INTEGRATED CIRCUIT TECHNICAL DATA

MICROCOMPUTER DATA BOOK

2nd Edition

Mar., 1982

Toshiba TOSHIBA

PREFACE

Since LSIs for the microcomputer manufactured first in Japan by Toshiba was put on the market in 1973, Toshiba has devoted efforts in strengthening its semiconductor devices for microcomputer. Toshiba has been developing 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 users.

This manual covers the technical description of LSIs for microcomputers supplied by Toshiba, including TLCS-43 Series (NMOS), TLCS-46A Series (CMOS) and TLCS-47 Series (NMOS and CMOS) 4-bit single chip microcomputer, TLCS-84 Series (NMOS and CMOS) 8-bit single chip microcomputer and TLCS-85A Family (NMOS) 8-bit microcomputer.

In addition to this manual, the technical description of the microcomputer development system is also available.

Further, examples of application circuits listed in this manual are only for reference in using the LSIs. Toshiba will not assume any responsibility for problem relative to patent rights which may be generated from use of the said application circuits.

CONTENTS

Page

4BIT SINGLE CHIP MICROCOMPUTER

TLCS-43 (NMOS)

- TMP4310AP/TMP4315BP/TMP4320AP/TMP4300C 7
- TMP4310APL/TMP4310APLL 82
- TMP4321AP 88
- TMP4399C 93

4BIT SINGLE CHIP MICROCOMPUTER

TLCS-46A (CMOS)

- TCP4620AP/TCP4630AP 99
- TCP4620AF/TCP4630AF 173
- TCP4620BP 179
- TCP4621AP 195
- TCP4632BF . . . . PRELIMINARY 201
- TCP4600AC 230

4BIT SINGLE CHIP MICROCOMPUTER

TLCS-47 (NMOS/CMOS) . . . . . . PRELIMINARY

- TMP4700C/TMP4720P/TMP4740P/TMP499C (NMOS) 257
- TMP47C20P/TMP47C40P/TMP47C22F (CMOS) /TMP47C41P/TMP47C21P 357

8BIT SINGLE CHIP MICROCOMPUTER

TLCS-84 (NMOS/CMOS)

- TMP8048P/TMP8035P/TMP8048PI/TMP8035PI 509
- TMP8049P/TMP8049P-6/TMP8039P/TMP8039P-6 /TMP8049PI-6/TMP8039PI-6 542
- TMP8243P/TMP8243PI 575
- TMP8022P 584
- TMP80C49-6/TMP80C39P-6 . . . . PRELIMINARY 602

- TMP82C43P 615

8BIT MICROPROCESSOR
TLCS-85A (NMOS)

- TMP8085AP 623

8BIT MICROPROCESSOR
PERIPHERALS (NMOS)

- TMP8155/TMP8156P 655
- TMP8355P 672
- TMP8755AC 682
- TMP8259AP 694

MICROCOMPUTER APPLICATION DEVELOPMENT TOOL 701

TLCS-43 (4 BIT SINGLE CHIP NMOS MICROCOMPUTER)

TLCS-43 SERIES (NMOS)

Device Number	RAM	ROM	I/O	Instruction Cycle Time (Min.)	Power Supply Current (Typ.)	Power Supply	Operating Temperature	Package Type	Alternate Source
TMP4300C	128 × 4	Without ROM	35 Lines	4 μs	70 mA	5V±10%	-10°C ∿ 70°C	Ceramic 64 pin	Toshiba Original
TMP4399C	128 × 4	Without ROM	35 Lines	4 μs	70 mA	5V±10%	-10°C ∿ 70°C	Ceramic 42 pin	
TMP4310AP	48 × 4	1024 × 8	22 Lines	4 μs	40 mA	5V±10%	-10°C ∿ 70°C	Plastic 28 pin	
TMP4315BP	64 × 4	1536 × 8	35 Lines	4 μs	40 mA	5V±10%	-10°C ∿ 70°C	Plastic 42 pin	
TMP4320AP	128 × 4	2048 × 8	35 Lines	4 μs	40 mA	5V±10%	-10°C ∿ 70°C	Plastic 42 pin	
TMP4321AP	128 × 4	2048 × 8	35 Lines	4 μs	40 mA	5V±10%	-10°C ∿ 70°C	Plastic 42 pin	
TMP4310APL	48 × 4	1024 × 8	22 Lines	5 μs	30 mA	5.5V±10%	0°C ∿ 55°C	Plastic 28 pin	
TMP4310APLL				10 μs	15 mA				

TLCS-46A (4 BIT SINGLE CHIP CMOS MICROCOMPUTER)

TLCS-46A SERIES (CMOS)

Device Number	RAM	ROM	I/O	Instruction Cycle Time (Min.)	Power Supply Current (Typ.)	Power Supply	Operating Temperature	Package Type	Alternate Source
TCP4600AC	160 × 4	Without ROM	34 Lines	10 μs	0.15 mA at 100kHz	4 ∿ 6V	-10°C ∿ 70°C	Ceramic 64 pin	Toshiba Original
TCP4620AP	96 × 4	2048 × 8			0.4 mA at 400kHz	4 ∿ 6V	-30°C ∿ 85°C	Plastic 42 pin	
TCP4620BP									
TCP4630AP	160 × 4	3072 × 8							
TCP4620AF	96 × 4	2048 × 8					-20°C ∿ 70°C	Plastic 43 pin (Flat)	
TCP4620BF									
TCP4630AF	160 × 4	3072 × 8							
TCP4621AP	96 × 4	2048 × 8						Plastic 42 pin	
TCP4632BF	160 × 4	3072 × 8	57 Lines					Plastic 67 pin (Flat)	

TLCS-47 (4 BIT SINGLE CHIP NMOS/CMOS MICROCOMPUTER)

TLCS-47 SERIES (NMOS/CMOS)

	Device Number	RAM	ROM	I/O	Instruction Cycle Time (Min.)	Power Supply Current (Typ.)	Power Supply	Operating Temperature	Package Type	Alternate Source
NMOS	TMP4720P	128 × 4	2048 × 8	35 Lines	2 μs	50 mA	5V±10%	-30°C ∿ 70°C	Plastic 42 Pin	Toshiba Original
	TMP4740P	256 × 4	4096 × 8							
	TMP4700C	256 × 4	without ROM						QIC 80 Pin	
	TMP4799C	256 × 4	without ROM						DIC 42 Pin	
CMOS	TMP47C20P	128 × 4	2048 × 8		4 μs	5 mA			Plastic 42 Pin	
	TMP47C40P	256 × 4	4096 × 8							
	TMP47C21P	128 × 4	2048 × 8							
	TMP47C41P	256 × 4	4096 × 8							
	TMP47C22F	192 × 4	2048 × 8	55 Lines					Plastic 67 Pin (Flat)	

TLCS-84 (8 BIT SINGLE CHIP NMOS/CMOS MICROCOMPUTER)

TLCS-84 SERIES (NMOS/CMOS)

	Device Number	RAM	ROM	I/O	Instruction Cycle Time (Min.)	Power Supply Current (Max.)	Power Supply	Operating Temperature	Package Type	Alternate Source
NMOS	TMP8048P	64 × 8	1024 × 8	27 Lines	2.5 μs	135 mA	5V±10%	0°C ∿ 70°C	Plastic 40 pin	i8048
	TMP8035P	64 × 8	–		2.5 μs	135 mA				i8035L
	TMP8049P-6	128 × 8	2048 × 8		2.5 μs	170 mA				
	TMP8039P-6	128 × 8	–		2.5 μs	170 mA				i8039-6
	TMP8049P	128 × 8	2048 × 8		1.36 μs	170 mA				i8049
	TMP8039P	128 × 8	–		1.36 μs	170 mA				i8039
	TMP8048PI	64 × 8	1024 × 8		2.5 μs	145 mA		-40°C ∿ 85°C		iD8048
	TMP8035PI	64 × 8	–		2.5 μs	145 mA				iD8035
	TMP8049PI-6	128 × 8	2048 × 8		2.5 μs	170 mA				iD8049
	TMP8039PI-6	128 × 8	–		2.5 μs	170 mA				iD8039
	TMP8243P	I/O Expander for TMP8048P/TMP8049P				20 mA	5V±10%		Plastic 24 pin	i8243
	TMP8243PI	I/O Expander for TMP8048PI-6/8049PI-6				20 mA	5V±10%	-40°C ∿ 85°C	Plastic 24 pin	iD8243
CMOS	TMP80C49P-6	128 × 8	2048 × 8	27 Lines	2.5 μs	10 mA	5V±10%	-40°C ∿ 85°C	Plastic 40 pin	
	TMP80C39P-6	128 × 8	–		2.5 μs	10 mA				

TLCS-85A

- 8 BIT MICROPROCESSOR

Device Number	Description	Instruction Cycle Time (Min.)	Power Supply Current (Max.)	Power Supply	Operating Temperature	Package Type	Alternate Source
TMP8085AP	8 Bit Microprocessor	1.3 μs	170 mA	5V±10%	0°C ∿ 70°C	Plastic 40 pin	i8085A

- PERIPHERALS (NMOS)

Device Number	Description	Power Supply Current (Max.)	Power Supply	Operating Temperature	Package Type	Alternate Source
TMP8155P	256 Byte RAM with I/O Ports and Timer (CE: Active Low)	180 mA	5V±5%	0°C ∿ 70°C	Plastic 40 pin	i8155
TMP8156P	256 Byte RAM with I/O Ports and Timer (CE: Active High)	180 mA	5V±5%		Plastic 40 pin	i8156
TMP8355P	2K Byte Mask ROM with I/O Ports	180 mA	5V±5%		Plastic 40 pin	i8355
TMP8755AC	2K Byte EPROM with I/O Ports	180 mA	5V±5%		Ceramic 40 pin	i8755A

4BIT SINGLE CHIP MICROCOMPUTER

TLCS-43(NMOS)

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL
INTEGRATED CIRCUIT
TMP4310AP TMP4315BP
TMP4320AP TMP4300C
Silicon Monolithic
N-Channel Silicon Gate Depression Load

GENERAL DESCRIPTION

TLCS-43 is a complete single chip micro computer series having an internal 4 bit parallel processing function which is suitable for controller applications.
It contains ROM (read only memory) which stores the control programs and the fixed data, RAM (read/write memory) which temporarily stores various data nd a plural number of input/output ports.

In order to provide for a variety of applications the TLCS-43 provides short instruction execution time, multiple subroutine nesting, and flexible input/output ports.
By combining index instructions with processing instructions, the same instruction can be executed for all the registers and all the input/output ports enabling highly efficient programes to be written.

In TLCS-43, there are three versions, TMP4310AP, TMP4315BP and TMP4320AP each of which has different memory capacity and different number of input/output lines, so that the optimum version for a specific application can be selected. Furthermore, TMP4300C is available as the evaluator chip.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4310AP TMP4315BP
TMP4320AP TMP4300C

FEATURES

- TMP4310AP
  1024 x 8 ROM
  48 x 4 RAM
  22 I/O Lines

- TMP4315BP
  1536 x 8 ROM
  64 x 8 RAM
  35 I/O Lines

- TMP4320AP
  2048 x 8 ROM
  128 x 4 RAM
  35 I/O Lines

- TMP4300C
  Evaluator Chip for TLCS-43

- 35 Basic instructions
  31 Processing Instructions
  4 Index Instructions

- 4 Level Subroutine Nesting

- Single Level External Interrupt

- 4 μs Instruction Executuion Time

- Single 5V Supply

- ROM Data Readout Instructions

- LED Direct Drive Capability
  (Except TMP4315BP)

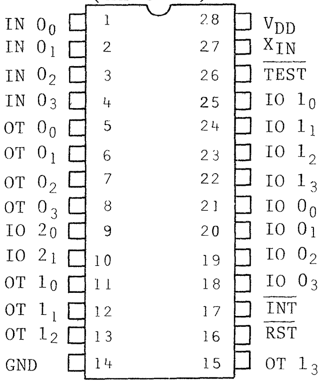
TABLE OF FEATURES

Feature	TMP4310AP	TMP4315BP	TMP4320AP	TMP4300C
ROM Capacity	1,024 Words x 8 Bits	1,536 Words x 8 Bits	2,048 Words x 8 Bits	External connection 2,048 Words x 8 Bits
RAM Capacity	48 Words x 4 Bits	64 Words x 4 Bits	128 Words x 4 Bits	128 Words x 4 Bits
Input Port	1 Port (4 Bits)	3 Port (12 Bits)	3 Port (12 Bits)	3 Ports (12 Bits)
Output Port	2 Port (8 Bits)	4 Port (15 Bits)	4 Port (15 Bits)	3 Port (12 Bits)
Input/Output Port	3 Port (10 Bits)	2 Port (8 Bits)	2 Port (8 Bits)	3 Port (11 Bits)
Subroutine Nesting Level	4 Levels (including interrupt)			
Interrupt Level	1 Level			
Nunber of Instructions	35 Basic Instructions			
Execution Time of Basic Instruction	4 µS (1 Cycle Instruction), 8 µS (2 Cycle Instruction)			
Input/Output Level	TTL Compatible			
Power Supply	5V ± 10 %			
Power Dissipation	200mW (TYP.)	200mW (TYP.)	200mW (TYP.)	350mW (TYP.)
Operating Ambient Temp.	-10°C to 70°C			
Package	28 Pin Plastic DIP	42 Pin Plastic DIP	42 Pin Plastic DIP	64 Pin Ceramic DIP
Process	N-Channel E/D MOS			

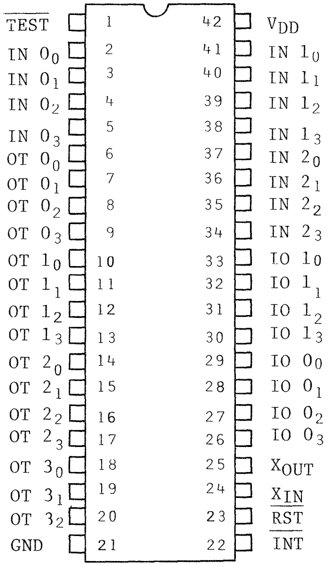
PIN CONNECTIONS

Top View

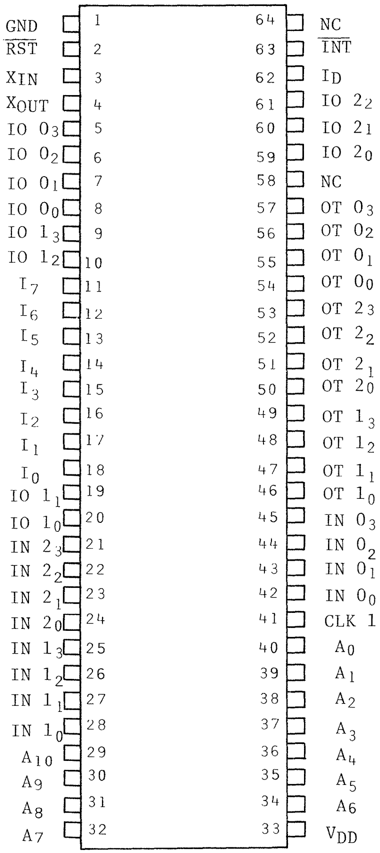
(TMP4310AP)



(TMP4315BP/TMP4320AP)



(TMP4300C)



PIN NAMES & PIN DESCRIPTION

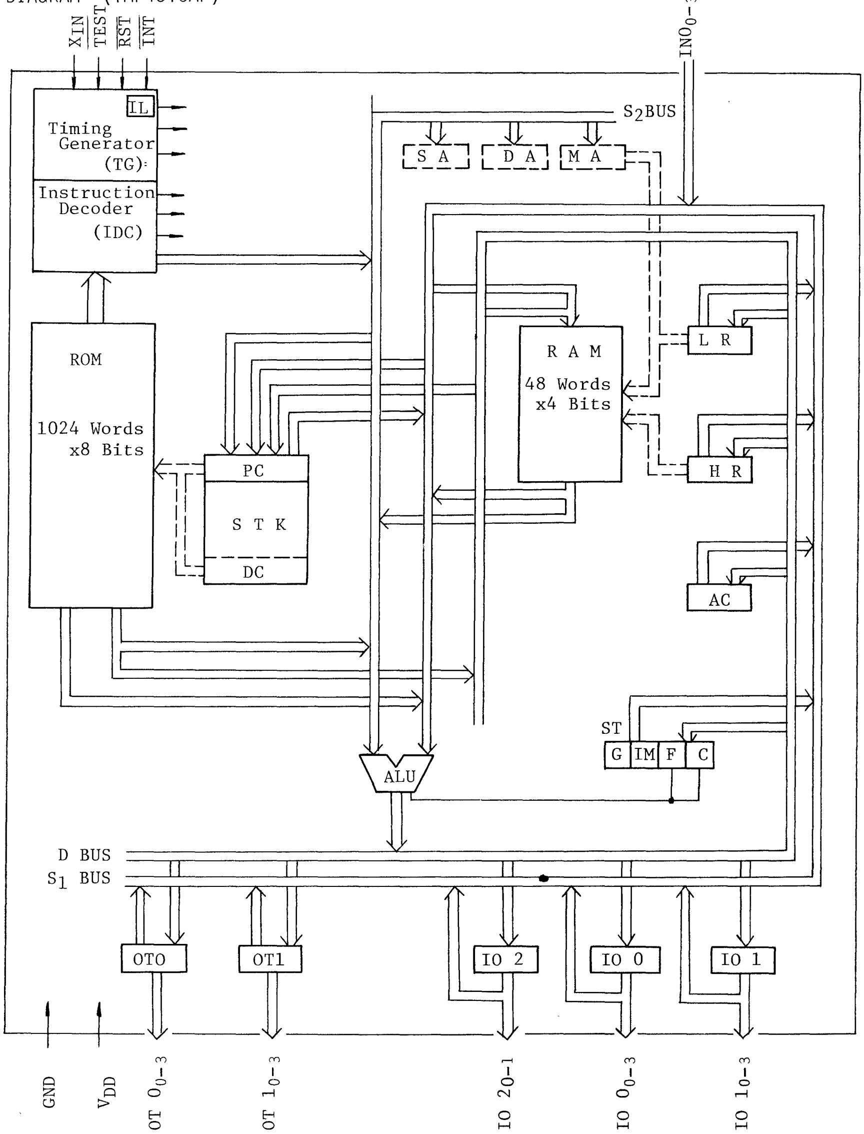
Pin Name	Input/ output	Function	4310	4315/ 4320	4300
IN0o-IN03	Input	4-bit Input Port IN0	○	○	○
IN1o-IN13	Input	4-bit Input Port IN1	—	○	○
IN2o-IN23	Input	4-bit Input Port IN2	—	○	○
OT0o-OT03	Output	4-bit Output Port OT0	○	○	○
OT1o-OT13	Output	4-bit Output Port OT1 Large sink current(IOL TYP=20mA,VOL=2V) is possible in TMP4310AP/20AP/00C.	○	○	○
OT2o-OT23	Output	4-bit Output Port OT2 Large sink current(IOL TYP=20mA;VOL=2V) is possible in TMP4320AP/00C.	—	○	○
OT3o-OT32	Output	3-bit Output Port OT3 IO2o-IO22 of TMP4300C are available in evaluation.	—	○	—
IO0o-IO03	Input/ Output	4-bit Input/Output Port IO0	○	○	○
IO1o-IO13	Input/ Output	4-bit Input/Output Port IO1	○	○	○
IO2o-IO22	Input/ Output	3-bit Input/Output Port IO2 2-bit Port (IO2o-IO21) in TMP4310AP	○	—	○
$\overline{\text{RST}}$	Input	Initialize Signal Input The initialize operation is performed by placing $\overline{\text{RST}}$ terminal at low level for more than four clock cycles.	○	○	○
$\overline{\text{INT}}$	Input	Interrupt Request Signal Input The interrut request is accepted by placing $\overline{\text{INT}}$ terminal at low level for more than our clock cycles. The repetetive interrupt should be requested, after keeping $\overline{\text{INT}}$ terminal at high level for two clock cycles or more.	○	○	○
$\overline{\text{TEST}}$ (Note 1)	Input	LSI Test Signal Input $\overline{\text{TEST}}$ should be always kept at high level (open or connect an oscillation resistance in TMP4310AP) except in LSI test mode.	○	○	—

XIN (Note 1)	Input	Basic Clock Terminal XIN is used as an external clock input pin, or a oscillator connection pin.	○	○	○
XOUT	Output	Basic Clock Terminal XOUT is used as a oscillator connection pin.	—	○	○
Ao-A_{10}	Output	ROM Address Output (MSB:A_{10},LSB:Ao)	—	—	○
Io-I_7	Input	ROM Data Input (MSB:I_7,LSB:Io)	—	—	○
CLK1	Output	Internal Clock Output	—	—	○
ID	Input	Interrupt Operation Inhobit Input ID is a dedicated terminal only for TDS400/43, and should be always kept at low level except in TDS400/ 43.	—	—	○
V_{DD}		+5V (Power Supply)	○	○	○
GND		OV (Power Supply)	○	○	○

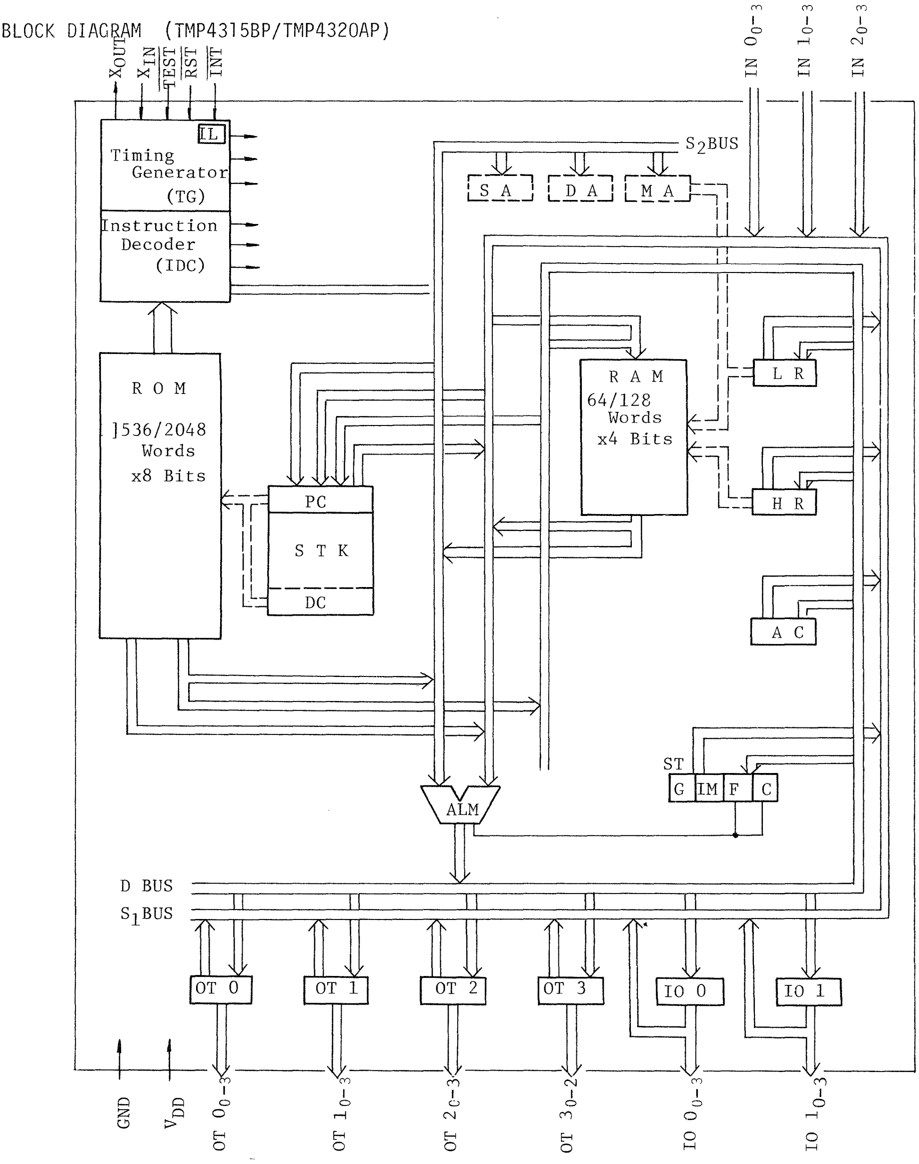
Note 1 The basic clock of TMP4310AP

- o Internal oscillation (with resistance externally installed between X_{IN} and $\overline{\text{TEST}}$) and external clock supply can be available.
- o $\overline{\text{TEST}}$ terminal should be kept open when the basic clock is supplied by an external oscillator circuit.

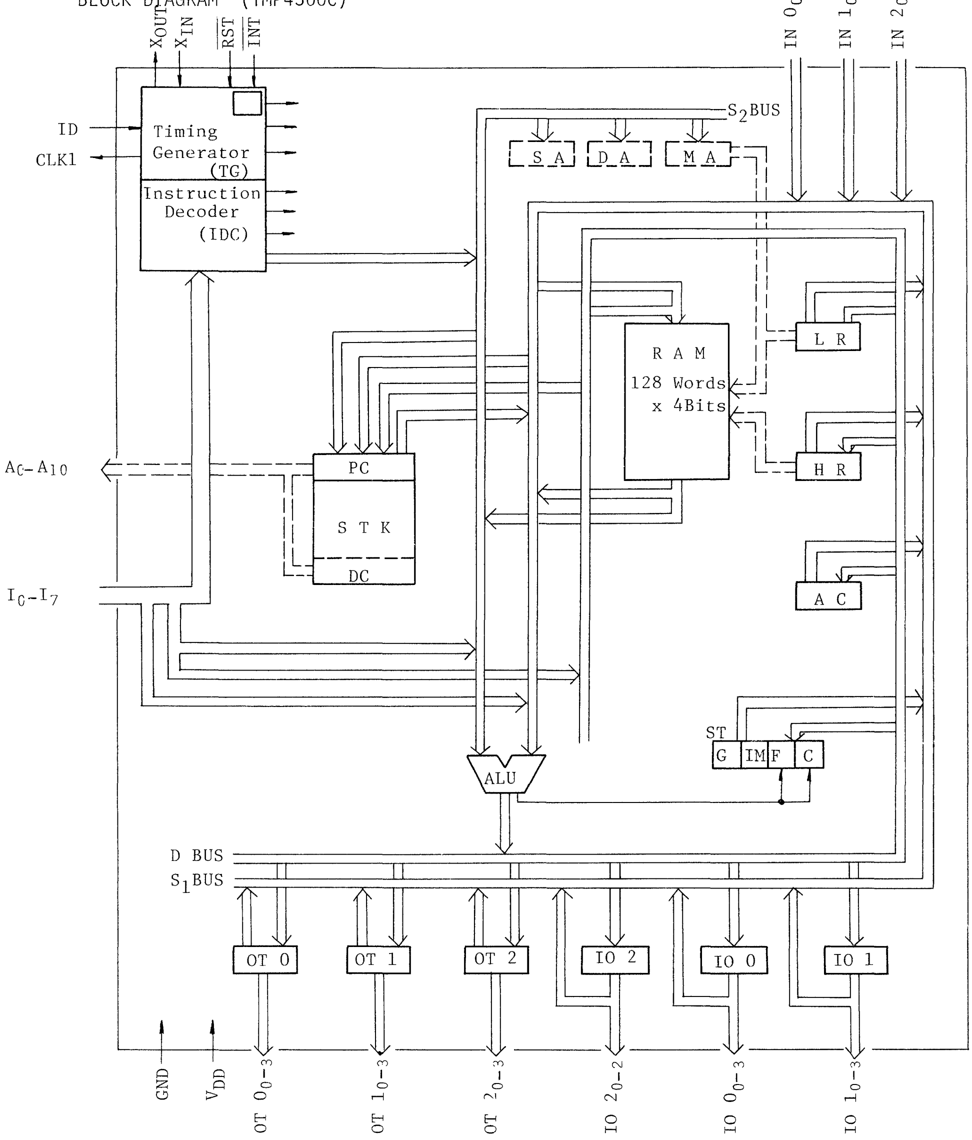
BLOCK DIAGRAM (TMP4310AP)



BLOCK DIAGRAM (TMP4315BP/TMP4320AP)



BLOCK DIAGRAM (TMP4300C)



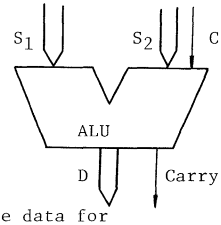
FUNCTIONAL DESCRIPTION

[Block Description]

For all registers and I/O ports, MSB is the High order bit and LSB is the Low order bit.

1. Arithmetic and Logical Unit (ALU)

The ALU is the central 4 bit parallel processing function of the TLCS-43. S_1 and S_2 are the two 4 bit input words and C is the carry input from some previous calculation. The ALU processed these and outputs one 4 bit result (D) and a carry bit.



2. Accumulator (AC)

The accumulator is a four bit register, which stores the data for arithmetic and logical operations. In addition the accumulator is also used to store the results of arithmetic and logical calculations.

3. Status Register (ST)

The status register is a four bit register which contains fields to represent the carry flag (C), branch condidtion flag (F), interrupt flag (IM) and general purpose flag (G).

	MSB 3	2	1	0 LSB
ST	G	IM	F	C

1) C

Bit 0 of the status register is called C flag and used to indicate Carry (or Borrow) during arithmetic operation with multiple number of digits.

2) F

Bit 1 of the status register is called F flag and set or reset according to the result of logical operation or arithmetic operation just executed. And this bit is referred to during execution of conditional branch instruction in a program.

3) IM

Bit 2 of the status register is the interrupt flag called IM flag which is set or reset by program.

IM flag being "1" indicates the interrupt enabled condition and IM flag is cleared to "0" as soon as an interrupt routine is initiated. This is also cleared to "0" by the initialize operation.

4) G

Bit 3 of the status register is called G flag and this one bit flag is used generally by programs.

4. L Register (LR)

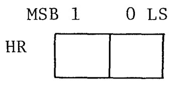
L Register (lower address register) is a four bit register which indicates the lower order 4 bits of RAM address and is used in conjunction with H Register for addressing RAM.

5. H Register (HR)

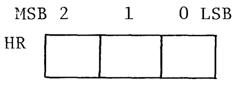
H Register (Higher address register) is a two bit or three bit register which indicates the higher order two or three bits of RAM address, and used in conjunction with L Register for addressing RAM.

When a program reads, the undefined higher order two bits (bit 3 and bit 2) or one bit (bit 3) are always processed to be zero.

(TMP4310AP/TMP4315BP)



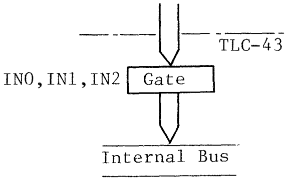
(TMP4320AP/TMP4300C)



6. Input Port (IN0, IN1, IN2)

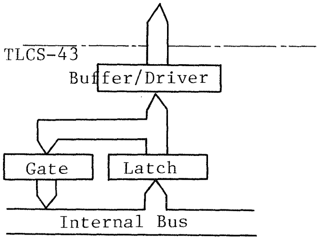
All of IN0, IN1 and IN2 ports are the dedicated input ports having four bit configuration and read the data sent from outside.

The input ports are non-latch type ports.



7. Output Port (OT0, OT1, OT2, OT3)

OT0,OT1 and OT2 have four bit configuration and OT3 has three bit configuration, all of which are the dedicated output ports. The content of each port is output to outside and retained until replaced with new data. And the content of a port can read by program. In this case the undefined bit 3 of OT3 port is always processed,to be zero. All the bits of all the output ports are set to "1" by the initialize operation.



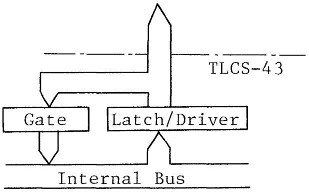
8. Input/Output Port (IO0, IO1, IO2)

IO0 and IO1 have four bit configuration and IO2 has two or three bit configuration, all of which are the input/output ports.

The content of each port is output to outside and retained until replaced with new data. It can also input data from outside. However, the output data must be set to "1" whenever the input operation is performed.

Whenver no data is input from outside, the output data to outside can be read by program. When the output data is read, undefined bit 3 and bit 2 of IO2 port are always processed to be zero.

All bits of the output data are set to "1" by the initialize operation.



9. Program Counter (PC)

The program counter is 11 bit counter which addresses the program stored in ROM (refer to (Note) in the ROM paragraph.)

While the normal instructions are executed, the program counter is incremented by word length of instruction just excuted. However, for branch instructions, subroutine call and interrupt operation, the counter is set to the values designated by the instructions. The counter is reset to "0" by the initialize operation.

10. Stack

The stack is a group of 4 words x 11 bits registers including the data counter.
The stack is used as the save area of the program counter during subroutine call and interrupt operation. If it is already occupied up to level 2, the data counter becomes to be the stack area of level 3.

11. Data Counter (DC)

The data counter is an 11 bit counter which addresses fixed data stored in ROM (refer to (Note) in the ROM paragraph.)

The content of the data counter can be set by program. The data counter is also used as the deepest stack level (level 3) and when nesting has been done up to level 2, if further nesting is performed, the content as the data counter is destroyed.

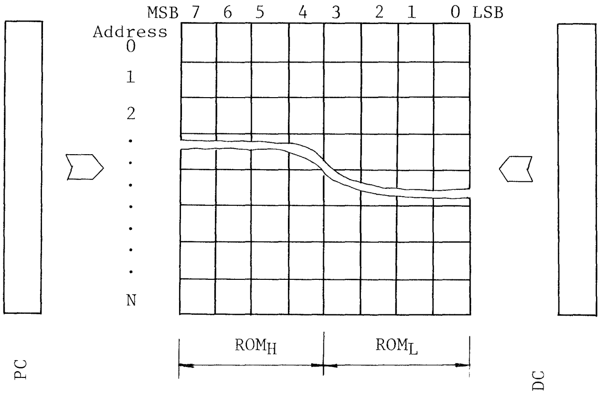
And when the data counter is being used as stack level 3, if setting operation is performed to the data counter, the content as stack level 3 is destroyed.

12. Read Only Memory (ROM)

The read only memory (ROM) can also store fixed data as well as programs which are required by users.

ROM has a maximum capacity of 2,048 words x 8 bits and is independently addressed by the program counter which addresses the storage area for programs and by the data counter which addresses the storage area for fixed data.

For storing programs the ROM is processed as 8 bit words but for fixed data, 8 bit word is divided into the higher order 4 bits and the lower order 4 bits, namely divided to two 4 bit words of ROM_H and ROM_L.



(TMP4310AP).... N=1,023 (TMP4315BP).... N=1,535 (TMP4320AP)... N=2,047

(Note) In the case of TMP4310AP, both of the program counter (PC) and the data counter (DC) are 11 bit counters, and if bit 10 is "1", any contents of ROM are not accessed.

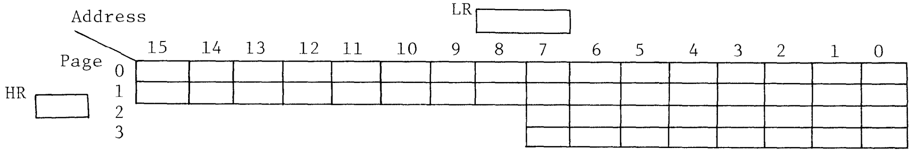
13. Read/Write Memory (RAM)

The read/write memory (RAM) can be used as the working area for data processing.

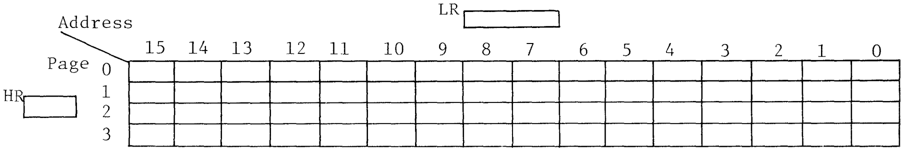
RAM has the maximum configuration of 128 words x 4 bits and is addressed by H Register which designates a page and L Register which designates an address in a page.

In addition to the above, another addressing method of RAM is to access an address in page 0 using the index instructions (M instructions) which will be explained later. This method is effective to save the contents of registers on the interrupt operation.
The configuration of each version is as follows.

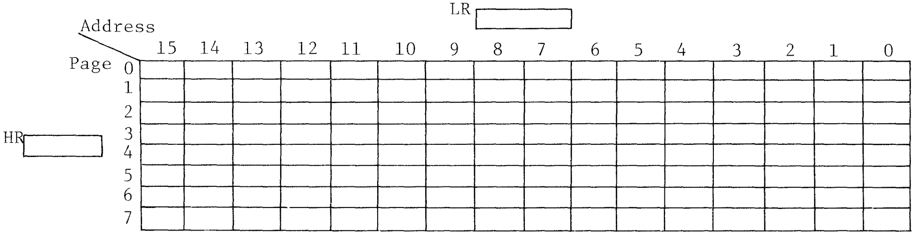
(TMP4310AP)



(TMP4315BP)



(TMP4320AP/TMP4300C)



14. Timing Generator (TG) and Instruction Decoder (IDC)

The timing generator produces a clock frequency that is dependant on the oscillator connected externally.
Based on this timing the instruction decoder reads and decodes the fields unique to each instruction.

Timing of interrupts, initialize requests etc. are also synchronized by the Timing Generator.

15. Registers Dedicated to Index Instructions (SA, DA and MA)

..... Refer to the paragraph of Index Instructions.

These are 4 bit registers used by the index instructions which are explained later,and there are three kinds, namely SA (Source Address Register), DA (Destination Address Register) and MA (Memory Address Register). Source register code, destination register code and RAM address which are activated by the index instructions are input to SA, DA and MA respectively, and these are temporarily retained until the following one operation instruction is completely executed.

The registers dedicated to the index instructions can not be used by program as additional data registers.

1. Features of TLCS-43 Machine Instructions

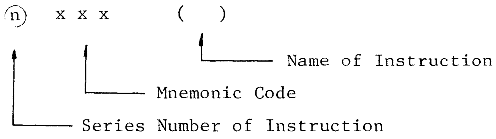
One of the features of the TLCS-43 Machine Instruction set is the existance of index instructions. In the case of processing instructions, usually the source and destination of data have been inherently defined. The index instructions modify the processing instructions to change the source of data to be processed or/and the destination of the processing results. Therefore, it is much simpler to write programs which require sequential operations through an area of memory. The extensive use of index instructions produces efficient programs in terms of the number of program steps.

Another great feature is that the machine instructions with Data Counter (DC) maintaining ROM address in addition to PC have the instructions which can read out the content of ROM directly, which allows a greater amount of fixed data to be efficiently read.

Furthermore, with the addition of four level subroutine nesting ability, the subroutine call instructions of 1 byte length are available. This is effective in reducing overall program size.

2. Format of Machine Instructions

The explanation of each instructions is described according to the following format.



<Symbol Instruction> Mnemonic Operation Code Operand

[Machine Instructions]

TLCS-43 series microcomputer is provided with 35 kinds of machine instructions. Unless otherwise mentioned the machine instructions are described as just instructions.

Among the machine instructions of TLCS-43, 30 instructions are of 1-byte length and 5 instructions are of 2-byte length. As regards the execution time of machine instructions, 28 instructions are of 1-machine cycle and 7 instructions are of 2-machine cycle.

Machine Instructions are classified by their functions as follows:

				Number of Instructions
TLCS-43 Machine Instructions	Index Instructions			4
	Processing Instructions	Data Processing Instructions	Data Transfer Instructions	7
			Logical Operating Instructions	10
			Bit Processing Instructions	3
			ROM Readout Instructions	3
		Subroutine Instructions		3
		Branch Instructions		5
			Total	35

< Machine Code >

No.1 Byte

MSB 7	6	5	4	3	2	1	LSB 0
x	x	x	x	x	x	x	x

No.2 Byte
(Address next to (+1) No.1 Byte)

MSB 7	6	5	4	3	2	1	LSB 0
x	x	x	x	x	x	x	x

(The correspondence of the operand of symbol instruction and the machine code is recorded, if necessary.)

< Function > The logical peration performed by this instruction is explained with symbols.

< Status Flag > (F): (C): The status after the execution of status flag is described.

(Dependent on the data when designated to store data in status register.)

< Execution Cycle > The number of machine cycles necessary for executing instructions is described.

< Function Explanation > The function of instructions is explained.

< Modifiable Index Instructions > In processing instructions, modifiable index instructions are described.

In this section, the symbols defined in Table A are used more concisely to express the functions of machine instructions. The storage elements, including input ports, related directly to the operation of instructions are collected in Table B.

Table A. Symbol and their Meanings for Instructions

Symbol	Meaning
(a)	The content of storage element "a".
M[(H.L)]	The content of RAM address designated by the contents of H register and L register.
TEMP	Temporary register
ZR	Virtual register of which content is "0".
$i_n\ i_n-1...i_o$	Data of n+1 bit
$\bar{b}$	Values inverted "1" to "0", "0" to "1" every bit of "b".
$a \leftarrow b$	"a" is replaced by the value of "b"/
a + 1	Value added 1 to "a".
a + b	Value added "b" to "a".
a - b	Value subtracted "b" from "a"/
$a \wedge b$	Value of logical product of "a" and "b" for every bit.
$a \vee b$	Value of logical sum of "a" and "b" for every bit.
a - b	Value of exclusive logical sum of "a" and "b" for every bit.

Ones <b>	4-bit data having 1's at bit positions only indicated by "b" and 0's at all other bit positions.
ROM_H[(DC)]	Higher order 4 bits in the content of ROM address indicated by the content of data counter DC.
ROM_L[(DC)]	Lower order 4 bits in the content of ROM address indicated by the content of data counter DC.
M[a]	Content of address "a" of RAM.
a<b>	Value of bit position "b" of "a".
DC_H	Higher order 3 bits of data counter
DC_M	Intermediate order 4 bits of data counter
DC_L	Lower order 4 bits of data counter
Carry	Carry resulted by operations (overflow)
Borrow	Borrow resulted by operations (underflow)
a = b	"a" equals to "b"
if a then b else c	If the condition of "a" is satisfied, "b" is performed; if not, "c" is performed
$	ROM address in which instructions are stored (No.1 byte address for 2-byte instruction)

Table B. Storage Elements Related Directly to Operation of Instructions

Name	Mnemonic	Function
Accumulator	AC	4-bit register
Carry flag	C	Carry flag of multiple digit operation
Branch flag	F	Condition flag exclusive for branch

Name	Mnemonic	Function
Interrupt mask flag	IM	Flag masking interrupt-peration In case of "1", interrupt is permitted.
General purpose flag	G	1 bit flag used by program
L register	LR	Register showing lower order 4 bits of RAM address
H register	HR	Rsgister showing higher order 3 bits of RAM address
Input port	IN0, IN1, IN2	Ports for input of external data.
Output port	OT0, OT1 OT2, OT3	Ports for output of data
I/O port	IO0, IO1 IO2	Ports for input or output of data
Data counter	DC	Counter to read out ROM contents as data
Program counter	PC	Counter to read out the instruction under program
Stack	STK	Stack storing return address from interrupt routine or subroutine (PC evacuation area)
Read/Write memory	RAM	Memory temporaly maintaing data
Read only memory	ROM	Memory maintaining program or fixed data

3. Index Instructions

The index instructions indicate source or/and destination of data. The data is processed by the instructions following the index instructions. The fixed data source and destination are designated for the instructions themselves, but if the instructions are modified by the index instructions, the data source and destination become those designated by the index instructions.

Since the index instructions cannot process effective data by themselves, they are invariably used in combination with the processing instructions. Therefore, the interrupt operation cannot be performed after execution of the index instructions, but can be done only after completion of the processing instructions.

One processing instruction can be modified by maximum three index instructions. When sources or destinations have been indicated in duplicate, the initially indicated one becomes effective.

① S (Designate Source register) : Source Register Index Instruction

< Symbol Instruction >

S r or = r

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
1	0	1	0	r_3	r_2	r_1	r_0

Operand $r = r_3\ r_2\ r_1\ r_0$

< Function > (SA) ← r

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 1 Machine Cycle

< Explanation of Function > Write address r of register/port, which becomes data source, in the source address register SA. The source register designated by this instruction is effective until the processing instruction is executed. However, if there are pural numbers of Instructions indicating the source register before the processing instruction, the initial index is effective.

② D (Designate Destination register) : Destination Register Index Instruction

< Symbol Instruction >

D r or = r

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
1	0	0	1	r_3	r_2	r_1	r_0

Operand $r = r_3\ r_2\ r_1\ r_0$

< Function > (DA) ← r

< Status Flag > (F) : No change

(C) : No change

< Executive Cycle > 1 Machine Cycle

< Explanation of Function > Write address r of register, which becomes a destination as a result of processing, in destination address register DA. The destination register designated by this instruction is effective until the processing instruction is executed. However, if there were plural numbers of instructions indicating the destination register before the processing instruction, the initial index becomes effective.

③ SD (Designate Source and Destination register) : Source & Destination Register Index Instruction

< Simbyl Instruction >

SD r or = r

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
1	0	0	0	r_3	r_2	r_1	r_0

Operand $r = r_3\ r_2\ r_1\ r_0$

< Function > (SA) ← r   (DA) ← r

< Status Flag > (F) : No change

(C) : No change

< Executive Cycle > 1 Machine Cycle

< Explanation of Function > Write address r of the register (same register) becoming data source and destination in source address register SA and destination address register DA. The register designated by this instruction is effective until the processing instruction is executed.
However, if the source register or the destination register has been indicated numbers times to one processing instruction in the same way as the instructions of S and D, the index initially made to the respective registers becomes effective.

④ M (Designate RAM address) : RAM Address Index Instruction

< Symbol Instruction >

M   r or = r

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
1	0	1	1	r_3	r_2	r_1	r_0

Operand $r = r_3\ r_2\ r_1\ r_0$

< Function > (MA) ← r

< Status Flag > (F) : No change
(C) : No change

< Executive Cycle > 1 Machine Cycle

< Explanation of Function > RAM address can be directly designated by this instruction without using H register and L register.
The RAM address "r" is written in memory address register MA; however, the RAM which can be designated by this instruction is limited to addressed 0 - 15 (16 words in "0" page). These RAM addresses are effective until the processing instruction is executed; that is, the RAM address of the processing instruction modified by this instruction is designated by the memory address register MA regardless of the content of H register and L register.
There are no changes in the contents of H register and L register.
If there are numbers of RAM address index instructions before the processing instruction the value initially indicated becomes effective.

4. Decision of Source and Destination by Index Instructions

The way of modifying the processing instruction by index instructions, or the decision of source and destination of the processing instruction, is regulated as follows :

(0) The elements not modified are unique source and destination for each instruction.

(1) Instructions related to registers and RAM (9 instructions)

In regard to the nine instructions, such as LDM, SWP, STR, ADD, ADC, SUB, CND, ORM, and EOR, their sources and destinations can be changed by the index instructions (S, D, and SD). When modified by M instruction, RAM data becomes the addresses (0 -15) designated by M instruction.

(2) Instructions related to register (13 Instructions)

In regard to the thirteen instructions, such as LDA, LDT, LDI, LLI, CMA, NGT, ADI, ALI, SSB, RSB, LFB, LRL, and LRH, their sources and destinations can be changed by the index instructions (S, D, and SD). When modified by M instruction, the sources and destinations by the S or D instruction become RAM addressed (0 - 15) designated by M instruction.

(3) As for SDC instruction, the source register can be designated to the intermediate order 4 bits of DC by S Instruction. In this case, the logical sum of the designated source register content and the immediate data is set to the intermediate order 4 bits of DC. When modified by M instructions the intermediate order 4 bits of DC become the contents of RAM addressed (0 - 15) designated by M instruction.

(4) Subroutine and branch instructions (8 instructions)

In regard to the eight instructions, such as CAL, CLS, RTN, BCF, BCB, JCS, JCC, and JMP, the index instructions cannot be modified. (If they are modified, their operations cannot be guaranteed.)

The following figure shows diagramatically the relationship between the above mentioned source and destination selection regulation and hardware.

The address of source register is stored in source address register SA by S instruction, and the source register is selected by the source selector according to this address.
The data from the selected source register is input into S1 of ALU. Either RAM or ROM is selected by the data selector as data input into S2 of ALU. The selection by this data selector is decided by the instructions.

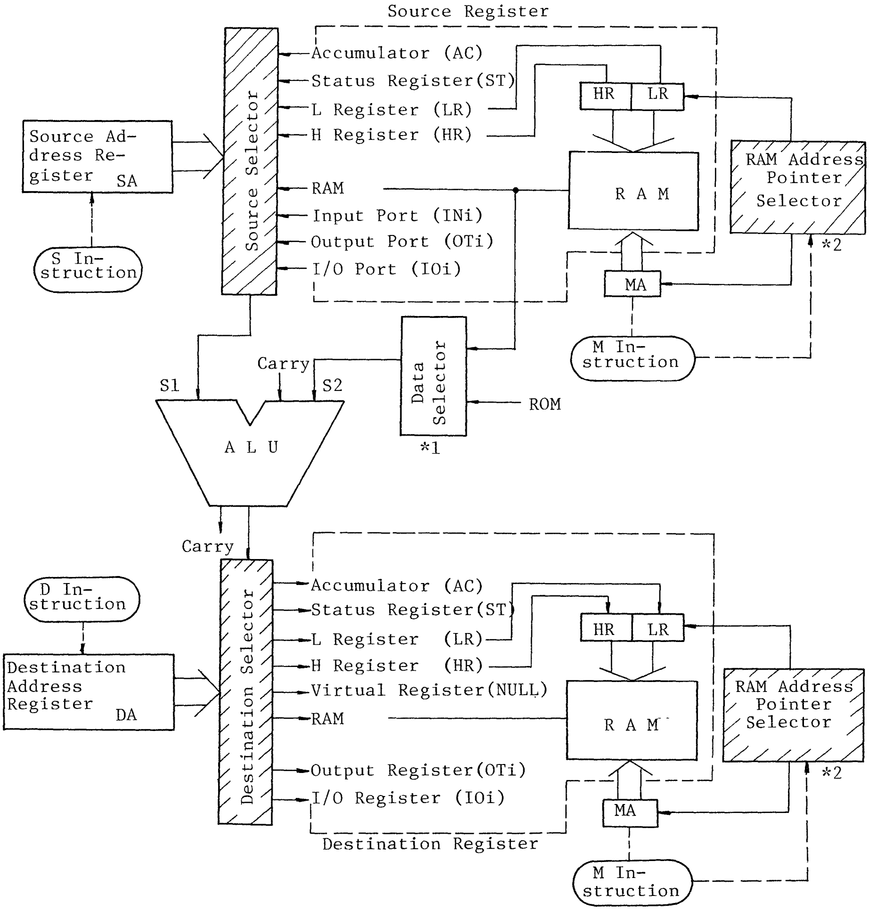
The address pointer of RAM has HR·LR and memory address register MA. Usually HR·LR is selected, but when modification is made by M instruction, MA is selected. The memory address is stored in MA by M instruction. Since either HR·LR or MA is used as address pointer, if RAM is used as source and destination register, the same address is selected.
The address of destination register is stored in the destination address register DA by D instruction. The destination selector selects destination register according to this address.
The process results output from ALU are stored in the destination register.

By using SD instruction, the same register address is stored into source address register SA and destination address register DA; therefore, the source register of S1 of ALU and the destination of output of ALU become the same.

As shown in the following figure the selector which changes the flow of processing data according to index instructions has the following three kinds of versions:

(1) Source selector (Selection of source register)
(2) Destination selector (Selection of destination register)
(3) RAM address pointer selector

In the case where no modification is made by index instructions, it may be thought that each selector makes the selection of the source or destination.



*1 Such a selector is determined according to the kind of instruction; in many cases RAM is selected, but in case of ROM readout instruction or immediate instruction, ROM is selected.

*2 Usually HR and LR is selected, but in case of processing instruction that modification is conducted by M instruction, MA is selected.

Registers can be designated by Index Instructions

Register code	Symbol	Name	TMP4310AP	TMP4315BP	TMP4320AP	TMP4300C
0	AC	Accumulator	○	○	○	○
I	ST	Status Register	○	○	○	○
2	LR	L Register	○	○	○	○
3	M[(H·L)]	RAM	○	○	○	○
4	INO	In. Port 0	○	○	○	○
5	NULL	NULL	○	○	○	○
6	IOO	I/O Port 0	○	○	○	○
7	IOl	I/O Port 1	○	○	○	○
8	OTl	Out. Port 1	○	○	○	○
9	OT	Out. Port 0	○	○	○	○
A	HR	H Register	○	○	○	○
B	-	-	Not used	Not used	Not used	Not used
C	OT3	Out. Port 3	—	○	○	—
	IO2	I/O Port 2	○	—	—	○
D	OT2	Out. Port 2	Not used	○	○	○
E	OT2	In. Port 1	Not used	○	○	○
F	IN2	In. Port 2	Not used	○	○	○

* Register code C designates IO2 for TMP4310AP and TMP4300C and OT3 for TMP4315BP and TMP4320AP. Therefore if it is required to perform evaluation of TMP4315BP and TMP4320AP using TMP4300C, IO2 is used as the matching port for OT3.

5. Data Processing Instructions

The data processing instructions are classified in four types, data transfer instruction, logical operation instruction, bit processing instruction, and ROM readout instruction.

5.1 Data transfer instruction

Data is handled in 4-bit units. The use of these instructions provide the setting of immediate data as well as the data transfer between two registers or between a register and RAM.

⑤ LDM (Load from Memory) : Load from Memory Instruction

< Symbol Instruction >

LDM

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	0	1	1	0

< Function > (AC) ← M[(H·L)]

< Status Flag > (F) : No change
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of RAM address designated by H register and L register is loaded into accumulator.

< Modifiable Index Instruction > D, M

⑥ SWP (Swap) : SWAP Instruction

< Symbol Instruction >

SWP

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	0	1	0	1

< Function > (TEMP) ← (AC)
(AC) ← M [(H·L)]
M[(H.L)] ← (TEMP)

< Status Flag > (F) : No change
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > This instruction exhanges the content of RAM address designated by H register and L register content of the accumulator.

< Modifiable Index Instruction > SD, M

⑦ LDA (Load from Accumulator) : Load from Accumulator Instruction

< Symbol Instruction >

LDA

< Machine Code >

MSB							LSB
7	6	5	4	3	2	1	0
0	0	1	0	1	0	1	1

< Function > (AC) ← (AC)

< Status Flag > (F) : No change
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of accumulator is loaded into the accumulator. If used independetly, this instruction becomes a no-operation instruction.

< Modifiable Index Instruction > S, D, M

⑧ STR (Store) : Store Instruction

< Symbol Instruction >

STR

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	0	1	0	0

< Function > (AC) ←M[(H·L)]

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of accumulator is stored in the RAM address designated by H register and L register.

< Modifiable Index Instruction > S, M

(9) LDT (Load and Test) : Load & Test Instruction

< Symbol Instruction >

LDT

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	1	0	1	0

< Function > (AC) ← (AC)

< Status Flag > If (AC) = 0 then (F) ← 1, else

(F) ← 0

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of accumulator is loaded into the accumulator. If the data is zero, F is set to "1", but if not, F is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

⑩ LDI (Load Immediate data) : Load Immediate Data Instruction

< Symbol Instruction >

LDI i $(0 \leqq i \leqq 15)$

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	1	1	0	i_3	i_2	i_1	i_0

Operand $i = i_3\ i_2\ i_1\ i_0$

< Function > (AC) ← i

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > Immediate Data i is loaded into accumulator.

< Modifiable Index Instruction > D, M

⑪ LLI (Load Immediate data to L register) : Load Immediate Data to L Register Instruction

< Symbol Instruction >

LLI i $(0 \leqq i \leqq 15)$

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	1	0	0	i_3	i_2	i_1	i_0

Operand $i = i_3\ i_2\ i_1\ i_0$

< Function > (LR) ← i

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > Immediate Data i is loaded into L Register

< Modifiable Index Instruction > D, M

5.2 Logical operation instruction

The instructions of CMA, CND, ORM and EOR are used for logical operation of every bit of 4-bit data. The others are mainly used for arithmetic operation. For 2-operand instruction one data source is a register and another is RAM.

For the nine instructions except CMA, branch condition flag F is set, and for ADC instruction, carry flag C is set.

12 CMA (Complement Accumulator) : Complement Accumulator Instruction

< Symbol Instruction >

CMA

< Machine Code >

MSB							LSB
7	6	5	4	3	2	1	0
0	0	1	1	1	0	1	1

< Function > $(AC) \leftarrow \overline{(AC)}$

< Status Flag > (F) : No change
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of accumulator is loaded into the accumulator after inverting "0" to "1" and "1" to "0" every bit.

< Modifiable Index Instruction > S, D, SD, M

13 NGT (Negate) : Negate (2's complement) Instruction

< Symbol Instruction >

NGT

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	1	1	0	0	1

< Function > $(AC) \leftarrow \overline{(AC)} + 1$

< Status Flag > if Carry then (F)←1, else (F)←0
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > 2's complement of the content of accumulator is loaded into the accumulator. If the original data is 0 ((AC)=0), 2's complement is 0. In this case only, F flag is set to "1", but in the other cases, F flag is cleared to "0". In this case F flag is used in both meanings of carry and zero decision.

< Modifiable Index Instruction > S, D, SD, M

(14) ADD (Add) : Add Instruction

< Symbol Instruction >

ADD

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	1	1	0	1

< Function > (AC) ← (AC) + M [(H·L)]

< Status Flag > if carry then (F)←1, else (F)←0
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of RAM address designated by the contents of H register and L register is added to the content of the accumulator, and the result is loaded into the accumulator. If the resultant carry is "1" , F flag is set to "1", and if it is "0", F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

⑮ ADC (Add with Carry) : Add with Carry Instruction

< Symbol Instruction >

ADC

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	1	0	1	0	1

< Function > (AC) ← (AC) + M [(H·L)] + (C)

< Status Flag > if carry then (F)←1, (C)←1, else (F)←0
(C)←0

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of RAM address designated by H register and L register and the content of C flag are added to the content of accumulator, and the result is loaded into the accumulator. The conditions of Carry cause F flag and C flag to be set/reset.

< Modifiable Index Instruction > S, D, SD, M

⑯ SUB (Subtract) : Subtract Instruction

< Symbol Instruction >

SUB

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	1	1	1	0	1

< Function > $(AC) \leftarrow M[(H\cdot L)] - (AC)$

< Status Flag > if borrow then (F) ← 0, else (F) ← 1
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The content of accumulator is subtracted from the content of RAM address designated by H register and L register, and the result is loaded into the accumulator. The conditions of underflow cause F flag to be set/reset.

< Modifiable Index Instruction > S, D, SD, M

⑰ CND (Complement & AND) : Complement and AND Instruction

< Symbol Instruction >

CND

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	1	1	1	0	0

< Function > $(AC) \leftarrow (AC) \wedge \overline{M[(H\cdot L)]}$

< Status Flag > if (AC) = 0 then (F) ← 1, else (F) ← 0
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The logical product of every bit of the content of accumulators and the 1's complement of the content of RAM address designated by the content of H register and L register is loaded into the accumulator. If the result is "0", F flag is set to "1"; in the other cases, F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

⑱ ORM (OR) : OR Instruction

< Symbol Instruction >

ORM

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	1	1	1	0

< Function > $(AC) \leftarrow (AC) \vee M[(H \cdot L)]$

< Status Flag > if $(AC)=0$ then $(F) \leftarrow 1$, else $(F) \leftarrow 0$

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The logical sum of every bit of the content of accumulator and the content of RAM address designated by H register and L register is loaded into the accumulator. If the result is "0", F flag is set to "1"; in the other cases, F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

⑲ EOR (Exclusive OR) : Exclusive OR Instruction

< Symbol Instruction >

EOR

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	1	1	1	1

< Function > $(AC) \leftarrow (AC) \veebar M[(H \cdot L)]$

< Status Flag > if $(AC)=0$ then $(F) \leftarrow 1$, else $(F) \leftarrow 0$
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The exclusive logical sum of every bit of the content of accumulator and the content of RAM address designated by H register and L register is loaded into the accumulator.
If the result is "0", F flag is set to "1"; in the other cases, F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

⑳ ADI (Add Immediate data) : Add Immediate Data Instruction

< Symbol Instruction >

ADI i ($0 \leqq i \leqq 15$)

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	1	1	1	i_3	i_2	i_1	i_0

Operand $i = i_3\ i_2\ i_1\ i_0$

< Function > $(AC) \leftarrow (AC) + i$

< Status Flag > if Carry then (F) ← 1, else (F) ← 0,
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > Immediate data "i" is added to the content of accumulator and the result is loaded into the accumulation.
If the resultant carry is generated, F flag is set to "1"; in the other case, F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

(21) ALI (Add Immediate data to L register) :
Add Immediate Data to L Register Instruction

Symbol Instruction

ALI i $(0 \leqq i \leqq 15)$

< Machine Code >

MSB							LSB
7	6	5	4	3	2	1	0
0	1	0	1	i_3	i_2	i_1	i_0

Operand $i = i_3\ i_2\ i_1\ i_0$

< Function > (LR) ← (LR) + i

< Status Flag > if Carry then (F) ← 1, else (F) ← 0
(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > Immediate date "i" is added to the content of L register, and the result is loaded into L register. If resultant carry is generated, F flag is set to "1"; in the other cases, F flag is cleared to "0".

< Modifiable Index Instruction > S, D, SD, M

5.3 Bit manipulation instruction

Ordinary data is manipulated as 4-bit units, however the use of these instructions enables the data to be manipulated bit by bit. These instructions are mainly applied to status register ST, but their functions can be extended to aribitrary register, output port and RAM by a combination of index instructions.

(22) SSB (Set Status Bit) : Set Status Bit Instruction

< Symbol Instruction >

SSB b $(3 \geqq b \geqq 0)$

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	0	1	0	1	b_1	b_0

Operand $b = b_1\ b_0$

< Function > $(ST) \leftarrow (ST) \vee \text{Ones} < b >$ $(ST < b > \leftarrow 1)$

< Status Flag > There may be flags (bits) changed by the execution itself of this instruction, but there are no flags changed by the result of the execution.

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The bit field of the status register defined by low order 2 bits "b" of the instruction is set to a "1".

< Modifiable Index Instruction > SD, M

(23) RSB (Reset Status Bit) : Reset Status Bit Instruction

< Symbol Instruction >

SRB b $(3 \geqq b \geqq 0)$

< Machine Code >

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	1	1	1	b_1	b_0

Operand $b = b_1 b_0$

< Function > $(ST) \leftarrow (ST) \wedge \overline{\text{Ones}} \langle b \rangle$ $(ST \langle b \rangle \leftarrow 0)$

< Status Flag > There may be flags (bits) changed by the execution itself of this instruction, but there are no flags changed by the result of the execution.

< Execution Cycle > 1 Machine cycle

< Explanation of Function > The bit field of the status register defined by the low order 2 bits "b" of the instruction is set to a "0".

< Modifiable Index Instruction > SD, M

㉔ LFB (Load complemented status Bit to F)

Load and Complement Status Bit to F Instruction

< Symbol Instruction >

LFB b $(3 \geq b \geq 0)$

< Machine Code >

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	1	1	0	b_1	b_0

Operand $b = b_1 b_0$

< Function > $(F) \leftarrow \overline{ST} \langle b \rangle$

< Status Flag > (F) : According to condition of ST <b>

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > If the bit content of status register designated by lower order 2 bits of the instruction is 1, it is inverted to 0, and it is 0, it is inverted to 1, and then it is loaded into F bit.

< Modifiable Index Instruction > S, M

5.4 ROM readout instruction

SDC is the instruction by which the address on ROM of fixed data is set into data counter DC. LRL and LRH are the instructions by which ROM data are readout in the 4-bit unit.

(25) SDC (Set Data Counter) : Set Data Counter Instruction

< Symbol Instruction >

SDC a $16 \leq a \leq 2032$, where "a" is
(integral multiples of 16.)
i.e. $a = 16n, 1 \leq n \leq 127$

< Machine Code >

1st Byte

MSB							LSB
7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0

2nd Byte

7	6	5	4	3	2	1	0
a_7	a_6	a_5	a_4	0	a_{10}	a_9	a_8

• Operand $a = a_{10}\ a_9\ a_8\ a_7\ a_6\ a_5\ a_4\ 0000$

< Function >

$$\left.\begin{array}{lll} (DC_H) & \leftarrow & a_{10} a_9 a_8 \\ (DC_M) & \leftarrow & a_7 a_6 a_5 a_4 \vee (ZR) \\ (DCL) & \leftarrow & M[(H \cdot L)] \end{array}\right\} \quad \text{i.e.} \quad (DC) \leftarrow a + M[(H \cdot L)]$$

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 2 Machine cycle

< Explanation of Function >

This instruction sets ROM address into the data counter. The higher order 3 bits of ROM address to be set are immediate values, and the intermediate 4 bits are the logical sum of intermediate value and virtual source register ZR, of which content is "0", and the lower order 4 bits are the content of RAM address indicated by H register and L register. When source register is designated by the index instructions, ZR OR-ed to the intermediate 4 bits becomes the content of its source register, being effectively activated.
The content of data counter after the execution of instruction is as follows, if it is not modified by index instructions:

Data Counter

MSB 10	9	8	7	6	5	4	3	2	1	LSB 0
a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	M[(H·L)]			

< Modifiable Index Instruction >

S, M

(26) LRL (Load ROM Lower data) : Load ROM Lower Data Instruction

< Symbol Instruction >

LRL

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 0
0	0	1	0	0	0	1	0

< Function > $(AC) \leftarrow ROM_L\ [(DC)]$

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 2 Machine cycle

< Explanation of Function > This instruction loads the accumulator with the lower order 4 bits of the content of ROM address designated by the content of data counter.

< Modifiable Index Instruction > D, M

(27) RLH (Load ROM Higher data) : Load ROM Higher Data Instruction

< Symbol Instruction >

LRH

< Machine Code >

MSB LSB

7	6	5	4	3	2	1	0
0	0	1	1	0	0	1	0

< Function > $(AC) \leftarrow ROM_H\ [(DC)]$

$(DC) \leftarrow (DC) + 1$

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 2 Machine cycle

< Explanation of Function > This instruction loads the accumulator with the higher order 4 bits of the content of ROM address designated by the content of data counter; after-ward, the content of data counter is increased by one.

< Modifiable Index Instruction > D, M

6 Subroutine Instructions

When subroutine call and interrupt operation are performed, the value of PC should be stored in a stack in order to keep return address. Four levels of PC stacks are provided for the interrup operation and the subroutine call, enabling four-level nesting to be performed. The 4th stack serves for data counter DC; therefore, when data counter is in use, the operation of nesting should be restricted to three levels. Otherwise, the content of data counter is destroyed, resulting in the deletion of the return address. When the four-level nesting is in operation, no interrupt operation is performed. Also, no storage of PC by call subroutine instruction is performed.

㉘ CAL (Call subroutine) : Call Instruction (2 Bytes)

< Symbol Instruction >

CAL a $(0 \leq a \leq 2047)$

< Machine Code >

MSB LSB

	7	6	5	4	3	2	1	0
1st Byte	1	1	0	0	1	a_{10}	a_9	a_8
2nd Byte	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operand $a = a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0$

< Function >

(PC) → (STK) push down

(PC) ← a

< Status Flag >

(F) : No change

(C) : No change

< Execution Cycle >

2 Machine cycle

< Explanation of Function >

This is a subroutine call instruction which directly indicates the entry address of subroutine.

The return address is put into a stack, and the designated value (the entry address of subroutine) a=a10a9a8a7a6a5a4a3a2a1a0 is set to PC.

The stack of return address is a push down stack which can be stacked up to 4 levels. Since the fourth level is a data counter, precautions for use should be taken.

< Modifiable Index Instruction >

No instruction

㉙ CLS (Call Subroutine by single byte) : Call by Single Byte Instruction

< Symbol Instruction >

CLS i (0 ≦ i ≦ 15)

< Machine Code >

MSB … LSB

7	6	5	4	3	2	1	0
0	0	0	0	i_3	i_2	i_1	i_0

Operand $i=i_3i_2i_1i_0$

< Function >

(PC) ⟶ (STK) push down

(PC) ⟵ i x 8 + 4

< Status Flag >

(F) : No change

(C) : No change

< Execution Cycle >

1 Machine cycle

< Explanation of Function >

This instruction stores a return address into a stack, calculates (=000 $i_3i_2i_1i_0$100) i x 8 + 4 as subroutine entry address from the entry address No. i designated to the operand, and sets the result to PC.

The stack of return address is of push-down stack which can stack as high as 4 levels.

Since the fourth level is a data counter, precautions for use should be taken.

< Modifiable Index Instruction > No instruction

㉚ RTN (Return) : Return Instruction

< Symbol Instruction >

RTN

< Machine Code >

MSB LSB

7	6	5	4	3	2	1	0
0	0	1	0	0	1	1	1

< Function > (PC) ← (STK) last in data

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 1 Machine cycle

< Explanation of Function > This instruction causes a return from an interrupt routine and a subroutine to the main program. The newest return address stored in the stack is loaded into the program counter.

< Modifiable Index Instruction > No instruction

7 Branch Instruction

Among the branch instructions, there is an on-condition-set/cleared branch instruction of which condition is decided whether or not branch operation is performed depending upon the value of F flag in the program status register; therefore, it should be considered how the value of F flag is changed by the data processing instruction just before the use of the conditional branch instruction. BRC is limited in branch range, but this instruction can reduce the number of bytes of ROM because of a single byte instruction.

The unconditional jump instruction shifts unconditionally the execution flow of the instruction to the address indicated by the address field of this instruction.

㉛ BRC (ranch on Condition set) : Branch on (F) = 1 Forward Instruction

㉜ BRC (Branch on Condition set) : Branch on (F) = 1 Backward Instruction

< Symbol Instruction >

BRC a $0 \leq a \leq 2047$,
($-16 \leqq a - \$ \leqq 15$)

< Machine Code >

MSB LSB

7	6	5	4	3	2	1	0
1	1	$\overline{S}$	1	i_3	i_2	i_1	i_0

$a-S=i_3i_2i_1i_0$

$a-\$ \geqq 0$, $\overline{S}=1$ ... Branch on(F)=1 Forward Instruction

$a-\$ \leqq 0$, $\overline{S}=0$ ... Branch on(F)=1 Backward Instruction

< Function >

if (F)=1 then (PC)←a, else No operation ((PC)←(PC) + 1)

< Status Flag >

(F) : No change
(C) : No change

< Execution Cycle >

1 Machine cycle

< Explanation of Function >

If F flag of status flag is set to "1", the value of program counter is changed to the absolute address "a" defined by the operand field of the instruction. In the other cases, the program counter advances by one and execution of the next instruction will be initiated without any other operations.

The range of absolute address "a" is -16 - 15 against ROM address "$" stored by this instruction itself.

< Modifiable Index Instruction > No instruction

(33) JCS (Jump on Condition Set) : Jump on(F)= 1 Instruction

< Symbol Instruction >

JCS a ($0 \leqq a \leqq 2047$)

< Machine Code >

MSB LSB

1st Byte

7	6	5	4	3	2	1	0
1	1	1	0	0	a_{10}	a_9	a_8

2nd Byte

7	6	5	4	3	2	1	0
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operand $a=a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0$

< Function > if (F)=1 then (PC)←a,else No Operation

< Status Flag > (F) : No change

(C) : No change

< Execution Cycle > 2 Machine cycle

< Explanation of Function > If F flag is set to "1", this instruction causes to branch to the address "a" indicated by the lower order 11 bits of this instruction. In the other cases, execution of the next instruction with be initiated without any other operations.

< Modifiable Index Instruction > No instruction

(34) JCC (Jump on Condition Cleared) : Jump on (F)=0 Instruction

< Symbol Instruction >

JCC a ($0 \leqq a \leqq 2047$)

< Machine Code >

1st Byte

	MSB 7	6	5	4	3	2	1	LSB 0
1st Byte	1	1	1	0	1	a_{10}	a_9	a_8

2nd Byte

	7	6	5	4	3	2	1	0
2nd Byte	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operand $a=a_{10}\ a_9\ a_8\ a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$

< Function > if (F)=0 then (PC)←a, else No Operation

< Status Flag > (F) : No change
(C) : No change

< Execution Cycle > 2 Machine cycle

< Explanation of Function > If F flag is cleared to "0", this instruction causes to branch to the address "a" indicated by the lower order 11 bits of the instruction. In the other cases, execution of the next instruction will be initiated without any other operations.

< Modifiable Index Instruction > No instruction

(35) JMP (Jump) : Unconditional Jump Instruction

< Symbol Instruction >

JMP a $(0 \leqq a \leqq 2047)$

< Machine Code >

	MSB 7	6	5	4	3	2	1	LSB 0
1st Byte	1	1	0	0	0	a_{10}	a_9	a_8

	7	6	5	4	3	2	1	0
2nd Byte	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Operand $a=a_{10}\ a_9\ a_8\ a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$

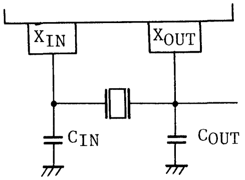
< Function >	$(PC) \leftarrow a$
< Status Flag >	(F) : No change (C) : No change
< Executio Cycle >	2 Machine cycle
< Explanation of Function >	This instruciton causes an unconditional branch to the address "a" indicated by the lower order 11 bits of the instruction.
< Modifiable Index Instruction >	No instruction

[OPERATION DESCRIPTION]

1. Basic Clock

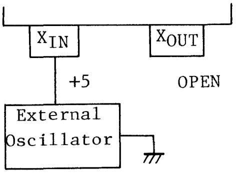
The basic clock generates the basic timing sequences required for the operations of TLCS-43. There are the following methods of options generating the basic clock.

(1) Direct Connection of the Oscillator



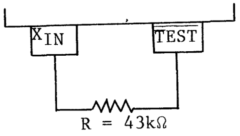
When a crystal oscillator, or a ceramic oscillator, or a IFT, is connected as shown at the left, the frequency of TLCS-43 basic clock is equal to that of the oscillator. (Except TMP4310AP)

(2) Supply of External Clock



The basic clock of TLCS-43 can also be supplied by an external oscillator circuit as shown at the left.
The external clock input should be sinusoidal or square wave vibrating with levels between 0 volts and 5 volts.

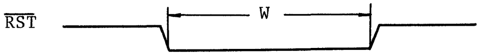
(3) Internal Oscillation



The basic clock of TMP4310AP can be obtained by connecting the resistance as shown in the left figure.
(fosc = 250kHz - 450 kHz at R=43kΩ)
(Except TMP4315BP, TMP4320AP, TMP4300C)

2. Initialize Operation

The initialize operation of TLCS-43 is performed by placing $\overline{RST}$ terminal at the low level as shown below. The minimum time period of four basic clock cycle is required for the low level pulse width.



$0 \leq W < 1.5$ cycles	1.5 cycles $\leq W < 4$ cycles	4 cycles $\leq W$
The initialize operation is not performed.	Whether or not the initialize operation is not definite.	The initialize operation is performed.

One cycle is equal to one cycle of the basic clock.

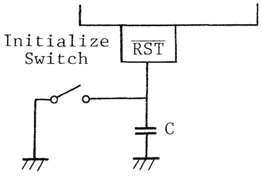
The initialize operation performs the following functions.

Block	Symbol	Initialize Function
Program counter	PC	Cleared to "0"
Interrupt latch	IL	Cleared
Interrupt flag	IM	Cleared to "0" disabling interrupt.
Output port	OTO	All bits are set to "1".
	T01	All bits are set to "1".
	OT2	All bits are set to "1".
	OT3	All bits are set to "1".
Input/Output port	I00	All output bits are set to "1".
	I01	All output bits are set to "1".
	I02	All output bits are set to "1".
Stack	STK	Made empty.

While $\overline{\text{RST}}$ terminal is held at the low level, only the above functions are taken place and other operations, such as execution of program are not performed.

When $\overline{\text{RST}}$ terminal is returned to the high level, the program starting from address 0 is executed.

(Note) C, F and G of the status register are not reset.
Therefore, these must be taken care of by the program.
And any index instructions executed prior to the initialization are ignored.



If a capacitor is connected to $\overline{\text{RST}}$ terminal as shown at the left, the initialize operation is automatically performed when the power supply is turned on. And if it is required to perform the initialize operation manually, a switch should be connected for this purpose.

3. Interrupt Operation

(1) Interrupt Operation

TLCS-43 has the function which allows the interrupt operation to be triggered externally. The interrupt operation is performed by holding $\overline{\text{INT}}$ terminal at the low level for four basic clock cycles or more. However, several conditions must be satisfied to initiate the interrupt operation. Such conditions are as follows.

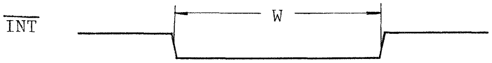
1. Interrupt flag IM has been set and one or more instructions have veen executed after the flag was set.
2. Four levels of subroutine nesting have not been performed.

The interrupt request signal from outside is retained in interrupt latch IL located inside and once an interrupt is accepted, IL and inteerupt flag IM are cleared.

If it is desired to trigger an inteerupt again, $\overline{\text{INT}}$ terminal must be returned to the high level (for two basic clock cycles or more) and must be placed at the low level again. Repetitive interrupts are not accepted keeping $\overline{\text{INT}}$ terminal at the low level.

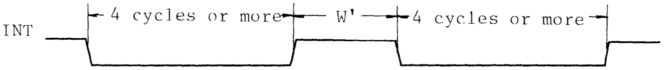
Setting and repetitive setting of IL by $\overline{\text{INT}}$ signal are performed in the following timings.

(Setting Timing of IL).



$0 \leq W < 1$ cycle	1 cycle $\leq W < 4$ cycles	4 cycles $\leq W$
IL is not set.	Whether or not IL is set is not definite.	IL is set.

(Repetitive Setting of IL)



W' = 0	0<W'<2 cycles	2 cycles≦W'
IL is not set again.	Whether or not IL is set again is not definite.	IL is set again.

Where one cycle is equal to one cycle of the basic clock.

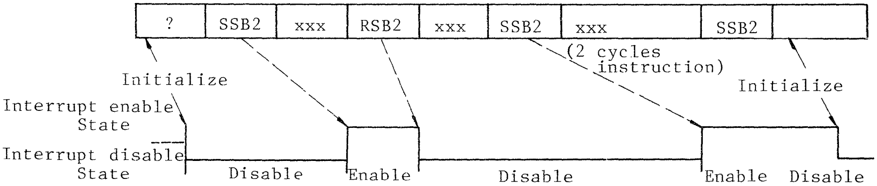
When an interrupt is accepted, the content of program counter PC is pushed down to stack STK and the entry address (address 2) of the interrupt service routine is set in the program counter.

The accumulator, the status register, L register and H register which are used in the interrupt program must be saved into RAM area in the service routine.

When returning to execution of the main routine after completing the interrupt routine the saved registers are returned and the interrupt flag which has been reset is set to "1". Then, execution of Return instruction causes to return to execution of the main routine.

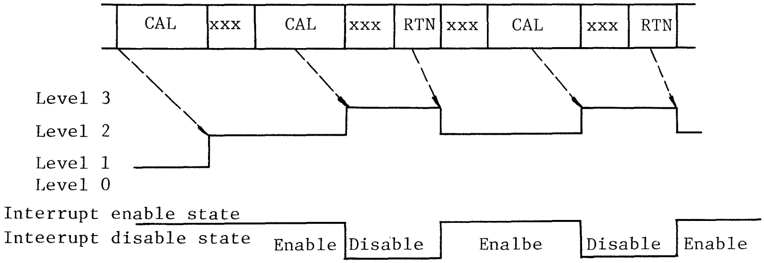
(2) Timing of Interrupt Enable/Disable

① Interrupt Flag Set/Reset



The interrupt is enabled, when one instruction has been executed after SSB2 instruction was executed, and immediately disabled by execution of RSB2 instruction.

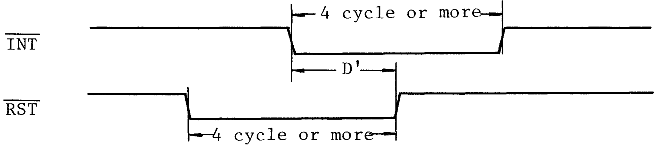
② When stack is used up to the deepest level (level 3)
(Assumption: IM flag is set "1" constantly)



Even if IL and IM have been set, if the stack is occupied fully up to the deepest level (level 3), any interrupt are not accepted and must wait until level 3 becomes available.

(3) Relationship between $\overline{\text{INT}}$ Signal and $\overline{\text{RST}}$ Signal

When the interrupt request signal and the initialize signal occur simultaneously, the operation shown in the following examples takes place.



D' <1 cycle .... The interrupt request is accepted after the initialize operation.

1 cycle ≦D' <5 cycles ... Acceptance of the interrupt request is not definite after the initialize operation.

D'≧ 5 cycles .... Only the initialize operation performed and the interrupt request is not accepted.

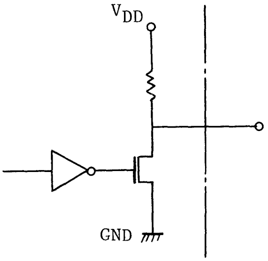
Where one cycle is equal to one cycle of the basic clock.

4. Types of Output Buffers

For the output buffers of output ports and input/output ports, the following types (A) or (B) can be specified by designating a mask option. Since these types are selected by the same mask as the user program mask,these must be specified based on the mask ROM data type format. (Refer to the mask ROM data tape format.)

When data is input from the input/output ports, the output data must have been set to "1" in advance for both (A) and (B) types.

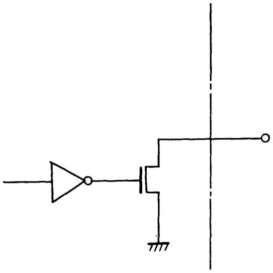
For the electrical characteristics, refer to the paragraphs of ABSOLUTE MAXIMUM RATINGS and DC CHARACTERISTICS. Since the pull-up resistors are provided by MOS transistors, the characteristics are somewhat different from normal resistors.



$V_{OH} \geq 2.4V(I_{OH}=-100\ A)$

$V_{OL} \leq 0.4V(I_{OL}=\ 1.6mA)$

(A) With Pull-up Resistance



$I_{LO} \leq 20\ A(V_{OUT}=V_{DD})$

$V_{OL} \leq 0.4V(I_{LO}=1.6mA)$

(B) Open Drain

Since all the bits of TMP4300C output ports and input/output ports are the open drain type, the mask option can not be specified.

5. Pull-up Resistors of $\overline{RST}$ Terminal and $\overline{INT}$ Terminal

All versions of TLCS-43 are provided with the pull-up quasi resistors (typical valve of 100k ohms) fabricated with MOS transistors for $\overline{RST}$ and $\overline{INT}$ terminals. The guaranteed valve of general electrical characteristics of these resistors are I_{IL2} MAX.=-0.1 mA(V_{IN}=0.6V). (Refer to DC CHARACTERISTICS.)

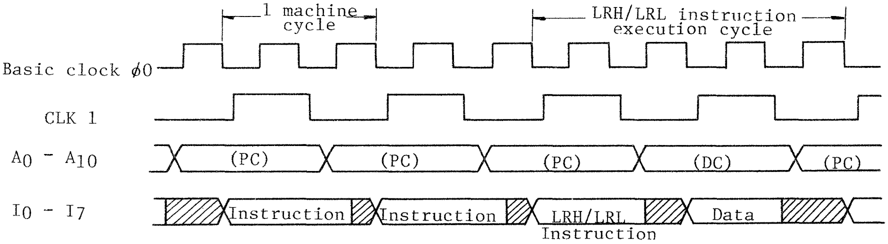
[EVALUATOR CHIP DESCRIPTION]

TMP4300C is the evaluator chip which is used for development of the application systems (or programs) for TLCS-43.
For these purposes, some terminals and functions have been added to TMP4300C in addition to those of other TLCS-43 chips.

The terminals and the functions dedicated to TLCS-43 development tool (TDS400/43) are also provided, and refer to the operation manual of TDS400/43 for details.

1. Operation Timing

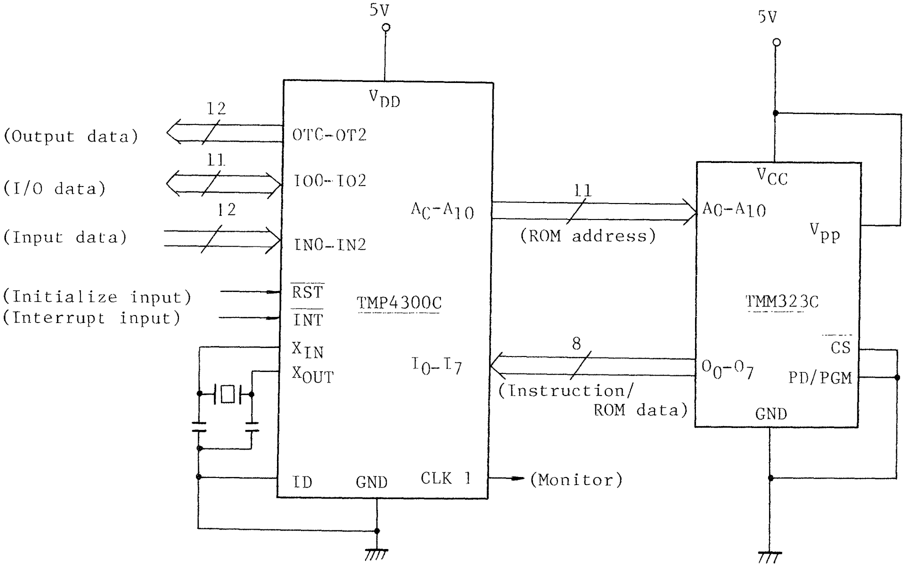
The normal operation timing of TMP4300C is shown in the figure below. The timings of the initialize operation and the interrupt operation are exactly same as the operation timings of other TLCS-43 chips.



2. Example of TMP4300C Application

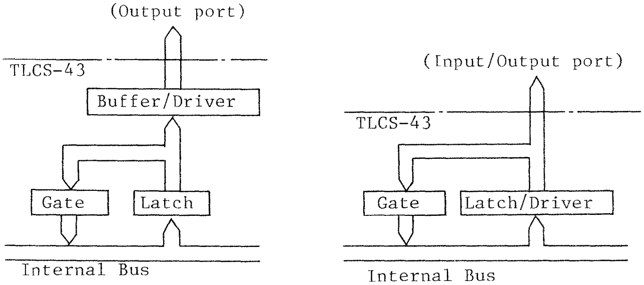
The diagram below illustrates an example of connection with an EPROM, which allows the program confirmation before confirming program to ROM.

* TMM323C is 2716 type EPROM with 16K bits (2,048 words x 8 bits)



3 Caution for Using IO2 Port

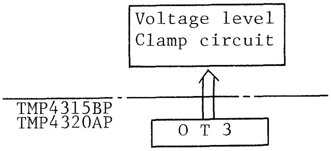
The output port and the input/output port of TLCS-43 have the configurations shown below.



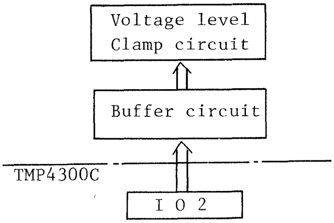
Therefore, when a circuit which clamps the output voltage level is directly connected externally to the terminal, and the output data is referenced by the program, it can be correctly read for the output port. However for the input/output ports there is possibility of reading erroneous data if the input voltage level is not secured by the clamp circuit. In order to read the data correctly, the terminals and the clamp circuits are required to be separated by the buffer circuits.

When a system development is conducted for TMP4315BP and TMP4320AP using TMP4300C, the pairing port for the output port OT3 is the input/output port IO2. Therefore, when it is required for a program to reference output data in OT3 port in a system using TMP4315BP or TMP4320AP,care should be taken not to directly connect a circuit which clamps the output level to IO2 port, during the evaluation stage using TMP4300C.

(Circuit of Actual Application)



(Circuit of Evaluation Stage)



TMP4310AP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{DD}	V_{DD} Supply Voltage	-0.5V to 7V
V_{DD}	Input Voltage	-0.5 V to 7V
V_{OUT1}	Output Voltage (Except Open Drain Pins)	-0.5V to 7V
V_{OUT2}	Output Voltage (Open Drain Pins)	-0.5V to 10V
I_{OUT1}	Average Output Current (Except OT1)	4mA
I_{OUT2}	Output Current (OT1)	30mA
P_D	Power Dissopation (TA=70°C)	700mA
T_{SOLDER}	Soldering Temperature (Soldering Time 10sec.)	260°C
T_{STG}	Storage Temperature	-55°C to 125°C
T_{OPR}	Operating Temperature	-10°C to 70°C

DC CHARACTERISTICS

T_A=-10°C to 70°C , V_{DD}=5V ± 10 %, Unless Otherwise Noted

Symbol	Parameter		Test Conditions	Min.	Typ.	Max.	Units
V_{IH1}	Input High Voltage (INO,IOO,IO1,IO2,$\overline{RST}$)			2.2		V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{INT}$)			3.5		V_{DD}	V
V_{IL}	Input Low Voltage			0		0.1	V
V_{CH}	Clock Input High Voltage (X_{IN})		External Drive	3.8		V_{DD}	V
V_{CL}	Clock Input Low Voltage (X_{IN})		External Drive	0		0.6	V
I_{IN1}	Input Current (INO)		V_{IN}=V_{DD}			20	μA
I_{IN2}	Input Current(IOO,IO1,IO2)	Open Drain	V_{IN}=V_{DD}			20	μA
		Pull Up				-	-
I_{IL1}	Input Low Current (IOO,IO1,IO2)	Open Drain				-	-
		Pull Up	V_{IN}=0.6V			-1.6	mA

Symbol	Parameter		Test Condition	Min.	Typ.	Max.	Unch
I_{IL2}	Input Low Current($\overline{RST}$, $\overline{INT}$)		V_{IN}=0.6V			-0.1	mA
I_{LO}	Output Lead Current (OTO, OT1)	Open Drain	$V_{OUT}=V_{DD}$			20	μA
		Pull Up				-	-
V_{OH}	Output High Voltage	Open Drain				-	-
		Pull Up	I_{OH}=-100uA	2.4			V
V_{OL}	Output Low Voltage (Note)		I_{OL}=1.6mA			0.4	V
I_{DD}	V_{DD} Supply Current				40	80	mA

Note: Output port OT1 can sink large current. (I_{OL} TYP.=20mA,V_{OL}=2.0V) While sinking large current, the output low voltage (V_{OL}) limit is the following value.

V_{OL} Max.=0.5V (I_{OL}=1.6mA)

AC CHARACTERISTICS Refer to TIMING WAVEFORMS (1).

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

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$t_{\phi o}$	Clock Cycle Time		2		5	μs
t_S	Input Set up Time		0.9			μs
t_H	Input Hold Time		0.9			μs
t_D	Output Delay Time	C_L=50PF,R(Pull-up)=50k ,1TTL			1.8	μs
t_{INT}	$\overline{INT}$ Low Level Pulse Width		4			Clock Cycle
t_{RST}	$\overline{RST}$ Low Level Pulse Width		4			Clock Cycle
f_{OSC}	Internal Oscillation Frequence	R=43k	250		450	kHz

TMP4315BP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{DD}	V_{DD} Supply Voltage	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to 7V
V_{OUT1}	Output Voltage (Except Open Drain Pins)	-0.5V to 7V
V_{OUT2}	Output Voltage (Open Drain Pins)	-0.5V to 10V
I_{OUT1}	Average Output Current	4mA
P_D	Power Dissopation (TA=70°C)	700mW
I_{SOLDER}	Soldering Temperature (Soldering Time, 10sec.)	260°C
T_{STG}	Storage Temperature	-55°C to 125°C
T_{OPR}	Operating Temperature	-10°C to 70°C

DC CHARACTERISTICS

TA=-10°C to 70°C, V_{DD}=5V ± 10 %, Unless Otherwise Noted.

Symbol	Parameter		Test Condition	Min.	Typ.	Max.	Units
V_{IH1}	Input High Voltage (IN0,IN1,IN2,IO0,IO1, $\overline{RST}$)			2.2		V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{INT}$)			3.5		V_{DD}	V
V_{IL}	Input Low Voltage			0		0.6	V
V_{CH}	Clock Input High Voltage (X_{IN})		External Drive	3.8		V_{DD}	V
V_{CL}	Clock Input Low Voltage (X_{IN})		External Drive	0		0.6	V
I_{IN1}	Input Current (IN0,IN1,IN2)		VIN=VDD			20	μA
I_{IN2}	Input Current(IO0,IO1)	Open Drain	V_{IN}=V_{DD}			20	μA
		Pull up				-	-
I_{IL1}	Input Low Current (IO0,IO1)	Open Drain				-	-
		Pull Up	V_{IN}=0.6V			-1.6	mA
I_{IL2}	Input Low Current($\overline{RST}$,$\overline{INT}$)		V_{IN}=0.6V			-0.1	mA
I_{LO}	Output Lead Current (OT0,OT1,OT2,OT3)	Open Drain	V_{OUT}=V_{DD}			20	μA
		Pull Up				-	-
V_{OH}	Output High Voltage (Except X_{OUT})	Open Drain				-	-
		Pull Up	I_{OH}=-100uA	2.4			V
V_{OL}	Low Output Voltage(Except X_{OUT})		I_{OL}=1.6mA			0.4	V
I_{DD}	V_{DD} Supply Current				40	80	mA

AC CHARACTERISTICS Refer to TIMING WAVEFORMS (1).

TA=-10°C to 70°C, V_{DD}=5V ± 10 %, Unless Otherwise Noted.

Symbol	Parameter		Min.	Typ.	Max.	Units
$t_{\phi o}$	Clock Cycle Time	Test Condition	2		5	µs
t_S	Input Set up Time		0.9			µs
t_H	Input Hold Time		0.9			µs
t_D	Output Delay Time	C_L=50PF,R(Pull up)=50kΩ ,1TTL			1.8	µs
t_{INT}	$\overline{\text{INT}}$ Low Level Pulse Width		4			Clock Cycle
t_{RST}	$\overline{\text{RST}}$ Low Level Pulse Width		4			Clock Cycle

TMP4320AP ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{DD}	V_{DD} Supply Voltage	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to 7V
V_{OUT1}	Output Voltage (Except Open Drain Pins)	-0.5V to 7V
V_{OUT2}	Output Voltage (Open Drain Pins)	-0.5V to 10V
I_{OUT1}	Average Output Current (Except OT1,OT2)	4mA
I_{OUT2}	Output Current (OT1, OT2)	30mA
P_D	Power Dissipation (TA=70°C)	850mW
T_{SOLDER}	Soldering Temperature (Soldering Time 10sec.)	260°C
T_{STG}	Storage Temperature	-55°C to 125°C
T_{OPR}	Operating Temperature	-10°C to 70°C

DC CHARACTERISTICS

TA=-10°C to 70°C, V_{DD}=5V ± 10 %, Unless Otherwise Noted

Symbol	Parameter		Test Conditions	Min.	Typ.	Max.	Units
V_{IH1}	Input High Voltage (IN0, IN1,IN2,IO0,IO1,$\overline{RST}$)			2.2		V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{INT}$)			3.5		V_{DD}	V
V_{IL}	Input Low Voltage			0		0.6	V
V_{CH}	Clock Input High Voltage (X_{IN})		External Drive	3.8		V_{DD}	V
V_{CL}	Clock Input Low Voltage (X_{IN})		External Drive	0		0.6	V
I_{IN1}	Input Current (IN0, $\overline{IN1}$, $\overline{IN2}$)		V_{IN}=V_{DD}			20	μA
I_{IN2}	Input Current (IO0,IO1)	Open Drain	V_{IN}=V_{DD}			20	μA
		Pull up				-	-
I_{IL1}	Input Low Current (IO0, IO1)	Open Drain				-	-
		Pull Up	V_{IN}=0.6V			-1.6	mA
I_{IL2}	Input Low Current($\overline{RST}$,$\overline{INT}$)		V_{IN}=0.6V			-0.1	mA
I_{LO}	Output Lead Current (OT0, OT1, OT2, OT3)	Open Drain	V_{OUT}=V_{DD}			20	μA
		Pull Up				-	-
V_{OH}	Output High Voltage (Except X_{OUT})	Open Drain				-	-
		Pull Up	I_{OH}=-100uA	2.4			V
V_{OL}	Output Low Voltage (Except X_{OUT}) (Note)		I_{OL}=1.6mA			0.4	V
I_{DD}	V_{DD} Supply Current				40	80	mA

Note: Output port OT1 and OT2 can sink large current. (I_{OL}.TYP.=20mA, V_{OL}=2.0V)

While sinking large current, the output low voltage (V_{OL}) limit is the following value.

V_{OL} Max.=0.5V (I_{OL}=1.6mA)

AC CHARACTERISTICS Refer to TIMING WAVEFORMS (1).

TA=-10°C to 70°C, V_{DD}=5V ± 10%, Unless Otherwise Noted.

SYMBOL	Parameter	Test Conditions	Min.	Typ.	Max.	Units
$t_{\phi o}$	Clock Cycle Time		2		5	µs
t_S	Input Set up Time		0.9			µs
t_H	Input Hold Time		0.9			µs
t_D	Output Delay Time	C_L=55pF,R(Pull up)=50KΩ,1TTL			1.8	µs
t_{INT}	INT Low Level Pulse Width		4			Clock Cycle
t_{RST}	RST Low Level Pulse Width		4			Clock Cycle

TMP4300C ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{DD}	V_{DD} Supply Voltage	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to 7V
V_{OUT1}	Output Voltage (Except Open Drain Pins)	-0.5V to 7V
V_{OUT2}	Output Voltage (Open Drain Pins)	-0.5V to 10V
I_{OUT1}	Average Output Current (Except OT1, OT2)	4mA
I_{OUT2}	Output Current (OT1, OT2)	30mA
P_D	Power Dissipation (TA=70°C)	1W
T_{SOLDER}	Soldering Temperature (Soldering Time 10 sec.)	260°C
T_{STG}	Storage Temperature	-55°C to 125°C
T_{OPR}	Operating Temperature	-10°C to 70°C

DC CHARACTERISTICS

TA=-10°C to 70°C, V_{DD}=5V ± 10 % , Unless Otherwise Noted.

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
V_{IH1}	Input High Voltage (Except $\overline{\text{INT}}$, X_{IN}) (Note)		2.2		V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{\text{INT}}$)		3.5		V_{DD}	V
V_{IL}	Input Low Voltage		0		0.6	V
V_{CH}	Clock Input High Voltage (X_{IN})	External Drive	3.8		V_{DD}	V
V_{CL}	Clock Input Low Voltage (X_{IN})	External Drive	0		0.6	V
I_{IN}	Input Current (Except $\overline{\text{RST}}$, $\overline{\text{INT}}$)	$V_{IN}=V_{DD}$			20	μA
I_{IL2}	Input Low Current ($\overline{\text{RST}}$, $\overline{\text{INT}}$)	V_{IN}=0.6V			-0.1	mA
I_{LO}	Output Lead Current(OT0,OT1,OT2)	$V_{OUT}=V_{DD}$			20	μA
V_{OH}	Output High Voltage (A0-A10,CLK1)	I_{OH}=-100uA	2.4			V
V_{OL}	Output Low Voltage (Except X_{OUT}) (Note)	I_{OL}=1.6mA			0.4	V
I_{DD}	V_{DD} Supply Current			70	120	mA

Note: Output Port OT1 and OT2 can sink large current. (I_{OL} TYP.=20mA, V_{OL}=2.0V) While sinking large current, the Output Low Voltage (V_{OL}) limit and the Input High Voltage (V_{IH1}) limit are the following values.

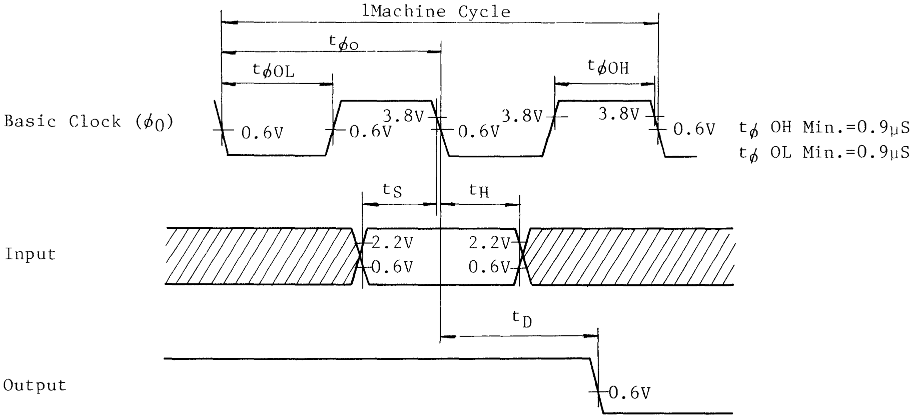
V_{OL} Max.=0.5V to 0.6V (I_{OL}=1.6mA)
V_{IH1} Min.=2.3V to 2.4V

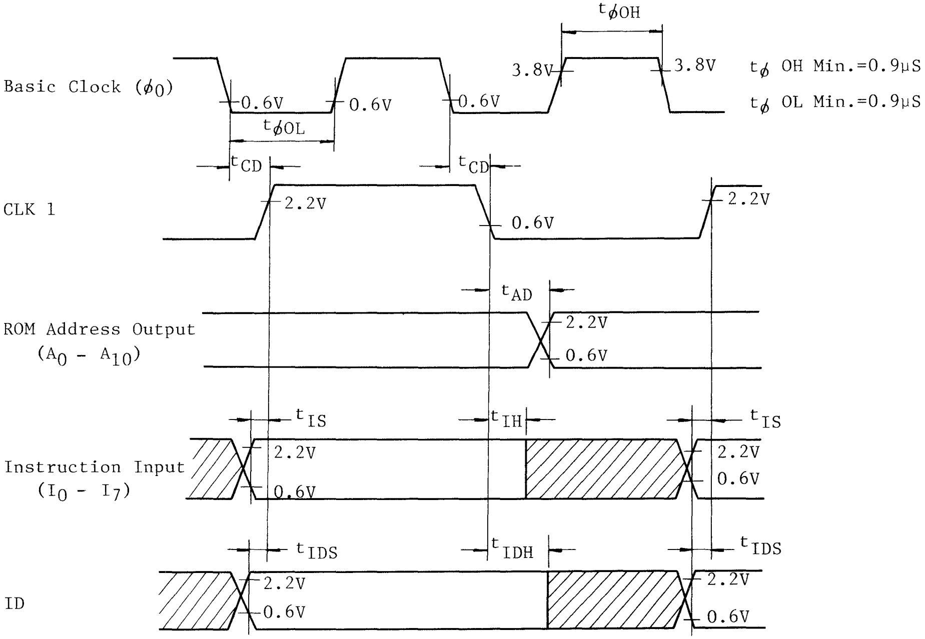
AC CHARACTERISTICS Refer to TIMING WAVEFORMS (1) (2).

TA=-10°C to 70°C, V_{DD}=5V ± 10 %, Unless Otherwise Noted.

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units.
$t_{\phi o}$	Clock Cycle Time		2		5	μs
t_s	Input Set up Time		0.9			μs
t_H	Input Hold Time		0.9			μs
t_D	Output Delay Time	C_L=50PF,R(Pull up)=50kΩ ,1 TTL			1.8	μs
t_{INT}	$\overline{\text{INT}}$ Low Level Pulse Width		4			Clock Cycle
t_{RST}	$\overline{\text{RST}}$ Low Level Pulse Width		4			Clock Cycle
t_{CD}	Clock Output Delay Time	C_L=50PF, 1 TIL			0.4	μs
t_{AD}	Address Output Delay Time	C_L=50PF, 1 TIL			0.95	μs
t_{IS}	Instruction Input Set up Time		0.4			μs
t_{IH}	Instruction Input Hold Time		0			μs
t_{IDS}	ID Input Set up Time		0.4			μs
t_{IDH}	ID Input Hold Time		0.95			μs

TIMING WAVEFORMS

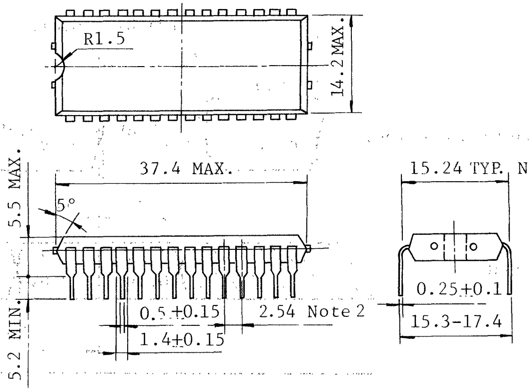




OUTLINE DRAWINGS

(TMP4310AP)

Unit in mm

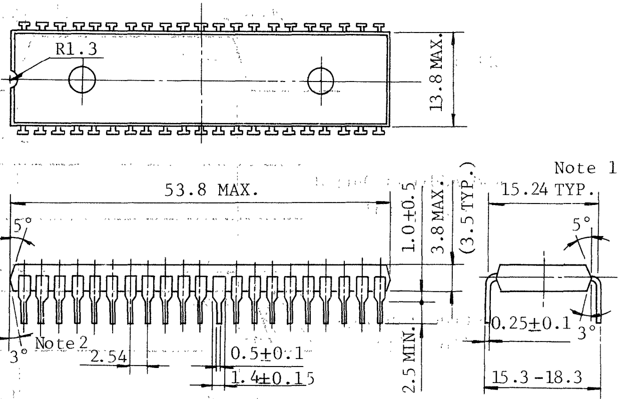


Note 1. This dimension is measured at the center of bending point of leads.

2. Each lead pitch is 2.54mm, and all the leads are located within ±0.25mm from their theoretical positions with respect to No. 1 and No. 28 leads.

(TMP4315BP/TMP4320AP)

Unit in mm

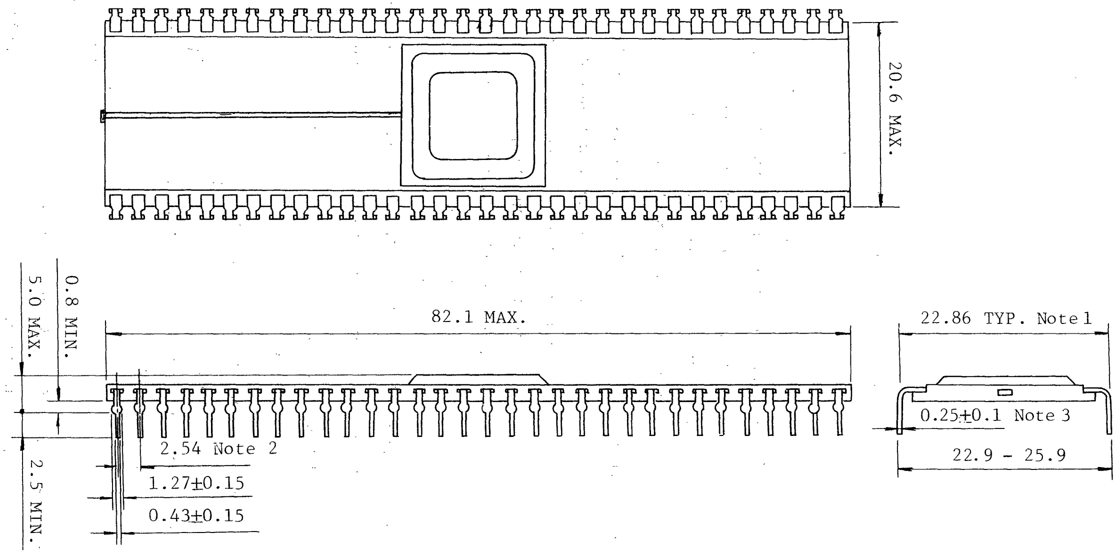


Note 1. This dimension is measured at the center of bending point of leads.

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

(TMP4300C)

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 theoretical positions with respect to No. 1 and No. 64 leads.

3. The metal on the side of the package is GND Level.

MASK ROM DATA TAPE FORMAT

⋮	---- Leader 50 or more characters of "NULL"
'COMMENT" (CR) (LF)	------- Comment When designated by mask option the characters of "MASK OPTION" are output with apostrophe.
(PUNNNi) (CR) (LF)	------- Outputs symbolic names of bits of output port or input/output port to which pull-up resistors are connected.
⋮	Outputs symbolic names of bits to which pull-up resistors are connected for all the bits of output port and input/output port which has been designated as mask option.
'COMMENT"	------- Comment Outputs six characters starting from the beginning of character train defined by TTL statement of source program and two characters of serial number, with apostrophe. (When no TTL statement exist, six characters of space code and two characters of serial number are output.)
N8; (CR) (LF)	------- Outputs "N8" which indicates that the data pattern is 8 bits long. (The program data follow this code.)
RXXXXX ;	------- Outputs the program start address following "R", in four frames of decimal ASCII code.
X XX PX;	------- Data and check sum of the first address.
X XX PX;	------- Data and check sum of the second address.
⋮	⋮
X XX PX; (CR) (LF)	------- Data and check sum of the eighth address.

PUNNNi
- i — Bit Name
- NN — Port Name

X XX PX
- PX — Check Sum
  Number of data bits having "1" is output
- XX — Data
  Two characters of eight bit data are output in two frames of hexadecimal ASCII codes.

```
 R XXXX ;   ------ Nineth program address.

 X XX PX;   ------ Data and check sum of the nineth address
     .
     .             Outputs repetitively through the last data.
     .
     .
     .
     .
 (CR) (LF)

     $      ------- Outputs symbol "$" to indicate the end of program data.
     .     }
     .     }
     .     }   Trailer   50 or more characters of "NULL".
     .     }
     .     }
     .
```

SPECIFICATION OF TMP4310APL/TMP4310APLL

This specification is applied to TMP4310APL/TMP4310APLL, one of versions of TLCS-43.

TMP4310APL/TMP4310APLL is the low power version of TMP4310AP. There are some differences in electrical characteristics between TMP4310APL/TMP4310APLL and TMP4310AP; however, their functions, instructions and pin connections are compatible. When using and examining TMP4310APL/TMP4310APLL, therefore, it is recommended that this specification be used together with the technical data on TMP4310AP.

Main differences in electrical characteristics between AP and APL/APLL are as follows:

1. Basic clock

   Internal oscillation (with resistance externally installed between X_{IN} and $\overline{TEST}$) and external clock supply can be available.

   1.1 Frequency of external supply clock

   f=200kHz to 400kHz (APL) (V_{DD}=5.5V ± 20%, T_{opr}=0°C to 55°C)
   f=50kHz to 200kHz (APLL)

   1.2 Internal oscillation frequency

   f=200kHz to 400kHz (APL, R_X=56kΩ) (V_{DD}=5.5V ± 20%, T_{opr}=0°C to 55°C)
   f=110kHz to 200kHz (APLL, R_X=110kΩ)

2. Low input voltage and clock low input voltage (at time of external supply)

   V_{IL} MAX.=V_{CL} MAX.=0.55V (V_{DD}=5.5V ± 20%, T_{opr}=0°C to 55°C)

3. High output voltage

   V_{OH} MIN.=2.4V (V_{DD}=5.5V ± 20%, T_{opr}=0°C to 55°C, I_{OH}=-50μA)

4. Supply current

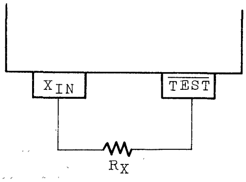
   I_{DD} TYP.=30mA, I_{DD} MAX.=45mA (APL) (V_{DD}=6.0V, T_{opr}=25°C)
   I_{DD} TYP.=15mA, I_{DD} MAX.=27mA (APLL)

5. Pull up resistance of $\overline{RST}$ and $\overline{INT}$ terminals is not contained.

TMP4310APL/TMP4310APLL BASIC CLOCK

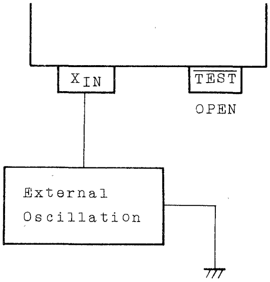
The methods of generating and supplying the basic clock of TMP4310APL/TMP4310APLL are as follows:

(1) Internal oscillation



The basic clock of TMP4310APL/TMP4310APLL can be obatined by connecting the resistance as shown in the left figure.
(f_{osc}=200kHz to 400kHz at R_X=56kΩ, APL)
(f_{osc}=110kHz to 200kHz at R_X=110kΩ, APLL)

(2) Supply of external clock



The basic clock of TMP4310APL/TMP4310APLL can be supplied from the external oscillation circuit as shown in the left figure.

