Lecture 8
Computer Technology

First Grade

2018-2019

Assistant Prof. Dr. Emad I Abdul Kareem

College of Education

Computer Science Department

الجامعة المستنصرية| Mustansiriyah University

Lecture Eight
Machine language

Introduction to Microcomputer Programming

8.1 Instruction

is a unit of information that used to indicate to the computer what operation it is to performs.

8.2 Program

is a set of instructions that are used together to accomplish a complete computational task.

8.3 Instruction types

Because of their limited size, microcomputers do not usually have a sufficient instructions repertoire to be able to perform each of these operations with just one instruction. In fact a measure of the sophistication of a MP is the number of these operations that can be done with only one instruction and the flexibility these instructions have in performing their operations.

1- **Transfer:** reg-reg, mem-reg and transfer onto and off of the stack
2- **Arithmetic:** +,-,*,/, and negation, the types of data can be used to represent the numbers (int, float,BCD)
3- **Logical:** OR, AND, exclusive OR and complement.
4- **Shift and Rotate:** SR,SL,RR,RL, and multiple bytes shifts and rotates.
5- **Indexing and Counting**: incrementing & decrementing.
6- **Bit Manipulation:** selectively setting, clearing, and testing bits within a byte or word.

7- **Looping:** A combination of incrementing or decrementing, comparing, testing, and branching; the purpose being to repetitively execute a program segment.

8- **Branching:** Ordinary instructions are taken form consecutive memory locations, however, it some necessary to jump out of the ordinary sequence of code. The branching instructions perform this function. They fall into three main categories

   i- Unconditional jump

   ii- Conditional jump.

   iii- Subroutines jump and Returns.

9- **I/O communication & Transfer:** for initiating and performing I/O data transfers, testing the I/O status, giving I/O commands, and so on.

8.4 Addressing modes

In order to communicate with their various components, all computers must have some means of identifying the individual external memory locations, stack, CPU register, and I/O interface register. Usually a computer will have separate address spaces for its memory and its CPU registers set, and will sometimes have separate space for it I/O interface registers and stack. This mean that there would be two to four storage unites with address 0; one memory location, one CPU register, and perhaps one I/O device register and one stack or internal memory location. Which of the two to four units is intended would be determined by the instruction.
Because memory is used to store instructions and data, all computers include a variety of methods, called addressing modes for accessing it.

1- **Direct Addressing Mode**: The instruction contains the exact memory address of the data item.

2- **Register Addressing Mode**: the instruction specifies a register or a register pair which the data is located.

3- **Register Indirect Addressing Mode**: The instruction specifies a register pair which contains the memory address where the data is locate.

4- **Immediate Addressing Mode**: The instruction contains the data itself.
5- **Base Addressing Mode**: The address is formed by adding the contents of a memory location or register to a specified number called a *displacement*.
6- **Indexing**: Is the process of increment or decrement on address as the computer sequences through a set of consecutive address.
7- **Auto-incrementing / Auto-decrementing**: A form of indexing in which the index is automatically incremented (or decremented) by the instruction.
8- **Relative Addressing Mode**: The address is the sum of a number and the current contents of the Program Counter.
9- **Page Addressing Mode**: A form of indirect addressing mode in which the specified address is formed from a page address, which determines the high-order bits of the address and a displacement which determines the low-order bits (the page address normally comes from a special register called the "page address register").

The 8085 have four addressing mode for addressing data stored in memory or in registers: (direct, register, register indirect, immediate).

8.5 Instruction Format

All instructions are made up of a sequence of bytes, each byte being a combination of 1's and 0's. The portion of an instruction that specifies what the instruction does is called the ***Operation Code*** **(OP Code)**. Any address or a piece of data that is needed by an instruction in order to complete its execution is called ***Operand***.

There are four type of address instruction

1- Three address instruction: **ADD A,B,C** this mean A=B+C

2- Two address instruction (Double operand instruction): **ADD A,B** this mean A=A+B

3- One address instruction: (Single operand instruction): **ADD B** this mean Acc=Acc+B

4- Zero Address instruction: (Zero operand instruction): Ex: **ADD** this type of instruction is use the stack to get the operands and then store the result again in the stack.

The instruction size must be small as possible to reduce fetch time and the space required to store the instruction. To reduce the number of bits needed to specify the operands is obtained by:

1- Permit only branch instruction to include the address of the next instruction.
2- Using the location of one of the data item being operated on to store the result.

3- Having either one or both operands contained in the CPU registers (register address needs few bits to designate a register address).

OP-Code	Destination	Source

Source: is the operand from which the information is taken.

Destination: is the location that is changed.

A typical Assembly language instruction has the following format:

LABLE: OPERATION-MNEMONIC OPERANDS; COMMENTS