Date :- 07/01/2020)

Word size

Ability of microprocessor to process the no. of bits in one instruction or one clock cycle.

Example

motorola
↓
8bits microprocessor → 8085, 68000

16 bits 8086 Μp

1NB = 2^{90}

1HB =

$(FEFE)_H$ = 2 Byte

$(FE)_H$ = 1 Byte
↓
FEH

#Difference b/w Microprocessor & Micro controller

Μp unit	Micro crontroller
1.) Based on John Von Neumam architecture	Based on Harvard architecture
2.) Micro = CPU + RAM + I/o device not integrated on single chip	Micro= Micro+ RAM + I/o on single chip integrated on single chip.
Micro = control + ALU + registers on single chip	All the embedded system are application of microcontooller.

10/01/2020

8085 ← works with single accumulator organisation

Some features of 8085

1) 8085 can process 8-bit data at a time so it is known as 8-bit microprocessor.

2) It can accept process and provide 8-bit data simultaneously.

3) It operates on +3V power supply → two pins Vcc and GND.

4.) It has on chip clock generator. The internal clock generator require LC, RC or crystal circuit for tuning the circuit.

- The internal clock generator divides this oscillator frequency by 2.
- So 8085 μp can operate on clock frequency 3MHz.

Version of 8085 → 8085-A, 5MHz.

5.) The 8085 has 16 address line so it can access 64K bytes of memory.

6.) The 8085 μp provides 8-bit I/O addresses to access 2^8 I/O ports (or 256 I/O ports)

7. In 8085 μp the Lower 8-bit address lines are multiplexed with data lines. (Reduce the no. of available lines)

A_7 — A_0 (lower 8-bit address)

Lower
multiplexed
with

D_7 — D_0 (Data Lines)

~~Why the~~

To reduce the no. of pin. ~~To reduce the cost of the system~~. They are seperated or demultiplexed by using latch.

8) The 8085 μp uses 5 addressing modes
Immediate, register, direct, indirect, implied

9) 8085 μp is using 74 instruction.

10) It consist 8 general purpose register each of 8 bit designated as ~~A, B, C, D, E, F, G, H,~~

A, B, C, D, E, H, L, & FLAG

→ 8085 μp working on single accumulator organisation.

Accumulator	FLAG
B (8)	C (8)
D (8)	E (8)
H (8)	L (8)

FLAG

It is a special register which specify status & condition of the result.

8085 μp consist of 5 FLAG

Sign, S
Carry, Cy
Auxilliary Carry, Ac
Zero Flag, Z
Parity Flag, P

The 8085 works on 5 hardware interrupts

→ Trap ← Non maskable interrupt

→ RST 7·5

→ RST 6·5

→ RST 5·5

→ INTR (interrupt request)

(Non maskable interrupt cannot be stoped.)

14/01/2020

Stack stores PSW (Program Status Word)
Application of stack is in subroutine calls.

Q→ If the 8085 adds 87H and 79H, specify the contents of the accumulator and the status of S, Z and CY flag?

Solⁿ

```
   1 1
     8 7 H
     7 9 H
  ---------
  1) 0 0 H  ← Accumulator.
  ---------
```

