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.

Toshiba 東芝 INTEGRATED CIRCUIT

TECHNICAL DATA

TCP4620AP
TCP4630AP

FEATURES

o TCP4620AP

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

o TCP4630AP

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

o TCP4600AP
Evaluator Chip
for TLCS-46A

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

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

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

o 52 Instructions
   46 One Cycle Instructions
   6 Two Cycle Instructions

o Single Level Subroutine Nesting

o Single Level External Interrupt

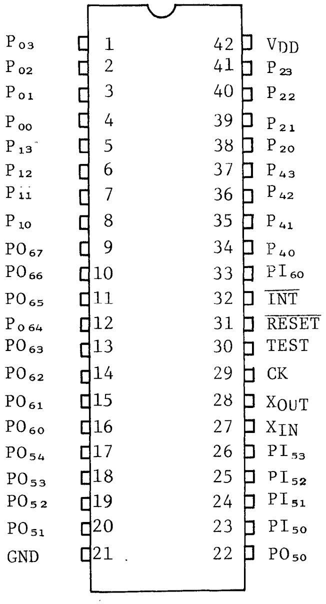
o 10μs Instruction Execution Time

o PLA and Decode Matrix for Display Operation

o Many Kinds of Mask Options for Optimum Application Systems

TCP4620AP
TCP4630AP

PIN CONNECTIONS (TOP VIEW)

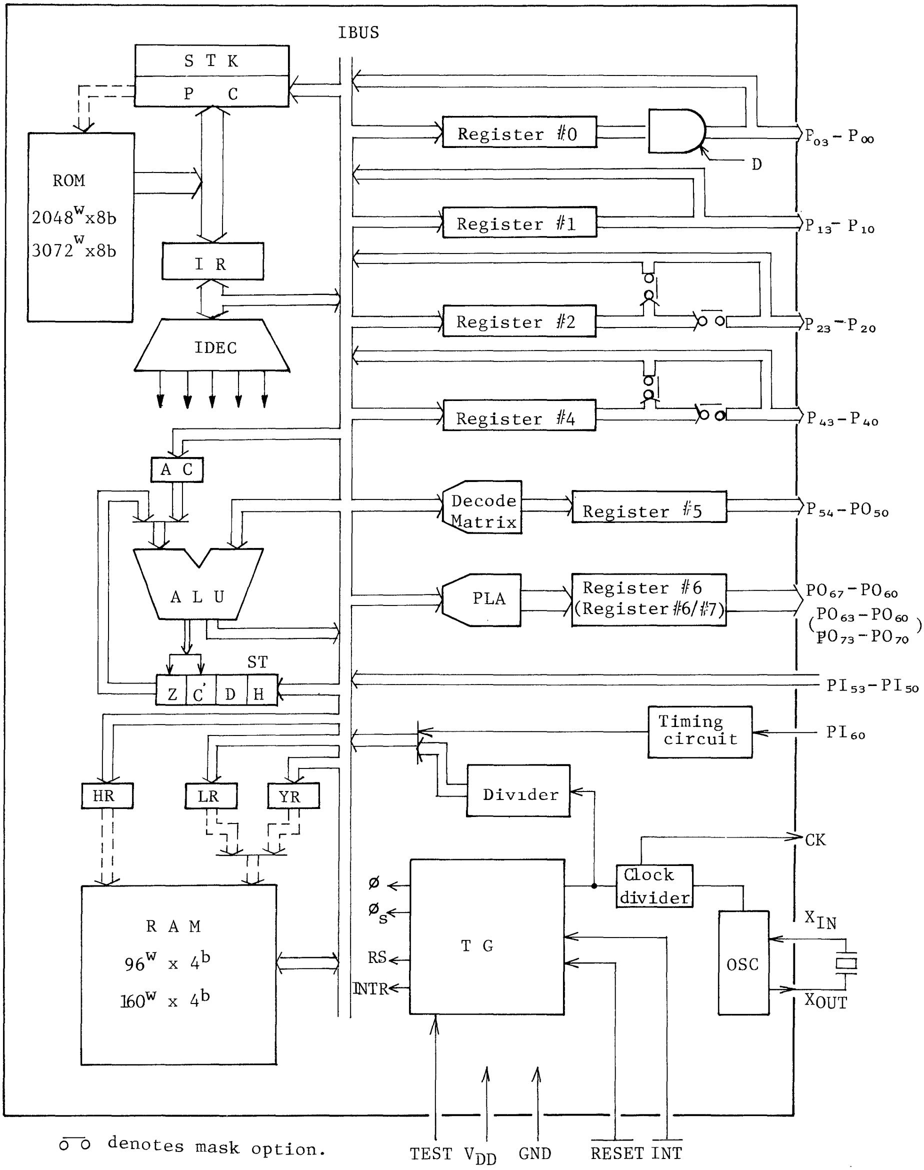


PIN NAMES & PIN DESCRIPTION

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 interinternal 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 connecting to GND
V_{DD}		Power supply
GND		GND

BLOCK DIAGRAM



o͞ o denotes mask option.

