Computer Architecture

Lecture 6
Instruction Set

Assit. Lect.:Noor H. Hassoon

Instruction Set

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions, called the **instruction set**, determines what functions the microprocessor can perform. An Instruction is a command given to the computer to perform a specified operation on given data. The instructions described are of Intel 8085. These instructions are of Intel Corporation.

- They cannot be used by other microprocessor manufactures. The programmer can write a program in assembly language using these instructions.

Instruction Set

- These instructions can be classified into the following five functional categories:
  - **Data Transfer Instruction,**
  - **Arithmetic Instructions,**
  - **Logical Instructions,**
  - **Branching Instructions,**
  - **Control Instructions,**

Instruction Set

Machine Control Operations

- These instructions control machine functions such as Halt, Interrupt, or do nothing. The microprocessor operations related to data manipulation can be summarized in four functions:

1. Copying data
2. Performing arithmetic operations
3. Performing logical operations
4. Testing for a given condition and alerting the program sequence

Instruction Set

Some important aspects of the instruction set are noted below:

1. In data transfer, the contents of the source are not destroyed; only the contents of the destination are changed. The data copy instructions do not affect the flags.

2. Arithmetic and Logical operations are performed with the contents of the accumulator, and the results are stored in the accumulator (with some expectations). The flags are affected according to the results.

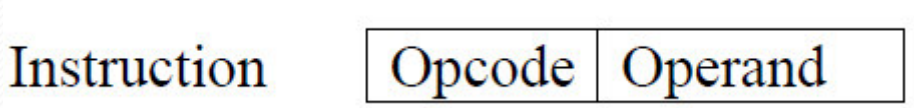
3. Any register including the memory can be used for increment and decrement.

4. A program sequence can be changed either conditionally or by testing for a given data condition.

Instruction Set

Instruction Format

- An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: the first is used to the task to be performed, called the operation code (opcode), and the second is the data to be operated on, called the operand. 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 8-bit (or 16-bit) address. In some instructions, the operand is implicit.



Instruction Set

Instruction Word Size

- 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. However, instructions are commonly referred to in terms of bytes rather than words.

Instruction Set

1- One – Byte Instructions

- A 1-byte instruction includes the opcode and operand in the same byte. Operand(s) are internal register and are coded into the instruction. For example:

Task	Op code	Operand	Binary Code	Hex Code
Copy the contents of the accumulator in the 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 (compliment) each bit in the accumulator.	CMA		0010 1111	2FH

Instruction Set

- These instructions are 1-byte instructions performing three different tasks.
  - ***In the first instruction***, both operand registers are specified.
  - ***In the second instruction***, the operand B is specified and the accumulator is assumed.
  - ***Similarly, in the third instruction***, the accumulator is assumed to be the implicit operand. These instructions are stored in 8- bit binary format in memory; each requires one memory location.

Instruction Set

2- Two – Byte Instructions

- In a two-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. Source operand is a data byte immediately following the opcode. For example:

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

Instruction Set

3- Three – Byte Instructions

- In a three-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. opcode + data byte + data byte.

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 85 20	First byte Second Byte Third Byte

Instruction Set

Data Transfer Instruction

- The data transfer instructions move data between registers or between memory and registers.

