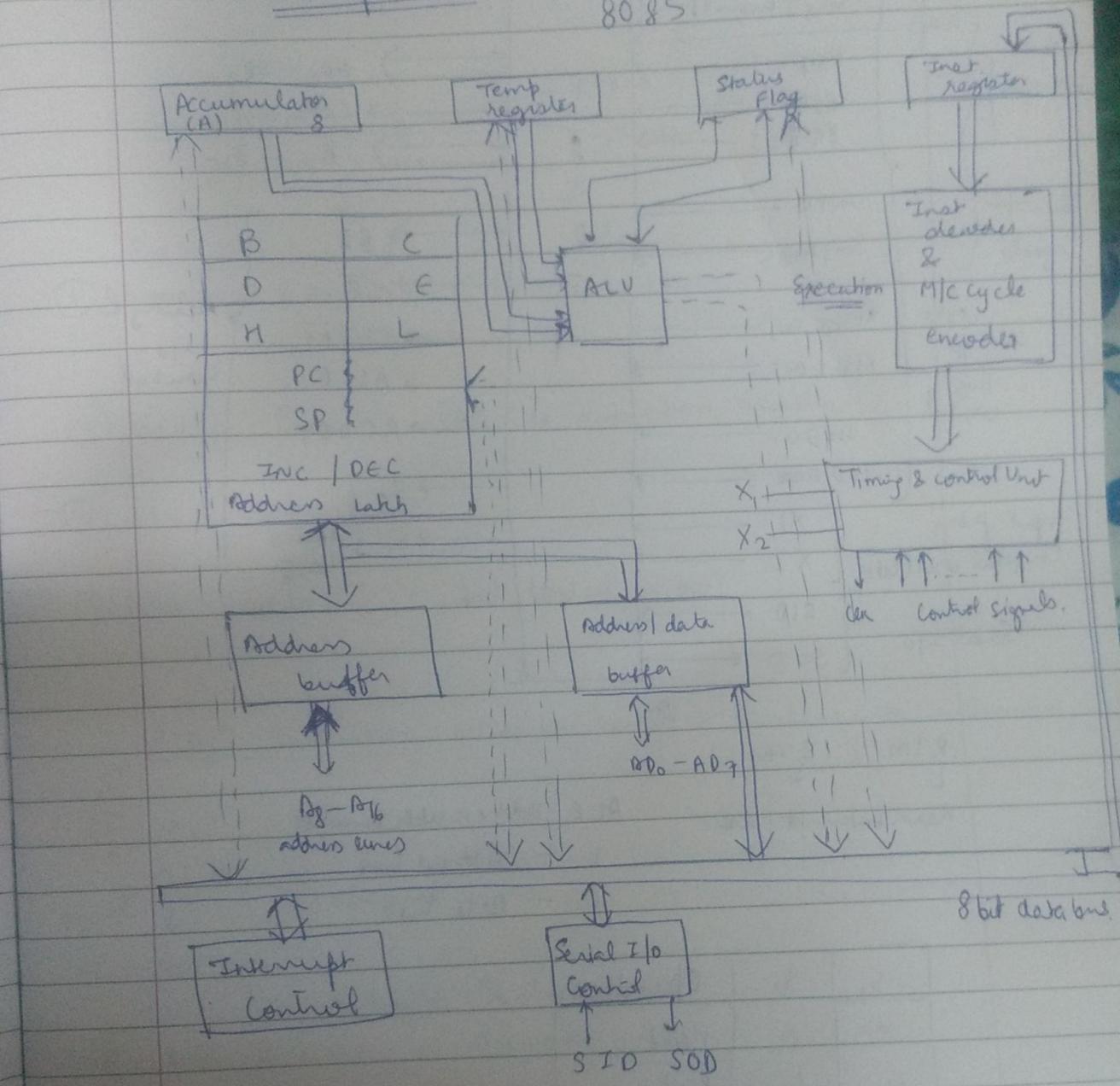
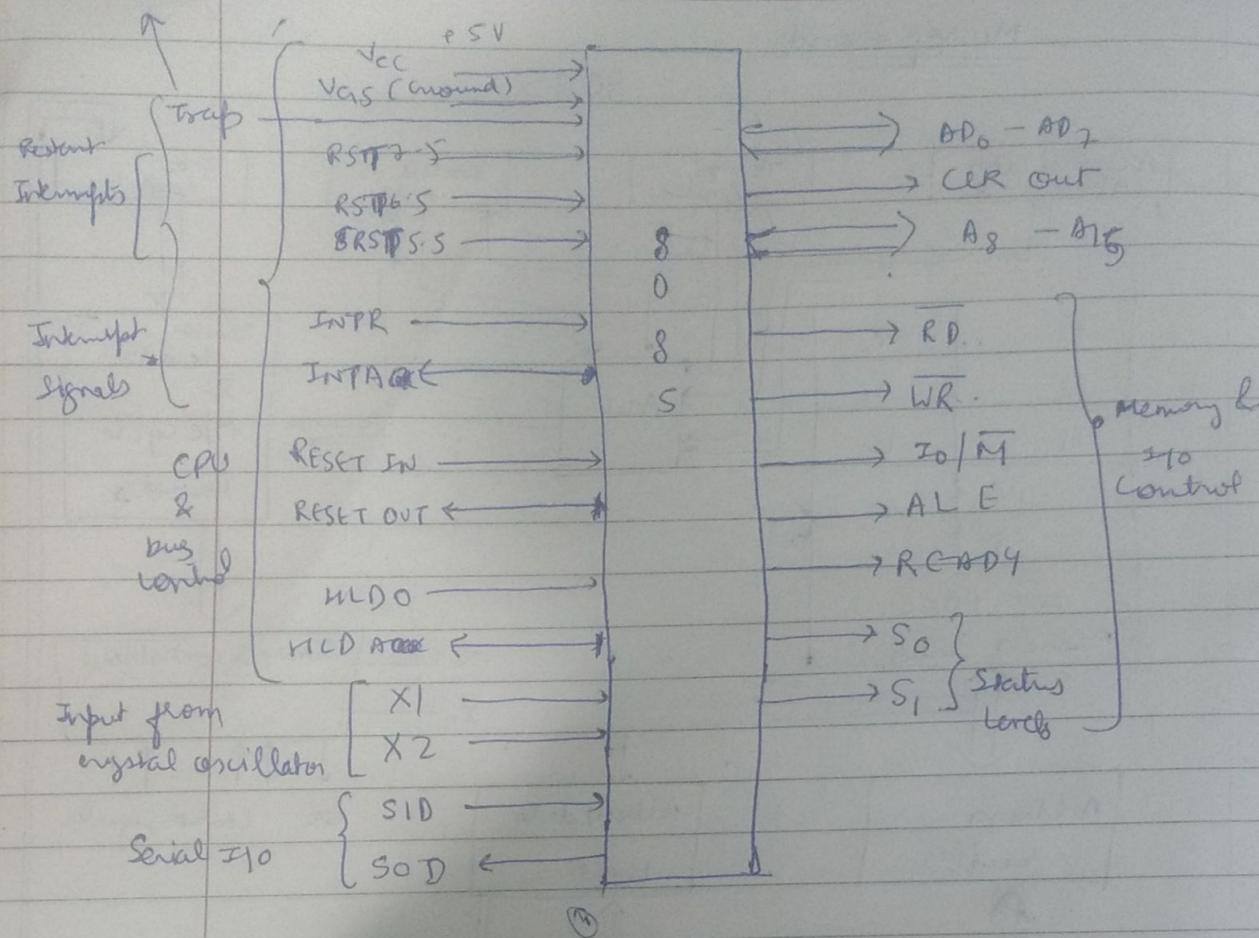


Microprocessor

8085

Pin Diagram :→ 8085,

Highest priority, non maskable interrupts



RIM S₁ IN
↓ ↓
Read Interrupt Mask

ALE - Address latch Enable.

