Unit -II

Addressing Modes and Instruction Cycle:

Addressing Mode:

Addressing mode is the process or way or method to define the data for the instruction.

The 8085 Microprocessor has 5 types of Addressing modes is as follows-

1] Register Addressing Mode

2] Immediate Addressing Mode

3]Direct Addressing Mode

4] Indirect addressing mode

5] Implicit/Implied Addressing Mode

1] Register Addressing Mode:

The addressing mode in which the data is given in one of the general purpose registers such as A, B, C, D, E, H and L is called as Register Addressing Mode. All these registers are user accessible i.e. used by the user.

ex.1. MOV A,B

Format:

[A] ← [B]

This instruction moves the contents (data) of register B into register A. In this instruction the data is in register A and B.

ex.2. ADD C

Format:

[A] ← [A] + [C]

This instruction adds the contents (data) of register C with the contents of register A and the result is stored back into register A i.e. Accumulator. In this instruction the data is in register A and C.

ex.3. SUB D

Format:

[A] ← [A] - [D]

This instruction subtracts the contents (data) of register D from the contents of register A and the result is stored back into register A i.e. Accumulator. In this instruction the data is in register A and D.

ex.4. INR E

Format:

[E] ← [E] + 1

This instruction increments the contents (data) of register E by one. In this instruction the data is in register E.

ex.5. DCR H

Format:

[H] ← [H] - 1

This instruction decrements the contents (data) of register H by one. In this instruction the data is in register H.

2] Immediate Addressing Mode:

The addressing mode in which the data is given in the instruction itself is called as Immediate Addressing Mode.

or

The addressing mode in which the data is the part of the instruction is called as Immediate Addressing Mode.

ex.1. MVI A,06H

Format:

[A] ← [06H]

This instruction moves the immediate data 06H into register A. Here the 06H data is given in the instruction itself.

ex.2. ADI 26H

Format:

[A] ← [A] + [26H]

This instruction adds immediate data 26H with the contents of register A and the result is stored back into register A i.e. Accumulator. Here 26H data is given in this instruction itself.

ex.3. SUI 48H

Format:

[A] ← [A] - [48H]

This instruction subtracts immediate data 48H from the contents of register A and the result is stored back into register A i.e. Accumulator. Here 48H data is given in this instruction itself.

ex.4. ANI 57H

Format:

[A] ← [A] ^ [57H]

This instruction logically AND's immediate data 57H with the contents of register A and the result is stored back into register A i.e. Accumulator. Here 57H data is given in this instruction itself.

ex.5. ORI 86H

Format:

[A] ← [A] v [86H]

This instruction logically OR's immediate data 86H with the contents of register A and the result is stored back into register A i.e. Accumulator. Here 86H data is given in this instruction itself.

3]Direct Addressing Mode:

The addressing mode in which the address of the data is given in the instruction itself is called as Direct Addressing Mode.

ex.1. LDA 6000H

Format:

[A] ← [6000H]

This instruction Loads Directly Accumulator from 6000H memory location. Here 6000H is the address of memory location which is directly given in the instruction itself.

ex.2. STA 5000H

Format:

[5000H] ← [A]

This instruction stores the contents (data) of Accumulator at 5000H memory location. Here 5000H is the address memory location, which is directly given in the instruction itself.

ex.3. LHLD 8001H

Format:

[L] ← [8001H]

[H] ← [8002H]

This instruction Loads HL pair Directly from 8001H and 8002H memory locations. Here the contents of register L are loaded first from 8001H memory location and then the contents of register H are loaded from the next memory location 8002H. Here 8001H address of memory is directly given in the instruction itself.

ex.4. SHLD 2001H

Format:

[2001H] ← [L]

[2002H] ← [H]

This instruction Stores HL pair Directly at 2001H and 2002H memory locations. Here the contents of register L are stored at 2001H memory location and then the contents of register H are stored at 8002H memory location. Here 2001H address of memory is directly given in the instruction itself.

ex.5. IN 01H

Format:

[A] ← [01H]

This instruction inputs the data to the processor's Accumulator from 01H I/O port address. Here 01H is the I/O port address which is directly given in the instruction itself.