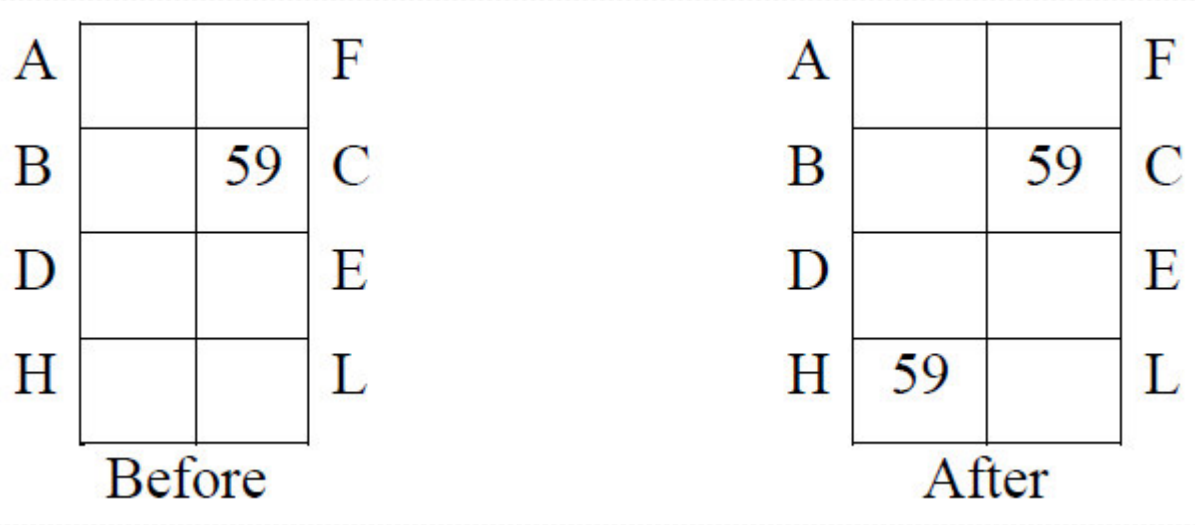
Copy from source to destination		
MOV	Rd, Rs	This instruction copies the contents of the source register into the destination register, the contents of Rd, M the source register are not altered. If one of the operands is a memory location, its location is specified by the contents of the HL registers. **Example:** MOV B, C or MOV B, M
	M, Rs	
	Rd, M	

Instruction Set

Example

- Copy the content of the register C to Register H

MOV H, C



Instruction Set

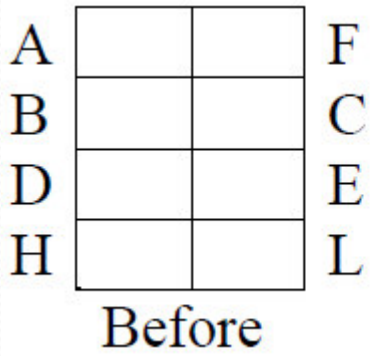
Move immediate 8-bit		
MVI	Rd, data	The 8-bit data is stored in the destination register or memory. If the operand is a memory location, its location is specified by the contents of the HL registers. **Example:** MVI B, 57H or MVI M, 57H
	M, data	

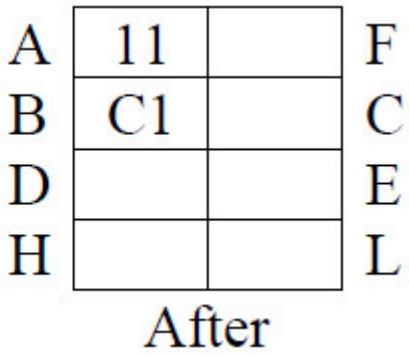
Example

- Load the register B with C1 h and the accumulator with 11 h

MVI B, C1 h

MVI A, 11h





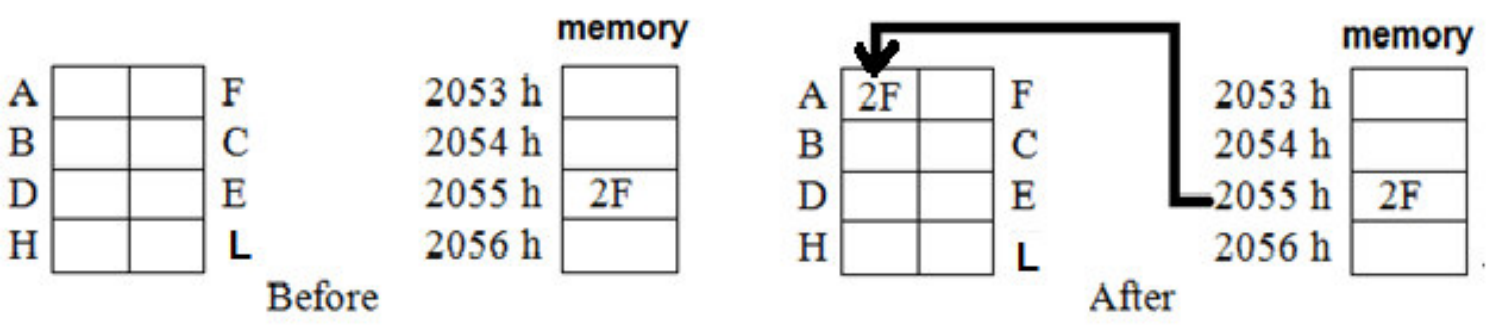
Instruction Set

Load accumulator		
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. **Example:** LDA 2034H