So, CY = 1

Z = 1

S = 0

Software interrupts

RST 0 to RST 7

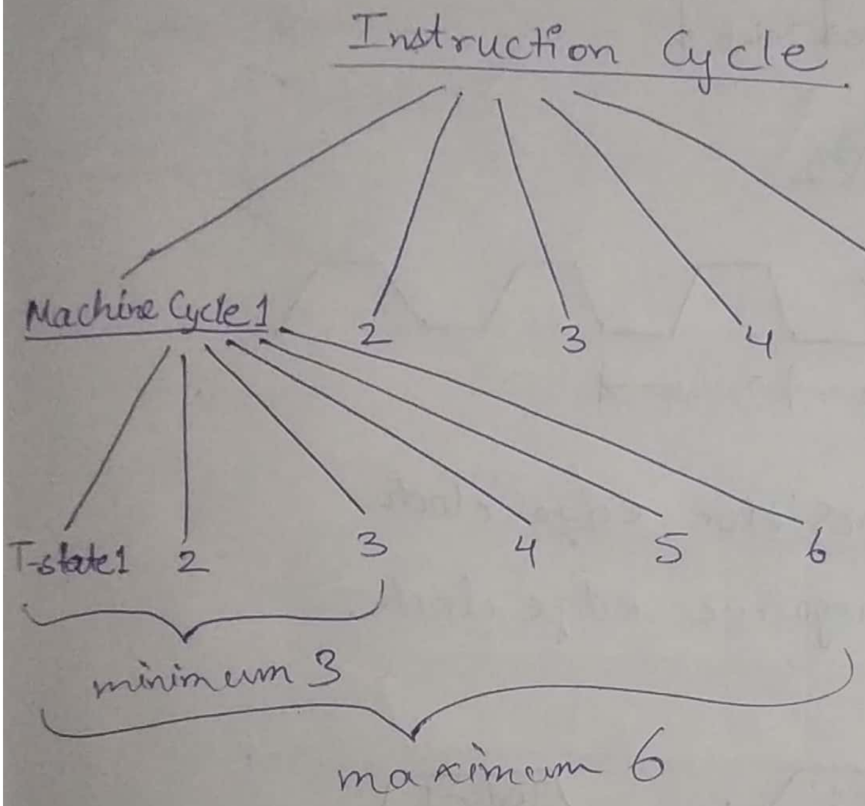
16/01/2020

Address Latch Enable
74 LS 373 -> 8 bit Latch
Ch 11 -> Morris Mano -> Flip Flo

Pin Configuration of 8085 :-

Reset Out , $\overline{\text{Reset In}}$ — H.W

Relation between Instruction cycle, Machine cycle, T-state (clock cycle) :-



Therefore,

$$1 M/C = 3 \text{ to } 6 \text{ T-state}$$

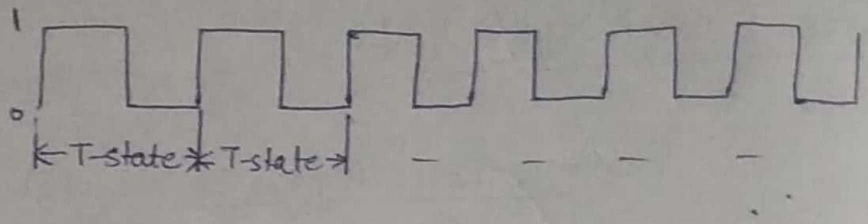
During normal operation μp sequentially fetches, decodes and executes one instruction.

The fetching, decoding and execution of a single instruction constitutes an instruction cycle which consist 1 to 5 read or write operation between μp and memory or I/O devices.

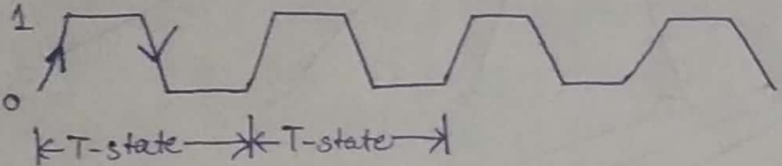
Each memory operation require a particular time period called machine cycle.

Each machine cycle consist 3 to 6 clock periods / cycles referred as T-state.

Clock →



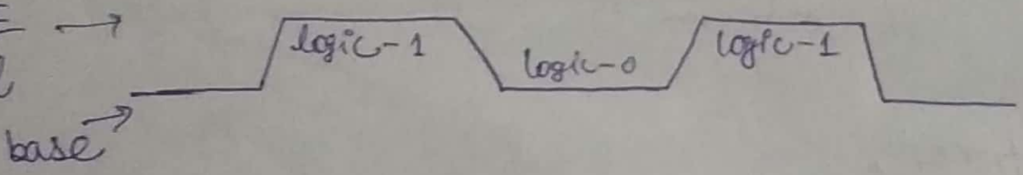
Actual →

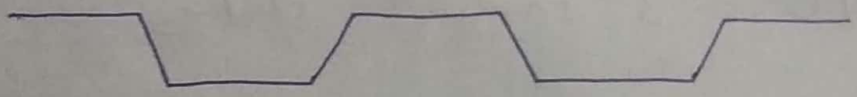


from 0 to 1 → positive edge clock

from 1 to 0 → negative edge clock.

