

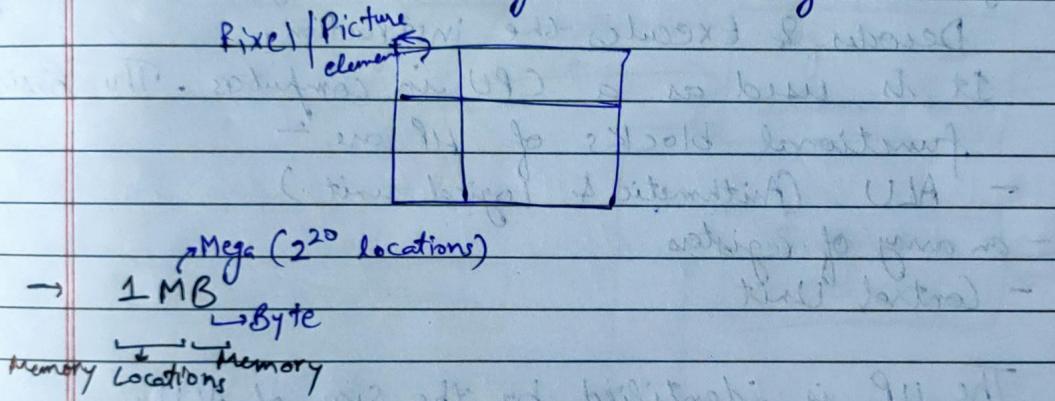
## 8085 μP

- ★ (b) Bit - A bit is a digit of the binary no. or code
- ★ Nibble - A group of 4 bits / 4 bit Binary no.
- ★ (B) Byte - 8 bits

$$(0110)_2 \rightarrow (0,1)$$

Each digit is a bit (in binary system)

→ In all other systems it is just called digit.



- ★ Word Size - 16 Bit Binary no.
- ★ Double Word Size - 32 Bit Binary no.
- ★ Multiple Word - 64, 128 ...
- ★ Data - Key quantity that is operated by the instruction of program is called data.
- ★ Address - It is an identification no. of a memory location.

→ 8085 μP uses 16 Bit address.

★ Memory Word Size:- The memory word is the size of the binary information that can be stored on a memory location.  
 → The Memory Word Size for 8085 μP is 8 bit. 1 byte  
like in 1MB

★ Microprocessor (μP):- μP is a semi programmable semi-conductor device which fetches the instruction & data from the memory. Decodes & Executes the instruction.

# It is used as a CPU in computers. The basic functional blocks of μP are :-  
 - ALU (Arithmetic & Logical unit)  
 - an array of registers  
 - Control Unit

★ The μP is identified by the size of ALU.

\* If ALU is 8 bit then μP is 8 bit  
 \* " .. is 16 bit .. .. 16 bits

★ Bus:- Data, Address & Control Signal to Transport

$\downarrow K_0$ Data Bus Koregi	$\downarrow K_0$ Address Bus Koregi	$\downarrow K_0$ Control Bus Koregi
--	---	---

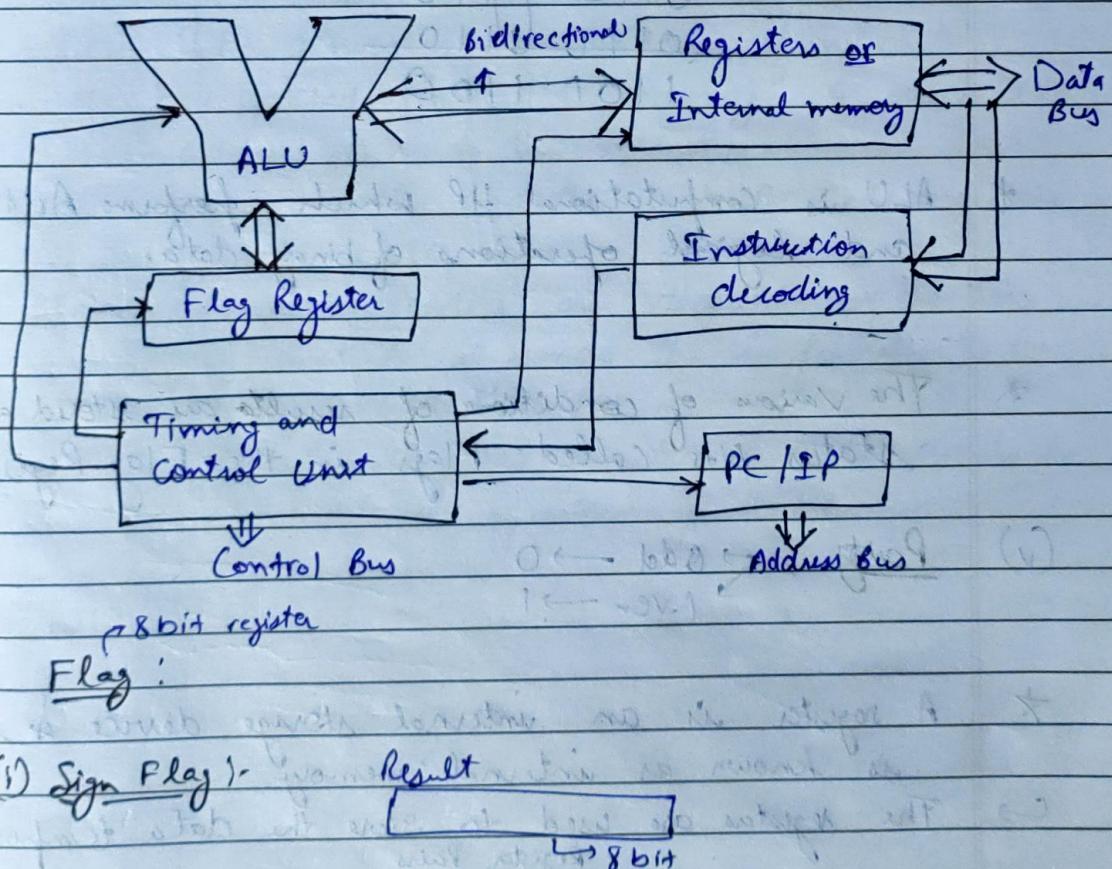
↳ A bus is group of conducting lines that carries data, address & control signals.

\* CPU Bus :- CPU The group of conducting lines that are directly connected to the CPU is called a CPU Bus.

\* In a CPU Bus the signals are multiplexed.

More than 1 signal is passed through the same line but at different timing.

\* Functional Building Blocks of 8085 HP :-

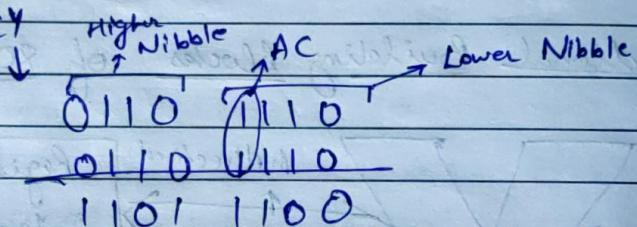


(i) Sign Flag :- Result

8 bit

$MSB = 1 \rightarrow S = 1$

otherwise  $S = 0$

(ii) Zero Flag :-  $Z = 1$  $[00000000]$ → All bits 0  
then  $Z = 1$ (iii) CY (Carry) :-If MSB generates carry then  $F_{lg} = 1$   
otherwise  $F_{lg} = 0$ (iv) AC (Auxiliary Carry Flag) :-

\* ALU is computational IP which performs Arithmetic and logical operations of binary data.

\* The various conditions of results are stored as status bits called Flags in the Flag Register.

(v) Parity :-  $\begin{cases} \text{Odd} \rightarrow 0 \\ \text{Even} \rightarrow 1 \end{cases}$ 

\* A register is an internal storage device &amp; it is known as internal memory.

↳ The registers are used to store the data temporarily.

