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.

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)

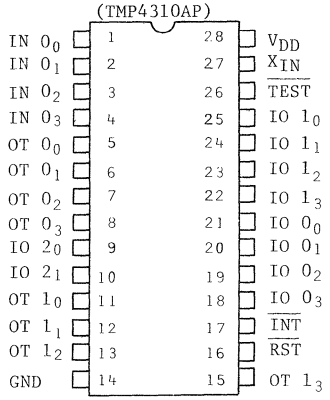
TABLE OF FEATURES

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			

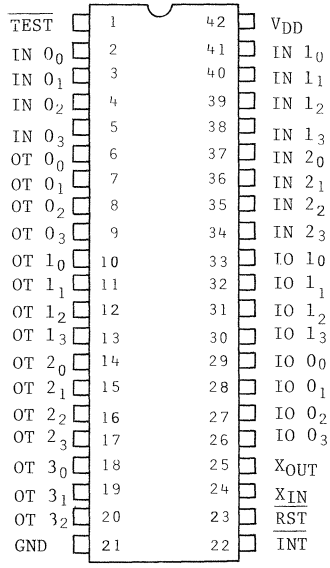
PIN CONNECTIONS

Top View

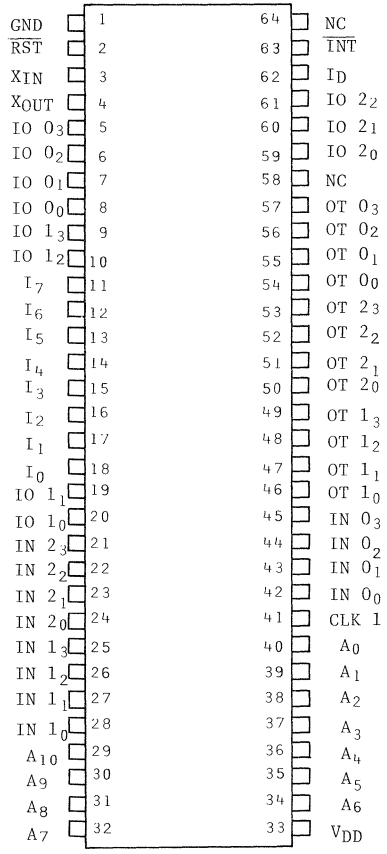
(TMP4310AP)



(TMP4315BP/TMP4320AP)



(TMP4300C)



PIN NAMES & PIN DESCRIPTION

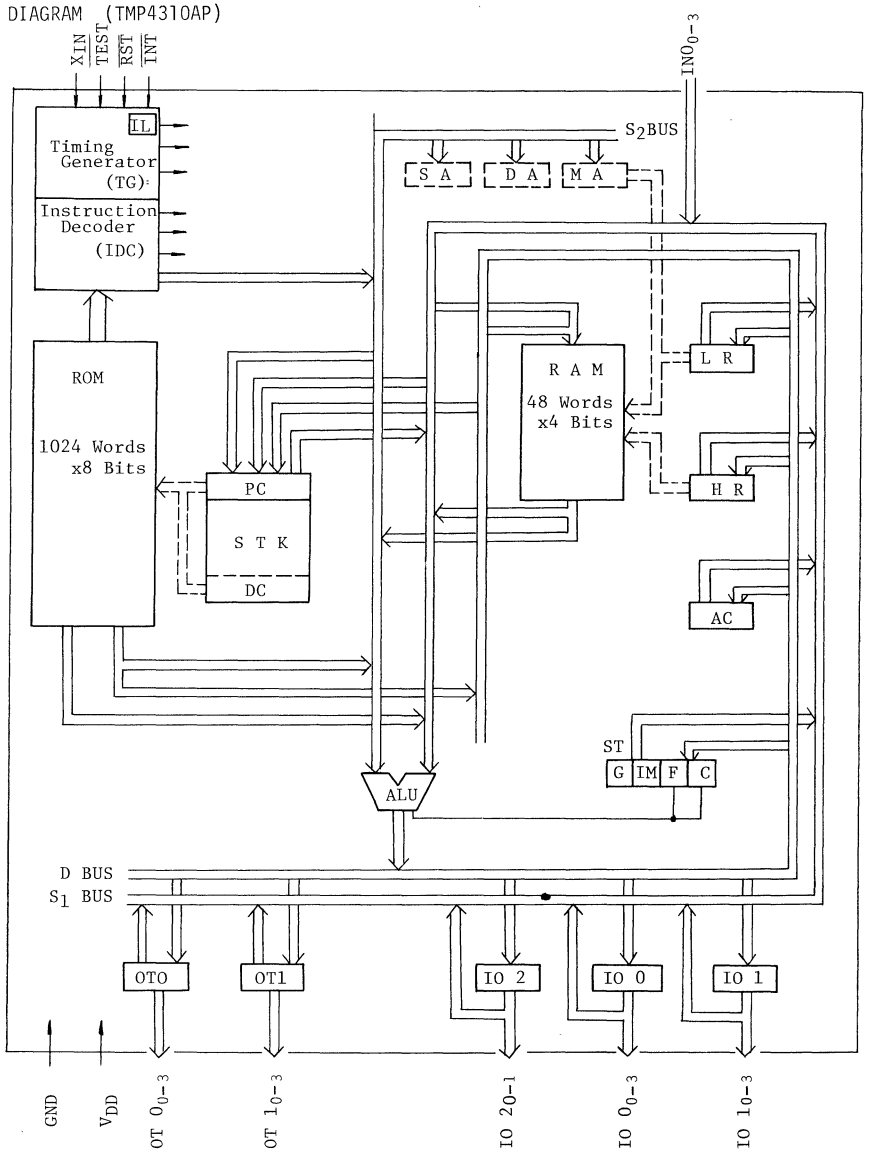
Pin Name	Input/ output	Function	4310	4315/ 4320	4300
IN0o-IN03	Input	4-bit Input Port IN0	○	○	○
IN1o-IN13	Input	4-bit Input Port IN1	—	○	○
IN2o-IN23	Input	4-bit Input Port IN2	—	○	○
OT0o-OT03	Output	4-bit Output Port OT0	○	○	○
OT1o-OT13	Output	4-bit Output Port OT1 Large sink current (IOL TYP=20mA, VOL=2V) 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	○	—	○
$\overline{RST}$	Input	Initialize Signal Input The initialize operation is performed by placing $\overline{RST}$ terminal at low level for more than four clock cycles.	○	○	○
$\overline{INT}$	Input	Interrupt Request Signal Input The interrut request is accepted by placing $\overline{INT}$ terminal at low level for more than our clock cycles. The repetetive interrupt should be requested, after keeping $\overline{INT}$ terminal at high level for two clock cycles or more.	○	○	○
$\overline{TEST}$ (Note 1)	Input	LSI Test Signal Input $\overline{TEST}$ 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)	○	○	○