FUNCTIONAL DESCRIPTION

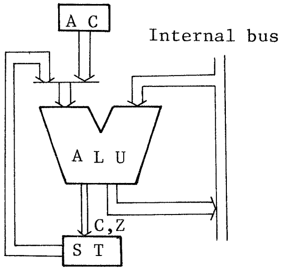
[SYSTEM CONFIGURATION]

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.

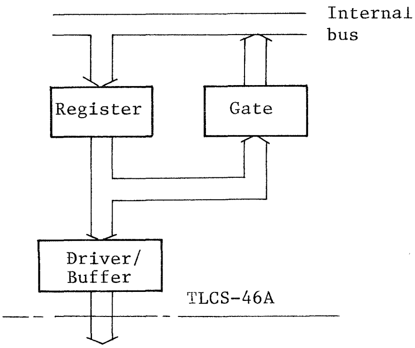
(5) Port (P), Port Register (Register)

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

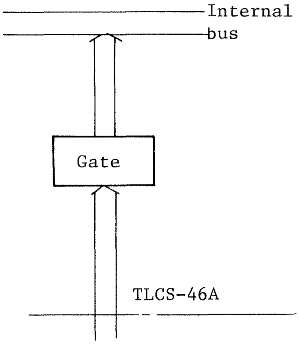
When a port is used as input, it is processed as non-latch input, and therefore, it is necessary to hold external input data till completion of read-in.

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 output 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 (P_{43},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.

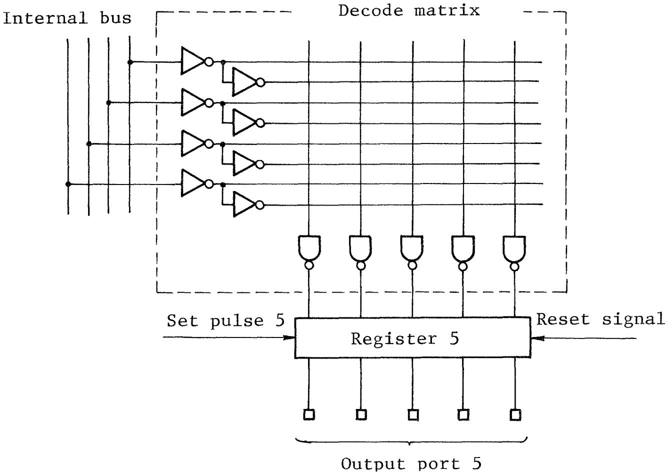
④ Output Port 5 (PO5)

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)

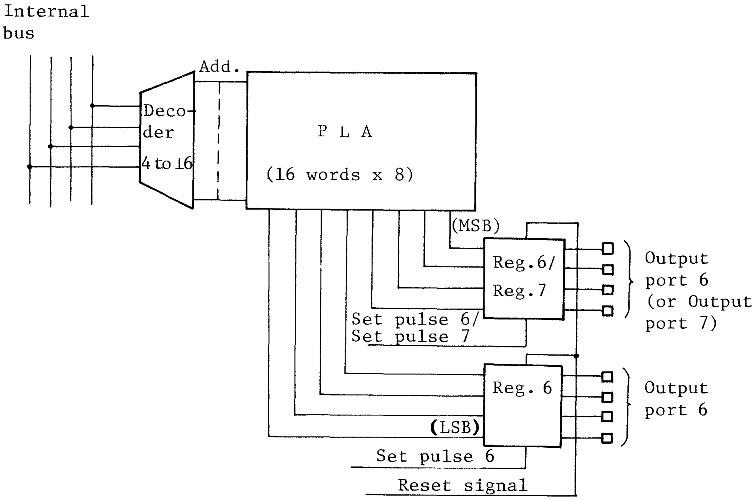
Output Port 6 is a 8-bit output port for segment data output in dynamic display.

