALL	DIS I	JTF	INC A	INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
2	XCH A,@R0	XCH A,@R1		MOV A,#	JMP	EN TCNTI	JNTO	CLR A	XCH A,R0	XCH A,R1	XCH A,R2	XCH A,R3	XCH A,R4	XCH A,R5	XCH A,R6	XCH A,R7
3	XCHD A,@R0	XCHD A,@R1			CALL	DIS TCNTI	JTO	CPL A		OUTL P1,A	OUTL P2,A		MOVD P4,A	MOVD P5,A	MOVD P6,A	MOVD P7,A
4	ORL A,@R0	ORL A,@R1	MOV A,T	ORL A,#	JMP	STRT CNT	JNT1	SWAP A	ORL A,R0	ORL A,R1	ORL A,R2	ORL A,R3	ORL A,R4	ORL A,R5	ORL A,R6	ORL A,R7
5	ANL A,@R0	ANL A,@R1		ANL A,#	CALL	STRT T	JT1	DA A	ANL A,R0	ANL A,R1	ANL A,R2	ANL A,R3	ANL A,R4	ANL A,R5	ANL A,R6	ANL A,R7
6	ADD A,@R0	ADD A,@R1	MOV ?,A		JMP	STOP TCNT		RRC A	ADD A,R0	ADD A,R1	ADD A,R2	ADD A,R3	ADD A,R4	ADD A,R5	ADD A,R6	ADD A,R7
7	ADDC A,@R0	ADDC A,@R1			CALL			RR A	ADDC A,R0	ADDC A,R1	ADDC A,R2	ADDC A,R3	ADDC A,R4	ADDC A,R5	ADDC A,R6	ADDC A,R7
8	RAD			RET	JMP	SEL ANO							ORLD P4,A	ORLD P5,A	ORLD P6,A	ORLD P7,A
9	OUTL P0,A			RETI	CALL	SEL AN1	JNZ	CLR C					ANLD P4,A	ANLD P5,A	ANLD P5,A	ANLD P7,A
A	MOV @R0,A	MOV @R1,A		MOVP A,@A	JMP			CPL C	MOV R0,A	MOV R1,A	MOV R2,A	MOV R3,A	MOV R4,A	MOV R5,A	MOV R6,A	MOV R7,A
B	MOV @R0,#	MOV @R1,#		JMPP @A	CALL				MOV R0,#	MOV R1,#	MOV R2,#	MOV R3,#	MOV R4,#	MOV R5,#	MOV R6,#	MOV R7,#
C					JMP		JZ									
D	XRL A,@R0	XRL A,@R1		XRL A,#	CALL				XRL A,R0	XRL A,R1	XRL A,R2	XRL A,R3	XRL A,R4	XRL A,R5	XRL A,R6	XRL A,R7
E					JMP		JNC	RL A	DJNZ R0,#	DJNZ R1,#	DJNZ R2,#	DJNZ R3,#	DJNZ R4,#	DJNZ R5,#	DJNZ R6,#	DJNZ R7,#
F	MOV A,@R0	MOV A,@R1			CALL		JC	RLC A	MOV A,R0	MOV A,R1	MOV A,R2	MOV A,R3	MOV A,R4	MOV A,R5	MOV A,R6	MOV A,R7

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
VCC	Supply Voltage	-0.5 ∿ +7	V
VINA	Input Voltage(Except PRESET,PROG,T1)	-0.5 ∿ +7	V
VINB	Input Voltage (Only PRESET,PROG,T1)	-0.5 ∿ +13	V
P_D	Power Dissipation	1.0	W
T_{opr}	Operating Temperature	0 ∿ 70	°C
T_{stg}	Storage Temperature	-55 ∿ 150	°C

DC CHARACTERISTICS T_{opr} = °C ∿ 70°C, VCC = 5.5V ± 1V, V_{SS} = 0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
VIL	Input Low Voltage	VTH Open	-0.5		0.8	V
VIL	Input Low Voltage (PORT0)		-0.5		VTH -0.1	V
VIH	Input High Voltage (All Except XTAL, RESET)	VCC=5.0V±10% VTH Open	2.0		VCC	V
VIH1	Input High Voltage (All Except XTAL, RESET)	VCC=5.5V±1V VTH Open	3.0		VCC	V
VIH2	Input High Voltage (PORT0)		VTH +0.1		VCC	V
VIH3	Input High Voltage (PRESET,XTALI)		3.0		VCC	V
VTH	PORT0 Threshold Comparison Voltage		0		0.4 VCC	V
VOL	Output Low Voltage	IOL = 1.6mA			0.45	V
VOL1	Output Low Voltage (P10, P11)	IOL = 7 mA			2.5	V
VOH	Output High Voltage (All unless Open Drain Option-Port 0)	IOH = -50μA	2.4			V
IL1	Input Current (T_1)	VSS=0.45V≤VIN≤VCC			±200	μA
IL0	Output Leak Current (Open Drain Option-Port 0)	VSS=0.45V≤VIN≤VCC			±10	μA
ICC	V_{CC} Supply Current		50		100	mA

AC CHARACTERISTICS 1 Ta = 0°C ∿ 70°C V_{CC} = 5.5V ± 1V, V_{SS} = 0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	MAX.	UNIT
t_{CY}	Cycle Time	At 3MHz XTAL 10µs	8.38	50.0	µs
V_{ZX}	Zero-cross Detection Input (T1)		1	3	VACpp
A_{ZX}	Zero-cross Acuracy	60Hz Sinewave		±135	mV
F_{ZX}	Zero-cross Detection Input Frequency		0.05	1	KHz

AC CHARACTERISTICS 2 Ta = 0° ∿ 70°C V_{CC} = 5.5V ± 1V, V_{SS} = 0V

SYMBOL		PARAMETER	CONDITION	MIN.	MAX.	UNIT
t_{CP}	Expander Operation	Port Control Setup Before Falling Edge of Prog		0.5		µs
t_{PC}		Port Control Hold After Falling Edge of Prog		0.8		µs
t_{PR}		Prog to Time P2 Input must be Valid			1.0	µs
t_{DP}		Output Data Setup Time		7.0		µs
t_{PD}		Output Data Hold Time		8.3		µs
t_{PF}		Input Data Hold Time		0	150	µs
t_{PP}		PROG Pulse Width		8.3		µs
t_{PRL}	Normal Operation	ALE to Time P2 Input must be Valid			3.6	µs
t_{PL}		Output Data Setup Time		0.8		µs
t_{LP}		Output Data Hold Time		1.6		µs
t_{PFL}		Input Data Hold Time		0		µs
t_{LL}		ALE Pulse Width	Max. at t_{CY}=8.38µs	3.9	23.0	µs

Test Condition t_{CY} = 8.38 µs C_L = 80 pF

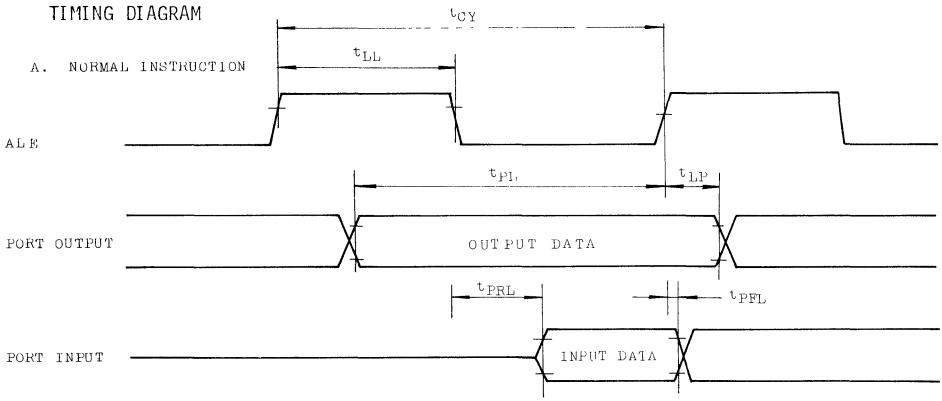
A/D CONVERTER CHARACTERISTICS Ta=0°C 70°C, V_{CC}=5.5V±1V, V_{SS}=0V, AV_{CC}=5.5±1V, AV_{SS}=0V, $AV_{CC}/2 \leq VAREF \leq AV_{CC}$

PARAMETER	MIN.	TYP.	MAX.	UNIT	REMARK
Resolution	8			Bits	
Absolute Accuracy			.8% FST ± 1/2LSB	LSB	Note 1)
Sample Setup Before Falling Edge of ALE (t_{SS})		0.20		t_{CY}	
Sample Hold After Falling Edge of ALE (t_{SH})		0.10		t_{CY}	
Input Capacitance		1		pF	
Conversion Time	4		4	t_{CY}	

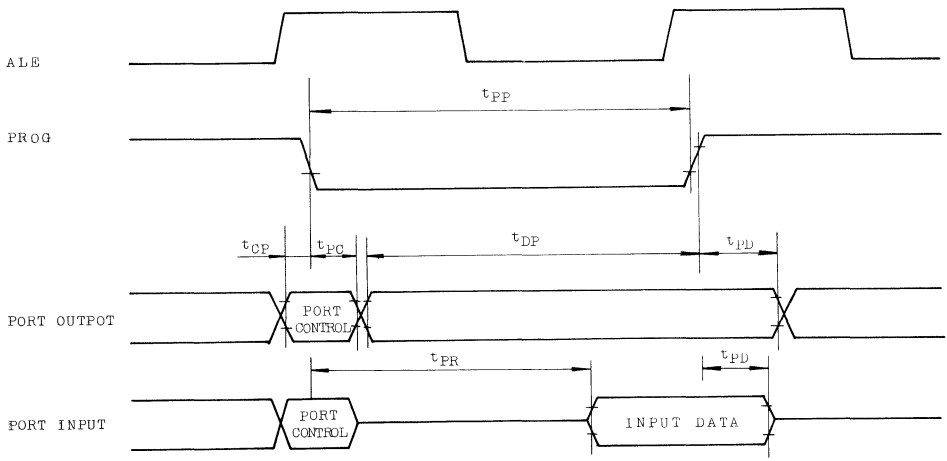
Note 1) It is required that the analog input terminal is kept at a constant voltage during the smapling time (t_{SS} + t_{SH}).

TIMING DIAGRAM

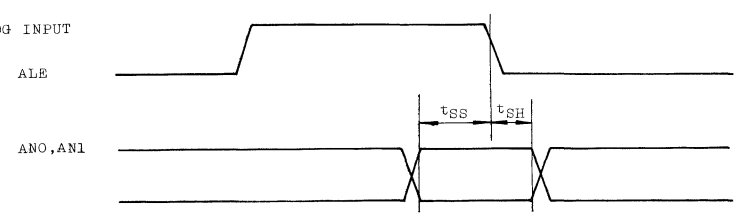
A. NORMAL INSTRUCTION



B. EXPANDER INSTRUCTION



C. ANALOG INPUT



PROGRAM DELIVERY OF TMP8022P

The program delivery of the TMP8022P is performed by using a paper tape of the following format. At the same time, it is required that mask options should be clearly designated. The format of the paper tape is the same as the Intel's type object tape (hexadecimal tape output by Intel MDS system, PROMPT 48 Development Tool, etc.)

(1) Mask Option

It is required that the presence of pull-up resistors is designated as to the 8 bits of PORT0, and the T1 terminal.

It is required that a mask option designation form attached to the ES Order Instruction Sheet is used for designation of mask option.

It is required that the mask option designation form is submitted together with the ES Order Instruction Manual within two weeks before the sumbission date of tape.

Example of mask option designation

0 : Without pull-up register 1 : With pull-up register

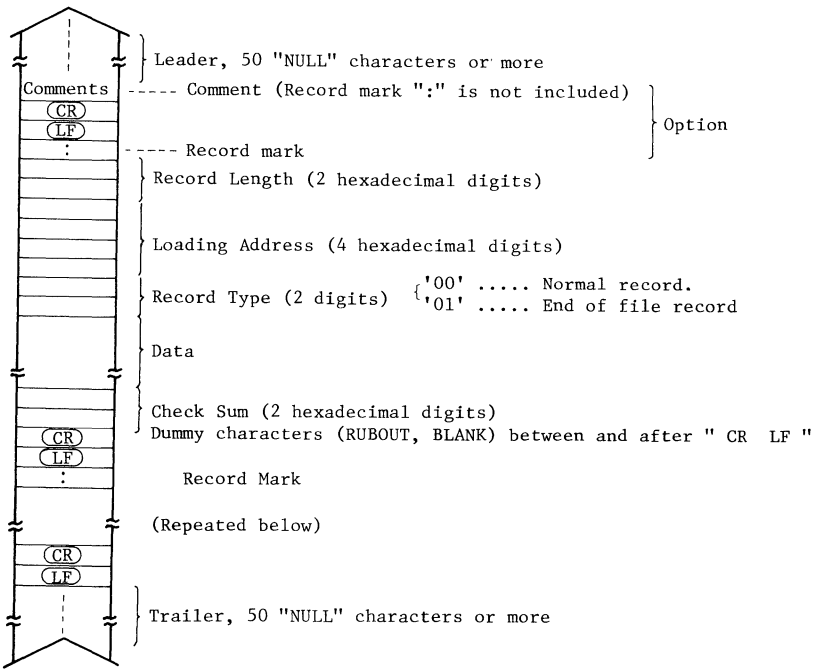
Terminal name / Option designation	PORT0								T1
	7	6	5	4	3	2	1	0	
Presence of pull-up resister	1	0	0	0	1	1	0	0	1

In this case, the presence of pull-up resistors is as follows:

Pins with pull-up resistors ..... P07, P03, P02 and T1

Pins without pull-up resistors ... P06, P05, P04, P01 and P00

(2) Tape Format

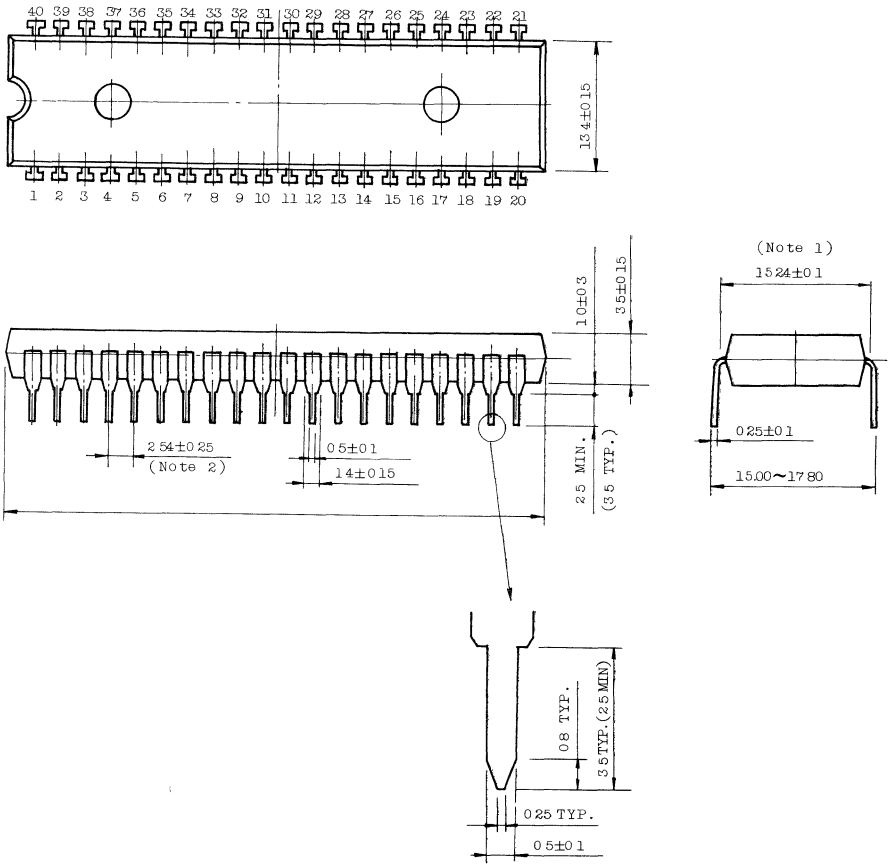


(3) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLSC-84
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884dde67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                    :
                    :
                    :
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1E41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20BC372BF60BD6
: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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8243P/TMP8243PI

N-CHANNEL SILICON GATE MOS

INPUT/OUTPUT EXPANDER

GENERAL DESCRIPTION

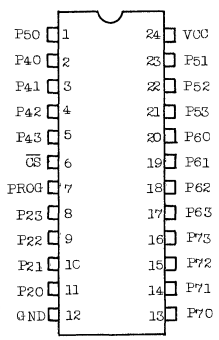
The TMP8243P is an input/output expander designed specifically to provide a low cost means of I/O expansion for the TLCS-84 family.

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.

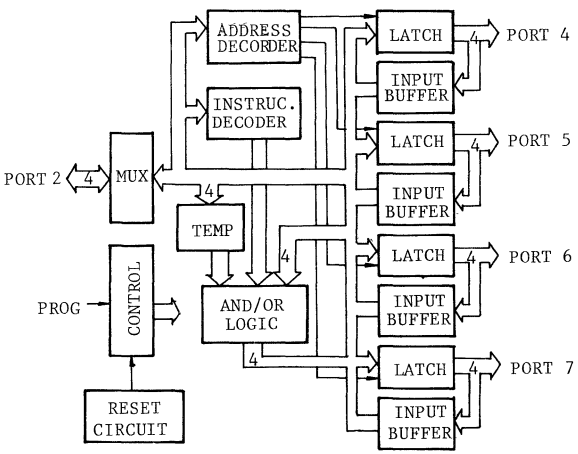
FEATURES

- 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

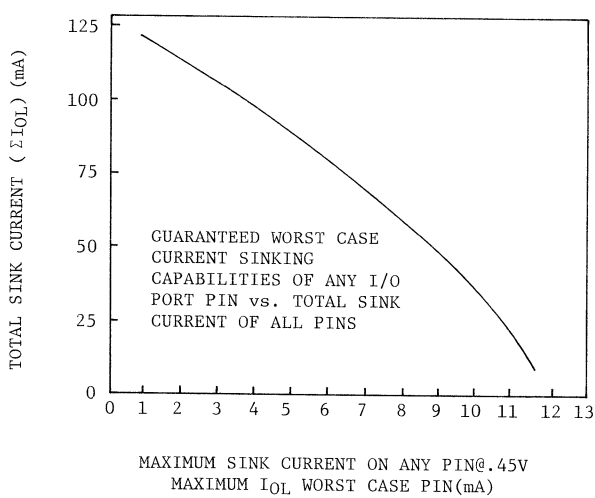
The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi, A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi, A takes new data AND's it with the old data and then writes it to the port. Operation code and 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_{LP1}	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%

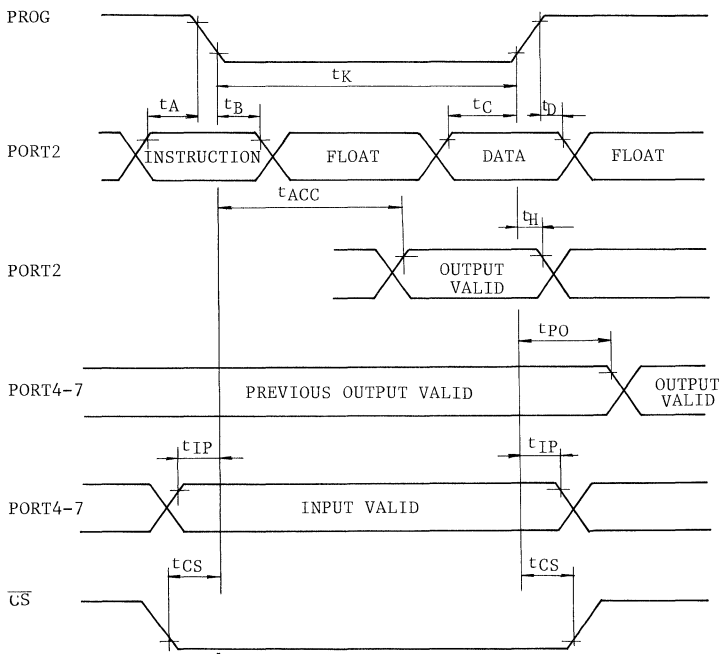
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
V_{OL1}	Outptu Low Voltage Ports 4-7	I_{OL}=4.5mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL}=20mA			1	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.6mA			0.45	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%

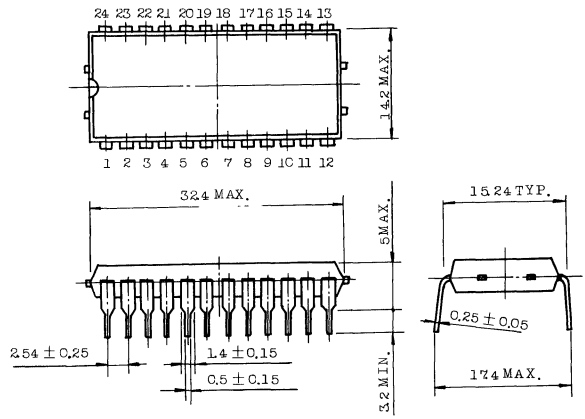
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
t_A	Code Valid before PROG	C_L = 80pF	100			ns
t_B	Code Valid after PROG	C_L = 20pF	60			ns
t_C	Data Valid before PROG	C_L = 80pF	200			ns
t_D	Data Valid after PROG	C_L = 20pF	20			ns
t_H	Floating after PROG	C_L = 20pF	0		150	ns
t_K	PROG Negative Pulse 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_{LP1}	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.

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-48

CMOS DEVICES

July. 1 9 8 4

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL
INTEGRATED CIRCUIT

TMP80C48AP , TMP80C35AP

Silicon Monolithic
CMOS Silicon Gate

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP80C48AP 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 × 8 RAM data memory, 1K × 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

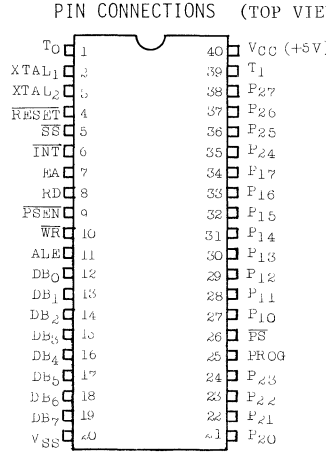
The TMP80C48AP is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

The TMP80C35AP is the equivalent of a TMP80C48AP without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

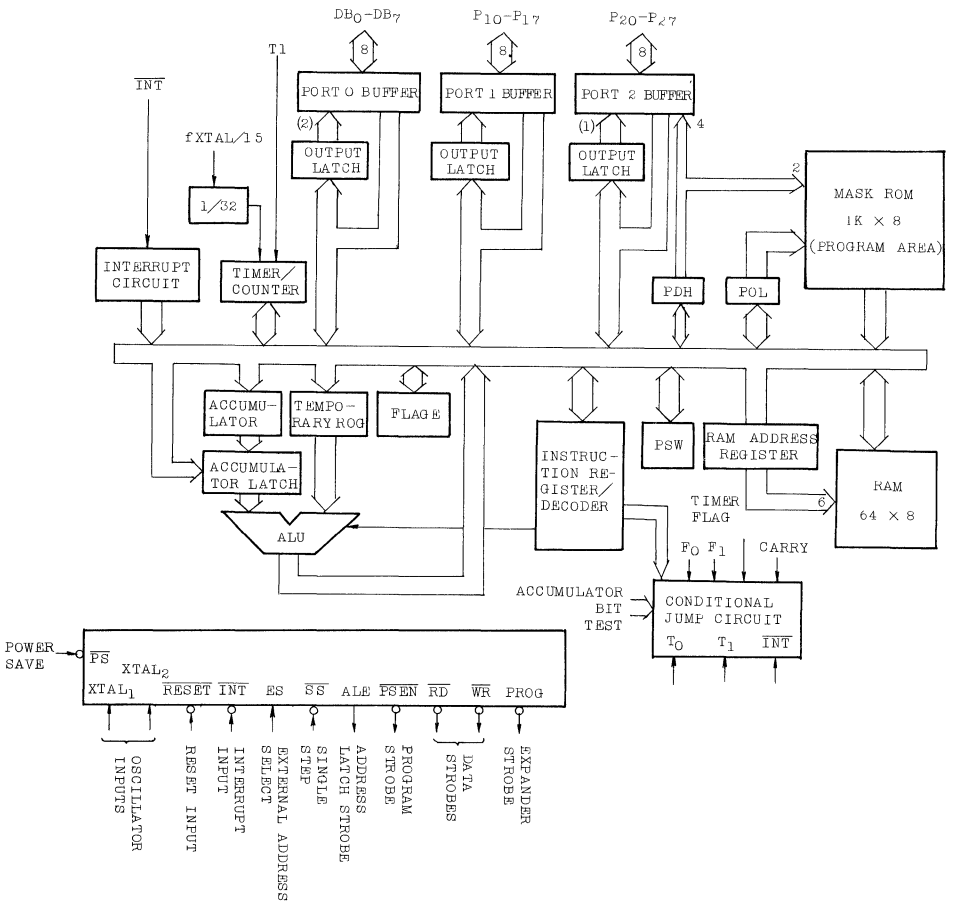
FEATURES

- 2.5 μs Instruction Cycle Time
  -40°C to 85°C, 5V±20%
- Software Compatible with
  TMP80C49AP/-6
- Software Upward Compatible with
  TMP8049P/TMP80C49P-6/INTEL's 8049
- HALT Instruction (Additional Instruction)
- Low Power
  10mA MAX. in Normal Operation
  (V_{CC}=5V, f_{XTAL}=6MHz)
  10μA MAX. in Power Down Mode
  (V_{CC}=5V, f_{XTAL}:DC)
- Single power supply
- 1K × 8 masked ROM
- 64 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Power Down Mode (Stand-by Mode)
- Halt Mode (Idel Mode)

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

V_{SS} (Power Supply)
Circuit GND potential

V_{CC} (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 and 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 $\overline{WR}$.

T_0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T_0 can be designated as a clock output using ENT0 CLK instruction.

T_1 (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 the power down mode.

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 executed. Also used during the powerdown mode.

EA (Input)
External Access input which forces all porgram 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.

INSTRUCTION SET

Refer to TMP80C49AP/-6 INSTRUCTION SET.

TMP80C48AP/TMP80C35AP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to V_{CC}+0.5V
V_{INB}	Input Voltage (Only EA)	-0.5V to 13V
P_D	Power Dissipation (Ta=85°C)	250mW
T_{SOLDER}	Soldering Temperature (Soldering Timer 10 sec)	260°C
T_{STG}	Storage Temperature	-65°C to 150°C
T_{OPR}	Operating Temperature	-40°C to 85°C

DC CHARACTERISTICS (I)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±10%, V_{SS}=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	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.7V_{CC}$	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH11}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-1.6mA	2.4	-	-	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC}-0.8	-	-	V
V_{OH21}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-50μA	2.4	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	-50	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leq V_{IN} \leq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Currevt	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz $V_{IH}=V_{CC}$-0.2V	-	-	10	mA
I_{CCH1}		HALT Mode	V_{IL}=0.2V	-	-	T.B.D.	

TMP80C48AP/TMP80C35AP ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (II)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage			-0.5	-	$0.15V_{CC}$	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.5V_{CC}$	-	V_{CC}	V
V_{IH1}	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.7V_{CC}$	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC}-0.8	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.1}$	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.01}$	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz $V_{IH}=V_{CC}$-0.2V	-	-	10	mA
I_{CCH1}		HALT Mode	V_{IL}=0.2V	-	-	T.B.D.	

TMP80C48AP/TMP80C35AP ELECTRICAL CHARACTERISTICS

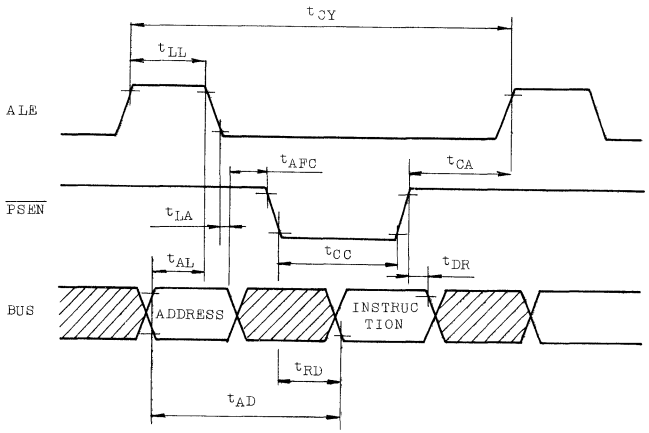
AC CHARACTERISTICS T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		400	-	-	ns
t_{AL}	Address Setup Time (ALE)		150	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		700	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		500	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	120	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	500	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	950	ns
t_{AFC}	Address Fload Time ($\overline{RD}$, $\overline{PSEN}$)		0	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		110	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		130	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	810	ns
t_{DP}	Output Data Setup Time (PROG)		220	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		65	-	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	-	150	ns
t_{PP}	PROG Pulse Width		1510	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		600	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		150	-	-	ns

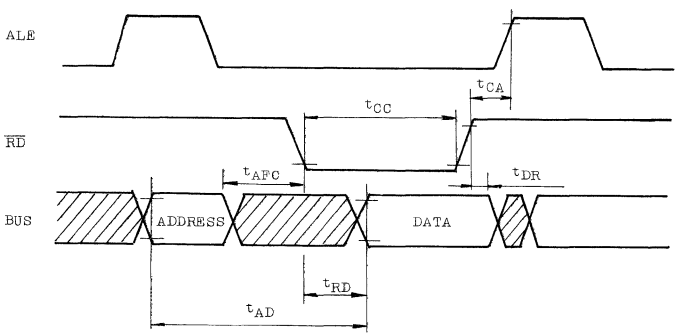
Note: t_{CY}=2.5μs (f_{XTAL}=6MHz)
Control Outputs: C_L=80pF, BUS Outputs: C_L=150pF

TIMING WAVEFORM

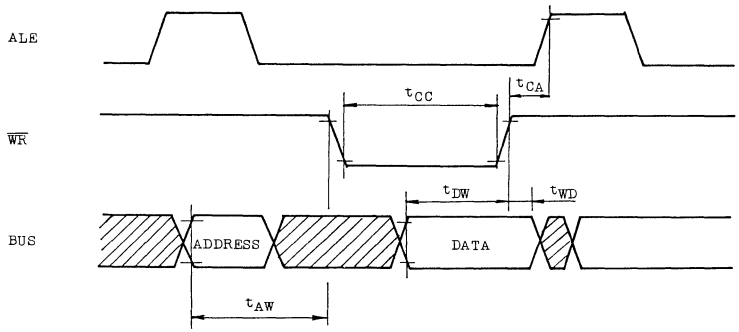
A. Instruction Fetch from External Program Memory



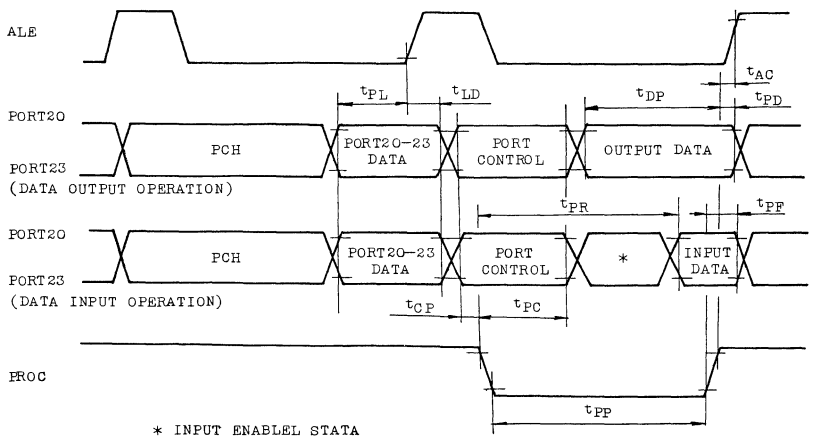
B. Read from External Data Memory



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 V_{CC} in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resum oscillation after V_{CC} 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 T_{OPR}=-40°C to 85°C, V_{SS}=0V

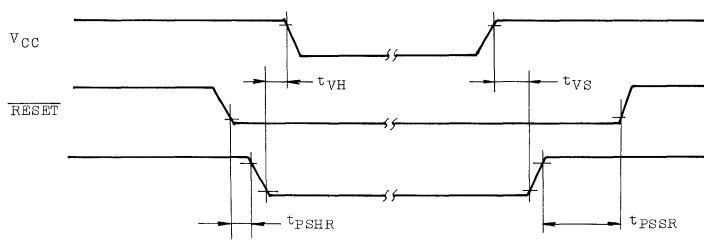
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB1}	Standby Voltage(1)		2.0	-	6.0	V
I_{SB1}	Standby Current(1)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS T_{OPR}=-40°C to 70°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHR}	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μs
t_{PSSR}	Power Save Setup Time ($\overline{RESET}$)		10	-	-	ms
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μs
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μs

Note: t_{CY}=2.5μs (f_{XTAL}=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 V_{CC} in this mode is 3V.

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

DC CHARACTERISTICS T_{OPR}=-40°C to 85°C, V_{SS}=0V

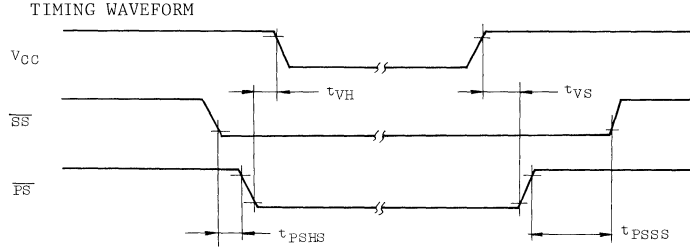
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB2}	Supply Voltage (2)		3.0	-	6.0	V
I_{SB2}	Standby Current (2)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

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

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHS}	Power Save Hold Time ($\overline{SS}$)		10	-	-	μs
t_{PSSS}	Power Save Setup Time ($\overline{SS}$)		10	-	-	ms
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μs
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μs

Note: t_{CY}=2.5μs (f_{XTAL}=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, TMP80C48AP, TMP80C35AP entry HALT MODE.

• 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logics are disabled. The status of all 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 descibed in the following table.

• 4 Release from HALT MODE

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

(1) $\overline{RESET}$ Release Mode: An active $\overline{RESET}$ input signal causes the normal reset function. TMP80C48AP, TMP80C35AP start the program at address "000H".

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

In case of interrupt enable mode (EI MODE), TMP80C48AP, TMP80C35AP 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), TMP80C48AP, TMP80C35AP 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)

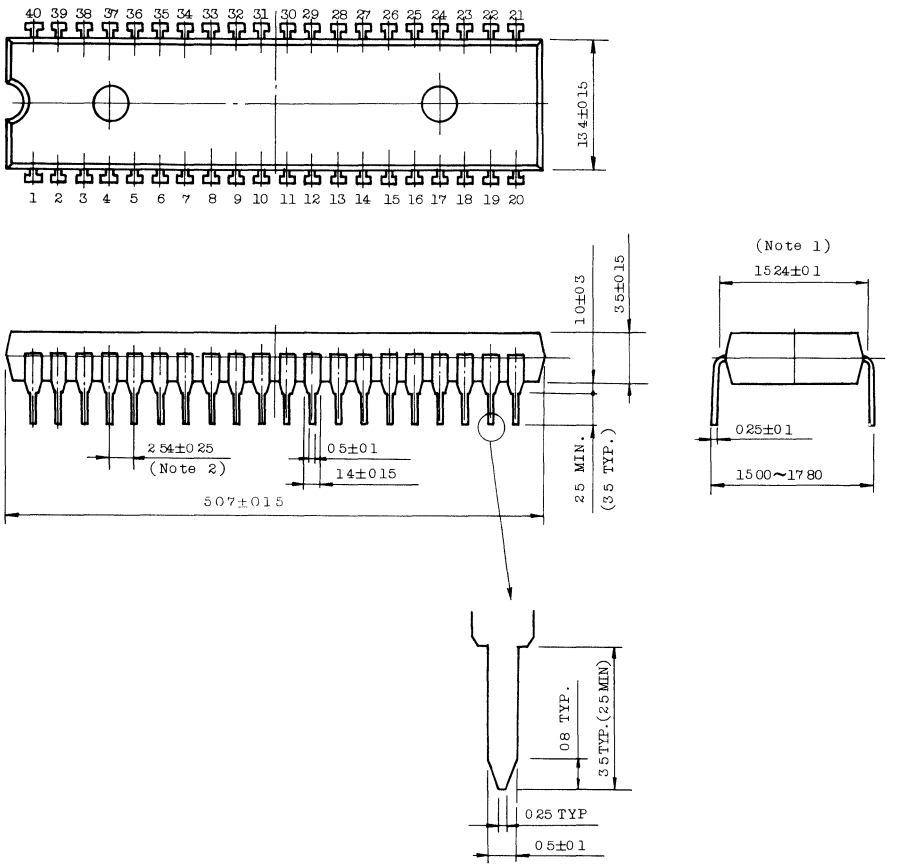
PIN NAME	STATUS
DB0 - DB7	High impedance
P10 - P17	Input disabled
P20 - P27	
T0	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{\text{RESET}}$, $\overline{\text{SS}}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{\text{INT}}$, 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	Values prior to the execution of HALT INSTRUCTION are maintained.
P10 - P17	
P20 - P27	
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

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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL
INTEGRATED CIRCUIT

TMP80C49AP , TMP80C39AP
TMP80C49AP-6 , TMP80C39AP-6

8-BIT SINGLE-CHIP MICROCOMPUTER

Silicon Monolithic
CMOS Silicon Gate

GENERAL DESCRIPTION

The TMP80C49AP/-6 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 × 8 RAM data memory, 2K × 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

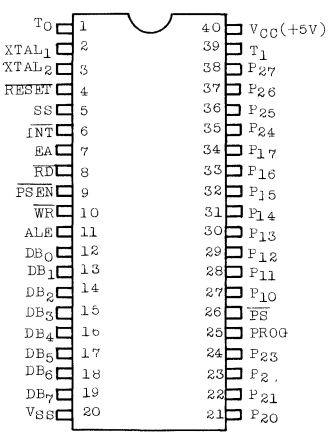
The TMP80C49AP/-6 is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

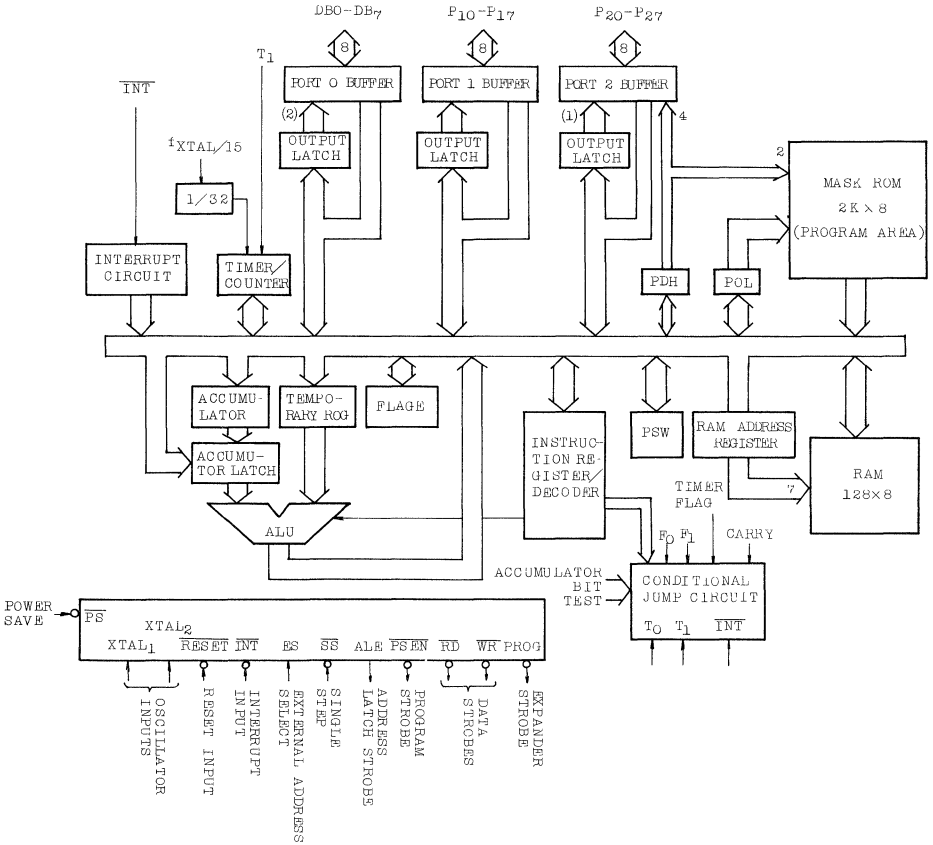
The TMP80C39AP/-6 is the equivalent of a TMP80C49AP/-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

FEATURES

- TMP80C49AP/TMP80C39AP
  1.36μs Instruction Cycle Time
  0°C to 70°C, 5V ±10%
- TMP80C49AP-6/TMP80C39AP-6
  2.5 μs Instruction Cycle Time
  -40°C to 85°C, 5V ±20%
- Software Upward Compatible with TMP8049P/TMP80C49P-6/INTEL's 8049
- HALT Instruction (Additional Instruction)
- Low Power
  10mA MAX. in Normal Operation
  (V_{CC}=5V, f_{XTAL}=6MHz)
  10μA MAX. in Power Down Mode
  (V_{CC}=5V, f_{XTAL}:DC)
- Single power supply
- 2K × 8 masked ROM
- 128 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Power Down Mode (Stand-by Mode)
- Halt Mode (Idel Mode)

PIN CONNECTIONS (TOP VIEW)





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

V_{SS} (Power Supply)
Circuit GND potential

V_{CC} (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 $\overline{WR}$.

T_0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T_0 can be designated as a clock output using ENT0 CLK instruction.

T_1 (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 (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 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.

INSTRUCTION SET

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 d7	0 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data	2	2	o	o
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 d7	0 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)+Data+(C)	2	2	o	o
ANL A,Rr	0	1	0	1	1	r	r	r	(A)←(A)∧(Rr) r = 0 - 7	1	1	-	-
ANL A,@Rr	0	1	0	1	0	0	0	r	(A)←(A)∧((Rr)) r = 0, 1	1	1	-	-
ANL A,#Data	0 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∧Data	2	2	-	-
ORL A,Rr	0	1	0	0	1	r	r	r	(A)←(A)∨(Rr) r = 0 - 7	1	1	-	-
ORL A,@Rr	0	1	0	0	0	0	0	r	(A)←(A)∨((Rr)) r = 0, 1	1	1	-	-
ORL A,#Data	0 d7	1 d6	0 d5	0 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∨Data	2	2	-	-
XRL A,Rr	1	1	0	1	1	r	r	r	(A)←(A)∀(Rr) r = 0 - 7	1	1	-	-
XRL A,@Rr	1	1	0	1	0	0	0	r	(A)←(A)∀((Rr)) r = 0, 1	1	1	-	-
XRL A,#Data	1 d7	1 d6	0 d5	1 d4	0 d3	0 d2	1 d1	1 d0	(A)←(A)∀Data	2	2	-	-
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	-	-
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,P_P	0	0	0	0	1	0	P	P	(A)←(P_P) P = 1, 2	1	2	-	-
OUTL P_P,A	0	0	1	1	1	0	P	P	(P_P)←(A) P = 1, 2	1	2	-	-
ANL P_P,#Data	1	0	0	1	1	0	P	P	(P_P)←(P_P)∧Data P = 1, 2	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
ORL P_P,#Data	1	0	0	0	1	0	P	P	(P_P)←(P_P)∨Data P = 1, 2	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
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)∧Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
ORL BUS,#Data	1	0	0	0	1	0	0	0	(BUS)←(BUS)∨Data	2	2	-	-
	d7	d6	d5	d4	d3	d2	d1	d0					
MOVD A,P_P	0	0	0	0	1	1	P	P	(A0-3)←(P_P) (A4-7)←0 P = 4 - 7	1	2	-	-
MOVD P_P,A	0	0	1	1	1	1	P	P	(P_P)←(A0-3) P = 4 - 7	1	2	-	-
ANLD P_P,A	1	0	0	1	1	1	P	P	(P_P)←(P_P)∧(A0-3) P = 4 - 7	1	2	-	-
ORLD P_P,A	1	0	0	0	1	1	P	P	(P_P)←(P_P)∨(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)	2	2	-	-
	a7	a6	a5	a4	a3	a2	a1	a0	if C = 1 (PC) = (PC)+2 if C = 0				

Mnemonic	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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) ≠ 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) ≠ 0				
									(PC)←(PC)+2				
									if (A) = 0				
JT0 Address	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				
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 $\overline{INT}$ = 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)				

Subroutine Instruction

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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

Mnemonics	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

Mnemonics	Instruction								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	-	-
	d7	d6	d5	d4	d3	d2	d1	d0	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				

Mnemonics	Instruction Code								Operation	Bytes	Cycles	Flag	
	D7	D6	D5	D4	D3	D2	D1	D0				C	AC
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

Mnemonics	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

Mnemonics	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	-	-
ENT0 CLK	0	1	1	1	0	1	0	1	T_0 is enabled to act as the clock output.	1	1	-	-
NOP	0	0	0	0	0	0	0	0	No operation	1	1	-	-
HALT	0	0	0	0	0	0	0	1	CPU enters HALT mode.	1	1	-	-

TMP80C49AP/TMP80C39AP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to V_{CC}+0.5V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=70°C)	250mW
T_{SOLDER}	Soldering Temperature (Soldering Time 10 sec)	260°C
T_{STG}	Storage Temperature	-65°C to 150°C
T_{OPR}	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS T_{OPR}=0°C to 70°C, V_{CC}=+5V±10%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.8	V
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7V_{CC}	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH11}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-1.6mA	2.4	-	-	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC} -0.8	-	-	V
V_{OH21}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-50μA	2.4	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC} -0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	-50	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, TO) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal operation	V_{CC}=5V, f_{XTAL}=6MHz V_{IH}=V_{CC}-0.2V	-	-	10	mA
I_{CCH1}		HALT Mode	V_{IL}=0.2V	-	-	T.B.D.	
I_{CC2}	V_{CC} Supply Current	Normal operation	V_{CC}=5V, f_{XTAL}=11MHz V_{IH}=V_{CC}-0.2V	-	-	15	mA
I_{CCH2}		HALT Mode	V_{IL}=0.2V	-	-	T.B.D.	

TMP80C49AP/TMP80C39AP ELECTRICAL CHARACTERISTICS

AC CHARACTERISTICS

T_{OPR}=0°C to 70°C, V_{CC}=+5V±10%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		150	-	-	ns
t_{AL}	Address Setup Time (ALE)		160	-	-	ns
t_{LA}	Address Hold Time (ALE)		50	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		350	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		390	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	CL=20pF	40	-	-	ns
t_{CY}	Cycle Time		1.36	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	110	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	210	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		400	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	570	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		10	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		100	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		160	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	700	ns
t_{DP}	Output Data Setup Time (PROG)		400	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		90	-	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	-	140	ns
t_{PP}	PROG Pulse Width		700	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		160	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		40	-	-	ns

Note : t_{CY}=1.36μs (f_{XTAL}=11 MHz)

Control Outputs : C_L=80pF, BUS Outputs : C_L=150pF

TMP80C49AP-6/TMP80C39AP-6 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to V_{CC}+0.5V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=85°C)	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

DC CHARACTERISTICS (I)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±10%, V_{SS}=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	-	V_{CC}	V
V_{IH1}	Input High Voltage ($\overline{XTAL1}$, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7V_{CC}	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH11}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-1.6mA	2.4	-	-	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC} -0.8	-	-	V
V_{OH21}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-50μA	2.4	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC} -0.8	-	-	V
I_{LI}	Input Leak Current ($\overline{T1}$, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	-50	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz V_{IH}=V_{CC}-0.2V	-	-	10	mA
I_{CCH1}		HALT Mode	V_{IL}=0.2V	-	-	T.B.D.	

TMP80C49AP-6/TMP80C39AP-6 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (I1)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage			-0.5	-	$0.15 V_{CC}$	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.5V_{CC}$	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.7V_{CC}$	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC} -0.8	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC} -0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	$-\frac{V_{CC}}{0.1}$	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leq V_{IN} \leq V_{CC}$	-	-	$-\frac{V_{CC}}{0.01}$	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz, $V_{IH}=V_{CC}$-0.2V, V_{IL}=0.2V	-	-	10	mA
I_{CCH1}		HALT Mode		-	-	T.B.D.	

TMP80C49AP-6/TMP80C39AP-6 ELECTRICAL CHARACTERISTICS

AC CHARACTERISTICS

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

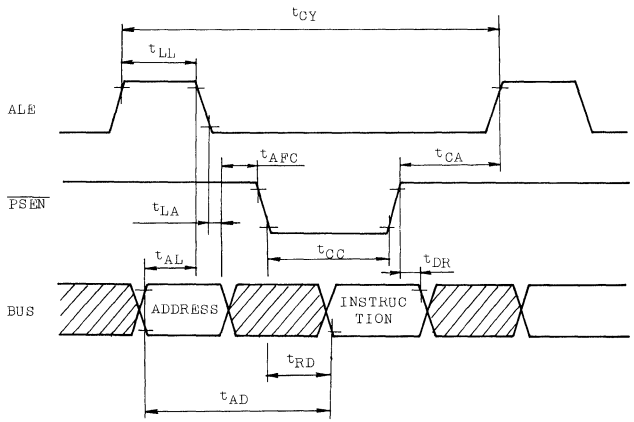
SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNITS
t_{LL}	ALE Pulse Width		400	-	-	ns
t_{AL}	Address Setup Time (ALE)		150	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		700	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		500	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	120	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	500	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	950	ns
t_{AFC}	Address Fload Time ($\overline{RD}$, $\overline{PSEN}$)		0	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		110	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		130	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	810	ns
t_{DP}	Output Data Setup Time (PROG)		220	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		65	-	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	-	150	ns
t_{PP}	PROG Pulse Width		1510	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		600	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		150	-	-	ns

Note : t_{CY}=2.5μs (f_{XTAL}=6 MHz)

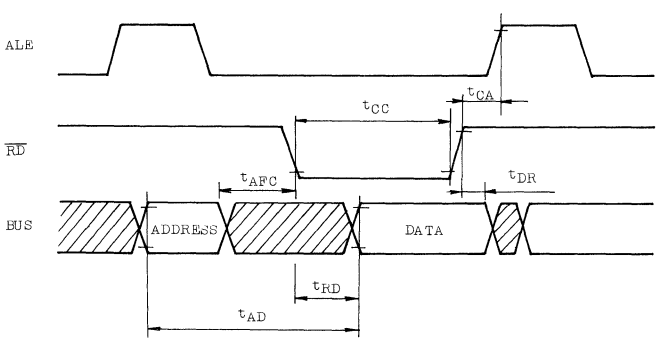
Control Outputs : C_L=80pF, BUS Outputs : C_L=150pF

TIMING WAVEFORM

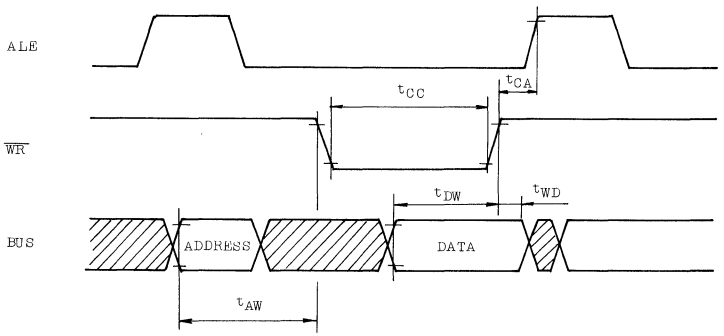
A. Instruction Fetch from External Program Memory



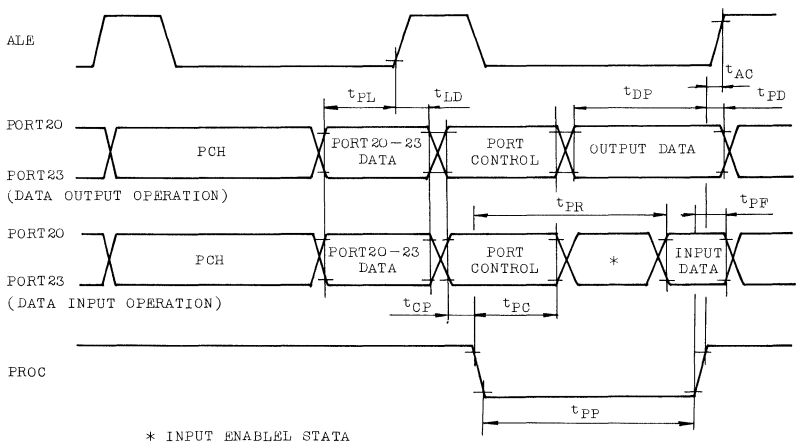
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



* INPUT ENABLEL STATA

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 V_{CC} in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resum oscillation after V_{CC} 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/TMP80C39AP : T_{OPR}=0°C to 70°C, V_{SS}=0V

TMP80C49AP-6/TMP80C39AP-6 : T_{OPR}=-40°C to 85°C, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB1}	Standby Voltage(1)		2.0	-	6.0	V
I_{SB1}	Standby Current(1)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS

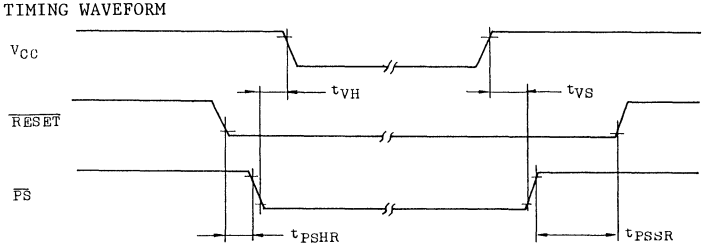
TMP80C49AP/TMP80C39AP : T_{OPR}=0°C to 70°C, V_{CC}=5V±10%, V_{SS}=0V

TMP80C49AP-6/TMP80C39AP-6 : T_{OPR}=-40°C to 70°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHR}	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μS
t_{PSSR}	Power Save Setup Time ($\overline{RESET}$)		10	-	-	mS
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μS
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μS

Note : t_{CY}=2.5μs (f_{XTAL}=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 V_{CC} in this mode is 3V.

$\overline{PS}$ terminal is set to high level to resume oscillation after V_{CC} 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/TMP80C39AP : T_{OPR}=0°C to 70°C, V_{SS}=0V

TMP80C49AP-6/TMP80C39AP-6 : T_{OPR}=-40°C to 85°C, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB2}	Standby Voltage(2)		3.0	-	6.0	V
I_{SB2}	Standby Current(2)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS

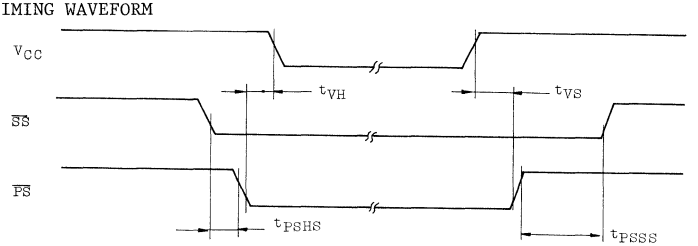
TMP80C49AP/TMP80C39AP : T_{OPR}=0°C to 70°C, V_{CC}=5V±10%, V_{SS}=0V

TMP80C49AP-6/TMP80C39AP-6 : T_{OPR}=-40°C to 85°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHS}	Power Save Hold Time ($\overline{SS}$)		10	-	-	μS
t_{PSSS}	Power Save Setup Time ($\overline{SS}$)		10	-	-	mS
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μS
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μS

Note : t_{CY}=2.5μs (f_{XTAL}=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, TMP80C49AP/-6, TMP80C39AP/-6 enter HALT MODE.

• 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logics are disabled. The status of all 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 ($\overline{RESET}$, $\overline{INT}$).

(1) $\overline{RESET}$ Release Mode : An active $\overline{RESET}$ input signal causes the normal reset function. TMP80C49AP/-6, TMP80C39AP/-6 start the program at address "000H".

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

In case of interrupt enable mode (EI MODE), TMP80C49AP/-6. TMP80C39AP/-6 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), TMP80C49AP/-6, TMP80C39AP/-6 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)

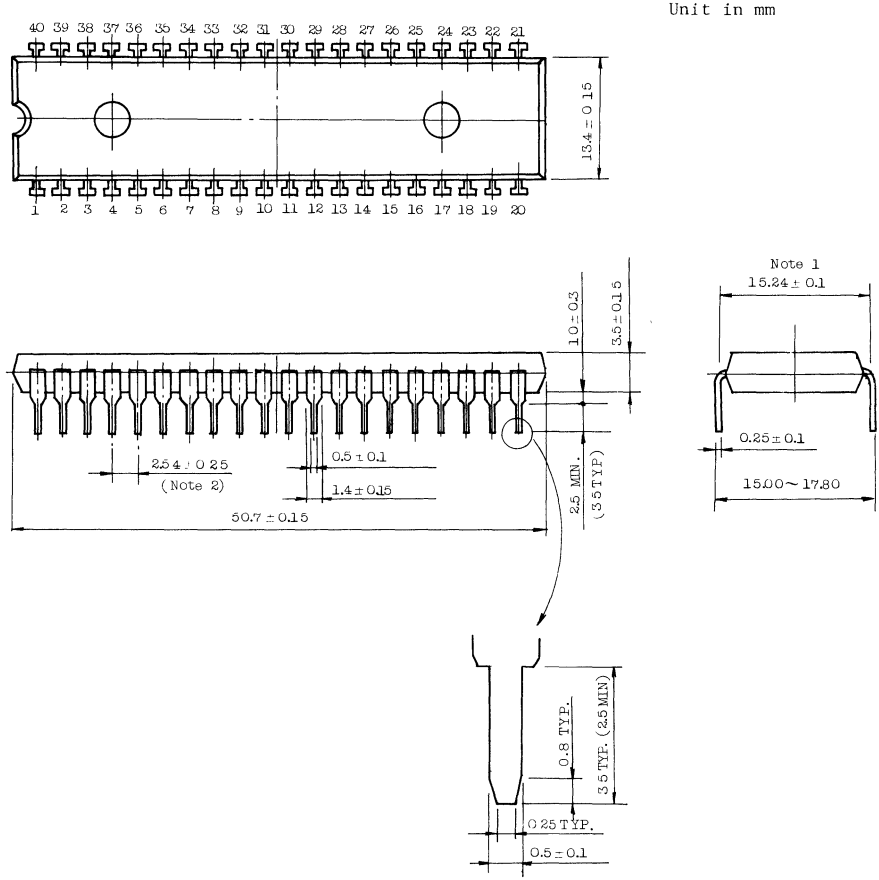
PIN NAME	STATUS
DB_0 - DB_7	High impedance Input disabled
P_{10} - P_{17}	
P_{20} - P_{27}	
T0	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{RESET}$, $\overline{SS}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{INT}$, EA	Input disabled when oscillator is stopped.
$\overline{RD}$, $\overline{WR}$, ALE PROG, $\overline{PSEN}$	High impedance

PIN STATUS IN HALT MODE

PIN NAME	STATUS
DB_0 - DB_7	Values prior to the execution of HALT INSTRUCTION are maintained.
P_{10} - P_{17}	
P_{20} - P_{27}	
T0	Status prior to the execution of HALT INSTRUCTION is maintained.
T1	Input disabled
XTAL1, XTAL2	Continue oscillation
$\overline{RESET}$, $\overline{INT}$	Input enabled
$\overline{SS}$, $\overline{EA}$	Input disabled
$\overline{RD}$, $\overline{WR}$, PROG, $\overline{PSEN}$	Output "High" level
ALE	Output "Low" level

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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL
INTEGRATED CIRCUIT

TMP80C50AP , TMP80C40AP
TMP80C50AP-6 , TMP80C40AP-6

Silicon Monolithic
CMOS Silicon Gate

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP80C50AP/-6 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 × 8 RAM data memory, 4K × 8 ROM program memory, 27 I/O lines and an 8-bit timer/event counter.