Example

- Load the accumulator with content of memory location 2055 h

LDA 2055h



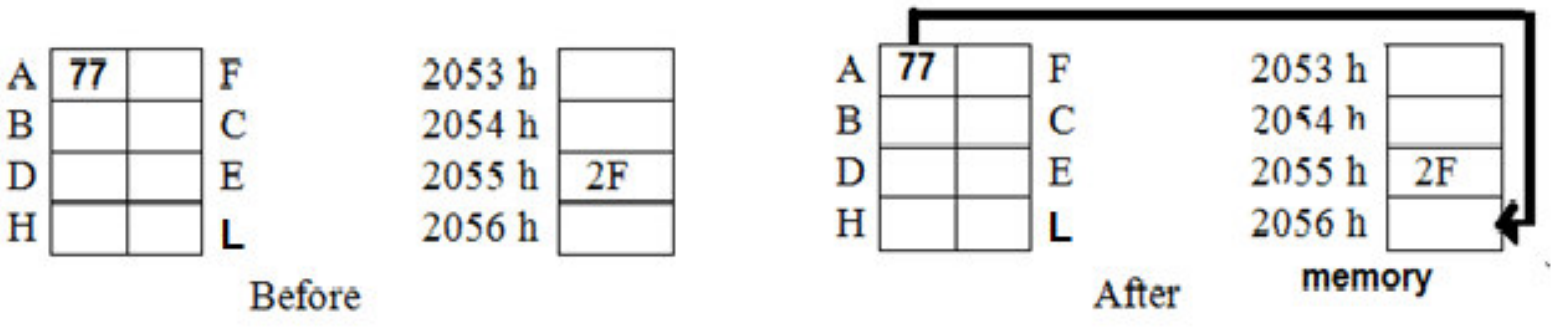
Instruction Set

Store accumulator direct		
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. **Example:** STA 4350H

Example

- Store the content of accumulator to memory location 2056 h

STA 2056h



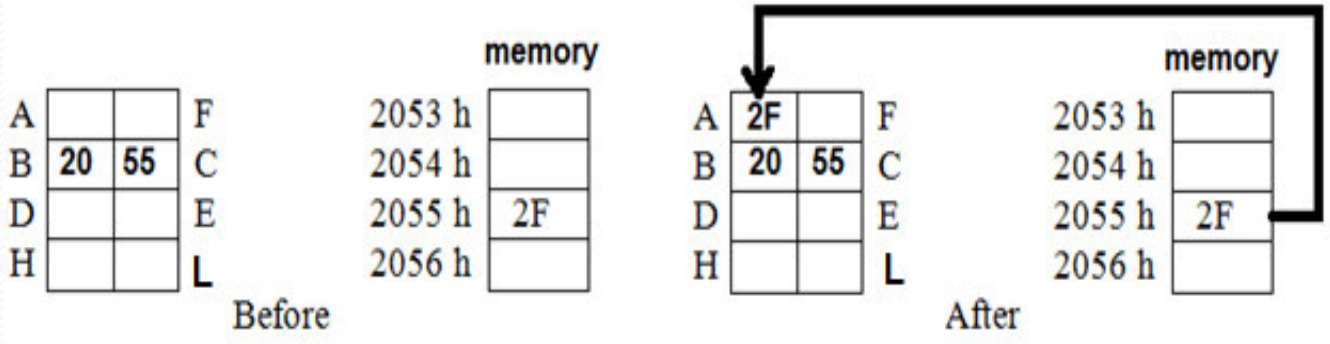
Instruction Set

Load accumulator indirect		
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. **Example:** LDAX B

Example

- Load the content of memory location to accumulator if the address specify in register pair B

LDAX B



Instruction Set

Store accumulator Indirect		
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. **Example:** STAX B

Example

- Store the content of register H to memory location if the address specify in register pair B

```
MOV A, H
STAX B
```

Instruction Set

Load register pair immediate		
LXI	Reg. pair, 16-bit data	The instruction loads 16-bit data in the register pair designated in the operand. **Example:** LXI H, 2034H or LXI H, XYZ

Example

- Copy the content of register B to the memory location 2053 h

  LXI H, 2053 H

  MOV M, B

Example

- Load the memory location 2053 h with data F8 h.