TMP4310APL/TMP4310APLL ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATING

SYMBOL	I T E M	RATING
V_{DD}	Supply Voltage	-0.5V to 7V
V_{IN}	Input Voltage	-0.5V to 7V
V_{OUT1}	Output Voltage (Except open drain terminal)	-0.5V to 7V
V_{OUT2}	Output Voltage (Open drain terminal)	-0.5V to 10V
Tstg	Storage Temperature	-55°C to 125°C
Topr	Operating Temperature	0°C to 55°C
Tsld	Soldering Temperature	260°C (10sec.)

D.C. CHARACTERISTICS(V_{DD}=5.5V±20%, Topr=0°C to 55°C) Unless otherwise noted.

SYMBOL	PARAMETER		TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IH1}	Input High Voltage	(INO,IOO,IO1,IO2,$\overline{RST}$)	-	2.2	-	VDD	V
V_{IH2}		($\overline{INT}$)	-	3.5	-	VDD	V
V_{IL}	Input Low Voltage		-	0	-	0.55	V
V_{CH}	Clock Input High Voltage (X_{IN})		External clock supply	3.8	-	VDD	V
V_{CL}	Clock Input Low Voltage (X_{IN})			0	-	0.55	V
I_{IN1}	Input Current (INO,$\overline{RST}$,$\overline{INT}$)		V_{IN} = VDD	-	-	20	μA
I_{IN2}	Input Current (IOO, IO1, IO2)	Open Drain	V_{IN} = V_{DD}	-	-	20	μA
		Pull Up	-	-	-	-	-
I_{IL1}	Input Low Current (IOO, IO1, IO2)	Open Drain	-	-	-	-	-
		Pull Up	V_{IN} = 0.55V	-	-	-1.6	mA
I_{LO}	Output Leak Current (OTO, OT1)	Open Drain	V_{OUT} = V_{DD}	-	-	20	μA
		Pull Up	-	-	-	-	-
V_{OH}	Output High Voltage	Open Drain	-	-	-	-	-
		Pull Up	I_{OH} = -50μA	2.4	-	-	V
V_{OL}	Output Low Voltage		I_{OL} = 1.6mA	-	-	0.4	V
I_{DD}	Supply Current	APL	V_{DD}=6.0V, Ta=25°C	-	30	45	mA
		APLL		-	15	27	mA

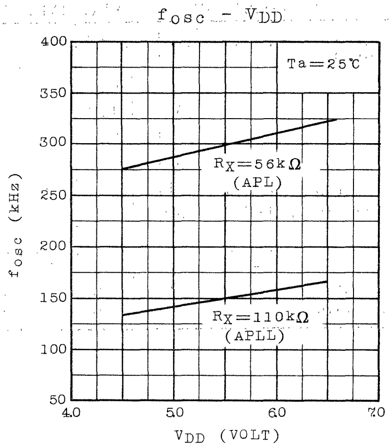
Pull-Up resistance of $\overline{RST}$ and $\overline{INT}$ terminals is not contained.

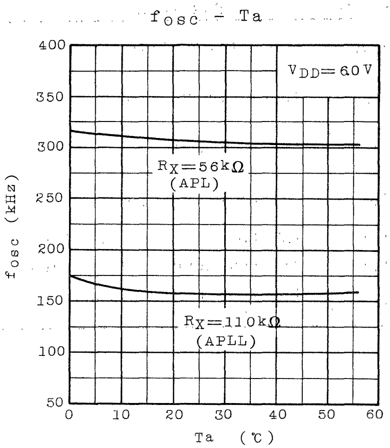
Low output voltage: It is possible that output port OT1 sinks large output current I_{OL} Typ.=20mA (V_{OL}=2V) for each pins. At the time when OT1 sinks large output current, low output voltage (VOL) becomes V_{OL} Max.≈0.5V (I_{OL}=1.6mA).

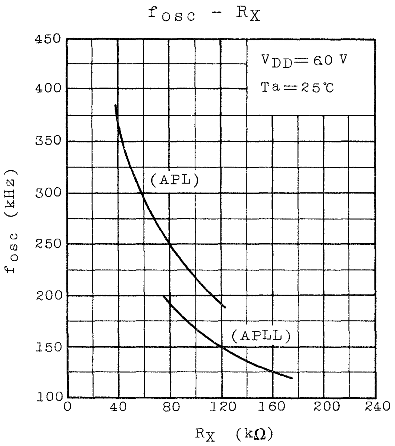
A.C. CHARACTERISTICS (V_{DD}=5.5V±20%, Topr=0°C to 55°C)

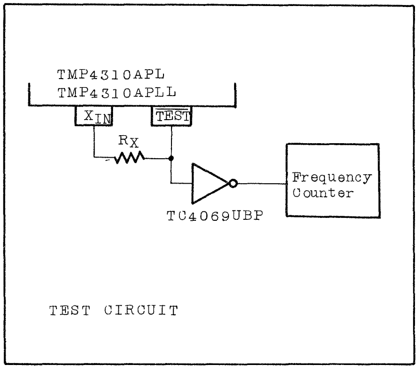
SYMBOL	PARAMETER	TEST CONDITION		MIN.	TYP.	MAX.	UNIT
$t_{\phi O}$	Clock Cycle Time	External clock supply	APL	2.5	-	5.0	μs
			APLL	5	-	20	μs
t_S	Input Set Up Time	-	APL	0.9	-	-	μs
			APLL	1.8	-	-	μs
t_H	Input Hold Time	-	APL	0.9	-	-	μs
			APLL	1.8	-	-	μs
t_D	Output Delay Time	C_L=50pF, R(Pull up)=50kΩ, 1TTL	APL	-	-	1.8	μs
			APLL	-	-	3.6	μs
t_{INT}	$\overline{\text{INT}}$ Low Level Pulse Width	-		4	-	-	Cycle
t_{RST}	$\overline{\text{RST}}$ Low Level Pulse Width	-		4	-	-	Cycle
f_{osc}	Internal Oscillation Frequency	R =56kΩ	APL	200	-	400	kHz
		R =110kΩ	APLL	110	-	200	kHz

TYPICAL CHARACTERISTICS

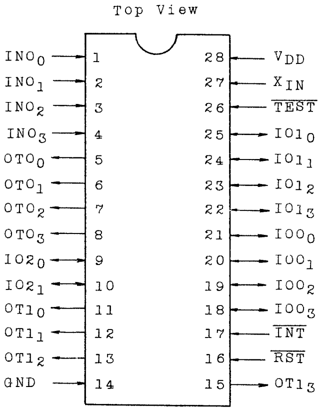






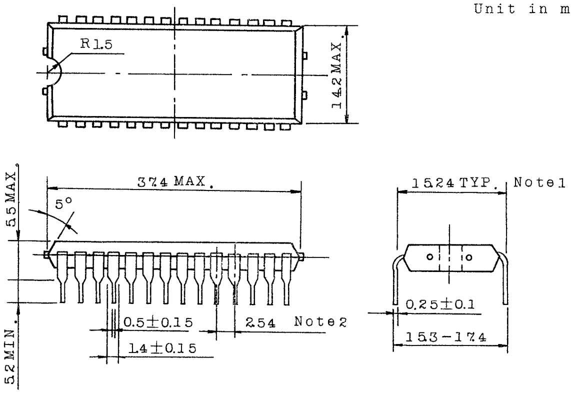


PIN CONNECTIONS



OUTLINE DRAWINGS

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 theoretical position with respect to No.1 and No.28 leads.

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4321AP

TOSHIBA MOS Digital Integrated Circuit
Silicon Monolithic
N-channel Silicon Gate Depression Load

GENERAL DESCRIPTION

TMP4321AP is one version of single-chip microcomputer TLCS43 series processed with NMOS.

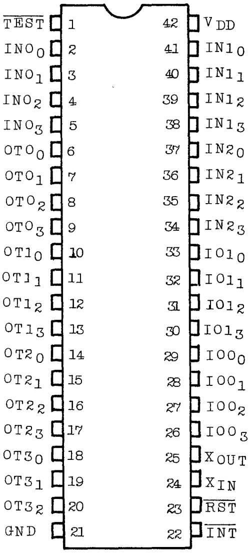
Since TMP4321AP can make periodic interrupt gained by dividing the CPU clock to 512 or 1024 using the built-in prescaler, it is most appropriate for computer control, such as sequence control.

Refer to the technical data of TLCS43 when using or considering TMP4321AP.

FEATURES

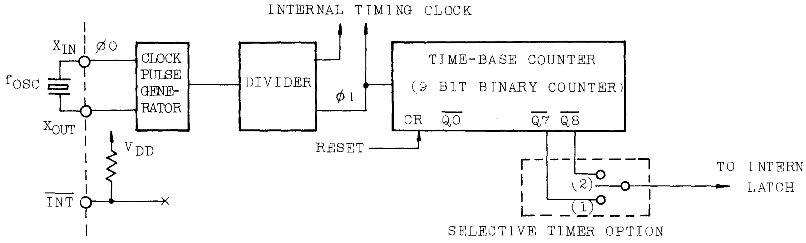
- o Software compatible with TLCS43 series
- o ROM capacity: 2048W x 8 Bits
- o RAM capacity: 128W x 4 Bits
- o Instruction executing rate: 4 μs
- o Input ports: 3 x 4 Bits
- o Input/output ports: 2 x 4 Bits
- o Output ports: 3 x 4 Bits
  1 x 3 Bits
- o Interrupt level: 1 level (Interrupt by timer)
- o Subroutine nesting: 4 levels (including interrupt level)
- o ROM data readout instructions
- o I/O level: TTL compatible
- o Ports available for large current output: 2 x 4 Bits
- o +5V single power operating (operating voltage margin: 5V ± 10%)
- o Operating temperature range: -10 ∿ 70°C

PIN CONNECTIONS



TIMER INTERRUPT

The following figure shows the configuration of timer interrupt circuit of TMP4321AP.



As shown in the above figure, interrupt is not caused by external source via $\overline{\text{INT}}$ pin, but it is only caused by the output divided by the built-in prescaler. The $\overline{\text{INT}}$ pin is usually set to "H" level (open) by the pull-up resistor, but it is not connected with the internal circuit.

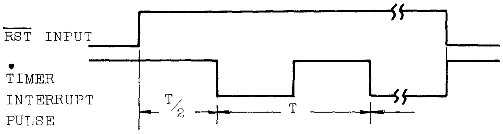
Assignment of selective timer option is as follows :

(1) Option INTDV7 is interrupted at $\frac{1}{\text{fosc}}$ x 512(S) cycle ($\overline{\text{Q7}}$).

(2) Option INTDV8 is interrupted at $\frac{1}{\text{fosc}}$ x 1024(S) cycle ($\overline{\text{Q8}}$).

(Refer to "Mask Option Assignment"in the technical data of TLCS43 for further details.)

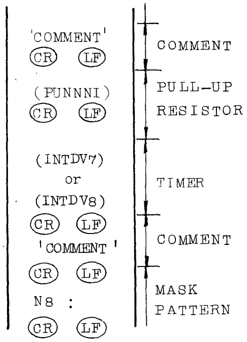
The relation between timer interrupt and reset is shown in the following figure.



T: INTERRUPT CYCLE BY OPTION ASSIGNMENT

Initial interrupt occurs at T/2 interval after the $\overline{\text{RST}}$ input is released, and then repeatedly occurs by every T cycle.

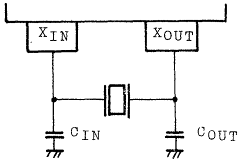
MASK ROM DATA TAPE FORMAT



BASIC CLOCK

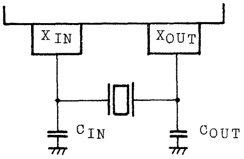
The following are four methods to generate and supply the basic clock pulse in the TMP4321AP.

(1) Direct connection of crystal oscillator



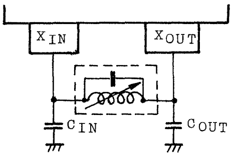
When a crystal oscillator has been connected to TMP4321AP as shown in the left figure, the basic clock pulse is generated at the frequency inherent in the crystal oscillator.
Recommendations for C_{IN} and C_{OUT}
$C_{IN}=C_{OUT}=60pF$ ($f_{OSC} \approx$ 500KHz)

(2) Direct connection of ceramic oscillator



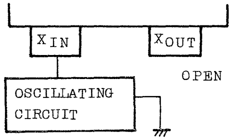
When a ceramic oscillator has been connected to TMP4321AP as shown in the left figure, the basic clock pulse is generated at the frequency inherent in the ceramic oscillator.
Recommendations for C_{IN} and C_{OUT}
$C_{IN}=C_{OUT}=100pF$ ($f_{OSC} \approx$ 500KHz)

(3) Direct connection of IFT



When a IFT has been connected to TMP4321AP as shown in the left figure, the basic clock pulse is determined according to the circuit constant.
Recommendations for C_{IN} and C_{OUT}
$C_{IN}=C_{OUT}=100pF$ ($f_{OSC} \approx$ 500KHz)

(4) Supply of external clock



The basic clock pulse of TMP4321AP can be supplied by an external oscillating circuit as shown in the left figure.

MAXIMUM RATINGS

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.5 ∿ 7	V
V_{IN}	Input Voltage	-0.5 ∿ 7	V
V_{OUT1}	Output Voltage (except open drain terminal)	-0.5 ∿ 7	V
V_{OUT2}	Output Voltage (open drain terminal)	-0.5 ∿ 10	V
I_{OUT1}	Output Current (except OT1, OT2)	4	mA
I_{OUT2}	Output Current (OT1, OT2)	30	mA
Tstg	Storage Temperature	-55 ∿ 125	°C
Topr	Operating Temperature	-10 ∿ 70	°C
Tsld	Solder Temperature	260 (10 sec.)	°C
P_W	Power Dissipation (Ta = 70°C)	850	mW

D.C. ELECTRICAL CHARACTERISTICS (V_{DD} = 5V±10%, Topr = -10 ∿ 70°C)

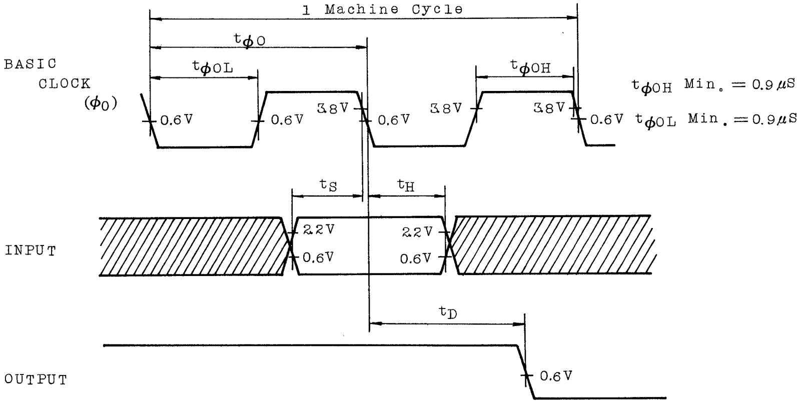
SYMBOL	PARAMETER		TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IH1}	Input High Voltage (IN0,IN1,IN2,IO0,IO1,$\overline{RST}$)		-	2.2	-	V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{INT}$)		-	3.5	-	V_{DD}	V
V_{IL}	Input Low Voltage		-	0	-	0.6	V
V_{CH}	Input High Clock Voltage (X_{IN})		-	3.8	-	V_{DD}	V
V_{CL}	Input Low Clock Voltage (X_{IN})		-	0	-	0.6	V
I_{IN1}	Input Current (INo, IN1, IN2)		V_{IN} = V_{DD}	-	-	20	μA
I_{IN2}	Input Current (IO0, IO1)	Open Drain	V_{IN} = V_{DD}	-	-	20	μA
		Pull Up	-	-	-	-	-
I_{IL1}	Input Low Current (IO0, IO1)	Open Drain	-	-	-	-	-
		Pull Up	V_{IN} = 0.6V	-	-	-1.6	mA
I_{IL2}	Input Low Current ($\overline{RST}$, $\overline{INT}$)		V_{IN} = 0.6V	-	-	-0.1	mA
I_{LO}	Output Leak Current (OT0,OT1 OT2,OT3)	Open Drain	V_{OUT} = V_{DD}	-	-	20	μA
		Pull Up	-	-	-	-	-
V_{OH}	Output High Voltage (except X_{OUT})	Open Drain	-	-	-	-	-
		Pull Up	I_{OH} =-100μA	2.4	-	-	V
V_{OL}	Output Low Voltage (except X_{OUT})		I_{OL} = 1.6 mA	-	-	0.4	V
I_{DD}	Supply Current		-	-	40	80	mA

Output Low Voltage : Output large current I_{OL} TYP.=20mA (V_{OL}=2V) is made possible by output ports OT1, OT2. When output large current sinks, output low voltage becomes V_{OL} MAX.≈0.5V(I_{OL}=1.6mA).

A.C. ELECTRICAL CHARACTERISTICS (V_{DD} = 5V±10%, Topr = -10 ∿ 70°C) Refer to Timing Chart.

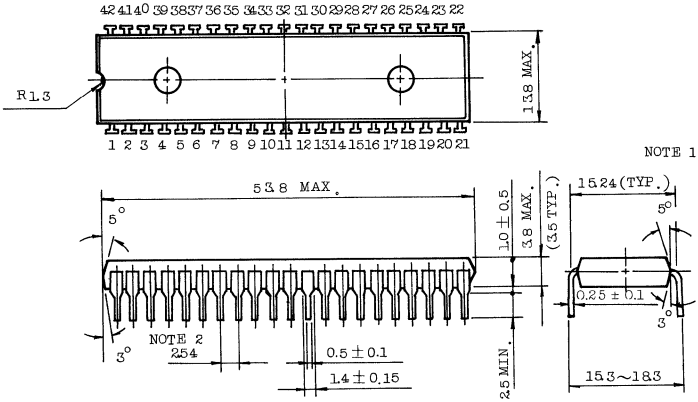
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
$t_{\phi 0}$	Clock Cycle Time	-	2	-	5	μS
t_S	Input Setup Time	-	0.9	-	-	μS
t_H	Input Hold Time	-	0.9	-	-	μS
t_D	Output Delay Time	C_L=50pF, R(Pull Up)=50kΩ,1TTL	-	-	1.8	μS
t_{INT}	$\overline{INT}$ (Low Level Width)	-	4	-	-	Cycle
t_{RST}	$\overline{RST}$ (Low Level Width)	-	4	-	-	Cycle

TIMING CHART



EXTERNAL VIEW OF PACKAGE (Plastic Package)

Unit: mm



Note 1. The dimension of this external view indicates the lead bending center.

Note 2. Lead pitch is 2.54mm and tolerance is ± 0.25mm against theoretical center of each lead that is obtained on the basis of No.1 and No.42 leads.

TMP4399C

TOSHIBA MOS Digital Integrated Circuit
Silicon Monolithic
N-channel Silicon Gate Depletion Load

DESCRIPTION

TMP4399C is a single-chip microcomputer for evaluation, provided with the general purpose 16K EPROM (TMM323D) inserting socket, being capable of developing and checking hardware and software of TLCS-43 Application System by use of the EPROM.

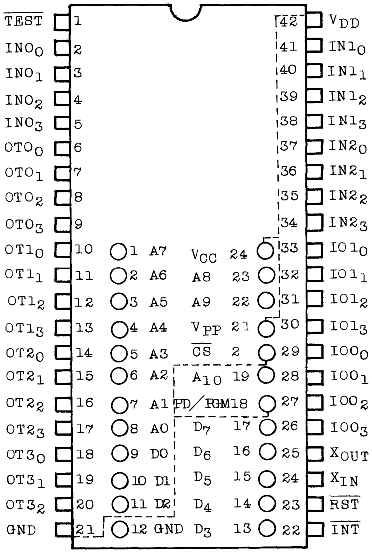
TMP4399C is equipped with a 24-pin socket which may directly mount the general purpose 16K EPROM (TMM323D) on the surface of the package. Therefore, when the fixed program written in the general purpose 16K EPROM is mounted on the package, TMP4399C becomes pin-compatible with TMP4315BP/ TMP4320AP as the end product which the program has been written in the mask ROM. The former operates the same as the latter.

TMP4399C can be used within the range of a microcomputer for evaluating the TLCS-43 system as well as for mounting an equipment made on an experimental basis.

FEATURES

- o General purpose 16K EPROM TMM323D (equivalent to INTEL2716) can be used.
- o Compatible with TLCS-43 single-chip microcomputer family TMP4315BP/ 4320AP in pin.
- o Compatible with TLCS-43 Series in software.
- o ROM 2KW x 8 BIT RAM 128W x 4 BIT
- o Can be operated by +5V single power supply. Voltage margin is ±10% equal to that of mass-produced products.
- o Operating temperature range is -10 ∿ 70°C equal to that of mass-produced products.

PIN CONNECTIONS

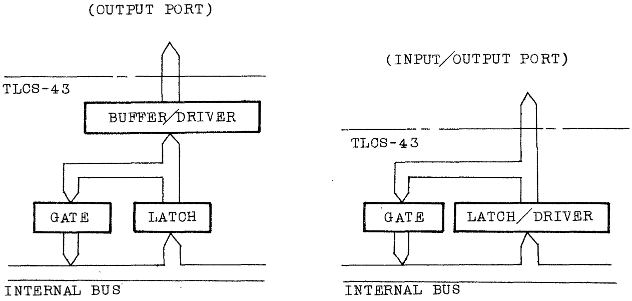


(Note) ○ mark: Socket for TMM323D

PRECAUTIONS FOR USING TMP4399C

1. Precautions for using OT3 Output Port

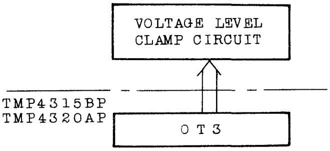
The output port and input/output port of TLCS-43 are composed as shown in the following figures.



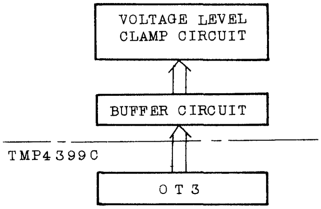
Therefore, when the circuit which clamps the output voltage level is connected directly to the outside of terminal, if the output data is referred to by the program, for the output port such data can be normally read, but for the input/output port erroneous data may be read unless the voltage level can be ensured as an input signal by the clamp circuit. For the purpose of reading correct data, it is necessary to separate the terminal from the clamp circuit by the buffer circuit.

The output port OT3 of TMP4399C serves as an input/output port owing to the configuration of the chip. In case of the system using TMP4315BP or TMP4320AP, therefore, if the output data of OT3 Port must be referred to by the program, care should be taken that the circuit which clamps the output level may not be connected directly to OT3 Port at the stage of development where TMP4399C is used.

(ACTUAL APPLICATION CIRCUIT)



(BUFFER CIRCUIT)



2. Precautions for using large output current

In case of TMP4399C, each terminal of the output port OT1 and OT2 may sink large output current I_{OL} =20mA TYP. (V_{OL} =2V) when there ports sink large output current, a part of the DC electrical characteristics is changed. Attention should be paid to the main characteristics that the high input voltage of V_{IH1} and the low output voltage of V_{OL} at every terminal to be applied will become the valurs shown as follows:

V_{IH1} MIN. ≈ 2.3 ∼ 2.4V

V_{OL} MAX. ≈ 0.5 ∼ 0.6V (I_{OL} = 1.6mA.)

The maximum rating of large output current is 30 mA.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Symbol	CHARACTERISTICS	RATING	UNIT
V_{DD}	Supply Voltage	-0.5 ∿ 7	V
V_{IN}	Input Voltage	-0.5 ∿ 7	V
V_{OUT1}	Output Voltage(excepting open drain terminal)	-0.5 ∿ 7	V
V_{OUT2}	Output Voltage(Open drain terminal)	-0.5 ∿ 10	V
I_{OUT1}	Output Current(Excepting OT1, OT2)	4	mA
I_{OUT2}	Output Current (OT1, OT2)	30	mA
T_{stg}	Storage Temperature	-55 ∿ 125	°C
Topr	Operating Temperature	-10 ∿ 70	°C
Tsld	Soldering Temperature	260(10 sec.)	°C
P_W	Power Consumption (Ta = 70°C)	1	W

D.C. ELECTRICAL CHARACTERISTICS (V_{DD} = 5V ± 10%, T_{opr} = -10 ∿ 70°C)

Symbol	CHARACTERISTICS	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{IH1}	Input High V.(excepting $\overline{INT}$,X_{IN})	-	2.2	-	V_{DD}	V
V_{IH2}	Input High Voltage ($\overline{INT}$)	-	3.5	-	V_{DD}	V
V_{IL}	Input Low Voltage	-	0	-	0.6	V
V_{CH}	Clock Input High Voltage (X_{IN})	-	3.8	-	V_{DD}	V
V_{CL}	Clock Input Low Voltage (X_{IN})	-	0	-	0.6	V
I_{IN}	Input Current(excepting $\overline{RST}$, $\overline{INT}$)	V_{IN} = V_{DD}	-	-	20	μA
I_{IL2}	Input Low Current ($\overline{RST}$, $\overline{INT}$)	V_{IN} = 0.6V	-	-	-0.1	mA
I_{LO}	Output Leak Current(OT0,OT1,OT2)	V_{OUT} = VDD	-	-	20	μA
V_{OH}	Output High Voltage (A0 ∿ A_{10})	I_{OH} = -100μA	24	-	-	V
V_{OL}	Output Low V. (excepting X_{OUT})	I_{OL} = 1.6 mA	-	-	0.4	V
I_{DD}	Supply Voltage	-	-	70	120	mA

[Operation of Evaluator Chip] Refer to the paragraph entitled "Precautions for Using Large Output Current".

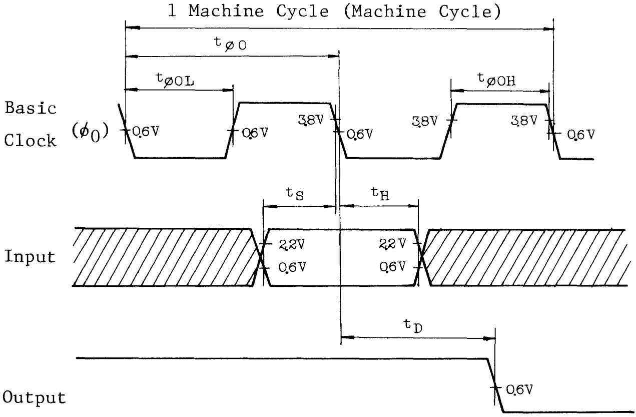
Refer to timing charts (1) and (2).

A.C. ELECTRICAL CHARACTERISTICS (V_{DD} = 5V ± 10%, T_{opr} = -10 ∿ 70°C)

Symbol	CHARACTERISTIC	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
$t\phi 0$	Clock Cycle Time	-	2	-	5	μs
t_S	Input Setup Time	-	0.9	-	-	μs
t_H	Input Hold Time	-	0.9	-	-	μs
t_D	Output Delay Time	C_L=50PF R(Pull UP) =50kΩ,1TTL	-	-	1.8	μs
t_{INT}	$\overline{INT}$ Low Level Width	-	4	-	-	Cycle
t_{RST}	$\overline{RST}$ Low Level Width	-	4	-	-	Cycle
t_{CD}	Clock Output Delay Time	C_L = 50PF, 1TTL	-	-	0.4	μs
t_{AD}	Address Output Delay Time	C_L = 50PF, 1TTL	-	-	0.95	μs
t_{IS}	Instruction Input Setup Time	-	0.4	-	-	μs
t_{IH}	Instruction Input Hold Time	-	0	-	-	μs

TIMING CHART

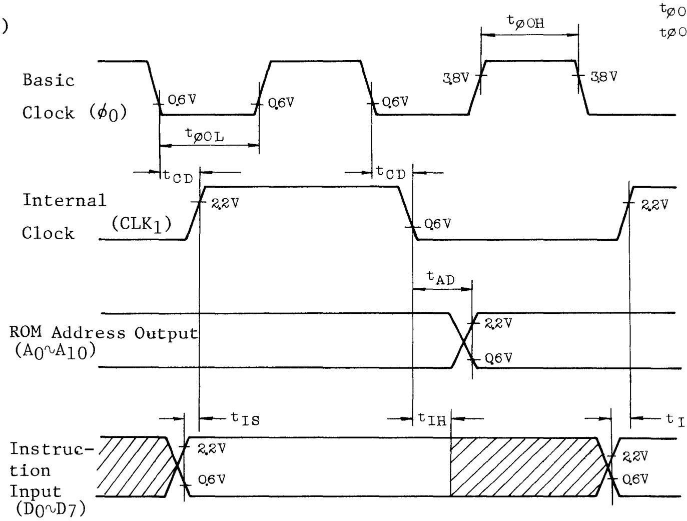
(1)



$t_{\phi OH}$ MIN. = 0.9μs
$t_{\phi OL}$ MIN. = 0.9μs

(2)

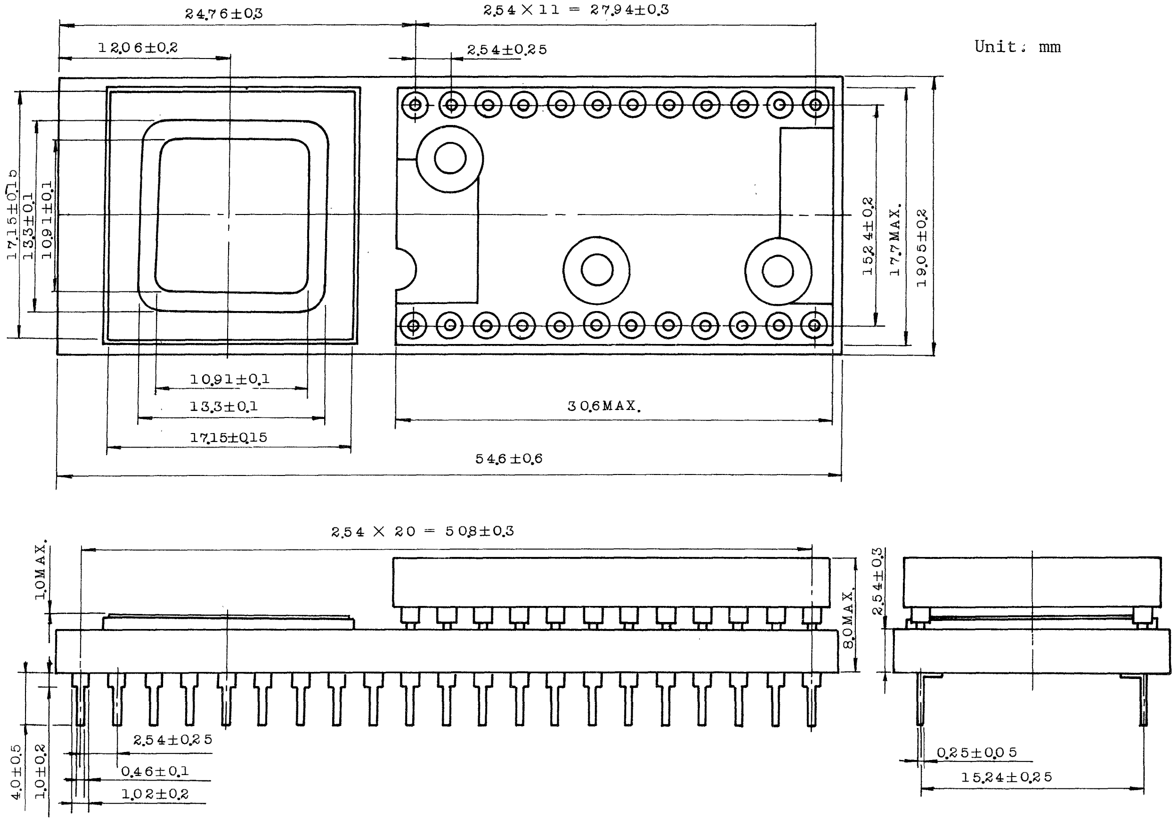
$t_{\phi OH}$ MIN. = 0.9μs
$t_{\phi OL}$ MIN. = 0.9μs



(Note) Since internal clock (CLK_1) is not output, it cannot be seen from the outside.

EXTERNAL SHAPE OF PACKAGE

Unit : mm



4BIT SINGLE CHIP MICROCOMPUTER

TLCS-46A(CMOS)

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

"C^2MOS" DIGITAL INTEGRATED CIRCUIT

TCP4620AP
TCP4630AP

SILICON MONOLITHIC

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

GENERAL DESCRIPTION

TLCS-46A is a C^2MOS high speed and low power 4-bit single chip microcomputer for consumer applications.

A single and integral microcomputer has been composed of a 4-bit parallel arithmetic and logical unit (ALU), accumulator (AC), program memory (ROM), data memory (RAM), input/output ports, clock generator, and divider incorporated.

TLCS-46A Family consists of two kinds of chips having different ROM/RAM capacities for mass production and evaluator for system development.

東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TCP4620AP
TCP4630AP

FEATURES

- TCP4620AP
  - 2048 x 8 ROM
  - 96 x 4 RAM
  - 34 I/O Lines

- TCP4630AP
  - 3072 x 8 ROM
  - 160 x 4 RAM
  - 34 I/O Lines

- TCP4600AP
  Evaluator Chip
  for TLCS-46A

- Low Power Dissipation by Employment of C^2MOS Process
  Typical Supply Current : 400μA
  (at 400 kHz Basic Clock)

- Single 5V Supply
  Wide Operating Range : 3V to 6V

- Wide Operating Temperature Range: -30°C to 85°C

- 52 Instructions
  - 46 One Cycle Instructions
  - 6 Two Cycle Instructions

- Single Level Subroutine Nesting

- Single Level External Interrupt

- 10μs Instruction Execution Time

- PLA and Decode Matrix for Display Operation

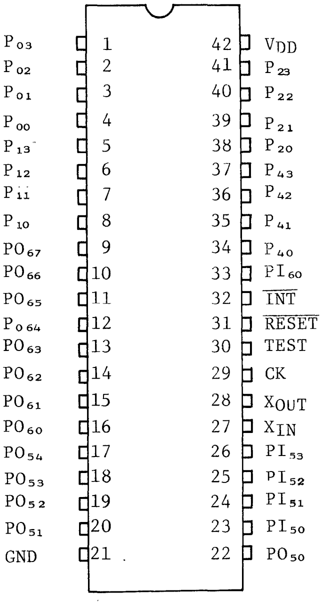
- Many Kinds of Mask Options for Optimum Application Systems

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TCP4620AP
TCP4630AP

PIN CONNECTIONS (TOP VIEW)

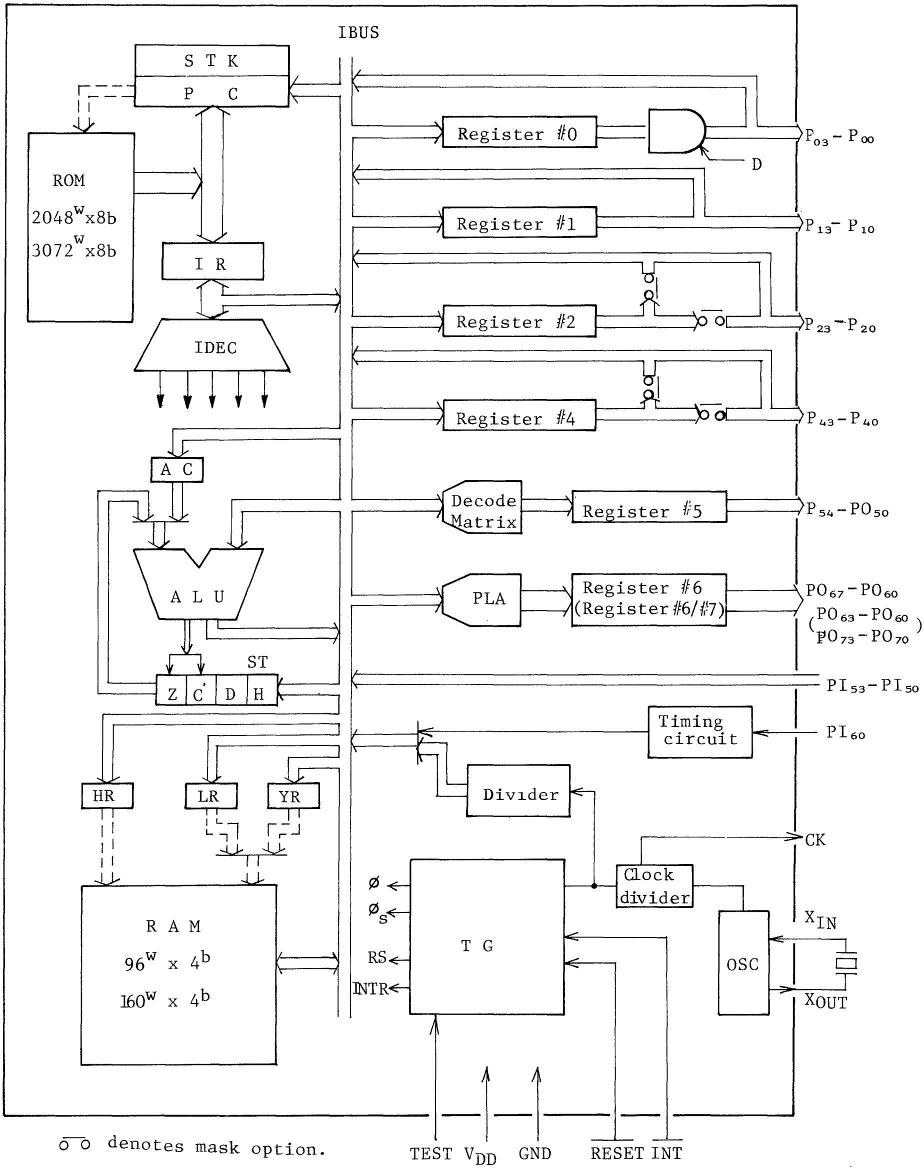


PIN NAMES & PIN DESCRIPTION

Pin Name	Input/Output	Function	
$P_{03}-P_{00}$	Input/Output	4-bit general purpose I/O port (I/O is designated by a program).	
$P_{13}-P_{10}$	Output	4-bit general purpose output port.	
$P_{23}-P_{20}$	Input or Output	4-bit general purpose Input/Output port	Input/Output is designated by mask options.
$P_{43}-P_{40}$	Input or Output	4-bit general purpose Input/Output port	
$PO_{54}-PO_{50}$	Output	5-digit output port for display. (Can be used as the general purpose 5-bit output port)	

Pin Name	Input/Output	Function
PO$_{67}$-PO$_{60}$	Output	8-segment output port for display (can be used as the general purpose 8-bit output port)
PI$_{53}$-PI$_{50}$	Input	4-bit general purpose input port.
PI$_{60}$	Input	1-bit general purpose input port (with a internal Schmitt circuit).
$\overline{\text{RESET}}$	Input	Initialize signal input (with a internal Schmitt circuit).
$\overline{\text{INT}}$	Input	Interrupt request signal input (with a inter-internal Schmitt circuit).
X$_{\text{IN}}$	Input	Oscillator connecting terminal
X$_{\text{OUT}}$	Output	Oscillator connecting terminal
CK	Output	External timing output
TEST	Input	LSI test signal input, used by connecting to GND
V$_{\text{DD}}$		Power supply
GND		GND

BLOCK DIAGRAM



o͞ o denotes mask option.

FUNCTIONAL DESCRIPTION

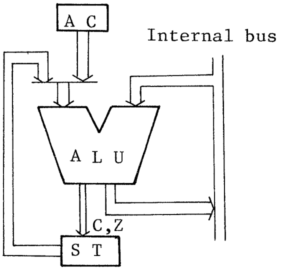
[SYSTEM CONFIGURATION]

TLCS-46A consists of the following elements.

1. BASIC ELEMENTS

(1) Arithmetic and Logical Unit (ALU)
(2) Accumulator (AC)
(3) Status Register (ST)
(4) H Register (HR), L Register (LR), Y Register (YR), Y Register Flag (EYR)
(5) Port (P), Port Register (Register)
(6) Internal Bus (IBUS)
(7) Data Memory (RAM)
(8) Program Memory (ROM)
(9) Program Counter (PC)
(10) Stack (STK), Stack Flag (FSTK)
(11) Instruction Register (IR), Instruction Decoder (IDEC)
(12) Clock Generator (OSC)
(13) Divider
(14) Timing Generator (TG)

(1) Arithmetic and Logical Unit (ALU)



ALU performs the principal function involved in the data processing of TLCS-46A, and consists of 4-bit binary parallel arithmetic circuit.

One of the inputs of the arithmetic and logical unit is the accumulator or the status register, and another input is the internal bus, and the result of operation is output to the internal bus and at the same time, carry (barrow) and zero are detected.

(2) Accumulator (AC)

The accumulator is a 4-bit register that temporarily stores data for arithmetic process, arithmetic result and data from/to the input/output ports.

(3) Status Register (ST)

The status register is a 4-bit register having a meaning per bit, and is called H flag, D flag, C flag and Z flag in that order from LSB side.

(MSB)			(LSB)
3	2	1	0
Z	C	D	H

① Zero Flag (Z)

The zero flag (Z) is set to "1",if the result generated by certain instructions is zero.
The zero flag is cleared, if the result is not zero.

Further, z flag is used not only for judgement of zero but also as a branch condition for program flow.

② C Flag

When an instruction indicating update is executed, C flag is set at "1" if carry is resulted at time of addition and increment, and at "0" if no carry is resulted.

Further C flag is also set at "0" when borrow is resulted at time of substruction and at "1" when no borrow is resulted.
C flag is used for judging size of data and for multiple digit arithmatic operation.

③ D Flag

D flag is used by a program as an input/output designating signal of the input/output port (Po). When D flag is "0", the input/output port serves as the input port and when D flag is "1", it serves as the output port.

When the input/output port is used exclusively as the output port (specified by mask option), D flag becomes the general purpose flag bit that can be optionally used by user.
D flag is reset at "0" by the initialize operation.

④ H Flag

H flag is used as a control signal for hold operation. When "1" is set on H flag, the timing generator is placed in hold mode and the operation is held suspensed. The restart from the hold mode is accomplished by resetting H flag at "0" in the hardware processing, and after released from the hold mode, the process that was held suspended prior to the hold operation is resumed. However, interrupt request is not accepted under the hold operation.

When the hold operation is not used (specified by mask option), H flag becomes a general purpose flag bit that can be optionally used by user.

H flag is reset at "0" by the initialize operation.

(4) H Register (HR), L Register (LR), Y Register (YR), Y Register Flag (EYR)

H register and L register are 4-bit registers and function as an address pointer of the data memory (RAM) or a general purpose register.

When H register and L register are used as an address pointer of the data memory, H register represents high order 4 bits of an address and L register represents low order 4 bits, and they designate an address space of total 8 bits (256 words). Therefore, when 16 words in the address space of the data memory are expressed as one page, H register designates a page address and L register designates an address in the page.

When an undefined region without data memory is read with H register and L register used as address pointers of the data memory, the data memory contents are regarded as being undefined. Further, data write into an undefined region should be avoided.

Y register is a 4-bit register and functions as an address pointer in page 0 of the data memory or a general purpose register.

Y register flag is an 1-bit flag that shows as to whether H register or L register is used as an address pointer of the data memory. When Y register flag is "1" (EYR=1), Y register is selected as an address pointer of the data memory, and when Y register flag is "0" (EYR=0), H register and L register are selected as address pointers.

Y register flag is set at "1" by the execution of Y register data setting instruction. And Y register flag is cleared at "0" by the execution of the instruction which does not set data to data memory. Further,while Y register flag is set at "1", it is kept in interrupt disabled (waiting) state.
Y register flag is reset at "0" by the initialize operation.

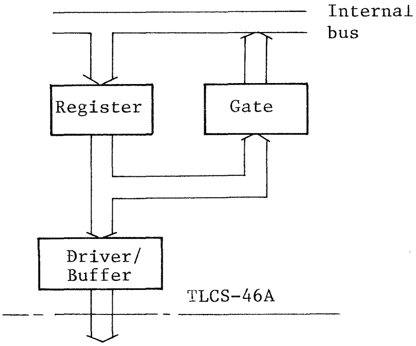
(5) Port (P), Port Register (Register)

TCP4620AP/TCP4630AP is provided with a total 34 ports; input port, output port, input output port, and input/output port.

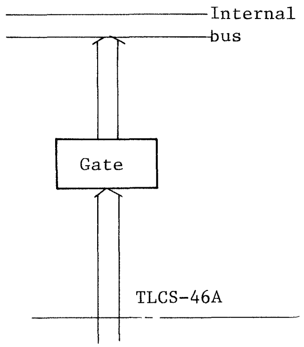
When a port is used as input, it is processed as non-latch input, and therefore, it is necessary to hold external input data till completion of read-in.

When a port is used as output, output data is set in the port register by an instruction and contents of this prot register are output. On some ports it is possible to read contents of the port registers by an instruction.

Further, the port register is reset at "0" by initialize operation. In addition, the input level is compatible with CMOS, and the output level is compatible with CMOS/TTL.



General purpose output port configuration



General purpose input port configuration

Port Configuration

Port Name	Symbol	Pin Name	Register Number	Port Function	Remarks
Input Out-Put port	P_0	$P_{03} - P_{00}$	0	Input out-put port	Control designating of I/O by D flag. This port can be specified as an exclusive use output port by mask option.

Port Name	Symbol	Pin Name	Register Number	Port Function	Remarks
Output Port	P_1	$P_{13}-P_{10}$	1	Output port	General purpose output port. Contents of registers can be read by an instruction.
I/O Port	P_2	$P_{23}-P_{20}$	2	I/O port	Input/output is specified by mask option. Contents of registers can be read by instruction at the output mode.
I/O Port	P_4	$P_{43}-P_{40}$	4	I/O port	
Key Input Port	PI_5	$PI_{53}-PI_{50}$	5	Input port	General purpose input port with a 150KΩ(Typical) resistor. (Pull-up/down can be specified by mask options).
Digit Output Port	PO_5	$PO_{54}-PO_{50}$		Output port	Digit output port for dynamic display. Can be specified as a general purpose output port by mask options.
Input Port	PI_6	PI_{60}	6	Input port	General purpose input port.
Segment Output Port	PO_6	$PO_{67}-PO_{60}$		Output port	8-segment output port for dynamic display. Can be specified as a general purpose output port by mask options.

Generalization of Segment Output Port (Sepcified by Mask Options)

Port Name	Symbol	Pin Name	Register Number	Port Function	Remarks
Output Port	PO_6	PO_{63} - PO_{60}	6	Output port	General purpose output port.
Output Port	PO_7	PO_{73} - PO_{70}	7	Output port	General purpose output port.

① Port 0 (Po)

Port 0 is a 4-bit general purpose input/output port. This port is selected when Register 0 is specified in an instruction.

Input/output designation is made by D flag of the status register. When D flag is "0", this port becomes the input and when D flag is "1", it becomes the output and output the content of Register 0.

Port 0 can be used as an exclusive output port by mask options.

② Port 1 (P_1)

Port 1 is a 4-bit general purpose output port. This port is selected when Register 1 is specified in an instruction and the content of Register 1 is output.

Further, the content of Register 1 can be read by an instruction.

③ Port 2 (P_2), Port 4 (P_4)

Port 2 and Port 4 are 4-bit input/output ports that can be specified as either input or output by mask options. When Register 2 and Register 4 are specified in an instruction, Port 2 and Port 4 are selected, respectively.

In specifying input or output in a mask option, the following combination is possible.

(a) p2/F/, P4/F/-Port 2 and Port 4 as output.

(b) P2/F/, P4/3/-Port 2 as output.
High order 2 bits (P_{43},P_{42}) of Port 4 as input, low order 2 bits (P41, P40) as output.

(c) P2/0/, P4/F/-Port 2 as input.
Port 4 as output.

(d) Specify P2/θ/, P4/3/-Port 2 as input.
High order 2 bits (P43,P42) of Port 4 as input low order 2 bits (P41, P40) of Port 4 as output.

(e) Specify P2/0/, P4/0/-Port 2 and P4 as input.

Further, in case of input bits these ports operate as non-latch inputs, and in case of output bits they operate functionally same as in Port 1.

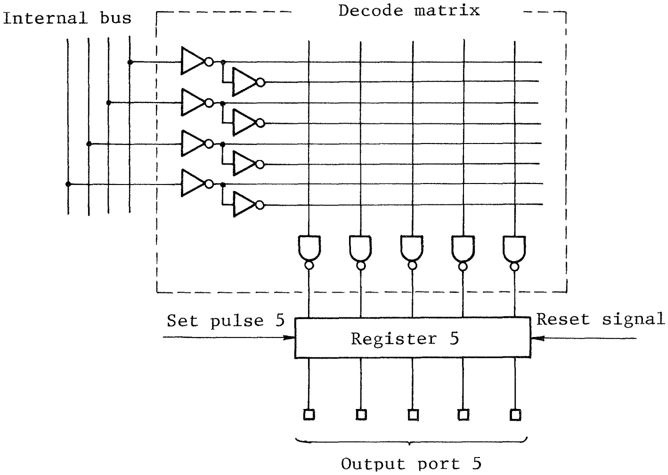
④ Output Port 5 (P05)

Output Port 5 is a 5-bit output port with the purpose of digit data output in dynamic display.

When an instruction for writing data in Register 5 is executed, 4-bit data on the internal bus is converted into 5-bit data by the decoder matrix, this data is written into the 5-bit Register 5 and further, output to Port 5.

User is able to specify the content of the decode matrix optionally by mask options.

Further, the content of Register 5 cannot be read by an instruction.



⑤ Input Port 5 (PI_5)

Input Port 5 is a 4-bit input port. This port is selected by a Read Register 5 instruction.

Input Port 5 is equipped with a 150kΩ(Typ.) input resistor, and the pull-up/down is specified by mask options.

⑥ Output Port 6 (PO_6), Output Port 7 (PO_7)

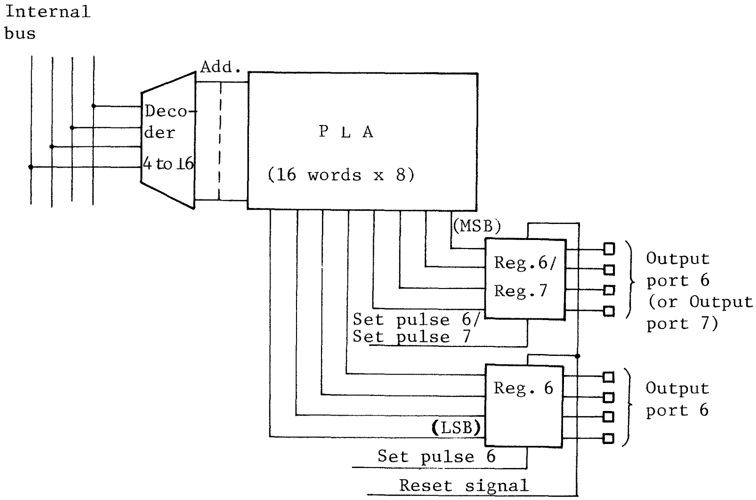
Output Port 6 is a 8-bit output port for segment data output in dynamic display.

When an instruction for writing data in Register 6 is executed, a output data is read from 16 words x 8 bits PLA with a 4-bit data on the internal bus used as an address, this output data is written into the 8-bit Register 6, and output to Port 6.

User is able to specify the content of PLA optionally by mask options.

Further, it is possible to specify and use output Port 6 as two 4-bit general purpose output ports by mask options. In this case, the 8-bit output port is split into high order 4 bits and low order 4 bits, and the high order 4 bits are assigned to output Port 7 (Register 7) and low order 4 bits to Output Port 6 (Register 6).

The contents of Register 6 (and Register 7) cannot be read by an instruction.



⑦ Input Port 6 (PI_6), Counter Buffer

Input Port 6 is the input port of 1 terminal (PI_{60}), but internally it is treated as a 4-bit data in combination with 3-bit data that are output from the divider (refer to Item for Divider).

The read circuit which reads this 4-bit data is called Counter Buffer/Input Port 6 or Counter Buffer simply. This counter buffer is selected by a Read Register 6 instruction and processed as 4-bit parallel non-latch input as in other input ports.

The bit configuration of the counter buffer is such that external input from PI_{60} terminal is connected to LSB side 1 bit and output from the divider is connected to MSB side 3 bits.

Further, the output stage of the divider to be connected is specified by mask options, but is fixed by a combination of an oscillator to be used and internal basic frequency. (Refer to Item for Divider.)

A Schmitt circuit and a timing shaping shift register are connected to PI_{60} terminal, which therefore cannot be operated at frequency above the internal basic clock frequency. The internal signal from the input terminal is subject to a time delay of maximum.

$$\frac{3}{\text{Internal basic frequency x 2}} \quad \text{(sec.)}$$

(6) Internal Bus (IBUS)

The internal bus consists of 4 bits, connects various registers and blocks such as the accumulator, status register, data memory, H register, L register, Y register, port register, ALU, etc., and data to be processed and data of process result are transfered through the internal bus.

(7) Data Memory (RAM)

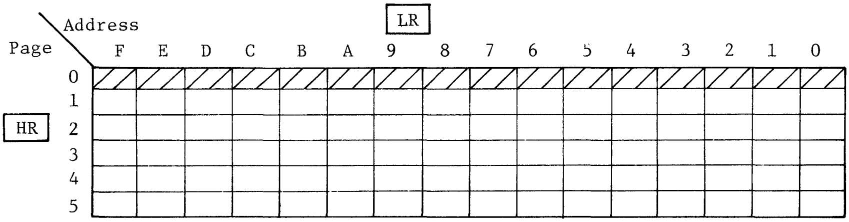
TLCS-46A Family has the following internal data memories in order to store user's process data.

Type	Capacity
TCP4620AP	96 words x 4 bits
TCP4630AP	160 words x 4 bits
(TCP4600AC	160 words x 4 bits)

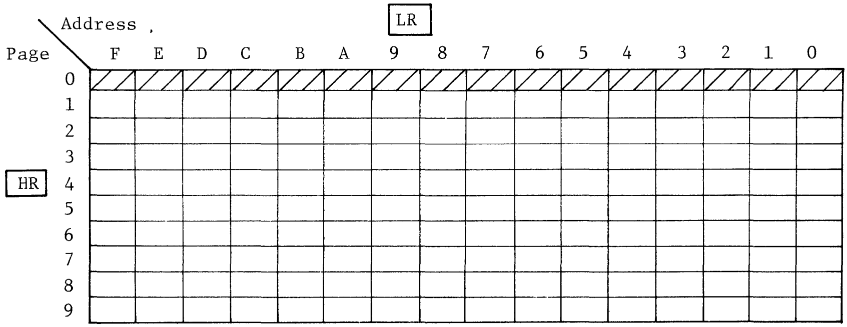
The data memory consists of the static memory cells.

Addressing of the data memory is executed by contents of H register/L register or Y register.

TCP4620AP



TCP4630AP/ (TCP4600AC)



▨ portion can be referred by Y register.

Addressing of Data Memory

(8) Program Memory (ROM)

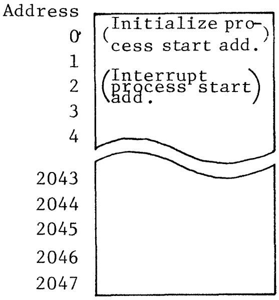
TLCS-46A Family has the following internal program memory in order to store user's process programs.

Type	Capacity
TCP4620AP	2048 words x 8 bits
TCP4630AP	3072 words x 8 bits
(TCP4600AC	No internal ROM)

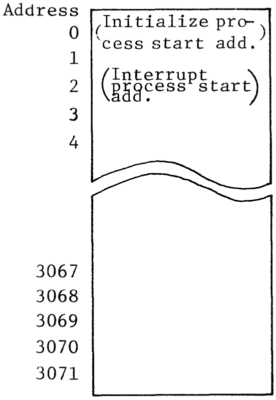
The program memory addressing is executed by the program counter (PC). Inherent meaning is given to Addresses 0 and 2 by the hardware, normal user process programs are place in Address 4 and subsequent addresses.

Further, the program counter consists of 12 bits and is capable of directly specifying addresses up to 4095, however, data in an undefined region having no program memory loaded becomes unstable.

TCP4620AP


Address
0
1
2
3
4
2043
2044
2045
2046
2047
(Initialize process start add.)
(Interrupt process start add.)

TCP4630AP


Address
0
1
2
3
4
3067
3068
3069
3070
3071
(Initialize process start add.)
(Interrupt process start add.)

(9) Program Counter (PC)

The program counter consists of a 12-bit binary counter, adds count increment at every instruction fetch, and makes the addressing of a program memory in which an instruction to be executed.

When a branch instruction, an interrupt operation, or subroutine instruction is executed, the contents of the program counter are changed.

Addresses 0 and 2 are compulsorily set in the program counter by a initialize signal and an interrupt request signal, respectively.

(10) Stack (STK), Stack Flag (FSTK)

The stack is used for temporary evacuation of the contents of the program counter when an interrupt request is accepted or a subroutine is to be executed.

The stack flag is a flag indicating whether the contents of the program counter have been evacuated in the stack.

When the contents of the program counter are pushed down in the stack, the stack flag is set at "1" (FSTK=1). And when the contents of the stack are popped up by a RTN instruction and returned to the original program flow, the stack flag is set at "0" (FSTK=0). At time of FSTK=1, the interrupt request becomes the disable (waiting) state.

Further, the stack flag is set at FSTK=1 by the initialize operation.

(11) Instruction Register (IR), Instruction Decoder (IDEC)

The instruction register latches a 8-bit data from the program memory and outputs it to the instruction decoder. (Program memory data may be used for direct internal control.)

The instruction decoder receives data from the instruction register and outputs a control signal required for processing.

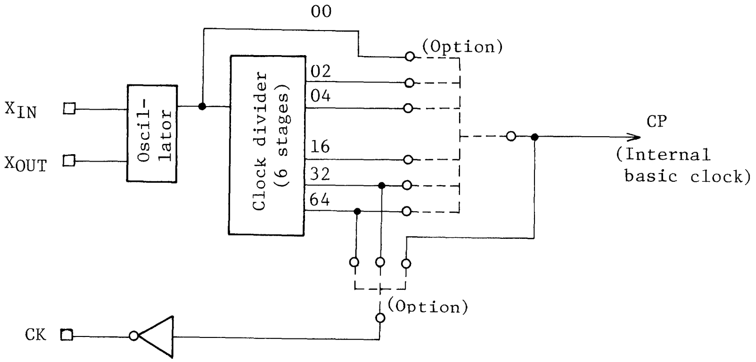
(12) Clock Generator (OSC)

TCP4620AP/TCP4630AP has a internal clock generator. By externally installing a crystal oscillator/ceramic oscillator or a LC oscillator, required clock is easily obtained. Furhter, clock can be supplied externally, however, in this case, the clock is input through X_{IN} terminal and X_{OUT} terminal is kept open. While TCP4620AP/TCP4630AP starts by $\overline{\text{RESET}}$, the clock should be always continuously supplied.

Oscillation frequency shall be selected from several frequencies ranging from 20KHz to 4.2MHz by mask options.
The clock generator has a internal 6-stage clock divider and specifies the optimum divide ratio to obtain internal basic clock (CP) on the basis of oscillation frequency by mask options.

The clock generator is provided with the output terminal CK for external timing, and output frequency is specified by mask options.

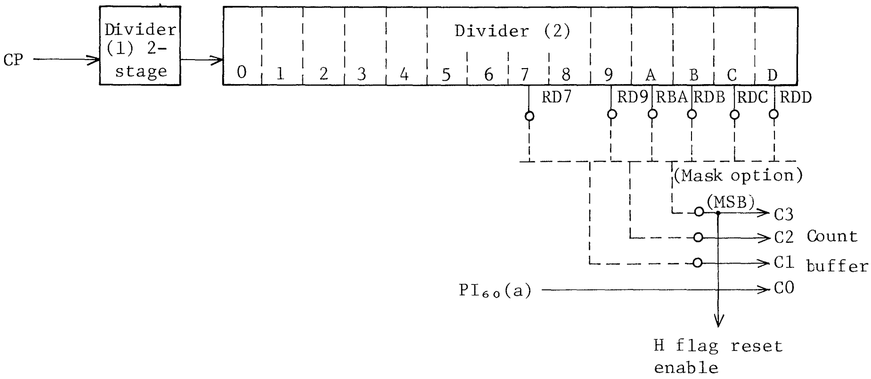
Further, a divide ratio for obtaining internal basic clock and external timing output frequency specified by mask options (refer to Item for Divider) are determined by an oscillator (oscillation frequency) to be used and internal basic clock frequency.



Clock Generator Circuit Configuration

(13) Divider

The divider is a binary 16-stage divider provided for a clock counter, timer, etc., and makes the count of internal basic clock (CP).



Divider Configuration

Of output from the 16-stage divider, 3 bits are connected to the counter buffer by mask options and can be read by a program as the counter buffer contents.

The relation between output 3 bits that can be specified by mask options and the divider output stage is determined by an oscillator (oscillation frequency) to be used and internal basic frequency as shown below.

Oscillator		32.76K X tal	100K X tal	400 - 500K Ceramic /IFT	400K ceramic /IFT	3 - 4.2M X tal	3 - 4.2M X tal	
Internal basic clock	CP	00	00	04	02	64	32	
External timing	CK	32	64	CP	CP	CP	CP	
Counter buffer	C1	RD9	RDB	RDB	RD7	RDA	RD7	
	C2	RDA	RDC	RDC	RDA	RDB	RDA	
	C3	RDB	RDD	RDD	RDD	RDC	RDD	

Further, the most significant bit (C3) of the counter buffer is used as a hold release signal (H flag reset enable signal of the status register) when the hold operation is used.

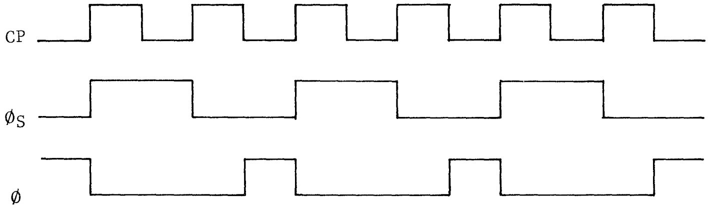
In addition, only when a 100K Xtal is used as an oscillator, the divider (2) is reset when the count value of the divider (2) reaches "12500" and the count value is reset to zero and then the count is resumed.

(14) Timing Generator (TG)

The timing generator consists of the internal timing signal generation circuit, initialize control circuit and interrupt control circuit.

① Internal Timing Signal Generation Circuit

This circuit receives the internal basic clock (CP) from the clock generator and generates two internal basic timing signals ϕ and ϕ_S.



② Initialize Control Circuit

This circuit shapes the timing of external $\overline{\text{RESET}}$ signal and generates an internal initialize signal. This initialize signal ia used for internal initialization. (For details of the initialize operation refer to "Operation Description".)

③ Interrupt Control Circuit

This circuit shapes the timing of external interrupt request signal $\overline{\text{INT}}$, store it in the internal interrupt latch, and makes a judgement as to if the internal state is interrupt enable. When it is in evable state, this circuit generates an interrupt request signal internally and starts an interrupt operation, but if it is in disable state, controls the interrupt request to wait till it becomes enables.

(For details of the interrupt operation refer to "Operation Description".)

[MACHINE INSTRUCTION]

1. Symbol Meaning

The explanation of symbols used for the following description is collected in the following table.

Symbol	Meaning
PC	Program counter
AC	Accumulator
ST	Status register
M	Data memory
LR	L register
HR	H register
YR	Y register
EYR	Y register flag
C	C flag of status register
Z	Z flag of status register
R	Port register, Field designating register under instructions
I	Value of immediate data field
G	Field designating bit position under bit processing instructions
AP	Address branched by branch instructions
AP_H	Field showing higher order 4 bits of AP
AP_M	Field showing intermediate order 4 bits of AP
AP_L	Field showing lower order 4 bits of AP
(A)	Content of A
$(\overline{A})$	Values that contents of A are inverted every bit
+	Binary addition
-	Binary subtraction
$\wedge$	Logical AND for every bit
$\vee$	Logical OR for every bit

Symbol	Meaning
∀	Exclusive OR for every bit
←	Value of left side is equalized to that of right side
A<a : b>	Content from b bitth to a bitth of A
°	Value is made into one by connecting two fields
*	Value is updated by operation result
-	Value is not changed by operation result
Push PC	Content of program counter is saved in stack
Pop PC	Content of stack is returned to program counter
Null	No operation

2. Instruction Description

The machine instruction of TLCS-46A consists of 52 kinds of instruction, being divided roughly into 6 kinds as follows:

Machine Instruction (52 kinds).

Data Transfer Instruction	(19 kinds)
Operation Instruction	(18 kinds)
Bit manipulation Instruction	(8 kinds)
Branch Instruction	(3 kinds)
Interrupt/Subroutine Instruction	(3 kinds)
Other Instruction	(1 kinds)

In this paragraph, the function of each instruction is described according to the following instruction description description format.

<table>
<tr><td rowspan="4">Assembler mnemonic</td><td colspan="4">Instruction name</td></tr>
<tr><td colspan="3">Machine code</td><td>Execution cycle</td></tr>
<tr><td>Function</td><td>Z flag after execution</td><td>C flag after execution</td><td>EYR after execution</td></tr>
<tr><td colspan="4">Operation description</td></tr>
</table>

Instruction Description Format

(1) Data Transfer Instruction

The data transfer instruction mainly provides the data transfer among accumulator, status register, memory (RAM), and registers.

<table>
<tr><td rowspan="4">LAR R</td><td colspan="4">Load Accumulator from Regsiter</td></tr>
<tr><td colspan="3">7 0
0 ¦ 0 ¦ 0 ¦ 1 ¦ 0 ¦ R   0 ≤ R ≤ 7</td><td>1</td></tr>
<tr><td>(AC) ← (R)</td><td>-</td><td>-</td><td>0</td></tr>
<tr><td colspan="4">The content of register or input port designated by R field is loaded in accumulator.</td></tr>
</table>

LTR R	Load Accumulator from Register and Test			
	7 0 0 0 0 1 1 R 0 ≤ R ≤ 7			1
	(AC) ← (R)	*	-	0
	The content of resister or input port designated by R field is loaded in accumulator. After execution Z flag is updated.			
L A M	Load Accumulator from Memory			
	7 0 0 0 0 0 0 1 0 0			1
	(AC) ← (M)	-	-	0
	The content of memory is loaded in accumulator Note 1			
L T M	Load Accumulator from Memory and Test			
	7 0 1 0 0 0 0 1 1 1			1
	(AC) ← (M)	*	-	0
	The content of memory is loaded in accumulator. After execution Z flag is updated. Note 1			

L S M	Load Status-register from Memory			
	7 0 0 0 0 0 0 1 0 1			1
	(ST) ← (M)	Note 3 *	Note 3 *	0
	The content of memory is loaded in status register. Note 1			
L A L	Load Accumulator from L-register			
	7 0 0 0 0 0 0 1 1 0			1
	(AC) ← (LR)	-	-	0
	The content of L register is loaded in accumulator.			
L A H	Load Accumulator from H-register			
	7 0 0 0 0 0 0 1 1 1			1
	(AC) ← (HR)	-	-	0
	The content of H register is loaded in accumulator.			
LAI I	Load Accumulator Immediate			
	7 0 0 1 0 0 I 0 ≤ I ≤ 15			1
	(AC) ← I	-	-	0
	The value of I field is set to accumulator.			

LLI I	Load L-register Immediate			
	7 0 0 ¦ 1 ¦ 1 ¦ 0 ¦ I 0 ≤ I ≤ 15			1
	(LR) ← I	-	-	0
	The value of I field is set to L register.			
LHI I	Load H-register Immediate			
	7 0 0 ¦ 1 ¦ 1 ¦ 1 ¦ I 0 ≤ I ≤ 15			1
	(HR) ← I	-	-	0
	The value of I field is set to H register.			
LYI I	Load Y-register Immediate			
	7 0 0 ¦ 1 ¦ 0 ¦ 1 ¦ I 0 ≤ I ≤ 15			1
	(YR) ← I	-	-	1
	The value of I field is set to Y register. After execution, Y register flag is set to "1".			
SAR R	Store Accumulator to Register			
	7 0 0 ¦ 0 ¦ 1 ¦ 0 ¦ 1 ¦ R 0 ≤ R ≤ 7			1
	(R) ← (AC)	-	-	0
	The content of accumulator is stored in the register designated by R field.			

S A M	Store Accumulator to Memory			
	7 0 0 0 0 0 1 1 0 0			1
	(M) ← (AC)	-	-	-
	The content of accumulator is stored in memroy. Note 1, Note 2			
S S M	Store Status-register to Memory			
	7 0 0 0 0 0 1 1 0 1			1
	(M) ← (ST)	-	-	-
	The content of status register is stored in memory Note 1, Note 2			
S A L	Store Accumulator to L-register			
	7 0 0 0 0 0 1 1 1 0			1
	(LR) ← (AC)	-	-	0
	The content of accumulator is stored in L register			
S A H	Store Accumulator to H-register			
	7 0 0 0 0 0 1 1 1 1			1
	(HR) ← (AC)	-	-	0
	The content of accumulator is stored in H register.			

S A Y	Store Accumulator to Y-register			
	7 0 0 0 0 0 1 0 0 0			1
	(YR) ← (AC)	-	-	1
	The content of accumulator is stored in Y register. After execution, Y register flag is set to "1".			
SYR R	Store Y-register to Register			
	7 0 0 0 1 1 0 R 4 ≤ R ≤ 7			1
	(R) ← (YR)	-	-	0
	The content of Y register is stored in register designated by R field.			
CLR R	Clear Register			
	7 0 0 0 1 0 0 R 0 ≤ R ≤ 7			1
	(R) ← 0 (0 ≦ R ≦ 4)	-	-	0
	"0" is written in the register designated by R field (0≦R≦4). For 5≦R≦7, decode matrix output or PLA output corresponding to data "0" is written in the register.			

(2) Operation Instruction

Operation instruction has 2 operand instruction and 1 operand instruction, and performs arithmetic operation and logical operation.

ADI I	Add Accumulator Immediate			
	7 0 1 0 0 1 I 0 ≤ I ≤ 15			1
	(AC) ← (AC) + I	*	*	0
	The value of I field is added to the content of accumulator. After execution of instruction, Z flag and C flag are updated.			
A D A	Add and Store to Accumulator			
	7 0 1 0 0 0 0 0 0 0			1
	(AC) ← (M) + (AC)	*	*	0
	The content of accumulator is added to the content of memory, and the result is loaded in accumulator. After execution of instruction, Z flag and C flag are updated.			
A D M	Add and Store to Memory			
	7 0 1 0 0 0 1 0 0 0			1
	(M) ← (M) + (AC)	*	*	-
	The content of accumulator is added to the content of memory, and the result is stored in memory. After execution of instruction, Z flag and C flag are updated. Note 1, Note 2			
A C A	Add with Carry and Store to Accumulator			
	7 0 1 0 0 0 0 0 0 1			1
	(AC) ← (M) + (AC) + (C)	*	*	0
	The content of accumulator and the content of C flag are added to the content of memory, and the result is loaded in accumulator. After execution of instruction,Z flag and C flag are updated. Note 1			

A C M	Add with Carry and Store to Memory			
	7 0 1 0 0 0 1 0 0 1			1
	(M) ← (M) + (AC) + (C)	*	*	-
	The content of accumulator and the content of C flag are added to the content of memory, and the result is sotred in memory. After execution of instruction, Z flag and C flag are updated. Note 1, Note 2			
S U A	Subtract and Store to Accumulator			
	7 0 1 0 0 0 0 0 1 0			1
	(AC) ← (M) - (AC)	*	*	0
	The content of accumulator is subtracted from the content of memory, and the result is loaded in accumulator. After execution of instruction, Z flag and C flag are updated. Note 1			
S U M	Subtract and Store to Memory			
	7 0 1 0 0 0 1 0 1 0			1
	(M) - (M) - (AC)	*	*	-
	The content of accumulator is subtracted from the content of memory, and the result is stored in memory. After execution of instruction, Z flag and C flag are updated. Note 1, Note 2			
O R A	Or and Store to Accumulator			
	7 0 1 0 0 0 0 1 0 0			1
	(AC) ← (M) ∨ (AC)	*	-	0
	The logical sum of every bit of the content of memory and the content of accumulator is loaded in accumulator. After execution of instruction, Z flag is updated. Note 1			

O R M	Or and Store to Memory			
	7 0 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0			1
	(M) ← (M) ∨ (AC)	*	-	-
	The logical sum of every bit of the content of memory and the content of accumulator is stored in memory. After the execution of instruction, Z flag is updated. Note 1, Note 2			
E O A	Exclusive-or and Store to Accumulator			
	7 0 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1			1
	(AC) ← (M) ⊻ (AC)	*	-	0
	The exclusive OR of every bit of the content of memory and the content of accumulator is loaded in accumulator. After execution of instruction, Z flag is updated. Note 1			
E O M	Exclusive-or and Store to Memory			
	7 0 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1			1
	(M) ← (M) ⊻ (AC)	*	-	-
	The exclusive OR of every bit of the content of memory and the content of accumulator is stored in memory. After execution of instruction, Z flag is updated. Note 1, Note 2			
C M A	Complement Accumulator			
	7 0 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0			1
	(AC) ← $\overline{(AC)}$	*	-	0
	The content of accumulator is inverted every bit. After exectuion of instruction, Z flag is updated.			

C M M	Complement Accumulator and Store to Memory			
	7 0 1 0 0 0 1 1 1 0			1
	(M) ← $\overline{(AC)}$	*	-	-
	The content of accumulator is inverted every bit, and the result is stored in memory. After execution of instruction, Z flag is updated. Note 1, Note 2			
I C A	Load Accumulator and Increment			
	7 0 1 0 0 0 0 0 1 1			1
	(AC) ← (M) + 1	*	*	0
	"1" is added to the content of memory, and the result is loaded in accumulator. After execution of instruction, Z flag and C flag are updated. Note 1.			
I C M	Increment Memory			
	7 0 1 0 0 0 1 0 1 1			1
	(M) ← (M) + 1	*	*	-
	"1" is added to the content of memory. After execution of instruction, Z flag and C flag are updated. Note 1, Note 2			
I C L	Increment L-register			
	7 0 0 0 0 0 0 0 1 0			1
	(LR) ← (LR) + 1	-	-	0
	"1" is added to the content of L register.			

I C H	Increment H-register			
	7 0 0 0 0 0 0 0 1 1			1
	(HR) ← (HR) + 1	-	-	0
	"1" is added to the content of H register.			
T S M	Test Memory			
	7 0 1 0 0 0 1 1 1 1			1
	(M) ← (M)	*	-	-
	Z flag is set according to the content of memory. Note 1, Note 2			

(3) Bit Manipulation Instruction

The bit manipulation instruction is performed to each bit of accmulator or status register.

TBA G	Test a bit for Accumulator			
	7 0 1 0 1 0 0 0 G $0 \leq G \leq 3$			1
	Null ← (AC) $\wedge$ 2^G	*	-	0
	The Z flag is updated, according to the bit of accumulator designated by G field.			

TBS G	Tast a bit for Status-register			
	7 0 1 | 0 | 1 | 1 | 0 | 0 | G $0 \le G \le 3$			1
	Null ← (ST) ∧ 2^G	*	-	0
	The Z flag is updated, according to the bit of status register designated by G field.			
SBA G	Set a bit for Accumulator			
	7 0 1 | 0 | 1 | 0 | 0 | 1 | G $0 \le G \le 3$			1
	(AC) ← (AC) ∨ 2^G	-	-	0
	The bit of accumulator designated by G field is set to "1".			
SBS G	Set a bit for Status-register			
	7 0 1 | 0 | 1 | 1 | 0 | 1 | G $0 \le G \le 3$			1
	(ST) ← (ST) ∨ 2^G	Note 3 *	Note 3 *	0
	The bit of status-register designated by G field is set to "1".			
CBA G	Clear a bit for Accumulator			
	7 0 1 | 0 | 1 | 0 | 1 | 0 | G $0 \le G \le 3$			1
	(AC) ← (AC) ∧ $\overline{2^G}$	-	-	0
	The bit of accumulator designated by G field is cleared to "0".			

CBS G	Clear a bit for Status register			
	7 0 1 0 1 1 1 0 G $0 \leq G \leq 3$			1
	(ST) ← (ST) ∧ $\overline{2^G}$	Note 3 *	Note 3 *	0
	The bit of status register designated by G field is cleared to "0".			
IBA G	Invert a bit for Accumulator			
	7 0 1 0 1 0 1 1 G $0 \leq G \leq 3$			1
	(AC) ← (AC) $\veebar$ 2^G	-	-	0
	The bit of accumulator designated by G field is inverted.			
IBS G	Invert a bit for Status register			
	7 0 1 0 1 1 1 1 G $0 \leq G \leq 3$			1
	(ST) ← (ST) $\veebar$ 2^G	Note 3 *	Note 3 *	0
	The bit of status register designated by G field is inverted.			

(4) Branch nstruction

The branch instruction performs unconditional or conditional branch.

JMP AP	Jump			
	7 0 15 8 1 1 1 0 | AP_L | AP_H | AP_M 0 ≤ AP ≤ 4095			2
	(PC) ← AP	-	-	0
	The value of AP field is set to program counter. After execution of instruction, therefore, program sequence changes to the AP address.			
BCS AP	Branch on Condition Set			
	7 0 15 8 1 1 0 0 | AP_L | AP_H | AP_M 0 ≤ AP ≤ 4095			2
	If Z = 1 then (PC) ← AP, else Null	-	-	0
	The content of Z flag is "1",the value of AP field is set to program counter. After execution of instruction, therefore, program sequence changes to the AP address. If the content of Z flag is "0", the program sequence follows the next address without any operation.			
BCC AP	Branch on Condition Clear			
	7 0 15 8 1 1 0 1 | AP_L | AP_H | AP_M 0 ≤ AP ≤ 4095			2
	If Z = 0 then (PC) ← AP, else Null	-	-	0
	The content of Z flag is "0",the value of AP field is set to program counter. After execution of instruction,therefore,program sequence changes to the AP address. If the content of Z flag is "1", the program sequence follows the next address without any operation.			

(5) Interrupt/Subroutine Instruction

The interrupt/subroutine instruction is used for performing a call of subroutine and a return from interrupt routine or subroutine.

In addition NOP instruction is available as one other instruction.

CAL AP	Call Subroutine			
	7 015 8 1 1 1 1 AP_L AP_H AP_M 0 ≤ AP ≤ 4095			2
	push (PC), and (PC) ← AP	-	-	0
	The content (the next address) of program counter is stored in the stack, and the value of AP field is set to the program counter. The interrupt is placed in disable (queued) state during execution of subroutine.			
R T N	Return			
	7 0 0 0 0 0 0 0 0 1			2
	POP (PC)	-	-	0
	The content of the stack is restored to program counter. After execution of instruction, the interrupt is placed in enable state.			
J A C	Jump by Accumulator			
	7 0 0 0 0 0 1 0 0 1			2
	(PC)←(PC)+1,and(PC)←(PC)<11:4>∘(AC)	-	-	0
	The content (address where JAC instruction is sroted,plus 2) of the incremented program counter is taken as the higher 8 bits,and the content of AC is taken as the lower 4 bits; then, the value of 12 bits in total that these contents are connected is set to the program counter.			
N O P	No Operation			
	7 0 0 0 0 0 0 0 0 0			1
	Null	-	-	0
	This is an instruction by which nothing is executed. However, Y register flag is reset to "0".			

Note 1: The address of data memory becomes as follows;

EYR = 0 , (HR∘LR)
EYR = 1 , (0∘YR).

Note 2: Care should be taken to the tact that, even after this instruction is ececuted, Y register flag does not change.

Note 3: A flag may be updated as a result of processing.

3. LIST OF INSTRUCTIONS

	Mne-monic	Machine code		Operation	Update of flag	Remarks
		Hexa-decimal	Binary			
Data Transfer Instruction	LAR	$1\underline{i}$	00010iii	(AC)←(Register iii)		
	LTR	$1\underline{i}$	00011iii	(AC)←(Register iii)	Z	
	LAM	04	00000100	(AC)←(M)		
	LTM	87	10000111	(AC)←(M)	Z	
	LSM	05	00000101	(ST)←(M)	ZC	
	LAL	06	00000110	(AC)←(LR)		
	LAH	07	00000111	(AC)←(HR)		
	LAI	4i	0100iiii	(AC)←iiii		
	LLI	6i	0110iiii	(LR)←iiii		
	LHI	7i	0111iiii	(HR)←iiii		
	LYI	5i	0101iiii	(YR)←iiii		EYR is set
	SAR	$2\underline{i}$	00101iii	(Register iii)←(AC)		
	SAM	0C	00001100	(M) ← (AC)		
	SSM	0D	00001101	(M) ← (ST)		
	SAL	0E	00001110	(LR)←(AC)		
	SAH	0F	00001111	(HR)←(AC)		
	SAY	08	00001000	(YR)←(AC)		EYR is set
	SYR	$3\underline{i}$	001101ii	(Register 1ii)←(YR)		
	CLR	$2\underline{\underline{i}}$	00100iii	(Register iii)←0		
Operation Instruction	ADI	9i	1001iiii	(AC)←(AC)+iiii	ZC	
	ADA	80	10000000	(AC)←(M)+(AC)	ZC	
	ADM	88	10001000	(M) ← (M)+(AC)	ZC	
	ACA	81	10000001	(AC)←(M)+(AC)+(C)	ZC	
	ACM	89	10001001	(M) ← (M)+(AC)+(C)	ZC	
	SUA	82	10000010	(AC)←(M)−(AC)	ZC	
	SUM	8A	10001010	(M) ← (M)−(AC)	ZC	
	ORA	84	10000100	(AC)←(M)∨(AC)	Z	
	ORM	8C	10001100	(M) ← (M)∨(AC)	Z	
	EOA	85	10000101	(AC)←(M)⊻(AC)	Z	
	EOM	8D	10001101	(M) ← (M)⊻(AC)	Z	
	CMA	86	10000110	(AC)←$\overline{(AC)}$	Z	
	CMM	8E	10001110	(M) ← $\overline{(AC)}$	Z	
	ICA	83	10000011	(AC)←(M)+1	ZC	
	ICM	8B	10001011	(M) ← (M)+1	ZC	
	ICL	02	00000010	(LR)←(LR)+1		
	ICH	03	00000011	(HR)←(HR)+1		
	TSM	8F	10001111	(M)←(M), if (M)=0 then Z←1 else Z←0	Z	

	Mne-monic	Machine code		Operation	Update of flag	Remarks
		Hexa-decimal	Binary			
Bit Manipulation Instruction	TBA	$\mathrm{A}\underline{\mathrm{i}}$	101000ii	if (AC)<ii>=0 then Z ← 1 eles Z ← 0	Z	
	TBS	$\mathrm{B}\underline{\mathrm{i}}$	101100ii	if (ST)<ii>=0 then Z ← 1 eles Z ← 0	Z	Asteris (*) denotes that, when
	SBA	$\mathrm{A}\underline{\mathrm{i}}$	101001ii	(AC)<ii>←1		Z flag and C flag
	SBS	$\mathrm{B}\underline{\mathrm{i}}$	101101ii	(ST)<ii>←1	**	are designated,
	CBA	$\mathrm{A}\underline{\mathrm{i}}$	101010ii	(AC)<ii>←0		the fags are
	CBS	$\mathrm{B}\underline{\mathrm{i}}$	101110ii	(ST)<ii>←0	**	updated.
	IBA	$\mathrm{A}\underline{\mathrm{i}}$	101011ii	(AC)<ii>←$\overline{\text{(AC)<ii>}}$		
	IBS	$\mathrm{B}\underline{\mathrm{i}}$	101111ii	(ST)<ii>←$\overline{\text{(ST)<ii>}}$	**	
Branch Instruction and Others	JMP	Ek ij	1110kkkk iiiijjjj	(PC)←iiiijjjjkkkk		2-byte,2-cycle instruction
	BCS	Ck ij	1100kkkk iiiijjjj	if Z=1 then (PC)←iiiijjjjkkkk		"
	BCC	Dk ij	1101kkkk iiiijjjj	if Z=0 then (PC)←iiiijjjjkkkk		"
	CAL	Fk ij	1111kkkk iiiijjjj	(PC)↓, (PC)←iiiijjjjkkkk		"
	RTN	01	00000001	(PC)↑		1-byte, 2-cycle instruction
	JAC	09	00001001	(PC)←((PC)+1)<11:4>∘(AC)		"
	NOP	00	00000000	No Operation		EYR is reset.

Note: Pay attention to the conversion of the underlined machine codes.

4. TLCS-46A Instruction map

L H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	*1 RTN	ICL	ICH	LAM	LSM	LAL	LAH	SAY	*1 JAC			SAM	SSM	SAL	SAH
1	LAR								LTR							
2	CLR								SAR							
3					SYR											
4	LAI															
5	LYI															
6	LLI															
7	LHI															
8	ADA	ACA	SUA	ICA	ORA	EOA	CMA	LTM	ADM	ACM	SUM	ICM	ORM	EOM	CMM	TSM
9	ADI															
A	TBA				SBA				CBA				IBA			
B	TBS				SBS				CBS				IBS			
C	BCS *2															
D	BCC *2															
E	JMP *2															
F	CAL *2															

*1 - 1 byte,2-cycle instruction
*2 - 2 byte,2-cycle instruction

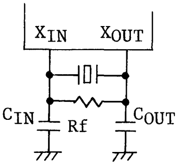
[OPERATION DESCRIPTION]

1. Basic Clock

The basic clock is used to generate the basic timing signals for the operation of TLCS-46A and as the clock for the divider.

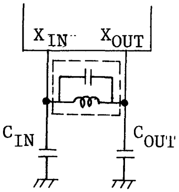
For the generation and supply of basic clock, there are following methods:

(1) Connecting crystal oscillator/ceramic oscillator



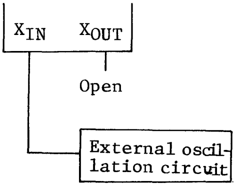
In case the crystal oscillator/ceramic oscillator is connected as shown in the left figure, it is possible to obtain the oscillation frequency characteristic to crystal oscillator/ceramic oscillator.

(2) Connecting IFT



In case the IFT is connected as shown in the left figure, it is possible to obtain the oscillation frequency characteristic to IFT.

(3) Suppling the external clock



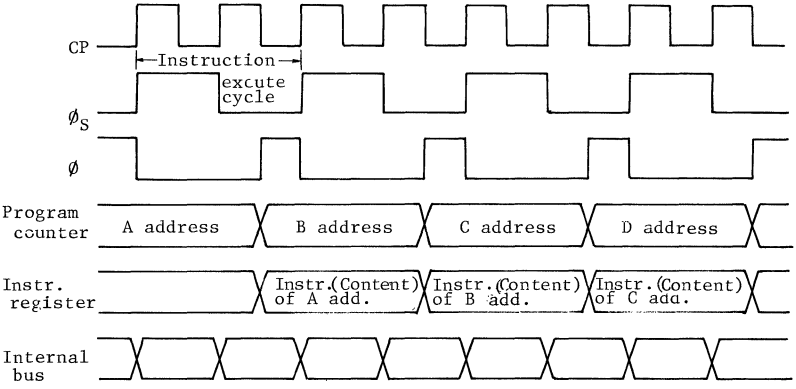
If the external oscillator is connected to X_{IN} terminal as shown in the left figure, it is possible to supply the clock from the external.

In this case, X_{OUT} terminal shall be kept open.

2. Instruction Execute Cycle

The instruction execute cycle consists of two internal timing signals, ϕs and ϕ, supplied by the timing generator.

ϕ_S is the signal expressing the instruction execute cycle, while ϕ is the signal which comes to the set timing of data memory, register, etc.

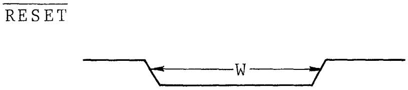

CP
Instruction execute cycle
ϕ_S
ϕ
Program counter
A address
B address
C address
D address
Instr. register
Instr.(Content) of A add.
Instr.(Content) of B add.
Instr.(Content) of C add.
Internal bus

Instruction Execution Timing Chart

3. Initialize operation (System initialization)

The initialize operation is carried out without fail after impressing power source on TLCS-46A, which is returned to the initial state in the interior.

By keeping the $\overline{\text{RESET}}$ terminal to "L" level, the internal initialize signal is formed, whereby the initialize operation of TLCS-46A is executed. At this time, more than 2 cycles are required for the time of "L" level.



W < 2 cycles	W ≥ 2 cycles
Whether or not initialize operation is executed is not difinite.	Initialize operation is executed.

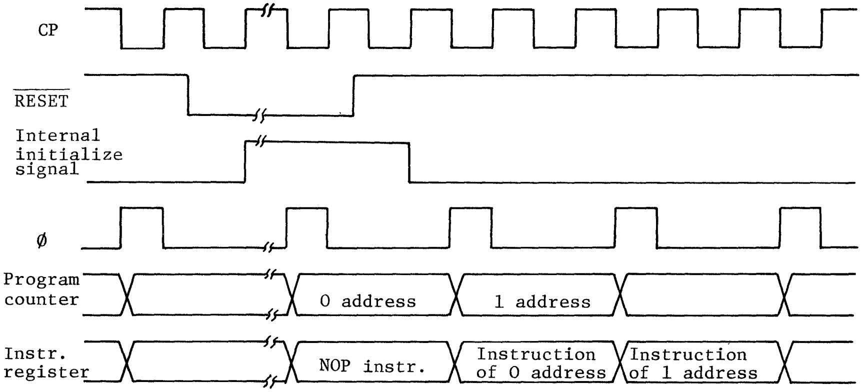
However, 1 cycle corresponds to 2 period of internal basic clock (CP).

By the initialize operation, the interior of TLCS-46A comes to the conditions as given below:

① Resetting H flag and D flag of status register to "0".
② Resetting the interrupt latch to "0".
③ Resetting the port register "0".
(All output pins come to "L" level.)
④ Setting the stack flag to "1".
⑤ Setting the program counter to 0 address.

There is no influence on accumulator, data memory, H register, L register, Y register, C flag and Z flag of status register, etc.

The program is executed from 0 address when $\overline{\text{RESET}}$ signal comes to "H" level and the internal initialize signal comes to "L" level.



Start from initialization

4. Interrupt Operation

TLCS-46A is provided with the function permitting the external interrupt operation. The interruption is an operation enabling other processing to be preferentially made by suspending the processing underway.
The request for interrupt operation is made by changing $\overline{\text{INT}}$ terminal from "H" level to "L" level and by holding the terminal on "L" level for more than 2 cycles.

There are several conditions enabling TLCS-46A to start the interrupt operation. If the interrupt operation request is made before the conditions are satisfied, the interrupt operation is in standby state to wait until the conditions are satisfied.
The conditions are as follows:

① Data memory is not made reference with Y register.
(EYR = 0)

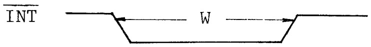
② Stack is empty. (FSTK = 0)

③ No hold operation (Interrupt request is ignored in hold operation.)

The external interrupt request signal is preserved in the internal interrupt latch, and the interrupt latch is cleared when the interrupt is accepted.

In case of repetitive interrupt request, it is required to make $\overline{\text{INT}}$ terminal "H" level (more than 2 cylces) and to make $\overline{\text{INT}}$ terminal "L" level. Repetitive interrupt request is not accepted when $\overline{\text{INT}}$ terminal is "L" level without making it "H" level.

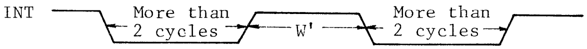
① Internal latch set timing



W < 2 cycles	W ≥ 2 cycles
Whether or not interruption latch is set is not difinite.	Interruption latch is set

However, 1 cycle corresponds to 2 period of internal basic clock (CP).

② Internal latch repetitive set timing



W' = 0	0 < W' < 2 cycles	W' ≥ 2 cycles
Interruption latch is not set again.	Whether or not interruption latch is set again is not difinite.	Interruption latch is set again

However, 1 cycle coresponds to 2 period of internal basic clock (CP).

When TLCS-46A accepts the interrupt, the following processings are made:

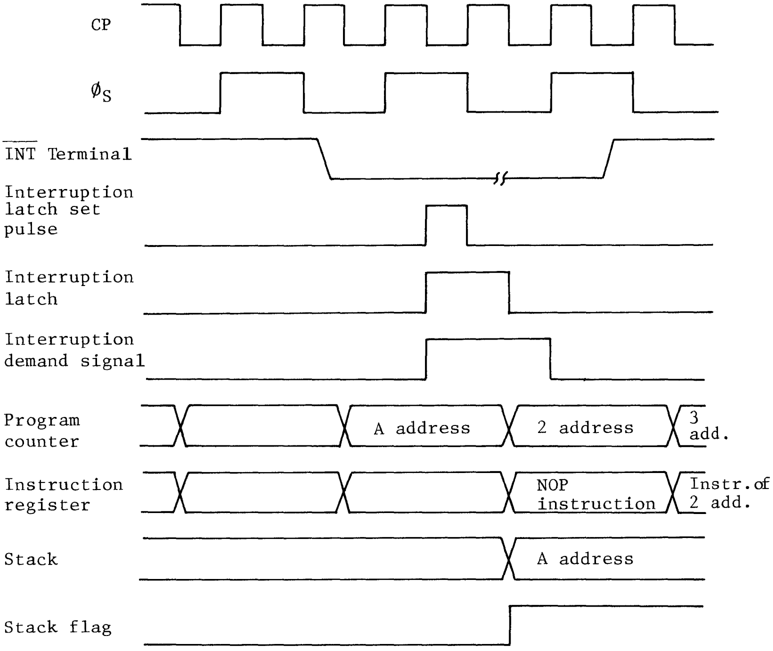
1 The content of program counter is stored in a save register (stack).

2 The entry address (address 2) is set in the program counter.

3 The interrupt latch is cleared.

Therefore, the interrupt service routine defines 2 address of program memory as the top address.

When TLCS-46A returns to the main program by completing the interrupt process, it follows the execution of RTN instruction.



Interrupt Acceptance Timing Chart

5. Hold Operation

In case the hold operation is designated by mask options, the operation state (normal/hold) of TLCS-46A is controlled by the H flag of status register.

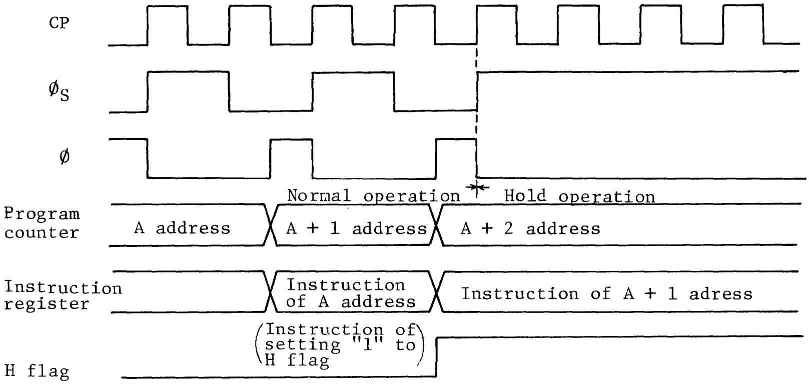
When each instruction of SBS, LBS and LSM is executed, and "1" is set to the H flag, the internal timing signals of ϕ_S and ϕ stop after the completion of the execution of the instruction, whereby the hold operation begins. At this time, the program counter, instruction register and other internal registers are held in the state which was before the hold operation, but the interrupt request under waiting is ignored because the interrupt latch is cleared.

As the clock generator does not stop, there is no influence on the internal basic clock and the count function of divider.

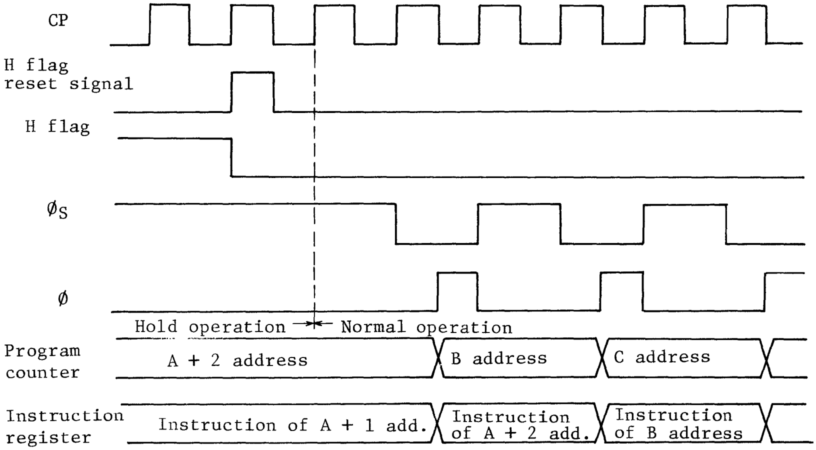
Restart from the hold operation is made by resetting H flag to "0" in hardware. For the signal for resetting H flag, the output of divider (the MSB input (C3) of counter buffer) is used, and the resetting is made by detecting the trailing edge of the bit input (C3).

At the input of divider is the internal clock (CP), restart is made at constant intervals.

When H flag is reset, the internal timing signals of ϕ_S and ϕ start, whereby the execution is countinued to start from the condition which is the same as that just before starting the hold operation.



Hold Operation Start Timing



Restart Timing from Hold Operation

6. Reference to Data Memory

There are two methods of making reference to data memory.

(1) The page of data memory is designated by H register, while the address in the page is designated by L register.
In the case of access to data memory, it is so arranged as to make access to the address designated by L register in the page designated by the contents of H register.
(1 page consists of 16 words.)

	7 4	3 0
Data Memory Address	HR	LR

(2) The address in 0 page is designated by Y register. In the case of access to data memory immediately after writing data in Y register by the instruction of LYI or SAY, the access is made to the address designated by the contents of Y register in 0 page.

	7 4	3 0
Data Memory Address	0 0 0 0	YR

At the time,of making reference to data memory, it depends on the value of internal Y register flag (EYR) to select H register/ L register or Y register for designating address. When EYR=0, the method (1) above is adopted. When EYR = 1, the method (2) above is adopted. When data are written in Y register by SAY instruction or LYI instruction, "1" is set to EYR. At the time of instruction execution when data are not set to data memory, EYR is reset to "0".

7. Input and Output Timing for Port

The input of data from the input port of TLCS-46A and the output of data to the output port are made by the data transfer instruction between accumulator and each port.

Port Input

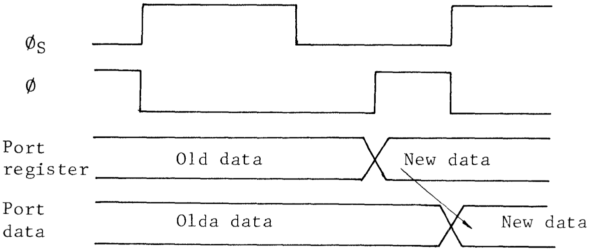
As the input ports of TLCS-46A are all non-latch inputs, the input data is required to be stable until the completion of reading.

At the instruction cycle of LAR or LTR instruction, input data is set to the accumulator.

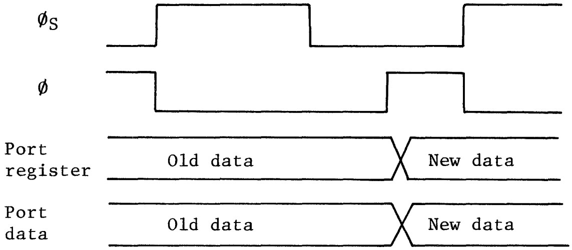
Port Output

The output ports of TLCS-46A output the content of port register. Data are written in the port register when SAR, SYR and CLR instructions are executed.

The digit output port (PO5) and segment output port (PO6) for display are different in output timing from other output ports.



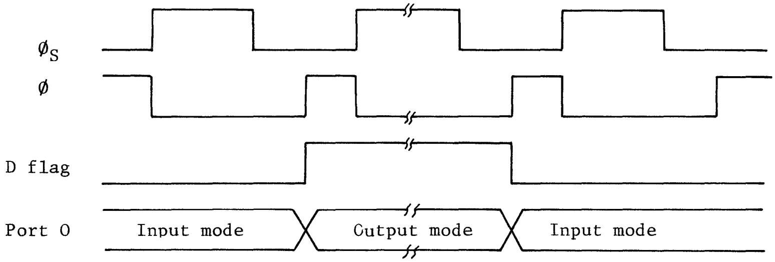
Output Timing (Port5, Port6)



Output Timing (except Port5, Port6)

8. Input/Output Switching of Port 0

The switching of input/output of port 0 is made by the D flag of status register. By each instruction of SBS, CBS, IBS and LSM, the value of D flag is changed, whereby the switching. of input/output is carried out.



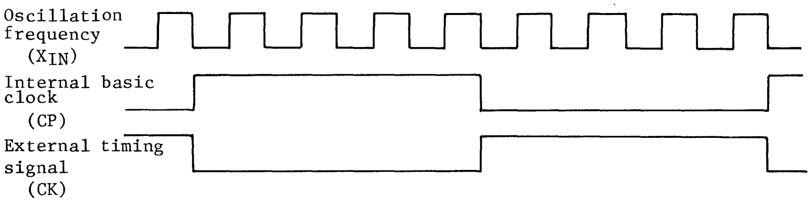
Input/Output Switching Timing of Port 0

9. Output Terminal for External Timing

CK terminal is equipped for taking out the output of the divider of clock generator.

The output frequency is designated by the mask options but is decided by the oscillator (oscillation frequency) used. The output frequency is either 1/32 or 1/64 of oscillation frequency or internal basic clock (CP).

The output signal on CK terminal is generated after inverting the phase of the output of divider.


Oscillation frequency (X_{IN})
Internal basic clock (CP)
External timing signal (CK)

Example of the internal basic clock and external timing waveform.

(The internal basic clock (CP) is 1/4 of the oscillation frequency. The external timing signal is CP.)

[Mask Options]

TCP4620AP/TCP4630AP have many kinds of mask options on the clock generator, input output ports, etc. so as to meet the extensive requirements from the users.

Option Name	Function
O S C	Prescribing the oscillator and oscillation frequency.
C P	Prescribing the dividing ratio for obtaining the internal basic clock.

Option Name	Function
CK	Prescribing the output frequency for the external timing.
COUNTER	Prescribing the reset timing of divider.
COUNTER BUFFER	Prescribing the input of counter buffer.
HOLD	Prescribing the function of H flag.
RSTH	Prescribing the restart signal from hold opration.
STD	Prescribing the function of port 0.
P 2	Prescribing the function of port 2.
P 4	Prescribing the function of port 4.
PI5	Prescribing the function of input resistance of port 5.
DECODER	Prescribing the contents of decode matrix.
PO6	Prescribing the function of port 6.
PLA	Prescribing the contents of PLA.

(Note) o The mask options of OSC-RSTH are decided by the oscillator (oscillation frequency) used.

o Use the designation paper for mask options which is attached to the ES order sheet.

o Submit to us the designation paper for mask options together with the ES order sheet by the data two weeks prior to the date of submitting the mask tape.

TCP4620AP
TCP4630AP

1. Designation of Mask Options

Option Item		Option Name		TYP001	TYP002	TYP003	TYP004	TYP005	TYP006
Oscillation frequency			OSC	003 32.76K Xtal	012 100K Xtal	050 400-500K ceramic /IFT	050 400K ceramic /IFT	400 3 - 4.2K Xtal	400 3-4.2K Xtal
Dividing ratio for internal clock			CP	00	00	04	02	64	32
External timing output			CK	32	64	CP	CP	CP	CP
Count-er	Divider 1 Input	Counter	PD	CP	CP	CP	CP	CP	CP
	Divider 3 Input		PDR	PD2	PD2	PD2	PD2	PD2	PD2
	Reset timing		PDR	N	/12500/	N	N	N	N
Counter Buffer	Buffer 0 Input	Counter Buffer	C0	PI_{60}	PI_{60}	PI_{60}	PI_{60}	PI_{60}	PI_{60}
	Buffer 1 Input		C1	RD9	RDB	RDB	RD7	RDA	RD7
	Buffer 2 Input		C2	RDA	RDC	RDC	RDA	RDB	RDA
	Buffer 3 Input		C3	RDB	RDD	RDD	RDD	RDC	RDD
H flag			HOLD	H	H	H	0	0	0
Restart condition			RSTH	C3	C3	C3	0	0	0
Input/Output port	Port 0		STD	D(PROG)	1(OUT)				
I/O port	Port 2		P2	/F/(OUT)	/F/(OUT)	/0/(IN)	/0/(IN)	/0/(IN)	
	Port 4		P4	/F/(OUT)	/3/(IN/OUT)	/F/(OUT)	/3/(IN/OUT)	/0/(IN)	
Input resistance (Input port 5)			PI5	0(UP)	1(DOWN)				
Decode matrix	Line 0	DECODER	DEC0	/ /					
	Line 1		DEC1	/ /					
	Line 2		DEC2	/ /					
	Line 3		DEC3	/ /					
	Line 4		DEC4	/ /					
Output port 6/7			P06	1(P6/P7)	0(P6)				
P L A	Line 0	PLA	PLA0	/ 00 /	/ /				
	Line 1		PLA1	/ 11 /	/ /				
	Line 2		PLA2	/ 22 /	/ /				
	Line 3		PLA3	/ 33 /	/ /				
	Line 4		PLA4	/ 44 /	/ /				
	Line 5		PLA5	/ 55 /	/ /				
	Line 6		PLA6	/ 66 /	/ /				
	Line 7		PLA7	/ 77 /	/ /				
	Line 8		PLA8	/ 88 /	/ /				
	Line 9		PLA9	/ 99 /	/ /				
	Line A		PLAA	/ AA /	/ /				
	Line B		PLAB	/ BB /	/ /				
	Line C		PLAC	/ CC /	/ /				
	Line D		PLAD	/ DD /	/ /				
	Line E		PLAE	/ EE /	/ /				
	Line F		PLAF	/ FF /	/ /				

Note 1: As to each option item, one of the thick frames horizontally lined shall be selected. (The whole thick frame is enclosed with a circule.)

Note 2: Please enter the data entry mode (decode matrix, PLA) by yourself.

Note 3: The formal name of option is that naming continuously the option name and the data in the designation column of option.

(1) OSC - RSTH Options

OSC-RSTH options can construct six kinds of combination. However, it is impossible to use the options by changing the combinations.

Name of Option		TYP 001	TYP002	TYP 003	TYP 004	TYP 005	TYP 006
	OSC	003	012	050	050	400	400
	CP	00	00	04	02	64	32
	CK	32	64	CP	CP	CP	CP
COUNTER	PD	CP	CP	CP	CP	CP	CP
	PDR	PD2	PD2	PD2	PD2	PD2	PD2
	PDR	N	/12500/	N	N	N	N
COUNTER BUFFER	CO	PI$_{60}$	PI$_{60}$	PI$_{60}$	PI$_{60}$	PI$_{60}$	PI$_{60}$
	C1	RD9	RDB	RDB	RD7	RDA	RD7
	C2	RDA	RDC	RDC	RDA	RDB	RDA
	C3	RDB	RDD	RDD	RDD	RDC	RDD
	HOLD	H	H	H	0	0	0
	RSTH	C3	C3	C3	0	0	0

① OSC: Prescribing the oscillator and oscillation frequency.

o OSC003 - 32.768 KHz oscillation by Xtal oscillator.

o OSC012 - 100 KHz oscillation by Xtal oscillator.

o OSC050 - 400 KHz (400K∿500KHz) oscillation by ceramic oscillator or IFT.

o OSC400 - 4194.304 KHz oscillation by Xtal oscillator.

Note (1): It is possible to make driving from the external oscillator by opening X_{OUT} terminal and by arrangeing X_{IN} terminal for input.

Note (2): The formal name of option is that naming continuously the option name and the data in the designation column of option.

② CP: Prescribing the frequency dividing ratio for obtaining the internal basic clock.

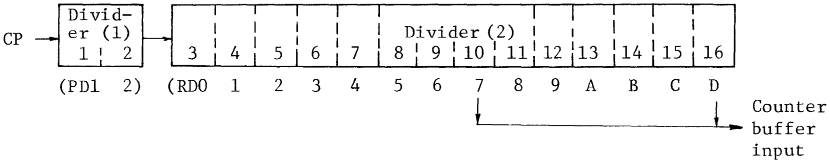
- CP00 - Oscillation signal is made to clock.
- CP02 - Signal of dividing oscillation frequency into 2.
- CP04 - Signal of dividing oscillation frequency into 4.
- CP32 - Signal of dividing oscillation frequency into 32.
- CP64 - Signal of dividing oscillation frequency into 64.

③ CK: Prescribing the output frequency for external timing

- CK32 - Signal of dividing oscillation frequency into 32.
- CK64 - Signal of dividing oscillation frequency into 64.
- CKCP - Signal of the internal basic clock frequency.

④ COUNTER: Dividers

The divider consists of 16-stage divider and con be disassumbled into the basic construction of the front 2 stages [divider (1)] and the rear 14 stages [divider (2)].



- PDCP - Setting the input of divider (1) to the internal basic clock (CP).
- PDRPD2 - Setting the input of divider (2) to the output of divider (1).
- PDRN - Setting the function of divider (1) and divider (2) as the usual binary counters.
- PDR/12500/ - This is the designation only at the time of use of 100K Xtal. When the count value of divider (2) is "12500", the divider (2) is reset to return the count value to "0".

⑤ COUNTER BUFFER Prescribing the input of counter buffer

The counter buffer is composed of 4 bits, which are marked C0, C1, C2 and C3 from LSB side (Bit 0).

- COPI60 - PI60 terminal input to Bit 0.
- C1RD7 - The output of divider 10 to Bit 1.
- C1RD9 - The output of divider 12 to Bit 1.
- C1RDA - The output of divider 13 to Bit 1.
- C1RDB - The output of divider 14 to Bit 1.
- C2RDA - The output of divider 13 to Bit 2.
- C2RDB - The output of divider 14 to Bit 2.
- C2RDC - The output of divider 15 to Bit 2.
- C3RDB - The output of divider 14 to Bit 3.
- C3RDC - The output of divider 15 to Bit 3.
- C3RDD - The output of divider 16 to Bit 3.

⑥ HOLD, RSTH: Prescribing the function of H flag and the signal for restart from the hold state.

- o HOLDH - Hold control is carried out by H flag.
- o HOLDO - No hold is maintained. H flag is capable of being used as the general flag.
- o RSTHC3 - Interlocking with HOLDH. Counter buffer bit 3 is used as the hold restart signal.
- o RSTHO - Interlocking with HOLDO. No hold is maintained.

(2) STD - PI5 Options

STL - PI5 options prescribe the state of input/output of port. It is possible to independently designate STD and P15 options respectively.

Option names	Designation of option				
STD	D(PROG)	1 (OUT)			
P2	/F/(OUT)	/F/(OUT)	/0/(IN)	/0/(IN)	/0/(IN)
P4	/F/(OUT)	/3/(IN/OUT)	/F/(OUT)	/3/(IN/OUT)	/0/(IN)
PI5	0(UP)	1(DOWN)			

① STD: Prescribing the function of port 0.

