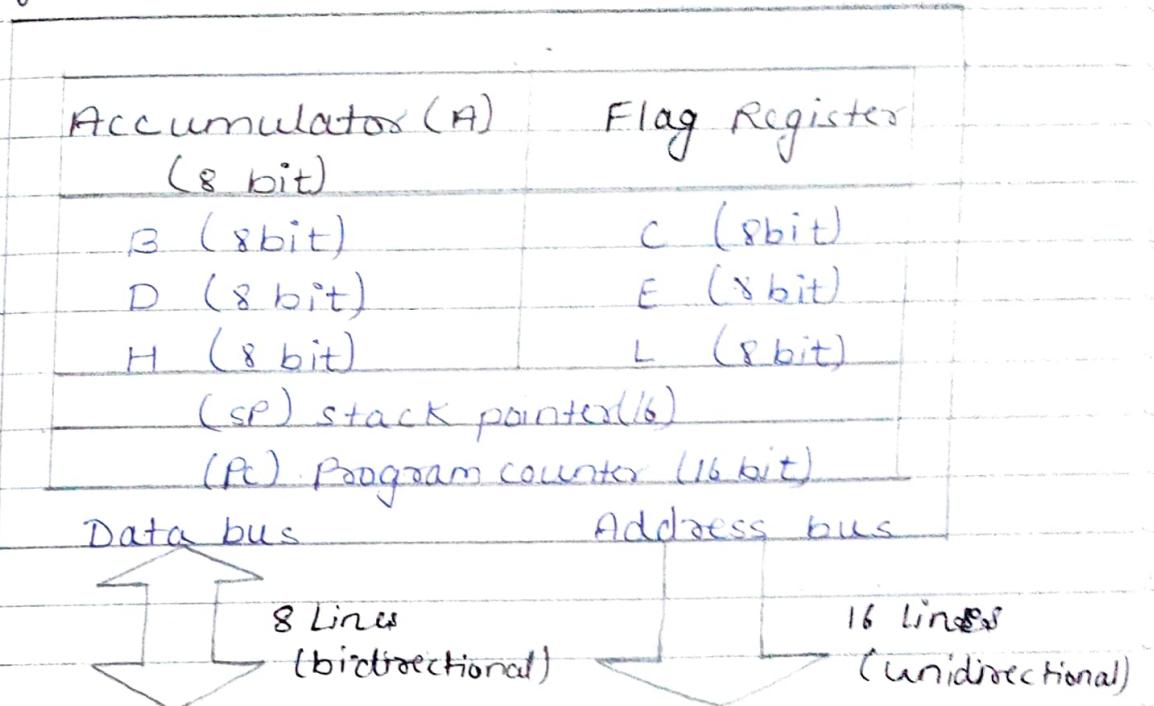


## UNIT-2

# software architecture Registers / Programming model of register / 8085-Microprocessor



→ The programming model of 8085 - microprocessor contain various 8-bit and 16-bit registers. Registers are actually a set of flip-flops. Registers are used to store data temporarily for processing. Enormous data is stored in memory, which is on a separate chip; hence more time is consumed to access data from memory by the microprocessor.

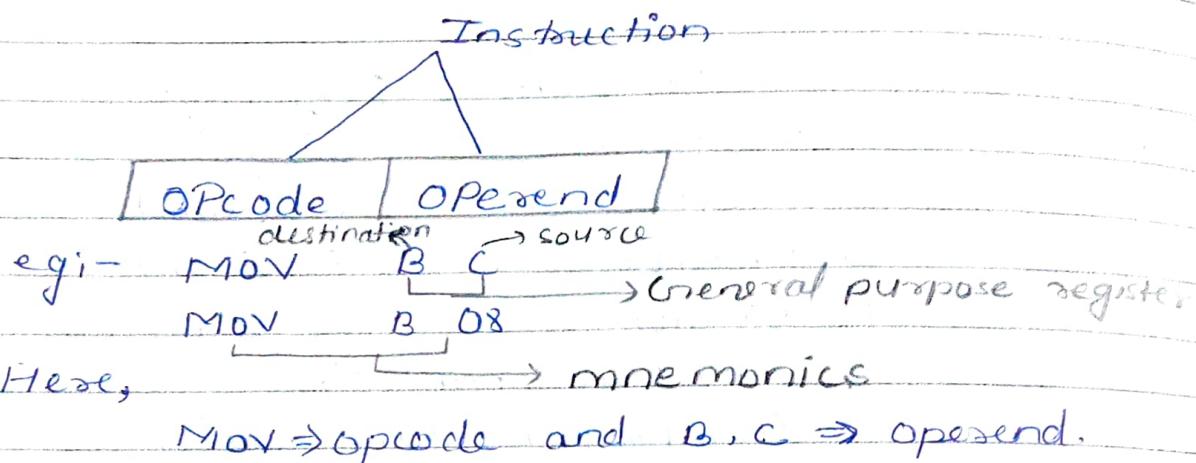
Therefore, the data which need processing are copied into registers within microprocessor to save processing time.