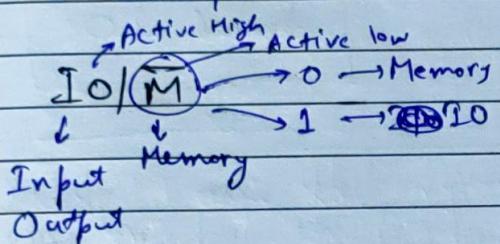
Register Pairs
BC → 16 BIT
DE → 16 BIT
HL

PC → Address of Next Instruction to be fetched.  
↳ 16 Bit for 8085 HP

PAGE NO.	/ /
DATE	

\* For any HP, there will be a set of instructions given by the manufacturer;

\* For doing any useful work with the HP we have to first write a program using these instructions & store them in a memory device external to the HP.



\* The IP generates the address of instructions to be fetched from the memory & sends it to the address bus to the memory. The memory will send the instruction code and data through the data bus.

The instruction codes are decoded by the Instruction Decoding Unit & it sends the information to the Timing & Control Unit.

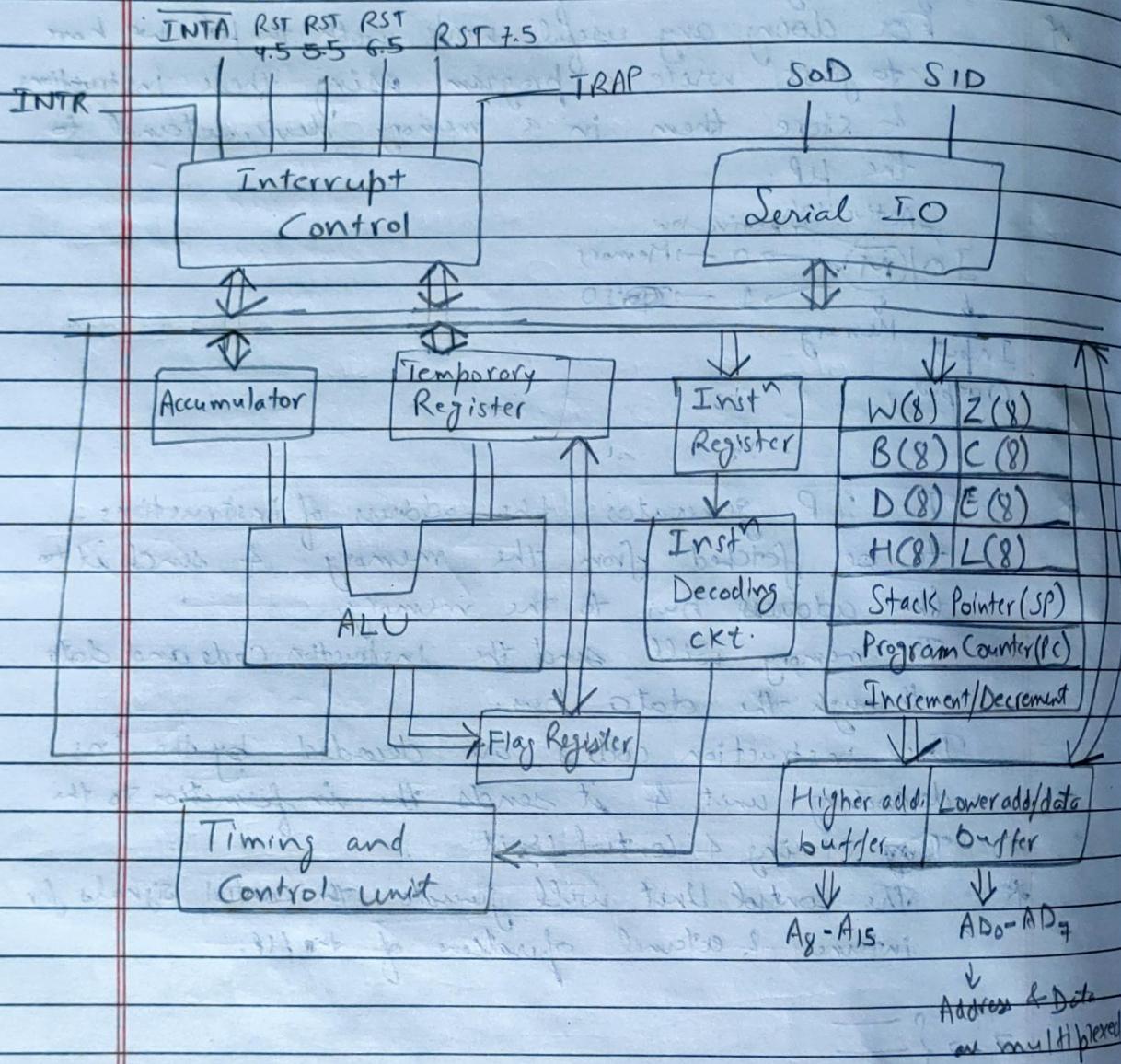
\* The Control Unit will generate the control signals for internal & external operations of the HP.

$Z = 0 \left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right.$

JZ 001017  
↓ ↓ Then jump  
Jump zero to this address

$$\begin{aligned} A > B &\rightarrow A - B > 0 && \xrightarrow{\text{positive}} \text{Sign Flag} = 1 \\ A = B &\rightarrow A - B = 0 && \rightarrow \text{Zero Flag} = 1 \\ A < B &\rightarrow A - B < 0 \end{aligned}$$

## 8085 Architecture



1976 → 8085  
mainly based on 8 bit

PAGE NO.	
DATE	/ /

ALE (Address Latch Enable) → used for demultiplexing data from Address or via <sup>versa</sup>

Related to  
Timing +  
Control  
Unit

Reset IN

RESET OUT

I/O M

RD

WR

READY

HOLD

S<sub>0</sub> } Status  
S<sub>1</sub> } Signals

x<sub>1</sub>

x<sub>2</sub>

Inst<sup>n</sup>  
Set

1	1	0	0	1	0	0
S	Z	X	AC	X	P	CY

Flag Register

- \* 8085 includes an ALU, a timing & Control Unit, Inst<sup>n</sup> Register, Decoder circuit, a register array & interrupt control & Serial I/O Control.
- \* The operations performed by the ALU of an 8085 CPU are addition, subtraction, Increment/Decrement, Logical AND, OR, ~~exclusive OR~~, complement, compare, left or right shift.
- \* The Accumulator & temporary register are used to hold the data during ~~the~~ Arithmetic & Logical operation. After an operation the result is stored in an accumulator & the flag register is set/reset according to the result.
- \* The Accu. & Flag register together is called as program status word (PSW).  
 $\leftarrow$  Group of 16 bit

\* If PSW is 0100H then Accumulator is 011H (Higher bits)  
Flag register is 001H. (Lower bits)

\* There are 5 flags in the flag register.

\* Sign Flag:- If the MSB is 1 then result is -ve, SF = 1  
If the MSB is 0 then result is +ve, SF = 0

Zero Flag:-

\* If the result is 0 then the zero flag is 1.  
otherwise 0.

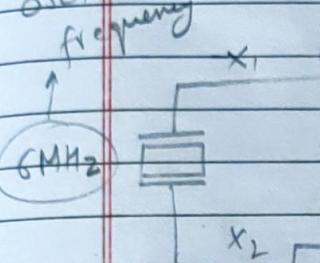
\* Auxiliary Carry:- If the lower Nibble generates the carry then  
Add of two  
that carry is AC and AC = 1.