- o STDD - Switching input/output of port 0 is made according to the contents of D flag.
- o STD1 - Using as the output of port 0. In this case, D flag can be used as the general flag.

② P2, P4: Prescribing the function of port 2 and port 4.

Port 2 and port 4 are the 4-bit ports to which the input or the output can be designated.
The 4-bit ports are expressed by the one-figure numeric characters of hexadecimal digit by defining the input as "0" and the output as "1" for each bit.

(Example)

	(MSB)			(LSB)
Bit	3	2	1	0
	0	0	1	1

(0 : Input
1 : Output) Hexadecimal = 3

- o P2/F/, P4/F/ - Prescribing both port 2 and port 4 as the output.
- o P2/F/, P4/3/ - Prescribing the upper 2 bits of port 4 the input, with the others being the output.
- o P2/0/, P4/F/ - Prescribing port 2 as the input and port 4 as the output.
- o P2/0/, P4/3/ - Prescribing the lower 2 bits of port 4 as as the output, with the others being the input.
- o P2/0/, P4/0/ - Prescribing both port 2 and port 4 as the input.

③ P15: Prescribing the function of the input resistance at port 5.

- o PI50 - Input port 5 comes to the input with pull-up resistance.
- o PI51 - Input port 5 comes to the input with pull-down resistance.

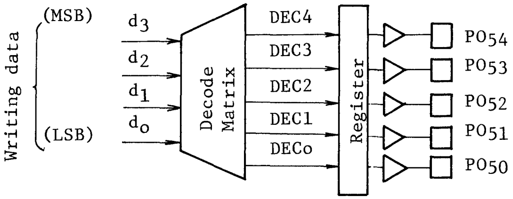
(3) DECODER - PLA Options

The decoder and PLA options are the data writing mode in which user designates the data. P06 option is used according to the function of PLA.

Option Names	Designation of Oprions	
DECODER	Data writing mode	
PO6 PLA	1 (P6/P7) General output port	0 (P6) Data writing mode

1 DECODER: Prescribing the contents of decode matrix.

Operation processing and internal data are processed with 4 bits. However, at the output port 5, 5-bit data are produced as the output after the conversion into 5 lines by the decode matrix.



The following explanation is made for the purpose of explaining the contents of decode matrix and has no direct relations with the design patterning in the actual LSI. Give your kind attention to this point.

The logic of decode matrix can be expressed by the following formula. (See the above diagram.)

DECi = $D_3 \cdot D_2 \cdot D_1 \cdot D_0$ (i = 0 - 4)
Where, the selection is practicable as to
Dj(j = 0 - 3) = dj or $\overline{\text{dj}}$ or 1

As to the designation of decode matrix, it is possible to define the selection as "1", to define non-selection as "0", and to express $\overline{d_3}$, d_3, $\overline{d_2}$, d_2, $\overline{d_1}$, d_1, $\overline{d_0}$, and d_0 as the continuous 8-bit data ($\overline{d_3}$ shall be MSB) with the figures of hexadecimal two digits.

$\overline{d_3}$, d_3		$\overline{d_2}$, d_2		$\overline{d_1}$, d_1		$\overline{d_o}$, d_o	
×	×	×	×	×	×	×	×

x = 1 (Selection) or
0 (Non-selection)

Hexadecimal two digits

Note: $D_j = \overline{d_j} \rightarrow (\overline{d_j},\ d_j)$ = (Selection,Non-selection)=(1, 0)
$D_j = d_j \rightarrow (\overline{d_j},\ d_j)$ = (Non-selection,Selection)=(0, 1)
$D_j = 1 \rightarrow (\overline{d_j},\ d_j)$ = (Non-selection,Non-selection)=(0,0)
It is meaningless to select both $\overline{d_j}$ and d_j

Attention shall be given to the fact that the output comes to "1" except initialization in case where the designation of (Non-selection, Non-selection) is made to all bits (DECi = 1).

o In case where PO_{50} terminal output is "H" when the data "1" are written in the output port 5,and the output is "L" when other data are written:

Logic: $DEC0 = \overline{d3} \cdot \overline{d2} \cdot \overline{d1} \cdot d0$

Designation:

	$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	do	
DECO =	1	0	1	0	1	0	0	1	= X' A9'

o In case where PO_{54} terminal output is "H" when the data "C" - "F" are written in the output port 5,and the output is "L" when other data are written:

Logic: $DEC4 = d3 \cdot d2$

Designation:

	$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	d0	
DEC4 =	0	1	0	1	0	0	0	0	= X' 50'

o In the case of PO_{51} terminal when the output of data written in the output port 5 is desired:

Logic: DEC = d1

		$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	d0	
Designation:	DEC1 =	0	0	0	0	0	1	0	0	= X' 04'

* Refer to the examples of data designation.

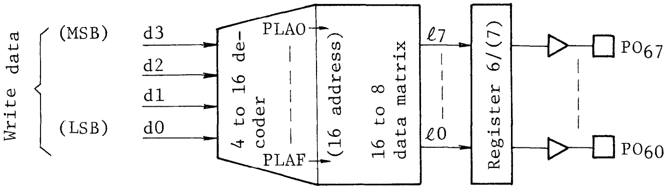
② P06: Prescribing thc function of port 6.

o P060 - Port 6 is used as the 8-bit PLA data output port.

o P061 - Port 6 is used as the two 4-bit general output port. (No designation can be made as to the contents of PLA.)

③ PLA: Prescribing the contents of PLA.

Operation processing and internal data are processed with 4 bits. However, at the output port 6, 8-bit data are produced as the output after the conversion into 8 lines by PLA.



The following explanation is made for the purpose of explaining the contents of PLA and has no direct relations with the design patterning in the actual LSI.
Give your kind attention to this point.

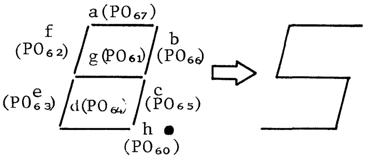
PLA can be expressed as the 16 words x 8 bits memory. (Refer to the above diagram.) Nemaly, the 4-bit write data is the address of PLA, and the 8-bit data read out from PLA comes to the output data on the output port terminals.

Write data	Address	Write data	Address	Write data	Address	Write data	Address
0	PLA0	4	PLA4	8	PLA8	12	PLAC
1	PLA1	5	PLA5	9	PLA9	13	PLAD
2	PLA2	6	PLA6	10	PLAA	14	PLAE
3	PLA3	7	PLA7	11	PLAB	15	PLAF

The designation of PLA data is made to addresses by expressing the 8-bit data, which are MSB in PO67 terminal output and LSB in PO60 terminal output, with 2-figure numeric characters of hexadecimal digit.

o In case each output between PO67 terminal and PO60 terminal comes to H, L, H, H. L, H. H, and L when data "5" are written in the 8-bit output port 6,

```
             PO66    PO64     PO62    PO60
          PO67    PO65    PO63     PO61
PLA 5 =  | 1   0   1   1 | 0   1   1   0 |  = X' B6'
```



o In case data "3" are written in output port 6 or output port 7 to use it as the 4-bit general output port,

```
             PO66    PO64     PO62    PO60
          PO67    PO65     PO63    PO61
PLA 3 =  | 0   0   1   1 | 0   0   1   1 |  = X' 33'
          \___Port 7____/ \___Port 6____/
```

Therefore, if data "3" are written in output port 7, each output between P067 terminal and P064 terminal comes to L, L, H, and H. At this time, each input to register 6 (connected to P063 terminal- P060 terminal) comes likewise to 0, 0, 1, and 1, but each output does not change because no setting is made to the register. This applies to the reverse case comes likewise.

o Refer to the examples of the designation of data.

Examples of the Designation of Data for Decode Matrix and PLA

	Examples of General port	Examples of 5-1 Fig. dynamic indication
Logic	DEC0 = d0	DEC0 = $\overline{d3}$ $\overline{d2}$ $\overline{d1}$ d0
	DEC1 = d1	DEC1 = $\overline{d3}$ $\overline{d2}$ d1 $\overline{d0}$
	DEC2 = d2	DEC2 = $\overline{d3}$ $\overline{d2}$ d1 d0
	DEC3 = d3	DEC3 = $\overline{d3}$ d2 $\overline{d1}$ $\overline{d0}$
	DEC4 = $\overline{d3}$	DEC4 = $\overline{d3}$ d2 $\overline{d1}$ d0
Designated Data	DEC0 = /01/	DEC0 = /A9/
	DEC1 = /04/	DEC1 = /A6/
	DEC2 = /10/	DEC2 = /A5/
	DEC3 = /40/	DEC3 = /9A/
	DEC4 = /80/	DEC4 = /99/

(Evaluator built-in option)

General port	Examples of segment display
PLA 0 = /00/	PLA 0 = /FC/
PLA 1 = /11/	PLA 1 = /60/
PLA 2 = /22/	PLA 2 = /DA/
PLA 3 = /33/	PLA 3 = /F2/
PLA 4 = /44/	PLA 4 = /66/
PLA 5 = /55/	PLA 5 = /B6/
PLA 6 = /66/	PLA 6 = /BE/
PLA 7 = /77/	PLA 7 = /E4/
PLA 8 = /88/	PLA 8 = /FE/
PLA 9 = /99/	PLA 9 = /F6/
PLA A = /AA/	PLA A = /FD/
PLA B = /BB/	PLA B = /00/ (blank)
PLA C = /CC/	PLA C = /02/
PLA D = /DD/	PLA D = /CE/
PLA E = /EE/	PLA E = /9E/
PLA F = /FF/	PLA F = /8E/

(Evaluator built-in option)

MASK ROM DATA TAPE FORMAT

	..... Leader Output of more than 50 characters of "NULL" (No other symbols than "NULL" should be existent.)
'COMMENT' CR LF	..... Comment Output of six characters starting from the beginning of character train defined by TTL statement of source program and two characters of serial number, with apostrophe. (when no TTL statement exist, output of six characters of space code and two chareaters of serial number.)
N8 ; CR LF	..... Output of "N8" indicating that the date pattern is 8 bits long. (The program data follow this code.)
RXXXX ;	..... Output of the program start address following "R" in four frames of decimal ASCII code.
XXX PX ;	..... 1st address data and check sum
XXX PX ;	2nd address data and check sum
XXX PX ; CR LF	8th address data and check sum
RXXXX ;	..... 9th program address
XXX PX ;	9th address data and check sum
	Repeated output through the last data
CR LF $	..... Output of indicating the end of program data
	Trailer: Output of more than 50 characters of "NULL".

XXXPX

Check sum: Number of data bits having "1" is output.

Data: Two-characters of eight bit data are output, in two frames of hexadecimal ASCII code.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	Supply Voltage	-0.3V to + 7.0V
V_{IN}	Input Voltage	-0.3V to V_{DD} + 0.3V
V_{OUT}	Output Voltage	-0.3V to V_{DD} + 0.3V
P_D	Power Dissipation	600 mW
T_{sol}	Soldering Temperature	260°C (10 SEC)
T_{stg}	Storage Temperature	-55°C to + 125°C
T_{opr}	Operating Temperature	-30°C to + 85°C

ALLOWABLE OPERATING CONDITIONS

SYMBOL	ITEM	Condition	
		V_{DD} = 3V to 6V	V_{DD} = 4V to 6V
Ta	Ambient Temepratture	-30°C to + 85°C	-30°C to 85°C
V_{OH}	Output High Voltage	Min. V_{DD} -3.5V(≥1.5V)	Min.V_{DD} -3.5V (≥1.5V)
V_{OL}	Output Low Voltage	Max. 3V	Max. 3V
fx	Xtal Operating Frequency	20KHz to 2MHz	20KHz to 4.2MHz
tcy	Cycle Time	40μs to 100μs	10μs to 100μs

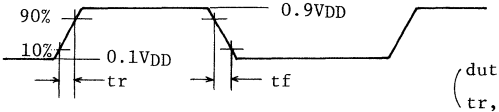
DC CHARACTERISTICS (Ta=-30°C to +85°C, V_{DD}=3V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP. (Note 1)	MAX.	UNIT
V_{IH}	Input High Voltage		V_{DD}x0.75	-	V_{DD}	V
		V_{DD} ≧ 4V	V_{DD}x0.7	V_{DD}x0.55	V_{DD}	
V_{IHS}	Input High Voltage (Schmitt)		V_{DD}x0.9	V_{DD}x0.75	V_{DD}	
		V_{DD} ≧ 4V	V_{DD}x0.85	-	V_{DD}	
V_{IHC}	Input High Voltage (X_{IN} Input)		V_{DD}x0.75	-	V_{DD}	
V_{IL}	Input Low Voltage		0	V_{DD}x0.45	V_{DD}x0.3	
V_{ILS}	Input Low Voltage (Schmitt)		0	V_{DD}x0.35	V_{DD}x0.1	
		V_{DD} ≧ 4V	0	-	V_{DD}x0.15	
V_{ILC}	Input Low Voltage (X_{IN} Input)		0	-	V_{DD}x0.25	
I_{IH}	Input High Current	V_{DD}=6V, V_{IN}=6V	-	-	20	μA
I_{IL}	Input Low Current	V_{DD}=6V, V_{IN}=0V	-	-	-20	
R_{IN}	Input Resistance (PI5)	V_{DD}=5V	75	150	350	kΩ
V_{OH}	Output High Voltage	V_{DD}=5V, Output Open	4.7	4.9	-	V
V_{OL}	Output Low Voltage		-	0.1	0.3	
I_{OH}	Output High Current	V_{DD}=4.5V, V_{OH}=2.4V	-0.7	-2	-	mA
I_{OH1}	Output High Current (P05, P06)		-2.5	-6	-	
		V_{DD}=5V, V_{OH}=4.2V	-1.1	-2.5	-	
I_{OL}	Output Low Current	V_{DD}=4.5V, V_{OL}=0.45V	1.6	4	-	
I_{OL1}	Output Low Current (P05,P06)		3.5	8	-	
I_{DDO}	V_{DD} Supply Current in Normal Operation (fx=32.8 KHz)	V_{DD}=6V V_{IN}=5.9V/0.1V (All valid) PI5 Open C_L = 50pF (Note 3)	-	50	300	μA
	(fx=100 KHz)		-	150	450	
	(fx=400 KHz)		-	400	1200	
	(fx=4.19 MHz)		-	1000	3000	
I_{DIH}	V_{DD} Supply Current in Hold Operation (fx=32.8 KHz)		-	15	80	
	(fx=100 KHz)		-	40	120	
	(fx=400 KHz)		-	150	450	

Note 1: Typical values are at Ta=25°C and V_{DD}=5V.

Note 2: Output characteristic excludes X_{OUT} terminal.

Note 3: X_{IN} input waveform at the time of measuring V_{DD} supply current.



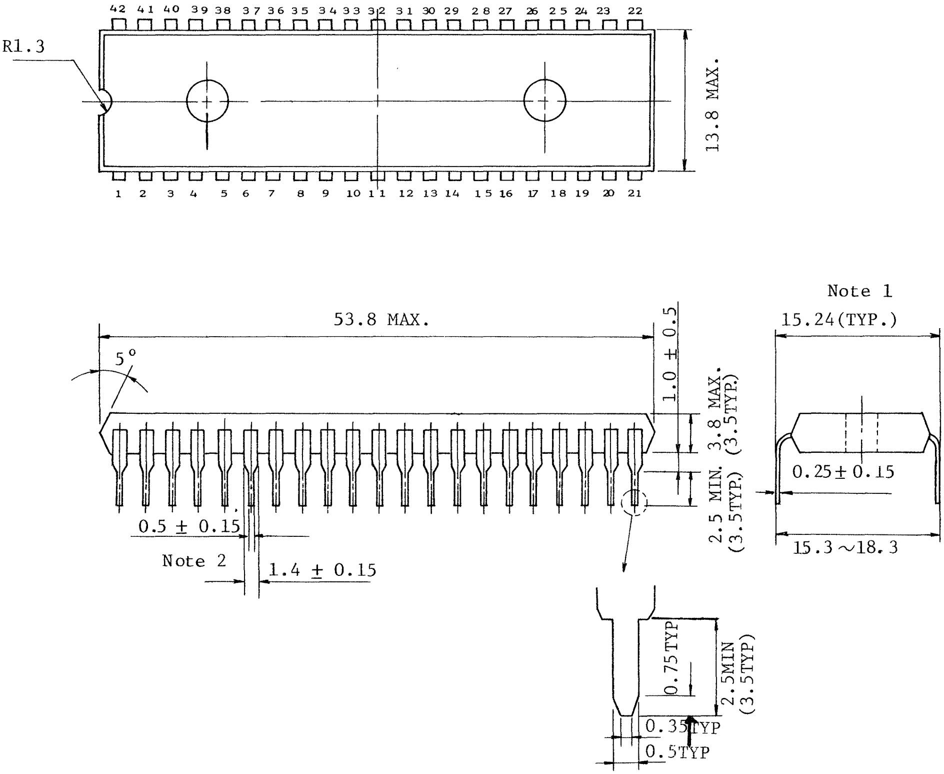
(duty = 50%
tr,tf < 50ns)

AC CHARACTERISTICS (Ta=-30°C to +85°C, V_{DD}=3V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{WXIN}	XIN Pluse Width	External Input $V_{IN}=V_{IHC}/V_{ILC}$	0.4/fx	-	$0.6/f_x$	SEC
t_{WRESET}	$\overline{\text{RESET}}$ Pulse Width	$V_{IN}=V_{IHS}/V_{ILS}$	2 tcy	-	-	μs
t_{WINT}	$\overline{\text{INT}}$ Pulse Width		2 tcy	-	-	
t_{WPI60}	PI60 Pulse Width		2 tcy	-	-	

OUTLINE DRAWINGS

Unit in mm



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.

"C^2MOS" DIGITAL INTEGRATED CIRCUIT

TCP4620AF
TCP4630AF

SILICON MONOLITHIC

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

This is the specification for TCP4620AF/TCP4630AF in the TLCS-46A family.

TCP4620AF/TCP4630AF is a flat package version of TCP4620AP/TCP4630AP. There are some differences in electrical characteristics between TCP4620AF/TCP4630AF and TCP4620AP/TCP4630AP; however, their function, instruction, and pin description are compatible.
When using and examining TCP4620AF/TCP4630AF, therefore, it is recommended that this specification be used together with the technical data on TCP4620AP/TCP4630AP.

The differences in electrical characteristics between the two are as follows:

1. Power Dissipation

   P_D = 400 mW MAX

2. Operating Temperature and Ambient Temperature

   T_{opr} = -20 to 70°C

   T_a = -20 to 70°C

INTEGRATEDCIRCUIT

TECHNICAL DATA

TCP4620AF
TCP4630AF

TCP4620AF /TCP4630AF ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	Rating
V_{DD}	Supply Voltage	-0.3V to +7.0V
V_{IN}	Input Voltage	-0.3V to V_{DD}+0.3V
V_{OUT}	Output Voltage	-0.3V to V_{DD}+0.3V
P_D	Power Dissipation	400mw
T_{sol}	Soldering Temperature	260°C(10 SEC)
T_{stg}	Storage Temperature	-55°C to +125°C
T_{opr}	Operating Temperature	-20°C to +70°C

ALLOWABLE OPARATING CONDITION

SYMBOL	ITEM	Condition	
		V_{DD}=3V to 6V	V_{DD}=4V to 6V
Ta	Ambient Temperature	-20°C to +70°C	-20°C to +70°C
V_{OH}	Output High Voltage	Min. V_{DD} -3.5V(>1.5V)	Min. V_{DD} -3.5V(>1.5V)
V_{OL}	Output Low Voltage	Max. 3V	Max. 3V
fx	Xtal Operating Frequency	20KHz to 2MHz	20KHz to 4.2MHz
tcy	Cycle Time	40μs to 100μs	10μs to 100μs

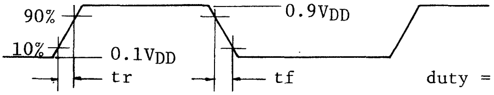
DC CHARACTERISTICS (Ta=-20°C to 70°C, V_{DD}=3V to 6V)

SYMBOL	PARAMETER		TEST CONDITION	MIN.	TYP. (Note 1)	MAX.	UNIT
V_{IH}	Input High Voltage			V_{DD}x0.75	-	V_{DD}	V
			V_{DD} ≧ 4V	V_{DD}x0.7	V_{DD}x0.55	V_{DD}	
V_{IHS}	Input High Voltage (Schmitt)			V_{DD}x0.9	V_{DD}x0.75	V_{DD}	
			V_{DD} ≧ 4V	V_{DD}x0.85	-	V_{DD}	
V_{IHC}	Input High Voltage (X_{IN} Input)			V_{DD}x0.75	-	V_{DD}	
V_{IL}	Input Low Voltage			0	V_{DD}x0.45	V_{DD}x0.3	
V_{ILS}	Input Low Voltage (Schmitt)			0	V_{DD}x0.35	V_{DD}x0.1	
			V_{DD} ≧ 4V	0	-	V_{DD}x0.15	
V_{ILC}	Input Low Voltage (X_{IN} Input)			0	-	V_{DD}x0.25	
I_{IH}	Input High Current		V_{DD}=6V, V_{IN}=6V	-	-	20	μA
I_{IL}	Input Low Current		V_{DD}=6V, V_{IN}=0V	-	-	-20	
R_{IN}	Input Resistance (PI5)		V_{DD}=5V	75	150	350	KΩ
V_{OH}	Output High Voltage		V_{DD}=5V, Output Open	4.7	4.9	-	V
V_{OL}	Output Low Voltage			-	0.1	0.3	
I_{OH}	Output High Current		V_{DD}=4.5V, V_{OH}=2.4V	-0.7	-2	-	mA
I_{OH1}	Output High Current (P05, P06)			-2.5	-6	-	
			V_{DD}=5V, V_{OH}=4.2V	-1.1	-2.5	-	
I_{OL}	Output Low Current		V_{DD}=4.5V, V_{OL}=0.45V	1.6	4	-	
I_{OL1}	Output Low Current (P05, P06)			3.5	8	-	
I_{DDO}	V_{DD} Supply Current in Normal Operation	(fx=32.8 KHz)	V_{DD}=6V V_{IN}=5.9V/0.1V (all valid) PI5 Open C_L = 50pF (Note 3)	-	50	300	μA
		(fx=100 KHz)		-	150	450	
		(fx=400 KHz)		-	400	1200	
		(fx=4.19 MHz)		-	1000	3000	
I_{DDH}	V_{DD} Supply Current in Hold Operation	(fx=32.8 KHz)		-	15	80	
		(fx=100 KHz)		-	40	120	
		(fx=400 KHz)		-	150	450	

Note 1: Typical values are at Ta=25°C and V_{DD}=5V.

Note 2: Output characteristic excludes X_{OUT} terminal.

Note 3: X_{IN} input waveform at the time of measuring V_{DD} Supply Current.



duty = 50%

tr, tf < 50ns

AC CHARACTERISTICS (Ta=-20°C to +70°C, V_{DD}=3V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{WXIN}	XIN Pluse Width	External Input $V_{IN}=V_{IHC}/V_{ILC}$	0.4/fx	-	$0.6/f_x$	SEC
t_{WRESET}	$\overline{\text{RESET}}$ Pulse Width	$V_{IN}=V_{IHS}/V_{ILS}$	2 tcy	-	-	μs
t_{WINT}	$\overline{\text{INT}}$ Pulse Width		2 tcy	-	-	
t_{WPI60}	P160 Pulse Width		2 tcy	-	-	

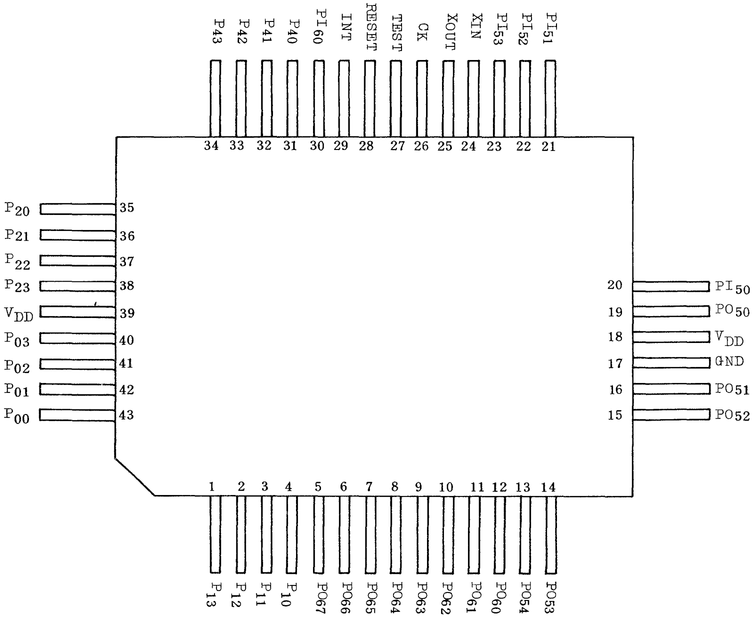
Note: Flat packages have a merit in assembly space, but they should be installed in better humidity and temperature environment than DIP's.

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TCP4620AF
TCP4630AF

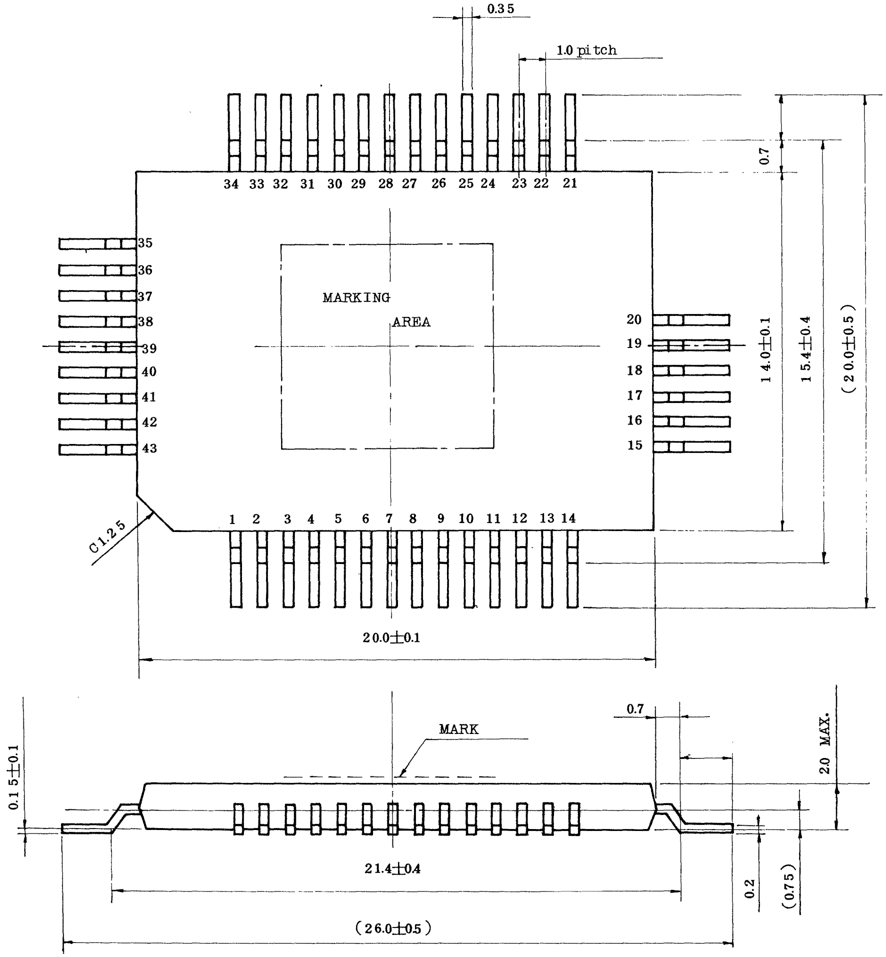
PIN CONNECTIONS (TOP VIEW)



Note) Pins 18 and 39, power supply terminals, are connected in the package.

OUTLINE DRAWINGS

Unit in mm



Toshiba 東芝 **INTEGRATED CIRCUIT**

TECHNICAL DATA

"C^2MOS" DIGITAL INTEGRATED CIRCUIT

TCP4620BP

SILICON MONOLITHIC

TENTATIVE

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

GENERAL DESCRIPTION

This is the specification for TCP4620BP in TLCS-46A family.

TCP4620BP is one version of TLCS-46A developed for the purpose of reducing the power consumption at hold operation so as to correspond to the applications having utilized the features of a CMOS microcomputer. TCP4620BP has realized the following charactristics.

Supply Current in hold operation	I_{DDHS} TYP.=0.1μA (V_{DDHS}=2V)
Hold voltage in hold operation	V_{DDHS} = 2V to 6V

TCP4620BP is a version of TCP4620AP to which the oscillation control (stop/start) function of a clock generator in hold operation has been added. There are some differences in function, pin description, electrical characteristics, and mask option concering hold operation and oscillation contril between TCP4620BP and TCP4620AP; however, other functions, electrical characteristics, instructions, and pin connections are compatible.

When using and examining TCP4620BP, therefore, it is recommended that this specification be used together with the technical data on TCP4620AP.

Main differences with TCP4620AP are as follows:

1. Function

o In hold operation, the input level of PI_{60} terminal makes possible the oscillation control (stop/start) of the oscillator, and the oscillation stop enables the power consumption to be reduced.

Supply Current in hold operation (Oscillation stop)

I_{DDHS} TYP.=0.1μA (Ta=25°C, V_{DDHS}=2V)

Hold Voltage in hold operation (Oscillation stop)

V_{DDHS} =2 V to 6 V

2. Pin Description

o In hold opeartion, PI$_{60}$ terminal will function as an oscillation control terminal.

PI$_{60}$ = "H" .... Oscillation stop

PI$_{60}$ = "L" .... Oscillation start (Continued.)

o In either case of normal operation/hold operation, PI$_{60}$ terminal is the reset input terminal of the divider (2).

PI$_{60}$ = "H" .... The divider (2) is reset.

PI$_{60}$ = "L" .... The divider (2) is not reset. (Reset is released.)

o No Schmitt circuits are built in the following two terminals:

PI60, $\overline{\text{INT}}$

3. Mask Option

The contents only indicated on the mask option sheet of TCP4620BP can be designated., 400 KHz ceramic oscillator/IFT oscillator can be used.

PIN NAMES AND PIN DESCRIPTION

Pin Name	Input/Output	Function
P_{03} P_{00}	Input/Output	4-bit general purpose I/O port (I/O is designated by a programe). Can be used as a dedicated output port. (Designated by mask options.)
P_{13} P_{10}	Output	4-bit general purpose output port.
P_{23} P_{20}	Input or Output	4-bit general purpose Input/Output port. Input/Output is design ted by mask options.
P_{43} P_{40}	Input or Output	4-bit general purpose Input/Output port.
P_{54} P_{50}	Output	5-digit output port for display. (Can be used as the general purpose output port.)
P67 P60	Output	8-segment output port for display. (Can be used as the general purpose 8-bit output port (4-bit x 2))
PI_{53} PI_{50}	Input	4-bit general purpose input port.
PI_{60}*	Input	1-bit general purpose input port/reset input of divider(2) [Note 1] (without Schmitt circuit). During hold operation this input functions as an input for stopping the oscillation and for controlling restart of the oscillator. *
$\overline{RESET}$	Input	Reset signal input
$\overline{INT}$	Input	Interrupt request signal input (without Schmitt circuit).
X_{IN}	Input	Oscillator connecting terminal.
X_{OUT}	Output	Oscillator connecting terminal.
CK	Output	External timing output.
TEST	Input	LSI test signal input, used by connection to GND.
V_{DD}		Power Supply
GND		GND

Note 1) In either case of normal operation or hold operation, the divider (2) is reset at "H" level input.

Note 2) * denotes the differences between TCP4620BP and TCP4620AP/TCP4630AP.

HOLD OPERATION

The hold operation of TCP4620BP is performed by setting "1" to H flag. In other words, immediately when "1" is set to H flag by execution of SBS, IBS, and LSM instructions, the internal timing signals ϕs and ϕ are stopped and TCP4620BP begins hold operation. (Refer to Figs. 1 and 3.)

During hold operation the program counter, instruction register, data memory, and other internal registers hold the contents stored before hold operation. However, the interrupt latch is cleared during hold operation, the waiting interrupt request and a new interrupt request are ignored.

The oscillation of oscillator is suspended by holding PI$_{60}$ terminal to "H" level during hold operation, so that the power consumption may be saved greatly.That is, while H flag $\wedge$ PI$_{60}$ = 1, the oscillation is suspended. (Refer to Fig. 3.)

TCP4620BP is restarted from hold operation due to reset of H flag to "0" by the rising signal of the counter buffer C2 bit.

Since the counter buffer is the output of the divider (2) which is counting the internal basic clock (cp), TCP4620BP is restarted at regular intervals if oscillation is not suspended during hold operation. (Refer to Fig. 2.) This is the same cperation as the hold operation of TCP4620AP/TCP4630AP. (However, there is a difference in nothing but restart condition by mask option between them.)

When the oscillation is suspended during hold operations, restart operation begins in the lapse of a certain period of time after a start of oscillation by holding PI$_{60}$ terminal to "L" level. (Refer to Fig. 4.) Special attention should be paid to the fact that in TCP4620BP the divider (2) is reset to "0" by holding PI$_{60}$ terminal to "H" level.

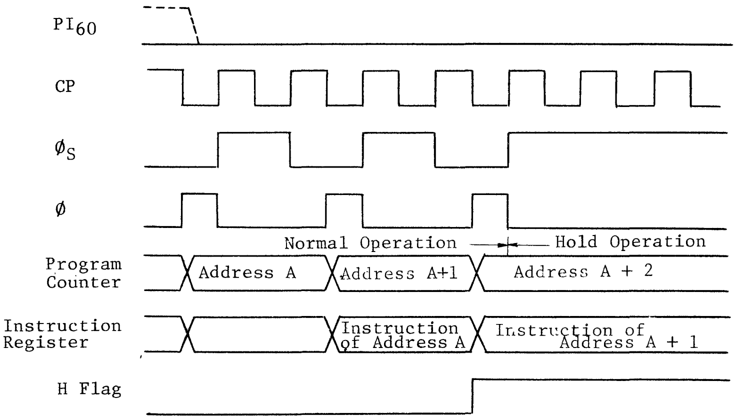


Fig. 1 Timing to Hold Operation (Oscillation Continuity)

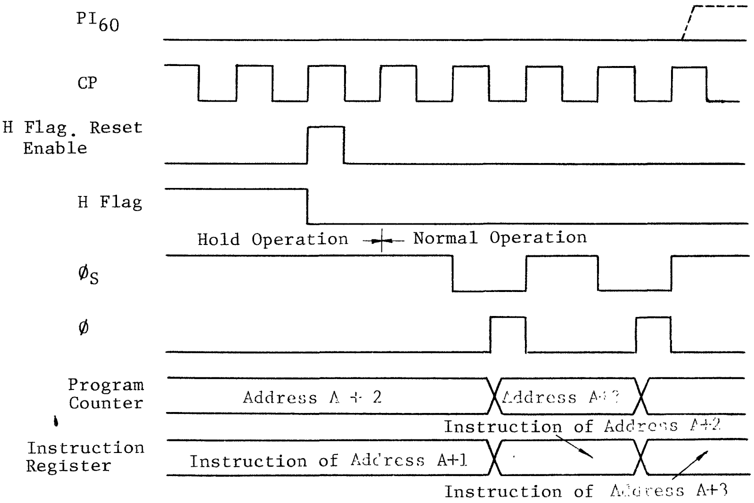


Fig. 2 Restart Timing from Hold Operation (Oscillation Continuity)

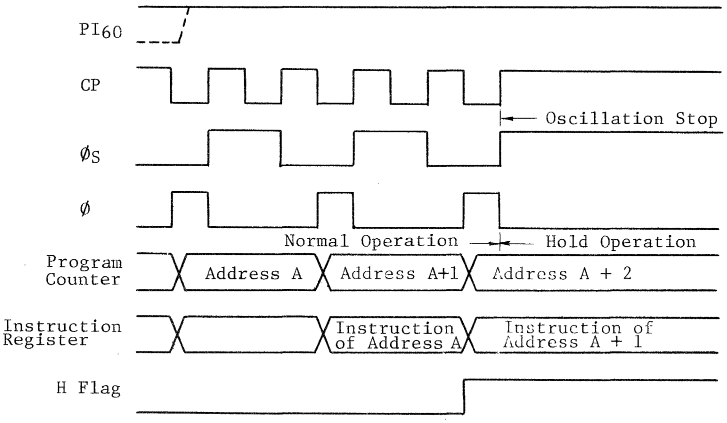


Fig. 3 Timing to Hold Operation (Oscillation Stop)

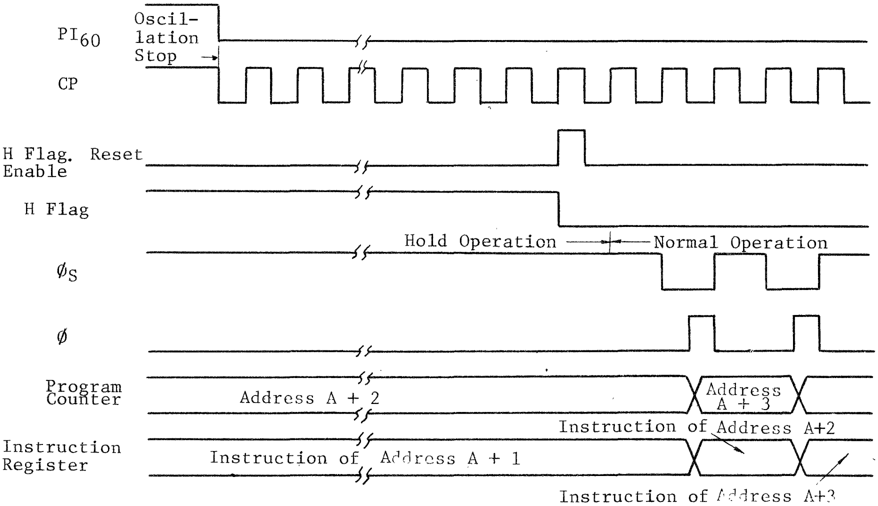


Fig. 4 Restart Timing from Hold Operation (Oscillation Stop)

TCP4620BP MASK OPTION

				TYP011				
Oscillating Frequency			OSC	0 5 0 400K Ceramic IFT				
Frequency division for internal clock			CP	02				
Ext. timing output			CK	CP				
COUNTER	Divider 1 input	COUNTER	PD	CP				
	Divider 3 input		PDR	PD2				
	Reset Timing		PDR	N				
COUNTER BUFFER	Buffer 0 input	COUNTER BUFFER	C0	PI60				
	Buffer 1 input		C1	RD7				
	Buffer 2 input		C2	RDA				
	Buffer 3 input		C3	RDD				
H Flag			HOLD	H				
Restart Signal			RSTH	C2 Rise				
I/O Port	Port 0		STD	D(PROG)	1 (OUT)			
Input/Output port	Port 2		P2	/F/(OUT)	/F/(OUT)	/0/(IN)	/0/(IN)	/0/(IN)
	Port 4		P4	/F/(OUT)	/3/(IN/OUT)	/F/(OUT)	/3/(IN/OUT)	/0/(IN)
Input resistance (Input port 5)			PI5	0 (UP)	1 (DOWN)			
DECODE MATRIX	Line 0	DECODER	DEC0	/ /				
	Line 1		DEC1	/ /				
	Line 2		DEC2	/ /				
	Line 3		DEC3	/ /				
	Line 4		DEC4	/ /				
Output Port	6/7		P06	1(P6/P7)	0 (P6)			
P L A	Line 0	P L A	PLA0	/ 0 0 /	/ /			
	Line 1		PLA1	/ 1 1 /	/ /			
	Line 2		PLA2	/ 2 2 /	/ /			
	Line 3		PLA3	/ 3 3 /	/ /			
	Line 4		PLA4	/ 4 4 /	/ /			
	Line 5		PLA5	/ 5 5 /	/ /			
	Line 6		PLA6	/ 6 6 /	/ /			
	Line 7		PLA7	/ 7 7 /	/ /			
	Line 8		PLA8	/ 8 8 /	/ /			
	Line 9		PLA9	/ 9 9 /	/ /			
	Line A		PLAA	/ A A /	/ /			
	Line B		PLAB	/ B B /	/ /			
	Line C		PLAC	/ C C /	/ /			
	Line D		PLAD	/ D D /	/ /			
	Line E		PLAE	/ E E /	/ /			
	Line F		PLAF	/ F F /	/ /			

- o $\overline{\mathrm{INT}}$ and PI60 are normal input terminal containing no Schmitt circuit.
- o In Hold operation, oscillation is stopped at PI60="H" and started at PI60="L".
- o Divider (2) is reset at PI60="H".

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	Supply Voltage	-0.3V to +7.0V
V_{IN}	Input Voltage	-0.3V to V_{DD}+0.3V
V_{OUT}	Output Volatge	-0.3V to V_{DD}+0.3V
P_D	Power Dissipation	600 mW
T_{sol}	Soldering Temperature	260°C(10 SEC)
T_{stg}	Storage Temperature	-55°C to +125°C
T_{opr}	Operating Temperature	-30°C to + 85°C

ALLOWABLE OPERATING CONDITIONS

SYMBOL	ITEM	CONDITION
V_{DD}	Supply Voltage	4V to 6V
Ta	Ambient Temperature	-30°C to +85°C
V_{OH}	Output High Voltage	Min. V_{DD} -3.5V(≥1.5V)
V_{OL}	Output Low Voltage	Max. 3V
f_X	Xtal Operating Frequency	40KHz to 400KHz
tcy	Cycle Time	10μs to 100μs
V_{DDHS}	Hold Voltage at Hold Operation (Oscillation Stop)	2V to 6V

Normal Operation

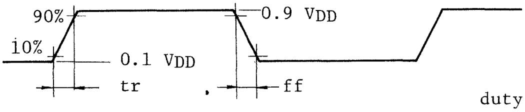
DC CHARACTERISTICS (Ta=-30°C to +85°C, V_{DD}=4V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP. (Notel)	MAX.	UNIT
V_{IH}	Input High Voltage		V_{DD}x0.7	-	V_{DD}	V
V_{IHS}	Input High Voltage (Schmitt)		V_{DD}x0.85	-	V_{DD}	
V_{IHC}	Input High Voltage (X_{IN})		V_{DD}x0.75	-	V_{DD}	
V_{IL}	Input Low Voltage		0	-	V_{DD}x0.3	
V_{ILS}	Input Low Voltage (Schmitt)		0	-	V_{DD}x0.15	
V_{ILC}	Input Low Voltage (X_{IN})		0	-	V_{DD}x0.25	
I_{IH}	Input High Current	V_{DD}=6V,V_{IN}=6V	-	-	20	μA
I_{IL}	Input Low Current	V_{DD}=6V,V_{IN}=0V	-	-	-20	
R_{IN}	Input Resistance (PI5)	V_{DD}=5V	75	150	350	kΩ
V_{OH}	Output High Voltage	V_{DD}=5V,Output Open	4.7	4.9	-	V
V_{OL}	Output Low Voltage		-	0.1	0.3	
I_{OH}	Output High Current	V_{DD}=4.5V,V_{OH}=2.4V	-0.7	-2	-	mA
I_{OH1}	Output High Current (P05, P06)		-2.5	-6	-	
		V_{DD}=5V, V_{OH}=4.2V	-1.1	-2.5	-	
I_{OL}	Output Low Current	V_{DD}=4.5V, V_{OL}=0.45V	1.6	4	-	
I_{OL1}	Output Low Current (P05,P06)		3.5	8	-	
I_{DDO}	V_{DD} Supply Current in Normal Operation	V_{DD}=6V, f_X=400kHz V_{IN}=5.9V/0.1V PI5 Open, CL=50pF (Note 3)	-	400	1200	μA
I_{DDH}	V_{DD} Supply Current in Hold operation (Oscillation Cont.)		-	150	450	

Note : Typical values are at Ta=25°C

Note : Output characteristic excludes X_{OUT} terminal.

Note : X_{IN} input waveform at the time of measuring V_{DD} Supply Curren



duty = 50%

tr,tf < 50ns

Normal Operation

AC CHARACTERISTICS (Ta=-30°C to +85°C, V_{DD}=4V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{WXIN}	XIN Pluse Width	External Input $V_{IN}=V_{IHC}/V_{ILC}$	0.4/fx	-	0.6/fx	SEC
t_{WRESET}	$\overline{\text{RESET}}$ Pulse Width	$V_{IN}=V_{IHS}/V_{ILS}$	2 tcy	-	-	μs
t_{WINT}	$\overline{\text{INT}}$ Pulse Width	$V_{IN}=V_{IH}/V_{IL}$	2 tcy	-	-	
t_{WPI60}	P160 Pulse Width (Note 1)		2tcy	-	-	

Note 1 : PI60 terminal used for general purpose input port.

Hold Operation (Oscillation stop)

DC CHARACTERISTICS (Ta=-30°C to 85°C, V_{DD}=2V to 6V)

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP. (Note1)	MAX.	UNIT
V_{DD}	Hold Voltage in Hold operation		2.0	-	6.0	V
I_{DDHS}	V_{DD} Supply Current in Hold Mode	V_{DDHS}=2V, V_{IN}=1.9V/ 0.1V PI5 Open $V_{\overline{RESET}}$=V_{DDHS}	-	0.1	T.B.D.	μA
		V_{DDHS}=5V, V_{IN}=4.9V/ 0.1V PI5 Open $V_{\overline{RESET}}$=V_{DDHS}	-	0.2	T.B.D.	
V_{IHHS}	Input High Voltage (P160)	V_{DDHS}=2V	1.9	-	-	V
R_{IN}	Input Registance (PI5)		75	150	350	KΩ
I_{OHHS}	Output High Current	V_{DDHS}=2V, V_{OH}=1V	-	-0.4	-	mA
I_{OH1HS}	Output High Current (PI5,PI6)	V_{DDHS}=2V, V_{OH}=1V	-	-0.9	-	mA
I_{OLHS}	Output High Current	V_{DDHS}=2V, V_{OL}=0.45V	-	2.0	-	mA
I_{OL1HS}	Output High Current (PO5,PO6)	V_{DDHS}=2V, V_{OL}=0.45V	-	3.5	-	mA

Note 1 : Typical values are at Ta=25°C.

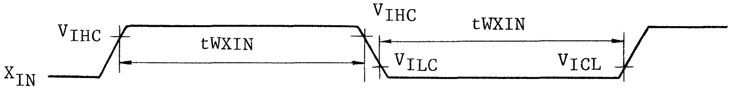
Note 2 : The limits of V_{IHHS}, I_{OHHS}, I_{OLHS} and I_{OL1HS} are equal to the limits of VIH, IOH, IOH1, IOC and I_{OC1} in Normal Operation respectively.
(V_{DDHS} 4V)

AC CHARACTERISTICS (Ta=-30°C to 85°C)

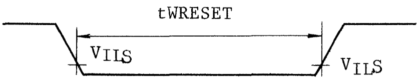
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
t_{WRST}	Restort Internal from Hold operation	V_{IN}=V_{IH}/V_{IL} V_{DD}=4V to 6V	-	2050tcy	-	μS

TIMING WAVEFORMS

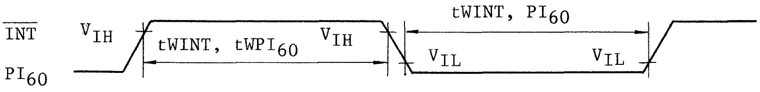
(1) X_{IN} Input Waveform



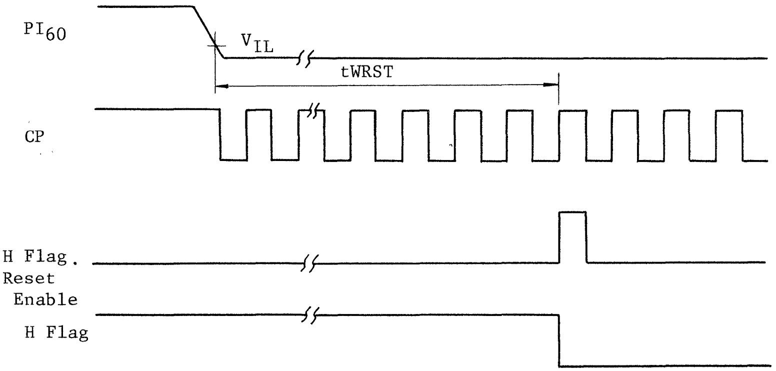
(2) $\overline{\text{RESET}}$ Input Waveform



(3) $\overline{\text{INT}}$, PI_{60} Input Waveform



(4) Restart timing from Hold Operation after Oscillation Stop.



APPLICATION EXAMPLES

[1] Battery Backup

It is thinkable that there is a backup method which utilizes a battery and a capacitor, as application of the system required to uninterruptedly resume the process before cutting-off of the power supply when the power supply has been restored after the power supply was cut off during the operation of the system.

Fig. 1 shows the example of utilizing a battery and Fig. 2 shows the example of utilizing a capacitor. In both the cases, the operations are almoѕt identical. Let us explain the circuit operation of the example (Fig. 1) of utilizing a battery.

Under the condition of "Main Power Voltage > Backup Voltage," the power (V_{DD}) of TCP420BP is supplied from the main power supply. When Q_1 is ON, "L" level is applied to PI_{60} terminal.

In other words, the oscillator is in oscillation state, and the operation of TCP4620BP depends on a program.
Under the condition of "Main Power Voltage < Backup Voltage," the power (V_{DD}) of TCP4620BP is supplied from the battery, and when Q_1 comes into OFF state, "H" level is applied to PI_{60} terminal.

In other words, the oscillator depends on the operating state of TCP4620BP, and if it is in hold state,.oscillation is stopped.

When TCP4620BP is put to holl state immediately after the power supply has been cut off, the oscillation is stopped, permitting a backup for a long period time.

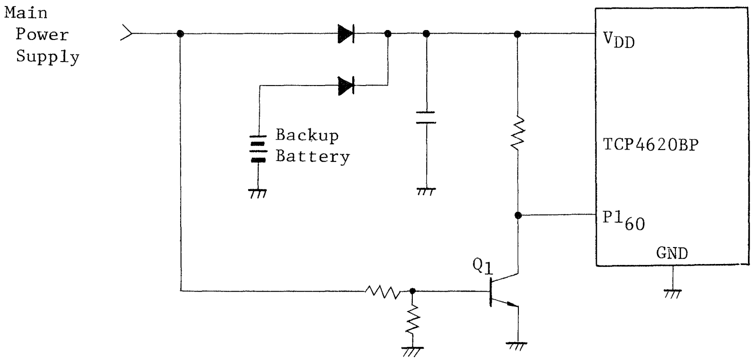


Fig. 1 Example of Backup Circuit by Use of Battery

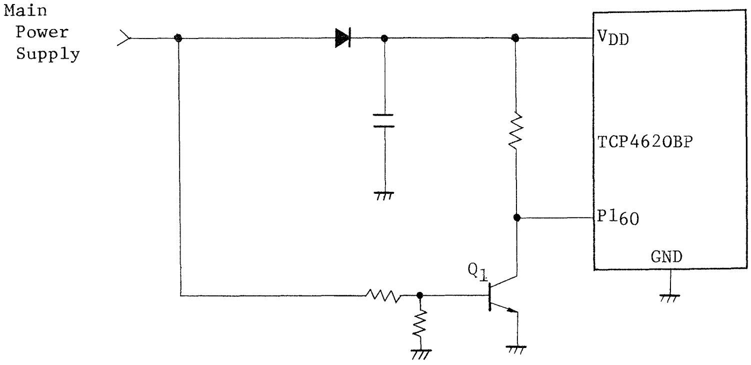


Fig. 2 Example of Backup Circuit by Use of Capacitor

[2] Restart by Key Input

The possibility of a remote control system utilizing TCP4620BP can be considered as application of a system which operates only when the key is fed.

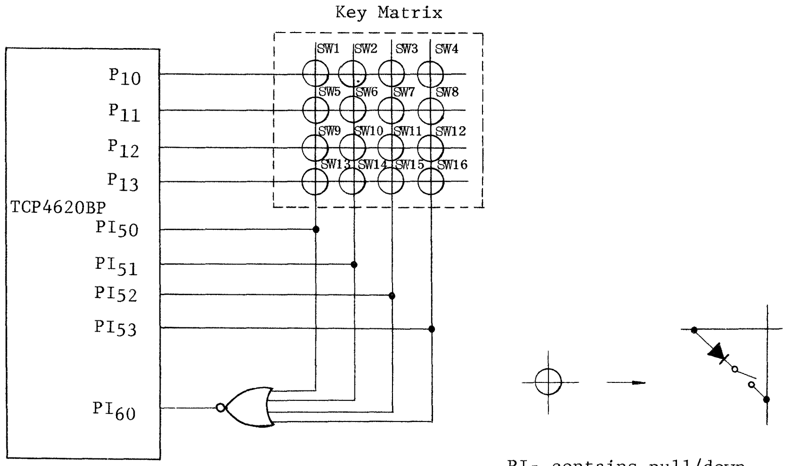
Let us explain the example of application in Fig. 3. Usually TCP4620BP is in hold state, and only when the key is fed, the hold state is released to perform the process according to the input data.

In other words, TCP4620BP is put to hold state after data X'F' has been output (putting SW1 to SW16 to selection state) to its output port 1 (P1).

When the keys (SW1 to SW16) are depressed in this state, "L" level is applied to PI_{60} terminal and the hold state is released after certain period of time.

The keys are scanned after the release of hold state, and the process is performed according to the key input data. After processing, the data output (X'F') to the output port 1 (P1) and a hold state are established.

In such an application, TCP4620BP operates only when a key is depressed, permitting a sharp reduction in power consumption and a better display of the functions of TCP4620BP.



PI_5 contains pull/down resistance at mask option.

Fig. 3 Example of Restart Circuit by Key Input

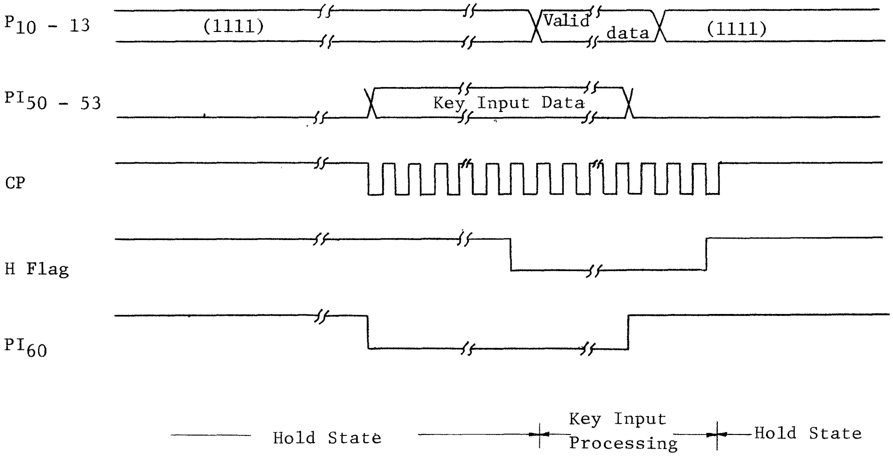


Fig. 4 Example of Circuit (Fig. 3) Timing Chart

TCP4621AP

C^2-MOS DIGITAL INTEGRATED CIRCUIT

SILICON MONOLITHIC

GENERAL DESCRIPTION

The TCP4621AP is a CMOS 4-bit single chip microcomputer most suitable for applications of driving fluorescent display tubess and of interfacing with high voltage circuits.

Output high breakdown voltage:
V_{DD} - 32V (Absolute max. rating V_{DD} - 35V)

'Output high breakdown voltage port:
14 ports ($P0_{60}$ to $P0_{67}$, $P0_{50}$ to $P0_{54}$, P_{10})

The TCP4621AP has adopted the P-channel Open Drain structure of some output ports of the TCP4620AP, resulting in high breakdown voltage. There are some difference between the TCP4620AP and the TCP4621AP in electrical characterisitcs, pin functions, and mask options, but the other functions, electrical characteristics, instructions, and pin connections of the TCP4621AP are compatible with those of the TCP4620AP. For the use and study of this device, use jointly with TCP4620AP Technical Data.

The following items are the points of difference in electrical characteristics.

1. Operating temperature and ambient temperature
   Topr = -20 to +70°C
   Ta = -20 to +70°C

2. Output high breakdown voltage port and output high level current
   IOH1 ($P0_{60}$ to $P0_{67}$) = -1mA MIN. (V_{DD}=4.5V, V_{OH}=2.5V)
   IOH1 ($P0_{50}$ to $P0_{54}$, P_{10}) = -7mA MIN. (V_{DD}=4.5V, V_{OH}=2.5V)

3. Mask option
   The contents only which are shown on the TCP4621AP mask option sheet can be designated. 400 KHz ceramic/IFT can be used as an oscillator.

PIN NAMES & PIN DESCRIPTION

Pin Name	Pin No.	Input/Output	Function
P_{03} to P_{00}	4	Input/Output	4-bit general purpose I/O port (I/O is changed over by a program.) This port can be used as a dedicated output. (I/O is designated by mask options.)
P_{13} to P_{10}	4	Output	4-bit general purpose output port * P_{10} only is a output high breakdown voltage port
P_{23} to P_{20}	4	Input or Output	4-bit general purpos Input/Output port
P_{43} to P_{40}	4	Input or Output	Input/Output is designated by mask options.
PO_{54} to PO_{50}	5	Output	* 5-digit Output high breakdown voltage port for display. (Can be used as the general purpose output port.)
PO_{67} to PO_{60}	8	Output	* 8-segment output high breakdown port for display [Can be used as the general purpose 8-bit output port (4 × 2 bits).]
PI_{53} to PI_{50}	4	Input	4-bit general purpose input port
PI60	1	Input	1-bit general purpose input port (with an internal Schmitt circuit)
$\overline{RESET}$	1	Input	Reset input terminal (with an internal Schmitt circuit)
$\overline{INT}$	1	Input	Interrupt request input terminal (with an internal Schmitt circuit)
X_{IN}	1	Input	Oscillator connecting terminal
X_{OUT}	1	Output	Oscillator connecting terminal
C K	1	Output	External timing output
TEST	1	Input	Used by connecting to GND at all times
V_{DD}	1		Power supply
GND	1		GND

Note) The asterisk(*) indicates the points of difference between the TCP4620AP and this device.

Output high breakdown voltage ports are of P-channel open drain structure.

TCP4621AP MASK OPTIONS

				TYP021	TYP022			
Oscillation frequency			OSC	050 400K ceramic IFT	050 400K ceramic IFT			
Dividing ratio for internal clock			CP	02	02			
External timing output			CK	CP	PD2			
Count-er	Divider 1 Input	COUNTER	PD	CP	CP			
	Divider 3 Input		PDR	PD2	PD2			
	Reset timing		PDR	N	/12500/			
Counter Buffer	Buffer 0 Input	COUNTER BUFFER	CO	PI60	PI60			
	Buffer 1 Input		C1	RD7	RDB			
	Buffer 2 Input		C2	RDA	RDC			
	Buffer 3 Input		C3	RDD	RDD			
H flag			HOLD	0	H			
Restart condition			RSTH	0	C3			
Input/Out-put port	Port 0		STD	D(PROG)	1 (OUT)			
I/O Port	Port 2		P2	/F/(OUT)	/F/(OUT)	/0/(IN)	/0/(IN)	/0/(IN)
	Port 4		P4	/F/(OUT)	/3/(IN/OUT)	/F/(OUT)	/3/(IN/OUT)	/0/(IN)
Input resistance (Input port 5)			P15	0 (UP)	1 (DOWN)			
Decode matrix (PO5)	Line 0	DECODER	DECO	/ /				
	Line 1		DEC1	/ /				
	Line 2		DEC2	/ /				
	Line 3		DEC3	/ /				
	Line 4		DEC4	/ /				
Output port	6/7		P06	1(P6/P7)	0 (P6)			
PLA (P06)	Line 0	PLA	PLA0	/ 00 /	/ /			
	Line 1		PLA1	/ 11 /	/ /			
	Line 2		PLA2	/ 22 /	/ /			
	Line 3		PLA3	/ 33 /	/ /			
	Line 4		PLA4	/ 44 /	/ /			
	Line 5		PLA5	/ 55 /	/ /			
	Line 6		PLA6	/ 66 /	/ /			
	Line 7		PLA7	/ 77 /	/ /			
	Line 8		PLA8	/ 88 /	/ /			
	Line 9		PLA9	/ 99 /	/ /			
	Line A		PLAA	/ AA /	/ /			
	Line B		PLAB	/ BB /	/ /			
	Line C		PLAC	/ CC /	/ /			
	Line D		PLAD	/ DD /	/ /			
	Line E		PLAE	/ EE /	/ /			
	Line F		PLAF	/ FF /	/ /			

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.3 to +7.0	V
V_{IN}	Input Voltage	-0.3 to V_{DD}+0.3	V
V_{OUT1}	Output Voltage (except $P0_5$,$P0_6$,$P1_0$)	-0.3 to V_{DD}+0.3	V
V_{OUT2}	Output Voltage ($P0_5$,$P0_6$,$P1_0$)	V_{DD} - 35 to V_{DD}+0.3	V
P_D	Power Dissipation	600	mW
Tsol	Soldering Temperature·Time	260 (10 SEC)	°C
Tstg	Sotrage Temperature	-55 to +125	°C
Topr	Operating Temperature	-20 to +70	°C

ALLOWABLE OPERATING CONDITIONS

SYMBOL	ITEM	RATING	UNIT
		V_{DD} = 4 to 6V	
Ta	Ambient Temperature	- 20 to +70	°C
V_{OUT}	Output Voltage ($P0_6$, $P0_5$, $P1_0$)	Max. V_{DD} - 32	V
f_x	X'tal Operating Frequency	40 to 400	KHz
tcy	Cycle Time	10 to 100	µs

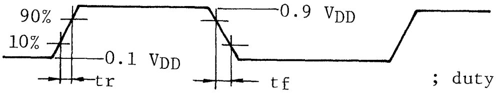
DC CHARACTERISTICS (Ta = -20°C to +70°C, V_{DD} = 4V to 6V)

SYMBOL	PARAMETER		TEST CONDITION	RATING			UNIT
				MIN.	TYP.	MAX.	
V_{IH}	Input High Voltage			VDDx0.7	VDDx0.55	VDD	V
V_{IHS}	Input High Voltage (Schmitt)			VDDx0.85	VDDx0.75	VDD	
V_{IHC}	Input High Voltage (XIN input)			VDDx0.75	-	VDD	
V_{IL}	Input Low Voltage			0	VDDx0.45	VDDx0.3	
V_{ILS}	Input Low Voltage (Schmitt)			0	VDDx0.35	VDDx0.15	
V_{ILC}	Input Low Voltage (XIN input)			-	-	VDDx0.25	
I_{IH}	Input High Current		VDD=6V, VIN=6V	-	-	20	μA
I_{IL}	Input Low Current		VDD=6V, VIN=0V	-	-	-20	
R_{IN}	Input Resistance (PI5)		VDD=5V	75	150	350	KΩ
V_{OH}	Output High Voltage		VDD=5V, Output Open	4.7	4.9	-	V
V_{OL}	Output Low Voltage			-	0.1	0.3	
I_{OH}	Output High Current		VDD=4.5V,VOH=2.4V	-0.7	-	-	mA
I_{OH1}	Output High Current	(P06)	VDD=4.5V,VOH=2.5V	-1	-	-	
		(P05,P10)	VDD=4.5V,VOH=2.5V	-7	-	-	
I_{OL}	Output Low Current		VDD=4.5V,VOL=0.45V	1.6	-	-	
I_{LO}	Output Leak Current (P05, P06, P10)		VDD=6V, VOUT=-26V	-	-	-20	μA
I_{DDO}	VDD Supply Current in Normal Operation (f_X= 400kHz)		VDD=6V,VIN=5.9V/0.1V (all valid) PI5 Open CL=50pF Output Open	-	400	1200	μA
I_{DDH}	VDD Supply Current in Hold Operation (f_X= 400kHz)			-	150	450	

Note 1: Typical values are at Ta=25°C and V_{DD}=5V.

2: Output characteristic excludes X_{OUT} terminal.

3: X_{IN} input waveform at the time of measuring V_{DD} supply current.



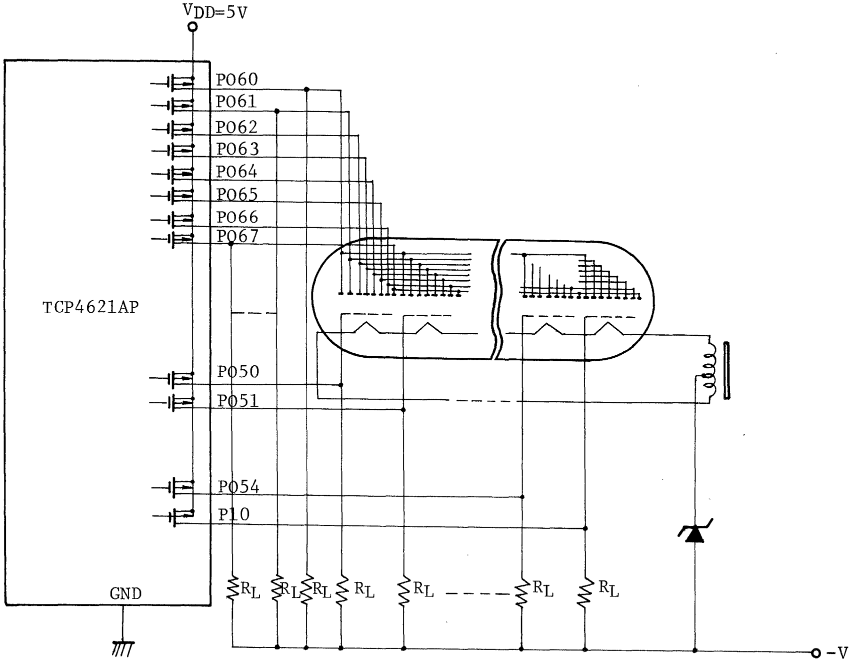
; duty = 50%

; tr, tf < 50 ns

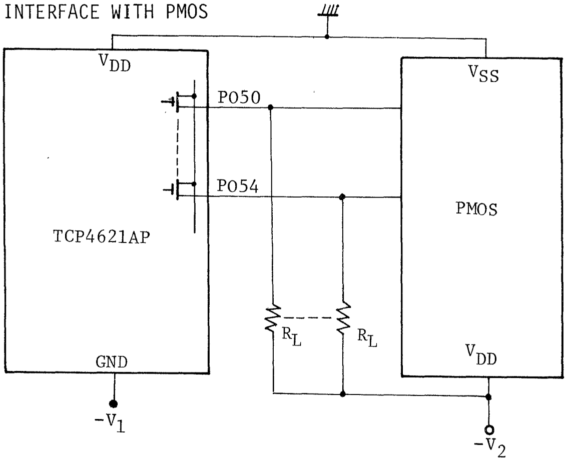
AC CHARACTERISTICS (Ta = -20 to +70°C, V_{DD} = 4 to 6V)

SYMBOL	PARAMETER	TESTCONDITION	MIN.	TYP.	MAX.	UNIT
t_{WXIN}	X_{IN} Pulse Width	External Input V_{IN}=V_{IHC}/V_{ILC}	0.4/f_X	-	0.6/f_X	SEC
t_{WRESET}	$\overline{\text{RESET}}$ Pulse Width	V_{IN}=V_{IHS}/V_{ILS}	2 tcy	-	-	μS
t_{WINT}	$\overline{\text{INT}}$ Pulse Width		2 tcy	-	-	
t_{WPI60}	PI60 Pulse Width		2 tcy	-	-	

INTERFACE OF FLUORESCENT DIAPLAY TUBE



INTERFACE WITH PMOS



TCP4632BF

C^2-MOS DIGITAL INTEGRATED CIRCUIT
SILICON MONOLITHIC

PRELIMINARY

GENERAL DESCRIPTION

The TCP4632BF is a C^2MOS 4-bit single-chip microcomputer capable of directly driving liquid crystal display provided with 1/3 or 1/4 duty, and is most suitable for use in a wide range of application fields, such as a system requiring low power dissipation.

The TCP4632BF contains the controller and driver for driving the liquid crystal display (hereinafter referred to as LCD) to the TCP4630AP. For the use and study of this device, refer to the technical data for the TCP4630AP in addition to this technical data.

FEATURES

o C^2MOS 4-bit single-chip microcomputer of low power dissipation

o Memory capacity

Program region (ROM)	3072 words x 8-bit
Data region (RAM)	160 words x 4-bit (including RAM for display)

o Instruction execution time: 10 μs (400kHz ceramic oscillator/IFT)

o 52 instructions

o Built-in LCD controller
1/3 or 1/4 duty (designated by mask options)

o Built-in LCD direct driver

Common output	4 pins
Segment output	24 pins

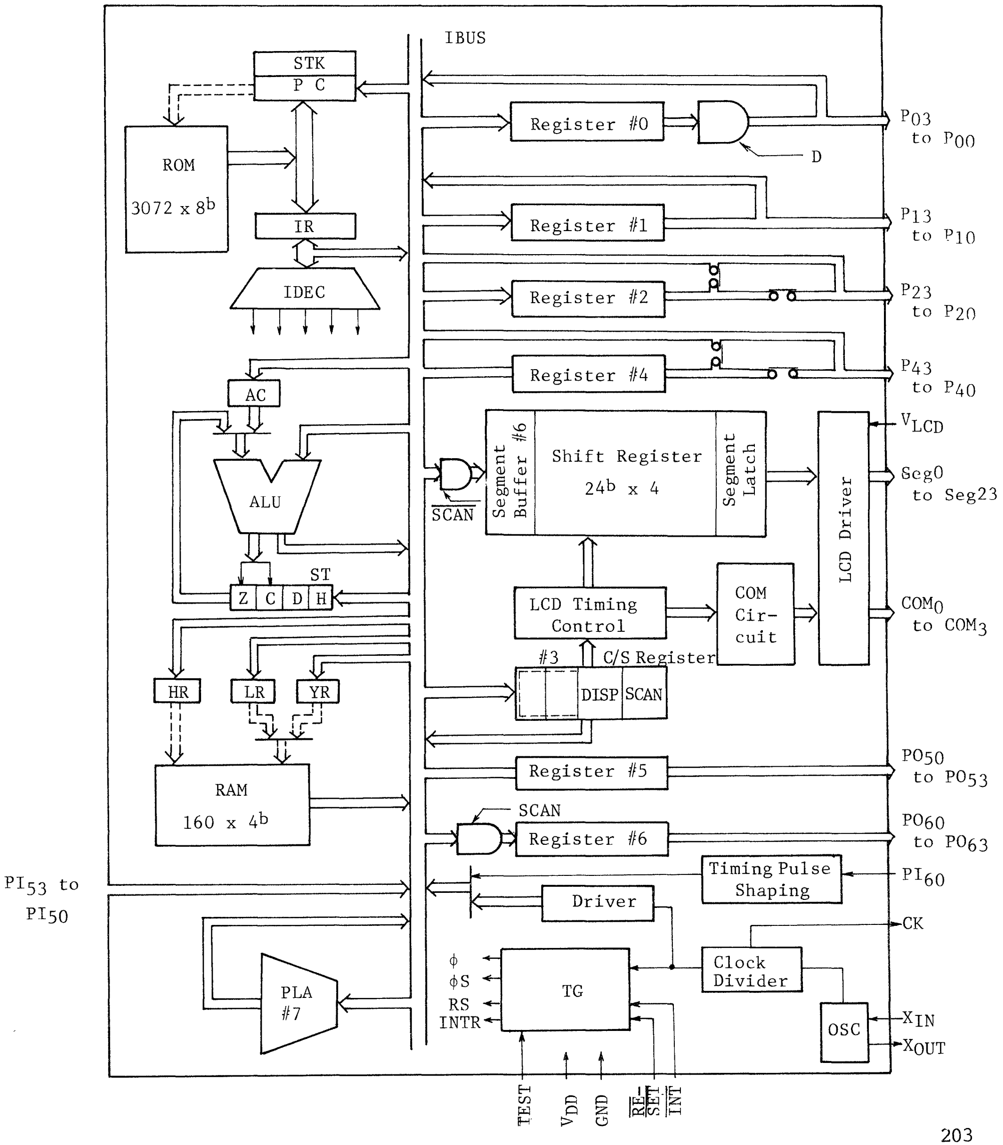
o Input/Output port

I/O port	1 x 4-bit (I/O can be switched by programs.)
	1 x 4-bit (I/O can be designated by mask options.)
	1 x 4-bit (I/O can be designated by mask options.)

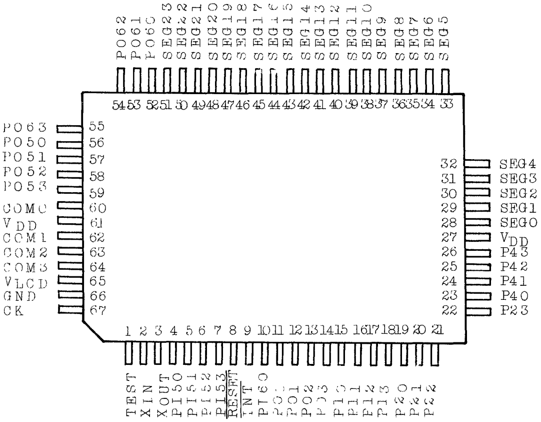
Output port	3 x 4-bit	(General purpose output port)
Input port	1 x 4-bit	(Pull-up/Pull-down resistor with input resistance,can be selected.)
	1 x 4-bit	

- o External power supply is available for LCD (V_{LCD})
- o Built-in display decoder (Can be designated by mask options with decoder data.)
- o Blanking operation
- o Hold operation (LCD can be displayed during hold operation.)
- o Power-down mode (Display blanking)
- o 67-pin plastic package

BLOCK DIAGRAM



PIN CONNECTIONS (Top View)



PIN NAMES/PIN DESCRIPTION

Pin Name	Input/ Output	TCP4620AP/30AP		TCP4632BF	
		No.of pins	Function	No.of pins	Function
TEST	Input	1	LSI test input (always kept set at 0.)	1	LSI test input (always kept set at 0.)
X_{IN}	Input	1	Oscillator connecting terminal	1	Oscillator connecting terminal
X_{OUT}	Output	1	Oscillator connecting terminal	1	Oscillator connecting terminal
PI_5	Input	4	General purpose input port	4	General purpose input port
$\overline{RESET}$	Input	1	Reset input	1	Reset input
$\overline{INT}$	Input	1	Interrupt input	1	Interrupt input
PI_6	Input	1	General purpose input	1	General purpose input

Pin Name	Input/ Output	TCP4620AP/30AP		TCP4632BF	
		No.of pins	Function	No.of pins	Function
P_0	I/O	4	I/O port	4	I/O port
P_1	Output	4	General purpose output port	4	General purpose output port
P_2	Input/ Output	4	General purpose input/ output port	4	General purpose input/ output port
P_4	Input/ Output	4	General purpose input/ output port	4	General purpose input/ output port
V_{DD}		1	Power supply	2	Power supply
$SEG_{0\sim23}$	Output			24	LCD segment driver output
$P0_6$	Output	4	PLA output port	4	General purpose output port
$P0_7$	Output	4	PLA output port		
$P0_5$	Output	5	Decode matrix output port	4	General purpose output port
$COM_{0\sim3}$				4	LCD common driver output
V_{LCD}				1	LCD external power supply terminal
GND		1	GND	1	GND
CK	Output	1	External timing output	1	External timing output

CONFIGURATION OF PORT AND REGISTER

	Name	Symbol	Pin Name	Reg. No.	Function	Remarks
*	I/O Port	P_0	P_{03} to P_{00}	0	I/O Port	I/O switching is controlled by D flag of ST. This port can be designated as a dedicated output port by mask options.
	Output Port	P_1	P_{13} to P_{10}	1	Output Port	General purpose output port. The contents of register can be referred to by a program.
*	Input/Output Port	P_2	P_{23} to P_{20}	2	Input/Output Port	Input/output can be designated by mask options. The contents of register can be referred to by a program.
*	Command Register	C/S	———	3	Command Register	LCD display is controlled by a 2-bit register. The contents of register can be referred to by a program.
*	Input/Output Port	P_4	P_{43} to P_{40}	4	Input/Output Port	Input/output can be designated by mask options. The contents of register can be referred to by a program.
	Input Port	PI_5	PI_{53} to PI_{50}	5	Input Port	General purpose input port with 150kΩ (TYP.) resistance. Pull-up/Pull-down designation can be made by mask options.
*	Output Port	PO_5	PO_{53} to PO_{50}		Output Port	General purpose output port. The contents of register can not be referred to by a program.
	Input Port	PI_6	PI_{60}	6	Input Port	1-bit general purpose input port. (Built-in Schmitt circuit).
*	Output Port	PO_6	PO_{63} to PO_{60}		Output Port	General purpose output port. The contents of register can not be referred to by a program.
*	PLA	PLA	———	7	PLA	PLA. PLA can be used as a display data decoder or a data decoder. The contents of decode is designated by mask options.

Note The asterisk (*) indicates the points of difference between TCP 4620AP/TCP4630AP and this device.

(a) Neither command register nor PLA (available for reference by a program) is contained in the TCP4620AP/TCP4630AP.

(b) Since PO_5 and PO_6 contain neither decode matrix nor PLA, the number of output pins of each port is 4.

(c) Care should be taken to the fact that P_0, P_2, P_4, PO_5 and PO_6 of this device are different from those of TCP4620AP/TCP4630AP in electrical characteristics.

LCD CONTROLLER/DRIVER

The TCP4632BF contains a driver circuit capable of directly, dynamically driving the LCD provided with 1/3 or 1/4 duty by the 1/3 bias method.

LCD connection terminals amount to 29 in total as follows:

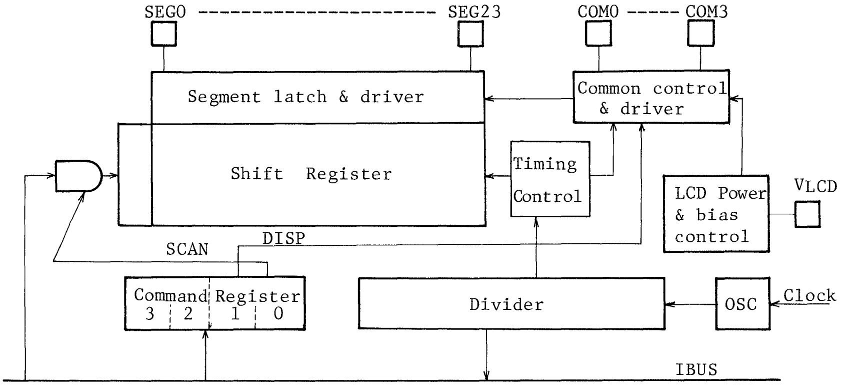
Common driver output	(COM0 to COM3)	4
Segment driver output	(SEG0 to SEG23)	24
V_{LCD} terminal as LCD driving power terminal		1

The display ability makes it possible directly to drive the following LCDs according to the number of time divisions:

(a) 1/4 duty (1/3 bias) LCD
Max. 96 segments (12 digits x 8 segments)

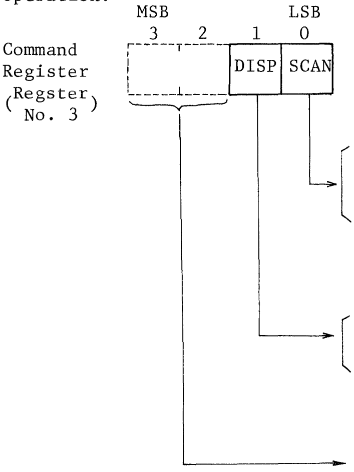
(b) 1/3 duty (1/3 bias) LCD
Max. 72 segments (8 digits x 9 segments)

BLOCK DIAGRAM OF LCD DRIVE CONTROLLER



CONTROL OF DRIVE CIRCUIT

The operation of LCD drive circuit is controlled by the contents of command register. The command register is a 2-bit register and is accessed as register No. '3'. Both of two bits are reset to "0" by an initialize operation.



Control of Display/Display Data Rewrite

0: Rewritable status of display data (The use of P06 is irhibited.)
1: Display status of display data (The use of P06 is permitted)

Control of Dispaly

0: Establishment of blanking status or blanking status
1: Release of blanking status or display status

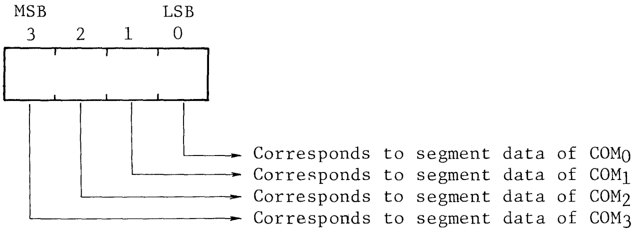
Undefined Bit

In case of reference by program, data is undefined.

DISPLAY OPERATION

SETUP OF DISPLAY DATA

The TCP4632BF internal processing data corresponds to LCD drive terminal in such a way as bit 0, bit 1, bit 2 and bit 3 of the processing data correspond to each segment data of COM0, COM1, COM2, and COM3, respectively.


MSB
LSB
3
2
1
0
Corresponds to segment data of COM$_0$
Corresponds to segment data of COM$_1$
Corresponds to segment data of COM$_2$
Corresponds to segment data of COM$_3$

The conversion of LCD display data is arbitrarily set up by a program or by a built-in PLA register (setup of the contents by mask options).

TRANSFER OF DISPLAY DATA

LCD display data setup by a program or PLA register is displayed by transferring it to the shift register of LCD driver by the program.

The transfer method is divided into two categories: one is that the converted data is transferred as it is, and another is that a displaying data region set up in the data memory (RAM) is transferred after it has been once stored. Even by either method, each bit of LCD segment (dot) and TCP4632BF internal processing data corresponds to each other by one to one.

Address \ Bit	MSB 3	2	1	LSB 0	
0					(SEG_0)
1					(SEG_1)
2					(SEG_2)
3					(SEG_3)
	≈	≈	≈	≈	
22					(SEG_{22})
23					(SEG_{23})
	(COM3)	(COM2)	(COM1)	(COM0)	

In case data memory is provided with display region

Register \ Bit	MSB 3	2	1	LSB 0	
					(SEGi)
	(COM3)	(COM2)	(COM1)	(COM0)	(i = 0 to 23)

In case processing data intact is transferred

Each bit of display data expresses data of segment (dot) equivalent to (SEGi COMj : $0 \leq i \leq 23$, $0 \leq j \leq 3$). When the data displays "1", the display lights. In case of the use of LCD with 1/3 futy (COM0 to COM2), the data of bit 3 (MSB) has no meaning as a LCD display data.

Transfer and rewrite of display data is made by a program as follows:

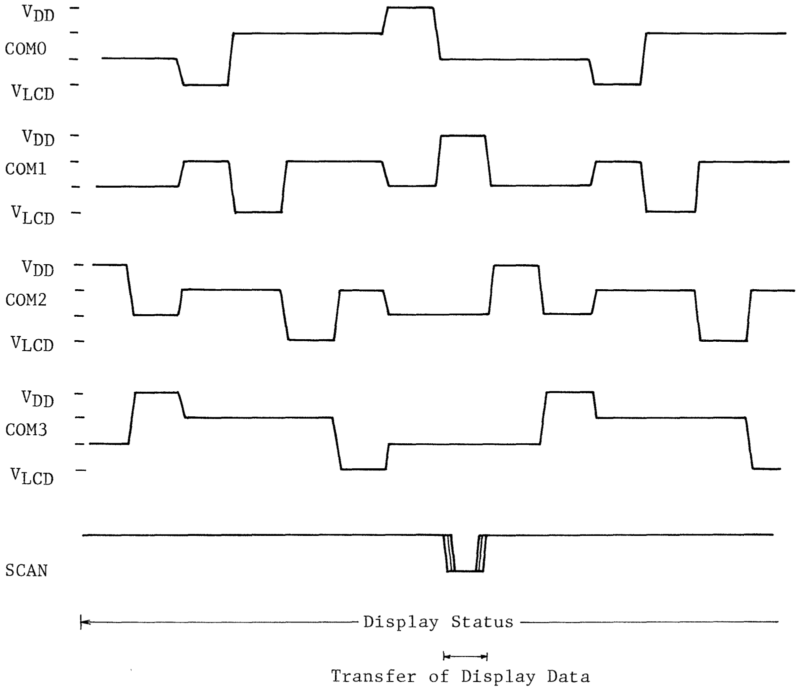
First, clear SCAN bit to '0'. (Immediately after initialization, the bit is at '0'.) Next, transfer 4-bit data equivalent to SEG. 23 to Register No. 6 (SAR 6), and then transfer 4-bit data equivalent to SEG 22. Continuously transfer 4-bit data in sequence by programs until the transfer of data equivalent to SEG. 0 is comleted.

This operation sllows 4×24-bit data to be written into the shift register within LCD drive circuit. After the data transfer of 24 words has been finished, SCAN bit is set to '1'. When SCAN bit goes to '1', writing into the shift register is inhibited, and LCD begins to be driven by the data newly held. Data switching is made from COMj to come next (data output equivalent to bit j to SEGi) and is automatically scanned to the next COMj+1.

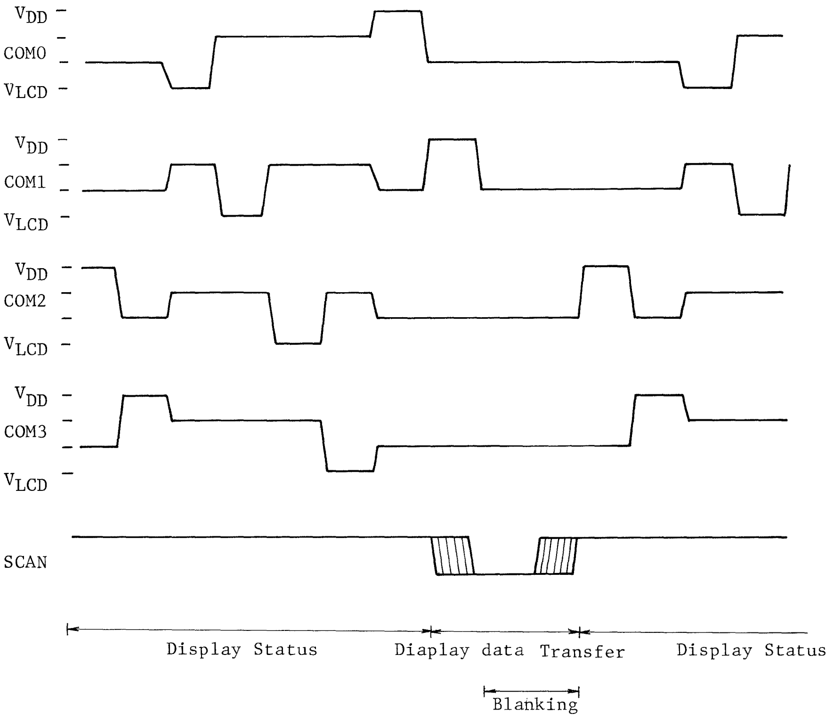
In the case that the period (period of SCAN = 0) of 24-word data transfer by means of a program is within the one COM period (1/4fF), the blank period does not come into the transfer period, and the display contents are switched. However, in the case that the period (period of SCAN = 0) of 24-word data transfer by means of a program crosses switching of COM (this phenomenon presents by timing of data transfer period or it presents when the data transfer period becomes longer than one COM period), no correct display data is secured during the period; thus, a blank is automatically inserted in the period.

When the display data is transferred to a shift register equivalent to Register No.6 during the period of SCAN = 0, all of 24 words must be transferred to the shift register. However, if SEG0 to SEGk-1 outputs only, not all of SEG0 ∿ SEG23 outputs, are used, k word only can be transferred.

In case of SCAN = 0, write in Register No. 6 (SAR = 6) means write in the shift register as a display data. In case of SCAN = 1, however, it means write in output port PO_6.



As the display data transfer period is short, it does not come in the blanking period.



As the display data transfer period is long, it comes in the blanking period.

LCD FRAME FREQUENCY

Frame frequency (LCD drive frequency) is given by the built-in devider. The frame frequency f_F is given from internal clock frequency f_{CP} by the following formula:

$$f_F = f_{CP} \times \frac{1}{2^m \times n}$$

Where,m is the number of divider stages. When fx = 400 kHz and f = 200 kHz, m is set to 9. n is the number of time division. Therefore,

In case of 1/3 duty	f_F ≒ 130 Hz
In case of 1/4 duty	f_F ≒ 98 Hz

The period equivalent to one COM is

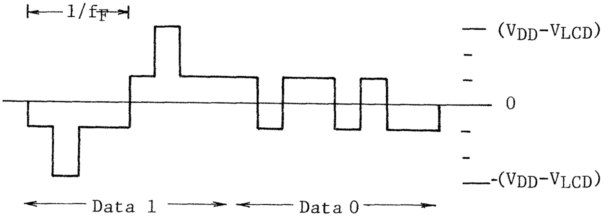
$$\frac{2m}{f_{CP}} \quad (\fallingdotseq \frac{2m-1}{f_{CY}})$$

therefore, when m = 9, the period is equivalent to 256 steps.

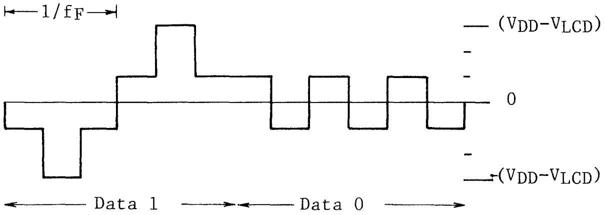
LCD DRIVE WAVEFORM

TCP4632BF is capable of directly, dynamically driving the LCD provided with 1/3 or 1/4 duty by 1/3 bias method.

In case of 1/3 or 1/4 duty, the voltage applied to LCD is + (V_{DD} - V_{LCD}) to -(V_{DD} - V_{LCD}).



1/4 duty (1/3 bias) drive

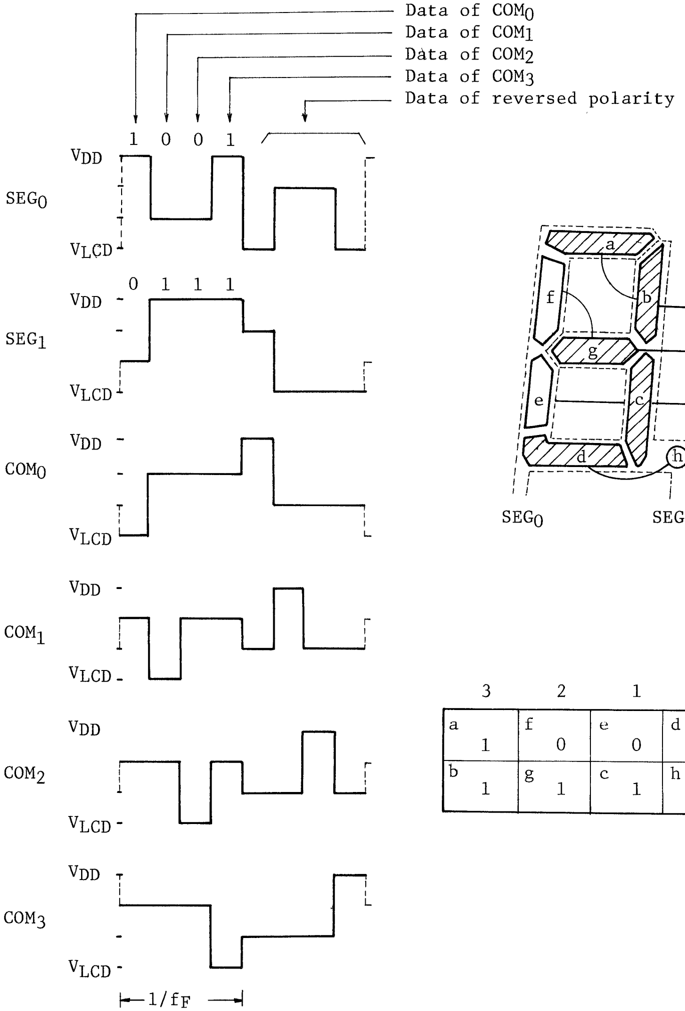


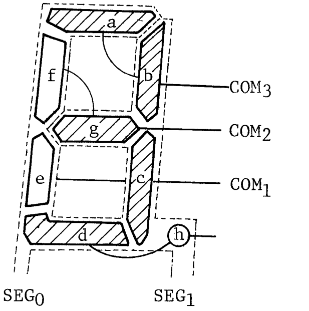
1/3 duty (1/3 bias) drive

(Note) f_F : LCD frame frequency

V_{LCD} : V_{LCD} terminal voltage

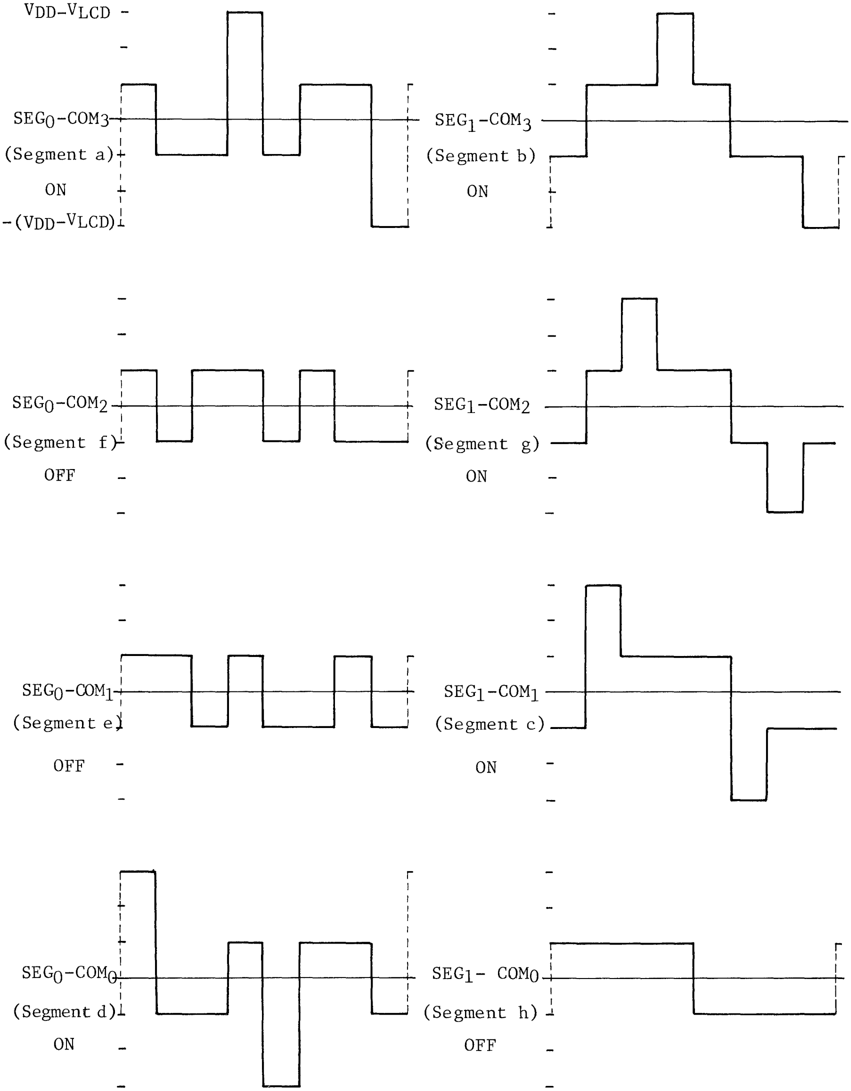
Each terminal waveform in case of displaying data "3" at 1/4 duty is as follows:





3	2	1	0	
a 1	f 0	e 0	d 1	SEG$_0$
b 1	g 1	c 1	h 0	SEG$_1$

The voltage between each SEG - COM is as follows:



LCD DRIVE VOLTAGE

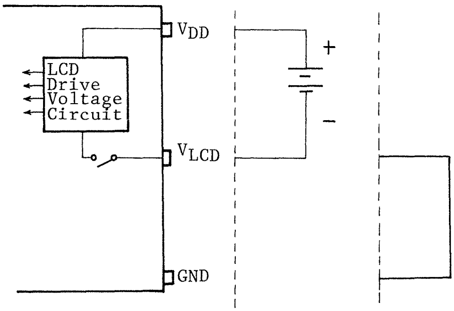
V_{LCD} is an input terminal of LCD driving power. LCD drive voltage is required to be supplied between V_{DD}-V_{LCD}.

If the operation voltage of the TCP4632BF is identical with the LCD drive voltage, V_{LCD} is connected to GND.

The supply of drive voltage generated by LCD drive circuit is controlled within this device according to the operating status of CPU. In the following cases that there is a possibility of keeping LCD drive circuit stationary for a long period of time, the supply of LCD drive voltage is cut with the built-in switch.

① Initialize operation

② SCAN = 0 and H = 1



The switch turned off by initialize operation is turned on by setting the SCAN bit of command register to '1', thus resulting in the supply of LCD drive voltage. After that, even if the SCAN bit of command register is reset to '0', LCD drive voltage is kept supplied.

For performing hold operation, H must be set to '1' in a state of SCAN = 1 and at time of normal LCD display because rewrite of display data has been already finished. In this case, LCD drive circuit is still in operation to continue display. However, while display data is being rewritten, if H is set to '1' at time of SCAN = 0, the supply of LCD drive voltage is cut.

When the supply of LCD drive voltage is cut with the built-in switch, all of SEG and COM outputs attain V_{DD} lelvel.

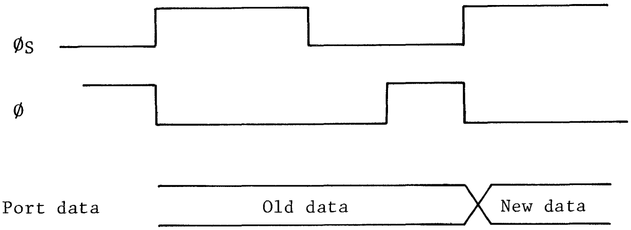
OUTPUT PORT (P05, P06)

P05 and P06 are 4-bit general purpose output ports. Each port is provided with a latch, and its contents are held until they are rewritten by a program. Each bit is reset to "0" by initialize operation, but the contents being held cannot be referred to by the program.

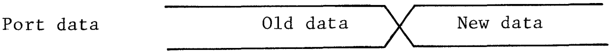
The authorization/inhibit of content rewrite by the program of P06 is controlled by SCAN bit of command register.

Command register SCAN	Object of content rewrite by program
0	Output to shift register (LCD segment data)
1	Output to P06

Unlike P05 and P06 of TCP4630AP, those of TCP4632BF do not contain decode matrix, and PLA; therefore, there is difference between them in output timing.



Output timing of P05 and P06 of TCP4630AP



Output timing of P05 and P06 of TCP4632BF

PLA

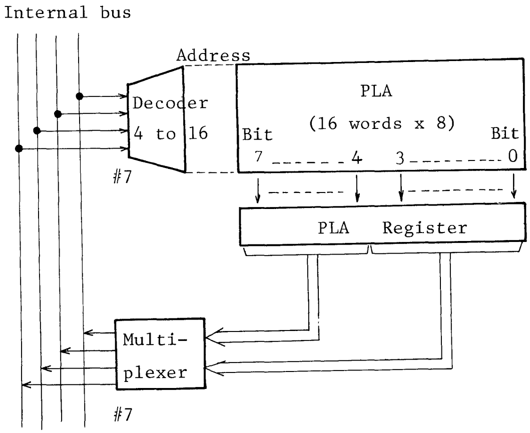
A program can make reference of PLA by use of a decoder for display data or the like. If 4-bit BCD data is written in PLA (Register No.7) by a program, one of sixteen addresses is selected by 4-to-16 decoder within PLA, and the corresponding 8-bit data is written in PLA register.

PLA contents of PLA register can be read out by executing the instruction for making reference of PLA. How to read out 8-bit data of PLA register varies with option designation of 1/3 duty or 1/4 duty. PLA register is not affected by reset operations.

(a) Designating 1/4 duty by options

The 8-bit data can be read out by executing the instruction for making reference of PLA register two times. In other words, 4-bit (bit 3 to 0) data on the lower level can be read out by the first execution of the instruction (LAR 7) for making reference of PLA after BCD data has been written in PLA (SAR 7), and similarly 4-bit (bit 7 to 4) data on the higher level can be read out by the second execution of the same instruction.

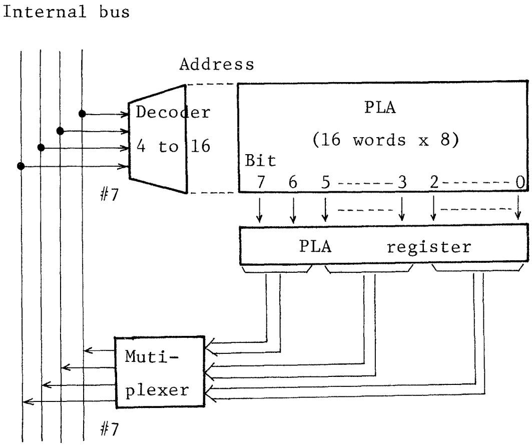
When the same data is read out again, and when PLA register is required to be updated, the reabout must be performed after BCD data has been written in the PLA over again.



(b) Designating 1/3 duty by options

The 8-bit data can be read out by executing the instruction for making reference of PLA register three times. In other words, 3-bit (bit 2 to 0) data on the lower level can be read out by the first execution of the instruction (LAR 7) for making reference of PAL after BCD data has been written in PLA (SAR 7), and similarly 3-bit (bit 5 to 3) data on the intermediate level can be read out by the second execution of the same instruction, and also 2-bit (bit 7 to 6) data on the higher level by the third execution of the same instruction, respectively.

While the reference instruction above mentioned is being executed, 1 bit (bit 3) on MSB side of the lower or intermediate level and 2 bits (bit 3 and 2) on MSB side of the higher level are always read out as zero. When the same data is read out again, and when PLA register is required to be updated, the readout is performed after BCD data has been written in the PLA over again.

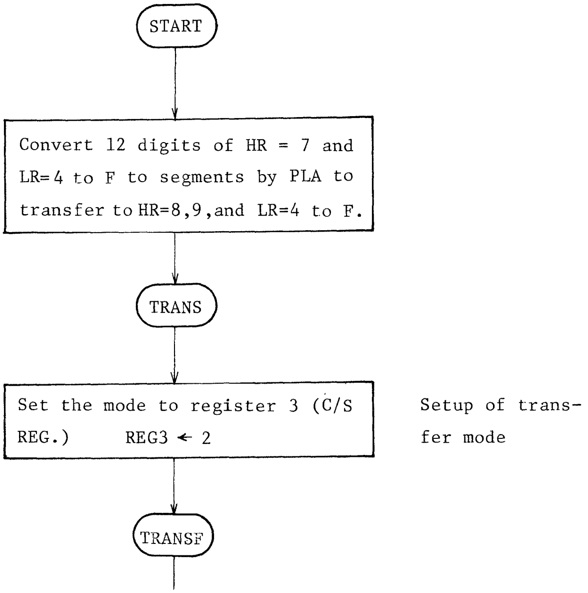


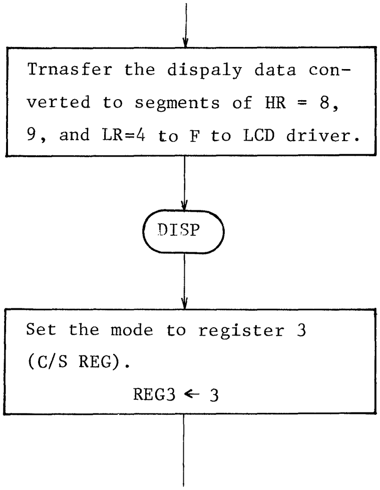
LCD DATA TRANSFER PROGRAM - EXAMPLE

HR \ LR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7					1 Digit	2	3	4	5	6	7	8	9	10	11	12- digit
8					SEG23	21	19	17	15	13	11	9	7	5	3	SEG 1
9					SEG22	20	18	16	14	12	10	8	6	4	2	SEG 0

Example of a program which the data of HR = 7 and LR = 4 to F are displayed on LCD through segment conversion

LCD: 1/4 duty, 12 digits (24 Seg.)


START
Convert 12 digits of HR = 7 and LR=4 to F to segments by PLA to transfer to HR=8,9,and LR=4 to F.
TRANS
Set the mode to register 3 (C/S REG.) REG3 ← 2
Setup of transfer mode
TRANSF



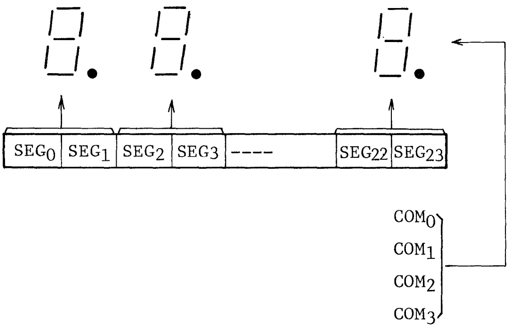
Trnasfer the dispaly data converted to segments of HR = 8, 9, and LR=4 to F to LCD driver.

Set the mode to register 3 (C/S REG).

REG3 ← 3

Setting of display mode

```
                    10   ;    ***   TCP4632BF LCD DATA TRANSFER PROGRAM
                    20   ;
0100                30              ORG  100H
0100   64           40   START:     LLI  4
0101   77           50   NEXT:      LHI  7
0102   04           60              LAM
0103   2F           70              SAR  7
0104   78           80              LHI  8
0105   17           90              LAR  7
0106   0C           100             SAM
0107   79           110             LHI  9
0108   17           120             LAR  7
0109   0C           130             SAM
010A   02           140             ICL
010B   06           150             LAL
010C   90           160             ADI  0
010D   D110         170             BCC  NEXT
010F   42           180   TRANS:    LAI  2
0110   2B           190             SAR  3     ;  DISP=1,SCAN=0   SET
0111   64           200             LLI  4
0112   78           210   TRANSF:   LHI  8
0113   04           220             LAM
0114   2E           230             SAR  6
0115   79           240             LHI  9
0116   04           250             LAM
0117   2E           260             SAR  6
0118   02           270             ICL
0119   06           280             LAL
011A   90           290             ADI  0
011B   D211         300             BCC  TRANSF
011D   43           310   DISP:     LAI  3
011E   2B           320             SAR  3     ;  DISP=SCAN=1  SET
                    330  ;
                    340  ;
```



TCP4632BF MASK OPTIONS

Oscillation frequency			OSC	TYP032 050 400K ceramic IFT				
Dividing ratio for internal clock			CP	02				
External timing output			CK	C1				
Counter	Divider 1 Input	COUNTER	PD	CP				
	Divider 3 Input		PDR	PD2				
	Reset timing		PDR	N				
Counter Buffer	Buffer 0 Input	COUNTER BUFFER	C0	PI60				
	Buffer 1 Input		C1	RD7				
	Buffer 2 Input		C2	RDA				
	Buffer 3 Input		C3	RDD				
H flag			HOLD	H				
Restart condition			RSTH	C3				
Input/Output port	Port 0		STD	D(PROG)	1 (OUT)			
Input/Output port	Port 2		P2	/F/(OUT)	/F/(OUT)	/0/(IN)	/0/(IN)	/0/(IN)
	Port 4		P4	/F/(OUT)	/3/(IN/OUT)	/F/(OUT)	/3/(IN/OUT)	/0/(IN)
Input resistance (Input port 5)			P15	0 (UP)	1 (DOWN)			
P L A (#7)	Line 0	P L A	PLA0	/ /				
	Line 1		PLA1	/ /				
	Line 2		PLA2	/ /				
	Line 3		PLA3	/ /				
	Line 4		PLA4	/ /				
	Line 5		PLA5	/ /				
	Line 6		PLA6	/ /				
	Line 7		PLA7	/ /				
	Line 8		PLA8	/ /				
	Line 9		PLA9	/ /				
	Line A		PLAA	/ /				
	Line B		PLAB	/ /				
	Line C		PLAC	/ /				
	Line D		PLAD	/ /				
	Line E		PLAE	/ /				
	Line F		PLAF	/ /				
Oscillation Stop Control PI60			RSTX	N				
LCD Duty			DUT	4	3			

Toshiba 東芝 **INTEGRATED CIRCUIT**

TECHNICAL DATA

TCP4632BF

PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	-0.3 to +7.0	V
V_{IN}	Input Voltage	-0.3 to V_{DD} +0.3	V
V_{OUT}	Output Voltage	-0.3 to V_{DD} +0.3	V
P_D	Power Dissipation	400	mW
T_{sol}	Soldering Temperature·Time	260 (10 SEC)	°C
T_{slg}	Storage Temperature	-55 to +125	°C
T_{opr}	Operating Temperature	-20 to +70	°C

ALLOWABLE OPERATING CONDITIONS

SYMBOL	ITEM	RATING	UNIT
V_{DD}	Supply Voltage	4 to 6	V
Ta	Ambient Temperature	-20 to +70	°C
I_{OUT}	Max. Output Current	±3	mA
f_x	X'tal Operating Frequency	40 to 400	kHz
t_{CY}	Cycle Time	10 to 100	μs
V_{LCD}	LCD Supply Voltage (V_{LCD} is with respect to V_{DD})	$-V_{DD}$ to -2.7	V

DC CHARACTERISTICS (Ta = -20 to +70°C, V_{DD} = 4 to 6V)

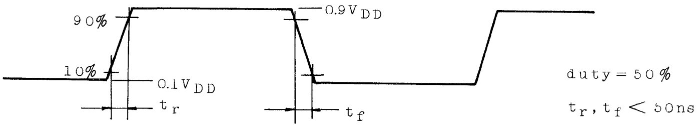
SYMBOL	PARAMETER	TEST CONDITION	MIN.	(Note 1) TYP.	MAX.	UNIT
V_{IH}	Input High Voltage		V_{DD}x0.7	-	V_{DD}	V
V_{IHS}	Input High Voltage (Schmitt)		V_{DD}x0.85	-	V_{DD}	
V_{IHC}	Input High Voltage(x_{IN} Input)		V_{DD}x0.75	-	V_{DD}	
V_{IL}	Input Low Voltage		0	-	V_{DD}x0.3	
V_{ILS}	Input Low Voltage (Schmitt)		0	-	V_{DD}x0.15	
V_{ILC}	Input Low Voltage (X_{IN} Input)		0	-	V_{DD}x0.25	
I_{IH}	Input High Current	V_{DD}=6V, V_{IN}=6V	-	-	20	µA
I_{IL}	Input Low Current	V_{DD}=6V, V_{IN}=0V	-	-	-20	
R_{IN}	Input Resistance (PI_5)	V_{DD}=5V	75	150	350	kΩ
V_{OH}	Output High Voltage (Port)	V_{DD}=5V, Output open	4.7	4.9	-	V
V_{OL}	Output Low Voltage (Port)	V_{DD}=5V, Output open	-	0.1	0.3	
I_{OH}	Output High Current (P_1)	V_{DD}=4.5V,V_{OH}=2.4V	-0.7	-	-	mA
I_{OH2}	Output High Current (P_0,P_2,P_4,CK)	V_{DD}=4.5V,V_{OH}=2.4V	-0.35	-	-	
I_{OH3}	Output High Current (P_5,P_6)	V_{DD}=4.5V,V_{OH}=2.4V	-0.15	-	-	
I_{OL}	Output Low Current (P_1)	V_{DD}=4.5V,V_{OL}=0.45V	1.6	-	-	
I_{OL2}	Output Low Current (P_0,P_2,P_4,CK)	V_{DD}=4.5V,V_{OL}=0.45V	0.8	-	-	
I_{OL3}	Output Low Current (P_5,P_6)	V_{DD}=4.5V,V_{OL}=0.45V	0.4	-	-	
R_{OS3}, R_{OS0}	Output Impedance (SEG)	V_{DD}=5V,V_{DD}-V_{LCD}=3V (Note 5) V_{OUT}=V_{LCD}+0.5V/ V_{DD}-0.5V	-	2	T.B.D	kΩ
R_{OC3}, R_{OC0}	Output Impedance (COM)		-	2	T.B.D	
R_{OS2}, R_{OS1}	Output Impedance (SEG)	V_{DD}=5V,V_{DD}-V_{LCD}=3V (Note 5) V_{OUT}=3+0.5V/ 4-0.5V	-	20	T.B.D	
R_{OC2}, R_{OC1}	Output Impedance (COM)		-	20	T.B.D	
V_{OS1} V_{OS2}	Output Intermediate Voltage (SEG,COM)	V_{DD}=5V,V_{DD}-V_{LCD}=3V	3 - 0.2 4 - 0.2	3 4	3 + 0.2 4 + 0.2	V
I_{DD0}	V_{DD} Supply Current in Normal Operation	(Note 4) f_x = 400 kHz	-	600	T.B.D	µA
I_{DDH}	V_{DD} Supply Current in Hold operation		-	250	T.B.D	

Note 1) Typical values are at Ta=25°C and V_{DD}=5V.