Type of <sup>flag</sup> register - 5

Carry flag, auxiliary carry flag, zero flag, Parity flag, sign flag.

## # Classification of Instruction set:-

### Instruction format:-



Each instruction of 8085 microprocessor has two specific information fields one is the operation code (opcode) which satisfy the operation to be performed.

- The operation is specified by binary code hence the name is operation code second is operend which say about the data on which task to be performed.

### Classification of instruction set:-

The whole instruction set can be classified into different categories on the basis of three different criteria:-

- 1) Operation performed by instruction.
- 2) Length of the instruction.
- 3) Addressing mode of the instruction.

1) Operation performed by instruction:-Data transfer graph / Instruction:-a) Mov (Move) Rd, Rs :-

Mov the copy of content of source together to destination register.

- 1 byte instruction.
- Register addressing.
- 1 machine cycle.
- 4-T state.
- No flag affected.

(b) MOV R, M :-

M = memory pointer pointing the memory location whose address is stored in HL pair.

- Move the copy content of memory location whose address is in HL pair to register R.
- 1 byte instruction.
- 2 machine cycle.
- 7-T state.
- In direct addressing mode.
- No flag is affected.

(c) MVI R, 8 bit :-

Mov immediate 8 bit data to R.

- Two byte instruction.
- 2 machine cycle.
- 7-T state
- Immediate addressing mode.
- No flag is affected.

d) MVI M , 8-bit data:-

M = memory location whose address is stored in HL pair.

Move immediately 8 bit data to M.

- 2 byte instruction.
- 3 machine cycle.
- 10-T state
- Immediate addressing mode.
- No flag is affected.

e) LXI Rp - 16-bit data :-

Move 16 bit data immediately in register pair.

- 3 byte instruction.
- 3 machine cycle.
- 10-T state.
- Immediate addressing mode.
- No flag is affected.

f) LDA , 16-bit address:-

Load the accumulator with the content of memory location whose address is given in instruction.

- 3 byte instruction.
- Direct addressing mode.
- 4 machine cycle.
- 13-T state.
- No flag is affected.

g) STA 16-bit addressing:-

Store the content of accumulator of memory location given in instruction.

- 3 byte instruction.
- 4 machine cycle
- 13-T state
- Direct addressing mode.
- No flag is affected.

h) LDAX , Rp:-

Load the accumulator with content of memory location that address is stored in Rp.

- 1 byte instruction.
- 2 machine cycle.
- Indirect addressing mode.
- No flag is affected.
- 7-T state.

i) STAX , Rp:-

Store the content of accumulator of the memory location whose address is stored in Rp.

- 1 byte instruction.
- 2 machine cycle.
- 7-T state.
- Indirect addressing mode.
- No flag is affected.

j) PCHL:-

Copy the contents of register pair to the program counter.

- 1 byte instruction.
- 1 machine cycle.

- Register addressing mode.
- 4-T state.
- No flag is affected.

K) SPHLi-

Copy the content of register pair (HL)  
to the state pointer.

- 1 byte instruction.
- 1 machine cycle.
- 4-T state.
- Register addressing mode.
- No flag is affected.

L) XCHGi-

Exchange the content of HKL with DKE.

- 1 byte instruction.
- 1 machine cycle.
- 4-T state.
- Register addressing mode.
- No flag is affected.

$$\begin{bmatrix} H \leftrightarrow D \\ L \leftrightarrow E \end{bmatrix}$$

M) XTHLi-

Exchange Hard L with top of the stack.