\* Parity Flag:- odd no. of 1's then PC = 0  
otherwise PC = 1

\* Carry Flag:- If the addition of 2 nos. generates the  
carry after adding the MSB bits then that  
carry is carry flag, CF = 1

# \* 8085 HP Pin diagram & assignments :-

oscillator (Crystal)  
frequency



CLKOUT

Reset IN

Reset OUT

Ready

I/O/M

RD

WR

S<sub>1</sub>

S<sub>0</sub>

(5V) +Vcc

(0V) GND

8085

HO PIN  
IC

3MHz

$\Rightarrow A_{15} - A_8 \rightarrow HOA$

$\Leftrightarrow AD_7 - AD_0 \rightarrow$  Lower order Address

ALE → Address

SID → Latch enable

SOD → Serial communication

RST 7.5 → Highest Priority

RST 6.5

RST 5.5 → Interrupt

INTR → Interrupt Request

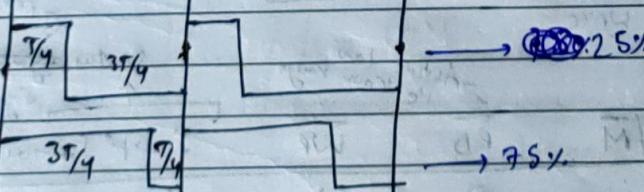
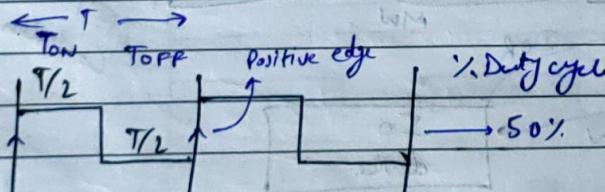
INTA → Acknowledgment

HOLD

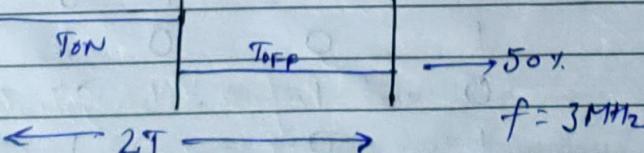
DMA

HLDA

X<sub>1</sub>, X<sub>2</sub>  
6MHz



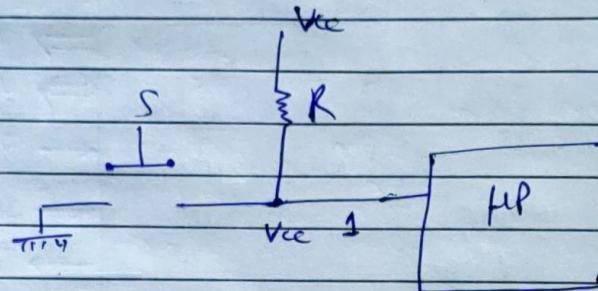
$$f_{clock} = \frac{f_{crystal}}{2}$$



$f = 3MHz$

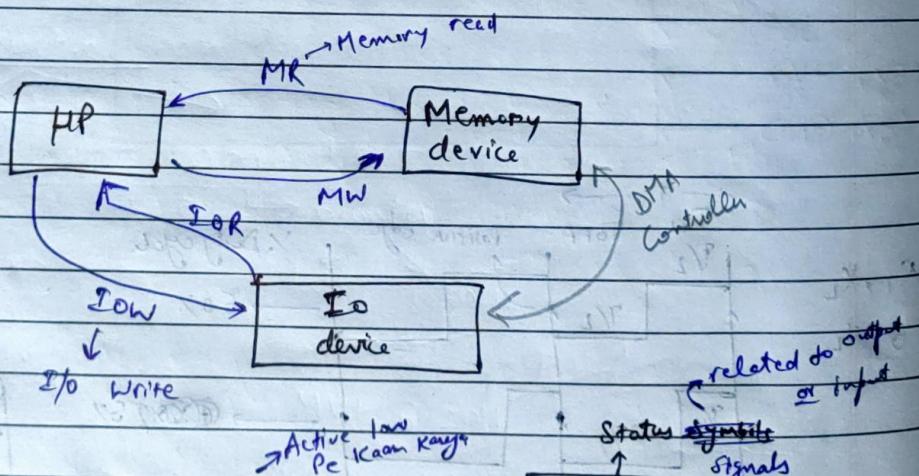
when this is 1  
then P will be  
toggled

$$\% \text{ duty Cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$



Reset IN → Active low ( $\bar{0}$ ) <sup>work</sup> <sub>high</sub>

- \* Ready → It is used to tell the processor whether other devices in the communication ready to communicate or not.
- eg. Printer
- For slow devices.



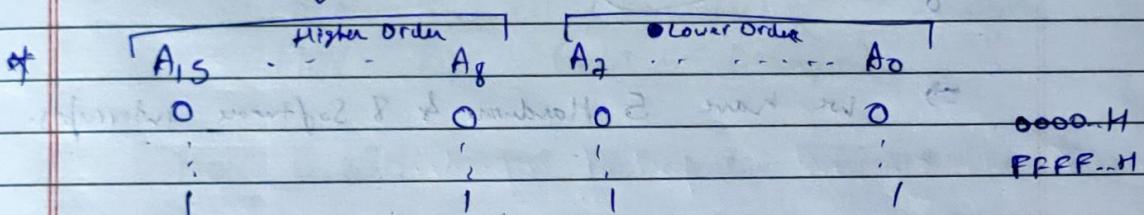
I/O/M	RD	WR	S <sub>1</sub> Input	S <sub>0</sub> Output	Notes
MR	0	0	1	1	0
MW	0	1	0	0	1
IOW	1	0	1	1	0
IOW	1	1	0	0	1

e.g. Music

MR  $\rightarrow$  HP  $\rightarrow$  IOW  
 (Player) (Song)

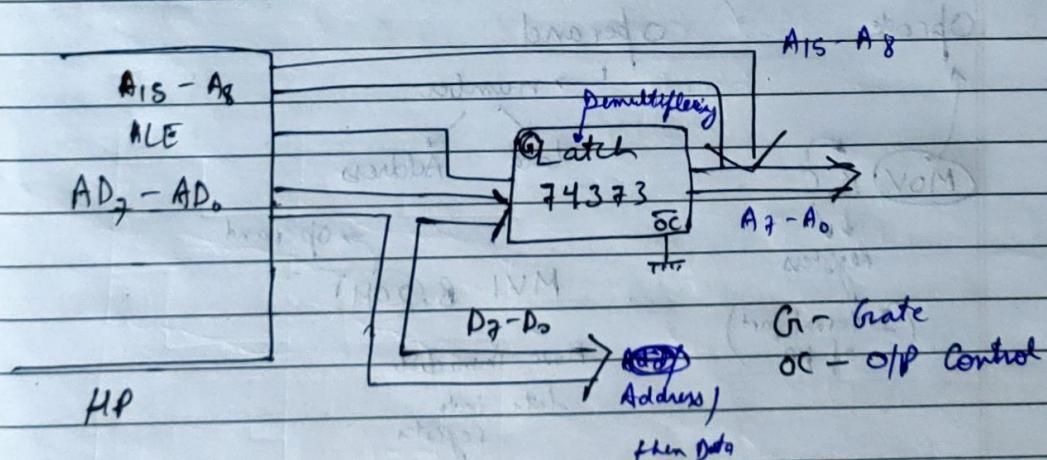
e.g. Keyboard: IOR  $\rightarrow$  HP  $\rightarrow$  MW

\* Status signals (S<sub>1</sub>, S<sub>0</sub>) are defined for HP.



\* ALE = 1, address selected

ALE = 0, data selected



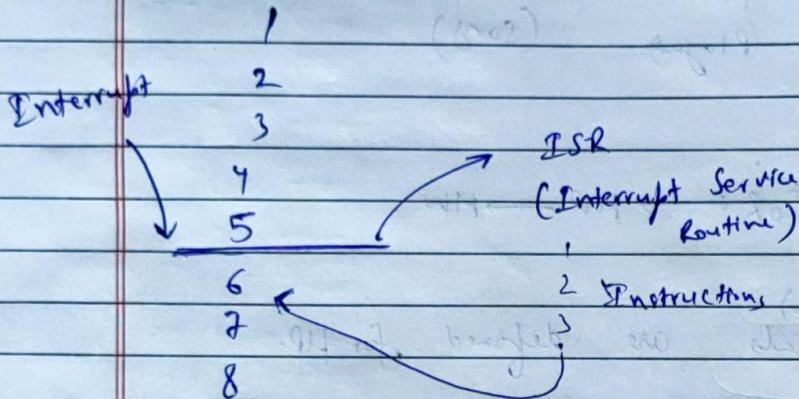
\* Serial Comm.: It is used for long distance.