1 - Address bus

0 - Data line

S ₁	S ₀	Operation
0	0	Halt
0	1	Read
1	0	Write
1	1	Fetch

Program status word → SF flag bit + 3 undefined bits

↓
8 bit PSW + 8 bit AC

↓ = SP (16 bit Register).

PSW

Clas

One bi
MOVE

① MVI

② MV

③ LDA

Q by

④ MVI

06, C

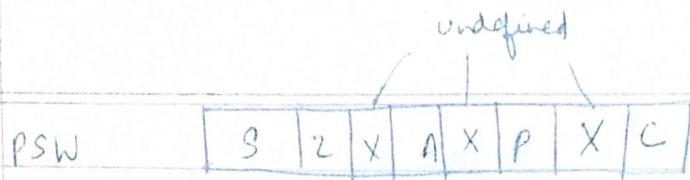
opcode ↓

MVI

⑤ LD

21

Clas



Classification of Inst *operand*
 opcode

One byte 2 byte 3 byte

① MOVE MOV BA (78H). 01111111
 01000111

② MVI B (AC is destination register).

③ MV I B data

④ LDA addr (Addr loaded in AC)

2 byte Instruction (Memory location & along with instruction)

① MVI B, 05.

05 → operand
 ↑
 opcode

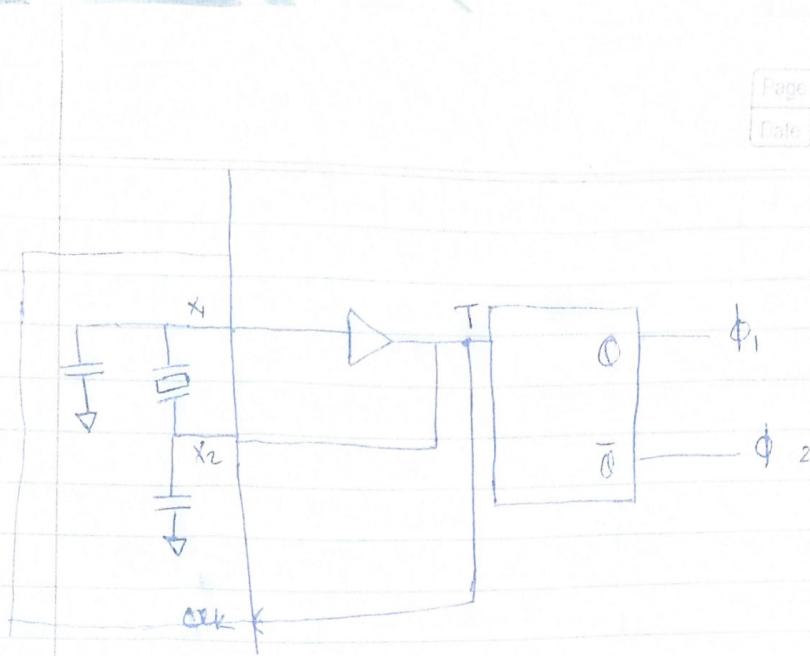
MVI B

② 3 byte Instruction
 LD A LX1 N 2400H

21
 ↑
 [21, 00] 24

clock → Single Phase

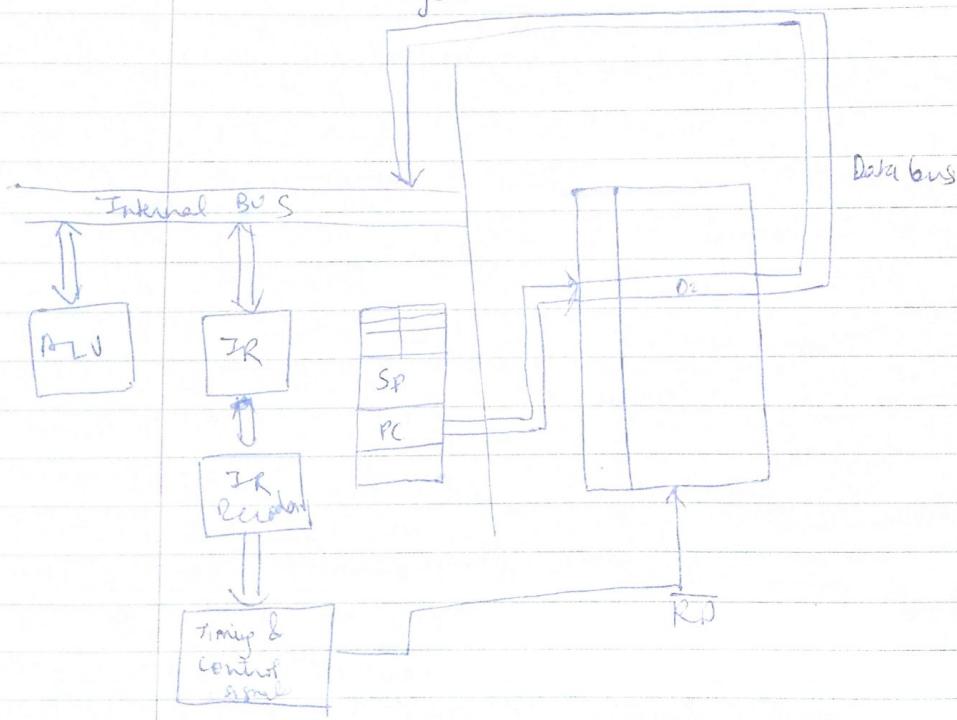
→ Two Phase



Instruction cycle: →

$$IC = FC + EC$$

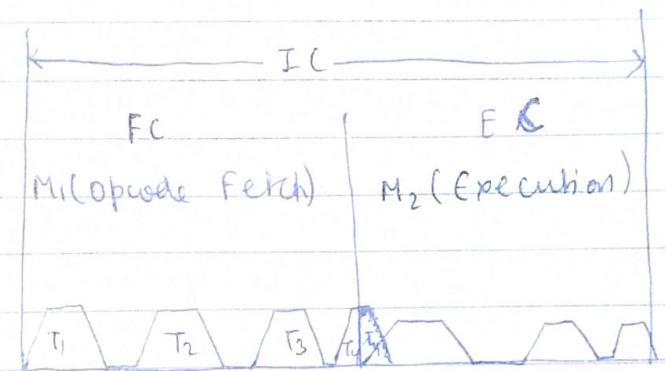
↓ fixed. ↓ very
 max vary
 3 clk
 cycle





$$IC = FC + EC$$

cycle / states
Machine cycle



① 1 byte instruction one cycle to fetch & execute

② 2 or 3 byte instructions take 2 or more than 2 machine cycles

Timing Diagram for opcode fetch cycle: →

μP fetches opcode from main memory

1 byte Instruction

CR

SD/M

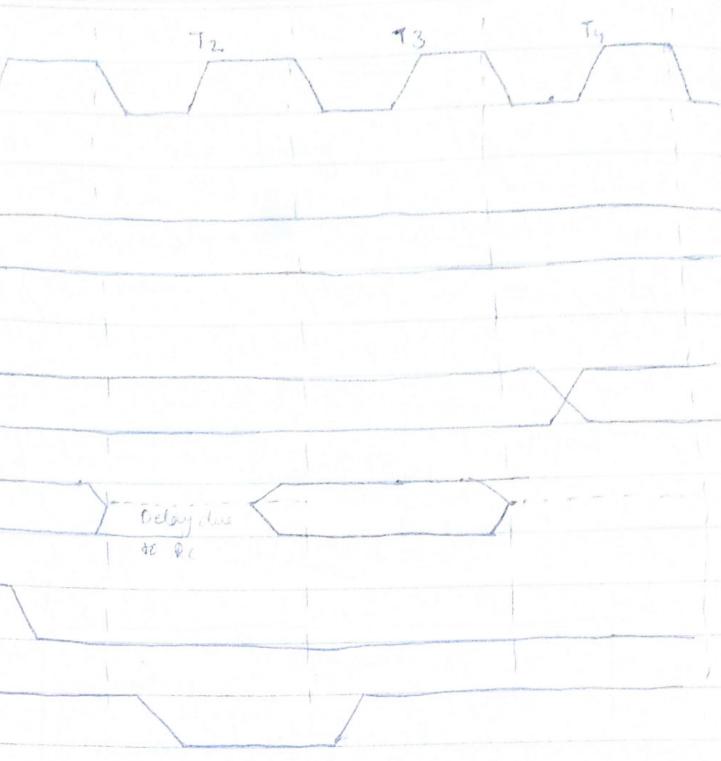
S₀, S₁

A₈-A₁₅

AD₀-AD₇

ALE

RD



Read

Write



A₈-A₁₅

AD₀-AD₇

BLF

WR

H G
Date

Page No. _____
Date _____

T₁ T₂ T₃

Read

Write

Page No. _____
Date _____

Instruction → Command given to μC or μP to fetch some data.

assembly language → mnemonics.

Classification of 8086 Inst.

1. Data Transfer Instruction → $R \rightarrow R$, $R \rightarrow M$, $M \rightarrow R$.
does not alter contents of register.
2. Arithmetic Add, Sub [ADD, SUB, INR, DEC ... etc].
3. Logical (AND, OR, CMP, RXOR, ANA, ORA, RAL, ^(MF)).
4. Branch, control group (Conditional & unconditional jobs).
5. I/O and Machine control group. (IN, OUT, PUSH, POP, INT, etc).

9/10/2022
Data Transfer → (MOV, MVI, LDI, LDA)

Branch & Control → JMP, CALL, JC, JZ, RST - etc

* Trap is non maskable interrupt. It cannot be ignored.

Trap subroutine → 0024 ← PC set to

INTR - address of subroutine is not fixed. It is specified by interrupt controller.

mention

What we can

→ Instruction →

- ① 8 bit or 16 bit addresses.
- ② Address of memory location, Input-output port or I/O device where data resides.
- ③ Registers [Operands are specified in instruction.]