4] Indirect addressing mode:

The addressing mode in which the address of data is given in the register pair is called as **Indirect addressing mode.**

ex.1. LXI H, 2000H ; Load HL pair with memory address 2000H

MOV A,M ; Move the contents of memory whose address is in HL pair.

Format:

[A] ← [[HL]]

Here the memory address 2000H is given in HL register pair.

ex.2. LXI H, 3000H **; Load HL pair with memory address 3000H**

ADD M ; Add the contents of memory whose address is in HL pair with reg. A i.e. Accumulator and result is stored back into the Accumulator.

Format:

[A] ← [A] + [[HL]]

Here the memory address 3000H is given in HL register pair.

ex.3. LXI H, 3000H **; Load HL pair with memory address 3000H**

SUB M ; Subtracts the contents of memory whose address is in HL pair from reg. A i.e. Accumulator and result is stored back into Accumulator.

Format:

[A] ← [A] + [[HL]]

Here the memory address 3000H is given in HL register

pair.

ex.4. LXI H, 1000H ; Load HL pair with memory address

3000H

INR M ; Increment the contents of memory whose address is in HL pair by one.

Format:

[[HL]] ← [[HL]] + 1

Here the memory address 1000H is given in HL register

pair.

ex.5. LXI H, 7000H ; Load HL pair with memory address

3000H

DCR M ; Decrement the contents of memory whose address is in HL pair by one.

Format:

[[HL]] ← [[HL]] - 1

Here the memory address 7000H is given in HL register .

pair.

5] Implicit/Implied Addressing Mode:

The addressing mode in which no data is given in the instruction is called as Implicit or Implied Addressing Mode.

In this addressing mode the operation is directly performed on the contents of Accumulator.

ex.1. CMA

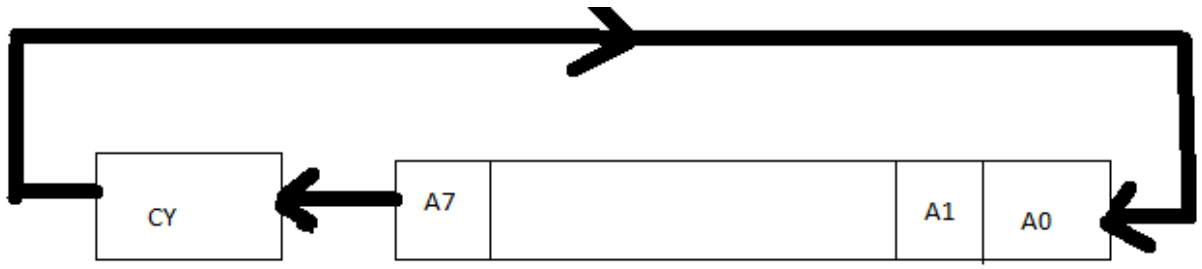
Format:

$$[A] \leftarrow [\bar{A}]$$

This instruction one's complement the contents of accumulator. Here no data is given in the instruction.

ex.2. RAL

Format:



[A1] ← [A0]

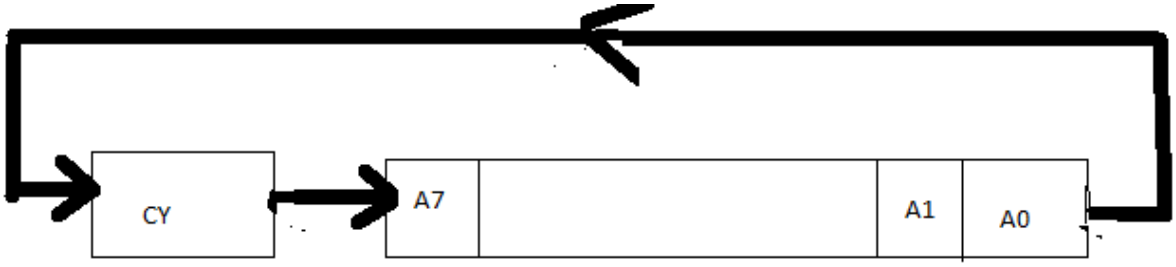
[CY] ← [A7]

[A0] ← [CY]