2) Output characteristic excludes X_{OUT} terminal.

3) Output current at the time when other terminals than those to be tested are open.

4) Test conditions of current dissipation
V_{DD} = 6V, V_{IN} = V_{IH}/V_{IL}(all valid), PI_5 Open, CL = 50pF
X_{IN} Input waveform



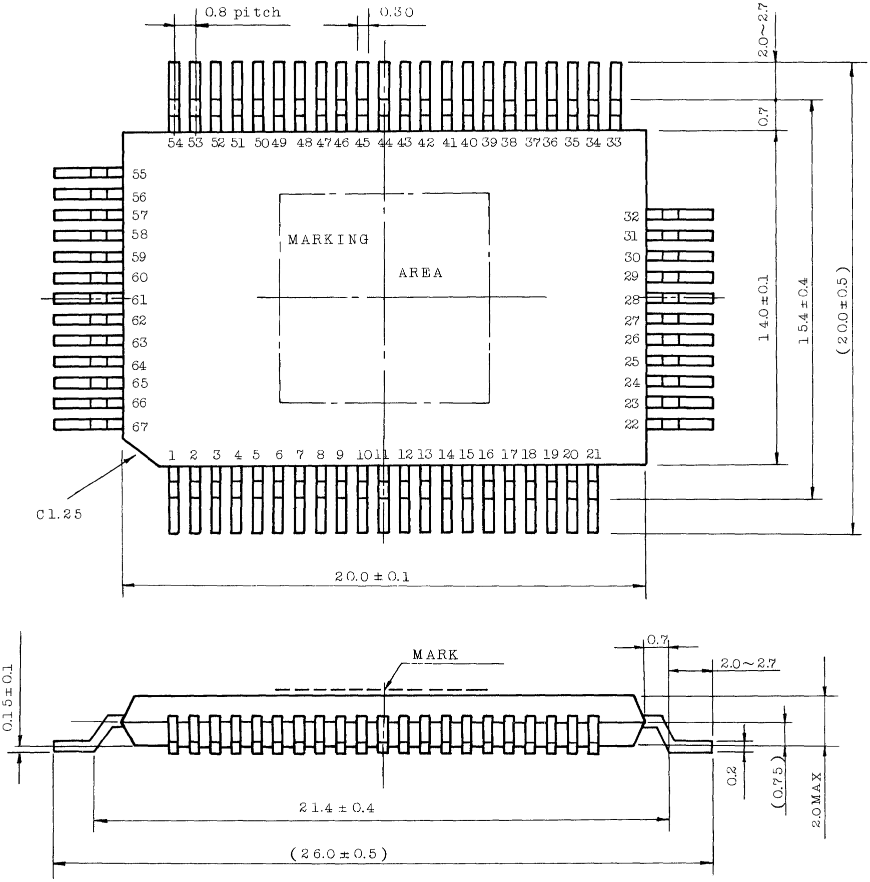
5) When supplying V_{LCD} power, When switching input/output.

TCP4632BF

PRELIMINARY

OUTLINE DRAWINGS

Unit mm



Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

"C^2MOS" DIGITAL INTEGRATED CIRCUIT

TCP4600AC

SILICON MONOLITHIC

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

GENERAL DESCRIPTION

TCP4600AC is a system development evaluator of TLCS-46A family, and makes it possible to configure a system equal to TCP4620AP/TCP4630AP by program memory (ROM) and the external circuits relative to mask options.

Since this technical data mainly makes mention of the function of an evaluator of TCP4600AC, it is recommended that instruction and detail be referred to the technical data of TCP4620AP/TCP4630AP.

FEATURES

o External ROM 4K x 8 Max.

o Internal RAM 160 x 4

o Single 5V Supply
Wide Operating Range: 4V to 6V

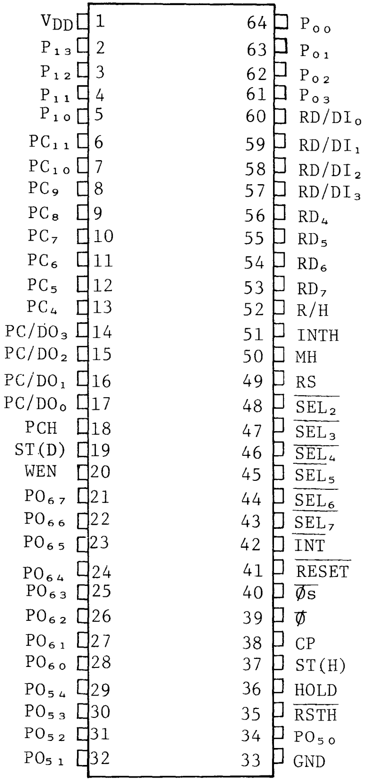
o Wide Operating Temperature Range: -10°C to 70°C

o Terminals for System Debugging

° External-Circuit Terminal for Realization of all the Mask Option of TLCS-46A

TCP4600AC

PIN CONNECTIONS (TOP VIEW)

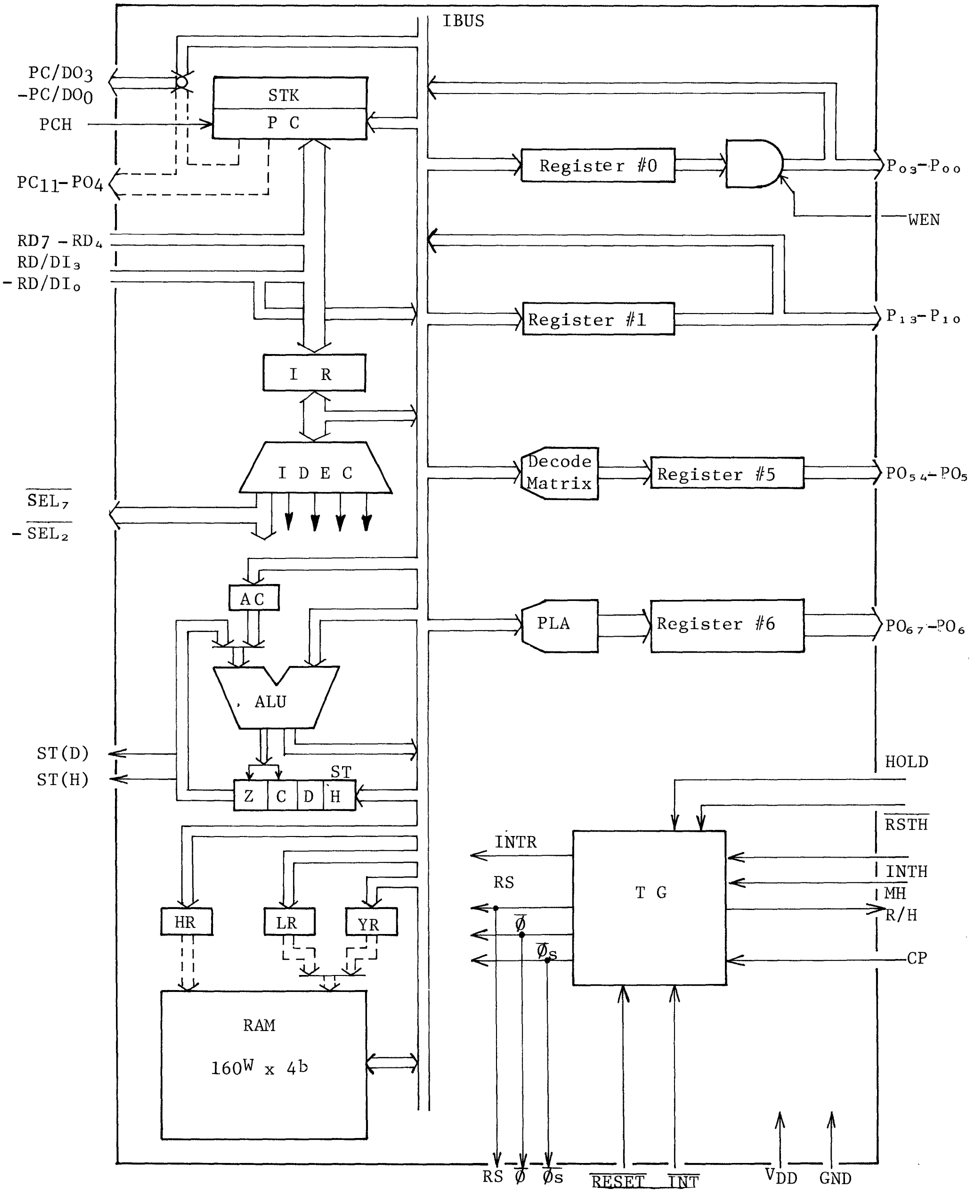


PIN NAMES & PIN DESCRIPTION

Pin Name	Input/Output	Function
P_{03} - P_{00}	Input/Output	4-bit general purpose I/O port State of input/output is designated by WEN terminal input. (common to TLCS-46A family)
P_{13} - P_{10}	Output	4-bit general purpose output port (common to TLCS-46A family)
PO_{54} - PO_{50}	Output	5-digit output port for display Content of built-in decode matrix is already designated (common to TLCS-46A family)
PO_{67} - PO_{60}	Output	8-segment output port for display Built-in PLA is already designated (common to TLCS-46A family)
$\overline{RESET}$	Input	Initialize Signal Input (without Schmitt circuit) (common to TLCS-46A family)
$\overline{INT}$	Input	Interrupt request signal input (without Schmitt circuit) (common to TLCS-46A family)
PC_{11} - PC_4	Output	The higher order 8-bit address output to external ROM
PC/DO_3 - PC/DO_0	Output	The lower order 4-bit address output to external ROM/Port data output
RD_7 - RD_4	Input	The higher order 4-bit data input from external ROM
RD/DI_3 - RD/DI_0	Input	The lower order 4-bit data input from external ROM/Input port
$\overline{SEL_7}$ - $\overline{SEL_2}$	Output	Selection signal output for external port register and input port

Pin Name	Input/Output	Function
WEN	Input	Designate signal input for Input/Output of port 0
HOLD	Input	Hold operation control signal input
$\overline{\text{RSTH}}$	Input	Restart control signal input at time of hold operation
ST(D)	Output	D flag output
ST(H)	Output	H flag output
CP	Input	Internal basic clock input
$\overline{\phi}_S$, $\overline{\phi}$	Output	Basic timing output
RS	Output	Internal reset signal output
MH	Input	Monitor hold control signal input (for system debug)
R/H	Output	Run/Hold status signal output (for system debug)
PCH	Input	Program counter hold control signal input (for system debug)
INTH	Input	Interrupt hold control signal input (for system debug)
V_{DD}		Power supply
GND		GND

BLOCK DIAGRAM



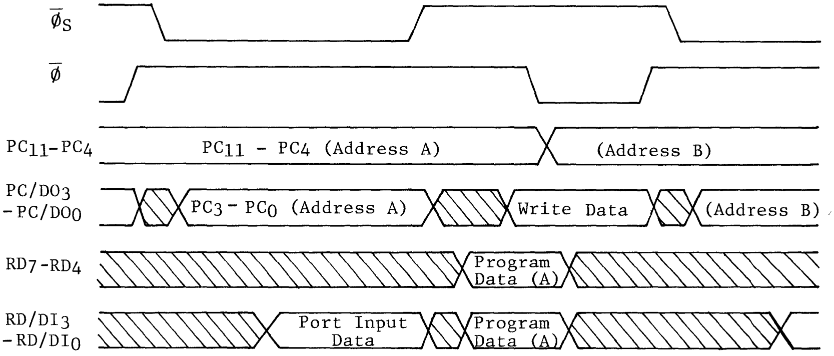
FUNCTIONAL DESCRIPTION

TCP4600AC is a system development evaluator which is used for developing TLCS-46A application system (program) and for confirming its operation. This makes it possible to form a configuration equal to TCP4620AP/TCP4630AP by program memory (hereinafter called ROM) and the external circuits relative to mask options.

1. Connection of Program Memory

Program counter (PC, 12 bits) output is used as address signal of external ROM (maximum capacity is 4096 words x 8 bits). Since the lower order 4 bits of address signal is multiplexed with the output of port data, it is necessary that they are externally separated.

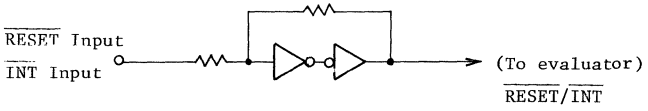
The data (8 bits) read out of ROM is read as instruction code in TCP4600AC. At this time, the data line of the lower order 4 bits is also used as port input data, so that it is necessary that data line is externally multiplexed.



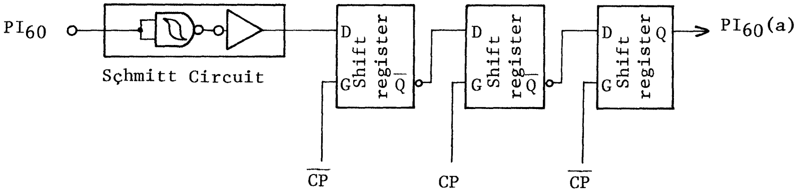
2. Schmitt Circuit

Each terminal of $\overline{\text{RESET}}$, $\overline{\text{INT}}$, and PI_{60} of TCP4620AP/TC4630AP are input terminals with Schmitt circuit. However, since no Schmitt circuit is contained in TCP4600AC, the circuit must be externally configured.

It is necessary to add shift register for timing shaping to PI_{60} terminal.


$\overline{\text{RESET}}$ Input
$\overline{\text{INT}}$ Input
(To evaluator)
$\overline{\text{RESET}}/\overline{\text{INT}}$

Schmitt Circuit



3. Oscillator and Divider

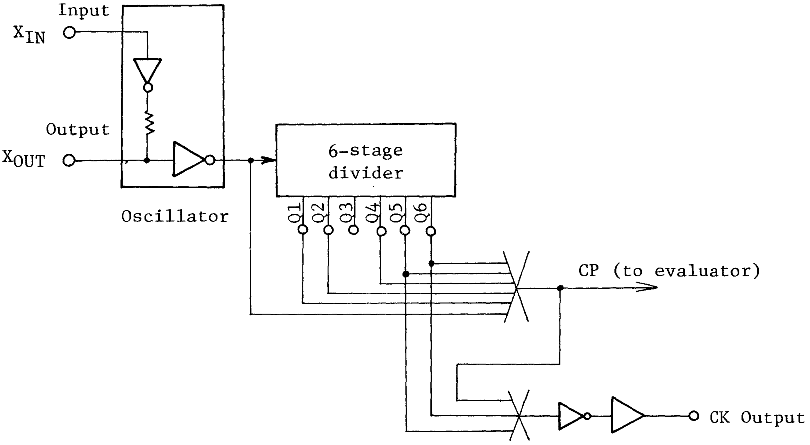
The mask options, which relate to the oscillator and divider of TCP4620AP/TCP4630AP, differ according to an oscillator to be used. For TCP4600AP these mask options are composed of external circuits.

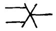
(1) Oscillator

The basic clock (CP) of TCP4600AC is externally supplied. In this case, the divider forming the basic clock as well as the oscillation circuits of the oscillator to be used are externally composed, and the output of a divider suitable for oscillation frequency is supplied, as a basic clock, to the CP terminal of TCP4600AC.

Since TCP4600AC is not provided with the CK terminal for external timing, a signal equivalent to the CK terminal must be also composed of external circuit.

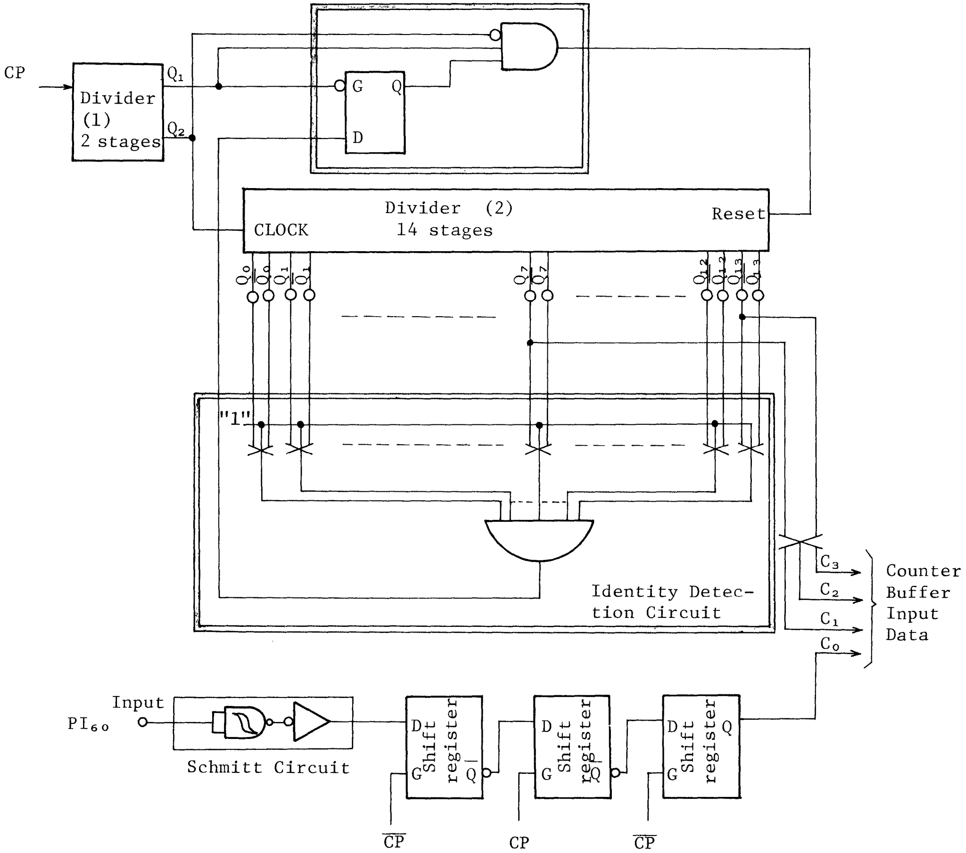
While the power is being supplied to the evaluator chip, the clock must be being supplied to the CP terminal at all times.



(Note) "  " means that either is connected.
(So with the following figures.)

(2) Divider

An example of cricuit configuration around a divider is shown as follows:



The portion of ▭ is required only when 100K Xtal is designated by mask options of mass production chip, and further 14-stage divider (2) is used as a resettable divider of count value "12500".

4. Input Port and Output Port

The input port and output port in TCP4620AP/TCP4630AP have plentiful mask options. In TCP4600AC, such mask options are composed of external circuits.

(1) Input port

The data of input port is input from RD/DI terminal. When the data of input port (port No. i) is input to RD/DI terminal during the period of time when the equation,

$$SEL_i \cdot \phi_S = 1 \text{ (i : port No.)},$$

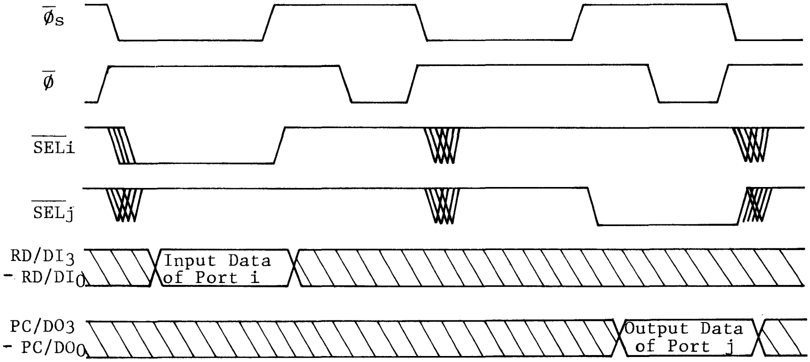
holds, TCP4600AC starts reading it as input port data.

(2) Output port

The data of output port is output from PC/DO terminal. Since TCP4600AC outputs the data of output port (port No. i) from PC/DO terminal during the period of time when the equation,

$$SEL_i \cdot \phi = 1 \text{ (i : port No.)},$$

hold, this data is written in the externally installed register. Further, since at time of system reset an internal initialize signal is output from RS terminal, it is necessary to clear (all "0") the port register by use of this signal and to initialize it.



(3) Port 0

Prot 0 is a port that can make switching of input/output by program (D flag of ST), and can be designated to a port exclusivelly used for output by mask options.

In TCP4600AC, the mask option of this port can be realized by use of each terminal of WEN and ST(D).

WEN terminal is an input line that defines whether port 0 is placed in an output mode or an input mode. When the terminal is at "1" level, it places port 0 in an output mode and when it is at "0" level, it places port 0 in an input mode.

A state of D flag of status register is output from ST (D) terminal.

o When input and output are switched by use of program,
  WEN = ST(D)
    WEN terminal is externally connected to ST(D) terminal.

o When port 0 is used as port exclusivelly for output,
  WEN = "1"
    WEN terminal is fixed to "1" level.

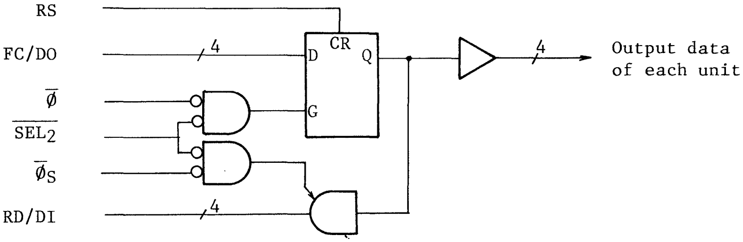
(4) Port 2 and Port 4

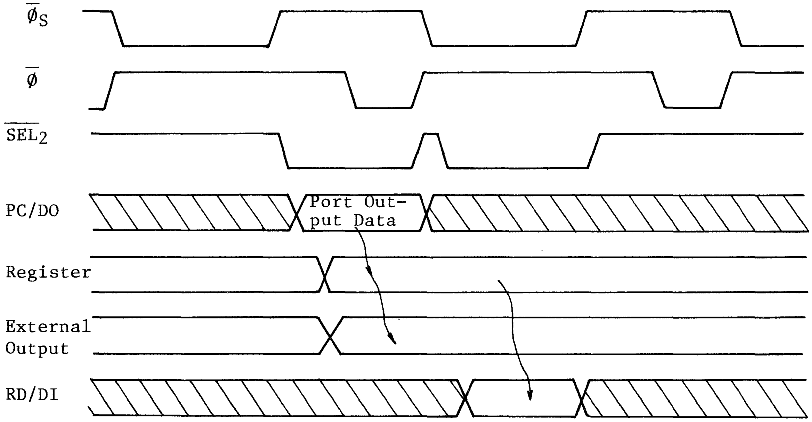
Port 2 and Port 4 are input/output selectable ports by use of mask options.
When these ports are used as input ports, the external input data is input to RD/DI terminal during the period when each of the following equations holds: $SEL_2 \cdot \phi_S = 1$ and $SEL_4 \cdot \phi_S = 1$.

On the other hand, when these prots are used as output ports, output data are output to PC/DO terminal during the period when each of the following equations holds:

$\mathrm{SEL}_2 \cdot \phi = 1$ and $\mathrm{SEL}_4 \cdot \phi = 1$.

Therfore, this data is written in the register externally installed, and is output to the outside through a buffer. And, at the time when the equations, such as $\mathrm{SEL}_2 \cdot \phi_S = 1$ and $\mathrm{SEL}_4 \cdot \phi_S = 1$, hold, the outputs of respective registers are input to RD/DI terminal. The externally installed register must be cleared (all "0") by the internal initialize signal from RS. terminal.





(5) Port 3

Port 3 is I/O port exclusively used for TCP4600AC. The function of port 3 is the same as the function of other ports.

This port is designated as register No. "3" even in case where it is used as either of input port or output port.

(6) Output port 5 and output port 6 (output port 7)

For TCP4600AC, each output terminal of P0$_5$ and P0$_6$ is available as output for display. However, when it is used as general output port or when it is used after changing the contents of decode matrix or PLA, it is configured by external circuit.

The contents of decode matrix and PLA built in TCP4600AC are as follows:

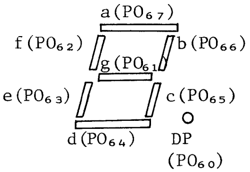
Contents of Decode Matrix

Write Data		F-6	5	4	3	2	1	0
Output State	P0$_{54}$	0	1	0	0	0	0	0
	P0$_{53}$	0	0	1	0	0	0	0
	P0$_{52}$	0	0	0	1	0	0	0
	P0$_{51}$	0	0	0	0	1	0	0
	P0$_{50}$	0	0	0	0	0	1	0

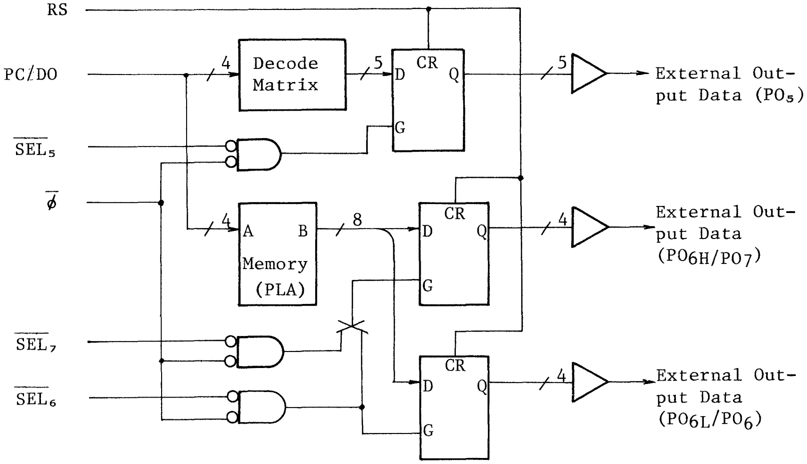
PLA

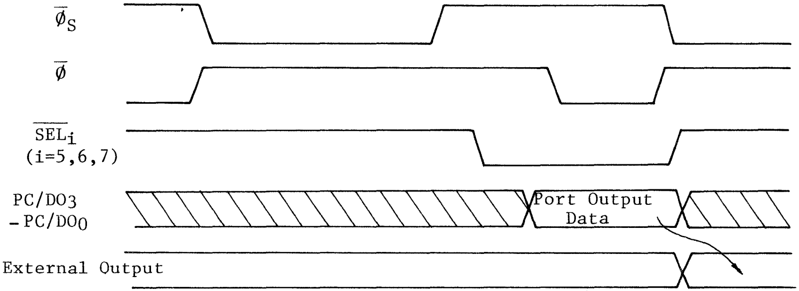
PLA can be expressed as memory of 16 words x 8 bits. The 4-bit write data for port 6 is the address of PLA, and the 8-bit data read out from PLA comes to the output data of the output port. When the data of 8 bits output at this time is expressed by hexadecimal 2 digits, PLA built-in TCP4600AC is as follows:

Address	Data	Form of Character
0	FC	0
1	60	1
2	DA	2
3	F2	3
4	66	4
5	B6	5
6	BE	6
7	E4	7
8	FE	8
9	F6	9
A	FD	0.
B	00	(Blank)
C	02	-
D	CE	P
E	9E	E
F	8E	F



When the built-in decode matrix and PLA are used changing their contents or are used as general output ports, they are externally configured by use of data output PC/DO, select signals $\overline{SEL_5}$, $\overline{SEL_6}$ and $\overline{SEL_7}$ and timing signal $\overline{\phi}$.





Output Timing of PO_5, PO_6 (PO_7)

5. Hold Operation

TCP4620AP/TCP4630AP is provided with hold function. In TCP4600AC, this function is configured by each terminal of HOLD, $\overline{\text{RSTH}}$ and ST(H), and the siganl of C_3 which is MSB of counter buffer input configured by external circuits. (For C_3, refer to the figure of divider.)

In case hold operation is performed:

HOLD = ST(H) : Designation of hold operation application

$\overline{\text{RSTH}}$ = C_3 : Designation of hold operation release signal

Each terminal of HOLD and ST(H) is externally connected. C_3 signal that is MSB of counter buffer is input to $\overline{\text{RSTH}}$ terminal.

In case hold operation is not performed:

HOLD = "0" : Designation of no use of hold operation

$\overline{\text{RSTH}}$ = "0" : Designation of no use of hold operation

Each terminal of HOLD and $\overline{\text{RSTH}}$ is fixed to the "0" level.

CONTROL TERMINALS FOR DEBUG

There are some functions and terminals to make easy the development of TLCS-46A application system (program).

(1) MH terminal (Monitor hold control input)

Monitor hold function stops the operation of TCP4600AC by unit of execution instruction.

The operation stops after completion of current excecution instruction by inputting "1" level into MH terminal. However, during "1" period of Y register flag (EYR = 1), the operation does not stop. When the system enters into the interrupt service routine, if stop request is made, the system stops after having jumped to the interrupt service routine.

In the stop state, program counter output the next address. Restart is made by inputting "0" level into MH terminal, and the first cycle makes instruction fetch and at the same time executes NOP instruction internally.

(2) R/H terminal (Run/Hold status output)

Run/Hold monitor function is the response singnal of monitor hold function. During stop of operation by monitor hold function,"0" level is output from R/H terminal (which is reset by the instruction cycle just short of stopping by MH signal), and during normal operation "1" level is output from R/H terminal.

(3) PCH terminal (Program counter hold control input)

Program counter hold function holds the previous state without updating the value of program counter. This function is operated by inputting "1" level into PCH terminal.

(4) INTH terminal (Interupt hold control input)

Interrupt hold function is a function to make interrupt request waiting. Even if interrupt latch is set by inputting "1" level into INTH terminal, jump to interrupt routine is not performed. In this case, interrupt request is kept waiting until "0" level is input into INTH terminal.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

SYMBOL	ITEM	RATING
V_{DD}	Supply Voltage	-0.3V to +7.0V
V_{IN}	Input Voltage	-0.3V to V_{DD}+0.3V
V_{OUT}	Output Volatage	-0.3V to V_{DD}+0.3V
P_D	Power Dissipation	600 mW
T_{sol}	Soldering Temperature & Time	260°C (10 SEC)
T_{stg}	Storage Temperature	-55°C to +125°C
T_{opr}	Operating Temperature	-10°C to +70°C

ALLOWABLE OPERATING CONDITIONS

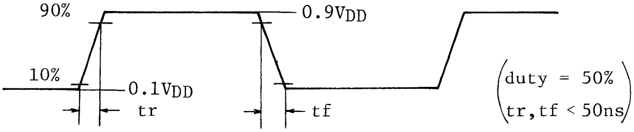
SYMBOL	PARAMETER	RATING
T_a	Ambient Temperature	-10°C to +70°C
V_{DD}	Supply Voltage	4V to 6V
V_{OH}	Output High Voltage	Min. V_{DD}-3.5V($\geq$1.5V)
V_{OL}	Output Low Voltage	Max. 3V
fcp	Basic Clock Frequency	20KHz to 200KHz

DC CHARACTERISTICS (Ta=-10°C to 70°C, V_{DD}=4V to 6V)

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP. (Note1)	MAX.	UNIT
V_{IH}	Input High Voltage		V_{DD}x0.7	V_{DD}x0.55	V_{DD}	V
V_{IL}	Input Low Voltage		0	V_{DD}x0.45	V_{DD}x0.3	
I_{IH}	Input High Current	V_{DD}=6V, V_{IN}=6V	-	-	20	μA
I_{IL}	Input Low Current	V_{DD}=6V, V_{IN}=0V	-	-	-20	
V_{OH}	Output High Voltage	V_{DD}=5V, Output Open	4.7	4.9	-	V
V_{OL}	Output Low Voltage		-	0.1	0.3	
I_{OH}	Output High Current	V_{DD}=4.5V, V_{OH}=2.4V	-0.7	-2	-	mA
I_{OHI}	Output High Current (P05, P06)		-2.5	-6	-	
		V_{DD}=5V, V_{OH}=4.2V	-1.1	-2.5	-	
I_{OL}	Output Low Current	V_{DD}=4.5V, V_{OL}=0.45V	1.6	4	-	
I_{OL1}	Output Low Current (P05, P06)		3.2	8	-	
I_{DDO}	V_{DD} Supply Current in Normal Operation	V_{DD}=6V, fcp=100KHz Output Open V_{IN}=5.9V/0.1V (Note 2)	-	150	450	μA
I_{DDH}	V_{DD} Supply Current in Hold Operation		-	40	120	

Note1: Typical values are at Ta=25°C and V_{DD}=5V.

Note 2: CP input waveform at the time of V_{DD} Supply Current

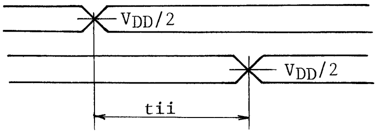


AC CHARACTERISTICS (Ta=-10°C to +70°C, V_{DD}=4V to 6V)

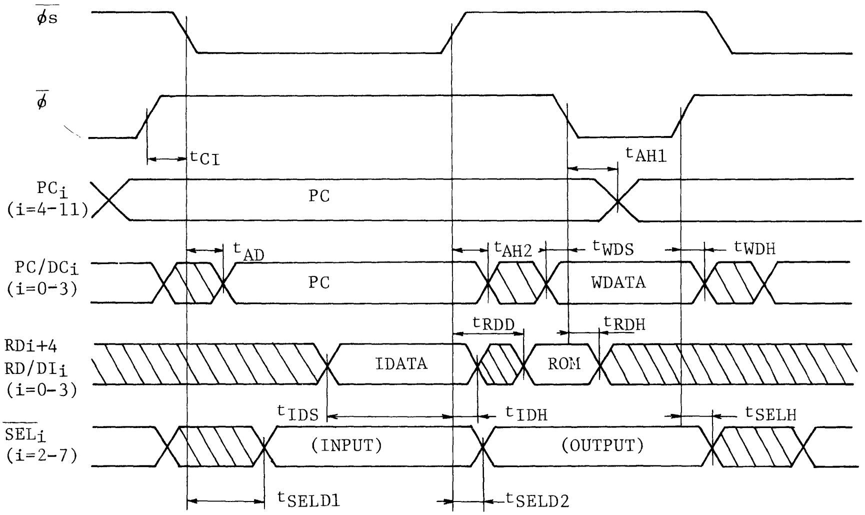
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP. (NOTE 1)	MAX.	UNIT
t_{CI}	Timing Signal Inhibit Time	CL = 50 pF (Note 2)	150	-	1500	ns
t_{AD}	Address Delay Time		-	-	2000	ns
t_{AH1}	Address Hold Time		80	-	-	ns
t_{AH2}	" "		200	-	-	ns
t_{RDD}	Data Delay Time		-	-	1000	ns
t_{RDH}	Data Hold Time		80	-	-	ns
t_{SELD1}	SEL Delay Time (for Input)		-	-	1000	ns
t_{SELD2}	SEL Delay Time (for Output)		100	-	500	ns
t_{SELH}	SEL Hold Time		300	-	-	ns
t_{WDS}	Write Data Set Up Time		500	-	-	ns
t_{WDH}	Write Data Hold Time		300	-	-	ns
t_{IDS}	Read Data Set Up Time		1000	-	-	ns
t_{IDH}	Read Data Hold Tiem		100	-	-	ns
t_{MHD}	MH Input Delay Time		-	-	500	ns
t_{MHH}	MH Input Hold Time		100	-	-	ns
t_{PCHD}	PCH Input Delay Time		-	-	500	ns
t_{PCHH}	PCH Input Hold Time		100	-	-	ns
t_{INTHD}	INTH Input Delay Time		-	-	500	ns
t_{INTHH}	INTH Input Hold Time		100	-	-	ns
t_{WCP}	CP Pulse Width	$V_{IN}=V_{IH}/V_{IL}$	$0.4/f_{cp}$	-	$0.6/f_{cp}$	SEC
t_{WRESET}	$\overline{\text{RESET}}$ Pulse Width		2 tcy	-	-	μs
t_{WINT}	$\overline{\text{INT}}$ Pulse Width		2 tcy	-	-	μs

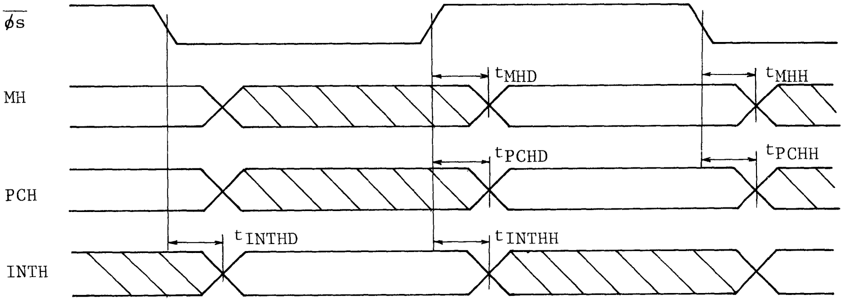
Note 1 : Typical values are at Ta=25°C and V_{DD}=5V.

Note 2 : AC test condition



TIMING WAVEFORMS

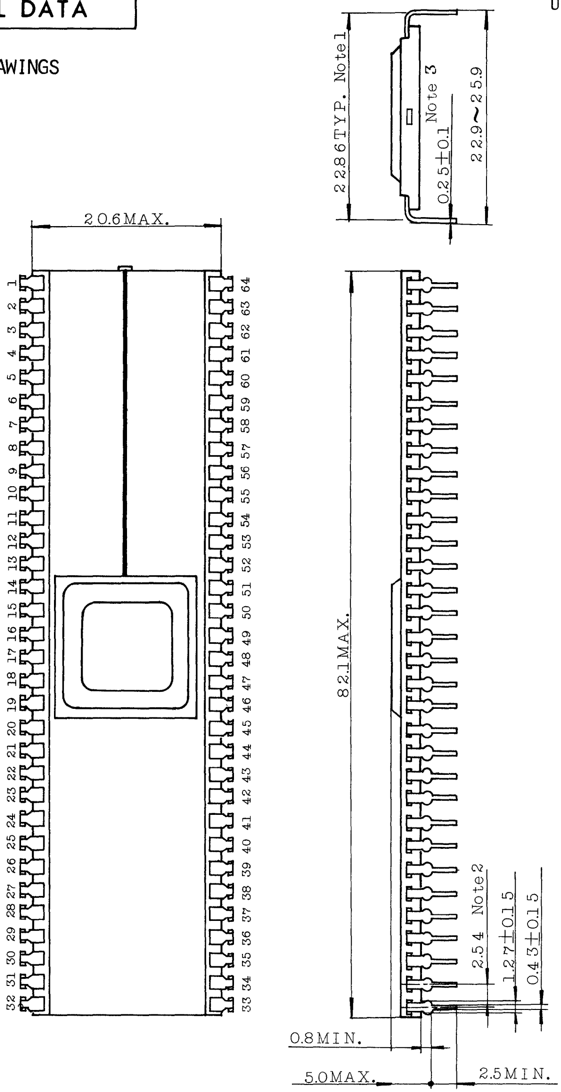




TCP4600AC

OUTLINE DRAWINGS

Unit in mm



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

Note 2: The 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. 64 leads.

Note 3; The metal on the side of the package is GND level.

PRECAUTIONS FOR USE

(1) Clock at time of power supplied

When TLCS-46A starts by $\overline{\text{RESET}}$, clock must be always supplied. There is no problem when clock is obtained by externally mounting the oscillator of crystal, celamic or IFT by use of internal clock generator in TCP4620AP/TCP4630AP, but care should be exercised when clock is externally supplied or when TCP4600AC is used.

(2) Control of I/O port

In port 0, Input/Output can be switched by program; therefore, care should be taken not to become driver short between the port and external circuit.

(3) Current capacity of output port

Each of output port 5 and output port 6 has a large current capacity as port for display. However, if the output port is supplied with a large current exceeding its capacity, it may cause destruction or deterrioration in character.

(4) Handling of unused terminal

When an unused input terminal is left open, it may cause malfunction and increase in power dissipation; therefore, insure to connect the terminals to V_{DD} or GND level. (It is the safest method to connect termianl via resistance.)

Further, insure to connect TEST terminal to GND because this terminal is installed for LSI testing.

(5) Electrostatic destruction

In TLCS-46A series, the input circuit is provided with protective resistance and protective diode for preventing electrostatic destruction. However, the surge equivalent weight is small as compared with discrete resistance and diode; therefore, carry TLCS-46A series in a conductive container so as not to add the surge directly into the system, when required.

(6) Latch-up phenomenon

When the voltage lower than GND or the voltage higher than V_{DD} is applied to input terminal and output terminal, a phenomenon called latch-up characteristic of CMOS presents. This phenomenon may cause destruction and degradation of elements; therefore, care should be taken not to apply the voltage exceeding the maximum rating to the input terminal and output terminal.

4BIT SINGLE CHIP MICROCOMPUTER

TLCS-47(NMOS/CMOS)
ADVANCED INFORMATION

東芝 Toshiba **INTEGRATED CIRCUIT**

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TLCS-47

SILICON MONOLITHIC

PRELIMINARY

4-BIT SINGLE CHIP MICROCOMPUTER TLCS-47

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.

FEATURES

- 4-bit, single chip microcomputer with built-in
  ROM, RAM, input/output port, divider, timer/counter, and serial port.
- Memory capacity
  - ROM : Max 4,096 × 8 bits
  - RAM : Max 256 × 4 bits
- Instruction execution time
  - NMOS family 2μs(at 4MHz clock)
  - CMOS family 4μs(")
- Efficient instruction set
  - 90 instructions
  - Software compatible in the series
- Subroutine nesting: Max. 15 levels
- 6 interrupts (External : 2, Internal : 4)
  Independently latched control and multiple interrupt control
- Input/Output (Standard)

· 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 conversion function (instruction)
  output of data to output port (8-bit)
- Data table look-up and table search function (instruction)
  Table can be set up in the whole ROM area .
- 12-bit timer/counter (2 channels)
- Serial port with 4-bit buffer
- 18-stage divider (with 4-stage prescaler)
- Built-in high output current terminals (NMOS family)
  Typ. 20mA ×8 bits, LED direct drive is available.
- Built-in high breakdown voltage output terminals (CMOS version)
  Max. 42V breakdown voltage
  FL tube direct drive is available.
- Built-in LCD drive circuit (automatic display)(CMOS version)
  - LCD direct drive is available (1/4 duty LCD, Max. 12-digit display)
  - 1/4, 1/3, 1/2 duties or static LCD drive are programmably selectable.
- Stand-by operation (NMOS/CMOS)
  Battery back-up, battery operation and condenser back-up are available.
- On chip oscillator
- TTL/CMOS compatible
- +5V single power supply

Configuration of TLCS-47 Series

Family			TLCS-47N				
Item		Unit / Name	TMP4740P	TMP4720P	TMP4700C	TMP4799C	
ROM Capacity		Bytes	4,096	2,048	(External) 4,096	(External) 4,096	
RAM Capacity		Words	256	128	256	256	
Instruction Execution time		μs	2				
No. of instructions			90				
Subroutine nesting		Levels	Max. 15				
Interrupts	External		2				
	Internal		4 (Serial I/O, timer/counter overflow(2), timer of divider)				
Timer/counter		Channels	2				
	(Bit length)	Bits	12				
	(Mode)		(Event counter, timer and pulse width 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		Stage	18 (With 4-stage prescaler)				
Input/Output Ports	Input	Bits	4				
	Output (corresponding to PLA)		8				
	I/O		16				
	I/O (Combined use)		7				
	Total		35				
With built-in high output current terminals		Bits	8				
With built-in high break-down voltage terminals			—				
With built-in LCD driver			—				
Memory Standby function			YES				
Hold function			—				
Clock oscillator			With built-in				
Power supply		V	+ 5.0				
Process			Nch Si Gate E/D MOS LSI				
Package			DIP - 42		QIC-80	DIC-42	
Remark					Evaluator Chip	Evaluator Chip(Piggy back type)	

Toshiba 東芝 **INTEGRATED CIRCUIT**

TECHNICAL DATA

TLCS-47

PRELIMINARY

Configuration of Series

Series		TLCS - 47					
Family		TLCS-47N		TLCS-47C			(Evaluator chip)
Item	Unit	TMP4740P	TMP4720P	TMP47C40P	TMP47C20P	TMP47C22F	TMP4700C
ROM Capacity	Bytes	4,096	2,048	4,096	2,048	2,048	(External) 4,096
RAM Capacity	Words	256	128	256	128	192	256
Instruction Execution time	μs	2		4			2
No. of instructions		90					
Subroutine nesting	Levels	Max. 15					
Interrupts External		2					
Interrupts Internal		4 (Serial I/O, timer/counter overflow(2), timer of divider)					
Timer/counter	Channels	2					
(Bit length)	Bits	12					
(Mode)		(Event counter, timer and pulse width 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	Stage	18 (With 4-stage prescaler)					
Input/Output Ports: Input	Bits	4				4	4
Output (corresponding to PLA)	Bits	8				0	8
I/O	Bits	16				16	16
I/O (Combined use)	Bits	7				7	7
Total	Bits	35				27	35
With built-in high output current terminals		YES					YES
With built-in high breakdown voltage terminals				YES			
With built-in LCD driver						YES	
Memory Standby function		YES					YES
Hold function				YES		YES	(YES)
Clock oscillator		With built-in					
Power supply	V	+ 5.0					
Process		Nch Si Gate E/D MOS LSI		Si Gate C^2MOS LSI			Nch Si Gate E/D MOS LSI
Package		DIP - 42				FP - 67	QIC - 80

TLCS-47(N)

- TMP4740P
- TMP4720P
- TMP4700C
- TMP4799C

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4740P TMP4720P

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

PRELIMINARY

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 TMP4700C 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
TMP4700C	Externally provided (4,096 × 8)	256 × 4

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

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
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 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

Toshiba 東芝

INTEGRATED CIRCUIT

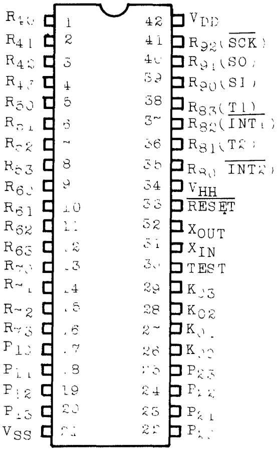
TECHNICAL DATA

TMP4740P

TMP4720P

PRELIMINARY

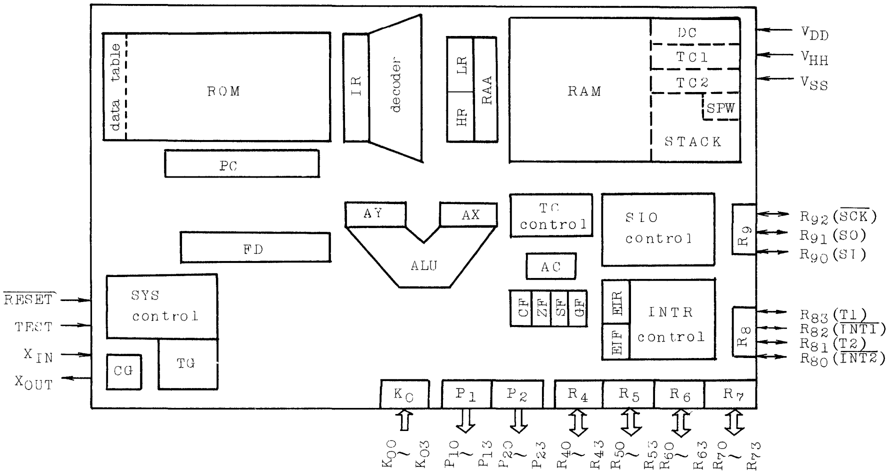
PIN CONNECTIONS (TOP VIEW)



PIN NAMES AND PIN DESCRIPTION

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

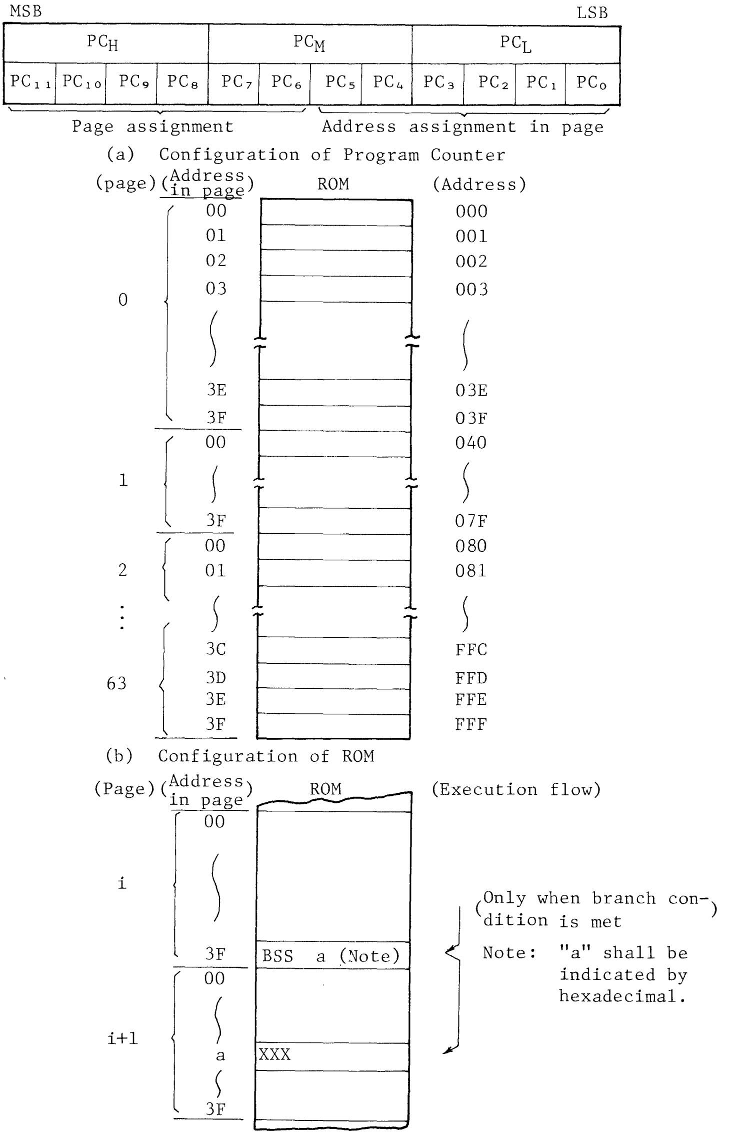
It is a 12-bit binary counter, and the contents of the program counter indicate the address of program memory in which the next instruction to be executed is stored.

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 Countei 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)	Call address by instruction (CALLS a)
086 (Note)	
FEO ~ 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 FEO - FFF) is, therefore, located in addresses 7EO - 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.

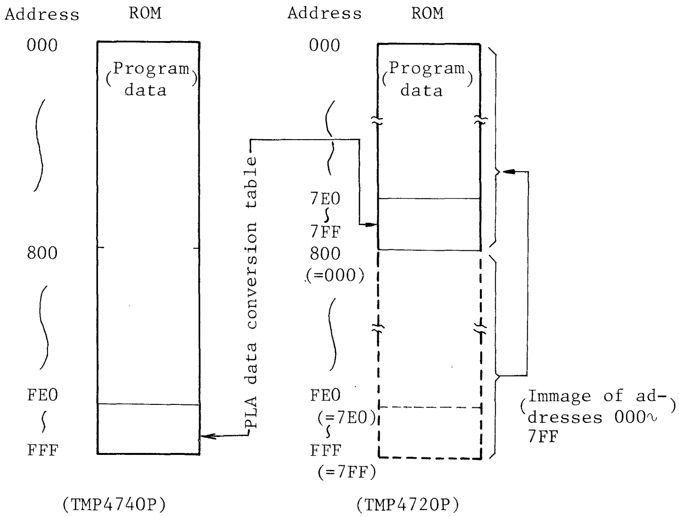


Fig. 1.2.1 ROM Capacity and Address

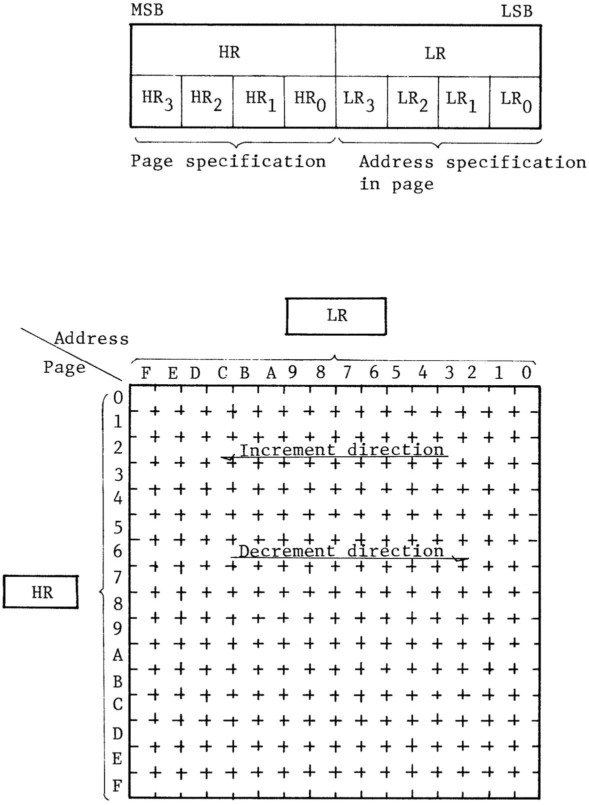
1.3 H Register (HR), L Register (LR), and RAM Address Buffer Register (RAA)

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.


MSB
LSB
HR
LR
HR3 HR2 HR1 HR0 LR3 LR2 LR1 LR0
Page specification
Address specification in page
LR
Address
Page
F E D C B A 9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9 A B C D E F
Increment direction
Decrement direction
HR

Configuration of RAM

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)

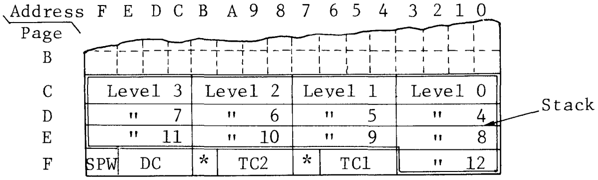
The address FF in the data memory is called a stack pointer word and decides the stack pointer. The stack is contained in the RAM, and accessed by the stack pointer.

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

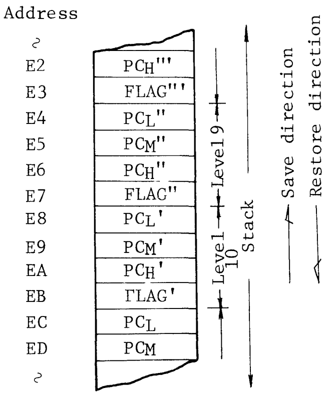
(a) Specific purposive map of RAM

(Stack pointer word RAM address : FF)

	MSB 3	2	1	LSB 0
	SPW_3	SPW_2	SPW_1	SPW_0

	MSB 7	6	5	4	3	2	1	LSB 0
(Stack pointer)	1	1	SPW_3	SPW_2	SPW_1	SPW_0	0	0

(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

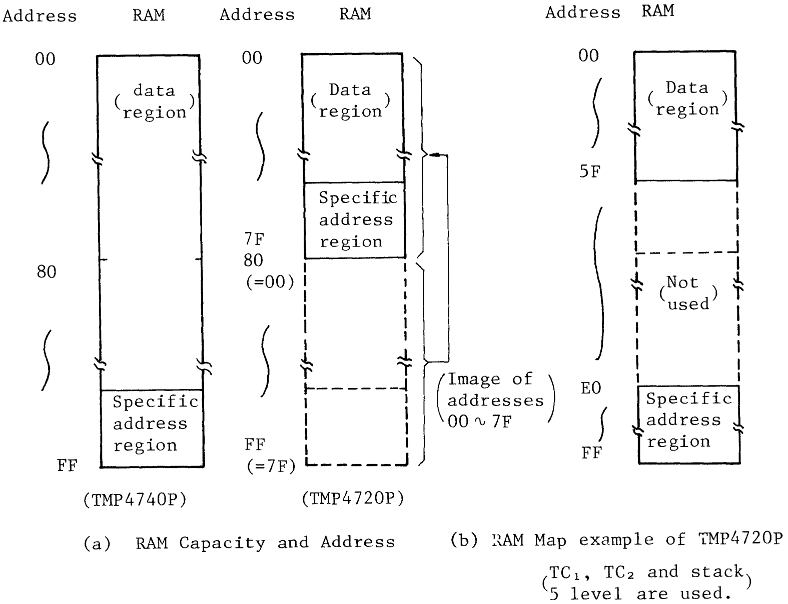
(5) Page 0 in Data Memory

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 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 CO - FF in a program, but the RAM equivalent to addresses 40 - 7F are assigned in the TMP4720P.



(a) RAM Capacity and Address

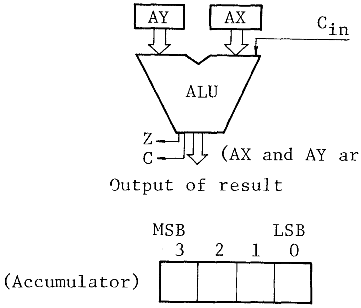
(b) RAM Map example of TMP4720P (TC₁, TC₂ and stack 5 level are used.)

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.



(AX and AY are temporary registers.)

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.

3	2	1	0
CF	ZF	SF	GF

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 2.6 (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 / ——	0					0	
01	IP01/OP01	P_1 Output latch/ P_1 Output 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	——							
0C	IP0C/OP0C	——							
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	/ ——							
18	/OP18	/ ——							
19	/OP19	/ (a)		0					
1A	/OP1A	/ ——							
1B	/OP1B	/ ——							
1C	/OP1C	/ (b)		0					
1D	/OP1D	/ (c)		0					
1E	/OP1E	/ ——							
1F	/OP1F	/ (d)		0					

(*) Serial buffer register (Reception)
(**) Serial buffer register (Transmission)

(a) Control with timer interrupt of divider
(b) Timer/Counter 1 control
(c) Timer/Counter 2 control
(d) Serial port control

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.

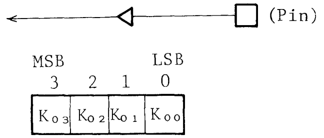


Fig. 1.7.1 K_0 Port

(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 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.

	MSB 7	6	5	4	3	2	1	LSB 0
Conversion data	P_{23}	P_{22}	P_{21}	P_{20}	P_{13}	P_{12}	P_{11}	P_{10}

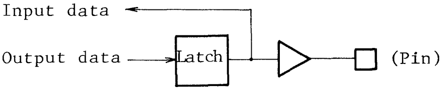


Fig. 1.7.2 P_1 and P_2 Ports

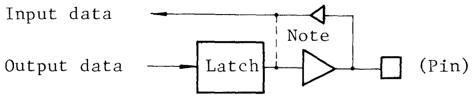
(3) $R_4(R_{43} \sim R_{40})$, $R_5(R_{53} \sim R_{50})$, $R_6(R_{63} \sim R_{60})$, $R_7(R_{73} \sim R_{70})$ Port

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

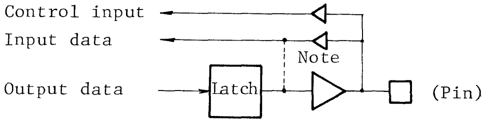
(4) R_8 ($R_{83} \sim R_{80}$) Port

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} ($\overline{\text{INT1}}$) 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} ($\overline{\text{INT2}}$) 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.

MSB 3	2	1	LSB 0
R_{83} (T1)	R_{82} ($\overline{\text{INT1}}$)	R_{81} (T2)	R_{80} ($\overline{\text{INT2}}$)



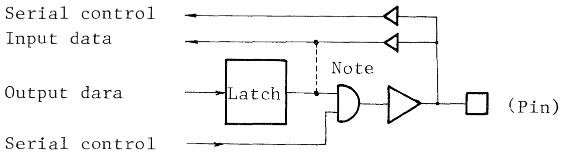
Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.4 R_8 Port

(5) $R_9(R_{92} \sim R_{90})$ 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 R_9 port is also used as serial port. The latch must be set to "1" when R_9 port is used as serial port. The port used as ncrmal 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.

MSB 3	2	1	LSB 0
	R_{92} ($\overline{\text{SCK}}$)	R_{91} (SO)	R_{90} (SI)



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

EIF

Interrupt enabling register (EIR)

MSB			LSB
3	2	1	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

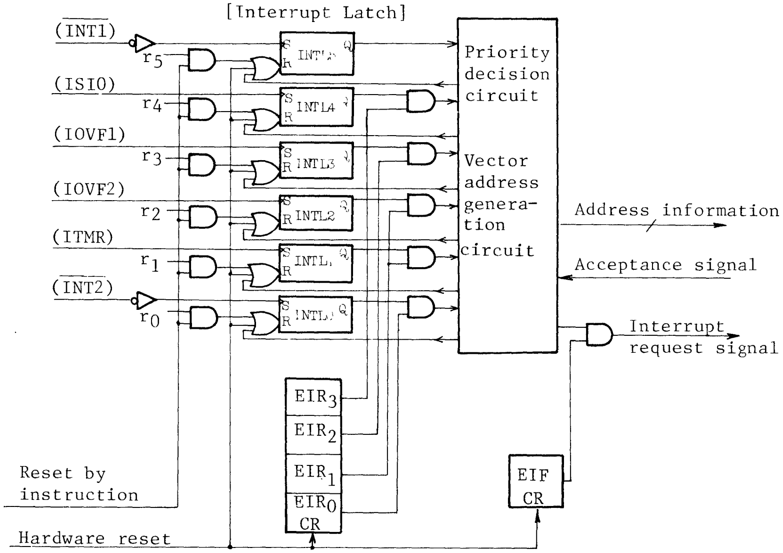
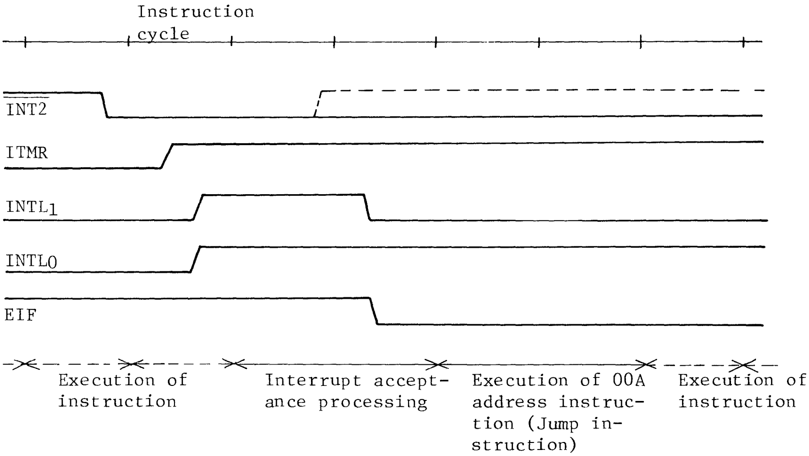


Fig. 1.8.1 Interrupt Control Circuit



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.

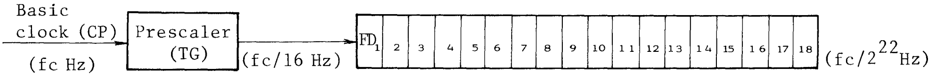
The basic clock (fc Hz) is divided into sixteen by the timing generator and input to the divider; therefore, the output frequency at the last stage is $fc/2^{22}$ Hz.
It is reset to "0" at the initialization operation.

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)

(*: don't care)

MSB 3	2	1	LSB 0			
*	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.

	MSB		LSB
(Timer/Counter 1)	TC1		
	TC1H	TC1M	TC1L
(RAM Address)	(F6)	(F5)	(F4)

	MSB		LSB
(Timer/Counter 2)	TC2		
	TC2H	TC2M	TC2L
(RAM Address)	(FA)	(F9)	(F8)

Fig. 1.10.1 Timer/Counter

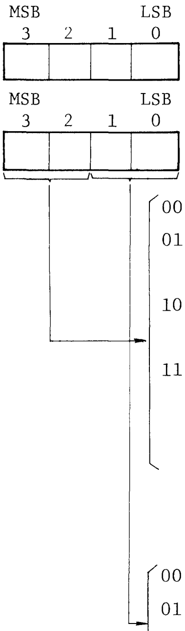
(1) Timer/Counter Control

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)].

TC1 Command register
(Port Address OP1C)

TC2 Command register
(Port Address OP1D)



Selection of Count Mode

00 : Disable state
01 : Event count mode
(Counts external input)
10 : Timer mode
(Counts internal pulse.)
11 : Pulse width measurement mode
(Counts the pulse obtained by sampling of external input by use of internal pulse.)

Selection of Internal Pulse (Divider Output) Rate

00 : $fc/2^{10}$ Hz is counted.
01 : $fc/2^{14}$ Hz is counted.
10 : $fc/2^{18}$ Hz is counted.
11 : $fc/2^{22}$ Hz is counted.
(fc: Basic clock frequency)

(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

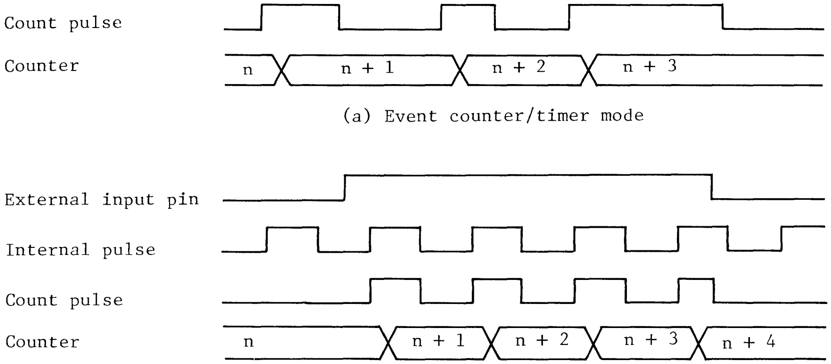
(2) Count Operation

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≒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.

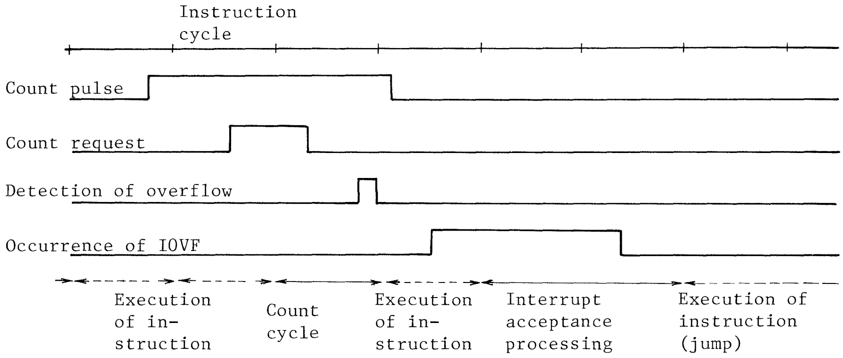


Fig. 1.10.4 Timing Chart of Timer/Counter in Interrupt by Overflow

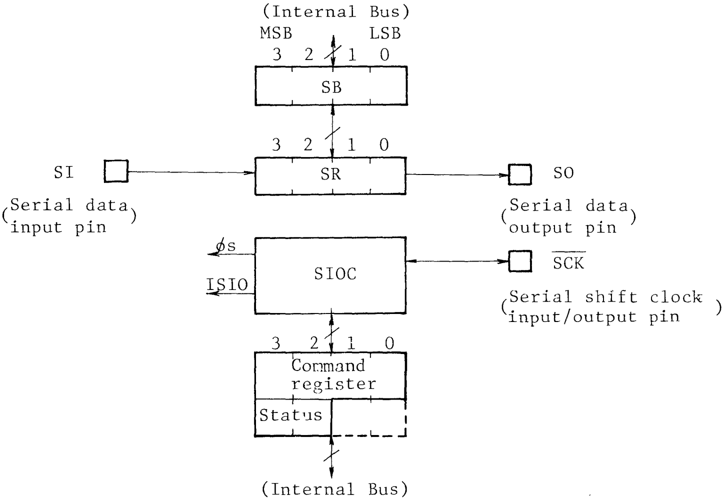
1.11 Serial Port (SIO)

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 ($\overline{SCK}$), R9$_1$ (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 R9$_1$ 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.

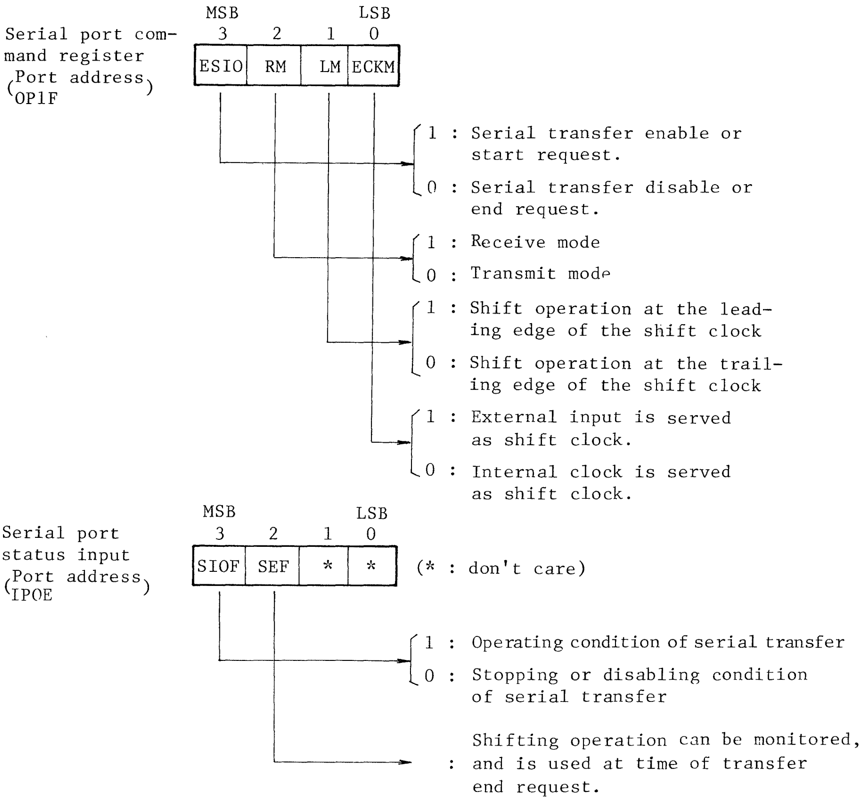


SR :	4-bit shift register	SIOC :	Serial port control circuit
SB :	4-bit buffer register	ϕs :	Internal shift clock
		ISIO :	Interrupt request

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.



Serial port command register (Port address OP1F)

MSB 3	2	1	LSB 0
ESIO	RM	LM	ECKM

ESIO:
1 : Serial transfer enable or start request.
0 : Serial transfer disable or end request.

RM:
1 : Receive mode
0 : Transmit mode

LM:
1 : Shift operation at the leading edge of the shift clock
0 : Shift operation at the trailing edge of the shift clock

ECKM:
1 : External input is served as shift clock.
0 : Internal clock is served as shift clock.

Serial port status input (Port address IPOE)

MSB 3	2	1	LSB 0
SIOF	SEF	*	*

(* : don't care)

SIOF:
1 : Operating condition of serial transfer
0 : Stopping or disabling condition of serial transfer

SEF:
: Shifting operation can be monitored, and is used at time of transfer end request.

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 $\overline{\text{SCK}}$ 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 $\overline{\text{SCK}}$ pin serves as an input.

Leading edge shift mode

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

Trailing edge shift mode

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

The $\overline{\text{SCK}}$ pin must be set to the "high" level when the serial transfer is started. In the internal clock mode, the $\overline{\text{SCK}}$ 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".

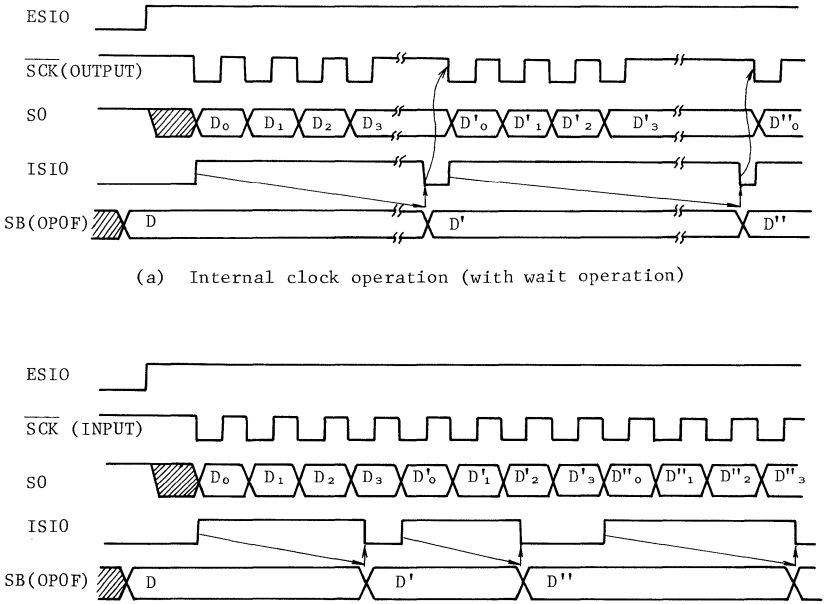
Internal clock operation

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

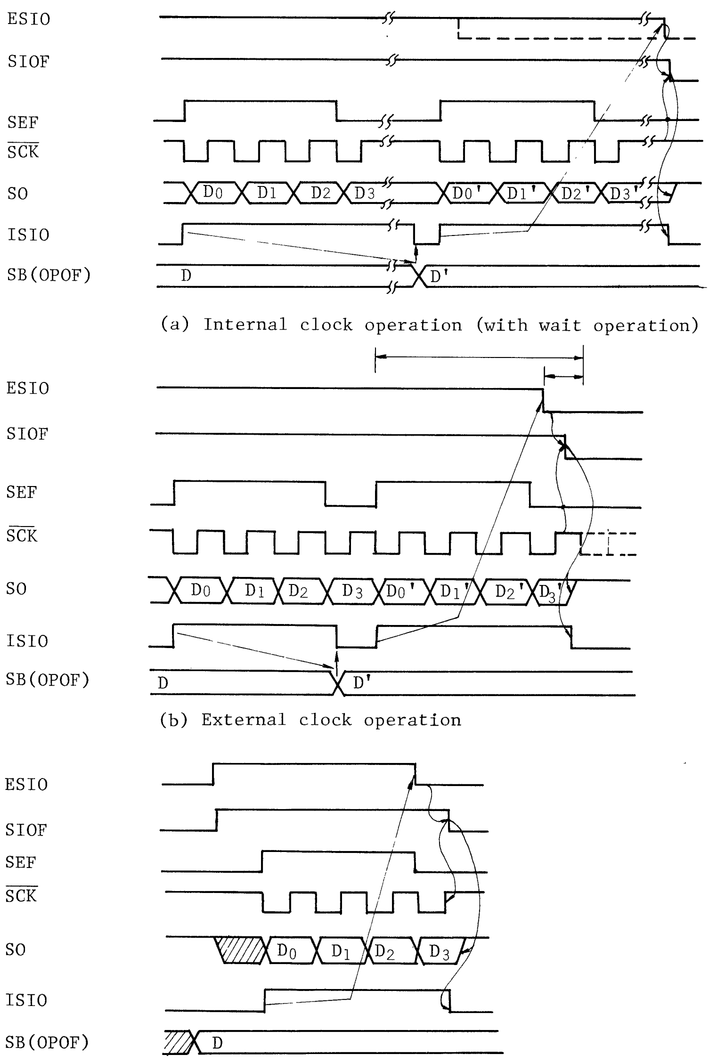
Fig. 1.11.3 Transmit Mode

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.

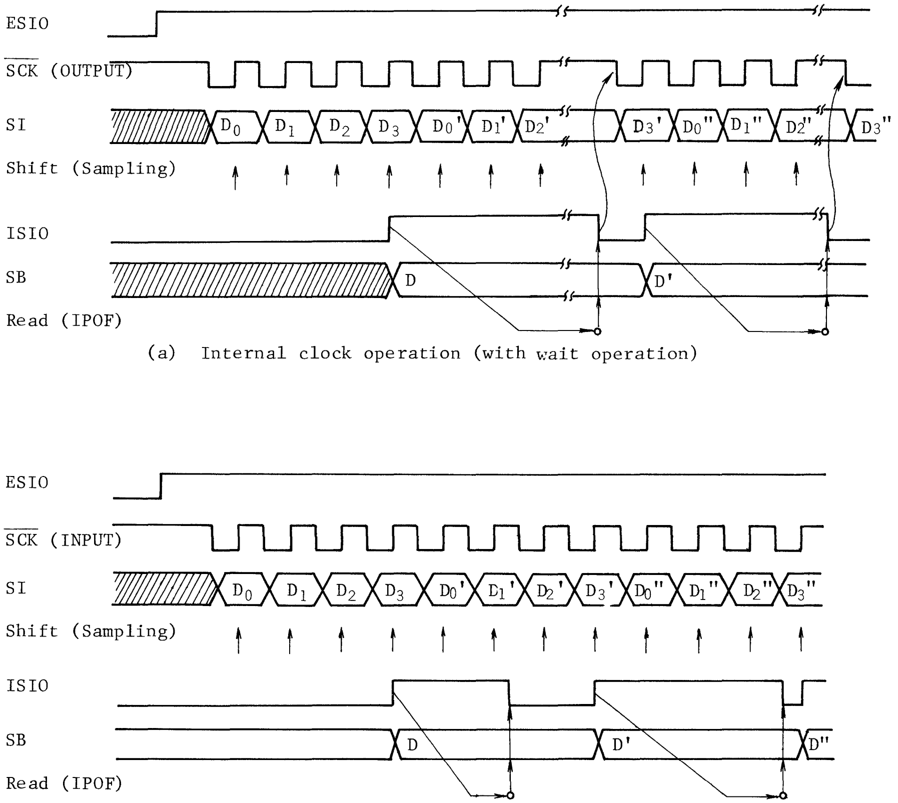
Internal clock operation

During the operation of the internal clock of fc/2^7Hz, 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.

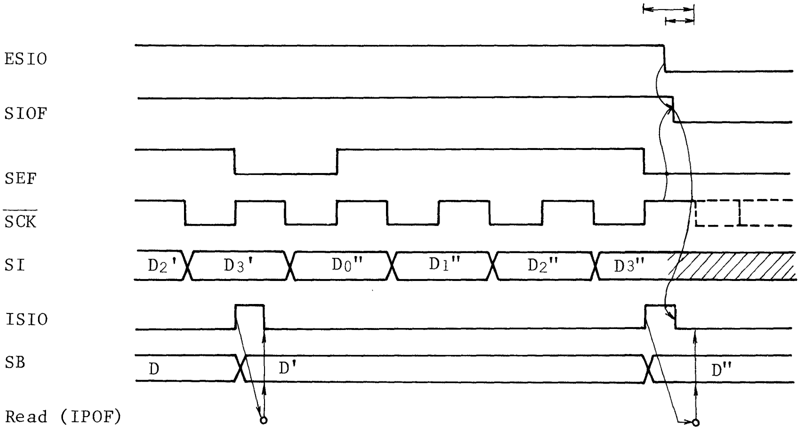


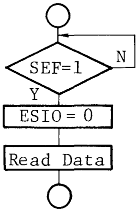
Fig. 1.11.6 Completion of Receiving (at slow transfer rate)

(b) Fast transfer rate

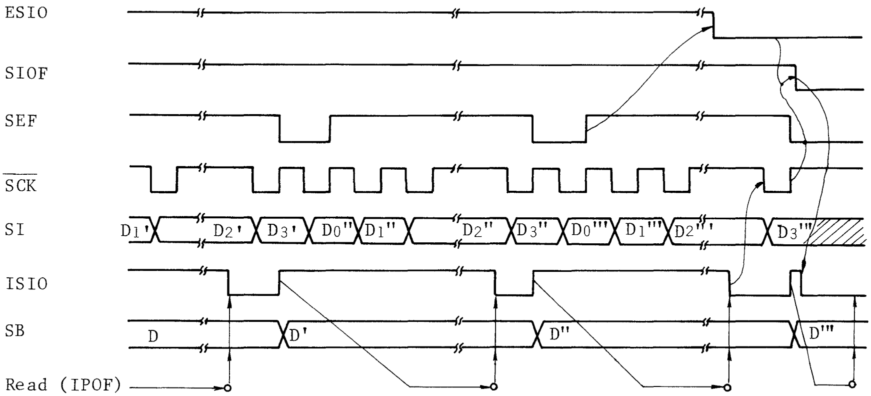
If the shift operation for the next data may start before the current data is read out by receipt of the interrupt request becuase the transfer rate is too fast, the interrupt service program which is loaded to read out the last data but one should be used to reset the ESIO to "0" after confirming that 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

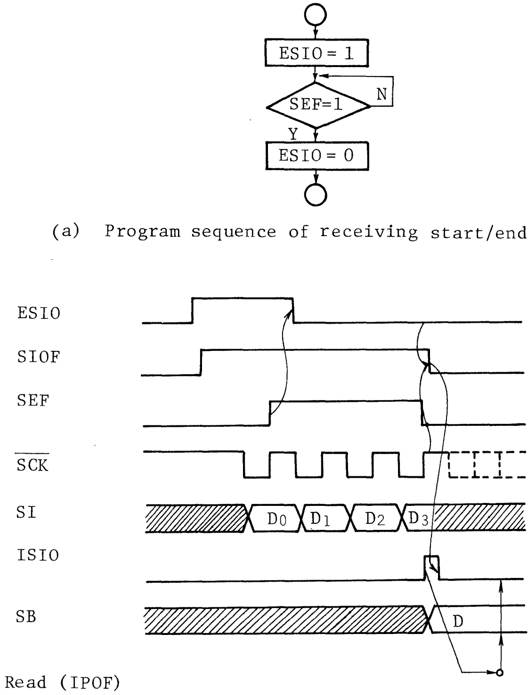


(b) Timing Chart
(in case of internal clock operation with wait operation)

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)

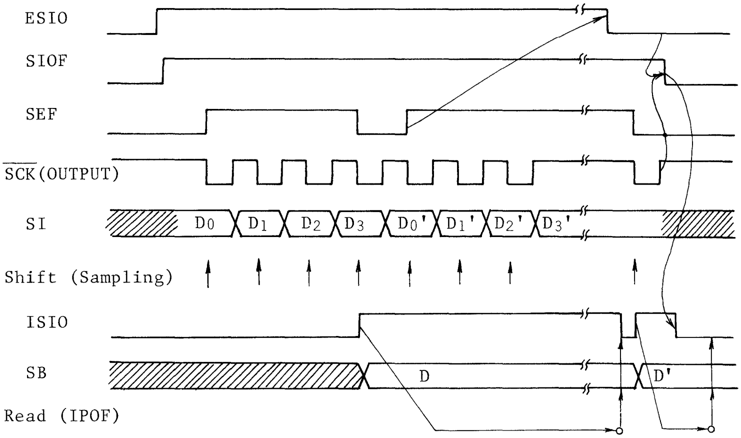
Receive (leading edge shift) mode

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

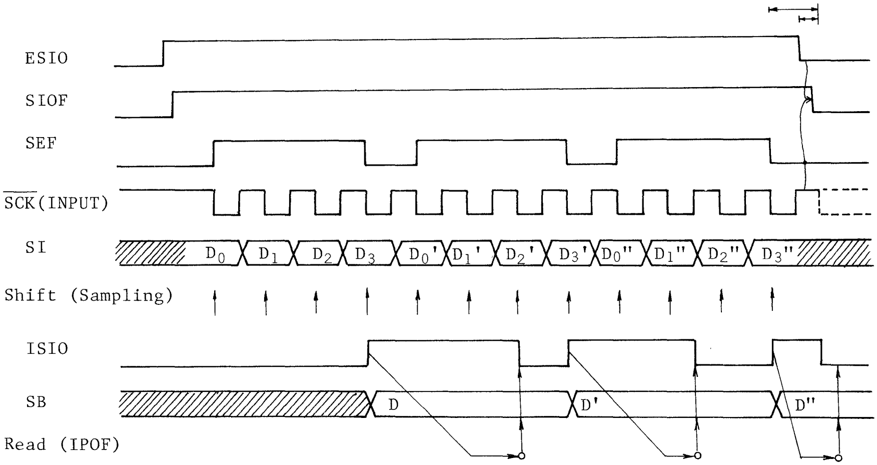
When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, the interrupt (ISIO) occurs to request the data reading (buffer full). Since the shift register is transferring the data to the buffer register, the shift operation 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.

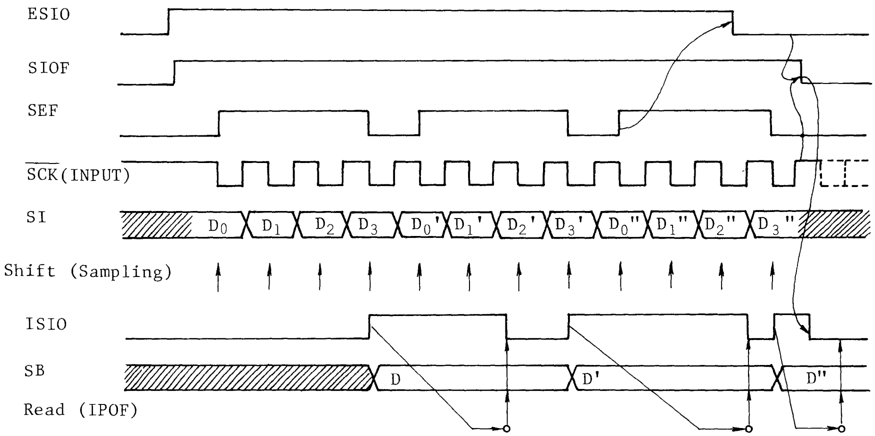
The basic operation in the receive (leading edge shift) mode is equivalent to that in the receive (trailing edge shift) mode except that the edge for the shift clock is different, and that at time of the transfer start, the first shifted data has been already input from the external circuitry before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



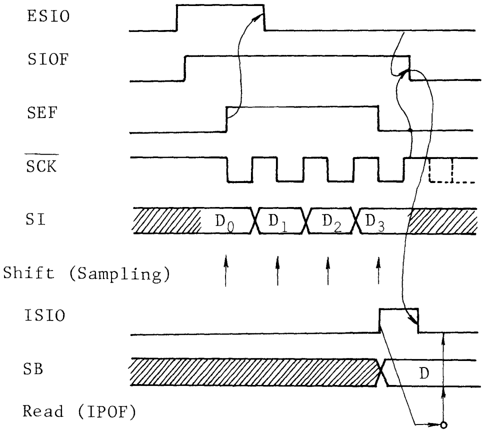
(a) Internal clock operation (with wait operation)



(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. 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.

2.1 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.

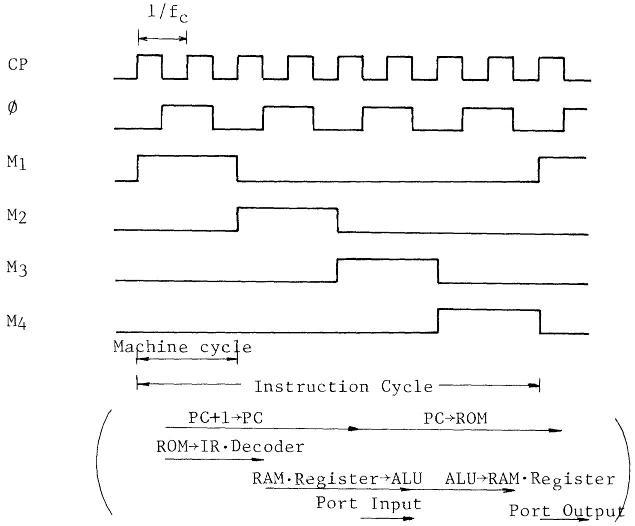


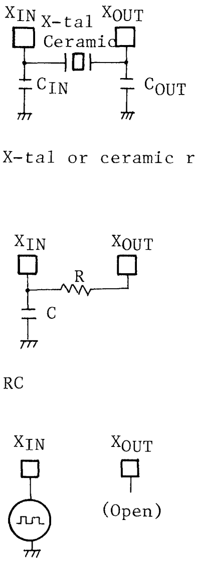
Fig. 2.1.1 Instruction Cycle

2.2 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. 2.2.1 Resonator Connections

2.3 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 (≈200kΩ TYP., MOS-load resister).

2.4 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.

2.5 Interrupt input

Two pins ($\overline{\text{INT1}}$, $\overline{\text{INT2}}$) 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.

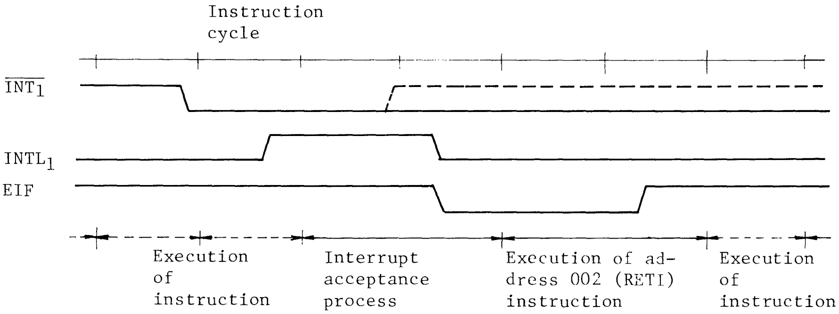


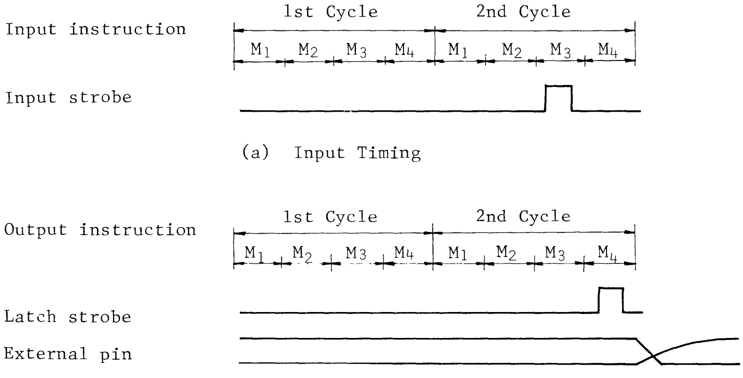
Fig. 2.5.1 Interrupt Timing (Dummy process of INT_1 interrupt)

2.6 Input/output port

(1) Input/output timing

The timing to read the external data from the input port or I/O port is in M3 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 M4 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. 2.6.1 Input/Output Timing

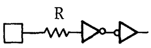
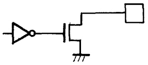
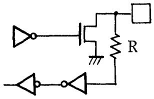
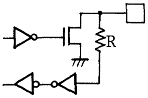
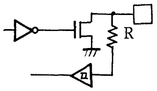
(2) Input/Output circuit format

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.

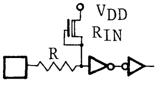
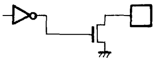
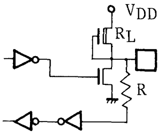
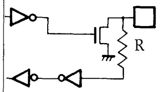
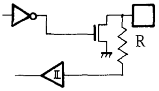
"I$\overline{\text{O}}$C$\overline{\text{O}}$DE AA" is employed if not specified.

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

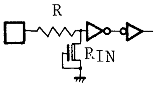
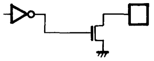
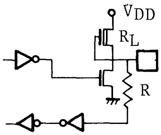
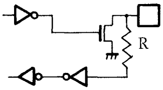
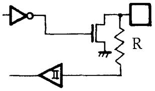
Port Circuit	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
I/O equivalent Circuit	 R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	○ High threshold input. ○ No resistor is contained.	○ Sink open drain output. ○ High output current. ○ Output latch is initialized 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 TMP4700C is made up with the circuit system equivalent to this input/output circuit system; therefore, the system of the TMP4700C can become equivalent to that of the TMP4740P or the TMP4720P by externally installing EPROM (program memory) on the TMP4700C (but TMP4700C is not contained the pull-up resistor with $\overline{\text{RESET}}$ pin and the pull-down resistor with TEST pin.).

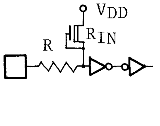
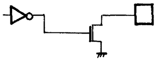
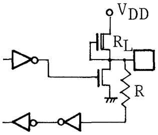
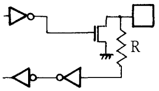
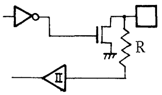
Input/Output Circuit Code (IOCODE) AE

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

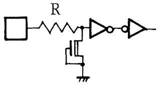
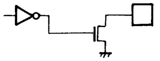
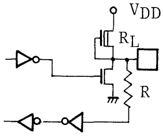
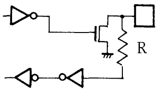
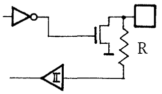
Input/Output Circuit Code (IOCODE) AF

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 (IOCODE) 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 (IOCODE) 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

2.7 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 R9 port. These pins can be used as the pins of the R9 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 (≈ 70kΩ TYP., MOS-load resistor).

3. 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 3.0.1 Classification of Instructions.

3.1 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

3.2 Description of instructions (*): Note 1 (**): Exec.cycle (⁂): Hexadecimal

Item / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Move Instruction	LD A, @HL	0 0 0 0 1 1 0 0	0 C	(AC)←M[(H·L)] Loads the contents of the data memory specified by the H and L registers in the accumulator.	-	Z	1	1
	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] Loads the contents of the data memory specified by the x of the instruction field in the accumulator.	-	Z	1	2
	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] $x'=x_7x_6x_5x_4x_3x_2 00$ 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.	-	-	1	2
	LD A, #k	0 1 0 0 $k_3k_2k_1k_0$	4 k	(AC)←k Loads the immediate data k of the instruction field in the accumulator. Serves as the clear instruction when k = 0.	-	Z	1	1

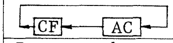
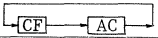
Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Move Instruction	LD H, #k	1 1 0 0 $k_3k_2k_1k_0$	C k	(HR)←k Loads the immediate data k of the instruction field in the H register. Serves as the clear instruction when k = 0.	-	-	1	1
	LD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

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
Move Instruction	ST #k, y	0010 1101 $k_3k_2k_1k_0$ $y_3y_2y_1y_0$	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
	MOV H, A	0001 0000	1 0	(AC)←(HR) Loads the contents of the H register in the accumulator.	-	Z	1	1
	MOV L, A	0001 0001	1 1	(AC)←(LR) Loads the contents of the L register in the accumulator.	-	Z	1	1
	XCH A, H	0011 0000	3 0	(HR)⇆(AC) Exchanges the contents of the accumulator for those of the H register. [Note 2]	-	Z	1	2
	XCH A, L	0011 0001	3 1	(LR)⇆(AC) Exchanges the contents of the accumulator for those of the L register. [Note 2]	-	Z	1	2
	XCH A,EIR	0001 0011	1 3	(EIR)⇆(AC) Exchanges the contents of the accumulator for those of the interrupt enable register.	-	-	1	1
	XCH A,@HL	0000 1101	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	0011 1101 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	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	0010 1001 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 9 x_Hx_L	M[x']⇆(LR),M[x'+1]⇆(HR) x'=$x_7x_6x_5x_4x_3x_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		Function	Flag(*)			(**)
		Binary	(**)	Functional Description	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.	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		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
Arithmetic Instruction	INC @HL	0 0 0 0 1 0 1 0	0 A	M[(H·L)]←M[(H·L)]+1	-	Z	$\overline{C}$	1
				Increments the contents of the data memory specified by the H and L registers.				
	DEC A	0 0 0 0 1 0 0 1	0 9	(AC)←(AC)-1	-	Z	$\overline{B}$	1
				Decrements the contents of the accumulator.				
	DEC L	0 0 0 1 1 0 0 1	1 9	(LR)←(LR)-1	-	Z	$\overline{B}$	1
				Decrements the contents of the L register.				
	DEC @HL	0 0 0 0 1 0 1 1	0 B	M[(H·L)]←M[(H·L)]-1	-	Z	$\overline{B}$	1
				Decrements the contents of the data memory specified by the H and L registers.				
	ADDC A,@HL	0 0 0 1 0 1 0 1	1 5	(AC)←(AC)+M[(H·L)]+(CF)	C	Z	$\overline{C}$	1
				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.				
	ADD A,@HL	0 0 0 1 0 1 1 1	1 7	(AC)←(AC)+M[(H·L)]	-	Z	$\overline{C}$	1
				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.				
	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	-	Z	$\overline{C}$	2
				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.				
	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	-	Z	$\overline{C}$	2
				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.				

Items / Class	Assembler Mnemonic	Object Code		Function				(**)
		Binary	(***)	Functional Description	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	-	Z	$\overline{C}$	2
				Adds the immediate data k of the instruction field to the contents of the L register, and places the result in the L register.				
	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	-	Z	$\overline{C}$	2
				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.				
	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	-	Z	$\overline{C}$	2
				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.				
	SUBRC A,@HL	0 0 0 1 0 1 0 0	1 4	(AC)←M[(H·L)]-(AC)-($\overline{CF}$)	$\overline{B}$	Z	$\overline{B}$	1
				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.				
	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)	-	Z	$\overline{B}$	2
				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.				

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
Arithmetic Instruction	SUBR @HL,#k	0011 1000 0101 $k_3k_2k_1k_0$	38 5k	M[(H·L)]←k-M[(H·L)]	-	Z	$\overline{B}$	2
				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.				
Logical Instruction	ROLC A	0000 0101	05	 (rotate left by 1 bit)	C	Z	$\overline{C}$	1
				Rotates the contents of the accumulator and carry flag to the left by one bit. [Note 4]				
	RORC A	0000 0111	07	 (rotate right by 1 bit)	C	Z	$\overline{C}$	1
				Rotates the contents of the accumulator and carry flag to the right by one bit. [Note 4]				
	AND A,@HL	0001 1110	1E	(AC)←(AC)∧M[(H·L)]	-	Z	$\overline{Z}$	1
				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.				
	AND A,#k	0011 1000 0011 $k_3k_2k_1k_0$	38 3k	(AC)←(AC)∧k	-	Z	$\overline{Z}$	2
				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.				
	AND @HL,#k	0011 1000 0111 $k_3k_2k_1k_0$	38 7k	M[(H·L)]←M[(H·L)]∧k	-	Z	$\overline{Z}$	2
				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.				

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

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Bit Manipulation Instruction	TEST y, b	0 0 1 1 1 0 0 1 1 0 $b_1 b_0$ $y_3 y_2 y_1 y_0$	3 9 8+b y	$(SF) \leftarrow \overline{M[y]<b>}$ 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 $b_1 b_0$ $p_3 p_2 p_1 p_0$	3 B 8+b P	$(SF) \leftarrow \overline{P[p]<b>}$ 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	$(SF) \leftarrow \overline{P[(LR)<3:2>+4]<(LR)<1:0>>}$ Places the inverse contents of the bit, which is specified by the lower-order two bits of the L register, of the ports R4 - R7 (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 $b_1 b_0$ $y_3 y_2 y_1 y_0$	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(*)			(**)
		Binary	(**)	Functional Description	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>	-	-	*	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		Function	Flag(*)			(**)
		Binary	(* **)	Functional Description	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>Λr<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>Λr<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>Λr<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		Function	Flag(*)			(**)
		Binary	(*/**)	Functional Description	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	$M[(H\cdot L)]\leftarrow P[p]$ 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 $8+2\overline{p}_4p$	$P[p]\leftarrow(AC)$, $P=P_4P_3P_2P_1P_0$ 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 $C+2\overline{p}_4p$	$P[p]\leftarrow M[(H\cdot L)]$, $P=P_4P_3P_2P_1P_0$ 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	$P[p]\leftarrow k$ 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	$P[2]\cdot P[1]\leftarrow ROM[F\cdot(E+(CF))\cdot M[(H\cdot L)]]$ Outputs the data (eight bits) of the program memory located in addresses FEO - 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(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
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(*)			(**)
		Binaty	(⁂)	Functional Description	CF	ZF	SF	
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	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.

ELECTRICAL CHARACTERISTICS

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

SYMBOL	ITEM	RATING	UNITS
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 (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} 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 (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.

Toshiba 東芝

INTEGRATED CIRCUIT

TECHNICAL DATA

TMP4740P
TMP4720P

PRELIMINARY

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

SYMBOL	PARAMETER	CONDITION	MIN.	(Note 1) TYP.	MAX.	UNITS
V_{HS}	Hysteresis Voltage (Schmitt Circuit Input)		-	0.5	-	V
I_{IN1}	Input Current (Note 2) (K_0, $\overline{RESET}$, TEST)	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 Pull-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	100	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.

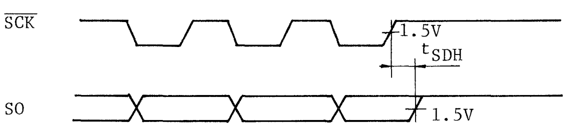
(Note 2) When an input resistor is built in the device, the input current through the resistor is eliminated.

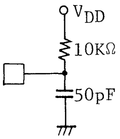
AC CHARACTERISTICS (V_{SS}=0V, V_{DD}=V_{HH}=5V±10%, T_{opr}=-30 ∿ 70°C)

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)

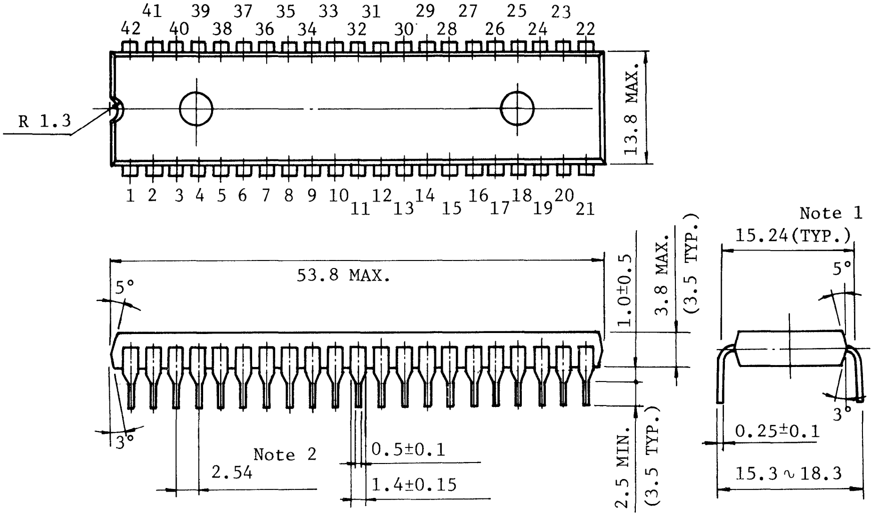




(Note 1) External circuit for serial ports $\overline{SCK}$ and SO

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.

Specification of program tape and input/output circuit format

Engineering Samples(ES) of the TMP4740P and TMP4720P will be made if you specify the program data and input/output circuit format by use of a paper tape.

The paper tape format is equivalent to the Hex. format of Intel Co. (Format I).

The program data should be specified within the address space corresponding to the built-in ROM capacity; the addresses 000 - 7FF denote the address range in the TMP4720P. Accordingly, if the PLA data conversion table (addresses FE0 - FFF) is used, the table data must be assigned as the data located in addresses 7E0 - 7FF.

1. Specification of input/output circuit format

The paper tape of Format I starts recording the program data after record mark ":", but the input/output circuit code should be specified just before the first record mark.

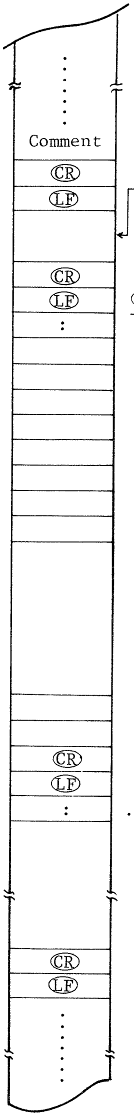
The "IŌCŌDE XX" format is used to define the input/output circuit code. XX denotes the proper input/output circuit code (two alphabets).

(Note) If the input/output circuit code is not specified, "IŌCŌDE AA" is employed. It should be noted that if the specified format is different from the standard one, and if the specified input/output circuit code is illegal, such specifications may be considered to have not been made.

(Example of tape list)

```
TŌSHIBA MICRŌCŌMPUTER TLCS-47
IŌCŌDE AA
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67E31F5D8ABA6DF292F113F5C1
:100020004FF1F
                        :
                        :
                        :
:1007E000B53D42E0EC32546025B7308CDD52063D1D
:1007F000B4BE9E9E345B6138060B20BC372BF60BD6
:00000001FF
```

2. Program tape format (Format I)



.... Leaders More than 50 characters of "NULL"

.... Comment (record mark ":" and the input/output circuit format are not included.)

Specification of the input/output circuit format.

Not specially required.

... Specification of the input/output circuit format IOCODE XX (XX denotes the input/output circuit code) (It does not matter if there is "NULL" from immediately after CR, LF to immediately before record marking.)

It is considered to be no specification, if not specified.

.... Record mark (start of record)

.... Record length (the number of program data within one record is represented by two-digit hexadecimal notation. "00" denotes the end of data.)

.... Load address (the first address of the record is represented by four-digit hexadecimal notation.)

... Record type ("00" ..... Normal record
"01" ..... End of file record)

.... Data (1-byte data is represented by two-digit hexadecimal notation.)

... Checksum (two-digit hexadecimal notation is used to represent the data of the lower-order eight bits in the value which the data covering from the record length up to immediately before the checksum is subtracted from the initial value 0.)

...... Record mark (the same procedure is repeated hereinafter.)