$\swarrow$  Bit by bit transferred  
 $\searrow$  Slow

Serial Input device
Serial Output device

\*

Parallel Comm. - all transfer in one go.  
↳ Fast



⇒ we have 5 Hardware & 8 Software interrupts.

#

8085 CPU Addressing modes :-

Instruction :-

Opcode

MOV

Registers

(Internal part  
of CPU)

operand

→ number

↓  
data

↓  
Address

MVI B,20H

Move Immediate  
data into  
register

operand

\*

Register :-

→ General Purpose (A, B, C, D, E, H, L)  
→ Specific (Stack pointer, PC, Flag Register)

\* Stack → Arrangement of memory locations.

\* Addressing Mode :- The manner in which an operand is given in the instruction.

↳ Immediate A.M. :-

→ Data is directly given in the instruction  
 1 byte for opcode                          1 byte for operand

MVI B, 25H (Example) → 2 byte

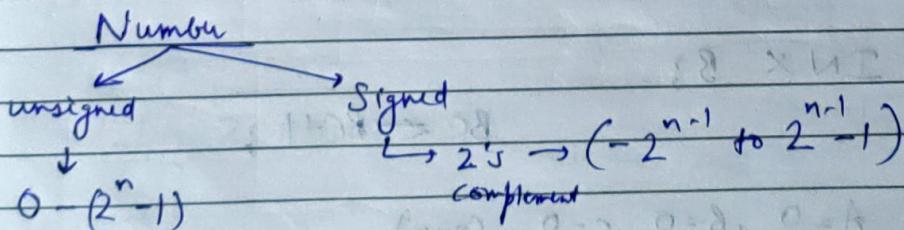
MOV B,C (Not an Example)

MOV B, 25H → Not valid instruction for 8085 CPU

Note :-

$$\begin{array}{r}
 25H \\
 + 3AH \\
 \hline
 5FH
 \end{array}
 \quad
 \begin{array}{r}
 00100101 \\
 00110100 \\
 \hline
 0101\ 111
 \end{array}$$

$$\begin{array}{r}
 38H \\
 + 3AH \\
 \hline
 72H
 \end{array}
 \quad
 \begin{array}{l}
 \text{Hex. decimal} \\
 F = 15 \\
 10 = 16 \\
 11 = 17 \\
 12 = 18
 \end{array}$$



↳ Dream 11 aphi banaani  
 kai toh jeh no gae  
 Karoge.

12/11/070

Advantage — You Know Data.

Disadvantage — Memory Size big (MA)

Ex:-

L X I B, 2000H  
Data

Register Pair

B C ?  
D E }  
H L }

→ Use to  
store to  
bit data

$$\begin{aligned} \rightarrow B &= 20H \\ C &= 00H \\ D &= 20H \\ E &= 00H \end{aligned}$$

\* No. is not a part of opcode.

↳ 2) Register Addressing Mode :-

↳ Data is given in register.

Ex:-

MOV B, C ; , B = C

opcode  
↳ 1 byte

Increment

Ex:-

INR B ;  $\leftarrow B+1$   
1 Byte

Ex:-

INX B ;

$BC \leftarrow BC+1$  ;

Ex:-

A = 0, B = 0, C = 0 (say)

(a) MVI A, 00H  
 MVI B, 00H  
 MVI C, 00H ] → 6 Byte

(b) MVI A, 00H  
~~MVI~~ MOV B, A  
 MOV C, A ] → 4 Byte

(c) SUB A  
 MOV B, A  
 MOV C, A ] → 3 Byte → More efficient

SUB A, <sup>Accumulator</sup>  
 A ← A - A

SUB C  
 A ← A - C

(d) LXI B, 0000H  
 Mov A, B ] → 4 Byte

↳ 3) Direct Addressing Mode :-

→ address is given in an instruction

Ex- LDA 2000H;  
 ↓  
 Load  
 Accumulator

Ex

ST A 2000H ;

→ Memory write

↓  
store

data from A to 2000H  
(Memory location)

A → [2000H]

Ex

address

2000H

A  
+  
B = A

3000H

MVI B, 02H

→ 2 Byte

LDA 2000H ;

→ 3 Byte

ADD B ;

→ 1 Byte

STA 3000H ;

→ 3 Byte

9 ByteEx

2000H

A  
↓  
B

3000H

+  
A = A

4000H

↪ 4) Indirect Addressing Mode :-

↪ Address is given in register pair

↓                    ↓                    ↓  
16 Bits              BC              DE              HL

1 Byte              ↗ Address  
Ex. LDAX B, 4200H      ↓      BC      Ka data Path hona chahiye isko  
Register              Pair              implement Karne se Pehle  
BC = 4200H

Content of BC is loaded into accumulator.

A ← [BC]  
↓  
Address

Ex. STAX B, 4200H

↪ Implement Karne se Pehle B aur C Ka data path hona chahiye

↪ BC register pair mein A Ka data store ho jaayega

\* Most of the Operations are performed using indirect addressing mode.

LDAX B

LXI B, 2000H  
LDAX B

→ 5) Implied Addressing Mode -

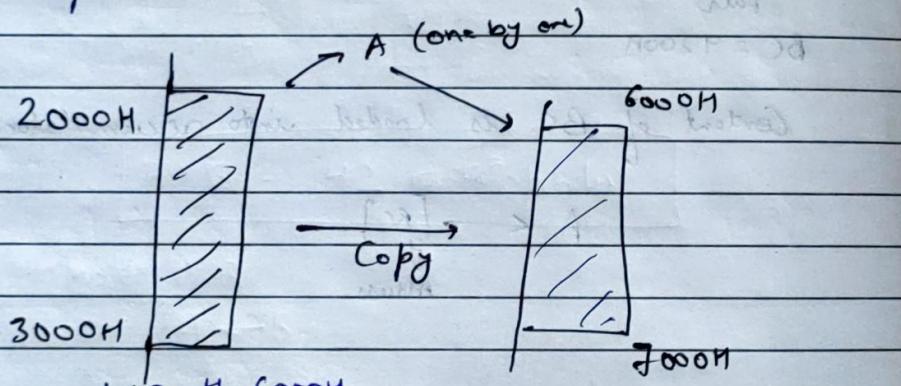
→ No operand (included in opcode)

STC → Set Carry Flag = 1

CNC → Complement Carry Flag = 1

→ No data/Address is specified / operand is not specified.

Ex



INX B

INX H

Increment

Now increment B<sup>th</sup>

2001H

6001H

↓ B

↓ A

↓ H

↓ A

## # Instruction Set of 8085 HP :-

Do not affect any  
Flag

8 bits

$2^8 \rightarrow \text{inst}^n$

Categorized into 5 groups

we will study 74 inst<sup>n</sup>

- 1) Data Transfer Group inst<sup>n</sup>: - This group will deal with the transfer of data from one place to another place.

MVI R, 35H

MVI R, 8 bits data  $\leftrightarrow$  2 Bytes inst<sup>n</sup>

Move whenever I is there then next given is data (not address).