ALE →
signal



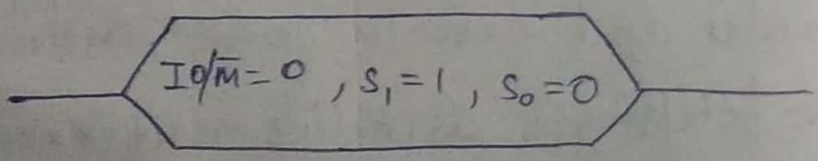


Multiple signal

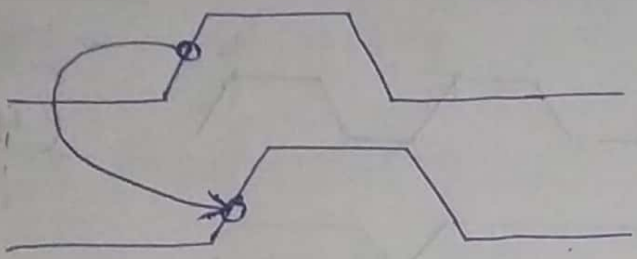
IO/$\overline{M}$, S_1, S_0

Address lines

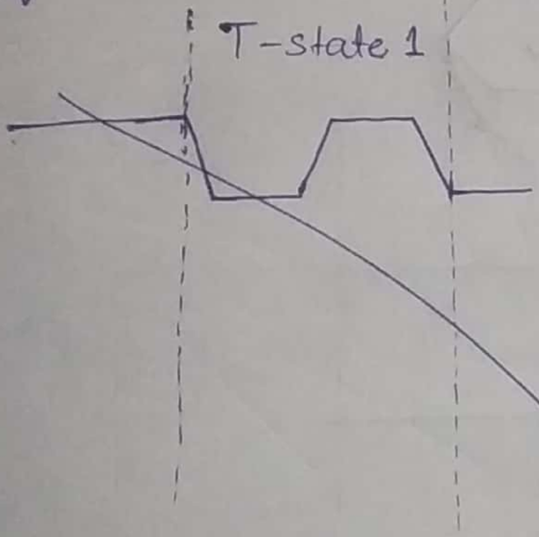
Data lines



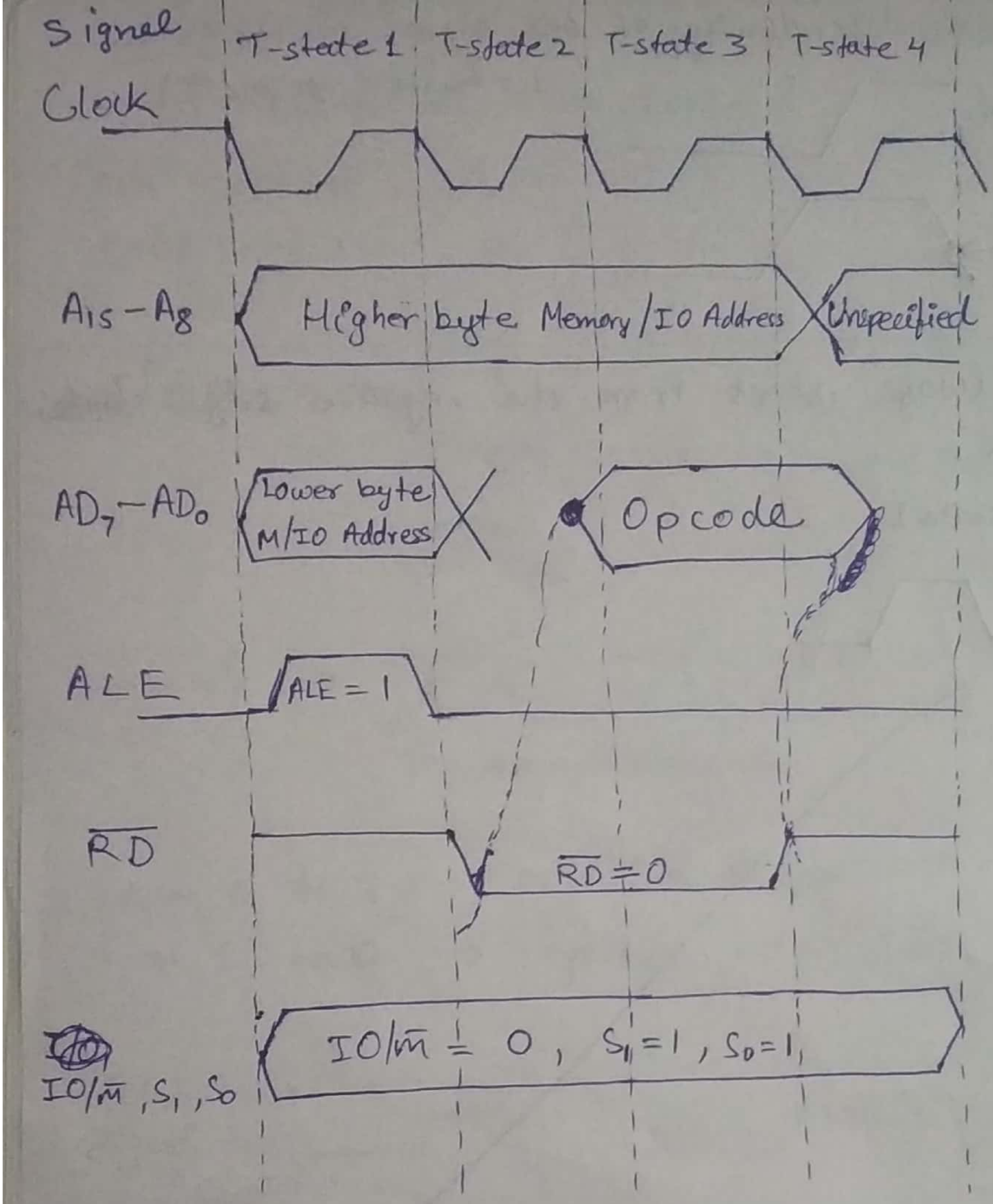
How to show dependency of one signal on another (between signals).