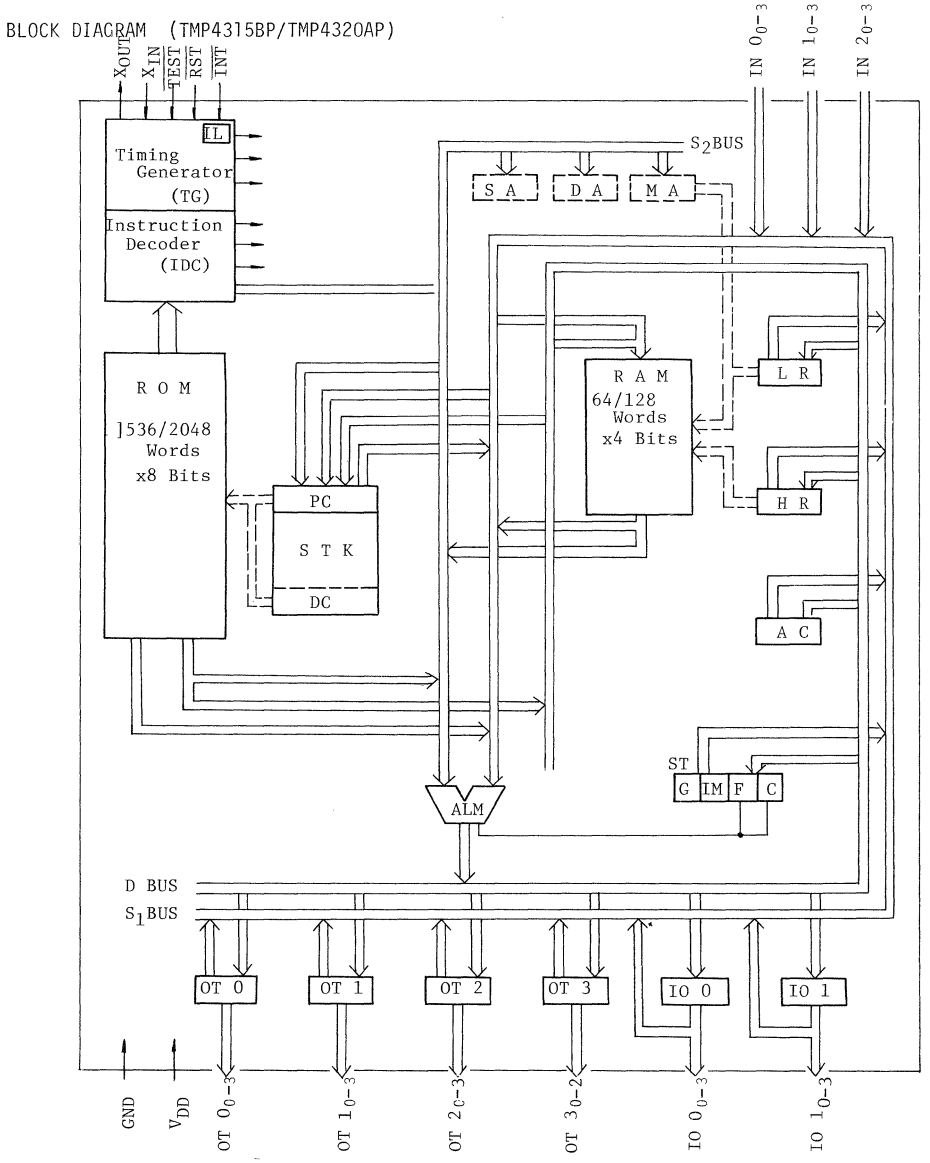
Note 1 The basic clock of TMP4310AP

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

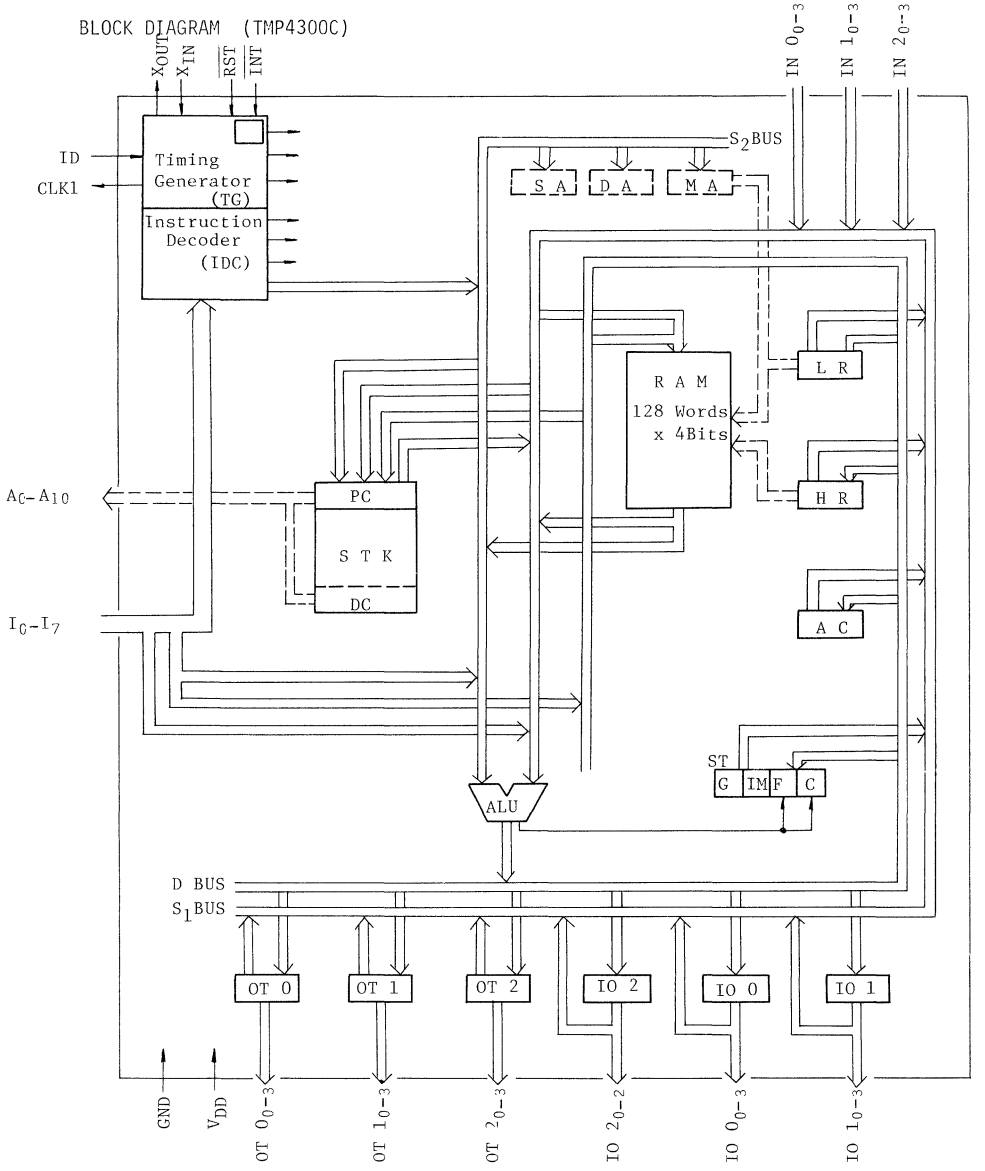
BLOCK DIAGRAM (TMP4310AP)



BLOCK DIAGRAM (TMP4315BP/TMP4320AP)



BLOCK DIAGRAM (TMP4300C)



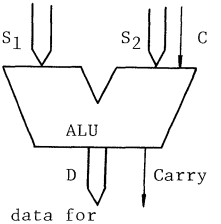
FUNCTIONAL DESCRIPTION

[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.

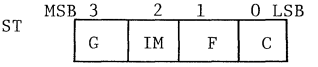


2. Accumulator (AC)

   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).



   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)

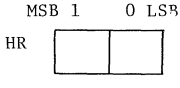
L Register (lower address register) is a four bit register which indicates the lower order 4 bits of RAM address and is used in conjunction with H Register for addressing RAM.

5. H Register (HR)

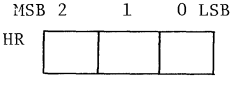