... Trailers More than 50 characters of "NULL"

LIST OF INSTRUCTIONS

Item / Classification	Assembler Mnemonic	Object Code				Function	*1 Flags			Execution Cycle	
		Binary		Hexadecimal							
		1st Byte	2nd Byte	1st Byte	2nd Byte		CF	ZF	SF		
Move	LD A ,@HL	00 00 11 00		0 C		(AC)←M[(H•L)]	–	Z	1	1	
	LD A , x	00 11 11 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 C	x_Hx_L	(AC)←M[x]	–	Z	1	2	
	LD HL, x	00 10 10 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 8	x_Hx_L	(LR)←M[x'],(HR)←M[x'+1], $x'=x_7x_6x_5x_4x_3x_2$00	–	–	1	2	
	LD A ,#k	01 00 $k_3k_2k_1k_0$		4 k		(AC)←k	–	Z	1	1	
	LD H ,#k	11 00 $k_3k_2k_1k_0$		C k		(HR)←k	–	–	1	1	
	LD L ,#k	11 10 $k_3k_2k_1k_0$		E k		(LR)←k	–	–	1	1	
	LDL A ,@DC	00 11 00 11		3 3		(AC)←ROM_L[(DC)]	–	Z	1	2	
	LDH A ,@DC+	00 11 00 10		3 2		(AC)←ROM_H[(DC)],(DC)←(DC)+1	–	Z	1	2	*2
	ST A ,@HL	00 00 11 11		0 F		M[(H•L)]←(AC)	–	–	1	1	
	ST A ,@HL+	00 01 10 10		1 A		M[(H•L)]←(AC),(LR)←(LR)+1	–	Z	$\overline{C}$	1	*3
	ST A ,@HL–	00 01 10 11		1 B		M[(H•L)]←(AC),(LR)←(LR)–1	–	Z	$\overline{B}$	1	*3
	ST A , x	00 11 11 11	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 F	x_Hx_L	M[x]←(AC)	–	–	1	2	
	ST #k ,@HL+	11 11 k3k2k1k0		F k		M[(H•L)]←k,(LR)←(LR)+1	–	Z	$\overline{C}$	1	*3
	ST #k, y	00 10 11 01	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 D	k y	M[y]←k	–	–	1	2	
	MOV H , A	00 01 00 00		1 0		(AC)←(HR)	–	Z	1	1	
	MOV L , A	00 01 00 01		1 1		(AC)←(LR)	–	Z	1	1	
	XCH A , H	00 11 00 00		3 0		(HR)⇄(AC)	–	Z	1	2	*2
	XCH A , L	00 11 00 01		3 1		(LR)⇄(AC)	–	Z	1	2	*2
	XCH A ,EIR	00 01 00 11		1 3		(EIR)⇄(AC)	–	–	1	1	
	XCH A ,@HL	00 00 11 01		0 D		M[(H•L)]⇄(AC)	–	Z	1	1	*2
	XCH A , x	00 11 11 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 D	x_Hx_L	M[x]⇄(AC)	–	Z	1	2	*2
	XCH HL, x	00 10 10 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 9	x_Hx_L	M[x']⇄(LR),M[x'+1]⇄(HR), $x'=x_7x_6x_5x_4x_3x_2$00	–	–	1	2	
Compare	CMPR A ,@HL	00 01 01 10		1 6		null←M[(H•L)]–(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR A , x	00 11 11 10	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 E	x_Hx_L	null←M[x]–(AC)	$\overline{B}$	Z	$\overline{Z}$	2	
	CMPR A ,#k	11 01 $k_3k_2k_1k_0$		D k		null←k–(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR H ,#k	00 11 10 00	11 01 $k_3k_2k_1k_0$	3 8	D k	null←k–(HR)	–	Z	$\overline{B}$	2	
	CMPR L ,#k	00 11 10 00	10 01 $k_3k_2k_1k_0$	3 8	9 k	null←k–(LR)	–	Z	$\overline{B}$	2	
	CMPR y ,#k	00 10 11 10	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 E	k y	null←k–M[y]	$\overline{B}$	Z	$\overline{Z}$	2	
Arithmetic	INC A	00 00 10 00		0 8		(AC)←(AC)+1	–	Z	$\overline{C}$	1	
	INC L	00 01 10 00		1 8		(LR)←(LR)+1	–	Z	$\overline{C}$	1	
	INC @HL	00 00 10 10		0 A		M[(H•L)]←M[(H•L)]+1	–	Z	$\overline{C}$	1	
	DEC A	00 00 10 01		0 9		(AC)←(AC)–1	–	Z	$\overline{B}$	1	
	DEC L	00 01 10 01		1 9		(LR)←(LR)–1	–	Z	$\overline{B}$	1	
	DEC @HL	00 00 10 11		0 B		M[(H•L)]←M[(H•L)]–1	–	Z	$\overline{B}$	1	
	ADDC A ,@HL	00 01 01 01		1 5		(AC)←(AC)+M[(H•L)]+(CF)	C	Z	$\overline{C}$	1	
	ADD A ,@HL	00 01 01 11		1 7		(AC)←(AC)+M[(H•L)]	–	Z	$\overline{C}$	1	
	ADD A ,#k	00 11 10 00	00 00 $k_3k_2k_1k_0$	3 8	0 k	(AC)←(AC)+k	–	Z	$\overline{C}$	2	
	ADD H ,#k	00 11 10 00	11 00 $k_3k_2k_1k_0$	3 8	C k	(HR)←(HR)+k	–	Z	$\overline{C}$	2	
	ADD L ,#k	00 11 10 00	10 00 $k_3k_2k_1k_0$	3 8	8 k	(LR)←(LR)+k	–	Z	$\overline{C}$	2	
	ADD @HL,#k	00 11 10 00	01 00 $k_3k_2k_1k_0$	3 8	4 k	M[(H•L)]←M[(H•L)]+k	–	Z	$\overline{C}$	2	
	ADD y ,#k	00 10 11 11	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 F	k y	M[y]←M[y]+k	–	Z	$\overline{C}$	2	
	SUBRC A ,@HL	00 01 01 00		1 4		(AC)←M[(H•L)]–(AC)–$\overline{(CF)}$	$\overline{B}$	Z	$\overline{B}$	1	
	SUBR A ,#k	00 11 10 00	00 01 $k_3k_2k_1k_0$	3 8	1 k	(AC)←k–(AC)	–	Z	$\overline{B}$	2	
	SUBR @HL,#k	00 11 10 00	01 01 $k_3k_2k_1k_0$	3 8	5 k	M[(H•L)]←k–M[(H•L)]	–	Z	$\overline{B}$	2	
Logical	ROLC A	00 00 01 01		0 5		$\boxed{CF}\leftarrow\boxed{AC}$ (rotate left by 1 bit)	C	Z	$\overline{C}$	1	*4
	RORC A	00 00 01 11		0 7		$\boxed{AC}\leftarrow\boxed{CF}$ (rotate right by 1 bit)	C	Z	$\overline{C}$	1	*4
	AND A ,@HL	00 01 11 10		1 E		(AC)←(AC)∧M[(H•L)]	–	Z	$\overline{Z}$	1	
	AND A ,#k	00 11 10 00	00 11 $k_3k_2k_1k_0$	3 8	3 k	(AC)←(AC)∧k	–	Z	$\overline{Z}$	2	
	AND @HL,#k	00 11 10 00	01 11 $k_3k_2k_1k_0$	3 9	7 k	M[(H•L)]←M[(H•L)]∧k	–	Z	$\overline{Z}$	2	
	OR A ,@HL	00 01 11 01		1 D		(AC)←(AC)∨M[(H•L)]	–	Z	$\overline{Z}$	1	
	OR A ,#k	00 11 10 00	00 10 $k_3k_2k_1k_0$	3 8	2 k	(AC)←(AC)∨k	–	Z	$\overline{Z}$	2	
	OR @HL,#k	00 11 10 00	01 10 $k_3k_2k_1k_0$	3 8	6 k	M[(H•L)]←M[(H•L)]∨k	–	Z	$\overline{Z}$	2	
	XOR A ,@HL	00 01 11 11		1 F		(AC)←(AC)∀M[(H•L)]	–	Z	$\overline{Z}$	1	

(continued)

Item / Classification	Assembler Mnemonic	Object Code				Function	Flags *1			Execution Cycle	
		Binary		Hexadecimal							
		1st Byte	2nd Byte	1st Byte	2nd Byte		CF	ZF	SF		
Bit Manipulation	TEST CF	00 00 01 10		0 6		$(SF)\leftarrow(\overline{CF}),(CF)\leftarrow 0$	0	—	*	1	
	TEST A , b	01 01 11 b_1b_0		5 C+b		$(SF)\leftarrow\overline{(AC)\langle b\rangle}$	—	—	*	1	
	TEST @HL, b	01 01 10 b_1b_0		5 8+b		$(SF)\leftarrow\overline{M[(H\cdot L)]\langle b\rangle}$	—	—	*	1	
	TEST y , b	00 11 10 01	10 $b_1b_0y_3y_2y_1y_0$	3 9	8+b y	$(SF)\leftarrow\overline{M[y]\langle b\rangle}$	—	—	*	2	
	TEST %p , b	00 11 10 11	10 $b_1b_0p_3p_2p_1p_0$	3 B	8+b p	$(SF)\leftarrow\overline{P[p]\langle b\rangle}$	—	—	*	2	
	TEST @L	00 11 01 11		3 7		$(SF)\leftarrow\overline{P[(LR)\langle 3:2\rangle+4]\langle(LR)\langle 1:0\rangle\rangle}$	—	—	*	2	
	TESTP CF	00 00 01 00		0 4		$(SF)\leftarrow(CF),(CF)\leftarrow 1$	1	—	*	1	
	TESTP ZF	00 00 11 10		0 E		$(SF)\leftarrow(ZF)$	—	—	*	1	
	TESTP GF	00 00 00 01		0 1		$(SF)\leftarrow(GF)$	—	—	*	1	
	TESTP y , b	00 11 10 01	11 $b_1b_0y_3y_2y_1y_0$	3 9	C+b y	$(SF)\leftarrow M[y]\langle b\rangle$	—	—	*	2	
	TESTP %p. b	00 11 10 11	11 $b_1b_0p_3p_2p_1p_0$	3 B	C+b p	$(SF)\leftarrow P[p]\langle b\rangle$	—	—	*	2	
	SET GF	00 00 00 11		0 3		$(GF)\leftarrow 1$	—	—	1	1	
	SET @HL, b	01 01 00 b_1b_0		5 b		$M[(H\cdot L)]\langle b\rangle\leftarrow 1$	—	—	1	1	
	SET y , b	00 11 10 01	00 $b_1b_0y_3y_2y_1y_0$	3 9	b y	$M[y]\langle b\rangle\leftarrow 1$	—	—	1	2	
	SET %p , b	00 11 10 11	00 $b_1b_0p_3p_2p_1p_0$	3 B	b p	$P[p]\langle b\rangle\leftarrow 1$	—	—	1	2	
	SET @L	00 11 01 00		3 4		$P[(LR)\langle 3:2\rangle+4]\langle(LR)\langle 1:0\rangle\rangle\leftarrow 1$	—	—	1	2	
	CLR GF	00 00 00 10		0 2		$(GF)\leftarrow 0$	—	—	1	1	
	CLR @HL, b	01 01 01 b_1b_0		5 4+b		$M[(H\cdot L)]\langle b\rangle\leftarrow 0$	—	—	1	1	
	CLR y , b	00 11 10 01	01 $b_1b_0y_3y_2y_1y_0$	3 9	4+b y	$M[y]\langle b\rangle\leftarrow 0$	—	—	1	2	
	CLR %p , b	00 11 10 11	01 $b_1b_0p_3p_2p_1p_0$	3 B	4+b p	$P[p]\langle b\rangle\leftarrow 0$	—	—	1	2	
	CLR @L	00 11 01 01		3 5		$P[(LR)\langle 3:2\rangle+4]\langle(LR)\langle 1:0\rangle\rangle\leftarrow 0$	—	—	1	2	
	CLR IL , r	00 11 01 10	11 $r_5r_4r_3r_2r_1r_0$	3 6	C+r_Hr_L	$(INTL)\langle 5:0\rangle\leftarrow(INTL)\langle 5:0\rangle\wedge r\langle 5:0\rangle$	—	—	1	2	
	EICLR IL , r	00 11 01 10	01 $r_5r_4r_3r_2r_1r_0$	3 6	4+r_Hr_L	$(EIF)\leftarrow 1,(INTL)\langle 5:0\rangle\leftarrow(INTL)\langle 5:0\rangle\wedge r\langle 5:0\rangle$	—	—	1	2	
	DICLR IL , r	00 11 01 10	10 $r_5r_4r_3r_2r_1r_0$	3 6	8+r_Hr_L	$(EIF)\leftarrow 0,(INTL)\langle 5:0\rangle\leftarrow(INTL)\langle 5:0\rangle\wedge r\langle 5:0\rangle$	—	—	1	2	
Input/Output	IN %p , A	00 11 10 10	00 10 $p_3p_2p_1p_0$	3 A	2 p	$(AC)\leftarrow P[p]$	—	Z	$\overline{Z}$	2	
	IN %p ,@HL	00 11 10 10	01 10 $p_3p_2p_1p_0$	3 A	6 p	$M[(H\cdot L)]\leftarrow P[p]$	—	—	$\overline{Z}$	2	
	OUT A ,%p	00 11 10 10	10 $\overline{p_4}$0 $p_3p_2p_1p_0$	3 A	8+2$\overline{p_4}$p	$p[p]\leftarrow(AC)$, $p=p_4\cdot p_3p_2p_1p_0$	—	—	1	2	
	OUT @HL,%p	00 11 10 10	11 $\overline{p_4}$0 $p_3p_2p_1p_0$	3 A	C+2$\overline{p_4}$p	$p[p]\leftarrow M[(H\cdot L)]$, $p=p_4\cdot p_3p_2p_1p_0$	—	—	1	2	
	OUT #k ,%p	00 10 11 00	$k_3k_2k_1k_0p_3p_2p_1p_0$	2 C	k p	$p[p]\leftarrow k$	—	—	1	2	
	OUTB @HL	00 01 00 10		1 2		$P[2]\cdot P[1]\leftarrow ROM[F\cdot(E+(CF))\cdot M[(H\cdot L)]]$	—	—	1	2	
Branch•Subroutine	BS a	01 10 $a_{11}a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	6 a_H	a_M a_L	If SF=1 then(PC)←a else null.	—	—	1	2	
	BSS a	10 $d_5d_4d_3d_2d_1d_0$		8+d_Hd_L		If SF=1 then(PC)←a else null,a=(PC)⟨11:6⟩•d	—	—	1	1	*5
	CALL a	00 10 0 $a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	2 a_H	a_M a_L	$STACK[(SPW)]\leftarrow(PC),(SPW)\leftarrow(SPW)-1,$ $(PC)\leftarrow a, 0\leqq a\leqq 2{,}047$	—	—	—	2	*5
	CALLS a	01 11 $n_3n_2n_1n_0$		7 n		$STACK[(SPW)]\leftarrow(PC),(SPW)\leftarrow(SPW)-1,$ $(PC)\leftarrow a, a=8n+6\,(n\neq 0), 134\,(n=0)$	—	—	—	2	*5
	RET	00 10 10 10		2 A		$(SPW)\leftarrow(SPW)+1$, $(PC)\leftarrow STACK[(SPW)]$	—	—	—	2	
	RETI	00 10 10 11		2 B		$(SPW)\leftarrow(SPW)+1$, $(FLAG\cdot PC)\leftarrow STACK[(SPW)]$, $(EIF)\leftarrow 1$	*	*	*	2	
Other	NOP	00 00 00 00		0 0		no operation	—	—	—	1	

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.

Operation Code Map

lover higher	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	TESTP GF	CLR GF	SET GF	TESTP GF	ROLC A	TEST CF	RORC A	INC A	DEC A	INC @HL	DEC @HL	LD A,@HL	XCH A,@HL	TESTP ZF	ST A,@HL
1	MOV H ,A	MOV L,A	OUTB @HL	XCH A,EIR	SUBRC A,@HL	ADDC A,@HL	CMPR A,@HL	ADD A,@HL	INC L	DEC L	ST A,@HL+	ST A,@HL-		OR A,@HL	AND A,@HL	XOR A,@HL
2	CALL a								LD HL,x	XCH HL,x	RET	RETI	OUT #k,%p	ST #k,y	CMPR y,#k	ADD y,#k
3	XCH A,H	XCH A,L	LDH A,@DC+	LDL A,@DC	SET @L	CLR @L	(OP36)	TEST @L	(OP38)	(OP39)	(OP3A)	(OP3B)	LD A,x	XCH A,x	CMPR A,x	ST A,x
4	LD A,#k															
5	SET @HL,b				CLR @HL,b				TEST @HL,b				TEST A,b			
6	BS a															
7	CALLS a															
8	BSS a															
9																
A																
B																
C	LD H , #k															
D	CMPR A , #k															
E	LD L , #k															
F	ST #k , @HL+															

(continued)

1st byte code	OP36	OP38	OP39	OP3A	OP3B
2nd byte code(higher)	36	38	39	3A	3B
0		ADD A, #k	SET • y, b		SET %p , b
1		SUBR A, #k			
2		OR A, #k		IN %p , A	
3		AND A, #k			
4	EICIR IL, r	ADD @HL, #k	CLR y, b		CLR %p , b
5		SUBR @HL, #k			
6		OR @HL, #k		IN %p ,@HL	
7		AND @HL, #k			
8	DICIR IL,r	ADD L, #k	TEST y, b	OUT A,%p	TEST %p , b
9		CMPR L, #k			
A				OUT A,%p	
B					
C	CLR IL,r	ADD H, #k	TESTP y, b	OUT @HL,%p	TESTP %p , b
D		CMPR H, #k			
E				OUT @HL,%p	
F					

(caution)

1. Blank codes are reserved.
2. □ is indicated 2-byte instruction.

PRELIMINARY

東芝 Toshiba

INTEGRATED CIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4700C

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

PRELIMINARY

NMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47N) TMP4700C

GENERAL DESCRIPTION

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

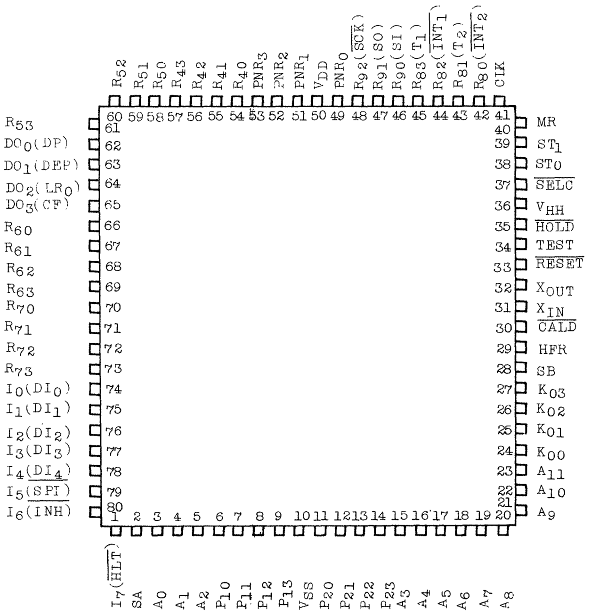
The TLCS-47 has variously powerful functions in order to meet with the advanced and complicated applications, which will be made in near future. In addition, software compatible NMOS family (TLCS-47N) and CMOS family (TLCS-47C) are also provided.

The TMP4700C 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 TMP4700C.

Furhter, when the TMP4700C is used, the evaluation boards equivalent to respective versions of the TLCS-47 should be used.

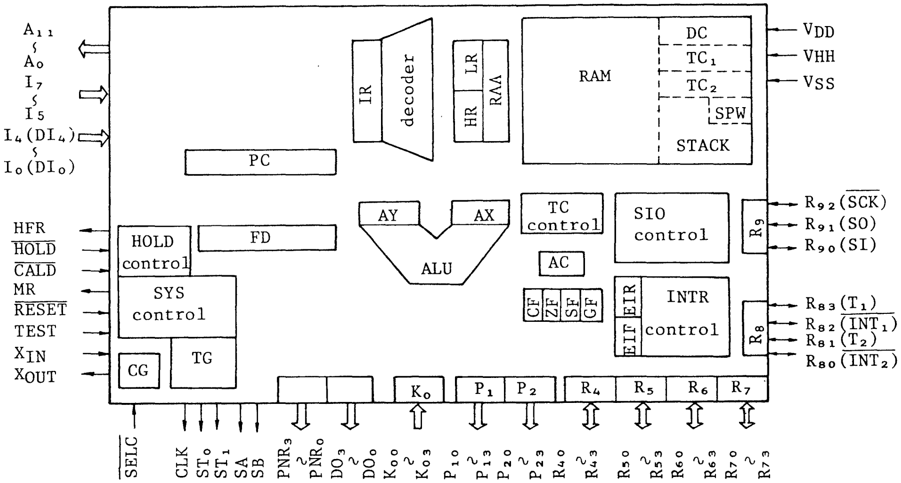
PIN CONNECTIONS (Top View)



PIN NAMES AND PIN DESCRIPTIONS

Pin Names	No.of Pins	I/O	Functions
K_{03} ∿ K_{00}	4	Input	Input port
P_{13} ∿ P_{10}	4	Output	Output port (corresponding to PLA)
P_{23} ∿ P_{20}	4	Output	" (")
R_{43} ∿ R_{40}	4	I/O	I/O port
R_{53} ∿ R_{50}	4	I/O	"
R_{63} ∿ R_{60}	4	I/O	"
R_{73} ∿ 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} ∿ 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)
$I_4(DI_4)$ ∿ $I_0(DI_0)$	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 ∿ 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
XIN, XOUT	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 TMP4700C 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 (IOCODE AA) and in the case of (IOCODE AE) and (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 TMP4700C 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 TMP4700C 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 (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.

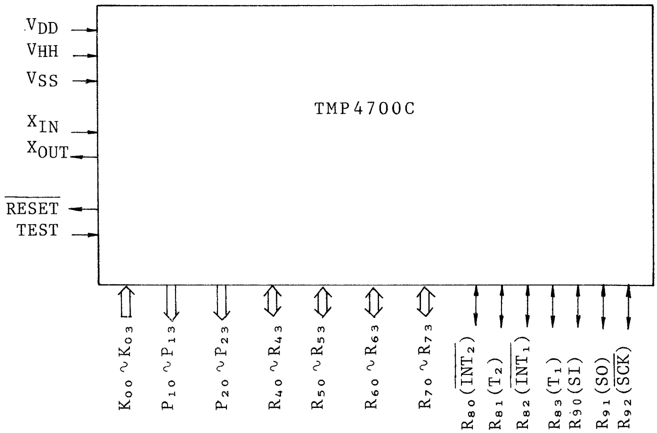
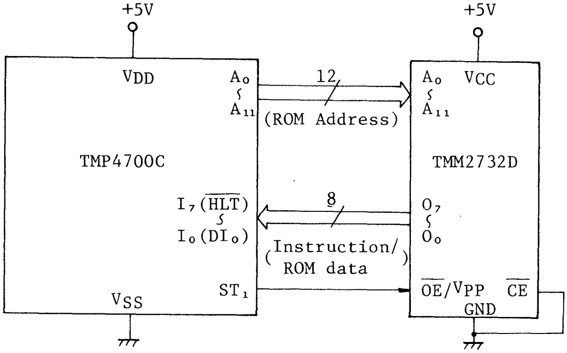


Fig.1.1 TLCS-47N Standard Chip (I$\overline{\text{O}}$C$\overline{\text{O}}$DEAA) Equivalent Terminals

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.

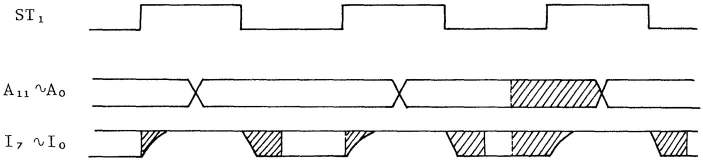
The connecting method of a program memory and the timing chart are shown in Fig. 2.1.
Further, A_{11} and I_7 ($\overline{\text{HLT}}$) terminals in the diagram are MSB, respectively.



Note 1. When the TMM323D-1 is used, the TMP4700C 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

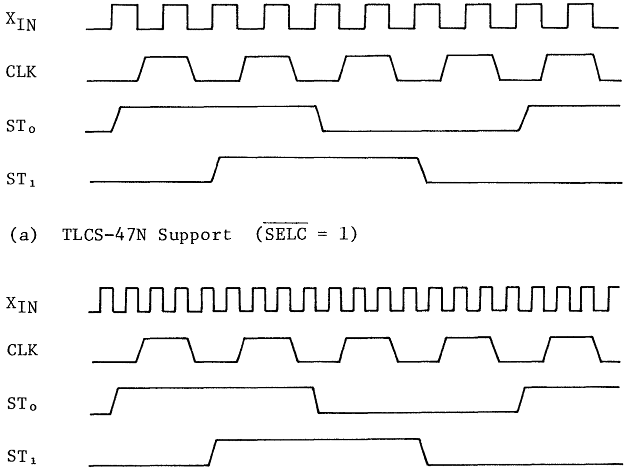
3. Control Terminals for External Circuits

(1) Timing signals (CLK, 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 TMP4700C.

The TMP4700C 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 ($\overline{\text{HLT}}$)

$\overline{\text{HLT}}$ signal is the halt request signal input to the TMP4700C at time of the system debugging. $\overline{\text{HLT}}$ 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 $\overline{\text{HLT}}$ signal input and accepted, the TMP4700C starts the halt 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 ($\overline{\text{HLT}}$) terminal has a built-in pull-up resistor.

I_6 ($\overline{\text{INH}}$)

$\overline{\text{INH}}$ 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. $\overline{\text{INH}}$ 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 $\overline{\text{INH}}$ input and is being accepted, the TMP4700C 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 ($\overline{\text{INH}}$) terminal has a built-in pull-up resistor.

$\overline{\text{HOLD}}$

This input is equivalent to the $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ 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 TMP4700C operating current.

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

$\overline{\text{CALD}}$

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 $\overline{\text{CALD}}$ input and accepted, the TMP4700C executes the LCD data fetch cycle (one instruction cycle).

Further, this $\overline{\text{CALD}}$ 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 TMP4700C (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 TMP4700C 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 ∿ 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 TMP4700C 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.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATING (VSS = 0V)

SYMBOL	ITEM	RATING	UNIT
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 (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 (VSS = 0V)

SYMBOL	ITEM	CONDITION	MIN.	MAX.	UNIT
Topr	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 (Note 1)		2.2	VDD	V
V_{IH2}	High Level Input Voltage (Note 2)		3	VDD	
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 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 (VSS = 0V, VDD = VHH = 5V±10%, Topr = -30 ∿ 70°C)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP. (*)	MAX.	UNIT
V_{HS}	Hysteresis Voltage (Schmitt Circuit Input)		-	0.5	-	V
IIN1	Input Current (KO,$\overline{RESET}$,TEST)	VDD=VHH=5.5V,VIN=5.5V	-	-	20	µA
IIN2	Input Current (R Port)	VDD=5.5V, VIN=5.5V	-	-	20	
IIL	Current (**)	VDD=5.5V, VIN=0.4V	-	-	-2	mA
ILO	Output Leakage Current (P, R Port)	VDD=5.5V, VOUT=5.5V	-	-	20	µA
VOH	High Level Output Voltage (***)	VDD=4.5V,IOH=-400µA	2.4	-	-	V
V_{OL}	Low Level Output Voltage (Except XOUT)	VDD=4.5V, I_{OL}=1.6mA	-	-	0.4	
IOL	Low Level Output Current (P_1, P_2)	VDD=5V, VOL=1V	-	20	-	mA
IDD+IHH	Supply Current	VDD=VHH=5.5V	-	70	150	mA
IHH1	Supply Current (Memory stand-by)	VDD=VSS, VHH=3.5V	-	5	10	

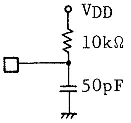
(*) TYP. values are at Topr=25°C, VDD=VHH=5V.
(**) Application terminals: $\overline{HOLD}$, $\overline{CALD}$, $\overline{SELC}$, I_7($\overline{HLT}$) ∿ I_0(DI_0).
(***) Application terminals: Control output terminal specific to evaluation.

A.C. CHARACTERISTICS (VSS = 0V, VDD = VHH = 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

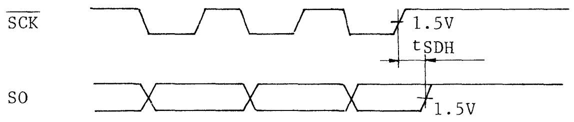


(2)

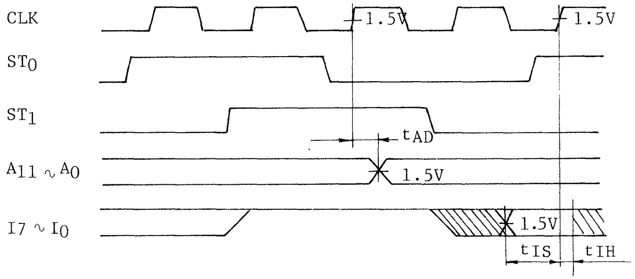
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tAD	Address Delay Time	CL = 100pF	-	-	270	nS
tIS	Data Set-up Time	"	150	-	-	
tIH	Data Hold Time	"	50	-	-	

A.C. Timing Chart

(1) Serial Port (Completion of transmission)

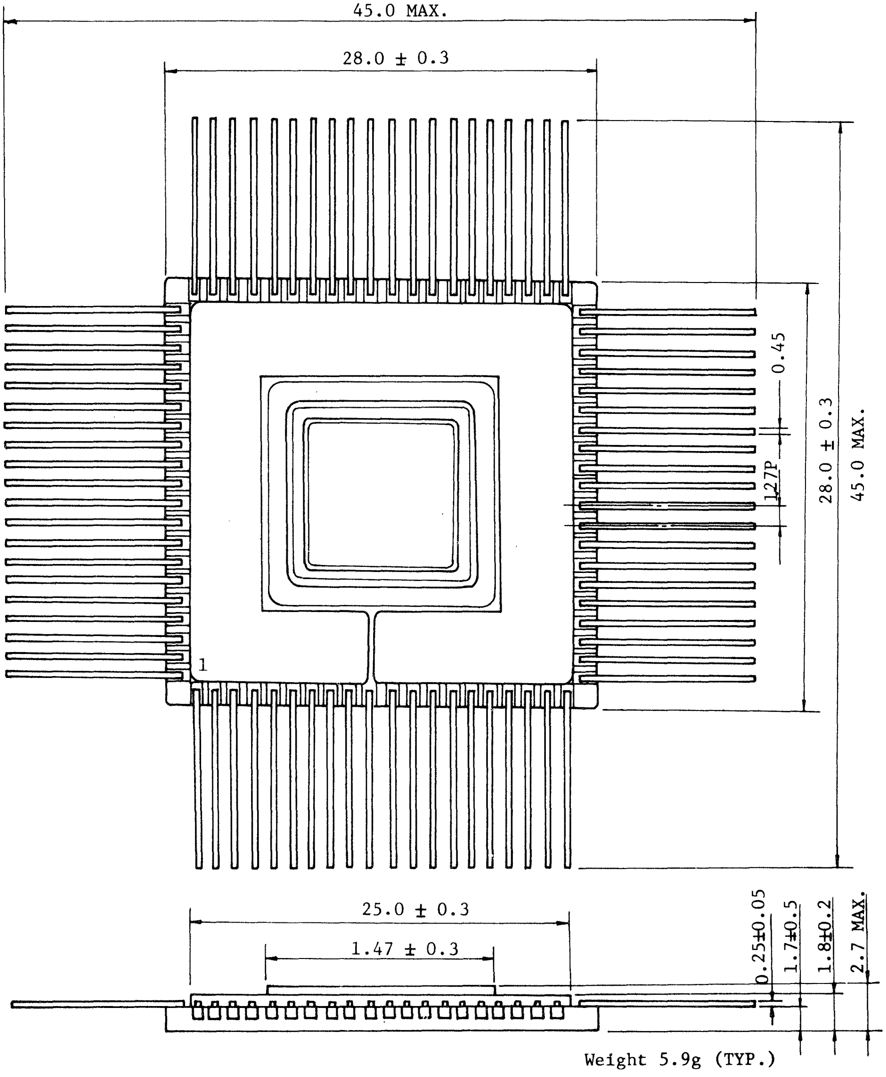


(2)



EXTERNAL DIMENSIONS

Unit : mm



Weight 5.9g (TYP.)

Toshiba 東芝 **INTEGRATED CIRCUIT**

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP4799C

SILICON MONOLITHIC

N-CHANNEL SILICON GATE DEPRESSION LOAD

PRELIMINARY

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

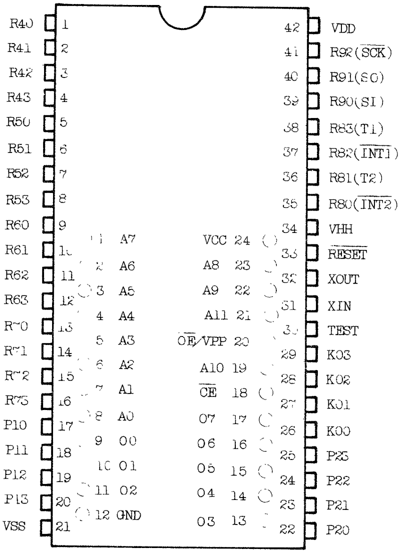
TMP4799C

PRELIMINARY

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)



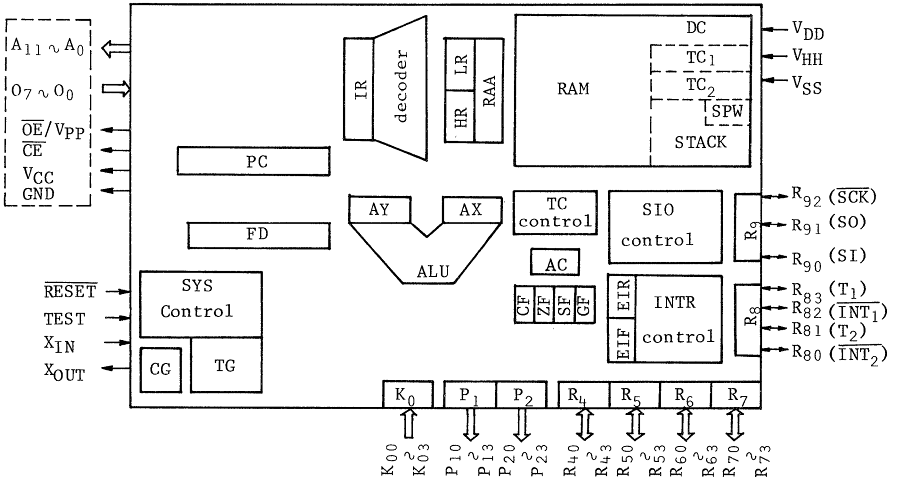
(NOTE) Mark : Socket for TMM2732D

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 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
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 7EO - 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 FEO - FFF.

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

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

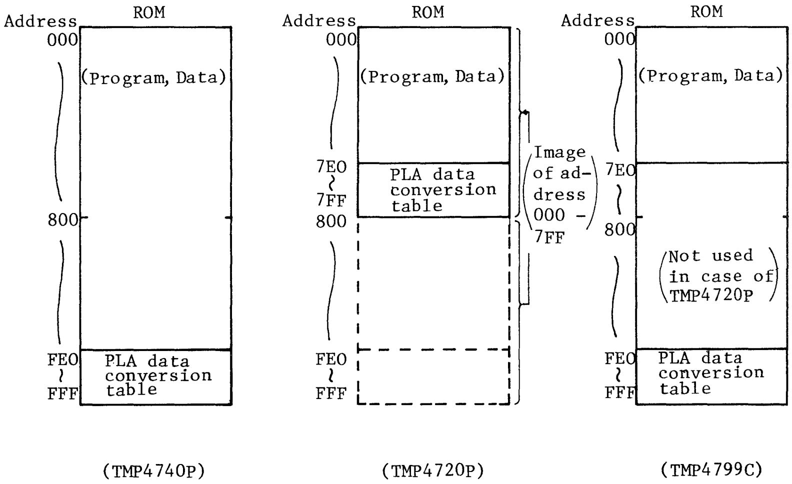


Fig. 1.1 ROM Capacity and Address

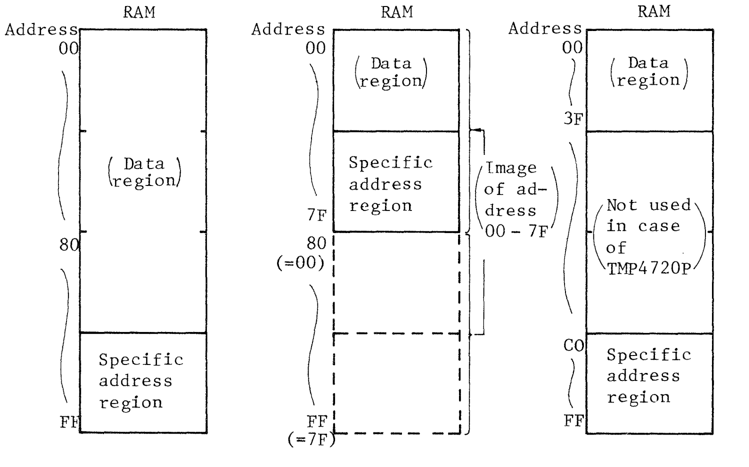
2. Data Memory (RAM) and RAM address

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 CO - 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.

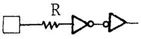
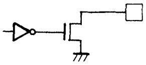
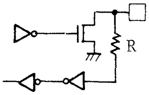
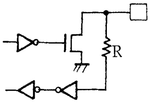
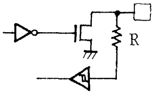


(TMP4740P) (TMP4720P) (TMP4799C)

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.

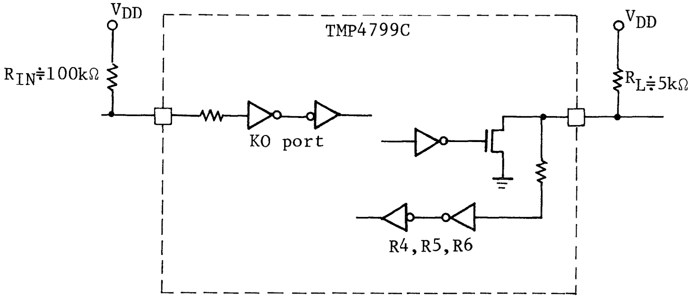
port Circuit	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
I/O equivalent Circuit	 R=1kΩ (TYP.)		 R=1kΩ (TYP.)	 R=1kΩ (TYP.)	 R=1kΩ (TYP.)
Remark	◦ High threshold input. ◦ No resistor is contained.	◦ Sink open-drain output. ◦ High output current. ◦ Output latch is initialized 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 (≈ 200kΩ TYP.) and to provide TEST pin with the pull-down resister (≈ 70kΩ TYP.), respectively.

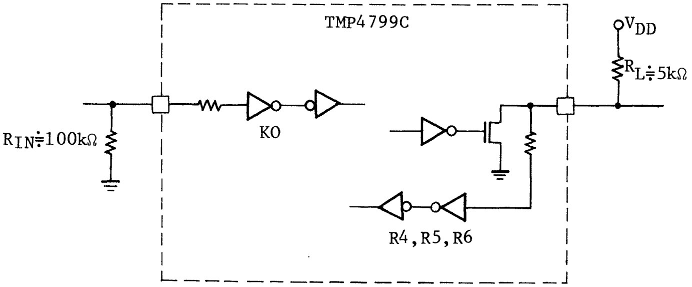
Fig. 3.1 Input/Output circuit format of TMP4799C

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

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 ∿ 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.

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	
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}+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	

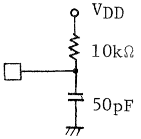
(*) TYP. values are at T_{opr}=25°C, V_{DD}=V_{HH}=5V.
(**) Application terminals : O_7 ∿ O_0
(***) Application terminals : A_{11} ∿ A_0, $\overline{OE}$/V_{PP}

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

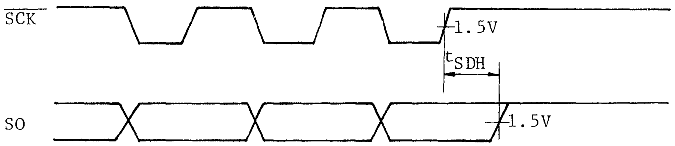


(2)

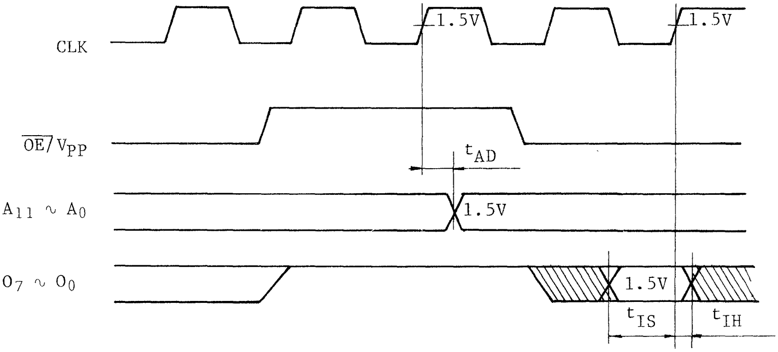
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
t_{AD}	Address Delay Time	C_L=100pF	-	-	270	nS
t_{IS}	Data Set-up Time	"	150	-	-	
t_{IH}	Data Hold Time	"	50	-	-	

A.C. Timing Chart

(1) Serial Port (Completion of transmission)

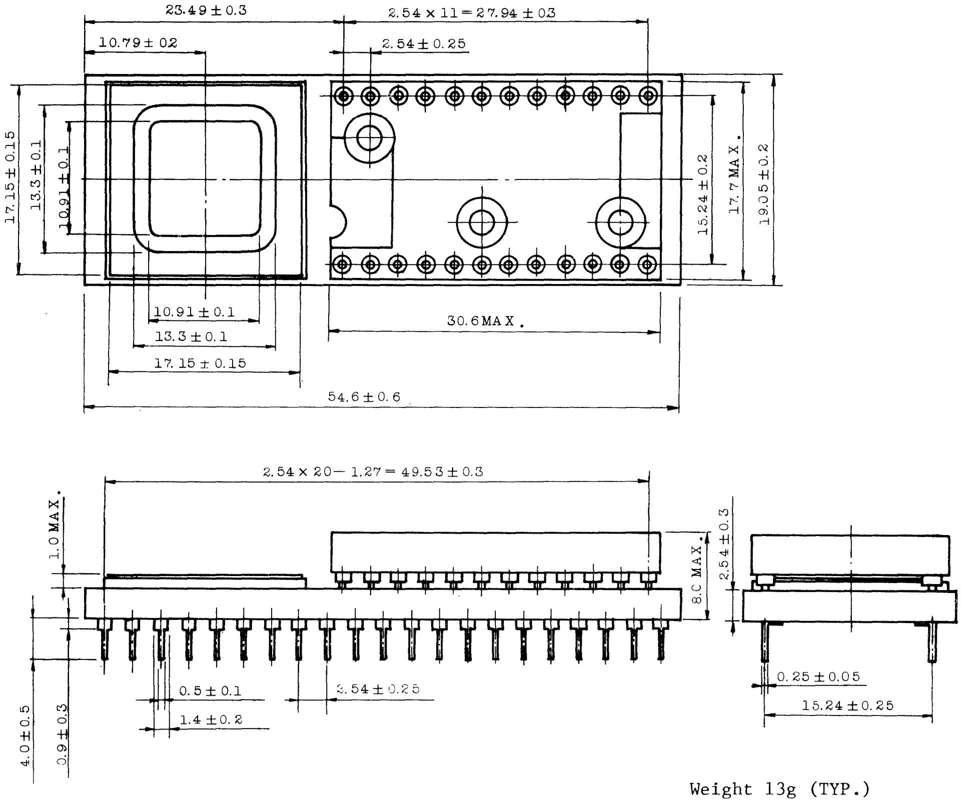


(2)



EXTERNAL DIMENSION VIEW

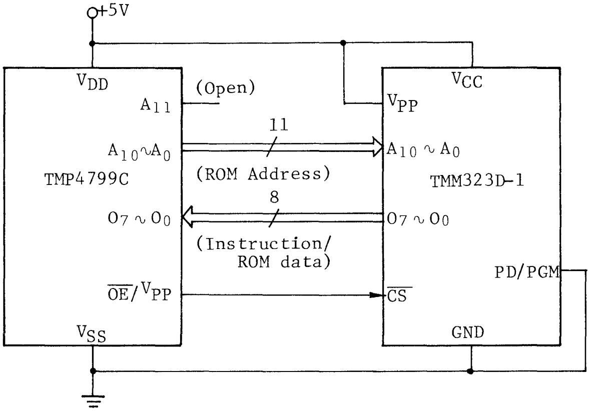
Unit in mm



Weight 13g (TYP.)

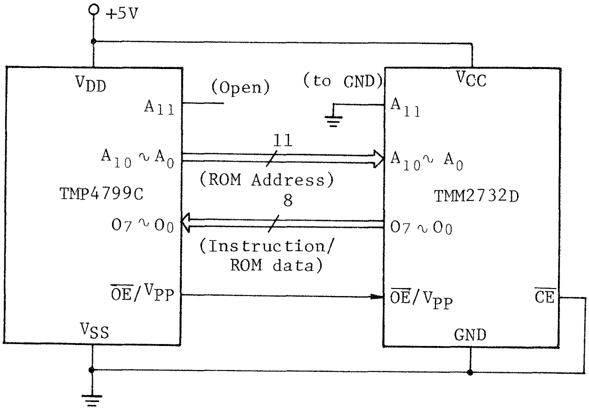
CONNECTION OF PROGRAM MEMORY

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}.

TLCS-47(C)

- TMP47C40P
- TMP47C20P
- TMP47C41P
- TMP47C21P
- TMP47C22F

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT

TMP47C40P, TMP47C41P
TMP47C20P, TMP47C21P

SILICON MONOLITHIC SILICON GATE CMOS

PRELIMINARY

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 TMP4700C (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
TMP4700C	Externally provided (4,096 × 8)	256 × 4

TMP47C40P, TMP47C41P
TMP47C20P, TMP47C21P

PRELIMINARY

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

Toshiba 東芝

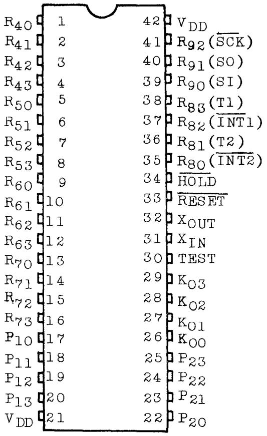
INTEGRATED CIRCUIT

TECHNICAL DATA

TMP47C40P, TMP47C41P
TMP47C20P, TMP47C21P

PRELIMINARY

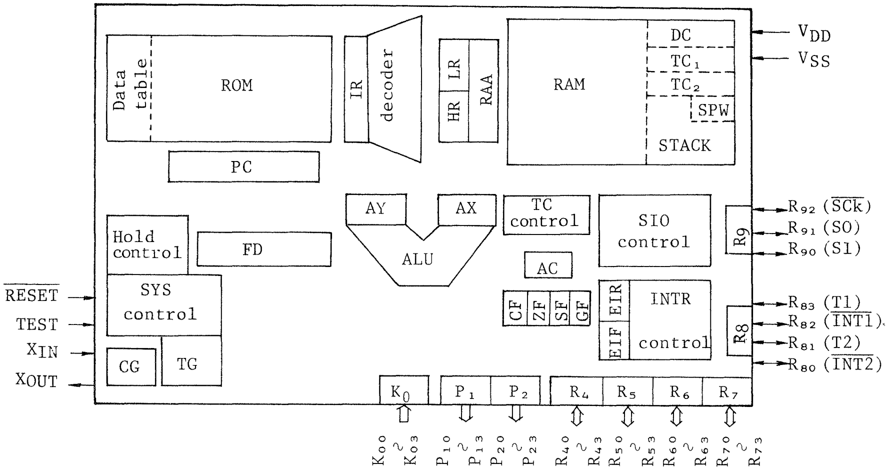
PIN CONNECTIONS (Top View)



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}, XOUT	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

東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TMP47C40P

TMP47C20P

PRELIMINARY

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)

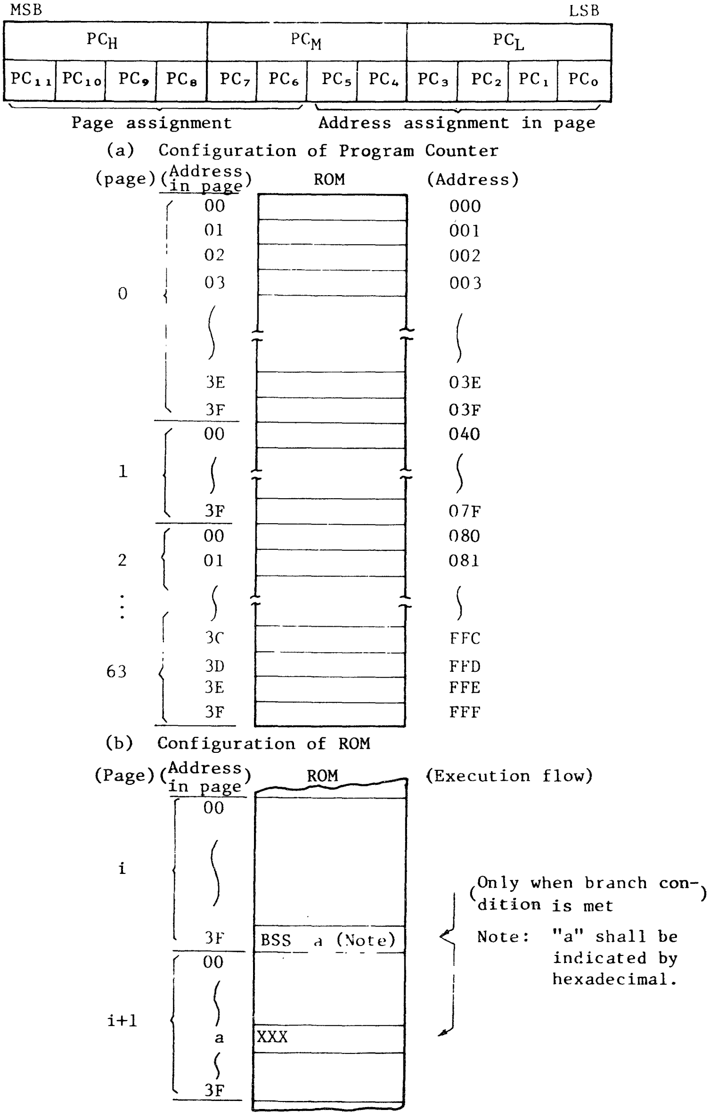
It is a 12-bit binary counter, and the contents of the program counter indicate the address of program memory in which the next instruction to be executed is stored.

The program counter generally gains increment at every instruction fetch by the number of 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 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 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 Countei and Program Memorv (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 acculator. 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)	Call address by instruction (CALLS a)
086 (Note)	
FEO ≀ 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 FEO - FFF) is, therefore, located in addresses 7EO - 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.

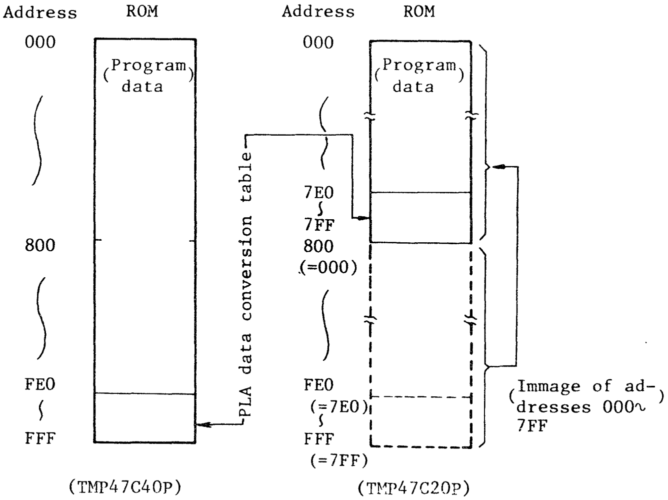


Fig. 1.2.1 ROM Capacity and Address

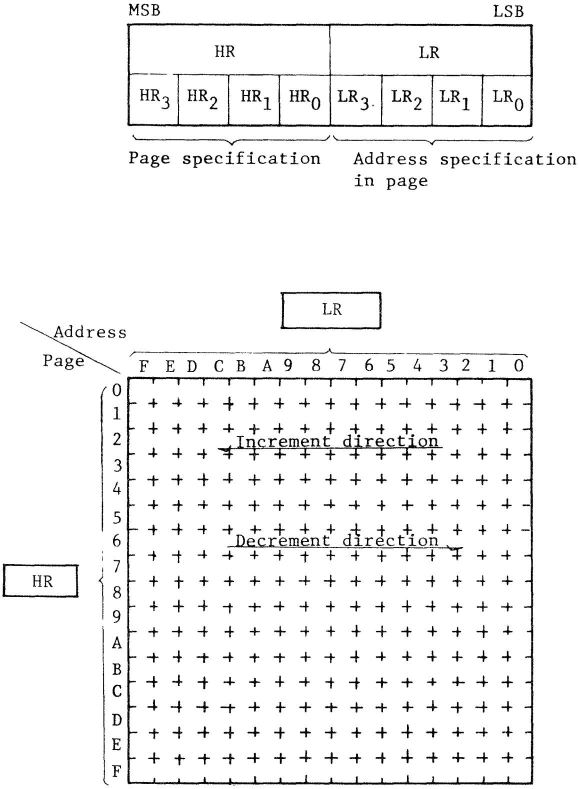
1.3 H Register (HR), L Register (LR), and RAM Address Buffer Register (RAA)

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.


MSB
LSB
HR
LR
HR3 HR2 HR1 HR0 LR3 LR2 LR1 LR0
Page specification
Address specification in page
LR
Address
Page
F E D C B A 9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9 A B C D E F
Increment direction
Decrement direction
HR

Configuration of RAM

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)

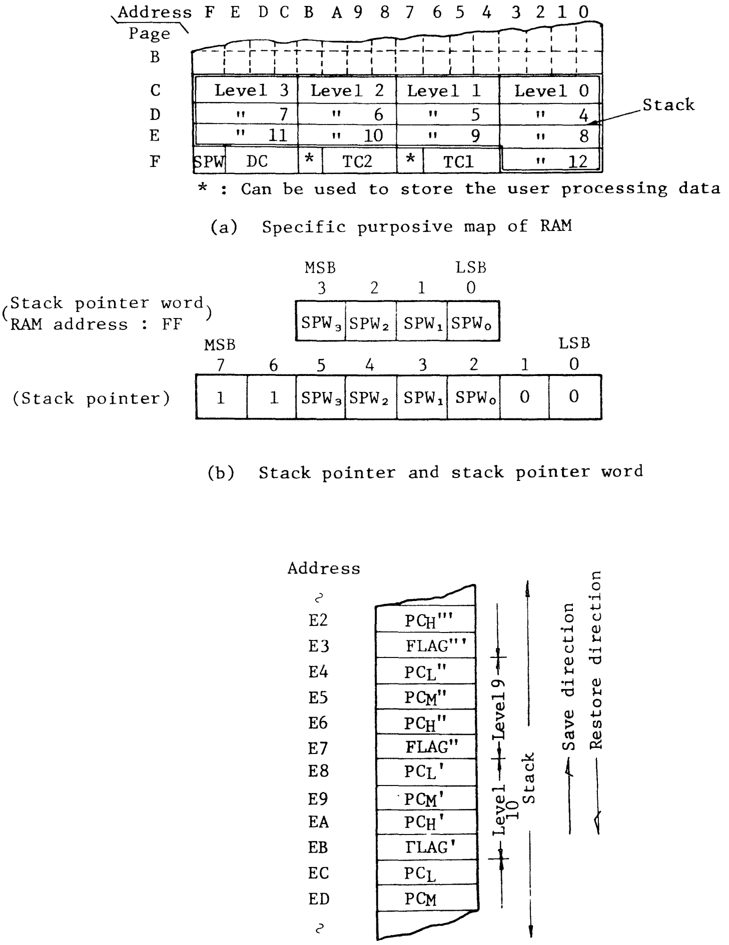
The address FF in the data memory is called a stack pointer word and decides the stack pointer. The stack is contained in the RAM, and accessed by the stack pointer.

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

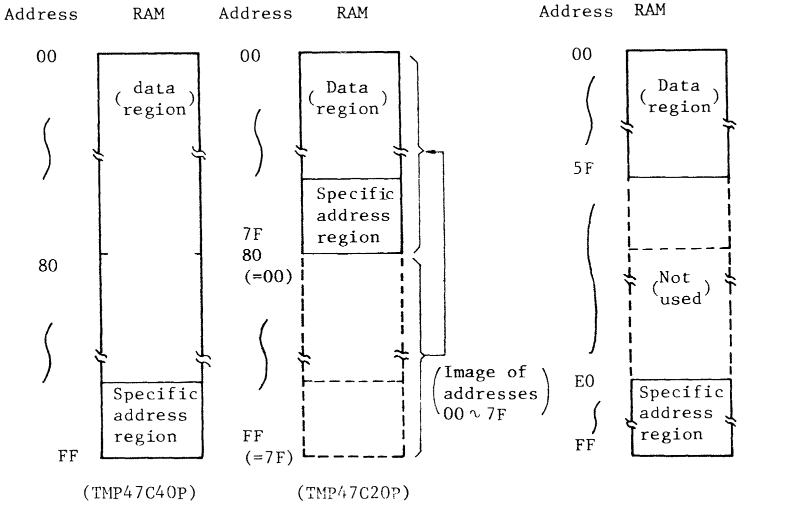
(5) Page 0 in Data Memory

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.



(a) RAM Capacity and Address

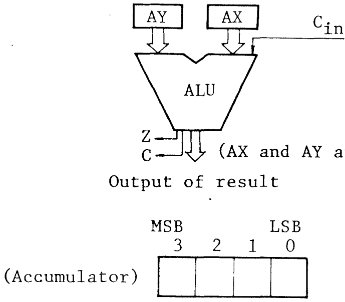
(b) RAM Map example of TMP47C20P (TC₁, TC₂ and stack 5 level are used.)

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.



(AX and AY are temporary registers.)

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 "$\overline{\text{C}}$".

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.

3	2	1	0
CF	ZF	SF	GF

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 2.6 (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 / ———	0					0	
01	IP01/OP01	P_1Output latch/ P_1 Output 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	———							
0C	IP0C/OP0C	———							
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	/ ———							
16	/OP16	/ ———							
17	/OP17	/ ———							
18	/OP18	/ ———							
19	/OP19	/ (b)		0					
1A	/OP1A	/ ———							
1B	/OP1B	/ ———							
1C	/OP1C	/ (c)		0					
1D	/OP1D	/ (d)		0					
1E	/OP1E	/ ———							
1F	/OP1F	/ (e)		0					

(*) Serial buffer register (Reception)
(**) Serial buffer register (Transmission)

(a) Hold control
(b) Control with timer interrupt of divider
(c) Timer/Counter 1 control
(d) Timer/Counter 2 control
(e) Serial port control

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} ∿ K_{00}) Port

This is a 4-bit port used for input.

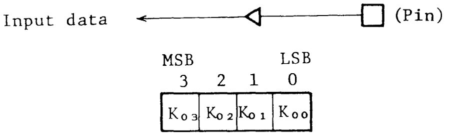


Fig. 1.7.1 K_0 Port

(2) P_1 (P_{13} ∿ P_{10}), P_2 (P_{23} ∿ 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 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.

	MSB 7	6	5	4	3	2	1	LSB 0
Conversion data	P_{23}	P_{22}	P_{21}	P_{20}	P_{13}	P_{12}	P_{11}	P_{10}

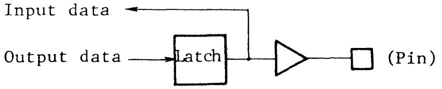


Fig. 1.7.2 P_1 and P_2 Ports

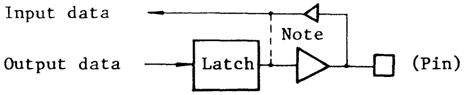
(3) $R_4(R_{43} \sim R_{40})$, $R_5(R_{53} \sim R_{50})$, $R_6(R_{63} \sim R_{60})$, $R_7(R_{73} \sim R_{70})$ Port

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				Corresponding Pin	L register				Corresponding Pin
3	2	1	0		3	2	1	0	
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

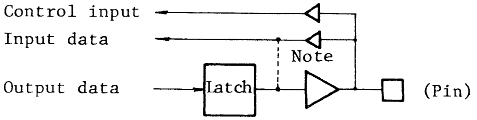
(4) R_8 ($R_{83} \sim R_{80}$) Port

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} ($\overline{\text{INT1}}$) 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} ($\overline{\text{INT2}}$) 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.

MSB 3	2	1	LSB 0
R_{83} (T1)	R_{82} ($\overline{\text{INT1}}$)	R_{81} (T2)	R_{80} ($\overline{\text{INT2}}$)



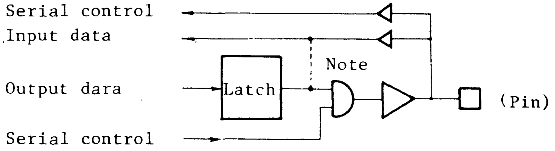
Note: For bit set/reset of port, latch output serves as input data.

Fig. 1.7.4 R_8 Port

(5) R9(R92 ∿ R90) 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.

MSB 3	2	1	LSB 0
	R_{92} ($\overline{\text{SCK}}$)	R_{91} (SO)	R_{90} (SI)



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

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

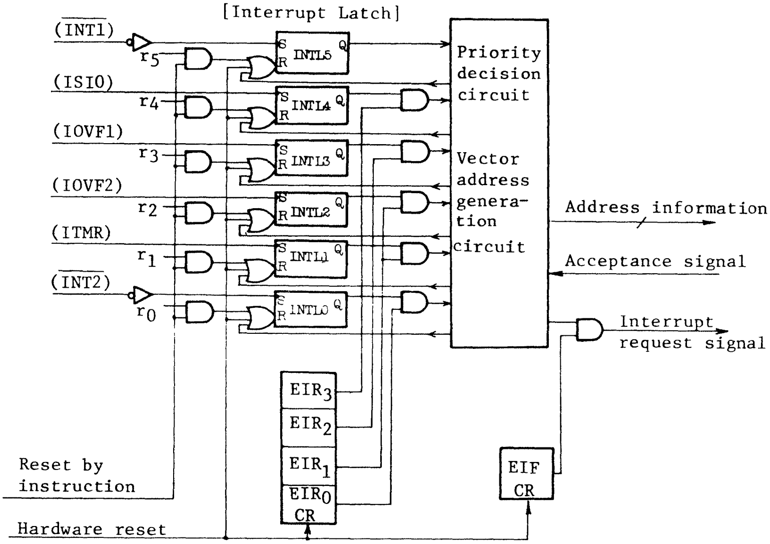
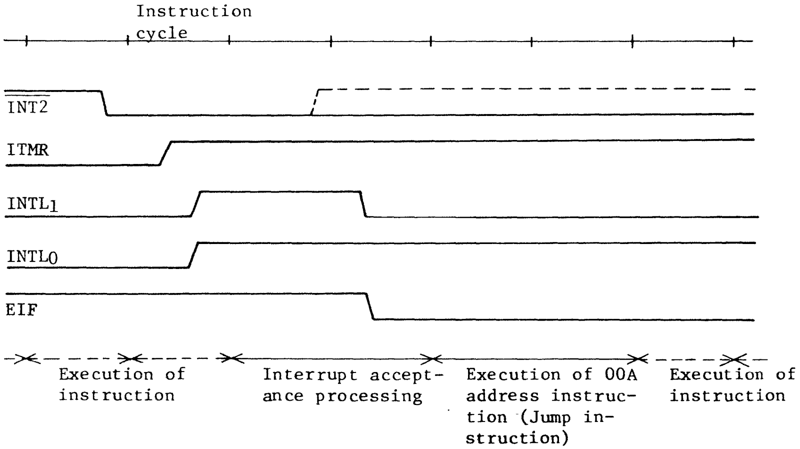


Fig. 1.8.1 Interrupt Control Circuit



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.

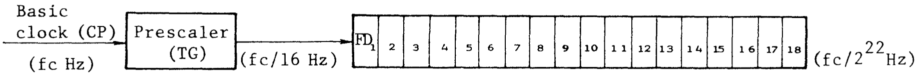
The basic clock (fc Hz) is divided into sixteen by the timing generator and input to the divider; therefore, the output frequency at the last stage is $fc/2^{22}$ Hz.
It is reset to "0" at the initialization operation.

Timer Interrupt of divider (ITMR)

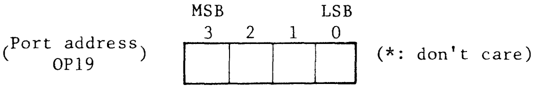
The divider is capable of sending the interrupt request for a certain frequency. Four different frequencies can be selected for timer interrupt by instructions.

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



3	2	1	0			
*	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 fre-quency (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.

	MSB		LSB
(Timer/Counter 1)	TC1		
	TC1H	TC1M	TC1L
(RAM Address)	(F6)	(F5)	(F4)

	MSB		LSB
(Timer/Counter 2)	TC2		
	TC2H	TC2M	TC2L
(RAM Address)	(FA)	(F9)	(F8)

Fig. 1.10.1 Timer/Counter

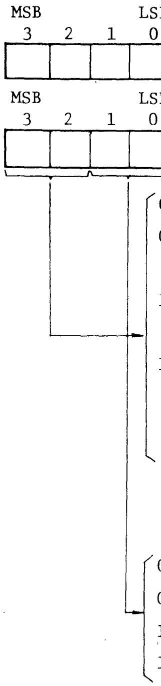
(1) Timer/Counter Control

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)].

TC1 Command register
(Port Address OP1C)

TC2 Command register
(Port Address OP1D)



Selection of Count Mode

- 00 : Disable state
- 01 : Event count mode (Counts external input)
- 10 : Timer mode (Counts internal pulse.)
- 11 : Pulse width measurement mode (Counts the pulse obtained by sampling of external input by use of internal pulse.)

Selection of Internal Pulse (Divider Output) Rate

- 00 : $fc/2^{10}$ Hz is counted.
- 01 : $fc/2^{14}$ Hz is counted.
- 10 : $fc/2^{18}$ Hz is counted.
- 11 : $fc/2^{22}$ Hz is counted.

(fc: Basic clock frequency)

(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

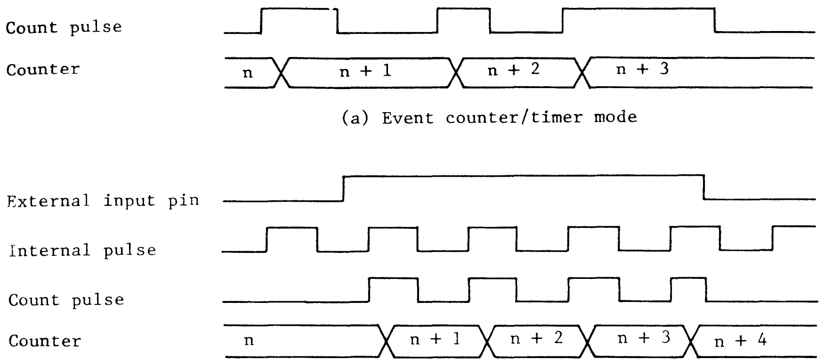
(2) Count Operation

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≒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.

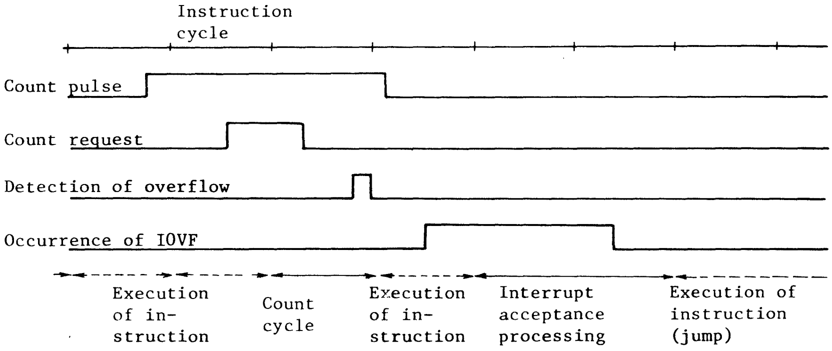


Fig. 1.10.4 Timing Chart of Timer/Counter in Interrupt by Overflow

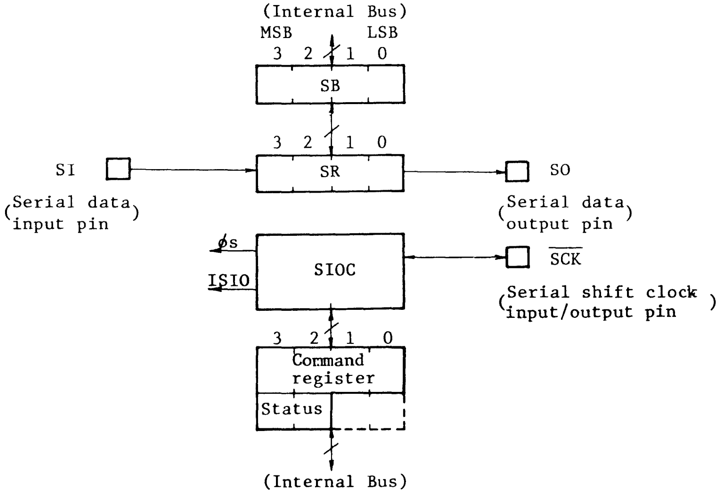
1.11 Serial Port (SIO)

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 ($\overline{SCK}$), $R9_1$ (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 $R9_1$ 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.

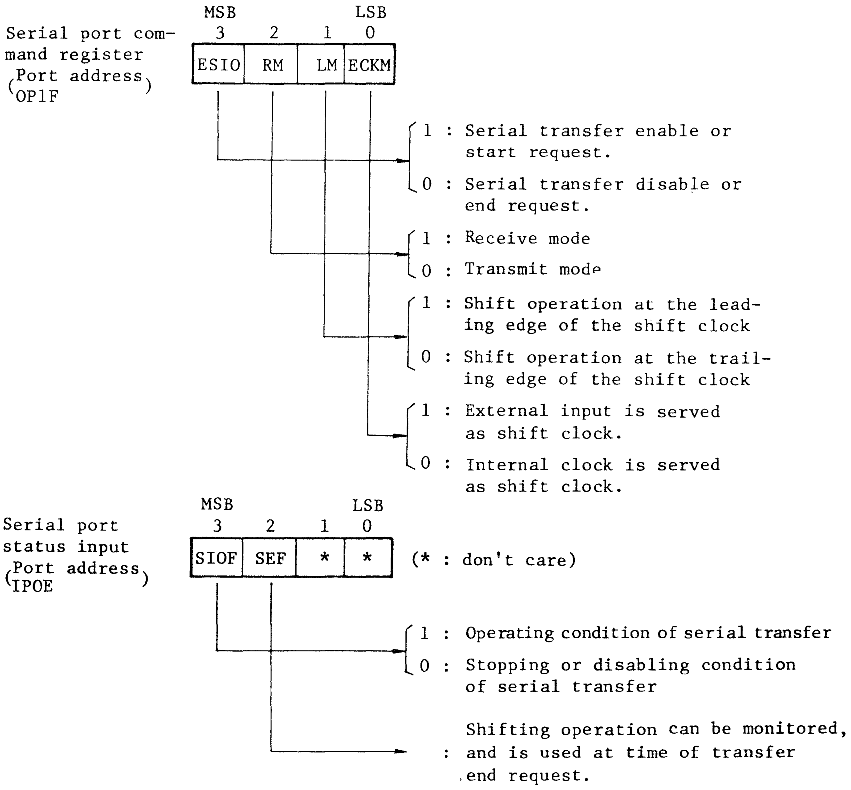


SR : 4-bit shift register
SB : 4-bit buffer register
SIOC : Serial port control circuit
ϕs : Internal shift clock
ISIO : Interrupt request

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.



Serial port command register (Port address OP1F)

MSB 3	2	1	LSB 0
ESIO	RM	LM	ECKM

ESIO:
1 : Serial transfer enable or start request.
0 : Serial transfer disable or end request.

RM:
1 : Receive mode
0 : Transmit mode

LM:
1 : Shift operation at the leading edge of the shift clock
0 : Shift operation at the trailing edge of the shift clock

ECKM:
1 : External input is served as shift clock.
0 : Internal clock is served as shift clock.

Serial port status input (Port address IPOE)

MSB 3	2	1	LSB 0
SIOF	SEF	*	*

(* : don't care)

SIOF:
1 : Operating condition of serial transfer
0 : Stopping or disabling condition of serial transfer

SEF:
: Shifting operation can be monitored, and is used at time of transfer end request.

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".

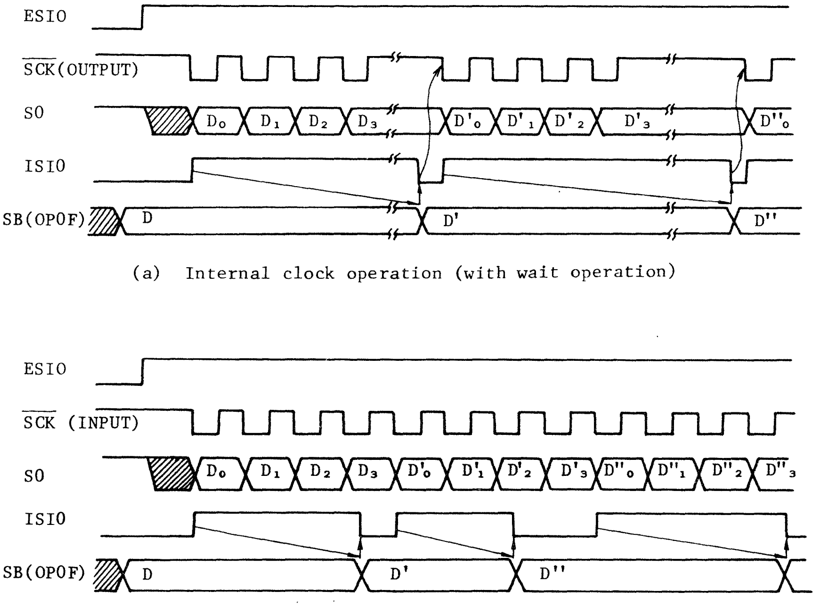
Internal clock operation

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

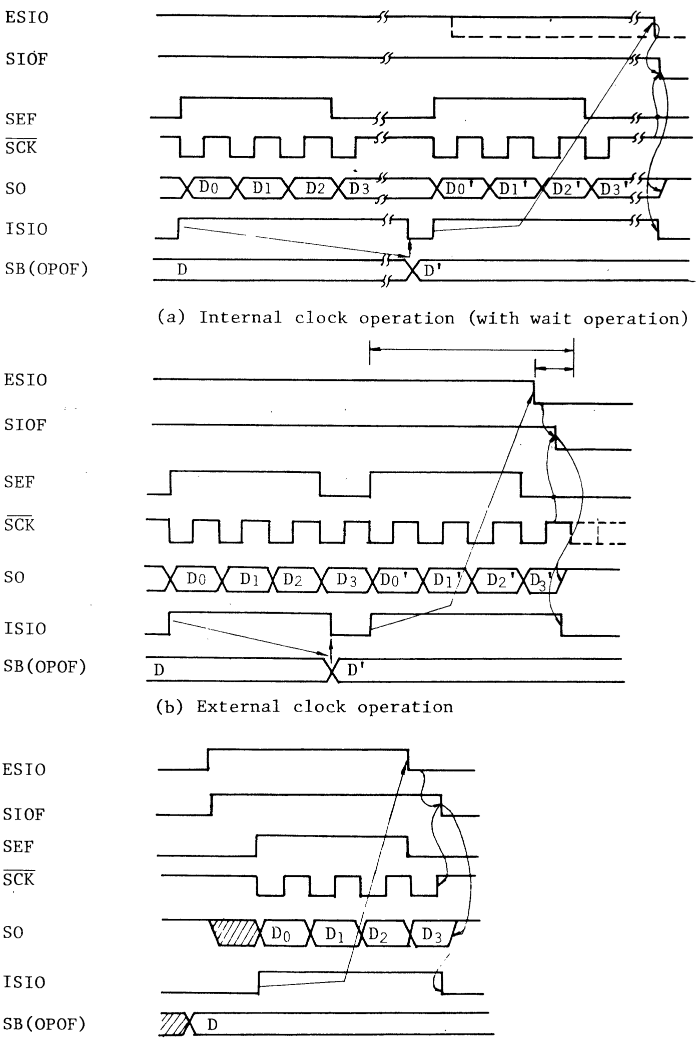
Fig. 1.11.3 Transmit Mode

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

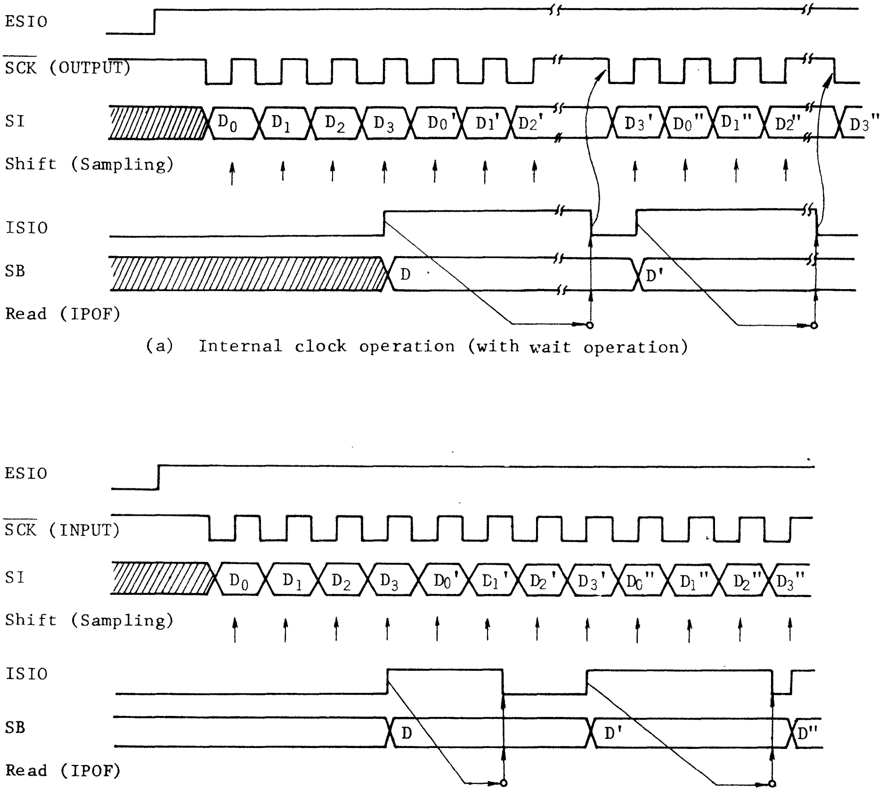
During the operation of the internal clock of fc/2^7Hz, if the next 4-bit data is not read out of the buffer register (the IPOF has not been accessed) in the program though the 4-bit data has been entirely input, the shift clock·automatically stops, and the wait operation is taking place until the data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

PRELIMINARY



(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.

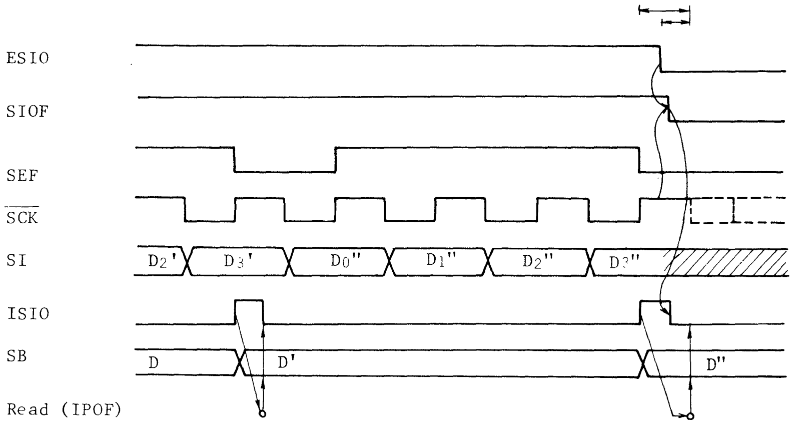


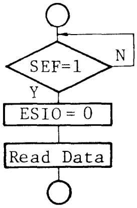
Fig. 1.11.6 Completion of Receiving (at slow transfer rate)

(b) Fast transfer rate

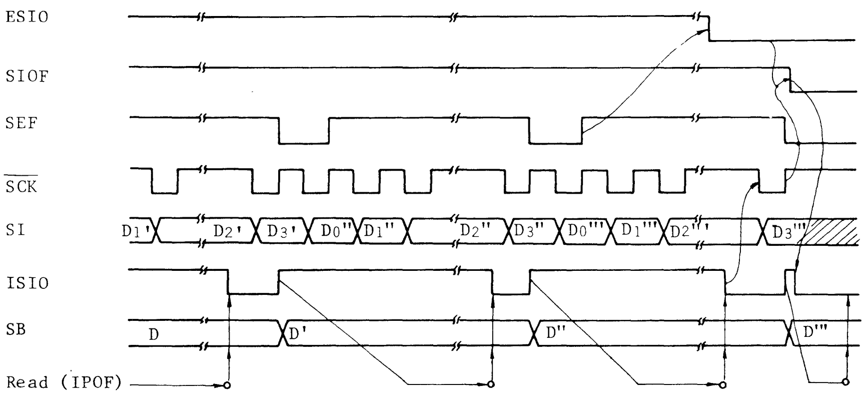
If the shift operation for the next data may start before the current data is read out by receipt of the interrupt request becuase the transfer rate is too fast, the interrupt service program which is loaded to read out the last data but one should be used to reset the ESIO to "0" after confirming that 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

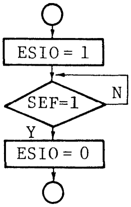


(b) Timing Chart
(in case of internal clock operation with wait operation)

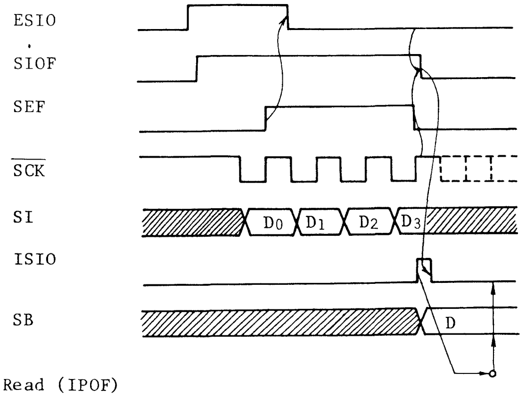
Fig. 1.11.7 Completion of Receiving (at fast transfer rate)

(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)

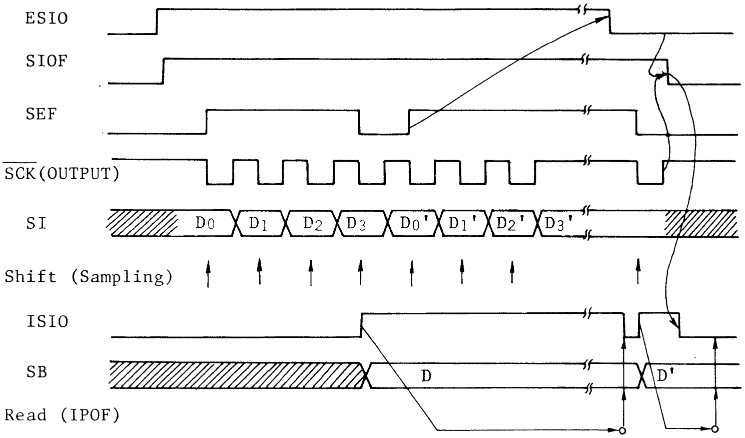
Receive (leading edge shift) mode

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

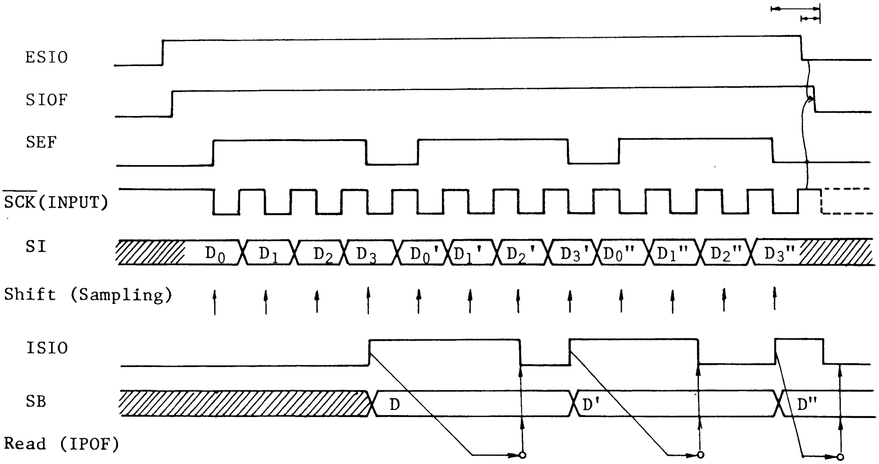
When the four data are received from the SI pin, the 4-bit data is transferred from the shift register to the buffer register. At the same time, the interrupt (ISIO) occurs to request the data reading (buffer full). Since the shift register is transferring the data to the buffer register, the shift operation 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.

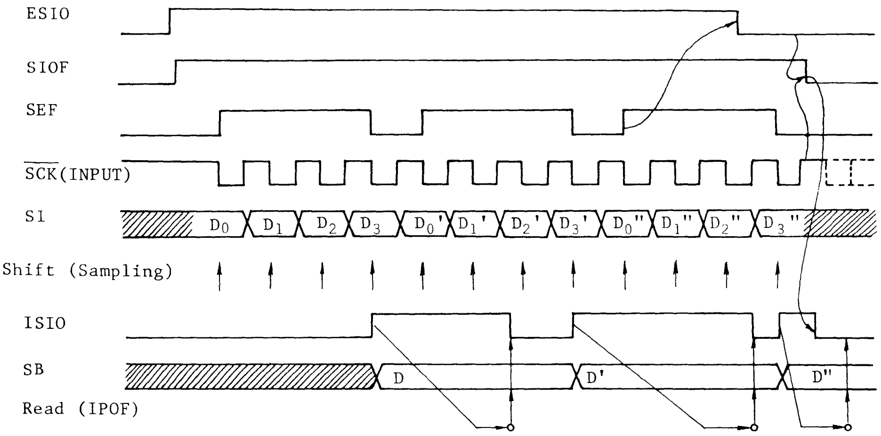
The basic operation in the receive (leading edge shift) mode is equivalent to that in the receive (trailing edge shift) mode except that the edge for the shift clock is different, and that at time of the transfer start, the first shifted data has been already input from the external circuitry before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



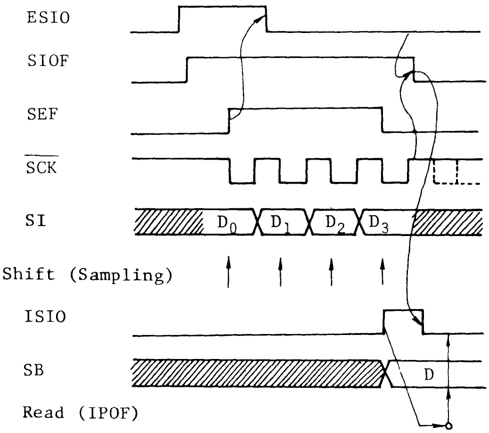
(a) Internal clock operation (with wait operation)



(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.

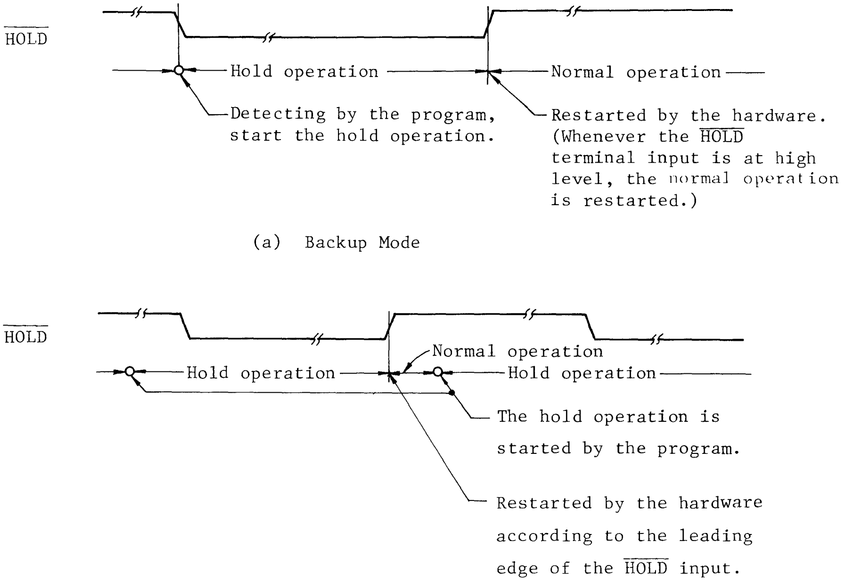
There are two operation modes for the hold operation as shown below. The designation for the hold operation start is made by the command in either mode.

(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.


$\overline{\text{HOLD}}$
Hold operation
Normal operation
Detecting by the program, start the hold operation.
Restarted by the hardware. (Whenever the $\overline{\text{HOLD}}$ terminal input is at high level, the normal operation is restarted.)
$\overline{\text{HOLD}}$
Hold operation
Normal operation
Hold operation
The hold operation is started by the program.
Restarted by the hardware according to the leading edge of the $\overline{\text{HOLD}}$ input.

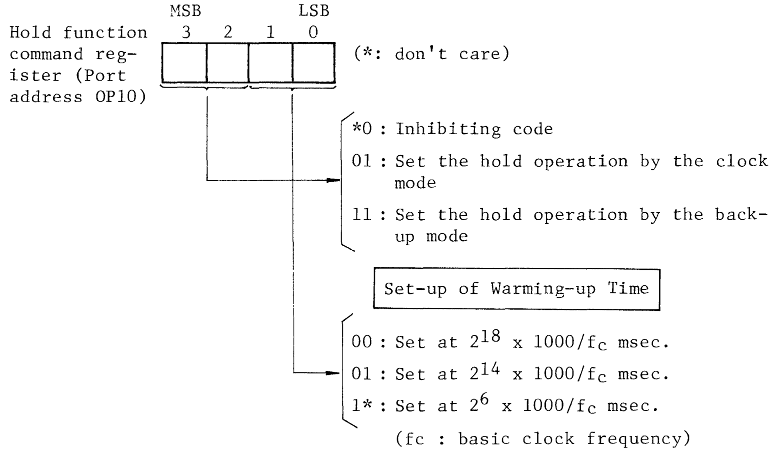
(a) Backup Mode

(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 $\overline{\text{HOLD}}$ terminal input from the status input. The status input is accessed as the port address IPOE.


MSB
LSB
Hold function command register (Port address OP10)
3
2
1
0
(*: don't care)
Set-up of Warming-up Time

*0 : Inhibiting code

01 : Set the hold operation by the clock mode

11 : Set the hold operation by the back-up mode

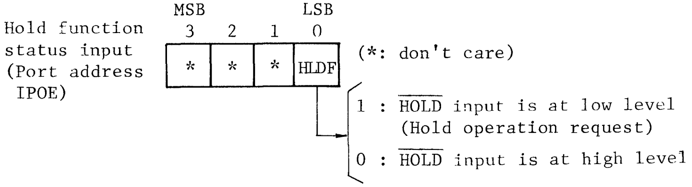
00 : Set at 2^{18} x $1000/f_c$ msec.

01 : Set at 2^{14} x $1000/f_c$ msec.

1* : Set at 2^6 x $1000/f_c$ msec.

(fc : basic clock frequency)

Warming-up Time (m SEC)	For example, at f_c=4MHz (m SEC)
2^{18} x $1000/f_c$	≒ 65.5
2^{14} x $1000/f_c$	≒ 4.1
2^6 x $1000/f_c$	≒ 0.016



1 : $\overline{\text{HOLD}}$ input is at low level (Hold operation request)

0 : $\overline{\text{HOLD}}$ input is at high level

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 $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ input to the $\overline{\text{INT}_1}$ input as an interrupt request.

The hold operation is released when the $\overline{\text{HOLD}}$ terminal input becomes the high level. That is, under the backup mode, the hold status is kept held as long as the $\overline{\text{HOLD}}$ input is at the low level. However, if the $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ 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. 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.

2.1 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.

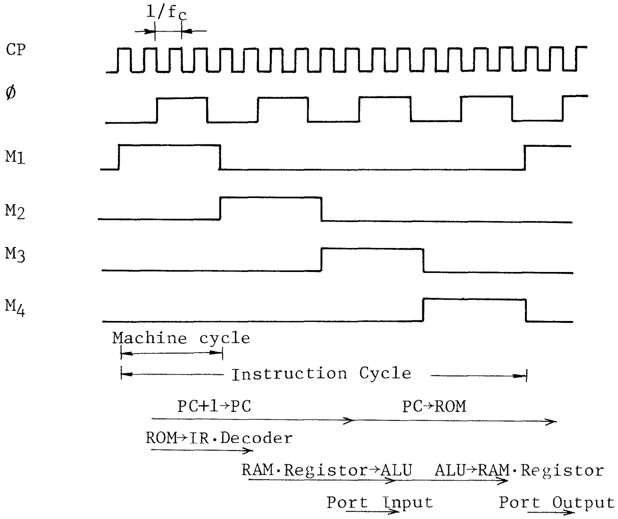


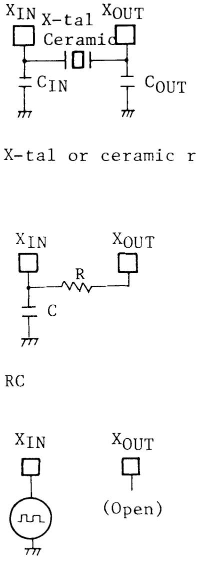
Fig. 2.1.1 Instruction Cycle

2.2 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, thc 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. 2.2.1 Resonator Connections

2.3 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 (≈ 200kΩ TYP., MOS-load resistor).

2.4 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.

2.5 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 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 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.

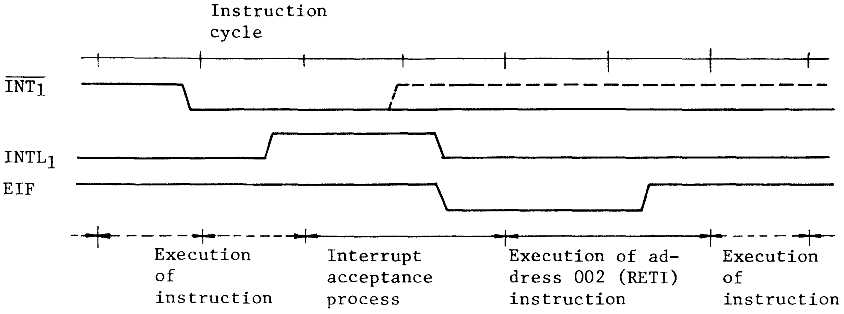


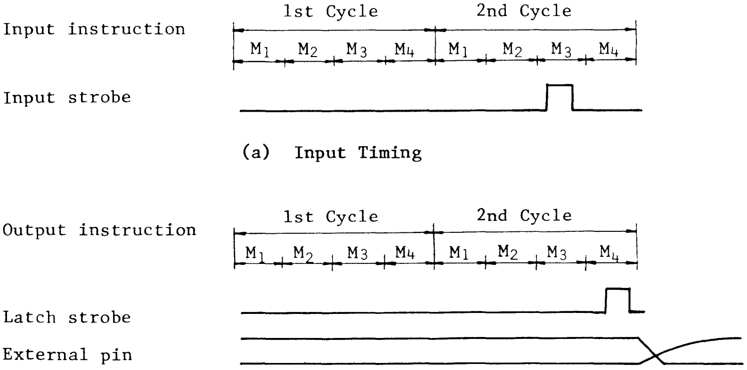
Fig. 2.5.1 Interrupt Timing (Dummy process of INT_1 interrupt)

2.6 Input/output port

(1) Input/output timing

The timing to read the external data from the input port or I/O port is in M3 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 M4 machine cycle in the second cycle of the output instruction (two-cycle instruction), but this timing cannot be externally recognized.



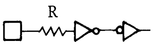
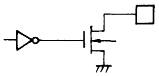
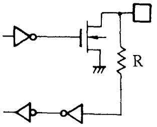
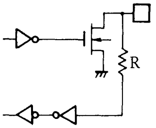
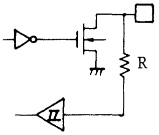
(b) Output Timing

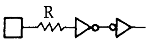
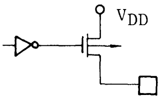
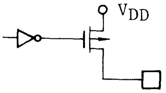
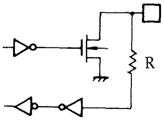
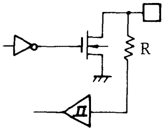
Fig. 2.6.1 Input/Output Timing

(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

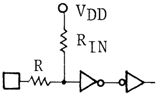
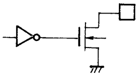
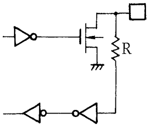
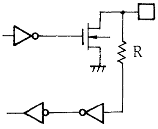
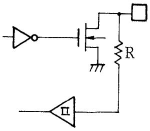
Input/Output Circuit Code (IOCODE) FA					
* **	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
Input/Output Circuit Format	 R = 1kΩ (TYP.)		 R = 1kΩ(TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained.	o Sink open drain output. o Output latch is initialized 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 (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.)

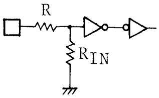
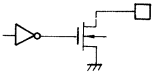
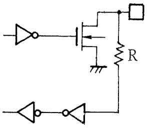
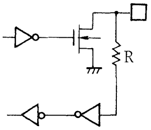
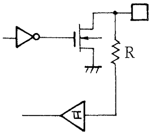
*: Port **: Circuit

Input/Output Circuit Code (IOCODE) FB

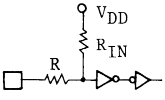
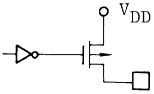
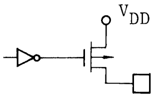
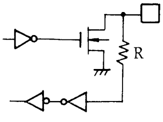
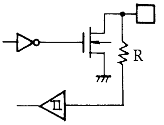
** \ *	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
Input/Output Circuit Format	 RIN=70kΩ(TYP.) R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-up resistor is contained.	o Sink open drain output o Output latch is initialized to the high level.	o 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 (IOCODE) FC

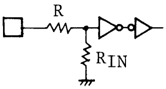
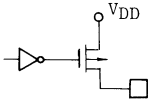
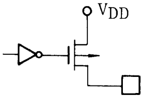
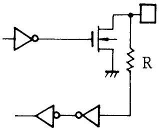
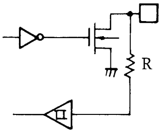
** \ *	Input (K_0)	Output (P_1,P_2)	I/O (R_4,R_5,R_6)	I/O (R_7)	I/O (R_8,R_9)
Input/Output Circuit Format	 RIN=70kΩ (TYP.) R = 1kΩ (TYP.)		 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o Pull-down resistor is contained.	o Sink open drain output o Output latch is initialized to the high level.	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 (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 (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.)

2.7 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 R9 port. These pins can be used as the pins of the R9 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 (≈ 70kΩ TYP.).

3. 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 3.0.1 Classification of Instructions.

3.1 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

3.2 Description of instructions (*): Note 1 (**): Exec.cycle (⁂): Hexadecimal

Item Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Move Instruction	LD A, @HL	0000 1100	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	0011 1100 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	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	0010 1000 $x_7x_6x_5x_4$ $x_3x_2x_1x_0$	2 8 x_Hx_L	(LR)←M[x'],(HR)←M[x'+1] $x'=x_7x_6x_5x_4x_3x_2 00$	-	-	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	0100 $k_3k_2k_1k_0$	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.				

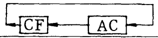
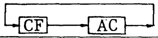
Items / Class	Assembler Mnemonic	Object Code: Binary	Object Code: (***)	Function / Functional Description	Flag(*) CF	Flag(*) ZF	Flag(*) SF	(**)
Move Instruction	LD H, #k	1 1 0 0 $k_3k_2k_1k_0$	C k	(HR)←k Loads the immediate data k of the instruction field in the H register. Serves as the clear instruction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

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	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
	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
	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_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		Function	Flag(*)			(**)
		Binary	(**)	Functional Description	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.	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		Function	Flag(*)			(**)
		Binary	(**)	Functional Description	CF	ZF	SF	
Arithmetic Instruction	INC @HL	0000 1010	0A	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	0000 1001	09	(AC)←(AC)-1 Decrements the contents of the accumulator.	-	Z	$\overline{B}$	1
	DEC L	0001 1001	19	(LR)←(LR)-1 Decrements the contents of the L register.	-	Z	$\overline{B}$	1
	DEC @HL	0000 1011	0B	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	0001 0101	15	(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	0001 0111	17	(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	0011 1000 0000 $k_3k_2k_1k_0$	38 0k	(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	0011 1000 1100 $k_3k_2k_1k_0$	38 Ck	(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		Function	Flag(*)			(**)
		Binary	($\overset{*}{**}$)	Functional Description	CF	ZF	SF	
Arithmetic Instruction	ADD L,#k	0 0 1 1 1 0 0 0	3 8	(LR)←(LR)+k	-	Z	$\overline{C}$	2
		1 0 0 0 $k_3k_2k_1k_0$	8 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.				
	ADD @HL,#k	0 0 1 1 1 0 0 0	3 8	M[(H·L)]←M[(H·L)]+k	-	Z	$\overline{C}$	2
		0 1 0 0 $k_3k_2k_1k_0$	4 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.				
	ADD y,#k	0 0 1 0 1 1 1 1	2 F	M[y]←M[y]+k	-	Z	$\overline{C}$	2
		$k_3k_2k_1k_0$ $y_3y_2y_1y_0$	k y	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.				
	SUBRC A,@HL	0 0 0 1 0 1 0 0	1 4	(AC)←M[(H·L)]-(AC)-($\overline{CF}$)	$\overline{B}$	Z	$\overline{B}$	1
				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.				
	SUBR A,#k	0 0 1 1 1 0 0 0	3 8	(AC)←k-(AC)	-	Z	$\overline{B}$	2
		0 0 0 1 $k_3k_2k_1k_0$	1 k	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.				

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
Arithmetic Instruction	SUBR @HL,#k	0011 1000 0101 $k_3k_2k_1k_0$	38 5k	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	0000 0101	05	 (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	0000 0111	07	 (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	0001 1110	1E	(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	0011 1000 0011 $k_3k_2k_1k_0$	38 3k	(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	0011 1000 0111 $k_3k_2k_1k_0$	38 7k	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		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
Logical Instruction	OR A, @HL	0 0 0 1 1 1 0 1	1 D	$(AC) \leftarrow (AC) \vee M[(H \cdot L)]$ 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.	-	Z	$\overline{Z}$	1
	OR A, #k	0 0 1 1 1 0 0 0 0 0 1 0 $k_3k_2k_1k_0$	3 8 2 k	$(AC) \leftarrow (AC) \vee k$ 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.	-	Z	$\overline{Z}$	2
	OR @HL, #k	0 0 1 1 1 0 0 0 0 1 1 0 $k_3k_2k_1k_0$	3 8 6 k	$M[(H \cdot L)] \leftarrow M[(H.L)] \vee k$ 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.	-	Z	$\overline{Z}$	2
	XOR A, @HL	0 0 0 1 1 1 1 1	1 F	$(AC) \leftarrow (AC) \forall M[(H \cdot L)]$ 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.	-	Z	$\overline{Z}$	1
Bit Manipulation Instruction	TEST CF	0 0 0 0 0 1 1 0	0 6	$(SF) \leftarrow (\overline{CF})$, $(CF) \leftarrow 0$ Places the inverse contents of the carry flag in the status flag, and then resets the carry flag to "0".	0	-	*	1
	TEST A, b	0 1 0 1 1 1 b_1b_0	5 C+b	$(SF) \leftarrow \overline{(AC)<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the accumulator, in the status flag.	-	-	*	1
	TEST @HL, b	0 1 0 1 1 0 b_1b_0	5 8+b	$(SF) \leftarrow \overline{M[(H \cdot L)]<b>}$ Places the inverse contents of the bit, which is specified by the b of the instruction field, of the data memory specified by the H and L registers, in the status flag.	-	-	*	1

Items / Class	Assembler Mnemonic	Object Code		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Bit Manipulation Instruction	TEST y, b	0 0 1 1 1 0 0 1	3 9	(SF)←$\overline{\text{M[y]<b>}}$	-	-	*	2
		1 0 b_1b_0 $y_3y_2y_1y_0$	8+b y	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	3 B	(SF)←$\overline{\text{P[p]<b>}}$	-	-	*	2
		1 0 b_1b_0 $p_3p_2p_1p_0$	8+b P	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)←$\overline{\text{P[(LR)<3:2>+4]<(LR)<1:0>>}}$	-	-	*	2
				Places the inverse contents of the bit, which is specified by the lower-order two bits of the L register, of the ports R4 - R7 (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	3 9	(SF)←M[y]<b>	-	-	*	2
		1 1 b_1b_0 $y_3y_2y_1y_0$	C+b y	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		Function	Flag(*)			(**)
		Binary	(⁂)	Functional Description	CF	ZF	SF	
Bit Manipulation Instruction	TESTP %P,b	0 0 1 1 1 0 1 1 1 1 $b_1 b_0$ $p_3 p_2 p_1 p_0$	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 $b_1 b_0$	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 $b_1 b_0$ $y_3 y_2 y_1 y_0$	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 $b_1 b_0$ $p_3 p_2 p_1 p_0$	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 R_4 - R_7 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 $b_1 b_0$	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: 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		Function	Flag(*)			(**)
		Binary	(***)	Functional Description	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 FEO - 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(*)			(**)
		Binary	(***)	Functional Description	CF	ZF	SF	
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, $a=(PC)\langle 11\cdot 6\rangle\cdot d$	-	-	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, $0 \le a \le 2{,}047$	-	-	-	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(*)			(**)
		Binaty	(⁂)	Functional Description	CF	ZF	SF	
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	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.

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
Topr	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} ⩾ 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} ⩾ 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 ∿ P_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 (VSS = 0V, V_{DD} = 5V±10%, Topr = -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,$\overline{RESET}$,$\overline{HOLD}$,TEST) [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]	VDD(V_{DDH})=5.5V, fC=4MHz, V_{IN}=5.3/0.2V (all valid) CL=50pF, C_{XIN}=C_{XOUT}=10pF	-	5	T.B.D	mA
I_{DDH}	Holding Supply Current [Note 3]		-	0.5	T.B.D	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.

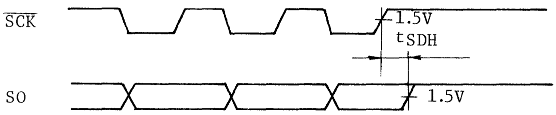
(Note 3) When K0 port has a built-in input resistor, current value is that at time of open. Further, voltage level at R port is valid.

A.C. CHARACTERISTICS (VSS = 0V, VDD = 5V±10%, Topr = -30 ∿ 70°C)

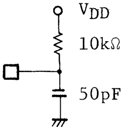
SYMBOL	Parameter	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	Instruction Cycle Time		3.8	-	40	μs
t_{SDH}	Shift Data Hold Time	(Note 1)	0.5tcy-300	-	-	ns

A.C. TIMING CHART

- Serial Port (Completion of Transmission)

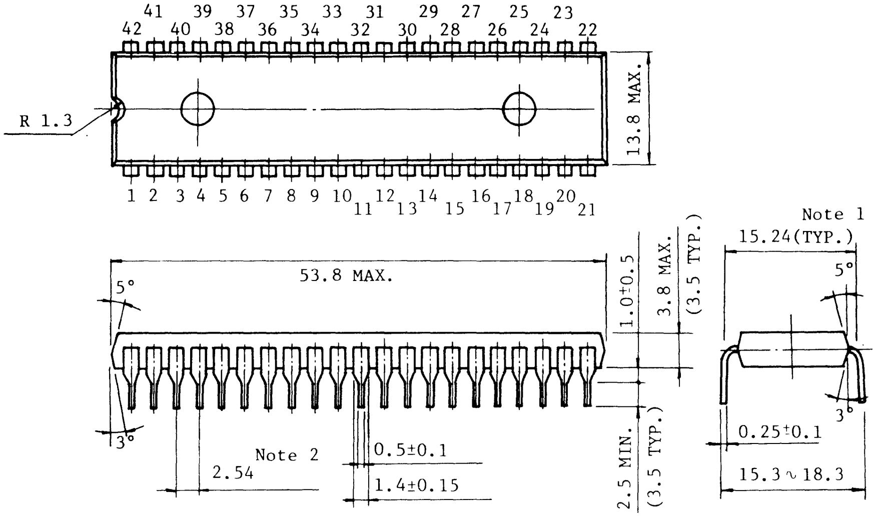


(Note 1) $\overline{SCK}$, SO terminal external circuit



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.

Specification of program tape and input/output circuit format

Engineering Samples(ES) of the TMP47C40P, TMP47C20P, TMP47C41P and TMP47C21P will be made if you specify the program data and input/output circuit format by use of a paper tape.

The paper tape format is equivalent to the Hex. format of Intel Co. (Format I).

The program data should be specified within the address space corresponding to the built-in ROM capacity; the addresses 000 - 7FF denote the address range in the TMP47C20P and TMP47C21P. Accordingly, if the PLA data conversion table (addresses FE0 - FFF) is used, the table data must be assigned as the data located in addresses 7E0 - 7FF.

1. Specification of input/output circuit format

The paper tape of Format I starts recording the program data after record mark ":", but the input/output circuit code should be specified just before the first record mark.

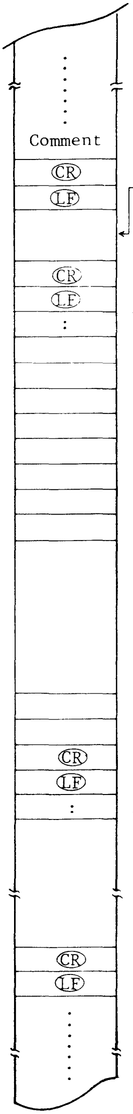
The "IŌCŌDE XX" format is used to define the input/output circuit code. XX denotes the proper input/output circuit code (two alphabets).

(Note) If the input/output circuit code is not specified, "IŌCŌDE FA" OR "IŌCŌDE HA" are employed. It should be noted that if the specified format is different from the standard one, and if the specified input/output circuit code is illegal, such specifications may be considered to have not been made.

(Example of tape list)

```
TŌSHIBA MICRŌCŌMPUTER TLCS-47
IŌCŌDE FA
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67E31F5D8ABA6DF292F113F5C1
:100020004FF1F
                        :
                        :
                        :
:1007E000B53D42E0EC32546025B7308CDD52063D1D
:1007F000B4BE9E9E345B6138060B20BC372BF60BD6
:00000001FF
```

2. Program tape format (Format I)



.... Leaders More than 50 characters of "NULL"

.... Comment (record mark ":" and the input/output circuit format are not included.) } Not specially required.

Specification of the input/output circuit format.

... Specification of the input/output circuit format IOCODE XX (XX denotes the input/output circuit code) (It does not matter if there is "NULL" from immediately after CR, LF to immediately before record marking.) } It is considered to be no specification, if not specified.

.... Record mark (start of record)

.... Record length (the number of program data within one record is represented by two-digit hexadecimal notation. "00" denotes the end of data.)

.... Load address (the first address of the record is represented by four-digit hexadecimal notation.)

... Record type ("00" ..... Normal record
"01" ..... End of file record)

.... Data (1-byte data is represented by two-digit hexadecimal notation.)

... Checksum (two-digit hexadecimal notation is used to represent the data of the lower-order eight bits in the value which the data covering from the record length up to immediately before the checksum is subtracted from the initial value 0.)

...... Record mark (the same procedure is repeated hereinafter.)