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)

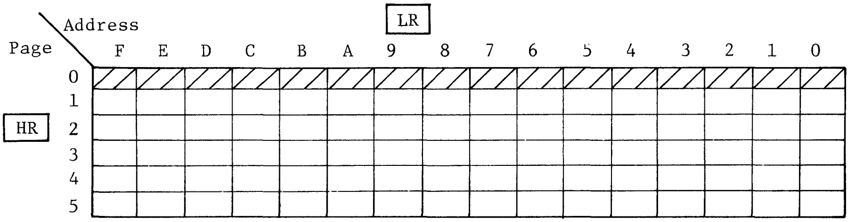
TLCS-46A Family has the following internal data memories in order to store user's process data.

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

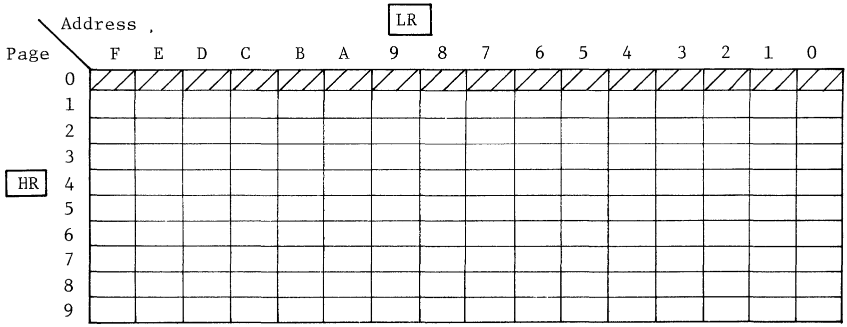
The data memory consists of the static memory cells.

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)

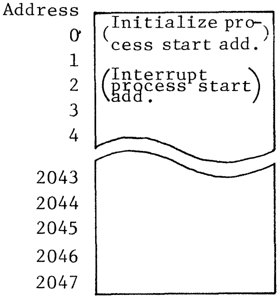
TLCS-46A Family has the following internal program memory in order to store user's process programs.

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

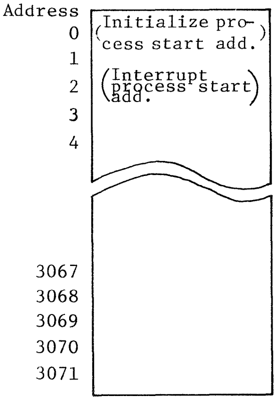
The program memory addressing is executed by the program counter (PC). Inherent meaning is given to Addresses 0 and 2 by the hardware, normal user process programs are place in Address 4 and subsequent addresses.

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.

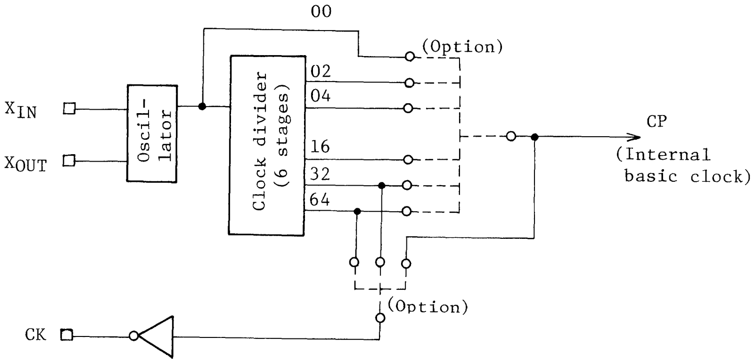
(12) Clock Generator (OSC)

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 optinum 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.

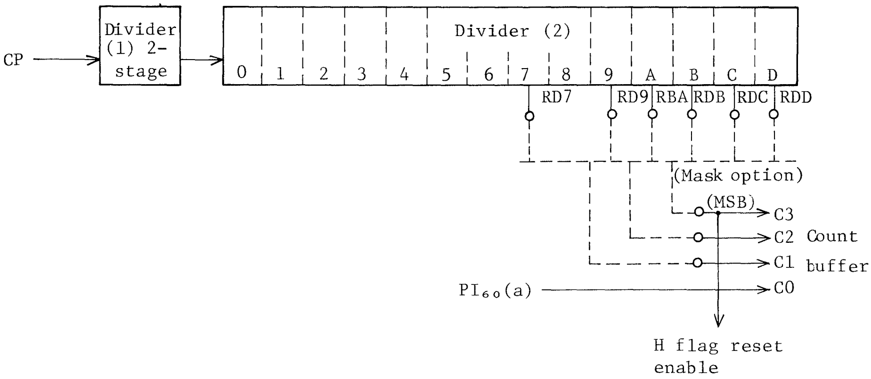
Further, a divide ratio for obtaining internal basic clock and external timing output frequency specified by mask options (refer to Item for Divider) are determined by an oscillator (oscillation frequency) to be used and internal basic clock frequency.