H Register (Higher address register) is a two bit or three bit register which indicates the higher order two or three bits of RAM address, and used in conjunction with L Register for addressing RAM.

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)



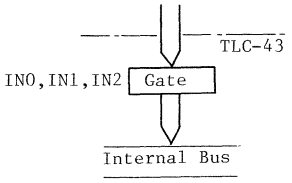
(TMP4320AP/TMP4300C)



6. Input Port (IN0, IN1, IN2)

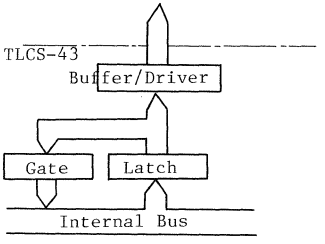
All of IN0, IN1 and IN2 ports are the dedicated input ports having four bit configuration and read the data sent from outside.

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.



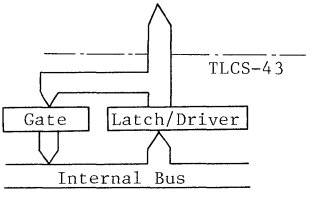
8. Input/Output Port (IO0, IO1, IO2)

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.

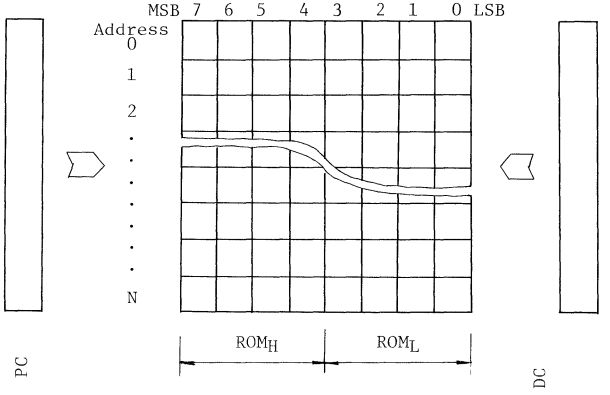
And when the data counter is being used as stack level 3, if setting operation is performed to the data counter, the content as stack level 3 is destroyed.

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.