... Trailers More than 50 characters of "NULL"

PRELIMINARY

LIST OF INSTRUCTIONS

Item / Classification	Assembler Mnemonic	Object Code				Function	*1 Flags			Execution Cycle	
		Binary		Hexadecimal							
		1st Byte	2nd Byte	1st Byte	2nd Byte		CF	ZF	SF		
Move	LD A ,@HL	00 00 11 00		0 C		(AC)←M[(H•L)]	—	Z	1	1	
	LD A , x	00 11 11 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 C	x_Hx_L	(AC)←M[x]	—	Z	1	2	
	LD HL, x	00 10 10 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 8	x_Hx_L	$(LR)\leftarrow M[x'],(HR)\leftarrow M[x'+1],x'=x_7x_6x_5x_4x_3x_2 00$	—	—	1	2	
	LD A ,#k	01 00 $k_3k_2k_1k_0$		4 k		(AC)←k	—	Z	1	1	
	LD H ,#k	11 00 $k_3k_2k_1k_0$		C k		(HR)←k	—	—	1	1	
	LD L ,#k	11 10 $k_3k_2k_1k_0$		E k		(LR)←k	—	—	1	1	
	LDL A ,@DC	00 11 00 11		3 3		$(AC)\leftarrow ROM_L[(DC)]$	—	Z	1	2	
	LDH A ,@DC+	00 11 00 10		3 2		$(AC)\leftarrow ROM_H[(DC)],(DC)\leftarrow(DC)+1$	—	Z	1	2	*2
	ST A ,@HL	00 00 11 11		0 F		M[(H•L)]←(AC)	—	—	1	1	
	ST A ,@HL+	00 01 10 10		1 A		M[(H•L)]←(AC),(LR)←(LR)+1	—	Z	$\overline{C}$	1	*3
	ST A ,@HL−	00 01 10 11		1 B		M[(H•L)]←(AC),(LR)←(LR)−1	—	Z	$\overline{B}$	1	*3
	ST A , x	00 11 11 11	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 F	x_Hx_L	M[x]←(AC)	—	—	1	2	
	ST #k ,@HL+	11 11 $k_3k_2k_1k_0$		F k		M[(H•L)]←k,(LR)←(LR)+1	—	Z	$\overline{C}$	1	*3
	ST #k, y	00 10 11 01	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 D	k y	M[y]←k	—	—	1	2	
	MOV H , A	00 01 00 00		1 0		(AC)←(HR)	—	Z	1	1	
	MOV L , A	00 01 00 01		1 1		(AC)←(LR)	—	Z	1	1	
	XCH A , H	00 11 00 00		3 0		(HR)⇄(AC)	—	Z	1	2	*2
	XCH A , L	00 11 00 01		3 1		(LR)⇄(AC)	—	Z	1	2	*2
	XCH A ,EIR	00 01 00 11		1 3		(EIR)⇄(AC)	—	—	1	1	
	XCH A ,@HL	00 00 11 01		0 D		M[(H•L)]⇄(AC)	—	Z	1	1	*2
	XCH A , x	00 11 11 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 D	x_Hx_L	M[x]⇄(AC)	—	Z	1	2	*2
	XCH HL, x	00 10 10 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 9	x_Hx_L	$M[x']\rightleftarrows(LR),M[x'+1]\rightleftarrows(HR),x'=x_7x_6x_5x_4x_3x_2 00$	—	—	1	2	
Compare	CMPR A ,@HL	00 01 01 10		1 6		null←M[(H•L)]−(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR A , x	00 11 11 10	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 E	x_Hx_L	null←M[x]−(AC)	$\overline{B}$	Z	$\overline{Z}$	2	
	CMPR A ,#k	11 01 $k_3k_2k_1k_0$		D k		null←k−(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR H ,#k	00 11 10 00	11 01 $k_3k_2k_1k_0$	3 8	D k	null←k−(HR)	—	Z	$\overline{B}$	2	
	CMPR L ,#k	00 11 10 00	10 01 $k_3k_2k_1k_0$	3 8	9 k	null←k−(LR)	—	Z	$\overline{B}$	2	
	CMPR y ,#k	00 10 11 10	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 E	k y	null←k−M[y]	$\overline{B}$	Z	$\overline{Z}$	2	
Arithmetic	INC A	00 00 10 00		0 8		(AC)←(AC)+1	—	Z	$\overline{C}$	1	
	INC L	00 01 10 00		1 8		(LR)←(LR)+1	—	Z	$\overline{C}$	1	
	INC @HL	00 00 10 10		0 A		M[(H•L)]←M[(H•L)]+1	—	Z	$\overline{C}$	1	
	DEC A	00 00 10 01		0 9		(AC)←(AC)−1	—	Z	$\overline{B}$	1	
	DEC L	00 01 10 01		1 9		(LR)←(LR)−1	—	Z	$\overline{B}$	.1	
	DEC @HL	00 00 10 11		0 B		M[(H•L)]←M[(H•L)]−1	—	Z	$\overline{B}$	1	
	ADDC A ,@HL	00 01 01 01		1 5		(AC)←(AC)+M[(H•L)]+(CF)	C	Z	$\overline{C}$	1	
	ADD A ,@HL	00 01 01 11		1 7		(AC)←(AC)+M[(H•L)]	—	Z	$\overline{C}$	1	
	ADD A ,#k	00 11 10 00	00 00 $k_3k_2k_1k_0$	3 8	0 k	(AC)←(AC)+k	—	Z	$\overline{C}$	2	
	ADD H ,#k	00 11 10 00	11 00 $k_3k_2k_1k_0$	3 8	C k	(HR)←(HR)+k	—	Z	$\overline{C}$	2	
	ADD L ,#k	00 11 10 00	10 00 $k_3k_2k_1k_0$	3 8	8 k	(LR)←(LR)+k	—	Z	$\overline{C}$	2	
	ADD @HL,#k	00 11 10 00	01 00 $k_3k_2k_1k_0$	3 8	4 k	M[(H•L)]←M[(H•L)]+k	—	Z	$\overline{C}$	2	
	ADD y ,#k	00 10 11 11	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 F	k y	M[y]←M[y]+k	—	Z	$\overline{C}$	2	
	SUBRC A ,@HL	00 01 01 00		1 4		$(AC)\leftarrow M[(H\bullet L)]-(AC)-\overline{(CF)}$	$\overline{B}$	Z	$\overline{B}$	1	
	SUBR A ,#k	00 11 10 00	00 01 $k_3k_2k_1k_0$	3 8	1 k	(AC)←k−(AC)	—	Z	$\overline{B}$	2	
	SUBR @HL,#k	00 11 10 00	01 01 $k_3k_2k_1k_0$	3 8	5 k	M[(H•L)]←k−M[(H•L)]	—	Z	$\overline{B}$	2	
Logical	ROLC A	00 00 01 01		0 5		$\boxed{CF}\leftarrow\boxed{AC}$ (rotate left by 1 bit)	C	Z	$\overline{C}$	1	*4
	RORC A	00 00 01 11		0 7		$\boxed{AC}\rightarrow\boxed{CF}$ (rotate right by 1 bit)	C	Z	$\overline{C}$	1	*4
	AND A ,@HL	00 01 11 10		1 E		(AC)←(AC)∧M[(H•L)]	—	Z	$\overline{Z}$	1	
	AND A ,#k	00 11 10 00	00 11 $k_3k_2k_1k_0$	3 8	3 k	(AC)←(AC)∧k	—	Z	$\overline{Z}$	2	
	AND @HL,#k	00 11 10 00	01 11 $k_3k_2k_1k_0$	3 8	7 k	M[(H•L)]←M[(H•L)]∧k	—	Z	$\overline{Z}$	2	
	OR A ,@HL	00 01 11 01		1 D		(AC)←(AC)∨M[(H•L)]	—	Z	$\overline{Z}$	1	
	OR A ,#k	00 11 10 00	00 10 $k_3k_2k_1k_0$	3 8	2 k	(AC)←(AC)∨k	—	Z	$\overline{Z}$	2	
	OR @HL,#k	00 11 10 00	01 10 $k_3k_2k_1k_0$	3 8	6 k	M[(H•L)]←M[(H•L)]∨k	—	Z	$\overline{Z}$	2	
	XOR A ,@HL	00 01 11 11		1 F		(AC)←(AC)∀M[(H•L)]	—	Z	$\overline{Z}$	1	

(continued)

Item / Classification	Assembler Mnemonic	Object Code: Binary 1st Byte	Binary 2nd Byte	Hexadecimal 1st Byte	Hexadecimal 2nd Byte	Function	Flags *1 CF	ZF	SF	Execution Cycle	
Bit Manipulation	TEST CF	00 00 01 10		0 6		(SF)←$\overline{\text{(CF)}}$,(CF)←0	0	—	*	1	
	TEST A , b	01 01 11 b_1b_0		5 C+b		(SF)←$\overline{\text{(AC)<b>}}$	—	—	*	1	
	TEST @HL, b	01 01 10 b_1b_0		5 8+b		(SF)←$\overline{\text{M[(H•L)]<b>}}$	—	—	*	1	
	TEST y , b	00 11 10 01	10 $b_1b_0y_3y_2y_1y_0$	3 9	8+b y	(SF)←$\overline{\text{M[y]<b>}}$	—	—	*	2	
	TEST %p , b	00 11 10 11	10 $b_1b_0p_3p_2p_1p_0$	3 B	8+b p	(SF)←$\overline{\text{P[p]<b>}}$	—	—	*	2	
	TEST @L	00 11 01 11		3 7		(SF)←$\overline{\text{P[(LR)<3:2>+4]<(LR)<1:0>>}}$	—	—	*	2	
	TESTP CF	00 00 01 00		0 4		(SF)←(CF),(CF)←1	1	—	*	1	
	TESTP ZF	00 00 11 10		0 E		(SF)←(ZF)	—	—	*	1	
	TESTP GF	00 00 00 01		0 1		(SF)←(GF)	—	—	*	1	
	TESTP y , b	00 11 10 01	11 $b_1b_0y_3y_2y_1y_0$	3 9	C+b y	(SF)←M[y]<b>	—	—	*	2	
	TESTP %p. b	00 11 10 11	11 $b_1b_0p_3p_2p_1p_0$	3 B	C+b p	(SF)←P[p]<b>	—	—	*	2	
	SET GF	00 00 00 11		0 3		(GF)←1	—	—	1	1	
	SET @HL, b	01 01 00 b_1b_0		5 b		M[(H•L)]<b>←1	—	—	1	1	
	SET y , b	00 11 10 01	00 $b_1b_0y_3y_2y_1y_0$	3 9	b y	M[y]<b>←1	—	—	1	2	
	SET %p , b	00 11 10 11	00 $b_1b_0p_3p_2p_1p_0$	3 B	b p	P[p]<b>←1	—	—	1	2	
	SET @L	00 11 01 00		3 4		P[(LR)<3:2>+4]<(LR)<1:0>>←1	—	—	1	2	
	CLR GF	00 00 00 10		0 2		(GF)←0	—	—	1	1	
	CLR @HL, b	01 01 01 b_1b_0		5 4+b		M[(H•L)]<b>←0	—	—	1	1	
	CLR y , b	00 11 10 01	01 $b_1b_0y_3y_2y_1y_0$	3 9	4+b y	M[y]<b>←0	—	—	1	2	
	CLR %p , b	00 11 10 11	01 $b_1b_0p_3p_2p_1p_0$	3 B	4+b p	P[p]<b>←0	—	—	1	2	
	CLR @L	00 11 01 01		3 5		P[(LR)<3:2>+4]<(LR)<1:0>>←0	—	—	1	2	
	CLR IL , r	00 11 01 10	11 $r_5r_4r_3r_2r_1r_0$	3 6	C+r_Hr_L	(INTL)<5:0>←(INTL)<5:0>∧r<5:0>	—	—	1	2	
	EICLR IL , r	00 11 01 10	01 $r_5r_4r_3r_2r_1r_0$	3 6	4+r_Hr_L	(EIF)←1,(INTL)<5:0>←(INTL)<5:0>∧r<5:0>	—	—	1	2	
	DICLR IL , r	00 11 01 10	10 $r_5r_4r_3r_2r_1r_0$	3 6	8+r_Hr_L	(EIF)←0,(INTL)<5:0>←(INTL)<5:0>∧r<5:0>	—	—	1	2	
Input/Output	IN %p , A	00 11 10 10	00 10 $p_3p_2p_1p_0$	3 A	2 p	(AC)←P[p]	—	Z	$\overline{Z}$	2	
	IN %p ,@HL	00 11 10 10	01 10 $p_3p_2p_1p_0$	3 A	6 p	M[(H•L)]←P[p]	—	—	$\overline{Z}$	2	
	OUT A ,%p	00 11 10 10	10 $\overline{p_4}$0 $p_3p_2p_1p_0$	3 A	8+2$\overline{p_4}$ p	p[p]←(AC) , p = p_4•$p_3p_2p_1p_0$	—	—	1	2	
	OUT @HL,%p	00 11 10 10	11 $\overline{p_4}$0 $p_3p_2p_1p_0$	3 A	C+2$\overline{p_4}$ p	p[p]←M[(H•L)], p = p_4•$p_3p_2p_1p_0$	—	—	1	2	
	OUT #k ,%p	00 10 11 00	$k_3k_2k_1k_0p_3p_2p_1p_0$	2 C	k p	p[p]←k	—	—	1	2	
	OUTB @HL	00 01 00 10		1 2		P[2]•P[1]←ROM[F•(E+(CF))•M[(H•L)]]	—	—	1	2	
Branch•Subroutine	BS a	01 10 $a_{11}a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	6 a_H	a_M a_L	If SF=1 then(PC)←a else null.	—	—	1	2	
	BSS a	10 $d_5d_4d_3d_2d_1d_0$		8+d_Hd_L		If SF=1 then(PC)←a else null,a=(PC)<11:6>•d	—	—	1	1	*5
	CALL a	00 10 0 $a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	2 a_H	a_M a_L	STACK[(SPW)]←(PC),(SPW)←(SPW)−1 , (PC)←a, $0 \leqq a \leqq 2{,}047$	—	—	—	2	*5
	CALLS a	01 11 $n_3n_2n_1n_0$		7 n		STACK[(SPW)]←(PC),(SPW)←(SPW)−1, (PC)←a,a=8n+6(n≠0),134(n=0)	—	—	—	2	*5
	RET	00 10 10 10		2 A		(SPW)←(SPW)+1 , (PC)←STACK[(SPW)]	—	—	—	2	
	RETI	00 10 10 11		2 B		(SPW)←(SPW)+1 , (FLAG•PC)←STACK[(SPW)], (EIF)←1	*	*	*	2	
Other	NOP	00 00 00 00		0 0		no operation	—	—	—	1	

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","$\overline{C}$","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.

Operation Code Map

higher \ lover	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	TESTP GF	CLR GF	SET GF	TESTP GF	ROLC A	TEST CF	RORC A	INC A	DEC A	INC @HL	DEC @HL	LD A,@HL	XCH A,@HL	TESTP ZF	ST A,@HL
1	MOV H ,A	MOV L, A	OUTB @HL	XCH A,EIR	SUBRC A,@HL	ADDC A,@HL	CMPR A,@HL	ADD A,@HL	INC L	DEC L	ST A,@HL+	ST A,@HL-		OR A,@HL	AND A,@HL	XOR A,@HL
2	CALL a								LD HL,x	XCH HL,x	RET	RETI	OUT #k,%p	ST #k,y	CMPR y,#k	ADD y,#k
3	XCH A,H	XCH A, L	LDH A,@DC+	LDL A,@DC	SET @L	CLR @L	(OP36)	TEST @L	(OP38)	(OP39)	(OP3A)	(OP3B)	LD A,x	XCH A,x	CMPR A,x	ST A,x
4	LD A,#k															
5	SET @HL, b				CLR @HL,b				TEST @HL,b				TEST A,b			
6	BS a															
7	CALLS a															
8	BSS a															
9																
A																
B																
C	LD H , #k															
D	CMPR A , #k															
E	LD L , #k															
F	ST #k , @HL+															

(continued)

1st byte code	OP36	OP38	OP39	OP3A	OP3B
2nd byte code(higher)	36	38	39	3A	3B
0		ADD A, #k	SET y, b		SET %p , b
1		SUBR A, #k			
2		OR A, #k		IN %p , A	
3		AND A, #k			
4	EICIR IL, r	ADD @HL, #k	CLR y, b		CLR %p , b
5		SUBR @HL, #k			
6		OR @HL, #k		IN %p , @HL	
7		AND @HL, #k			
8	DICIR IL, r	ADD L, #k	TEST y, b	OUT A , %p	TEST %p , b
9		CMPR L, #k			
A				OUT A , %p	
B					
C	CLR IL, r	ADD H, #k	TESTP y, b	OUT @HL, %p	TESTP %p , b
D		CMPR H, #k			
E				OUT @HL, %p	
F					

(caution)

1. Blank codes are reserved.
2. ◸ is indicated 2-byte instruction.

TOSHIBA MOS TYPE DIGITAL INTEGRATED CIRCUIT
TMP47C22F
SILICON MONOLITHIC SILICON GATE CMOS

PRELIMINARY

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER (TLCS-47C) TMP47C22F

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 TMP47C22F is a chip countaining LCD driver for the TLCS-47C. The memory capacity consists of ROM 2,048 x 8 bits and RAM 192 x 4 bits. The TMP4700C (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

Toshiba 東芝

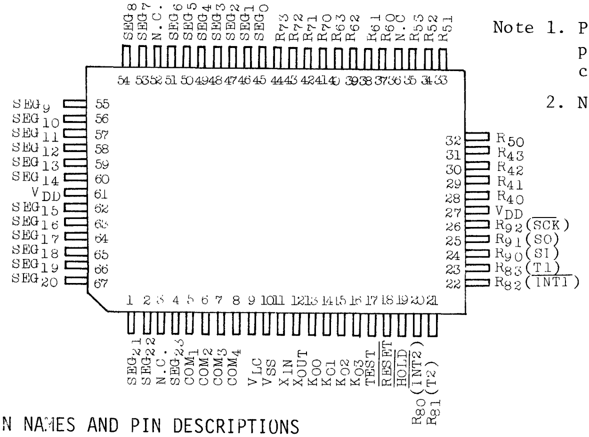
INTEGRATED CIRCUIT

TECHNICAL DATA

TMP47C22F

PRELIMINARY

PIN CONNECTIONS (Top View)



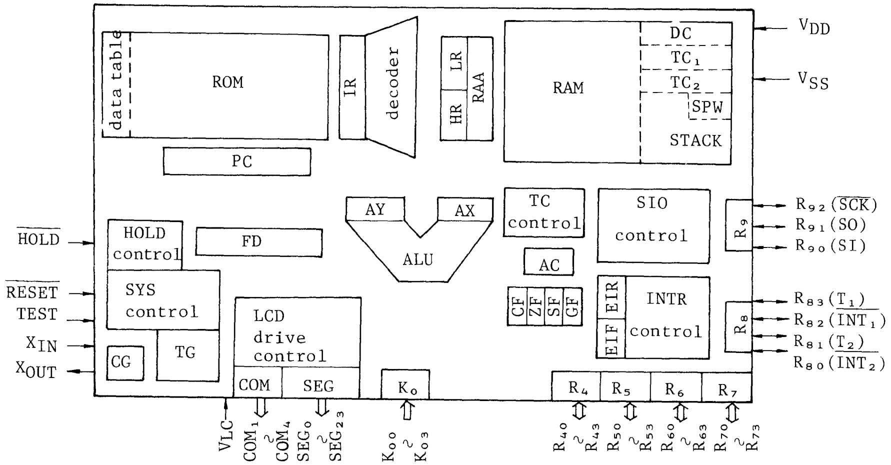
Note 1. Pin 27 is connected to pin 61 through external circuit.

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} ($\overline{T1}$)	1	I/O	I/O port or timer/counter input
R_{82} (INT1)	1	I/O	I/O port or interrupt input
R_{81} ($\overline{T2}$)	1	I/O	I/O port or timer/counter input
R_{80} (INT2)	1	I/O	I/O port or interrupt input
R_{92} ($\overline{SCK}$)	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
VSS	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)
TC₁, TC₂	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

The configuration will be explained with priority given primarily to the LCD drive circuit.

1.1 Program Counter (PC), 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 x 8 bit (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. 1.3.1.

1.2 H Register (HR), L Register (LR)

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

1.3 RAM Address Buffer Register (RAA), 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. 1.3.1.

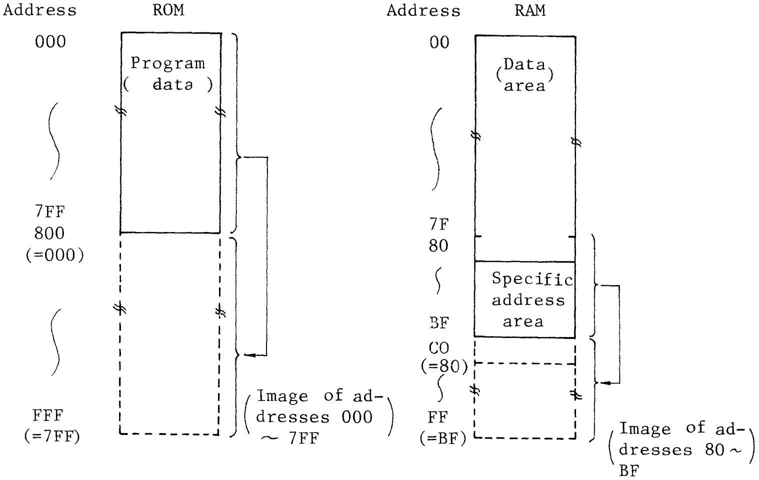


Fig. 1.3.1 ROM/RAM Capacity and Addresses

1.4 ALU, Accumulator (AC), Flag (FLAG)

The ALU is a circuit used for various 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.

Flag is a 4-bit register used to store the conditions 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 condition immediately before the interrupt is accepted.

1.5 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.

Since the TMP47C22F has no built-in outputs ports P_1 and P_2, (OUTB @HL) instruction cannot be used; therefore 12 kinds of instructions become available as effective 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).

Port ad-dress	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/ -	0					0	
01	IP01/OP01	-							
02	IP02/OP02	-							
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	-							
0C	IP0C/OP0C	-							
0D	IP0D/OP0D	-							
0E	IP0E/OP0E	Status input/ -	0					0	
0F	IP0F/OP0F	(*) / (**)	0	0	0				
10	/OP10	/Hold control		0					
11	/OP11	/ -							
12	/OP12	/ -							
13	/OP13	/ -							
14	/OP14	/ -							
15	/OP15	/ -							
16	/OP16	/ -							
17	/OP17	/ -							
18	/OP18	/ -							
19	/OP19	/ (a)		0					
1A	/OP1A	/ (b) (1)		0					
1B	/OP1B	/ " (2)		0					
1C	/OP1C	/ (c)		0					
1D	/OP1D	/ (d)		0					
1E	/OP1E	/ -							
1F	/OP1F	/ (e)		0					

(*) Serial buffer register (Reception)
(**) Serial buffer register (Transmission)

(a) Control with timer interrupt of divider
(b) LCD drive control (1), (2)
(c) Timer/counter 1 control
(d) Timer counter 2 control
(e) Serial port control

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 user's program.

Table 1.5.1 Port Address Allocation and Input/Output Instructions

(1) K$_0$ (K$_{03}$ ∿ K$_{00}$) Port

This is a 4-bit port used for input.

(2) R$_4$ (R$_{43}$∿R$_{40}$), R$_5$ (R$_{53}$∿R$_{50}$), R$_6$ (R$_{63}$∿R$_{60}$), R$_7$ (R$_{73}$∿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.

(3) R$_8$ (R$_{83}$∿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 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.

(4) R$_9$ (R$_{92}$∿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 R$_9$ port is also 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.

1.6 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 is generated to the CPU.
The interrupt latch is set when the edge of the input signal is detected.

1.7 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.

1.8 Timer/counter (TC_1, TC_2)

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

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 maximum frequency applied to the timer/counter is as follows.
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 disignated internal pulse rate is applied to the external input pin.

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/64 Hz. When 2 channels are operated simultaneously, timer/counter 1 is fc/64 Hz and timer/counter 2 is fc/80 Hz.

(b) When LCD display is enabled

Frequency applied at time of a single channel operation is fc/128 Hz. When 2 channels are operated simultaneously, both timer/counter 1 and timer/counter 2 are fc/144 Hz.

1.9 Serial Port (SIO)

A 4-bit serial port with a buffer is provided to transfer the serial data from/to the external circuitry. According to the contents of the command register, either one of transmit mode, receive (trailing edge shift) mode or receive (leading edge shift) mode can be selected.

1.10 Hold Control Circuit (HOLDC)

The hold function is the function to hold the status immediately before the system operation is stopped at low power consumption making the most of the features of CMOS.

The hold function is controlled by $\overline{\text{HOLD}}$ terminal input and command register.

1.11 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. 1.11.1.

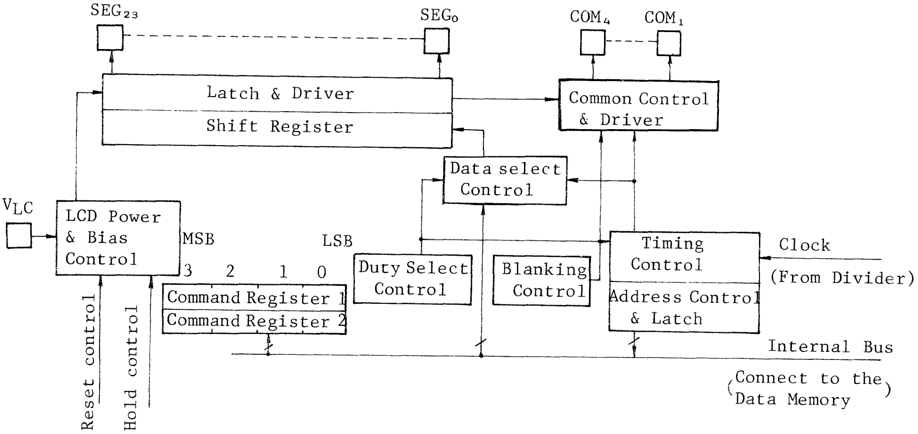
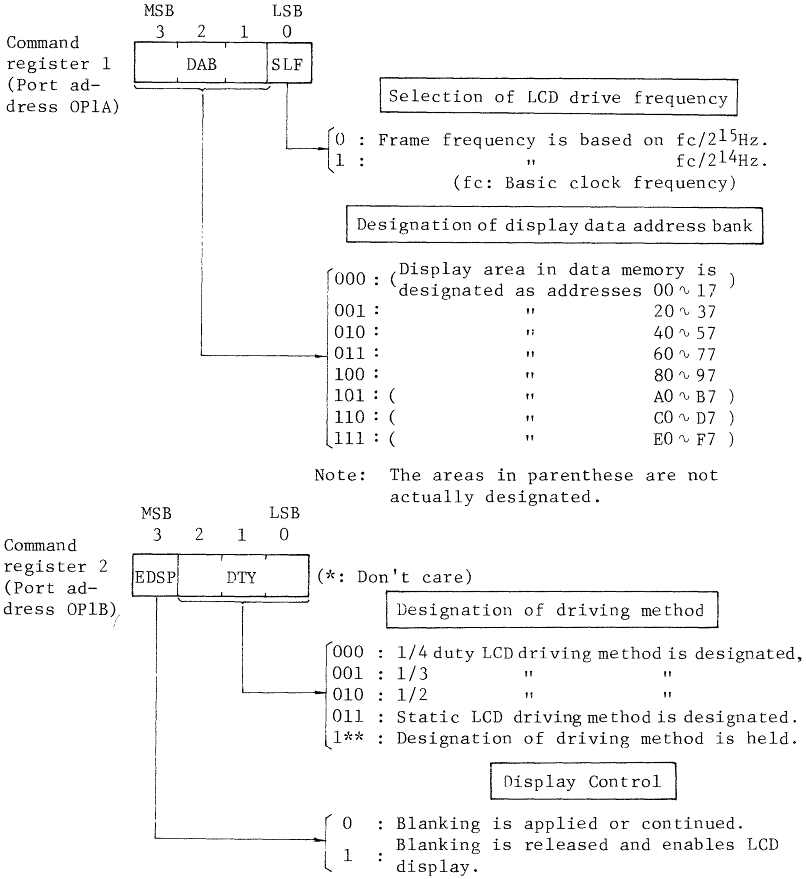


Fig. 1.11.1 LCD Drive Circuit

(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.


MSB
LSB
3
2
1
0
Command register 1 (Port address OP1A)
DAB
SLF
MSB
LSB
3
2
1
0
Command register 2 (Port address OP1B)
EDSP
DTY

Selection of LCD drive frequency

0 : Frame frequency is based on $fc/2^{15}$Hz.
1 : " $fc/2^{14}$Hz.
(fc: Basic clock frequency)

Designation of display data address bank

000 : (Display area in data memory is designated as addresses 00 ∿ 17)
001 : " 20 ∿ 37
010 : " 40 ∿ 57
011 : " 60 ∿ 77
100 : " 80 ∿ 97
101 : (" A0 ∿ B7)
110 : (" C0 ∿ D7)
111 : (" E0 ∿ F7)

Note: The areas in parenthese are not actually designated.

(*: Don't care)

Designation of driving method

000 : 1/4 duty LCD driving method is designated,
001 : 1/3 " "
010 : 1/2 " "
011 : Static LCD driving method is designated.
1** : Designation of driving method is held.

Display Control

0 : Blanking is applied or continued.
1 : Blanking is released and enables LCD display.

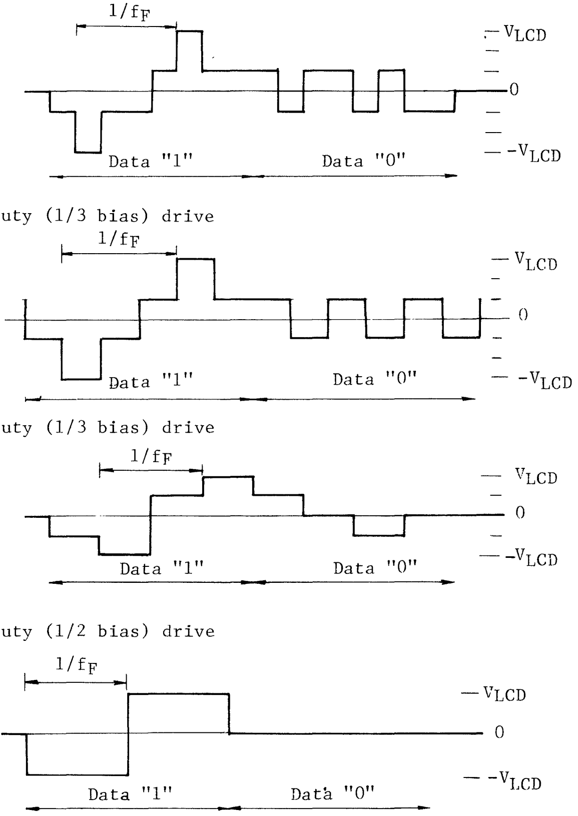
Fig. 1.11.2 Control of Drive Circuit

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. 1.11.3.



(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. 1.11.3 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	$\frac{f_c}{2^{14}}$	$\frac{f_c}{2^{15}}$	$\frac{4}{3}\cdot\frac{f_c}{2^{15}}$	$\frac{4}{2}\cdot\frac{f_c}{2^{15}}$	$\frac{f_c}{2^{15}}$
	(f_c=2 MHz)	≑ 122	≑ 163	≑ 244	≑ 122

(f_c: Basic clock frequency)

Table 1.11.1 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.

(3) Display operation

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. 1.11.4.

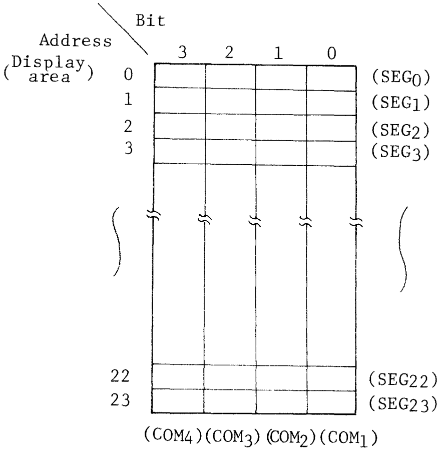


Fig. 1.11.4 LCD Diaply Data Area (Data Memory)

Where, each bit of the display data memory shows data of segment (dot) equivalent to SEGi, COMj ($0 \leqq i \leqq 23$, $1 \leqq j \leqq 4$), 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. 1.11.2.).

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 Figl 1.11.6 - Fig. 1.11.9.

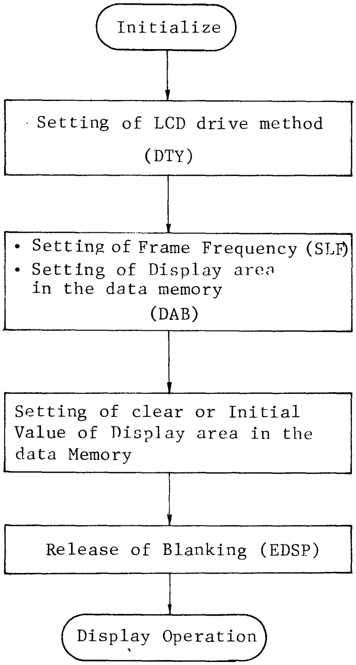
When EDSP is set at "1", the LCD display is enabled and the LCD display is made according to data stored in the display area of the data memory.

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. 1.11.5.

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.


Initialize
Setting of LCD drive method (DTY)
• Setting of Frame Frequency (SLF)
• Setting of Display area in the data memory (DAB)
Setting of clear or Initial Value of Display area in the data Memory
Release of Blanking (EDSP)
Display Operation

(* : Setting command)

: Set "00**" at port address OP1B. [Note 1] (Thereafter, use "1/0 1**" only.

: Set "****" at port address OP1A. [Note 2]

: Set "11**" at port address OP1B.

[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. 1.11.5 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 1.11.2. For the connecting method of COM terminals and SEG terminals, the example shown in Fig. 1.11.6 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	0.	1 1 0 1	1 1 1 1	5	5	1 0 1 1	0 1 0 1
1	1	0 0 0 0	0 1 1 0	6	6	1 1 1 1	0 1 0 1
2	2	1 1 1 0	0 0 1 1	7	7	0 0 0 1	0 1 1 1
3	3	1 0 1 0	0 1 1 1	8	8	1 1 1 1	0 1 1 1
4	4	0 0 1 1	0 1 1 0	9	9	1 0 1 1	0 1 1 1

Table 1.11.2 Examples of Display Data (1/4 Duty LCD)

Further, examples of display data when a numeral display similar to Table 1.11.2 is made by using the 1/3 duty LCD are shown in Table 1.11.3. For the connecting method of COM terminals and SEG terminals, the example shown in Fig. 1.11.7 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

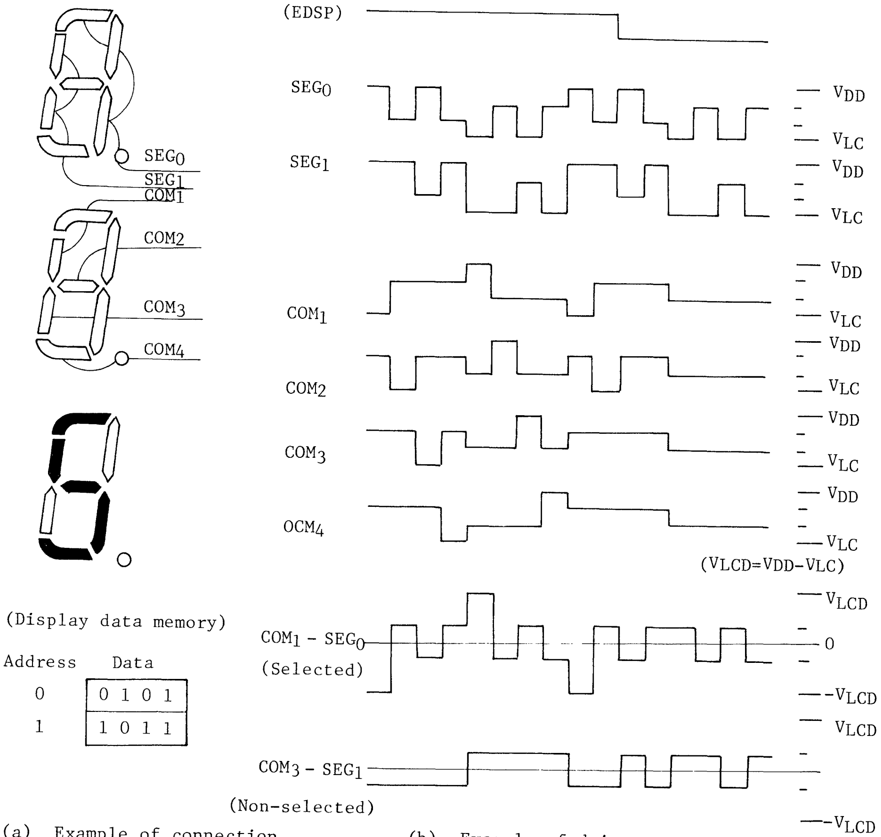
(* : don't care)

Table 1.11.3 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)

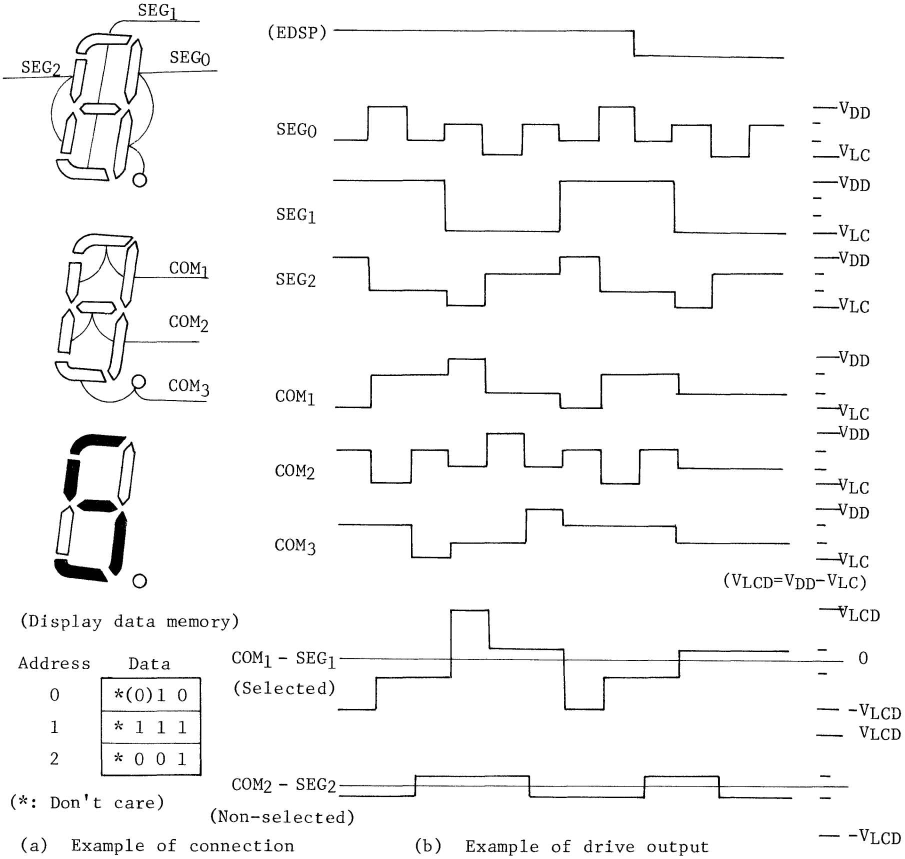
Address	Data
0	0 1 0 1
1	1 0 1 1

(a) Example of connection and display character

(b) Example of drive output

Fig. 1.11.6 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 0 1

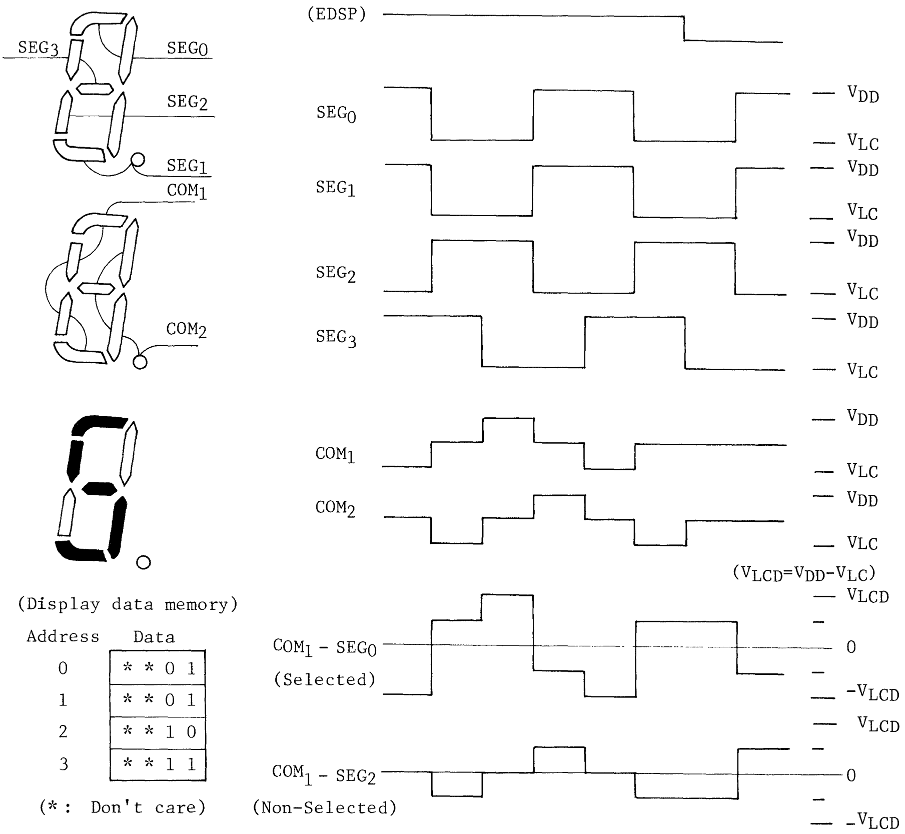
(*: Don't care)

(a) Example of connection and display character

(b) Example of drive output

Fig. 1.11.7 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

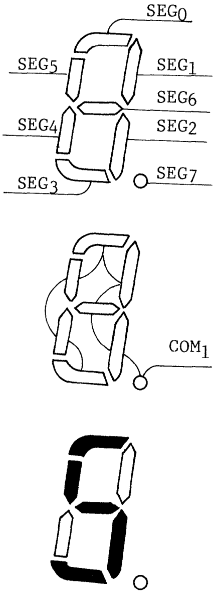
(* : Don't care)

(a) Example of connection and display character

(b) Example of drive output

Fig. 1.11.8 Example of 1/2 Duty LCD Display Output

Static Drive

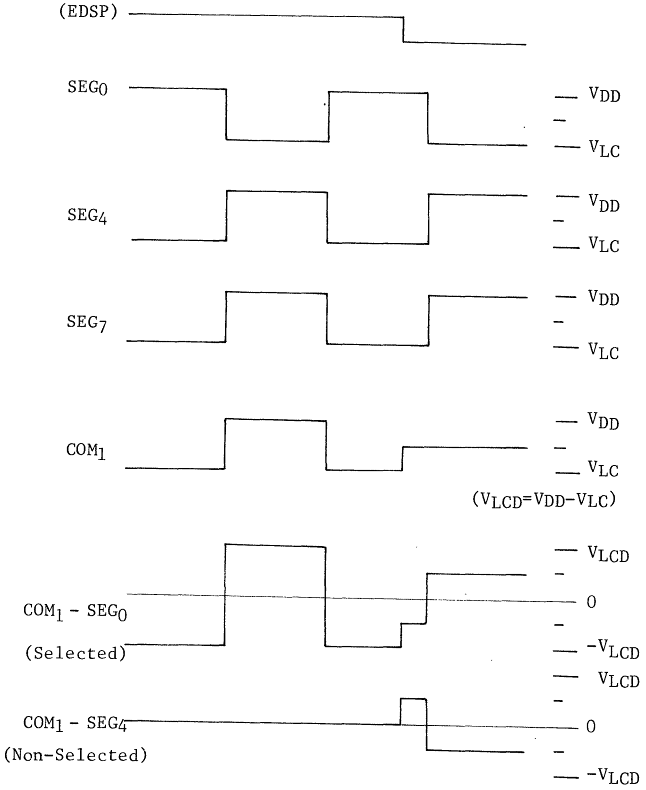


(Display data memory)

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. 1.11.9 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.

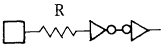
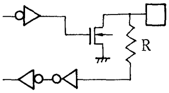
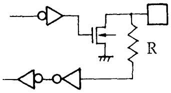
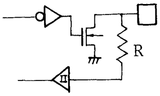
Two pins ($\overline{INT_1}$, $\overline{INT_2}$) are provided for the external interrupt input. Since these pins are common pins with R_8 port, they can be used as I/O pins respectively, if not used as the interrupt input pins. The interrupt latch is set by the falling edge of the external interrupt inputs.

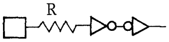
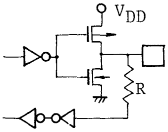
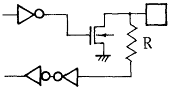
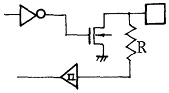
The TEST terminal is used at time of the shippint test. When a user's system is to be operated, low level voltage should be positively applied. Further, the TEST terminal of the TMP 47C22F has no built-in pull-down resistor.

2.3 Input/Output port

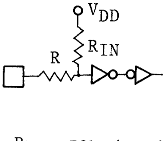
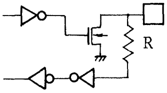
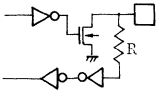
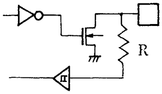
Input/output Circuit Format

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. "IOCODE GD" is employed if not specified.

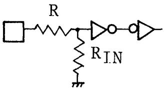
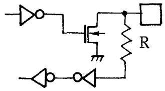
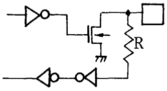
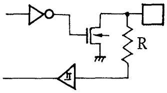
Input/Output Circuit Code (IOCODE) GA				
Port Circuit	Input (K_0)	I/O (R_4, R_5, R_6)	I/O (R_7)	I/O (R_8, R_9)
I/O equivalent circuit	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)	 R = 1kΩ (TYP.)
Remark	o No resistor is contained	o Sink open drain output o Output latch is initialized to the high level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

Input/Output Circuit Code (IOCODE) GD				
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 = 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 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

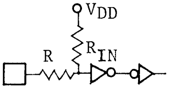
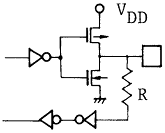
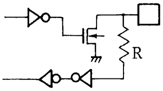
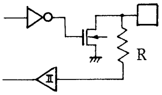
Input/Output Circuit Code (I$\overline{\text{O}}$C$\overline{\text{O}}$DE) GB

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

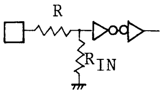
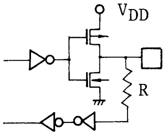
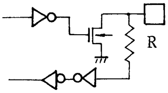
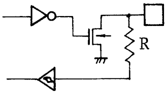
Input/Output Circuit Code (I$\overline{\text{O}}$C$\overline{\text{O}}$DE) GC

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 (IOCODE) 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 (IOCODE) 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

3. 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.

The TMP47C22F is software compatible with other versions of the TLCS-47 series; however, since it has no built-in output ports P_1 and P_2, (OUTB @HL) instruction cannot be used, so that 89 instructions become available as effective instructions.

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

(a) Classification by byte/cycle

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

(Note): Including ROM data referring instructions

(b) Classification by function

Table 3.0.1 Classification of instructions

Item Classification	Assembler Mnemonic	Object Code				Function	Flags *1			Execution Cycle	
		Binary		Hexadecimal							
		1st Byte	2nd Byte	1st Byte	2nd Byte		CF	ZF	SF		
Move	LD A ,@HL	00 00 11 00		0 C		(AC)←M[(H•L)]	–	Z	1	1	
	LD A , x	00 11 11 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 C	x_Hx_L	(AC)←M[x]	–	Z	1	2	
	LD HL, x	00 10 10 00	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 8	x_Hx_L	(LR)←M[x'],(HR)←M[x'+1], x'=$x_7x_6x_5x_4x_3x_2$00	–	–	1	2	
	LD A ,#k	01 00 $k_3k_2k_1k_0$		4 k		(AC)←k	–	Z	1	1	
	LD H ,#k	11 00 $k_3k_2k_1k_0$		C k		(HR)←k	–	–	1	1	
	LD L ,#k	11 10 $k_3k_2k_1k_0$		E k		(LR)←k	–	–	1	1	
	LDL A ,@DC	00 11 00 11		3 3		(AC)←ROM_L[(DC)]	–	Z	1	2	
	LDH A ,@DC+	00 11 00 10		3 2		(AC)←ROM_H[(DC)],(DC)←(DC)+1	–	Z	1	2	*2
	ST A ,@HL	00 00 11 11		0 F		M[(H•L)]←(AC)	–	–	1	1	
	ST A ,@HL+	00 01 10 10		1 A		M[(H•L)]←(AC),(LR)←(LR)+1	–	Z	$\overline{C}$	1	*3
	ST A ,@HL–	00 01 10 11		1 B		M[(H•L)]←(AC),(LR)←(LR)–1	–	Z	$\overline{B}$	1	*3
	ST A , x	00 11 11 11	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 F	x_Hx_L	M[x]←(AC)	–	–	1	2	
	ST #k,@HL+	11 11 $k_3k_2k_1k_0$		F k		M[(H•L)]←k,(LR)←(LR)+1	–	Z	$\overline{C}$	1	*3
	ST #k, y	00 10 11 01	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 D	k y	M[y]←k	–	–	1	2	
	MOV H , A	00 01 00 00		1 0		(AC)←(HR)	–	Z	1	1	
	MOV L , A	00 01 00 01		1 1		(AC)←(LR)	–	Z	1	1	
	XCH A , H	00 11 00 00		3 0		(HR)⇄(AC)	–	Z	1	2	*2
	XCH A , L	00 11 00 01		3 1		(LR)⇄(AC)	–	Z	1	2	*2
	XCH A ,EIR	00 01 00 11		1 3		(EIR)⇄(AC)	–	–	1	1	
	XCH A ,@HL	00 00 11 01		0 D		M[(H•L)]⇄(AC)	–	Z	1	1	*2
	XCH A , x	00 11 11 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 D	x_Hx_L	M[x]⇄(AC)	–	Z	1	2	*2
	XCH HL, x	00 10 10 01	$x_7x_6x_5x_4x_3x_2x_1x_0$	2 9	x_Hx_L	M[x']⇄(LR),M[x'+1]⇄(HR), x'=$x_7x_6x_5x_4x_3x_2$00	–	–	1	2	
Compare	CMPR A ,@HL	00 01 01 10		1 6		null←M[(H•L)]–(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR A , x	00 11 11 10	$x_7x_6x_5x_4x_3x_2x_1x_0$	3 E	x_Hx_L	null←M[x]–(AC)	$\overline{B}$	Z	$\overline{Z}$	2	
	CMPR A ,#k	11 01 $k_3k_2k_1k_0$		D k		null←k–(AC)	$\overline{B}$	Z	$\overline{Z}$	1	
	CMPR H ,#k	00 11 10 00	11 01 $k_3k_2k_1k_0$	3 8	D k	null←k–(HR)	–	Z	$\overline{B}$	2	
	CMPR L ,#k	00 11 10 00	10 01 $k_3k_2k_1k_0$	3 8	9 k	null←k–(LR)	–	Z	$\overline{B}$	2	
	CMPR y ,#k	00 10 11 10	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 E	k y	null←k–M[y]	$\overline{B}$	Z	$\overline{Z}$	2	
Arithmetic	INC A	00 00 10 00		0 8		(AC)←(AC)+1	–	Z	$\overline{C}$	1	
	INC L	00 01 10 00		1 8		(LR)←(LR)+1	–	Z	$\overline{C}$	1	
	INC @HL	00 00 10 10		0 A		M[(H•L)]←M[(H•L)]+1	–	Z	$\overline{C}$	1	
	DEC A	00 00 10 01		0 9		(AC)←(AC)–1	–	Z	$\overline{B}$	1	
	DEC L	00 01 10 01		1 9		(LR)←(LR)–1	–	Z	$\overline{B}$	1	
	DEC @HL	00 00 10 11		0 B		M[(H•L)]←M[(H•L)]–1	–	Z	$\overline{B}$	1	
	ADDC A ,@HL	00 01 01 01		1 5		(AC)←(AC)+M[(H•L)]+(CF)	C	Z	$\overline{C}$	1	
	ADD A ,@HL	00 01 01 11		1 7		(AC)←(AC)+M[(H•L)]	–	Z	$\overline{C}$	1	
	ADD A ,#k	00 11 10 00	00 00 $k_3k_2k_1k_0$	3 8	0 k	(AC)←(AC)+k	–	Z	$\overline{C}$	2	
	ADD H ,#k	00 11 10 00	11 00 $k_3k_2k_1k_0$	3 8	C k	(HR)←(HR)+k	–	Z	$\overline{C}$	2	
	ADD L ,#k	00 11 10 00	10 00 $k_3k_2k_1k_0$	3 8	8 k	(LR)←(LR)+k	–	Z	$\overline{C}$	2	
	ADD @HL,#k	00 11 10 00	01 00 $k_3k_2k_1k_0$	3 8	4 k	M[(H•L)]←M[(H•L)]+k	–	Z	$\overline{C}$	2	
	ADD y ,#k	00 10 11 11	$k_3k_2k_1k_0y_3y_2y_1y_0$	2 F	k y	M[y]←M[y]+k	–	Z	$\overline{C}$	2	
	SUBRC A ,@HL	00 01 01 00		1 4		(AC)←M[(H•L)]–(AC)–$\overline{(CF)}$	$\overline{B}$	Z	$\overline{B}$	1	
	SUBR A ,#k	00 11 10 00	00 01 $k_3k_2k_1k_0$	3 8	1 k	(AC)←k–(AC)	–	Z	$\overline{B}$	2	
	SUBR @HL,#k	00 11 10 00	01 01 $k_3k_2k_1k_0$	3 8	5 k	M[(H•L)]←k–M[(H•L)]	–	Z	$\overline{B}$	2	
Logical	ROLC A	00 00 01 01		0 5		$\boxed{CF}\leftarrow\boxed{AC}\leftarrow$ (rotate left by 1 bit)	C	Z	$\overline{C}$	1	*4
	RORC A	00 00 01 11		0 7		$\leftarrow\boxed{AC}\rightarrow\boxed{CF}$ (rotate right by 1 bit)	C	Z	$\overline{C}$	1	*4
	AND A ,@HL	00 01 11 10		1 E		(AC)←(AC)∧M[(H•L)]	–	Z	$\overline{Z}$	1	
	AND A ,#k	00 11 10 00	00 11 $k_3k_2k_1k_0$	3 8	3 k	(AC)←(AC)∧k	–	Z	$\overline{Z}$	2	
	AND @HL,#k	00 11 10 00	01 11 $k_3k_2k_1k_0$	3 8	7 k	M[(H•L)]←M[(H•L)]∧k	–	Z	$\overline{Z}$	2	
	OR A ,@HL	00 01 11 01		1 D		(AC)←(AC)∨M[(H•L)]	–	Z	$\overline{Z}$	1	
	OR A ,#k	00 11 10 00	00 10 $k_3k_2k_1k_0$	3 8	2 k	(AC)←(AC)∨k	–	Z	$\overline{Z}$	2	
	OR @HL,#k	00 11 10 00	01 10 $k_3k_2k_1k_0$	3 8	6 k	M[(H•L)]←M[(H•L)]∨k	–	Z	$\overline{Z}$	2	
	XOR A ,@HL	00 01 11 11		1 F		(AC)←(AC)∀M[(H•L)]	–	Z	$\overline{Z}$	1	

(continued)

Item Classification	Assembler Mnemonic	Object Code				Function	Flags *1			Execution Cycle	
		Binary		Hexadecimal							
		1st Byte	2nd Byte	1st Byte	2nd Byte		CF	ZF	SF		
Bit Manipulation	TEST CF	00 00 01 10		0 6		(SF)←$\overline{(CF)}$,(CF)←0	0	–	*	1	
	TEST A , b	01 01 11 b_1b_0		5 C+b		(SF)← $\overline{(AC)\langle b\rangle}$	–	–	*	1	
	TEST @HL, b	01 01 10 b_1b_0		5 8+b		(SF)← $\overline{M[(H•L)]\langle b\rangle}$	–	–	*	1	
	TEST y , b	00 11 10 01	10 $b_1b_0y_3y_2y_1y_0$	3 9	8+b y	(SF)← $\overline{M[y]\langle b\rangle}$	–	–	*	2	
	TEST %p , b	00 11 10 11	10 $b_1b_0p_3p_2p_1p_0$	3 B	8+b p	(SF)← $\overline{P[p]\langle b\rangle}$	–	–	*	2	
	TEST @L	00 11 01 11		3 7		(SF)← $\overline{P[(LR)\langle 3:2\rangle+4]\langle (LR)\langle 1:0\rangle\rangle}$	–	–	*	2	
	TESTP CF	00 00 01 00		0 4		(SF)←(CF),(CF)←1	1	–	*	1	
	TESTP ZF	00 00 11 10		0 E		(SF)←(ZF)	–	–	*	1	
	TESTP GF	00 00 00 01		0 1		(SF)←(GF)	–	–	*	1	
	TESTP y , b	00 11 10 01	11 $b_1b_0y_3y_2y_1y_0$	3 9	C+b y	(SF)←M[y]⟨b⟩	–	–	*	2	
	TESTP %p, b	00 11 10 11	11 $b_1b_0p_3p_2p_1p_0$	3 B	C+b p	(SF)←P[p]⟨b⟩	–	–	*	2	
	SET GF	00 00 00 11		0 3		(GF)←1	–	–	1	1	
	SET @HL, b	01 01 00 b_1b_0		5 b		M[(H•L)]⟨b⟩←1	–	–	1	1	
	SET y , b	00 11 10 01	00 $b_1b_0y_3y_2y_1y_0$	3 9	b y	M[y]⟨b⟩←1	–	–	1	2	
	SET %p , b	00 11 10 11	00 $b_1b_0p_3p_2p_1p_0$	3 B	b p	P[p]⟨b⟩←1	–	–	1	2	
	SET @L	00 11 01 00		3 4		P[(LR)⟨3:2⟩+4]⟨(LR)⟨1:0⟩⟩←1	–	–	1	2	
	CLR GF	00 00 00 10		0 2		(GF)←0	–	–	1	1	
	CLR @HL, b	01 01 01 b_1b_0		5 4+b		M[(H•L)]⟨b⟩←0	–	–	1	1	
	CLR y , b	00 11 10 01	01 $b_1b_0y_3y_2y_1y_0$	3 9	4+b y	M[y]⟨b⟩←0	–	–	1	2	
	CLR %p , b	00 11 10 11	01 $b_1b_0p_3p_2p_1p_0$	3 B	4+b p	P[p]⟨b⟩←0	–	–	1	2	
	CLR @L	00 11 01 01		3 5		P[(LR)⟨3:2⟩+4]⟨(LR)⟨1:0⟩⟩←0	–	–	1	2	
	CLR IL , r	00 11 01 10	11 $r_5r_4r_3r_2r_1r_0$	3 6	C+r_Hr_L	(INTL)⟨5:0⟩←(INTL)⟨5:0⟩∧r⟨5:0⟩	–	–	1	2	
	EICLR IL , r	00 11 01 10	01 $r_5r_4r_3r_2r_1r_0$	3 6	4+r_Hr_L	(EIF)←1,(INTL)⟨5:0⟩←(INTL)⟨5:0⟩∧r⟨5:0⟩	–	–	1	2	
	DICLR IL , r	00 11 01 10	10 $r_5r_4r_3r_2r_1r_0$	3 6	8+r_Hr_L	(EIF)←0,(INTL)⟨5:0⟩←(INTL)⟨5:0⟩∧r⟨5:0⟩	–	–	1	2	
Input/Output	IN %p , A	00 11 10 10	00 10 $p_3p_2p_1p_0$	3 A	2 p	(AC)←P[p]	–	Z	$\overline{Z}$	2	
	IN %p ,@HL	00 11 10 10	01 10 $p_3p_2p_1p_0$	3 A	6 p	M[(H•L)]←P[p]	–	–	$\overline{Z}$	2	
	OUT A ,%p	00 11 10 10	10 $\overline{p}_4$0 $p_3p_2p_1p_0$	3 A	8+2$\overline{p}_4$ p	p[p]←(AC) , p = p_4•p_3 p_2 p_1 p_0	–	–	1	2	
	OUT @HL,%p	00 11 10 10	11 $\overline{p}_4$0 $p_3p_2p_1p_0$	3 A	C+2$\overline{p}_4$ p	p[p]←M[(H•L)], p = p_4•p_3 p_2 p_1 p_0	–	–	1	2	
	OUT #k ,%p	00 10 11 00	$k_3k_2k_1k_0p_3p_2p_1p_0$	2 C	k p	p[p]←k	–	–	1	2	
	OUTB @HL	00 01 00 10		1 2		P[2]•P[1]←ROM[F•(E+(CF))•M[(H•L)]]	–	–	1	2	
Branch•Subroutine	BS a	01 10 $a_{11}a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	6 a_H	a_M a_L	If SF=1 then(PC)←a else null.	–	–	1	2	
	BSS a	10 $d_5d_4d_3d_2d_1d_0$		8+d_Hd_L		If SF=1 then(PC)←a else null,a=(PC)⟨11:6⟩•d	–	–	1	1	*5
	CALL a	00 10 0 $a_{10}a_9a_8$	$a_7a_6a_5a_4a_3a_2a_1a_0$	2 a_H	a_M a_L	STACK[(SPW)]←(PC),(SPW)←(SPW) −1 , (PC)←a, 0≦a≦2,047	–	–	–	2	*5
	CALLS a	01 11 $n_3n_2n_1n_0$		7 n		STACK[(SPW)]←(PC),(SPW)←(SPW)-1, (PC)←a,a=8n+6(n≠0),134(n=0)	–	–	–	2	*5
	RET	00 10 10 10		2 A		(SPW)←(SPW)+1 , (PC)←STACK[(SPW)]	–	–	–	2	
	RETI	00 10 10 11		2 B		(SPW)←(SPW)+1 , (FLAG•PC)←STACK[(SPW)], (EIF)←1	*	*	*	2	
Other	NOP	00 00 00 00		0 0		no operation	–	–	–	1	

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.

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	Poser Dissipation (Topr = 70°C)	400	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
Topr	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)	$V_{DD} \geqslant$ 4.5V	V_{DD}x0.7	V_{DD}	V
V_{IH2}	High Level Input Voltage (Schmitt circuit input)		V_{DD}x0.75	V_{DD}	
V_{IH3}	High Level Input Voltage	$V_{DD} <$ 4.5V	V_{DD}x0.9	V_{DD}	
V_{IL1}	Low Level Input Voltage (Except Schmitt circuit input)	$V_{DD} \geqslant$ 4.5V	0	V_{DD}x0.3	
V_{IL2}	Low Level Input Voltage (Schmitt circuit input)		0	V_{DD}x0.25	
V_{IL3}	Low Level Input Voltage	$V_{DD} <$ 4.5V	0	V_{DD}x0.1	
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}=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, RESET, HOLD, TEST) (NOTE2)		V_{DD}=5.5V, V_{IN}=5.5/0V	–	–	±2	μA
I_{IN2}	INPUT CURRENT (OPEN DRAIN R PORT			–	–	±2	
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	–	–	2	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 V_{OUT}=V_{DD}-0.2V/V_{LC}+0.2V	–	10	T.B.D	kΩ
R_{OC4}, R_{OC0}		HIGH·LOW LEVEL (COM)(NOTE.4,5)		–	10	T.B.D	
R_{OS3}, R_{OS1}		2/3, 1/3 LEVEL (SEG)(NOTE.4	V_{DD}=5V, V_{LCD}=3V V_{OUT}=4-0.2V/3+0.2V	–	10	T.B.D	
R_{OC3}, R_{OC1}		2/3, 1/3 LEVEL (COM)(NOTE.4)		–	10	T.B.D	
R_{OS2}		1/2 LEVEL (SEG)(NOTE.5)	V_{DD}=5V, V_{LCD}=3V V_{OUT}=3.5±0.2V		10	T.B.D	
R_{OC2}		1/2 LEVEL (COM)(NOTE.5)		–	10	T.B.D	
V_{O3}	OUTPUT VOLTAGE	2/3 LEVEL (SEG, COM)(NOTE.4)	V_{DD}=5V, V_{LCD}=3V	4-0.2	4	4+0.2	V
V_{O2}		1/2 LEVEL (SEG, COM)(NOTE.5)		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	T.B.D	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	T.B.D	μA

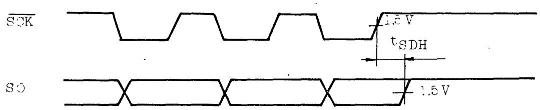
(NOTE.1) TYP.VALUES SHOW THOSE WHEN T_{OPR}=25°C, V_{DD}=5V.
(NOTE.2) WHEN THE KO PORT HAS A BUILT-IN INFUT RESISTER, CURRENT BY RESISTER IS EXCLUDED.
(NOTE.3) V_{LCD}=V_{DD}-V_{LC}.
(NOTE.4) SHOWS ON-RESISTANCE AT TIME OF LEVEL SWITCHING WHEN THE 1/4 OR 1/3 DUTY LCD IS USED.
(NOTE.5) SHOWS ON-RESISTANCE 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)

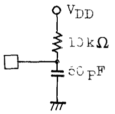
SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
tcy	INSTRUCTION CYCLE TIME		3.8	–	40	μs
t_{SDH}	SHIFT DATA HOLD TIME	(NOTE 1)	0.5tcy-300	–	–	ns

A.C. TIMING CHART

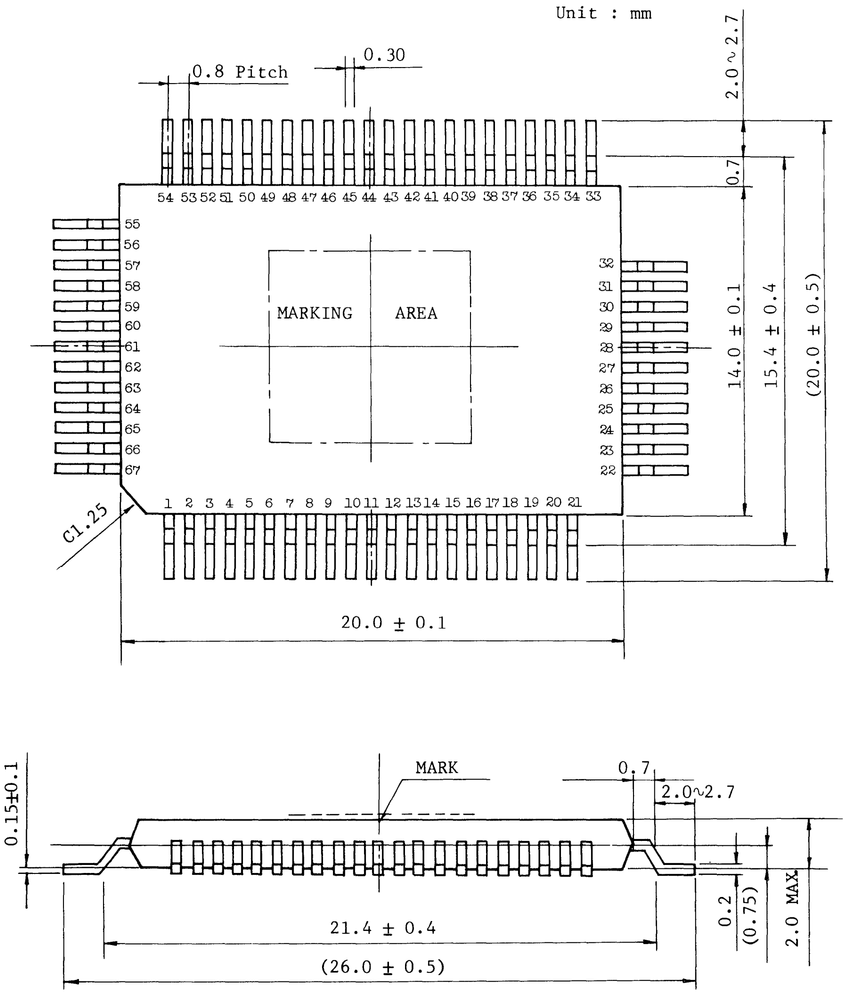
- SERIAL PORT (Completion of Transmission)



(NOTE.1) SCK, SO TERMINAL EXTERNAL CIRCUIT



EXTERNAL DIMENSIONS



Weight 1.3g (TYP.)

Specification of program tape and input/output circuit format

The TMP47C22F will be able to made enginnering samples (ES) if you specify the program data and input/output circuit format by use of a paper tape.

The paper tape format is equivalent to the Hex. format of Intel Co. (Format I).

The program data should be specified within the address space corresponding to the built-in ROM capacity; the addresses 000 - 7FF denote the address range in the TMP47C22F.

1. Specification of input/output circuit format

The paper tape of Format I starts recording the program data after record mark ":", but the input/output circuit code should be specified just before the first record mark.

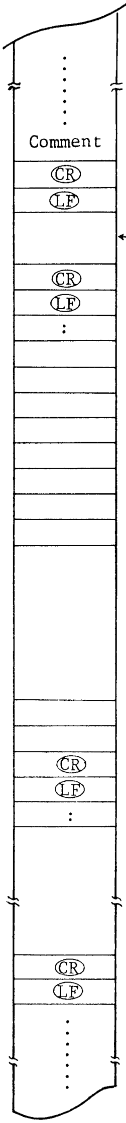
The "IŌCŌDE XX" format is used to define the input/output circuit code. XX denotes the proper input/output circuit code (two alphabets).

(Note) If the input/output circuit code is not specified, "IŌCŌDE GD" is employed. It should be noted that if the specified format is different from the standard one, and if the specified input/output circuit code is illegal, such specifications may be considered to have not been made.

(Example of tape list)

```
TŌSHIBA MICRŌCŌMPUTER TLCS-47
IŌCŌDE GD
:10000000066 5C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE76E31F5D8ABA6DF292F113F5C1
:100020004FF1F
                     :
                     :
                     :
:1007E000B53D42E0EC32546025B7308CDD52063D1D
:1007F000B4BE9E9E345B6138060B20BC372BF60BD6
:00000001FF
```

2. Program tape format (Format I)



.... Leaders More than 50 characters of "NULL"

.... Comment (record mark ":" and the input/output circuit format are not included.) } Not specially required.

Specification of the input/output circuit format.

... Specification of the input/output circuit format IOCODE XX (XX denotes the input/output circuit code) (It does not matter if there is "NULL" from immediately after CR, LF to immediately before record marking.) } It is considered to be no specification, if not specified.

.... Record mark (start of record)

.... Record length (the number of program data within one record is represented by two-digit hexadecimal notation. "00" denotes the end of data.)

.... Load address (the first address of the record is represented by four-digit hexadecimal notation.)

... Record type ("00" ..... Normal record
"01" ..... End of file record)

.... Data (1-byte data is represented by two-digit hexadecimal notation.)

... Checksum (two-digit hexadecimal notation is used to represent the data of the lower-order eight bits in the value which the data covering from the record length up to immediately before the checksum is subtracted from the initial value 0.)

...... Record mark (the same procedure is repeated hereinafter.)

... Trailers More than 50 characters of "NULL"

8BIT SINGLE CHIP MICROCOMPUTER

TLCS-84(NMOS/CMOS)

Toshiba 東芝 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.

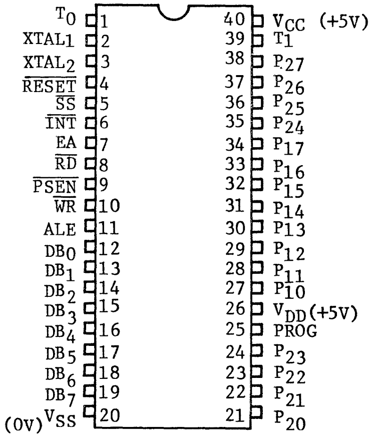
The TMP8048 is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

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.

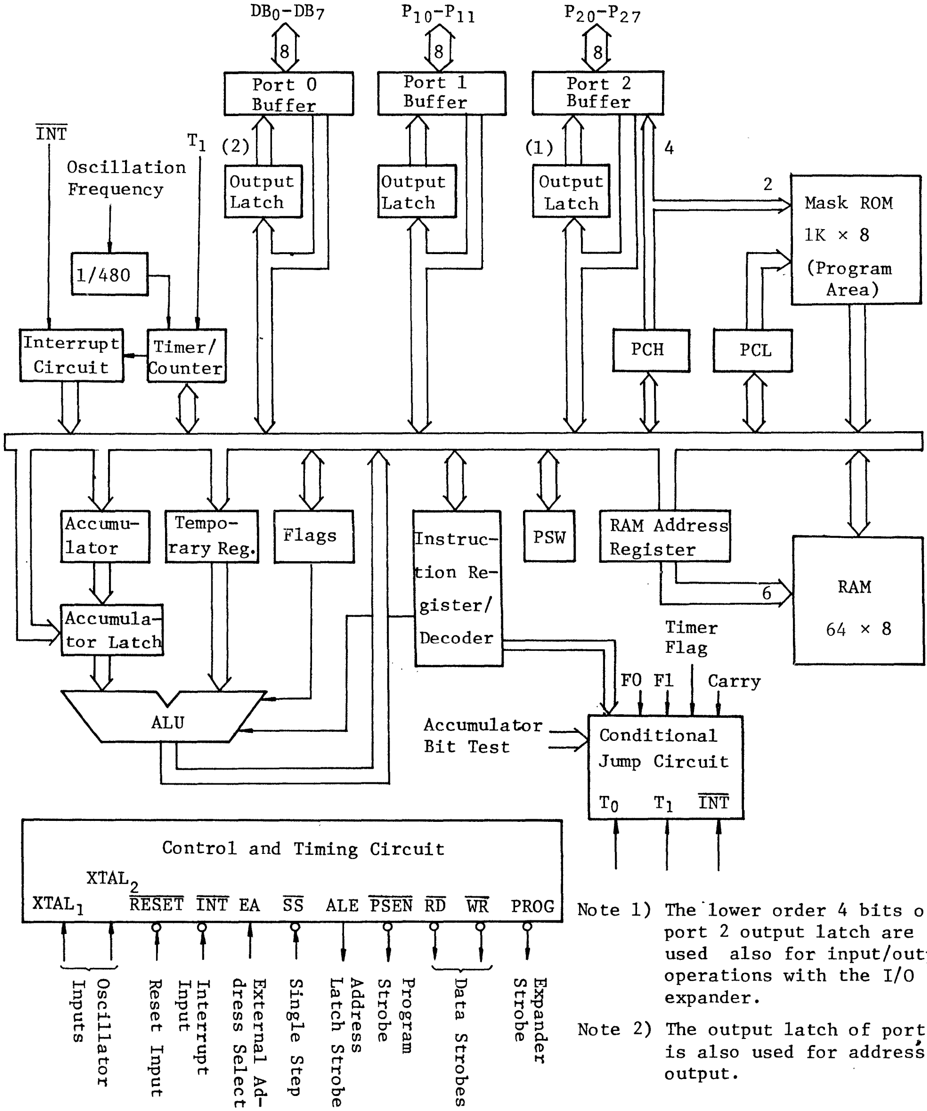
FEATURES

- 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/outpu 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≅50KΩ).

P_{20}-P_{27} (Input/Output) Port 2

8-bit quasi-bidirectional port (Internal Pullup≅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.

$\overline{\text{RESET}}$ (Input)

Active Low signal which is used to initialize the Processor. Also used during Power down.

ALE (Output)

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

$\overline{\text{PSEN}}$ (Output)

Program Store Enable. This output occurs only during a fetch to external program memory (Active Low).

$\overline{\text{SS}}$ (Input)

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

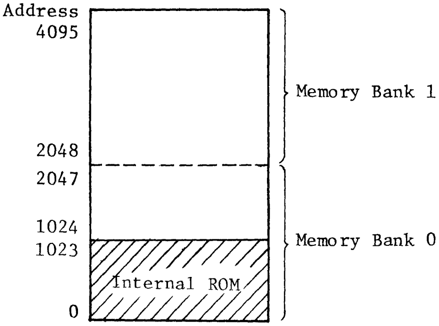
The following system functions of the TMP8048 are described in detail.

(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

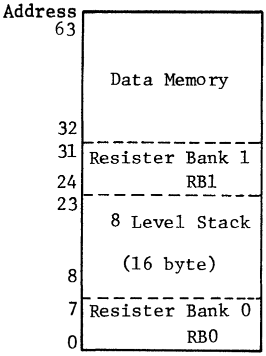
  A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

- 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.

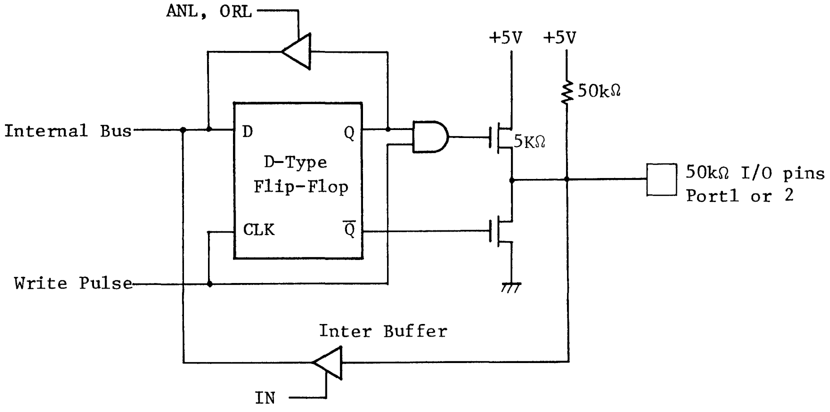


Fig.1 Input/Output Circuit of Port 1, Port 2

- 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 Tl 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.

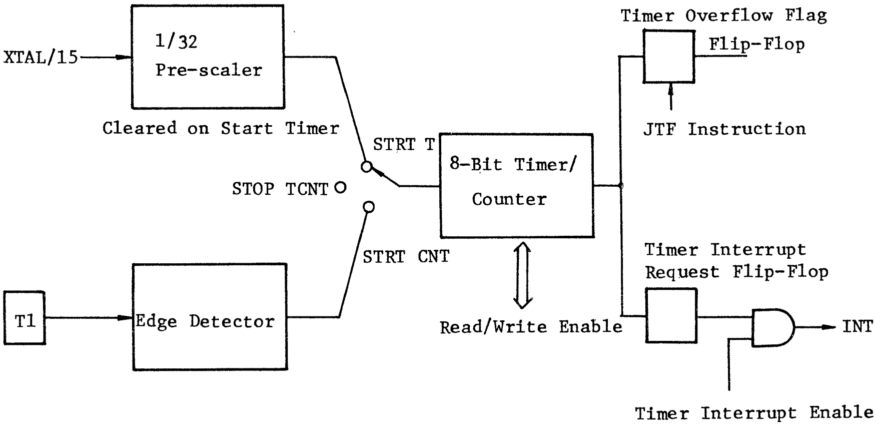


Fig.2 Concept of Timer Circuit

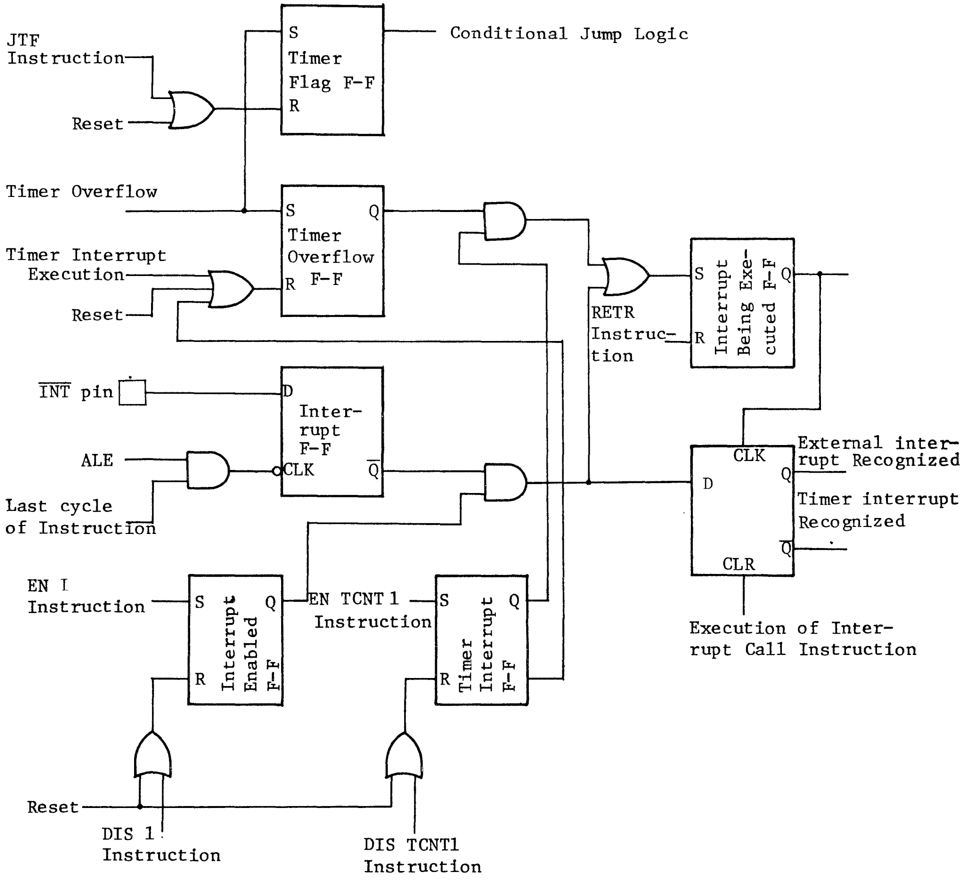


Fig.3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

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

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

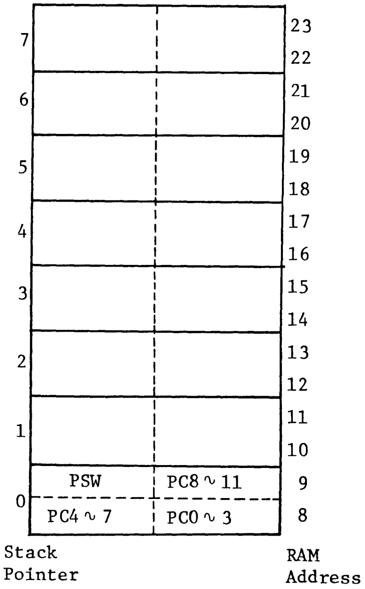
(2) Timer Interrupt caused by timer overflow

- 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 $\overline{\text{INT}}$ pin. $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$ and times overflow occur simultaneously then external request $\overline{\text{INT}}$ takes precedence.
- If an extra external interrupt is needed in addition to $\overline{\text{INT}}$ 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.

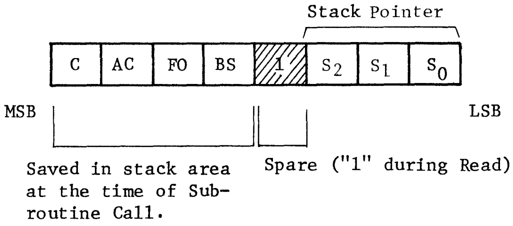


(7) Flag 0, Flag 1, (F0, F1)

- 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

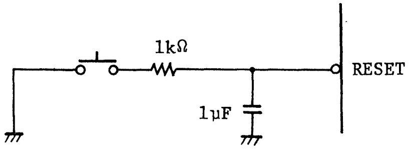
Bit 5 : Flag 0 (FO)

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.

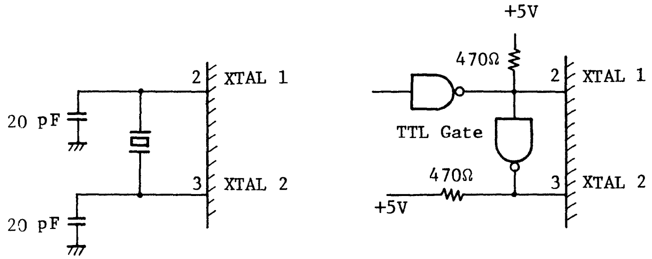
(ix) Clears Timer Flag.

(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.
- ϕ2 clock shown in Fig.4 is derived to outside by ENTO CLK instruction.
- ALE can be also used as the clock to indicate the machine cycle as well as giving the external address latch timing.

(2) External Memory Access Timing

(i) Program Memory Access

- 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.

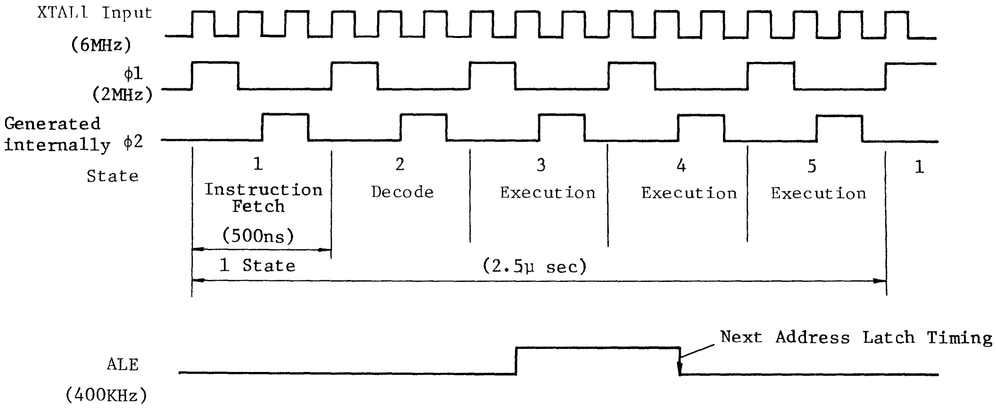


Fig.4 Instruction Cycle Timing

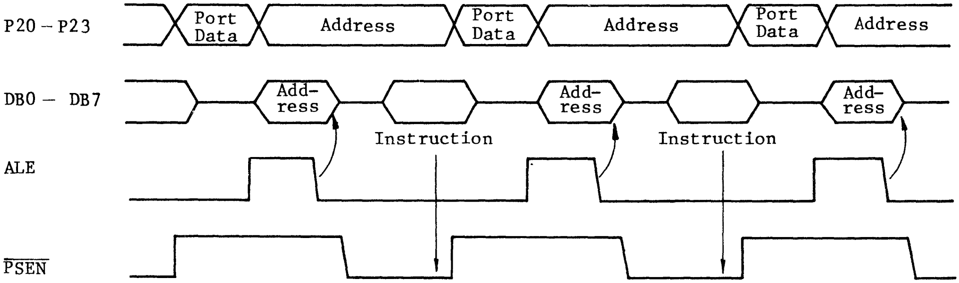
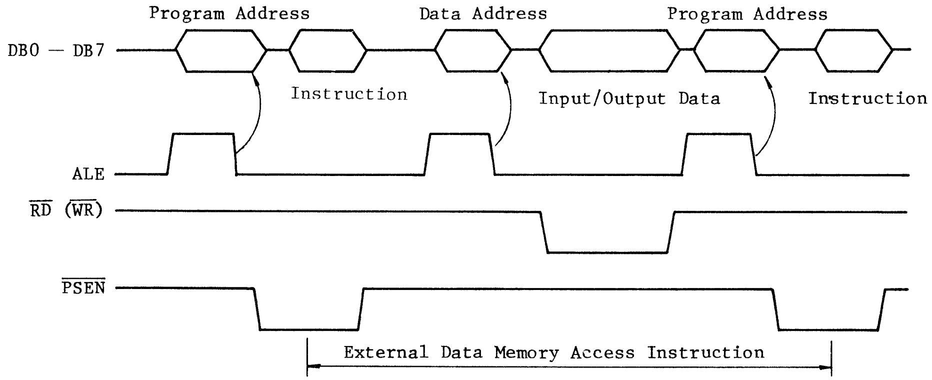


Fig.5 Timing of External Program Memory Access



Suggest we have two diagrams

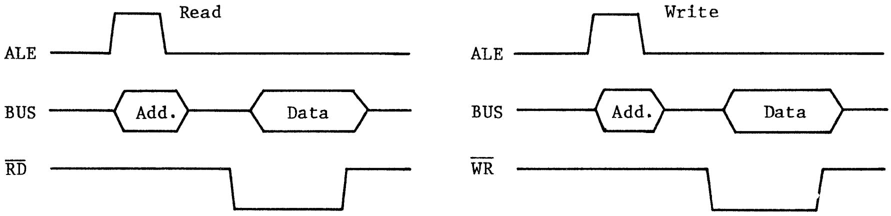


Fig.6 Timing of Accessing External Data Memory

- 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.

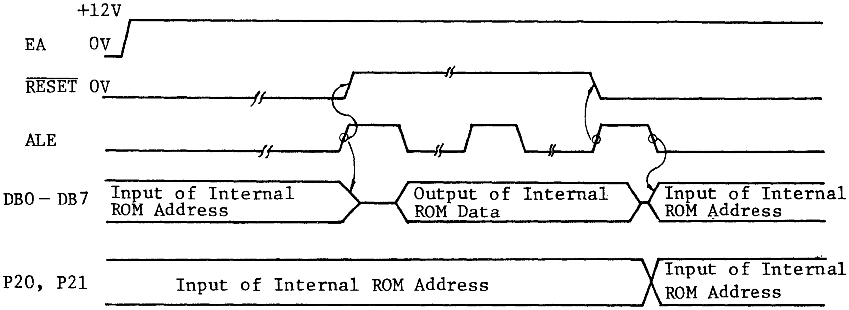


Fig.7 Timing of Rading Internal Program Memory

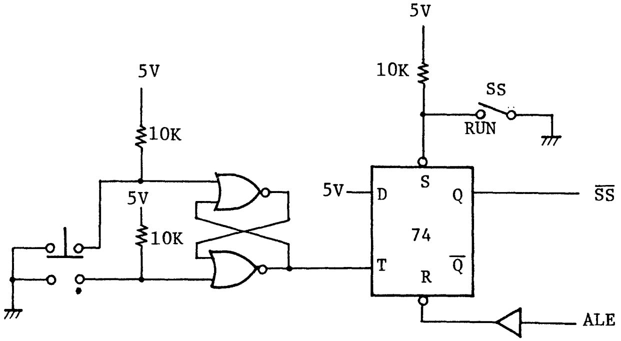


Fig.8(a) Single Step Circuit

(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 $\overline{SS}$ 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.

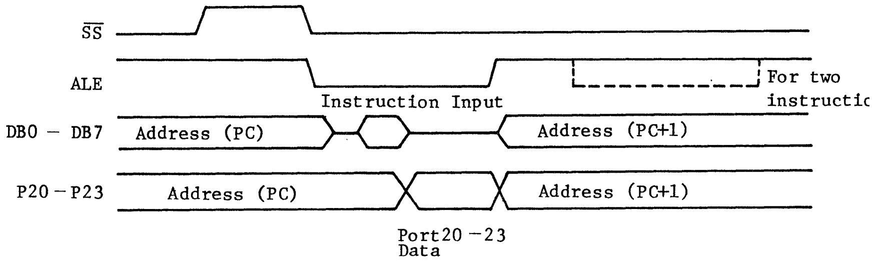


Fig.8(b) Single Step Operation Timing

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	–	–
JT0 Address	0 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T0 = 1 (PC)←(PC)+2 if T0 = 0	2	2	–	–
JNT0 Address	0 a7	0 a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T0 = 0 (PC)←(PC)+2 if T0 = 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 TCNT1	0	0	1	0	0	1	0	1	Timer interrupt is enabled	1	1	-	-
DIS TCNT1	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	-	-

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}=-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 other output pins)	I_{OH}=-40μA	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 TA=0°C to 70°C, $V_{CC}=V_{DD}=+5V\pm10\%$, $V_{SS}=0V$, 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 Note

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, TO) (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

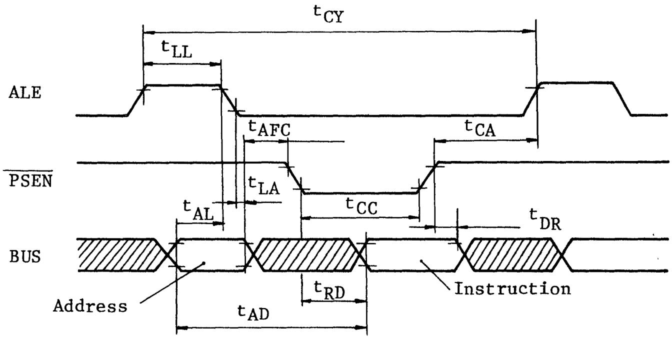
TA=-40°C to 85°C, V_{CC}=V_{DD}=+5V±10%, V_{SS}=0V, Unless otherwise Noted.

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($\overline{\text{PSEN}}$,$\overline{\text{RD}}$,$\overline{\text{WR}}$)		400	-	-	ns
t_{DW}	Data Setup Time ($\overline{\text{WR}}$)		420	-	-	ns
t_{WD}	Data Hold Time ($\overline{\text{WR}}$)	C_L=20pF	80	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	µs
t_{DR}	Data Hold Time ($\overline{\text{PSEN}}$, $\overline{\text{RD}}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{\text{PSEN}}$,$\overline{\text{RD}}$)		-	-	400	ns
t_{AW}	Address Setup Time ($\overline{\text{WR}}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	600	ns
t_{AFC}	Address Float Time ($\overline{\text{RD}}$, $\overline{\text{PSEN}}$)		-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

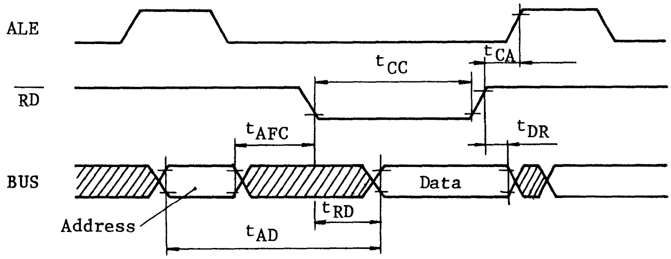
Note : tcy=2.5µs, Control Output: C_L=80pF, BUS Output: C_L=150pF, PORT 20 - 23: C_L=80pF.

TIMING WAVEFORM

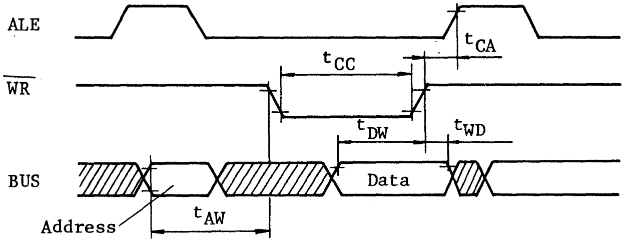
A. Instruction Fetch from External Program Memory



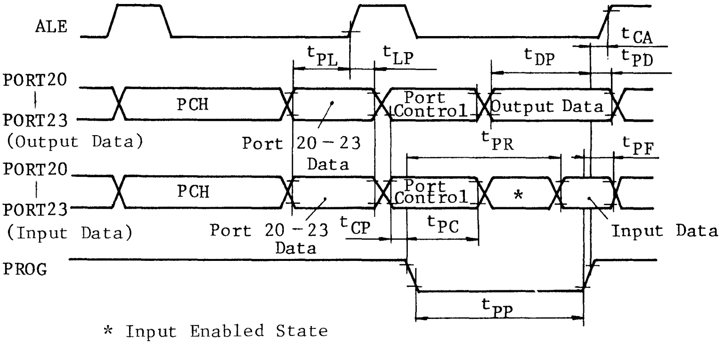
B. Read from External Data Memory



C. Write into External Data Memory

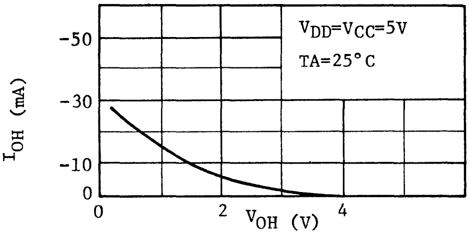


D. Timing of Port 2 during Expander Instruction Execution

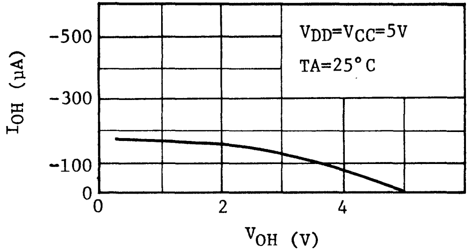


TYPICAL CHARACTERISTICS

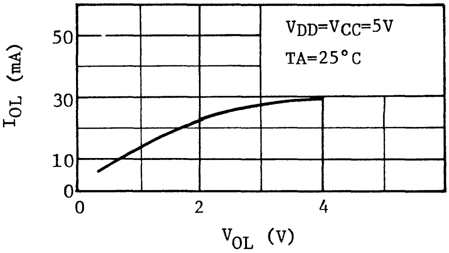
1) BUS: I_{OH} - V_{OH}



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



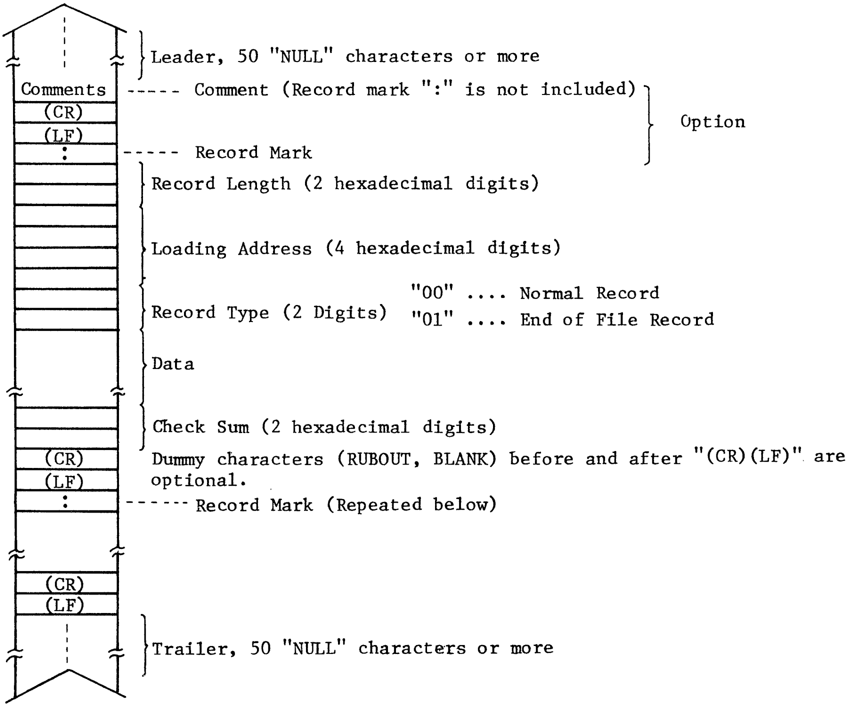
3) BUS, P1, P2: I_{OL} - V_{OL}



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


Leader, 50 "NULL" characters or more
Comments
(CR)
(LF)
:
Comment (Record mark ":" is not included)
Record Mark
Option
Record Length (2 hexadecimal digits)
Loading Address (4 hexadecimal digits)
Record Type (2 Digits)
"00" .... Normal Record
"01" .... End of File Record
Data
Check Sum (2 hexadecimal digits)
(CR)
(LF)
:
Dummy characters (RUBOUT, BLANK) before and after "(CR)(LF)" are optional.
Record Mark (Repeated below)
(CR)
(LF)
Trailer, 50 "NULL" characters or more

(2) Example of Tape List

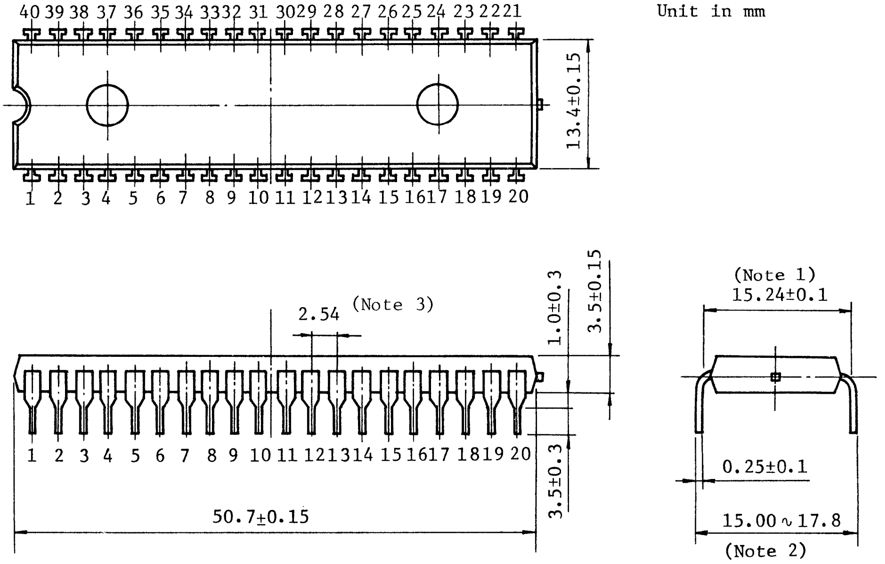
```
TOSHIBA MICRO COMPUTER TLCS-84
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:1000300019735 2F729F12F79AA9C057C5B851EED77
                         .
                         .
                         .
                         .
                         .
                         .
                         .
                         .
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20VC372BF60BD6
:00000001FF
```

Toshiba 東芝 **INTEGRATED CIRCUIT**

TECHNICAL DATA

TMP8048P/8048PI
TMP8035P/8035PI

OUTLINE DRAWING



Note : 1. This dimension shows the center of curvature of leads.

2. This dimension shows spread of leads.

3. The pitch of leads is 2.54 and the tolerance is ±0.25 from the theroretical center of each lead obtained having No.1 lead and No. 40 lead as the reference.

Note : Toshiba does not assume any responsibility for use of any circuitry described; no circuit patent licenses are implled, and Toshiba reserves the right, at any time without notice, to change said cricuitry.

Toshiba 東芝 **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.

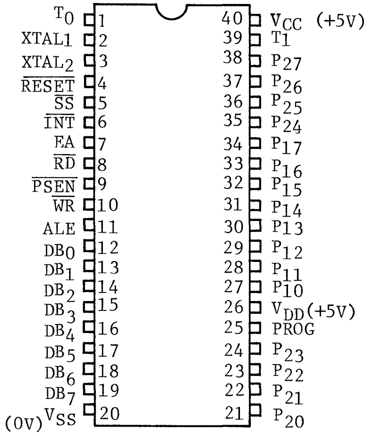
The TMP8039P is the equivalent of a TMP8049 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

The TMP8049P-6/TMP8039P-6 is a lower speed (6MHz) version of the TMP8049P/TMP8039P.

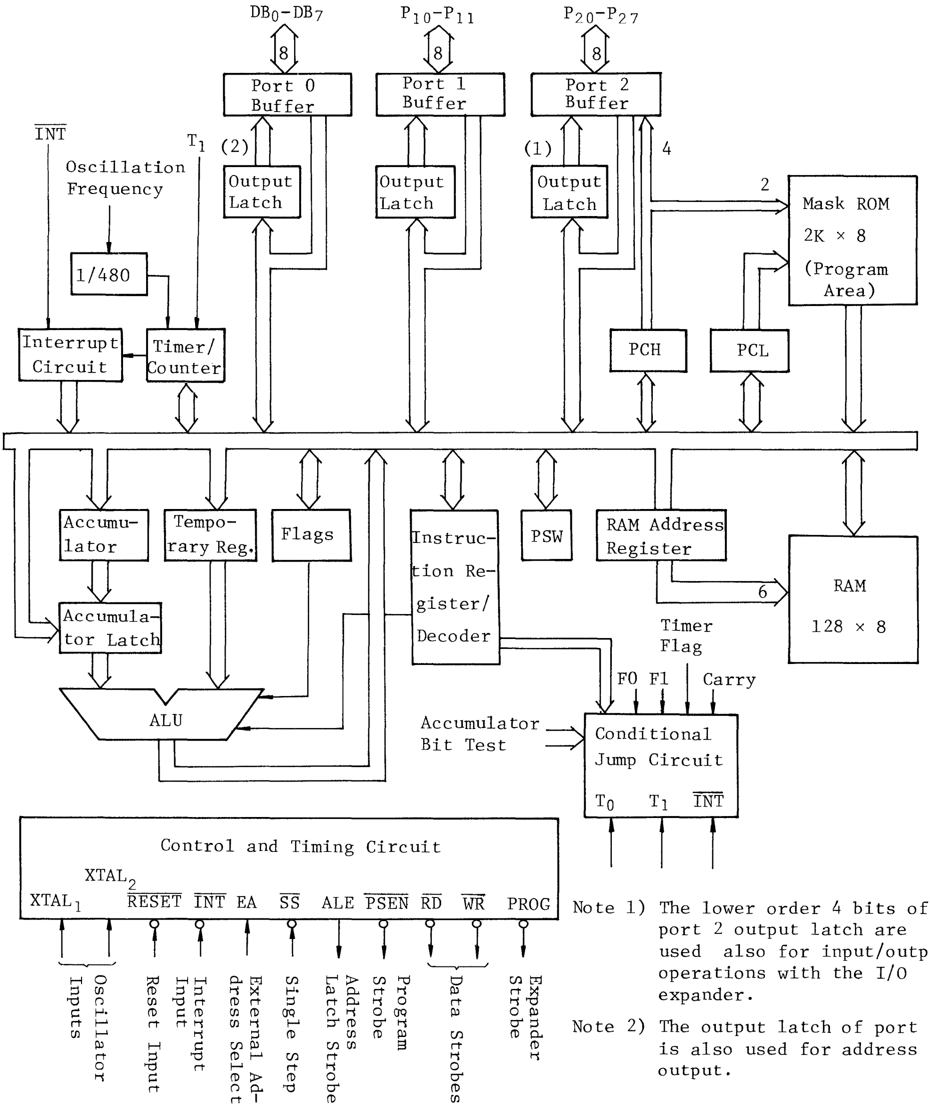
FEATURES

- 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≅50KΩ).

P_{20}-P_{27} (Input/Output) Port 2

8-bit quasi-bidirectional port (Internal Pullup≅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 JTl and JNTl 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 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

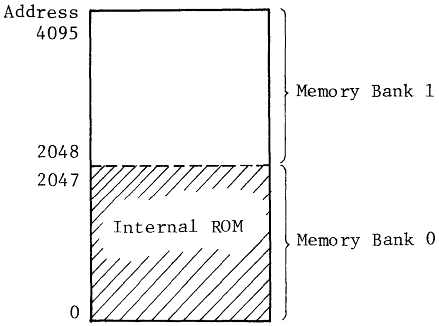
The following system functions of the TMP8049 are described in detail.

(1) Program Memory
(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 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

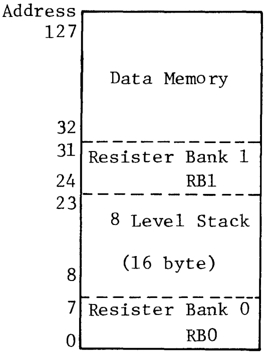
  A timer/counter interrupt resulting from a timer/counter overflow (if enabled) causes a jump to a subroutine defined by address held in Location 7.

- 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.

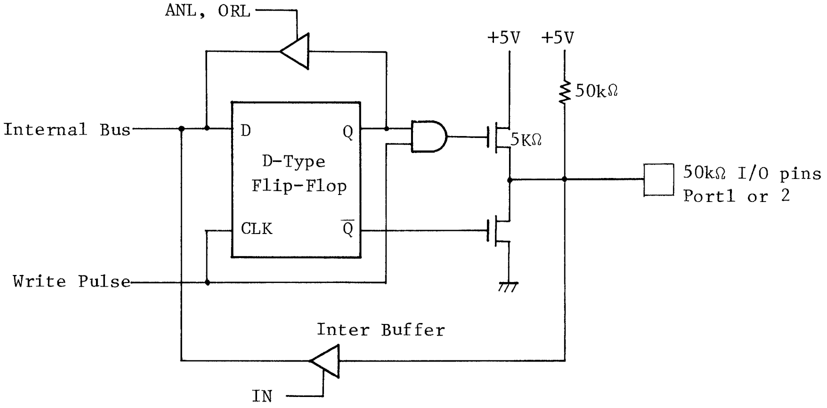


Fig.1 Input/Output Circuit of Port 1, Port 2

- 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.

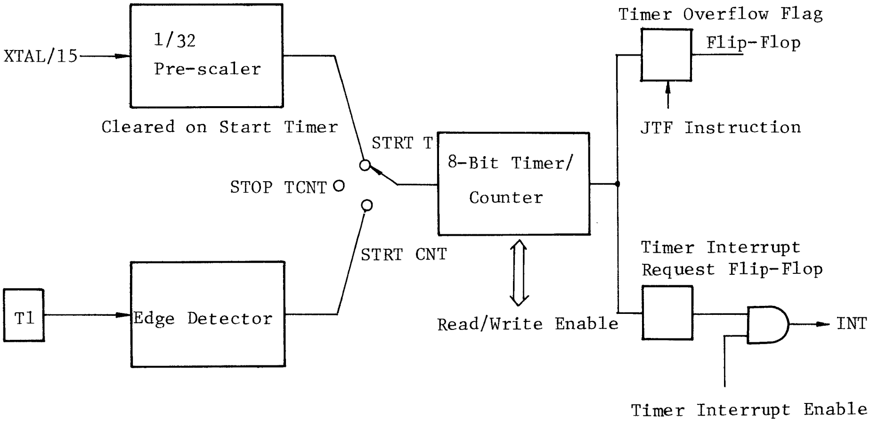


Fig.2 Concept of Timer Circuit

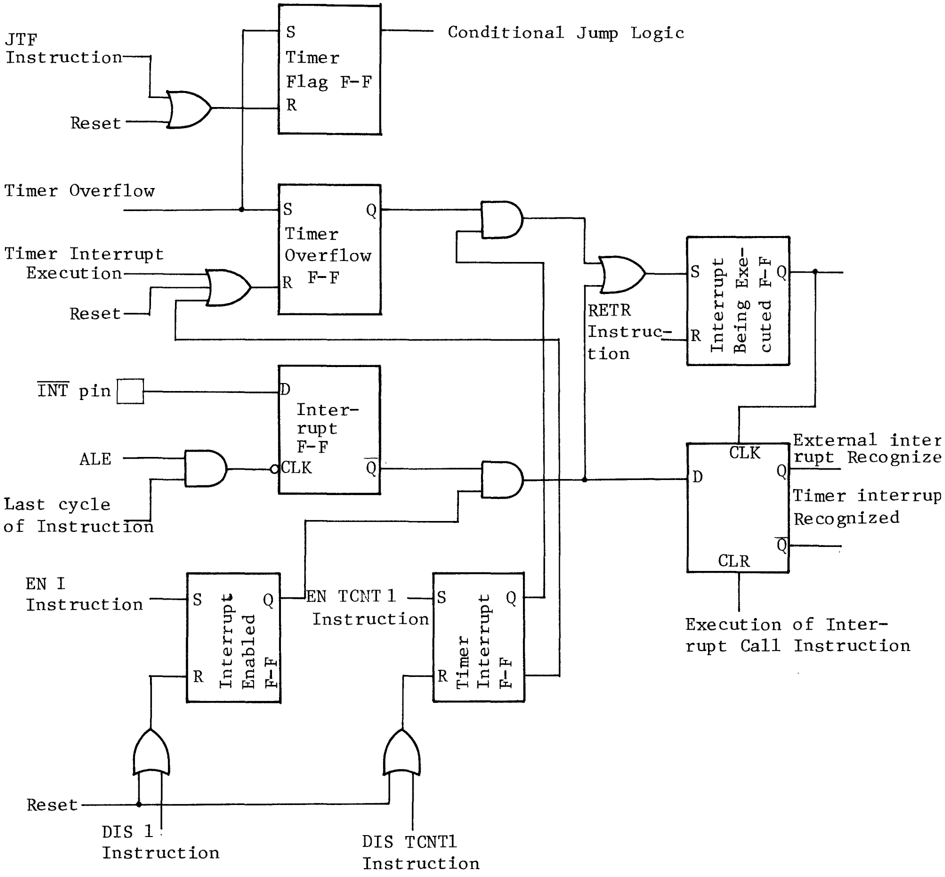


Fig.3 Concept of Interrupt Control Circuit

(5) Interrupt Control Circuit

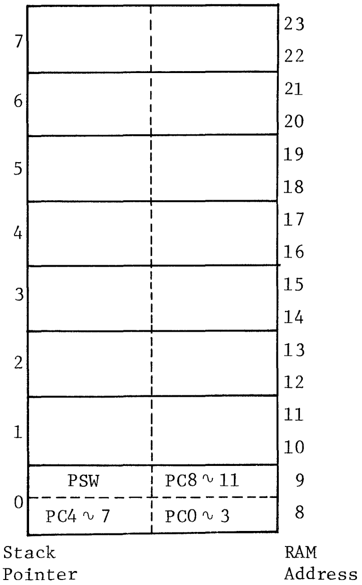
- There are two distinct types of Interrupts in the TMP8049.
  - (1) External Interrupt from the $\overline{\text{INT}}$ terminal
  - (2) Timer Interrupt caused by timer overflow

- 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 $\overline{\text{INT}}$ pin. $\overline{\text{INT}}$ 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 $\overline{\text{INT}}$ and times overflow occur simultaneously then external request $\overline{\text{INT}}$ takes precedence.
- If an extra external interrupt is needed in addition to $\overline{\text{INT}}$ 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.

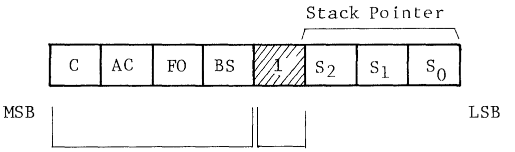


(7) Flag 0, Flag 1, (F0, F1)

- 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.



Saved in stack area at the time of Subroutine Call.

Spare ("1" during Read)

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

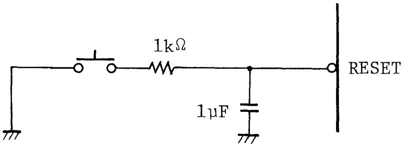
Bit 5 : Flag 0 (FO)

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.

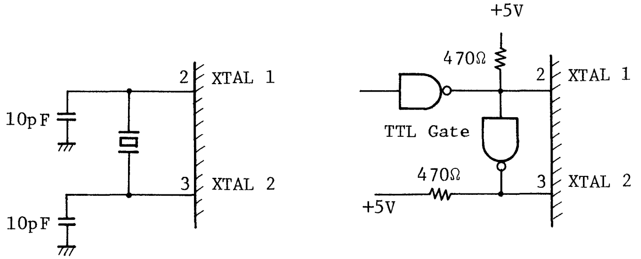
(ix) Clears Timer Flag.

(x) Clears F0 and F1.

(xi) Disables clock output TO.

(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.