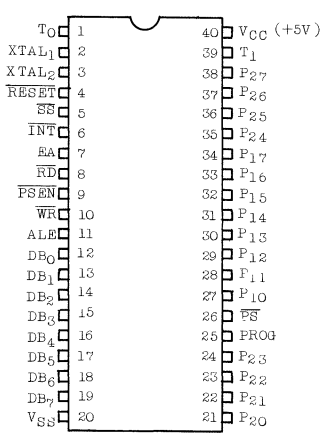
The TMP80C50AP/-6 is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

The TMP80C40AP/-6 is the equivalent of a TMP80C50AP/-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

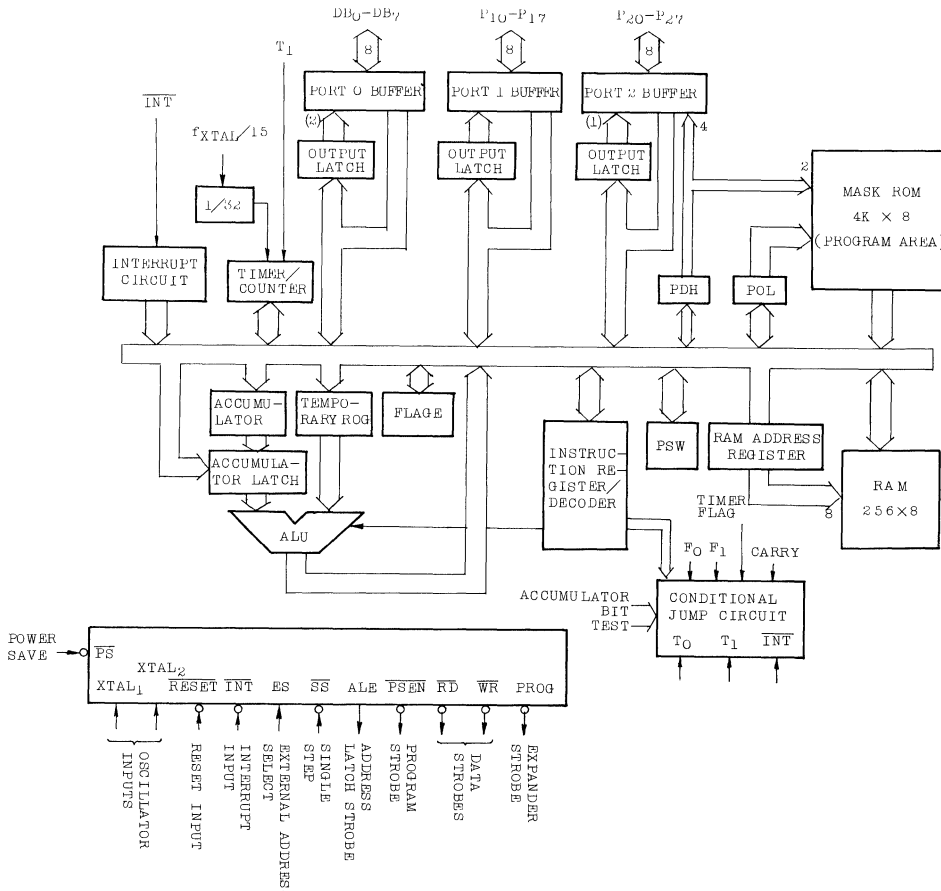
FEATURES

- TMP80C50AP/TMP80C40AP
  1.36μs Instruction Cycle Time
  0°C to 70°C, 5V±10%
- TMP80C50AP-6/TMP80C40AP-6
  2.5 μs Instruction Cycle Time
  -40°C to 85°C, 5V±20%
- Software Compatible with TMP80C49AP/-6
- Software Upward Compatible with TMP8049P/TMP80C49P-6/INTEL's 8049
- HALT Instruction (Additional Instruction)
- Low Power
  10mA MAX. in Normal Operation
  (V_{CC}=5V, f_{XTAL}=6MHz)
  10μA MAX. in Power Down Mode
  (V_{CC}=5V, f_{XTAL}:DC)
- Single power supply
- 4K × 8 masked ROM
- 256 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Power Down Mode (Stand-by Mode)
- Halt Mode (Idel Mode)

PIN CONNECTION (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

V_{SS} (Power Supply)
Circuit GND potential

V_{CC} (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 $\overline{WR}$.

T_0 (Input/Output)
Input pin testable using the conditional transfer instructions JT0 and JNT0. T_0 can be designated as a clock output using ENT0 CLK instruction.

T_1 (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 (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 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.

INSTRUCTION SET
Refer to TMP80C49AP/-6 INSTRUCTION SET.

TMP80C50AP/TMP80C40AP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to V_{CC}+0.5V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=70°C)	250mW
T_{SOLDER}	Soldering Temperature (Soldering Time 10 sec)	260°C
T_{STG}	Storage Temperature	-65°C to 150°C
T_{OPR}	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS (T_{OPR}=0°C to 70°C, V_{CC}=+5V±10%, V_{SS}=0V, unless otherwise noted.)

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.8	V
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)			-0.5	-	0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			0.7V_{CC}	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH11}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-1.6mA	2.4	-	-	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC}-0.8	-	-	V
V_{OH21}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-50μA	2.4	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	-50	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current ($\overline{BUS}$, T0) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal operation	V_{CC}=5V, f_{XTAL}=6MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	-	10	mA
I_{CCH1}		HALT Mode		-	-	T.B.D.	
I_{CC2}	V_{CC} Supply Current	Normal operation	V_{CC}=5V, f_{XTAL}=11MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	-	15	μA
I_{CCH2}		HALT Mode		-	-	T.B.D.	

TMP80C50AP/TMP80C40AP ELECTRICAL CHARACTERISTICS

AC CHARACTERISTICS T_{OPR}=0°C to 70°C, V_{CC}=+5V±10%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		150	-	-	ns
t_{AL}	Address Setup Time (ALE)		160	-	-	ns
t_{LA}	Address Hold Time (ALE)		50	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		350	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		390	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	40	-	-	ns
t_{CY}	Cycle Time		1.36	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	110	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	210	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		400	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	570	ns
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		10	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		100	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		160	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	700	ns
t_{DP}	Output Data Setup Time (PROG)		400	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		90	-	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	-	140	ns
t_{PP}	PROG Pulse Width		700	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		160	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		40	-	-	ns

Note: t_{CY}=1.36μs (f_{XTAL}=11MHz)

Control Outputs: C_L=80pF, BUS Outputs: C_L=150pF

TMP80C50AP-6/TMP80C40AP-6 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{INA}	Input Voltage (Except EA)	-0.5V to V_{CC}+0.5V
V_{INB}	Input Voltage (Only EA)	-0.5V to +13V
P_D	Power Dissipation (Ta=85°C)	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

DC CHARACTERISTICS (I)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±10%, V_{SS}=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	-	V_{CC}	V
V_{IH1}	Input High Voltage ($\overline{RESET}$, $\overline{PS}$)			0.7V_{CC}	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH11}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-1.6mA	2.4	-	-	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC}-0.8	-	-	V
V_{OH21}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-50μA	2.4	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	-50	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	-500	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	-	10	mA
I_{CCH1}		HALT Mode		-	-	T.B.D.	

TMP80C50AP-6/TMP80C40AP-6 ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS (II)

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage			-0.5	-	$0.15V_{CC}$	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.5V_{CC}$	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$, $\overline{PS}$)			$0.7V_{CC}$	-	V_{CC}	V
V_{OL}	Output Low Voltage (Except P10-P17, P20-P27)		I_{OL}=1.6mA	-	-	0.45	V
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)		I_{OL}=1.2mA	-	-	0.45	V
V_{OH12}	Output High Voltage (Except P10-P17, P20-P27)		I_{OH}=-400μA	V_{CC} -0.8	-	-	V
V_{OH22}	Output High Voltage (P10-P17, P20-P27)		I_{OH}=-25μA	V_{CC} -0.8	-	-	V
I_{LI}	Input Leak Current (T1, $\overline{INT}$, EA, $\overline{PS}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{SS}$, $\overline{RESET}$)		$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.1}$	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.01}$	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)		V_{SS}+0.45V$\leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC1}	V_{CC} Supply Current	Normal Operation	V_{CC}=5V, f_{XTAL}=6MHz $V_{IH}=V_{CC}$-0.2V, V_{IL}=0.2V	-	-	10	mA
I_{CCH1}		HALT Mode		-	-	T.B.D.	

TMP80C50AP-6/TMP80C40AP-6 ELECTRICAL CHARACTERISTICS

AC CHARACTERISTICS

T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, unless otherwise noted.

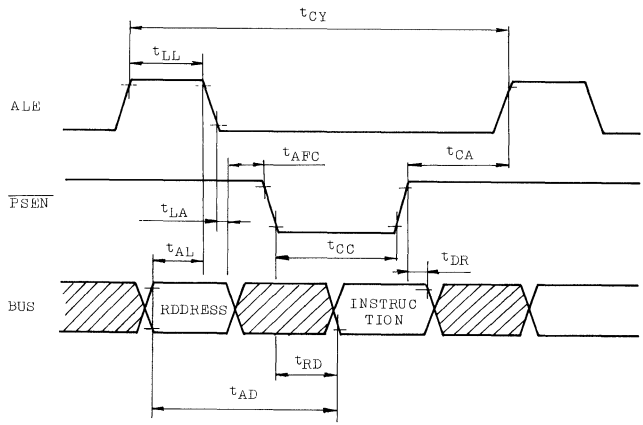
SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{LL}	ALE Pulse Width		400	-	-	ns
t_{AL}	Address Setup Time (ALE)		150	-	-	ns
t_{LA}	Address Hold Time (ALE)		80	-	-	ns
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		700	-	-	ns
t_{DW}	Data Setup Time ($\overline{WR}$)		500	-	-	ns
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	120	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	μs
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	-	500	ns
t_{AW}	Address Setup Time ($\overline{WR}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	950	ns
t_{AFC}	Address Fload Time ($\overline{RD}$, $\overline{PSEN}$)		0	-	-	ns
t_{CP}	Port Control Setup Time (PROG)		110	-	-	ns
t_{PC}	Port Control Hold Time (PROG)		130	-	-	ns
t_{PR}	Port 2 Input Data Set Time (PROG)		-	-	810	ns
t_{DP}	Output Data Setup Time (PROG)		220	-	-	ns
t_{PD}	Output Data Hold Time (PROG)		65	-	-	ns
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	-	150	ns
t_{PP}	PROG Pulse Width		1510	-	-	ns
t_{PL}	Port 2 I/O Data Setup Time		600	-	-	ns
t_{LP}	Port 2 I/O Data Hold Time		150	-	-	ns

Note: t_{CY}=2.5μs (f_{XTAL}=6MHz)

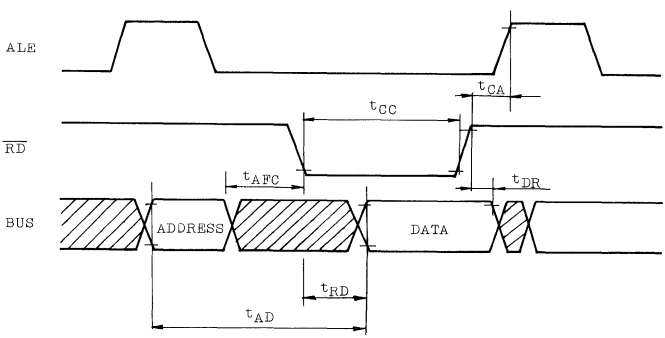
Control Outputs: C_L=80pF, BUS Outputs: C_L=150pF

TIMING WAVEFORM

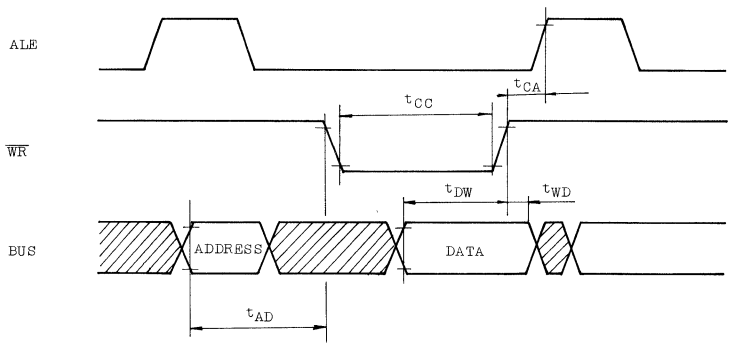
A. Instruction Fetch from External Program Memory



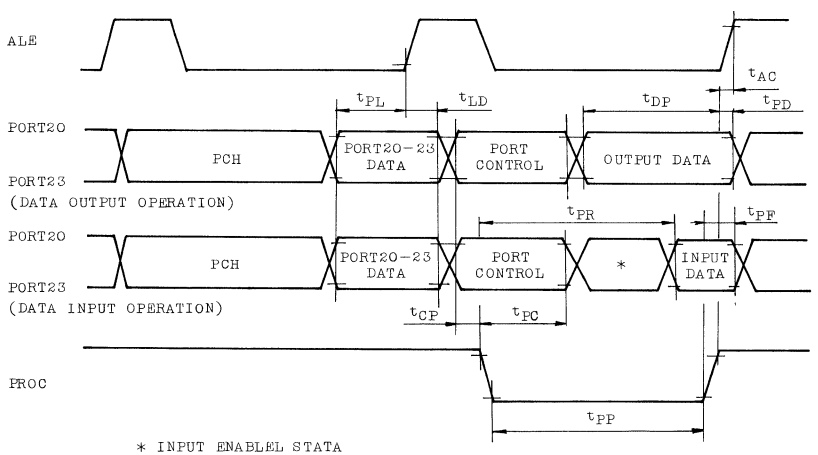
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



* INPUT ENABLEL STATA

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 V_{CC} in this mode is 2V.

$\overline{PS}$ terminal is set to high level to resum oscillation after V_{CC} 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/TMP80C40AP: T_{OPR}=0°C to 70°C, V_{SS}=0V

TMP80C50AP-6/TMP80C40AP-6: T_{OPR}=-40°C to 85°C, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{SB1}	Standby Voltage (1)		2.0	-	6.0	V
I_{SB1}	Standby Current (1)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS

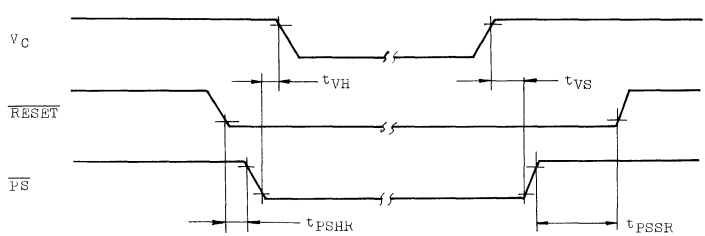
TMP80C50AP/TMP80C40AP: T_{OPR}=0°C to 70°C, V_{CC}=5V±10%, V_{SS}=0V

TMP80C50AP-6/TMP80C40AP-6: T_{OPR}=-40°C to 70°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHR}	Power Save Hold Time ($\overline{RESET}$)		10	-	-	μs
t_{PSSR}	Power Save Setup Time ($\overline{RESET}$)		10	-	-	ms
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μs
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μs

Note: t_{CY}=2.5μs (f_{XTAL}=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 V_{CC} in this mode is 3V.

$\overline{PS}$ terminal is set to high level to resume oscillation after V_{CC} 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/TMP80C40AP: T_{OPR}=0°C to 70°C, V_{SS}=0V

TMP80C50AP-6/TMP80C40AP-6: T_{OPR}=-40°C to 85°C, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{SB2}	Standby Voltage (2)		3.0	-	6.0	V
I_{SB2}	Standby Current (2)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS

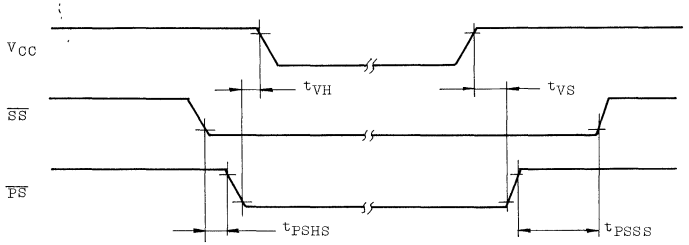
TMP80C50AP/TMP80C40AP: T_{OPR}=0°C to 70°C, V_{CC}=5V±10%, V_{SS}=0V

TMP80C50AP-6/TMP80C40AP-6: T_{OPR}=-40°C to 85°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{PSHS}	Power Save Hold Time ($\overline{SS}$)		10	-	-	μs
t_{PSSS}	Power Save Setup Time ($\overline{SS}$)		10	-	-	ms
t_{VH}	V_{CC} Hold Time ($\overline{PS}$)		5	-	-	μs
t_{VS}	V_{CC} Setup Time ($\overline{PS}$)		5	-	-	μs

Note: t_{CY}=2.5μs (f_{XTAL}=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, TMP80C50AP/-6, TMP80C40AP/-6 enter HALT MODE.

• 3 Status in HALT MODE

The oscillator continues its operation, however, the internal clocks and internal logics are disabled. The status of all 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. TMP80C50AP/-6, TMP80C40AP/-6 start the program at address "000H".

(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), TMP80C50AP/-6, TMP80C40AP/-6 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), TMP80C50AP/-6, TMP80C40AP/-6 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)

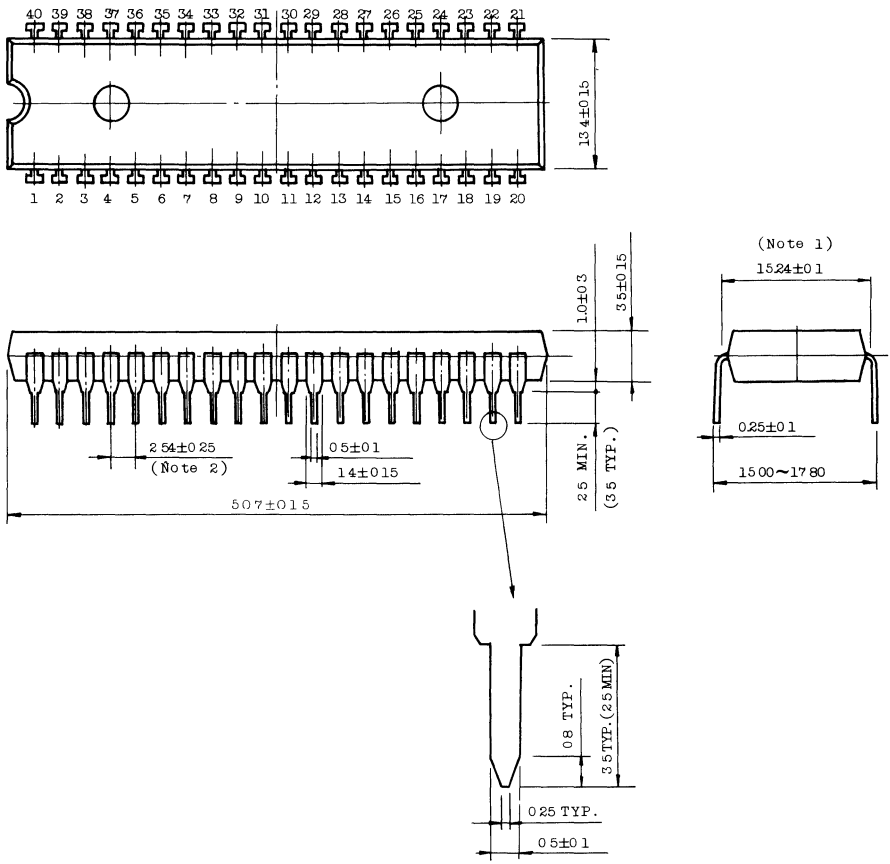
PIN NAME	STATUS
DB0 - DB7	High impedance Input disabled
P10 - P17	
P20 - P27	
T0	High impedance, input disabled
T1	Input disabled
XTAL1	High impedance
XTAL2	Output "High" Level
$\overline{\text{RESET}}$, $\overline{\text{SS}}$	Input disabled when oscillator is stopped. Pull-up transistors turn off.
$\overline{\text{INT}}$, EA	Input disabled when oscillator 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	Values prior to the execution of HALT INSTRUCTION are maintained.
P10 - P17	
P20 - P27	
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

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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRAED CIRCUIT

TMP82C43P

SILICON MONOLITHIC CMOS SILICON GATE

INPUT/OUTPUT EXPANDER

GENERAL DESCRIPTION

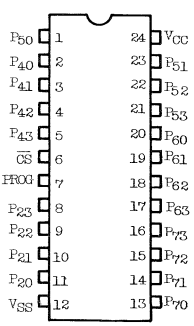
The TMP82C43P is an input/output expander designed specifically to provide a low cost means of I/O expansion for the TLCS-84C family.

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.

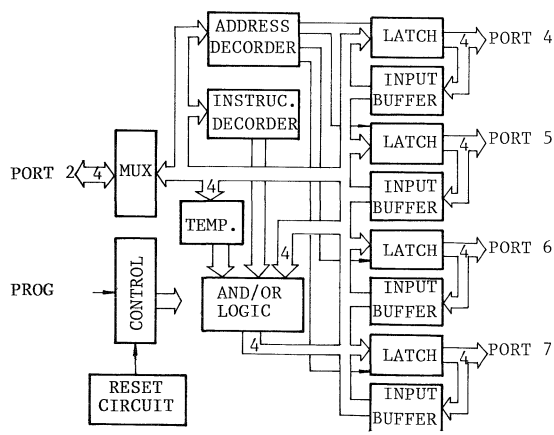
FEATURES

- 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

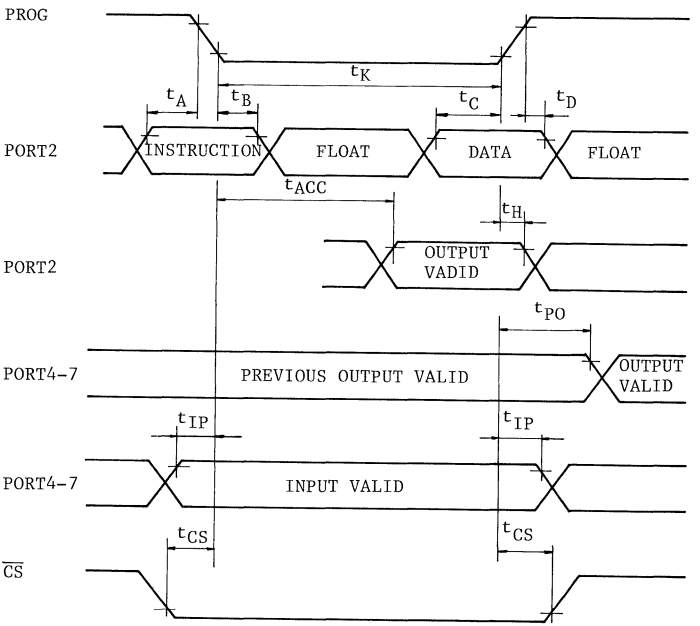
D.C. CHARACTERISTICS (II) T_{OPR}=-40°C to 85°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	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

A.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 $\frac{\text{Before}}{\text{After}}$ PROG		100			ns
t_{ACC}	Port 2 Valid After PROG	C_L=80pF			650	ns

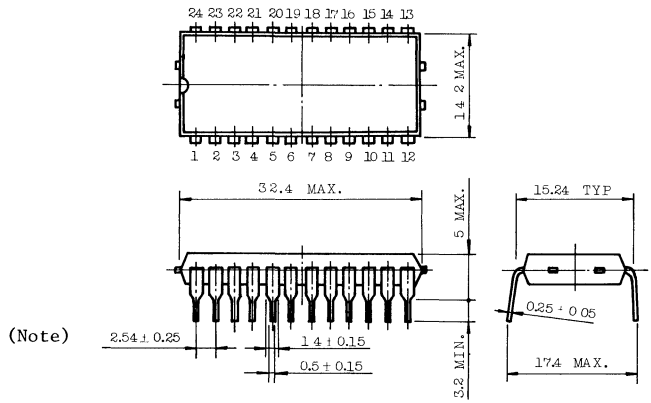
TIMING WAVEFORM



OUTLINE DRAWINGS

PLASTIC PACKAGE

Unit in mm



Note : 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.

POSTSCRIPT

This manual is a reference for the customer applying the TLCS-48 series. It contains the functions and specifications of each LSI device of the TLCS-48. The examples of application circuits described here are shown as the reference for explanation. Toshiba assumes no responsibility for any problems caused by using these examples. The information herein is subject to change without prior notice.

This manual has been prepared by the following section.

Microcomputer LSI Application Engineering Section
Integrated Circuit Division, Toshiba Corporation

1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 210, Japan
Phone: Japan (81)44-511-3111

PART 4

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-Z80

LSI DEVICES

July. 1 9 8 4

PREFACE

This Part desceibes the detail functions and specefications of the LSI deveces of the TLCS-Z80 microprocessor family. The TLCS-Z80 fammily contains the Z80 MPU and associated peripheral devices which are fabricated by a new silicon-gate CMOS technology. These CMOS devices have conpatible architecture, speed and pin configuration with NMOS Z80 family. The CMOS Z80 family have superior performance such as lower power consumption and wide operationg temperature range than NMOS family.
Besides, the TLCS-Z80 family has a new clock generator/Controller device which makes the MPU peripheral ciruit simpler and compact.
The complete devilopment systems have already been prepared for the TLCS-Z80 aplications.

CONTENTS

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMPZ84C00P-3/TMPZ84C00P

SILICON MONOLITHIC

CMOS SILICON GATE

PRELIMINARY

CMOS Z80® 8-BIT MICROPROCESSOR

GENERAL DESCRIPTION

The TMPZ84C00P-3/TMPZ84C00P (from here on referred to as Z80 or CPU) is CMOS version of Z80 CPU which provides low power operation and high performance.

Built into the CMOS Z80 microprocessor are all bus control, memory control, and timing signals in addition to eight general purpose 16-bit registers and an arithmetic-and-logic unit. The CMOS Z80 is fabricated using Toshiba's C^2MOS Silicon Gate Technology.

FEATURES

- Software Compatible with the Zilog Z80 CPU
- DC to 4MHz Operation (TMPZ84C00P)
  DC to 2.5MHz Operation (TMPZ84C00P-3)
- Single 5V Power Supply
  4MHz @ 5V±10% (TMPZ84C00P)
  2.5MHz @ 5V±10% (TMPZ84C00P-3)
- Powerful Set of 158 Instructions
- Duplicate Sets of Both General-purpose and Flag Registers
- Two Interrupt Inputs
  - Non-maskable Interrupt (NMI)
  - 3 Modes of Maskable Interrupt (INT)
    - 8080 Compatible (Non-Z80 Peripheral Device) (Mode 0)
    - Restart (Mode 1)
    - Z80 Family Peripheral with Daisy Chain (Mode 2)
- Low Power Consumption
  15mA Typ. @ 4MHz @ 5V (TMPZ84C00P), 9mA Typ. @ 2.5MHz @ 5V (TMPZ84C00P-3)
  Less than 10μA @ 5V (Power down)
- Extended Operating Temperature
  - 40°C to 85°C

Z80® is a trademark of Zilog Inc.

PRELIMINARY

- Two Indexed Registers
- 10 Addressing Modes
- On-chip Dynamic Memory Refresh Counter

PIN CONNECTIONS (TOP VIEW)

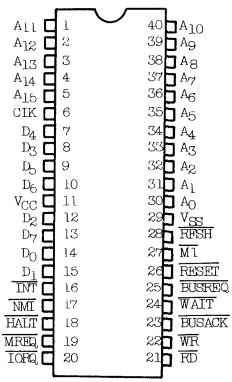


FIGURE 1. Z80 PINOUT DIAGRAM

BLOCK DIAGRAM

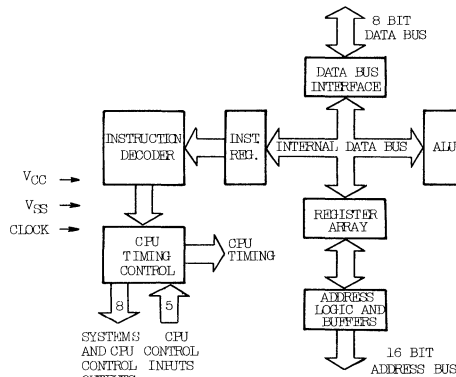


FIGURE 2. Z80 BLOCK DIAGRAM

PRELIMINARY

PIN NAMES AND PIN DESCRIPTION

A_0-A_{15}. Address Bus (output, active High, 3-state)

A_0-A_{15} form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

$\overline{BUSACK}$. Bus Acknowledge (output, active Low)

Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ have entered their high impedance states. The external circuitry can now control these lines.

$\overline{BUSREQ}$. Bus Request (input, active Low)

Bus Request has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle. $\overline{BUSREQ}$ forces the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ to go to a high-impedance state so that other devices can control these lines. $\overline{BUSREQ}$ is normally wire-ORed and requires an external pullup for these applications. Extended $\overline{BUSREQ}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

D_0-D_7. Data Bus (input/output, active High, 3-state)

D_0-D_7 constitute an 8-bit bidirectional data bus, used for data exchange with memory and I/O.

$\overline{HALT}$. Halt State (output, active Low)

HALT indicates that the CPU has executed a Halt instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

$\overline{INT}$. Interrupt Request (input, active Low)

Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. $\overline{INT}$ is normally wire-ORed and requires an external pullup for these applications.

$\overline{IORQ}$. Input/Output Request (output, active Low, 3-state)

$\overline{IORQ}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{IORQ}$ is also generated concurrently with $\overline{M1}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

$\overline{M1}$. Machine Cycle One (output, active Low)

$\overline{M1}$, together with $\overline{MREQ}$, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes always begin with CB_H, DD_H, ED_H or FD_H. $\overline{M1}$ also occurs with $\overline{IORQ}$ to indicate an interrupt acknowledge cycle.

$\overline{MREQ}$. Memory Request (output, active Low, 3-state)

$\overline{MREQ}$ indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{NMI}$. Non-Maskable Interrupt (input, active Low)

$\overline{NMI}$ has a higher priority than $\overline{INT}$. $\overline{NMI}$ is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

$\overline{RD}$. Memory Read (output, active Low, 3-state)

$\overline{RD}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{RESET}$. Reset (input, active Low)

$\overline{RESET}$ initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that $\overline{RESET}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

$\overline{RFSH}$. Refresh (output, active Low)

$\overline{RFSH}$ together with $\overline{MREQ}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

$\overline{WAIT}$. Wait (input, active Low)

$\overline{WAIT}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended $\overline{WAIT}$ periods can prevent the CPU from refreshing dynamic memory properly.

$\overline{WR}$. Memory Write (output, active Low, 3-state)

$\overline{WR}$ indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

CLK. Clock (input)

Single phase system clock input. When CLK is a DC state (either a high or low level), CPU stops its operation and maintains resisters and control signals.

V_{CC}. Power Supply

+5V

V_{SS}. Power Supply

Ground reference (0V).

FUNCTIONAL DESCRIPTION

CPU REGISTERS

The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

PRELIMINARY

The Z80 also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register.

Figure 3 shows the registers within the Z80 CPU.

Table 1 provides further information on these resisters.

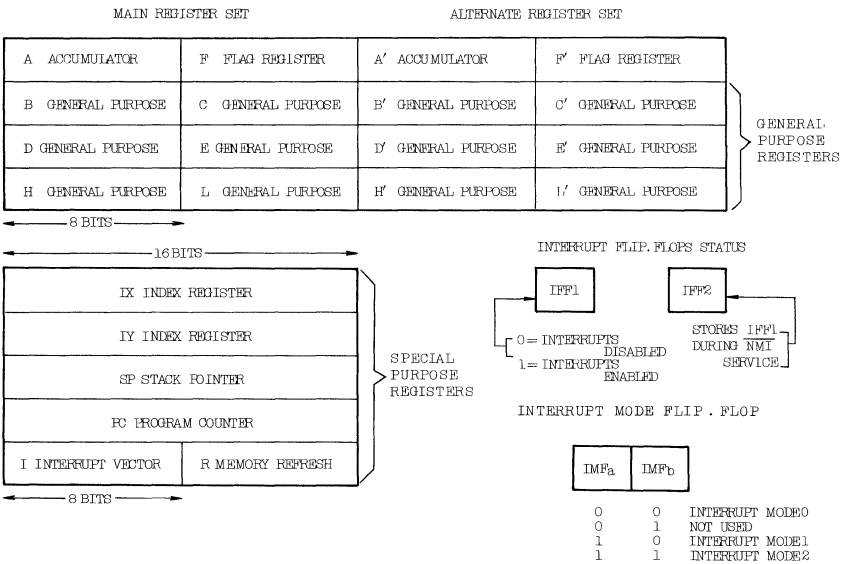


FIGURE 3. CPU REGISTERS

PRELIMINARY

Register		Size (Bits)	Remarks
A, A'	Accumulator	8	Stores an operand or the results of an operation.
F, F'	Flags	8	See Instruction Set.
B, B'	General Purpose	8	Can be used separately or as a 16-bit register with C.
C, C'	General Purpose	8	See B, above.
D, D'	General Purpose	8	Can be used separately or as a 16-bit register with E.
E, E'	General Purpose	8	See D, above.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	See H, above. Note: The (B,C), (D,E), and (H,L) sets are combined as follows: B - High byte C - Low byte D - High byte E - Low byte H - High byte L - Low byte
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Same as IX, above.
SP	Stack Pointer	16	Stores addresses or data temporarily. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction.
IFF_1-IFF_2	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 3).
IMFa-IMFb	Interrupt Mode	Flip-Flops	Reflect Interrupt mode (see Figure 3).

TABLE 1. Z80 CPU REGISTERS

(1) Special Purpose Registers

• Program Counter (PC)

The program counter is 16-bit counter and holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new valve is automatically placed in the PC, overriding the incrementer.

• Stack Pointer (SP)

The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack from specific CPU registers or popped off of the stack into specific CPU registers through the execution of PUSH and POP instructions. The data popped from the stack is always the last data pushed onto it. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.

• Two Index Registers (IX & IY)

The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's

complement signed integer. This mode of addressing greatly simplifies many types of programs, especially where tables of data are used.

- Interrupt Page Address Register (I)

The Z80CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I Register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrup routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.

- Memory Refresh Register (R)

The Z80CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. Seven bits of this 8-bit register are automatically incremented after each instruction fetch. The eighth bit will remain as programmed as the result of an LD R, A instruction. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer. During refresh, the contents of the I register are placed on the upper 8 bits of the address bus.

(2) Accumulator and Flag Registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8 or 16-bit operations, such as indicating whether or not the result of an operation is equal to zero. The programmer selects the accumulator and flag pair that he wishes to work with a single exchange instruction so that he may easily work with either pair.

(3) General Purpose Registers

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or as 16-bit register pairs by the programmer. One set is called BC, DE and HL while the complementary set is called BC', DE' and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator-flag register may be reserved for handling this very fast routine. Only a simple exchange commands need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

ARITHMETIC & LOGIC UNIT (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus.

The type of functions performed by the ALU include:

Add	Left or right shifts or rotates (arithmetic and logical)
Subtract	Increment
Logical AND	Decrement
Logical OR	Set bit
Logical Exclusive OR	Reset bit
Compare	Test bit

INSTRUCTION REGISTER AND CPU CONTROL

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control section performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, controls the ALU and provides all required external control signals.

FLAGS

Each of the two Z80 CPU Flag registers contains six bits of information which are set or reset by various CPU instructions. Four of these bits are testable; that is, they are used as conditions for jump, call or return instructions. The four testable flag bits are:

1) Carry Flag (C) - This flag is the carry from the highest order bit of the accumulator. For example, the carry flag will be set during an add instruction where a carry from the highest bit of the accumulator is generated. This flag is also set if a borrow is generated during a subtraction instruction. The shift and rotate instructions also affect this bit.

2) Zero Flag (Z) - This flag is set if the result of the operation loaded a zero into the accumulator. Otherwise it is reset.

3) Sign Flag (S) - This flag is intended to be used with signed numbers and it is set if the result of the operation was negative. Since bit 7 (MSB) represents the sign of the number (A negative number has a 1 in bit 7), this flag stores the state of bit 7 in the accumulator.

4) Parity/Overflow Flag (P/V) - This dual purpose flag indicates the parity of the result in the accumulator when logical operations are performed (such as AND A, B) and it represents overflow when signed two's complement arithmetic operations are perfomed. The Z80 overflow

flag indicates that the two's complement number in the accumulator is in error since it has exceeded the maximum possible (+127) or is less than the minimum possible (-128) number that can be represented two's complement notation.

There are also two non-testable bits in the flag register. Both of these are used for BCD arithmetic.

1) Half carry (H) = This is the BCD carry or borrow result from the least significant four bits of operation. When using the DAA (Decimal Adjust Instruction) this flag is used to correct the result of a previous packed decimal add or subtract.

2) Add/Subtract Flag (N) - Since the agorithim for correcting BCD operations is different for addition or subtraction, this flag is used to specify what type of instruction was executed last so that the DAA operation will be correct for either addition or subtraction.

The Flag register can be accessed by the programmer and its format is as follows:

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	H	X	P/V	N	C

X means flag is indeterminate.

The Table 2 lists how each flag is affected by various CPU instructions.

- '_' indicates that the instruction does not change the flag.
- 'X' means that the flag goes to an indeterminate state.
- 'R' means that it is reset.
- 'S' means that it is set.
- 'O' indicates that it is set or reset according to the previous discussion.

Note) Any instruction not appearing in the table 2 does not affect any of the flags.

Table 2 includes a few special cases that must be described for clarity. Notice that the block search instruction sets the Z flag if the last compare operation indicated a match between the source and the accumulator data. Also, the parity flag is set if the byte counter (register pair BC) is not equal to zero. This same use of the parity flag is made with the block move instructions. Another special case is during block input or output instructions, here the Z flag is used to indicate the state of register B which is used as a byte counter. Notice that when the I/O block transfer is complete, the zero flag will be reset to a zero (i.e. B=0) while in the case of a block move command the parity flag is reset when the operation is complete. A final case is when the refresh or I register is loaded into the accumulator, the interrupt enable flip flop is loaded into the parity flag so that the complete state of the CPU can be saved at any time.

PRELIMINARY

Instruction	D7 S	D6 Z	D5	D4 H	D3	D2 P/V	D1 N	D0 C	Comments
ADD A,s;ADC A,s	o	o	x	o	x	V	R	o	8-bit add or add with carry
SUB s; SBC A,s; CP s; NEG	o	o	x	o	x	V	S	o	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	o	o	x	S	x	P	R	R	Logical operations
OR s; XOR s	o	o	x	R	x	P	R	R	Logical operations
INC s	o	o	x	o	x	V	R	-	8-bit increment
DEC s	o	o	x	o	x	V	S	-	8-bit decrement
ADD DD, SS	-	-	x	x	x	-	R	o	16-bit add
ADC HL, SS	o	o	x	x	x	V	R	o	16-bit add with carry
SBC HL, SS	o	o	x	x	x	V	S	o	16-bit subtract with carry
RLA; RLCA; RRA; RRCA	-	-	x	R	x	-	R	o	Rotate accumulator
RL s; RLC s; RR s; RRC s; SLA s; SRA s; SRL s	o	o	x	R	x	P	R	o	Rotate and shift locations
RLD; RRD	o	o	x	R	x	P	R	-	Rotate digit left and right
DAA	o	o	x	o	x	P	-	o	Decimal adjust accumulator
CPL	-	-	x	S	x	-	S	-	Complement accumulator
SCF	-	-	x	R	x	-	R	S	Set carry
CCF	-	-	x	x	x	-	R	o	Complement carry
IN r, (C)	o	o	x	R	x	P	R	-	Input register indirect
INI; IND; OUTI; OUTD	x	o	x	x	x	x	S	x	Block input and output
INIR; INDR; OTIR; OTDR	x	S	x	x	x	x	S	x	Z=0 if B≠0 otherwise Z=1
LDI; LDD	x	x	x	R	x	o	R	-	Block transfer instructions
LDIR; LDDR	x	x	x	R	x	R	R	-	P/V=1 if BC≠0, otherwise P/V=0

PRELIMINARY

Instruction	D7 S	D6 Z	D5	D4 H	D3	D2 P/V	D1 N	D0 C	Comments
CPI; CPIR; CPD; CPDR	o	o	x	o	x	o	S	-	Block search instructions Z=1 if A=(HL), otherwise Z=0 P/V=1 if BC≠0, otherwise P/V=0
LD A, I; LD A, R	o	o	x	R	x	IFF	R	-	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag.
BIT b, s	x	o	x	S	x	x	R	-	The state of bit b of location s is copied into the Z flag

TABLE 2. SUMMARY OF FLAG OPERATION

The following notation is used in this table:

SYMBOL	OPERATION
C	Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
Z	Zero flag. Z=1 if the result of the operation is zero.
S	Sign flag. S=1 if the MSB of the result is one.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
H	Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.

SYMBOL	OPERATION
N	Add/Subtract flag. N=1 if the previous operation was a subtract. H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format. The flag is affected according to the result of the operation.
-	The flag is unchanged by the operation.
R	The flag is reset by the operation.
S	The flag is set by the operation.
o	The flag is affected according to the result of the operation.
x	The flag is a "don't care".
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
I	I resister
R	Refresh counter.
n	8-bit value in range <0, 255>
nn	16-bit value in range <0, 65535>

INTERRUPT

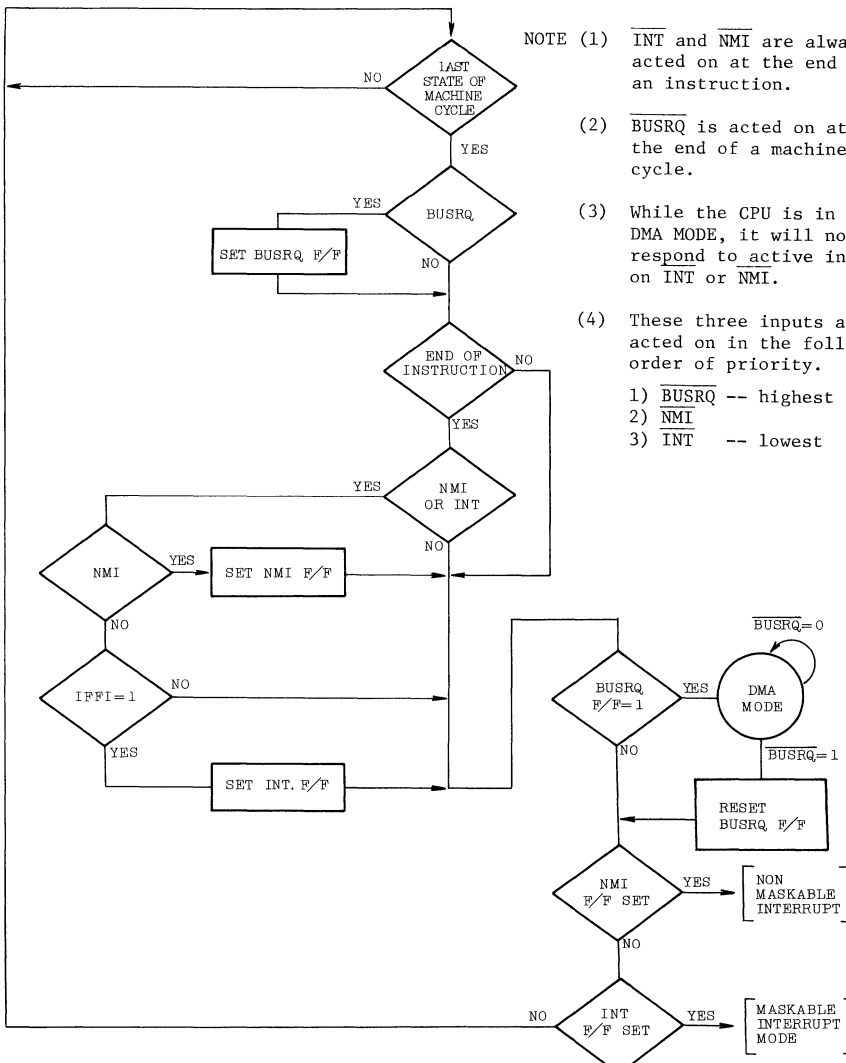
The CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt since it requires that interrupts be enabled in software in order to operate. Either $\overline{NMI}$ or $\overline{INT}$ can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, $\overline{INT}$, has three programmable response modes available.
These are:

- Mode 0 — compatible with the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.
- Mode 2 — a vectored interrupt scheme, usually daisy-chained, for use with Z80 family and compatible peripheral devices.

Both the $\overline{INT}$ and $\overline{NMI}$ inputs are sampled by the CPU on the rising edge of CLK in the last T state of the last Machine (M) cycle of any instruction. However, if $\overline{BUSRQ}$ is active at the same time, it will be processed before any interrupts. Figure 4 illustrates the Z80 interrupt service sequence.

PRELIMINARY



NOTE (1) $\overline{INT}$ and $\overline{NMI}$ are always acted on at the end of an instruction.

(2) $\overline{BUSRQ}$ is acted on at the end of a machine cycle.

(3) While the CPU is in the DMA MODE, it will not respond to active inputs on $\overline{INT}$ or $\overline{NMI}$.

(4) These three inputs are acted on in the following order of priority.

1) $\overline{BUSRQ}$ -- highest
2) $\overline{NMI}$
3) $\overline{INT}$ -- lowest

FIGURE 4. Z80 CPU INTERRUPT SEQUENCE

(1) Non-Maskable Interrupt ($\overline{NMI}$)

The non-maskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

(2) Maskable Interrupt ($\overline{INT}$)

Regardless of the interrupt mode set by the user, the Z80 CPU response to a maskable interrupt input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in a normal $\overline{M1}$ cycle. In addition, this special $\overline{M1}$ cycle is automatically extended by two $\overline{WAIT}$ states, to allow for the time required to acknowledge the interrupt request and to place the interrupt vector on the bus.

• Mode 0 Interrupt Operation

This mode is compatible with the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus.

This is normally a Restart Instruction, which will initiate an unconditional jump to the selected one of eight restart locations in page zero of memory.

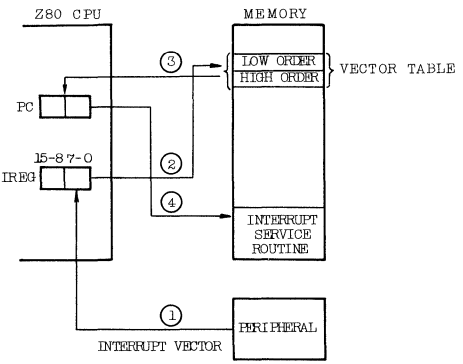
- Mode 1 Interrupt Operation

Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has a vector address of 0038H only.

- Mode 2 Interrupt Operation

This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit address vector on the data bus during the interrupt acknolwedge cycle. The high-order byte of the interrupt service routine address is supplied by the I (Interrupt) register. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines.
These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.
Figure 5 illustrates the vector processing sequence.

PRELIMINARY



NOTES:

1) Interrupt vector generated by peripheral is read by CPU during interrupt acknowledge cycle.

2) Vector combined with I register contents form 16-bit memory address pointing to vector table.

3) Two bytes are read sequentially from vector table. These 2 bytes are read into PC.

4) Processor control is transferred to interrupt service routine and execution continues.

FIGURE 5. VECTOR PROCESSING SEQUENCE

(3) Interrupt Priority (Daisy Chaining and Nested Interrupts).

The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

(4) Interrupt Enable/Disable Operation.

In the Z80-CPU there is an enable flip flop (called IFF) that is set or reset by the programmer using the Enable Interrupt (EI) and Disable Interrupt (DI) instructions. When the IFF is reset, an interrupt (except $\overline{NMI}$) cannot be accepted by the CPU.

Actually, there are two enable flip flops, called IFF_1 and IFF_2.

IFF_1	IFF_2
Actually disables interrupts from being accepted.	Temporary storage location for IFF_1.

A reset to the CPU will force both IFF_1 and IFF_2 to the reset state so that interrupts are disabled. They can then be enabled by an EI instruction at any time by the programmer. When an EI instruction is executed, any pending interrupt request will not be accepted until after the instruction following EI has been executed. This single instruction delay is necessary for cases when the following instruction is a return instruction and interrupts must not be allowed until the return has been completed. Both IFF_1 and IFF_2 can be enabled by execution of the EI instruction. When an interrupt is accepted by the CPU, both IFF_1 and IFF_2 are automatically reset, inhibiting further interrupts until a new EI instruction is executed. Note that for all of the previous cases, IFF_1 and IFF_2 are always equal.

The purpose of IFF_2 is to save the status of IFF_1 when a non-maskable interrupt occurs. When a non-maskable interrupt is accepted, IFF_1 is reset to prevent further interrupts until reenable by the programmer. Thus, after a non-maskable interrupt has been accepted maskable interrupts are disabled but the previous state of IFF_1 has been saved so that the complete state of the CPU just prior to the non-maskable interrupt can be restored at any time. When a Load Register A with Register I (LD A, I) instruction or a Load Register A with Register R (LD A, R) instruction is executed, the state of IFF_2 is

copied into the parity flag where it can be tested or stored.

A second method of restoring the status of IFF_1 is thru the execution of a Return From Non-Maskable Interrupt (RETN) instruction. Since this instruction indicates that the non maskable interrupt service routine is complete, the contents of IFF_2 are now copied back into IFF_1, so that the status of IFF_1 just prior to the acceptance of the non-maskable interrupt will be restored automatically.

Operation of the two flip-flops is described in Table 3.

Action	IFF_1	IFF_2	Comments
CPU Reset	0	0	Maskable interrupt $\overline{INT}$ disabled
DI instruction execution	0	0	Maskable interrupt $\overline{INT}$ disabled
EI instruction execution	1	1	Maskable interrupt $\overline{INT}$ enabled
LD A,I instruction execution	-	-	$IFF_2 \rightarrow$ Parity flag
LD A,R instruction execution	-	-	$IFF_2 \rightarrow$ Parity flag
Accept $\overline{NMI}$	0	IFF_1	$IFF_1 \rightarrow IFF_2$ (Maskable interrupt $\overline{INT}$ disabled)
RETN instruction execution	IFF_2	-	$IFF_2 \rightarrow IFF_1$ at completion of an $\overline{NMI}$ service routine.
Accept $\overline{INT}$	0	0	
RETI	-	-	

Note) "-" indicates no change.

TABLE 3. STATE OF FLIP-FLOPS

CPU TIMING

The Z80 CPU executes instructions by proceeding through a specific sequence of operations. These include:

Memory read or write

I/O device read or write

Interrupt acknowledge

All instructions are merely a series of these basic operations. Each of these basic operations can take from three to six clock periods to complete or they can be lengthened to synchronize the CPU to the speed of external devices. The basic clock periods are referred to as T states and the basic operations are referred to as M (for machine) cycles. Figure 6 illustrates how a typical instruction will be merely a series of specific M and T cycles. Notice that this instruction consists of three machine cycles (M1, M2 and M3). The first machine cycle of any instruction is a fetch cycle which is four, five or six T states long (unless lengthened by the wait signal). The fetch cycle (M1) is used to fetch the OP code of the next instruction to be executed. Subsequent machine cycles move data between the CPU and memory or I/O devices and they may have anywhere from three to five T cycles (again they may be lengthened by wait states to synchronize the external devices to the CPU). The following paragraphs describe the timing which occurs within any of the basic machine cycles.

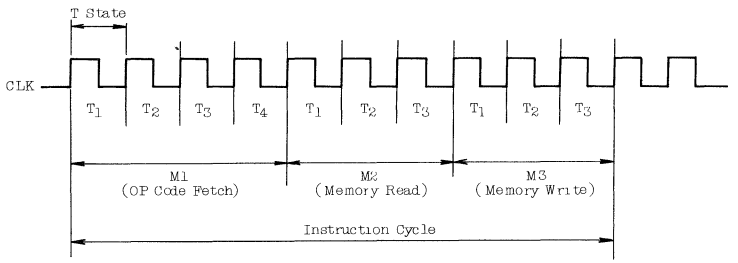


FIGURE 6. BASIC CPU TIMING EXAMPLE

All CPU timing can be broken down into some very simple timing diagrams as shown in Figure 7 through 14. These diagrams show the following basic operations with and without wait states (wait states are added to synchronize the CPU to slow memory or I/O devices).

Fig. 7 Instruction OP code fetch (M1 cycle)

Fig. 8 Memory data read or write cycles

Fig. 9 I/O read or write cycles

Fig. 10 Bus Request/Acknowledge Cycle

Fig. 11 Interrupt Request/Acknowledge Cycle

Fig. 12 Non maskable Interrupt Request/Acknowledge Cycle

Fig. 13 Exit from a HALT instruction

Fig. 14 Reset Cycle

(1) Instruction fetch

Figure 7-0 shows the timing during an M1 cycle (OP code fetch). Notice that the PC is placed on the address bus at the beginning of the M1 cycle. One halt clock time later the $\overline{MREQ}$ signal goes active. At this time the address to the memory has had time to stabilize so that the falling edge of $\overline{MREQ}$ can be used directly as a chip enable clock to dynamic memories. The $\overline{RD}$ line also goes active to indicate that the memory read data should be enabled onto the CPU data bus. The CPU samples the data from the memory on the data bus with the rising edge of the clock of state T3 and this same edge is used by the CPU to turn off the $\overline{RD}$ and $\overline{MREQ}$ signals. Thus the data has already been sampled by the CPU before the $\overline{RD}$ signal becomes inactive. Clock state T3 and T4 of a fetch cycle are used to refresh dynamic memories. (The CPU uses this time to decode and execute the fetched instruction so that no other operation could be performed at this time). During T3 and T4 the lower 7-bits of the address bus contain a memory refresh address and the $\overline{RFSH}$ signal becomes active to indicate that a refresh read of all dynamic memories should be accomplished. Notice that a $\overline{RD}$ signal is not generated during refresh time to prevent data from different memory segments from being gated onto the data bus. The $\overline{MREQ}$ signal during refresh time should be used to perform a refresh read of all memory elements. The refresh signal cannot be used by itself since the refresh address is only guaranteed to be stable during $\overline{MREQ}$ time.

PRELIMINARY

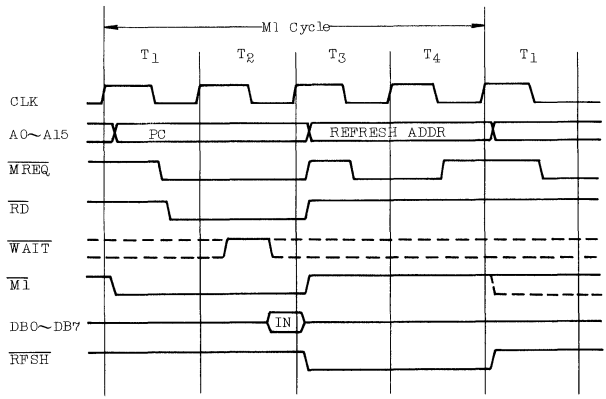


FIGURE 7-0. INSTRUCTION OP CODE FETCH

Figure 7-1 illustrates how the fetch cycle is delayed if the memory activates the $\overline{WAIT}$ line. During T2 and every subsequent Tw, the CPU samples the $\overline{WAIT}$ line with the falling edge of CLK. If the $\overline{WAIT}$ line is active at this time, another wait state will be entered during the following cycle. Using this technique the read cycle can be lengthened to match the access time of any type of memory device.

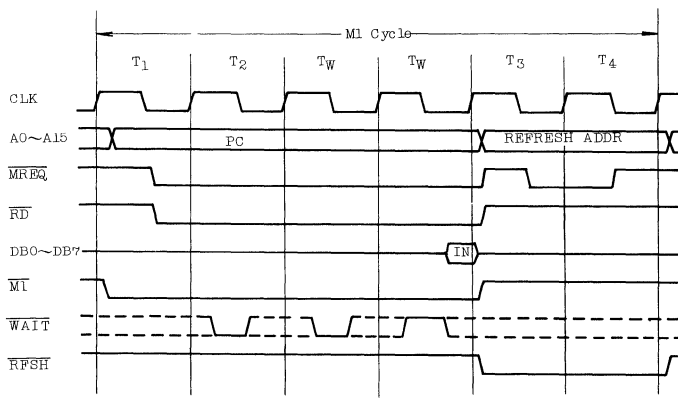


FIGURE 7-1. INSTRUCTION OP CODE FETCH WITH WAIT STATES

(2) Memory read or write

Figure 8-0 illustrates the timing of memory read or write cycles other than an OP code fetch (M1 cycle). These cycles are generally three clock periods long unless wait states are requested by the memory via the $\overline{WAIT}$ signal. The $\overline{MREQ}$ signal and the $\overline{RD}$ signal are used the same as in the fetch cycle. In the case of a memory write cycle, the $\overline{MREQ}$ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The $\overline{WR}$ line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory.

Furthermore the $\overline{WR}$ signal goes inactive one halt T state before the address and data bus contents are changed so that the overlap requirements for virtually any type of semiconductor memory type will be met.

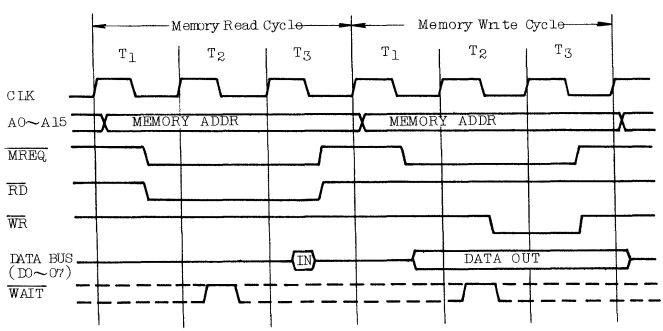


FIGURE 8-0. MEMORY READ OR WRITE CYCLES

Figure 8-1 illustrates how a $\overline{WAIT}$ request signal will lengthen any memory read or write operation. This operation is identical to that previously described for a fetch cycle. Notice in this figure that a separate read and a separate write cycle are shown in the same figure although read and write cycles can never occur simultaneously.