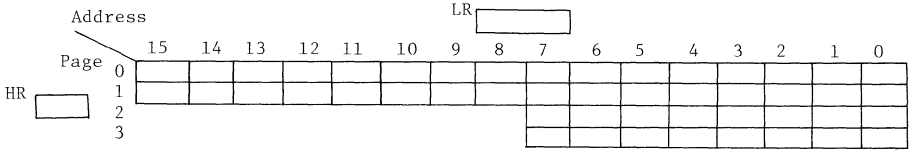
13. Read/Write Memory (RAM)

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

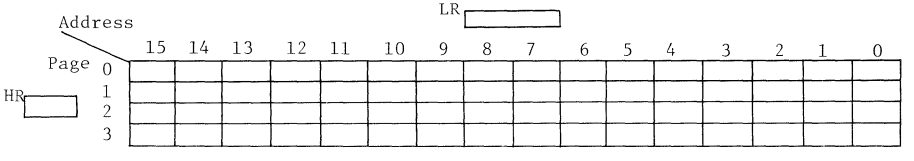
RAM has the maximum configuration of 128 words x 4 bits and is addressed by H Register which designates a page and L Register which designates an address in a page.

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.

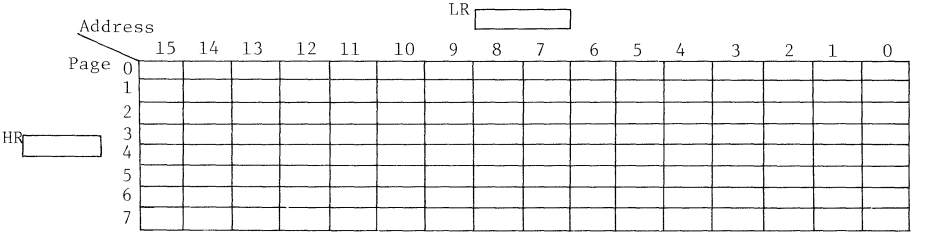
(TMP4310AP)



(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

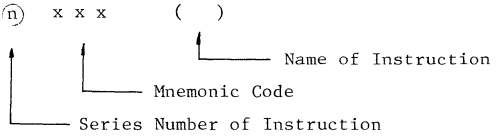
One of the features of the TLCS-43 Machine Instruction set is the existance of index instructions. In the case of processing instructions, usually the source and destination of data have been inherently defined. The index instructions modify the processing instructions to change the source of data to 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) maintaing 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.



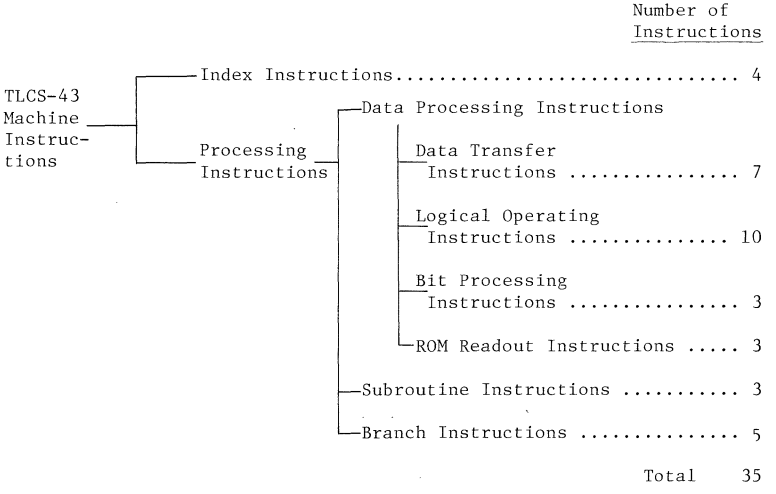
<Symbol Instruction> Mnemonic Operation Code Operand

[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.

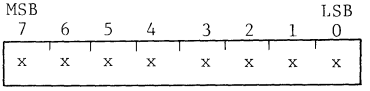
Among the machine instructions of TLCS-43, 30 instructions are of 1-byte length and 5 instructions are of 2-byte length. As regards the execution time of machine instructions, 28 instructions are of 1-machine cycle and 7 instructions are of 2-machine cycle.

Machine Instructions are classified by their functions as follows:

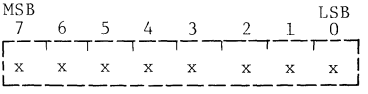


< Machine Code >

No.1 Byte



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



(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 ← 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 :

(O) 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.

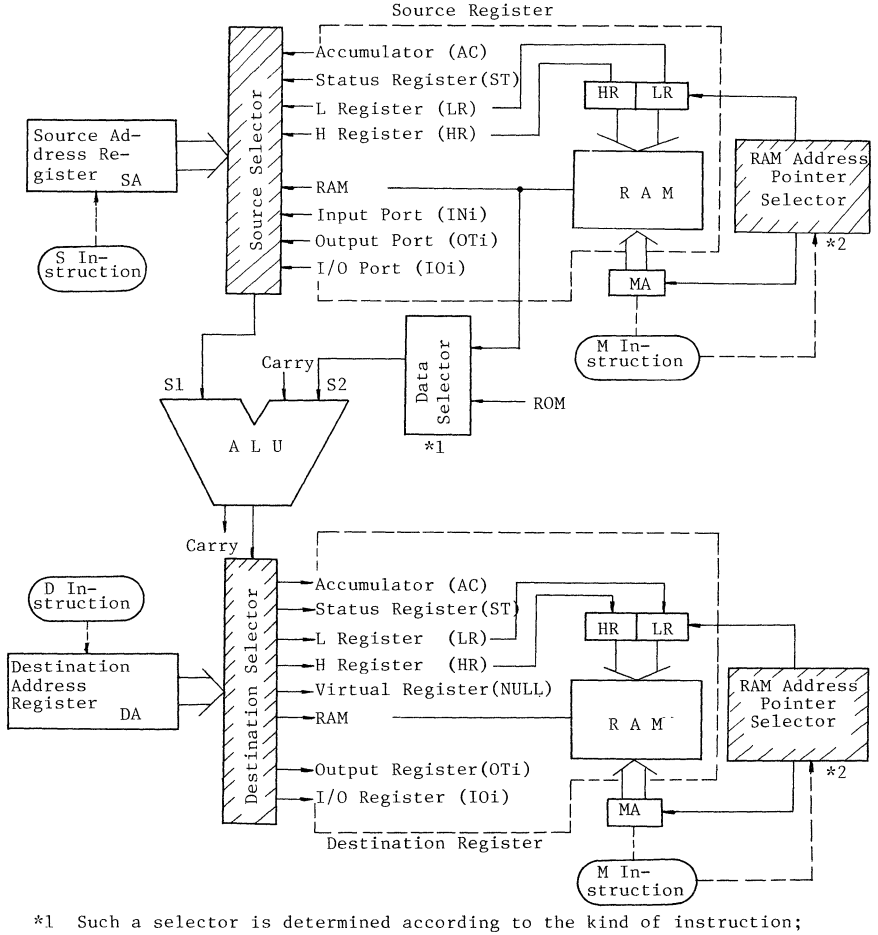
The address pointer of RAM has HR·LR and memory address register MA. Usually HR·LR is selected, but when modification is made by M instruction, MA is selected. The memory address is stored in MA by M instruction. Since either HR·LR or MA is used as address pointer, if RAM is used as source and destination 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	IN0	In. Port 0	○	○	○	○
5	NULL	NULL	○	○	○	○
6	IO0	I/O Port 0	○	○	○	○
7	IO1	I/O Port 1	○	○	○	○
8	OT1	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".