8085HP

A

B

C

D

E

H

L

General Purpose Registers

Ex- MVI A, 32H

A  $\leftarrow$  32H

2) LX(I) Rp , 16-bits data

Rp  $\rightarrow$  Register Pair

load

BC DE HL

Ex- LX8 B, 2000H  $\rightarrow$  3 Bytes

BC  $\leftarrow$  2000H

Same inst<sup>n</sup> can be done by -

MVI B, 20H } 14 bytes  
MVI C, 00H }

Note :- MOV R<sub>i</sub>, R<sub>j</sub>

→ It can only transfer from register to register

3) MOV R<sub>d</sub>, R<sub>s</sub>

↓      ↗ Source reg.  
Destination reg.

$$7 \times 6 = 42 \text{ opcodes}$$

Ex. MOV A, B

A ← B  
25H      25H

4) MVI M, data

↓      → 2 Byte

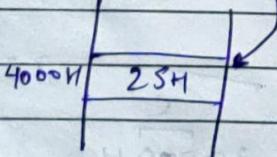
Memory location is given by HL reg. pair

Ex. MVI M, 25H

[HL] ← 25H

HP

25H	←
H	2
40	00H

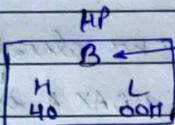


\* In this M is the memory location whose address is specified in H, L register pair.

(1 opcode)

5)  $\text{MOV } R, M$  → 1 Byte  
 ← (HL) Rp      Indirect Addressing mode

Ex.  $\text{MOV } B, M$



4000H 25H

\* (5 opcodes) are possible

6)  $\text{MOV } M, R$  → Register Addressing mode  
 ← (R)

(5 opcodes) are possible

7)  $\text{LDA } 2000H$  → Direct Addressing mode  
 ↓  
 Address

$A \leftarrow [2000H]$

8) STA 3000H → Direct Addressing mode  
 ↓  
 Address

$$A \rightarrow [3000H]$$

9) LDA X B ; 1 opode

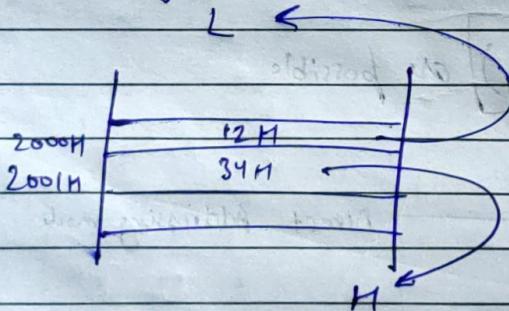
Both are same  
 ↗ LDA X H ;

MOV A, M → This one we already studied  
 therefore LDA X H is not used.

10) STAX D ;

High Byte → Lower Byte  
 11) LHLD 2000H  
 ↓  
 Address

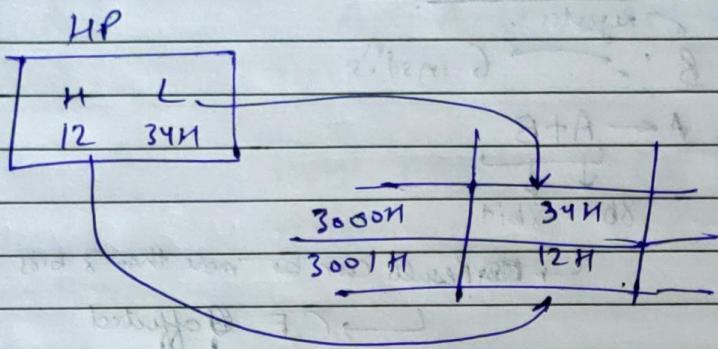
Directly  
 Address is given  
 load H & L KO 2000H wala location ke data se  
 but humein direct data nahi de rakha. Toh 2000H  
 wala L mein jayega aur H mein 2001H wala  
 jayega.



$$\begin{aligned} L &\leftarrow [2000H] \\ H &\leftarrow [2001H] \end{aligned}$$

$$[Address] \rightarrow A$$

12) SHLD 3000H



13)  $PC \leftarrow HL \Rightarrow$  Register Addressing Mode

$$PC \leftarrow HL$$

↓ 16 Bit    ↓ 16 Bit

HL  
4000H  
PCHL  
↓  
H000H

States the address of next inst<sup>n</sup>

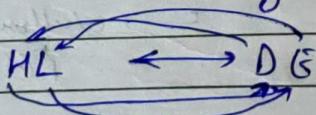
Stack Pointer  
SP gives the address of top of the stack

13) ~~SPHL~~

$SP \leftarrow HL$

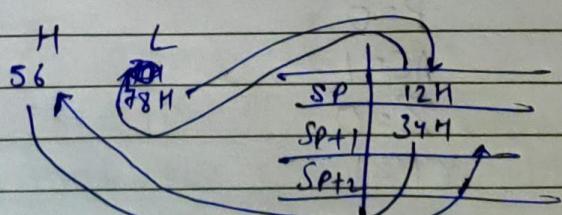
Arrangement of memory location

14) XCHG exchange.



15) XTHL

↓  
Exchanging with Stack Pointer



Flag may be affected

## # Arithmetic instruction set :-

→ Register.

1) ADD B :- → 6 instns  
 ↓  
 1 byte      A  $\leftarrow A + B$   
 ↓            8bit    8bit

→ Result can be more than 8 bits

→ CF affected  
 ↓  
 carry

2) ADD M :- → Direct Addressing mode  
 ↓  
 Memory location  
 1 byte      A  $\leftarrow A + [M]$

→ Carry Flag may be affected

3) ADD 25H :- → Data  
 ↓  
 Immediate 2 bytes      A  $\leftarrow A + 25H$   
 A.M.

Note:- FFH

+ FFH  
 ① FEH  
 CP      A

1111 1111  
 ① 1111 1110

→ overflow flag is set  
 FE display flag

Sign Flag = 1

Seeing in PE

P=0

AC=1

Z=0

CF=1

4) SUB B  
 $A \leftarrow A - B$

5) SUB M  
 $A \leftarrow A - [M]$

6) SUB 25H  
 $A \leftarrow A - 25H$

All Flags can be affected

Borrow Flag -

$$\begin{array}{r}
 \text{A} \xrightarrow{\text{2's complement}} \\
 + (-B) \xrightarrow{\text{2's complement}} \\
 \hline
 \text{Result A} \xrightarrow{\text{2's complement}}
 \end{array}$$

Ex  

$$\begin{array}{r}
 23H \\
 -15H \\
 \hline
 \end{array}$$

08H

aana chahiye

$$\begin{array}{r}
 0010\ 0011 \xrightarrow{\text{1's complement}} 1101\ 1100 \\
 - 0001\ 0101 \xrightarrow{\text{2's complement}} 1110\ 1010 \\
 \hline
 1101\ 1101 \\
 \hline
 1110\ 1011 \\
 \hline
 \textcircled{1} \ 1100\ 1000 \\
 \hline
 \Rightarrow \textcircled{C} \ 08H
 \end{array}$$

$+23H \rightarrow (35)_{10} \Rightarrow 0010\ 0011 \xrightarrow{\text{SMR}}$

$-15H \rightarrow -(21)_{10} \Rightarrow 0001\ 0101 \xrightarrow{\text{2's}}$

$$\begin{array}{r}
 0010\ 0011 \\
 1110\ 1011 \\
 \hline
 \textcircled{1} \ 0000\ 1110 \\
 \hline
 0EH
 \end{array}$$

but yeh aaya

0111 1100  
0111 1101

PAGE NO.	/ /
DATE	/ /

$$64 + 32 + 16 + 8 + 4 + 1$$

Ex

$$-23H \rightarrow 0010\ 0011$$

$$-31H$$

$$-54H$$

aaaa

change

$$1101\ 1100$$

+1

$$\underline{1101\ 1101}$$

$$\text{cf} \quad \begin{array}{r} 1100\ 1111 \\ 1010\ 1100 \end{array}$$

↓ 2's complement

$$0101\ 0100$$

$$\Leftrightarrow 54$$

Ex

$$0011 + 3 = 0011$$

$$0101 - 3 = 1010\ 1000$$

SMR  $\rightarrow$  1011

$\boxed{1011} \rightarrow 1101$

2's

Carry flag higher nibble se check karte hai  
A.C. lower nibble  
So check karte hai

