Microprocessors

Dr Mridul Gupta

Assistant Professor

Dept. of ECE

UNIT-2

8085 Instructions and Assemble Language Programming

8085 Instructions

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function.
- The entire group of instructions that a microprocessor supports is called **instruction set**.
- An **assembly language program** is a set of instructions written in the mnemonics of a given microprocessor.
- 8085 is an 8-bit device it can have up to 2^8 (256) instructions.
- However, the 8085 only uses 246 combinations that represent a total of 74 instructions.
  - Most of the instructions have more than one format.

8085 Instructions

- Microprocessor instructions can be classified based on the parameters such functionality, length and operand addressing.

Classification based on functionality (**five** different groups):

- Data Transfer Operations
- Arithmetic Operations
- Logic Operations
- Branch Operations
- Machine Control Operations

8085 Instructions

1. Data transfer operations:
   - This group of instructions copies data from source to destination.
   - The content of the source is not altered.
2. Arithmetic operations:
   - Instructions of this group perform operations like addition, subtraction, increment & decrement.
   - One of the data used in arithmetic operation is stored in accumulator and the result is also stored in accumulator.

8085 Instructions

3. Logical operations:
   - Logical operations include AND, OR, EXOR, NOT.
   - The operations like AND, OR and EXOR uses two operands, one is stored in accumulator and other can be any register or memory location. The result is stored in accumulator.
   - NOT operation requires single operand, which is stored in accumulator.

8085 Instructions

4. Branching operations:
   - Instructions in this group can be used to transfer program sequence from one memory location to another either conditionally or unconditionally.
5. Machine control operations:
   - Instruction in this group control execution of other instructions and control operations like interrupt, halt etc.

8085 Instructions

Classification based on length:

1. One-byte instructions:
   - Instruction having one byte in machine code.
2. Two-byte instructions:
   - Instruction having two byte in machine code.
3. Three-byte instructions:
   - Instruction having three byte in machine code.

8085 Instructions

- Each instruction has **two** parts.
  - The first part is the task or operation to be performed.
    - This part is called the “opcode” (operation code).
  - The second part is the data to be operated on
    - Called the “operand”.

8085 Instructions

- Examples of One Byte instructions

Task	Opcode	Operand*	Binary Code	Hex Code
Copy the contents of the accumulator in register C.	MOV	C,A	0100 1111	4FH
Add the contents of register B to the contents of the accumulator.	ADD	B	1000 0000	80H
Invert (complement) each bit in the accumulator.	CMA		0010 1111	2FH

*the destination register C is shown first, followed by the source register.

8085 Instructions

- Examples of Two Byte instructions

Task	Opcode	Operand	Binary Code	Hex Code	
Load an 8-bit data byte in the accumulator.	MVI	A,32H	0011 1110	3E	First Byte
			0011 0010	32	Second Byte
Load an 8-bit data byte in register B.	MVI	B,F2H	0000 0110	06	First Byte
			1111 0010	F2	Second Byte

8085 Instructions

- Examples of Three Byte instructions

Task	Opcode	Operand	Binary Code	Hex Code*	
Load contents of memory 2050H into A.	LDA	2050H	0011 1010	3A	First Byte
			0101 0000	50	Second Byte
			0010 0000	20	Third Byte
Transfer the program sequence to memory location 2085H.	JMP	2085H	1100 0011	C3	First Byte
			1000 0101	85	Second Byte
			0010 0000	20	Third Byte

*the 16-bit addresses are stored in memory locations in reversed order, the low-order byte first, followed by the high-order byte..

8085 Instructions

- Each instruction in many ways is similar to our everyday conversation. E.g.:

1. Pass the butter (the request specifies the exact item).
   - It is similar to the instruction for loading a specific data byte in a register.
2. Pass the bowl (mentions the bowl rather than its content).
   - Similar to MOV C,A, where registers (bowls) are requested rather than data.
3. Let us eat (assumes that one knows what to eat).
   - Similar to the instruction “Complement” which implicitly assumes that the operand is the Accumulator.

8085 Instructions

- Each instruction in many ways is similar to our everyday conversation. E.g.:

4. I will have combination 17 on the menu (location of the item).
   - Similar to transfer the data byte from the location 2050H.
5. I will have what Susie ordered (specified indirectly).
   - Specifies memory location through the contents of a register pair.

- These various ways of specifying data are called **Addressing Modes**.

Addressing Modes in Instructions

- The process of specifying the data to be operated on by the instruction is called addressing.
- The various formats for specifying operands are called **addressing modes**.
- The 8085 has the following **five types of addressing**:
  1. Immediate addressing
  2. Memory direct addressing
  3. Register direct addressing
  4. Indirect addressing
  5. Implicit addressing

Addressing Modes in Instructions

Immediate Addressing:

- In this mode, the operand given in the instruction - a byte or word – transfers to the destination register or memory location.
  - E.g.: MVI A, 9AH
- The operand is a part of the instruction.
- The operand is stored in the register mentioned in the instruction.

