Instruction Set

Lecture objectives: at the end of this lecture the student will able to:

1- **Define the instruction and program.**

2- **Classify the instructions set of 8085 microprocessor.**

3- **Understand the assembly and machine languages**

4- **Explain the operation of data transfer instructions in 8085 microprocessor.**

2.1 Definition of instruction and program in 8085 microprocessor:

The second part of microprocessor system is the software. Software is the programs that can be executed by the system.

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***. A 8085 microprocessor has ***246*** instructions. Each instruction is represented by an ***8-bit*** binary value. These 8-bits of binary value is called ***Op-Code*** or ***Instruction Byte***.

A program is some of instructions written in certain method to achieve the certain task or function. The program is the communication method between person and system. Therefore, to write and execute any program by using 8085 microprocessor must studying the instructions that used in this processor.

Any instruction or program in 8085 microprocessor is written in two methods, the first is assembly language and the second is machine language. In assembly language the instruction is written as words such as (**MOV, LXI, ADD C, JMP, PUSH PSW**, etc). The other form of instruction is a binary number which called the machine language, where each instruction has special code consist of two digits written by hexadecimal system called the Opcode such as (**76** which is Opcode of **HLT** instruction, **7E** which is Opcode of **MVI M** instruction, etc).

2.2 Instruction Set Classification

The 8085 microprocessor instructions are classified into five groups as in below:

1- Data transfer group.
2- Arithmetic group.
3- Logic group.
4- Branch group.
5- Stack, I/O, and machine control group.

2.3 Data Transfer Group Instructions:

These instructions move data between registers, or between memory and registers. These instructions copy data from source to destination, where the contents of source are not modified through copying.

Opcode	Operand	Description
MOV	Rd, Rs M, Rs Rd, M	Copy from source to destination.

A- **MOV (destination), (source) [one byte instruction]**

Transferring the contents of source to destination where the contents of source is 8-bit number or two digits in hexadecimal system.

Where Rd (register destination) and Rs (register source) are one of registers (**A, B, C, D, E, H**, and **L**) and $M_{(HL)}$ is the memory location which addressing by contents of registers pair (**HL**).

Examples

MOV A,B transferring the contents of register **B** to register **A**.

MOV C,M transferring the contents of memory location which addressing by contents of registers pair **HL** to register **C.**

MOV M,D transferring the contents of register **D** to memory location which addressing by contents of register pair **HL**.

Opcode	Operand	Description
MVI	Rd, Data M, Data	Move immediate 8-bit

B- **MVI destination, data [two bytes instruction]**

Loading the destination with 8-bit data immediately. Where Rd is one of registers (**A, B, C, D, E, H**, and **L**) and $\mathbf{M_{HL}}$ is the memory location addressing by contents of register pair (**HL**).

Examples

MVI H,34 Loading the register **H** with data **34** immediately.

MVI M,34 Loading the Mem. location which addressing by (**HL**) with data **34** immediately.

Opcode	Operand	Description
LXI	Reg. pair, 16-bit data	Load register pair immediate

C- **LXI destination,(16-bit)Data [three bytes instruction]**

Loading the destination with data 16-bit immediately, where the destination is Registers Pair (**BC, DE, HL**) or 16 bit register **SP.**

LXI B,3456 Loading registers pair **BC** with data **3456** immediately where **34** loaded in register **B** and **56** loaded in register **C.**

Opcode	Operand	Description
LDA	16-bit address	Load Accumulator

D- **LDA 16-bit number** **[three bytes instruction]**

Transferring the contents of memory location which addressed by the operand to accumulator. Operand is (16-bit) number used as address to memory location.

Example

LDA AD21 transferring the contents of memory location which has address **AD21** to **A**

Opcode	Operand	Description
STA	16-bit address	Store accumulator direct

E- **STA 16-bit number** **[three bytes instruction]**

Transferring the contents of accumulator to memory location which addressed by operand. Operand is (16-bit) number used as address to memory location.

Example

STA AD21 transferring (**A**) to memory location which has address **AD21**.

Opcode	Operand	Description
LDAX	B/D Register Pair	Load accumulator indirect

F- **LDAX R.P.** **[one byte instruction]**

Transferring the contents of memory location which addressed by Rp to accumulator. Rp is on of the register pair (**BC, DE**)

Example

LDAX D transferring the contents of memory location which addressing by (**DE**) to **A**

Opcode	Operand	Description
STAX	Reg. pair	Store accumulator indirect

G- **STAX R.P.** **[one byte instruction]**

Transferring the contents accumulator to memory location which addressed by R.P. R.P. is on of the register pair (**BC, DE**)

Example

STAX B transferring (**A**) to memory location which addressing by (**BC**) .

Opcode	Operand	Description
LHLD	16-bit address	Load H-L registers direct

H- **LHLD 16-bit number** **[three bytes instruction]**

Transferring the ($M_{16\text{-bit number}}$) to register **L** and transfer the ($M_{16\text{-bit number}+1}$) to register **H**.

Example

LHLD 2000 transferring (M_{2000}) to register **L** and (M_{2001}) to register **H.**

Opcode	Operand	Description
SHLD	16-bit address	Store H-L registers direct

I- **SHLD 16-bit number** **[three bytes instruction]**

Transferring the (**L**) to (**M**$_{16\text{-bit number}}$) and transfer the (**H**) to (**M**$_{16\text{-bit number}+1}$).

Example

SHLD 2000 transferring (**L**) to (**M**$_{2000}$) and transfer the (**H**) to (**M**$_{2001}$)**.**

Opcode	Operand	Description
XCHG	None	Exchange H-L with D-E

J- XCHG **[one byte instruction]**

Replacement the (**D**) with (**H**) and (**E**) with (**L**).

Opcode	Operand	Description
SPHL	None	Copy H-L pair to the Stack Pointer (SP)

K- SPHL **[one byte instruction]**

This instruction loads the contents of **HL** pair into **SP**.

Opcode	Operand	Description
XTHL	None	Exchange H–L with top of stack

L-XTHL **[one byte instruction]**

The contents of reg. **L** is exchanged with the stack location pointed out by the contents of the SP and the contents of reg. **H** are exchanged with the stack location pointed out by the contents of the (SP + 1).

Opcode	Operand	Description
PCHL	None	Load program counter with H-L contents

M- PCHL **[one byte instruction]**

The contents of registers **H** and **L** are copied into the program counter (**PC**). The contents of **H** are placed as the high-order byte and the contents of **L** as the low-order byte.

Homework:

Transfer the contents of stack memory locations (4000, 4001, 4002, 4003) to the memory locations (2030, 2031, 2032, 2033) [***Note:*** the second location not stack memory].