* for 8085 always start from the negative edge ~~clock~~ of the clock.



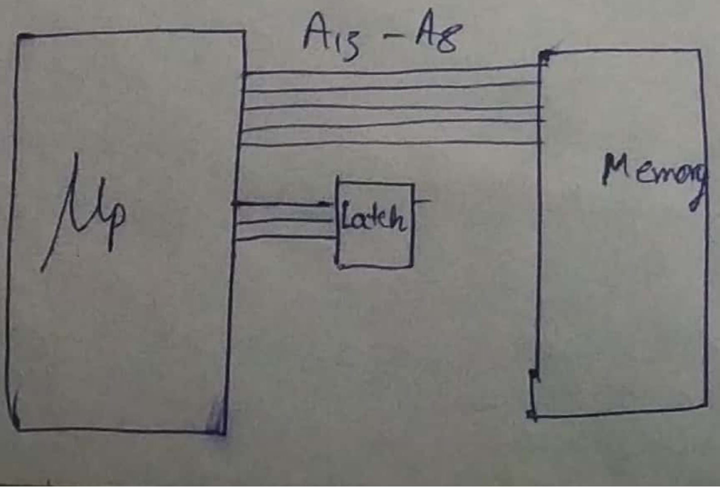
Op-code fetch cycle (4T)