- ④ Data is implied in some instruction that is instruction operates on content of Accumulator.

These techniques to specify data for instruction known as addressing modes.

In 8085, there are 4 addressing modes:

- ① Direct addressing.
- ② Register " "
- ③ Register indirect " "
- ④ Immediate addressing.
- ⑤ Implicit addressing

① Direct addressing → In this mode of addressing, the address of the operand is given in the instruction.

itself. , opcode
 Eg: STA 2400H , operand

Store MZ contents @ to 2400 memory location

But in coded form

32, 00, 24 → 3 byte instruction

- ② Register Addressing → In this mode of addressing, the operand is in one of the registers (general purpose register). The opcode specifies the address of the register in addition to the operation to be performed.

Eg: MOV A, B copies content of B to A
 runs in CPU only.

78 H single byte instruction.

ADD B

80 H

- ③ Register indirect addressing → In this mode of addressing, the address of one operand is specified by the register pair.

Eg: LXI H 2500H Load n-L pair with 2500H.
 Contents of Memory to BC.

n-L

MOV A, M

Registers indirect addressing →
 The contents of memory are stored in
 n-L pair

(d) LX1 H, 2500H

ADD M

HLT

register indirect addressing mode

Immediate

Immediate Addressing → In this addressing mode, the operand is specified within the instruction itself.

Eg: ① MVI A, 05

Move 05 to register A

② ADJ 06

Add 06 to Accumulator

2 byte instruction
/ →
opcode operand.

(e)

Implicit Addressing → operates on contents of AC.

→ Inst. do not require address of operand.

Eg: → CMA, RAL, RAR... etc.

→ addr - 16 bit address of memory.

→ data - 8 bit data

→ data 16 - 16 bit data

M - Memory whose addr is in n-l pairs

sp - register pair

rh - higher order register pair.

[E] - Content of register identified in bracket.

[E J] - Content of memory loc where value is in register pair.



Data Transfer

is - copy status

24/11/18

Instruction Set of 8085

(I) Data Transfer / Movement Instruction :-

(II) Between Registers

(a) MOV R₁, R₂

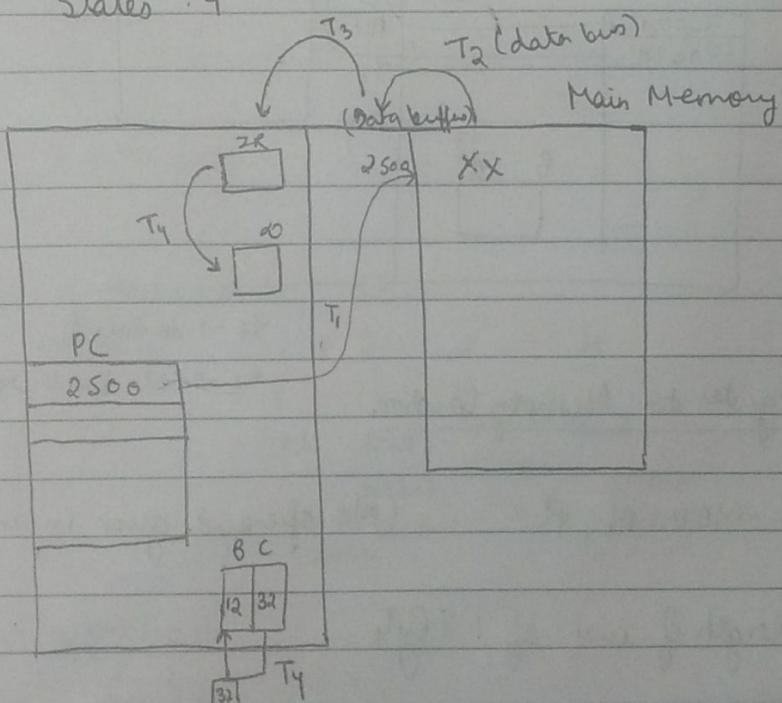
Length of Instruction : 1 Byte

(b) MOV B, C (2500)

One byte instruction

Machine cycle → 1

States : 4



At T₁ ALE = 1

Address bus → 16 bit address of memory location.

After T₁ ALE = 0, transfer contents to data bus & stored in data buffers.

① MVI R, data
(8 bit data).