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 ce-ramic /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.

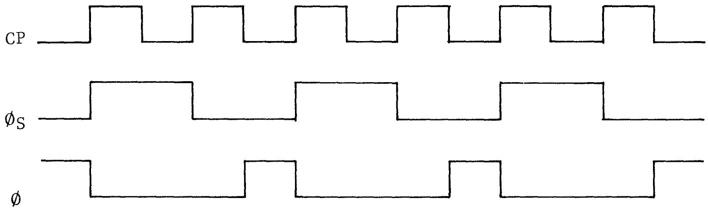
In addition, only when a 100K Xtal is used as an oscillator, the divider (2) is reset when the count value of the divider (2) reaches "12500" and the count value is reset to zero and then the count is resumed.

(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.

Assembler mnemonic	Instruction name			
	Machine code			Execution cycle
	Function	Z flag after execution	C flag after execution	EYR after execution
	Operation description			

Instruction Description Format

(1) Data Transfer Instruction

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

LAR R	Load Accumulator from Regsiter			
	7 0 0 0 0 1 0 R $0 \leq R \leq 7$			1
	(AC) ← (R)	-	-	0
	The content of register or input port designated by R field is loaded in accumulator.			

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 $0 \leq I \leq 15$			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) $\veebar$ (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) $\veebar$ (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 $0 \leq G \leq 3$			1
	$\text{Null} \leftarrow (ST) \wedge 2^G$	*	-	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 $0 \leq G \leq 3$			1
	$(AC) \leftarrow (AC) \vee 2^G$	-	-	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 $0 \leq G \leq 3$			1
	$(ST) \leftarrow (ST) \vee 2^G$	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 $0 \leq G \leq 3$			1
	$(AC) \leftarrow (AC) \wedge \overline{2^G}$	-	-	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 ≤ G ≤ 3			1
	(ST) ← (ST) $\wedge$ $2^{\overline{G}}$	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 ≤ G ≤ 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 ≤ G ≤ 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			
	7 015 8 1 ┆ 1 ┆ 1 ┆ 0 ┆ AP$_L$ ┆ AP$_H$ ┆ AP$_M$ $0 \le AP \le 4095$			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			
	7 015 8 1 ┆ 1 ┆ 0 ┆ 0 ┆ AP$_L$ ┆ AP$_H$ ┆ AP$_M$ $0 \le AP \le 4095$			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			
	7 015 8 1 ┆ 1 ┆ 0 ┆ 1 ┆ AP$_L$ ┆ AP$_H$ ┆ AP$_M$ $0 \le AP \le 4095$			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 thc 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;

$$\mathrm{EYR} = 0\ ,\ (\mathrm{HR} \circ \mathrm{LR})$$
$$\mathrm{EYR} = 1\ ,\ (0 \circ \mathrm{YR}).$$

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	$2\underline{i}$	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	A$\underline{i}$	101000ii	if (AC)<ii>=0 then Z←1 eles Z←0	Z	
	TBS	B$\underline{i}$	101100ii	if (ST)<ii>=0 then Z←1 eles Z←0	Z	Asteris (*) denotes that, when Z flag and C flag are designated, the fags are updated.
	SBA	A$\underline{i}$	101001ii	(AC)<ii>←1		
	SBS	B$\underline{i}$	101101ii	(ST)<ii>←1	**	
	CBA	A$\underline{i}$	101010ii	(AC)<ii>←0		
	CBS	B$\underline{i}$	101110ii	(ST)<ii>←0	**	
	IBA	A$\underline{i}$	101011ii	(AC)<ii>←$\overline{\text{(AC)<ii>}}$		
	IBS	B$\underline{i}$	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

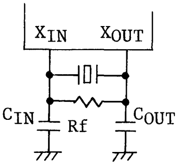
[OPERATION DESCRIPTION]

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.