12/01/2020

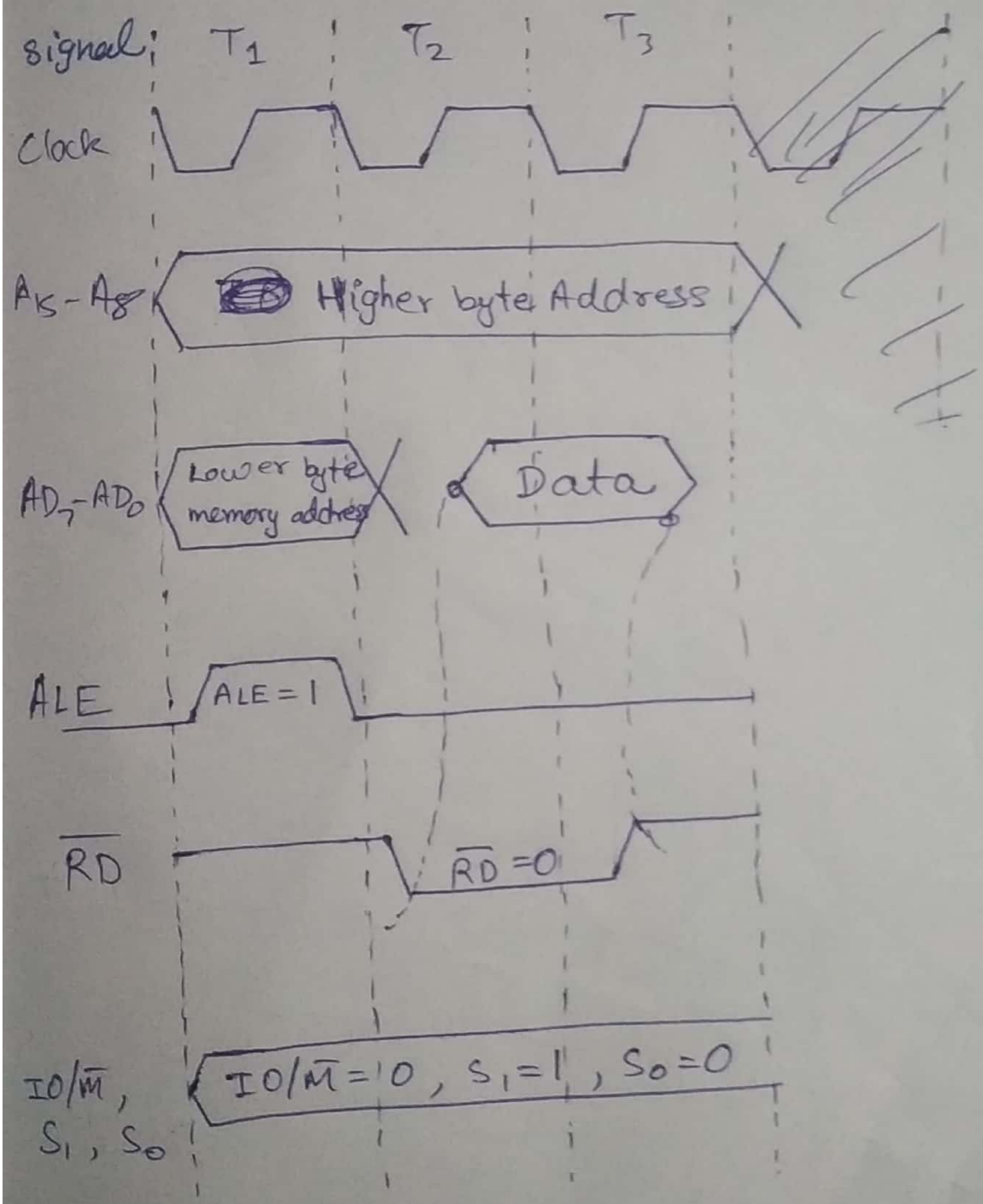
opcode fetch machine cycle

$IO/\overline{M} = 0$, $S_1 = 1$, $S_0 = 1$



Q In case of op code fetch machine cycle, T_4 is used for decoding.

Memory Read machine cycle (3T)



MVI B, FFH

- → 2 byte
- → 2 Memory
  - → Op code (MVI B) ← Op code fetch. (4T)
  - → operand (43H) ← Memory Read. (3T)

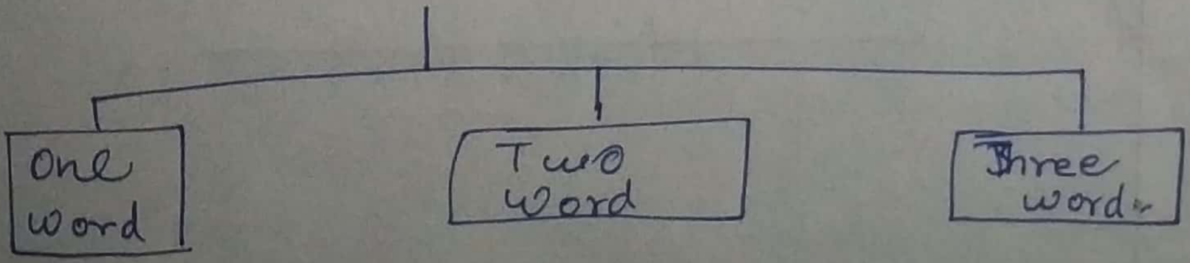
LDA 2000 H.

Instruction Set Classification

1) On the basis of length of an instruction.

2) On the basis of function.

1) On the basis of length



On the basis of function :-

1) Data transfer Instruction set

i) Register to Register :-

MOV R_d, R_s $\qquad$ $R_{DES} \leftarrow R_{SOURCE}$

MOV D, H

ii) Register to Memory.

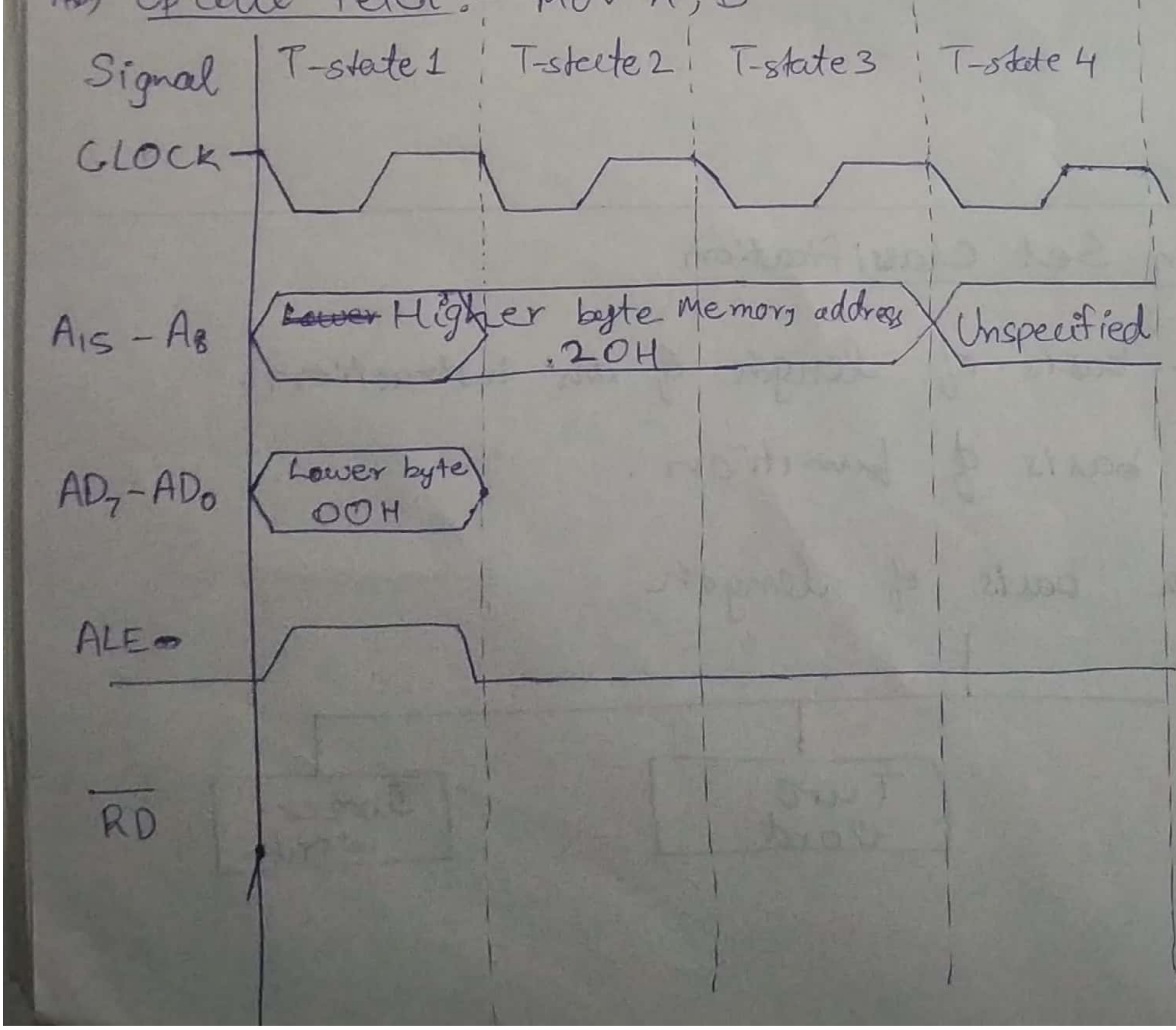
iii) Memory to Register.