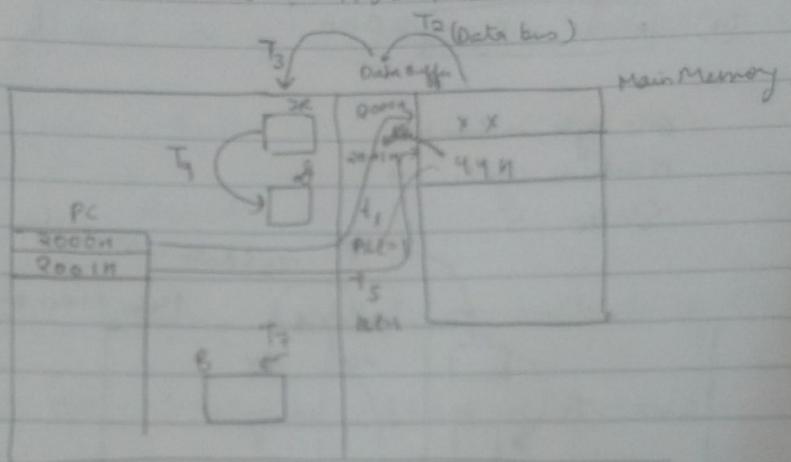
e.g. - MNMVI B, 44H. (Immediate Addressing)

length of instruction - 2 bytes.

M/C - 2 (1+1)

/ \
 4 3

$$\text{States} = 4 + 3 = 7 \text{ states}$$



\rightarrow T2 → Data Bus

\rightarrow T2 → Data Bus to Register

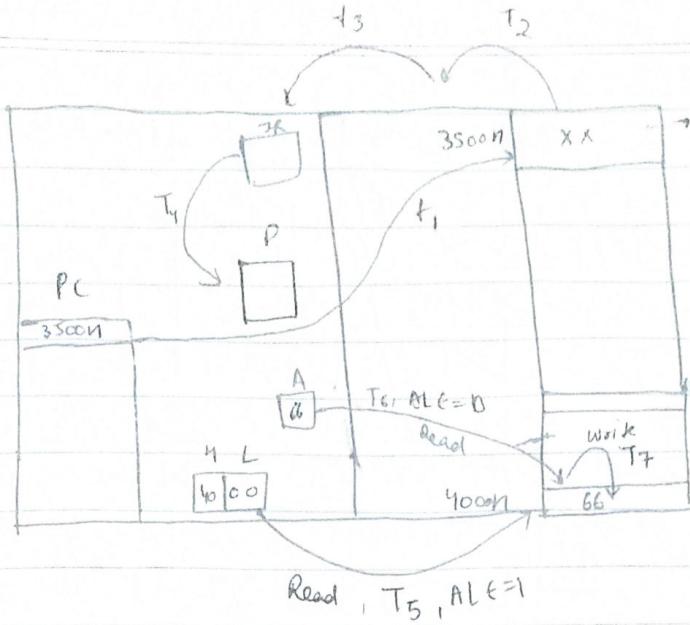
Register to Memory location

MOV M, R (no operand given in instr.)

length of inst of : 1 Byte

e.g. \rightarrow MOV M, A [whenever A is given, address present in HL pair]

PC will fetch only once as it is one byte inst.



$$M/C \rightarrow 1 + 1 = 2$$

states \rightarrow 4 states (Fetch) + 3 states (Write) = 7

L X 1 R_p , 16-bit value [Direct Addressing].
Register pair

Length of instruction 3.B.

$$M/C \quad 1 + 2 + 2 = 3$$

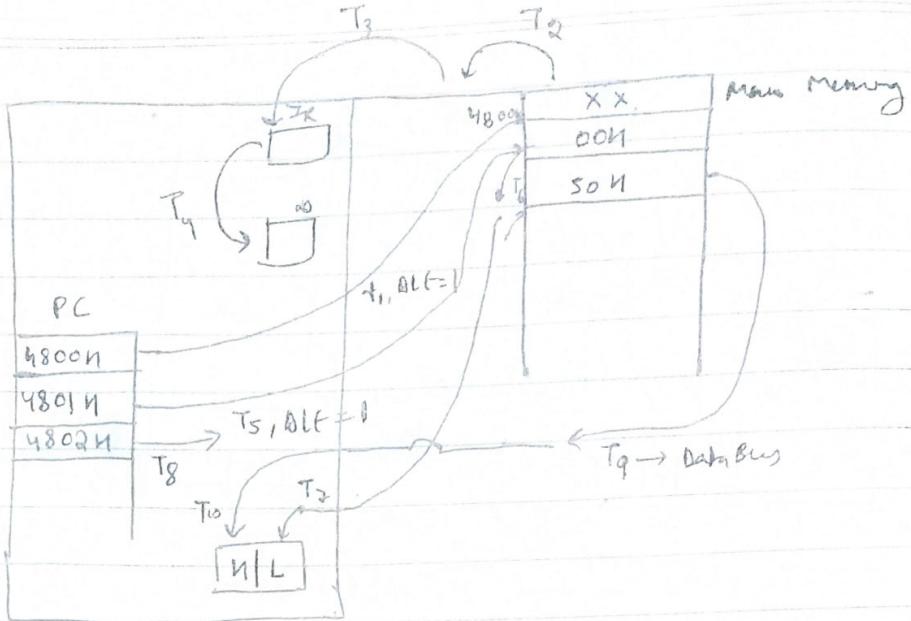
↓ ↓ ↘

Fetch Read Load Read

16-bit
8-bit value Reg R_p . of higher value.

e.g. L X 1 $n, \underline{\underline{5000H}}$
High Low

Total states = 10



(iii) LDA 16 bit address
 \downarrow
 Load acc

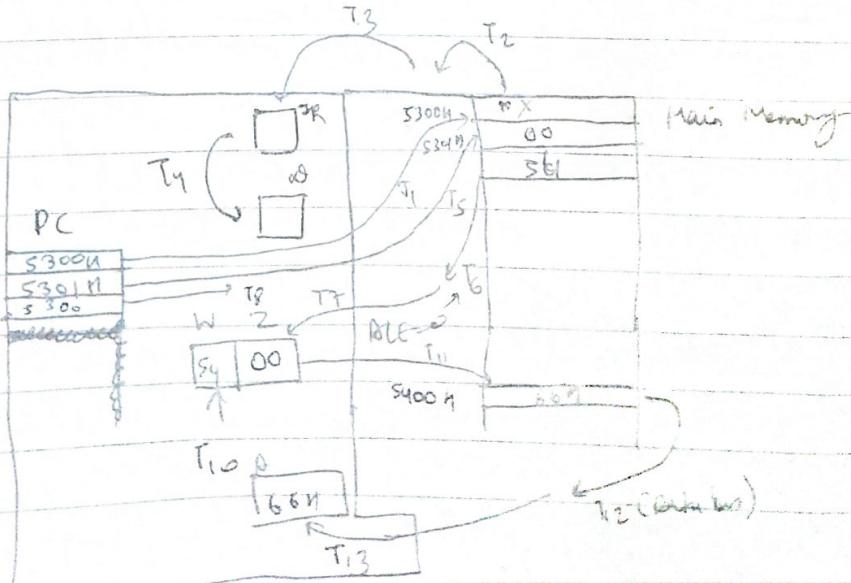
eg \rightarrow LDA 5400H [Direct addressing mode]
 [operand is given in acc]

Length of inst \rightarrow 3 B.