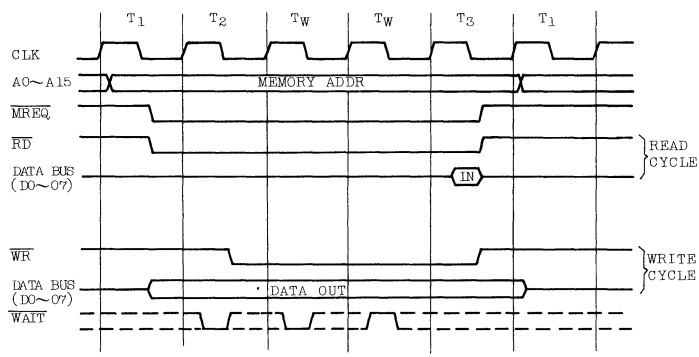


FIGURE 8-1. MEMORY READ OR WRITE CYCLES WITH WAIT STATES

(3) Input or output cycles

Figure 9-0 illustrates an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted. The reason for this is that during I/O operations, the time from when the $\overline{IORQ}$ signal goes active until the CPU must sample the $\overline{WAIT}$ line is very short and without this extra state sufficient time does not exist for an I/O port to decode its address and activate the $\overline{WAIT}$ line if a wait is required. Also, without this wait state it is difficult to design MOS I/O devices that can operate at full CPU speed. During this wait state time the $\overline{WAIT}$ request signal is sampled. During a read I/O operation, the RD line is used to enable the addressed port onto the data bus just as in the case of a memory read. For I/O write operations, the $\overline{WR}$ line is used as a clock to the I/O port, again with sufficient overlap timing automatically provided so that the rising edge may be used as a data clock.

Figure 9-1 illustrates how additional wait states may be added with the $\overline{WAIT}$ line. The operation is identical to that previously described.

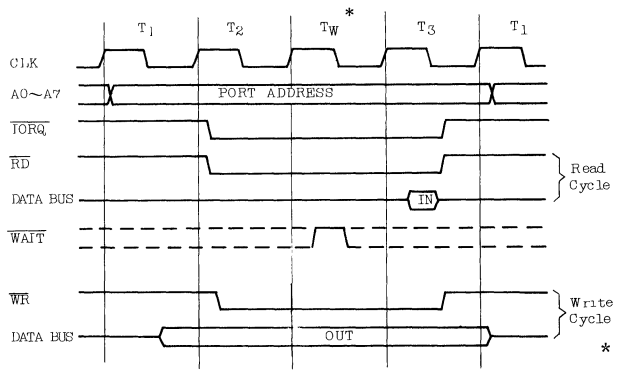


FIGURE 9-0. INPUT OR OUTPUT CYCLES

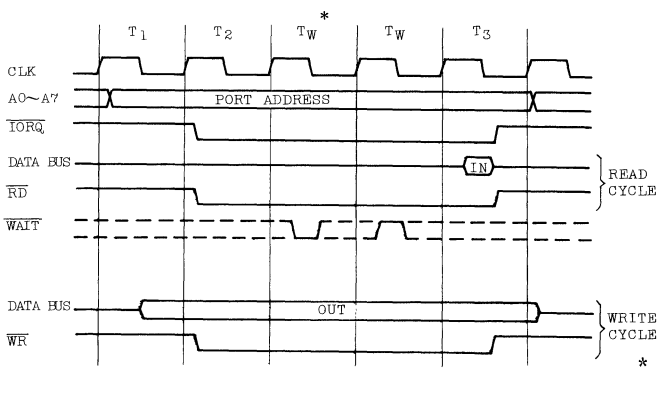


FIGURE 9-1. INPUT OR OUTPUT CYCLES WITH WAIT STATES

(4) Bus request/acknowledge cycle

Figure 10 illustrates the timing for a Bus Request/Acknowledge cycle. The $\overline{BUSRQ}$ signal is sampled by the CPU with the rising edge of the last clock period of any machine cycle. If the $\overline{BUSRQ}$ signal is active, the CPU will set its address, data and tri-state control signals to the high impedance state with the rising edge of the next clock pulse. At that time any external device can control the buses to transfer data between memory and I/O devices. (This is generally known as Direct Memory Access [DMA] using cycle stealing).

The maxunyn time for the CPU to respond to a bus request is the length of a machine cycle and the external controller can maintain control of the bus for as many clock cycles as is desired.
Note, however, that if very long DMA cycles are used, and dynamic memories are being used, the external controller must also perform the refresh function. This situation only occurs if very large blocks of data are transferred under DMA control. Also note that during a bus request cycle, the CPU cannot be interrupted by either a $\overline{NMI}$ or an $\overline{INT}$ signal.

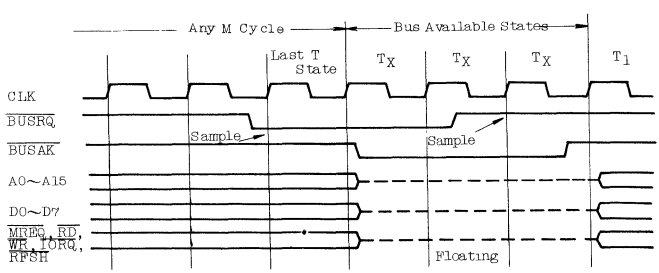


FIGURE 10. BUS REQUEST/ACKNOWLEDGE CYCLE

(5) Interrupt request/acknowledge cycle

Figure 11-0 illustrates the timing associated with an interrupt cycle. The interrupt signal ($\overline{INT}$) is sampled by the CPU with the rising edge of the last clock at the end of any instruction. The signal will not be accepted if the internal CPU software controlled interrupt enable flip-flop is not set or if the $\overline{BUSRQ}$ signal is active. When the signal is accepted a special M1 cycle is generated. During this special M1 cycle the $\overline{IORQ}$ signal becomes active (instead of the normal $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. Notice that two wait states are automatically added to this cycle. These states are added so that a ripple priority interrupt scheme can be easily implemented. The two wait states allow sufficient time for the ripple signals to stabilize and identify which I/O device must insert the response vector.

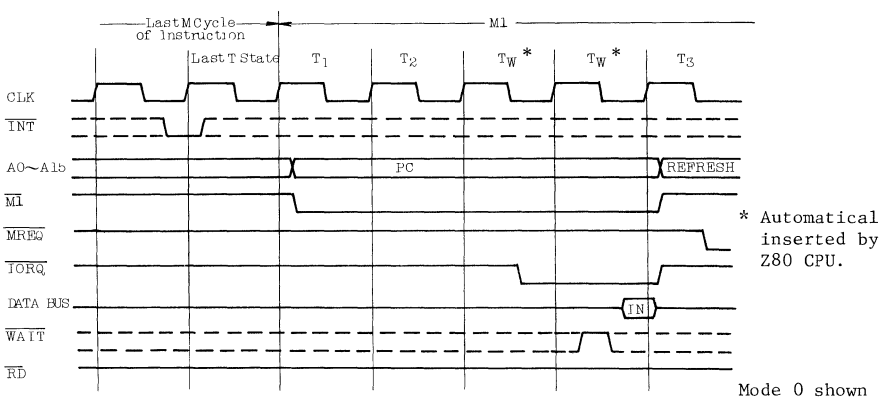
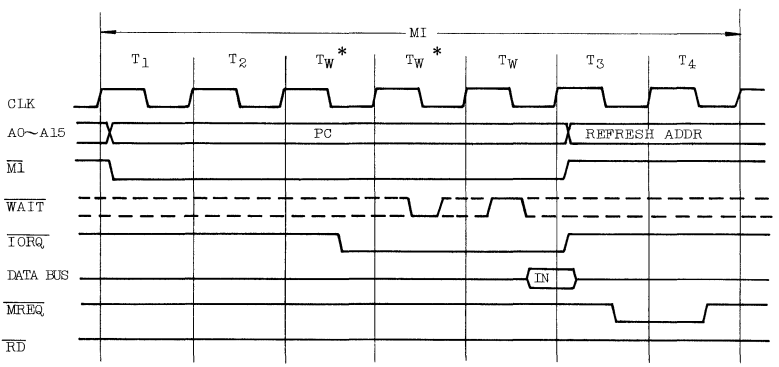


FIGURE 11-0. INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

Figure 11-1 illustrates how additional wait states can be added to the interrupt response cycle. Again the operation is identical to that previously described.



Mode 0 shown

FIGURE 11-1. INTERRUPT REQUEST/ACKNOWLEDGE WITH WAIT STATES

(6) Non maskable interrupt response

Figure 12 illustrates the request/acknowledge cycle for the non-maskable interrupt. A pulse on the $\overline{NMI}$ input sets an internal NMI latch which is tested by the CPU at the end of every instruction. This NMI latch is sampled at the same time as the interrupt line, but this line has priority over the normal interrupt and it cannot be disabled under software control. Its usual function is to provide immediate response to important signals such as an impending power failure. The CPU response to a non maskable interrupt is similar to a normal memory read operation. The only difference being that the content of the data bus is ignored while the processor automatically stores the PC in the external stack and jumps to location 0066H. The service routine for the non maskable interrupt must begin at this location if this interrupt is used.

(7) Halt acknowledge cycle and exit

Whenever a software halt instruction is executed the CPU begins executing NOP's until an interrupt is received (either a non-maskable or a maskable interrupt while the interrupt flip flop is enabled). The two interrupt lines are sampled with the rising clock edge during each T4 state as shown in Figure 13. If a non-maskable interrupt has been received or a maskable interrupt has been received and the interrupt enable flip-flop is set, then the halt state will be exited on the next rising clock edge. The following cycle will then be an interrupt acknowledge cycle corresponding to the type of interrupt that was received. If both are received at this time, then the

non maskable one will be acknowledged since it was highest priority. The purpose of executing NOP instructions while in the halt state is to keep the memory refresh signals active. Each cycle in the halt state is a normal M1 (fetch) cycle except that the data received from the memory is ignored and a NOP instruction is forced internally to the CPU. The halt acknowledge signal is active during this time to indicate that the processor is in the halt state.

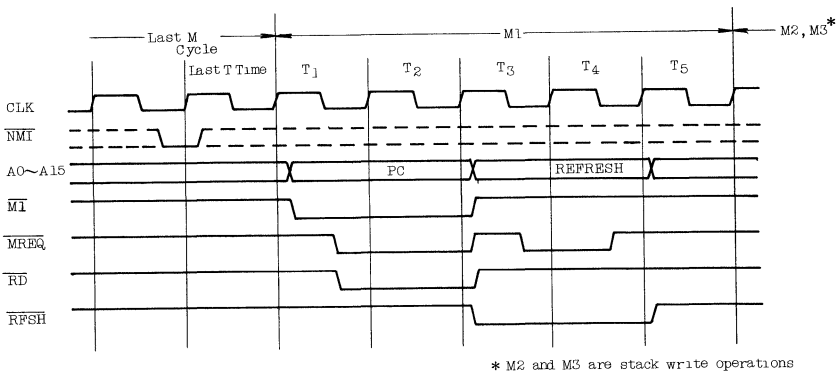


FIGURE 12. NON MASKABLE INTERRUPT RESPONSE

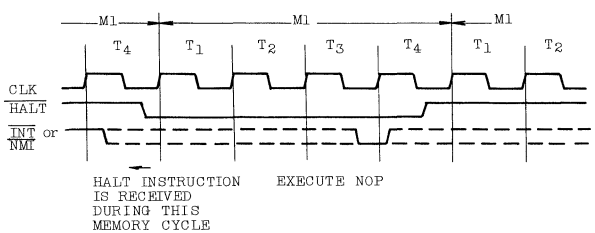


FIGURE 13. HALT ACKNOWLEDGE CYCLE AND EXIT

(8) Reset cycle

$\overline{\text{RESET}}$ must be active for at least three clock cycles for the CPU to properly accept it. As long as $\overline{\text{RESET}}$ remains active, the address and data buses float, and the control outputs are inactive.

Once $\overline{\text{RESET}}$ goes inactive, two internal T cycles are consumed before the CPU resumes normal processing operation. $\overline{\text{RESET}}$ clears the PC (program counter), so the first OPcode fetch will be to location 0000_H.

(See Figure 14.)

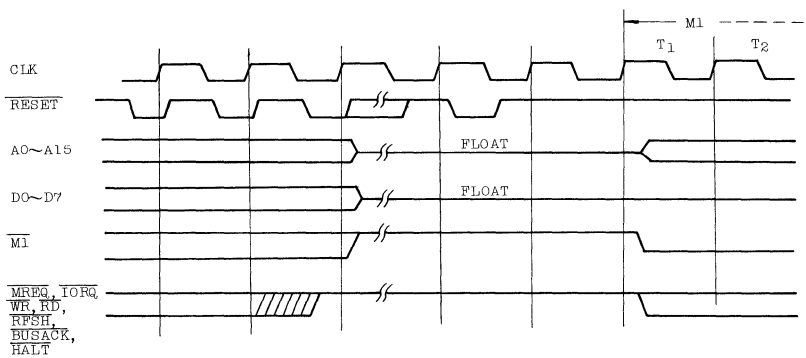


FIGURE 14. RESET CYCLE

POWER DOWN FUNCTION

When system clock to Z80 CPU is stopped at either a high or low level, Z80 CPU stops its operation and maintains registers and control signals.

However I_{CC2} Stand-by Supply Current is guaranteed only when the supplied system clock is stopped at a low level during T4 state of the following machine cycle (actually that is M1 cycle and executes NOP instruction) next to OPcode fetch cycle of HALT instruction. The timing diagram when POWER DOWN FUNCTION is implemented by HALT instruction is shown as figure 15.

This function can be easily realized when T6497 clock generator controller is connected with Z80 CPU.

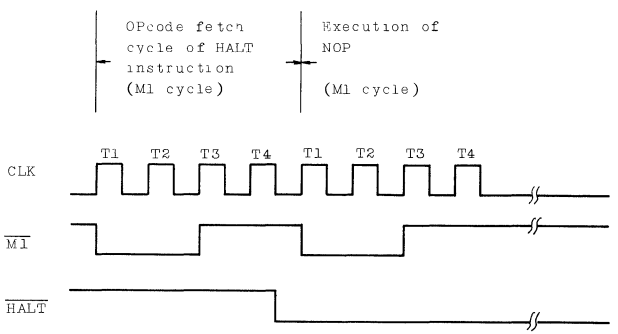


FIGURE 15. TIMING DIAGRAM OF POWER DOWN FUNCTION BY HALT INSTRUCTION

RELEASE FROM POWER DOWN STATE

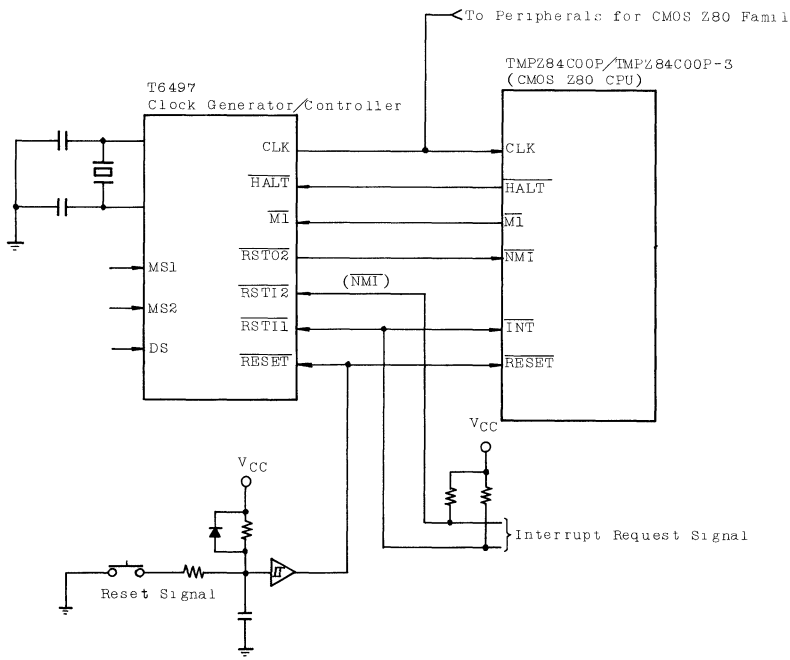
The system clock must be supplied to Z80 CPU to release power down state. When the system clock is supplied to CLK terminal of Z80 CPU, CPU restarts operation continuously from the state when power down function has been implemented.

Note the followings when release from power down state.

(1) When external oscillator has been stopped to enter power down state, some warming-up time may be required to obtain precious and stable system clock for release from power down state.

(2) When HALT instruction is executed to enter power down state, Z80 CPU will enter HALT state. An interrupt signal ($\overline{NMI}$ or INT) or $\overline{RESET}$ signal must be generated to Z80 CPU after the system clock is supplied to release power down state. Otherwise Z80 CPU is still in HALT state even if the system clock is supplied.

Figure 16 shows an example to connect with T6497 clock generator/controller.



INSTRUCTION SET

The Z80 microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor.
It includes such unique operations as a block move for fast, efficient data transfers within memory or between memory and I/O. It also allows operations on any bit in any location in memory.

The Z80 CPU can execute 158 different instruction types including all 78 of the 8080A CPU.

The instructions are devided into the following categories:

- 8-bit loads
- 16-bit loads
- Exchanges, block transfers, and searches
- 8-bit arithmetic and logic operations
- General-purpose arithmetic and CPU control
- 16-bit arithmetic operations
- Rotates and shifts
- Bit set, reset, and test operations
- Jumps
- Calls, returns, and restarts
- Input and output operations

A variety of addressing modes are implemented to permit efficient and data transfer between various registers, memory locations, and input/output devices. These addressing modes are as follows:

- Immediate
- Immediate extended
- Modified page zero
- Relative
- Extended
- Indexed
- Resister
- Resister indirect
- Implied
- Bit

PRELIMINARY

8-BIT LOAD GROUP

Mnemonic	Instruction $D_7D_6D_5D_4D_3D_2D_1D_0$	Code Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
LDr,r	0 1 ← r → ← r' →		r ← r'	–	–	–	–	–	–	1	1	4	r,r Reg.
LDr,n	0 0 ← r → 1 1 0 ◄——— n ———►		r ← n	–	–	–	–	–	–	2	2	7	000 B 001 C
LDr,(HL)	0 1 ← r → 1 1 0		r ← (HL)	–	–	–	–	–	–	1	2	7	010 D
LDr,(IX+d)	1 1 0 1 1 1 0 1 0 1 ← r → 1 1 0 ◄——— d ———	DD	r ← (IX+d)	–	–	–	–	–	–	3	5	19	011 E 100 H 101 L
LDr,(IY+d)	1 1 1 1 1 1 0 1 0 1 ← r → 1 1 0 ◄——— d ———	FD	r ← (IY+d)	–	–	–	–	–	–	3	5	19	111 A
LD(HL),r	0 1 1 1 0 ← r →		(HL) ← r	–	–	–	–	–	–	1	2	7	
LD(IX+d),r	1 1 0 1 1 1 0 1 0 1 1 1 0 ← r → ◄——— d ———	DD	(IX+d) ← r	–	–	–	–	–	–	3	5	19	
LD(IY+d),r	1 1 1 1 1 1 0 1 0 1 1 1 0 ← r → ◄——— d ———	FD	(IY+d) ← r	–	–	–	–	–	–	3	5	19	
LD(HL),n	0 0 1 1 0 1 1 0 ◄——— n ———	36	(HL) ← n	–	–	–	–	–	–	2	3	10	
LD(IX+d),n	1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 ◄——— d ——— ◄——— n ———	DD	(IX+d) ← n	–	–	–	–	–	–	4	5	19	
LD(IY+d),n	1 1 1 1 1 1 0 1 0 0 1 1 0 1 1 0 ◄——— d ——— ◄——— n ———	FD 36	(IY+d) ← n	–	–	–	–	–	–	4	5	19	
LD A, (BC)	0 0 0 0 1 0 1 0	0A	A ← (BC)	–	–	–	–	–	–	1	2	7	
LD A, (DE)	0 0 0 1 1 0 1 0	1A	A ← (DE)	–	–	–	–	–	–	1	2	7	
LD A, (nn)	0 0 1 1 1 0 1 0 ◄——— n ——— ◄——— n ———	3A	A ← (nn)	–	–	–	–	–	–	3	4	13	
LD(BC),A	0 0 0 0 0 0 1 0	02	(BC) ← A	–	–	–	–	–	–	1	2	7	
LD(DE),A	0 0 0 1 0 0 1 0	12	(DE) ← A	–	–	–	–	–	–	1	2	7	
LD(nn),A	0 0 1 1 0 0 1 0 ◄——— n ——— ◄——— n ———	32	(nn) ← A	–	–	–	–	–	–	3	4	13	
LD A, I	1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 1	ED 57	A ← I	0	0	R	IFF	R	–	2	2	9	
LD A, R	1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1	ED 5F	A ← R	0	0	R	IFF	R	–	2	2	9	
LDI,A	1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1	ED 47	I ← A	–	–	–	–	–	–	2	2	9	
LD R, A	1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 1	ED 4F	R ← A	–	–	–	–	–	–	2	2	9	

Notes: r,r' means any of the registers A, B, C, D, E, H, L
IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
Flag Notation: – = flag not affected, R = flag reset, S = flag set.
0 = flag is affected according to the result of the operation.

PRELIMINARY

16-BIT LOAD GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd,nn	0 0 d d 0 0 0 1 ← n → ← n →		dd ← nn	–	–	–	–	–	–	3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LDIX,nn	1 1 0 1 1 1 0 1 0 0 1 0 0 0 0 1 ← n → ← n →	DD 21	IX ← nn	–	–	–	–	–	–	4	4	14	
LDIY,nn	1 1 1 1 1 1 0 1 0 0 1 0 0 0 0 1 ← n → ← n →	FD 21	IY ← nn	–	–	–	–	–	–	4	4	14	
LD HL, (nn)	0 0 1 0 1 0 1 0 ← n → ← n →	2A	H ← (nn+1) L (nn)	–	–	–	–	–	–	3	5	16	
LDdd, (nn)	1 1 1 0 1 1 0 1 0 1 d d 1 0 1 1 ← n → ← n →	ED	dd_H ← (nn+1) dd_L ← (nn)	–	–	–	–	–	–	4	6	20	
LDIX, (nn)	1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 0 ← n → ← n →	DD 2A	IX_H ← (nn+1) IX_L ← (nn)	–	–	–	–	–	–	4	6	20	
LDIY, (nn)	1 1 1 1 1 1 0 1 0 0 1 0 1 0 1 0 ← n → ← n →	FD 2A	IY_H ← (nn+1) IY_L ← (nn)	–	–	–	–	–	–	4	6	20	
LD(nn),HL	0 0 1 0 0 0 1 0 ← n → ← n →	22	(nn+1) ← H (nn) ← L	–	–	–	–	–	–	3	5	16	
LD(nn),dd	1 1 1 0 1 1 0 1 0 1 d d 0 0 1 1 ← n → ← n →	ED	(nn+1) ← dd_H (nn) ← dd_L	–	–	–	–	–	–	4	6	20	
LD(nn),IX	1 1 0 1 1 1 0 1 0 0 1 0 0 0 1 0 ← n → ← n →	DD 22	(nn+1) ← IX_H (nn) ← IX_L	–	–	–	–	–	–	4	6	20	
LD(nn),IY	1 1 1 1 1 1 0 1 0 0 1 0 0 0 1 0 ← n → ← n →	FD 22	(nn+1) ← IY_H (nn) ← IY_L	–	–	–	–	–	–	4	6	20	
LD SP,HL	1 1 1 1 1 0 0 1	F9	SP ← HL	–	–	–	–	–	–	1	1	6	
LD SP, IX	1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 1	DD F9	SP ← IX	–	–	–	–	–	–	2	2	10	
LD SP, IY	1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1	FD F9	SP ← IY	–	–	–	–	–	–	2	2	10	

PRELIMINARY

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
PUSHqq	1 1 q q 0 1 0 1		$(SP-2)\leftarrow qq_L$ $(SP-1)\leftarrow qq_H$ $SP\rightarrow SP-2$	-	-	-	-	-	-	1	3	11	qq Pair 00 BC 01 DE
PUSH IX	1 1 0 1 1 1 0 1 1 1 1 0 0 1 0 1	DD E5	$(SP-2)\leftarrow IX_L$ $(SP-1)\leftarrow IX_H$ $SP\rightarrow SP-2$	-	-	-	-	-	-	2	4	15	10 HL 11 AF
PUSH IY	1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1	FD E5	$(SP-2)\leftarrow IY_L$ $(SP-2)\leftarrow IY_H$ $SP\rightarrow SP-2$	-	-	-	-	-	-	2	4	15	
POPqq	1 1 q q 0 0 0 1		$qq_H\leftarrow(SP+1)$ $qq_L\leftarrow(SP)$ $SP\rightarrow SP+2$	-	-	-	-	-	-	1	3	10	
POPIX	1 1 0 1 1 1 0 1 1 1 1 0 0 0 0 1	DD E1	$IX_H\leftarrow(SP+1)$ $IX_L\leftarrow(SP)$ $SP\rightarrow SP+2$	-	-	-	-	-	-	2	4	14	
POPIY	1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 1	FD E1	$IY_H\leftarrow(SP+1)$ $IY_L\leftarrow(SP)$ $SP\rightarrow SP+2$	-	-	-	-	-	-	2	4	14	

Notes: dd is any of the register pairs BC, DE, HL, SP
qq is any of the register pairs AF, BC, DE, HL
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively. e.g. $BC_L=C$, $AF_H=A$

Flag Notation: - = flag not affected, R = flag reset, S = flag set,
O=flag is affected according to the result of the operation.

PRELIMINARY

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	1 1 1 0 1 0 1 1	EB	DE↔HL	–	–	–	–	–	–	1	1	4	
EX AF, AF'	0 0 0 0 1 0 0 0	08	AF↔AF'	–	–	–	–	–	–	1	1	4	(Exx)
EXX	1 1 0 1 1 0 0 1	D9	(BC↔BC' DE↔DE' HL↔HL')	–	–	–	–	–	–	1	1	4	Register bank and auxiliary
EX(SP),HL	1 1 1 0 0 0 1 1	E3	H↔(SP+1) L↔(SP)	–	–	–	–	–	–	1	5	19	register bank ex-
EX(SP),IX	1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1	DD E3	IX_H↔(SP+1) IX_L↔(SP)	–	–	–	–	–	–	2	6	23	change
EX(SP),IY	1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1	FD E3	IY_H↔(SP+1) IY_L↔(SP)	–	–	–	–	–	–	2	6	23	
LDI	1 1 1 0 1 1 0 1 1 0 1 0 0 0 0 0	ED A0	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC–1	–	–	R	①	R	–	2	4	16	*1
LDIR	1 1 1 0 1 1 0 1 1 0 1 1 0 0 0 0	ED B0	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC–1 Repeat until BC=0	–	–	R	R	R	–	2 2	5 4	21 16	If BC≠0 If BC=0
LDD	1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 0	ED A8	(DE)←(HL) DE←DE–1 HL←HL–1 BC←BC–1	–	–	R	①	R	–	2	4	16	
LDDR	1 1 1 0 1 1 0 1 1 0 1 1 1 0 0 0	ED B8	(DE)←(HL) DE←DE–1 HL←HL–1 BC←BC–1 Repeat until BC=0	–	–	R	R	R	–	2 2	5 4	21 16	If BC≠0 If BC=0
CPI	1 1 1 0 1 1 0 1 1 0 1 0 0 0 0 1	ED A1	A–(HL) HL←HL+1 BC←BC–1	○	②	○	①	S	–	2	4	16	
CPIR	1 1 1 0 1 1 0 1 1 0 1 1 0 0 0 1	ED B1	A–(HL) HL←HL+1 BC←BC–1 Repeat until A=(HL) or BC=0	○	②	○	①	S	–	2 2	5 4	21 16	*2 *3
CPD	1 1 1 0 1 1 0 1 1 0 1 0 1 0 0 1	ED A9	A–(HL) HL←HL–1 BC←BC–1	○	②	○	①	S	–	2	4	16	

PRELIMINARY

Nmenomic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
CPDR	1 1 1 0 1 1 0 1	ED	A-(HL)	0	②	0	①	S	-	2	5	21	*2
	1 0 1 1 1 0 0 1	B9	HL←HL-1 BC←BC-1 Repeat until A=(HL)or BC=0							2	4	16	*3

Notes: ① P/V flag is 0 if the result of BC-1,=0, otherwise P/V=1
② Z flag is 1 if A=(HL), otherwise Z=0.

Flag Notation: - = flag not affected, R = flag reset, S = flag set.
0 = flag is affected according to the result of the operation.

*1 LDI: Load (HL) into (DE), increment the pointers and decrement the byte counter (BC).

*2 : If BC≠0 and A≠(HL)

*3 : If BC=0 or A=(HL)

PRELIMINARY

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A,r	1 0 [0 0 0] ← r →		A←A+r	0	0	0	V	R	0	1	1	4	r Reg.
ADD A,n	1 1 [0 0 0] 1 1 0 ← n →		A←A+n	0	0	0	V	R	0	2	2	7	000 B
ADD A,(HL)	1 0 [0 0 0] 1 1 0		A←A+(HL)	0	0	0	V	R	0	1	2	7	001 C
ADD A,(IX+d)	1 1 0 1 1 1 0 1 1 0 [0 0 0] 1 1 0 ← d →	DD	A←A+(IX+d)	0	0	0	V	R	0	3	5	19	010 D 011 E 100 H 101 L
ADD A,(IY+d)	1 1 1 1 1 1 0 1 1 0 [0 0 0] 1 1 0 ← d →	FD	A←A+(IY+d)	0	0	0	V	R	0	3	5	19	111 A
ADCA,s	[0 0 1]		A←A+s+CY	0	0	0	V	R	0				s is any of r, n, (HL),(IX+d), (IY+d)as shown for ADD instruction.
SUBs	[0 1 0]		A←A-s	0	0	0	V	S	0				
SBCA,s	[0 1 1]		A←A-s-CY	0	0	0	V	S	0				
ANDs	[1 0 0]		A←A ∧ s	0	0	S	P	R	R				
ORs	[1 1 0]		A←A ∨ s	0	0	R	P	R	R				
XORs	[1 0 1]		A←A ⊕ s	0	0	R	P	R	R				
CPs	[1 1 1]		A-s	0	0	0	V	S	0				
INCr	0 0 ← r → [1 0 0]		r←r+1	0	0	0	V	R	-	1	1	4	The indicated bits replace the [000] in the ADD set above.
INC(HL)	0 0 1 1 0 [1 0 0]		(HL)←(HL)+1	0	0	0	V	R	-	1	3	11	
INC(IX+d)	1 1 0 1 1 1 0 1 0 0 1 1 0 [1 0 0] ← d →	DD	(IX+d)←(IX+d)+1	0	0	0	V	R	-	3	6	23	
INC(IY+d)	1 1 1 1 1 1 0 1 0 0 1 1 0 [1 0 0] ← d →	FD	(IY+d)←(IY+d)+1	0	0	0	V	R	-	3	6	23	
DECs	[1 0 1]		s←s-1	0	0	0	V	S	-				s is any of r,(HL), (IX+d),(IY+d) as shown for INC. DEC same format and states as INC. Replace [100] with [101] in OP Code.

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V=1 means overflow, V=0 means not overflow, P=1 means parity of the result is even, P=0 means parity of the result is odd.

Flag Notation: - = flag not affected, R = flag reset, S = flag set,
0 = flag is affected according to the result of the operation.

PRELIMINARY

GENERAL PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	0 0 1 0 0 1 1 1	27	Converts acc, content into packed BCD following add or subtract with packed BCD operands	0	0	0	P	–	0	1	1	4	Decimal adjust accumulator
CPL	0 0 1 0 1 1 1 1	2F	$A \leftarrow \overline{A}$	–	–	S	–	S	–	1	1	4	
NEG	1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 0	ED 44	$A \leftarrow \overline{A}+1$	0	0	0	V	S	0	2	2	8	
CCF	0 0 1 1 1 1 1 1	3F	$CY \leftarrow \overline{CY}$	–	–	X	–	R	0	1	1	4	
SCF	0 0 1 1 0 1 1 1	37	CY←1	–	–	R	–	R	S	1	1	4	
NOP	0 0 0 0 0 0 0 0	00	No operation	–	–	–	–	–	–	1	1	4	
HALT	0 1 1 1 0 1 1 0	76	CPU halted	–	–	–	–	–	–	1	1	4	
DI*	1 1 1 1 0 0 1 1	F3	IFF←0	–	–	–	–	–	–	1	1	4	
EI*	1 1 1 1 1 0 1 1	FB	IFF←1	–	–	–	–	–	–	1	1	4	
IM 0	1 1 1 0 1 1 0 1 0 1 0 0 0 1 1 0	ED 46	Set interrupt mode 0	–	–	–	–	–	–	2	2	8	
IM 1	1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0	ED 56	Set interrupt mode 1	–	–	–	–	–	–	2	2	8	
IM 2	1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 0	ED 5E	Set interrupt mode 2	–	–	–	–	–	–	2	2	8	

Notes: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.

Flag Notation: - = flag not affected, R = flag reset, S = flag set, X = flag is unknown,
0 = flag is affected according to the result of the operation.

* Interrupts are not sampled at the end of EI or DI

16-BIT ARITHMETIC GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL,ss	0 0 s s 1 0 0 1		HL←HL+ss	–	–	X	–	R	0	1	3	11	ss Reg.
ADC HL,ss	1 1 1 0 1 1 0 1 0 1 s s 1 0 1 0	ED	HL←HL+ss+CY	0	0	X	V	R	0	2	4	15	00 BC 01 DE
SBC HL,ss	1 1 1 0 1 1 0 1 0 1 s s 0 0 1 0	ED	HL←HL-ss-CY	0	0	X	V	S	0	2	4	15	10 HL 11 SP
ADDIX,pp	1 1 0 1 1 1 0 1 1 1 p p 1 0 0 1	DD	IX←IX+pp	–	–	X	–	R	0	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADDIY,rr	1 1 1 1 1 1 0 1 0 0 r r 1 0 0 1	FD	IY←IY+rr	–	–	X	–	R	0	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INCss	0 0 s s 0 0 1 1		ss←ss+1	–	–	–	–	–	–	1	1	6	
INCIX	1 1 0 1 1 1 0 1 0 0 1 0 0 0 1 1	DD 23	IX←IX+1	–	–	–	–	–	–	2	2	10	
INCIY	1 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1	FD 23	IY←IY+1	–	–	–	–	–	–	2	2	10	
DECss	0 0 s s 1 0 1 1		ss←ss-1	–	–	–	–	–	–	2	2	6	
DECIX	1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 1	DD 2B	IX←IX-1	–	–	–	–	–	–	2	2	10	
DECIY	1 1 1 1 1 1 0 1 0 0 1 0 1 0 1 1	FD 2B	IY←IY-1	–	–	–	–	–	–	2	2	10	

Notes: ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: – = flag not affected, R = flag reset, S = flag set, X = flag is unknown.
0 = flag is affected according to the result of the operation.

ROTATE AND SHIFT GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA	0 0 0 0 0 1 1 1	07	CY ← 7 ← 0 ← A	–	–	R	–	R	0	1	1	4	•Rotate left circular accumulator
RLA	0 0 0 1 0 1 1 1	17	CY ← 7 ← 0 ← A	–	–	R	–	R	0	1	1	4	•Rotate left accumulator
RRCA	0 0 0 0 1 1 1 1	0F	7 → 0 → CY, A	–	–	R	–	R	0	1	1	4	•Rotate right circular accumulator
RRA	0 0 0 1 1 1 1 1	1F	7 → 0 → CY, A	–	–	R	–	R	0	1	1	4	•Rotate right accumulator
RLCr	1 1 0 0 1 0 1 1 0 0 0 0 0 ← r	CB	CY ← 7 ← 0 ← r, (HL), (IX+d), (IY+d)	0	0	R	P	R	0	2	2	8	•Rotate left circular register r
RLC(HL)	1 1 0 0 1 0 1 1 0 0 0 0 0 1 1 0	CB		0	0	R	P	R	0	2	4	15	
RLC(IX+d)	1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 0 0 0 0 0 1 1 0	DD CB		0	0	R	P	R	0	4	6	23	
RLC(IY+d)	1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 0 0 0 0 0 0 0 0	FD CB		0	0	R	P	R	0	4	6	23	
RLs	0 1 0		CY ← 7 ← 0 s≡r, (HL), (IX+d), (IY+d)	0	0	R	P	R	0				Instruction format and states are as shown for RLC's. To form new Op-Code replace 000 of RLC's with shown code.
RRCs	0 0 1		7 → 0 → CY s≡r, (HL), (IX+d), (IY+d)	0	0	R	P	R	0				
RRs	0 1 1		CY ← 7 ← 0 s≡r, (HL), (IX+d), (IY+d)	0	0	R	P	R	0				
SLAs	1 0 0		CY ← 7 ← 0 ← 0 s≡r, (HL), (IX+D), (IY+d)	0	0	R	P	R	0				
SRAs	1 0 1		7 → 0 → CY s≡r, (HL), (IX+D), (IY+d)	0	0	R	P	R	0				
SRLs	1 1 1		0 → 7 → 0 → CY s≡r, (HL), (IX+d), (IY+d)	0	0	R	P	R	0				

PRELIMINARY

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P_r/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLD	1 1 1 0 1 1 0 1 0 1 1 0 1 1 1 1	ED 6F	A (7-4)(3-0) ↔ (7-4)(3-0) (HL)	0	0	R	P	R	-	2	5	18	Rotate digit left & right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected.
RRD	1 1 1 0 1 1 0 1 0 1 1 0 0 1 1 1	ED 67	A (7-4)(3-0) → (7-4)(3-0) (HL)	0	0	R	P	R	-	2	5	18	

Flag Notation: - = flag not affected, R = flag reset, S = flag set,
O = flag is affected according to the result of the operation.

PRELIMINARY

BIT SET, RESET AND TEST GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b,r	1 1 0 0 1 0 1 1 0 1 ← b → ← r →	CB	$Z \leftarrow \overline{r_b}$	X	O	S	X	R	–	2	2	8	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
BIT b,(HL)	1 1 0 0 1 0 1 1 0 1 ← b → 1 1 0	CB	$Z \leftarrow (HL)_b$	X	O	S	X	R	–	2	3	12	
BIT b,(IX+d)b	1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 0 1 ← b → 1 1 0	DD CB	$Z \leftarrow \overline{(IX+d)_b}$	X	O	S	X	R	–	4	5	20	
BIT b,(IY+d)b	1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 0 1 ← b → 1 1 0	FD CB	$Z \leftarrow \overline{(IY+d)_b}$	X	O	S	X	R	–	4	5	20	b: 000 Bit 0, 001 1, 010 2, 011 3, 100 4, 101 5, 110 6, 111 7
SET b,r	1 1 0 0 1 0 1 1 1 1 ← b → ← r →	CB	$r_b \leftarrow 1$	–	–	–	–	–	–	2	2	8	
SET b,(HL)	1 1 0 0 1 0 1 1 1 1 ← b → 1 1 0	CB	$(HL)_b \leftarrow 1$	–	–	–	–	–	–	2	4	15	
SET b,(IX+d)	1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 1 1 ← b → 1 1 0	DD CB	$(IX+d)_b \leftarrow 1$	–	–	–	–	–	–	4	6	23	
SET b,(IY+d)	1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 ← d → 1 1 ← b → 1 1 0	FD CB	$(IY+d)_b \leftarrow 1$	–	–	–	–	–	–	4	6	23	
RES b,s	1 0		$s_b \leftarrow 0$ s≡r,(HL), (IX+d), (IY+d)	–	–	–	–	–	–				To form new Op Code replace 11 of SET b,s with 10 . Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) or location s.

Flag Notation: – = flag not affected, R = flag reset, S = flag set, X = flag is unknown,
O = flag is affected according to the result of the operation.

PRELIMINARY

JUMP GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
JPnn	1 1 0 0 0 0 1 1 ←n→ ←n→	C3	PC←nn	-	-	-	-	-	-	3	3	10	
JPcc,nn	1 1 C C 0 1 0 ←n→ ←n→		If condition cc is true PC←nn, otherwise continue	-	-	-	-	-	-	3	3	10	cc: 000 NZ non zero; 001 Z zero; 010 NC non carry; 011 C carry; 100 PO parity odd; 101 PE parity even; 110 P sign positive; 111 M sign negative
JRe	0 0 0 1 1 0 0 0 ←e-2→	18	PC←PC+e	-	-	-	-	-	-	2	3	12	
JR C,e	0 0 1 1 1 0 0 0 ←e-2→	38	If C=0, continue If C=1, PC←PC+e	-	-	-	-	-	-	2 2	2 3	7 12	If condition not met If condition is met
JR NC,e	0 0 1 1 0 0 0 0 ←e-2→	30	If C=1, continue If C=0, PC←PC+e	-	-	-	-	-	-	2 2	2 3	7 12	If condition not met If condition is met
JR Z,e	0 0 1 0 1 0 0 0 ←e-2→	38	If Z=0, continue If Z=1, PC←PC+e	-	-	-	-	-	-	2 2	2 3	7 12	If condition not met If condition is met
JR NZ,e	0 0 1 0 0 0 0 0 ←e-2→	20	If Z=1, continue If Z=0, PC←PC+e	-	-	-	-	-	-	2 2	2 3	7 12	If condition not met If condition is met
JP(HL)	1 1 1 0 1 0 0 1	E9	PC←HL	-	-	-	-	-	-	1	1	4	
JP(IX)	1 1 0 1 1 1 0 1 1 1 1 0 1 0 0 1	DD E9	PC←IX	-	-	-	-	-	-	2	2	8	
JP(IY)	1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1	FD E9	PC←IY	-	-	-	-	-	-	2	2	8	
DJNZ,e	0 0 0 1 0 0 0 0 ←e-2→	10	B←B-1 If B=0, continue If B≠0, PC←PC+e	-	-	-	-	-	-	2 2	2 3	8 13	If B=0 If B≠0

Notes: e represents the extension in the relative addressing mode. e is a signed two's complement number in the range ⟨-126,129⟩. e-2 in the op-code provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.

Flag Notation: - = flag not affected, R = flag reset, S = flag set,
0 = flag is affected according to the result of the operation.

PRELIMINARY

CALL AND RETURN GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	P_E/V	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALLnn	1 1 0 0 1 1 0 1 ← n → ← n →	CD	(SP-1)←PC_H (SP-2)←PC_L PC←nn	-	-	-	-	-	-	3	5	17	
CALLcc,nn	1 1 c c 1 0 0 ← n → ← n →		If condition cc is false continue, otherwise same is CALLnn	-	-	-	-	-	-	3 3	3 5	10 17	If cc is false If cc is true
RET	1 1 0 0 1 0 0 1	C9	PC_L←(SP) PC_H←(SP+1)	-	-	-	-	-	-	1	3	10	
RETcc	1 1 c c 0 0 0		If condition cc is false continue, otherwise same as RET	-	-	-	-	-	-	1 1	1 3	5 11	If cc is false If cc is true
RETI	1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 1	ED 4D	Return from interrupt	-	-	-	-	-	-	2	4	14	
RETN*	1 1 1 0 1 1 0 1 0 1 0 0 0 1 0 1	ED 45	Return from non maskable interrupt	-	-	-	-	-	-	2	4	14	
RSTp	1 1 ← t → 1 1 1		(SP-1)←PC_H (SP-2)←PC_L PC_H←0 PC_L←p	-	-	-	-	-	-	1	3	11	

cc	Condition
000	NZ non zero
001	Z zero
010	NC non carry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

* RETN loads $IFF_2 \rightarrow IFF_1$

Flag Notation: - = flag not affected, R = flag reset, S = flag set,
0 = flag is affected according to the result of the operation.

PRELIMINARY

INPUT AND OUTPUT GROUP

Mnemonic	Instruction Code $D_7D_6D_5D_4D_3D_2D_1D_0$	Hex	Operation	Flags S	Z	H	$P_{/V}$	N	C	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A,(n)	1 1 0 1 1 0 1 1 ←— n —→	DB	A←(n)	-	-	-	-	-	-	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
INr,(C)	1 1 1 0 1 1 0 1 0 1 r 0 0 0	ED	r←(C) If r=110 only the flags will be affected	0	0	0	P	R	-	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0	ED A2	(HL)←(C) B←B-1 HL←HL+1	X	①	X	X	S	X	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 0	ED B2	(HL)←(C) B←B-1 HL←HL+1 Repeat until B=0	X	1	X	X	S	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0	ED AA	(HL)←(C) B←B-1 HL←HL-1	X	①	X	X	S	X	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 0	ED BA	(HL)←(C) B←B-1 HL←HL-1 Repeat until B=0	X	1	X	X	S	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT(n),A	1 1 0 1 0 0 1 1	D3	(n)←A	-	-	-	-	-	-	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
OUT(C),r	1 1 1 0 1 1 0 1 0 1 r 0 0 1	ED	(C)←r	-	-	-	-	-	-	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 1	ED A3	B←B-1 (C)←(HL) HL←HL+1	-	①	-	-	S	X	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1	ED B3	B←B-1 (C)←(HL) HL←HL+1 Repeat until B=0	X	1	X	X	S	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 1	ED AB	(C)←(HL) B←B-1 HL←HL-1	X	①	X	X	S	X	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1	ED BB	(C)←(HL) B←B-1 HL←HL-1 Repeat until B=0	X	1	X	X	S	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

Notes: ① If the result of B-1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: - = flag not affected, R = flag reset, S = flag set, X = flag is unknown,
0 = flag is affected according to the result of the operation.

PRELIMINARY

TMPZ84C00P-3/TMPZ84C00P

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage with respect to V_{SS}	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to V_{CC} +0.5V
P_D	Power Dissipation (T_A=85°C)	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

DC CHARACTERISTICS (I)

T_A=-40°C to 85°C, V_{CC}=5V±10% , V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{ILC}	Clock Input Low Voltage		-0.3	-	0.6	V
V_{IHC}	Clock Input High Voltage		V_{CC} -0.6	-	V_{CC} +0.3	V
V_{IL}	Input Low Voltage (except CLK)		-0.5	-	0.8	V
V_{IH}	Input High Voltage (except CLK)		2.2	-	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL}=2.0mA	-	-	0.4	V
V_{OH1}	Output High Voltage (1)	I_{OH}=-1.6mA	2.4	-	-	V
V_{OH2}	Output High Voltage (2)	I_{OH}=-250μA	V_{CC} -0.8	-	-	V
I_{LI}	Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{LO}	3-State Output Leakge Current in Float	$V_{SS}+0.4 \leq V_{OUT} \leq V_{CC}$	-	-	±10	μA
I_{CC1}	Operating Supply Current	V_{CC}=5V, CLK=4MHz $V_{IH}=V_{CC}-0.2V$ V_{IL}=0.2V	-	15	25	mA
(1) I_{CC2}	Stand-by Supply Current	V_{CC}=5V CLK=(1) $V_{IL}=V_{CC}-0.2V$ V_{IH}=0.2V	-	0.5	10	μA

Note (1) I_{CC2} Stand-by Supply Current is guaranteed only when the supplied clock is stopped at a low level during T4 state of the following machine cycle (M1) next to OPcode fetch cycle of HALT instruction.

PRELIMINARY

TMPZ84C00P (4MHz Operation)

AC CHARACTERISTICS

T_A = -40°C to 85°C, Vcc = 5V ± 10%, Vss = 0 V, Unless otherwise noted.

NUMBER	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
1	TcC	Clock Cycle Time		250	-	DC	ns
2	TwCh	Clock Pulse Width (High)		110	-	DC	ns
3	TwCl	Clock Pulse Width (Low)		110	-	DC	ns
4	TfC	Clock Fall Time		-	-	30	ns
5	TrC	Clock Rise Time		-	-	30	ns
6	TdCr(A)	Clock ↑ to Address Valid Delay		-	-	110	ns
7	TdA(MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay		65	-	-	ns
8	TdCf(MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		-	-	85	ns
9	TdCr(MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay		-	-	85	ns
10	TwMREQh	$\overline{MREQ}$ Pulse Width (High)		110	-	-	ns
11	TwMREQl	$\overline{MREQ}$ Pulse Width (Low)		220	-	-	ns
12	TdCf(MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		-	-	85	ns
13	TdCf(RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay		-	-	95	ns
14	TdCr(RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay		-	-	85	ns
15	TsD(Cr)	Data Setup Time to Clock ↑		35	-	-	ns
16	ThD(RDr)	Data Hold Time to $\overline{RD}$ ↑		0	-	-	ns
17	TsWAIT(Cf)	$\overline{WAIT}$ Setup Time to Clock ↓	C_L=100pF	70	-	-	ns
* 18	ThWAIT(Cf)	$\overline{WAIT}$ Hold Time after Clock ↓		10	-	-	ns
19	TdCr(M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay		-	-	100	ns
20	TdCr(M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay		-	-	100	ns
21	TdCr(RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Delay		-	-	130	ns
22	TdCr(RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay		-	-	120	ns
23	TdCf(RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay		-	-	85	ns
24	TdCr(RDf)	Clock ↑ to $\overline{RD}$ ↓ Delay		-	-	85	ns
25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4 or M_5 Cycles		50	-	-	ns
26	TdA(IORQf)	Address Stable prior to $\overline{IORQ}$ ↓		180	-	-	ns
27	TdCr(IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay		-	-	75	ns
28	TdCf(IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay		-	-	85	ns
29	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓		80	-	-	ns

PRELIMINARY

NUMBER	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
30	TdCf(WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay	C_L=100pF	-	-	80	ns
31	TwWR	$\overline{WR}$ Pulse Width		220	-	-	ns
32	TdCf(WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay		-	-	80	ns
33	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓		-10	-	-	ns
34	TdCr(WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay		-	-	65	ns
35	TdWRr(D)	Data Stable from $\overline{WR}$ ↑		60	-	-	ns
36	TdCf(HALT)	Clock ↓ to $\overline{HALT}$ ↑ or ↓		-	-	300	ns
37	TwNMI	$\overline{NMI}$ Pulse Width		80	-	-	ns
38	TsBUSREQ(Cr)	$\overline{BUSREQ}$ Setup Time to Clock ↑		50	-	-	ns
*39	ThBUSREQ(Cr)	$\overline{BUSREQ}$ Hold Time after Clock ↑		10	-	-	ns
40	TdCr(BUSACKf)	Clock ↑ to $\overline{BUSACK}$ ↓ Delay		-	-	100	ns
41	TdCf(BUSACKr)	Clock ↓ to $\overline{BUSACK}$ ↑ Delay		-	-	100	ns
42	TdCr(Dz)	Clock ↑ to Data Float Delay		-	-	90	ns
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$)		-	-	80	ns
44	TdCr(Az)	Clock ↑ to Address Float Delay		-	-	90	ns
45	TdCTr(A)	$\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time		80	-	-	ns
46	TsRESET(Cr)	$\overline{RESET}$ to Clock ↑ Setup Time		60	-	-	ns
*47	ThRESET(Cr)	$\overline{RESET}$ to Clock ↑ Hold Time		10	-	-	ns
48	TsINTf(Cr)	$\overline{INT}$ to Clock ↑ Setup Time		80	-	-	ns
*49	ThINTr(Cr)	$\overline{INT}$ to Clock ↑ Hold Time		10	-	-	ns
50	TdM1f(IORQf)	$\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay		565	-	-	ns
51	TdCf(IORQf)	Clock ↓ to $\overline{IORQ}$ ↓ Delay		-	-	85	ns
52	TdCf(IORQr)	Clock ↑ to $\overline{IORQ}$ ↑ Delay		-	-	85	ns
53	TdCf(D)	Clock ↓ to Data Valid Delay		-	-	150	ns

NOTE 1) Timing Measurements are made at the following voltage.

Input V_{IH}=2.4V, V_{IL}=0.4V, V_{IHC}=V_{CC}-0.6V, V_{ILC}=0.6V

Output V_{OH}=2.2V, V_{IL}=0.8V

NOTE 2) The Items attached * Mark are not compatible with NMOS Z80 SPECS.

PRELIMINARY

TMPZ84C00P-3 (2.5MHz Operation)

AC CHARACTERISTICS

TA=-40°C to 85°C, Vcc=5V±10%, Vss=0V, Unless otherwise noted.

NUMBER	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
1	TcC	Clock Cycle Time	C_L=100pF	400	-	DC	ns
2	TwCh	Clock Pulse Width (High)		180	-	DC	ns
3	TwCl	Clock Pulse Width (Low)		180	-	DC	ns
4	TfC	Clock Fall Time		-	-	30	ns
5	TrC	Clock Rise Time		-	-	30	ns
6	TdCr(A)	Clock ↑ to Address Valid Delay		-	-	145	ns
7	TdA(MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay		125	-	-	ns
8	TdCf(MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		-	-	100	ns
9	TdCr(MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay		-	-	100	ns
10	TwMREQh	$\overline{MREQ}$ Pulse Width (High)		170	-	-	ns
11	TwMREQl	$\overline{MREQ}$ Pulse Width (Low)		360	-	-	ns
12	TdCf(MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		-	-	100	ns
13	TdCf(RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay		-	-	130	ns
14	TdCr(RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay		-	-	100	ns
15	TsD(Cr)	Data Setup Time to Clock ↑		50	-	-	ns
16	ThD(RDr)	Data Hold Time to $\overline{RD}$ ↑		0	-	-	ns
17	TsWAIT(Cf)	$\overline{WAIT}$ Setup Time to Clock ↓		70	-	-	ns
* 18	ThWAIT(Cf)	$\overline{WAIT}$ Hold Time after Clock ↓		20	-	-	ns
19	TdCr(M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay		-	-	130	ns
20	TdCr(M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay		-	-	130	ns
21	TdCr(RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Delay		-	-	180	ns
22	TdCr(RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay		-	-	150	ns
23	TdCf(RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay		-	-	110	ns
* 24	TdCr(RDf)	Clock ↑ to $\overline{RD}$ ↓ Delay		-	-	110	ns
25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4 or M_5 Cycles		60	-	-	ns
26	TdA(IORQf)	Address Stable prior to $\overline{IORQ}$ ↓		320	-	-	ns
* 27	TdCr(IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay		-	-	100	ns
28	TdCf(IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay		-	-	110	ns
29	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓		190	-	-	ns

PRELIMINARY

NUMBER	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
30	TdCf(WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay		-	-	90	ns
31	TwWR	$\overline{WR}$ Pulse Width		360	-	-	ns
32	TdCf(WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay		-	-	100	ns
33	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓		20	-	-	ns
* 34	TdCr(WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay		-	-	100	ns
35	TdWRr(D)	Data Stable from $\overline{WR}$ ↑		120	-	-	ns
36	TdCf(HALT)	Clock ↓ to $\overline{HALT}$ ↑ or ↓		-	-	300	ns
37	TwNMI	$\overline{NMI}$ Pulse Width		80	-	-	ns
38	TsBUSREQ(Cr)	$\overline{BUSREQ}$ Setup Time to Clock ↑		80	-	-	ns
* 39	ThBUSREQ(Cr)	$\overline{BUSREQ}$ Hold Time after Clock ↑		20	-	-	ns
40	TdCr(BUSACKf)	Clock ↑ to $\overline{BUSACK}$ ↓ Delay		-	-	120	ns
41	TdCf(BUSACKr)	Clock ↓ to $\overline{BUSACK}$ ↑ Delay	C_L=100pF	-	-	110	ns
42	TdCr(Dz)	Clock ↑ to Data Float Delay		-	-	90	ns
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$)		-	-	110	ns
44	TdCr(Az)	Clock ↑ to Address Float Delay		-	-	110	ns
45	TdCTr(A)	$\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time		160	-	-	ns
46	TsRESET(Cr)	$\overline{RESET}$ to Clock ↑ Setup Time		90	-	-	ns
* 47	ThRESET(Cr)	$\overline{RESET}$ to Clock ↑ Hold Time		20	-	-	ns
48	TsINTf(Cr)	$\overline{INT}$ to Clock ↑ Setup Time		80	-	-	ns
* 49	ThINTr(Cr)	$\overline{INT}$ to Clock ↑ Hold Time		20	-	-	ns
50	TdM1f(IORQf)	$\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay		920	-	-	ns
51	TdCf(IORQf)	Clock ↓ to $\overline{IORQ}$ ↓ Delay		-	-	110	ns
* 52	TdCf(IORQr)	Clock ↑ to $\overline{IORQ}$ ↑ Delay		-	-	115	ns
53	TdCf(D)	Clock ↓ to Data Valid Delay		-	-	230	ns

NOTE 1) Timing Measurements are made at the following voltage.

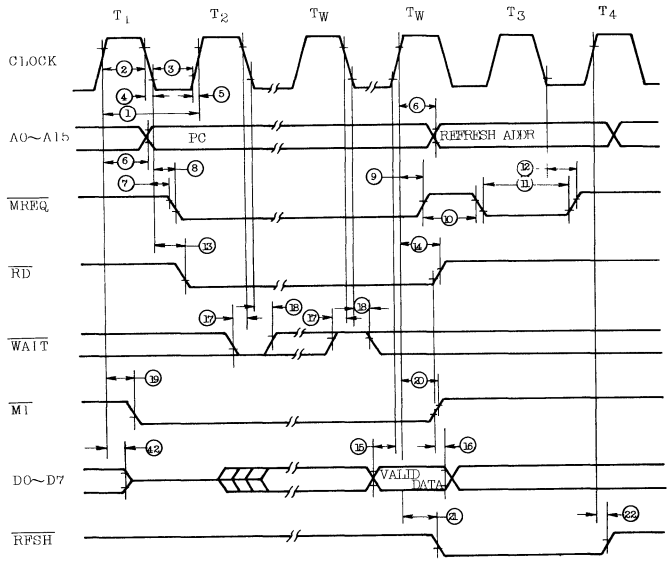
Input V_{IH}=2.4V, V_{IL}=0.4V, V_{IHC}=V_{CC}-0.6V, V_{ILC}=0.6V

Output V_{OH}=2.2V, V_{IL}=0.8V

NOTE 2) The Items attached * Mark are not compatible with NMOS Z80 SPECS.

PRELIMINARY

TIMING WAVEFORMS



NOTE: T_W-Wait cycle added when necessary for slow ancilliary devices.

FIGURE 17. INSTRUCTION OPCODE FETCH

PRELIMINARY

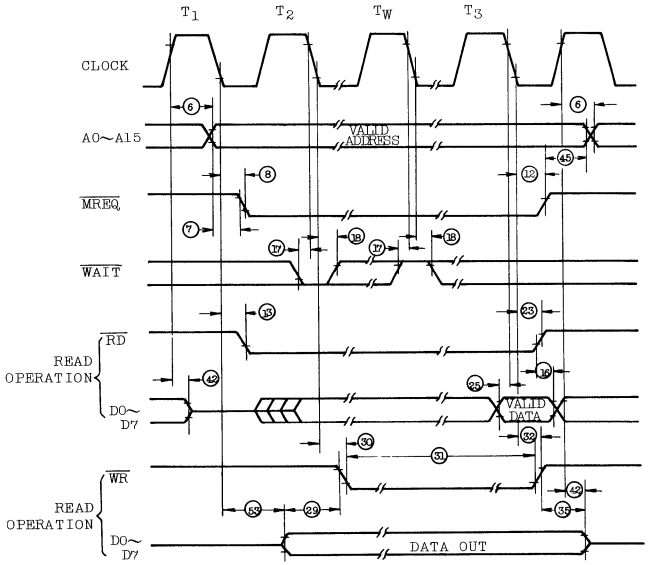
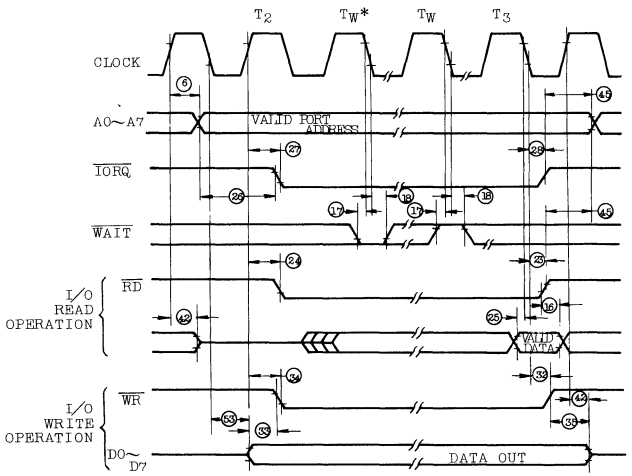


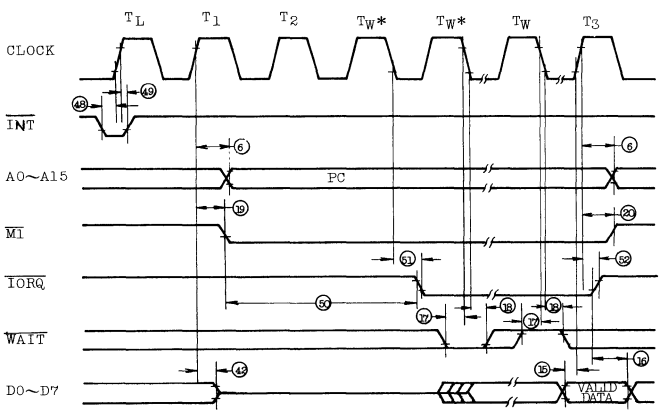
FIGURE 18. MEMORY READ OR WRITE CYCLES

PRELIMINARY



NOTE: T_W^* =One Wait cycle automatically inserted by CPU.