(iv) Initialization.

23/01/2020

Q) Draw the timing diag. for MOV A, B from address location 2000.

Ans) Op code fetch :- MOV A, B



M means → M[HL] ← HL Register pair
pointer must be created by HL Registe Pair.

For Immediate data Op-code contains I,

Ex ADI ,ACI , ~~etc~~. LXI , etc.

Add data at location 3000 with 05H

→ MVI A, 05H
LXI H, 3000H
ADD M.

Arithemetic Instructions :-

Addition

1.) Addition without carry :-

ADD R
ADD M ← A ← A + M[HL]
ADI 8bits data
↑
Immediate data.

Flags are not affected when we transfer data

Flags are affected in arithematic instruction

2.) With Carry :-

ADC R
ADC M
ACI 8 bits data
↑
Carry

Substraction

1.) without Borrow :-

SUB R

SUB M

SUI 8 bits data

2.) with borrow :-

SBB R

SBB M

SBI 8 bits data.

Increment

INR R

R ← R + 1

INX Rp

Register Pair

Ex 1) MVI B, 06H
INR B

2) LXI H, 2000 H
INX H.

Decrement :-

DCR R

DCX Rp — for 16-bit

Logical Instruction :-

ANA R

ANA M

ANI 8 bits data

ORA R

ORA M

ORI 8 bits data

XRA R

XRA M

XRI 8 bits data

Complement.

CMA

CM

~~LHLD 3000~~

Q Load HL register with the contain of memory location whose address specified with the instruction

LHLD 3000 ← special data transfer instruction

Consecutive memory locations can be stored. i.e.,

L ← M[3000]

H ← M[3001]

Q Store the HL data to the consecutive memory location whose add. specified with the instruction

→ SHLD 4000 ← special data transfer instruction

↓ i.e.,

M[4000] ← L

M[4001] ← H

~~Move m0~~

Load Register pair with immediate data (16-bit)

→ P.T.O

~~LXI D,~~ RIM & SIM

LXI D, 1010 H

LXI B, 1F1F H

D E

B C

```
MOV A, E
ADD C
MOV L, A
MOV A, D
ADC B
MOV H, A
SHLD 3000
HLT
```

← $\boxed{10+1F}$ A

To add the contents of DE register pair with contents of BC register pair and Store the result at consecutive memory location 3000 & 3001.