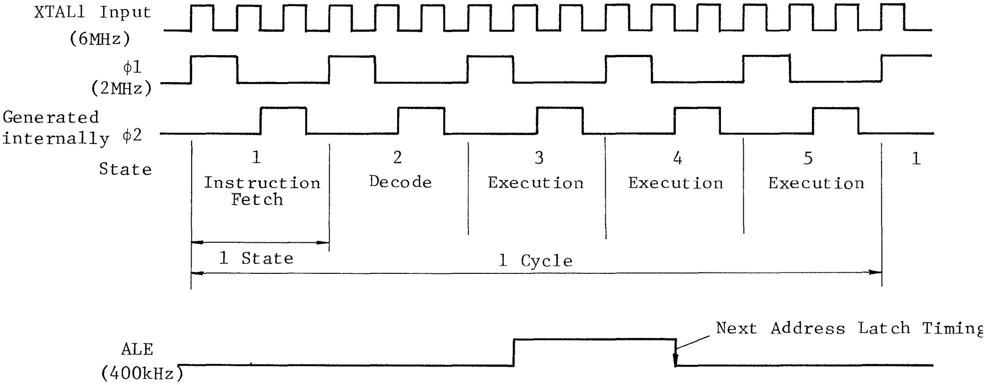


Fig.4 Instruction Cycle Timing

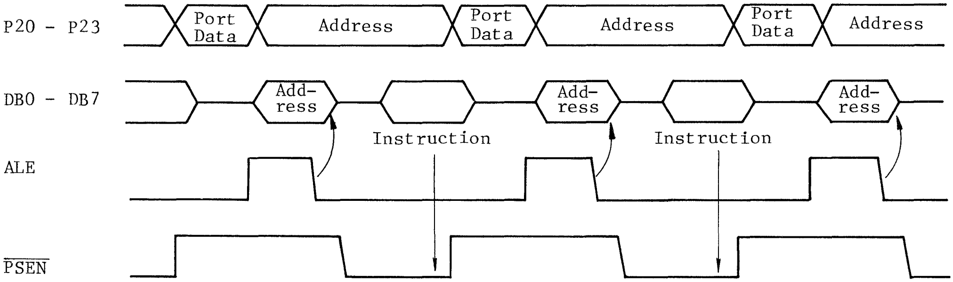
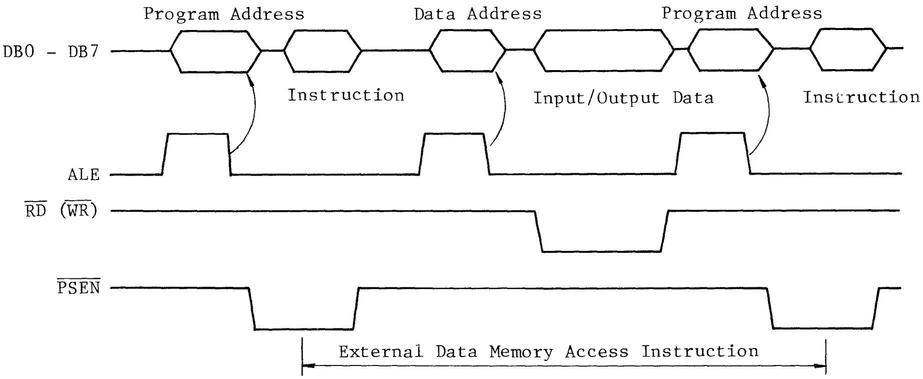


Fig.5 Timing of External Program Memory Access



Suggest we have two diagrams

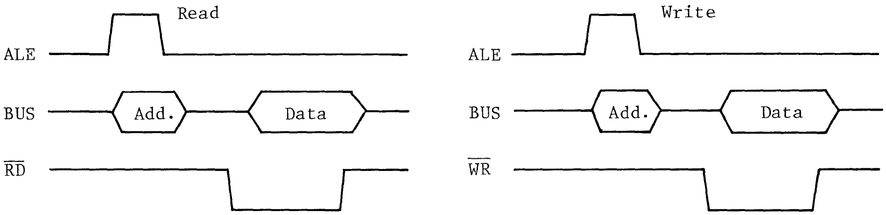


Fig.6 Timing of Accessing External Data Memory

- 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.

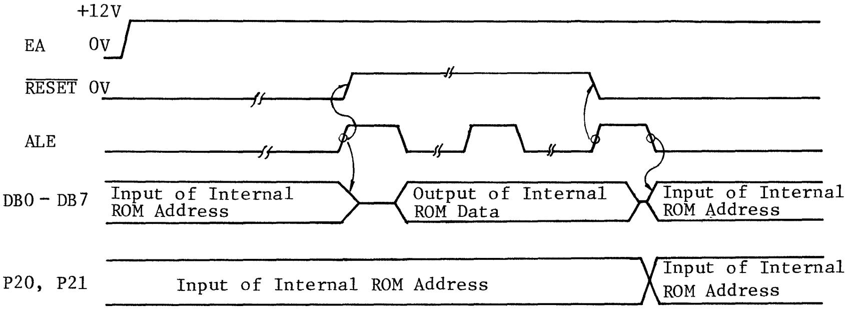

+12V
EA
0V
RESET 0V
ALE
DB0 - DB7
Input of Internal ROM Address
Output of Internal ROM Data
Input of Internal ROM Address
P20, P21
Input of Internal ROM Address
Input of Internal ROM Address

Fig.7 Timing of Rading Internal Program Memory

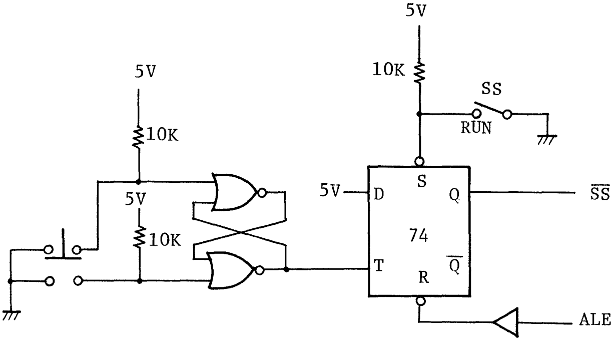


Fig.8(a) Single Step Circuit

(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 processo,r 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 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.

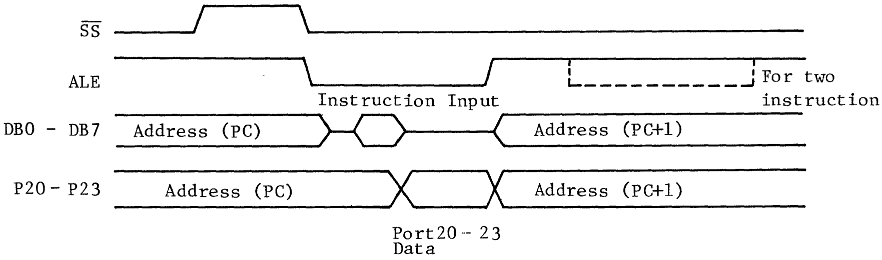


Fig.8(b) Single Step Operation Timing

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	–	–
JT0 Address	0 a7	0 a6	1 a5	1 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T0 = 1 (PC)←(PC)+2 if T0 = 0	2	2	–	–
JNT0 Address	0 a7	0 'a6	1 a5	0 a4	0 a3	1 a2	1 a1	0 a0	(PC0-7)←(a0-7) if T0 = 0 (PC)←(PC)+2 if T0 = 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	–	–
JNI 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 FO	1	0	0	0	0	1	0	1	(FO)←0	1	1	-	-
CPL FO	1	0	0	1	0	1	0	1	(FO)←NOT(FO)	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	-	-
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	-	-

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	-	VCC	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 tothe 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		-	-	50	mA
$I_{DD}+I_{CC}$	Total Supply Current		-	-	170	mA

AC CHARACTERISTICS

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

SYMBOL	PARAMETER	TEST CONDITION	TMP8049P/ TMP8039P		TMP8049P-6/ TMP8039P-6		UNI
			MIN.	MAX.	MIN.	MAX.	
t_{LL}	ALE Pulse Width		150	-	400	-	n
t_{AL}	Address Setup Time (ALE)		70	-	150	-	n
t_{LA}	Address Hold Time (ALE)		50	-	80	-	n
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		300	-	700	-	n
t_{DW}	Data Setup Time ($\overline{WR}$)		250	-	500	-	n
t_{WD}	Data Hold Time ($\overline{WR}$)	C_L=20pF	40	-	120	-	n
t_{CY}	Cycle Time	11MHz XTAL (6MHz XTAL for -6)	1.36	15.0	2.5	15.0	μ
t_{DR}	Data Hold Time ($\overline{PSEN}$, $\overline{RD}$)		0	100	0	200	n
t_{RD}	Data Input Read Time ($\overline{PSEN}$, $\overline{RD}$)		-	200	-	500	n
t_{AW}	Address Setup Time ($\overline{WR}$)		200	-	230	-	n
t_{AD}	Address Setup Time (Data Input)		-	400	-	950	n
t_{AFC}	Address Float Time ($\overline{RD}$, $\overline{PSEN}$)		-10	-	0	-	n
t_{CP}	Port Control Setup Time (PROG)		100	-	110	-	n
t_{PC}	Port Control Hold Time (PROG)		60	-	130	-	n
t_{PR}	Port 2 Input Data Set Time (PROG)		-	650	-	810	n
t_{DP}	Output Data Setup Time (PROG)		200	-	220	-	n
t_{PD}	Output Data Hold Time (PROG)		20	-	65	-	n
t_{PF}	Port 2 Input Data Hold Time (PROG)		0	150	0	150	n
t_{PP}	PROG Pulse Width		700	-	1510	-	n
t_{PL}	Port 2 I/O Data Setup Time		250	-	500	-	n
t_{LP}	Port 2 I/O Data Hold Time		120	-	150	-	n

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, $\overline{\text{RESET}}$)		2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{\text{RESET}}$)		3.8	-	V_{CC}	V
V_{OL}	Output Low Voltage (BUS)	$I_{OL}=1.6mA$	-	-	0.45	V
V_{OL1}	Output Low Voltage ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, 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 ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, 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, $\overline{\text{INT}}$)	$V_{SS} \leq V_{IN} \leq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current (P10-17, P20-P27, EA, $\overline{\text{SS}}$)	$V_{SS}+0.45 \leq V_{IN} \leq V_{CC}$	-	-	-700	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

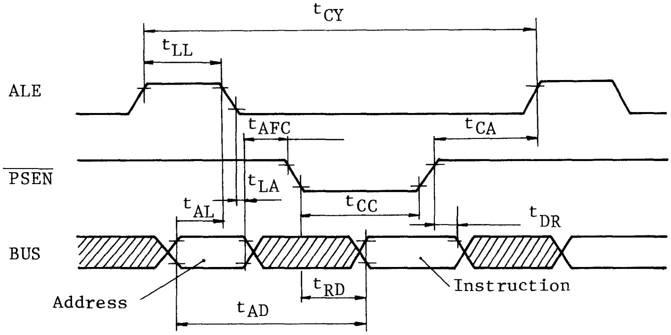
TA=-40°C to 85°C, V_{CC}=V_{DD}=+5V±10%, V_{SS}=0V, Unless otherwise Noted.

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

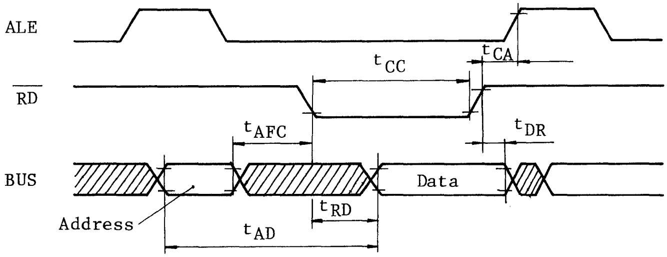
Note : tcy=2.5μs, Control Output: C_L=80pF, BUS Output: C_L=150pF, PORT 20-23: C_L=80pF.

TIMING WAVEFORM

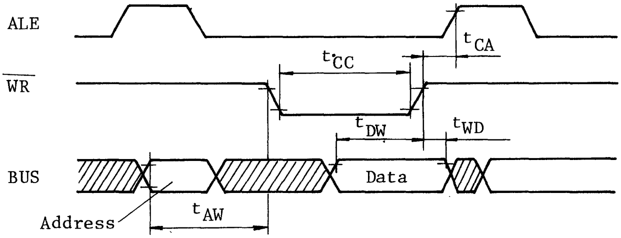
A. Instruction Fetch from External Program Memory



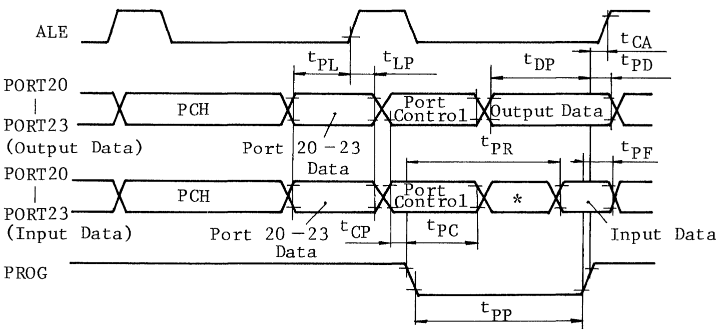
B. Read from External Data Memory



C. Write into External Data Memory



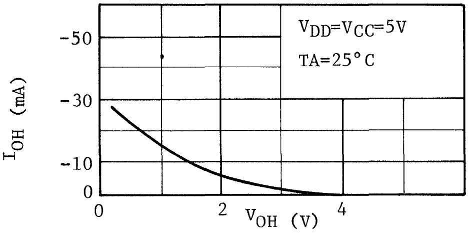
D. Timing of Port 2 during Expander Instruction Execution



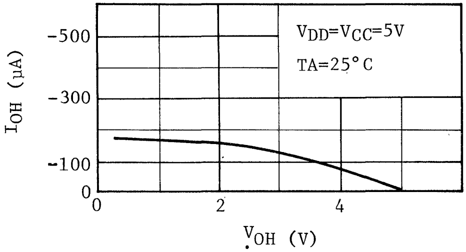
* Input Enabled State

TYPICAL CHARACTERISTICS

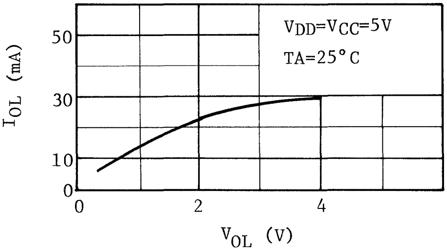
1) BUS:I_{OH} - V_{OH}



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



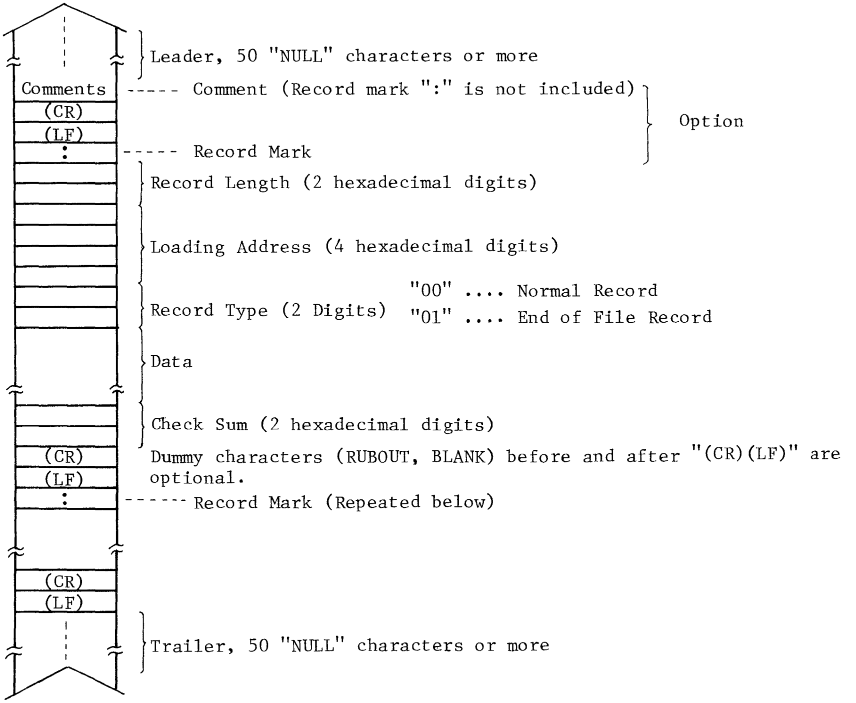
3) BUS, P1, P2: I_{OL} - V_{OL}



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



Leader, 50 "NULL" characters or more

----- Comment (Record mark ":" is not included)

----- Record Mark

Option

Record Length (2 hexadecimal digits)

Loading Address (4 hexadecimal digits)

Record Type (2 Digits) "00" .... Normal Record
"01" .... End of File Record

Data

Check Sum (2 hexadecimal digits)

Dummy characters (RUBOUT, BLANK) before and after "(CR)(LF)" are optional.

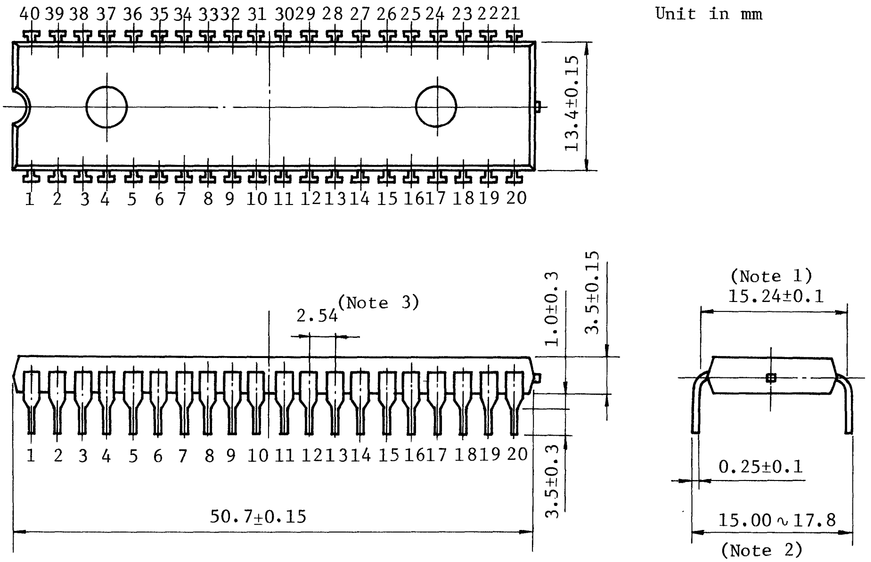
------ Record Mark (Repeated below)

Trailer, 50 "NULL" characters or more

(2) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLCS-84
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:1000300019735 2F729F12F79AA9C057C5B851EED77
                    .
                    .
                    .
                    .
                    .
                    .
                    .
                    .
                    .
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20VC372BF60BD6
:00000001FF
```

OUTLINE DRAWING



Note : 1. This dimension shows the center of curvature of leads.

2. This dimension shows spread of leads.

3. The pitch of leads is 2.54 and the tolerance is ±0.25 from the theroretical center of each lead obtained having No.1 lead and No. 40 lead as the reference.

Note : Toshiba does not assume any responsibility for use of any circuitry described; no circuit patent licenses are implled, and Toshiba reserves the right, at any time without notice, to change said cricuitry.

Toshiba 東芝 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

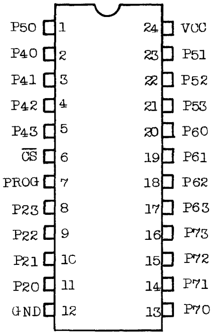
o High output drive

o Direct extension of resident TMP8048P/TMP8049P I/O ports.

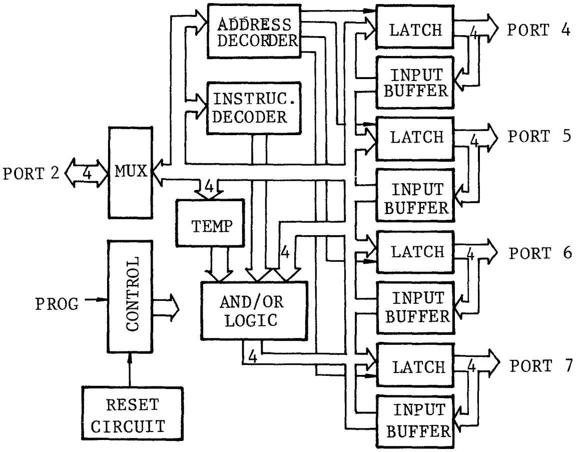
o Compatible with intel's 8243

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

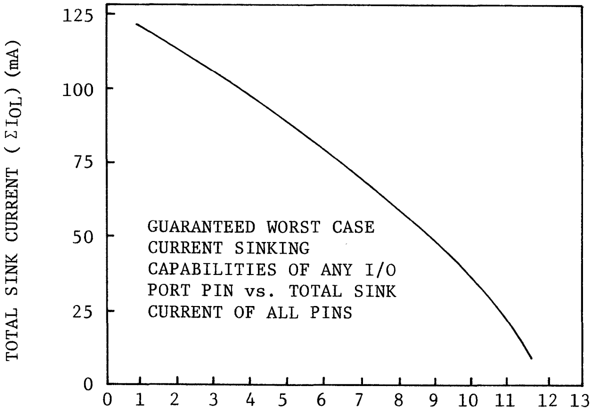
The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi, A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi, A takes new data AND's it with the old data and then writes it to the port. Operation code and 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.



MAXIMUM SINK CURRENT ON ANY PIN@.45V
MAXIMUM I_{OL} WORST CASE PIN(mA)

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.	Unit
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≤V_{IN}≤V_{CC}	-10		20	μA
I_{IL2}	Input Leadage Port 2, $\overline{CS}$, PROG	0V≤V_{IN}≤V_{CC}	-10		10	μA
I_{CC}	V_{CC} Supply Current			10	20	mA
I_{OL}	Sum of all I_{OL} 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.	Unit
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%

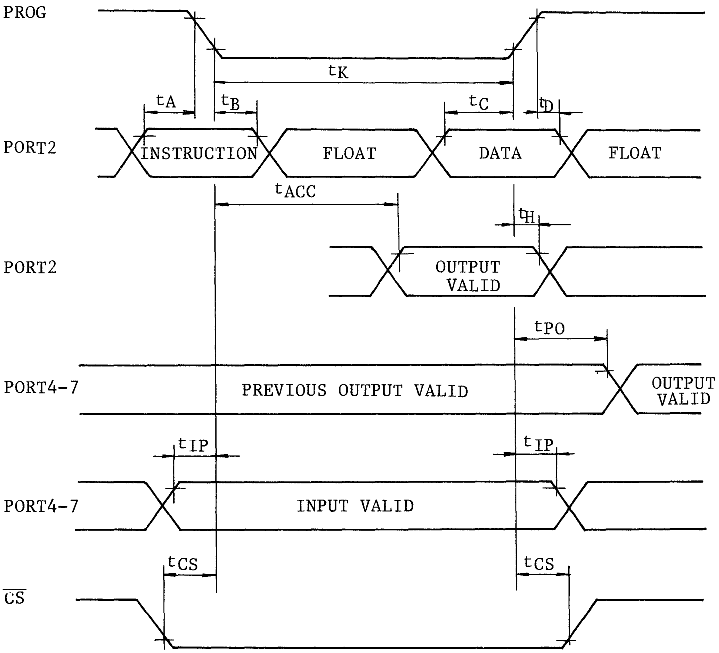
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage		-0.5		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
V_{OL1}	Outptu Low Voltage Ports 4-7	I_{OL}=4.5mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL}=20mA			1	V
V_{OL3}	Output Low Voltage Port 2	I_{OL}=0.6mA			0.45	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≤V_{IN}≤V_{CC}	-10		20	μA
I_{IL2}	Input Leakage Port 2, $\overline{CS}$,PROG	0V≤V_{IN}≤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%

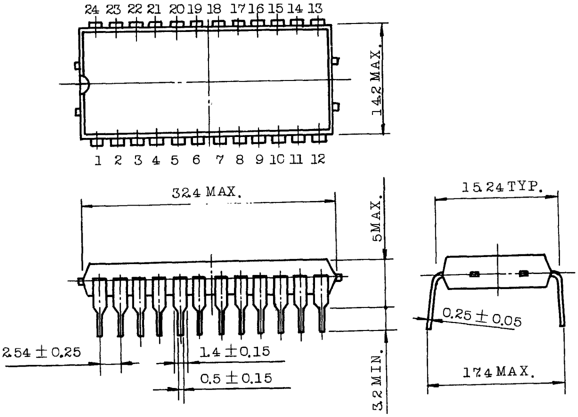
Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
t_A	Code Valid before PROG	C_L = 80pF	100			ns
t_B	Code Valid after PROG	C_L = 20pF	60			ns
t_C	Data Valid before PROG	C_L = 80pF	200			ns
t_D	Data Valid after PROG	C_L = 20pF	20			ns
t_H	Floating after PROG	C_L = 20pF	0		150	ns
t_K	PROG Negative Pulse 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.

GENERAL DESCRIPTION

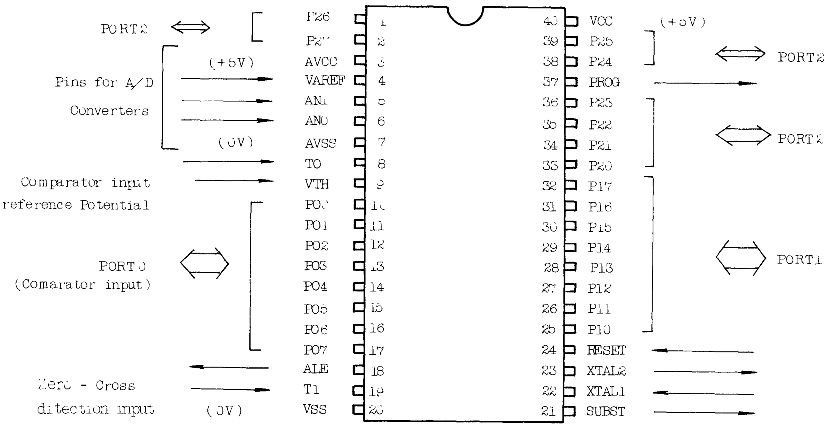
The TMP8022P is one version of the TLCS-84 family, which is an 8-bit single chip microcomputer containing A/D converter.

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 dictection circuit, etc. are all integraied 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

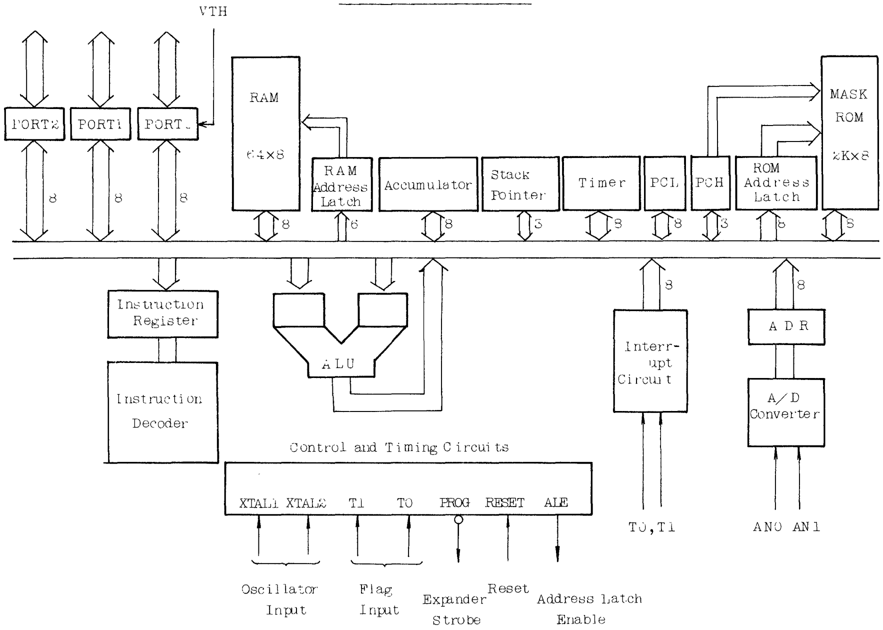
Pin Name	Pin No.	Input/ Output	Function
XTAL1 XTAL2	22 23	Input	A terminal for connecting the oscillator or an input terminal for the external clock.
RESET	24	Input	High active signal and initializes the chip. When a low level voltage is applied to this pin, a program starts from adress 0.

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	PORTO 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)

Classi-fication	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	$(A) \leftarrow (P_p)$ p=0,1,2	-	-	1	2	08, 09, 0A
	OUTL Pp,A	$(P_p) \leftarrow (A)$ p=0,1,2	-	-	1	2	90, 39, 3A
	MOVD A, Pp	$(A_{0-3}) \leftarrow (P_p)$ 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	$(P_p) \leftarrow (P_p) \wedge (A_{0-3})$ p=4-7	-	-	1	2	9C-9F
	ORLD Pp,A	$(P_p) \leftarrow (P_p) \vee (A_{0-3})$ p=4-7	-	-	1	2	8C-8F
Register Instruction	INC Rr	$(R_r) \leftarrow (R_r)+1$ r=0-7	-	-	1	1	18-1F
	INC @Rr	$((R_r)) \leftarrow ((R_r))+1$ 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	$(R_r) \leftarrow (R_r)-1$, 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

Classi-fication	Mnemonics	Functional Description	Effected Flag		Bytes	Cycles	Operation Code (Hexadecimal)
			C	AC			
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 Instruc-tion	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		Bytes	Cycles	Operation Code (Hexadecimal)
				C	AC			
Data Moves Instruction	XCH	A,Rr	(A) ⇄ (Rr) r=0-7	-	-	1	1	28-2F
	XCH	A,@Rr	(A) ⇄ ((Rr)) r=0,1	-	-	1	1	20-21
	XCHD	a,@Rr	(A_{0-3}) ⇄ ((Rr)) r=0,1	-	-	1	1	30-31
	MOVP	A,@A	(PC_{0-7})←(A) [Note] (A) ← ((PC))	-	-	1	2	A3
Timer/Counter Instruction	MOV	A,T	(A) ← (T)	-	-	1	1	42
	MOV	T,A	(T) ← (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) ← (CRR)	-	-	1	2	80
	SEL	ANO	ANO 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) ← (SP) - 1 (PC) ← ((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 IR04-7	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	DISI	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	JNT0	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	JT0	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 T,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 AN0							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	MOZ 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,XTAL1)		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
ILO	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

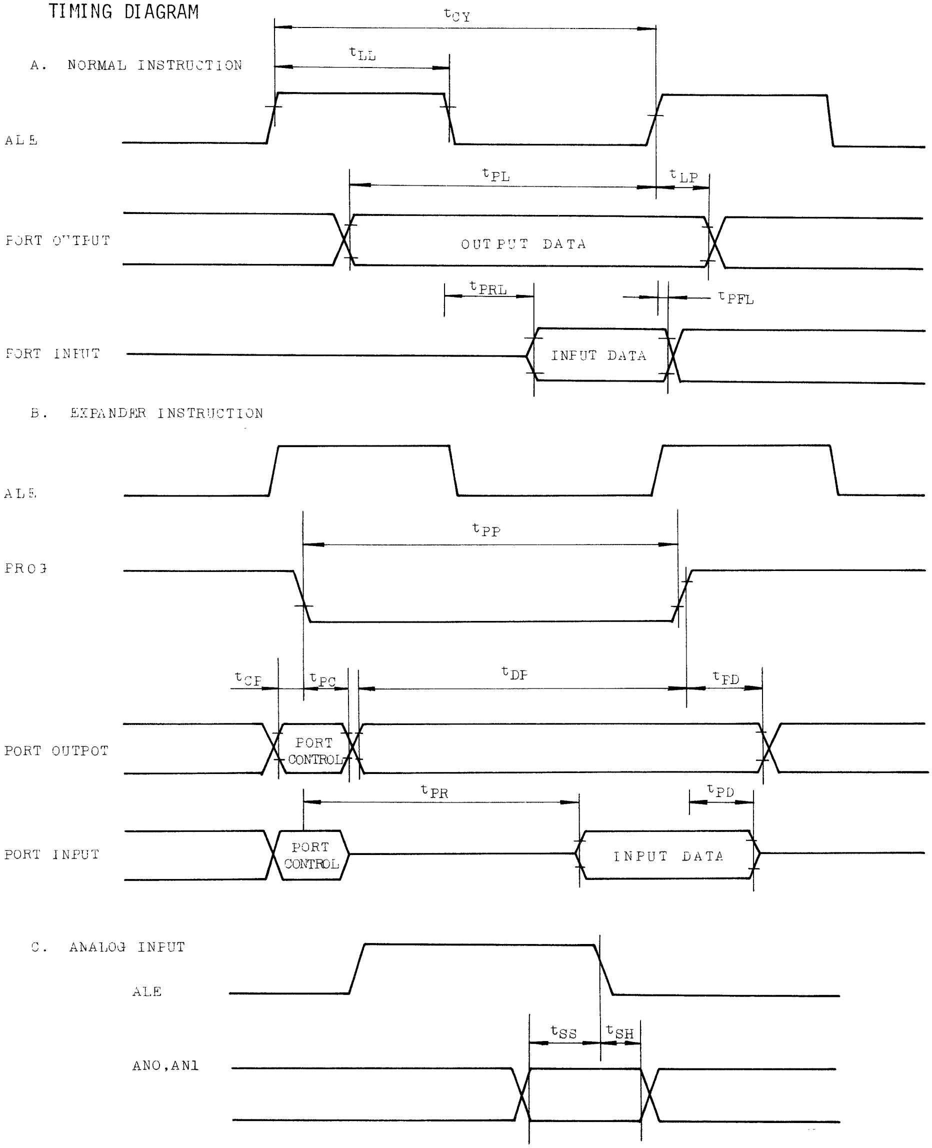
Test Condition t_{CY} = 8.38 μs C_L = 80 pF

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, $AV_{CC}/2 \leqq VAREF \leqq AV_{CC}$

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



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 PORTO, 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

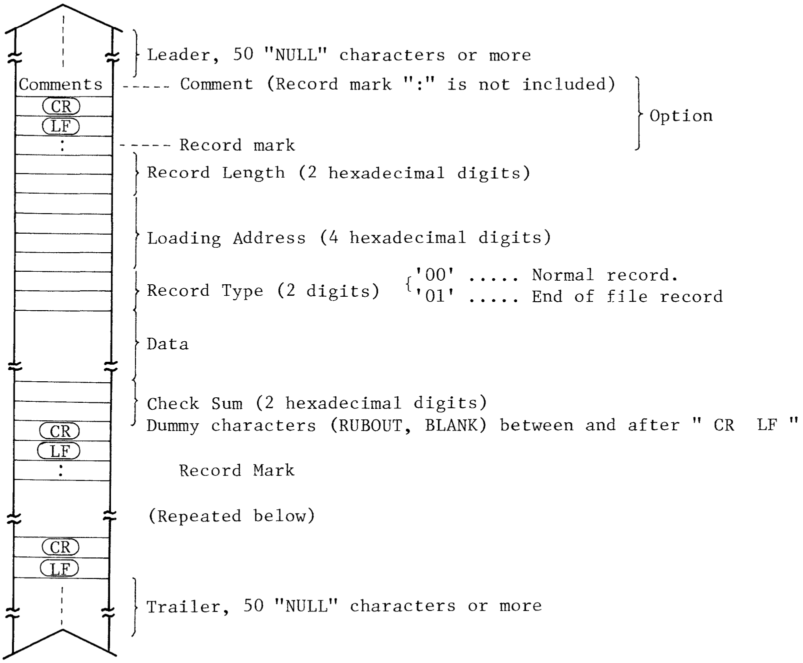
0 : Without pull-up register   1 : With pull-up register

Terminal name / Option designation	PORTO								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

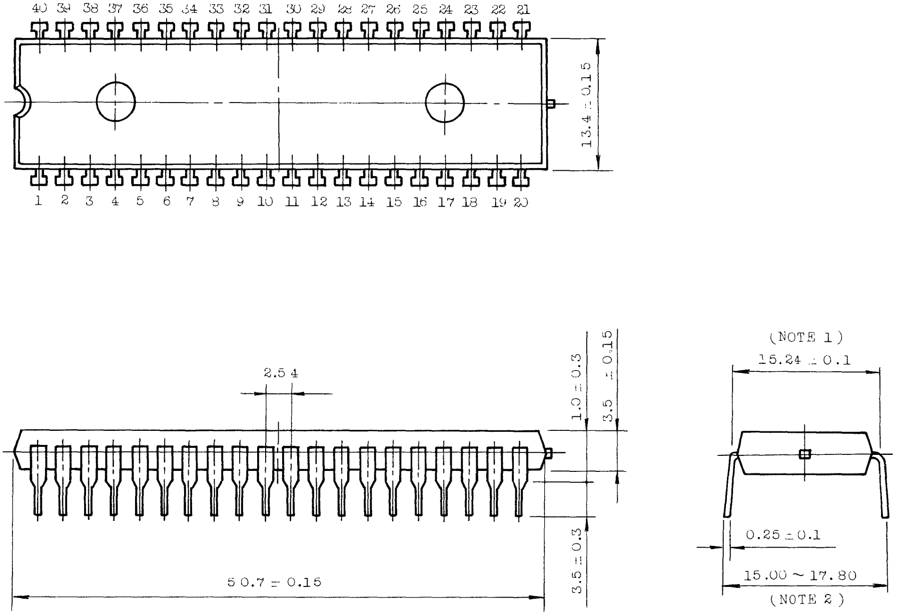
(2) Tape Format


Leader, 50 "NULL" characters or more
Comments
Comment (Record mark ":" is not included)
CR
LF
:
Record mark
Option
Record Length (2 hexadecimal digits)
Loading Address (4 hexadecimal digits)
Record Type (2 digits)
'00' ..... Normal record.
'01' ..... End of file record
Data
Check Sum (2 hexadecimal digits)
Dummy characters (RUBOUT, BLANK) between and after " CR LF "
CR
LF
:
Record Mark
(Repeated below)
CR
LF
Trailer, 50 "NULL" characters or more

(3) Example of Tape List