Addressing Modes in Instructions

Memory Direct Addressing:

- Memory direct addressing moves a byte or word between a memory location and register.
- The memory location address is given in the instruction.
  - E.g. LDA 850FH: This instruction is used to load the content of memory address 850FH in the accumulator.

Register Direct Addressing:

- Register direct addressing transfer a copy of a byte or word from source register to destination register.
  - E.g. MOV B,C : It copies the content of register C to register B.

Addressing Modes in Instructions

Indirect Addressing:

- Indirect addressing transfers a byte or word between a register and a memory location.
  - E.g. MOV A, M: Here the data is in the memory location pointed to by the contents of HL pair. The data is moved to the accumulator.

Implicit Addressing:

- In this addressing mode the data itself specifies the data to be operated upon.
  - E.g. CMA: The instruction complements the content of the accumulator. No specific data or operand is mentioned in the instruction.

Data Format

In an 8-bit microprocessor, data can be represented in one of four formats:

- ASCII
- BCD
- Signed Integer:
  - the most significant bit D7, is used for sign. Therefore the largest positive integer that can be processed by the 8085 at one time is 0111 1111 (7FH);
  - the remaining Hex numbers, 80H to FFH, are considered negative numbers.
  - However, all negative numbers in this microprocessor are represented in 2's complement format.
- Unsigned Integer

Data Format

- It is important to recognize that the microprocessor deals with 0's and 1's.
  - It deals with values as strings of bits.
  - It is the job of the user to add a meaning to these strings.

- Assume that after performing some operations the result in the accumulator is 0100 0001 (41H).

Data Format

- There are four ways of reading this value:
  - It is an unsigned integer expressed in binary, the equivalent decimal number would be 65.
  - It is a number expressed in BCD (Binary Coded Decimal) format. That would make it, 41.
  - It is an ASCII representation of a letter. That would make it the letter A.
  - It is a string of 0's and 1's where the 0^{th} and the 6^{th} bits are set to 1 while all other bits are set to 0.

Overview of 8085 Instruction Set

- The following NOTATIONS are used:

Notation	Meaning
M	Memory location pointed by HL register pair
r	8-bit register
rp	16-bit register pair
rs	Source register
rd	Destination register
addr	16-bit address / 8-bit address

Data Transfer (Copy) Instructions

- These instructions perform the following six operations:

  - Load an 8-bit number in a register
  - Copy from register to register
  - Copy between I/O and accumulator
  - Load 16-bit number in a register pair
  - Copy between register and memory
  - Copy between registers and stack memory

Data Transfer (Copy) Instructions

1. MVI r, data (8) This instruction directly loads a specified register with an 8-bit data given within the instruction. The register r is an 8-bit general purpose register such as A, B, C, D, E, H and L.

Operation : r ← 8-bit data (byte)

Example :

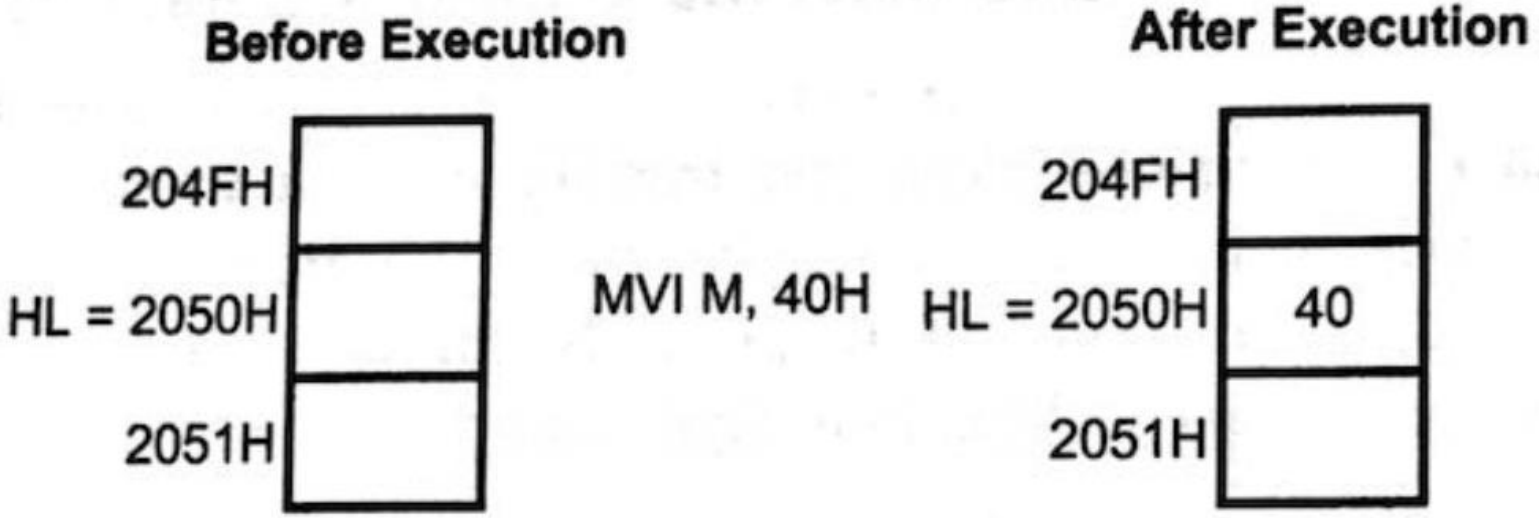
MVI B, 60H ; This instruction will load 60H directly into the B register.