24/01/2020 Branch Operation :-

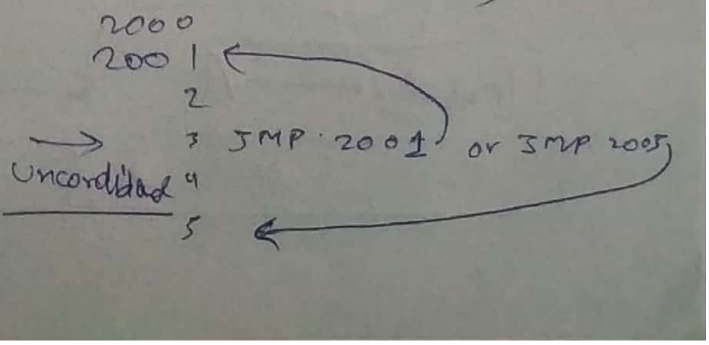
Branching :- Control switches from main program to sub program or special part. (subroutine calls)

Two types

- Unconditional branching
- Conditional branching (on the basis of status flag, control of execution switches from one instruction to another)

Instruction (unconditional)

* JMP <addr> Ex



CALL <addr>
RET
← Unconditional

R → RET.

Conditional branching op. :-

	J → C	J → R
JC	CC	RC
JNC	CNC	RNC
JP	CP	RP
JM	CM	RM
JPE	CPE	RPE
JPO	CPO	RPO
JZ	CZ	RZ
JNZ	CNZ	RNZ.

Machine Control Instruction :-

1.) HLT → stop the machine completely

2.) RST 5

3.) IN Port# → waiting for input of user

4.) OUT Port#

0 → no. of port

~~Port~~ A ← $\boxed{\text{Port \#}}$ ← IN Port #

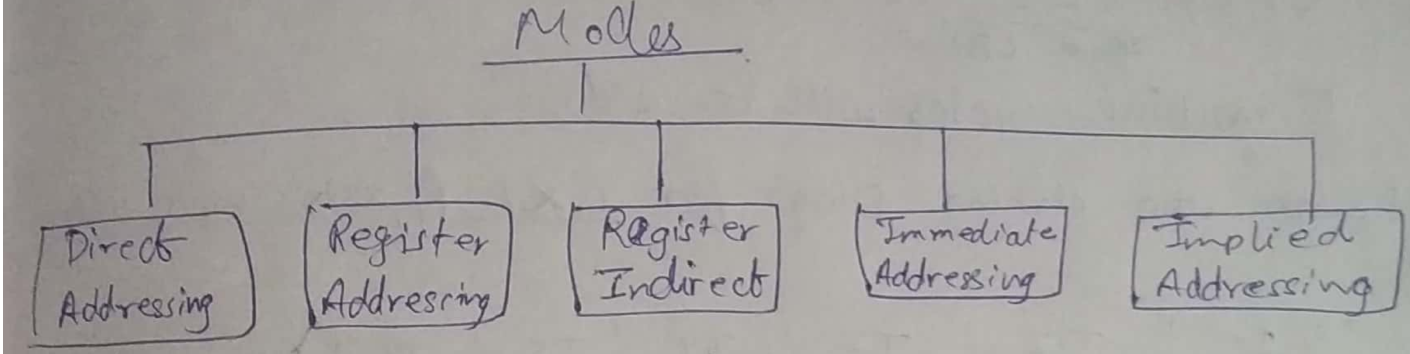
$\boxed{\text{Port\#}}$ ← A ← OUT Port #

Addressing Modes :-

Instruction format :-

Mode	Opcode	Data or Address

5 Addressing modes in 8085 :-



Direct Addressing :-

Ex: LDA D500H

Register (single byte)

Ex:- i) MOV A, B

ii) INR B

iii) ADD C

Immediate ~~value~~

Ex. LXI D,

30/01/2020

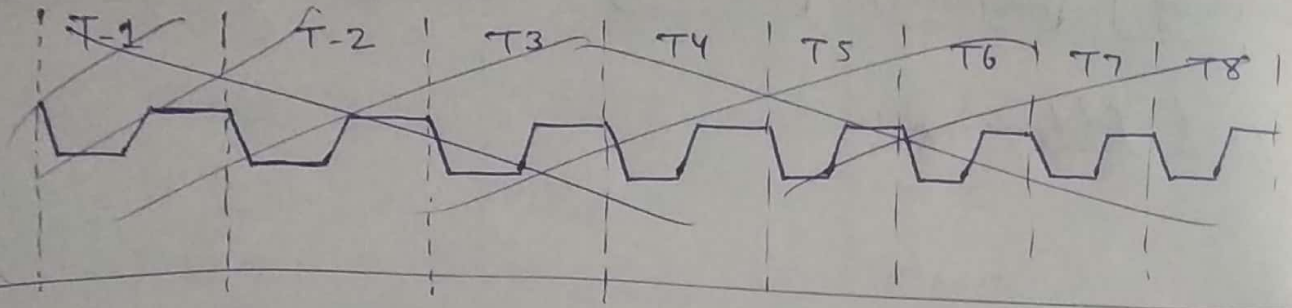
Q→ How many machine cycle will run for LHLD instruction?