$$\text{M/C} \rightarrow 1 + 2 + 1 = 4.$$

↓	↓	↓
Fetch	Read	Load
16 bit	into	
address	D	

States $\rightarrow 4 + 3 + 3 + 3 = 13$ states



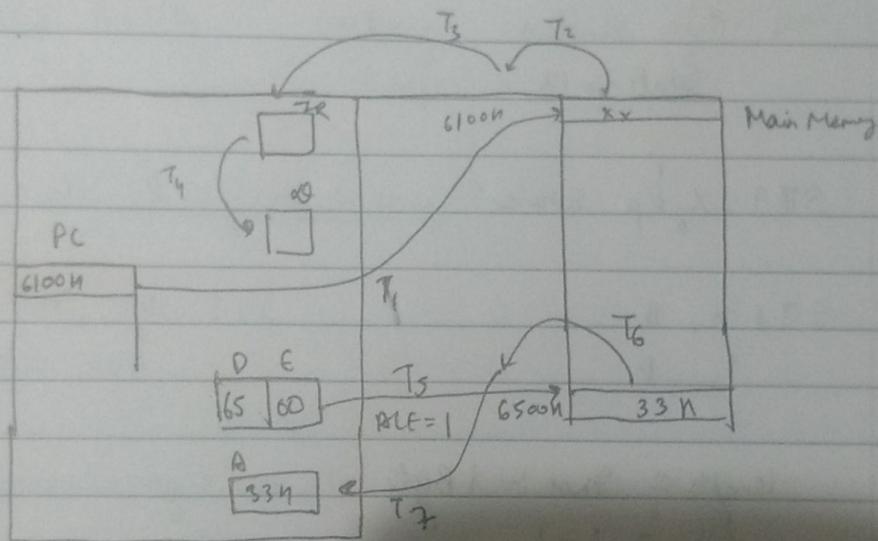
W-2 used to store temp data

(iv) ~~eg~~ LDA X, Rp

length of inst \rightarrow 1B.

PC \rightarrow will fetch only one
load Acc with data whose address is present in Rp.

eg: \rightarrow 6100; LDA X, D OR use MOV RM



$$\text{States} = 4 + 3 = 7$$

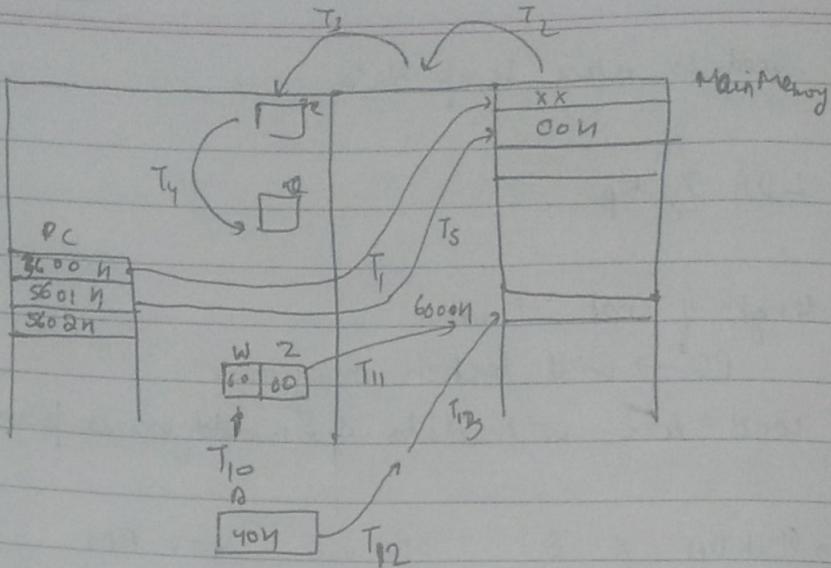
(v) ~~STA X, Rp~~ STA 16 bit add/value.
↓
↓
Store Acc

Store contents of Acc to 16 bit address

Inst. length: 3B.

Machine cycle: \rightarrow 4 = 1 \downarrow 2 \downarrow 1
fetch read write.

$$\text{states} = 13$$



States = 13

STAX X, fp store

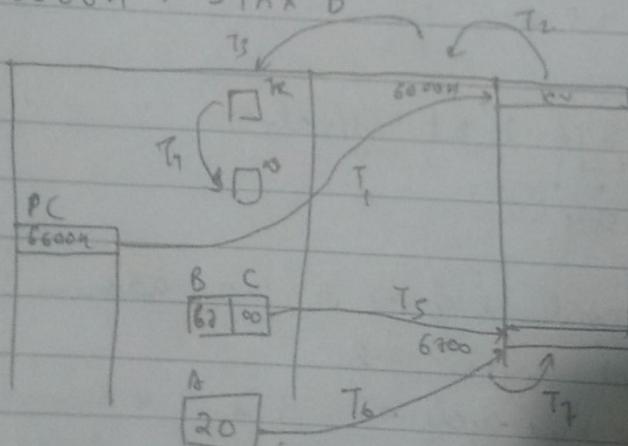
STAX X, B

" D

length of 1 byte :- 1 Byte

$$M/C = 1+1$$

6600H : STAX B



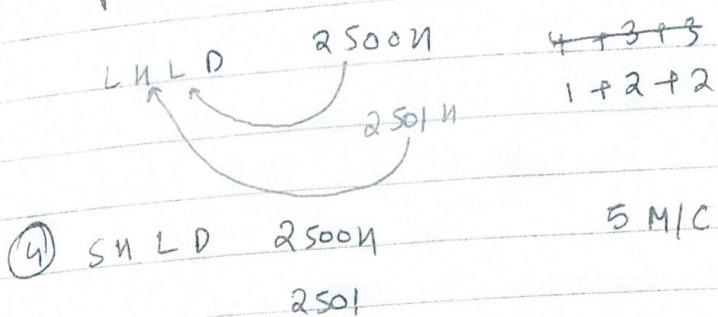
Data Transfer - ① MOV R, M [Move data from memory]

② MOV M, R

③ LHLD add [Load HL pair direct]

contents of Mem loc whose address is specified by

second & third byte of int is loaded into register L & contents of next mem location is loaded into register H.



⑤ XCNG Exchange contents of NL pair with DE.
1 M/c

(ii) Arithmetic & Logical Group :-

ADD, SUB, INC, DEC

OR, AND, XOR

One operand / byte is in AC

1. Addition :-

{ ADD R. [Add R data with AC & store result in AC].

Without
carry

{ ADD M [Add Memory content to AC]

{ ADD I 8 bit value [Add value to AC & store result in AC]
[Shift of content]

{ ADC R

With
carry

{ ADC M

{ ADC 8 bit value

Sub :-W/mov
Burrow

SUB R SUB M

SUB 8 bit value

With
Burrow

SBB R SBB M

SBI 8 bit value

carry flag treated as burrow flag.

Logical Inst :-

AND R AND M ANDI 8-bit value

OR A R ORA M ORI 8bit value

XRA R XRA M XRI 8-bit value

CMP R CMP M CPI 8-bit value

Increment/Decrement Inst :-

INR R INR M INX Rp

RP-H
B
D
SP

DCR R DCR M DCX Rp

CMA Implicit Inst :-

CMA CMC (complement carry status).

Rotate :- Rotate R only.