```
TOSHIBA MICRO COMPUTER TLSC-84
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884dde67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                     .
                     .
                     .
                     .
                     .
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1E41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20BC372BF60BD6
:00000001FF
```

UNIT : mm



Note: 1. This dimension shows the center of curvature of leads.

2. This dimension shows spread of leads.

3. The pitch of leads is 2.54 and the tolerance is ±0.25 from the theoretical center of each lead obtained No.40 lead as the reference.

Toshiba 東芝 INTEGRATEDCIRCUIT
TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL
INTEGRATED CIRCUIT
TMP80C49P-6, TMP80C39P-6
Silicon Monolithic
CMOS Silicon Gate

PRELIMINAR.

8-BIT SINGLE-CHIP MICROCOMPUTER

GENERAL DESCRIPTION

The TMP80C49P-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.

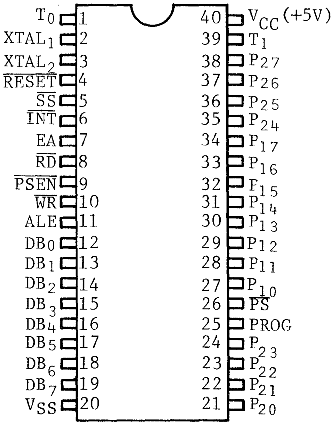
The TMP80C49P-6 is particularly efficient as a controller. It has extensive bit handling capability as well as facilities for both binary and BCD arithmetic.

The TMP80C39P-6 is the equivalent of a TMP80C49P-6 without ROM program memory on chip. By using this device with external EPROM or RAM, software debugging becomes easy.

FEATURES

- Software compatible with TMP8049P/i8049
- CMOS/LSI for low power dissipation - less than 50 mW at 5V, 6MHz
- High Noise Immunity
- 2.5μs Instruction Cycle
- Extended temperature operation:
  -40°C to +85°C
- Single power supply
- 2K × 8 masked ROM
- 128 × 8 RAM
- 27 I/O lines
- Interval Timer/Event Counter
- Power Down Mode

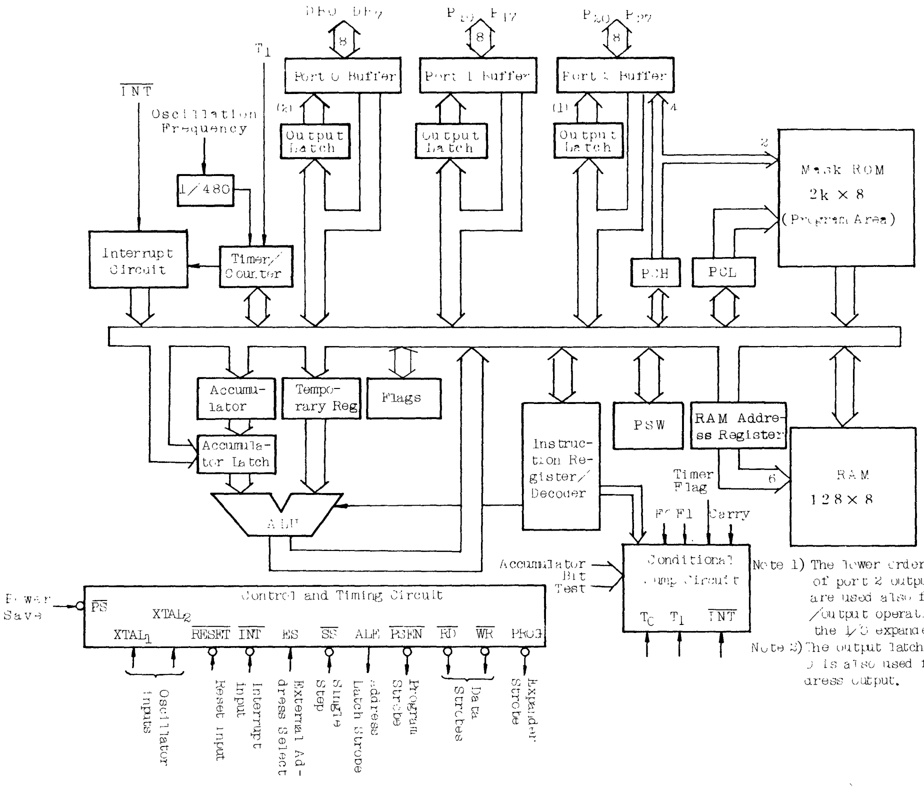
PIN CONNECTIONS (TOP VIEW)



TMP80C49P-6
TMP80C39P-6

PRELIMINARY

BLOCK DIAGRAM



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

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

PIN NAMES AND PIN DESCRIPTION

V_{SS} (Power Supply)
Circuit GND potential

V_{CC} (Power Supply)
+5V during operation

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

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

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

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

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

DB0-DB7 (Input/Output, Tri-State)
True bidrectional port which can be written or read synchronously using the $\overline{\text{RD}}$, $\overline{\text{WR}}$ 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 $\overline{\text{PSEN}}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{\text{RD}}$, and $\overline{\text{WR}}$.

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_0 is also used during programming.

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.

$\overline{\text{INT}}$ (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)

$\overline{\text{RD}}$ (Output)
Output strobe activated druing 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).

$\overline{\text{WR}}$ (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 Eanble. 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.

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, $\overline{RESET}$, T0, T1)		2.2	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		0.7V_{CC}	-	V_{CC}	V
V_{IH2}	Input High Voltage (T0, T1)		0.5V_{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_{CC}	V_{CC} Supply Current	V_{CC}=5V, f_{XTAL}=6MHz V_{IH}=V_{CC}-0.2V V_{IL}=0.2V, C_{X2}=0pF	-	-	10	mA

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	-	0.15 V_{CC}	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{\text{RESET}}$, T0, T1)		$0.5V_{CC}$	-	V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{\text{RESET}}$)		$0.7V_{CC}$	-	V_{CC}	V
V_{IH2}	Input High Voltage (T0, T1)		$0.5V_{CC}$	-	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, $\overline{\text{INT}}$, EA, $\overline{\text{PS}}$)	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{LI1}	Input Leak Current ($\overline{\text{SS}}$, $\overline{\text{RESET}}$)	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.1}$	μA
I_{LI2}	Input Leak Current (P10-P17, P20-P27)	$V_{SS}+0.45V \leqq V_{IN} \leqq V_{CC}$	-	-	$-\frac{V_{CC}}{0.1}$	μA
I_{LO}	Output Leak Current (BUS, T0) (High impedance condition)	$V_{SS}+0.45V \leqq V_{IN} \leqq V_{CC}$	-	-	±10	μA
I_{CC}	V_{CC} Supply Current	V_{CC}=5V, f_{XTAL}=6MHz $V_{IH}=V_{CC}-0.2V$, V_{IL}=0.2V, C_{X2}=0pF	-	-	10	mA

AC CHARACTERISTICS

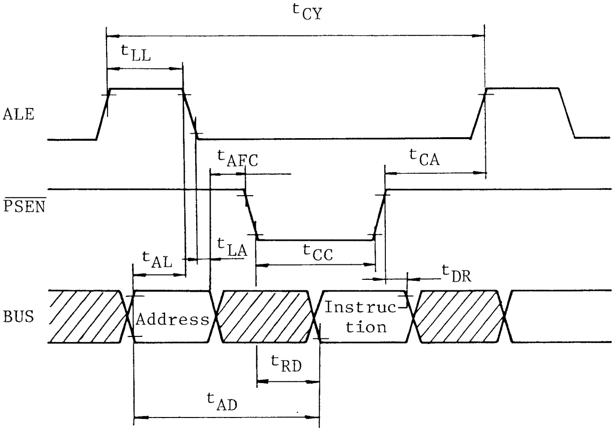
T_{OPR}=-40°C to 85°C, V_{CC}=+5V±20%, V_{SS}=0V, Unless Otherwise Noted.

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 ($\overline{\text{PSEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$)		700	-	-	ns
t_{DW}	Data Setup Time ($\overline{\text{WR}}$)		500	-	-	ns
t_{WD}	Data Hold Time ($\overline{\text{WR}}$)	C_L=20pF	120	-	-	ns
t_{CY}	Cycle Time		2.5	-	15.0	µs
t_{DR}	Data Hold Time ($\overline{\text{PSEN}}$, $\overline{\text{RD}}$)		0	-	200	ns
t_{RD}	Data Input Read Time ($\overline{\text{PSEN}}$, $\overline{\text{RD}}$)		-	-	500	ns
t_{AW}	Address Setup Time ($\overline{\text{WR}}$)		230	-	-	ns
t_{AD}	Address Setup Time (Data Input)		-	-	950	ns
t_{AFC}	Address Float Time ($\overline{\text{RD}}$, $\overline{\text{PSEN}}$)		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)		-	-	310	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

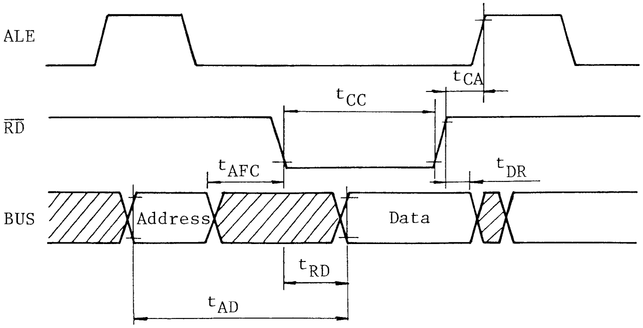
Note : t_{cy}=2.5µs (f_X=6MHz)
Control Outputs ; C_L=80pF, BUS Outputs ; C_L=150pF

TIMING WAVEFORM

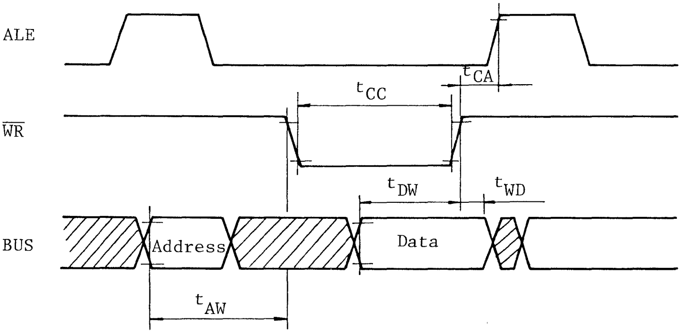
A. Instruction Fetch from External Program Memory



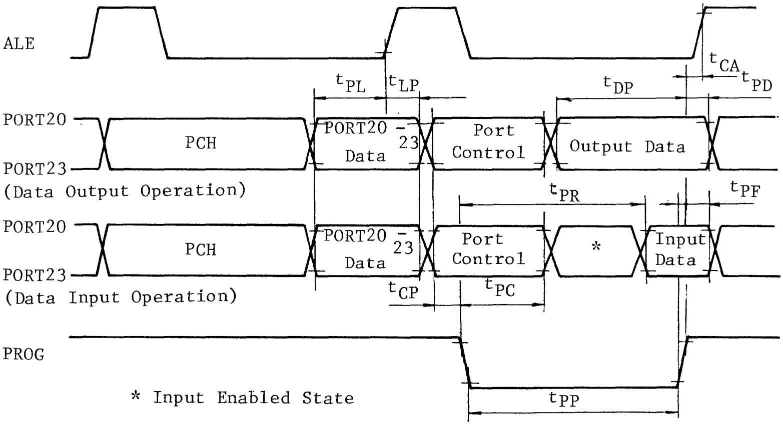
B. Read from External Data Memory



C. Write into External Data Memory



D. Timing of Port 2 during Expander Instruction Execution



* Input Enabled State

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 resume osillation 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)

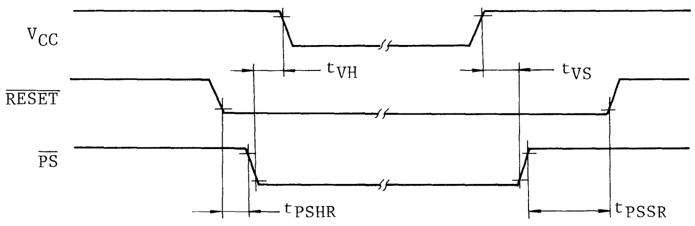
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB1}	Standby Voltage(1)		2.0	-	6.0	V
I_{SB1}	Standby Current(1)	V_{CC}=5V, V_{IH}=V_{CC}-0.2V, V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS (T_{OPR}=-40°C to 85°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_x=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 (T_{OPR}=-40°C to 85°C, V_{SS}=0V)

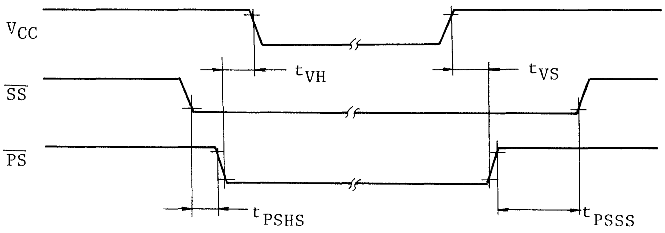
SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
V_{SB2}	Standby Voltage(2)		3.0	-	6.0	V
I_{SB2}	Standby Current(2)	V_{CC}=5V,V_{IH}=V_{CC}-0.2V,V_{IL}=0.2V	-	0.5	10	μA

AC CHARACTERISTICS (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 ($\overline{SS}$)		5	-	-	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_x=6MHz)

TIMING WAVEFORM

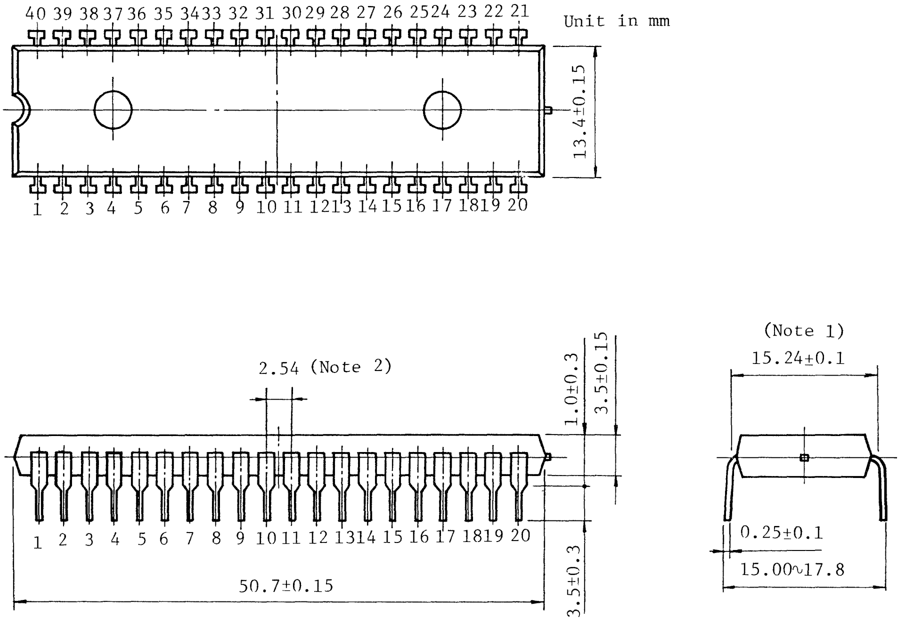


PIN STATUS IN THE POWER DOWN MODE

Each pin status in the power down mode (I)/(II) is shown in the following table.

PIN NAME	POWER DOWN MODE (I)	POWER DOWN MODE (II)
DB0 - DB7	High Impedance Input Disabled	High Impedance Input Disabled
P20 - P23	Output "0" (at EA=1) High Impedance (at EA=0) Input Disabled	Output PC_H
P24 - P27	High Impedance Input Disabled	Output the data Contained in the Port (Open Drain)
P10 - P17	High Impedance Input Disabled	Output the data Contained in the Port (Open Drain)
T_0	High Impedance Input Disabled	High Impedance Input Disabled
T_1	Input Disabled	Input Disabled
XTAL1 XTAL2	High Impedance	High Impedance
$\overline{RESET}$, $\overline{SS}$	Input Disabled when oscillator is stopped Pull-up transistor turned off	Input Disabled when oscillator is stopped Pull-up transistor turned off
$\overline{INT}$, EA	Input Disabled when oscillator is stopped	Input Disabled when oscillator is stopped
$\overline{RD}$, $\overline{WR}$	Output "1"	Output "1"
ALE	Output "0"	Output "1"
PROG	Output "1"	Output "1"
$\overline{PSEN}$	Output "0" (at EA=1) Output "1" (at EA=0)	Output "1"

OUTLINE DRAWING



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.42 leads.

東芝 INTEGRATEDCIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DIGITAL INTEGRAED CIRCUIT

TMP82C43P

SILICON MONOLITHIC CMOS SILICON GATE

INPUT/OUTPUT EXPANDER

GENERAL DESCRIPTION

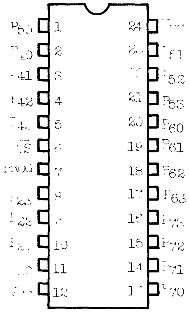
The TMP82C43P is an input/output expander designed specifically to provide a low cost means of I/O expansion for the TLCS-84C family.

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.

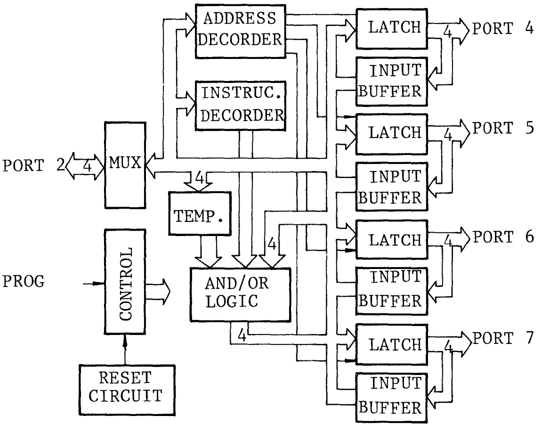
FEATURES

- ∘ 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.

$\overline{\text{CS}}$ (Input)

Chip Select Input. A high on $\overline{\text{CS}}$ 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

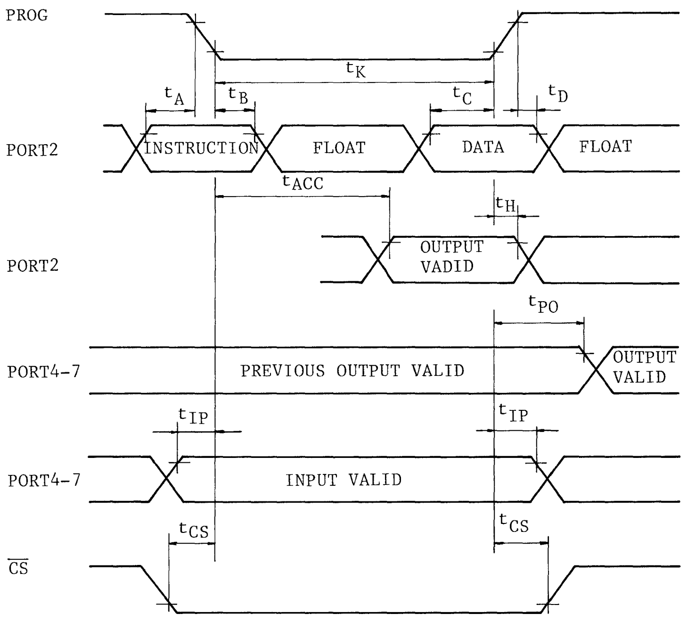
D.C. CHARACTERISTICS (Ⅱ) T_{OPR}=-40°C to 85°C, V_{CC}=5V±20%, V_{SS}=0V

SYMBOL	PARAMETER	TEST CONDITION	MIN.	TYP.	MAX.	UNITS
V_{IL}	Input Low Voltage	$4.0V \leq V_{CC} \leq 4.5V$	-0.5		$0.15V_{CC}$	V
V_{IH}	Input High Voltage	$5.5V \leq V_{CC} \leq 6.0V$	$0.5V_{CC}$		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 Before/After PROG		100			ns
t_{ACC}	Port 2 Valid After PROG	C_L=80pF			650	ns

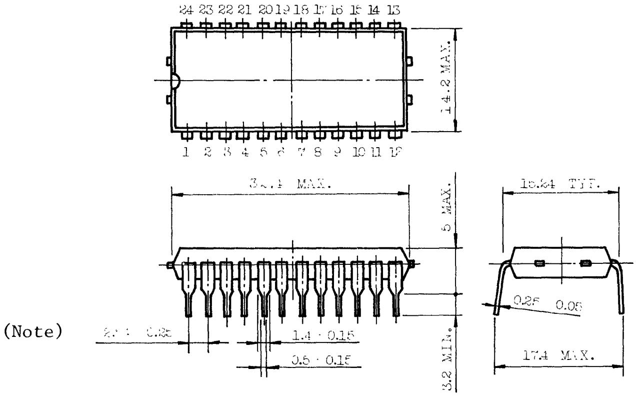
TIMING WAVEFORM



OUTLINE DRAWINGS

PLASTIC PACKAGE

Unit in mm



Note : 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.

8BIT MICROPROCESSOR

TLCS-85A(NMOS)

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)

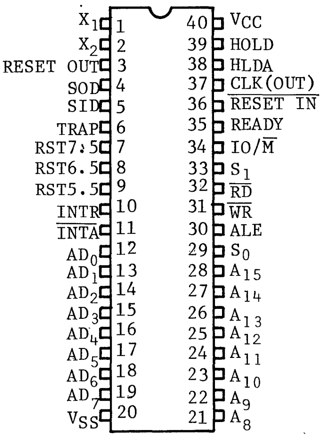


FIGURE 2. TMP8085A PINOUT DIAGRAM

BLOCK DIAGRAM

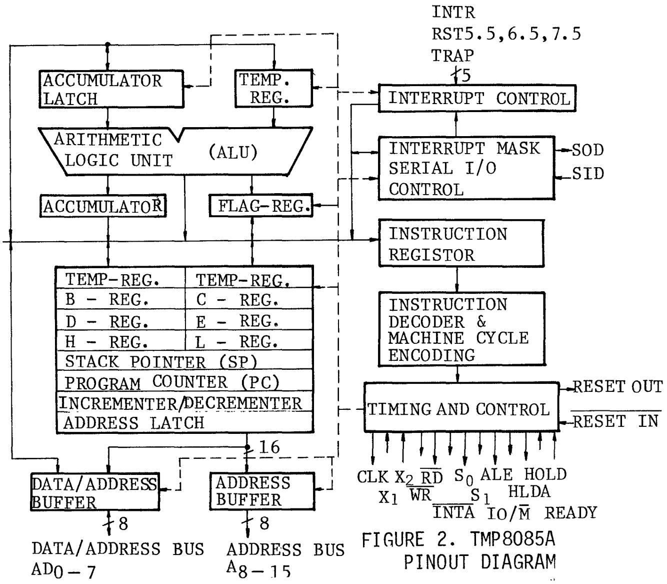


FIGURE 2. TMP8085A PINOUT DIAGRAM

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.

$\overline{\text{INTA}}$ (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) $\overline{\text{RD}}$ 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/$\overline{\text{M}}$ (Output)
Machine cycle status:

IO/$\overline{\text{M}}$	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/$\overline{\text{M}}$, 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 $\overline{\text{RD}}$ ($\overline{\text{INTA}}$) and $\overline{\text{WR}}$ 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	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
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
T1	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

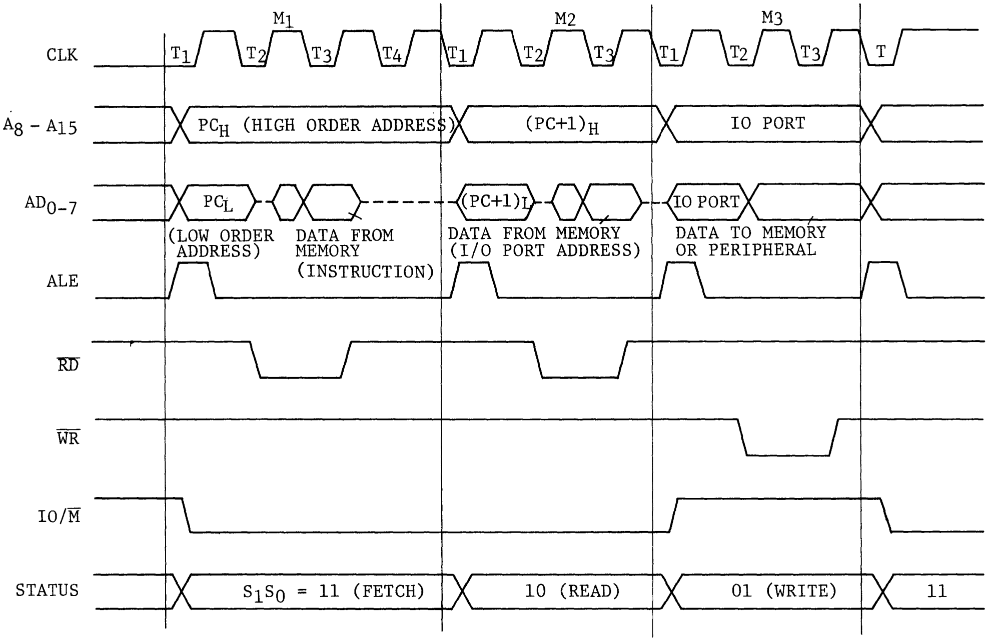


FIGURE 3. TMP8085A BASIC SYSTEM TIMING

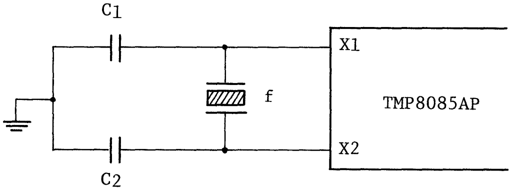
DRIVING THE X1 AND X2 INPUTS

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) ≤ 7 PF
- R_S (equivalent shunt resistance) ≤ 75 Ohms



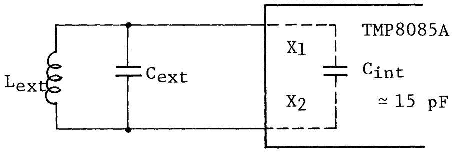
Note a value of the external capacitors C_1 and C_2 between X1, X2 and ground. In case of the crystal frequency above 4 MHz, it is recommended that you choose a value of 10pF for C_1 and C_2 and less than 4 MHz, 20pF capacitors are recommended.

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.

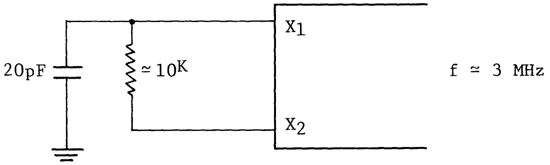
$$f = \frac{1}{2\pi \sqrt{L}\ (C_{ext} + C_{int})}$$

The use of an LC circuit is not recommended for frequencies higher than approximately 5 MHz.

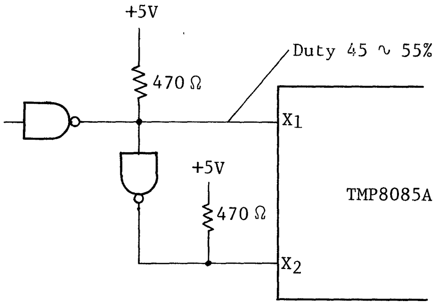


C. RC Circuit Clock Driver

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,rl,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

O — 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 B_2	(r) ← (B_2)	2	7	- - - - -
MVI M, d8	0 0 1 1 0 1 1 0 B_2	[(H)(L)] ← (B_2)	2	10	- - - - -
LDA addr	0 0 1 1 1 0 1 0 B_2 B_3	(A) ← [(B_3)(B_2)]	3	13	- - - - -
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 B_2 B_3	(L) ← [(B_3)(B_2)] (H) ← [(B_3)(B_2)+1]	3	16	- - - - -
LXI H, d16	0 0 1 0 0 0 0 1 B_2 B_3	(H) ← (B_3) (L) ← (B_2)	3	10	- - - - -
LXI D, d16	0 0 0 1 0 0 0 1 B_2 B_3	(D) ← (B_3) (E) ← (B_2)	3	10	- - - - -
LXI B, d16	0 0 0 0 0 0 0 1 B_2 B_3	(B) ← (B_3) (C) ← (B_2)	3	10	- - - - -
LXI SP, d16	0 0 1 1 0 0 0 1 B_2 B_3	$(SP)_H$ ← (B_3) $(SP)_L$ ← (B_2)	3	10	- - - - -
SHLD addr	0 0 1 0 0 0 1 0 B_2 B_3	[(B_3)(B_2)] ← (L) [(B_3)(B_2)+1] ← (H)	3	16	- - - - -
STA addr	0 0 1 1 0 0 1 0 B_2 B_3	[(B_3)(B_2)] ← (A)	3	13	- - - - -

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)	1	4	-	-	-	-	-
									(L) ← (E)							
XTHL	1	1	1	0	0	0	1	1	(L) ← [(SP)]	1	16	-	-	-	-	-
									(H) ← [(SP)+1]							
IN addr8	1	1	0	1	1	0	1	1	(A) ← (data)	2	10	-	-	-	-	-
				B_2												
OUT addr8	1	1	0	1	0	0	1	1	(data) ← (A)	2	10	-	-	-	-	-
				B_2												

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	(PC) ← $(B_3)(B_2)$	3	10	-	-	-	-	-
				B_2												
				B_3												
JNZ addr	1	1	0	0	0	0	1	0	If Z = 0	3	7/10	-	-	-	-	-
				B_2					(PC) ← $(B_3)(B_2)$,							
				B_3					If Z = 1							
									(PC) ← (PC) + 3							
JZ addr	1	1	0	0	1	0	1	0	If Z = 1	3	7/10	-	-	-	-	-
				B_2					(PC) ← $(B_3)(B_2)$,							
				B_3					If Z = 0							
									(PC) ← (PC) + 3							
JNC addr	1	1	0	1	0	0	1	0	If C = 0	3	7/10	-	-	-	-	-
				B_2					(PC) ← $(B_3)(B_2)$,							
				B_3					If C = 1							
									(PC) ← (PC) + 3							
JC addr	1	1	0	1	1	0	1	0	If C = 1	3	7/10	-	-	-	-	-
				B_2					(PC) ← $(B_3)(B_2)$,							
				B_3					If C = 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
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 B_2 B_3	If C = 0, the actions specified in the CALL instruction are performed. If C = 1 (PC) ← (PC) + 3	3	9/18	- - - - -
CC addr	1 1 0 1 1 1 0 0 B_2 B_3	If C = 1, the actions specified in the CALL instruction are performed. If C = 0 (PC) ← (PC) + 3	3	9/18	- - - - -
CPO addr	1 1 1 0 0 1 0 0 B_2 B_3	If P = 0, the actions specified in the CALL instruction are performed. If P = 1 (PC) ← (PC) + 3	3	9/18	- - - - -
CPE addr	1 1 1 0 1 1 0 0 B_2 B_3	If P = 1, the actions specified in the CALL instruction are performed. If P = 0 (PC) ← (PC) + 3	3	9/18	- - - - -
CP addr	1 1 1 1 0 1 0 0 B_2 B_3	If S = 0, the actions specified in the CALL instruction are performed. If S = 1 (PC) ← (PC) + 3	3	9/18	- - - - -

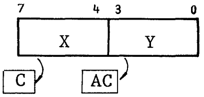
Mnemonic	Instruction Code								Operation	Bytes	States	Flag				
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				C	Z	S	P	AC
CM addr	1	1	1	1 B_2 B_3	1	1	0	0	If S = 1, the actions specified in the CALL instruction are performed. If S = 0 (PC) ← (PC) + 3	3	9/18	-	-	-	-	-
RET	1	1	0	0	1	0	0	1	(PCL) ← [(SP)] (PCH) ← [(SP)+1] (SP) ← (SP) + 2	1	10	-	-	-	-	-
RNZ	1	1	0	0	0	0	0	0	If Z = 0, the actions specified in the RET instruction are performed. If Z = 1 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RZ	1	1	0	0	1	0	0	0	If Z = 1, the actions specified in the RET instruction are performed. If Z = 0 (PC) ← (PC) + 1	1	6/12	-	-	-	-	-
RNC	1	1	0	1	0	0	0	0	If C = 0, the actions specified in the RET instruction are performed. If C = 1 (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
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				
	D7	D6	D5	D4	D3	D2	D1	D0				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				
	D7	D6	D5	D4	D3	D2	D1	D0				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 B2	0	1	1	0	(A) ← (A) + (B2)	2	7	0	0	0	0	0
ACI d8	1	1	0	0 B2	1	1	1	0	(A) ← (A)+(B2)+(C)	2	7	0	0	0	0	0
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 B2	0	1	1	0	(A) ← (A) - (B2)	2	7	0	0	0	0	0
SBI d8	1	1	0	1 B2	1	1	1	0	(A) ← (A) - (B2) - (C)	2	7	0	0	0	0	0

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 $\geqq$ 10 or AC=1, $(A) \leftarrow (A) + 6$ 2. If X $\geqq$ 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	$(A) \leftarrow (A) \wedge (r)$	1	4	R	0	0	0	S
ANA M	1	0	1	0	0	1	1	0	$(A) \leftarrow (A) \wedge [(H)(L)]$	1	7	R	0	0	0	S
ANI d8	1	1	1	0	0	1	1	0	$(A) \leftarrow (A) \wedge (B_2)$	2	7	R	0	0	0	S
				B_2												
XRA r	1	0	1	0	1	S	S	S	$(A) \leftarrow (A) \forall (r)$	1	4	R	0	0	0	R
XRA M	1	0	1	0	1	1	1	0	$(A) \leftarrow (A) \forall [(H)(L)]$	1	7	R	0	0	0	R
XRI d8	1	1	1	0	1	1	1	0	$(A) \leftarrow (A) \forall (B_2)$	2	7	R	0	0	0	R
				B_2												
ORA r	1	0	1	1	0	S	S	S	$(A) \leftarrow (A) \vee (r)$	1	4	R	0	0	0	R
ORA M	1	0	1	1	0	1	1	0	$(A) \leftarrow (A) \vee [(H)(L)]$	1	7	R	0	0	0	R
ORI d8	1	1	1	1	0	1	1	0	$(A) \leftarrow (A) \vee (B_2)$	2	7	R	0	0	0	R
				B_2												
CMP r	1	0	1	1	1	S	S	S	$(A) - (r)$	1	4	0	0	0	0	0
CMP M	1	0	1	1	1	1	1	0	$(A) - [(H)(L)]$	1	7	0	0	0	0	0
CPI d8	1	1	1	1	1	1	1	0	$(A) - (B_2)$	2	7	0	0	0	0	0
				B_2												
CMA	0	0	1	0	1	1	1	1	$(A) \leftarrow (\overline{A})$	1	4	-	-	-	-	-
RLC	0	0	0	0	0	1	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7)$ $(C) \leftarrow (A_7)$	1	4	0	-	-	-	-
RRC	0	0	0	0	1	1	1	1	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$ $(C) \leftarrow (A_0)$	1	4	0	-	-	-	-
RAL	0	0	0	1	0	1	1	1	$(A_{n+1}) \leftarrow (A_n)$ $(C) \leftarrow (A_7)$ $(A_0) \leftarrow (C)$	1	4	0	-	-	-	-
RAR	0	0	0	1	1	1	1	1	$(A_n) \leftarrow (A_{n+1})$ $(C) \leftarrow (A_0)$ $(A_7) \leftarrow (C)$	1	4	0	-	-	-	-

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] ← $D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$ 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 $70°C$, $V_{CC} = 5V \pm 5\%$

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_{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

TA = 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	C_L=150pF t_{CYC}=320ns	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 $\overline{\text{READ}}$ $\overline{\text{(INTA)}}$				0	ns
t_{AL}	A8-15 Valid before Trailing Edge of ALE[1]		115			ns
t_{ALL}	A0-7 Valid before Trailing Edge of ALE		90			ns
t_{ARY}	READY Valid from Address Valid				220	ns
t_{CA}	Address (A8 - A15) Valid after Control		120			ns
t_{CC}	Width of Control Low ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{INTA}}$) 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 $\overline{\text{WRITE}}$		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 $\overline{\text{READ}}$ to Re-Enabling of Address		150			ns
t_{RD}	$\overline{\text{READ}}$ (or $\overline{\text{INTA}}$)to Valid Data				300	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control		400			ns
t_{RDH}	Data Hold Time After $\overline{\text{READ}}$ $\overline{\text{INTA}}$		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 $\overline{\text{WRITE}}$		100			ns
t_{WDL}	LEADING Edge of $\overline{\text{WRITE}}$ to Data Valid				40	ns

Notes:

1. A_{8-15} address specs apply to IO/$\overline{\text{M}}$, S_0 and S_1 except A_{8-15} are undifined during T_4 - T_6 of OF cycle whereas IO/$\overline{\text{M}}$, 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}

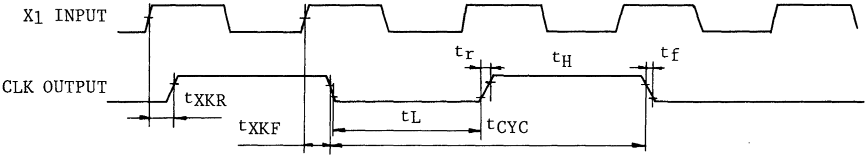


FIGURE 4. CLOCK TIMING WAVEFORM

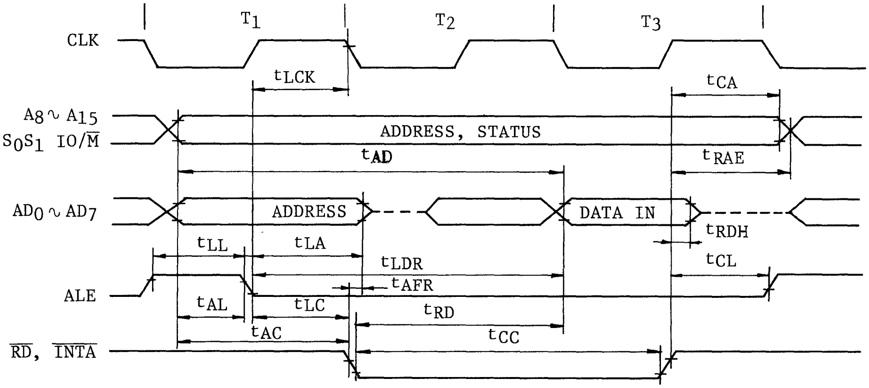


FIGURE 5. READ OPERATION

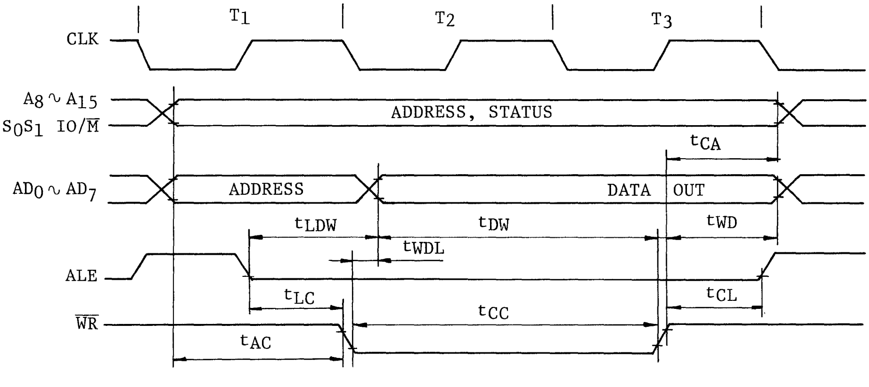


FIGURE 6. WRITE OPERATION

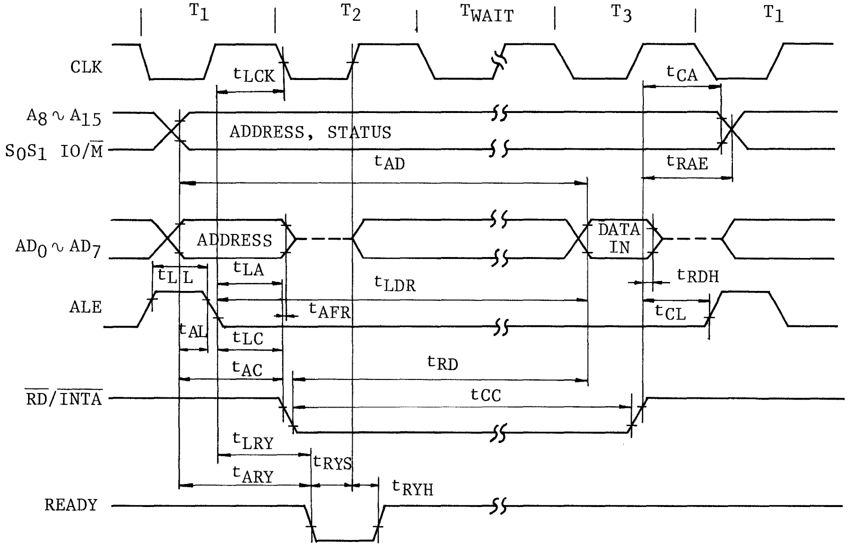


FIGURE 7. READ OPERATION WITH WAIT CYCLE (TYPICAL)
- SAME READY TIMING APPLIES TO WRITE OPERATION

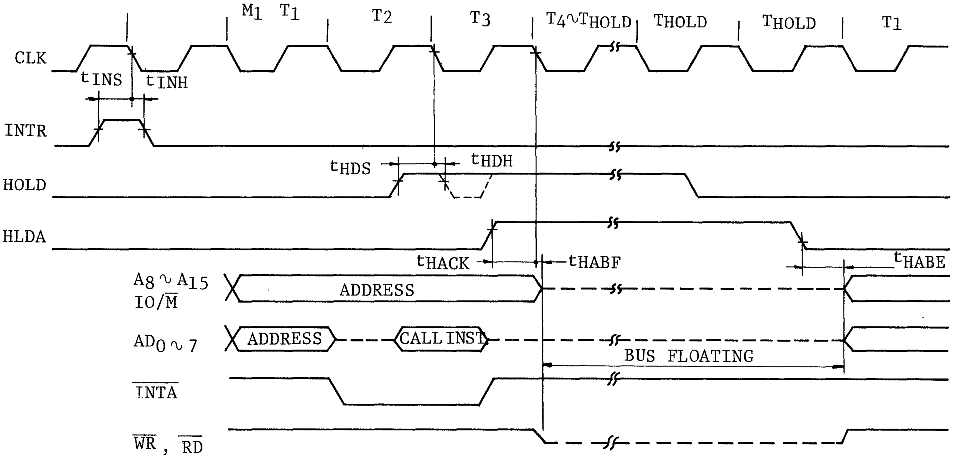
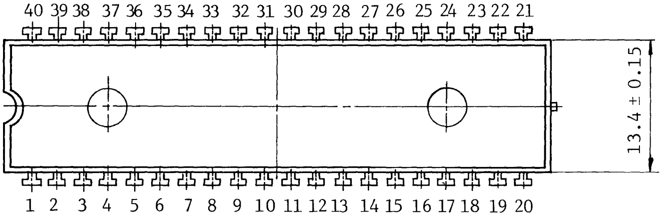


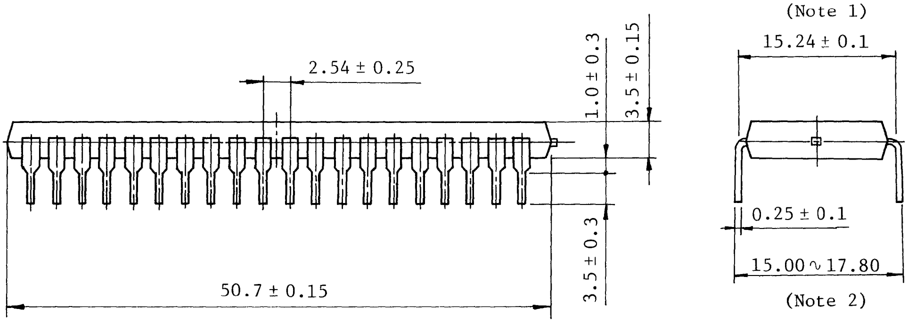
FIGURE 8. INTERRUPT AND HOLD TIMING

OUTLINE DRAWING

40 Pins Plastic Package

Unit in mm





Notes:
1. This dimension shows the center of curvature of leads
2. This dimension shows spread of leads.
3. All dimensions are in millimeters.

8BIT MICROPROCESSOR

PERIPHERALS(NMOS)

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)

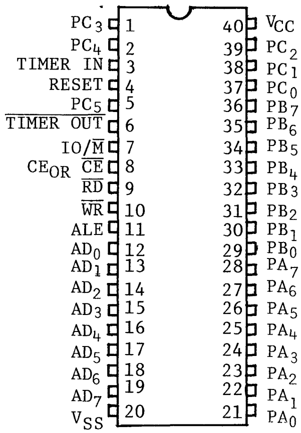


FIGURE 1 TMP8155P/8156P PINOUT DIAGRAM

BLOCK DIAGRAM

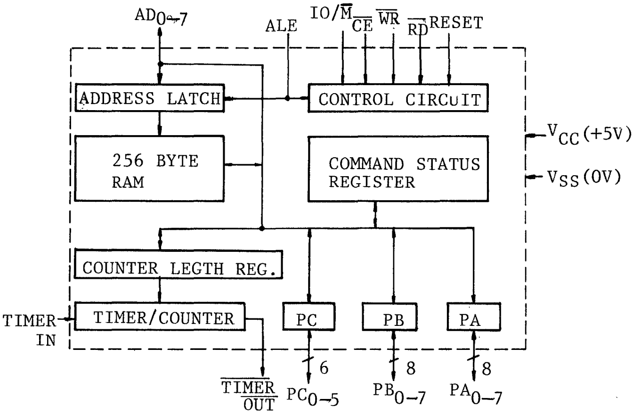


FIGURE 2 TMP8155P/8156P FUNCTIONAL BLOCK DIAGRAM

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)

$\overline{\text{IO/Memory}}$ 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

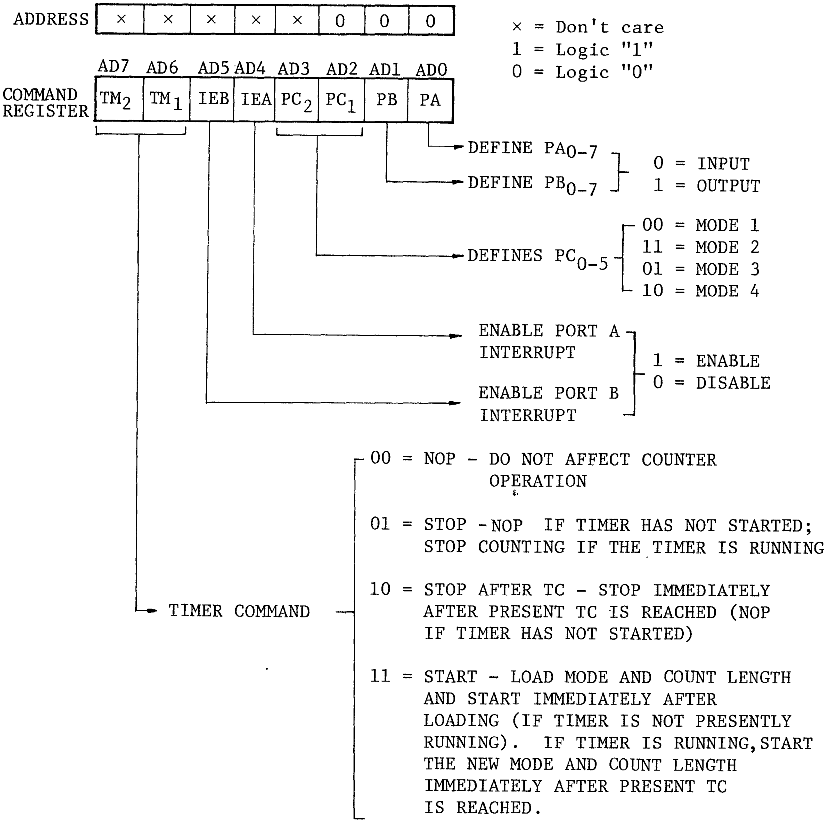
V_{SS} (Power)
Ground Reference

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.



00 = NOP - DO NOT AFFECT COUNTER OPERATION

01 = STOP -NOP IF TIMER HAS NOT STARTED; STOP COUNTING IF THE TIMER IS RUNNING

10 = STOP AFTER TC - STOP IMMEDIATELY AFTER PRESENT TC IS REACHED (NOP IF TIMER HAS NOT STARTED)

11 = 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.

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.

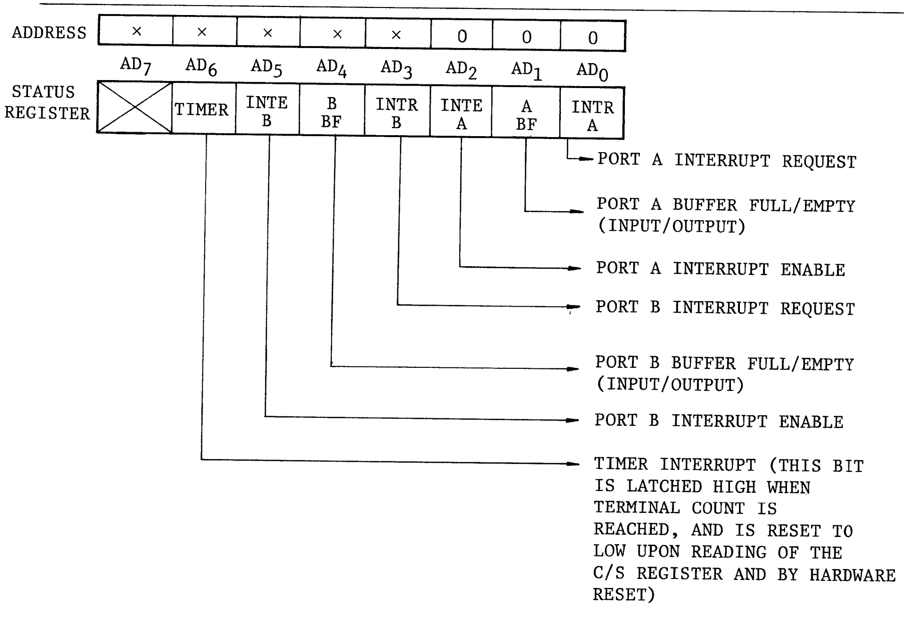

ADDRESS
×
×
×
×
×
0
0
0
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
STATUS REGISTER
TIMER
INTE B
B BF
INTR B
INTE A
A BF
INTR A
PORT A INTERRUPT REQUEST
PORT A BUFFER FULL/EMPTY (INPUT/OUTPUT)
PORT A INTERRUPT ENABLE
PORT B INTERRUPT REQUEST
PORT B BUFFER FULL/EMPTY (INPUT/OUTPUT)
PORT B INTERRUPT ENABLE
TIMER INTERRUPT (THIS BIT IS LATCHED HIGH WHEN TERMINAL COUNT IS REACHED, AND IS RESET TO LOW UPON READING OF THE C/S REGISTER AND BY HARDWARE RESET)

FIGURE 4 STATUS REGISTER BIT ASSIGNMENT

INPUT/OUTPUT SECTION

COMMAND/STATUS REGISTER (C/S)

Both register have the common address xxxxx000. 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 AD2 and AD3 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:

CONTROL / MODE	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 $\overline{STB}$(Port A strobe)	A $\overline{STB}$ (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 $\overline{STB}$ (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.

M_2	M_1	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8

TIMER MODE   MSB OF COUNT LENGTH

T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0

LSB OF COUNT LENGTH

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 equal s 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.

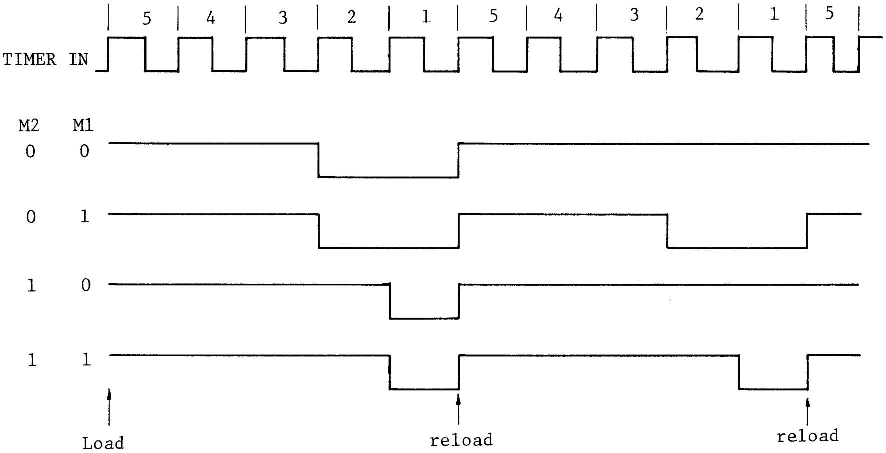


FIGURE 6 ASYMMETRICAL SQUARE-WAVE OUTPUT RESULTING FROM COUNT OF 5

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 $+70°C$, $V_{CC} = +5V \pm 5\%$

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	$V_{IN} = V_{CC}$ to 0V			±10	μA
I_{LO}	Output Leakage Current	$0.45V \leq V_{OUT} \leq V_{CC}$			±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

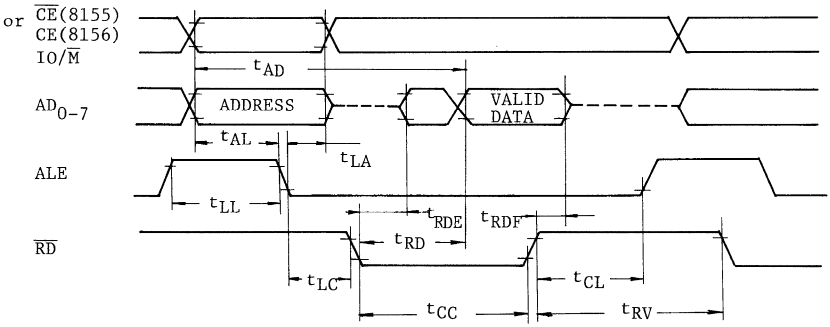
A.C. CHARACTERISTICS

TA=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

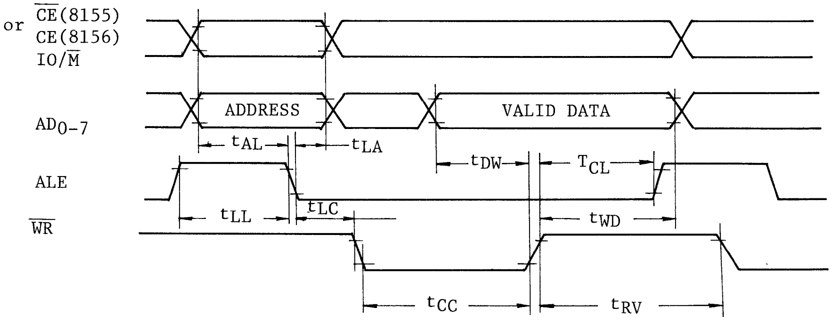
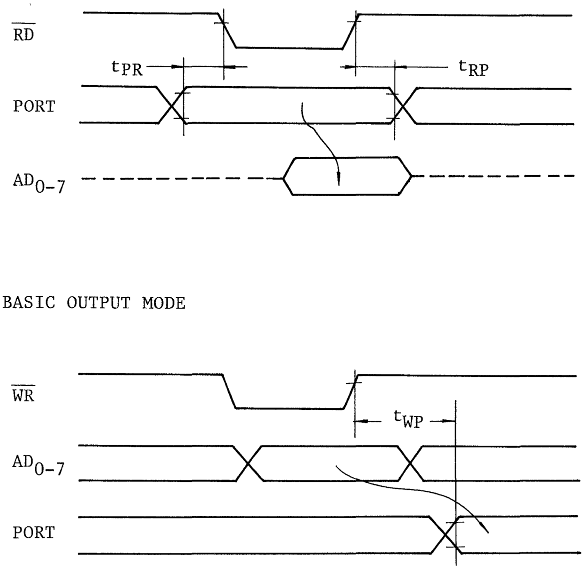


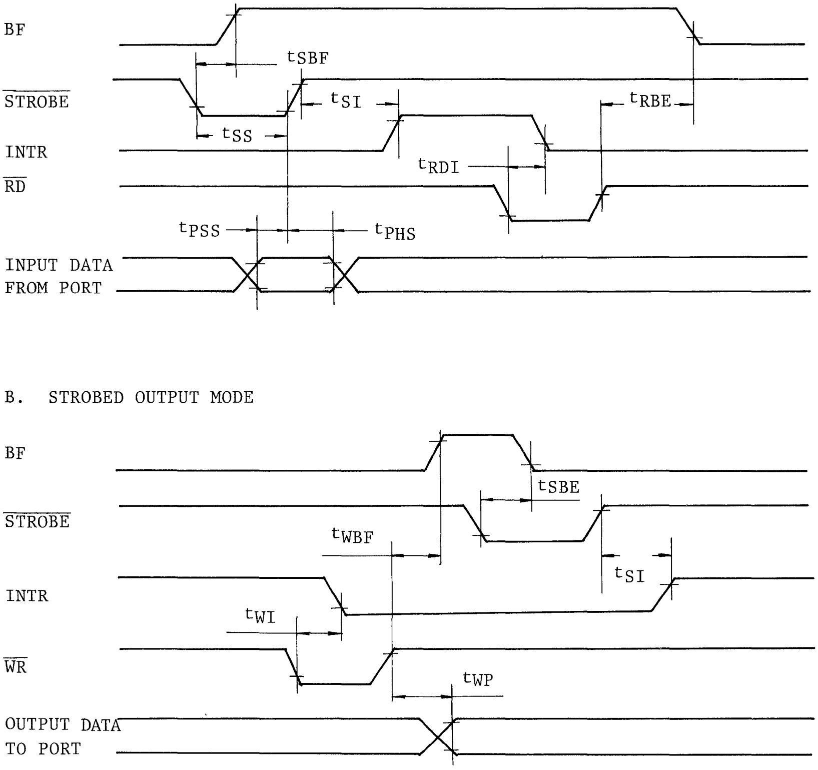
FIGURE 7 READ/WRITE TIMING DIAGRAMS



A. BASIC INPUT MODE

B. BASIC OUTPUT MODE

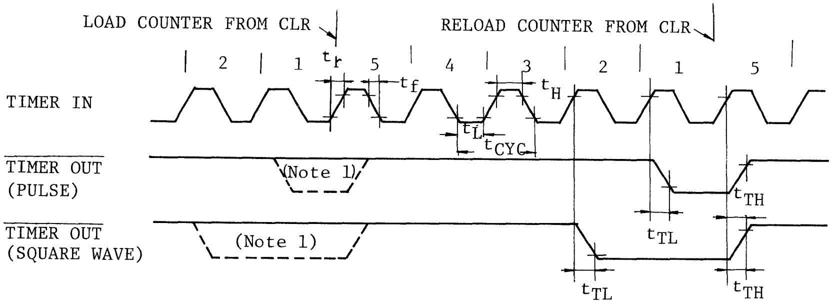
FIGURE 8 BASIC I/O TIMING WAVEFORM



A. STROBED INPUT MODE

B. STROBED OUTPUT MODE

FIGURE 9 STROBED I/O TIMING WAVEFORM



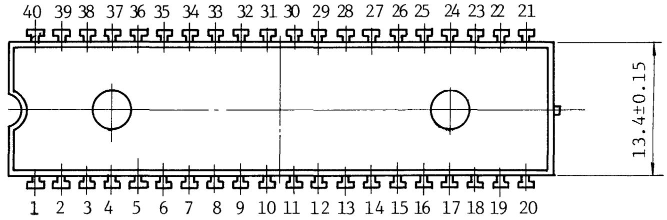
Note 1: The timer output is periodic if in an automatic reload mode (M_1 Mode Bit = 1)

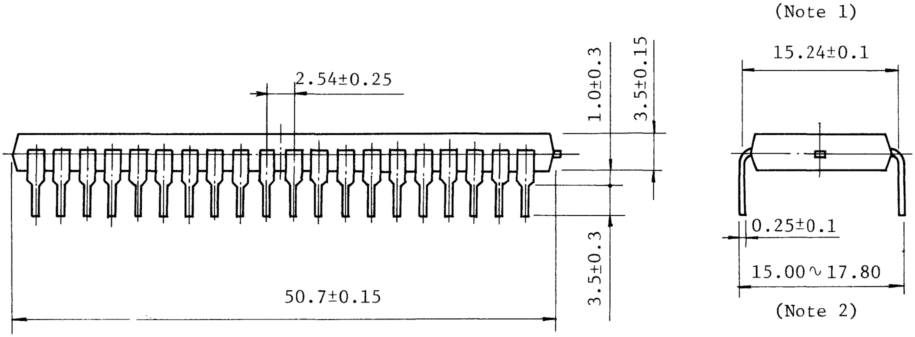
FIGURE 10 TIMER OUTPUT WAVEFORM COUNTDOWN FROM 5 TO 1

OUTLINE DRAWING

40 Pins Plastic Package

Unit in mm





Note: 1. This dimension shows the center of curvature of leads.

2. This dimension shows spread of leads.

3. All dimensions are in millimeters.

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)

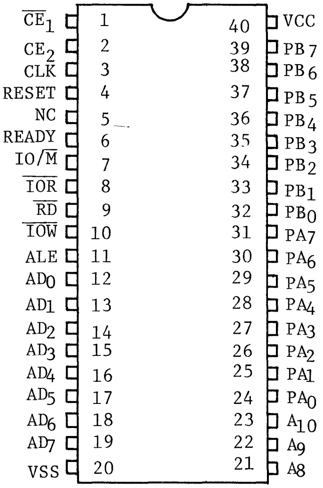


FIGURE 1 TMP8355P PINOUT DIAGRAM

BLOCK DIAGRAM

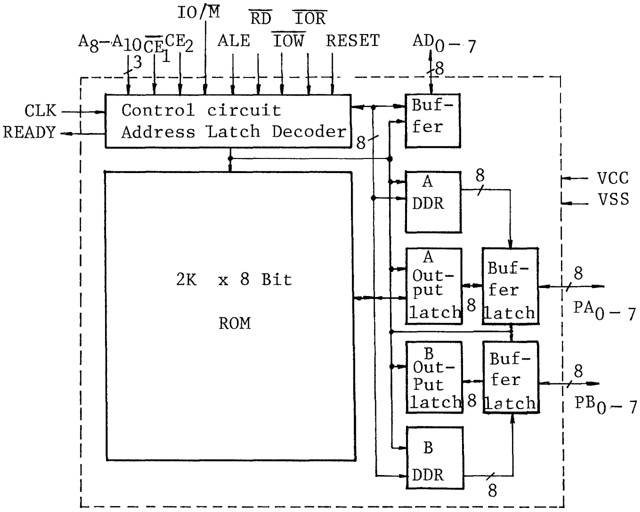


FIGURE 2 TMP8355P FUNCTIONAL BLOCK DIAGRAM

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 cirectly 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 $\overline{RD}$ 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 ± 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			± 10	μA
I_{LO}	Output Leakage Current	$0.45 \leq Vout \leq V_{CC}$			± 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

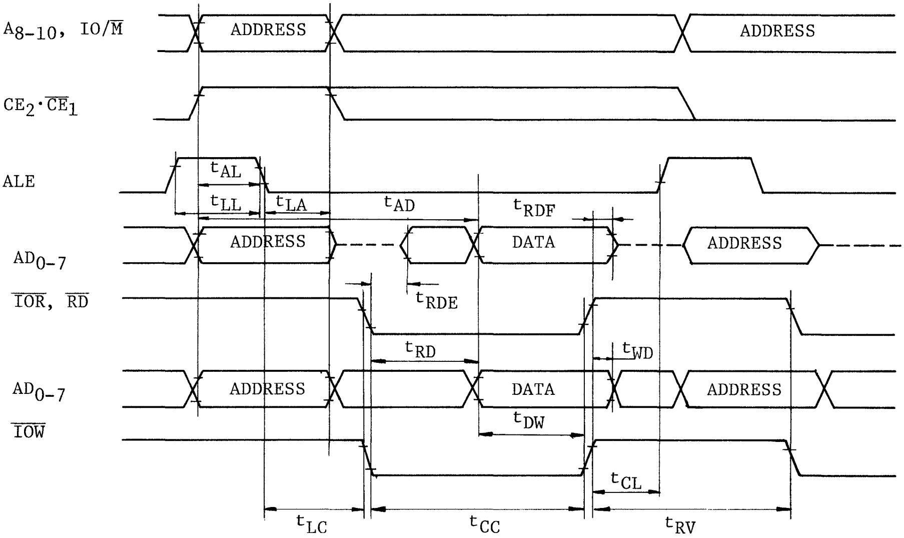


FIGURE 3 PROM READ, I/O READ, AND WRITE TIMING

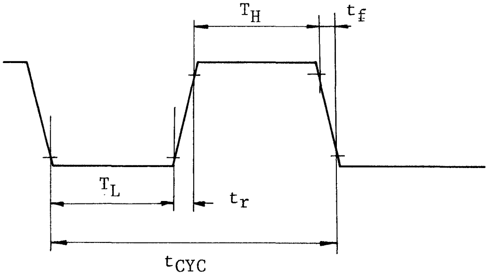


FIGURE 4 CLOCK SPECIFICATION FOR TMP8355P

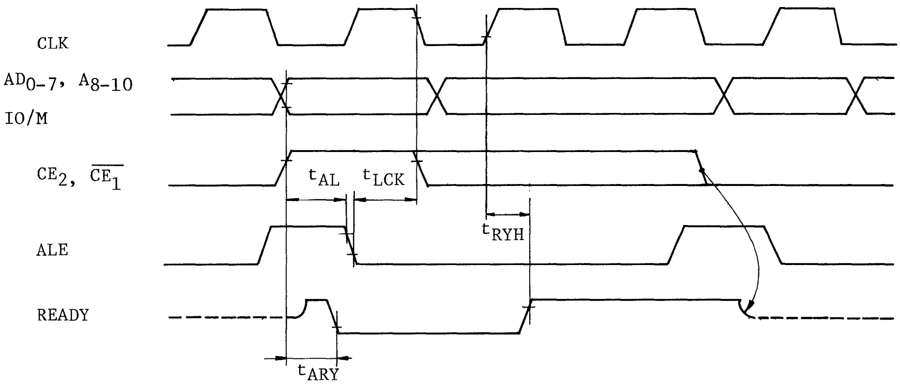


FIGURE 5 WAIT STATE TIMING (READY = 0)

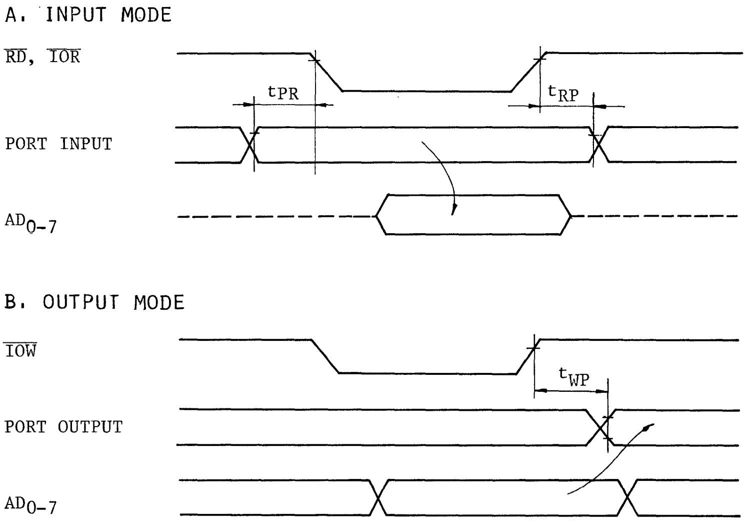
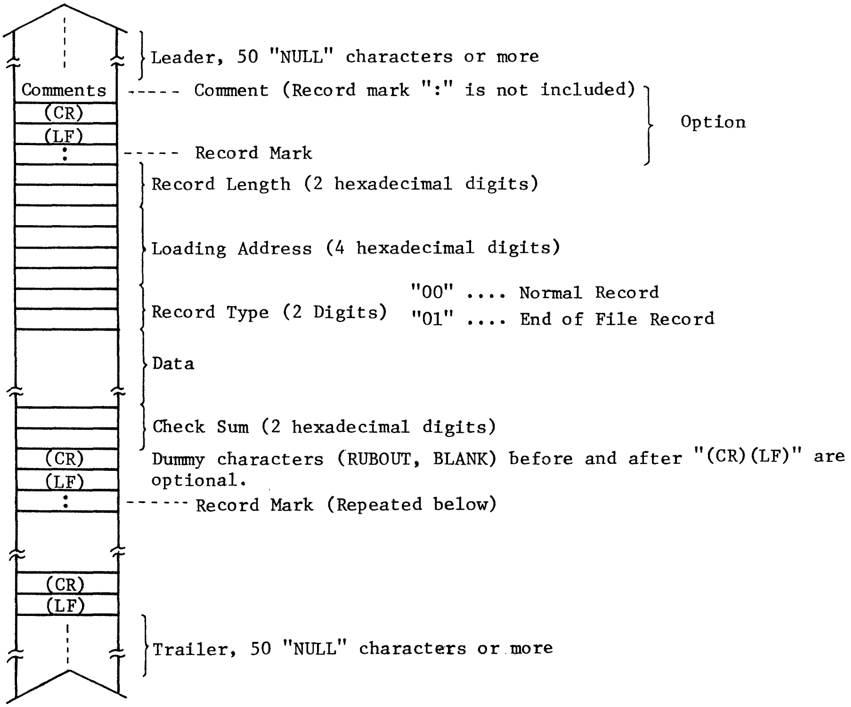


FIGURE 6 I/O PORT TIMING

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



Leader, 50 "NULL" characters or more

----- Comment (Record mark ":" is not included)

----- Record Mark

Option

Record Length (2 hexadecimal digits)

Loading Address (4 hexadecimal digits)

Record Type (2 Digits) "00" .... Normal Record
"01" .... End of File Record

Data

Check Sum (2 hexadecimal digits)

Dummy characters (RUBOUT, BLANK) before and after "(CR)(LF)" are optional.

------ Record Mark (Repeated below)

Trailer, 50 "NULL" characters or more

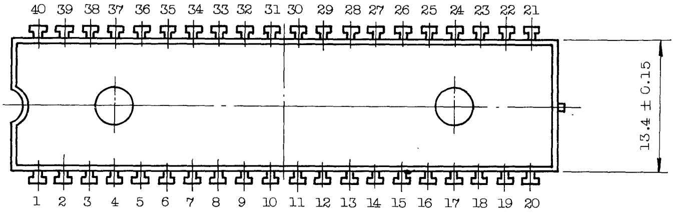
(2) Example of Tape List

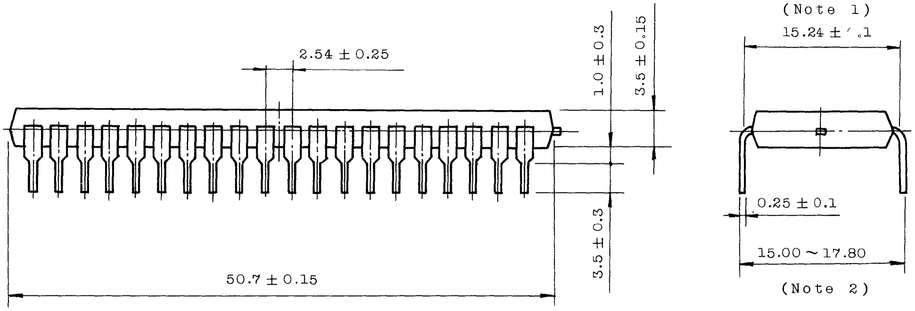
```
TOSHIBA MICRO COMPUTER TLCS-84
:100000000665C7D79CF50F3F951FED55A8FF16E570
:1000100088884DDE67D31F5D8ABA6DF292F113F5C1
:100020004FF1FB5DFFDAA96A99CF7DF94A346B7C09
:10003000197352F729F12F79AA9C057C5B851EED77
                  .
                  .
                  .
                  .
                  .
                  .
                  .
                  .
                  .
                  .
                  .
:1003C0005DFDB5E556A67277F61A51C631CF9F0E80
:1003D000BD2F6F20E8BB1977E3FB5AD1F41FDAA7E2
:1003E000B53D42E0EC32546025B7308CDD52063D1D
:1003F000B4BE9E9E345B6138060B20VC372BF60BD6
:00000001FF
```

OUTLINE DRAWING

40 Pins Plastic Package

Unit in mm





Note: 1. This dimension shows the center of curvature of leads.

2. This dimension shows spread of leads.

3. All dimensions are in millimeters.

Toshiba 東芝 INTEGRATEDCIRCUIT

TECHNICAL DATA

TOSHIBA MOS DIGITAL INTEGRATED CIRCUIT

TMP8755AC

N-CHANNEL SILICON TATE MOS

16,384 BIT EPROM WITH I/O PORTS

GENERAL DESCRIOTION

The TMP8755AC is an erasable and electrically reprogrammable ROM (EPROM) and I/O chip to be used in the TLCS-85A microcomputer system. The PROM portion is organized as 2,048 words by 8 bits.

The I/O portion consists of two general purpose I/O ports. Each I/O port has eight port lines, and each I/O port line is individually programmable as input or output.

FEATURES

- U.V. Erasable and Electrically Reprogrammable ROM (2,048 x 8)
- Single +5V Power Supply (V_{CC})
- Internal Address Latch
- 2 General Purpose 8 bit I/O Ports
- Access Time: 450 ns (MAX.)
- Each I/O Port Line ,Individually programmable as Input or Output.
- Multiplexed Address and Data Bus
- 40 pin DIP
- Compatible wih Intel's 8755A

PIN CONNECTIONS (TOP VIEW)

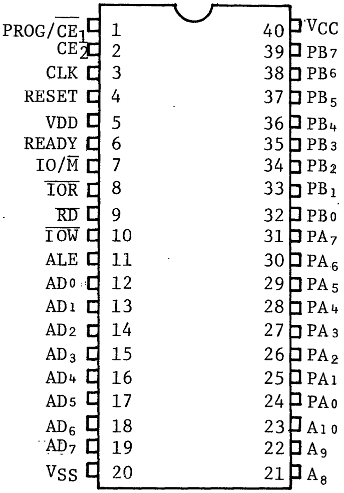


FIGURE 1 TMP8755AC PINOUT DIAGRAM

BLOCK DIAGRAM

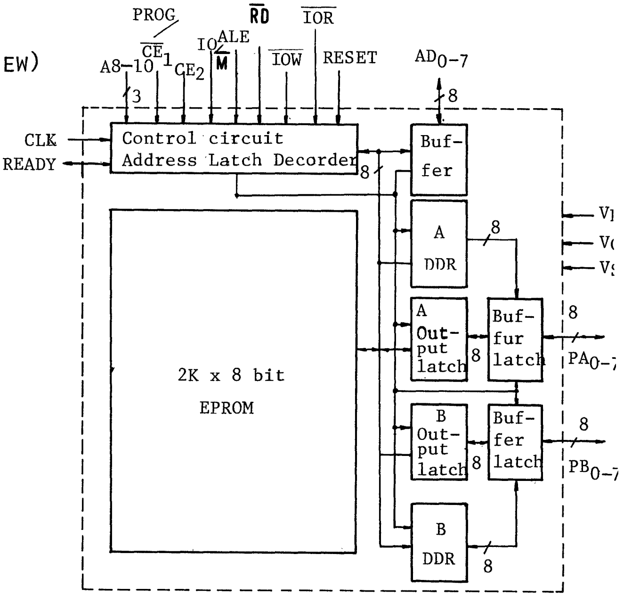


FIGURE 2 TMP8755AC FUNCTIONAL BLOCK DIAGRAM

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) are latched in at the trailing edge of ALE.

AD_{0-7} (Input/Output, 3-state)

Bi-directional Address/Data bus. The lower 8-bit of the PROM 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.

A8-10 (Input)

These are the high order bits of the PROM address. They do not affect I/O operations.

PROG/$\overline{CE1}$, 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 PROM. $\overline{CE}_1$ is also used as a programming pin.

$IO/\overline{M}$ (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 PROM.

$\overline{RD}$ (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 PROM location or I/O ports. 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, AD_1.
Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD_0 and AD_1 low, or IO/$\overline{M}$ high, RD low, active Chip Enables, and AD_0 and AD_1 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 and a 0 from AD_1.

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 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

V_{DD} (Power)

V_{DD} is a programming voltage and must be tied to +5V when the TMP8755AC is being read. For programming, a high voltage is supplied with V_{DD}=25V, typical.

FUNCTIONAL DESCRIPTION

PROM SECTION

The TMP8755AC contains an 8-bit address latch which allows it to interface directly to TLCS-85A microcomputer system without additional hardware.
The PROM portion of the chip is addressed by the 11-bit address (A8-10,AD_{0-7}) and CE. The address, $IO/\overline{M}$ and CE are latched into the address latches on the 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 EPROM location addressed by latched address are output onto the AD_{0-7} lines.

I/O SECTION

The I/O port portion consists 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 those of DDR's cannot be read. A port can be read out when the latched Chip Enables are active and either $\overline{RD}$ 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 correspinding 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 summarizes 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)

ERASURE CHARATERISTICS

The TMP8755AC can be erased by applying light with wavelengths shorter than 4000 Å. (1Å = 10^{-8} cm). Sunlight and the fluorescent lamps may include 3000 ∿ 4000 Å wavelength components consequently when used under such lighting for extended periods of time, an opaque seal will be required to protect the TMP8755AC. Generally, ultraviolet light with a wavelength of 2537 Å is recommended for TMP8755AC-erasing, and in this case the integrated does (ultraviolet light intensity [W/cm^2] × time [sec]) should be over 15 [W • sec /cm^2].

If the Toshiba sterilizing lamp GL-15 is used and the device is exposed at a distance of 1-cm from the lamp surface, erasure should be completed in about 60 minutes.

And using a lamp whose ultraviolet light intensity is a 12000 [μW/cm^2] will reduce the exposure time to about 20 minutes. (In this case the integrated dose should be 12000 [μW/cm^2] x (20 x 60)[sec] ≅ 15 [W · sec/cm^2].)

PROGRAMMING

Initially when received by customers all bits of the TMP8755AC are in the "1" state which is the erased state. Therefore programming is carried out by electrically writing in the "0" state at the desired bit locations.
A programmed "0" can only be changed to a "1" by ultraviolet erasure.
The program mode itself consists of programming a single address at a time, giving a single 50 ms pulse for every address.
Preliminary timing diagrams and parameter values pertaining to the TMP8755AC programming operation are contained in Figure 7.

ABSOLUTE MAXIMUM RATINGS

Symbol	Item	Rating
V_{CC}	V_{CC} Supply Voltage with Respect to V_{SS}	-0.5V to +7.0V
V_{DD}	V_{DD} Supply Voltage for Programming with Respect to V_{SS}	-0.5V to +26.5V
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	1.5 W
T_{SOLDER}	Seldering 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

$TA = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = V_{CC} \pm 0.6V$

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	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input Leakage Current	$V_{IN} = V_{CC}$ to 0V			±10	μA
I_{LO}	Output Leakage Current	$0.45 \leq V_{out} \leq V_{CC}$			±10	μA
I_{CC}	V_{CC} Supply Current				180	mA

A.C. CHARACTERISTICS

TA = 0℃ to 70℃ , V_{CC} = 5V ± 5% V_{DD} = V_{CC} ± 0.6V

Symbol	Parameter	Test Condition	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			450	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		30			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_{LD}	ALE to Data Out Valid				350	ns
t_{LCK}	ALE Low during CLK high		100			ns

D.C. CHARACTERISTICS FOR PROGRAMMING

TA = 25℃ ± 5℃ , V_{CC} = 5V ± 5% , V_{SS} = 0V

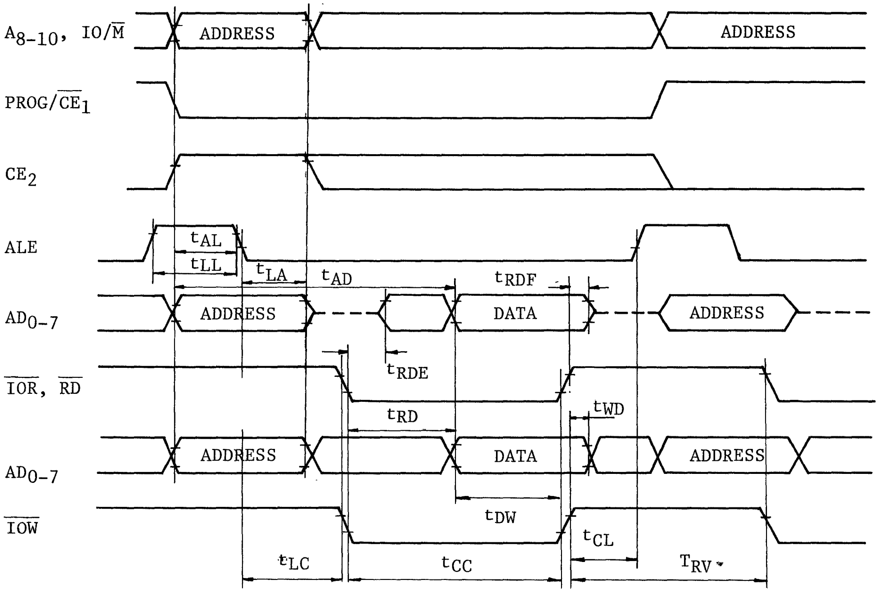
Symbol	Parameter	Min.	Typ.	Max.	Units
V_{DD}	Supply Voltage for programming	24	25	26	V
I_{DD}	Supply Current			30	mA

A.C. CHARACTERISTICS FOR PROGRAMMING

TA = 25℃ ± 5℃ , V_{CC} = 5V ±5% , V_{SS} = 0V

Symbol	Parameter	Min.	Typ.	Max.	Units
t_{PS}	Data Set Up Time	10			ns
t_{PD}	Data Hold Time	0			ns
t_S	Program Pulse Set Up Time	2			µs
t_H	Program Pulse Hold Time	2			µs
t_{PR}	Program Pulse Rise Time	10	2000		ns
t_{PF}	Program Pusle Fall Time	10	2000		ns
t_{PRG}	Program Pulse Width	45	50		ms

TIMING WAVEFORMS



Note: $\overline{CE_1}$ must remain low for the entire cycle.

FIGURE 3 PROM READ, I/O READ, AND WRITE TIMING

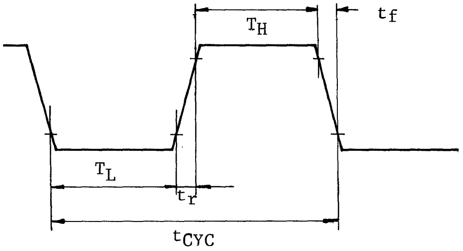


FIGURE 4 CLOCK SPECIFICATION FOR TMP8755AC

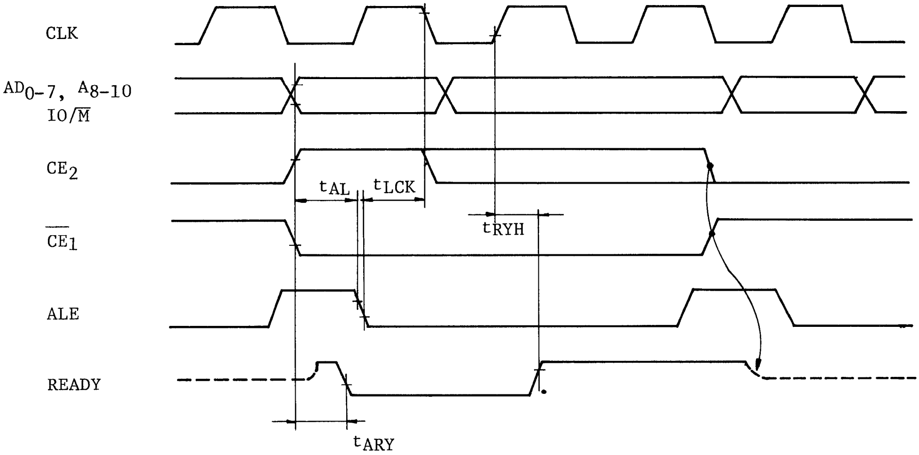
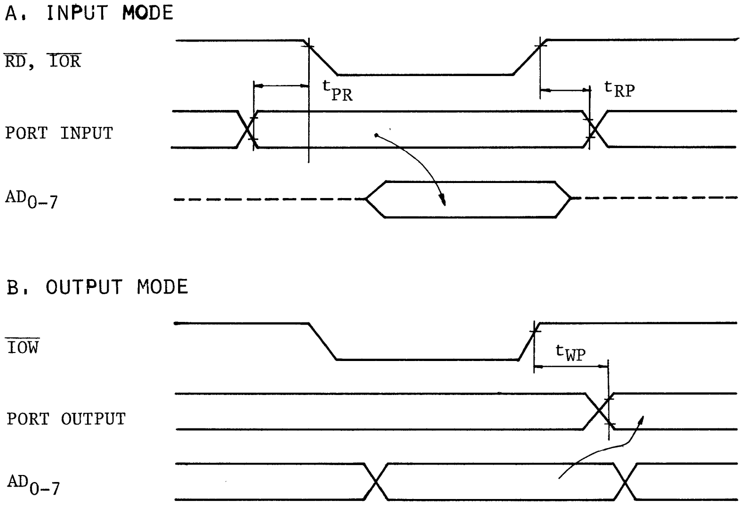
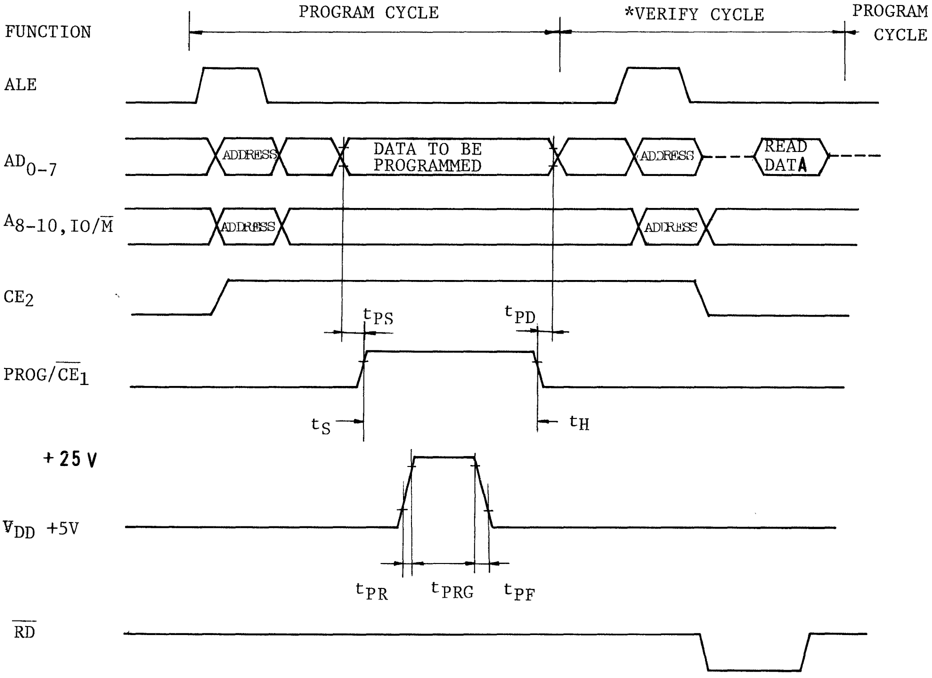


FIGURE 5 WAIT STATE TIMING (READY= 0)


A. INPUT MODE

B. OUTPUT MODE

FIGURE 6 I/O PORT TIMING



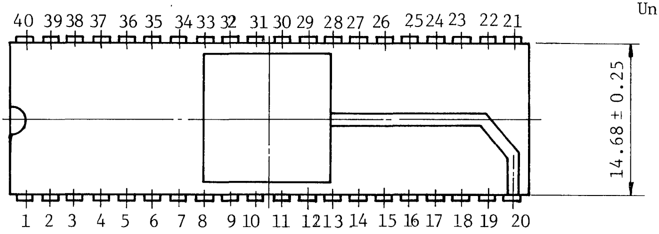
* VERIFY CYCLE IS A REGULAR MEMORY READ CYCLE
(WITH V_{DD} = 5V FOR TMP8755AC)

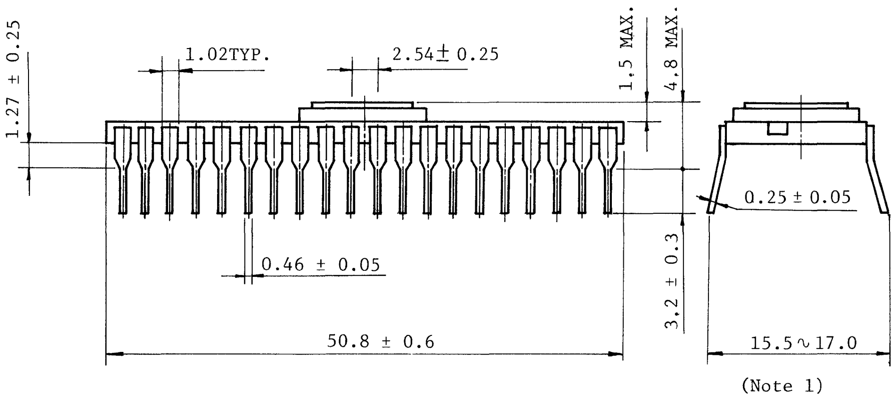
FIGURE 7 PROGRAM MODE TIMING DIAGRAM

OUTLINE DRAWING

40 Pins Ceramic Package

Unit in mm





Note 1. This dimension shows spread of leads.

2. All dimensions are in millimeters.

東芝

INTEGRATEDCIRCUIT

TECHNICAL DATA

TOSHIBA MOS TYPE DEGITAL
INTEGRATED CIRCUIT

TMP8259AP

Silicon Monolithic
N Channel Silicon Gate MOS

PROGRAMMABLE INTERRUPT CONTROLLER

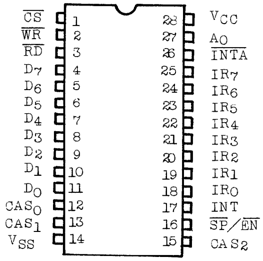
GENERAL DESCRIPTION

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)

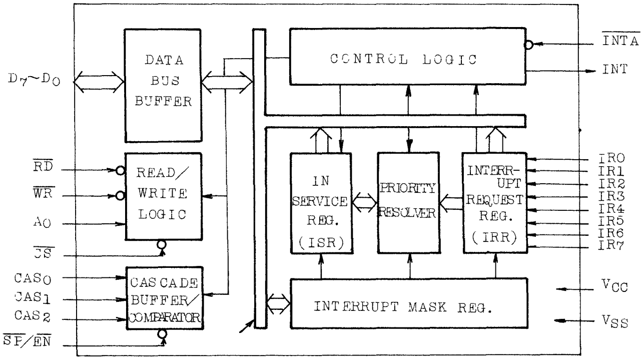


PIN NAMES AND PIN DESCRIPTION

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.
D0 ∿ D7	Input/Output	Biderectional Data Bus. Command status and interrupt-vector information is transfered via this bus.
CAS_0 ∿ 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 - IR_7	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).
$\overline{\text{INTA}}$	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.
A0	Input	A0 Address Line. This pin acts in conjunction with the $\overline{\text{CS}}$, $\overline{\text{WR}}$, and $\overline{\text{RD}}$ 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 ($\overline{RD}/\overline{INTA}$↓)	0	-	ns
TRHAX	$A_0/\overline{CS}$ Hold Time ($\overline{RD}/\overline{INTA}$↑)	0	-	ns
TRLRH	$\overline{RD}$ Pulse Width	235	-	ns
TAHWL	$A_0/\overline{CS}$ Setup Time ($\overline{WR}$↓)	0	-	ns
TWHAX	$A_0/\overline{CS}$ Hold Time ($\overline{WR}$↑)	0	-	ns
TWLWH	$\overline{WR}$ Pulse Width	290	-	ns
TDVWH	D_0 - D_7 Setup Time ($\overline{WR}$↑)	240	-	ns
TWHDX	D_0 - D_7 Hold Time ($\overline{WR}$↑)	0	-	ns
TJLJH	Interrupt Request Pulse Width (LOW)	100	-	ns
TCVIAL	Cascade Setup Time (Second or Third $\overline{INTA}$↓)	55	-	ns
TRHRL	$\overline{RD}$↑ to Next Command	160	-	ns
TWHRL	$\overline{WR}$↑ to Next Command	190	-	ns

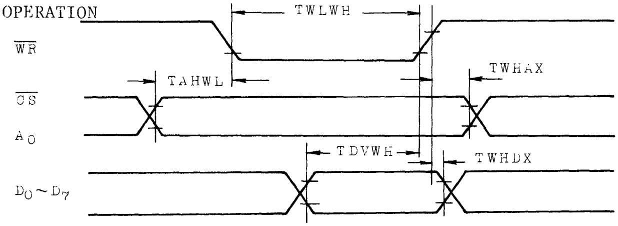
RESPONSE CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
TRLDV	Valid Data Delay ($\overline{RD}/\overline{INTA}$↓)		-	-	200	ns
TRHDZ	Data Floating ($\overline{RD}/\overline{INTA}$↑)	D_0 - D_7 C_L = 100pF	-	-	100	ns
TJHIH	Interrupt Output Delay (IR↑)		-	-	350	ns
TIALCV	Valid Cascade Delay ($\overline{INTA}$↓)	INT C_L = 100pF	-	-	565	ns
TRLEL	Enable Active ($\overline{RD}/\overline{INTA}$↓)		-	-	125	ns
TRHEH	Enable Inactive ($\overline{RD}/\overline{INTA}$↑)	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

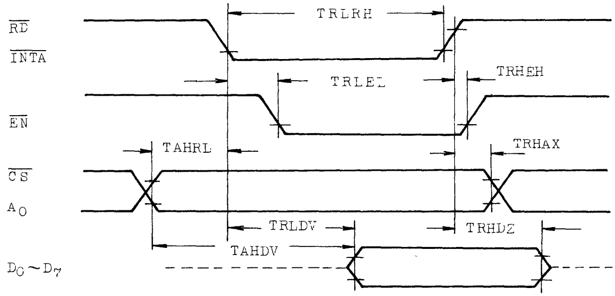
NOTE: AC TESTING. Inputs are driven at V_L = 0.45V and V_H = 2.4V.
Measurements are made at V_L = 0.8V and V_H = 2.0V.

TIMING WAVEFORMS

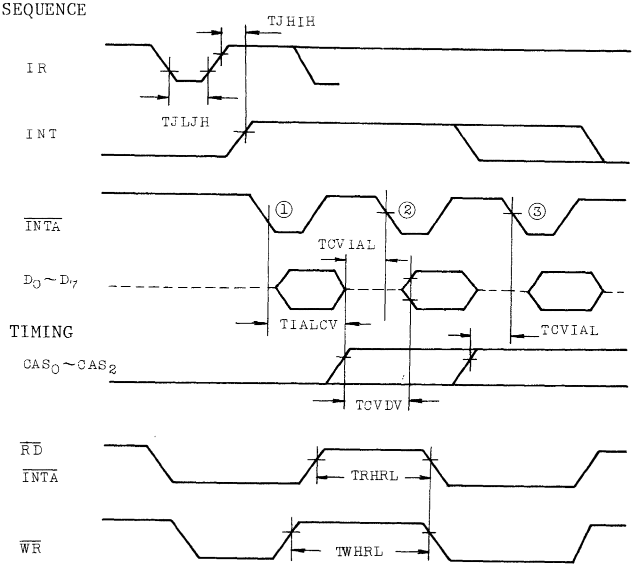
WRITE OPERATION



READ AND INTA OPERATION

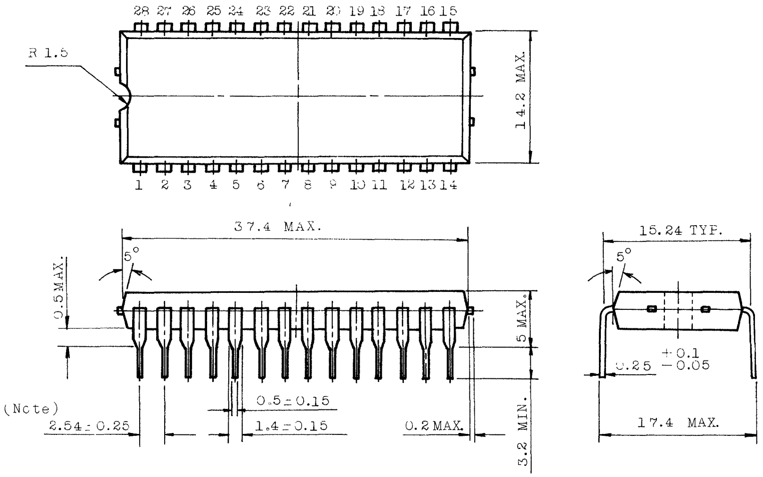


INTA SEQUENCE



OUTLINE DRAWING

UNIT : 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.42 leads.

MICROCOMPUTER APPLICATION

DEVELOPMENT TOOL

Toshiba 東芝

MICROCOMPUTER

TECHNICAL DESCRIPTION

Microcomputer Technical Description

MICROCOMPUTER APPLICATION DEVELOPMENT TOOL

GENERAL DESCRIPTION

DEVECOPMENT TOOLS

Toshiba's single-chip microcomputer development tool has the TDS400 microcomputer application development system and the dedicated debugging system.

MICROCOMPUTER APPLICATION DEVELOPMENT SYSTEM TDS400

By exchanging portions of the control board and software, the TDS400 system can be changed into the following development systems:

(1) TDS400/43 TLCS-43 Application Development System

(2) TDS400/46 TLCS-46A Application Development System

(3) TDS400/84 TLCS-84 Application Development System

The TDS400 system is hereinafter described under the name of the TDS400/xx (xx denotes the name of the microcomputer series of 43, 46 or 84).

DEDICATED DEBUGGING SYSTEM

The dedicated debugging system can be realized by combining the debugging board and the emulation board, consisting of the following products.

(1) BM4304 TLCS-43 debugging board + BM4303 TLCS-43 emulation board

(2) BM4604 TLCS-46A debugging board + BM4603B TLCS-46A emulation board

(3) BM8404 TLCS-84 debugging board + BM8403 TLCS-84 emulation board

1. MICROCOMPUTER APPLICATION DEVELOPMENT SYSTEM TDS400

1.1 SYSTEM CONCEPT

The TDS400/xx system is the application program development support system of Toshiba's single-chip microcomputer. Microcomputer application program development is divided into two main categories, programming and real time debugging. These functions can be performed consecutively through the TDS400/xx system.

o Programming:

(1) Programming and editing of source program

(2) Assembling source program (converting to machine code)

(3) Loading, modifying and writing into EPROM in the object program

o Real time debugging:

(1) Allows to execute and debug program at a state with the application circuit. (direct modification with machine codes)

(5) Monitoring and revising of internal status during program execution.

(6) Data exchange with EPROM (read/write)

The following mannuals are available as system documents relative to the functions and operating instructions of the TDS400/xx:

(A) Operating Instructions (Description of the functions and operations of the above-mentioned (1) to (6).

(B) Instruction Manuals (configuration description of hardware).

1.2 HARDWARE CONFIGURATION

The TDS400/xx is a sytem which a microcomputer consists of TMP8085A MPU, 16 K-byte ROM, and 48K-byte RAM has been equipped with various auxiliary memories and input/output interfaces. To ensure a more useful development system, it is provided with 16K EPROM wirter, emulation board, program memory (4 K-byte RAM) which can be substituted with ROM, and dedicated debugging console.

The TDS400/xx hardware configuration is given in Fig.1, and its conceptrual system view in Fig.2, respectively. In Fig.1 solid line boxes show basic configuration units, while dotted line boxes indicate optional units. The units enclosed with broken lines are supplied by users, as required. Shaded boxes indicate the specific units depending on the series of a microcomputer. Other units are common to both.

When used as a programming system, this system requires a minimum of one console I/O device for input/output of commands and application program data. When the system is used as a real-time debugging tool, operation is made through the attached debugging console. Real time execution can be obtained by removing the single chip microprocessor from the application circuit (user system) and plugging in the DIP socket from the emulation board.

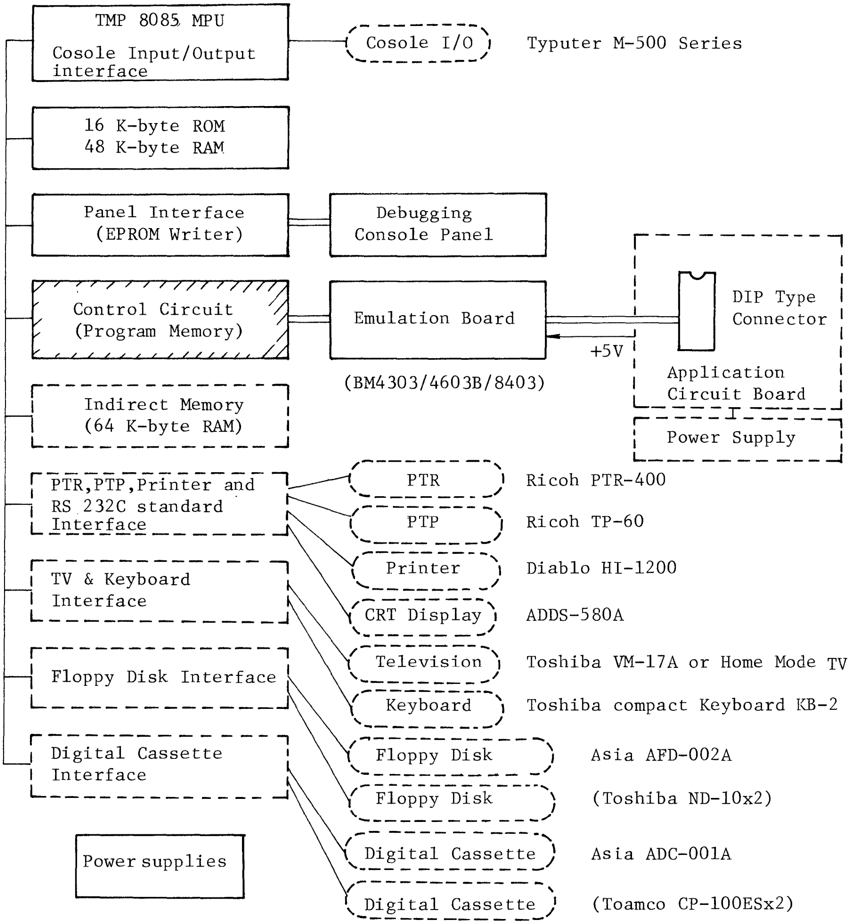


Fig. 1 TDS400/xx Hardware Configuration

Solid line boxes: basic configuration Dotted lines: Optional units
Broken lines: user-supplied Shaded line boxes: specific to microcomputer

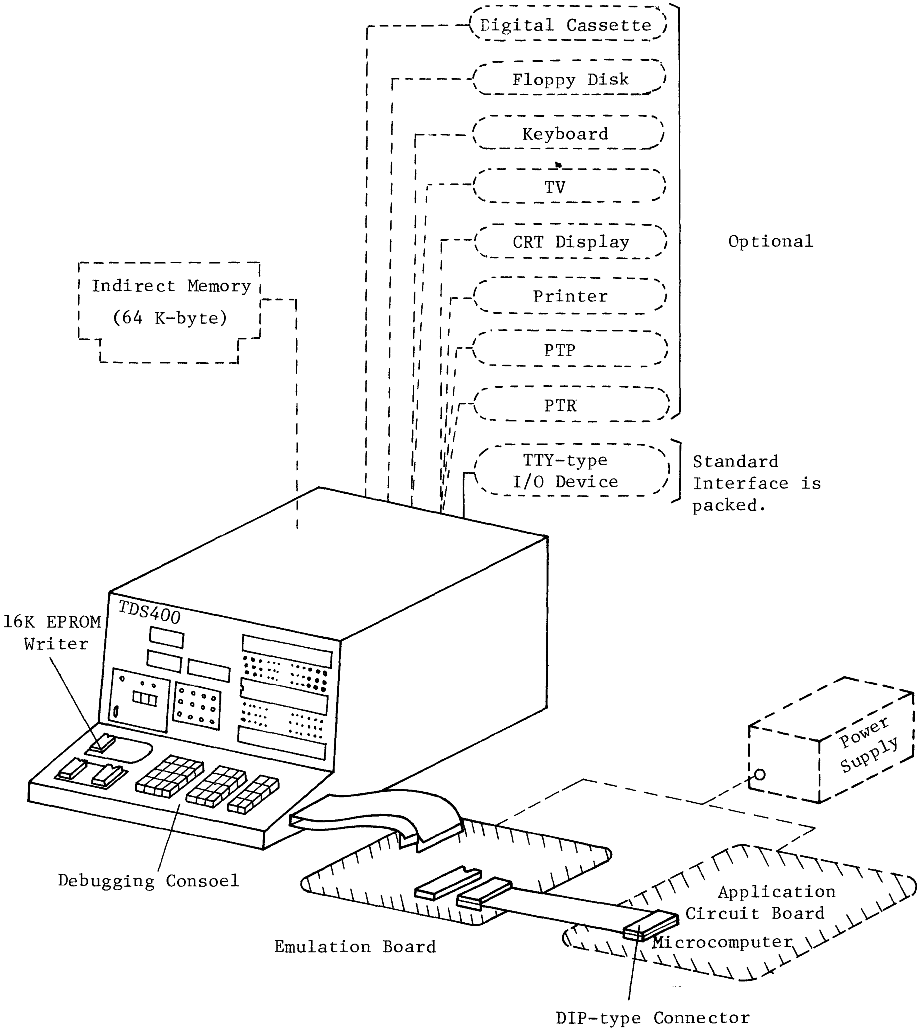


Fig. 2 Conceptual System View of TDS400/xx System

1.3 SOFTWARE CONFIGURATION

As shown in ·Fig.3, the TDS400/xx system software consists of a programming system segment and debugging console control segment. Shaded portions indicate the specific program depending on the microcomputer series. Other programs are common to both.

These programs are firmwared in 16 K-byte ROM within the main memory of the TMP8085A system: therefore, no loading of system programs is required, thus programs being able to be immediately operated. However, for operating option programs, such as assembler programs, the programs must be separately loaded.
The system program uses a 2 K-byte as dedicated data area. Furthermore, a 4K-byte RAM is designated as dedicated area (called the back-up memory) for internal save and transmission in the program memory. The remaining 42 K-byte RAM area is a free area used as user programs, symbol tables, current file, or assembler program. (See Fig. 4)

1.4 PROCESS FLOW OF APPLICATION PROGRAM DATA

Fig. 5 shows a conceptual flow chart of the application program data for application program development and evaluation by using the TDS400/xx system (as programming system or debugging console)

1.5 PROGRAMMING SYSTEM

The programming system is provided for facilitating the programming work (development) of application software.

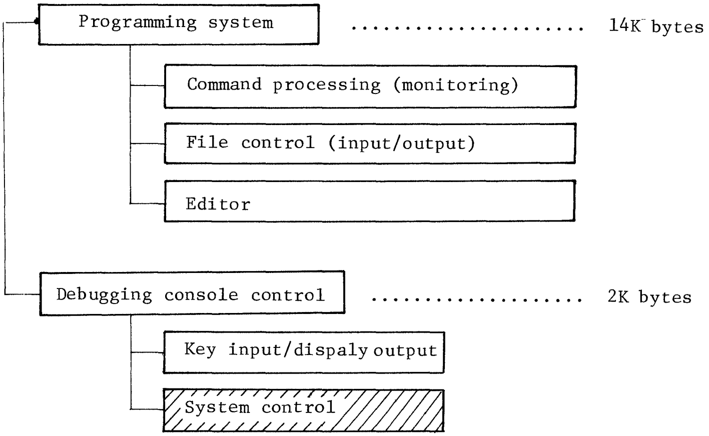

Programming system
14K bytes
Command processing (monitoring)
File control (input/output)
Editor
Debugging console control
2K bytes
Key input/dispaly output
System control

Fig. 3 TDS400/xx System Program Configuration
Shaded element: Portions depending to microcomputer series

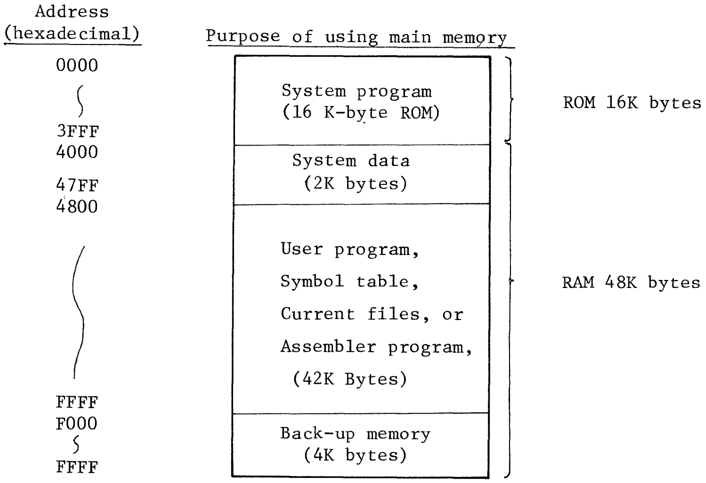

Address (hexadecimal)
Purpose of using main memory
0000
3FFF
4000
47FF
4800
FFFF
F000
FFFF
System program (16 K-byte ROM)
System data (2K bytes)
User program, Symbol table, Current files, or Assembler program, (42K Bytes)
Back-up memory (4K bytes)
ROM 16K bytes
RAM 48K bytes

Fig. 4 TDS400/xx Main Memory Configuration

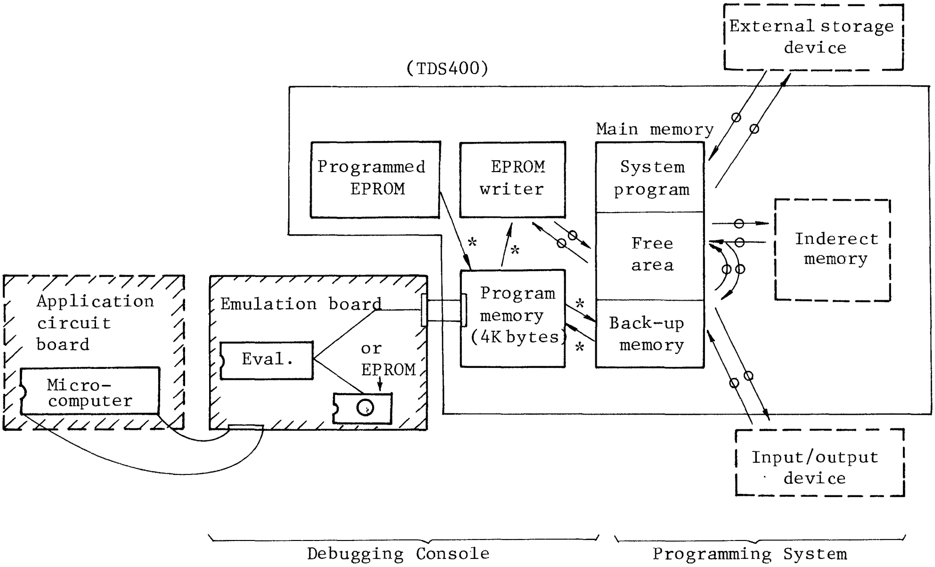


Fig. 5 Data Flow of Application Program

* : Operation from the debugging console.

o : Operation through the porgramming system.

The TDS400/xx programming system operates according to an input command from the input/output device. The following outline shows the outline of utilization form at each step in application software programming.

(1) In making a source program by expressing the control processing contents of application system in assembler language, the source program is entered directly through the keyboard or via paper tape, etc., revised and edited statement (line) by statement (line). If the program was structured into blocks, when it was written, the program can be easily edited into a single program and then punched onto paper tape, etc.

(2) In assembling the source program, the assembler program is activated. Each pass of the program listing, object tape output, and generation of object program (machine code) in the main memory can be selected from the commands of input/output device.

(3) At the machine code level, the following functions can be implemented: object program loading, transmission, modifying, dumping, format change on the paper tape and writing into 16K EPROM.

The entire programming system can be operated from a single input/output device[1], but for performing the operation at high-speed,multiple input/output devices can also be utilized.

Note 1) Can use teletype, CRT display, or TV·keyboard

(a) 64 K-byte indirect memory --- can be used as a temporary file memory, For example, if the paper tape source program is once stored in this memory, the program will become much faster and easier to excute. The source program, however, is restricted to about 1.5 ∿ 2 K-byte because of memory capacity.

(b) Floppy disk --- a file memory (random file) similar to indirect memory. Memory capacity is approximately 250 K-byte. Each disc can store 6 K∿8 K-step source program. The processing rate is lower than that of indirect memory.

(c) Paper tape reader and Paper tape punch --- allows faster processing of paper tape.
Printer -- speeds up program listing output
CRT display -- convenient for editing and modifying application programs.

(d) Digital cassette --- can be used as a sequential file, instead of paper tape. This eliminates the bother of rewinding and remounting paper tape, improving speed and ease of handling.

(e) Television receiver and simple keyboard --- useful as low cost input/output devices.

1.6 DEBUGGING CONSOLE

The debugging console is designed for easy real time debugging of the application system.

The debugging console, mounted in the front panel of the TDS400, is a dedicated console. Major functions performed by this console are as follows:

(1) Real time emulation can be performed. For application program memory, the user can select either program memory (RAM) within the TDS400/xx or external memory (EPROM) within the emulation board.

(2) Program memory read and write is done directly through the debugging console.

(3) Loading to program memory:
(i) The program can be loaded directly from program EPROM.
(ii) The program can also be loaded from the back-up memory within the development system[Note 2)].

(4) Contents of the program memory:
(i) Can be written directly into 16K EPROM.
(ii) Can be compared with the contents of programmed EPROM.
(iii) Can be transferred and saved to the back-up memory

(5) Direct read and write can be performed with RAM or register built into the single-chip microcomputer just as with program memory.

(6) Application programs can be initiated from any address.

(7) Application ptograms may be stopped at any time while in operation.

(8) Address-stop function provided.

Note 2) The back-up memory is a point of contact between the debugging console function and the programming system function. The programming system can load the back-up memory with an object program, and the contents of the back-up memory can be put out onto paper tape, etc. (in various formats).

(9) Single-step operation possible.

(10) Dummy interrupt signal can be entered through the debugging console.

(11) Logical levels of all pins of microcomputer chip are always indicated. Voltage of the threshold value can be see at any level.

1.7 EMULATION BOARD

In the last step of application system development, it becomes extremely important to be able to check the system operations as close to the end product as possible. For answering this need, emulation board (BM4303/BM603B/BM8403) is available, as shown in Table 1. These emulation board are integrated with the evaluator chip and are designed to use EPROM in the program memory section. Fig. 6 shows the conceptualized drawings of these emulation boards.

Besides the circuits attached to the evaluator chip, the emulation board has interface circuits for connecting to the application development system TDS400 and free space for mounting a small number of peripheral circuits. Fot the program memory, 24-pin DIP sockets are provided so as to make it possible to use 16K EPROM TMM323 (2716) unit.

When this emulation board is connected with the TDS400, the user can choose to put the application program either into program memory (RAM) within the TDS400/xx or into EPROM on the emulation board. The former freely permits reading and reloading of program memory contents through the TDS400/xx debugging console, making debugging of the application program during development very convenient.

The latter does not permit reading and writing of the application program contents. However, in either case the application program can easily be checked by using the function of the debugging console of TDS400/xx while monitoring or changing internal status of the microcomputer.

When the emulation board is disconnected from the TDS400, EPROM within the board is automatically used as program memory and the board can be used as an independent emulation board.

The BM4302/4602A makes a compact emulation boards, equipped with one/two 16K EPROM TMM323 ·(2716) as program memory. This board is useful for field-testing the applied product, but lacks free space and an interface to TDS400.

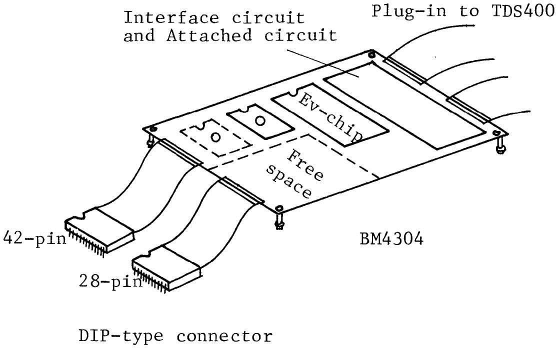
Each emulation board provides each microcomputer chip and a DIP-type equivalent connector. With the microcomputer chip removed from the application circuit (user system) and this connector connected with its place, operations can be performed as if a single-chip microcomputer were mounted.

When the application circuit is simple and no application circuit board has been provided, system development and evaluation can be accomplished by mounting a peripheral circuit in the free space on the emulation board.

Fig. 7 illustrates a concenptualized view of each using situation of these emulation boards. Optimum emulation board and evaluation means can be selected according to the evaluation objectives of the application system.

Table 1 Single-chip Microcomputer Emulation Board

Board Name	Evaluator Chip	Program memory EPROM		Oscil-lator (Hz)	Free Space	Power Supply Required	Remarks
		EPROM	Qt. of Socket				
BM4303	TMP4300C	16 K (one)	1	I F T. (455K)	Yes	5V	Can be used invididually or connected TDS400/43
BM4603B	TCP4600AC	16 K (two)	2	I F T (400K)	Yes	5V	Can be used individually or connected TDS400/46
BM8403	TMP8039-6	16 K (two)	2	Crystal 6M	Yes	5V	Can be used individually or connected TDS400/84



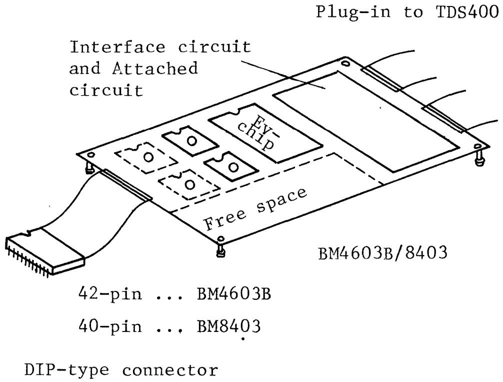


Fig. 6 Emulation Boards --- Conceptualized Drawings

(Note: Ev-chip is an abbreviation of evaluator chip.)

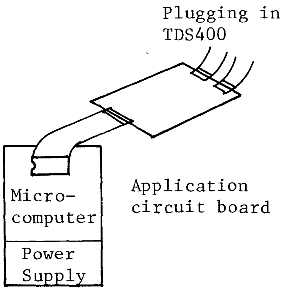
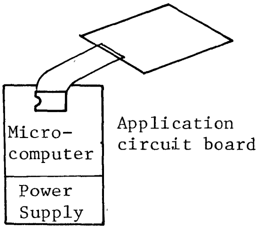
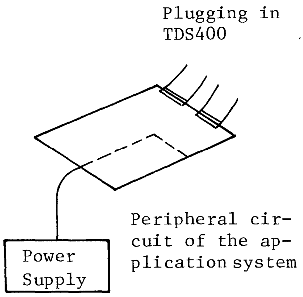
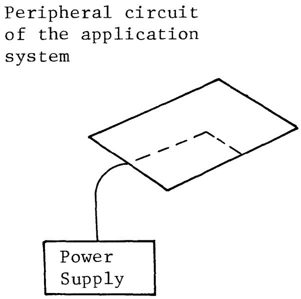
<table>
<tr>
<th>Plugging in the TDS400
Type of mounting for the application circuit (Peripheral circuit)</th>
<th>For plugging in the TDS400 to verify operations. To mount application program in:
o TDS400 program memory (RAM) or
o Emulation board EPROM</th>
<th>For verifying operations of the application system by use of the emulation board alone or for making field test.</th>
</tr>
<tr>
<td>Verifying or dubugging operations of systems by replacing the microcomputer chip within the applicaiton circuit board.</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>For verifying or debugging operations by mounting the microcomputer peripheral circuit of the application circuit on the free space.</td>
<td>
</td>
<td>
</td>
</tr>
</table>

Fig. 7 Examples of Emulation Board Usage

2. DEDICATED DEBUGGING SYSTEM

2.1 SYSTEM CONCEPT

The dedicated debugging system is formed by a combination of the debugging board (BM4304/4604/8404) and emulation board. The debugging board is a low cost development tool for debugging application software and hardware to which Toshiba's single-chip microcomputer is applied. For assembling a program, other means must be utilized.

Fig. 8 shows the conceptualized drawing of this development system. In this drawing, the portions enclosed with solid lines are the units of minimum requirement for program debugging, while those enclosed with broken lines are the units supplied by user as required. The shaded portions which the cable connecting the debugging board to emulation board has been removed, indicate the structure to make it possible to execute application programs through EPROM on the emulation board.

Mojor functions of the debugging board are as follows:

(1) Real-time emulation possible. The program memory (RAM 4K bytes) of the debugging board is used as an application program memory.

(2) The contents of the program memory can be read and written directly from the console.

(3) The program loading can be performed from the written EPROM to the program memory.

(4) Direct read and write can be performed to RAM, register and I/O device contained in the single-chip microcomputer in the same way as the program memory.

(5) Application programs can be initiated from any address.

(6) Application programs can be stopped at any time while in operation.

(7) Address-stop function provided.

(8) Single-step operation possible.

(9) Pseudo interrupt signal can be entered through the debugging console.

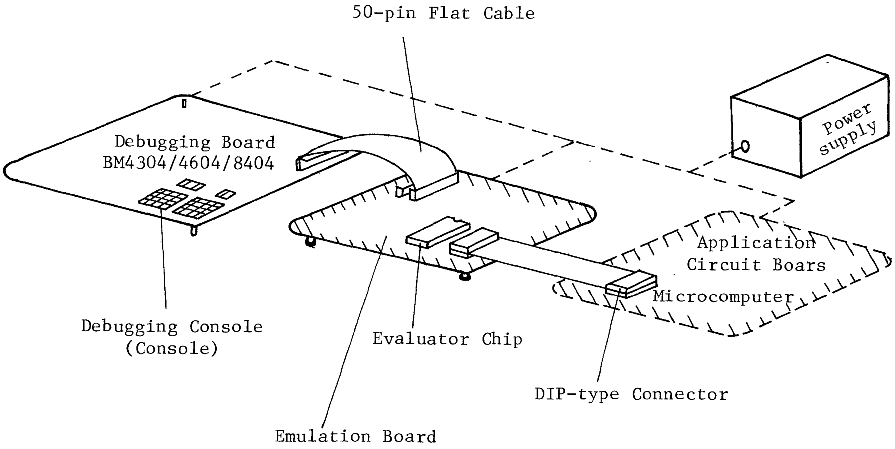


Fig. 8 Conceptualized Drawing of Dedicated Debugging System

3. PRODUCTS CONFIGURATION FOR APPLICATION DEVELOPMENT SUPPORT SYSTEM

Depending on the user's needs, the TDS400 and emulation board can be converted or expanded. For example, conversion of the TDS400/43 system to the TDS400/46 system can be accomplished by replacing BM4301A + BM4303 + SW4323 with BM4601A + BM4603B + SW 4623, and vice versa.

The optional I/O interface boards are common to all the TDS400 development systems. These interfaces can be attached directly to the supposed standard I/O equipment and/or auxiliary memory devices (See Fig. 1). Further, these equipment and devices are so designed as to be used without modification of the system software.

Therefore, it is very convenient for user to keep various kinds of several emulation boards at hand in order to forward verification work of application systems smoothly.

PRODUCTS CONFIGURATION FOR APPLICATION DEVELOPMENT TOOL

Family	Model No.	Description	Configuration of Products	Accompanying Materials	Remarks
Common to	TDS400	Microcomputer application system 400 (Toshiba MDS model 400)	BM4006A + SW4006A Optional board (BM4001 ∿ 5)	Refer to each system component	Products name of combined system, peripheral equipment is supplied by user.
	SW0051	Paper tape output program/ACOS (PT output program on ACOS)	Card image magnetic tape (9 tracks) of source program (FORTRAN)	SW0051 Instruction manual	For object output of cross assembler (SWxx53)
	SW0061	Paper tape output program/NOVA (PT output program on NOVA)	Paper tape (ASCII code) of source program (FORTRAN)	SW0061 Instruction manual	For object output of cross assembler (SWxx63)
	BM4001	64KB Indirect memory	Board module (1)	BM4001 Instruction manual	Mounted on BM4006A
	BM4002	General I/O interface	Board module (1), intra-board connection cable (2), connector board(1)	BM4002 Instruction manual	Ditto RS232C, PTR, PTP & Printer
	BM4003	TV·Keyboard interface	Board module (1), keyboard (1), intra-board connection cable (2), connector board (1)	BM4003 Instruction manual	Ditto Connected to NTSC type of TV set
	BM4004	Floppy disk interface	Board module (1), intra-boad connection cable (1) connector board (1)	BM4004 Instruction manual	Ditto Can be connected to two FD drives of single-sided single density

Family	Model No.	Description	Configuration of Products	Accompanying Materials	Remarks
Common to	BM4005	Digital cassette interface	Board module (1), intra-board connection cable (2), connector board (1)	BM4005 Instruction manual	Mounted on BM4006A Can be connected to two 10"/sec DC handlers
	BM4006A	TDS400 hardware main frame	A set of TDS400 basic hardware with power supply in cabinet	BM4006A Instruction manual	EPROM writer with debugging console, serial port (1)
	SW4006A	TDS400 system A monitor program	EPROM (8) (monitor program, 16K bytes)	TDS400 System A operation manual, SW4006A instruction manual	System control program packed in main memory region of BM4006A
43	TDS400/43	Toshiba MDS400 for TLCS-43	TDS400 + BM4301A + BM4303 + SW4323	Refer to each system component	Products name of combined system. Peripheral equipment is supplied by user
	BM4301A	TDS400/43 controller	Board module (1), EPROM (1), (control program, 2K bytes)	TDS400/43 debugger operation manual, BM4301A Instruction manual	EPROM is repalced with a part of SW4006A in main memory area
	BM4303	TLCS-43 emulation board	Board module (1), EPROM (1), Cables (2) to LSI socket	BM4303 Instruction manual	Board with functions equivalent to LSI of TLCS-43. With interface to TDS400
	BM4304	TLCS-43 debugging board	Board module (1), cable (1) connecting with BM4303	BM4303 debugging board operation manual. BM4303 instruction manual	Used for debugging by connecting with BM4303

Family	Model No.	Description	Configuration of Products	Accompanying Materials	Remarks
43	SW4323	TLCS-43 assembler on TDS400	EPROM (3) (Assembler program 6K bytes)	TLCS-43 assembly language manual. SW4323 Instruction manual	Loaded from EPROM writer (/reader) of TDS400
	SW4353	TLCS-43 cross assembler on ACOS	Card image magnetic tape (9 tracks) of source program (FORTRAN)	TLCS-43 assembly language manual. SW4353 Instruction manual	Used together with SW0051
	SW4363	TLCS-43 cross assembler on NOVA	Paper tape(ASCII code) of source program (FORTRAN)	TLCS-43 assembly language manual. SW4363 Instruction manual	Used together with SW0061
	SW4393S SW4393D	TLCS-43 assembler/ iMDS	Diskette(1) (ISIS-II file) S: single side single density. D: single side double density.	TLCS-43 assembly language manual. SW4393S, D instruction manual	Operates under ISIS-II
46A	TDS400/46	TLCS-46A application development	TDS400 + BM4601A + BM4603B + SW4623	Refer to each system component	Products name of combined system, peripheral equipment is supplied by user.
	BM4601A	TDS400/46 controller)	Board module (1), EPROM (1) (control porgram 2K bytes)	TDS400/46 debugger operation manual. BM4601A instruction manual	EPROM is replaced with a part of SW4006A in main memory area

Family	Model No.	Description	Configuration of Products	Accompanying Materials	Remarks
46A	BM4603B	TLCS-46A emulation board	Board module (1), EPROM (2), Cable (1) to LSI socket	BM4603B Instruction manual	Board with functions equivalent to LSI of TLCS-46A. With interface to TDS400
	BM4604	TLCS-46A debugging board	Board module (1), Cable (1) to BM4603B	BM4604 debugging board Operation manual. BM4604 instruction manual	Used for debugging by connecting with BM4603B
	SW4623	TLCS-46A assembler on TDS400	EPROM (3) (assembler program 6K bytes)	TLCS-46A assembly language manual. SW4623 instruction manual	Loaded from EPROM writer (/reader) of TDS400
	SW4653	TLCS-46A cross assembler on ACOS	Card image magnetic tape (9 tracks) of source program (FORTRAN)	TLCS-46A assembly language manual. SW4653 instruction manual	Used together with SW0051
	SW4663	TLCS-46A cross assembler on NOVA	Paper tape (ASCII code) of source program (FORTRAN)	TLCS-46A assembly language manual. SW4663 instruction manual	Used together with SW0061
	SW4693S SW4693D	TLCS-46A assembler on iMDS	Diskette (1) (ISIS-II file) S: single side single density. D: single side double density.	TLCS-46A assembly language manual. SW4693S, D instruction manual	Operates under ISIS-II
84	TDS400/84	Toshiba MDS400 for TLCS-84	TDS400 + BM8401A + BM8403 + SW8423A	Refer to each system component	Products name of combined system peripheral equipment is supplied by user.

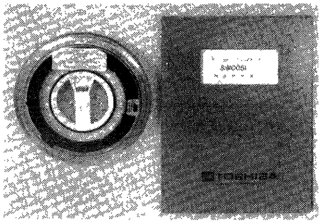
Family	Model No.	Description	Configuration of Products	Accompanying Materials	Remarks
84	BM8401A	TDS400/84 controller	Board module (1), EPROM (1) (control program 2K bytes)	TDS400/84 debugging operation manual. BM8401A instruction manual	EPROM is replaced with a part of SW4006A in main memory region
	BM8403	TLCS-84 emulation board	Board module (1), EPROM (1). Cable (1) to LSI socket	BM8403 Instruction manual	Board with functions equivalent to TMP8048, 8049-6. With interface to TDS400
	BM8404	TLCS-84 debugging board	Board module (1), Cable (1) to BM8403	BM8404 debugging board operation manual. BM8404 instruction manual	Used for debugging by connecting with BM8403
	SW8423A	TLCS-84 assembler on TDS400	EPROM (5) (assembler program 10K bytes)	TLCS-84 assembly language manual. SW8423A instruction manual	Loaded from EPROM writer (/reader) of TDS400
	SW8453	TLCS-84 cross assmebler on ACOS	Card image magnetic tape (9 tracks) of source program (FORTRAN)	TLCS-84 assembly language manual. SW8453 instruction manual	Used together with SW0051
85	SW8553	TLCS-85 cross assembler on ACOS	Card image magnetic tape (9 tracks) of source program (FORTRAN)	TLCS-85 assembly language manual. SW8553 instruction manual	Used together with SW0051

SW0051 PAPER TAPE OUTPUT PROGRAM / ACOS

GENERAL DESCRIPTION

This program outputs the object program translated with a cross assembler to the paper tape with various formats.
The program can be loaded on the general-purpose computer ACOS600 or 700.

PRODUCT CONFIGURATION



o Magnetic tape (9 tracks) (Card (IBM29 code) image of a source program written in FORTRAN (JIS7000, partially GMAP)).	1
o SW0051 Instruction manual	1

FEATURES AND MAIN SPECIFICATIONS

(1) Using this program, paper tapes can be made up to be used for various EPROM writers, production system of masked ROMs, and application development system.
(2) The formats of the object paper tape are divided broadly into 1-bit BNPF format and 4-bit hexadecimal expressing format.•
(3) For the hexadecimal format, 8 formats can be specified by selecting the output system such as start and end symbols of data, and addressing and checksum modes.
(4) Output range and block length can be specified.

OPERATING ENVIRONMENT (Host Machine)

Processing unit:	ACOS600/700 with more than 64K words (more than 17-bit words)
Operating system:	ACOS-6
Peripheral equipment:	Magnetic disk, magnetic tape, line printer, card reader, paper tape punch, etc.

OPERATING PROCEDURE

Compile the program first and load it in the host system. According to OS from the host machinè, specify the file and output format to operate the program.

TECHNICAL MATERIALS

(1) SW4353, SW4653, SW8453, SW8553 Instruction Manuals
(2) ACOS600/700 FORTRAN Language Description
(3) ACOS600/700 Job Control Language (JCL) Description
(4) ACOS600/700 IMCV Editing Program Manual

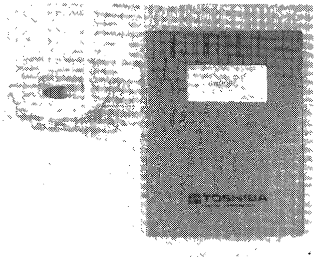
SW0061 PAPER TAPE OUTPUT PROGRAM / NOVA

GENERAL DESCRIPTION

This program outputs the object program translated with a cross assembler to the paper tape with various formats.
The program can be loaded on the micrompter NOVA.

PRODUCT CONFIGURATION

o Magnetic tape (ASCII code) (Source program written in FORTRAN (JIS7000))	1
o SW0061 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Using this program, paper tapes can be made up to be used for various EPROM writers, production system of masked ROMs, and application development systems.
(2) The formats of the object paper tape are divided broadly into 1-bit BNPF format and 4-bit hexadecimal expressing format.
(3) For the hexadecimal format, 8 formats can be specified by selecting the output systems such as start and end symbols of data, and addressing and checksum modes.
(4) Output range and block length can be specified.

OPERATING ENVIRONMENT (Host Machine)

Processing unit:	NOVA with more than 64K words (more than 16-bit words)
Operating system:	RDOS
Peripheral equipment:	Magnetic disk, line printer, console typewriter, paper tape reader, paper tape punch, etc.

OPERATING PROCEDURE

Compile the program first and load it in the host system. According to OS from the host machine, specify the file and output format to operate the program.

TECHNICAL MATERIALS

(1) SW4363, SW4663 Instruction Manuals
(2) NOVA FORTRAN IV Description Manual (Data General)
(3) NOVA RDOS Description Manual, CLI Series (Ditto)
(4) NOVA RDOS Description Manual, System Utility Series (Ditto)

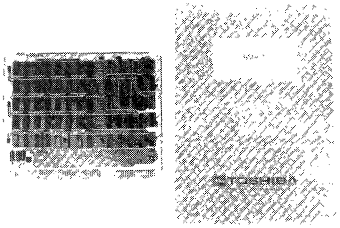
BM4001 64KB INDIRECT MEMORY

GENERAL DESCRIPTION

A memory board of 64,536 bytes (64KB) to be mounted in TDS400 (BM4006A). Generally this board is used as an auxiliary memory.

PRODUCT CONFIGURATION

o	Board module	1
o	BM4001 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) This memory can be used as a high-speed file memory.
(2) Assembling or the like is simple and fast if this memory is used as a file to store a source program or an object program.
(3) OS can specify this memory as a main memory.
(4) Thirty-two 16K dRAMs (TMM416P-3) are provided for BM4001.
(5) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.
(6) Approx. power dissipation is 5V × 1A, -5V × 1mA, or 12V × 0.1A.

OPERATING CONDITIONS

The indirect memory is inserted in the slot "MEM2" within the TDS400 (BM4006A), being operated by OS (control program) from SW4006A or the like.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual
(2) BM4006A Instruction Manual

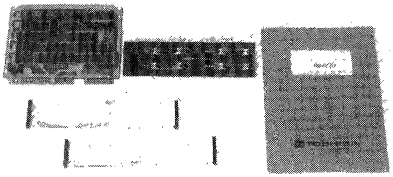
BM4002 GENERAL I/O INTERFACE

GENERAL DESCRIPTION

An interface board through which RS232C type I/O device, PTR, PTP, and printer are connected to the TDS400. The connectors for four I/O devices are provided separately.

PRODUCT CONFIGURATION

o	Board module	1
o	Connector board	1
o	Inter-board connection cable	2
o	BM4002 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Processing capability of the TDS400 can be sharply increased by the high-speed I/O device.
(2) RS232C ports are available for console I/O devices.
(3) 15 versions (50 - 9600 baud) can be selected for the baud rate of RS232C ports.
(4) RS232C type I/O devices (e.g. ADD's Model ADDS580) can be directly connected.
(5) Paper tape reader PTR-400 (Ricoh Denshi Industry Co.) can be directly connected.
(6) Paper tape punch TP-60 (Ricoh Denshi Industry Co.) can be directly connected.
(7) Printer 1200Hy Type 1 (Diablo System Inc.) can be directly connected.
(8) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.
(9) Approx. power dissipation is 5V × 0.7A, 12V × 0.1A, or -12V × 0.1A.

OPERATING CONDITIONS

The interface board is inserted in the slot "GIO" within the TDS400 (BM4006A). Mount the connector board on the back panel of the TDS400 and connect each unit with the inter-board connection cable. Each I/O device must be connected to the connector board with the dedicated cable (user's supplied connector for the TDS400 has 25 pins). The interface board is operated by OS (control program) of SW4006A or the like.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual
(2) BM4006A Instruction Manual
(3) CRT Terminal Model 580 Instruction Manual (Matsushita Electric Trading Co.)
(4) RICOM PTR-400 READER Specifications (Ricoh Denshi Industry Co.)
(5) RICOM TP-60 PERFORATOR Specification and Operating Manual (Ricoh Denshi Industry Co.)
(6) Model 1200 Hytype 1 Printer Maintenance Manual (Diablo System Inc.)

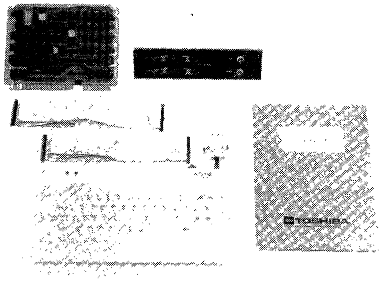
BM4003 TV·KEYBOARD INTERFACE

GENERAL DESCRIPTION

An interface board through which the NTSC system TV screen and a full keyboard are connected to the TDS400. That is, a CRT input/output device can be configured at low cost.

PRODUCT CONFIGURATION

o	Board module	1
o	Connector board with RF module	1
o	Inter-board connection cable	2
o	Full keyboard with a cable	1
o	BM4003 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) A low cost CRT input/output device usable for a console can be configured.

(2) Using BM4003, NTSC system home TV set or a monitoring TV set can be directly connected to the TDS400.

(3) Full keyboard (KB-2) with touch-key system is packed. (ACSII specification)

(4) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.

(5) Keyboard dimensions are 176mm × 345mm × 55mm.

(6) Approx. power dissipation is 5V × 0.7A, -5V × 30mA, or 12V × 50mA.

OPERATING CONDITIONS

The interface board is inserted in the slot "TV" within the TDS400 (BM4006A). The connector board is installed on the back panel of the TDS400 to connect each unit with the inter-board connection cables. Then user's TV set (the cable is user-supplied) and full keyboard are connected. This makes the TDS400 operate as a console I/O device or TV (similar to CRT display) output device.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual

(2) BM4006A Instruction Manual

(3) Toshiba Color TV 20G600 Instruction Manual

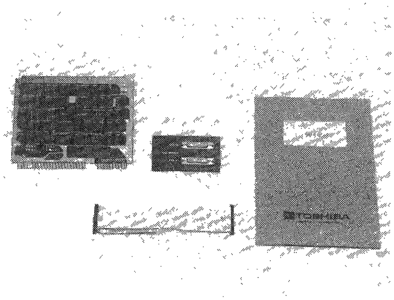
BM4004 FLOPPY DISK INTERFACE

GENERAL DESCRIPTION

An interface board through which two floppy disk drives (standard 8" single-sided single density) are connected to the TDS400.

PRODUCT CONFIGURATION

o	Board module	1
o	Connector Board	1
o	Inter-board connection cable	1
o	BM4004 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Mass storage (256K bytes/disk) random file can be made up by using BM4004 floppy disk interface.

(2) BM4004 controls two sets of floppy disk drive ND-10 (standard 8" single-sided single density).

(3) At least four sets of floppy disk drive can be controlled by using two boards of BM4004. However, the drivers, to the number of two, can be used for SW4006A.

(4) Available is a couple of floppy disk drive AFD-002 (Asia Mfg. Co.) which can be directly connected to the TDS400.

(5) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.

(6) Approx. power dissipation is 5V × 0.5A.

OPERATING CONDITIONS

The interface board is inserted in the slot "FDC1" within the TDS400 (BM4006A). The connector board is installed on the back panel of the TDS400 to connect each unit with the inter-board connection cable. The drive units, such as AFD-002A, are connected to the TDS400. The drive units are operated by a command from the console I/O device.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual

(2) BM4006A Instruction Manual

(3) AFD-002A Instruction Manual (Asia Mfg. Co.)

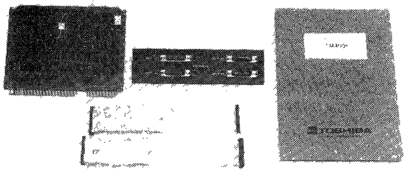
BM4005 DIGITAL CASSETTE INTERFACE

GENERAL DESCRIPTION

An interface board through which two digital cassette handlers (10"/sec) are connected to the TDS400.

PRODUCT CONFIGURATION

o	Board module	1
o	Connector board	1
o	Inter-board connection cable	2
o	BM4005 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Mass storage sequential file can be made up by using BM4005 digital cassette interface.
(2) Two sets of digital cassette handlers CP-100EB (Toshiba-Ampex 800BPI, 10"/sec) can be controlled by using the BM4005.
(3) Available is a couple of digital cassette handler ADC-001A (Asia Mfg. Co.) which can be directly connected to the TDS400.
(4) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.
(5) Approx. power dissipation is 5V × 0.8A.

OPERATING CONDITIONS

The interface board is inserted in the slot "DCC" within the TDS400 (BM4006A). The connector board is installed on the back panel of TDS400 to connect each unit with the inter-board connection cable. Then the handler, such as model ADC-001A, is connected to the TDS400. The handler is operated by a command from the console I/O device.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual
(2) BM4006A Instruction Manual
(3) ADC-001A Instruction Manual (Asia Mfg. Co.)

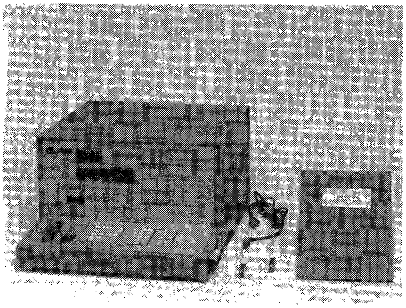
BM4006A TDS400 HARDWARE MAINFRAME

GENERAL DESCRIPTION

A basic hardware of microcomputer development system used for developing various microcomputer application system and application softwares. The system configuration is convenient for consecutively performing application software production, edit assembling, application system debugging, and test evaluation, etc.

PRODUCT CONFIGURATION

- o Set of TDS400 basic hardware (mainframe, power cord, connector)
- o BM4006A Instruction Manual 1



FEATURES AND MAIN SPECIFICATIONS

(1) The control display panel of debugging console is provided in front of the mainframe.
(2) Reader/writer for EPROM (equivalent to TMM323, 2714) is equipped with the TDS400.
(3) Console I/O interface (serial transmission, 50 ∿ 9,600 baud) is provided in the TDS400.
(4) The main memory consists of 48K-byte RAM and 16K-byte EPROM (socket). A monitor program (e.g. SW4006, option) is mounted in the EPROM.
(5) By installing optional interface boards, various peripheral equipment can be connected to the mainframe.
(6) Real-time debugging of application system, such as TLCS-43/46A/84, is available by exchanging some interface boards of EPROMs (control programs).
(7) Frame dimensions are 470mm (W) × 580mm (L) × 295mm (H).
(8) Approx. power dissipation is AC100V (50 or 60Hz), 350VA (Max.).

OPERATING CONDITIONS

By loading TDS400 monitor program (SW4006A) for system control, and by connecting the console I/O device such as teletypewriter (user-supplied), a minimum system can be configured. A system more usable can be built up by mounting various interface boards and by connecting peripheral equipment.

TECHNICAL MATERIALS

(1) TDS400 System A Operation Manual
(2) SW4006A Instruction Manual
(3) BM4001, BM4002, BM4003, BM4004 and BM4005 Instruction Manuals
(4) BM4301A, BM4601A and BM8401 Instruction Manuals
(5) Typuter MODEL-502 Type Specification (CASIO Computer)

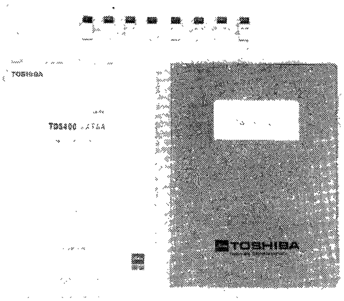
SW4006A TDS400 SYSTEM A MONITOR PROGRAM

GENERAL DESCRIPTION

A resident program within the EPROM of the TDS400 main memory. This system program can be used for various processing operations according to the command from the console I/O device.

PRODUCT CONFIGURATION

o	EPROM (TMM323) (16K-byte program)	8
o	TDS400 System A Operation Manual	1
o	SW4006A Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) This program facilitates creation, modification, and edit of source program for application systems.
(2) Control programs for the peripheral equipment, such as floppy disk, are contained in the monitor program.
(3) Assemblers and compilers are provided for various types of microcomputers (option).
(4) Loading or modifying the object programs of application systems as well as reading or writing into EPROM can be performed.
(5) This is a resident program within a main memory EPROM, so that it operates immediately after the power is turned on.

OPERATING ENVIRONMENT

Processor :	BM4006A (48K-byte RAM) and optional board modules.
Console I/O device :	Teletypewriter, etc.
Optional peripheral equipment:	Floppy disk, CRT, printer, PTR, PTP, digital cassette, etc.

OPERATING PROCEDURE

Insert the eight EPROMs in EPROM sockets, according to their corresponding numbers, on the BM4008 board of "MEM1" slot within BM 4006A. Select the I/O device used as a system console, and input the command from this console to control the operating system.

TECHNICAL MATERIALS

(1) BM4006A Instruction Manual
(2) BM4001, BM4002, BM4003, BM4004 and BM4005 Instruction Manuals

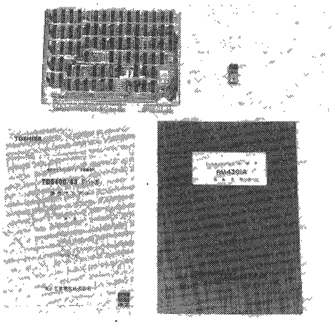
BM4301A TDS400/43 CONTROLLER

GENERAL DESCRIPTION

A special module for making the TDS400 to the TDS400/43 debugger. It can debug the application systems (software and hardware) of the TLCS-43 by connecting BM4303 (TLCS-43 emulation board).

PRODUCT CONFIGURATION

o Board module	1
o EPROM (TMM323) (Control program)	1
o TDS400/43 debugger operation manual	1
o BM4301A Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Application program can be executed with real-time operation by loading it in the dedicated program memory (RAM) within this board or in the EPROM on BM4303.

(2) Programs can be transmitted to and from EPROM or the back-up memory (a part of TDS400 main memory). Information stored in the back-up memory can be output to the paper tape, etc. in various formats (TDS400 function).

(3) Direct read/write of the registers and RAMs contained in the CPU, and program memory (corresponding to ROM) within this board can be performed.

(4) Various functions, such as execution and halt of programs, address stopping, single-step operation, and dummy interrupt signal input, are provided.

(5) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.

(6) Approx. power dissipation is 5V × 1.3A, -5V × 30mA, or 12V × 50mA.

OPERATING CONDITIONS

This board is inserted in slot "43C/46C" within the TDS400 (BM4006A). The control program is loaded in the BM4008 board of slot "MEM1" where EPROM No. 7 is mounted after having removed EPROM. BM4303 and application system (board) are connected to this board, which is operated by controlling the TDS400 debugging console.

TECHNICAL MATERIALS

(1) BM4006A Instruction Manual

(2) BM4303 Instruction Manual

(3) TDS400 System A Operation Manual

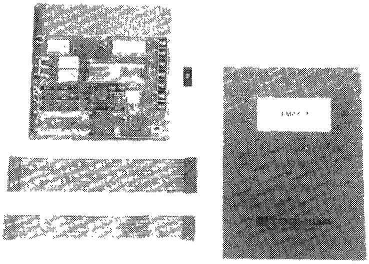
BM4303 TLCS-43 EMULATION BOARD

GENERAL DESCRIPTION

This is an emulation board that can be performed as equivalent to the single-chip microcomputer LSI of TLCS-43 series. Interface circuit is provided in this board for being connected to the host system (TDS400, BM4304). EPROM (for 2K bytes) can be installed as a program ROM.

PRODUCT CONFIGURATION

o Board module	1
o EPROM (TMM323)	1
o LSI socket connection cable	2
o BM4303 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) This is an emulation board that can be replaced directly with TMP4310, TMP4315 or TMP4320.
(2) 2 K-byte EPROM can be installed as a program ROM.
(3) Interface circuit to be connected to the TDS400 (BM4301A) or BM4304, and free space (for mounting application system circuit or interface) are provided in the board.
(4) As a program memory, this emulation board can select EPROM within the board or memory in the host system to execute the application program stored on these units.
(5) This emulation board has a function capable of executing the application program while monitoring or modifying the internal status on of the microcomputer under the control by the host system.
(6) When the emulation board is disconnected from the host system, it is also operated as an independent emulation board.
(7) Module dimensions are 230mm × 230mm × approx. 30mm.
(8) Power dissipation is less than 5V (±5%) × 0.5A. (Extra 12V and -5V power supplies are required when using 8K-bit EPROM.)

OPERATING CONDITIONS

EPROM, in which the application program has been written, is mounted on this board, and two cables from the board is inserted in the sockets of the microcomputer LSI of application system. The power supply (user-supplied) and host system (TDS400 or BM4304) are connected to the board for operation.

TECHNICAL MATERIALS

(1) TLCS-43 Integrated Circuit Technical Data
(2) TLCS-43 System Manual
(3) TDS400/43 Debugger Operation Manual, BM4301A Instruction Manual
(4) BM4304 Debugging Board Operation Manual, BM4304 Instruction Manual

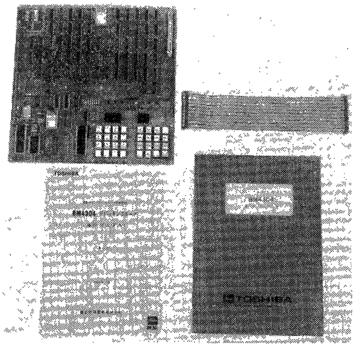
BM4304 TLCS-43 DEBUGGING BOARD

GENERAL DESCRIPTION

This board is a control board for debugging of the application system (or program) of the TLCS-43. The debugging system of the TLCS-43 is made up by connecting it to the BM4303 (TLCS-43 emulation board).

PRODUCT CONFIGURATION

o Board module	1
o BM4303 connection cable	1
o BM4304 debugging board operation Manual	1
o BM4304 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) BM4304 board is provided with a console unit consisting of a display for addresses, data and system condition, and 32 keys.
(2) Application program is executed by emulator chip (TMP4300) of BM4303 after having been transmitted to the EPROM program memory within this board via the EPROM socket.
(3) Direct read/write of register or RAM within the microcomputer LSI is available.
(4) Application program can be modified with machine code level.
(5) Application program can be started with any address and stopped at any time.
(6) Address stop and single step operation functions are provided.
(7) Dummy interrupt signals can be generated from the console on the board.
(8) Module dimensions are 300mm × 300mm × approx. 30mm.
(9) Power dissipation is less than 5V (±5%) × 2.5A.

OPERATING CONDITIONS

This board is connected with BM4303 that have been connected to the application system. The application program is written in EPROM, and EPROM is installed on the BM4304 board. The power (user-supplied) is turned on and the instructions from the console on BM4304 board are given to operate the system.

TECHNICAL MATERIALS

(1) BM4303 Instruction Manual

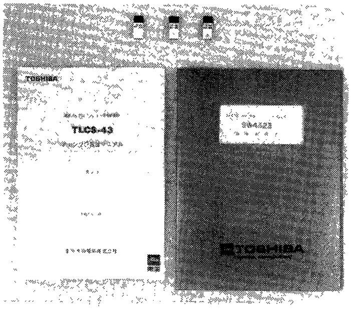
SW4323 TLCS-43 ASSEMBLER/TDS400

GENERAL DESCRIPTION

A cross assembler used for developing the application program of the TLCS-43. The TDS400 (Microcomputer Developing System 400) is used for operating this assembler.

PRODUCT CONFIGURATION

o	EPROM (TMM323) (Assembler program)	3
o	TLCS-43 Assembler Language Manual	1
o	SW4323 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx 950 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Decimal and hexadecimal constants can be handled.
(3) 16 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 7 assembler directives and 35 instructions.

(6) Source program can be input from any input device.
(7) Assemble list can be output to any output device.
(8) Object program is created as a ROM image in the TDS400. The TDS400 can output this program in various formats. Direct write in the EPROM is also available.

OPERATING ENVIRONMENT

Processor :	TDS400 (48K-byte RAM)
Operating system :	System A monitor program (SW4006A)
Peripheral equipment:	Console I/O device (e.g. teletypewriter), optional floppy disk, CRT, printer, PTR, PTP, etc.

OPERATING PROCEDURE

This assembler is operated by a command sent to the TDS400 from the console I/O device (or other devices). This assembler program itself should be lo loaded in the EPROM writer socket on the front panel of the TDS400.

TECHNICAL MATERIALS

(1) TLCS-43 Integrated Circuit Technical Data
(2) TLCS-43 System Manual
(3) TLCS-43 Collected subprograms
(4) TDS400 System A Operation Manual

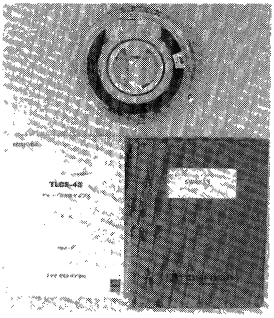
SW4353 TLCS-43 CROSS ASSEMBLER/ACOS

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-43. General-purpose computer ACOS600 or 700 can be used to load the program.

PRODUCT CONFIGURATION

o Magnetic tape (9 tracks) (Card (IBM29 code) image of source program written in FORTRAN (JIS7000, partially GMAP)	1
o TLCS-43 Assembler Language Manual	1
o SW4354 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 500 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 16 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 11 assembler directives and 38 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the paper tape output program ACOS (SW0051), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor :	ACOS600 or 700 with more than 64K-words (more than 17-bit words)
Operating system :	ACOS-6
Peripheral equipment:	Magnetic disk, magnetic tape, line printer, card reader, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First compile this cross assembler to load in the host system. According to the OS of the host machine, define various files, and specify the input and output conditions to perform assembling.

TECHNICAL MATERIALS

(1) TLCS-43 Integrated Circuit Technical Data
(2) TLCS-43 System Manual
(3) TLCS-43 Collected subprograms
(4) SW0051 Instruction Manual
(5) ACOS600/700 FORTRAN Language Description
(6) ACOS600/700 Job Control Language (JCL) Description
(7) ACOS600/700 IMCV Edit Program Description

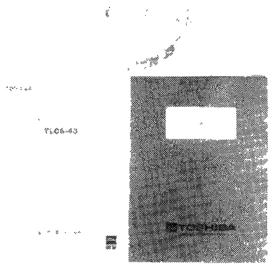
SW4363 TLCS-43 CROSS ASSEMBLER/NOVA

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-43. Mini-computer NOVA can be used to load the program.

PRODUCT CONFIGURATION

o	Paper tape (ASCII code) (Source program written in FORTRAN (JIS7000))	1
o	TLCS-43 Assembler Language Manual	1
o	SW4363 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to 100 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 16 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 11 assembler directives and 38 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the different paper tape output program/NOVA (SW0061), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor : NOVA with more than 32K-words. (16-bit words)
Operating system : RDOS
Peripheral equipment: Magnetic disk, line printer, console typewriter, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First compile this cross assembler to load in the host system. According to the OS of the host machine, specify various files to perform assembling.

TECHNICAL MATERIALS

(1) TLCS-43 Integrated Circuit Technical Data
(2) TLCS-43 System Manual
(3) TLCS-43 Collected Subprograms
(4) SW0061 Instruction Manual
(5) NOVA FORTRAN IV Description Manual (Japan Data General)
(6) NOVA RDOS Description Manual, CLI Series (Ditto)
(7) NOVA RDOS Description Manual, System Utility Series (Ditto)

SW4393S, SW4393D TLCS-43 ASSEMBLER/iMDS

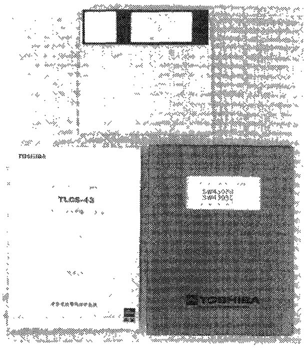
GENERAL DESCRIPTION

A cross assembler used for developing the application program of the TLCS-43. The intellec MDS is used for operating this assembler.

PRODUCT CONFIGURATION

o	Diskette Note) (Assembler program ISIS-II file)	1
o	TLCS-43 Assembler Language Manual	1
o	SW4393S, SW4393D Instruction Manual	1

Note) SW4393S is of single-sided single density.
SW4393D is of single-sided double density.



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 600 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Decimal and hexadecimal constants can be handled.
(3) 16 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 7 assembler directives and 35 instructions.

(6) The source program is input from ISIS-II diskette file.
(7) Assemble list can be output to any output device.
(8) The object program is created on the diskette in format I (Intel hexadecimal format) or format M (masked ROM production format). This program can be punched out to the paper tape by COPY command from ISIS-II, and also can be directly written in the EPROM.

OPERATING ENVIRONMENT

Processor :	iMDS with more than 32K bytes
Operating system :	ISIS-II
Peripheral equipment:	Console I/O device (e.g. teletypewriter), floppy disk, optional CRT, printer, PTR, PTP, PROM programmer, etc.

OPERATING PROCEDURE

Copy the program to the system diskette. Then specify the file and assembler parameter according to the OS from the host machine.

TECHNICAL MATERIALS

(1) TLCS-43 Integrated Circuit Technical Data
(2) TLCS-43 System Manual
(3) TLCS-43 Collected Subprograms
(4) ISIS-II System User's Guide (Intel)

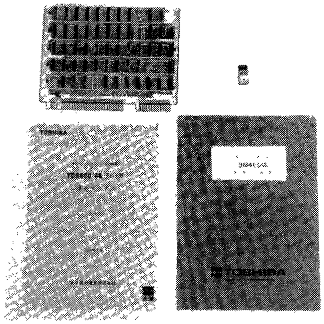
BM4601A TDS400/46 CONTROLLER

GENERAL DESCRIPTION

A special module for making the TDS400 to the TDS400/46 debugger. It can debug the application systems (software and hardware) of the TLCS-46A by connecting BM4603B (TLCS-46 emulation board).

PRODUCT CONFIGURATION

o	Board module	1
o	EPROM (TMM323) (Control program)	1
o	TDS400/46 Debugger Operation Manual	1
o	BM4601A Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Application program can be executed with real-time operation by loading it in the dedicated program memory (RAM) within this board or in the EPROM on BM4603B.

(2) Programs can be transmitted to and from EPROM or the back-up memory (a part of TDS400 main memory). Information stored in the back-up memory can be output to the paper tape, etc. in various formats (TDS400 function).

(3) Direct read/write of the registers and RAM contained in the CPU, and program memory (corresponding to ROM) within this board can be performed.

(4) Various functions, such as execution and halt of programs, address stopping, single-step operation, and dummy interrupt signal input, are provided.

(5) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.

(6) Approx. power dissipation is 5V × 1.3A, -5V × 30mA, or 12V × 50mA.

OPERATING CONDITIONS

This board is inserted in slot "43C/46C" within the TDS400 (BM4006A). The control program is loaded in the BM4008 board of slot "MEM1" where EPROM No. 7 is mounted after having removed EPROM. BM4603B and application system (board) are connected to this board, which is operated by controlling the TDS400 debugging console.

TECHNICAL MATERIALS

(1) BM4006A Instruction Manual

(2) BM4603B Instruction Manual

(3) TDS400 System A Operation Manual

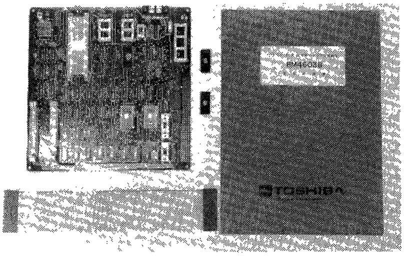
BM4603B TLCS-46A EMULATION BOARD

GENERAL DESCRIPTION

This is an emulation board that can be performed as equivalent to the single-chip microcomputer LSI of TLCS-46A series. Interface circuit is provided in this board for being connected to the host system (TDS400, BM4604). EPROM (for 4K bytes) can be installed as a program ROM.

PRODUCT CONFIGURATION

o	Board module	1
o	EPROM (TMM323)	2
o	LSI socket connection cable	1
o	BM4603B Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) This is an emulation board that can be replaced directly with TCP4620A, TCP4630A or TCP4620B.

(2) 4K-byte EPROM can be installed as a program ROM.

(3) Interface circuit to be connected to the TDS400 (BM4601A) or BM4604, and free space (for mounting application system circuit or interface) are provided in the board.

(4) As a program memory, this emulation board can select EPROM within the board or memory in the host system to execute the application program stored on these units.

(5) This emulation board has a function capable of executing the application program while monitoring or modifying the internal status on of the microcomputer under the control by the host system.

(6) When the emulation board is disconnected from the host system, it is also operated as an independent emulation board.

(7) Module dimensions are 230mm × 230mm × approx. 30mm.

(8) Power dissipation is less than 5V (±5%) × 0.5A. (Extra 12V and -5V power supplies are required when using 8K-bit EPROM.)

OPERATING CONDITIONS

EPROM, in which the application program has been written, is mounted on this board, and two cables from the board is inserted in the sockets of the microcomputer LSI of application system. The power supply (user-supplied) and host system (TDS400 or BM4604 are connected to the board for operation.

TECHNICAL MATERIALS

(1) TLCS-46A Integrated Circuit Technical Data

(2) TLCS-46A System Manual

(3) TDS400/46 Debugger Operation Manual, BM4601A Instruction Manual

(4) BM4604 Debugging Board Operation Manual, BM4604 Instruction Manual

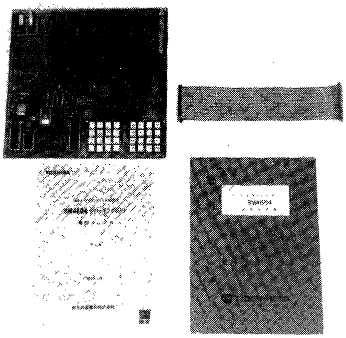
BM4604 TLCS-46A DEBUGGING BOARD

GENERAL DESCRIPTION

This board is a control board for debugging of the application system (or program) of the TLCS-46A. The debugging system of the TLCS-46A is made up by connecting it to the BM4603B (TLCS-46A emulation board).

PRODUCT CONFIGURATION

o	Board Module	1
o	BM4603B Connection cable	1
o	BM4604 Debugging board operation manual	1
o	BM4604 Instruction manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) BM4604 board is provided with a console unit consisting of a display for addresses, data and system condition, and 32 keys.
(2) Application program is executed by emulator chip (TCP4600A) of BM4603B after having been transmitted to the EPROM program memory within this board via the EPROM socket.
(3) Direct read/write of register or RAM within the microcomputer LSI is available.
(4) Application program can be modified with machine code level.
(5) Application program can be started with any address and stopped at any time.
(6) Address stop and single step operation functions are provided.
(7) Pseudo interrupt signal can be generated from the console on the board.
(8) Module dimensions are 300mm × 300mm × approx. 30mm.
(9) Power dissipation is less than 5V (±5%) × 2.5A.

OPERATING CONDITIONS

This board is connected with BM4603B that have been connected to the application system. The application program is written in EPROM, and EPROM is installed on the BM4604 board. The power (user-supplied) is turned on and the instructions from the console on BM4604 board are given to operate the system.

TECHNICAL MATERIALS

(1) BM4604B Instruction Manual

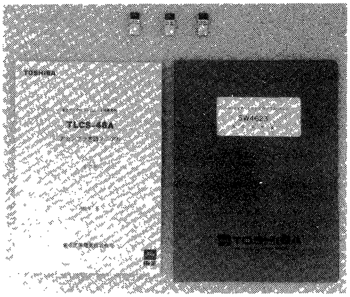
SW4623 TLCS-46A ASSEMBLER/TDS400

GENERAL DESCRIPTION

A cross assembler used for developing the application program of the TLCS-46A. TDS400 (Microcomputer Developing System 400) is used for operating this assembler.

PRODUCT CONFIGURATION

o	EPROM (TMM323) (Assembler program)	3
o	TLCS-46A Assembly language manual	1
o	SW4623 Instruction manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 950 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Decimal and hexadecimal constants can be handled.
(3) 13 reserved symbols are provided for I/O port name or the like.
(4) + and - are available for operators.
(5) Provided are 6 assembler directives and 53 instructions.

(6) Source program can be input from any input device.
(7) Assemble list can be output to any output device.
(8) Object program is created as a ROM image in the TDS400. The TDS400 can output this program in various formats. Direct write in the EPROM is also available.

OPERATING ENVIRONMENT

Processor : TDS400 (48K-byte RAM)
Operating system : System A monitor program (SW4006A)
Peripheral equipment: Console I/O device (e.g. teletypewriter), optional floppy disk, CRT, printer, PTR, PTP, etc.

OPERATING PROCEDURE

This assembler is operated by a command sent to the TDS400 from the console I/O device (or other devices). This assembler program itself should be loaded in the EPROM writer socket on the front panel of the TDS400.

TECHNICAL MATERIALS

(1) TLCS-46A Integrated Circuit Technical Data
(2) TLCS-46A System Manual
(3) TLCS-46A Collected Subprograms
(4) TDS400 System A Operation Manual

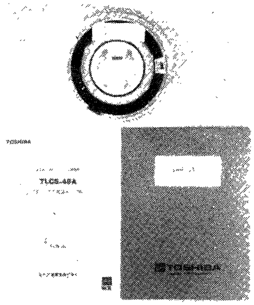
SW4653 TLCS-46A CROSS ASSEMBLER/ACOS

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-46A. General-purpose computer ACOS600 or 700 can be used to load the program.

PRODUCT CONFIGURATION

o Magnetic tape (9 tracks) (Card (IBM29 code) image of source program written in FORTRAN (JIS700, partially GMAP)	1
o TLCS-46A Assembly Language Manual	1
o SW4653 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 500 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 16 reserved symbols are provided for I/O port name or the like.
(4) + and - are available for operators.
(5) Provided are 11 assembler directives and 53 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the paper tape output program ACOS (SW0051), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor : ACOS600 or 700 with more than 64K-words (more than 17-bit words)
Operating system : ACOS-6
Peripheral equipment: Magnetic disk, magnetic tape, line printer, card reader, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First compile this cross assembler to load in the host system. According to the OS of the host machine, define various files, and specify the input and output conditions to perform assembling.

TECHNICAL MATERIALS

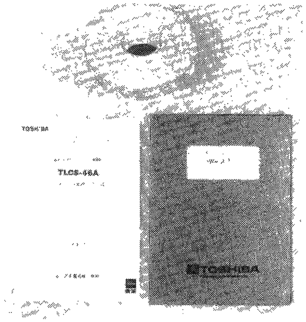
(1) TLCS-46A Integrated Circuit Technical Data
(2) TLCS-46A System Manual
(3) TLCS-46A Collected Subprograms
(4) SW0051 Instruction Manual
(5) ACOS600/700 FORTRAN Language Description
(6) ACOS600/700 Job Control Language (JCL) Description
(7) ACOS600/700 IMCV Edit Program Description

SW4663 TLCS-46A CROSS ASSEMBLER/NOVA

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-46A. Mini-computer NOVA can be used to lead the program.

PRODUCT CONFIGURATION



o Paper tape (ASCII code) (Source program written in FORTRAN (JIS7000))	1
o TLCS-46A Assembly Language Manual	1
o SW4363 Instruction Manual	1

FEATURES AND MAIN SPECIFICATIONS

(1) Up to 100 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 13 reserved symbols are provided for I/O port name or the like.
(4) + and - are available for operators.
(5) Provided are 11 assembler directives and 53 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the different paper tape output program/NOVA (SW0061), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor :	NOVA with more than 32K-words. (16-bit words)
Operating system :	RDOS
Peripheral equipment:	Magnetic disk, line printer, console typewriter, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First compile this cross assembler to load in the host system. According to the OS of the host machine, specify various files to perform assembling.

TECHNICAL MATERIALS

(1) TLCS-46A Integrated Circuit Technical Data
(2) TLCS-46A System Manual
(3) TLCS-46A Collected Subprograms
(4) SW0061 Instruction Manual
(5) NOVA FORTRAN IV Description Manual (Data General)
(6) NOVA RDOS Description Manual, CLI Series (Ditto)
(7) NOVA RDOS Description Manual, System Utility Series (Ditto)

SW4693S, 4693D TLCS-46A ASSEMBLER/iMDS

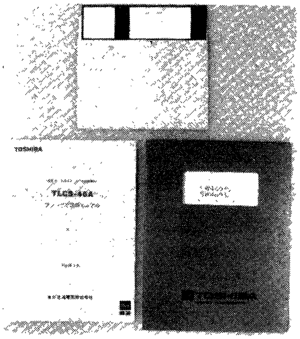
GENERAL DESCRIPTION

A cross assembler used for developing the application program of the TLCS-46A. The intellec MDS is used for operating this assembler.

PRODUCT CONFIGURATION

o Diskette Note) (Assembler program ISIS-II file)	1
o TLCS-46A Assembley Language Manual	1
o SW46935, SW4693D Instruction Manual	1

Note) SW4693A is of single-sided single density.
SW4693D is of single-sided double density.



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 600 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Decimal and hexadecimal constants can be handled.
(3) 13 reserved symbols are provided for I/O port name or the like.
(4) + and - are available for operators.
(5) Provided are 6 assembler directives and 53 instructions.

(6) The source program is input from ISIS-II diskette file.
(7) Assemble list can be output to any output device.
(8) The object program is created on the diskette in format I (Intel hexadecimal format) or format M (masked ROM production format). This program can be punched out to the paper tape by COPY command from ISIS-II, and also can be directly written in the EPROM.

OPERATING ENVIRONMENT

Processor	:	iMDS with more than 32K bytes
Operating system	:	ISIS-II
Peripheral equipment	:	Console I/O device (e.g. teletypewriter), floppy disk, optional CRT, printer, PTR, PTP, PROM programmer, etc.

OPERATING PROCEDURE

Copy the program to the system diskette. Then specify the file and assembler parameter according to the OS from the host machine.

TECHNICAL MATERIALS

(1) TLCS-46A Integrated Circuit Technical Data
(2) TLCS-46A System Manual
(3) TLCS-46A Collected Subprograms
(4) ISIS-II System User's Guide (Intel)

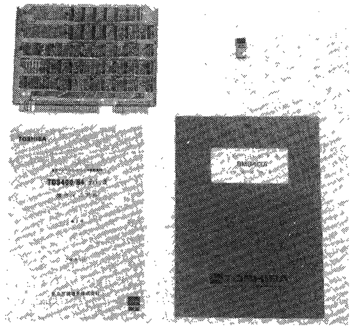
BM8401A TDS400/84 CONTROLLER

GENERAL DESCRIPTION

A special module for making the TDS400 to the TDS400/84 debugger. It can debug the application systems (software and hardware) of the TLCS-84 by connecting BM8403 (TLCS-84 emulation board).

PRODUCT CONFIGURATION

o	Board module	1
o	EPROM (TMM323) (Control program)	1
o	TDS400/84 Debugger operation manual	1
o	BM8401A Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Application program can be executed with real-time operation by loading it in the dedicated program memory (RAM) within this board or in the EPROM on BM8403.

(2) Programs can be transmitted to and from EPROM or the back-up memory (a part of TDS400 main memory). Information stored in the back-up memory can be output to the paper tape, etc. in various formats (TDS400 function).

(3) Direct read/write of the registers and RAM contained in the CPU, and program memory (corresponding to ROM) within this board can be performed.

(4) Various functions, such as execution and halt of programs, address stopping, single-step operation, and dummy interrupt signal input, are provided.

(5) Module dimensions are 180mm × 230mm × approx. 15mm with edge connector of 144 pins.

(6) Approx. power dissipation is 5V × 1A.

OPERATING CONDITIONS

This board is inserted in slot "43C/46C" within the TDS400 (BM4006A). The control program is loaded in the BM4008 board of slot "MEM1" where EPROM No. 7 is mounted after having removed EPROM. BM8403 and application system (board) are connected to this board, which is operated by controlling the TDS400 debugging console.

TECHNICAL MATERIALS

(1) BM4006A Instruction Manual
(2) BM8403 Instruction Manual
(3) TDS400 System A Operation Manual

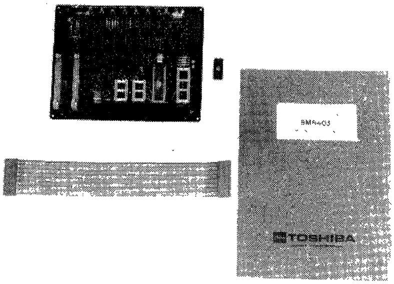
BM8403 TLCS-84 EMULATION BOARD

GENERAL DESCRIPTION

This is an emulation board that can be performed as equivalent to the single-chip microcomputer LSI of TLCS-84 series. Interface circuit is provided in this board for being connected to the host system (TDS400, EM8404). EPROM (for 4K bytes) can be installed as a program ROM.

PRODUCT CONFIGURATION

o	Board module	1
o	EPROM (TMM323)	1
o	LSI socket connection cable	2
o	BM8403 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) This is an emulation board that can be replaced directly with TM8048 or TMP8049. The emulator chip is TMP8039-6.
(2) 4K-byte EPROM can be installed as a program ROM.
(3) Interface circuit to be connected to the TDS400 (BM8401A) or BM8404, and free space (for mounting application system circuit or interface) are provided in the board.
(4) As a program memory, this emulation board can select EPROM within the board or memory in the host system to execute the application program stored on these units.
(5) This emulation board has a function capable of executing the application program while monitoring or modifying the internal status on of the microcomputer under the control by the host system.
(6) When the emulation board is disconnected from the host system, it is also operated as an independent emulation board.
(7) Module dimensions are 180mm × 230mm × approx. 30mm.
(8) Power dissipation is less than 5V (±5%) × 0.5A.

OPERATING CONDITIONS

EPROM, in which the application program has been written, is mounted on this board, and two cables from the board is inserted in the sockets of the micro-computer LSI of application system. The power supply (user-supplied) and host system (TDS400 or BM8404) are connected to the board for operation.

TECHNICAL MATERIALS

(1) TLCS-84 Integrated Circuit Technical Data
(2) TDS400/84 Debugger Operation Manual, BM8401A Instruction Manual
(3) BM8404 Debugging Board Operation Manual, BM8404 Instruction Manual

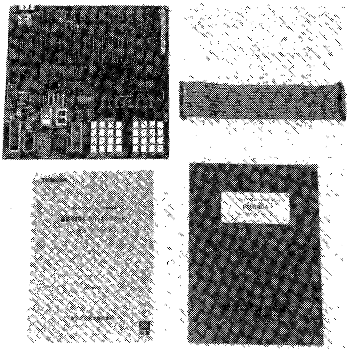
BM8404 TLCS-84 DEBUGGING BOARD

GENERAL DESCRIPTION

This board is a control board for debugging of the application system (or program) of the TLCS-84. The debugging system of the TLCS-84 is made up by connecting it to the BM8403 (TLCS-84 emulation board).

PRODUCT CONFIGURATION

o Board module	1
o BM8403 Connection cable	1
o BM8404 Debugging board Operation Manual	1
o BM8404 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) BM8404 board is provided with a console unit consisting of a display for addresses, data and system condition, and 32 keys.
(2) Application program is executed by emulator chip (TMP8035, TMP8039-6) of BM8403 after having been transmitted to the EPROM program memory within this board via the EPROM socket.
(3) Direct read/write of register or RAM within the microcomputer LSI is available.
(4) Application program can be modified with machine code level.
(5) Application program can be started with any address and stopped at any time.
(6) Address stop and single step operation functions are provided.
(7) Dummy interrupt signals can be generated from the console on the board.
(8) Module dimensions are 300mm × 300mm × approx. 30mm.
(9) Power dissipation is less than 5V (±5%) × 2.5A.

OPERATING CONDITIONS

This board is connected with BM8403 that have been connected to the application system. The application program is written in EPROM, and EPROM is installed on the BM8404 board. The power (user-supplied) is turned on and the instructions from the console on BM8404 board are given to operate the system.

TECHNICAL MATERIALS

(1) BM8403 Instruction Manual

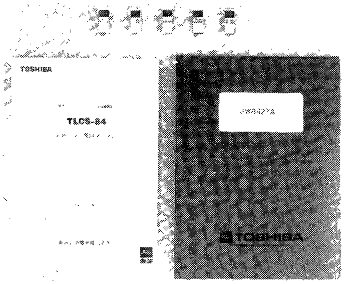
SW8423A TLCS-84 ASSEMBLER/TDS400

GENERAL DESCRIPTION

A cross assembler used for developing the application program of the TLCS-84. The TDS400 (Microcomputer Developing System 400) is under for operating this assembler.

PRODUCT CONFIGURATION

o EPROM (TMM323) (Assembler program)	5
o TLCS-84 Assembly Language Manual	1
o SW8423A Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 950 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Decimal and hexadecimal constants can be handled.
(3) 34 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 6 assembler directives and 108 instructions.

(6) Source program can be input from any input device.
(7) Assemble list can be output to any output device.
(8) Object program is created as a ROM image in the TDS400. The TDS400 can output this program in various formats. Direct write in the EPROM is also available.

OPERATING ENVIRONMENT

Processor :	TDS400 (48K-byte RAM)
Operating system :	System A monitor program (SW4006A)
Peripheral equipment:	Console I/O device (e.g. teletypewriter), optional floppy disk, CRT, printer, PTR, PTP, etc.

OPERATING PROCEDURE

This assembler is operated by a command sent to the TDS400 from the console I/O device (or other devices). This assembler program itself should be loaded in the EPROM writer socket on the front panel of the TDS400.

TECHNICAL MATERIALS

(1) TLCS-84 Integrated Circuit Technical Data
(2) TDS400 System A Operation Manual

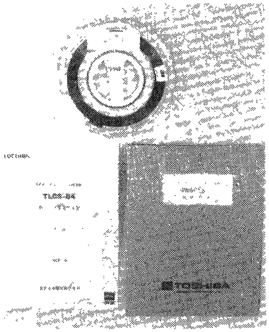
SW8453 TLCS-84 CROSS ASSEMBLER/ACOS

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-84. General-purpose computer ACOS600 or 700 can be used to load the program.

PRODUCT CONFIGURATION

o	Magnetic tape (9 tracks) (Card (IBM29 code) image of source program written in FORTRAN (JIS7000, partially GMAP)	1
o	TLCS-84 Assembly Language Manual	1
o	SW8453 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 500 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 34 reserved symbols are provided for register name or the like.
(4) + and - are available for operators.
(5) Provided are 11 assembler directives and 108 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the paper tape output program ACOS (SW0051), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor	: ACOS600 or 700 with more than 64K-words (more than 17-bit words)
Operating system	: ACOS-6
Peripheral equipment	: Magnetic disk, magnetic tape, line printer, card reader, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First, compile this cross assembler to load in the host system. According to the OS of the host machine, define various files, and specify the input and output conditions to perform assembling.

TECHNICAL MATERIALS

(1) TLCS-84 Integrated Circuit Technical Data
(2) SW0051 Instruction Manual
(3) ACOS600/700 FORTRAN Language Description
(4) ACOS600/700 Job Control Language (JCL) Description
(5) ACOS600/700 IMCV Edit Program Description

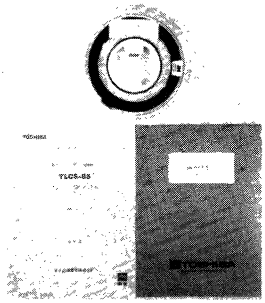
SW8553 TLCS-85 CROSS ASSEMBLER/ACOS

GENERAL DESCRIPTION

This cross assembler is used when developing the application program of the TLCS-85. General-purpose computer ACOS600 or 700 can be used to load the program.

PRODUCT CONFIGURATION

o	Magnetic tape (9 trucks) (Card (IBM29 code) image of source program written in FORTRAN (JIS 7000, partially GMAP))	1
o	TLCS-85 Assembly Language Manual	1
o	SW8553 Instruction Manual	1



FEATURES AND MAIN SPECIFICATIONS

(1) Up to approx. 500 symbols (name) each of which consists of less than 6 characters can be defined.
(2) Binary, octal, decimal, and hexadecimal constants can be handled.
(3) 10 reserved symbols are provided for register name or the like.
(4) 11 arithmetic and logical operators are available.
(5) Provided are 13 assembler directives and 113 instructions.

(6) Punched card or paper tape can be used for the input medium of the source program.
(7) Assemble list can be output to a line printer.
(8) Object program is created in the host system. Using the paper tape output program ACOS (SW0051), this program can be output to paper tapes in various formats.

OPERATING ENVIRONMENT

Processor : ACOS600 or 700 with more than 64K-words (more than 17-bit words)
Operating system : ACOS-6
Peripheral equipment: Magnetic disk, magnetic tape, line printer, card reader, paper tape reader, (and paper tape punch), etc.

OPERATING PROCEDURE

First, compile this cross assembler to load in the host system. According to the OS of the host machine, define various files, and specify the input and output conditions to perform assembling.

TECHNICAL MATERIALS

(1) TLCS-85 Integrated Circuit Technical Data
(2) SW0051 Instruction Manual
(3) ACOS600/700 FORTRAN Language Description
(4) ACOS600/700 Job Control Language (JCL) Description
(5) ACOS600/700 IMCV Edit Program Description

OVERSEAS OFFICES

U.S.A.:

TOSHIBA AMERICA, INC.

Chicago Office
2900 MacArthur Boulevard, Northbrook, Ill.
60062, U.S.A.
Cable: TOSHIBA CHICAGO, Telex: 35-4407,
Tel: (312)564-1200

Irvine Office
2151 Michelson Drive, Suite 190, Irvine, Calif.
92715, U.S.A.
Telex: 910-595-1127 TSBIRIN,
Tel: (714)955-1155

WEST GERMANY:

TOSHIBA DEUTSCHLAND G.m.b.H.

Head Office
Hammer Landstrasse 115, 4040 Neuss 1,
F. R. Germany
Telex: 8517926, Tel: (02101)1981

Muenchen Office
Dom Pedrostrasse 17, 8000 Muenchen 19
Telex: 05213363, Tel: (089)18 50 22

UNITED KINGDOM:

TOSHIBA (UK) LTD.
Toshiba House, Frimley Road, Frimley,
Camberley, Surrey Gu 16 5JJ, England
Telex: 858798, Tel: Camberley (0276)62222

FRANCE:

TOSHIBA CORPORATION—Paris Office
Tour Neptune 23rd Floor, Cedex No. 20,
92086 Paris La Defense , France
Telex: 613351 F, Tel: 773-7272

SWEDEN:

TOSHIBA CORPORATION—Stockholm Office
Banergatan 23, S-115 22 Stockholm, Sweden
Telex: 14169 TSB STK S, Tel: 08-616265

HONG KONG:

TÓSHIBA ELECTRONICS HONG KONG LTD.
Suite 423-5, Ocean Centre, Canton Road,
Kowloon, Hong Kong
Cable: TSBELHK, Telex: 38501 TSBEH HX,
Tel: 3-671141

TAIWAN:

TSU-RONG CORPORATION
Rm N-813, Chia Hsin 2-Bldg., 96 Sec. 2,
Chung Shan N. Rd., Taipei, Taiwan
Telex: 26874 TRCTP, Tel: 531-3934

SINGAPORE:

TOSHIBA SINGAPORE PTE., LTD.
2405/6 Orchard Towers, 400 Orchard Road,
Singapore 0923
Telex: RS 23892, Tel: 7373911

KOREA:

KOREA ELECTRONICS CO., LTD.
189-2 Guro-Dong, Guro-Gu, Seoul, Korea
Cable: KOREA TOSHIBA COM,
Telex: KECOS K27424, Tel: 853-3161

AUSTRALIA:

TOSHIBA CORPORATION—Sydney Office
14th Floor, Gold Fields, House No.1,
Alfred Street, Sydney Cove, Box R219, Sydney
N.S.W. 2000, Australia
Cable: TOSHIBA SYDNEY, Telex: 22268,
Tel: 27-4301~5

BRASIL:

TOSHIBA BRASILEIRA REPRESENTACOES LTDA.
Rua Peixoto Gomide, No.996-1, Andar,
Conj-130/140, Jardim Paulista, São Paulo, Brasil
Cable: TOSHIBA SPO, Telex: 011-22298,
Tel: 283-4511, 4714, 4964

The information in this guide has been carefully checked and is believed to be reliable, however, no responsibility can be assumed for inaccuracies that may not have been caught. All information in this guide is subject to change without prior notice. Furthermore, Toshiba cannot assume responsibility for the use of any license under the patent rights of Toshiba or any third parties.

Toshiba TOSHIBA

TOSHIBA CORPORATION TOKYO JAPAN

1-6, UCHISAIWAICHO 1-CHOME, CHIYODA-KU, TOKYO, 100, JAPAN
TELEX: J22587 TOSHIBA CABLE: TOSHIBA TOKYO TEL: 508-2704

PRINTED IN JAPAN