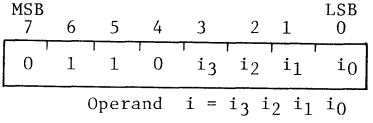
< Modifiable Index Instruction > S, D, SD, M

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

< Symbol Instruction >

LDI i (0 ≦ i ≦ 15)

< Machine Code >



< 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.

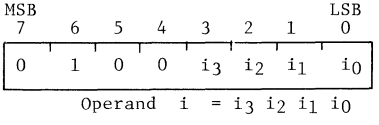
< Modifiable Index Instruction > D, M

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

< Symbol Instruction >

LLI i (0 ≦ i ≦ 15)

< Machine Code >



< 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) ← $\overline{(AC)}$ + 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)←M[(H·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) ← (AC) / $\overline{M[(H\cdot L)]}$

< 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) ← (AC) $\vee$ M[(H·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 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) ← (AC)∀M[(H·L)]

< Status Flag > if (AC)=0 then (F)←1, else (F)←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) ← (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 ≦ i ≦15)

< Machine Code >

MSB 7	6	5	4	3	2	1	LSB 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) ← (ST) ∨ Ones < b > (ST < b > ← 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".

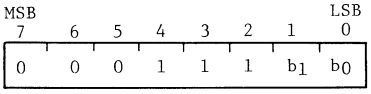
< Modifiable Index Instruction > SD, M

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

< Symbol Instruction >

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

< Machine Code >



Operand b = b_1 b_0

< Function > (ST) ← (ST) ∧ $\overline{\text{Ones}}$ < b > (ST < b > ← 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

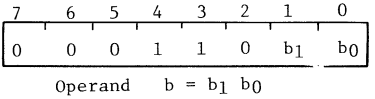
(24) LFB (Load complemented status Bit to F)

Load and Complement Status Bit to F Instruction

< Symbol Instruction >

LFB b (3 ≥ b ≥ 0)

< Machine Code >



Operand b = b_1 b_0

< Function > (F) ← $\overline{\text{ST}}$ < b >

< 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 7	6	5	4	3	2	1	LSB 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 >

$(DC_H) \leftarrow a_{10} a_9 a_8$
$(DC_M) \leftarrow a_7 a_6 a_5 a_4 \vee (ZR)$
$(DCL) \leftarrow M[(H \cdot L)]$

i.e. $(DC) \leftarrow a + M[(H \cdot L)]$

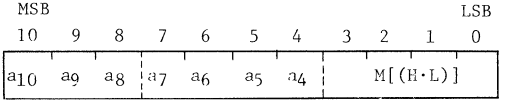
< Status Flag > (F) : No change
(C) : No change

< 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



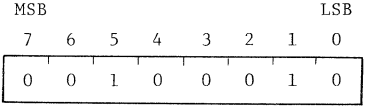
< Modifiable Index Instruction > S, M

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

< Symbol Instruction >

LRL

< Machine Code >



< Function > (AC) ← 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) ← ROM_H [(DC)]
(DC) ← (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

1st Byte

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

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

< 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=$i_3 i_2 i_1 i_0$

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.

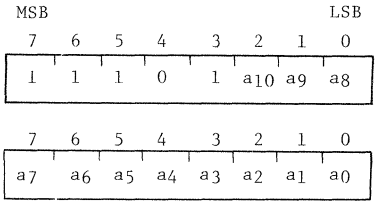
< Modifiable Index Instruction > No instruction

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

< Symbol Instruction >

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

< Machine Code >



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.

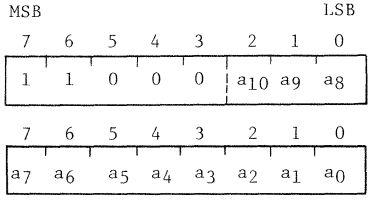
< Modifiable Index Instruction > No instruction

(35) JMP (Jump) : Unconditional Jump Instruction

< Symbol Instruction >

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

< Machine Code >



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

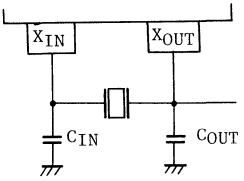
< Function >	(PC)←a
< Status Flag >	(F) : No change (C) : No change
< Executio Cycle >	2 Machine cycle
< Explanation of Function >	This instruciton causes an unconditional branch to the address "a" indicated by the lower order 11 bits of the instruction.
< Modifiable Index Instruction >	No instruction

[OPERATION DESCRIPTION]

1. Basic Clock

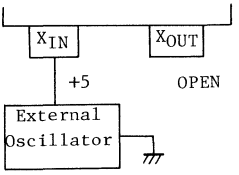
The basic clock generates the basic timing sequences required for the operations of TLCS-43. There are the following methods of options generating the basic clock.

