

## Unit - 3

8085 Instruction Set

Name : Trivedi Human A

No : 196170307149

Class : 3 - B

Ques. 1.) Give classification of 8085 instruction based on operation,

An instruction is a command to microprocessor to perform a particular task on specified data.

→ There are five group

- 1.) Arithmetic
- 2.) Logical
- 3.) Data transfer
- 4.) Branch
- 5.) stack I/O & machine control group

1.) Arithmetic :-

This group perform various arithmetic operation like addition, subtraction, increment & decrement

2.) Logical :-

This group perform logical operation like AND, OR, NOT.

3.) Branch :-

This includes various conditional and unconditional jump, restart and return

4.) Stack, I/O and machine control:

This group of instruction includes instruction for push and pop.

5.) Data transfer:

This moves data between register or memory location.

2.) Explain LXI, LDA, STA, XCHG, LHLD, GHLD:

⇒ LXI :-

LXI RP, data16

This instruction is used to transfer second byte instruction in lower order address and third byte instruction in higher order address.

Ex

LXI H, 2050h      [L] → 50  
                        [H] → 20

Here 50 will store in register L and 20 will store in register H.

- > It is a 3 byte instruction with immediate addressing mode.
- > Here 1 byte is for Opcode [CXI] and 2 byte for the data16 which is used to store in register pair given in instruction.

## 2.) LDA :-

LDA add

$A \leftarrow [add]$

This instruction is used to transfer the data of add given in the instruction into the accumulator.

- > It is a 3 byte instruction with direct addressing mode.

~~LDA~~ LDA 2030H

So Here, content of memory location 2030 is given to the accumulator.

3) STA :-

$[addr] \leftarrow A$

STA  $\rightarrow$  address

This instruction is used to store the content of accumulator to the location specified in the instruction.

⇒ STA  $\rightarrow$  8030h

So here, value of accumulator will be given to the address 8030h.

It is also of 3 byte instruction with direct addressing mode.

4) XCHG :-

$$\begin{array}{l} [H] \leftrightarrow [D] \\ [L] \leftrightarrow [E] \end{array}$$

This instruction is used to exchange the content of HL with DE register pair.

It is 1 byte instruction with implied addressing mode.

a) LHLD

LHLD add

$$\begin{array}{l} [L] \leftarrow [add] \\ [H] \leftarrow [add+1] \end{array}$$

This instruction is used to transfer the content of given memory location in L and next location in H.

b) LHLD .wdata

Here, data of zero will store in L register and content of .wdata will store in register H.

c) SHLD

SHLD add

$$\begin{array}{l} [add] \leftarrow [L] \\ [add+1] \leftarrow [H] \end{array}$$

This is used to transfer the content of register L to the location specified in instruction and H to the next location of specified location.

SHLD 2050h

so, data of L is stored in 2050.  
location and data of H will  
store in 2051.

3.) Explain ADI, SUB, DCR, INX,  
DAA, ADC.

1.) ADI

ADI data

$$[A] \leftarrow [A] + \text{data}$$

This is used to add 8bit data with the content of accumulator and stores the result in accumulator.

> It is a 2 byte instruction with immediate addressing mode.

ADI 04h

if A has 03 in it then after the above instruction accumulator will have 07h.

## 2.) SUB :-

SUB R/M

$$[A] \leftarrow [A] - [R]$$

$$[A] \leftarrow [A] - [HL] \quad \text{or}$$

This is used to subtract the content of Register or memory specified in the instruction from the accumulator and stores the result in it.

- > It is of byte instruction if the M is in its as its operand else it is of 1 byte.
- > If the register is as operand then it is of register addressing mode else it is of indirect mode.

## 3.) DCR :-

DCR R/M

$$[R] \leftarrow [R] - 1$$

$$[HL] \leftarrow [HL] - 1$$

This is used to decrement the content of Register / memory.

e.g. DCR B

Then it will decrement the value of B by 1.

DCR M

Then it will decrement the value of HL register pair by 1.

4) INX :-

INX Rp

$[Rp] \leftarrow [Rp + 1]$

4)

It is used to increment 16bit of register pair given in the instruction.

④

INX H

Then it will increment HL by 1.

## 5.) DAA : [Decimal adjust Accumulator]

This instruction adjusts the content of the accumulator to form two 4 bit binary-coded decimal digits.

- > If the value of lower 4 bits of the accumulator is greater than 9 or if the AC is set, 6 is added to the
- > If the value of higher 4 bits of the accumulator is greater than 9 or the CY is set, 6 is added to the higher 4 bits of the accumulator.

## 4.) Explain logical AND operation:-

AND instructions are used to perform bitwise ANDING of two numbers.

- > The result of ANDING instruction is stored into accumulator.

### 1.) ANA :-

$$\begin{aligned} [A] &\leftarrow [A] \wedge [R] \\ [A] &\leftarrow [A] \wedge [HL] \end{aligned}$$

After this instruction's execution,  
 $AC = 1$  and  $cy = 0$  and other  
flag change as per answers.

Eg:

ANA B

Here Anding will performe between  
ac content of accumulators and  
register B.