17

$$\begin{array}{r} 23H \\ + 31H \\ \hline 54H \end{array}$$

$$\begin{array}{r} 0010\ 0011 \\ 0011\ 0001 \\ \hline 0101\ 0100 \end{array}$$

$$\downarrow$$

$$54H$$

Carry Flag = 0

P=0

AC=0

Z=0

S=0

Q2)

$$\begin{array}{r} 37H \\ + 29H \\ \hline 60H \end{array}$$

$$1100\ 0100$$

$$1010\ 1111$$

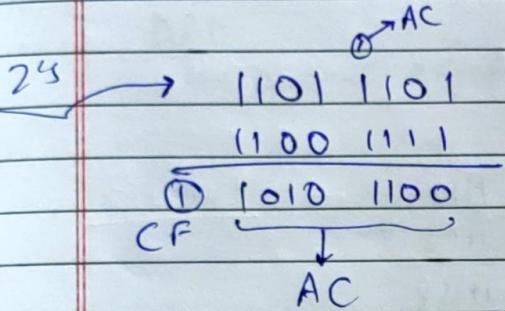
$$0111\ 0000$$

37

$$-23H$$

$$-31H$$

$$\underline{-54H}$$



$$S = 1$$

$$Z = 0$$

$$AC > 1$$

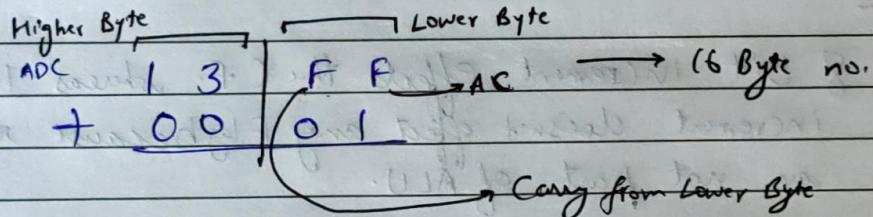
$$(F_2)$$

$$P = 1$$

7) ADC R  
Add with carry

$$A \leftarrow A + R + CF$$

\* It is used to add two 16 Bit numbers.



8) ADC M → 1 byte

$$A \leftarrow A + [M] + CF$$

9) ACI 25H

$$A \leftarrow A + 25H + CF$$

→ 2 Byte inst?

10) SBB M

$$A \leftarrow A - [M] - CF$$



→ Subtract with Borrow      → Carry in 2's complement system

11)

SBI 25H

$$A \leftarrow A - 25H - CF$$

12)

SBB R

$$A \leftarrow A - R - CF$$

13)

INR R → 8bit

$$R \leftarrow R + 1$$

14)

INX Rp → Register Pair  
→ 16Bit

$$Rp \leftarrow Rp + 1 + 8 - A \rightarrow A$$

Note -

\* 8 bit increment effects the C.R whereas 16 Bit increment doesn't effect any flag because registers are not part of ALU.

15)

INR M

↓ Decrement

16)

DCR R

17)

DCX Rp

18)

DCR M

↓ Double Addition

19)

DAD D

$$HL \leftarrow HL + DE$$

HL is acting as a accumulator

DAD B

$$HL \leftarrow HL + BC$$

$\rightarrow$  Multiplication by factor 2

DAD H

$HL \leftarrow HL + HL$

20) DAA

$\hookrightarrow$  Decimal Adjust after addition

$$\begin{array}{r}
 \textcircled{1} \\
 \begin{array}{r}
 49H \\
 + 39H \\
 \hline
 82H
 \end{array}
 \quad \begin{array}{r}
 45 \\
 + 45 \\
 \hline
 8A
 \end{array}
 \end{array}$$

(i)  $24H$

$$\begin{array}{r}
 25H \\
 - 49H \\
 \hline
 \end{array}$$

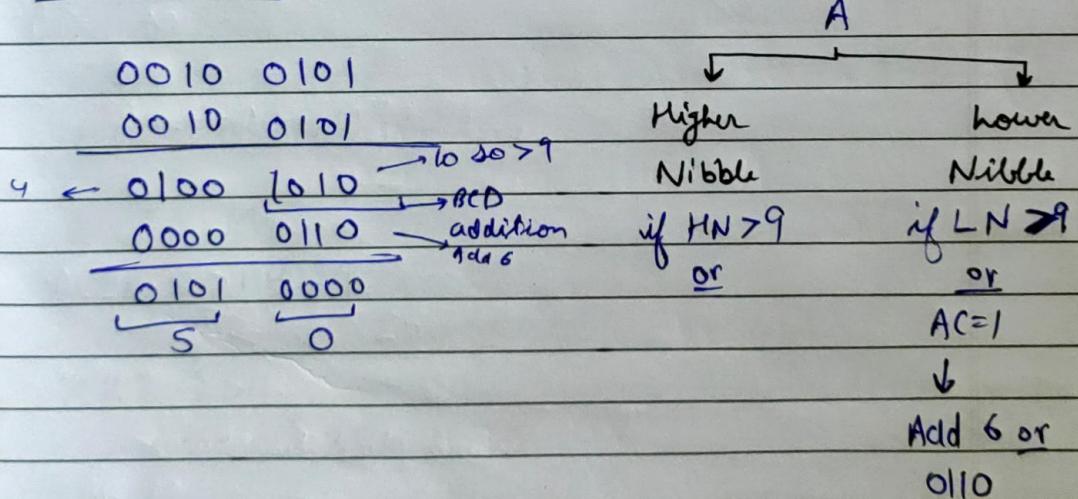
(ii)  $2.5$

$$\begin{array}{r}
 25 \\
 \hline
 50 \\
 \hline
 4A
 \end{array}$$

in BCD we have to  
add 6

in BCD  $\rightarrow 0-9 \rightarrow^{10} \rightarrow$  Add 6

in Hexadecimal  $\rightarrow 0-15 \rightarrow^{16}$



# logical Instructions :-

1) AND  
 $A \leftarrow A \wedge R$  → register

2) ANA M → Direct  
 $A \leftarrow A \wedge [M]$

3) ANI 25H → Immediate  
 $A \leftarrow A \wedge 25H$

4) ORA R  
 $A \leftarrow A \vee R$

5) ORA M  
 $A \leftarrow A \vee [M]$

6) ORI 25H  
 $A \leftarrow A \vee 25H$

7) XRA R  
 $A \leftarrow A \oplus R$

8) XRA M  
 $A \leftarrow A \oplus [M]$

9) XRI 25H  
 $A \leftarrow A \oplus 25H$

Assume  $A = 35H$

1)

Clear the lower nibble of register A

$$A = 35H = \begin{array}{c} HN \\ 0011 \end{array}, \begin{array}{c} LN \\ 1010 \end{array}$$

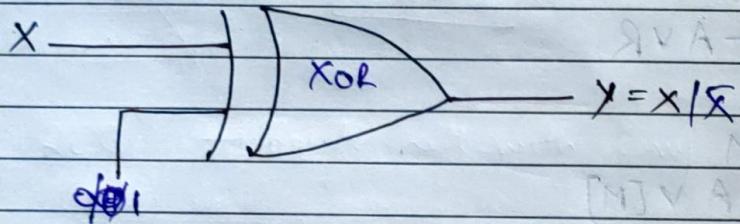
$1111, 0000 \rightarrow \text{AND}$

$$\text{Required} = 00110000$$

$\hookrightarrow 30H$

$\text{ANI } F, 0H$

$$\begin{array}{l} X \oplus X = 0 \\ X \oplus \bar{X} = 1 \end{array}$$



2)

Clear the upper nibble

$$A = 35H = \begin{array}{c} HN, LN \\ 00111010 \end{array}$$

$$\text{Required} = \underline{\underline{00001010}}$$

$\text{ANI } 0F H \quad \text{or}$   
 $XRI = 30H$

37 Set the lower nibble

ORI OF H

XRI OA H

47 Complement the lower nibble

~~ADD SUB~~

XRD OF H

~~XRD~~

10)  $\xrightarrow{\text{Comparison b/w A \& R}}$  CMP R  $\xrightarrow{\text{Here we only talk about flags not the result but in subtraction we talk about result also.}}$