(1) Direct Connection of the Oscillator



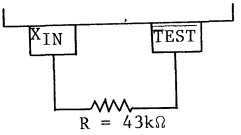
When a crystal oscillator, or a ceramic oscillator, or a IFT, is connected as shown at the left, the frequency of TLCS-43 basic clock is equal to that of the oscillator. (Except TMP4310AP)

(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.

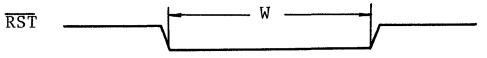
(3) Internal Oscillation



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.



$0 \leqq W < 1.5$ cycles	1.5 cycles $\leqq W < 4$ cycles	4 cycles $\leqq W$
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.

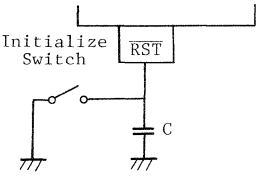
The initialize operation performs the following functions.

Block	Symbol	Initialize Function
Program counter	PC	Cleared to "0"
Interrupt latch	IL	Cleared
Interrupt flag	IM	Cleared to "0" disabling interrupt.
Output port	OT0	All bits are set to "1".
	TO1	All bits are set to "1".
	OT2	All bits are set to "1".
	OT3	All bits are set to "1".
Input/Output port	IO0	All output bits are set to "1".
	IO1	All output bits are set to "1".
	IO2	All output bits are set to "1".
Stack	STK	Made empty.

While $\overline{RST}$ 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 $\overline{RST}$ 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 $\overline{RST}$ 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 $\overline{INT}$ 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.

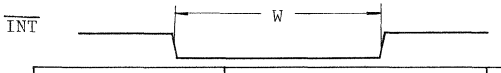
1. Interrupt flag IM has been set and one or more instructions have veen executed after the flag was set.
2. Four levels of subroutine nesting have not been performed.

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, $\overline{INT}$ 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 $\overline{INT}$ terminal at the low level.

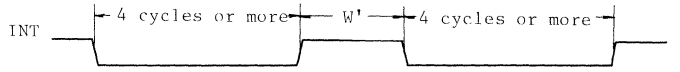
Setting and repetitive setting of IL by $\overline{INT}$ signal are performed in the following timings.

(Setting Timing of IL).



$0 \leqq W < 1$ cycle	1 cycle $\leqq W < 4$ cycles	4 cycles $\leqq W$
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.

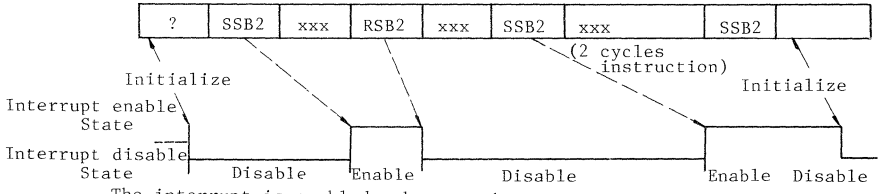
When an interrupt is accepted, the content of program counter PC is pushed down to stack STK and the entry address (address 2) of the interrupt service routine is set in the program counter.

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.

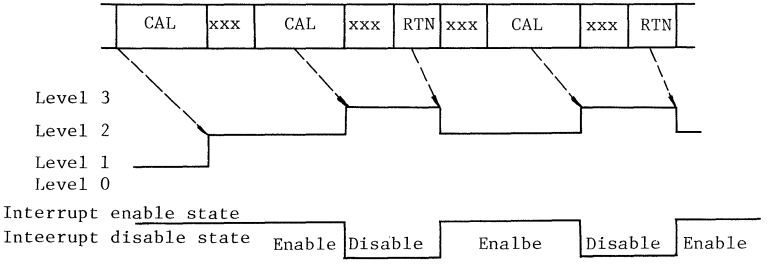
(2) Timing of Interrupt Enable/Disable

(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.

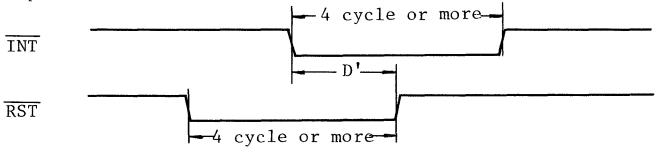
② When stack is used up to the deepest level (level 3)
(Assumption: IM flag is set "1" constantly)



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 $\overline{INT}$ Signal and $\overline{RST}$ 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 $\leq$ D' < 5 cycles ... Acceptance of the interrupt request is not definite after the initialize operation.

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

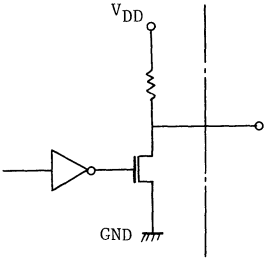
Where one cycle is equal to one cycle of the basic clock.

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.

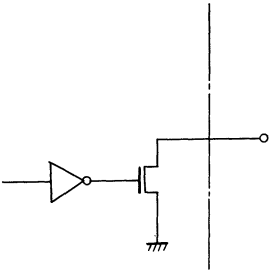
For the electrical characteristics, refer to the paragraphs of ABSOLUTE MAXIMUM RATINGS and DC CHARACTERISTICS. Since the pull-up resistors are provided by MOS transistors, the characteristics are somewhat different from normal resistors.



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

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

(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.)