RLC

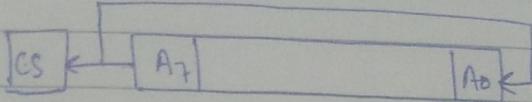
RR C

RAC

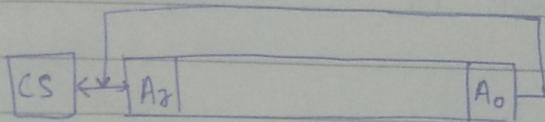
RAR

 Rotate →

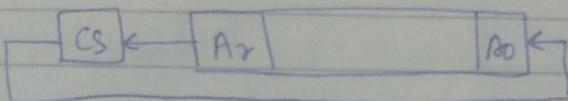
RLC →



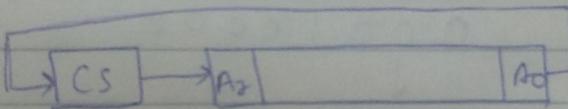
RRC



RAL → Rotate Ax left through Carry.



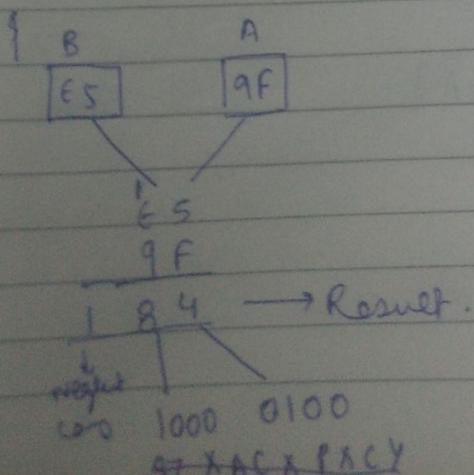
RAR →



Arithmetic shift →

① ADD R 1B, 1M/C, 4 to states

e.g. → ADD B.



Parity
Even = 01
Odd = 10

S2 X BC X P X CY
10 0 1 0 0 0 1

95

97 X AC X P X CY

PSW : \rightarrow R_c + Flag bit

PSW [84H | 95H]

(ii) ADD M

Eg: \rightarrow 9000H : ADD M



B-10
11
17
13

9000H	X X
9300H	
n L	

9300H	35
-------	----

A
20

$$\begin{array}{r}
 1 \\
 3 \cancel{5} 3 \\
 + 2 D F \\
 \hline
 6 \cancel{8} 2 H
 \end{array}$$

01000010

S Z X A C X P X C Y

0 0 0 1 0 0 0 0



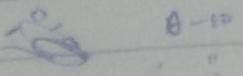
910H

PSW: \rightarrow AC + Flag bit

PSW [84H] 95H

(ii) ADD M

e.g. 9000H : ADD M



A-10
B-11
C-12
D-13

9300H	XX
9300H	35
HL	

18
20

$$\begin{array}{r} 1 \\ 353 \\ + 2DF \\ \hline 682H \end{array}$$

01000010

S Z X A C X P X C Y
0 0 0 1 0 0 0 0

↓

910H.

PSW [62H] 10H

Addition with Carry: \rightarrow

(i) RDC R \rightarrow

$$[A] \leftarrow [R + C + A]$$

1Byte, 1M/C, 4 states.

eg: ADC H

A	R	C	S Z X AC X P	X C Y
[4C]	[84]	[1]	D ₇ D ₆ D ₅	D ₄ D ₃ D ₂ D ₁ D ₀

1
4C F

B	X	I
1	00	H
1		

4C F
BY I
100

1000000001

PSW	S Z X AC X P X C Y
	0 0 0 1 0 0 0 1

1	4C F
1	BY I
1	100

PSW	0111	111
-----	------	-----

SUB

Signed \rightarrow S \rightarrow 0 \rightarrow +ve
 $1 \rightarrow -ve$.

Carry flag \rightarrow ignored.

Unsigned op \rightarrow C Y \rightarrow 0 \rightarrow +ve.
 \rightarrow 1 \rightarrow -ve.
S \rightarrow X \rightarrow ignored.

SUB C

AC	C
[87]	[23]

Manual Method \rightarrow

87
- 23
64 H.

LNR1 < LNR18
if TNEAE \rightarrow AC \rightarrow 0
AC \rightarrow 0

110 0100

Processor Method

$$A + (-C)$$

$$\begin{array}{r} C \quad 15 \quad 16 \\ \boxed{23} \quad F \quad 16n \end{array}$$

$$\begin{array}{r} 2 \quad 3 \\ D \quad \boxed{D} \quad n \end{array}$$

$$85 \quad 87 \quad 5$$

$$F \quad D \quad D \quad F$$

$$1 \quad \boxed{6} \quad 4 \quad n$$

$\downarrow \downarrow$

$$0110 \quad 0100 \quad n$$

$$\begin{array}{r} 87 \quad 23 \\ \hline F \quad 10 \end{array}$$

$$00100011$$

$$11011100$$

$$\begin{array}{r} + \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\ \hline 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \end{array}$$

$$\begin{array}{r} 82.5 \\ \hline DD \quad F \end{array}$$

$$\begin{array}{r} 1 \quad 6 \quad 4 \quad 1 \\ \hline 6 \quad 4 \end{array}$$

$$\begin{array}{r} 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \\ 3 \quad 2 \quad AC \quad P \quad \boxed{CY} \end{array}$$

$$000010000 \quad 0000$$

$$00010000$$

PSW

$$\boxed{64} \quad n \quad \boxed{10} \quad n$$

SUB M

$$1B, 2M | C, 7$$

↓

1 fetch

1 Read.

S0J 8bit valueS0I BC

hexadecimal number

$$\boxed{25}$$

$$A + (-C)$$

$$n - 10$$

$$B - 11$$

Manual : $\rightarrow 125 \quad F$

$$- BC$$

$$\begin{array}{r} 1 \quad 1 \quad 3 \\ \hline \end{array}$$

negative

$$\begin{array}{r} 16 \quad 11 \\ 1 \quad 2 \quad 5 \\ \hline 25 \end{array}$$

$$\begin{array}{r} 11 \rightarrow BC \\ 0011 \quad 0010 \\ \hline 0010 \quad 0011 \end{array}$$

$$\begin{array}{r} 16 \quad 11 \\ 25 \quad 5 \\ \hline 12 \end{array}$$

15 16

$$\begin{array}{r} 8 \\ - 4 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 25 \\ - 44 \\ \hline 69 \end{array}$$

$$\begin{array}{r} 01101001 \\ - 00000101 \\ \hline \end{array}$$

Complement carry bit in subtraction

Sub with Borrow

SBB R. 1B, 1M | C, 4t states.

$$[A - R - C_4]$$

$$[A - (R + C_4)]$$

$$\begin{array}{ccc} M & R & C_4 \\ \boxed{11} & \boxed{01} & \boxed{1} \end{array}$$

b

$$\begin{array}{r} DF \\ - 1 \\ \hline DF. \end{array}$$

$$\begin{array}{r} 16 \quad 16 \\ 71 \\ \hline DF \\ - 292 \\ \hline \end{array}$$

10010010

SZ X AXXP & C4

10000000 0

8

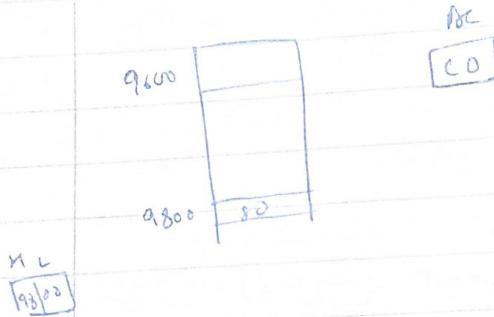
1

$L_N[\text{Result}] > L_N[M]$

Hence $A \times = 1$.