Ex: CMP B  
 $(A - B)$

$A > B$       ↓       $A = B$        $A < B$

	CY	Z
$A > B$	0	0
$A = B$	0	1
$A < B$	1	0

11) CMP M

$A - [M]$

12) CPI 25H

$A - 25H$

$\xrightarrow{\text{Set the Carry Flag}}$

13) STC

$CY = 1$

14) CMC

$\xrightarrow{\text{Complementing the Carry Flag}}$

$CY = 0$

$CY \rightarrow CY$

15)  $\text{CMA}$   $\xrightarrow{\text{Complement}}$   
 $A \rightarrow \bar{A}$

$\text{NAND} \rightarrow$   $\boxed{\text{ANAR}}$   $\xrightarrow{\text{AND & store in A}}$   
 $\text{CMA} \xrightarrow{\text{Complement A}}$

~~XOR~~  $\rightarrow$   
~~A  $\oplus$  B~~  
~~NOR~~  $\rightarrow$   $\boxed{\text{ORA R}}$   
 $\text{CMA}$

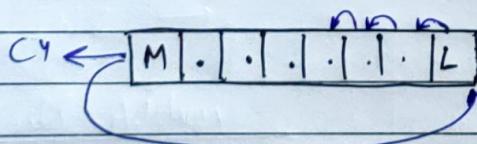
16) RLC  $\rightarrow$  Rotate left with Carry

17) RRC  $\rightarrow$  Rotate Right with Carry

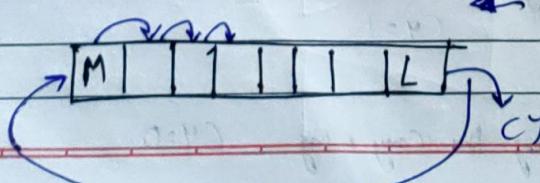
18) RAL  $\rightarrow$  Rotate Arithmetic Left

19) RAR  $\rightarrow$  Rotate Arithmetic Right

Ex. RLC :-

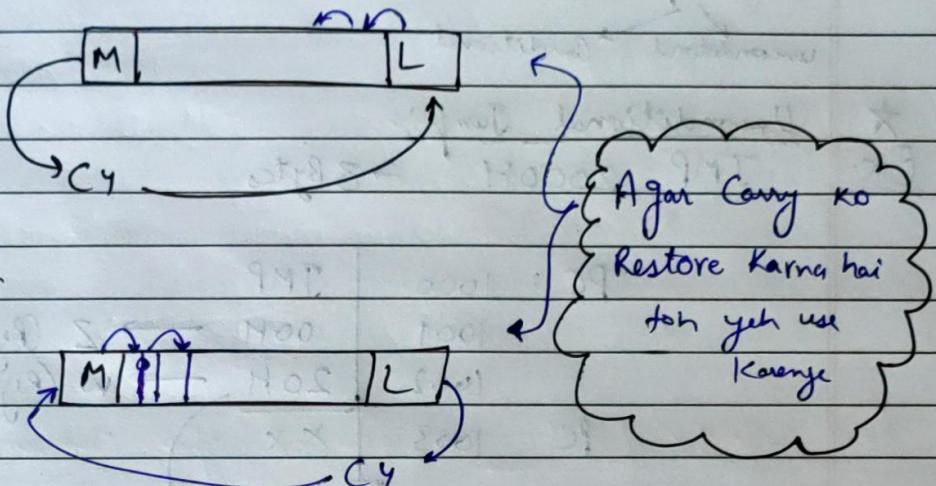


Ex. RRC :-

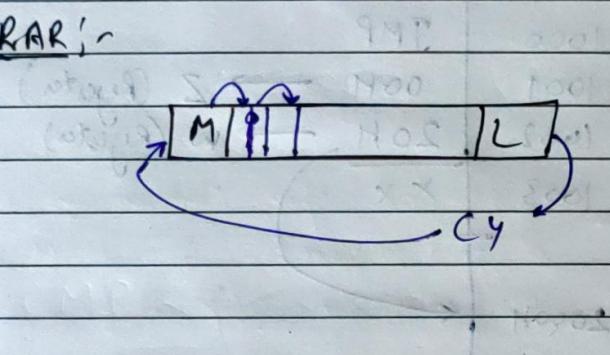


Here Carry doesn't participate in rotation

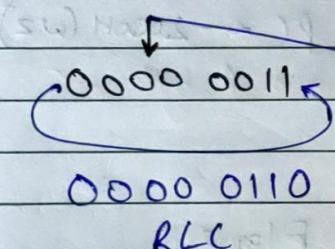
Ex. RAL :- Data can be changed in this



Ex. RAR :-



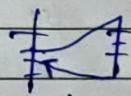
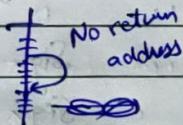
Ex.



If I want to determine this bit then I need to rotate it.

Note :

Jump Ins<sup>n</sup>



(Call ~~Reg~~) Ins<sup>n</sup> → address return hogा

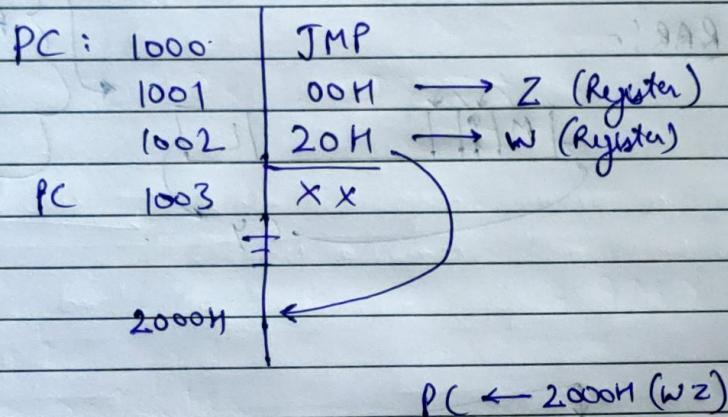
#

## Jump & Call Instruction :-

unconditional → Conditional

### ★ Unconditional Jump :-

Ex: JMP 2000H → 3 Bytes



★

### Conditional Jump :-

JC → Jump when Carry Flag = 1

JNC → " " " " " = 0

JM → " " Sign " " = 1

JP → " " Sign " " = 0

JPE → " " Parity Flag = 1 (Even)

JPO → " " Parity Flag = 0 (Odd)

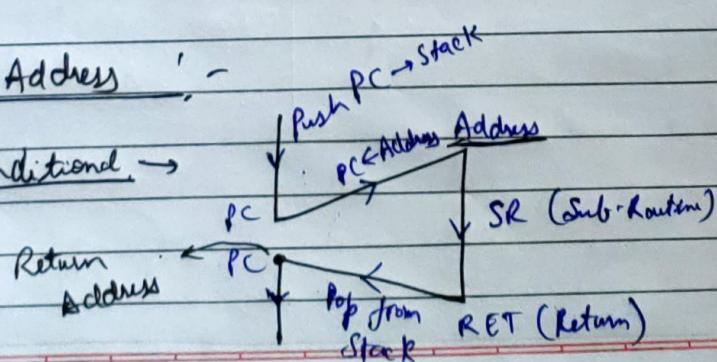
JZ → " " Zero Flag = 1

JNZ → " " " " " = 0

3)

### Call Address :-

(Unconditional) →



\* Stack pointer points to the top of the stack.

