Lecture 7

8085 Instruction Set

Classification:

1. On the basis of byte
2. On the basis of function

1. **On the basis of byte**

The 8085 instruction set is classified into the following three groups according to word size:

1. One-word or 1-byte instructions
2. Two-word or 2-byte instructions
3. Three-word or 3-byte instructions

In the 8085, "byte" and "word" are synonymous because it is an 8-bit microprocessor.

ONE-BYTE INSTRUCTIONS

A 1-byte instruction includes the opcode and the operand in the same byte. For example:

Task	Opcode	Operand	Binary Code	Hex code
Copy the contents of the accumulator in register C.	MOV	C,A	0100 1111	4FH

TWO-BYTE INSTRUCTIONS

In a 2-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. For example:

Task	Opcode	Operand	Binary Code	Hex code
Load an 8-bit data byte in the accumulator	MVI	A, Data	0011 1110 DATA	3E-First byte Data- Second byte

This instruction would require two memory locations to store in memory.

THREE-BYTE INSTRUCTIONS

In a 3-byte instruction, the first byte specifies the opcode and the following two bytes specify the 16-bit address. Note that the second byte is the low-order address and the third byte is the high- order address.

For example:

Task	Opcode	Operand	Binary Code	Hex code
Transfer the program sequence to the memory location 2085H.	JMP	2085H	1100 0011 1000 0101 0010 0000	C3 First byte 85Second byte 20 Third byte

This instruction would require three memory locations to store in memory.

2. On the basis of function

The 8085 instructions can be classified into the following five functional categories:

1. Data transfer (copy) operations
2. Arithmetic operations
3. logical operations
4. Branching operations and
5. Machine control operations.

An instruction is a command to the microprocessor to perform a given task on specified data. Each instruction has two parts: one is the task to be performed, called the operation code (opcode), and the second is the data to be operated on, called the operand (or data) can be specified in various ways. It may include 8-bit (or 16-bit) data, an internal register, a memory location, or an 8-bit (or 16-bit) address.

1. DATA TRANSFER (COPY) OPERATIONS

This group of instructions copies data from a location called a source to another location called destination, without modifying the contents of the source. The various types of data transfer are listed below together with examples of each type:

Types	Example
Between registers	Copy the contents of register B into Register D.
Specific data byte to a register or a memory location	Load register B with the data byte
Between a memory location and a register	From the memory location 2000H to register B.
Between an I/O device and the accumulator.	From an input keyboard to the accumulator

INSTRUCTIONS

The data transfer instructions copy data from a source into a destination without modifying the contents of the source. The previous contents of the destination are replaced by the contents of the source.

Instruction	Description	Example
MOV R_d, R_s **MOV M, R_s** **MOV R_s, M**	This instruction copies the contents of the source register into the destination register; the contents of the source register are not altered. If one of the operands is a memory location, its location is specified by the contents of the HL registers.	**MOV B, C** **MOV B, M**
MVI R_d, data **MVI M, data**	The 8-bit data is stored in the destination register or memory. If the operand is a memory location, its location is specified by the contents of the HL registers.	**MVI B, 57H** **MVI M, 57H**
LXI Reg.-pair, 16-bit data	The instruction loads 16-bit data in the register pair designated in the operand.	**LXI H, 2034H** **LXI H, XYZ**
LDA 16-bit address	The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator. The contents of the source are not altered.	**LDA 2034H**
STA 16-bit address	The contents of the accumulator are copied into the memory location specified by the operand. This is a 3-byte instruction, the second byte specifies the low- order address and the third byte specifies the high-order address.	**STA 4350H**
LDAX B/D Reg. pair	The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into the accumulator. The contents of either the register pair or the memory location are not altered.	**LDAX B**
STAX Reg. pair	The contents of the accumulator are copied into the memory location specified by the contents of the operand (register pair). The contents of the accumulator are not altered.	**STAX B**
LHLD 16-bit address	The instruction copies the contents of the memory location pointed out by the 16-bit address into register L and copies the contents of the next memory location into register H. The contents of source memory locations are not altered.	**LHLD 2040H**
SHLD 16-bit address	The contents of register L are stored into the memory location specified by the 16-bit address in the operand and the contents of H register are stored into the next memory location by incrementing the operand. The contents of registers HL are not altered. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address**.**	**SHLD 2470H**
XCHG	The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E.	**XCHG**
IN 8-bit port address	The contents of the input port designated in the operand are read and loaded into the accumulator.	**IN 8CH**
OUT 8-bit port address	The contents of the accumulator are copied into the I/O port specified by the operand.	**OUT F8H**

1. **Write an ALP to load register B with data 14H, register C with FFH, register D with 29H and register E with 67H.**

```
MVI B, 14H

MVI C, FFH

MVI D, 29H

MVI E, 67H

HLT
```

2. **Write an ALP to transfer data from register B to C.**

```
MVI B, 55H

MOV C, B

HLT
```

3. **Write an ALP to store data of register B into memory location 2050H.**

```
MVI B, 67H

MOV A, B

STA 2050H ; Store data of Accumulator at memory location 2050H

HLT
```

4. **Write an ALP which directly store data 56H into memory location 2050H.**

```
LXI H, 2050H
```

MVI M, 56H

HLT

5. **Write an 8085 assembly language program for exchanging two 8-bit numbers stored in memory locations 2050h and 2051h.**

LDA 2050H

MOV B, A

LDA 2051H

STA 2050H

MOV A, B

STA 2051H

HLT

6. **Write an ALP to interchange 16-bit data stored in register BC and DE.**

WITHOUT XCHG INSTRUCTION

MOV H, B

MOV L, C

XCHG ; The contents of register H are exchanged with the contents of register D, and the

```
        ; contents of register L are exchanged with the contents of register E.

MOV B, H

MOV C, L

HLT
```