  LXI H, 2053h

  MVI M, F8 h

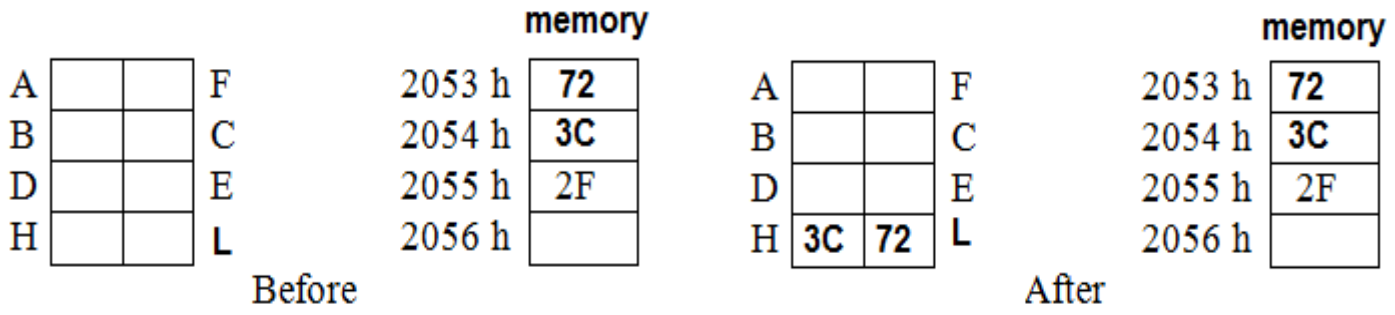
Instruction Set

Load H and L registers direct		
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. **Example**: LHLD 2040H

Example

- Load the content of the memory location 2053 h and 2054 to H and L registers

LHLD 2053h



Instruction Set

Store H and L registers direct		
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. **Example:** SHLD 2470H

Example

- Store the content of the H and L registers to memory location 2053 h and 2054

  SHLD 2053 h

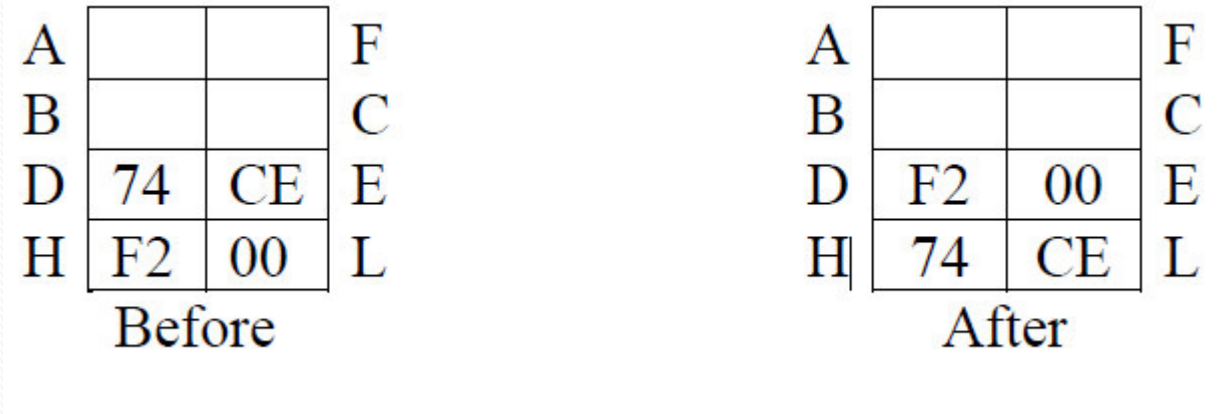
Instruction Set

Exchange H and L with D and E		
XCHG	none	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. **Example:** XCHG

Example

- Swap the content registers pair D with H

XCHG



Instruction Set

Copy H and L registers to the stack pointer		
SPHL	none	The instruction loads the contents of the H and L registers into the stack pointer register, the contents of the H register provide the high-order address and the contents of the L register provide the low-order address. The contents of the H and L registers are not altered. **Example:** SPHL

Example

- Load the content of the HL register pair to stack pointer

  SPHL

Instruction Set

Push register pair onto stack		
PUSH	Reg. pair	The contents of the register pair designated in the operand are copied onto the stack in the following sequence. The stack pointer register is decremented and the contents of the high- order register (B, D, H, A) are copied into that location. The stack pointer register is decremented again and the contents of the low-order register (C, E, L, flags) are copied to that location. **Example:** PUSH B or PUSH A

Example

- Store the content of register pair D in to stack memory

  PUSH D

Instruction Set

Pop off stack to register pair		
POP	Reg. pair	The contents of the memory location pointed out by the stack pointer register are copied to the low-order register (C, E, L, status flags) of the operand. The stack pointer is incremented by 1 and the contents of that memory location are copied to the high-order register (B, D, H, A) of the operand. The stack pointer register is again incremented by 1. **Example:** POP H or POP A

Example

- Load the register pair D from stack memory content.