2) ANI

ANI address  
[A]

$[A] \leftarrow [A] \text{ AND } [\text{data}]$

After

> Here anding will occur  
between content of accumula-  
tor & bit immediate data and  
result is stored in 8 bit A.

5) Explain OR Operation.

OR instruction are used to  
performe OR operation of two  
nos.

> The result is stored in  
accumulator.

## 1.) ORA

ORA R/M

$$[A] \leftarrow [A] \vee R$$

or

$$[A] \leftarrow [A] \vee [UL]$$

This instruction performs oring between accumulator or memory or register's value.

## OR A B

so here Oring of B and A will result and stores in A.

After this execution value of cy and AC will cleared [0].

## 2.) ORI :-

ORI data

$$[A] \leftarrow [A] \vee data$$

## ORI 04

so If accumulator has 03 and after

DATE \_\_\_\_\_  
प्राचीनी

this instruction accumulator will have  
07h.

### 6) Explain logical XOR.

- > XOR instructions are used to perform XOR operation of register/memory with accumulator.
- > The result is stored in accumulator.
- > After the execution of this instruction CY and AC flag cleared.

#### 7.) XRA :-

XRA R/M

$$\begin{aligned}[A] &\leftarrow [A] \oplus [R] \\ [A] &\leftarrow [A] \oplus [HL]\end{aligned}$$

XRA instruction is used to perform XOR operation of register/memory with content of accumulator.

2.) XRI :-

XRI data8

$[A] \leftarrow [A] \oplus \text{data8}$

> This is used to perform XOR operation immediately with content of accumulator.

7.) Explain compare instruction. Give status of CY and Z flag.

> Compare instruction is used to find relations equal, greater than or less than.

> The result can be shown from CY and Z flag.

> They are very useful in decision making situations.

8.) CMP :-

CMP R/M

$CMP \leftarrow [A] - [R]$

or

$CMP \leftarrow [A] - [HL]$

DATE \_\_\_\_\_  
 PAGE \_\_\_\_\_  
 शिवानी

> This instruction subtract content of register/memory location with the content of accumulator, content of accumulator remains unchanged but conditional flag change accordingly.

Instruction	Flag		Relation
CMP R	Y	Z	$[A] = [R]$
	0	1	$[A] < [R]$
	1	0	$[A] > [R]$
	0	0	
CMP M	0	1	$[A] = M$
	1	0	$[A] < M$
	0	0	$[A] > M$

2) CPI :-

CPI dates

$$CPI \rightarrow [A] - \text{dates}$$

CPI instruction is used to subtract the content of 8 bit date given in second byte with content of accumulator.

- > Content of accumulators remains unchanged but conditional accordingly.

Instruction	Flag	Relation
-------------	------	----------

	$y$	$z$
CPI 05h	0	1
	1	0
	0	0

$A = \text{data}$

$A < \text{data}$

$A > \text{data}$

- R] a) Explain unconditional flag.

It is very similar to goto statement in higher programming language.

- > It transfers control to an instruction labeled with label used in jump instruction.

- > It does not affect any flag.

- ,) Forward Flag:-

JMP Next

Next : ADD B

2.) Backward Jump:

NEXT: ADD B

- - -  
- - -  
JMP NEXT  
- - -  
- - -

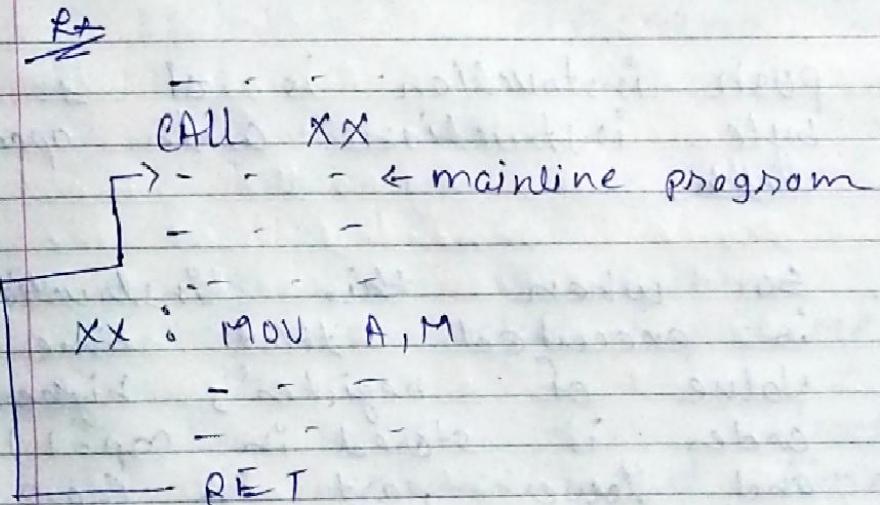
9.) Explain CALL and RET.

1.) CALL :-

- > This instruction stores the content of program counter on the stack and decrement stack pointer by 2.
- > Then it jump to subroutine unconditionally where address is given in second and third byte of instruction.
- > Addressing mode is immediate
- > It does not affect on conditional flag.