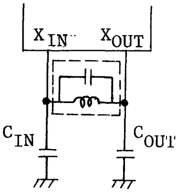
For the generation and supply of basic clock, there are following methods:

(1) Connecting crystal oscillator/ceramic oscillator



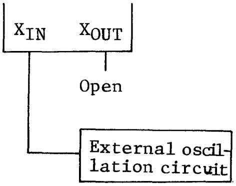
In case the crystal oscillator/ceramic oscillator is connected as shown in the left figure, it is possible to obtain the oscillation frequency characteristic to crystal oscillator/ceramic oscillator.

(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



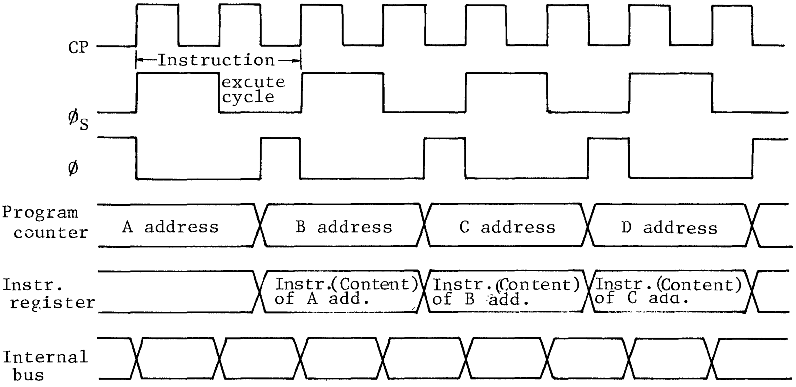
If the external oscillator is connected to X_{IN} terminal as shown in the left figure, it is possible to supply the clock from the external.

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

2. Instruction Execute Cycle

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

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

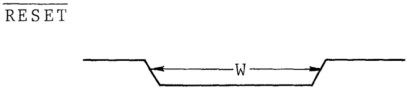

CP
Instruction execute cycle
∅S
∅
Program counter
A address
B address
C address
D address
Instr. register
Instr.(Content) of A add.
Instr.(Content) of B add.
Instr.(Content) of C add.
Internal bus

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.

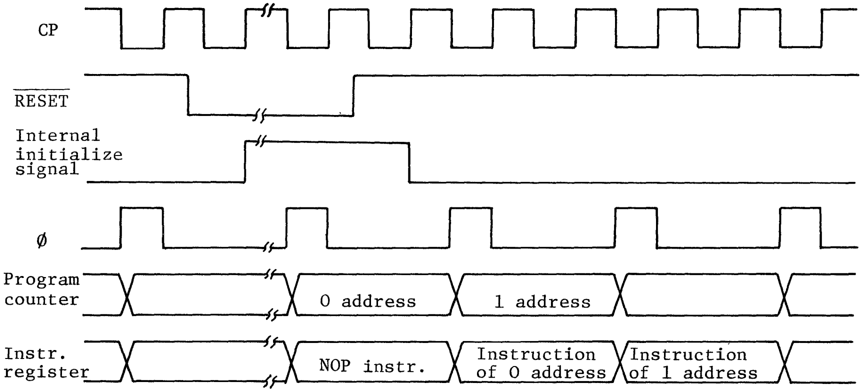
However, 1 cycle corresponds to 2 period of internal basic clock (CP).

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)

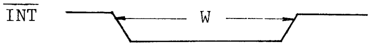
② Stack is empty. (FSTK = 0)

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.

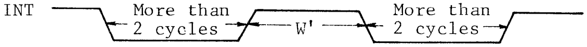
① Internal latch set timing



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:

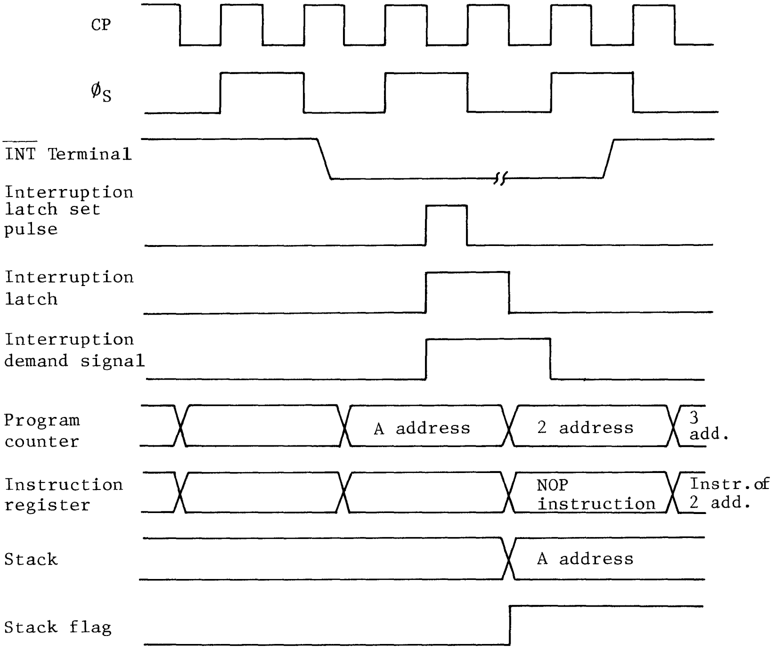
1 The content of program counter is stored in a save register (stack).

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.

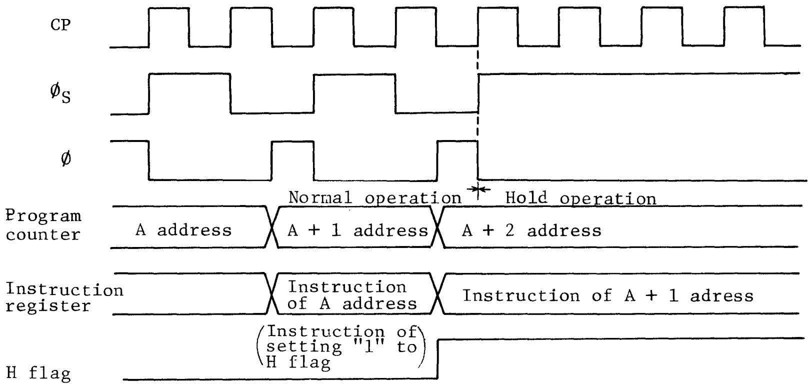
When each instruction of SBS, LBS and LSM is executed, and "1" is set to the H flag, the internal timing signals of ϕ_S and ϕ stop after the completion of the execution of the instruction, whereby the hold operation begins. At this time, the program counter, instruction register and other internal registers 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.

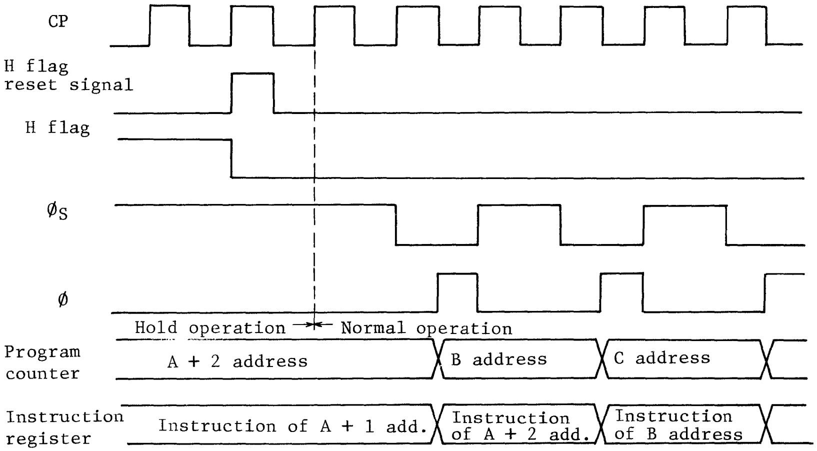
Restart from the hold operation is made by resetting H flag to "0" in hardware. For the signal for resetting H flag, the output of divider (the MSB input (C3) of counter buffer) is used, and the resetting is made by detecting the trailing edge of the bit input (C3).

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.)

Data Memory Address

7 4	3 0
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.

Data Memory Address

7 4	3 0
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

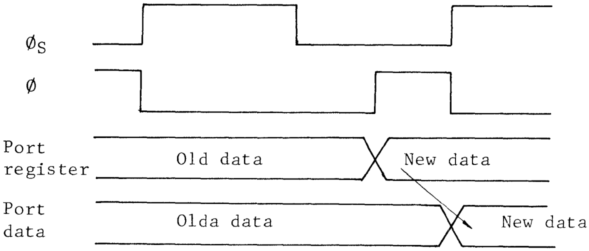
As the input ports of TLCS-46A are all non-latch inputs, the input data is required to be stable until the completion of reading.