* SBB M

9600n : SBB M



$$\begin{array}{r}
 \text{CO} \\
 \underline{80} \\
 \text{40} \quad \text{N} \\
 \hline
 0100\ 0000
 \end{array}$$

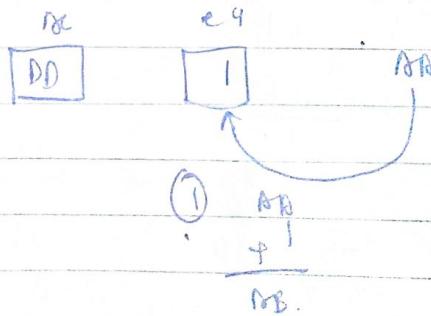
 SZ X AC X PX CY
 00000000.

* SBI 8 bit data

SBI AAH

Inst length = 2 byte
2 M/C

7 states



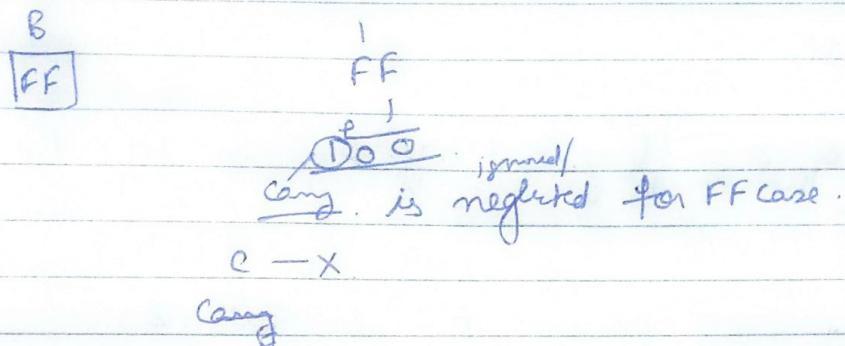
$$\begin{array}{r}
 \text{DD} \\
 + \text{BB} \\
 \hline
 \text{32 H}
 \end{array}$$

 00110010 BC \rightarrow 1 bit
 D > 2

SZ X AC X PX CY
 00010001

① INR R.

IR INR B.



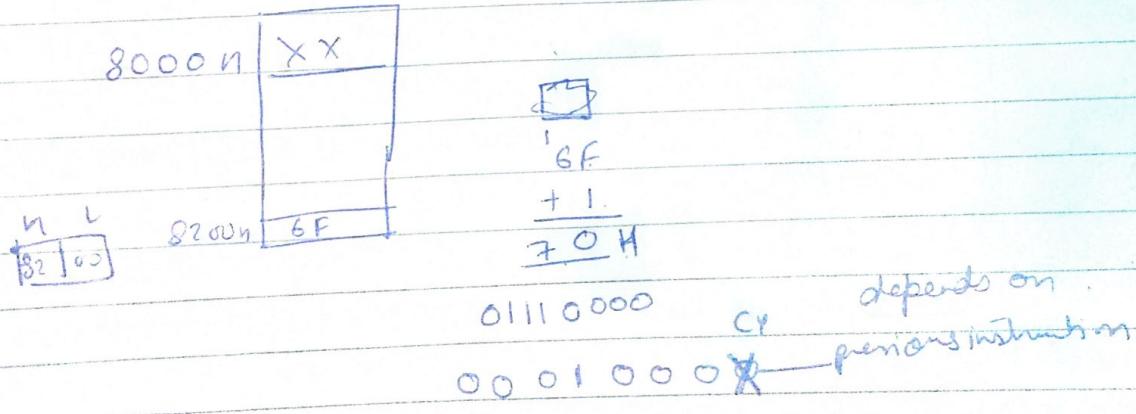
② INR M

IB, 3 M/C, 10 states.

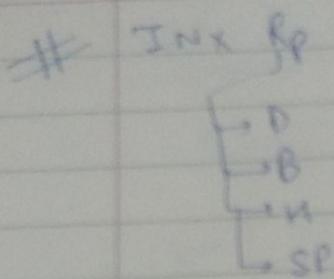
I
fetch
Read
Write

* In INR M & DECM take 3 M/C.

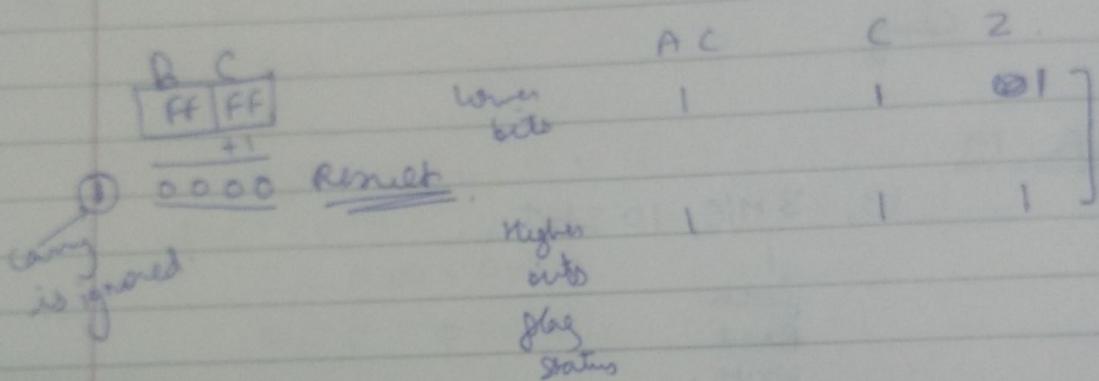
8000N : INC M



PSW 70H | 10H



* None of the flags are affected. in INX Rp, 2
DC X Rp



Due to two different values of flag bits for two bytes of $\text{DC} \times \text{Rp}$, the flag bits are neglected.

DCX Rp

Status of DC flag is checked at manual method of subtraction.

DCR C

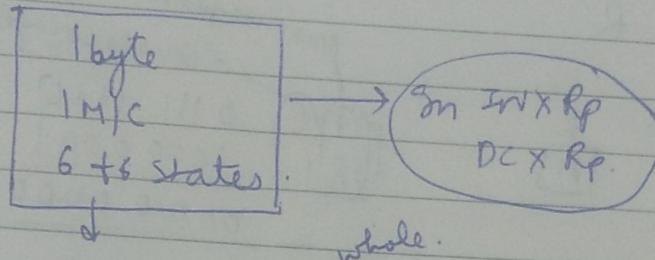
$$\begin{array}{r}
 \boxed{C} \\
 00 \\
 \end{array}
 \begin{array}{r}
 15 \ 16 \\
 0 \ 0 \\
 - \ 1 \\
 \hline
 F \ F
 \end{array}$$

1 0 0 1 0 1 0 1 X → Status of first

DCR M

3 M/C
|
Fetch, Read, Write.

DCX Rp.



Delay due to work done by processor hence
2 extra time cycle states is taken.

DC X H

H	E
50	00

$$\begin{array}{r}
 50\ 00 \\
 - 1 \\
 \hline
 4F\ FF
 \end{array} \text{ Result}$$

No flags affected.

* PUSH , INX , DCX → 6 time states due to delay .

logical instructions : →

AND, OR, XOR,

→ Carry flag is always 0. CY=0.
AC flag

- $AC = 01$ [AND]
 $AC = 0$ [OR and XOR].