## 2.) RET :-

- > This instruction copies two byte from top of the stack into program counter and increment stack pointer by 2.
- > As it copies the address of mainline instructions after call instruction from stack into pc, the control unconditionally return back to mainline.
- > The addressing mode is indirect.



10) Explain push and pop operation.

> PUSH Rp :-

$$\begin{aligned} [Sp-1] &\leftarrow [Rh] \\ [Sp-2] &\leftarrow [RL] \\ Sp &\leftarrow Sp - 2 \end{aligned}$$

> push Here, in push instruction register pair is used as operand.

- > So it is of indirect addressing mode.
- > push instruction is of 2 byte instruction with opcode fetch.
- > So when this instruction is executed that time value of register's higher order is stored in  $(Sp-1)$  and lower part in  $(Sp-2)$ .
- > And Sp is set to  $(Sp-2)$ .
- > It has 3 machine cycle  $Op + MW + MW$  with 12 instruction cycle.

Q

PUSH B

$$B = 20$$

$$C = 30$$

So, On  $(SP)$  stack pointer value of  $C$  which is 30 is stored and  $(SP - 1)$  has 20.

> POP RP :-

$$\begin{aligned} [RL] &\leftarrow (SP) \\ [RH] &\leftarrow (SP+1) \\ SP &\leftarrow SP+2 \end{aligned}$$

> So Here, RP is used as its operand. It is also of indirect addressing mode.

> It is also of 1 byte instruction of opcode fetch.

> It has 3 machine cycle of 12 tstate  $[Op + M8 + M12]$

> So when this instruction is executed that time value of register will take value from SP.

> Value of SP will store into lower order of RP and  $(SP+1)$ 's with a value stored into higher

order.

- > And SP will have address. ~~be at~~ SP+2

Ex. POP B

SP	20
	30

B will have 30 and C will take 20 and SP is set to SP+2.

11) Explain IN and OUT instruction

13) IN part :-

(A) ~~L~~ data 8

- > This instruction moves the 8-bit data placed on the data bus by the input device connected at the port address specified by the second byte of the instruction to the accumulator.

- > It does not affect the flag and uses direct addressing mode.

Ex. IN 05h

reads to the data byte from

the input device whose port address is 05h and stores it in accumulator.

2) OUT port :-

$$[\text{port add}] \leftarrow [A]$$

- This instruction sends the contents of accumulator to the output device whose port address is specified by the second byte of the instruction.
- It does not affect the flags and uses direct addressing mode.

OUT 08h

So, port address 08 sends the content to the accumulator.

12) Explain rotate instruction:-

We can rotate bit from left to right and right to left.

### 1) RLC :-

- > This instruction rotates the content of accumulator only.

Algo :-

$$\begin{aligned} [A_{n+1}] &\leftarrow [A_n] \\ [A_0] &\leftarrow [A_7] \\ [cy] &\leftarrow [A_2] \end{aligned}$$

- > This rotates the content of accumulator on position left.
- > The MSB of the accumulator is transferred into the cy flag and LSB.

### 2) RRC

Algo :-

$$\begin{aligned} [A_n] &\leftarrow [A_{n+1}] \\ [A_7] &\leftarrow [A_0] \\ [cy] &\leftarrow [A_0] \end{aligned}$$

- > This instruction rotates the content of accumulator one position right.
- > The LSB of the accumulator is transfer into both MSB and cy flag.

## 3.) RAL :-

Algo :-  $[A_{n+1}] \leftarrow [A_n]$   
 $[cy] \leftarrow [A_2]$   
 $[A_0] \leftarrow [cy]$

- > This rotates the content of accumulator to one position left through cy flag.
- > The MSB of the accumulator transfer into cy flag into LSB of accumulator.

## 4.)RAR :-

Algo :-  $[A_n] \leftarrow [A_{n+1}]$   
 $[cy] \leftarrow [A_0]$   
 $[A_2] \leftarrow [cy]$

- > This rotates content of accumulator to one position right through cy flag.
- > The LSB of accumulator is transferred into cy flag and cy flag into MSB of accumulator.

Q3) Explain 8085 addressing mode

There are five types of address.

- 1.) Implied
- 2.) Indirect
- 3.) Immediate
- 4.) direct
- 5.) Register

1.) Implied:-

This type of instruction does not have any operand such as register / memory.

e.g STC [set carry]  
CMC [complement carry]

2.) Register:-

In this mode, the operand of instructions are register.

for example, MOV A, B

Here A and B both are register so the addressing mode of this instruction is register.

### 3) Immediate :-

- In this mode, operand of instruction is 8 bit data.
- Store data immediately after the opcode.

e.g. MVI A, 03h

So here in this instruction, first address of opcode is stored in memory and the 03 is stored in memory.

### 4) Direct :-

- Three byte instruction contains direct addressing mode
- First byte is opcode, second is lower-bit address of 16bit, third byte is higher bit address of 16bit.

e.g. LDA 3050h

- The processor assemble second & third byte into 16bit memory address 3050h and value of processor store on accumulator.

### 5) Indirect :-

In this mode, address of operand is indirect in register.

Ex: MOV A, M

Here content of memory is transfer into accumulator where [M] stores the content of [HL] pair.