- 1 byte instruction.
- 5 machine cycle.
- 16-T state.
- Register addressing mode.
- No flag is affected.

$$\begin{bmatrix} H \leftrightarrow sp+1 \\ L \leftrightarrow sp \end{bmatrix}$$

L  $\leftarrow$  Address

H  $\leftarrow$  Address + 1

N) LHLD :- Addresses 16-bit

Load HL register pair from  $\text{wo}$  consecutive memory location.

- 3 byte instruction
- 5 machine cycle
- Direct addressing mode.
- No flag is affected.

L  $\rightarrow$  Address

O) SHLD :- Address - 16 bit

store content of HL register pair direct into two consecutive memory location.

- 3 byte instruction.
- 5 machine cycle.
- Direct addressing mode.
- No flag is affected.

P) IN, 8-bit Port address:-

Read 8-bit data into accumulator from 'I/O' device connected at the microprocessor.

- 2-byte instruction.
- 3 machine cycle.
- Direct addressing mode.
- No flag is affected.

Q) OUT, 8-bit Port address:-

Write 8-bit data from accumulator into I/O device that is connected to given 8-bit port address.

- 2 bit instruction.
- 3 machine cycle.
- Direct addressing mode
- No flag is affected.

## → Arithmetic Instruction:

① ADD, RI:- At the content of register to the content of accumulator and final result store in accumulator.

$$A \leftarrow A + R$$

→ 1 byte instruction.

→ Register addressing mode.

→ All flags are affected according to the result.

② ADD, MI:-

At the content of memory location pointed by the HL pair to the content of accumulator.

$$A \leftarrow M_{[HL]} + A$$

→ 1 byte instruction.

→ In direct addressing mode.

→ All flag are affected according to the result.

③ ADI, Data (8bit):-

$$A \leftarrow A + [8\text{-bit}]$$

8 bit immediate data to the accumulator and store the result into accumulator.

→ 2 byte instruction.

→ immediate addressing mode.

→ All flag are affected according to the result.

④ ADC, RI:-

$$A \leftarrow A + R + CY$$

Add the content of Register RI and status of carry flag 0 or 1 to the content of accumulator.

- 1 byte instruction.
- Register addressing mode.
- All flags are affected.

⑤ ADC, M :-  $A \leftarrow A + [HL] + CY$

Add the content of memory location pointed by Register pair and status of carry flag to the content of accumulator.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected.

⑥ ACI, data [8-bit] :-  $A \leftarrow A + 8\text{-bit data} + CY$

Add the content of 8-bit immediate data and status of carry flag to the content of accumulator.

- 2-byte instruction.
- Immediate addressing mode.
- All flags are affected.

⑦ DAD Rp :-  $HL \leftarrow HL + RP [BC \quad DE]$

Add the content of Register pair to the content of HL pair register and result is stored in HL pair register.

- 1 byte instruction.
- Register addressing mode.
- Only carry flag is affected.

⑧ SUB Ri :-  $A \leftarrow A - R$

Subtract the content of the Register from the content of accumulator.

- 1 byte instruction.
- Register addressing mode.
- All flags are affected.

⑨ SUB, M :-  $A \leftarrow A + [HL]$

subtract the content of memory location pointed by Register pair from the content of accumulator.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected.

SBB, M :-

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

subtract the content of memory location pointed by HL pair and status of carry flag from the content of accumulator.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected according to the result.

SBI, 8-bit (Data) :-  $A \leftarrow A - Data - CY$

subtract 8-bit immediate data and status of carry flag from the content of accumulator.

- 2 byte instruction.
- Immediate addressing mode. Direct data access
- All flags are affected according to the result.

INR, Ri

$R \leftarrow R + 1$

Increment the content of register.

- 1 byte instruction.
- Register addressing mode.
- Excluding the carry flag, all flags are affected according to result.

INR, Mi

$[HL] \leftarrow [HL] + 1$

Increment the content of memory location pointed by the HL pair register.

location pointed by the HL pair register.

- 1 byte instruction.
- Indirect addressing mode.
- Excluding the carry flag, all flags are affected according to result.

INX, Rp i

Increment the content of register pairs.