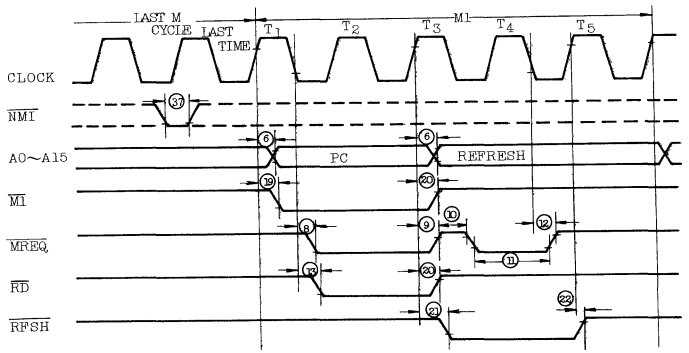
FIGURE 19. INPUT OR OUTPUT CYCLES



NOTE: 1) T_L =Last state of previous instruction.

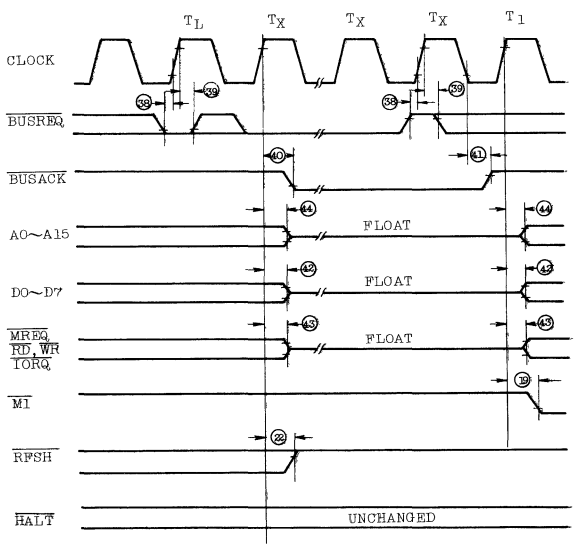
2) Two Wait cycles automatically inserted by CPU(*).

FIGURE 20. INTERRUPT REQUEST/ACKNOWLEDGE CYCLE



* Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding T_{LAST}.

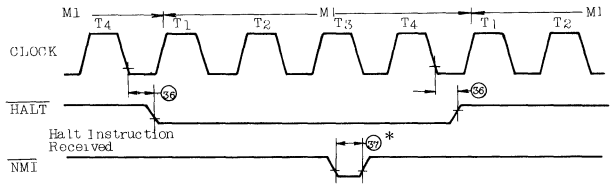
FIGURE 21. NON-MASKABLE INTERRUPT REQUEST OPERATION



NOTE: T_L = Last state of any M cycle. T_X = An arbitrary clock cycle used by requesting device.

FIGURE 22. BUS REQUEST/ACKNOWLEDGE CYCLE

PRELIMINARY



NOTE: $\overline{INT}$ will also force a Halt exit.

* See note, Figure 19.

FIGURE 23. HALT ACKNOWLEDGE CYCLE

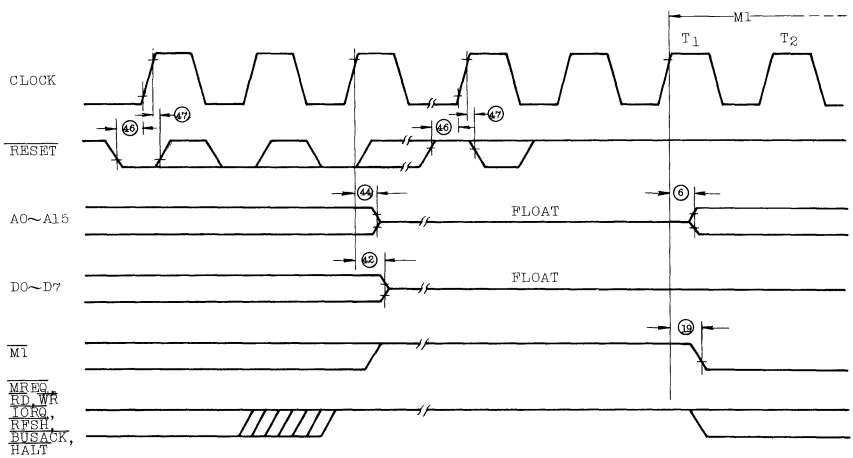


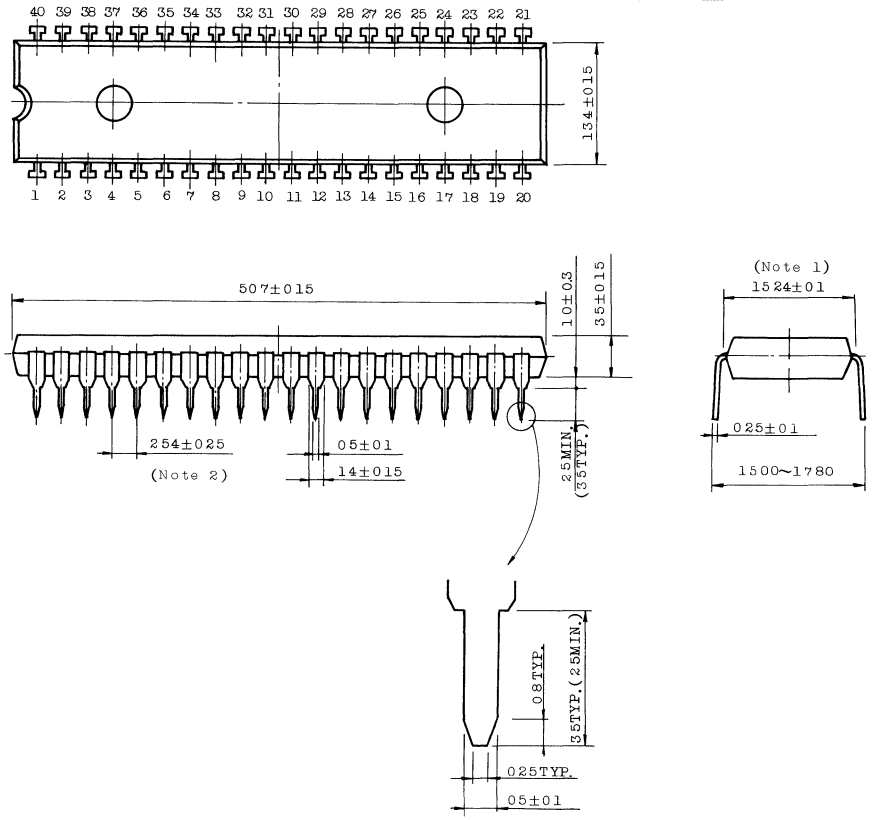
FIGURE 24. RESET CYCLE

PRELIMINARY

OUTLINE DRAWING

Plastic Pakcage

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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

T6497

SILICON MONOLITHIC CMOS SILICON GATE

PRELIMINARY

CMOS CLOCK GENERATOR/CONTROLLER FOR CMOS Z80®

GENERAL DESCRIPTION

The T6497 is a clock generator/controller for Toshiba CMOS Z80 microprocessor (TMPZ84C00P) and peripheral devices. The T6497 has two inputs for choosing one of three modes. When CPU executes HALT instruction, T6497 enters to one of three states described below.

(1) RUN MODE

The T6497 is always providing the clock (CLK) to Z80 CPU and peripheral devices. (CPU is actually in HALT state and executes NOP instruction until an interrupt signal or a reset signal is recognized).

(2) IDLE MODE

The T6497 stops providing the clock. However only the internal oscillator continues its operation.

(3) STOP MODE

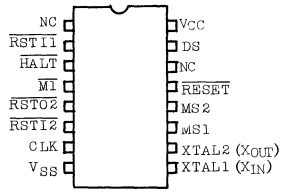
The T6497 stops its operation.

In STOP MODE, CMOS Z80 microcomputer system may stop its operation, so that power consumption to maintain microcomputer system will be extremely reduced.
An interrupt signal ($\overline{NMI}$ or $\overline{INT}$) or a reset signal ($\overline{RESET}$) makes CPU terminate HALT states. The T6497 is fabricated with Toshiba C^2MOS Silicon Gate Technology and molded in 16-pin standard dual-in-line plastic package.

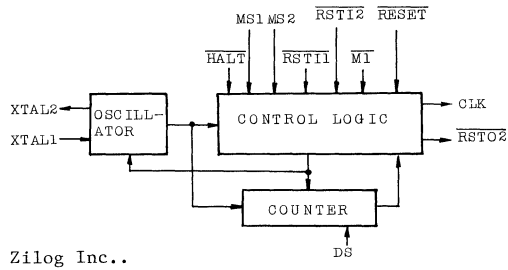
FEATURES

- Toshiba CMOS Z80 Compatible
- 5 Volt Single Power Supply
  5V±10%
- Selectable Three Modes
  RUN MODE
  IDLE MODE
  STOP MODE
- Low Power Consumption
  2mA Typ. @5V @4MHz
  500μA Typ. @5V @4MHz(IDLE MODE)
  10μA Max. @5V(STOP MODE)
- Extended Operating Temperature Range
  -40°C to 85°C

PIN CONNECTIONS (TOP VIEW)



BLOCK DIAGRAM



® Z80 is a trademark of Zilog Inc..

PRELIMINARY

PIN NAMES AND PIN DESCRIPTION

Pin Name	No. of Pins	I/O, 3-state	Description
MS1, MS2	2	Input	Input for Mode select.
XTAL1, XTAL2	2	Input	Terminals for a crystal.
$\overline{RSTI1}$	1	Input	Input to resume the CLK. (Level trigger) Usually input for $\overline{INT}$ request.
$\overline{RSTI2}$	1	Input	Input with a latch to resume the CLK. (Edge trigger) Usually input for $\overline{NMI}$ request.
$\overline{RSTO2}$	1	Output	Output corresponding to $\overline{RSTI2}$. Usually output for $\overline{NMI}$ terminal of CPU.
$\overline{M1}$	1	Input	Input for $\overline{M1}$ signal from CPU.
$\overline{HALT}$	1	Input	Input for $\overline{HALT}$ signal from CPU.
$\overline{RESET}$	1	Input	Input signal to resume the CLK. Usually input for $\overline{RESET}$ signal.
CLK	1	Output	Clock output. When HALT instruction is executed by Z80 CPU in either IDLE MODE or STOP MODE, CLK is kept a low level.
DS	1	Input	Input for selecting the number of counter stage. It is used to determine warming-up time when T6497 restarts from STOP MODE.
NC	2	-	No connection
V_{CC}	1	Power	Single 5V power supply.
V_{SS}	1	Power	Ground reference.

PRELIMINARY

FUNCTIONAL DESCRIPTION

Table 1 illustrates mode select and those functions.
There are two modes (IDLE and STOP) effective when HALT instruction is executed by Z80 CPU. The T6497 continuously provides the system clock (CLK) to Z80 CPU and peripherals unless HALT instruction is executed. In Idle Mode or Stop mode, $\overline{RSTI1}$, $\overline{RSTI2}$ or $\overline{RESET}$ makes the T6497 resume the CLK.

MS1	MS2	MODE	FUNCTIONS
1	1	RUN	Always provides the system clock (CLK).
0	(Note) X	IDLE	Stops the system clock (CLK), but keeps the oscillator operation. The CLK is kept low in this mode.
1	0	STOP	Stops all the internal operation and the CLK is kept low.

Note) X= Don't care

TABLE 1. OPERATION MODES

1. HALT OPERATION IN EACH MODE

(1) RUN MODE (MS1=1, MS2=1)

Figure 1 shows a basic timing when HALT instruction is executed. When Z80 CPU fetches the OPcode of HALT instruction (76H) from program memory, $\overline{HALT}$ signal goes active ("0" level) at the timing synchronized with the falling edge of T4 clock cycle and it shows that Z80 CPU is in the HALT state. In this mode, T6497 always provides the CLK.

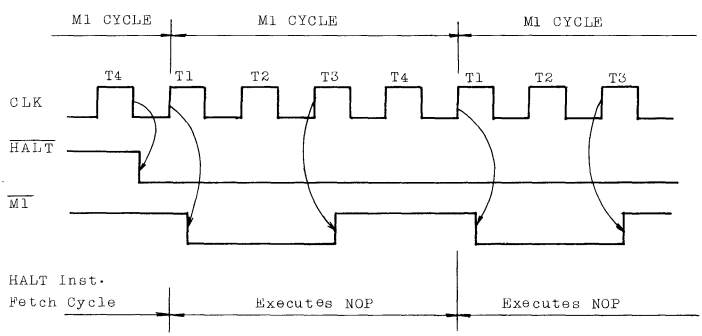


FIGURE 1. RUN MODE

PRELIMINARY

(2) IDLE MODE (MS1=0, MS2= don't care)

Figure 2 shows a basic timing when HALT instruction is executed in Idle Mode. When Z80 CPU fetches the OPcode of HALT instruction (76H) from program memory, $\overline{HALT}$ signal goes active at the timing synchronized with the falling edge of T4 clock cycle and it shows that Z80 CPU is in the HALT state.

T6497 stops providing the CLK at low level state during the T4 clock cycle of the following machine cycle next to OPcode fetch cycle of HALT instruction.

A rising edge of $\overline{M1}$ signal during active $\overline{HALT}$ signal makes the T6497 stop the CLK.

However the internal oscillator continuously works.

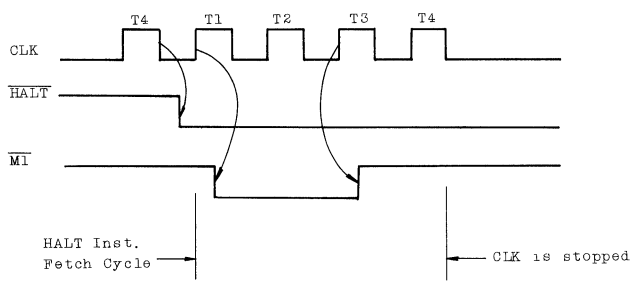


FIGURE 2. CLK STOP SEQUENCE IN IDLE/STOP MODE

(3) STOP MODE (MS1=1, MS2=0)

The same function as IDLE MODE is implemented when $\overline{HALT}$ instruction is executed. (See figure 2.)
Only difference from IDEL MODE is that the T6497 completely stops its operation.

PRELIMINARY

2. CLK RESTART SEQUENCE

There are three inputs to resume the CLK.
$\overline{RSTI1}$ (level trigger), $\overline{RSTI2}$ (edge trigger) or $\overline{RESET}$ (level trigger) can be used.

(1) IDLE MODE

Figure 3 shows the sequence to resume the CLK in IDLE MODE. IN IDLE MODE, the CLK will resume in small delay when a signal to terminate is generated as the internal oscillator is working.

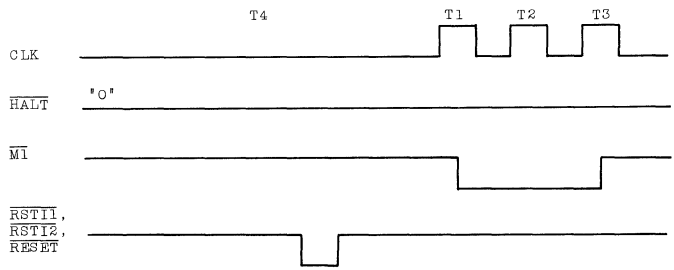


FIGURE 3. CLK RESTART SEQUENCE IN IDLE MODE.

(2) STOP MODE

Figure 4 shows the sequence to resume the CLK in STOP MODE. As the T6497 needs warming-up time to stabilize the frequency it uses the counter when a restart signal is generated.

DS (Divider Select) input must be used to determine warming-up time. External crystal frequency is devided by either 2^{17} or 2^{14}.

Figure 5 shows the block diagram regarding counter and Table 2 illustrates the warming-up time.

PRELIMINARY

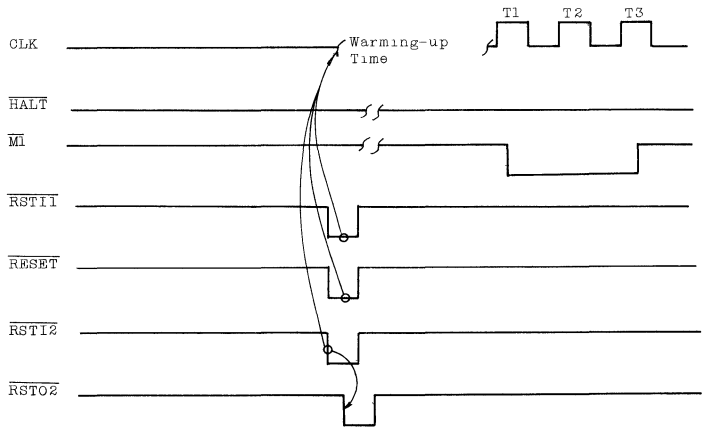


FIGURE 4. CLK RESTART SEQUENCE IN STOP MODE.

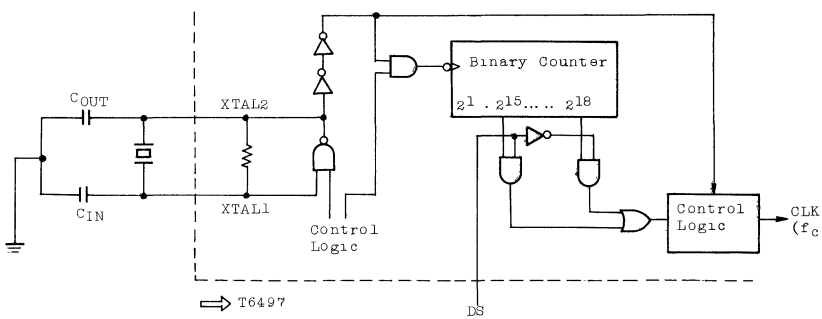


FIGURE 5. BLOCK DIAGRAM of COUNTER and CONTROL LOGIC.

DS	Counter Output	Warming-up Time		
		f_{XTAL}=4MHz	f_{XTAL}=2.5MHz	f_{XTAL}=400kHz
0	2^{18}	≒ 32.8ms	≒ 52.4ms	≒ 328ms
1	2^{15}	≒ 4 ms	≒ 6.6ms	≒ 40ms

TABLE 2. WARMING-UP TIME IN STOP MODE.

PRELIMINARY

Note 1)

Note that either interrupt input or $\overline{\text{RESET}}$ input must be generated to terminate the HALT state of Z80 CPU, where CLK is stopped at a low level during T4 state, in either IDLE MODE or STOP MODE.

(1) In case of $\overline{\text{RESET}}$ input signal is connected with both Z80 CPU $\overline{\text{RESET}}$ terminal and T6497 $\overline{\text{RESET}}$ terminal

$\overline{\text{RESET}}$ input signal to Z80 CPU must be kept active (Low) during at least three clock cycles. When $\overline{\text{RESET}}$ input signal goes inactive, CPU fetches the first OPcode from address 0000H after at least two dummy clock cycles. Thus CPU will terminate HALT state.

Note that if $\overline{\text{RESET}}$ input is connected with both Z80 CPU $\overline{\text{RESET}}$ terminal and T6479 $\overline{\text{RESET}}$ terminal, the $\overline{\text{RESET}}$ signal should be active for enough period to reset the Z80 CPU surely at power on reset. (See Figure 6.)

(2) In case of using an interrupt signal

Figure 7 shows the timing to resume the CLK and to terminate HALT state by an interrupt signal.
$\overline{\text{RSTI1}}$ or $\overline{\text{RSTI2}}$ input makes T6497 resume the CLK. And then an interrupt signal ($\overline{\text{INT}}$ or $\overline{\text{NMI}}$) must be generated to terminate HALT state.
Note that Z80 CPU in HALT state executes NOP instruction unless an interrupt is recognized.

a) In case of using $\overline{\text{NMI}}$

$\overline{\text{NMI}}$ of Z80 CPU is an input (edge trigger) with a latch. If active (low) $\overline{\text{NMI}}$ signal is accepted prior to sampling timing for an interrupt request signal, Z80 CPU recognizes $\overline{\text{NMI}}$. $\overline{\text{RSTI2}}$ of T6497 may be used as $\overline{\text{NMI}}$ input, since $\overline{\text{RSTI2}}$ has a latch and $\overline{\text{RSTO2}}$ may be connected with $\overline{\text{NMI}}$ input of Z80 CPU.

b) In case of using $\overline{\text{INT}}$

In maskable interrupt ($\overline{\text{INT}}$), interrupt enable flip flop (IFF) must be set by software before receiving an interrupt signal.

Figure 7 shows the timing when an interrupt signal is connected with both $\overline{\text{RSTI1}}$ terminal of T6497 and $\overline{\text{INT}}$ terminal of Z80 CPU.

Note 2)

The internal counter of T6497 to determine warming-up time is not used in stop mode when $\overline{\text{RESET}}$ input is activated to resume the clock, so Z80 CPU may not restart properly due to unstable clock when the oscillator restarts. Therefore connect $\overline{\text{RESET}}$ input of T6497 with that of Z80 CPU when $\overline{\text{RESET}}$ input of T6497 is used to restart the clock in stop mode. Also it is suggested that $\overline{\text{RESET}}$ input be kept low for enough period to initialize Z80 CPU.

PRELIMINARY

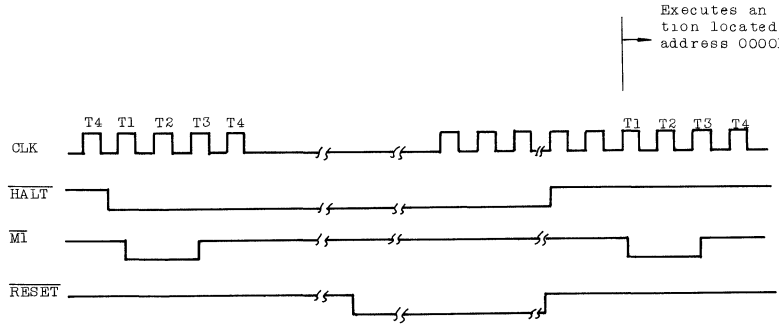


FIGURE 6. EXAMPLE of RESUMING CLK by $\overline{\text{RESET}}$

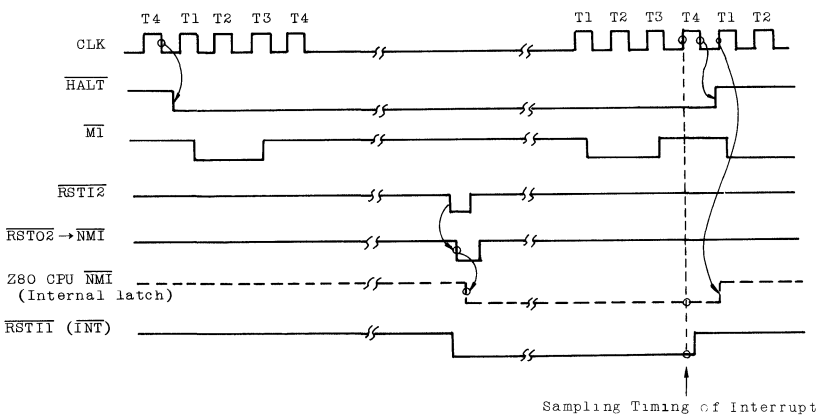


FIGURE 7. EXAMPLE of RESUMING CLK by $\overline{\text{RSTI1}}$ or $\overline{\text{RSTI2}}$

PRELIMINARY

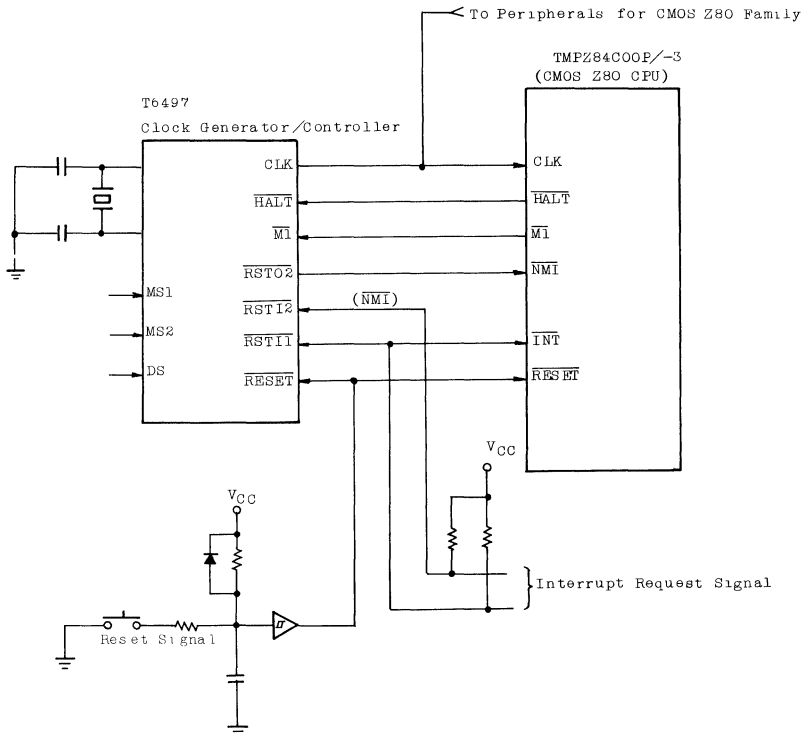


FIGURE 8. EXAMPLE of CONNECTION with Z80 CPU.

PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage with respect to V_{SS}	-0.5V to 7.0V
V_{IN}	Input Voltage	-0.5V to V_{CC}+0.5V
I_{IN}	Input Current	±10mA
P_D	Power Dissipation (TA=85°C)	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

DE CHARACTERISTICS (I)

TA=-40°C to 85°C, V_{CC}=5±10% , V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage (except XTAL1,2)		-0.5	-	0.8	V
V_{IH}	Input High Voltage (except XTAL1,2)		2.2	-	V_{CC}	V
V_{OLC}	Output Low Voltage (CLK)	I_{OL}=2.0mA	-	-	0.4	V
V_{OL}	Output Low Voltage (except CLK)	I_{OL}=2.0mA	-	-	0.4	V
V_{OHC}	Output High Voltage (CLK)	I_{OH}=-250μA	V_{CC}-0.6	-	-	V
V_{OH1}	Output High Voltage (except CLK)	I_{OH}=-1.6mA	2.4	-	-	V
V_{OH2}	Output High Voltage (except CLK)	I_{OH}=-250μA	V_{CC}-0.8	-	-	V
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±1	μA
I_{OL}	Outptu Leakage	V_{SS}+0.4V$\leq V_{IN} \leq V_{CC}$	-	-	±1	μA
I_{CC1}	V_{CC} Supply Current (NORMAL/RUN MODE)	V_{CC}=5V fXTAL=4MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	2	4	mA
I_{CC2}	V_{CC} Supply Current (STOP MODE)	V_{CC}=5V V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	0.3	10	μA
I_{CC3}	V_{CC} Supply Current (IDLE MODE)	V_{CC}=5V fXTAL=4MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	0.5	1	mA

PRELIMINARY

A.C. CHARACTERISTICS TA=-40°C to 85°C, V_{CC}=5V±10%, V_{SS}=0V

NUMBER	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
1	TcC	CLK Cycle Time		250	-	-	ns
2	TwCh	CLK Pulse Width (High)		110	-	-	ns
3	TwCl	CLK Pulse Width (Low)		110	-	-	ns
4	TrC	CLK Rise Time		-	-	15	ns
5	TfC	CLK Fall Time		-	-	15	ns
6	TsHALT(Mlr)	$\overline{HALT}$ Setup Time to Ml↑		10	-	-	ns
7	TwRSTIl	$\overline{RSTIl}$ Pulse Width (Low)		80	-	-	ns
8	TwRSTI2	$\overline{RSTI2}$ Pulse Width (Low)		200	-	-	ns
9	TdRSTO2 (RSTI2f)	$\overline{RSTI2}$↓ to $\overline{RSTO2}$↓ Delay		-	-	100	ns
10	TwRSTO2	$\overline{RSTO2}$ Pulse Width (Low)	C_L=100pF	80	-	-	ns
11	TwRESET	$\overline{RESET}$ Pulse Width (Low)		80	-	-	ns
12	TRSTlS	CLK Restart Delay by DS=0 $\overline{RSTIl}$ (Stop Mode) DS=1		- -	$(2^{17}+2.5)TcC$ $(2^{14}+2.5)TcC$	- -	ns ns
13	TRST2S	CLK Restart Delay by DS=0 $\overline{RSTI2}$ (Stop Mode) DS=1		- -	$(2^{17}+2.5)TcC$ $(2^{14}+2.5)TcC$	- -	ns ns
14	TRSTlI	CLK Restart Dealy by $\overline{RSTIl}$ (Idel Mode)		-	2.5 TcC	-	ns
15	TRST2I	CLK Restart Delay by $\overline{RSTI2}$ (Idel Mode)		-	2.5 TcC	-	ns
16	TRESETI	CLK Restart Dealy by $\overline{RESET}$ (Idle Mode)		-	1 TcC	-	ns

NOTE) A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.4V for a logic "0". Timing measurements are made at 2.2V for a logic "1" and 0.8V for a logic "0" except CLK output. CLK is made at V_{CC} - 0.6V for a logic "1" and 0.4V for a logic "0".

PRELIMINARY

TIMING WAVEFORMS

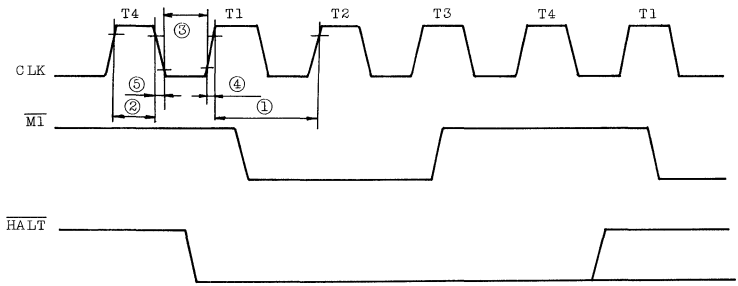


FIGURE 9. CLK TIMING

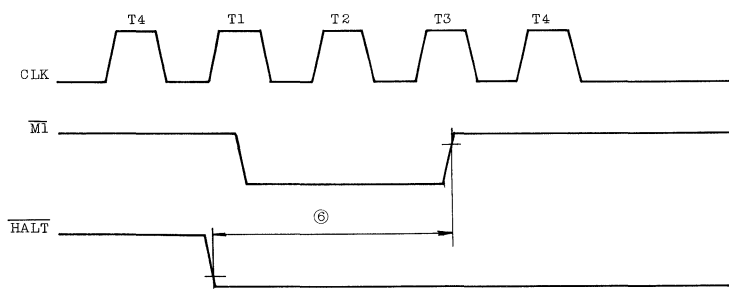


FIGURE 10. CLK STOP TIMING (IDLE/STOP MODE)

PRELIMINARY

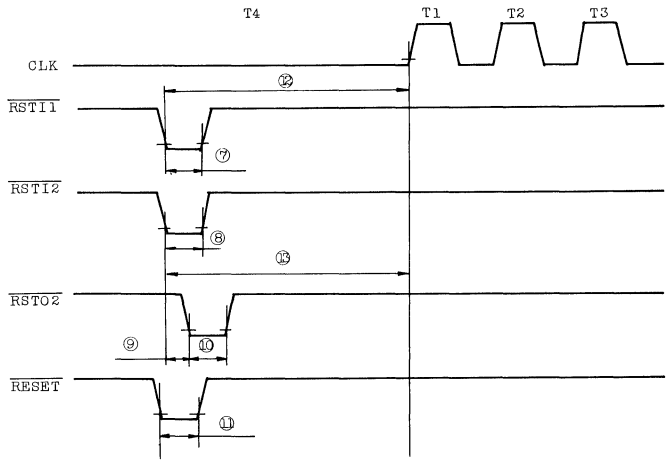


FIGURE 11. CLK RESTART TIMING IN STOP MODE

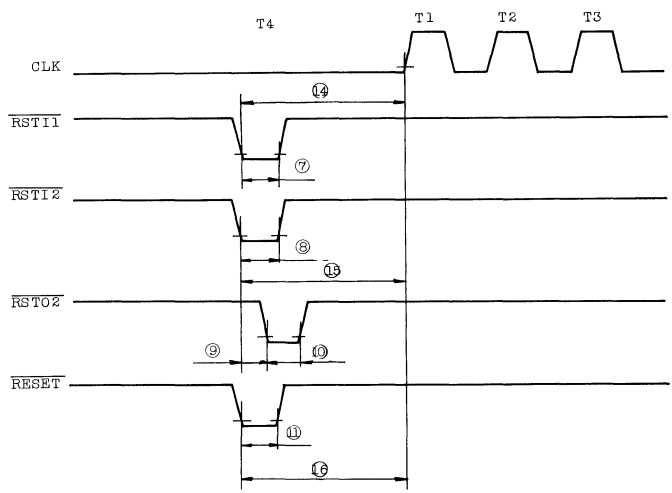


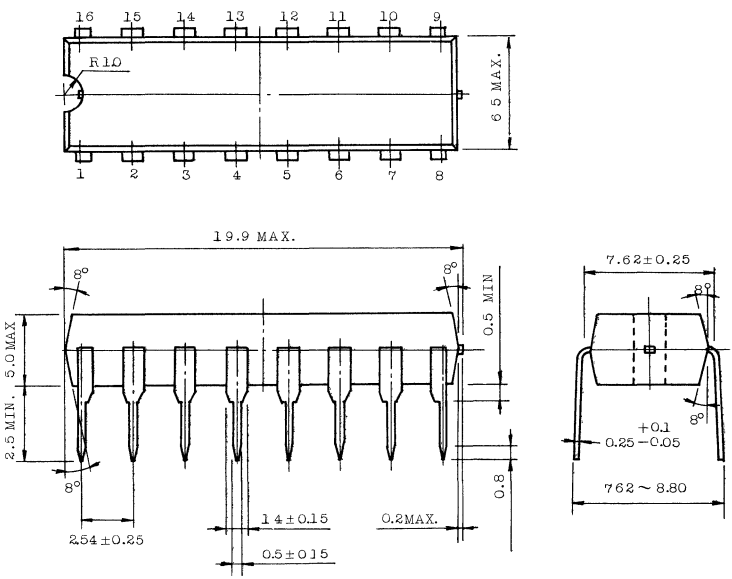
FIGURE 12. CLK RESTART TIMING IN IDLE MODE

PRELIMINARY

OUTLINE DRAWINGS

Plastic Package

Unit in mm



Note: Each lead pitch is 2.54mm. All leads are located within 0.25mm of their true longitudinal position with pespect to No.1 and No.16 leads.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TMPZ84C10P

ADVANCE INFORMATION

CMOS Z80® DAM: Direct Memory Access Controller

GENERAL DESCRIPTION

The TMPZ84C10P, (from here on referred to as Z80 DMA), is CMOS version of Z80 DMA (Direct Memory Access Controller) which provides low power, powerful and versatile operation. It is designed to improve system performance by allowing external devices to directly transfer data from the system memory. Memory-to-memory and I/O-to-I/O operations capability is also provided. The device is fabricated with Toshiba's C^2MOS Silicon Gate Technology.

FEATURES

- Z80 Compatible DMA
- DC to 4MHz Operation
- Single 5V Power Supply

  4MHz @ 5V ± 10%
- 2M Bytes/Sec. Data Rate @ 4MHz
- 64K Byte Max. Block Length
- Dual Port Address Generation with Incrementing, Decrementing, or Fixed Address in Both Ports
- Low Power Consumption

  Less than 10μA @ 5V (Power down)
- Extended Operating Temperature

  -40°C to 85°C
- Transfer, Search, or Transfer/Search Operations in Byte-at-a Time, Burst or Continuous Modes.
- Bit-maskable Byte Searching
- Z80 Compatible Daisy - Chain Interrupt Structure

® Z80 is a trademark of Zilog Inc.

ADVANCE INFORMATION

PIN CONNECTIONS (TOP VIEW)

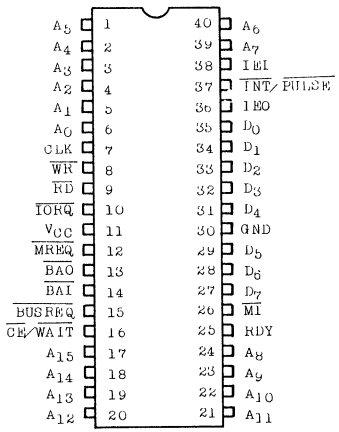


FIGURE 1. PINOUT DIAGRAM

BLOCK DIAGRAM

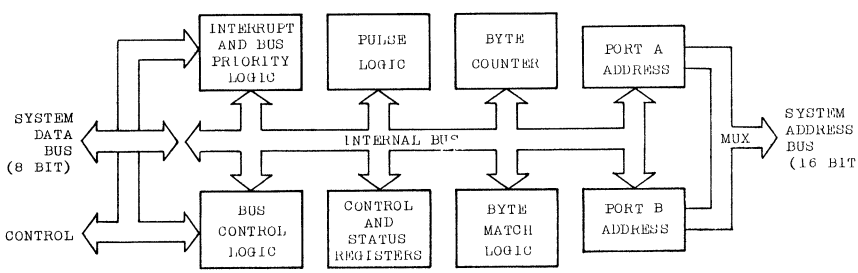


FIGURE 2. BLOCK DIAGRAM

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMPZ84C20P

SILICON MONOLITHIC

CMOS SILICON GATE

PRELIMINARY

CMOS Z80® PIO: Parallel Input/Output Controller

GENERAL DESCRIPTION

The TMPZ84C20P (from here on referred to as PIO), is CMOS version of Z80 PIO and have been designed to provide low power operation. The PIO is a programmable, dual-port device that provides a direct interface between Z80 microcomputer systems and peripheral devices. Also all logic necessary to implement a fully nested interrupt structure is included in the PIO.

The device is fabricated with Toshiba's C^2MOS Silicon Gate Technology.

FEATURES

- Z80 Compatible PIO
- DC to 4 MHz Operation
- Single 5V Power Supply

  4 MHz @ 5V ± 10%
- Four Programmable Operation Modes
  - Byte Input
  - Byte Output
  - Byte Input/Output (Port A Only)
  - Bit Input/Output
- Low Power Consumption

  2mA Typ. @ 4 MHz @ 5V

  Less than 10μA @ 5V (Power down)
- Extended Operating Temperature

  - 40°C to 85°C
- Two Independent 8-bit Bidirectional ports
- Interrupt-driven Handshake for Fast Response
- Eight Outputs of Port B Capable of Driving Darlington Transistors
- Z80 Compatible Daisy-chain Interrupt Structure

Z80® is a trademark of Zilog Inc.

PRELIMINARY

PIN CONNECTIONS (TOP VIEW)

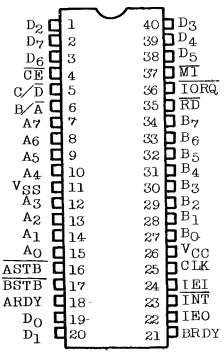


FIGURE 1. PINOUT DIAGRAM

BLOCK DIAGRAM

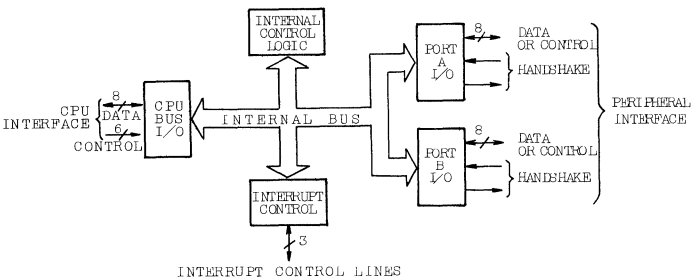


FIGURE 2. BLOCK DIAGRAM

PIN NAMES AND PIN DESCRIPTION

A_0-A_7. Port A Bus (bidirectional, 3-state).

This 8-bit bus transfers data, status, or control information between Port A of the PIO and a peripheral device. A_0 is the least significant bit of the Port A data bus.

ARDY. Register A Ready (output, active High)

The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. This signal goes active to indicate that the Port A output register has been loaded and the peripheral data bus is stable and ready for transfer to the peripheral device.

Input Mode. This signal is active when the Port A input register is empty and ready to accept data from the peripheral device.

Bidirectional Mode. This signal is active when data is available in the Port A output register for transfer to the peripheral device. In this mode, data is not placed on the Port A data bus, unless $\overline{\text{ASTB}}$ is active.

Control Mode. This signal is disabled and forced to a Low state.

$\overline{\text{ASTB}}$. Port A Strobe Pulse From Peripheral Device (input, active Low)

The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO.

Input Mode. The strobe is issued by the peripheral to load data from the peripheral into the Port A input register. Data is loaded into the PIO when this signal is active.

Bidirectional Mode. When this signal is active, data from the Port A output register is gated onto the Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data.

Control Mode. The strobe is inhibited internally.

B_0-B_7. Port B Bus (bidirectional, 3-state)

This 8-bit bus transfers data, status, or control information between Port B and a peripheral device. The Port B data bus can supply 1.5 mA at 1.5V to drive Darlington transistors. B_0 is the least significant bit of the bus.

B/$\overline{A}$. Port B Or A Select (input, High = B).

This pin defines which port is accessed during a data transfer between the CPU and the PIO. A Low on this pin selects Port A; a High selects Port B. Often address bit A_0 from the CPU is used for this selection function.

BRDY. Register B Ready (output, active High).

This signal is similar to ARDY, except that in the Port A bidirectional mode this signal is High when the Port A input register is empty and ready to accept data from the peripheral device.

$\overline{BSTB}$. Port B Strobe Pulse From Peripheral Device (input, active Low)

This signal is similar to $\overline{ASTB}$, except that in the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

$C/\overline{D}$. Control Or Data Select (input, High = C)

This pin defines the type of data transfer to be performed between the CPU and the PIO. A High on this pin during a CPU write to the PIO causes the Z80 data bus to be interpreted as a command for the port selected by the $B/\overline{A}$ Select line. A Low on this pin means that the Z80 data bus is being used to transfer data between the CPU and the PIO. Often address bit A_1 from the CPU is used for this function.

$\overline{CE}$. Chip Enable (input, active Low).

A Low on this pin enables the PIO to accept command or data inputs from the CPU during a write cycle or to transmit data to the CPU during a read cycle. This signal is generally decoded from four I/O port numbers for Ports A and B, data, and control.

CLK. System Clock (input)

The Z80 PIO uses the standard single-phase Z80 system clock. When CLK is a DC state (either a high or low level), PIO stops its operation and maintains registers and control signals so that power consumption is extremely reduced.

D_0-D_7. Z80 CPU Data Bus (bidirectional, 3-state)

This bus is used to transfer all data and commands between the Z80 CPU and the Z80 PIO. D_0 is the least significant bit.

IEI. Interrupt Enable In (input, active High).

This signal is used to form a priority-interrupt daisy chain when more than one interrupt-driven device is being used. A High level on this pin

indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.

IEO. Interrupt Enable Out (output, active High)

The IEO signal is the other signal required to form a daisy chain priority scheme. It is High only if IEI is High and the CPU is not servicing an interrupt from this PIO. Thus this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{INT}$. Interrupt Request (output, open drain, active Low)

When $\overline{INT}$ is active the Z80 PIO is requesting an interrupt from the Z80 CPU.

$\overline{IORQ}$. Input/Output Request (input from Z80 CPU, active Low)

$\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{RD}$ to transfer commands and data between the Z80 CPU and the Z80 PIO. When $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are active, the port addressed by B/$\overline{A}$ transfers data to the CPU (a read operation). Conversely, when $\overline{CE}$ and $\overline{IORQ}$ are active but $\overline{RD}$ is not, the port addressed by B/$\overline{A}$ is written into from the CPU with either data or control information, as specified by C/$\overline{D}$. Also, if $\overline{IORQ}$ and $\overline{M1}$ are active simultaneously, the CPU is acknowledging an interrupt; the interrupting port automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

$\overline{M1}$. Machine Cycle (input from CPU, active Low)

This signal is used as a sync pulse to control several internal PIO operations. When both the $\overline{M1}$ and $\overline{RD}$ signals are active, the Z80 CPU is fetching an instruction from memory. Conversely, when both $\overline{M1}$ and $\overline{IORQ}$ are active, the CPU is acknowledging an interrupt. In addition, $\overline{M1}$ has two other functions within the Z80 PIO: it synchronizes the PIO interrupt logic; when $\overline{M1}$ occurs without an active $\overline{RD}$ or $\overline{IORQ}$ signal, the PIO is reset. $\overline{M1}$ must be active for a minimum of two clock cycles to reset PIO.

$\overline{RD}$. Read Cycle Status (input from Z80 CPU, active Low)

If $\overline{RD}$ is active, or an I/O operation is in progress, $\overline{RD}$ is used with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{IORQ}$ to transfer data from the Z80 PIO to the Z80 CPU.

V_{CC}. Power Supply

+5V.

VSS. Power Supply

Ground Reference. (0V).

PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage with respect to V_{SS}	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to V_{CC}+0.5V
P_D	Power Dissipation (TA=85°C)	250 mW
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

DC CHARACTERISTICS (I)

T_A = -40°C to 85°C, V_{CC} = 5V ± 10%, V_{SS} = 0V, Unless otherwise noted.

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{ILC}	Clock Input Low Voltage		-0.3	-	0.6	V
V_{IHC}	Clock Input High Voltage		V_{CC}-0.6	-	V_{CC}+0.3	V
V_{IL}	Input Low Voltage (except CLK)		-0.5	-	0.8	V
V_{IH}	Input High Voltage (except CLK)		2.2	-	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL}=2.0 mA	-	-	0.4	V
V_{OH1}	Output High Voltage (1)	I_{OH}=-1.6 mA	2.4	-	-	V
V_{OH2}	Output High Voltage (2)	I_{OH}=-250 µA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	µA
I_{LO}	3-State Output Leakage Current in Float	V_{SS}+0.4≤V_{OUT}≤V_{CC}	-	-	±10	µA
I_{CC1}	Operating Supply Current	V_{CC}=5V, CLK=4MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	2	5	mA
I_{CC2}	Stand-by Supply Current	V_{CC}=5V CLK=V_{CC} V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	0.5	10	µA
*I_{OHD}	Darlington Drive Current	V_{OH}=1.5V, R_{EXT}=1.1kΩ	-1.5	-	-5.0	mA

NOTE 1) * Applied to Port B only

2) Typical Value is specified at 25°C.

PRELIMINARY

AC CHARACTERISTICS (4MHz Operation)

T_A = -40°C to 85°C, V_{CC} = 5V± 10%, V_{SS} = 0V, Unless otherwise noted.

No.	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
1	TcC	Clock Cycle Time	C_L=100 pF	250	–	DC	ns
2	TwCh	Clock Width (High)		105	–	DC	ns
3	TwCl	Clock Width (Low)		105	–	DC	ns
4	TfC	Clock Fall Time		–	–	30	ns
5	TrC	Clock Rise Time		–	–	30	ns
6	TsCS(RI)	$\overline{CE}$, B/$\overline{A}$, C/$\overline{D}$ to $\overline{RD}$, $\overline{IORQ}$ ↓ Setup Time		50	–	–	ns
*7	Th	Any Hold Times for Specified Setup Time		40	–	–	ns
8	TsRI(C)	$\overline{RD}$, $\overline{IORQ}$ to Clock ↑ Setup Time		115	–	–	ns
9	TdRI(DO)	$\overline{RD}$, $\overline{IORQ}$ ↓ to Data Out Delay		–	–	380	ns
10	TdRI(DOs)	$\overline{RD}$, $\overline{IORQ}$ ↑ to Data Out Float Delay		–	–	110	ns
11	TsDI(C)	Data In to Clock ↑ Setup Time		50	–	–	ns
12	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)		160	–	–	ns
13	TsM1(Cr)	$\overline{M1}$ ↓ to Clock ↑ Setup Time		90	–	–	ns
14	TsM1(Cf)	$\overline{M1}$ ↑ to Clock ↓ Setup Time (M1 Cycle)		0	–	–	ns
15	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt Immediately Preceding $\overline{M1}$ ↓)		–	–	(1) 190	ns
16	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)		(1) 140	–	–	ns
17	TsIEI(IEOf)	IEI ↓ to IEO ↓ Delay		–	–	130	ns
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED Decode)		–	–	160	ns
19	TcIO(C)	$\overline{IORQ}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)		200	–	–	ns
20	TdC(RDYr)	Clock ↓ to READY ↑ Delay		–	–	190	ns
21	TdC(RDYf)	Clock ↓ to READY ↓ Delay		–	–	140	ns
22	TwSTB (2)	$\overline{STROBE}$ Pulse Width		(2) 150	–	–	ns
23	TsSTB(C)	$\overline{STROBE}$ ↑ to Clock ↓ Setup Time (To Activate READY on Next Clock Cycle)		220	–	–	ns
24	TdIO(PD)	$\overline{IORQ}$ ↑ to PORT DATA Stable Delay (Mode 0)		–	–	180	ns
25	TsPD(STB)	PORT DATA to $\overline{STROBE}$ ↑ Setup Time (Mode 1)		230	–	–	ns
26	TdSTB(PD)	$\overline{STROBE}$ ↓ to PORT DATA Stable (Mode 2)		–	–	210	ns
27	TdSTB(PDr)	$\overline{STROBE}$ ↑ to PORT DATA Float Delay (Mode 2)		–	–	180	ns
28	TdPD(INT)	PORT DATA Match to $\overline{INT}$ ↓ Delay (Mode 3)		–	–	490	ns
29	TdSTB(INT)	$\overline{STROBE}$ ↑ to $\overline{INT}$ ↓ Delay		–	–	440	ns

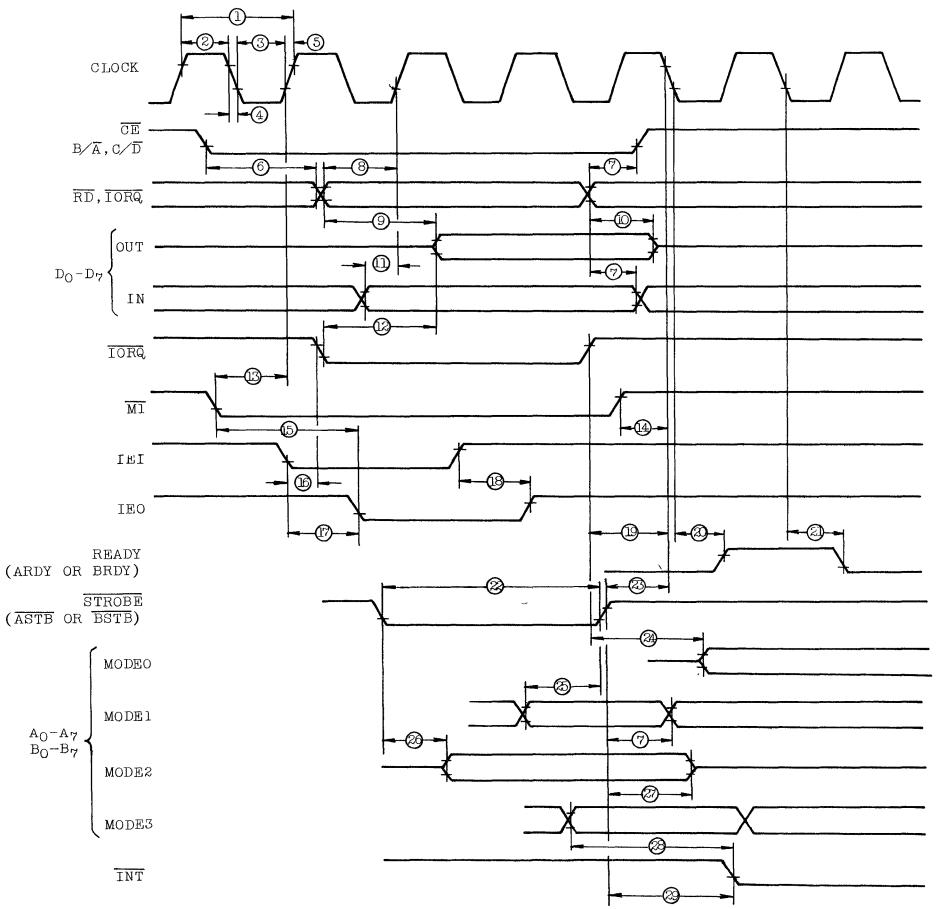
NOTES) (1) 2.5 TcC > (N-2) TdIEI (IEOf) + TdM1 (IEO) + TsIEI(IO) + TTL Buffer Delay, if any.

(2) In Mode 2; TwSTB > TsPD (STB)

(3) Spec No. 7 (Th) with * mark is not compatible with NMOS PIO.

PRELIMINARY

TIMING WAVEFORM



TEST CONDITIONS

AC test Conditions

- Inputs except CLK (clock) are driven at 2.4V for a logic "1" and 0.4V for a logic "0". Clock input is driven at V_{CC}-0.6V for a logic "1" and 0.6V for a logic "0".
- Timing measurements are made at 2.2V for a logic "1" and 0.8V for a logic "0".

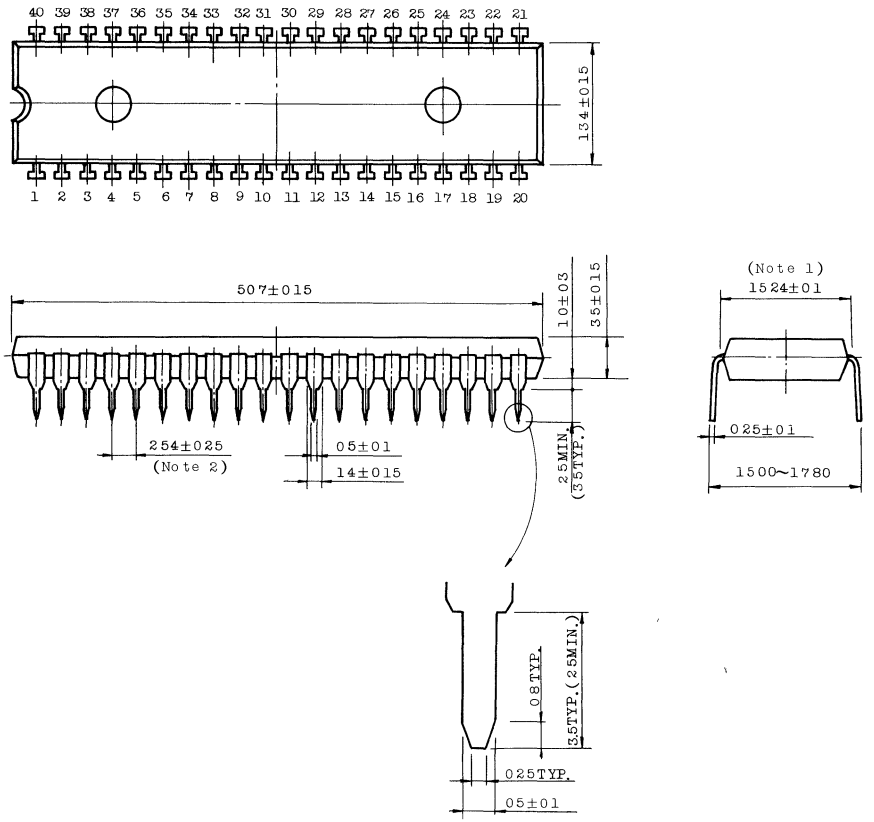
All AC parameters assume a load capacitance of 100 pF.

PRELIMINARY

OUTLINE DRAWING

Plastic 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.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMPZ84C30P

SILICON MONOLITHIC

CMOS SILICON GATE

PRELIMINARY

CMOS Z80 ® CTC : COUNTER/TIMER CIRCUIT

GENERAL DESCRIPTION

The TMPZ84C30P, (from here on referred to as CTC), is CMOS version of Z80 CTC (Counter/Timer circuit) that provides low power operation. The CTC has four independent counter/timer channels, each with a readable downcounter and a selectable 16 or 256 prescaler. Also standard Z80 family daisy-chain interrupt structure is provided. The device is fabricated with Toshiba's C^2MOS Silicon Gate Technology.

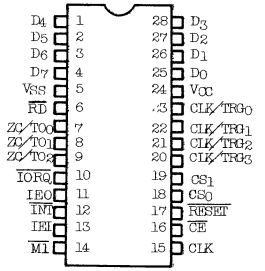
FEATURES

- Z80 Compatible CTC
- DC to 4 MHz Operation
- Single 5 V Power Supply
  4 MHz @ 5 V ±10%
- Low Power Consumption
  3mA Typ. @5V @4MHz
  Less than 10μA @ 5V (Power down)
- Extended Operating Temperature
  -40°C to 85°C
- Four Independent 8-bit Counter/Timer Channels
- More than One Counter Can be Cascaded for Counts greater than 256.
- Selectable in either Counter Mode or Timer Mode for Each Channel.
- Selectable Positive or Negative Trigger Operation
- Three Zero Count/Timeout Outputs Capable of Driving Darlington Transistors
- Z80 Compatible Daisy-chain Interrupt Structure

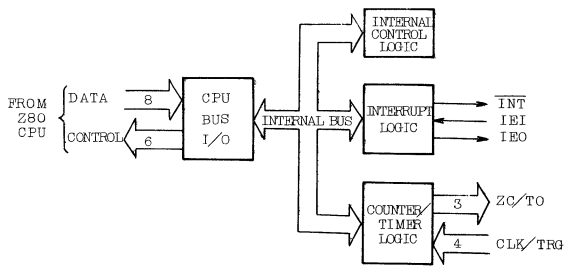
® Z80 is a trademark of Zilog Inc..

PRELIMINARY

PIN CONNECTIONS (TOP VIEW)



BLOCK DIAGRAM



PIN NAMES AND PIN DESCRIPTION

$\overline{CE}$. Chip Enable (input, active Low).