This instruction Rotates the contents of accumulator in left direction by one-bit with carry. Here no data is given in the instruction.

ex.3. RAR

Format:



[A0] ← [A1]

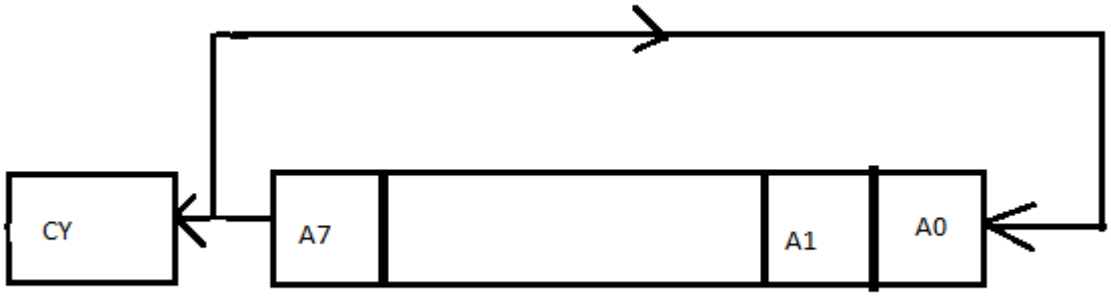
[CY] ← [A0]

[A7] ← [CY]

This instruction Rotates the contents of accumulator in right direction by one-bit with carry. Here no data is given in the instruction.

ex.4. RLC

Format:



[A1] ← [A0]

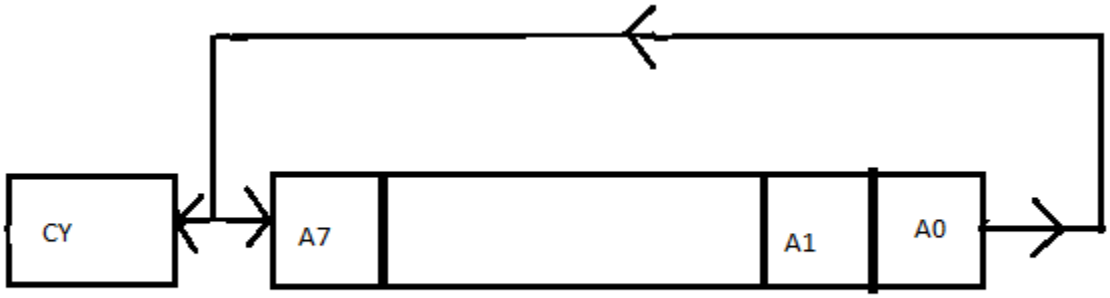
[CY] ← [A7]

[A0] ← [A7]

This instruction Rotates the contents of accumulator in left direction by one-bit without carry. Here no data is given in the instruction.

ex.5. RRC

Format:



[A0] ← [A1]

[CY] ← [A0]

[A7] ← [A0]

This instruction Rotates the contents of accumulator in right direction by one-bit without carry. Here no data is given in the instruction.

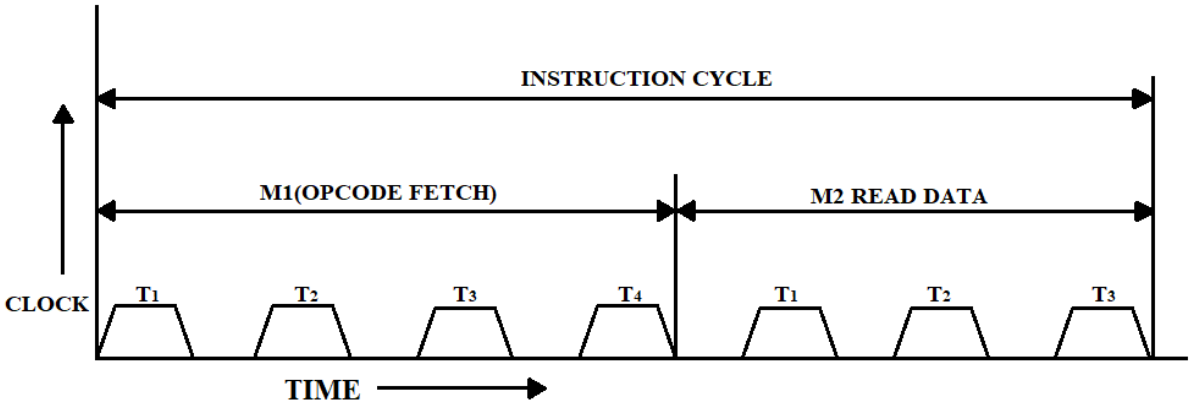
Instruction Cycle:

Instruction cycle is combination of two cycles i.e. Fetch cycle and Execute cycle.