- 1 byte instruction.
- Register addressing mode.
- No flags are affected.

$Rp \leftarrow Rp + 1$

DCR, Ri - Decrement the content of register.

same  
as  
above

DCR, Mi - Decrement the content of memory location pointed by the HL pair register.

DCX, Rp i - Decrement the content of register pairs.

- Logical instructions

AND operation

① ANA, Ri

$A \leftarrow A \text{ AND } R$

Logical AND the content of accumulator

with the contents of Register R.

- 1 byte instruction.
- Register addressing mode.
- All flags are affected.

② ANA, M :-  $A \leftarrow A \text{ AND } [HL]$

Logically AND the content of accumulator with the content of memory location pointed by register pair HL.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected.

③ ANI, 8-bit data :-  $A \leftarrow A \text{ AND } [8\text{-bit data}]$

Logically AND the 8-bit immediate data with the content of accumulator.

- 2-byte instruction.
- immediate addressing mode.
- All flags are affected.

OR operation :-

① DRA, R :-  $A \leftarrow A \text{ OR } R$

Logically OR the content of accumulator with contents of Register R.

- 1 byte instruction.
- Register addressing mode.
- All flags are affected.

② ORA, M:-

$$A \leftarrow A \text{ OR } [HL]$$

Logically OR the content of accumulator with the content of memory location pointed pair register HL.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected.

③ ORI, 8-bit Data:-

$$A \leftarrow A \text{ OR } [8\text{-bit data}]$$

Logically OR the 8-bit immediate data with content of accumulator.

- 2 byte instruction.
- Immediate addressing mode.
- All flags are affected.

X-OR operation:-

① XRA, Ri:-

$$A \leftarrow A \text{ X-OR } R$$

Logically X-OR the content of accumulator with content of Register R.

- 1 byte instruction.
- Register addressing mode.
- All flags are affected.

② XRA, M:-

$$A \leftarrow A \text{ X-OR } [HL]$$

Logically X-OR the content of accumulator with the content of memory location pointed pair register HL.

- 1 byte instruction.
- Indirect addressing mode.
- All flags are affected.

(3) XRI, 8-bit data:-  $A \leftarrow A \text{ X-OR } [8\text{-bit data}]$   
 Logically X-OR the 8-bit immediate data with the content of accumulator.  
 → 2-byte instruction.  
 → Immediate addressing mode.  
 → All flags are affected.

complement / NOT operation

CMA :-  $A \leftarrow \bar{A}$

Complement the content of accumulator.

- 1 byte instruction.
- Implicit addressing mode.
- No flag is affected.

CMP, Ri :-  $A - R$

Compare the content of register with the content of accumulator and result of subtraction operation is NOT stored.

- 1 byte instruction.
- Register addressing mode.
- All flag are affected, according to the result.

CMP, Mi :-  $A - [HL]$

Compare the content of memory location pointed by register pair with the content of accumulator. result is not stored.

- 1 byte instruction.
- Indirect addressing mode.
- All flag are affected.

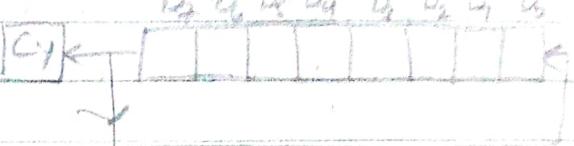
### CPI, 8-bit data:-

A - [8-bit data]

Compare the immediate data with the content of accumulator.

- 2 byte instruction.
- Immediate addressing mode.
- All flags are affected.

### Rotate instructions:-



#### RLC :-

Rotate accumulator left without carry.

- 1 byte instruction.
- Implicit addressing mode.
- Only carry flag is affected.

#### RAL :-

Rotate accumulator left with carry.

- 1 byte instruction.
- Implicit addressing mode.
- Only carry flag is affected.

#### RRc :-

Rotate accumulator right without carry.

- 1 byte instruction.
- Implicit addressing mode.
- Only carry flag is affected.

#### RAR :-

Rotate accumulator right with carry.

- 1 byte instruction.
- Implicit addressing mode.
- Only carry flag is affected.

## Branch - Instruction

→ unconditional  
→ conditional.