When enabled the CTC accepts control words, interrupt vectors, or time constant data words from the data bus during an I/O write cycle; or transmits the contents of the down-counter to the CPU during an I/O read cycle. In most applications this signal is decoded from the eight least significant bits of the address bus for any of the four I/O port addresses that are mapped to the four counter-timer channels.

CLK. System Clock (input)

Standard single-phase Z80 system clock. When CLK is a DC state (either a high or low level, CTC stops its operation and maintains registers and control signals.)

CLK/TRG_0-CLK/TRG_3. External Clock/Timer Trigger (input, user-selectable active High or Low).

Four pins corresponding to the four Z80 CTC channels. In counter mode, every active edge on this pin decrements the down-counter. In timer mode, an active edge starts the timer.

CS_0-CS_1. Channel Select (inputs active High).

Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to A_0 and A_1).

D_0-D_7. System Data Bus (bidirectional, 3-state).

Transfers all data and commands between the Z80 CPU and the Z80 CTC.

IEI. Interrupt Enable In (input, active High).

A High indicates that no other interrupting devices of higher priority in the daisy chain are being serviced by the Z80 CPU.

IEO. Interrupt Enable Out (output, active High).

High only if IEI is High and the Z80 CPU is not servicing an interrupt from any Z80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

$\overline{INT}$. Interrupt Request (output, open drain, active Low).

Low when any Z80 CTC channel that has been programmed to enable interrupts has a zero-count condition in its down-counter.

$\overline{IORQ}$. Input/Output Request (input from CPU, active Low).

Used with $\overline{CE}$ and $\overline{RD}$ to transfer data and channel control words between the Z80 CPU and the Z80 CTC. During a write cycle, $\overline{IORQ}$ and $\overline{CE}$ are active and $\overline{RD}$ inactive. The Z80 CTC does not receive a specific write signal; rather, it intemally generates its own from the inverse of an active $\overline{RD}$ signal. In a read cycle, $\overline{IORQ}$, $\overline{CE}$ and $\overline{RD}$ are active; the contents of the down-counter are read by the Z80 CPU. If $\overline{IORQ}$ and $\overline{M1}$ are both true, the CPU is acknowledging an interrupt request, and the highest priority interrupting channel places its interrupt vector on the Z80 data bus.

$\overline{M1}$. Machine Cycle One (input from CPU, active Low)

When $\overline{M1}$ and $\overline{IORQ}$ are active, the Z80 CPU is acknowledging an interrupt. The Z80 CTC then places an interrupt vector on the data bus if it has highest priority, and if a channel has requested an interrupt ($\overline{INT}$).

$\overline{RD}$. Read Cycle Status (input, active Low).

Used in conjunction with $\overline{IORQ}$ and $\overline{CE}$ to transfer data and channel control words between the Z80 CPU and the Z80 CTC.

$\overline{RESET}$. Reset (input active Low)

Terminates all down-counts and disables all interrupts by resetting the interrupt bits in all control registers; the ZC/TO and the Interrupt outputs go inactive; IEO reflects IEI; D_0-D_7 go to the high-impedance state. $\overline{RESET}$ must be active for a minimum of 3 clock cycles.

ZC/TO_0-ZC/TO_2. Zero Count/Timeout (output, active High).

Three ZC/TO pins corresponding to Z80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse when the down-counter decrements to zero.

V_{CC}. Power Supply

+5 V

V_{SS}. Power Supply

Ground Reference (0 V)

PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage with respect to V_{SS}	-0.5 V to 7 V
V_{IN}	Input Voltage	-0.5 V to V_{CC}+0.5 V
P_D	Power Dissipation (Ta=85°C)	250 mW
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

DC CHARACTERISTICS (I)

T_A = -40°C to 85°C, V_{CC} = 5 V ±10%, V_{SS} = 0 V

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{ILC}	Clock Input Low Voltage		-0.3	-	0.6	V
V_{IHC}	Clock Input High Voltage		V_{CC}-0.6	-	V_{CC}+0.3	V
V_{IL}	Input Low Voltage (except CLK)		-0.5	-	0.8	V
V_{IH}	Input High Voltage(except CLK)		2.2	-	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL}=2.0mA	-	-	0.4	V
V_{OH1}	Output High Voltage (1)	I_{OH}=-1.6mA	2.4	-	-	V
V_{OH2}	Output High Voltage (2)	I_{OH}=-250μA	V_{CC}-0.8	-	-	V
I_{LI}	Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{LO}	3-State Output Leakage Current in Float	V_{SS}+0.4≤V_{OUT}≤V_{CC}	-	-	±10	μA
I_{CC1}	Operating Supply Current	V_{CC}=5V, CLK=4MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	3	7	mA
I_{CC2}	Stand-by Supply Current	V_{CC}=5V CLK=V_{CC} V_{IH}=V_{CC}-0.2V V_{IL}=0.2V	-	0.5	10	μA
I_{OHD}	Darlington Drive Current(1)	V_{OH}=1.5V,R_{EXT}=1.1kΩ	-1.5	-	-5.0	mA

Note 1) Applied to ZC/TO_0, ZC/TO_1 and ZC/TO_2.

PRELIMINARY

AC CHARACTERISTICS (4MHz Operation)

T_A = -40°C to 85°C, V_{CC} = 5V ± 10%, V_{SS} = 0 V, Unless otherwise noted.

NUMBER	SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
1	TcC	Clock Cycle Time		250	-	DC	ns
2	TwCH	Clock Width (High)		105	-	DC	ns
3	TwCl	Clock Width (Low)		105	-	DC	ns
4	TfC	Clock Fall Time		-	-	30	ns
5	TrC	Clock Rise Time		-	-	30	ns
6	Th	All Hold Times		0	-	-	ns
7	TsCS(C)	CS to Clock ↑ Setup Time		160	-	-	ns
8	TsCE(C)	$\overline{CE}$ to Clock ↑ Setup Time		150	-	-	ns
9	TsIO(C)	$\overline{IORQ}$ ↓ to Clock ↑ Setup Time		115	-	-	ns
10	TsRD(C)	$\overline{RD}$ ↓ to Clock ↑ Setup Time		115	-	-	ns
11	TdC(DO)	Clock ↑ to Data Out Delay		-	-	200	ns
12	TdC(DOz)	Clock ↓ to Data Out Float Delay		-	-	110	ns
13	TsDI(C)	Data In to clock ↑ Setup Time		50	-	-	ns
14	TsMl(C)	$\overline{M1}$ to Clock ↑ Setup Time		90	-	-	ns
15	TdMl(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (Interrupt immediately preceding $\overline{M1}$)	C_L=100pF	-	-	190	ns
16	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTA Cycle)		-	-	160	ns
17	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		-	-	130	ns
18	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (After ED Decode)		-	-	160	ns
19	TdC(INT)	Clock ↑ to $\overline{INT}$ ↓ Delay		-	-	(1)TcC +140	ns
20	TdCLK(INT)	CLK/TRG ↑ to $\overline{INT}$ ↓ tsCTR(C) satisfied tsCTR(C) not satisfied		 - -	 - -	(2) TcC+160 2TcC+370	 ns ns
21	TcCTR	CLK/TRG Cycle Time		(2) 2TcC	-	-	ns
22	TrCTR	CLK/TRG Rise Time		-	-	50	ns
23	TfCTR	CLK/TRG Fall Time		-	-	50	ns
24	TwCTRl	CLK/TRG Width (Low)		200	-	-	ns

PRELIMINARY

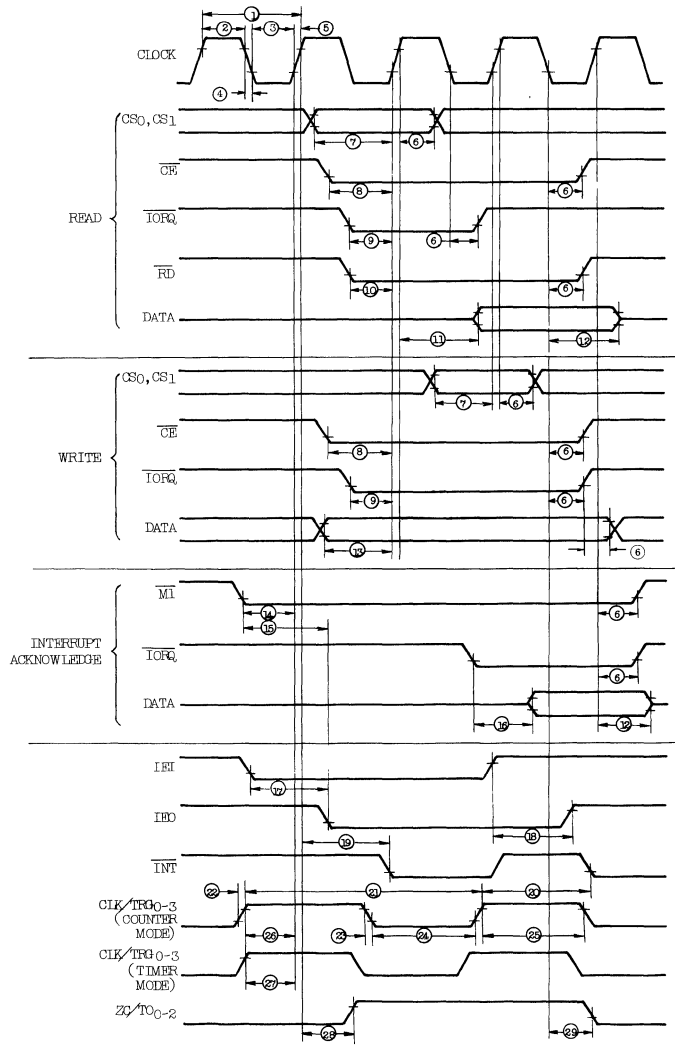
NUMBER	SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
25	TwCTRh	CLK/TRG Width (High)		200	-	-	ns
26	TsCTR(Cs)	CLK/TRG ↑ to Clock ↑ Setup Time for Immediate Count		(2) 210	-	-	ns
27	TsCTR(Ct)	CLK/TRG ↑ to Clock ↑ Setup Time for enabling of Prescaler on following clock ↑		(1) 210	-	-	ns
28	TdC(ZC/TOr)	Clock ↑ to ZC/TO ↑ Delay		-	-	190	ns
29	TdC(ZC/TOf)	Clock ↓ to ZC/TO ↓ Delay		-	-	190	ns

NOTES) (1) Timer mode

(2) Counter mode

PRELIMINARY

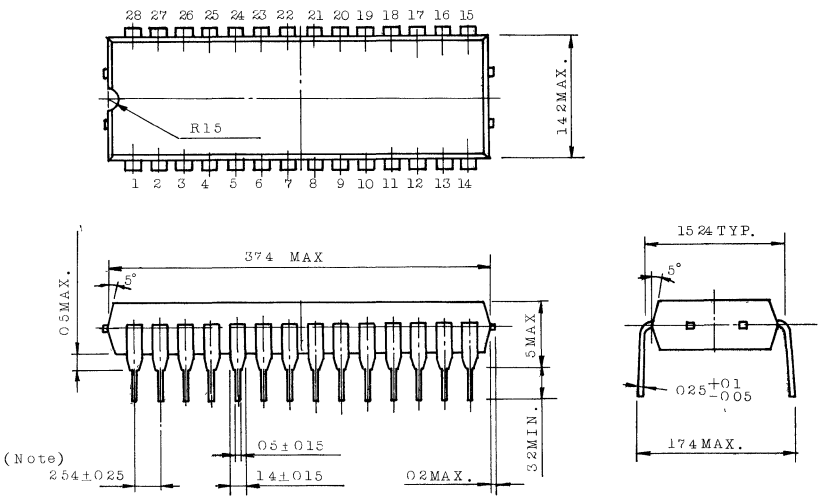
TIMING WAVEFORM



PRELIMINARY

OUTLINE DRAWING

Unit in mm



Note: 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.28 leads.

TOSHIBA

INTEGRATED CIRCUIT

TECHNICAL DATA

TMPZ84C40P/TMPZ84C41P/TMPZ84C42P

ADVANCE INFORMATION

CMOS Z80® SIO: SERIAL INPUT/OUTPUT CONTROLLER

GENERAL DESCRIPTION

The TMPZ84C40P (SIO/0), TMPZ84C41P (SIO/1) and TMPZ84C42P (SIO/2), (from here on referred to as Z80 SIO), are CMOS versions of Z80 SIO which provide low power operation and are designed to satisfy a wide variety of serial data communications requirements in microcomputer systems.

The Z80 SIO can handle asynchronous and synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC.

The CMOS Z80 SIOs are fabricated using Toshiba's C^2MOS Silicon Gate Technology.

FEATURES

- Z80 Compatible SIOs
- CCITT-X.25 Compatible
- HDLC/SDLC Compatible
- Data Rates up to 800 k/sec.
- Two Independent Full-duplex channels
- Asynchronous/Synchronous Protocols
- Automatic CRC Generation and Checking
- Z80 Compatible Daisy - Chain Interrupt Structure
- DC to 4MHz Operation
- Single 5V Power Supply

  4MHz @ 5V ± 10%
- Low Power Consumption

  Less than 10μA @ 5V (Power down)
- Extended Operating Temperature

  -40°C to 85°C

® Z80 is a trademark of Zilog Inc.

ADVANCE INFORMATION

PIN CONNECTIONS (TOP VIEW)

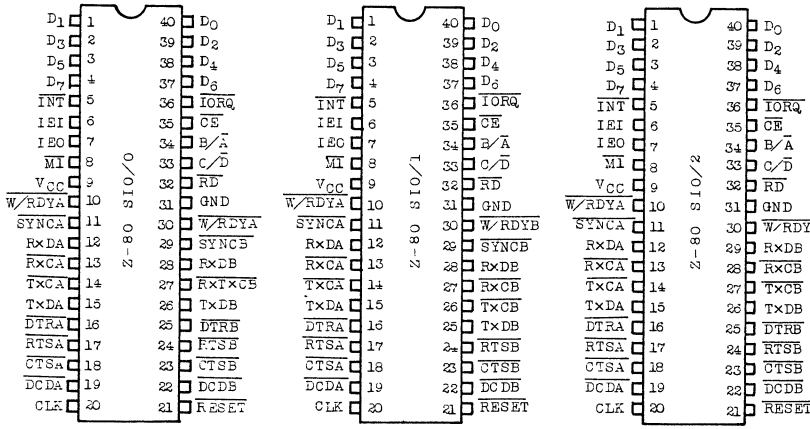


FIGURE 1. Z80 SIO/0, SIO/1, SIO/2 PINOUT DIAGRAMS

BLOCK DIAGRAM

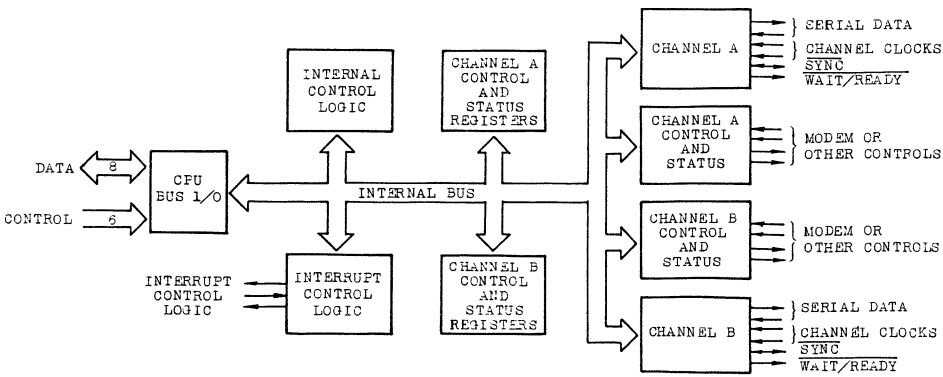


FIGURE 2. BLOCK DIAGRAM

POSTSCRIPT

This Manual is a reference for the customer applying the TLCS-Z80 family. It contains the function and specification of each LSI device of the TLCS-Z80. The examples of application circuits described here are shown as the reference for explanation. Toshiba assumes no responsidility for any prolems caused by using these examples. the information herein is subhict to change without prior notice.

This manual has been prepared by the following section.

Microconputer LSI Application Engineering Sctions
Integrated Circuit Division, Toshiba Corpration

1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 210, Japan
Phone: Japan (81)44-511-3111

PART 5

INTEGRATED CIRCUIT TECHNICAL DATA

TLCS-85

LSI DEVICES

July. 1 9 8 4

PREFACE

This Part describes the detail functions and specefications of the LSI devices of the TLCS-85 family. The TLCS-85 family consists of a popular 8085A MPU and seven basic peripheral devices which are fabricated with our latest NMOS technology. All these devices are considered to be industrial standards. Many other semiconductor manufacturers make compatible devices with ours. But our devices are expected to be superior on quality and reliability.

CONTENTS

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8085AP

N-CHANNEL SILICON GATE MOS

8-BIT SINGLE CHIP MICROPROCESSOR

GENERAL DESCRIPTION

The TMP8085AP, from here on referred to as the TMP8085A, is a new generation, complete 8 bit parallel central processing unit (CPU). Its instruction set is 100% software compatible with the TMP9080A (8080A) microprocessor, and it is designed to improve the present 9080's performance by higher system speed. Its high level of system integration allows a minimum system of there IC's : TMP8085A (CPU), TMP8155P/TMP8156P (RAM/IO) and TMP8755AC (EPROM/IO)/ TMP8355P (ROM/IO). The TMP8085A uses a multiplexed data bus. The address is plit between the 8 bit address bus and the 8 bit data bus. The on-chip address latches of TMP8155P/TMP8156P/TMP8755AC/TMP8355P memory products allow a direct interface with TMP8085P.

FEATURES

- 100% Software Compatible with TMP9080A
- 1.3 s Instruction Cycle
- Single +5V Power Supply
- On-Chip Clock Generator (with External Crystal or RC Network)
- On-Chip System Controller; Advanced Cycle status information available for Large System Control
- 4 Vectored Interrupts (One is Non-Maskable) Plus an TMP9080A compatible interrupt
- Decimal, Binary and Double Precision Arithmetic
- Serial In/Serial Out Port
- Direct Addressing Capability to 64K Bytes of Memory
- Compatible with Intel's 8085A

PIN CONNECTION

(TOP VIEW)

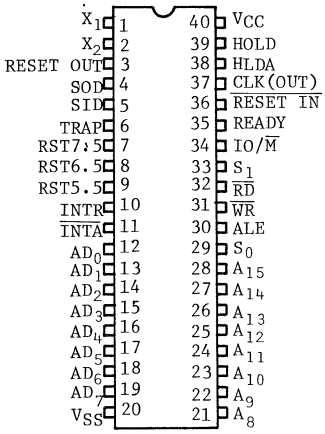


FIGURE 2. TMP8085A PINOUT DIAGRAM

BLOCK DIAGRAM

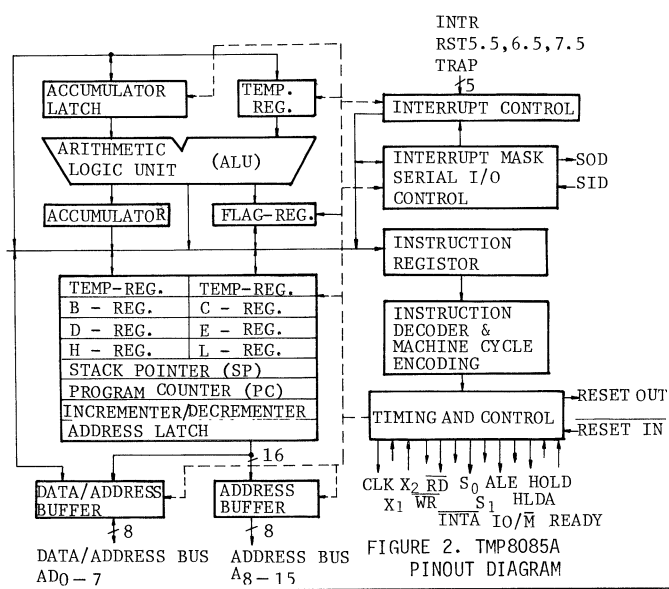


FIGURE 2. TMP8085A PINOUT DIAGRAM

PIN NAME AND PIN DESCRIPTION

X_1, X_2 (Input)

Crystal, LC, or RC network are connected to X_1 and X_2 to drive the internal clock generator. X_1 and X_2 can also be driven from an externally derived frequency source. The input frequency is devided by 2 to give the processor's internal operating frequency.

CLK (Output)

Clock Output for use as a system clock. The period of CLK is twice the X_1, X_2 input period.

$\overline{\text{RESET IN}}$ (Input)

The RESET Input initialize the processor by clearing the program counter, instruction register, SOD latch, Interrupt Enable flip-flop and HLDA flip-flop. The address and data buses and the control lines are 3-stated during RESET and because of the asynchronous nature of RESET, the processor's internal registers and flags may be altered by RESET with unpredictable results. $\overline{\text{RESET IN}}$ is a Schmitt-triggered input, allowing connection to an RC network for power on RESET delay. The TMP8085A is held in the reset condition as long as $\overline{\text{RESET IN}}$ is applied.

RESET OUT (OUTPUT)

The RESET OUT signal indicates that the TMP8085A is being reset. It can be used as a system reset. It is synchronized to the processor clock and lasts an integral number of clock periods.

SOD (Output)

Serial output data line. The output SOD is set or reset as specified by the SIM instruction.

SID (Input)

Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.

INTR (Input)

INTERRUPT REQUEST signal provides a mechanism for external devices to modify the instruction flow of the program in progress. It is sampled only during the next to the last clock cycle of an instruction and during Hold and Halt states. If it is recognized, the processor will complete the execution of the current instruction, and then the Program Counter (PC) will be inhibited from incrementing and an $\overline{\text{INTA}}$ will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by RESET and immediately after an interrupt is accepted.

$\overline{INTA}$ (Output)

INTERRUPT ACKNOWLEDGE: Occurs in response to an Interrupt input and indicates that the processor will be ready for an interrupt instruction on the data bus. It is used instead of (and has the same timing as) $\overline{RD}$ during the instruction cycle after an INTR is accepted.

RST 5.5
RST 6.5 } (Inputs)
RST 7.5

RESTART INTERRUPTS: These three inputs have the same timing as INTR exept they cause an internal RESTART to be automatically inserted. These interrupts have a higher priority than INTR. The priority of these interrupts is ordered as shown Table 1. They may be individually masked out using the SIM instruction.

TRAP (Input)

Trap interrupt is a nonmaskable RESTART interrupt. It is sampled at the same timing as INTR or RST 5.5 - 7.5. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt.

AD_0 - AD_7 (Input/Output, 3 - state)

Lower 8 bits of the memory address (or I/O address) appear on the bus during the first clock cycle (T_1 state) of a machine cycle. It then becomes the data bus during the second and third clock cycles.

A_8 - A_{15}(Output, 3-state)

The most significant 8 bits of the memory address or the 8 bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.

S_0, S_1, and IO/$\overline{M}$ (Output)
Machine cycle status:

IO/$\overline{M}$	S_1	S_0	Status
0	1	1	Opcode fetch
0	1	0	Memory read
0	0	1	Memory write
1	1	0	I/O read
1	0	1	I/O write
1	1	1	Interrupt Acknowledge
TS	0	0	Halt
TS	X	X	Hold
TS	X	X	Reset

Note: TS = 3-state (high impedance)
X = unspecified

ALE (Output)

Address Latch Enable: It occurs during the first clock state of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE can be used to strobe the status information. ALE is never 3-stated.

$\overline{WR}$ (Output, 3-state)

WRITE control: A low level on $\overline{WR}$ indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of $\overline{WR}$. It is 3-stated during Hold and Halt modes and during RESET.

$\overline{RD}$ (Output, 3-state)

READ control: A low level on $\overline{RD}$ indicates the selected memory or I/O device to be read and that the Data Bus is available for the data transfer, 3-stated during Hold and Halt modes and during RESET.

READY(Input)

When READY is absent (low), indicating the external operation is not complete, the processor will enter the Wait state. It will wait an integral number of clock cycles for READY to go high before completing the read or write cycle.

HOLD (Input)

The Hold input allows an external signal to cause the processor to relinquish control over the address bus and the data bus. When Hold goes active, the processor completes its current operation, activates the HLDA output, and puts the Address, Data, $\overline{RD}$, $\overline{WR}$, and IO/$\overline{M}$ lines into their high-impedance state. Internal processing can continue. The Holding device can then utilize the address and data buses without interference. The processor can regain the bus only after the Hold is removed.

HLDA (Output)

The Hold Acknowledge output signal is a response to a Hold input. It indicates that the processor has received the HOLD request and it will relinquish the bus in the next cycle. HLDA goes low after the Hold request is moved. The processor takes the bus one half clock cycle after HDLA goes low.

V_{CC}
+5 volt supply

V_{SS}
Ground Reference

FUNCTIONAL DESCRIPTION

The TMP8085A is a complete 8-bit parallel central processor. Its basic clock speed is 3 MHz. Also it is designed to fit into a minimum system of three IC's: The CPU (TMP8085A), a RAM I/O (TMP8155P or TMP8156P), and a ROM or EPROM I/O chip (TMP8355P or TMP8755AC).

The TMP8085A is provided with internal 8-bit registers and 16-bit registers. The TMP8085A has eight addressable 8-bit registers. Six of them can be used either as 8-bit registers or as 16-bit register pairs. In addition to the register pairs, the TMP8085A contains two more 16-bit registers. The TMP8085A register set is as follows:

- The accumulator (A Register) is the focus of all of the accumulator instructions, which include arithmetic, logic, load and store, and I/O instructions.
- The program counter (PC) always points to the memory location of the next instruction to be executed.
- General - purpose registers BC, DE, and HL may be used as 8-bit registers or as three 16-bit registers, interchangeably, depending on the instruction being performed.
- The stack pointer (sp) is a special data pointer that always points to the stack top (next available stack address).
- The flag register contains five one-bit flags, each of which records processor status information and may also control processor operation.

The five flags in the TMP8085A CPU are shown below:

(MSB)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		C

- The carry flag (C) is set and reset by arithmetic operations. An addition operation that resutls in an overflow out of the high-order bit of the accumulator sets the carry flag. The carry flag also acts as a "borrow" flag for subtract instruction.
- The auxiliary carry flag (AC) indicates overflow out of bit 3 of the accumulator in the same way that C flag indicates overflow out of bit 7. This flag is commonly used in BCD arithmetic.
- The sign flag (S) is set to the condition of the most significant (MSB) bit of the accumulator following the execution of arithmetic or logic instructions.
- The zero flag (Z) is set if the result generated by certain instructions is zero. The zero flag is cleared if the result is not zero.
- The parity flag (P) is set to 1 if the parity (number of 1-bits) of the accumulator is even. If odd, it is cleared.

In the TMP8085A microprocessor are contained the functions of clock generation, system bus control, and interrupt priority selection, in addition to execution of the instruction set. The TMP8085A uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first T state (T_1 clock cycle) of a machine cycle the lower order address is sent out on the Address/Data Bus. These lower 8 bits may be latched externally by the Address Latch Enable signal (ALE). During the rest of the machine cycle the data bus is used for memory or I/O data.

INTERRUPT AND SERIAL I/O

The TMP8085A has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5, and TRAP. INTR is identical in function to 9080A INT. Each of three RESTART inputs 5.5, 6.5, 7.5, has a programmable mask. TRAP is also a RESTART interrupt but it is nonmaskable.

The three RESTART interrupts cause the internal execution of RESTART if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes internal execution independent of the state of the interrupt enable or masks.

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are high levelsensitive like INTR and are recognized with the same timing as INTR.RST 7.5 is rising edge-sensitive.

For RST 7.5, only a pulse is required to set an internal flip-flop which generates the interrupt request. The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a $\overline{\text{RESET IN}}$ to the TMP8085A. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending : TRAP-highest priority. RST 7.5, RST 6.5, RST 5.5, INTR - lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt a RST 7.5 routine if the interrupts were reenabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic errors such as power failure or bus error. It is not affected by any flag or mask. The TRAP input is both edge and level sensitive. The TRAP input must go high and remain high until it is acknowledged. It will not be recognized again until it goes low, then high again. This avoids any false triggering due to noise or logic glitches.

TABLE 1. INTERRUPT PRIORITY, RESTART ADDRESS, AND SENSITIVITY

Name	Priority	Address Branched to When Interrupt Occurs	Type Trigger
TRAP	1	24 (Hex.)	Rising edge and high level until sampled.
RST 7.5	2	3C (Hex.)	Rising edge (latched).
RST 6.5	3	34 (Hex)	High level until sampled.
RST 5.5	4	2C (Hex.)	High level until sampled.
INTR	5	See Note (2)	High level until sampled.

Notes: (1) The processor pushes the PC on the stack before branching to the indicated address.

(2) The address branched to depends on the instruction provided to the TMP8085A when the interrupt is acknowledged.

The TRAP interrupt is special in that it disables interrupts, but preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instruction provide current interrupt enable status. Performing a RIM instruction following INTR, or RST 5.5 - 7.5 will provide current interrupt enable status, revealing that interrupts are disabled.

The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD date.

BASIC TIMING

The execution of each instruction by the TMP8085A consists of a sequence of from one to five machine cycles, and cach machine cycle consists of a minimum of from three to six clock cycles. Most machine cycles consist of three T states, (cycles of the CLK output) with the exception of opecode fetch, which normally has either four or six T states (unless WAIT or HOLD states are forced by the receipt of $\overline{\text{READY}}$ or HOLD inputs). Any T state must be one of ten possible states, shown in Table 3.
At the beginning of every machine cycle, the TMP8085A sends out three status signals ($IO/\overline{M}$, S_1, S_0) that define what type of machine cycle is about to take place. The TMP8085A also sends out a 16-bit address at the beginning of every machine cycle to identify the particular memory location or I/O port that the machine cycle applies to.

The special timing signal, ADDRESS LATCH ENABLE (ALE), is used a strobe to sample the lower 8-bits of address on the $AD_0 - AD_7$ lines. ALE is present during T_1 of every machine cycle. Control lines $\overline{RD}$ ($\overline{INTA}$) and $\overline{WR}$ become active later, at the time when the transfer of data is to take plece.
Figure 3 shows an instruction fetch, memory read and I/O write cycle (as would occur during processing of the OUT instruction).

TABLE 2. TMP8085A MACHINE CYCLE CHART

MACHINE CYCLE	IO/$\overline{M}$	S_1	SO	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OPCODE FETCH	0	1	1	0	1	1
MEMORY READ	0	1	0	0	1	1
MEMORY WRITE	0	0	1	1	0	1
I/O READ	1	1	0	0	1	1
I/O WRITE	1	0	1	1	0	1
ACKNOWLEDGE OF INTR	1	1	1	1	1	0
BUS IDLE : DAD	0	1	0	1	1	1
ACK. OF RST, TRAP	1	1	1	1	1	1
HALT	TS	0	0	TS	TS	1

NOTE: 0 = Logic "0", 1 = Logic "1", TS = High Impedance

TABLE 3. TMP8085A MACHINE STATE CHART

MACHINE STATE	S_1, S_0	IO/$\overline{M}$	A_8-A_{15}	AD_0-AD_7	$\overline{RD}, \overline{WR}$	$\overline{INTA}$	ALE
T_1	X	X	X	X	1	1	1°
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4	1	0†	X	TS	1	1	0
T_5	1	0†	X	TS	1	1	0
T_6	1	0†	X	TS	1	1	0
T_{RESET}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0
T_{HOLD}	X	TS	TS	TS	TS	1	0

NOTES: (1) 0 = Logic "0", 1 = Logic "1", TS = High Impedance, X = Unspecified

(2) °ALE not generated during 2nd and 3rd machine cycles of DAD instruction

(3) † IO/$\overline{M}$ = 1 during T_4 - T_6 of INA machine cycle

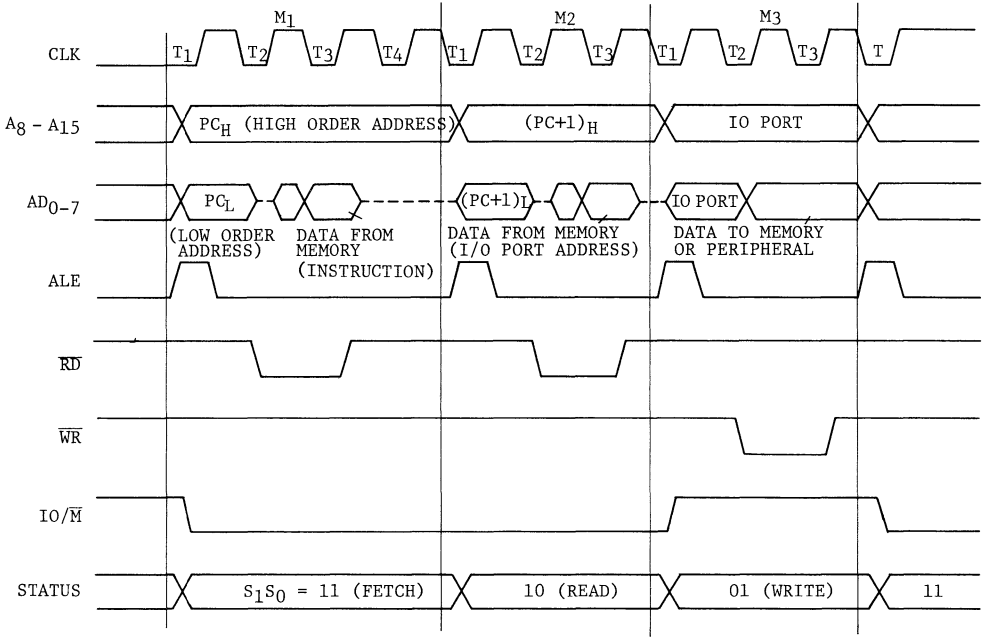


FIGURE 3. TMP8085A BASIC SYSTEM TIMING

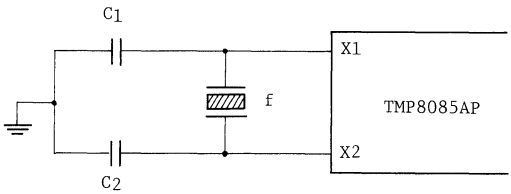
DRIVING THE X1 AND X2 INPUTS

You may drive the clock inputs of the TMP8085A with a crystal, an LC tuned circuit, an RC network or an external clock source. The driving frequency must be at least 1 MHz, and must be twice the desired internal clock frequency.

A. Quartz Crystal Clock Driver

If a crystal used, it must have the following characteristics.

- Parallel resonance at twice the clock frequency desired
- C_S (shunt capacitance) $\leq$ 7 PF
- R_S (equivalent shunt resistance) $\leq$ 75 Ohms



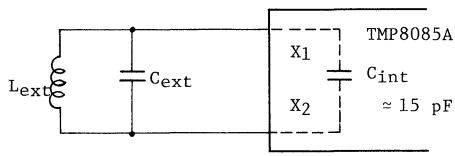
Note a value of the external capacitors C_1 and C_2 between X1, X2 and ground. In case of the crystal frequency above 4 MHz, it is recommended that you choose a value of 10pF for C_1 and C_2 and less than 4 MHz, 20pF capacitors are recommended.

B. LC Turned Circuit Clock Driver

A parallel-resonant LC circuit may be used as the frequency-determining network for the TMP8085A, providing that its frequency tolerance of approximately 10% is acceptable. The components are chosen from the formula.

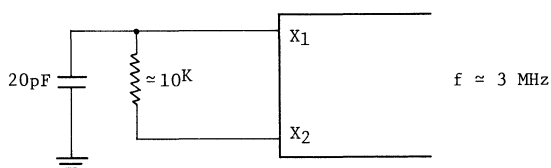
$$f = \frac{1}{2\pi \sqrt{L\ (C_{ext} + C_{int})}}$$

The use of an LC circuit is not recommended for frequencies higher than approximately 5 MHz.

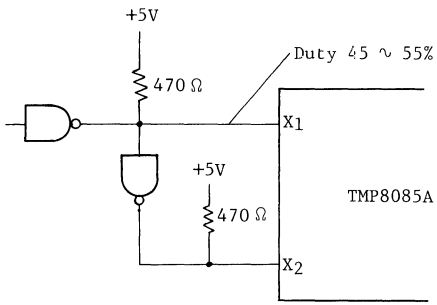


C. RC Circuit Clock Driver

An RC circuit may be used as the frequency - determining network for the TMP 8085A if maintaining a precise clock frequency is of no importance. Variations in the on-chip timing generation can cause a wide variation in frequency when using RC circuit. The driving frequency generated by the circuit shown is approximately 3 MHz. It is not recommended that frequencies greatly higher or lower than this be attempted.



D. External clock Driver Circuit



POWER ON AND $\overline{\text{RESET IN}}$

The TMP 8085A is not guaranteed to work until 10 ms after V_{CC} reaches 4.75 V. It is suggested that $\overline{\text{RESET IN}}$ be kept low during this period.
Note that the 10 ms period does not include the time it takes for the power supply to reach its 4.75 V level.

INSTRUCTION SET

Symbols and Abbreviations

SYMBOLS	DEFINITION
ddd,sss	The bit pattern designating one of the registers A,B,C,D,E,H,L (ddd=destination, sss=source):

ddd or sss	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L
110	M (Memory)

r,r1,r2	One of the registers A,B,C,D,E,H,L
d8	8-bit data quantity
d16	16-bit data quantity
addr8	8-bit address of an I/O device
addr	16-bit address quantity
RP	The bit pattern designating one of the register pairs B,D,H,SP:

RP	rp	REGISTER PAIR (rpH)(rpL)
00	B	B-C
01	D	D-E
10	H	H-L
11	SP	SP

B_2	The second byte of the instruction
B_3	The third byte of the instruction
0	Affected
S	Set
R	Reset
-	Not affected

Data Transfer

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
MOV r1, r2	0	1	d	d	d	S	S	S	(r1) ← (r2)	1	4	-	-	-	-	-
MOV M, r	0	1	1	1	0	S	S	S	[(H)(L)] ← (r)	1	7	-	-	-	-	-
MOV r, M	0	1	d	d	d	1	1	0	(r) ← [(H)(L)]	1	7	-	-	-	-	-
MVI r, d8	0	0	d	d	d	1	1	0	(r) ← (B_2)	2	7	-	-	-	-	-
					B_2											
MVI M, d8	0	0	1	1	0	1	1	0	[(H)(L)] ← (B_2)	2	10	-	-	-	-	-
					B_2											
LDA addr	0	0	1	1	1	0	1	0	(A) ← [(B_3)(B_2)]	3	13	-	-	-	-	-
					B_2											
					B_3											
LDAX B	0	0	0	0	1	0	1	0	(A) ← [(B)(C)]	1	7	-	-	-	-	-
LDAX D	0	0	0	1	1	0	1	0	(A) ← [(D)(E)]	1	7	-	-	-	-	-
LHLD addr	0	0	1	0	1	0	1	0	(L) ← [(B_3)(B_2)]	3	16	-	-	-	-	-
					B_2				(H) ← [(B_3)(B_2)+1]							
					B_3											
LXI H, d16	0	0	1	0	0	0	0	1	(H) ← (B_3)	3	10	-	-	-	-	-
					B_2				(L) ← (B_2)							
					B_3											
LXI D, d16	0	0	0	1	0	0	0	1	(D) ← (B_3)	3	10	-	-	-	-	-
					B_2				(E) ← (B_2)							
					B_3											
LXI B, d16	0	0	0	0	0	0	0	1	(B) ← (B_3)	3	10	-	-	-	-	-
					B_2				(C) ← (B_2)							
					B_3											
LXI SP, d16	0	0	1	1	0	0	0	1	$(SP)_H$ ← (B_3)	3	10	-	-	-	-	-
					B_2				$(SP)_L$ ← (B_2)							
					B_3											
SHLD addr	0	0	1	0	0	0	1	0	[(B_3)(B_2)] ← (L)	3	16	-	-	-	-	-
					B_2				[(B_3)(B_2)+1] ← (H)							
					B_3											
STA addr	0	0	1	1	0	0	1	0	[(B_3)(B_2)] ← (A)	3	13	-	-	-	-	-
					B_2											
					B_3											

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
STAX B	0	0	0	0	0	0	1	0	[(B)(C)] ← (A)	1	7	-	-	-	-	-
STAX D	0	0	0	1	0	0	1	0	[(D)(E)] ← (A)	1	7	-	-	-	-	-
SPHL	1	1	1	1	1	0	0	1	(SP) ← (H)(L)	1	6	-	-	-	-	-
XCHG	1	1	1	0	1	0	1	1	(H) ↔ (D) (L) ↔ (E)	1	4	-	-	-	-	-
XTHL	1	1	1	0	0	0	1	1	(L) ↔ [(SP)] (H) ↔ [(SP)+1]	1	16	-	-	-	-	-
IN addr8	1	1	0	1	1	0	1	1 B_2	(A) ← (data)	2	10	-	-	-	-	-
OUT addr8	1	1	0	1	0	0	1	1 B_2	(data) ← (A)	2	10	-	-	-	-	-

Branch

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
JMP addr	1	1	0	0	0	0	1	1 B_2 B_3	(PC) ← (B_3)(B_2)	3	10	-	-	-	-	-
JNZ addr	1	1	0	0	0	0	1	0 B_2 B_3	If Z = 0 (PC) ← (B_3)(B_2), If Z = 1 (PC) ← (PC) + 3	3	7/10	-	-	-	-	-
JZ addr	1	1	0	0	1	0	1	0 B_2 B_3	If Z = 1 (PC) ← (B_3)(B_2), If Z = 0 (PC) ← (PC) + 3	3	7/10	-	-	-	-	-
JNC addr	1	1	0	1	0	0	1	0 B_2 B_3	If C = 0 (PC) ← (B_3)(B_2), If C = 1 (PC) ← (PC) + 3	3	7/10	-	-	-	-	-
JC addr	1	1	0	1	1	0	1	0 B_2 B_3	If C = 1 (PC) ← (B_3)(B_2), If C = 0 (PC) ← (PC) + 3	3	7/10	-	-	-	-	-

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
JPO addr	1	1	1	0	0	0	1	0	If P = 0	3	7/10	-	-	-	-	-
					B_2				(PC) ← $(B_3)(B_2)$,							
					B_3				If P = 1							
									(PC) ← (PC) + 3							
JPE addr	1	1	1	0	1	0	1	0	If P = 1	3	7/10	-	-	-	-	-
					B_2				(PC) ← $(B_3)(B_2)$,							
					B_3				If P = 0							
									(PC) ← (PC) + 3							
JP addr	1	1	1	1	0	0	1	0	If S = 0	3	7/10	-	-	-	-	-
					B_2				(PC) ← $(B_3)(B_2)$,							
					B_3				If S = 1							
									(PC) ← (PC) + 3							
JM addr	1	1	1	1	1	0	1	0	If S = 1	3	7/10	-	-	-	-	-
					B_2				(PC) ← $(B_3)(B_2)$,							
					B_3				If S = 0							
									(PC) ← (PC) +3							
CALL addr	1	1	0	0	1	1	0	1	[(SP)-1] ← (PCH)	3	18	-	-	-	-	-
					B_2				[(SP)-2] ← (PCL)							
					B_3				(SP) ← (SP) - 2							
									(PC) ← $(B_3)(B_2)$							
CNZ addr	1	1	0	0	0	1	0	0	If Z = 0, the actions specified in the CALL instruction are performed.	3	9/18	-	-	-	-	-
					B_2											
					B_3											
									If Z = 1							
									(PC) ← (PC) + 3							
CZ addr	1	1	0	0	1	1	0	0	If Z = 1, the actions specified in the CALL instruction are performed.	3	9/18	-	-	-	-	-
					B_2											
					B_3											
									If Z = 0,							
									(PC) ← (PC) + 3							

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
CNC addr	1	1	0	1	0	1	0	0	If C = 0, the actions specified in the CALL instruction are performed. If C = 1 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
				B_2												
				B_3												
CC addr	1	1	0	1	1	1	0	0	If C = 1, the actions specified in the CALL instruction are performed. If C = 0 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
				B_2												
				B_3												
CPO addr	1	1	1	0	0	1	0	0	If P = 0, the actions specified in the CALL instruction are performed. If P = 1 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
				B_2												
				B_3												
CPE addr	1	1	1	0	1	1	0	0	If P = 1, the actions specified in the CALL instruction are performed. If P = 0 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
				B_2												
				B_3												
CP addr	1	1	1	1	0	1	0	0	If S = 0, the actions specified in the CALL instruction are performed. If S = 1 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
				B_2												
				B_3												

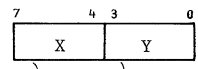
Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
CM addr	1	1	1	1	1	1	0	0	If S = 1, the actions specified in the CALL instruction are performed.	3	9/18	-	-	-	-	-
				B_2					If S = 0							
				B_3					(PC) ← (PC) + 3							
RET	1	1	0	0	1	0	0	1	(PCL) ← [(SP)]	1	10	-	-	-	-	-
									(PCH) ← [(SP)+1]							
									(SP) ← (SP) + 2							
RNZ	1	1	0	0	0	0	0	0	If Z = 0, the actions specified in the RET instruction are performed.	1	6/12	-	-	-	-	-
									If Z = 1							
									(PC) ← (PC) + 1							
RZ	1	1	0	0	1	0	0	0	If Z = 1, the actions specified in the RET instruction are performed.	1	6/12	-	-	-	-	-
									If Z = 0							
									(PC) ← (PC) + 1							
RNC	1	1	0	1	0	0	0	0	If C = 0, the actions specified in the RET instruction are performed.	1	6/12	-	-	-	-	-
									If C = 1							
									(PC) ← (PC) + 1							

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
RC	1	1	0	1	1	0	0	0	If C = 1, the actions specified in the RET instruction are performed. If C = 0 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RPO	1	1	1	0	0	0	0	0	If P = 0, the actions specified in the RET instruction are performed. If P = 1 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RPE	1	1	1	0	1	0	0	0	If P = 1, the actions specified in the RET instruction are performed. If P = 0 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RP	1	1	1	1	0	0	0	0	If S = 0, the actions specified in the RET instruction are performed. If S = 1 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RM	1	1	1	1	1	0	0	0	If S = 1, the actions specified in the RET instruction are performed. If S = 0 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
PCHL	1	1	1	0	1	0	0	1	(PCH) ← (H) (PCL) ← (L)	1	6	-	-	-	-	-
RST	1	1	A	A	A	1	1	1	[(SP)-1] ← (PCH) [(SP)-2] ← (PCL) (SP) ← (SP) - 2 (PC) ← (00000000 00AAA000)	1	12	-	-	-	-	-

Arithmetic

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
ADD r	1	0	0	0	0	S	S	S	(A) ← (A) + (r)	1	4	0	0	0	0	0
ADC r	1	0	0	0	1	S	S	S	(A) ← (A) + (r) + (C)	1	4	0	0	0	0	0
ADD M	1	0	0	0	0	1	1	0	(A) ← (A)+[(H)(L)]	1	7	0	0	0	0	0
ADC M	1	0	0	0	1	1	1	0	(A)←(A)+[(H)(L)]+(C)	1	7	0	0	0	0	0
ADI d8	1	1	0	0	0	1	1	0	(A) ← (A) + (B_2)	2	7	0	0	0	0	0
					B_2											
ACI d8	1	1	0	0	1	1	1	0	(A) ← (A)+(B_2)+(C)	2	7	0	0	0	0	0
					B_2											
DAD rp	0	0	R	P	1	0	0	1	(H)(L) ← (H)(L) + (rH)(rL)	1	10	0	-	-	-	-
SUB r	1	0	0	1	0	S	S	S	(A) ← (A) - (r)	1	4	0	0	0	0	0
SBB r	1	0	0	1	1	S	S	S	(A) ← (A) - (r) - (C)	1	4	0	0	0	0	0
SUB M	1	0	0	1	0	1	1	0	(A) ← (A)-[(H)(L)]	1	7	0	0	0	0	0
SBB M	1	0	0	1	1	1	1	0	(A)←(A)-[(H)(L)]-(C)	1	7	0	0	0	0	0
SUI d8	1	1	0	1	0	1	1	0	(A) ← (A) - (B_2)	2	7	0	0	0	0	0
					B_2											
SBI d8	1	1	0	1	1	1	1	0	(A) ← (A) - (B_2) - (C)	2	7	0	0	0	0	0
					B_2											

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
DAA	0	0	1	0	0	1	1	1	The 8-bit number in the accumulator is adjusted to form two 4-bit BCD digits by the following process. Accumulator  1. If $Y \geq 10$ or AC=1, $(A) \leftarrow (A) + 6$ 2. If $X \geq 10$ or C=1, $(A)_{4-7} \leftarrow (A)_{4-7} + 6$	1	4	0	0	0	0	0

Logical Instruction

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
ANA r	1	0	1	0	0	S	S	S	(A) ← (A) ∧ (r)	1	4	R	O	O	O	S
ANA M	1	0	1	0	0	1	1	0	(A) ← (A) ∧ [(H)(L)]	1	7	R	O	O	O	S
ANI d8	1	1	1	0	0	1	1	0	(A) ← (A) ∧ (B_2)	2	7	R	O	O	O	S
					B_2											
XRA r	1	0	1	0	1	S	S	S	(A) ← (A) ∀ (r)	1	4	R	O	O	O	R
XRA M	1	0	1	0	1	1	1	0	(A) ← (A) ∀ [(H)(L)]	1	7	R	O	O	O	R
XRI d8	1	1	1	0	1	1	1	0	(A) ← (A) ∀ (B_2)	2	7	R	O	O	O	R
					B_2											
ORA r	1	0	1	1	0	S	S	S	(A) ← (A) ∨ (r)	1	4	R	O	O	O	R
ORA M	1	0	1	1	0	1	1	0	(A) ← (A) ∨ [(H)(L)]	1	7	R	O	O	O	R
ORI d8	1	1	1	1	0	1	1	0	(A) ← (A) ∨ (B_2)	2	7	R	O	O	O	R
					B_2											
CMP r	1	0	1	1	1	S	S	S	(A) − (r)	1	4	O	O	O	O	O
CMP M	1	0	1	1	1	1	1	0	(A) − [(H)(L)]	1	7	O	O	O	O	O
CPI d8	1	1	1	1	1	1	1	0	(A) − (B_2)	2	7	O	O	O	O	O
					B_2											
CMA	0	0	1	0	1	1	1	1	(A) ← $(\overline{A})$	1	4	-	-	-	-	-
RLC	0	0	0	0	0	1	1	1	(A_{n+1}) ← (An) (A_0) ← (A_7) (C) ← (A_7)	1	4	O	-	-	-	-
RRC	0	0	0	0	1	1	1	1	(An) ← (A_{n+1}) (A_7) ← (A_0) (C) ← (A_0)	1	4	O	-	-	-	-
RAL	0	0	0	1	0	1	1	1	(A_{n+1}) ← (An) (C) ← (A_7) (A_0) ← (C)	1	4	O	-	-	-	-
RAR	0	0	0	1	1	1	1	1	(An) ← (A_{n+1}) (C) ← (A_0) (A_7) ← (C)	1	4	O	-	-	-	-

Increment and Decrement

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
INR r	0	0	d	d	d	1	0	0	(r) ← (r) + 1	1	4	-	0	0	0	0
INR M	0	0	1	1	0	1	0	0	[(H)(L)] ← [(H)(L)]+1	1	10	-	0	0	0	0
INX rp	0	0	R	P	0	0	1	1	(rH)(rL) ← (rH)(rL)+1	1	6	-	-	-	-	-
DCR r	0	0	d	d	d	1	0	1	(r) ← (r) - 1	1	4	-	0	0	0	0
DCR M	0	0	1	1	0	1	0	1	[(H)(L)] ← [(H)(L)]-1	1	10	-	0	0	0	0
DCX rp	0	0	R	P	1	0	1	1	(rH)(rL) ← (rH)(rL)-1	1	6	-	-	-	-	-

Stack

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
PUSH rp	1	1	R	P	0	1	0	1	[(SP)-1] ← (rH) [(SP)-2] ← (rL) (SP) → (SP) - 2 Note: Register pair rp=SP may not be specified.	1	12	-	-	-	-	-
PUSH PSW	1	1	1	1	0	1	0	1	[(SP)-1] ← (A) [(SP)-2] ← $D_7D_6D_5D_4D_3D_2D_1D_0$ S Z X AC X P X C MSB (SP) ← (SP) - 2	1	12	-	-	-	-	-
POP rp	1	1	R	P	0	0	0	1	(rL) ← [(SP)] (rH) ← [(SP)+1] (SP) ← (SP)+2	1	10	-	-	-	-	-
POP PSW	1	1	1	1	0	0	0	1	(C) ← $[(SP)]_0$ (P) ← $[(SP)]_2$ (AC) ← $[(SP)]_4$ (Z) ← $[(SP)]_6$ (S) ← $[(SP)]_7$ (A) ← [(SP)+1] (SP) ← (SP) + 2	1	10	0	0	0	0	0

Control

Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
HLT	0	1	1	1	0	1	1	0	Halt	1	5	-	-	-	-	-
STC	0	0	1	1	0	1	1	1	(C) ← 1	1	4	0	-	-	-	-
CMC	0	0	1	1	1	1	1	1	(C) ← $(\overline{C})$	1	4	0	-	-	-	-
EI	1	1	1	1	1	0	1	1	Enable interrupts Note: Interrupts are not recognized during the EI instruction.	1	4	-	-	-	-	-
DI	1	1	1	1	0	0	1	1	Disable interrupts Note: Interrupts are not recognized during the DI instruction.	1	4	-	-	-	-	-
NOP	0	0	0	0	0	0	0	0	No operation is performed.	1	4	-	-	-	-	-
RIM	0	0	1	0	0	0	0	0	(A) ← d_7 = SID d_6 = I7 d_5 = I6 d_4 = I5 d_3 = IE d_2 = M7 d_1 = M6 d_0 = M5	1	4	-	-	-	-	-
SIM	0	0	1	1	0	0	0	0	$IF(A)_6$ = 1; SOD ← $(A)_7$,$IF(A)_3$ = 1; M7 ← $(A)_2$ M6 ← $(A)_1$ M5 ← $(A)_0$,$IF(A)_4$ = 1; RST7.5 RESET	1	4	-	-	-	-	-

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Ratings
V_{CC}	V_{cc} Supply Voltage	-0.5V to 7.0V
V_{IN}	Input Voltage with Respect to V_{SS}	-0.5V to 7.0V
V_{OUT}	Output Voltage with Respect to V_{SS}	-0.5V to 7.0V
P_D	Power Dissipation	1.5W
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

DC CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V±5%

Symbol	Parameter	Test Conditions	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_{OL}	Output Low Voltage	I_{OL} = 2mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = -400μA	2.4			V
I_{CC}	Power Supply Current				170	mA
I_{IL}	Input Leakage	$V_{IN} = V_{CC}$			±10	μA
I_{LO}	Output Leakage	$0.45 \leq V_{OUT} \leq V_{CC}$			10	μA
V_{ILR}	Input Low Level (RESET)		-0.5		0.8	V
V_{IHR}	Input High Level (RESET)		2.4		V_{CC} +0.5	V
V_{HY}	Hysteresis (RESET)		0.25			V

AC CHARACTERISTICS

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

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
t_{CYC}	CLK Cycle Period		320		2000	ns
t_L	CLK Low Time - Standard 150pF Loading - Lightly Loaded [2]		80 100			ns ns
t_H	CLK High Time - Standard 150pF Loading - Lightly Loaded[2]		120 150			ns ns
t_r, t_f	CLK Rise and Fall Time				30	ns
t_{XKR}	X_1 Rising to CLK Rising		30		120	ns
t_{XKF}	X_1 Rising to CLK Falling		30		150	ns
t_{AC}	A_{8-15} Valid to Leading Edge of Control[1]		270			ns
t_{ACL}	A_{0-7} Valid to Leading of Control		240			ns
t_{AD}	A_{0-15} Valid to Valid Data In				575	ns
t_{AFR}	Address Float after Leading Edge of $\overline{READ}$ ($\overline{INTA}$)				0	ns
t_{AL}	A_{8-15} Valid before Trailing Edge of ALE[1]	C_L=150pF	115			ns
t_{ALL}	A0-7 Valid before Trailing Edge of ALE		90			ns
t_{ARY}	READY Valid from Address Valid	t_{CYC}=320ns			220	ns
t_{CA}	Address (A8 - A15) Valid after Control		120			ns
t_{CC}	Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$) Edge of ALE		400			ns
t_{CL}	Trailing Edge of Control to Leading Edge of ALE		50			ns
t_{DW}	Data Valid to Trailing Edge of $\overline{WRITE}$		420			ns
t_{HABE}	HLDA to Bus Enable				210	ns
t_{HABF}	Bus Float after HLDA				210	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK		110			ns
t_{HDH}	HOLD Hold Time		0			ns
t_{HDS}	HOLD Setup Time to Trailing Edge of CLK		170			ns
t_{INH}	INTR Hold Time		0			ns

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
t_{INS}	INTR, RST and TRAP Setup Time to Falling Edge of CLK		160			ns
t_{LA}	Address Hold Time after ALE		100			ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control		130			ns
t_{LCK}	ALE Low during CLK High		100			ns
t_{LDR}	ALE to Valid Data during Read				460	ns
t_{LDW}	ALE to Valid Data during Write				200	ns
t_{LL}	ALE Width		140			ns
t_{LRY}	ALE to READY Stable				110	ns
t_{RAE}	Trailing Edge of $\overline{READ}$ to Re-Enabling of Address		150			ns
t_{RD}	$\overline{READ}$ (or $\overline{INTA}$) to Valid Data				300	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control		400			ns
t_{RDH}	Data Hold Time After $\overline{READ}$ $\overline{INTA}$		0			ns
t_{RYH}	READY Hold Time		0			ns
t_{RYS}	READY Setup Time to Leading Edge of CLK		110			ns
t_{WD}	Data Valid After Trailing Edge of $\overline{WRITE}$		100			ns
t_{WDL}	LEADING Edge of WRITE to Data Valid				40	ns

Notes:
1. A_{8-15} address specs apply to IO/$\overline{M}$, S_0 and S_1 except A_{8-15} are undifined during T_4 - T_6 of OF cycle whereas IO/$\overline{M}$, S_0, and S_1 are stable.
2. Loading equivalent to 50 pF + 1 TTL input.
3. All timings are measured at output voltage V_L = 0.8 V, V_H = 2.0 V.
4. To calculate timing specifications at other value of t_{CYC} use Table 4.

TABLE 4. BUS TIMING SPECIFICATION AS A T_{CYC} DEPENDENT

t_{AL}	(1/2) T - 45	MIN
t_{LA}	(1/2) T - 60	MIN
t_{LL}	(1/2) T - 20	MIN
t_{LCK}	(1/2) T - 60	MIN
t_{LC}	(1/2) T - 30	MIN
t_{AD}	(5/2 + N) T - 225	MAX
t_{RD}	(3/2 + N) T - 180	MAX
t_{RAE}	(1/2) T - 10	MIN
t_{CA}	(1/2) T - 40	MIN
t_{DW}	(3/2 + N) T - 60	MIN
t_{WD}	(1/2) T - 60	MIN
t_{CC}	(3/2 + N) T - 80	MIN
t_{CL}	(1/2) T - 110	MIN
t_{ARY}	(3/2) T - 260	MAX
t_{HACK}	(1/2) T - 50	MIN
t_{HABF}	(1/2) T + 50	MAX
t_{HABE}	(1/2) T + 50	MAX
t_{AC}	(2/2) T - 50	MIN
t_{L}	(1/2) T - 80	MIN
t_{H}	(1/2) T - 40	MIN
t_{RV}	(3/2) T - 80	MIN
t_{LDR}	(4/2) T - 180	MAX

Note: N is equal to the total WAIT states.