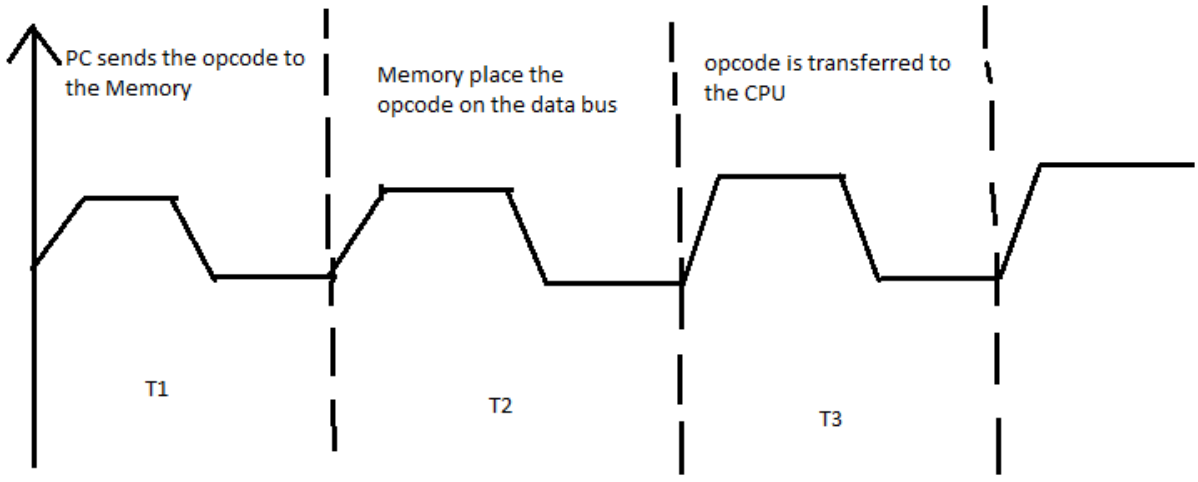
Instruction cycle = Fetch cycle + Execute cycle
i.e. IC = FC + EC

Defination:

Instruction cycle is defined as the process of accessing the opcode and data from the memory and performing operation on the data given in the instruction.



1] Fetch Cycle:



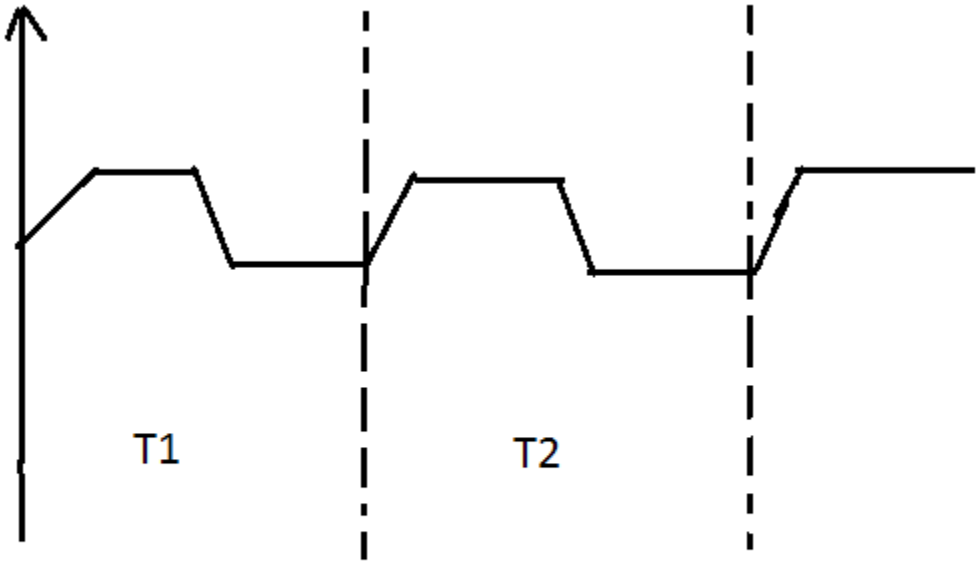
Fetch Cycle

Fetch cycle is defined as the process of accessing the opcode from the memory by the CPU.

- The main part of fetch cycle is opcode.
- Opcode means operational code , which defines the operation performed by the computer.
- The size of the opcode is 1-Byte i.e. 8-bit.
- So the time period of fetch cycle is always fixed i.e. fetch cycle requires 3 clock time T1, T2 and T3.
    - During T1 PC sends the opcode to the memory.
    - During T2 memory place the opcode on data bus.

- During T3 opcode is transferred to the CPU.

2] Execute Cycle:



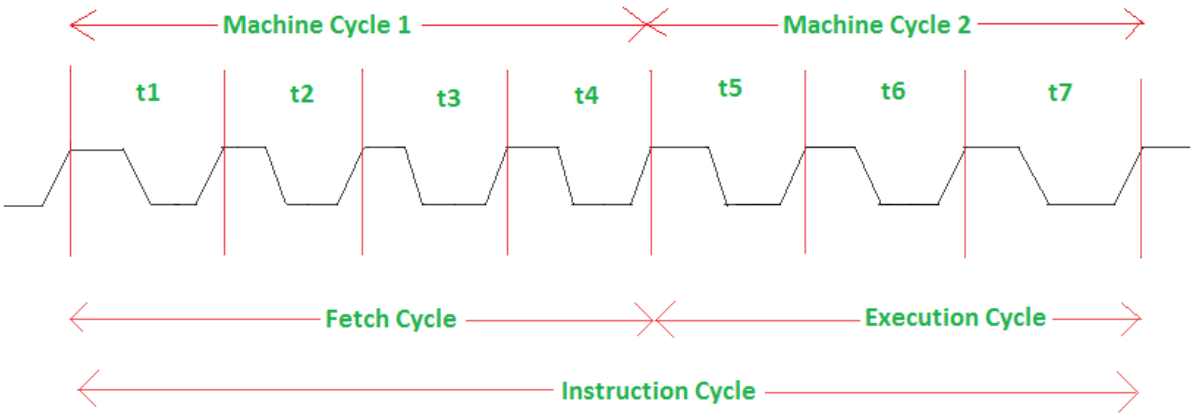
Execute Cycle

Execute cycle is the process of accessing the data from the memory if any and performing the operation on the data, which is given in the instruction.

- In this cycle the opcode is placed in the data register or data/address buffer and then opcode is transferred to the IR.
- From the IR the opcode is finally transferred to the decoder.

- So, the decoder will decode the opcodeand then the instruction is executed.
- The Execution cycle requires only one T-state, if the data is in the register i.e. the instruction is 1-byte.
- The Execution cycle requires two T-states, if the instruction is 2-byte or 3-byte.

Machine Cycle:



Instruction cycle in 8085 microprocessor

Machine cycle is defined as the process of accessing the memory or I/O devices.

or

Machine cycle is defined as the process of performing the operation of fetch, read or write operation.

- That means in one machine cycle only one basic operation is performed i.e. opcode fetch or memory read or memory write or I/O read or I/O write.
- Every Instruction cycle consist of one or more Machine cycles.
- In M1 machine cycle always the opcode of the instruction is fetched from the memory. M1 consist of 4 clock cycles T1,T2,T3 and T4 because in this cycle the opcode is fetch and executed. Whereas the other machine cycles such as M2, M3, M4,----- Mn consists of only three clock cycles i.e. T1,T2 and T3.
- In M2 machine cycle the low order 8-bit data is either read or write into the memory or I/O devices.
- In M3 machine cycle the high order 8-bit data is either read or write into the memory or I/O devices.