AND is used to reset bit of data

AND R

\boxed{A} \boxed{R}

one's complement
one's complement.

$\boxed{11111110}$	$\boxed{10000001}$
<hr/>	
00000000	

$$S = X \cdot AC \cdot P \cdot C.Y \\ 0 \ 010 \ 010100$$

* AND M

1B, 2M/C, 7+ states

$\boxed{B \mid C}$	\boxed{A}	<u>Result</u>
	34	$34 \text{ } N$
		<hr/>
		0010100

$$AC = 1$$

$$C = 0 \quad 00010000$$

AND 8bit Value.

AND 16N

\boxed{AC}	$\boxed{32}$	$\boxed{A \mid 16}$	$32 \quad 00110010$
			$\underline{00010110}$
			$\underline{00010010}$

$$S = X \cdot AC \cdot P \cdot C.Y \\ 00010100$$

OR Instruction  Immediate value.

(i) ORA R $C_4 = AC = 0$

(ii) ORA M $C_4 = AC = 0$

(iii) ORI

XOR Inst

$C_4 = AC = 0$

XRA R

XRA M $C_4 = AC = 0$

XRI 8bit Value $C_4 = AC = 0$

CMA \rightarrow complement contents of Acc
1B inst.
no flag affected.

CMC \rightarrow complement carry status.
other flags are not affected.

SDESTC \rightarrow Set carry status, others are not affected.

CMP R \rightarrow compare contents of reg into twin R
result store in R.

compare means subtraction.

flag status changes all to result.

Result is discarded and

$>, =, <$ is output.

Content of R are unchanged.

Result is given by status of C and Z flag.
The status of remaining flags must be found from the _____

~~#2/18~~

Stack / I/O and Machine Control Group:

- PUSH [subroutine call, return, interrupt].
- POP

PUSH fp — BC
— DE
— HL
— PSW

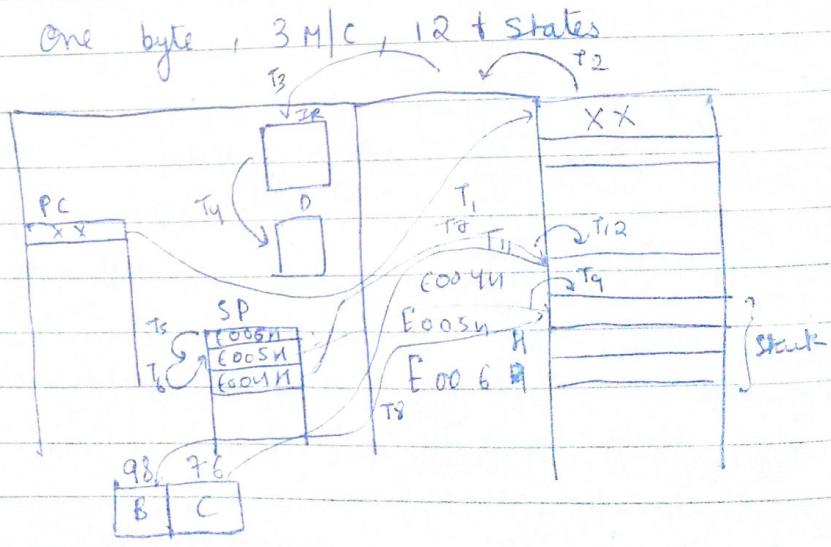
e.g.: → PUSH B

→ DEC SP

→ LIFO

→ uses Stack Memory.

One byte, 3 M/C, 12 t states



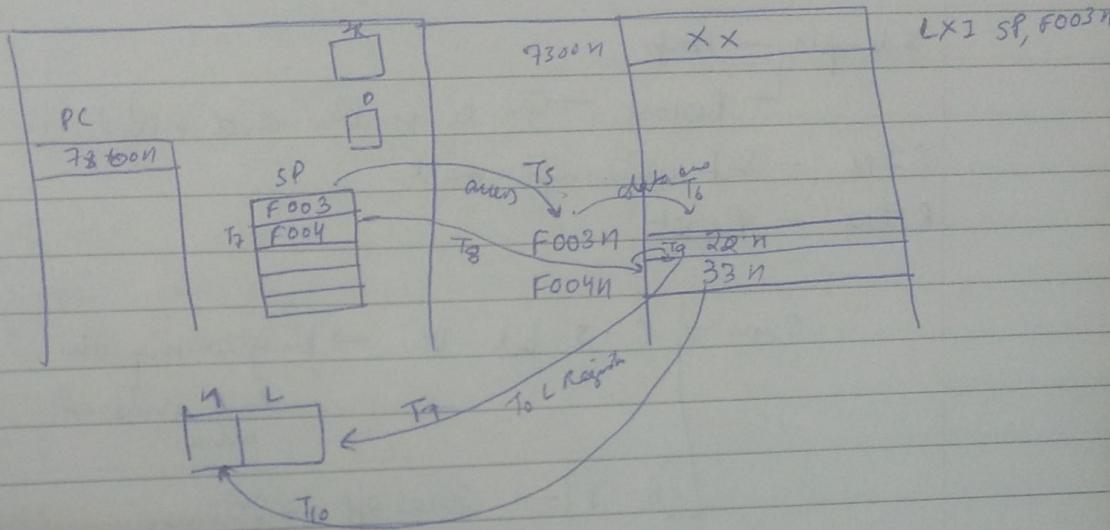
SP ← SP - 1

LXI SP, F006H.

$$\text{SP} \leftarrow \text{SP} - 1$$

2. POP RP 1B, 3, 1Q T states.

7300H : POP H



At T8, SP incremented.

Used in return instruction.

Input / Output InstructionsI/O ... IN port_addr
OUT

IN [0] port address

AC ← Input Port

OUT [0] port address

AC → Output Port

PUSH	PSW
POP	PSW

HLT [stop no op].

XTHL [exchange stack top with HL pair]

SPHL [move HL pair to SP].

EI [Enable interrupts]

DI [Disable interrupts].

Interrupts — Trap

Restart — can be enabled or disabled.

SIM — Set interrupt mask.

RIM — Reset "

SIM — $\begin{cases} 0-5 \text{ bits } 1\text{b} \\ \text{rest } 0\text{b} \end{cases}$ → programming the restart interrupt masks.

{6-7} → serial o/p → S0 line.

RIM → load A with pending interrupts.

→ Restart interrupt mask and contents of index S ID.

NOP → No operation [increase delay b/w two instructions].

#F

Branch Group

Conditional

Unconditional.

→ Jump addr (label) [2M/c] [unconditional jump].

→ cond. Jump addr (label) [3M/c] cost

(i) JC addr (label)

(ii) JNC " (")

- (iii) JZ
- (iv) JNZ
- (v) JP [Jump if Parity]
- (vi) RJM [Jump if Minus]
- (vii) JP [Jump if Plus]
- (viii) JPE
- (ix) JP O
call addr(label).

① Cond. Call

CC add \rightarrow Call subroutine if CY = 1.

① CC addr (label).

② CNC

③ CP

④ CM — Check sign flag.

⑤ CPG

⑥ CPO

⑦ CZ

⑧ CNZ

⑨ Return \rightarrow [Ret.]

RC addr (label)

① RNC

③ RP

④ RM

⑤ R2

⑥ RNZ

⑦ RPE

⑧ RPD

PCNL \rightarrow transfer HL Pair with to PC.