2. MVI M, data (8) This instruction directly loads an 8-bit data given within the instruction into a memory location. The memory location is specified by the contents of HL register pair.

Operation : M ← byte or (HL) ← byte

Example : H = 20H and L = 50H

MVI M, 40H ; This instruction will load 40H into memory whose address is 2050H.



3. MOV rd, rs This instruction copies data from the source register into destination register. The rs and rd are general purpose registers such as A, B, C, D, E, H and L. The contents of the source register remain unchanged after execution of the instruction.

Operation : rd ← rs

Example : A = 20H

```
MOV B, A    ; This instruction will copy the contents of register A (20H) into
              register B.
```

4. MOV M, rs This instruction copies data from the source register into memory location pointed by the HL register pair. The rs is an 8-bit general purpose register such as A, B, C, D, E, H and L.

Operation : (HL) ← rs

Example : If HL = 2050H, B = 30H.

MOV M, B ; This instruction will copy the contents of B register (30H) into the memory location whose address is specified by HL (2050H).

5. MOV rd, M This instruction copies data from memory location whose address is specified by HL register pair into destination register. The contents of the memory location remain unchanged. The rd is an 8-bit general purpose register such as A, B, C, D, E, H and L.

Operation : rd ← (HL)

Example : HL = 2050H, contents at 2050H memory location = 40H

MOV C, M ; This instruction will copy the contents of memory location pointed by HL register pair (40H) into the C register.

6. LXI rp, data (16) This instruction loads immediate 16 bit data specified within the instruction into register pair or stack pointer. The rp is 16-bit register pair such as BC, DE, HL or 16-bit stack pointer.

Operation : rp ← data (16)

Example :

LXI B,1020H ; This instruction will load 10H into B register and 20H into C register.

7. STA addr This instruction stores the contents of A register into the memory location whose address is directly specified within the instruction. The contents of A register remain unchanged.

Operation : (addr) ← A

Example : A = 50H

STA 2000H ; This instruction will store the contents of A register (50H) to memory location 2000H.

8. LDA addr This instruction copies the contents of the memory location whose address is given within the instruction into the accumulator. The contents of the memory location remain unchanged.

Operation : A ← (addr)

Example : (2000H) = 30H

LDA 2000H ; This instruction will copy the contents of memory location 2000H i.e. data 30H into the A register

9. SHLD addr This instruction stores the contents of L register in the memory location given within the instruction and contents of H register at address next to it. This instruction is used to store the contents of H and L registers directly into the memory. The contents of the H and L registers remain unchanged.

Operation : (addr) ← L and (addr + 1) ← H

Example : H = 30H, L = 60H

SHLD 2500H ; This instruction will copy the contents of L register at address 2500H and the contents of H register at address 2501H.

10. LHLD addr This instruction copies the contents of the memory location given within the instruction into the L register and the contents of the next memory location into the H register.

Operation : $L \leftarrow (addr), H \leftarrow (addr + 1)$

Example : $(2500H) = 30H, (2501H) = 60H$

LHLD 2500H ; This instruction will copy the contents of memory location 2500H i.e. data 30H into the L register and the contents at memory location 2501H i.e. data 60H into the H register.

11. STAX rp This instruction copies the contents of accumulator into the memory location whose address is specified by the specified register pair. The rp is BC or DE register pair. This register pair is used as a memory pointer. The contents of the accumulator remain unchanged.

Operation : (rp) ← A

Example : BC = 1020H, A = 50H

STAX B ; This instruction will copy the contents of A register (50H) to the memory location specified by BC register pair (1020H).

12. LDAX rp This instruction copies the contents of memory location whose address is specified by the register pair into the accumulator. The rp is BC or DE register pair. The register pair is used as a memory pointer.

Operation : $A \leftarrow (rp)$

Example : DE = 2030H, (2030H) = 80H

LDAX D This instruction will copy the contents of memory location specified by DE register pair (80H) into the accumulator.

13. XCHG This instruction exchanges the contents of the register H with that of D and of L with that of E.

Operation : $H \leftrightarrow D$ and $L \leftrightarrow E$

Example : DE = 2040H, HL = 7080H

```
XCHG    ; This instruction will load the data into registers as follows
          H = 20H, L = 40H, D = 70H and E = 80
```