7- Execution of the instruction

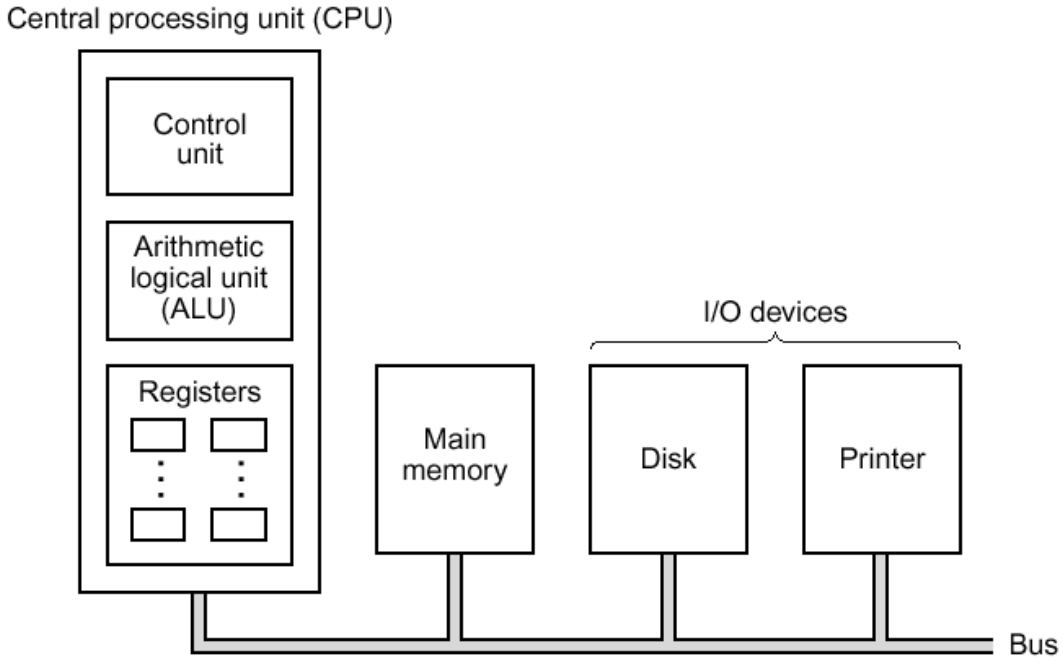
7-1 General Review

An instruction cycle consists of an instruction fetch, followed by zero or more operand fetches, followed by zero or more operand stores, followed by an interrupt check (if interrupts are enabled) .The major computer system components (processor, main memory, I/O modules) need to be interconnected in order to exchange data and control signals. The most popular means on interconnection is the use of a shared system bus consisting on multiple lines.

- **Processor Organization.**

  The aims of the processor are to :

  - Fetching an instruction from memory (through the bus);
  - Fetch the instruction (fetch);
  - Execute the instruction (execute);
  - identify the next instruction ready to be fetched and executed



- **The three components of the processor :**

(1) The Arithmetic and Logical Unit (ALU)

- Made up of circuits that performs the arithmetic and logical execution within the processor.
- It has no internal storage.

(2) The Control Unit (CU) : It contains circuits that direct and coordinate proper sequence, interpret each instruction and apply the proper signals to the ALU and registers.

Note1:

A computer can be programmed by using a small set of basic logic components that store binary data and perform arithmetic and logical operations on data. If a particular computation is to be performed, a configuration of logic components designed specifically for that computation could be constructed. Programming in hardware is termed a **hardwired program**. This customized hardware system accepts data and produces results. This customized hardware system is not very flexible because for each new program, this customized hardware must be rewired. As an alternative, construct a general-purpose configuration of arithmetic and logic functions. This set of hardware will perform various functions on data depending on control signals applied to the hardware. This general-purpose hardware system accepts data and control signals and produces results. For each new program, the programmer merely needs to supply a new set of control signals. How shall control signals be supplied? The entire program is actually a sequence of steps. At each step, some arithmetic or logic operation is performed on some data. For each step, a new set of control signals is needed. Simply provide a unique code for each possible set of control signals, and add to the general-purpose

hardware a segment that can accept a code and generate control signals. Programming is now much easier. Instead of rewiring the hardware for each new program, simply provide a new sequence of codes. Each code is, in effect, an instruction, and part of the hardware interprets each instruction and generates control signals. To distinguish this new method of programming, a sequence of codes or instructions is called **software** (**microprogram**).

(3) The Registers :

- Registers are high speed temporary data storage area within the processor to support execution activities.
- Both instructions and data can be stored in registers for processing by the ALU.
- All processors have a certain number of registers, the exact number varies between different CPUs.
- There are many registers such as:
  - **Instruction Register (IR)**
    - It holds the instruction that is currently being executed.
    - Its output is available to the control unit which generated the timing signals that control the various processing elements involved in executing the instruction.
  - **Program Counter (PC)**
    - It holds the address of the instruction to be fetched and executed.
    - During the execution of an instruction, its contents are updated to point to the next instruction to be executed.

- **Stack Pointer (SP) :** It contains the address of a section of memory known as ***stack*** which may be used for temporary storage of data or addresses.
- **Memory Address Register (MAR) :** It holds the address in memory to or from which data are to be transferred.
- **Memory Data Register (MDR) :**It contains the data to be written into or read out of the addressed memory location.
- **Status Register (SR) / Conditional Code Register (CCR) / Status Flags**
  - A register with individual bits (flags) to indicate condition of the processor as a result of an arithmetic/logical operations.
  - Common status flags:
    - Carry (C)
    - Positive result (P)
    - Zero result (Z)
    - Negative result (N)
    - arithmetic oVerflow (V)

7-2 Fetch and Execute Cycle.

- A sequence of instructions (program) to be executed by a computer are first loaded into consecutive memory locations through the input unit.
- Execution of the program starts when the Program Counter (PC) is set to point to the first instruction.
- The CPU fetches one instruction at a time and performs the specified operation.
- The CPU must keep track of the memory address of the next instruction by the PC.

- The cycle is repeated for the next instruction.
- The processor executes a program in memory is equivalent to execute the program in a series of fetch/execute operations for individual instruction as:

  (i) At the start of the instruction, the PC contains the address of the instruction to be executed.

  (ii) Fetch the instruction pointed to by the PC from memory into Instruction Register (IR).

  (iii) Update PC to point to the following instruction.

  (iv) Decode the instruction by the control unit (CU).

  Fetch phase ends here.

  (v) If the instruction uses data in memory, determine where they are.

  (vi) Fetch data, if any, into registers. (it may be necessary to repeat fetching data, depending on the instruction)

  (vii) Execute the instruction.

  (viii) Store the results in place as specified by the instruction.

  (ix) Go to (i) and execute next instruction.

Note2:

1. The **fetch cycle** (to fetch & decode instruction) is the same for all types of instructions
2. The **execute cycle** is determined by the instruction fetched from memory in the fetch cycle.
3. We can classify the **instruction** based on the nature of PC increment as shown below :