$T = t_{CYC}$

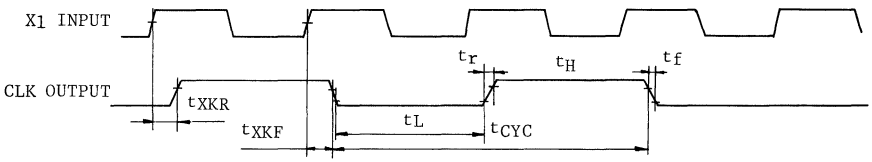


FIGURE 4. CLOCK TIMING WAVEFORM

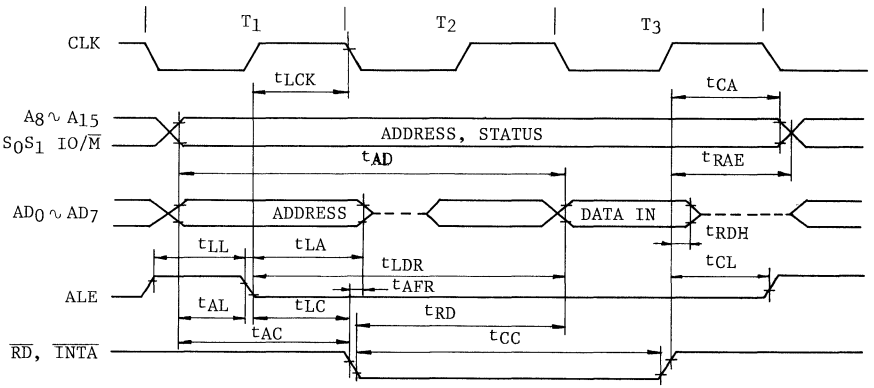


FIGURE 5. READ OPERATION

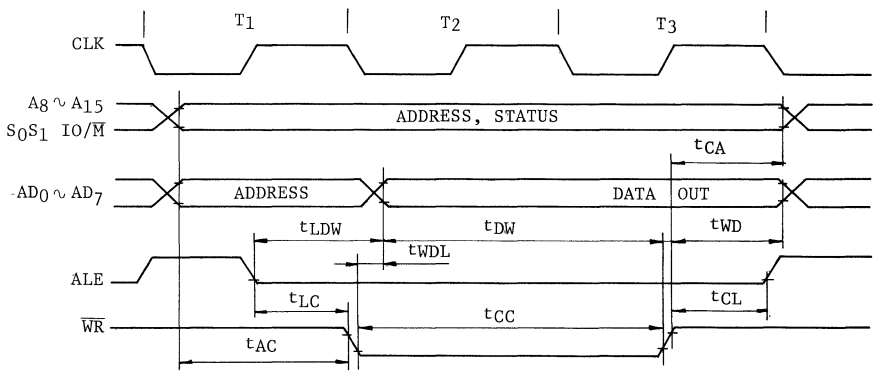


FIGURE 6. WRITE OPERATION

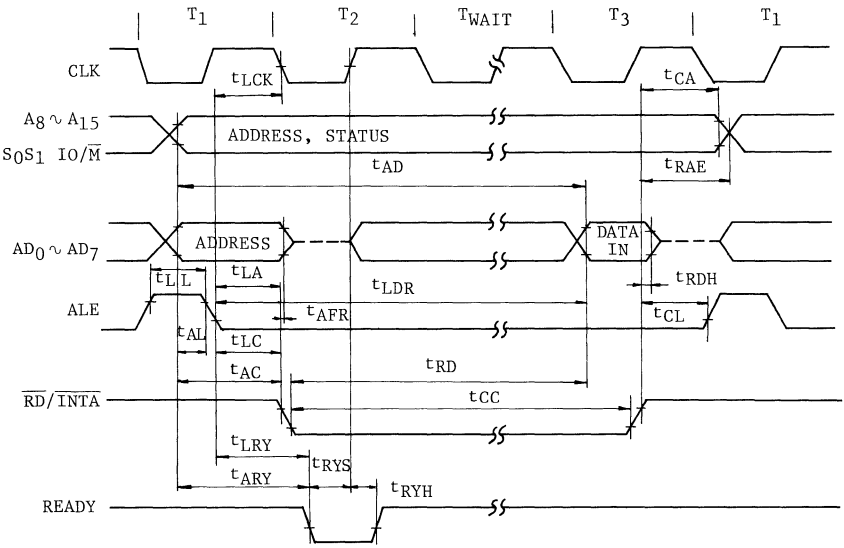


FIGURE 7. READ OPERATION WITH WAIT CYCLE (TYPICAL)
- SAME READY TIMING APPLIES TO WRITE OPERATION

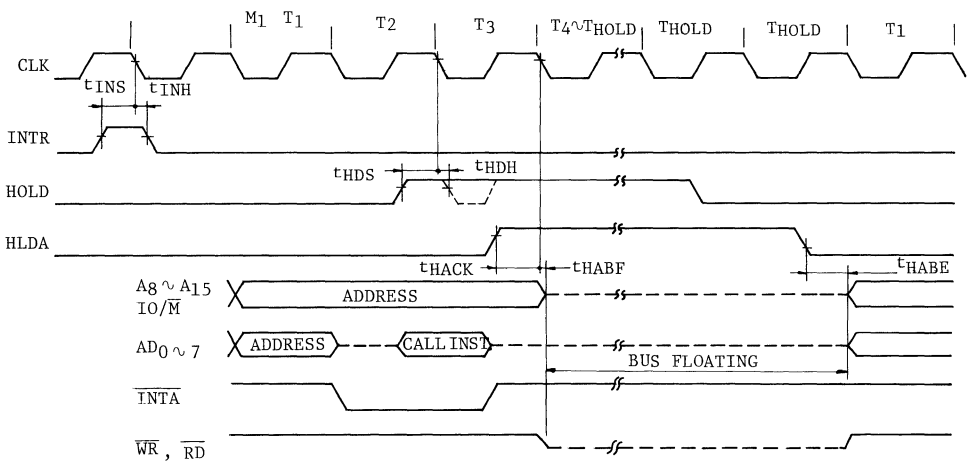
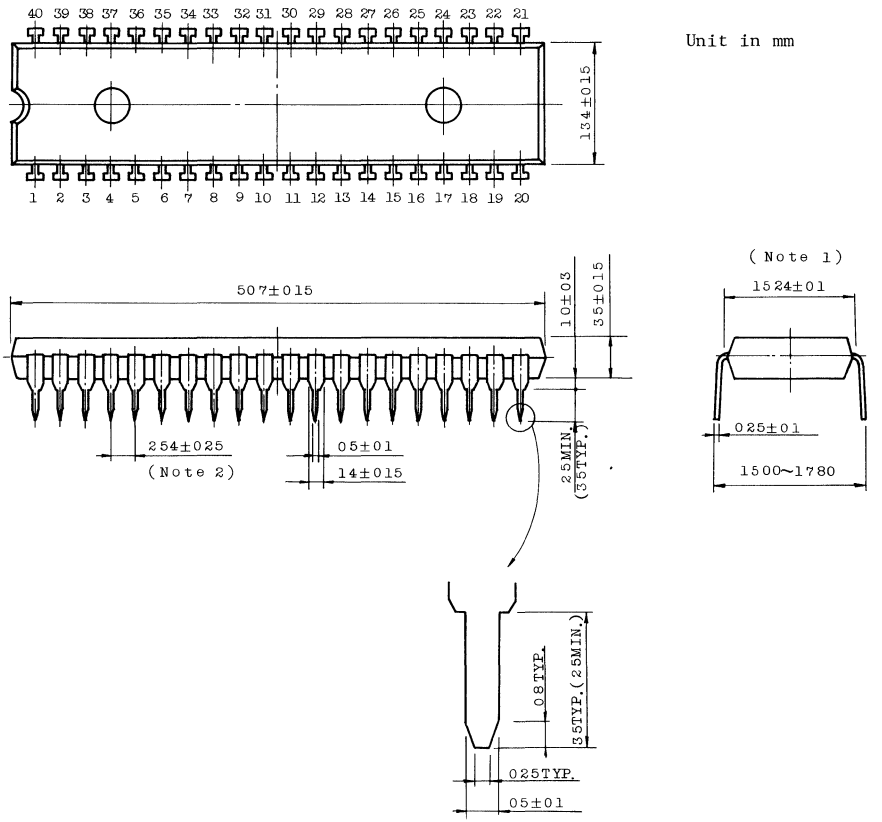


FIGURE 8. INTERRUPT AND HOLD TIMING

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.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8155P

TMP8156P

N-CHANNEL SILICON GATE MOS

2048 BIT STATIC MOS RAM WITH I/O PORTS AND TIMER

GENERAL DESCRIPTION

The TMP 8155P/8156P are RAM including I/O ports and counter/timer on the chip for using in the TLCS-85A microcomputer system. The RAM portion is designed with 2K bit static cells organized as 256 x 8. The 14 bit programmable counter/timer is the down counter. It provides either a square wave or terminal count pulse for the cpu system depending on timer mode.

The I/O portion is consists of 2 programmable 8 bit I/O ports and 1 programmable 6 bit I/O port. The programmable I/O ports can be operated by BASIC MODE and STROBE MODE.

FEATURES

- Compatible with Intel's 8155/8516
- Single +5 V Power Supply
- Access Time: 400 ns (MAX.)
- Internal Address Latch
- 2 Programmable 8 Bit I/O Ports and 1 Programmable 6 Bit I/O Port.
- 256 Word x 8 Bits RAM
- Programmable 14 Bit Binary Counter/Timer
- Multiplexed Address and Data Bus
- Chip Enable Active High (TMP8156P) or Low (TMP8155P)
- 40 pin DIP

PIN CONNECTION (TOP VIEW)

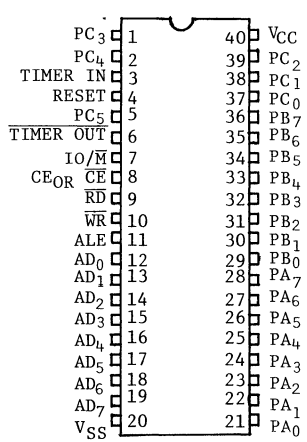


FIGURE 1 TMP8155P/8156P PINOUT DIAGRAM

BLOCK DIAGRAM

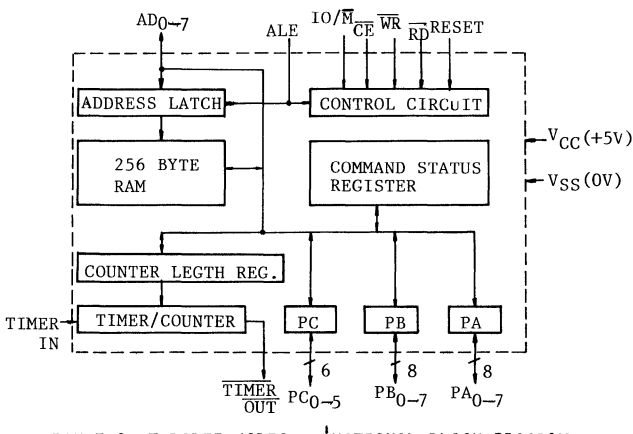


FIGURE 2 TMP8155P/8156P FUNCTIONAL BLOCK DIAGRAM

PIN NAMES AND PIN DESCRIPTION

RESET (INPUT)

The Reset signal is a pulse provided by the TMP8085A to initialize the system. Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two TMP8085A clock cycle times.

$AD_{0\sim7}$ (INPUT / OUTPUT, 3-STATE)

These are 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latche on the falling edge of the ALE. The address can be applied to the memory section or the I/O section depending on the polarity of the IO/$\overline{M}$ input signal. The 8-bit data is either written into the chip or read from the chip depending on the status of $\overline{WR}$ or $\overline{RD}$ input signal.

CE OR $\overline{CE}$ (INPUT)

Chip Enable: On the TMP8155P, this pin is $\overline{CE}$ and is ACTIVE LOW. On the TMP8156P, this pin is CE and is ACTIVE HIGH.

$\overline{RD}$ (INPUT)

Input low on this line with the Chip Enable active enables the $AD_{0\sim7}$ buffers. If IO/$\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/ status register will be read to the AD bus.

$\overline{WR}$ (INPUT)

Input low on this line with the Chip Enable active causes the data on the AD lines to be written to the RAM or I/O ports and command/status register depending on the polarity of IO/$\overline{M}$.

ALE (INPUT)

Address Latch Enable: This control signal latches both the address on the $AD_{0\sim7}$ lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.

IO/$\overline{M}$ (INPUT)

IO/$\overline{\text{Memory}}$ Select: This line selects the memory if low and selects the I/O and command/status register if high.

$PA_{0\sim7}$ (INPUT/OUTPUT, 3-STATE)

These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command Register.

PB0~7 (INPUT/OUTPUT, 3-STATE)

These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command Register.

PC0~5 (INPUT/OUTPUT, 3-STATE)

These 6 pins can function as either input port, output port, or as control signal for PA and PB. Programming is done through the Command Register. When $PC_{0\sim5}$ are used as control signals, they are defined the following:

PC_0 - A INTR (Port A Interrupt)

PC_1 - A BF (Port A Buffer Full)

PC_2 - A $\overline{STB}$ (Port A Strobe)

PC_3 - B INTR (Port B Interrupt)

PC_4 - B BF (Port B Buffer Full)

PC_5 - B $\overline{STB}$ (Port B Strobe)

TIMER IN (INPUT)

This is the input to the counter-timer.

$\overline{\text{TIMER OUT}}$ (OUTPUT)

This pin is the timer output. This output can be either a square wave or a pulse depending on the timer mode.

V_{CC} (Power)
+5 volt supply

V_{SS} (Power)
Ground Reference

FUNCTIONAL DESCRIPTION

PROGRAMMING OF THE COMMAND REGISTER

The command register consists of eitht latches. Four bits (0-3) define the mode of the ports, two bits (4-5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6-7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXX000 during a WRITE operation. The function of each bit of the command byte is defined in FUGURE 3.

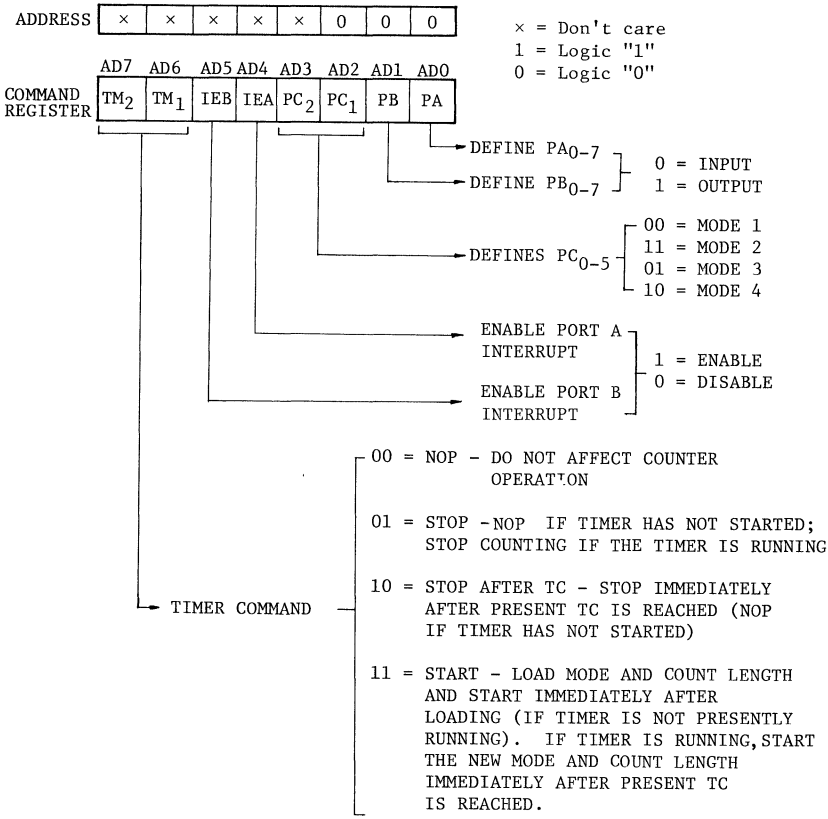


FIGURE 3 COMMAND REGISTER BIT ASSIGNMENT

READING THE STATUS REGISTER

The status register consists of seven latches, one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in FIGURE 4.

Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

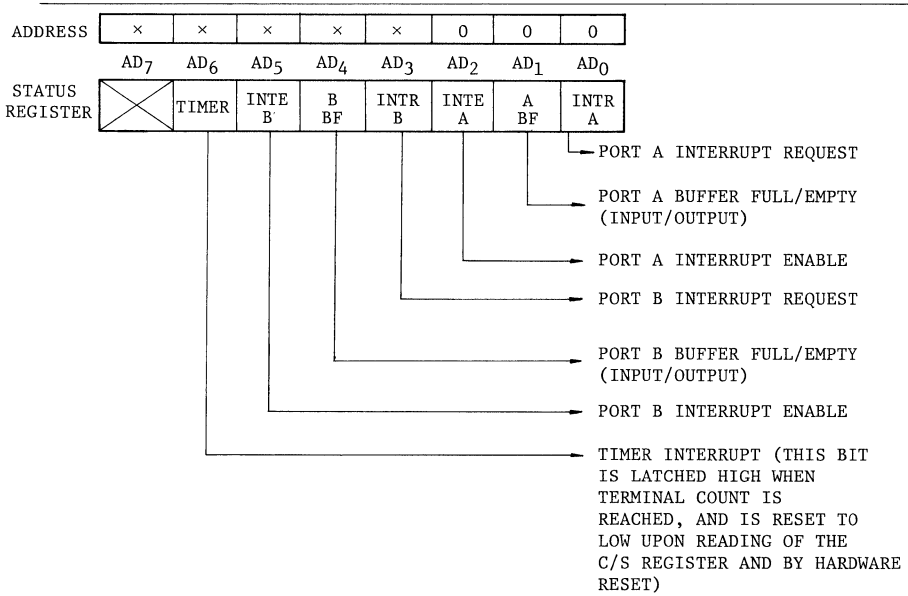


FIGURE 4 STATUS REGISTER BIT ASSIGNMENT

INPUT/OUTPUT SECTION

COMMAND/STATUS REGISTER (C/S)

Both register have the common address ××××××000. When the C/S registers are selected during WRITE operation, a command is written into the C/S register. The contents of this register are not accessible through the pins. When the C/S is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.

- PA Register — This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (See timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.

- PB Register — This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.

- PC Register — This register has the address XXXXX011 and contains only 6-bits. The 6-bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.

When PC_{0-5} is used as a control port, 3-bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the TMP8155P/8156P issues. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. See Table 2.

When the port C is programmed to either MODE 3 or MODE 4, the control signals for PA and PB are initialized as follows:

MODE \ CONTROL	BF	INTR	$\overline{STB}$
INPUT MODE	Low	Low	Input Control
OUTPUT MODE	Low	High	Input Control

To summarize, the register's assignments are shown TABLE 1.

TABLE 1 I/O PORT ADDRESSING SCHEME

I/O ADDRESS								PINOUTS	SELECTION	NO. OF BITS
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0			
X	X	X	X	X	0	0	0	Internal	Command/Status Register	8
X	X	X	X	X	0	0	1	PA_{0-7}	General Purpose I/O Port A	8
X	X	X	X	X	0	1	0	PB_{0-7}	General Purpose I/O Port B	8
X	X	X	X	X	0	1	1	PC_{0-7}	General Purpose I/O Port or Control	6
X	X	X	X	X	1	0	0		Low-Order 8 bits of Timer Count	
X	X	X	X	X	1	0	1		High 6 bits/2 bits of Timer Count	

TABLE 2 TABLE OF PORT CONTROL ASSIGNMENT

Pin	MODE 1	MODE 2	MODE 3	MODE 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{\text{STB}}$ (Port A strobe)	A $\overline{\text{STB}}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{\text{STB}}$ (Port B Strobe)

TIMER SECTION

The timer is a 14-bit down-counter that counts the 'timer input' pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register.

To program the timer, the COUNT LENGTH REGISTER is loaded first, one byte at a time, by selecting the timer addresses. Bits 0-13 will specify the length of the next count and bits 14-15 will specify the timer output mode. The value loaded into the count length register can have any value from 2_H through $3FFF_H$ in bits 0-13.

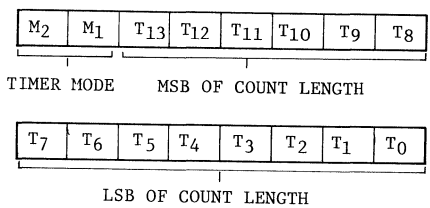


FIGURE 5 TIMER FROMAT

There are four timer modes which are defined by M2 and M1.

M2	M1	
0	0	——— Put out low during second half of count.
0	1	——— Continuous square wave; The period of the square-wave equals the count length programmed with automatic reload at terminal count.
1	0	——— Single pulse upon TC being reached.
1	1	——— Continuous pulses.

Note: In case of an odd-numbered count, the first half-cycle of the square-wave output, which is high, is one count longer than the second (low) half-cycle as shown in FIGURE 6.

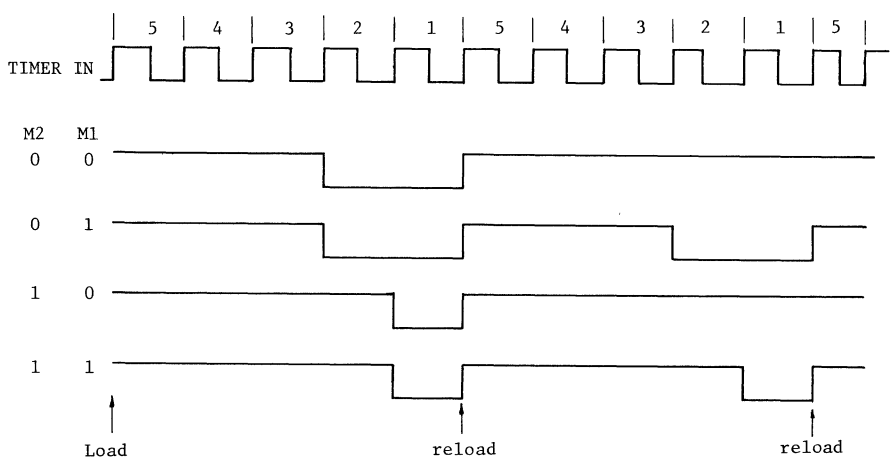


FIGURE 6 ASYMMETRICAL SQUARE-WAVE OUTPUT RESULTING FROM COUNT OF 5

Bits 6-7 (TM2 and TM1) of command register contents are used to start and stop the counter. There are four commands to choose from;

TM2	TM1	
0	0	NOP: Do not affect counter operation.
0	1	STOP: NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC: Stop immediately after present TC is reached. (NOP if timer has not started)
1	1	START: Load mode and count length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and count length immediately after present TC is reached.

Note that while the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be used by the counter, you must issue a START command to the counter. This applies even though you may only want to change the count and use the previous mode.

The counter in the TMP8155P/8156P is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore you must issue a START command via the C/S register, because counting cannot begin following RESET.

Please note that the timer circuit on the TMP8155P/8156P chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary). After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count.
2. Read in the 16-bit value from the count length registers.
3. Reset the upper two mode bits.
4. Reset the carry and rotate right one position all 16 bits through carry.
5. If carry is set, add 1/2 of the full original count (1/2 full count-1 if full count is odd.)

Note: If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the TMP8155P/8156P always counts out the right number of pulses in generating the $\overline{\text{TIMER OUT}}$ waveforms.

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{CC}	V_{CC} Supply Voltage with Respect to V_{SS}	-0.5V to +7.0V
V_{IN}	Input Voltage with Respect to V_{SS}	-0.5V to +7.0V
V_{OUT}	Output Voltage with Respect to V_{SS}	-0.5V to +7.0V
P_D	Power Dissipation	1.5W
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 ± 5%

Symbol	Parameter	Test Conditions	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_{OL}	Output Low Voltage	I_{OL} = 2mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = -400μA	2.4			V
I_{IL}	Input Leakage	V_{IN} = V_{CC} to 0V			±10	μA
I_{LO}	Output Leakage Current	0.45V ≤ V_{OUT} ≤ V_{CC}			±10	μA
I_{CC}	V_{CC} Supply Current				180	mA
$I_{IL(CE)}$	Chip Enable Leakage 8155 8156	V_{IN} = V_{CC} to 0V.			 +100 -100	 μA μA

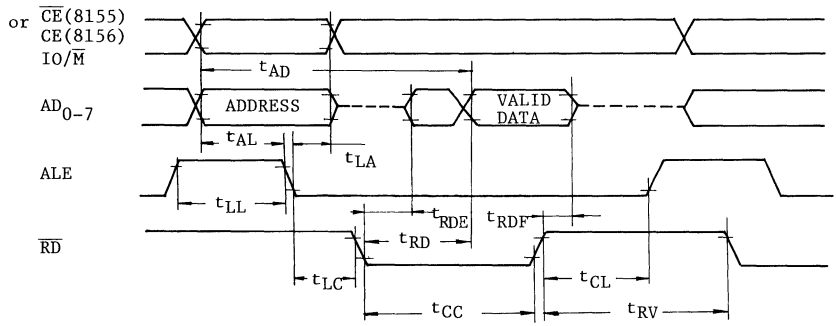
A.C. CHARACTERISTICS

T_A=0°C to +70°C, V_{CC}=+5V±5%

Symbol	Parameters	Test Condition	Min.	Typ.	Max.	Units
t_{AL}	Address to Latch Set Up Time	150pF Load	50			ns
t_{LA}	Address Hold Time after Latch		80			ns
t_{LC}	Latch to READ/WRITE Control		100			ns
t_{RD}	Valid Data out Delay from READ Control				170	ns
t_{AD}	Address Stable to Data Out Valid				400	ns
t_{LL}	Latch Enable Width		100			ns
t_{RDF}	Data Bus Float After READ		0		100	ns
t_{CL}	READ/WRITE control Latch Enable		20			ns
t_{CC}	READ/WRITE Control Width		250			ns
t_{DW}	Data In to WRITE Set Up Time		150			ns
t_{WD}	Data in Hold Time After WRITE		0			ns
t_{RV}	Recovery Time Between Controls		300			ns
T_{WP}	WRITE to Port Output				400	ns
T_{PR}	Port Input Setup Time		70			ns
t_{RP}	Port Input Hold Time		50			ns
t_{SBF}	Strobe to Buffer Full				400	ns
t_{SS}	Strobe Width		200			ns
t_{RBE}	READ to Buffer Empty				400	ns
t_{SI}	Strobe to INTR On				400	ns
t_{RDI}	READ to INTR Off				400	ns
t_{PSS}	Port Setup Time to Strobe		50			ns
t_{PHS}	Port Hold Time After Strobe		120			ns
t_{SBE}	Strobe to Buffer Empty				400	ns
t_{WBF}	WRITE to Buffer Full				400	ns
t_{WI}	WRITE to INTR Off				400	ns
t_{TL}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ Low				400	ns
t_{TH}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ High				400	ns
t_{RDE}	Data Bus Enable from READ Control		10			ns
t_L	TIMER-IN Low Time		80			ns
t_H	TIMER-IN High Time		120			ns

TIMING WAVEFORMS

A. READ CYCLE



B. WRITE CYCLE

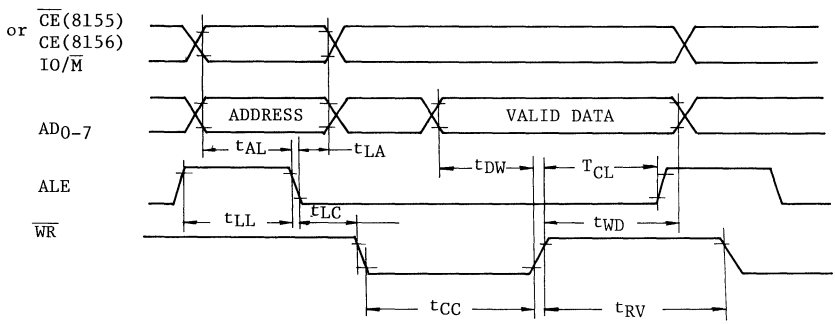
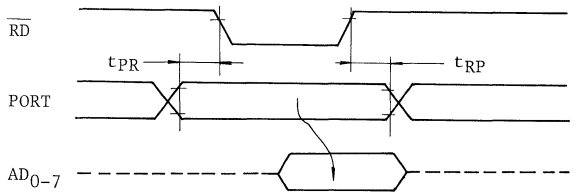


FIGURE 7 READ/WRITE TIMING DIAGRAMS

A. BASIC INPUT MODE



B. BASIC OUTPUT MODE

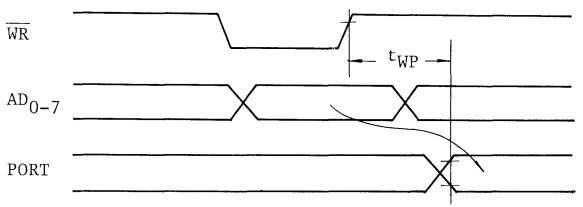
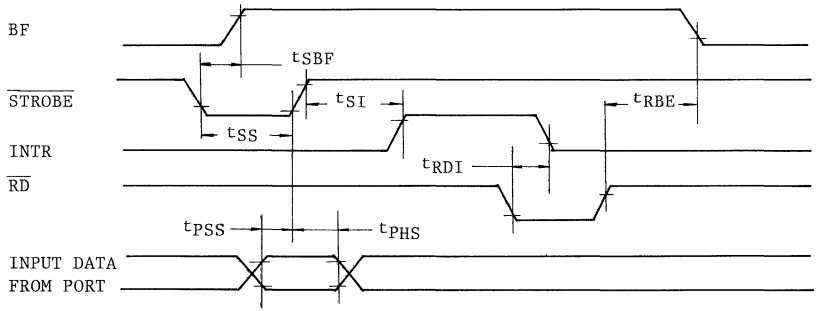


FIGURE 8 BASIC I/O TIMING WAVEFORM

A. STROBED INPUT MODE



B. STROBED OUTPUT MODE

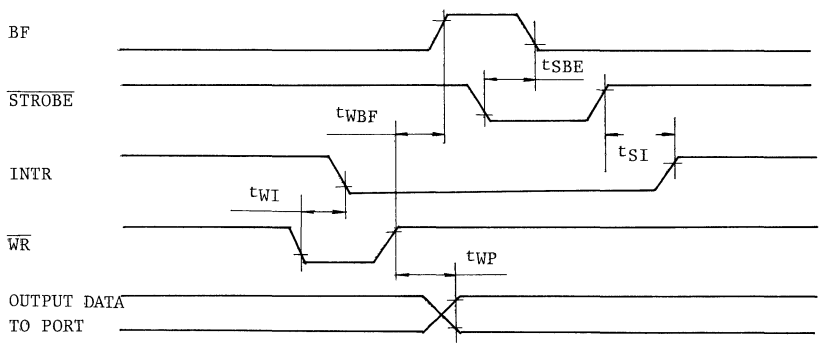
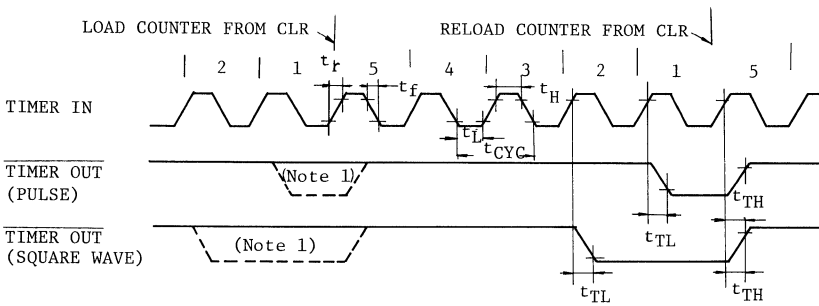


FIGURE 9 STROBED I/O TIMING WAVEFORM

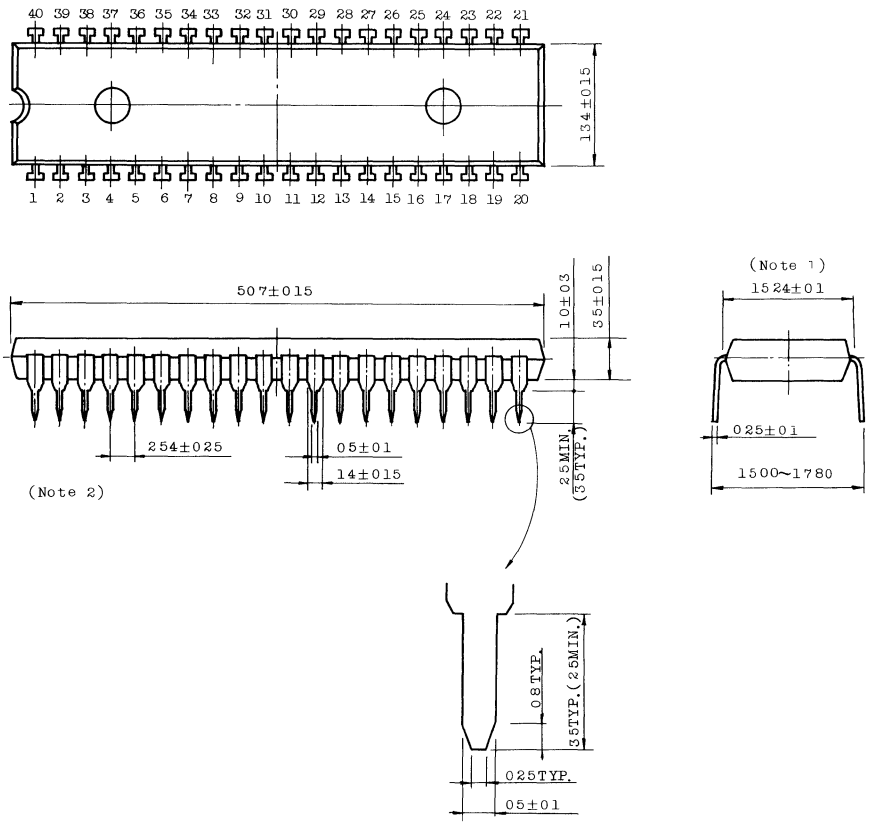


Note 1: The timer output is periodic if in an automatic reload mode (M_1 Mode Bit = 1)

FIGURE 10 TIMER OUTPUT WAVEFORM COUNTDOWN FROM 5 TO 1

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.

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP8251AP
Silicon Monolithic
N Channel Silicon Gate MOS

PROGRAMMABLE COMMUNICATION INTERFACE

GENERAL DESCRIPTION

The TMP8251AP is the industry standard Universal Synchronous/ Asynchronous Receiver/Transmitter (USART) that is fabricated using N-channel silicon gate MOS technology.

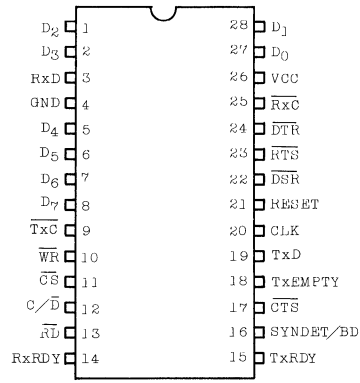
The TMP8251A is mainly used for 8-bit microcomputer extension systems, which require serial data communications.

FEATURES

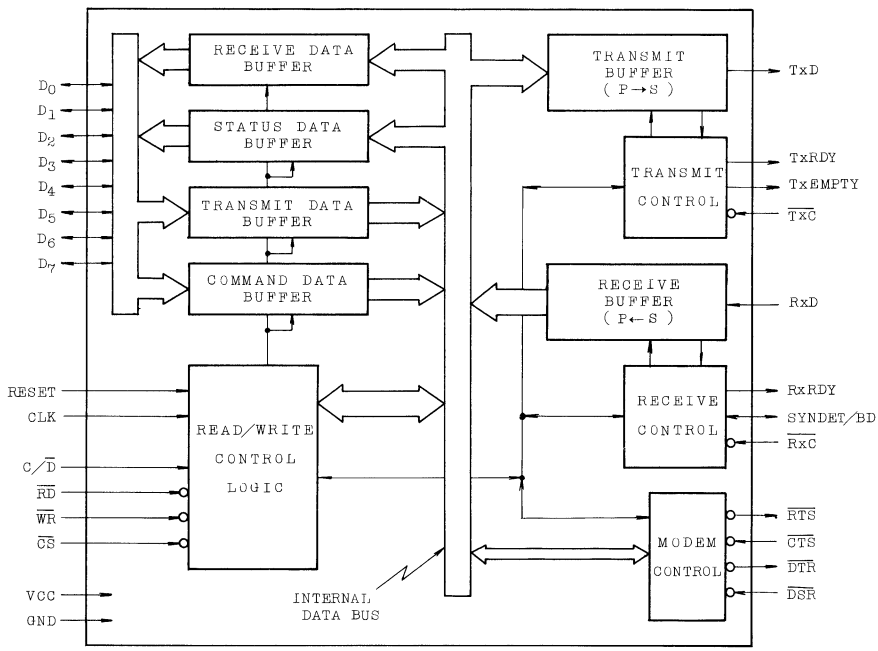
- Synchronous and Asynchronous Operation
  - Synchronous:
    - 5-8 Bit Characters
    - Internal or External Character Synchronization
    - Single or Double Character Synchronization (Internal)
    - Automatic Sync Insertion
  - Asynchronous:
    - 5-8 Bit Characters
    - Clock Rate - 1,16 or 64 Times Baud Rate
    - Break Character Generation
    - 1, $1\frac{1}{2}$, or 2 Stop Bits
    - False Start Bit Detection
    - Automatic Break Detect and Handling
- Baud Rate DC to 64K Baud (Synchronous)

  DC to 19.6K Baud (Asynchronous)
- Full-Duplex, Double-Buffered, Transmitter and Receiver
- Error Detection-Parity, Overrun and Framing
- Single +5V Supply
- Compatible with Intel's 8251A/S2657

PIN CONNECTIONS (Top View)



BLOCK DIAGRAM



PIN NAMES AND PIN DESCRIPTIONS

- Interface Signals to CPU (Main System)

$D_0 \sim D_7$ (Input/Output)

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received through the buffer upon execution of Input or Output Instructions of the CPU. Control Words, Command Words and Status Information are also transferred through the Data Bus Buffer.

$\overline{WR}$ (Input)

A "low" level signal on this input informs the 8251A that the CPU is Writing Data or Control Words to the 8251A.

$\overline{RD}$ (Input)

A "low" level signal on this input informs the 8251A that the CPU is Reading Data or Status Information from the 8251A.

$\overline{CS}$ (Input)

A "low" level signal on this input selects the 8251A. No reading or writing operation will occur unless the device is selected. When $\overline{CS}$ is "high" the Data Bus is in the floating state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

$C/\overline{D}$ (Input)

This input signal, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, imforms the 8251A that the word on the Data Bus is either a Data Character, Control Word or Status Infromation. A "high" level signal means Control or Status, a "low" level signal means Data.

$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A Receive DATA Buffer → DATA Bus
0	1	0	0	8251A Transmit DATA Buffer ← DATA Bus
1	0	1	0	8251A Status DATA Buffer → DATA Bus
1	1	0	0	8251A Command DATA Buffer ← DATA Bus
×	1	1	0	DATA Bus is in floating state.
×	×	×	1	"

CLK (Input)

The CLK input is used to generate internal device timing. No external input or output is referenced to CLK, but the frequency of CLK must be greater than 30 times the Receiver or Transmitter Data Bit Rates ($\overline{RxC}$ or $\overline{TxC}$) in Synchronous Operation, and greater than 4.5 times the Receiver Data Bit Rates ($\overline{RxC}$) in Asynchronous Operation.

RESET (Input)

A "high" level signal on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" untill a new set of Control Words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 tcy.

- MODEM Control Signals

$\overline{DSR}$ (Input)

The $\overline{DSR}$ input signal is a general purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read Operation. The $\overline{DSR}$ input is normally used to test MODEM conditions such as Data Set Ready signal.

$\overline{DTR}$ (Output)

The $\overline{DTR}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction Word. The $\overline{DTR}$ output signal is normally used for MODEM control such as Data Terminal Ready or Rate Select signal.

$\overline{RTS}$ (Output)

The $\overline{RTS}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction Word. The $\overline{RTS}$ output signal is normally used for MODEM control such as Request to Send signal.

$\overline{CTS}$ (Input)

A "low" level signal on this input enables the 8251A to transmit serial data, if the Tx Enable Bit in the Command Byte is set to a "one" (TxEN=1). If either a Tx Enable off (TxEN=0) or CTS off ($\overline{CTS}$=1) condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable Command before shutting down.

• Transmit Control Signals

$\overline{TxC}$ (Input)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous Transmission Mode, the Baud Rate (1x) is equal to the $\overline{TxC}$ frequency. In Asynchronous Transmission Mode the baud rate is a fraction of the actual $\overline{TxC}$ frequency. A portion of the Mode Instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{TxC}$.

For Example:

If Baud Rate equals 110 Baud,

$\overline{TxC}$ = 110 Hz (1x)

$\overline{TxC}$ = 1.76 KHz (16x)

$\overline{TxC}$ = 7.04 KHz (64x)

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251A.

TxD (Output)

This line is used to transmit the serial data. Serial output data on TxD is changed from parallel data to serial data in accordance with the format specified by the Control Words.
TxD line will be held in the marking state ('1' level) immediately on one of the followings.

- Master Reset
- Tx Disable (TxEN=0)
- CTS signal is high ($\overline{CTS}$=1)
- TxEMPTY signal is high (TxEMPTY=1)

TxRDY (Output)

This output informs the CPU that the transmitter is ready to accept a Data Character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by Tx Disable(TxEN=0), or, for polled Operation, the CPU can check TxRDY using a Status Read Operation. TxRDY is automatically reset by the trailing edge of $\overline{WR}$ when a Data Character is loaded from the CPU. The TxRDY pin output status (TxRDY (pin)) is different from the TxRDY status bit status (TxRDY (status bit)) as follows.

TxRDY (status bit) = (Transmit Data Buffer Empty)

TxRDY (pin) = (Transmit Data Buffer Empty)•(CTS=0)•(TxEN=1)

TxEMPTY (Output)

The TxEMPTY output will go "high" when the 8251A has no characters to send. It resets upon receiving a character from the CPU if the transmitter is enabled.

In Synchronous Mode, a "high" level signal on this output indicates that a Character has not been loaded and the SYNC Character or Characters are about to be or are being transmitted automatically as "fillers". TxEMPTY does not go "low" when the SYNC characters are being shifted out.

- Receive Control Signals

$\overline{RxC}$ (Input)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{RxC}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{RxC}$ frequency. A portion of the Mode Instruction selects this factor; 1, 1/16 or 1/64 the $\overline{RxC}$.

For Example:

if Baud Rate equals 2400 Baud,

$\overline{RxC}$ = 2.4 KHz (1x)

$\overline{RxC}$ = 38.4 KHz (16x)

$\overline{RxC}$ = 153.6 KHz (64x)

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

RxD (Input)

This line is used to receive the serial data. Serial input data on this line is changed to parallel data in accordance with the format specified by the Control Words, and then transfered to the Recive Data Buffer.

RxRDY (Output)

This output indicates that the 8251A contains a Data Character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU, or, for Polled Operation, the CPU can check the condition of RxRDY using a Status Read Operation.
Rx Enable off both masks and holds RxRDY in the Reset Condition.

SYNDET/BD (Input/Output)

This pin is used for SYNDET in Synchronous Mode and may be used as either input or output, programmable through the Control Word. It is reset to output mode "low" upon RESET. When used as an Output (Internal Sync Mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC Character in the Receive Mode. If the 8251A is programmed to use Double Sync Characters then SYNDET will go "high" in the middle of the last bit of the second SYNC Character. SYNDET is automatically reset upon a Status Read Operation. When used as an Input (External Sync Mode), a positive going signal will cause the 8251A to start assembling Data Characters on the rising edge of the next $\overline{RxC}$.

In Asynchronous Mode this pin is used for BD.
This output will go "high" whenever the receiver remains "low" through two consecutive Stop Bit Sequences (including the Start Bits, Data Bits, and Parity Bits). Break Detect may also be read as a Status Bit.
It is reset only upon a Master Chip Reset or Rx Data returning to a "one" state.
But, if the Rx Data returns to a "one" State during the last bit of the next character after the Break, Break Detect does not always reset.

- Power Supply
  - V_{CC} (Power)
    +5 Volt supply
  - GND (Power)
    0 Volt supply

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	Power 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 (Ta=70°C)	1W
T_{solder}	Soldering Temperature (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_{opr}=0°C to 70°C, V_{CC}=5V ±5%, GND=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		2.2	-	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL}=2.2mA	-	-	0.45	V
V_{OH}	Output High Voltage	I_{OH}=-400μA	2.4	-	-	V
V_{OFL}	Output Leak Current	$0.45V \leq V_{OUT} \leq V_{CC}$	-	-	±10	μA
I_{IL}	Input Leak Current	$0.45V \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{CC}	Power Supply Current	All Outputs="High"	-	-	100	mA

A.C. CHARACTERISTICS T_{opr}=0°C to 70°C, V_{CC}=5V±5%, GND=0V, Unless otherwise noted.

BUS READ CYCLE TIMING Note 1)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{AR}	$\overline{CS}$, C/$\overline{D}$ Set-up Time for $\overline{RD}$		50	-	-	ns
t_{RA}	$\overline{CS}$, C/$\overline{D}$ Hold Time for $\overline{RD}$		50	-	-	ns
t_{RR}	$\overline{RD}$ Pulse Width		250	-	-	ns
t_{RD}	Data Delay Time for $\overline{RD}$ Note 2)	C_L=150pF Note 3)	-	-	250	ns
t_{DF}	Data Hold Time for $\overline{RD}$		10	-	100	ns

BUS WRITE CYCLE TIMING Note 1)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{AW}	$\overline{CS}$, C/$\overline{D}$ Set-up Time for $\overline{WR}$		50	-	-	ns
t_{WA}	$\overline{CS}$, C/$\overline{D}$ Hold Time for $\overline{WR}$		50	-	-	ns
t_{WW}	$\overline{WR}$ Pulse Width		250	-	-	ns
t_{DW}	Data Set Up Time for $\overline{WR}$		150	-	-	ns
t_{WD}	Data Hold Time for $\overline{WR}$		50	-	-	ns
t_{RV}	Recovery Time Between WRITES	Note 4)	6	-	-	t_{cyc}

OTHER TIMING

SYMBOL	PARAMETER		MIN.	TYP.	MAX.	UNIT	
t_{cyc}	Clock Period Note 5), 6)		320	-	1350	ns	
t_H	Clock High Level Width		140	-	$t_{cyc}-90$	ns	
t_L	Clcok Low Level Width		90	-	-	ns	
t_R, t_F	Clock Rise and Fall Time		-	-	20	ns	
t_{DTx}	TxD Delay Time from Falling Edge of $\overline{TxC}$		-	-	1	μs	
f_{Tx}	Transmitter Input Clock Frequency	1x Baud Rate	DC	-	64	kHz	
		16x Baud Rate	DC	-	310		
		64x Baud Rate	DC	-	615		
t_{TPH}	Transmitter Input Clock High Level Width	1x Baud Rate	12	-	-	t_{cyc}	
		16x,64x Baud Rate	1	→	-		
t_{TPL}	Transmitter Input Clock Low Level Width	1x Baud Rate	15	-	-	t_{cyc}	
		16x,64x Baud Rate	3	-	-		
f_{Rx}	Receiver Input Clock Frequency	1x Baud Rate	DC	-	64	kHz	
		16x Baud Rate	DC	-	310		
		64x Baud Rate	DC	-	615		
t_{RPH}	Receiver Input Clock High Level Width	1x Baud Rate	12	-	-	t_{cyc}	
		16x,64x Baud Rate	1	-	-		
t_{RPL}	Receiver Input Clock Low Level Width	1x Baud Rate	15	-	-	t_{cyc}	
		16x,64x Baud Rate	3	-	-		
t_{TxRDY}	TxRDY Pin Delay Time from Center of Last Bit		-	-	8	t_{cyc}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY Clear Delay Time from Trailing Edge of $\overline{WR}$		-	-	6	t_{cyc}	Note 7
t_{RxRDY}	RxRDY Pin Delay Time from Center of Last Bit		-	-	24	t_{cyc}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY Clear Delay Time from Leading Edge of $\overline{RD}$		-	-	6	t_{cyc}	Note 7
t_{IS}	Internal SYNDET Delay Time from Rising Edge of $\overline{RxC}$		-	-	24	t_{cyc}	Note 7
t_{ES}	External SYNDET Set-Up Time fore Falling Edge of $\overline{RxC}$		16	-	-	t_{cyc}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay Time from Center of Last Bit		20	-	-	t_{cyc}	Note 7
t_{WC}	Control Delay Time from Rising Edge of $\overline{WR}$ (TxEN, $\overline{DTR}$, $\overline{RTS}$)		8	-	-	t_{cyc}	Note 7
t_{CR}	$\overline{DSR}$, $\overline{CTS}$ Set-Up Time for $\overline{RD}$		20	-	-	t_{cyc}	Note 7

Notes:

1) AC Test Conditions: Output measuring point V_{OH}=2.0V, V_{OL}=0.8V
   Input supply level V_{IH}=2.4V, V_{IL}=0.45V

2) Assumes that Address is valid before the falling edge of $\overline{RD}$.

3) C_L means load capacitance.

4) This recovery time is defined only for Mode Initialization. Write Data is allowed only when TxRDY=1. Recovery Time between Writes for Asynchronous Mode is 8 tcy and for Synchronous Mode is 16 tcy.

5) The TxC and RxC frequencies have the following limitations with respect to CLK:

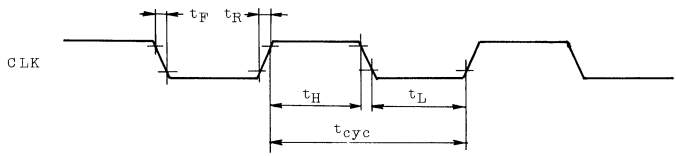
   For 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30tcy)$
   For 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5tcy)$

6) Minimum Reset Pulse Width is 6 tcy. System Clock must be running during Reset.

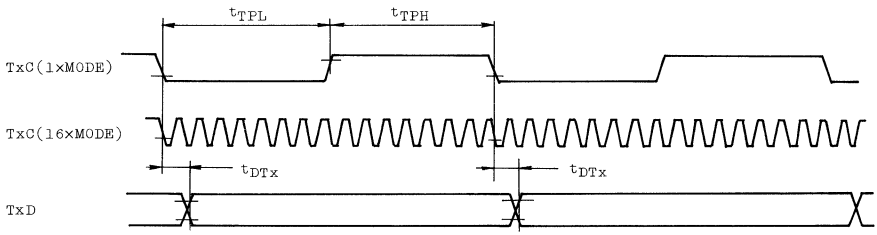
7) Status up data can have a maximum delay of 28 clock periods from the event affecting the status.

TIMING WAVEFORMS

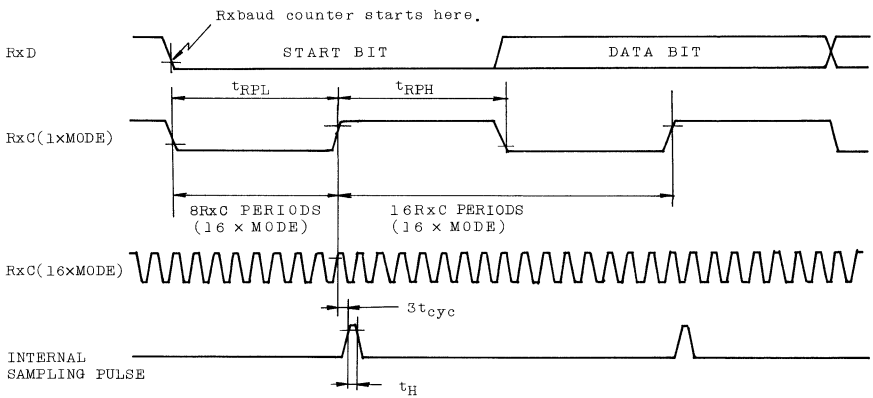
SYSTEM CLOCK INPUT



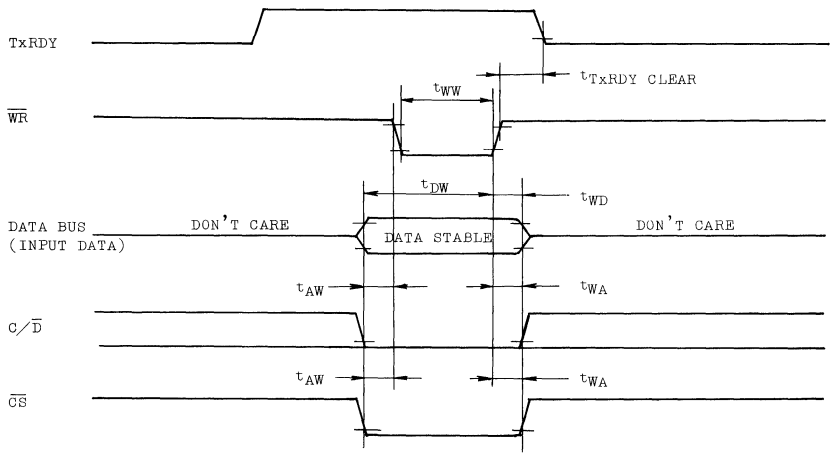
TRANSMITTER CLOCK AND DATA



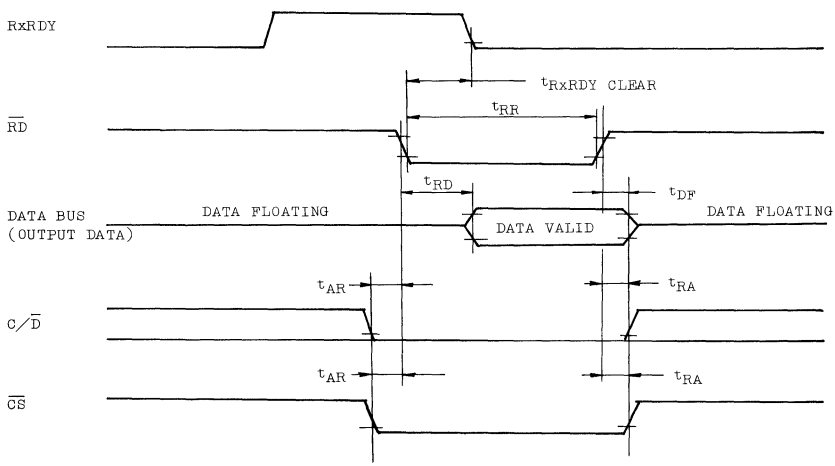
RECEIVER CLOCK AND DATA



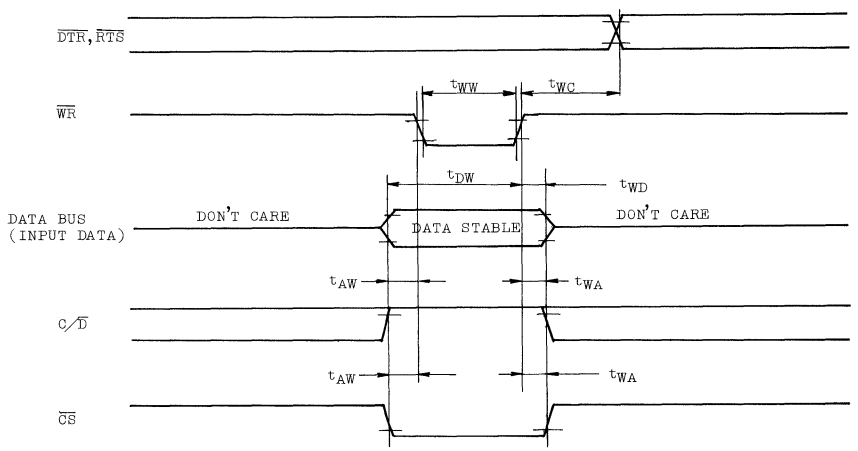
WRITE DATA CYCLE (CPU → 8251A)



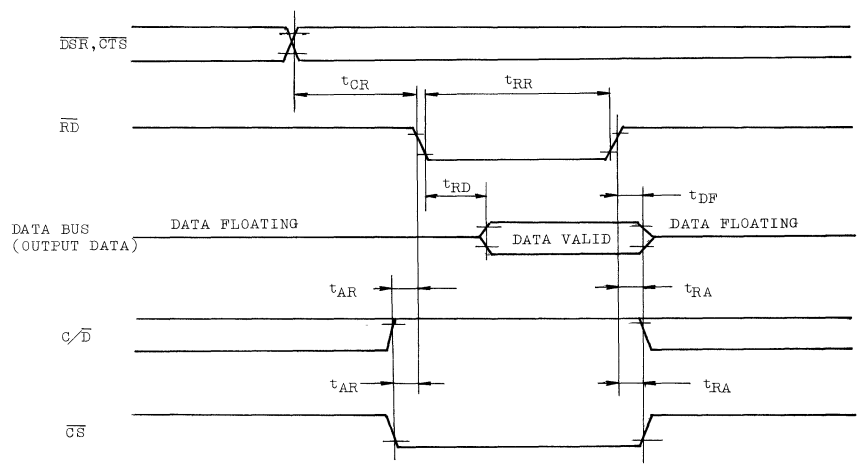
READ DATA CYCLE (8251A → CPU)



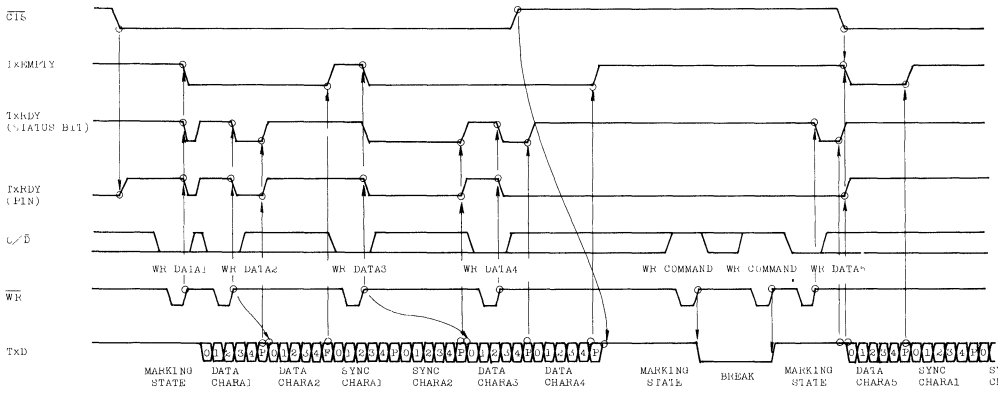
WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → 8251A)



READ CONTROL OR INPUT PORT CYCLE (8251A → CPU)

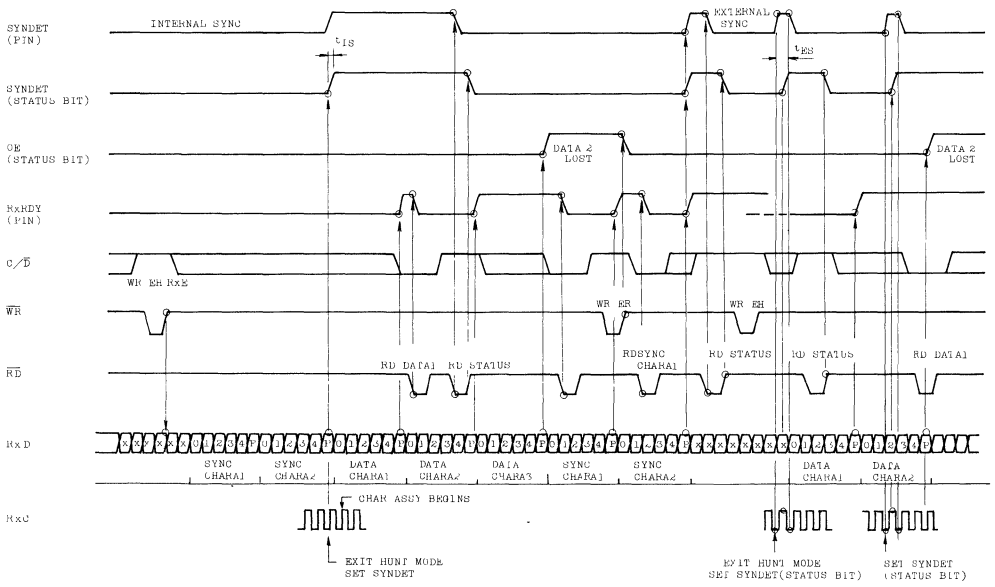


TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



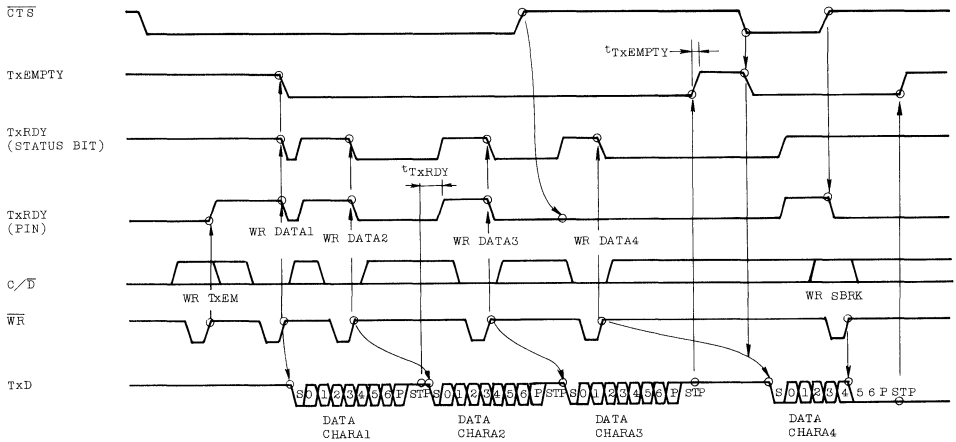
EXAMPLE FORMAT : 5 BIT CHARACTER WITH PARITY 2 SYNC CHARACTERS

RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



EXAMPLE FORMAT : 5 BIT CHARACTER WITH PARITY 2 SYNC CHARACTERS

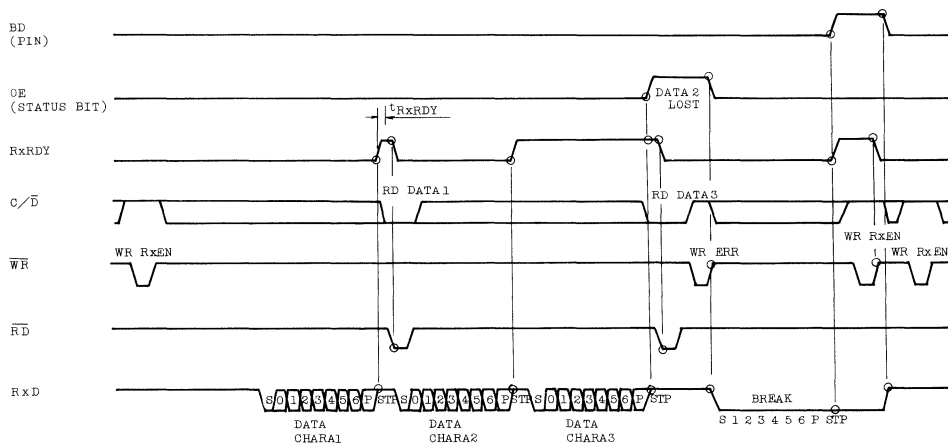
TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)



EXAMPLE FORMAT : 7 BIT CHARACTER & 2 STOP BITS.