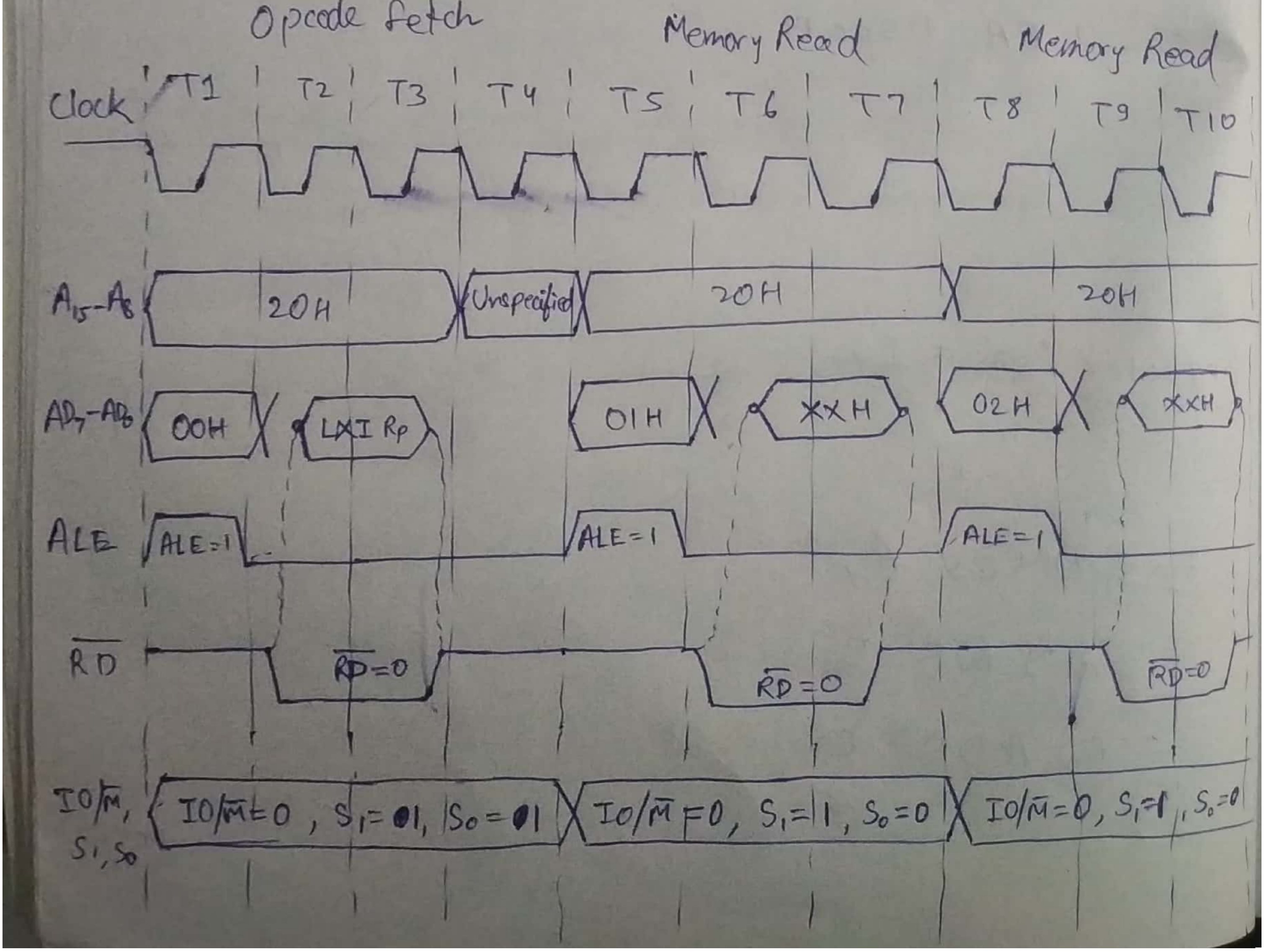
Solⁿ LHLD $\underbrace{\_\ \_}_{HB\,add}\ \underbrace{\_\ \_}_{LB\,add}$ ← 3byte

5 machine cycles will be used.

Q→ Design the timing Diag. for LXI Rp, xxxx 3 byte address

Solⁿ





LXI Rp, xxxx → 3byte, 3Memory

Let 2000 LXI Rp [Opcode fetch cycle 4T]
2001 XXH [Operands, Memory Read 3T]
2002 XX H [operands, Memory Read 3T],

Execution inside μp

- So No extra cycle Required.

31/01/2020

1's & 2's complement using Assembly language

1's complement

```
MVI B, F0h
MOV A, B
CMA
STA 3050
HLT
```

2's comp

```
MVI B, F0h
MOV A, B
CMA
ADI 01H / INR A
STA 3050
HLT
```