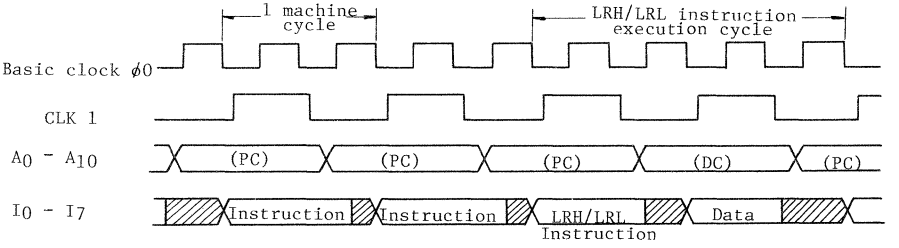
[EVALUATOR CHIP DESCRIPTION]

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

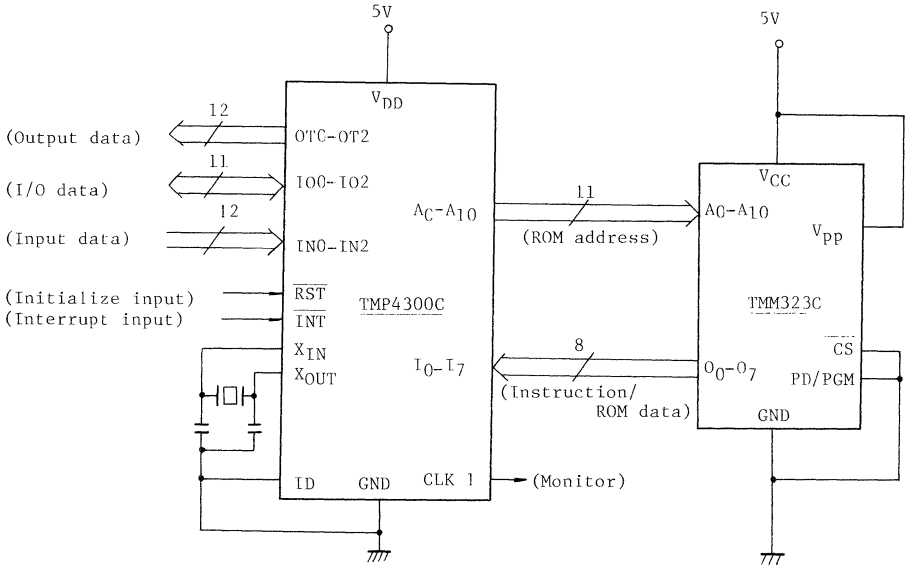
The normal operation timing of TMP4300C is shown in the figure below. The timings of the initialize operation and the interrupt operation are exactly same as the operation timings of other TLCS-43 chips.



2. Example of TMP4300C Application

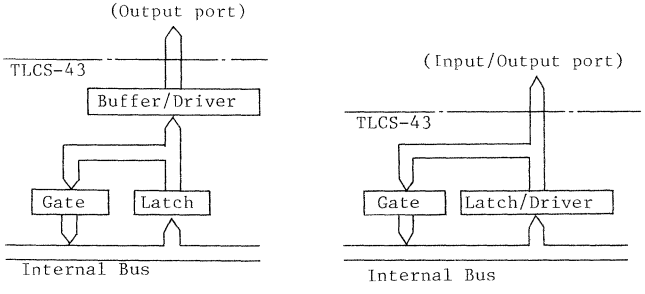
The diagram below illustrates an example of connection with an EPROM, which allows the program confirmation before confirming program to ROM.

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



3 Caution for Using IO2 Port

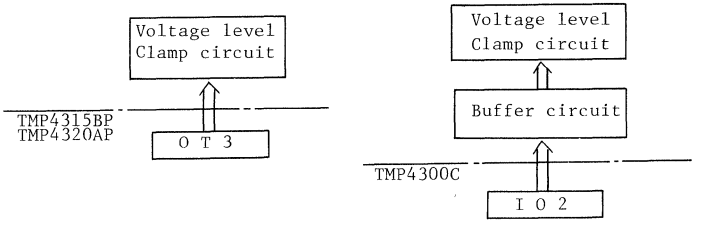
The output port and the input/output port of TLCS-43 have the configurations shown below.



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 10 sec)	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 $\pm$ 10 %, Unless Otherwise Noted

Symbol	Parameter		Test Conditions	Min.	Typ.	Max.	Units
V_{IH1}	Input High Voltage (IN0, IO0, 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 (IN0)		$V_{IN}=V_{DD}$			20	μA
I_{IN2}	Input Current (IO0, IO1, IO2)	Open Drain	$V_{IN}=V_{DD}$			20	μA
		Pull Up				-	-
I_{IL1}	Input Low Current (IO0, 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, OTl)	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 OTl 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
fOSC	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

'C 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 (INO, IN1, IN2, IOO, 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 (INO, IN1, IN2)		$V_{IN}=V_{DD}$			20	μA
I_{IN2}	Input Current(IOO, IO1)	Open Drain	$V_{IN}=V_{DD}$			20	μA
		Pull up				-	-
I_{IL1}	Input Low Current (IOO, 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 (OTO, 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 XOUT)		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}	$\overline{INT}$ Low Level Pulse Width		4			Clock Cycle
t_{RST}	$\overline{RST}$ 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 $\overline{INT}$, X_{IN}) (Note)		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_{IN}	Input Current (Except $\overline{RST}$, $\overline{INT}$)	$V_{IN}=V_{DD}$			20	μA
I_{IL2}	Input Low Current ($\overline{RST}$, $\overline{INT}$)	V_{IN}=0.6V			-0.1	mA
I_{LO}	Output Lead Current(OTO,OT1,OT2)	$V_{OUT}=V_{DD}$			20	μA
V_{OH}	Output High Voltage (AO-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.