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

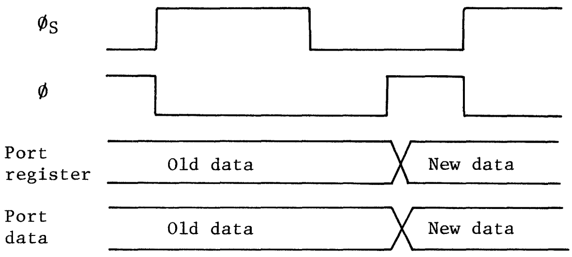
Port Output

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.



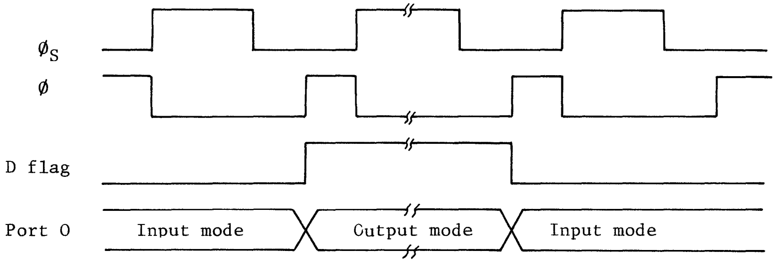
Output Timing (Port5, Port6)



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.



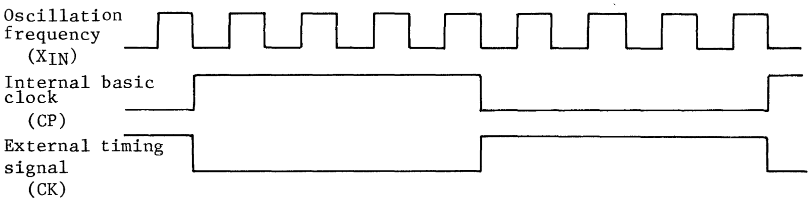
Input/Output Switching Timing of Port 0

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.
P2	Prescribing the function of port 2.
P4	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.

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	CO	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		PO6	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
	ЄK	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.

o CP00 - Oscillation signal is made to clock.

o CP02 - Signal of dividing oscillation frequency into 2.

o CP04 - Signal of dividing oscillation frequency into 4.

o CP32 - Signal of dividing oscillation frequency into 32.

o CP64 - Signal of dividing oscillation frequency into 64.

③ CK: Prescribing the output frequency for external timing

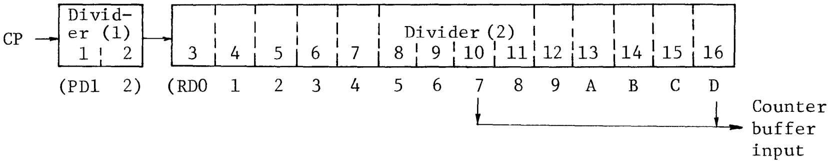
o CK32 - Signal of dividing oscillation frequency into 32.

o CK64 - Signal of dividing oscillation frequency into 64.

o 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.

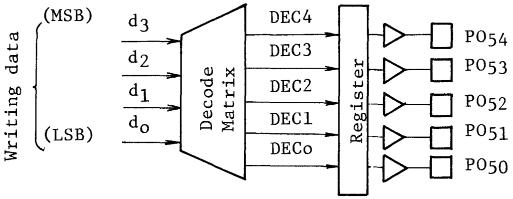
(3) DECODER - PLA Options

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.)

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

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

	$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	do	
Designation: 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: $DEC4 = d3 \cdot d2$

	$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	d0	
Designation: 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

Designation: DEC1 =

$\overline{d3}$	d3	$\overline{d2}$	d2	$\overline{d1}$	d1	$\overline{d0}$	d0
0	0	0	0	0	1	0	0

= X' 04'

* Refer to the examples of data designation.

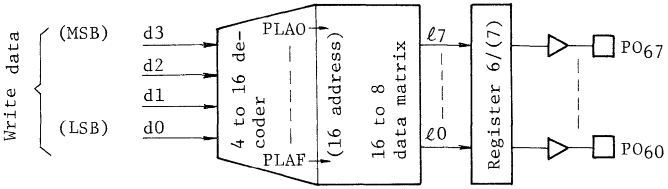
② P06: Prescribing thc function of port 6.

o $P0_{60}$ - Port 6 is used as the 8-bit PLA data output port.

o $P0_{61}$ - 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

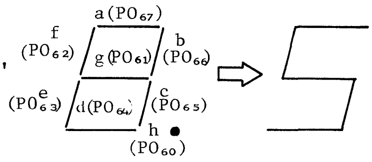
The designation of PLA data is made to addresses by expressing the 8-bit data, which are MSB in PO67 terminal output and LSB in PO60 terminal output, with 2-figure numeric characters of hexadecimal digit.