\* Conditional :-

CC → Call when CF = 1

CNC → ., CF = 0

CM →

CF →

~~CPF~~ →

CPO →

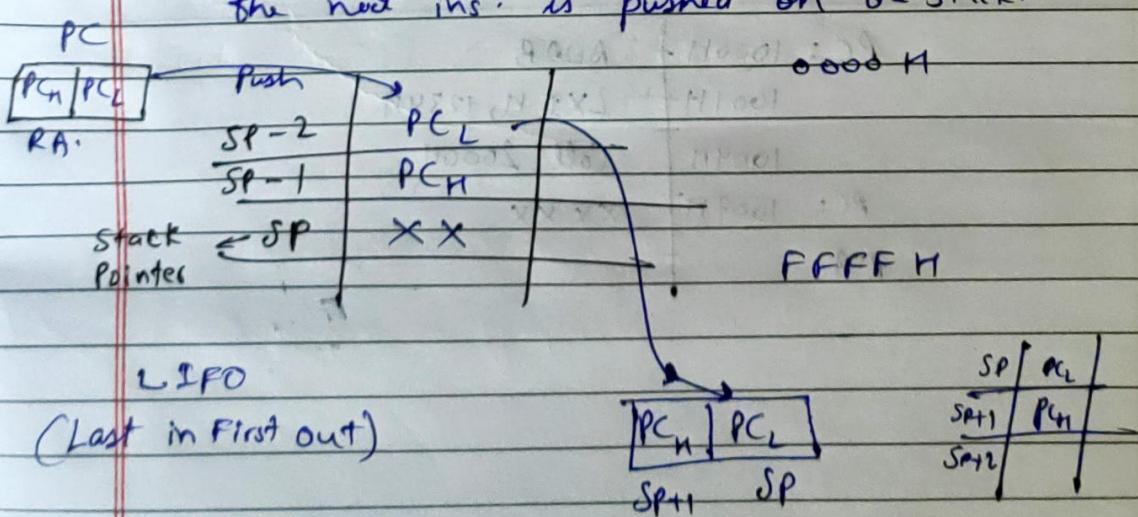
CZ →

CNZ →

\* Unconditional Call :-

No Flags are effected after the execution of Jump & Call ins.

\* Call Address :- A program sequence is transferred to the address specified by the operand. Before the transfer the address of the next ins. is pushed on the stack.



\*  $\overset{\text{Reset}}{\text{RST}_n} \rightarrow \text{Software Interrupts (8 SIs)}$

$(n \times 8)$

$\text{RST}_0 \quad 0 \times 8 = 0000H$

$\text{RST}_1 \quad 1 \times 8 = 0008H$

$\text{RST}_2 \quad 2 \times 8 = 0010H$

$\text{RST}_3 \quad 3 \times 8 = 0018H$

$\text{RST}_4 \quad 4 \times 8 = 0020H$

Real Life Ex: of Call  $\rightarrow$  when we open Camera in Instagram  
it goes to camera then takes us  
back to Instagram app.

\* Call has 3 byte, Reset has 1 byte in?

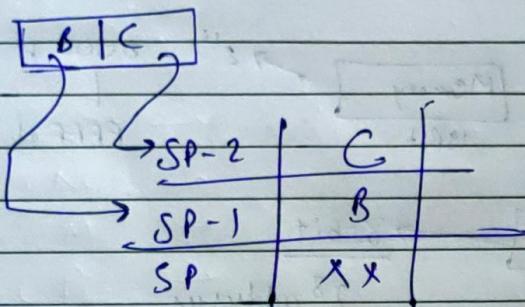
Note:- There are 5 Hardware Interrupts.

\* Real life Ex. of Infinite Loop - Digital watch  
(Runs until battery is charged)

PC: 1000H	ADD B
1001H	LXI H, 1234H
1004H	Call 2000H
PC: 1007H	XX XX
R.A.	

1) Push R<sub>p</sub>

Push B



Steps

1)  $SP \leftarrow SP - 1$

2)  $[SP] \leftarrow B$

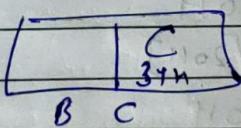
2) Pop R<sub>p</sub>

3)  $SP \leftarrow SP + 1$

4)  $[SP] \leftarrow C$

Ex- Pop B

SP	34H
SP+1	12H
SP+2	XX



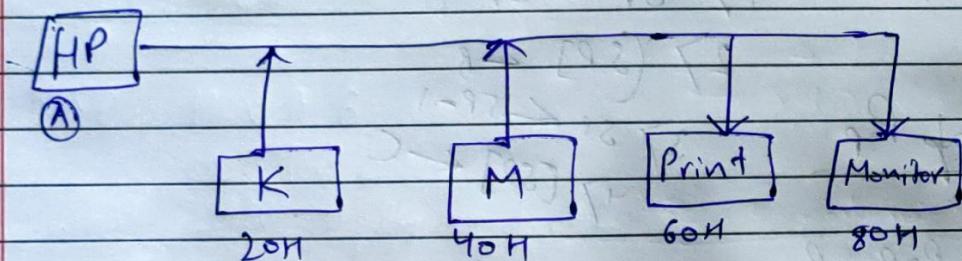
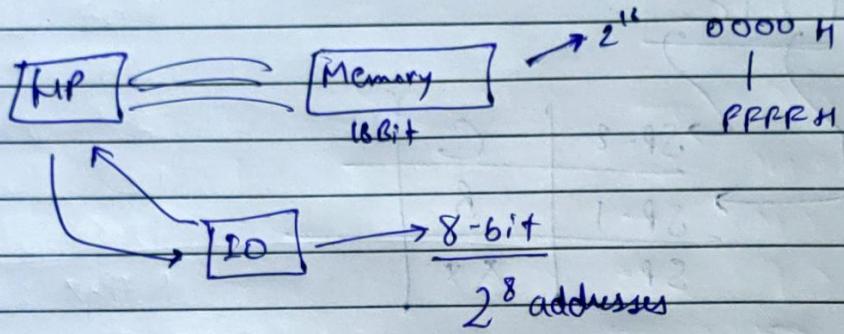
1)  $C \leftarrow [SP]$

2)  $SP \leftarrow SP + 1$

3)  $B \leftarrow [SP]$

4)  $SP \leftarrow SP + 1$

IO  $\longleftrightarrow$  IN address  
OUT address



HP se joh connected nota hai, uska ek Port Address nota hai.

IN 20H  
 $A \leftarrow [20H]$

OUT 60H  
 $A \rightarrow [60H]$

Ex:  $20H = 16$

IN 20H

(Input from keyboard)

OUT 80H

(Display on monitor ~~monitor~~)

IN 20H

$A \leftarrow$

- 1) SIM Set Interrupt Mask  
→ Ajga yaad nahi
- 2) RIM Read " " register mein kya value daadi:
- 3) EI Enable Interrupt
- 4) DI Disabled Interrupt
- 5) NOP No operation → used to provide the delays  
→ Like
- 6) HLT Hold → iske baad HP Kacche Karte band  
kar dega.

Q.

LXI H, 2000H

LDA 2002H

XRA M

HL = 2000H

MOV E,A

A = 02H

MVI D, 20H

A = 02H

LDAX D

E = 02H

OUT 01H

D = 20H

HLT → Address of Port

DE = 2002H

A = 02H

Memory

Content

2000H

00H

2002H

01H

2001H

02H

2003H

03H

(N)

O:- MVI A, Byte 1

MOV B,A

SUI 50H

JC DEL

MOV A,B

SUI 80H

JC DISP

~~DEL: XRA A~~

~~OUT Port 1~~

HLT

DISP: MOV A,B

OUT Port 2

what will display on port 2?

A=N

B=N

A  $\leftarrow$  N - 50

N  $\geq$  50

CY=0

H00 = A

H50H = ?

H80 = A

N < 80H  
CY=1  $\rightarrow$  DISP  
A = N  
Port2  $\rightarrow$  N

A = 00H  
Port1: 00H

Now, JC wala Jump karwa deya

A=N

~~H50H~~

N < 50H  $\rightarrow$  Port1: 00H

N < 80H  $\rightarrow$  Port2: N

N  $\geq$  50