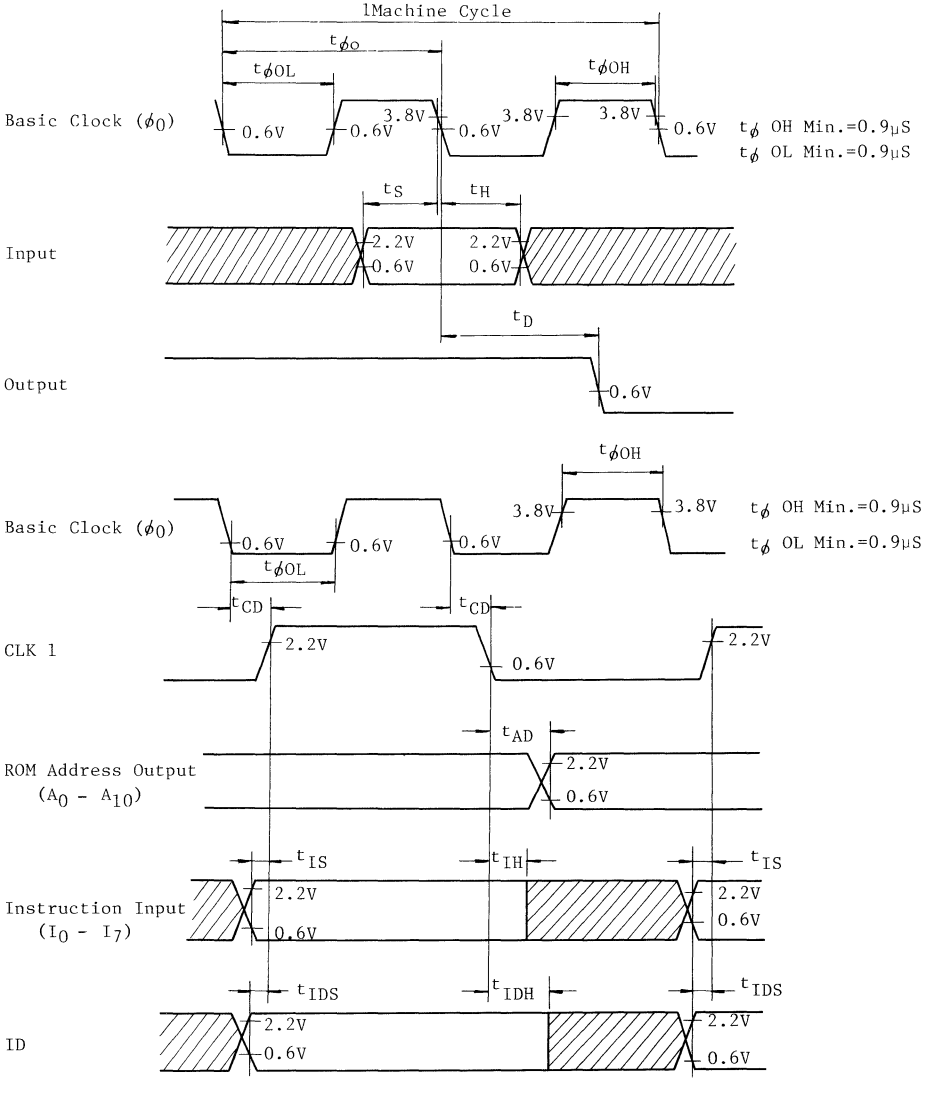
V_{OL} Max.=0.5V to 0.6V (I_{OL}=1.6mA)
V_{IH1} Min.=2.3V to 2.4V

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

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

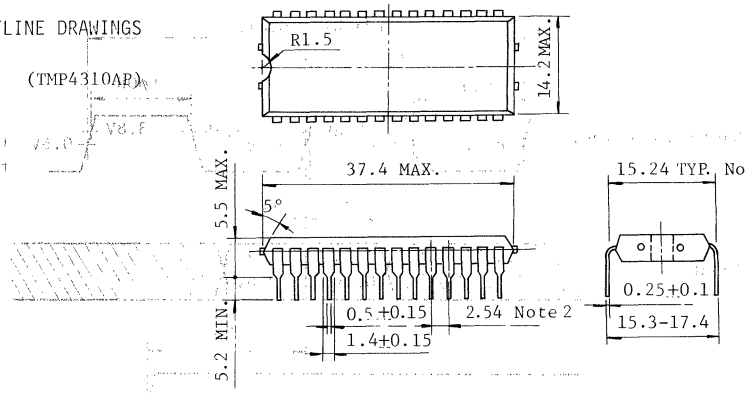
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units.
$t_{\phi o}$	Clock Cycle Time		2		5	μs
t_S	Input Set up Time		0.9			μs
t_H	Input Hold Time		0.9			μs
t_D	Output Delay Time	C_L=50PF,R(Pull up)=50kΩ ,1 TTL			1.8	μs
t_{INT}	$\overline{INT}$ Low Level Pulse Width		4			Clock Cycle
t_{RST}	$\overline{RST}$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

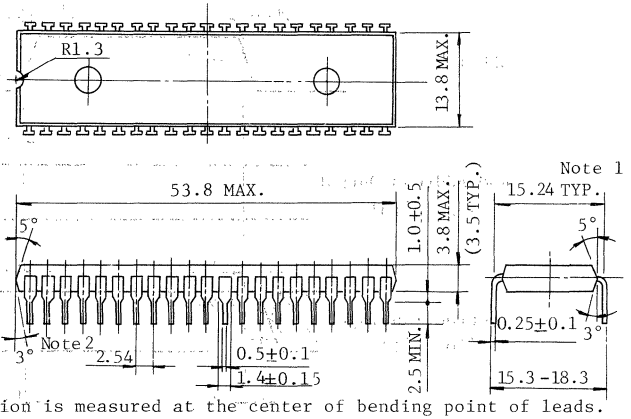
(TMP4310AP) Unit in mm



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.

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.

Unit in mm

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.)
RXXXX ;	------- Outputs the program start address following "R", in four frames of decimal ASCII code.
x xx P_X;	------- Data and check sum of the first address.
X XX P_X;	------- Data and check sum of the second address.
. . .	. . .
X XX P_X; (CR) (LF)	------- Data and check sum of the eighth address.

PUNNNi
- Bit Name
- Port Name

X XX P_X
- P_X: 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 P_X;	------ 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".
.	
.	
.	
.	
.	