  POP D

Instruction Set

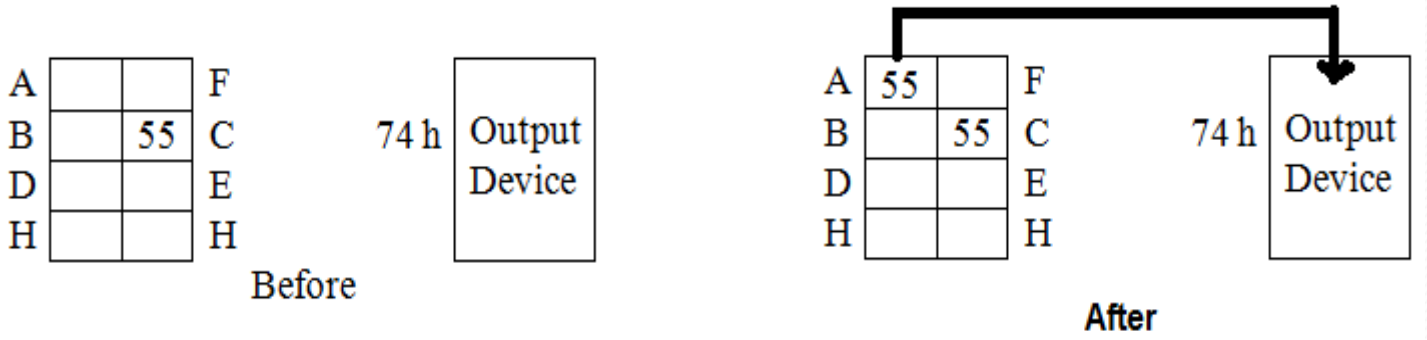
Output data from accumulator to a port with 8-bit address		
OUT	8-bit port address	The contents of the accumulator are copied into the I/O port specified by the operand. **Example:** OUT F8H

Example

- Store the content of register C to output device port 74 h

MOV A, B

OUT 74 h



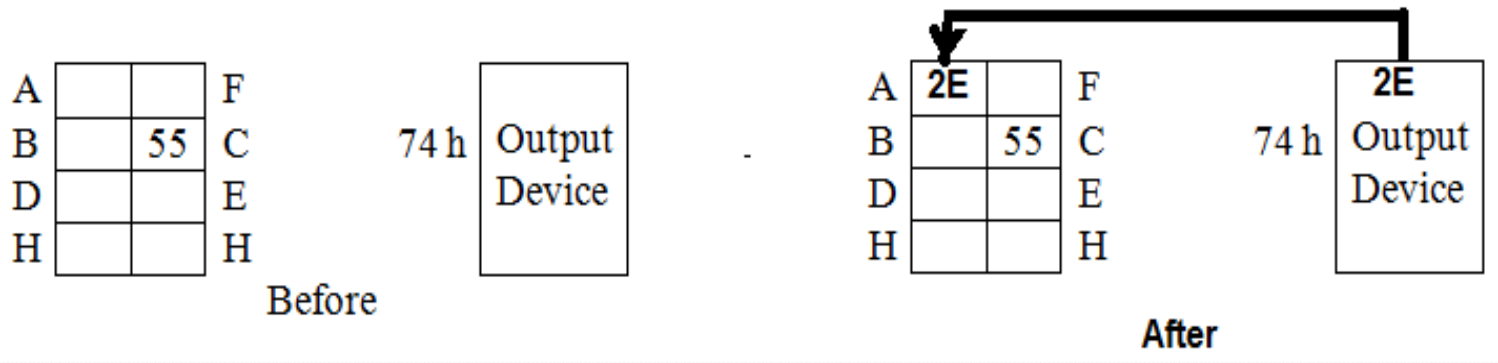
Instruction Set

Input data to accumulator from a port with 8-bit address		
IN	8-bit port address	8-bit port address The contents of the input port designated in the operand are read and loaded into the accumulator. **Example:** IN 8CH

Example

- Load the data from output device port 74 h to accumulator

IN 74 h




The End