### Unconditional :-

#### ① JMP, 16-bit address :-

Jump unconditionally to the specified address and start execution from that address.

- 3 byte instruction.
- Immediate addressing mode.
- Flags are not affected.

#### ② CALL, 16-bit address :-

Jump unconditionally to the specified address after storing the address of next instruction of main programme.

- 3 byte instruction.
- Immediate addressing mode.
- Flags are not affected.

#### ③ RET :-

Return from the sub routine unconditionally, returning address is obtain from stack and loaded to program counter.

- 1 byte instruction.
- Indirect addressing mode.
- Flags are not affected.

### conditional :-

#### ① J-conditional 16-bit address :-

Jump to specified if address if condition is true.

- 3 byte instruction

- Direct addressing mode.
- Flags are checked only but not affected.

② C-condition - 16-bit :-

call the subroutine at specified address if condition is true.

- 3 byte instruction.
- Immediate addressing mode.
- Flags are checked only but not affected.

Machine control instruction:-

① NOP :-

No operation.

- 1 byte instruction.
- Non addressing mode.
- Flags are not affected.

② HLT :-

stop the execution.

- 1 byte instruction.
- Non addressing mode.
- Flags are not affected.

③ STC :-

set carry flag.

- 1 byte instruction.
- Implicit addressing mode.
- Carry flag is set and other flags are not affected.

#### (4) CMCI-

Complement the carry flag.

- 1 byte instruction.
- Implicit addressing mode.
- Carry flag is complemented and all others are not affected.

#### ~~\*→~~ Addressing mode of instruction:-

The different ways that a microprocessor can access data are referred to as addressing mode.

The programmer can refer the store data in different ways

8085 - microprocessor having 5 type of addressing mode.

##### i) Immediate addressing mode:-

In this addressing mode immediate data of 8-bit or 16-bit is specified as a part of instruction.

The immediate data is operated with a register or register pair according to the instruction.

e.g. → MVI , 8 bit data

SBI , 8 bit data

SBI , 8 bit data

ADI , 8 bit data

### ii) Register addressing mode:

This addressing mode specifies that the source operand, destination operand or both the operands are stored in registers or register pairs.

Note: The register addressing mode instruction are faster in execution because there is no need to access memory for operands.

### iii) Implicit addressing mode:

In this addressing mode the opcode itself specifies the address of operand, this is also known as, implicit addressing mode.

### iv) Direct addressing mode:

In this addressing mode the address of data, is directly given in the instruction itself using its hex representation.

If 16-bit address is used, the second and third byte of instruction content contain this 16-bit address.

If 8-bit address is used, the second byte is the 8-bit, forth address.

### v) Indirect addressing mode:

If the address of data is stored in register pair and the name of that

register pair, is shown in instruction, that the instruction having indirect addressing mode.

### Programming techniques:-

- ① Indexing → To point the next data
- ② Counting → To count for the end of available data
- ③ Branching → Change the sequence and perform the task again.

### \* → Debugging of program -

Debugging of a program is similar to troubleshooting of a hardware but it is even much more difficult some clues keep alert to the program to know about errors made by him / her.

The debugging process can be divided into two parts -

#### 1) static debugging :-

static debugging is similar to visual inspection of hardware by a paper and pencil check of the flow chart and machine code.

The some common errors which can be find and available in static debugging are -

- selecting a <sup>ROM</sup> machine code.
- specifying the ROM group instruction.

- Giving Rom order of high and low order bits.
- Failure to set a flag before using jump instruction.
- Failure in indexing and counting.
- Failure to clear ~~to~~ the accumulator when it is used to arithmetic operation.
- Failure to clear register when it is used to track some data like carry.

## 2) Dynamic debugging:-

Sometimes even after <sup>static</sup> debugging program doesn't work then the user can go for dynamic debugging, the tools of dynamic debugging are:

- Single step
- Break point

i) single step - single step key on the keyboard of single board microprocessor always to execute one instruction at a time and after that program can check for the expected result by examine, memory or register and then continue.

ii) Break point - Break point facility provided a software service routine that allows the programmer to execute a program in section

Break point are inserted in program by writing RST instruction and as the execution key is pressed execution takes place upto the 1<sup>st</sup> break point and after that user can check for expected result and then go for the next break point.