Note : TxRDY(PIN) = (Transmit Data Buffer is empty) (TxEN=1) · (CTS=0)
TxRDY(STATUS BIT) = (Transmit Data Buffer is empty)

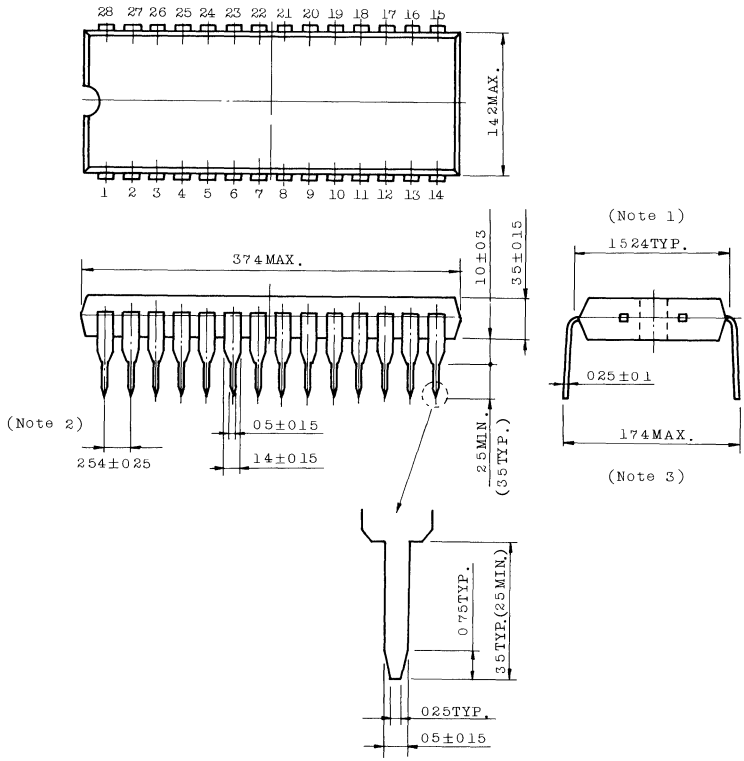
RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



EXAMPLE FORMAT : 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

OUTLINE DRAWINC

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.28 leads.

3. This dimension is to outside of leads.

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8253P-5

N-CHANNEL SILICON GATE MOS

PROGRAMMABLE INTERVAL TIMER

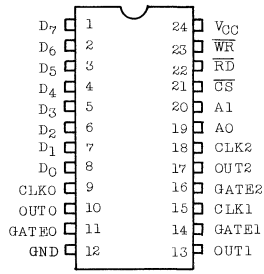
GENERAL DESCRIPTION

The TMP8253-5 is a programmable counter/timer chip designed for use as the TLCS-85A microcomputer peripheral. It is organized as 3 independent 16 bit counters, each operates with a count rate of up to 2.5MHz. All modes of operation are software programmable.

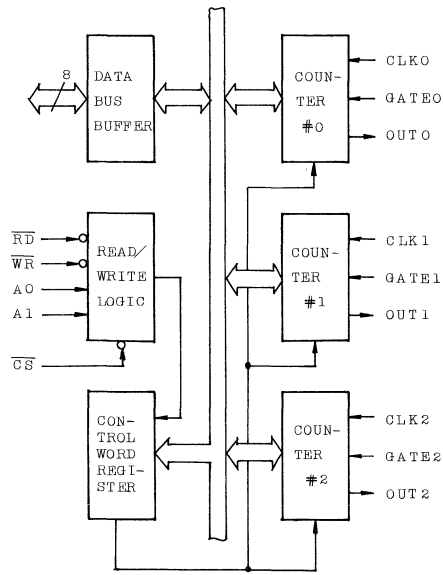
FEATURES

- Count Binary or BCD
- 3 Independent 16 Bit Counters
- Single +5V Supply
- Count rate DC to 2.5MHz
- 6 programmable Counter Modes
- Compatible with Intel's 8253-5

PIN CONNECTIONS



BLOCK DIAGRAM



PIN NAMES

$D_7 \sim D_0$	Data Bus (8 bit)
CLK N	Counter Clock Input
GATE N	Counter Gate Input
OUT N	Counter Output
$\overline{RD}$	Read Counter
$\overline{WR}$	Write Counter
$\overline{CS}$	Chip Select
$A_0 \sim A_1$	Counter Select
VCC	+5V
GND	Ground (0V)

PIN NAMES AND PIN DESCRIPTION

GND (Power Supply)

Ground.

V_{CC} (Power Supply)

+5V during operation.

$\overline{CS}$ (Input)

A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the TMP8253-5. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

A0, A1 (Input)

These inputs acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. This pin is used to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{WR}$ (Input)

A low on this pin when $\overline{CS}$ is low enables the TMP8253-5 to accept mode information or loading counters from the CPU.

$\overline{RD}$ (Input)

A low on this pin when $\overline{CS}$ is low enables the TMP8253-5 to release a counter value onto the data bus for the CPU.

D0 ∿ D7 (Input/Output)

Bidirectional Data Bus. Mode information, the information loading counter or the count values are transfered via this data bus.

CLK0 ∿ CLK2 (Input)

Clock inputs to counters. Falling edge on this pin enables the counter to count down.

GATE0 ∿ GATE2 (Input)

Gate inputs to counters. The function of this pin differs by the mode selection of counter operation.

Out0 ∿ Out2 (Output)

Outputs from the counters. The output signal from this pin differs by the mode selection of counter operation.

FUNCTIONAL DESCRIPTION

[Block Description]

Data Bus Buffer

This is 3-state, bi-directional, 8 bit buffer used for interfacing the TMP8253-5 to the system data bus. The Data Bus Buffer has three functions as follows. Programming the MODEs of the TMP8253-5, Loading the count registers, and Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter #0
0	1	0	0	1	Load Counter #1
0	1	0	1	0	Load Counter #2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter #0
0	0	1	0	1	Read Counter #1
0	0	1	1	0	Read Counter #2
0	0	1	1	1	Data Bus is in High-impedance state
1	x	x	x	x	
0	1	1	x	x	

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register. No reading of the contents of the Control Word Register is available.

Counter #0, Counter #1, Counter #2

These three blocks are identical so only a single counter will be described. Each counter consists of a single, 16 bit, presettable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES (Six MODES: MODE 0 to MODE 5) stored in the Control Word Register. Also the control word handles the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications. Special commands and logic are included in the TMP8253-5 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

[MODE Definition]

MODE 0: Interrupt on Terminal Count.

The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.

(2) Write 2nd byte starts the new count.

MODE 1: Programmable One Shot.

The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generater

Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator.

Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count.

This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for $(N + 1)/2$ counts and low for $(N - 1)/2$ counts.

MODE 4: Software Triggered Strobe.

After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

If the count register is reloaded between output pulses, counting will continue from the new value. The count will be inhibited while the gate input is low. Reloading the counter register will restart counting beginning with the new number.

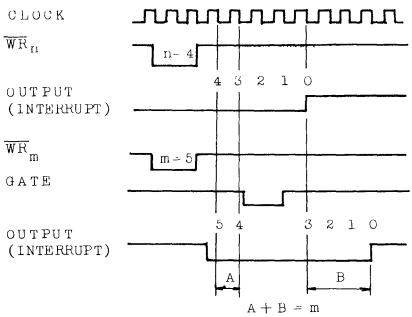
MODE 5: Hardware Triggered Strobe.

The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

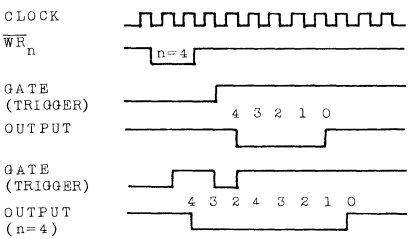
Status / Modes	Low or Going Low	Rising	High
0	Disables counting	-	Enables counting
1	-	(1) Initiates counting (2) Resets output after next clock	-
2	(1) Disables counting (2) Sets output immediately High	(1) Reloads counter (2) Initiates counting	Enables counting
3	(1) Disables counting (2) Sets output immediately High	Initiates counting	Enables counting
4	Disables counting	-	Enables counting
5	-	Initiates counting	-

Figure 1. Gate Pin Operations

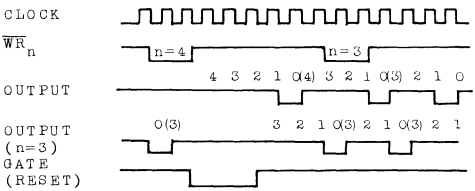
MODE 0: Interrupt on Terminal Count



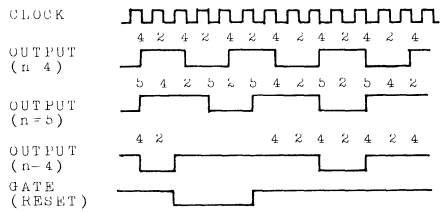
MODE 1: Programmable One-Short



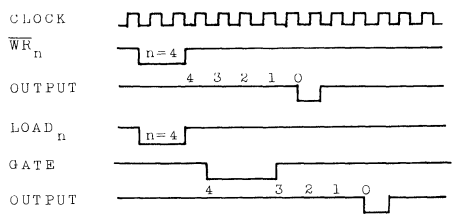
MODE 2: Rate Generator



MODE 3: Square Wave Generator



MODE 4: Software-Triggered Strobe



MODE 5: Hardware-Triggered Strobe

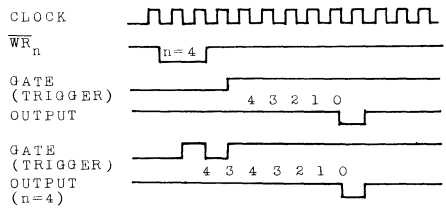
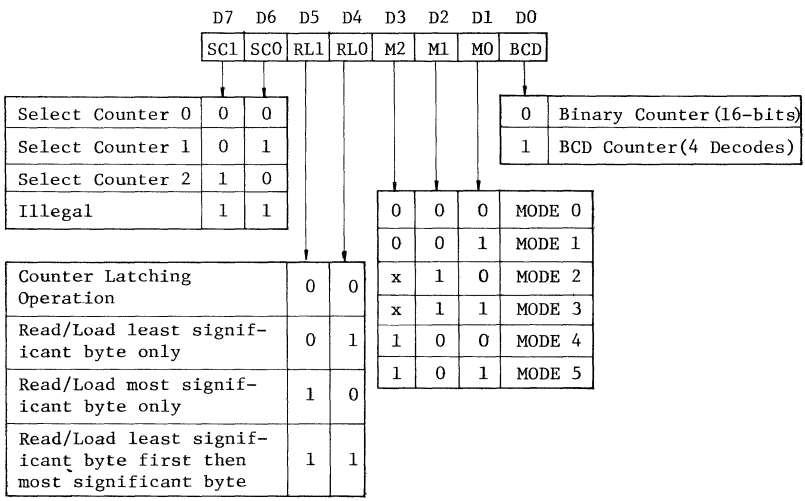


Figure 2. TMP8253-5 Timing Diagrams

PROGRAMMING the TMP8253-5

All of the MODEs for each counter are programmed by the systems software by simple I/O operations.

Each counter of the TMP8253-5 is individually programmed by writing a control word into the Control Word Register. ($\overline{CS}$=0, A0=A1=1, $\overline{WR}$=0)



NOTE. SC: Select Counter, RL: Read/Load, M: Mode,
BCD: Binary Coded Decimal.

Figure 3. Control Word Format

The programmer must write out to the TMP8253-5 a MODE Control Word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE Control Word can be in any sequence of counter selection.

The loading of the Count Register with actual count value, however, must be done in exactly the sequence programmed in the MODE Control Word (RL0, RL1)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock.

The count register must be loaded with the number of bytes programmed in the MODE Control Word. The one or two bytes to be loaded in the count regist.. do not have to follow the associated MODE Control Word. They can be programmed at any time following the MODE Control Word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Loading all zeros will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 and MODE 4, the new count will not restart until the load has been completed.

Read Operations

The TMP8253-5 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations. By controlling the A0, A1 inputs to the TMP8253-5, the programmer can select the counter to be read. The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input.

The contents of the counter selected must be read in the sequence programmed in the MODE Control Word (RL0, RL1). When RL0, RL1 is 11. First I/O Read contains the least significant byte (LSB), second I/O Read contains the most significant byte (MSB), and the two bytes must be read before any loading WR command can be sent to the same counter.

The second method allows the programmer to read the contents of any counter without effecting or disturbing the counting operation. When the programmer wishes to read the contents of a selected counter "On the fly", he loads the MODE register with a special code which latches the present

count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter. The contents of the latched register must be read in the sequence programmed in the MODE Control Word (RL0, RL1). This commands has no effect on the counters mode.

Program Example

```
Set up sequence for counter #0
    MVI  A, 0011000B ... #0, LSB→MSB, MODE 0, Binary
    OUT  CWAD        ... The address of Control Word Register
    MVI  A, 53H      ... LSB for counter #0
    OUT  CNTO        ... The address of counter #0
    MVI  A, 82H      ... MSB for counter #0
    OUT  CNTO        ... The address of counter #0
     ¦
READ the contents of counter #0
    MVI  A, 0000XXXXB
    OUT  CWAD        ... Latching count
    IN   CNTO        ... Read LSB of counter #0
    MOV  L, A
    IN   CNTO        ... Read MSB of counter #0
    MOV  H, A
     ¦
RELOAD to counter #0
    MVI  A, 82H      ...
    OUT  CNTO        ... Load LSB for counter #0
    MVI  A, 53H      ...
    OUT  CNTO        ... Load MSB for counter #0
```

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
V_{IN}	Input Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
V_{OUT}	Output Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
P_D	Power Dissipation	1W
Tsol	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 (Topr=0°C to 70°C, V_{CC}=5.0V±5%, GND=0V)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.2		V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL}=2.2 mA			0.45	V
V_{OH}	Output High Voltage	I_{OH}=-400 μA	2.4			V
I_{IL}	Input Leak Current	$0 \leq V_{IN} \leq V_{CC}$			±10	μA
I_{OFL}	Output Leak Current	$0 \leq V_{OUT} \leq V_{CC}$			±10	μA
I_{CC}	V_{CC} Supply Current				140	mA

INPUT CAPACITANCE (Ta=25°C, V_{CC}=GND=0V)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
C_{IN}	INPUT CAPACITANCE	f_C=1 MHz			10	pF
$C_{I/O}$	INPUT/OUTPUT CAPACITANCE	Unmeasured pins, 0V			20	pF

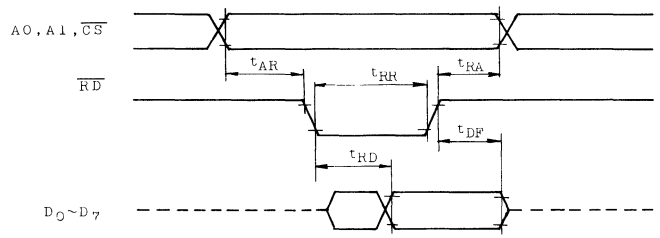
AC CHARACTERISTICS

(Topr=0°C to 70°C, V_{CC}=5.0V±5%, GND=0V)

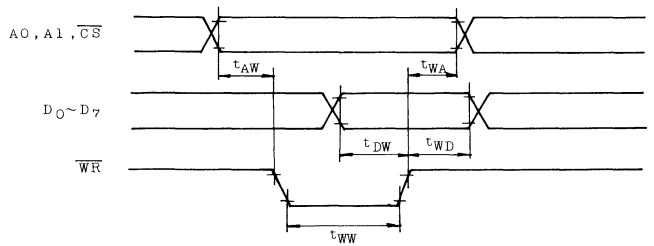
SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
t_{AR}	Address Set up Time ($\overline{RD}\downarrow$)		30			ns
t_{RA}	Address Hold Time ($\overline{RD}\uparrow$)		5			ns
t_{RR}	$\overline{RD}$ Pulse Width		300			ns
t_{RD}	Valid Data ($\overline{RD}\downarrow$)	C_L=150 pF			200	ns
t_{DF}	Data Floating ($\overline{RD}\uparrow$)		25		100	ns
t_{RV}	Recovery Time		1			μs
t_{AW}	Address Set up Time ($\overline{WR}\downarrow$)		30			ns
t_{WA}	Address Hold Time ($\overline{WR}\uparrow$)		30			ns
t_{WW}	$\overline{WR}$ Pulse Width		300			ns
t_{DW}	Data Set up Time ($\overline{WR}\uparrow$)		250			ns
t_{WD}	Data Hold Time ($\overline{WR}\uparrow$)		30			ns
t_{CLK}	Clock Period		380		DC	ns
t_{PWH}	CLK High Pulse Width		230			ns
t_{PWL}	CLK Low Pulse Width		150			ns
t_{GW}	GATE Width High		150			ns
t_{GL}	GATE Width Low		100			ns
t_{GS}	GATE Set up Time (CLK↑)		100			ns
t_{GH}	GATE Hold Time(CLK↑)		50			ns
t_{OD}	Output Delay From (CLK↓)	C_L=150 pF			400	ns
t_{ODG}	Output Delay From (GATE↓)	C_L=150 pF			300	ns

NOTE: AC timings measured at V_{OH}=2.2V, V_{OL}=0.8V

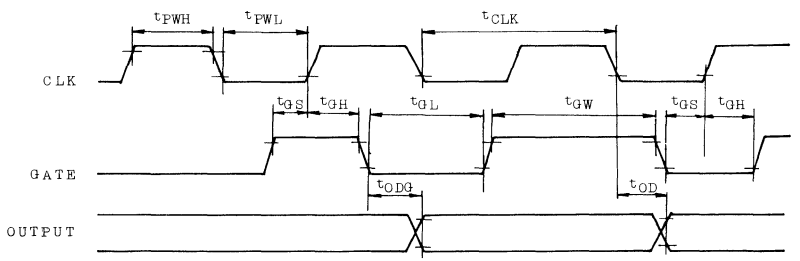
READ TIMING



WRITE TIMING



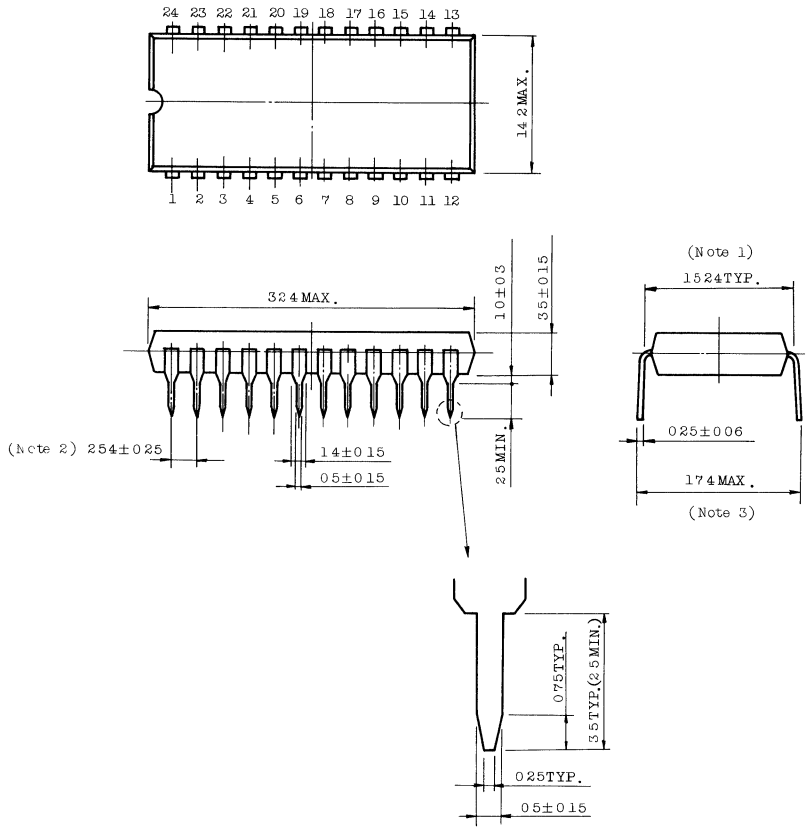
CLOCK & GATE TIMING



OUTLINE DRAWING

PLASTIC 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.24 leads.

3. This dimension is to outside of leads.

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP8255AP-5

TOSHIBA MOS TYPE DIGITAL IC

SILICON MONOLITHIC

PROGRAMMABLE PERIPHERAL INTERFACE (PPI)

GENERAL DESCRIPTION

TMP8255AP-5 is the high speed programmable peripheral interface LSI, capable of controlling parallel input/output data. This LSI is programmable in several operation modes and is capable of supplying a simple interface between micro-processors and peripherals equipment.

24 input/output pins are divided into three 8-bit ports and used either for input or output by programs.

All signal levels are TTL compatible.

Data transfer between the processor and TMP8255AP-5 is possible by using Chip Select Input ($\overline{CS}$) and Port Address A_0, A_1.

Data write/read operation to/from a specified port is possible by using Write Input ($\overline{WR}$) or Read Input ($\overline{RD}$).

FEATURES

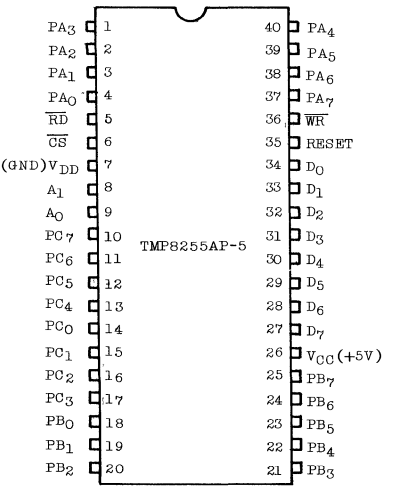
- Compatible with INTEL's 8255A-5
- 24 Programmable Input/Output Pins
- Programmable Operation Modes
- Direct Bit Set/Reset Capability
- Single +5V Power Supply

PIN NAMES

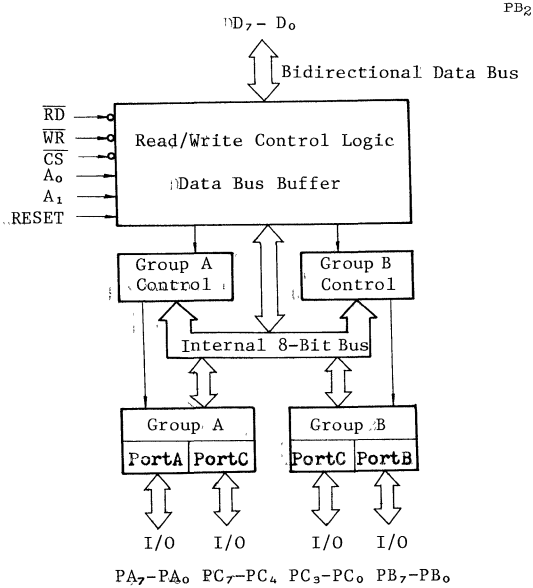
$PA_0 \sim PA_7$	Port A
$PB_0 \sim PB_7$	Port B
$PC_0 \sim PC_7$	Port C
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Signal
$\overline{WR}$	Write Signal
A_0 - A_1	Port Address
RESET	Reset
V_{CC} . V_{SS}	Power Supply

PIN CONNECTION

TOP VIEW



BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

I/O SIGNALS

TMP8255A-5 uses a 8-bit bidirectional data bus for data transfer to/from the processor. Data can be transfered between the data bus and control registers or between 2 output port groups (Group A and Group B), in the inside of TMP8255AP-5. There are 2 control registers.
Group A consists of high-order 4 bits of Port C and Port A. Group B consists of low-order 4 bits of Port C and Port B.

Port A 8 data bits are used for input latch/buffer, output latch/buffer, or bidirectional bus, respectively.

Port B 8 data bits are used for input buffer or output latch/buffer.

Port C 8 data bits are used for input buffer, output latch/buffer, or two 4-bit control ports in combination with Port A and B.

Operation of each port is controlled by programs.

When two inputs of Port Addresses A_0 and A_1 are used together with Read Input, Write Input and Chip Select, it is possible to select a specific port or control register.

Port A selection	$A_1 = 0$	$A_0 = 0$
Port B selection	$A_1 = 0$	$A_0 = 1$
Port C selection	$A_1 = 1$	$A_0 = 0$
Control register selection	$A_1 = 1$	$A_0 = 1$

(Note: Readout operation from a control register is impossible.)

Read ($\overline{RD}$)	Data read operation from TMP8255AP-5 to the data bus is controlled by $\overline{RD}$ signal (low active).
Write ($\overline{WR}$)	Data write operation from the data bus to TMP8255AP-5 is controlled by $\overline{WR}$ signal (low active).
Chip Select ($\overline{CS}$)	TMP8255AP-5 is selected by $\overline{CS}$ signal (low active). When $\overline{CS}$="1", the data bus driver is in the high impedance state.
Reset (RESET)	When RESET="1", all internal registers are cleared and all ports are in high impedance input mode.
Data Bus ($D_7 - D_0$)	The 8-bit data bus is used for transferring data and program information between the processor and TMP8255AP-5.

PROGRAMMING

To program the operations of TMP8255AP-5, first select the internal control function to be programmed by the processor. To do this, execute the write operation under the programming mode ($A_1 = A_0 = \overline{RD}$ = "1", $\overline{CS} = \overline{WR}$ = "0"). As a result, data bus information is written into one of two control registers. Input/Output of respective ports and operation mode of each group (Mode 0, 1 and 2 are available) can be selected by one of these control registers. Another control register is used for controlling set/reset of Port C bits. One of these two control registers is selected by Bit 7 of the data bus. When Bit 7 is "1", an operation mode is selected, while the set/reset function is selected when it is "0".

Bit 0 through 6 have different meanings depending upon a selected control mode.

(a) Operation Mode Control (DB7 = "1")

1	Control of Group A				Control of Group B		
DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0

- Control of Group A (DB_6 - DB_3)

  The operation mode is defined by Bit 6 and 5, and the port function (input or output) is selected by Bit 4 and 3.

- Control of Group B (Bit 2, 1 and 0)

  The operation mode is defined by Bit 2, and the port function (input or output) is selected by Bit 1 and 0.

The detail of operation modes and port input/output selection are described in the next item.

Relationship between operation modes and control bit are shown in the following table.

Operation Mode	Control Bit		
	Group A		Group B
	6	5	2
0	0	0	0
1	0	1	1
2	1	X	

Bit 6 and 5 define Group A modes and Bit 2 defines Group B modes.

Note: X mark shows that either 0 or 1 is acceptable.

(b) Bit Set/Reset Control (DB_7 = "0")

0	X	X	X	Bit Select			I/O
DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0

Bit Select

Bit 3, 2 and 1 select Port C bit to be set/reset.

Port C Bit Selection	Control Bit		
	3	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Set/Reset

When Bit 0 is "1", selected bit of Port C is set and when it is "0", selected bit of Port C is reset. In this case, Bit 6, 5 and 4 are not used. Therefore, either "0" or "1" is acceptable.

OPERATION MODES

TMP8255AP-5 is designed for various programs which control for interfacing with various peripherals.
For this purpose, there are 3 basic modes.

Mode 0

24 input/output pins of 3 ports are devided into 4 groups of Port A (8 bits), Port B (8 bits), Port C (high order 4 bits) and Port C (low order 4 bits).

Data is latched in the output port, but is not latched in the input port. In the data input operation, input data is placed on the data bus at $\overline{RD}$ = 0. The combination of input/output of these 4 groups is available in 16 ways. This combination is selected by the operation mode register of TMP8255AP-5 using Bit 0, 1, 3 and 4 of the data bus.

Programming under Mode 0

In the case $\overline{CS}$ = $\overline{WR}$ = "0", A_1 = A_0 = $\overline{RD}$ = "1", DB_7 = "1" and DB_6 = DB_5 = DB_2 = "0", the programming is shown in the following table.

Data Bus Bit				Port A	Port C (PC_7-PC_4)	Port B	Port C (PC_3-PC_0)
4	3	1	0				
0	0	0	0	Out	Out	Out	Out
0	0	0	1	Out	Out	Out	In
0	0	1	0	Out	Out	In	Out
0	0	1	1	Out	Out	In	In
0	1	0	0	Out	In	Out	Out
0	1	0	1	Out	In	Out	In
0	1	1	0	Out	In	In	Out
0	1	1	1	Out	In	In	In
1	0	0	0	In	Out	Out	Out
1	0	0	1	In	Out	Out	In
1	0	1	0	In	Out	In	Out
1	0	1	1	In	Out	In	In
1	1	0	0	In	In	Out	Out
1	1	0	1	In	In	Out	In
1	1	1	0	In	In	In	Out
1	1	1	1	In	In	In	In

Mode 1

Under Mode 1, Port A and high order 5 bits of Port C are correlated to Group A, and Port B and low order 3 bits of Port B are correlated to Group B, respectively.

Port C is used for a control signal to control input/output data of Port A or Port B.

The internal enable/disable flip-flop (INTE) can be controlled by setting/resetting Bit 4 and 2 of Port C when Ports A and B are used as the input ports using the bit set/reset function, and by setting/resetting Bit 6 and 2 when they are used as the output ports.

When bit set/reset is "1", the flip-flop is placed in the enable state, and when bit set/reset is "0", it is placed in the disable state.

Data transfer between the ports and peripherals is controlled by 3 control signals for input operation, and is also controlled by 3 control signals for output operation. Functions of individual bits of Port C are specified as shown in the following table.

	Control Functions	Related Port	
		A	B
Input	$\overline{STB}$	PC_4	PC_2
	IBF	PC_5	PC_1
	INTR	PC_3	PC_0
Output	$\overline{OBF}$	PC_7	PC_1
	$\overline{ACK}$	PC_6	PC_2
	INTR	PC_3	PC_0

Out of above stated control functions, those related to input are as follows.

- Strobe Input ($\overline{STB}$): When $\overline{STB}$ = "0", data is loaded into the input latch

- Input Buffer Full (IBF): This signal shows that data has been already loaded. IBF is set by $\overline{STB}$ = "0" and is reset at the rising edge of $\overline{RD}$.

- Interrupt Request (INTR): If INTE flag is in the enable state and IBF = "1", INTR signal becomes "1" at $\overline{STB}$ = "1".

  INTR signal can be directly connected to INT input of the processor, and when data is loaded on a port, an interrupt signal is generated. INTR is reset when $\overline{RD}$ signal from the processor is received into the port.

On the other hand, control functions related to output are as follows.

- Output Buffer Full (OBF): This is a flag showing that the processor has loaded data on a specific port ($\overline{OBF}$ = 0). $\overline{OBF}$ becomes "0" at the rising edge of $\overline{WR}$ signal and becomes "1" at the falling edge of $\overline{ACK}$ signal from peripherals.

- Acknowledge ($\overline{ACK}$): When data has been received from a TMP8255AP-5 port, a peripheral responds to TMP8255AP-5 by transmitting an acknowledge signal $\overline{ACK}$ (low active).

- Interrupt Request (INTR): This output can be used to interrupt the processor when a peripheral has accepted data transmitted by the processor. If INTE flag is in the enable state and $\overline{OBF}$ = "1", INTR signal is set by $\overline{ACK}$ = "1" and is reset by $\overline{WR}$ = "0".

Mode 2

Under mode 2, Port A is used as a bidirectional bus. Both input and output of Port A are latched under this mode.

5 bits of Port C are used for control between peripherals and TMP8255AP-5. Signals used for this control are as follows:

$\overline{STB}$, IBF, $\overline{OBF}$ and INTR: The functions of these signals are identical in Mode 1.

$\overline{ACK}$: When this signal becomes active (low), 3-state output buffer of Port A is enabled to transfer data to peripheral equipment. During other periods, the output buffer is in high impedance.

For the selection enable/disable of INTE flip-flop, Bit 6 is used for output operation, and Bit 4 is used for input operation. Data transfer between the ports and peripherals is executed by designating pins of Port C same as in operations under Mode 1.

Under Mode 1 and 2, Port C status and control bits can be tested when Port C contents are read out. All bits of Port C are not used for control and status functions, unspecified bits can be programmed for input or output as described below.

In the case Port C has been programmed as output, Pins (PC_7 - PC_4) of Group A operate Port C by using the bit set/reset function. Pins (PC_3 - PC_0) of Group B controls write operation into Port C or read operation by using the bit set/reset function.

ABSOLUTE MAXIMUM RATING

SYMBOL	PARAMETER	RATING
T_{stg}	Storage Temperature	-65°C to 150°C
T_{opr}	Operating Temperature	0°C to 70°C
V_{CC}	Supply Voltage	-0.5V to 7.0V
V_{OUT}	Output Voltage	-0.5V to 7.0V
V_{IN}	Input Voltage	-0.5V to 7.0V
P_W	Power Dissipation	1.0W

D.C CHARACTERISTICS (Ta = 0°C to 70°C, V_{CC}=5V±5%, V_{SS}=0V)

SYMBOL	PARAMETER		TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IL}	Input Low Voltage			-0.5		0.8	V
V_{IH}	Input High Voltage			2.0		V_{CC}	V
V_{OL}	Output Low Voltage	(DB)	I_{OL} = 2.5 mA			0.45	V
		(PER)	I_{OL} = 1.7 mA			0.45	V
V_{OH}	Output High Voltage	(DB)	I_{OH} = -400 μA	2.4			V
		(PER)	I_{OH} = -200 μA	2.4			V
I_{DAR} (Note 1)	Darlington Drive Current		V_{EXT} = 1.5V, R_{EXT} = 750Ω	-1.0		-4.0	mA
I_{IL}	Input Leak Current		0V ≤ V_{IN} ≤ V_{CC}			±10	μA
I_{OFL}	Output Leak Current (High Impedance State)		0V ≤ V_{OUT} ≤ V_{CC}			±10	μA
I_{CC}	Supply Current					120	mA

CAPACITANCE (Ta=25°C, V_{CC}=V_{SS}=0V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
C_{IN}	Input Capacity	fc = 1 MHz			10	pF
$C_{I/O}$	I/O Capacity				20	pF

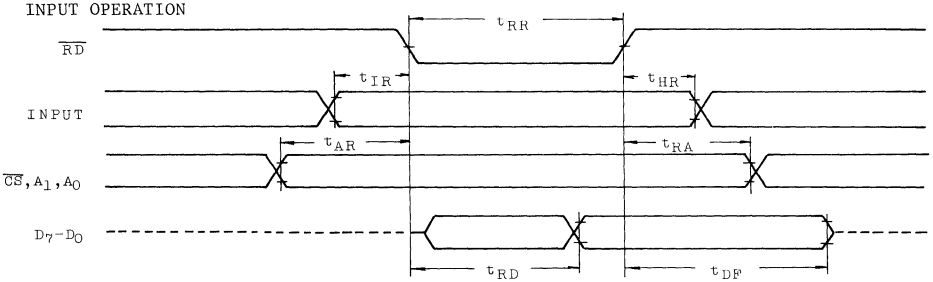
A.C. CHARACTERISTICS ($Ta=0°C$ to $70°C$, $V_{CC}=5V\pm5\%$, $V_{SS}=0V$)

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
t_{AR}	Address Stable before $\overline{RD}$	0			ns
t_{RA}	Address Stable after $\overline{RD}$	0			ns
t_{RR}	$\overline{RD}$ Pulse Width	300			ns
t_{RD}	Data Valid from $\overline{RD}$ (Note 2)			200	ns
t_{DF}	Data Float after $\overline{RD}$	10		100	ns
t_{RV}	Time between READs and/or WRITEs	850			ns
t_{AW}	Address Stable before $\overline{WR}$	0			ns
t_{WA}	Address Stable after $\overline{WR}$	20			ns
t_{WW}	$\overline{WR}$ Pulse Width	300			ns
t_{DW}	Data Valid to $\overline{WR}$	100			ns
t_{WD}	Data Valid after $\overline{WR}$	30			ns
t_{WB}	$\overline{WR}=1$ to Output Delay (Note 2)			350	ns
t_{IR}	Peripheral Data before $\overline{RD}$	0			ns
t_{HR}	Peripheral Data after $\overline{RD}$	0			ns
t_{AK}	$\overline{ACK}$ Pulse Width	300			ns
t_{ST}	$\overline{STB}$ Pulse Width	500			ns
t_{PS}	Peripheral Data before Rising Edge of $\overline{STB}$	0			ns
t_{PH}	Peripheral Data after Rising Edge of $\overline{STB}$	180			ns
t_{AD}	$\overline{ACK}=0$ to Output (Note 2)			300	ns
t_{KD}	$\overline{ACK}=1$ to Output Float	20		250	ns
t_{WOB}	$\overline{WR}=1$ to $\overline{OBF}=0$ (Note 2)			650	ns
t_{AOB}	$\overline{ACK}=0$ to $\overline{OBF}=1$ (Note 2)			350	ns
t_{SIB}	$\overline{STB}=0$ to IBF = 1 (Note 2)			300	ns
t_{RIB}	$\overline{RD}=1$ to IBF = 0 (Note 2)			300	ns
t_{RIT}	$\overline{RD}=0$ to INTR = 0 (Note 2)			400	ns
t_{SIT}	$\overline{STB}=1$ to INTR = 1 (Note 2)			300	ns
t_{AIT}	$\overline{ACK}=1$ to INTR = 1 (Note 2)			350	ns
t_{WIT}	$\overline{WR}=0$ to INTR = 0 (Note 2)			850	ns

TIMING WAVEFORMS

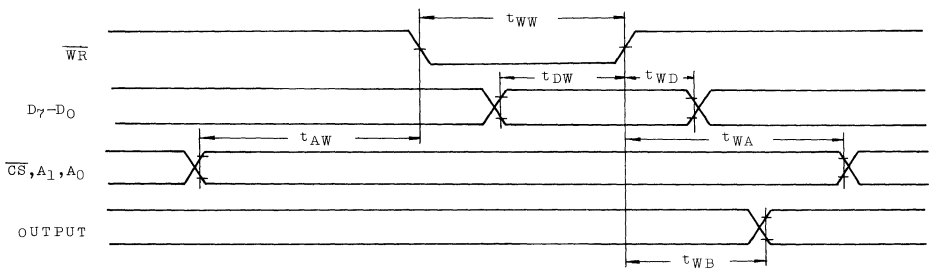
MODE 0

INPUT OPERATION



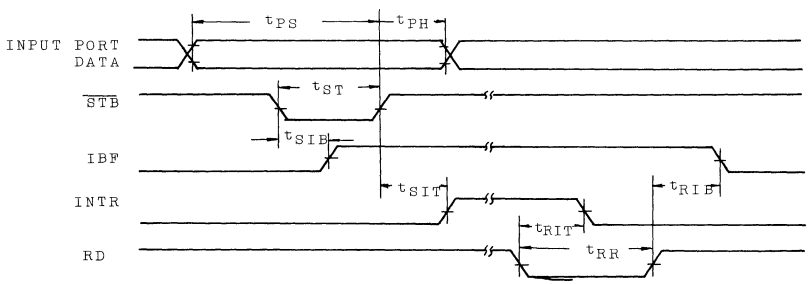
MODE 0

OUTPUT OPERATION



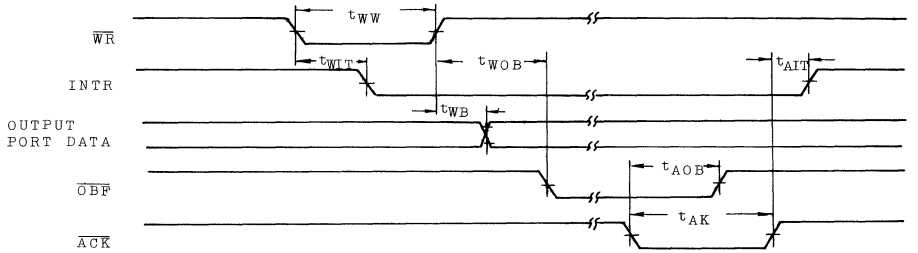
MODE 1

INPUT OPERATION



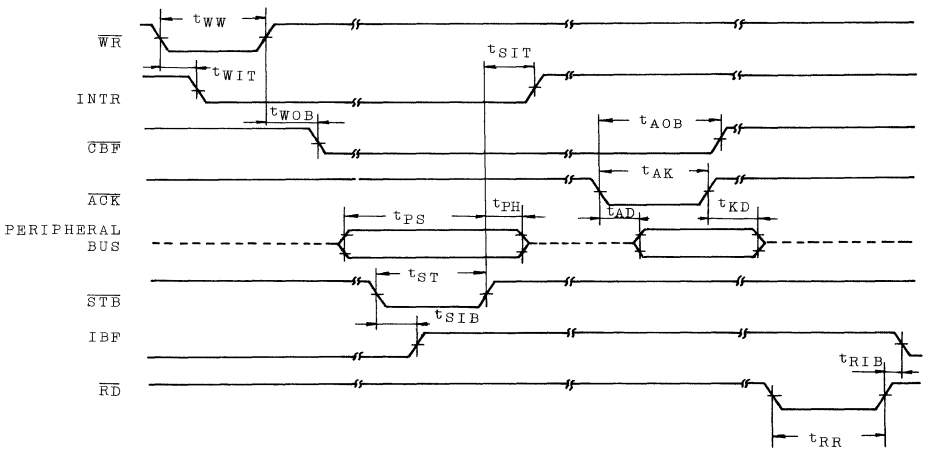
MODE 1

OUTPUT OPERATION



MODE 2

BIDIRECTIONAL OPERATION



Note 1. Available on any 8 pins of Port B and Port C.

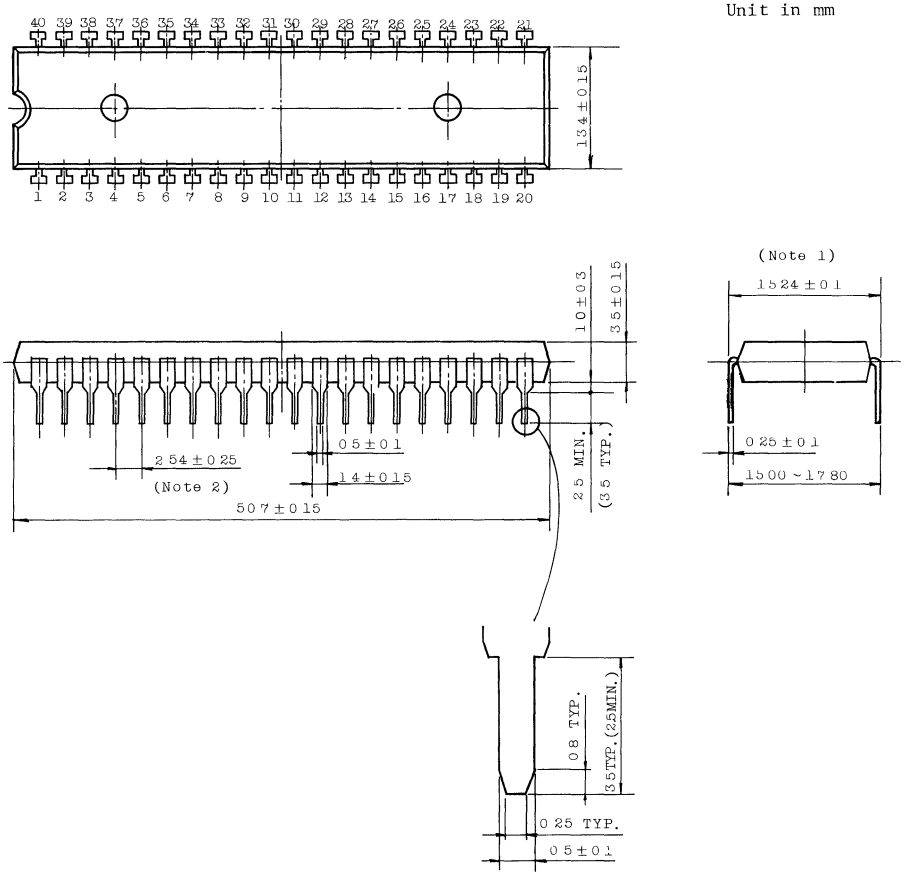
2. Test conditions; $C_L = 150pF$

3. Period of Reset pulse should be at least 50μs during or after power on. Subsequent Reset pulse can be 500ns min.

4. Timing measuring levels are as follows: high level = 2V
   low level = 0.8V

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.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DEGITAL
INTEGRATED CIRCUIT

TMP8259AP

Silicon Monolithic
N Channel Silicon Gate MOS

PROGRAMMABLE INTERRUPT CONTROLLER

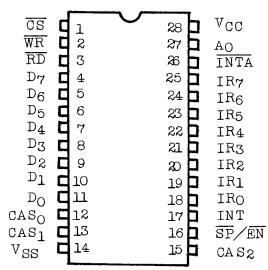
GENERAL DESCRIPTION

The TMP8259AP is a programmable interrupt controller designed for use with the TLCS-85A microcomputer system. It handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry.

FEATURES

- o Eight Level Priority Controller.
- o Expandable to 64 Level.
- o Interrupt Modes, Interrupt Mask, Vectored Address Programmable.
- o Single +5V Power Supply.
- o 8085A, 8086 Microcomputer System Compatible.
- o Compatible with Intel's 8259A.

PIN CONNECTIONS (TOP VIEW)

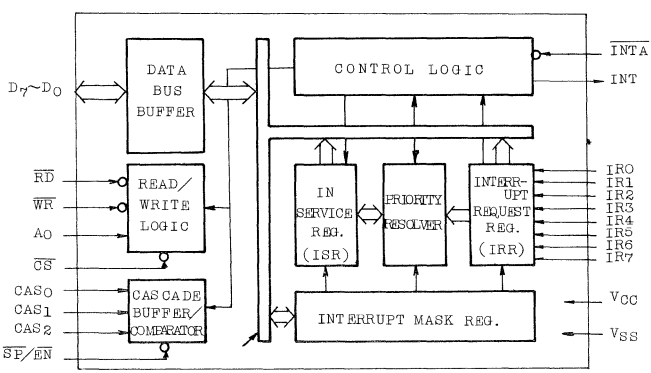


PIN NAMES AND PIN DESCRIPTION

Pin Name	Input/Output	Function
$\overline{CS}$	Input	Chip Select Input. A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. $\overline{INTA}$ functions are independent of $\overline{CS}$.
$\overline{WR}$	Input	Write Control Input. A low on this pin when $\overline{CS}$ is low enables the 8259A to accept command words from CPU.
$\overline{RD}$	Input	Read Control Input. A low on this pin when $\overline{CS}$ is low enables the 8259A to release status onto the data bus for the CPU.
$D_0 \sim D_7$	Input/Output	Biderectional Data Bus. Command status and interrupt-vector information is transfered via this bus.
$CAS_0 \sim CAS_2$	Input/Output	Cascade Lines. The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.
$\overline{SP}/\overline{EN}$	Input/Output	Slave Program/Enable Buffer. This is a dual function pin. When in the buffered mode it can be used as on Output to control buffer transceivers ($\overline{EN}$). When not in the buffered mode it is used as an input to designate a master 8259A ($\overline{SP}$=1) or a slave ($\overline{sp}$=0).
INT	Output	Interrupt Request Output. This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU. Thus it is connected to CPU's interrupt pin.

Pin Name	Input/Output	Function
IR_0 - IR7	Input	Interrupt Request Inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
$\overline{INTA}$	Input	Interrupt Acknowledge Input. This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	Input	A_0 Address Line. This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various command words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A_0 address line.
VCC		+5V Power Supply
VSS		Ground

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7V
V_{IN}	Input Voltage	-0.5V to +7V
P_D	Power Dissipation	1W
T_{sol}	Soldering Temperature	260°C
T_{stg}	Storage Temperature	-65°C to 150°C
T_{opr}	Operating Temperature	0°C to 70°C

DC CHARACTERISTICS T_{opr}=0°C to 70°C, V_{CC}=+5V±10%, V_{SS}=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		2.0	-	V_{CC}+0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.2mA	-	-	0.45	V
V_{OH}	Output High Voltage	I_{OH} = -400μA	2.4	-	-	V
$V_{OH(INT)}$	Output High Voltage (INT)	I_{OH} = -100μA	3.5	-	-	V
		I_{OH} = -400μA	2.4	-	-	V
I_{LI}	Input Leak Current	0V ≤ V_{IN} ≤ V_{CC}	-	-	±10	μA
I_{LOL}	Output Leak Current	0.45V < V_{IN} < V_{CC}	-	-	±10	μA
I_{CC}	V_{CC} Supply Current		-	-	85	mA
I_{LIR}	Input Current (IR)	V_{IN} = 0V	-	-	-300	μA
		V_{IN} = V_{CC}	-	-	10	μA

AC CHARACTERISTICS T_{opr}=0°C ∿ 70°C, V_{CC}=5V±10%, V_{SS}=0V, Unless otherwise noted.

TIMING REQUIREMENTS

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
TAHRL	$A_0/\overline{CS}$ Setup Time ($\overline{RD}/\overline{INTA}\downarrow$)	0	–	ns
TRHAX	$A_0/\overline{CS}$ Hold Time ($\overline{RD}/\overline{INTA}\uparrow$)	0	–	ns
TRLRH	$\overline{RD}$ Pulse Width	235	–	ns
TAHWL	$A_0/\overline{CS}$ Setup Time ($\overline{WR}\downarrow$)	0	–	ns
TWHAX	$A_0/\overline{CS}$ Hold Time ($\overline{WR}\uparrow$)	0	–	ns
TWLWH	$\overline{WR}$ Pulse Width	290	–	ns
TDVWH	$D_0 - D_7$ Setup Time ($\overline{WR}\uparrow$)	240	–	ns
TWHDX	$D_0 - D_7$ Hold Time ($\overline{WR}\uparrow$)	0	–	ns
TJLJH	Interrupt Request Pulse Width (LOW)	100	–	ns
TCVIAL	Cascade Setup Time (Second or Third $\overline{INTA}\downarrow$)	55	–	ns
TRHRL	$\overline{RD}\uparrow$ to Next Command	160	–	ns
TWHRL	$\overline{WR}\uparrow$ to Next Command	190	–	ns

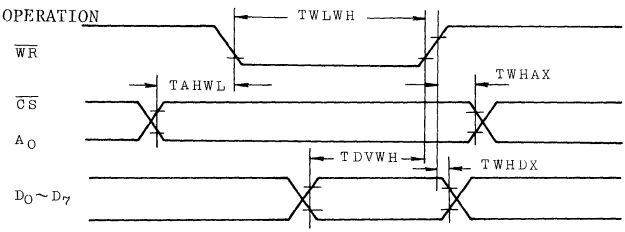
RESPONSE CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
TRLDV	Valid Data Delay ($\overline{RD}/\overline{INTA}\downarrow$)	$D_0 - D_7$, C_L = 100pF	–	–	200	ns
TRHDZ	Data Floating ($\overline{RD}/\overline{INTA}\uparrow$)		–	–	100	ns
TJHIH	Interrupt Output Delay (IR↑)	INT, C_L = 100pF	–	–	350	ns
TIALCV	Valid Cascade Delay ($\overline{INTA}\downarrow$)		–	–	565	ns
TRLEL	Enable Active ($\overline{RD}/\overline{INTA}\downarrow$)		–	–	125	ns
TRHEH	Enable Inactive ($\overline{RD}/\overline{INTA}\uparrow$)	$CAS_0 - CAS_2$, C_L = 100pF	–	–	150	ns
TAHDV	Valid Data Delay ($A_0/\overline{CS}$)		–	–	200	ns
TCVDV	Valid Data Delay (CAS_0-CAS_2)		–	–	300	ns

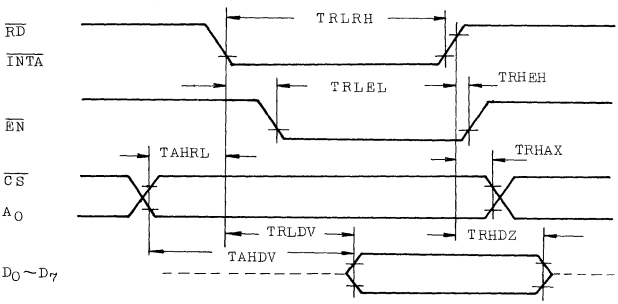
NOTE: AC TESTING. Inputs are driven at V_L = 0.45V and V_H = 2.4V.
Measurements are made at V_L = 0.8V and V_H = 2.0V.

TIMING WAVEFORMS

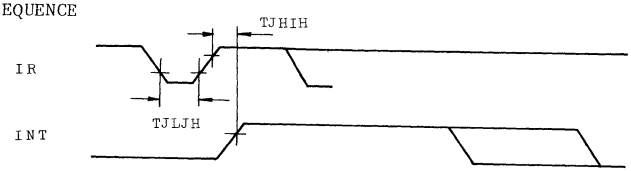
WRITE OPERATION

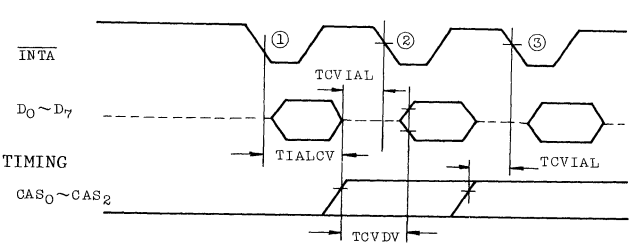


READ AND INTA OPERATION

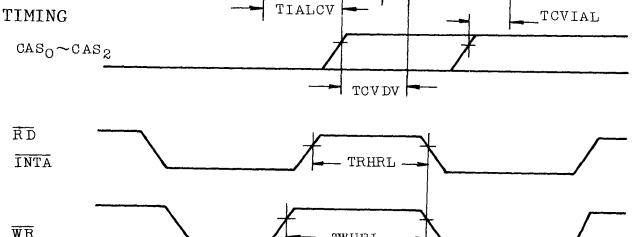


INTA SEQUENCE



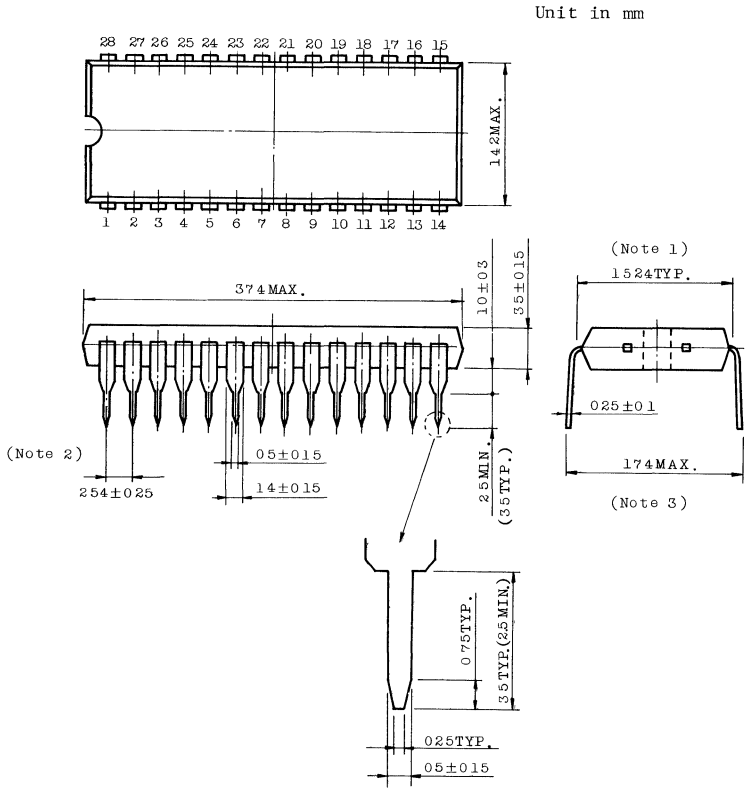


OTHER TIMING



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.28 leads.