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,

PLA 5 =

PO_{67}	PO_{66}	PO_{65}	PO_{64}	PO_{63}	PO_{62}	PO_{61}	PO_{60}
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,

PLA 3 =

PO_{67}	PO_{66}	PO_{65}	PO_{64}	PO_{63}	PO_{62}	PO_{61}	PO_{60}
0	0	1	1	0	0	1	1
Port 7				Port 6			

= X' 33'

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 Ceneral 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'         ..... Comment   Output of six characters starting  from the
 (CR) (LF)                         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  ;           ..... Output of "N8" indicating that the date pattern is 8
 (CR) (LF)               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     |      XXXPX
                                           |      |   |
 XXX PX    ;             2nd address data  |      |   └── Check sum
      |                  and check sum     }      |
      |                                    |      |       Number of data bits having
      |                                    |      |       "1" is output.
 XXX PX    ;             8th address data  |      └──Data
 (CR) (LF)               and check sum     )         Two-characters of eight bit
                                                     data are output, in two frames
 RXXXX     ;       ..... 9th program address         of hexadecimal ASCII code.

 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".
     |
```

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 Temepratu re	-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

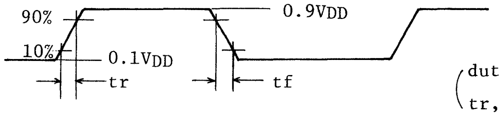
DC CHARACTERISTICS (Ta=-30°C to +85°C, V_{DD}=3V to 6V)

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.



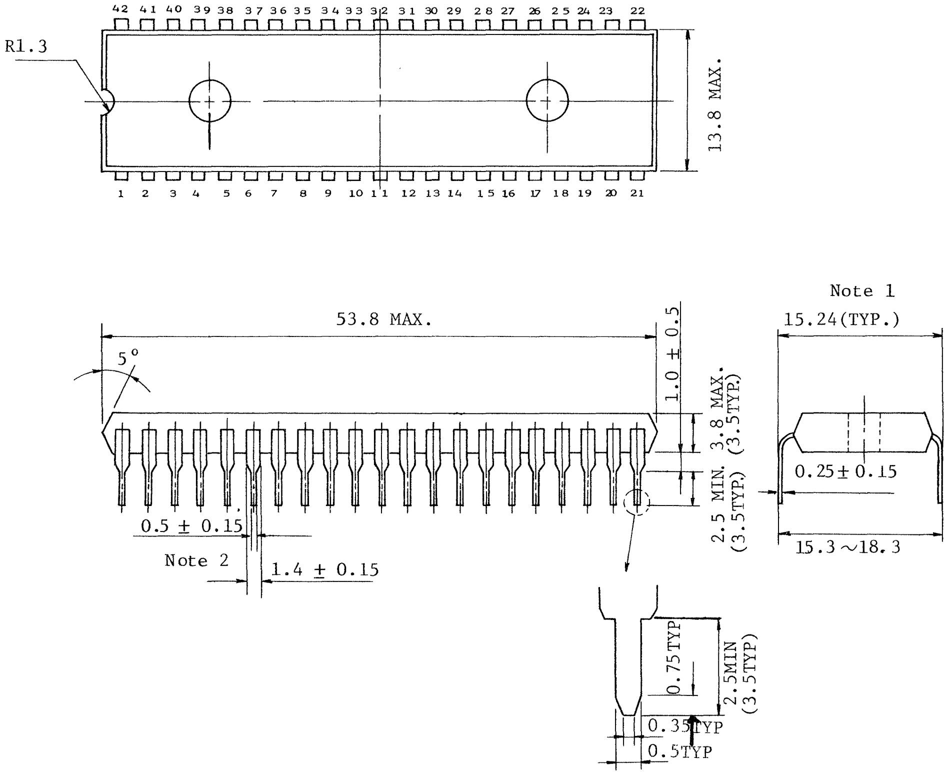
(duty = 50%
tr,tf < 50ns)

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.