3. This dimension is to outside of ledas.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8279P-5

N-CHANNEL SILICON GATE MOS

PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

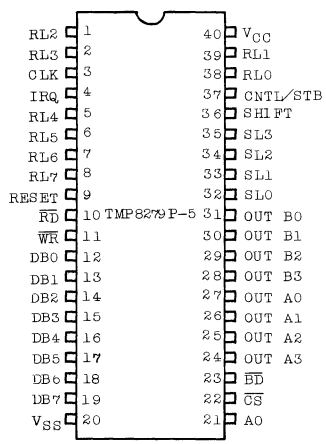
GENERAL DESCRIPTION

The TMP8279-5 is a progammable keyboard/display interface chip designed for use as the TLCS-85A microcomputer peripheral. The keyboard portion can provide a scanned interface to a 64-contact key matrix. Also, the keyboard portion will interface to an array of sensors or a strobed interface keyboard. Key depressions can be 2-key lockout or N-key rollover. The display portion has 16 × 8 bits display RAM which can be organized into dual 16 × 4 bits. Both right entry and left entry display formats are possible.

FEATURES

- Sumulataneous Keyboard Display operation is possible.
- Scanned Keyboard mode.
- Scanned Sensor Matrix mode.
- Strobed Input Entry mode.
- 8-Character FIFO is built in.
- 2 Key Lockout or N-key Rollover with contact De-bounce is programmable.
- 16 × 8 bit Display RAM is built in.
- Scan timing is programmable.
- Compatible with INTEL 8279-5.

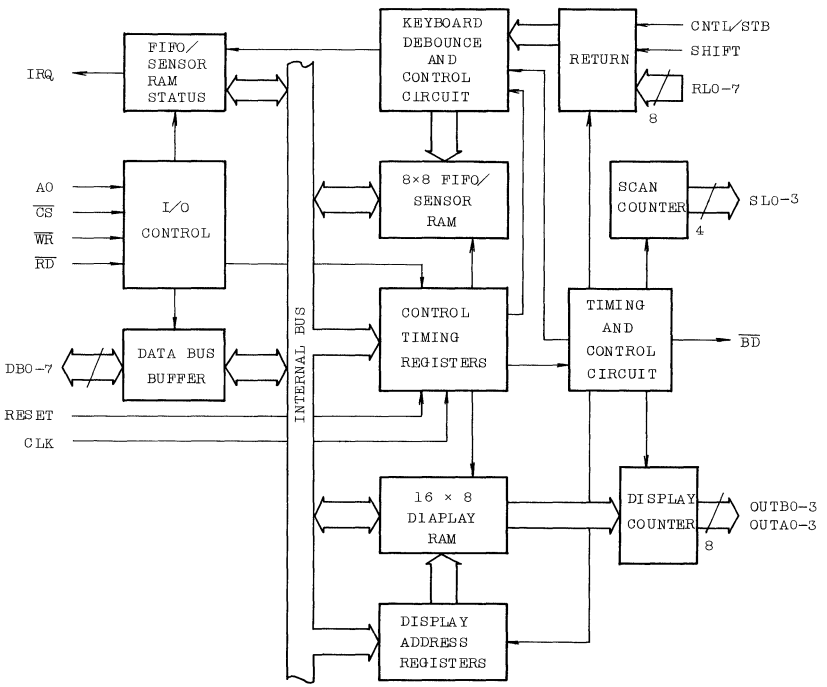
PIN CONNECTION



PIN NAME

DB0 - DB7	8-bit Bidirectional data bus
CLK	Clock input
RESET	Reset input
$\overline{CS}$	Chip select input
$\overline{RD}$	Read input
$\overline{WR}$	Write input
A0	Command/data control input
IRQ	Interrupt request output
SL0 - SL3	Scan lines
RL0 - RL7	Return lines
SHIFT	Shift input
CNTL/STB	Control/strobe input
OUTA0 - A3	Display (A) outputs
OUTB0 - B3	Display (B) outputs
$\overline{BD}$	Blanking display output
V_{CC}	+5V
V_{SS}	Ground

BLOCK DIAGRAM



PIN NAMES AND PIN DESCRIPTION

V_{SS} (Power Supply)

Ground

V_{CC} (Power Supply)

+5V during operation

DB_0 - DB_7 (Input/Output)

Bidirectional Data Bus. All data and commands are transfered via this data Bus.

CLK (Input)

CLOCK from system used to generate internal timing.

RESET (Input)

A high signal on this pin resets the TMP8279 . After being reset the TMP8279 is placed in the following state.

(1) 16 × 8 bit character display, left entry.
(2) Encode scan keyboard, 2 key lockout, clock pre-scale value is set to 31H.

$\overline{CS}$ (Input)

A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the TMP8279-5.

A0 (Input)

This inputs acts in conjunction with the $\overline{CS}$, $\overline{WR}$ and $\overline{RD}$ pins. A high on this pin indicates the signals on data bus are interpreted as command or status. A low indicates they are Data.

$\overline{WR}$ (Input)

A low on this pin when $\overline{CS}$ is low enables the TMP8279 to accept command or data from the CPU.

$\overline{RD}$ (Input)

A low on this pin when $\overline{CS}$ is low enables the TMP8279 to send data to data Bus.

IRQ (Output)

Interrupt request output. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.

SL_0 - SL_3 (Output)

Scan lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).

RL_0 - RL_7 (Input)

Return lines which are connected to the scan lines through the keys or sensor switches. They have internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.

SHIFT (Input)

This input status is stored along with the key position on key closure in the Scanned key board modes. It has a internal pullup to keep it high until a switch closure pulls it low.

CNTL/STB (Input)

For Keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into FIFO in the Strobed Input mode (Rising Edge). It has an internal pullup to keep it high until a switch closure pulls it low.

$OUTA_0$ - $OUTA_3$ (Output)

$OUTB_0$ - $OUTB_3$ (Output)

These two ports are the outputs for the 16 ×4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0 - SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8-bit port.

$\overline{BD}$ (Output)

This output is used to blank the display during digit switching or by a display blanking command.

FUNCTIONAL DESCRIPTION
[BLOCK DESCRIPTION]

I/O Control and Data Bus Buffer

The I/O control section uses the $\overline{CS}$, A0, $\overline{RD}$ and $\overline{WR}$ lines and controls the flow of data to and from various internal registers and buffers. $\overline{CS}$ input enables the all data flow to and from the TMP8279. The character of the information given by the CPU, is identified by A_0. $\overline{RD}$ and $\overline{WR}$ decide the direction of data flow through the data bus buffer. The data bus buffer is bidirectional buffer which is used for connecting the internal bus and a system bus. When $\overline{CS}$ is high, the buffer is in a high impedance state.

$\overline{CS}$	A_0	$\overline{RD}$	$\overline{WR}$	Functions
0	0	0	1	Read Data
0	0	1	0	Write Data
0	1	0	1	Read Status word
0	1	1	0	Write Command word
1	X	X	X	High-impedance state

Control Register, Timing Register and Timing Control Circuit

The keyboard and display modes or other operating conditions are programmed by the CPU. These modes are latched at the rising edge of $\overline{WR}$ when A_0 is high. The timing control contains the basic counter chains. The first counter is the 1/N prescaler that can be programmed to yield an basic internal frequency which gives a 5.1 mS keyboard scan time and a 10.3 mS debounce time. The other counters divide down the basic internal frequency to provide the proper keyboard matrix scan and display scan times.

Scan Counter

Two modes are available for the scan counter. In the encode mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the key board and display. In the decode mode, the scan counter decodes the least significant 2 bits internally and provides a decoded 1 of 4 scan. It is necessary to pay attention on the fact that only first 4 characters in the Display RAM are displayed.

Return Buffer and Keyboard Devounce Control circuit

The 8 return lines are latched by the return buffer. In the Keyboard mode, these lines are scanned to look for key closures in a row. If the debounce circuit detects a closed switch, it waits about 10 mS, and checks if the switch remains closed. If it does so, address of the switch in the matrix is transferred to the FIFO along with the status of SHIFT and CNTL lines.

FIFO/Sensor RAM and FIFO/Senser RAM Status

The FIFO/Sensor RAM is a dual function RAM. In the keyboard mode or In the Strobe Input mode, this FIFO/Sensor RAM serves as a FIFO. The FIFO status shows whether the FIFO is empty or full and keeps track of the number of characters in the FIFO. In addition, there is a flag to show an error in the case where too many reads or writes is recognized. The FIFO status can be read at $\overline{CS} = \overline{RD} = 0$, $A_0 = 1$. The FIFO status logic provides an IRQ signal when the FIFO is not empty. In the scanned sensor matrix mode, the memory serves as a sensor RAM. IRQ becomes high when a change in the sensor is detected.

Display Address Registers and Display RAM

The display address registers hold the address of the word currently being written or read by the CPU and the two 4 bit nibbles being displayed. The Display RAM stores data for display outputs. The read/write addresses are programmed by the CPU command. They also can be programmed to auto-increment after read or write. The Display RAM can be directry read out by the CPU after mode and address is set. The A and B nibbles of the Display RAM are outputted to the Display Outputs A and B syncronously with scan signals (SL_0 - SL_3). The A and B nibbles can be entered independently or as one word by the CPU command.

[COMMAND DESCRIPTION]

Keyboard/Display Mode Set

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	D	D	K	K	K

Display Mode

0	0	8 × 8 bit character display - Left entry
* 0	1	16 × 8 bit character display - Left entry
1	0	8 × 8 bit character display - Right entry
1	1	16 × 8 bit character display - Right entry

* Default after reset.

Keyboard Mode

* 0	0	0	Encoded Scan Keyboard - 2-Key Lockout
0	0	1	Decoded Scan Keyboard - 2-Key Lockout
0	1	0	Encoded Scan Keyboard - N-Key Rollover
0	1	1	Decoded Scan Keyboard - N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

Program Clock

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	P	P	P	P	P

The TMP8279 generates all timing and multiplexing signals by means of the internal prescaler. The prescaler generates internal reference clocks by dividing an external clock by a programmable value PPPPP. Any number of ranging from 2 to 31 can be set as a prescaler value. When this value is set to 0 or 1, it is interpreted to be 2. If the internal reference clock is set to 100kHz, it is possible to obtain 5.1mS keyboard scan time and 10.3mS debounce time. The value PPPPP is set to 31 after reset, but cannot be changed by the Clear command.

Read FIFO/Sensor RAM

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	AI	X	A	A	A

X=don't care

If this command is written, the subsequent data reads are set up for the FIFO/Senser RAM. Auto-increment flag (AI) and the RAM address bits AAA are valid only in Senser Matrix Mode. The address bits AAA select one of the 8 rows of the Sensor RAM. If AI = 1, the RAM address is incremented after each successive read. The Auto-incremented flag does not affect the auto-increment of the Display RAM.

Read Display RAM

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	1	AI	A	A	A	A

If this command is written, the subsequent data reads are set up for the Display RAM. The address bits AAAA select one of the 16 rows of the Display RAM. If AI = 1, the address is incremented after each read or write to the Display RAM. This command sets the next read or write address and the sense of the Auto-increment.

Write Display RAM

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	AI	A	A	A	A

If this command is written, the subsequent data writes are set up for the Display RAM. Note that writing this command does not switch the source of the subsequent data reads. The address register of the Display RAM is same for read/write operations. The addressing and Auto-increment function are identical to those for the Read Display RAM.

Display Write Inhibit/Blanking

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	1	X	IWA	IWB	BLA	BLB	X=don't care

The IWA or IWB bit can be used to mask A nibble or B nibble for entering the Display data independently. The BLA or BLB flag is available for the nibble A or B to blank the display. In the case where the Display Outputs are used as separate 4-bit display ports, the IWA or IWB bit is useful so as not to affect the other display port when the CPU writes a word to the Display RAM. The BLA or BLB bit is used for blanking the display independently without giving any affect to the other 4-bit display port. The blank code is determined by the last Clear command that has been programmed after reset. If the Display Output is used as an 8-bit port, it is necessary to set both BLA and BLB bits for blanking the display. Then $\overline{BD}$ signal becomes low.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
↕	↕	↕	↕	↕	↕	↕	↕
A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0

: Correspondence between Display Output and Data Bus

Clear

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	0	C_D	C_D	C_D	C_F	C_A

The C_D bits are used to clear all rows of the Display RAM to the following code shown below.

(D_4)	(D_3)	(D_2)	
C_D	C_D	C_D	
1	0	X	--- All Zeros (X = Don't Care)
1	1	0	--- All Hex 20H (0010 0000)
1	1	1	--- All Ones
0	X	X	--- not clear display if C_A = 0

↑ Enable clear display when C_D = 1 (or by C_A = 1)

While the Display RAM is being cleared, it may not write to the Display RAM. The MSB bit of the FIFO status word is set during this time. If the C_F bit is set to "1", the FIFO status is cleared and the interrupt request output (IRQ) is reset. Also, the Senser RAM pointer is set to the row 0.

The C_A bit has the combined effect of the C_D bit and C_F bit. It enables clear display code to the Display RAM and also clears the FIFO status. Furthermore, it re-synchronizes the internal timing chains.

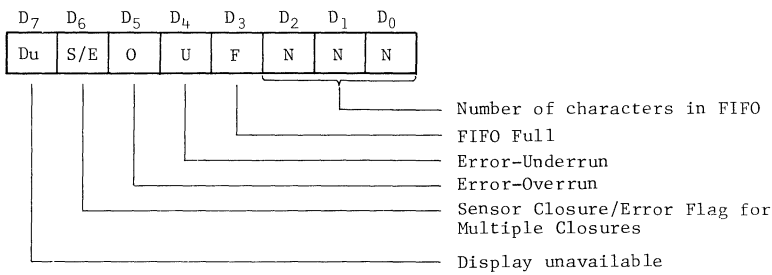
END Interrupt/Error Mode Set

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	1	E	X	X	X	X

X: don't care

In the Sensor Matrix mode, this command loweres the IRQ line and enables writing to the Sensor RAM. This means that a write to the Sensor RAM is inhibited when IRQ line is high. If the E bit is set to "1", the S/E bit of the FIFO status becomes "1" when any one of the sensor switches is closed. If E =0, the S/E bit is always "0". In the N-Key Rollorer, if the E bit is programmed to "1", the Special Error mode will be resulted.

FIFO status



Du : indicates that the Display RAM is unavailable because a Clear Display or Clear All command has not completed its clearing operation.

S/E : In a Sensor Matrix mode, if the E bit of END Interrupt/Error Mode Set is programmed to "1", this S/E bit is set to indicate that at least one sensor closure indication is contained in the Sensor RAM.
In Special Error Mode, this S/E bit is showing the error flag and serves as an indication to whether a simulataneous multiple closure error has occurred.

O : indicates that the entry of another character into a full FIFO was attempted.

U : indicates that the CPU tried to read an empty FIFO.

F : indicates that the FIFO is full of the eight characters.

NNN : indicate number of characters in the FIFO when in the Keyboard Mode or in the Strobe Input Mode.

[INTERFACE WITH KEYBOARD]

Scanned Keyboard, 2-key LOCKOUT

In this mode, if one key only is kept depressed during one debounce cycle (2 times of the key scan cycle), the key is recognized. When a key is depressed, the debounce logic is set and the other depressed keys are checked during the next two scan cycle. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If another depressed key are encountered, operates as follows.

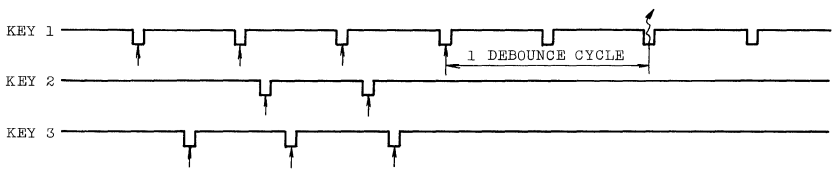
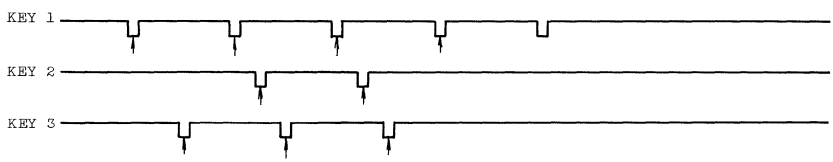


Fig. 1 Example of a case where a first depressed key is continuously kept to the last



↑ ..... Debounce logic is set

↯ ..... Entered to FIFO

Fig. 2 Example of a case where all depressed keys are ignored

As shown in Fig. 1, if all other keys are released before the first depressed key, the first depressed key is recognized. As shown in Fig. 2, if the first depressed key is released within one debounce cycle after the other keys was released, then all keys are ignored.

Scanned Keyboard, N-key Rollover

In this mode, each key depression is independently treated from all others. In the 2-Key lockout mode, if a key is depressed, the debounce logic is set. If other keys are depressed within one debounce cycle after it, the debounce logic is set again. The first depressed key is ignored. In the N-key Rollover mode, if a key is depressed waits one debounce cycle and then checks if the key is still down. If it is, the key is entered into the FIFO even if other keys are depressed.

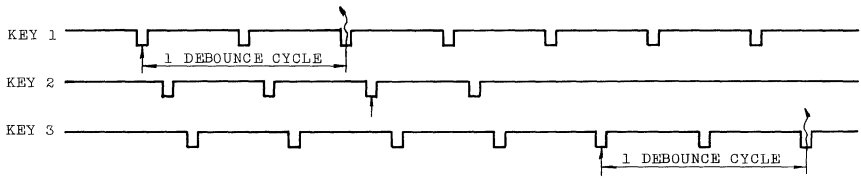


Fig. 3 Example of 3 keys being pushed simultaneously

In the example as shown in Fig. 3, the debounce circuit starts by Key 1, and checks if the key is still down after one debounce cycle. If it is, Key 1 is recognized and Key 2 is ignored not to be depressed for one debounce cycle.

Special Error Mode (N - Key Rollover)

This mode is set if the E bit of the End Interrupt/error Mode Set command is programmed to "1". In the normal N-Key Rollover Mode, the key information is entered to the FIFO according to the key scan timing even if a simulataneous multiple depression occures during one debounce cycle. In the Special Error Mode, if a simulataneous multiple depression occurs during one debounce cycle, sets the error flag (the S/E bit of the FIFO status word) to "1". This flag prevents any further writing into the FIFO and will set interrupt request (IRQ). The S/E bit is cleared if the normal Clear command is written with $C_F = 1$.

Senser Matrix Mode

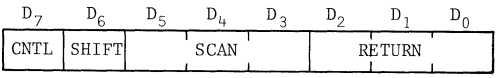
In Sensor Matrix Mode, the debounce circuit does not operate. The status of the senser switch is inputted directly to the Senser RAM. The CPU can know a validated closure in the keyboard, however this mode has such advantage that the CPU knows how long the sensor was closed and when it was released. If there is any change in the sensor value at the end of the sensor matrix scan, the IRQ line goes high. The IRO line is cleared by the first data read if the Auto-increment flag is "0" or by the End Interrupt/Error Mode Set command if $A_I = 1$.

Strobe Input Mode

In Strobe Input Mode, the debounce circuit does not operate. The data is inputted into the FIFO from the return lines at the rising edge of CNTL/STB Signal. When the data is entered into the FIFO, the IRQ line goes high. The functions of the FIFO and the FIFO status in this mode are same as those in the keyboard mode.

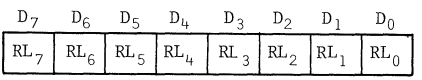
[DATA FORMAT]

Keyboard Mode



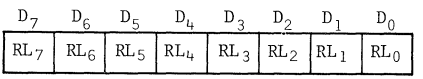
In this mode, the Data Format of the character entered into the FIFO is as follows. The MSB is the status of CNTL/STB line and the next MSB shows the status of SHIFT line. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to whitch return line the key was connected.

Sensor Matrix Mode



In this mode, the data on return lines is inputted in the row of the Senser RAM in order according to the scan. The data is entered even if there is no change in the status of the senser matrix switches. Each switch position maps to a Senser RAM position. CNTL and SHIFT signals are ignored.

Strobe Input Mode

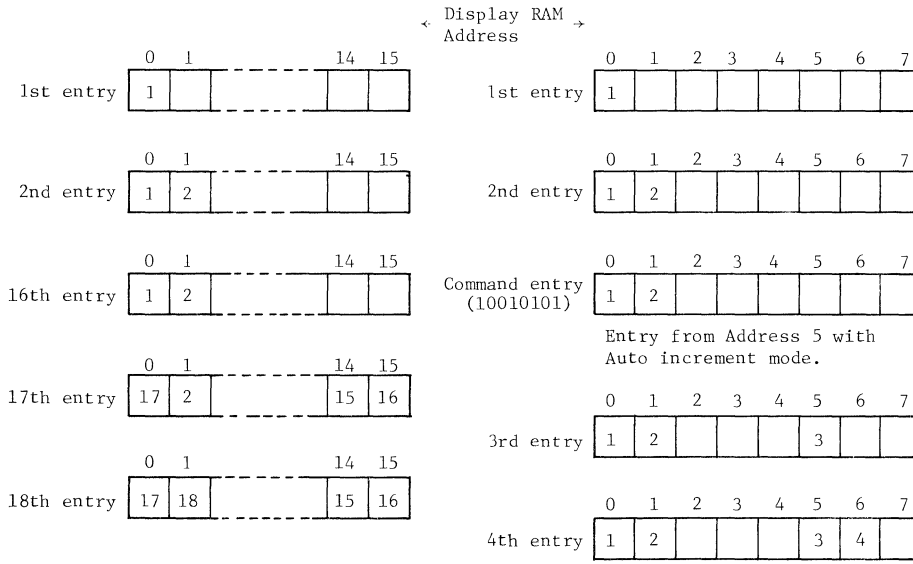


In this mode, the data on the return line is entered into the FIFO at the rising edge of CNTL/STB signal.

[INTERFACE WITH DISPLAY]

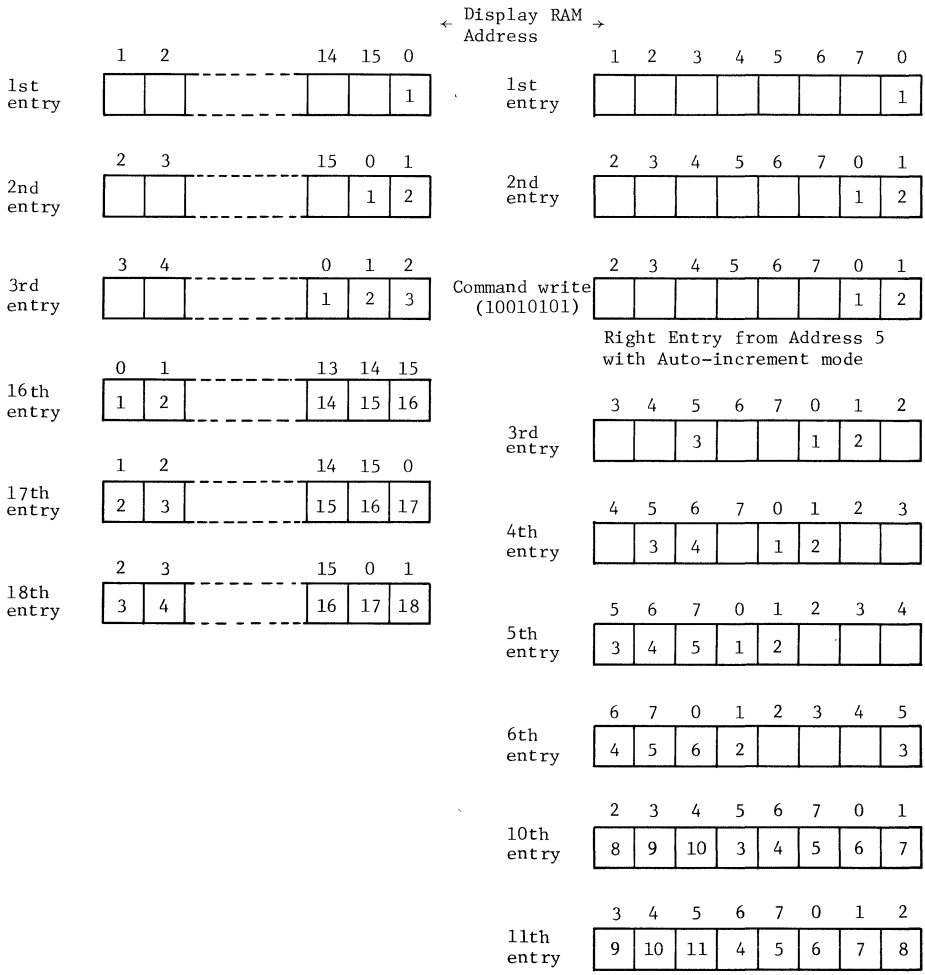
Left Entry

In Left Entry mode, address 0 of the Display RAM is the left-most side of the display and address 15 (address 7 in the case of 8-character display) is the right-most side. When characters are inputted onto the Display RAM with the auto-increment mode from address 0 of the display RAM, Characters are filled from the left-most position of the display. The 17th (or 9th) character is placed in the left-most position again. Address of the display RAM corresponds directry to each display position of the display, and so its position does not change every entry.



Right Entry

In Right Entry, the first entry is from the right-most position. Address of the Display RAM does not correspond to the display position.


← Display RAM Address →
1st entry
2nd entry
3rd entry
16th entry
17th entry
18th entry
Command write (10010101)
Right Entry from Address 5 with Auto-increment mode
3rd entry
4th entry
5th entry
6th entry
10th entry
11th entry

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{CC}	V_{CC} Supply Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
V_{IN}	Input Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
V_{OUT}	Output Voltage (with respect to GND (V_{SS}))	-0.5V to +7.0V
P_D	Power Dissipation	1W
T_{sol}	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. ELECTRICAL CHARACTERISTICS (Ta=0 ~ 70°C, V_{CC}=5V±10%, V_{SS}=0V)

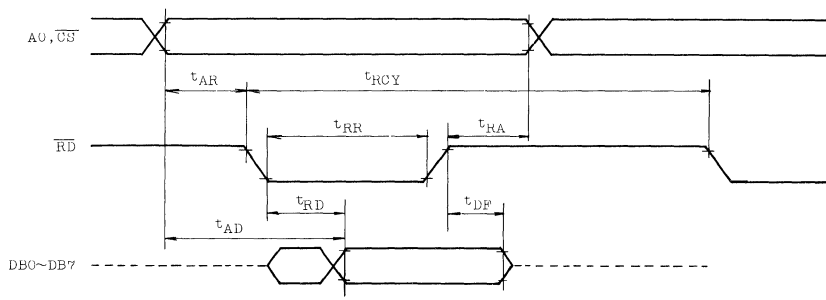
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IL1}	Input Low Voltage ($RL_0 \sim RL_7$)		-0.5		1.4	V
V_{IL2}	Input Low Voltage (Others)		-0.5		0.8	V
V_{IH1}	Input HIgh Voltage ($RL_0 \sim RL_7$)		2.2			V
V_{IH2}	Input High Voltage (Others)		2.0			V
V_{OL}	Output Low Voltage	I_{OL}=2.2mA			0.45	V
V_{OH1}	Output High Voltage (IRQ)	I_{OH}=-100μA	3.5			V
V_{OH2}	Output High Voltage (Others)	I_{OH}=-400μA	2.4			V
I_{IL1}	Input Leak Current (SHIFT, CNTL, $RL_0 \sim RL_7$)	$V_{IN}=V_{CC}$			+10	μA
		V_{IN}=0V			-100	
I_{IL2}	Input Leak Current (Others)	$0V \leqq V_{IN} \leqq V_{CC}$			±10	μA
I_{OFL}	Output Leak Current	$0.45V \leqq V_{OUT} \leqq V_{CC}$			±10	μA
I_{CC}	Supply Current				120	mA

INPUT CAPACITY

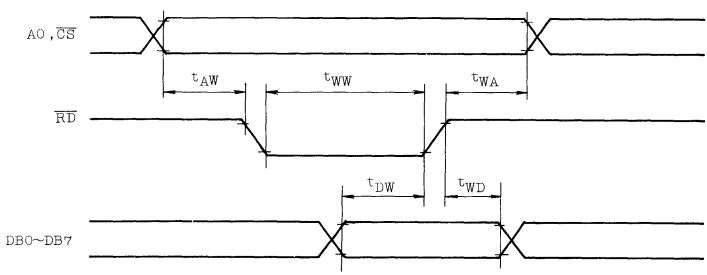
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
C_{IN}	Input Capacity	f_C=1MHz Unmeasured Pins returned to V_{SS}.		5	10	pF
C_{OUT}	Output Capacity			10	20	pF

A.C. ELECTRICAL CHARACTERISITCS ($Ta=0 \sim 70°C$, $V_{CC}=5.0V \pm 10\%$, $V_{SS}=0V$)

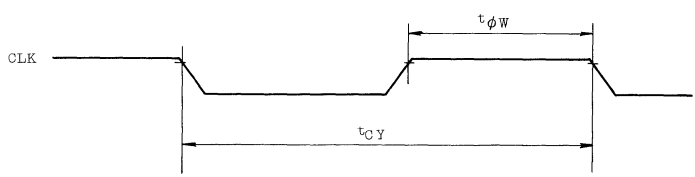
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{AR}	Address Set up Time ($\overline{RD}\downarrow$)		0			nS
t_{RA}	Address Hold Time ($\overline{RD}\uparrow$)		0			nS
t_{RR}	$\overline{RD}$ Pulse Width		250			nS
t_{RD}	Valid Data ($\overline{RD}\uparrow$)	$C_L=150pF$			150	nS
t_{AD}	Address to Valid Data	$C_L=150pF$			250	nS
t_{DF}	Data Floating ($\overline{RD}\uparrow$)		10		100	nS
t_{RCY}	Read Cycle Time		1			μS
t_{AW}	Address Set up Time ($\overline{WR}\downarrow$)		0			nS
t_{WA}	Address Hold Time ($\overline{WR}\uparrow$)		0			nS
t_{WW}	$\overline{WR}$ Pulse Width		250			nS
t_{DW}	Data Set up Time ($\overline{WR}\uparrow$)		150			nS
t_{WD}	Data Hold Time ($\overline{WR}\uparrow$)		0			nS
$t_{\phi W}$	CLK Pulse Width		120			nS
t_{CY}	Clock period		320			nS



Read-operation



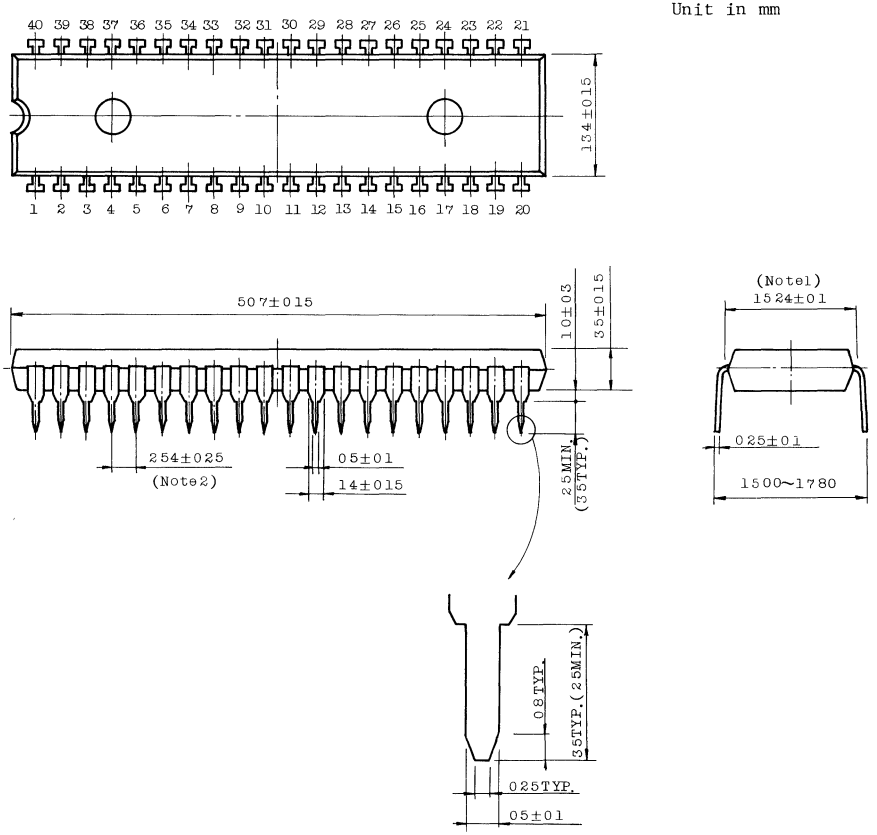
Write-operation



Clock input

OUTLINE DRAWING

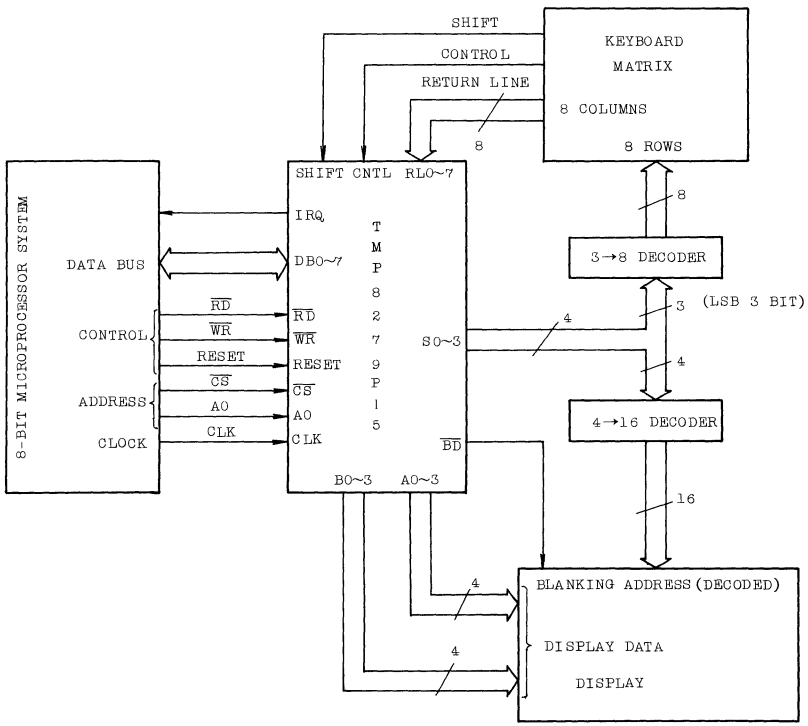
Unit in mm



Note: 1. This dimension is measured at the center of bening 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.

EXAMPLE OF APPLICATION CIRCUIT



Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8355P

N-CHANNEL SILICON GATE MOS

16,384 BIT ROM WITH I/O PORTS

GENERAL DESCRIPTION

The TMP8355P is a ROM and I/O chip to be used in the TLCS-85A microcomputer system. The ROM portion is organized as 2,048 words by 8 bits.
The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is individually programmable as input or output.

FEATURES

- 2048 words x 8 bits ROM
- Single + 5V Power Supply
- Internal Address Latch
- 2 General Purpose 8-Bit I/O Ports
- Access Time : 400 ns (MAX.)
- Each I/O Port Line Individually Programmable as Input or Output
- Multiplexed Address and Data Bus
- 40 pin DIP
- Compatible with Inptel's 8355

PIN CONNECTIONS (TOP VIEW)

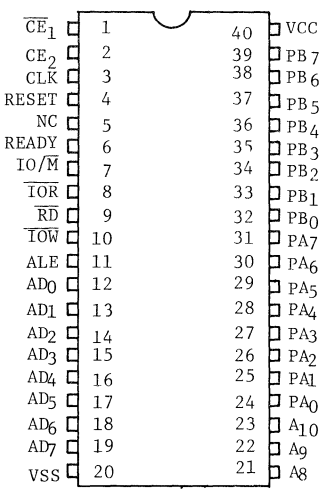


FIGURE 1 TMP8355P PINOUT DIAGRAM

BLOCK DIAGRAM

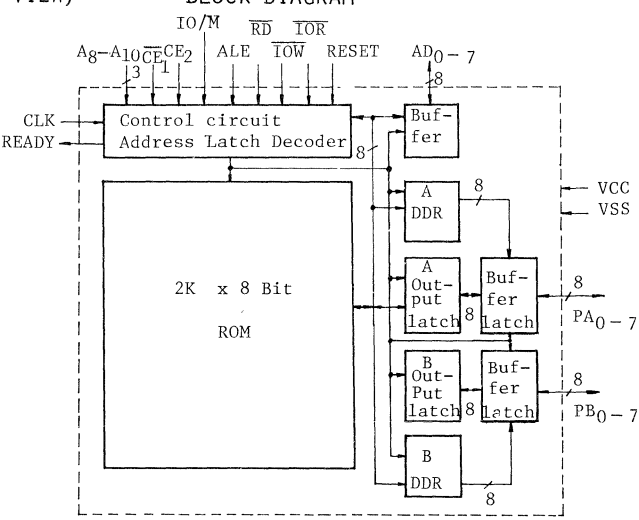


FIGURE 2 TMP8355P FUNCTIONAL BLOCK DIAGRAM

PIN NAMES AND PIN DESCRIPTION

ALE (INPUT)

When Address Latch Enable goes high, AD_{0-7}, IO/$\overline{M}$, A_{8-10}, CE_2, and $\overline{CE_1}$, enter the address latches. The signals (AD_{0-7}, IO/$\overline{M}$, A_{8-10}, CE_2, $\overline{CE_1}$) are latched in at the trailing edge of ALE.

AD_{0-7} (INPUT/OUTPUT, 3-STATE)

Bi-directional Address/Data bus. The lower 8-bits of the ROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B are selected based on the latched value of AD_0. If $\overline{RD}$ or $\overline{IOR}$ is low when the latched Chip Enables are active, the output buffers present data on the bus.

A_{8-10} (INPUT)

These are the high order bits of the ROM address. They do not affect I/O operations.

$\overline{CE1}$, CE2 (INPUT)

CHIP ENABLE INPUTS: $\overline{CE_1}$ is active low and CE_2 is active high. Both chip enables must be active to permit accessing the ROM.

IO/$\overline{M}$ (INPUT)

If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the ROM.

$\overline{RD}$ (INPUT)

If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected ROM location or I/O port.
When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are 3-stated.

$\overline{IOW}$ (INPUT)

If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}.
The state of IO/$\overline{M}$ is ignored.

CLK (INPUT)

The CLK is used to force the READY into its high state after it has been forced low by $\overline{CE_1}$ low, CE_2 high, and ALE high.

READY (OUTPUT, 3-STATE)

READY is a 3-state output controlled by $\overline{CE}_1$, CE_2, ALE and CLK.
READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK.

PA_0 - PA_7 (INPUT/OUTPUT, 3-STATE)

These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active, and $\overline{IOW}$ is low and a 0 was previously latched from AD_0.
Read operation is selected by either $\overline{IOR}$ low, active Chip Enables and AD_0 low, or IO/$\overline{M}$ high, $\overline{RD}$ low, active Chip Enables, and AD_0 low.

PB_0 - PB_7 (INPUT/OUTPUT, 3-STATE)

This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0.

RESET (INPUT)

In normal operation, an input high on RESET causes all pins in Ports A and B to assume input mode (clear DDR register).

$\overline{IOR}$ (INPUT)

When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination of IO/$\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V_{CC} "1".

V_{CC} (POWER)

+5 volt supply.

V_{SS} (POWER)

Ground Reference

FUNCTIONAL DESCRIPTION

ROM SECTION

The TMP8355P contains an 8-bit address latch which allows it to interface circetly to TLCS-85A microcomputer system without additional hardware.
The ROM portion of the chip is addressed by the 11-bit address (A8-10, AD_{0-7}) and CE. The address, IO/$\overline{M}$, CE_2 and $\overline{CE_1}$ are latched into the address latches on falling edge of ALE. If the Chip Enables (CE_2 and $\overline{CE_1}$) are active and IO/$\overline{M}$ is low when $\overline{RD}$ goes low, the contents of the ROM location addressed by the latched address are put out on the AD_{0-7} lines.

I/O SECTION

The I/O port portion consits of two 8-bit I/O ports and two 8-bit Data Direction Registers (DDR). The I/O portion of the chip is addressed by the latched value of AD_0 and AD_1. Contents of Port A and Port B can be read and written, but the contents of DDR's cannot be read. The contents of the selected I/O port can be read out when the latched Chip Enable are active and either $\overline{R}$ goes low with IO/$\overline{M}$ high, or $\overline{IOR}$ goes low.
The two 8-bit DDR's (DDRA and DDRB) are used to determine the input/output status of each pin in the corresponding port.
A '0' specifies an input mode and a '1' specifies an output mode.
The two 8-bit DDR's are cleared by RESET signal. The table 1 summarize Port and DDR designation.

TABLE 1, SELECTION OF PORT AND DDR DESIGNATION

AD_1	AD_0	Selection
0	0	Port A
0	1	Port B
1	0	Port A Data Direction Register (DDR A)
1	1	Port B Data Direction Register (DDR B)

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{CC}	V_{CC} Supply Voltage with Respect to V_{SS}	-0.5V to 7.0V
V_{IN}	Input Voltage with Respect to V_{SS}	-0.5V to 7.0V
V_{OUT}	Output Voltage with Respect to V_{SS}	-0.5V to 7.0V
P_D	Power Dissipation	1.5W
T_{SOLDER}	Soldering Temperature (Soldering Time 10sec.)	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 $\pm$ 5%

Symbol	Parameter	Test Conditions	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_{OL}	Output Low Voltage	I_{OL} = 2mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = -400μA	2.4			V
I_{IL}	Input Leakage Current	V_{IN} = V_{CC} to 0V			$\pm$ 10	μA
I_{LO}	Output Leakage Current	0.45 ≤ Vout ≤ V_{CC}			$\pm$ 10	μA
I_{CC}	V_{CC} Supply Current				180	mA

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 5%

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
t_{CYC}	Clock Cycle Time		320			ns
t_L	CLK Low Width		80			ns
t_H	CLK High Width		120			ns
t_r, t_f	CLK Rise and Fall Time				30	ns
t_{AL}	Address to Latch Set Up Time		50			ns
t_{LA}	Address Hold Time after Latch		80			ns
t_{LC}	Latch to READ/WRITE Control		100			ns
t_{RD}	Valid Data Out Delay from READ Control	150pF			170	ns
t_{AD}	Address Stable to Data Out Valid	Load			400	ns
t_{LL}	Latch Enable Width		100			ns
t_{RDF}	Data Bus Float after READ		0		100	ns
t_{CL}	READ/WRITE Control to Latch Enable		20			ns
t_{CC}	READ/WRITE Control Width		250			ns
t_{DW}	Data In to WRITE Set Up Time		150			ns
t_{WD}	Data In Hold Time after WRITE		10			ns
t_{WP}	WRITE to Port Output				400	ns
t_{PR}	Port Input Set Up Time		50			ns
t_{RP}	Port Input Hold Time		50			ns
t_{RYH}	READY Hold Time		0		160	ns
t_{ARY}	ADDRESS (CE) to READY				160	ns
t_{RV}	Recovery Time between Controls		300			ns
t_{RDE}	Data Out Delay from READ Controls		10			ns
t_{LCK}	ALE Low during CLK High		100			ns

TIMING WAVEFORMS

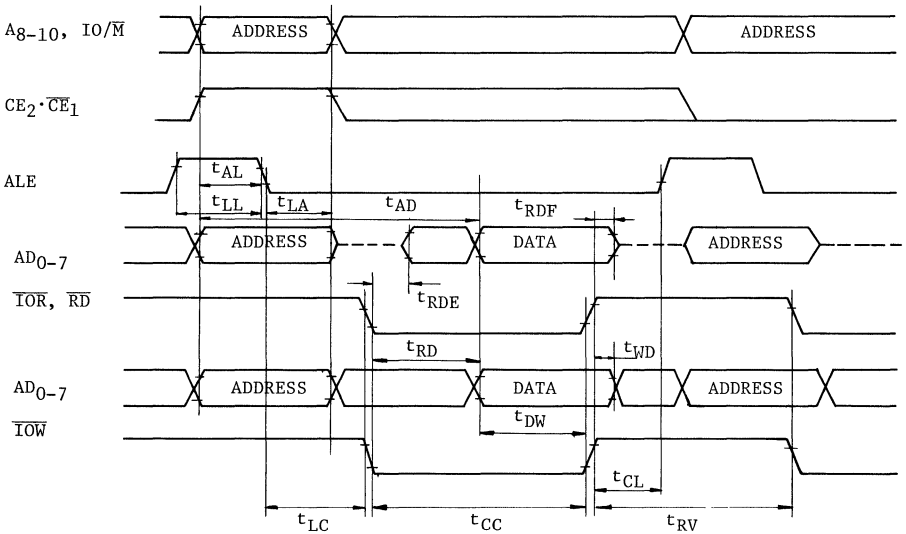


FIGURE 3 PROM READ, I/O READ, AND WRITE TIMING

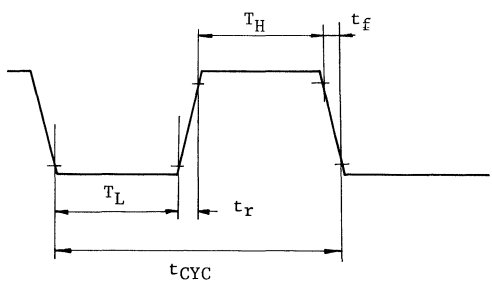


FIGURE 4 CLOCK SPECIFICATION FOR TMP8355P

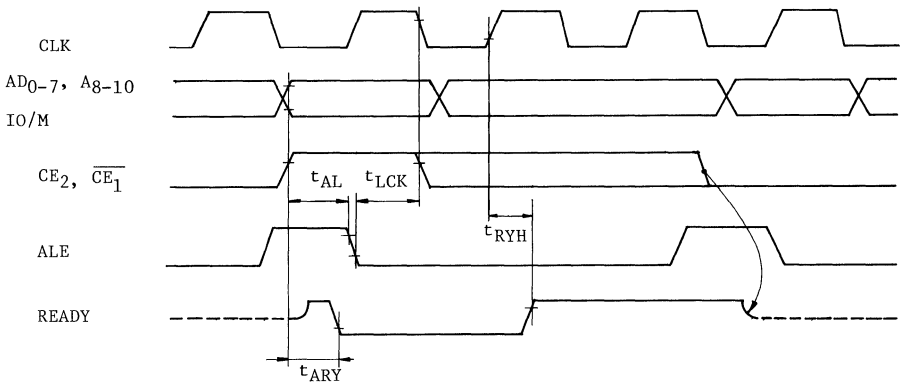
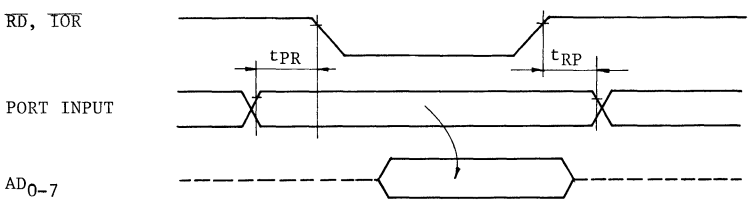


FIGURE 5 WAIT STATE TIMING (READY = 0)

A. INPUT MODE



B. OUTPUT MODE

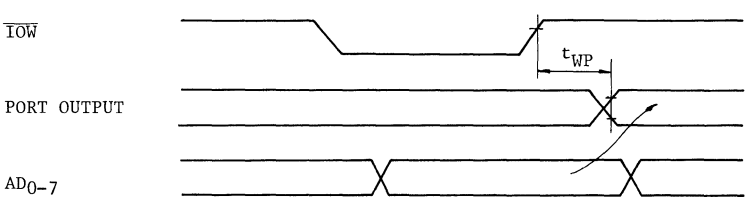
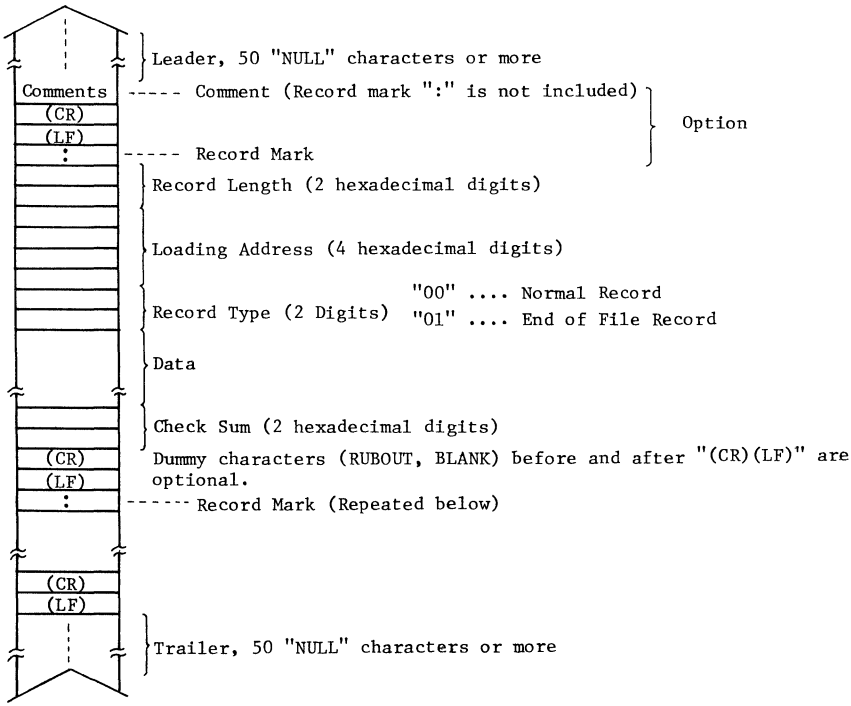


FIGURE 6 I/O PORT TIMING

PROGRAM TAPE FORMAT

TMP8355P programs are delivered in the form of punched paper tape or the 8755A from which to copy. In case of the 8755A, Toshiba needs two pieces.

(1) Tape Format

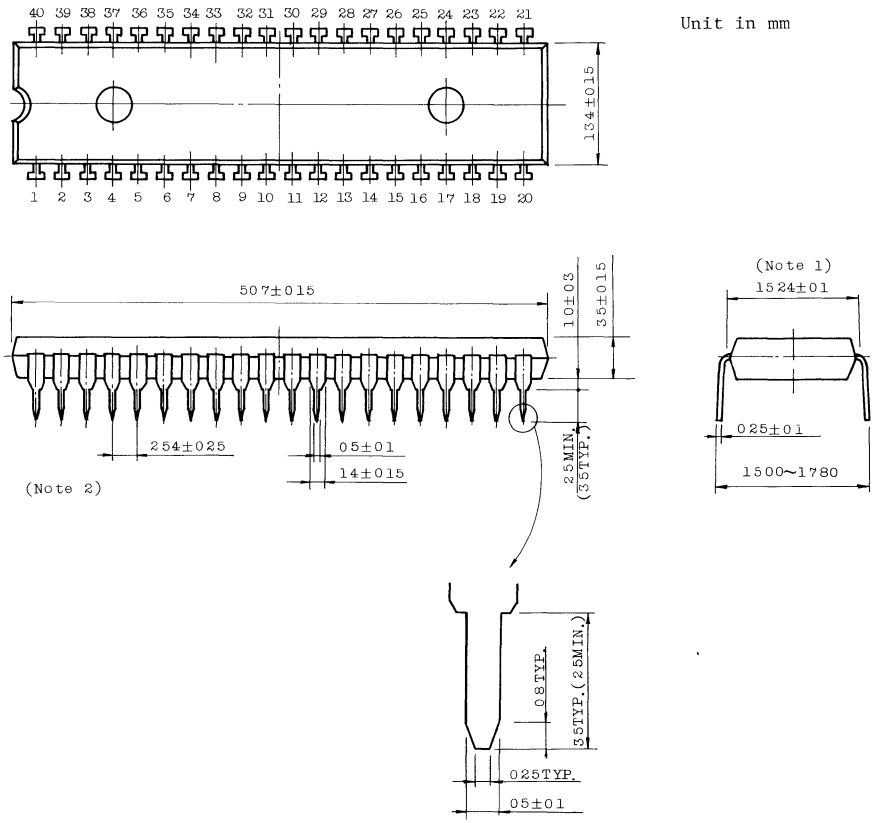


(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-84
: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.

POSTSCRIPT

This Manual is a reference for the customer applying the TLCS-85 family, It contains the functions and specifications of each LSI device of the TLCS-85, The examples of application circuits described here are shown as the reference for explanation. Toshiba assumes no responsibility for any problems caused by using these examples. the information herein is subject to change without prior notice.

This manual has been prepared by the following section.

Microconputer LSI Application Engineering Sections
Integrated Circuit Division, Toshiba Corpration

1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 210, Japan
Phone: Japan (81)44-511-3111

OVERSEAS OFFICES

São Paulo:
Toshiba Brasileira Representacoes Ltda.
Av. Paulista, 807 21 Andar Cjto. 2101, 2102, 2103.
Cerqueira Cesar.
Cep. 01311—Sao Paulo—S.P.—Brasil
Tel.: 283-4714, 4964, 285-4519

New Zealand:
Toshiba Corporation Representative in New Zealand
6th Floor, Databank House, 175 The Terrace
P.O. Box 3549 Wellington, New Zealand
Tel.: 726-001 Telex: NZ-3433
Cable: Toshiba Wellington

Athens:
Toshiba Corporation Athens Office
Athens Tower Bldg. A, 2-4 Mesogion Ave.,
Athens 610 Greece
Tel: 7799828-9 Telex: 21-6502 TSBA GR
Cable: Toshiba Athens

Iran:
Toshiba Corporation Iran Liaison Office
No. 79 Bucharest Ave., 3rd Floor, Argentine
Square, Teheran, Iran
P.O. Box 314/1696, Teheran, Iran
Tel.: 624709, 624710, 624729
Telex: 212531-TSBA IR Cable: Toshiba Teheran

SALES SUBSIDIARIES

Toshiba America, Inc.
Tustin Head Office
Electronic Components Division: 2441 Michelle Drive, Tustin, Ca 92680, U.S.A.
Tel.: (714)730-5000 Telex: 183-812
Fax: 714(730)8902
Branch Office
Chicago: 1101A Lake Cook RD. Deerfield, IL 60015
Tel.: (312)945-1500 Telex: 29-7131
Fax: 312-945-1044
Detroit: 26533 Evergreen RD. Suite 420 Southfield, Michigan 48076, U.S.A.
Tel.: 313(827)7700 Telex: 858798
Fax: 313(569)7174

Toshiba (UK) Ltd.
Frimley: Toshiba House, Frimley Road, Frimley Camberley, Surrey GU 165JJ England
Tel.: (0276)-62222 Telex: 858798
Fax: (0276)682256

Toshiba Europa (I.E.) GmbH
Head Office
Neuss: Hammer Landstrasse 115, 4040 Neuss 1, F.R. Germany
Tel.: (02101)1580 Telex: 8517926
Fax: (02101)158341 TOSD
Liaison Office
München: Büro München Arabellastr. 33/v 80000 München 81
Tel.: (089)915061-66 Telex: 5-212 363
Fax: 089-913579
Paris: Tour de Bureaux de Rosny 2, 93118 Rosny, Sous Bois, France
Tel.: 855 56 56 Telex: 613351
Fax: 855-5248 F TOSPAR
Stuttgart: Zeppelin Str., 41 7302 Ostfildern 4 F.R. Germany
Tel.: (0711)45 2054 Telex: 722403 TOSS

Toshiba Electronics Scandinavia A.B.
Banergatan 21-23, S-115 22
Stockholm Sweden
Tel.: (08)635240 Telex: 14169 TSBSTK S
Fax: (08)636533

Toshiba (Australia) Pty. Ltd.
Sydney: 84-92 Talavera Road North Ryde N.S.W. 2113 Australia
Tel.: (02)887-3322 Telex: AA27235

Toshiba Electronics Hong Kong Ltd.
Suite 423-5 Ocean Center, Canton Road,
Kowloon, Hong Kong
Tel.: 002-852-3-671-141~4

Toshiba Electronics Taiwan Corp.
7F, Min Sheng Commerical Bldg. 344, Min Sheng East Road, Taipei, Taiwan
Tel.: 2-53-3934 Telex: 26874 "TRCTR"

Toshiba Trading Singapore Pt. Ltd.
2405/06, 24th floor, Orchard Tower, 400 Orchard Road, Singapore 0923, Republic of Singapore
Tel.: 002-65-737-3911

MANUFACTURING SUBSIDIARIES AND JOINT VENTURES

Toshiba Singapore Pte. Ltd.
818/826, 8th Floor Block 2, PSA Multi-Storey Complex, Pasir Panjang Road, Singapore 0511
Tel.: 2718066

Thai Toshiba Electric Industries Co., Ltd.
18/1, Tivanon Road, Nonburi, Thailand
Tel.: 5880002, 5883010 TT Telex: 82020

Toshiba Electronic Malaysia Sdn., Bhd.
Batu 9-12 Telok Panglima, Garang, Kuala Langat, Selangor, Malaysia
Tel.: 03-37300~4 Telex: TOELMA 39506

Toshiba (Malaysia) Bhd.
Batu Tiga Industiral Estate,, Shan Ajam, Selangor, Malaysia

Toshiba Semiconductor (U.S.A.) Inc.
1220, Midas Way, Sunnyvale, P.O. Box 3509 Calif. 94086-3509, U.S.A.
Tel.: 408-739-0560 Fax: 408-746-0577

Radiola Toshiba Philippines, Inc.
19, Katarungan Street, Mandaluyong, Rizal. Philippines

Paris Toshiba Industrial Co., Ltd.
No. 55, Iranshahr Avenue, Teheran, Iran

Industrial Mexicana Toshiba S.A.
Calzada Aurora No. 303, Cuautitlan Edo de Mexico, Mexico
Paseo de La Reforma No. 30 4 Piso Mexico 1. D.F. Mexico
Tel.: 5-65-00-88 Telex: 017-72-560

Semp Toshiba Amazonas S.A.
Rua Içá No. 500 Distrito Industrial, Manaus, 69000 Am., Brasil

Toshiba Semiconductor GmbH
Grotrian-Steinweg Str. 10 3300
Braunschweig F.R. Germany
Tel.: (0531)310060 Telex: 952368
Fax: (0531)31006139 TSCD

TOSHIBA

TOSHIBA CORPORATION TOKYO JAPAN

INTERNATIONAL OPERATIONS-ELECTRONIC COMPONENTS
1-1 Shibaura 1-chome, Minato-ku, Tokyo, 105, JAPAN
Telex: J22587 TOSHIBA CABLE: TOSHIBA Tokyo
PHONE: 457-3495
Facsimile: 595-0348

'84-8 Ⓓ Printed in Japan