**OPERATING SYSTEM INTRODUCTION**

OS is a program that acts as an intermediary between a user of a computer and the computer hardware

E.g: Windows, Linux

**Operating system goals:**

1. Execute user programs and make solving user problems easier
2. Make the computer system convenient to use
3. Use the computer hardware in an efficient manner

**1.1 COMPUTER SYSTEM OVERVIEW- BASIC ELEMENTS**

At a top level, a computer consists of processor, memory, and I/O components, with one or more modules of each type. These components are interconnected to execute programs.

Four main structural elements:

**Processor**: Controls the operation of the computer and performs its data processing functions. When there is only one processor, it is often referred to as the central processing unit (CPU).

**Main memory**: Stores data and programs. It is typically volatile; that is, when the computer is shut down, the contents of the memory are lost. In contrast, the contents of disk memory are retained even when the computer system is shut down. Main memory is also referred to as real memory or primary memory.

**I/O modules**: Move data between the computer and its external environment. The external environment consists of a variety of devices, including secondary memory devices (e.g., disks), communications equipment, and terminals.

**System bus**: Provides for communication among processors, main memory, and I/O modules.

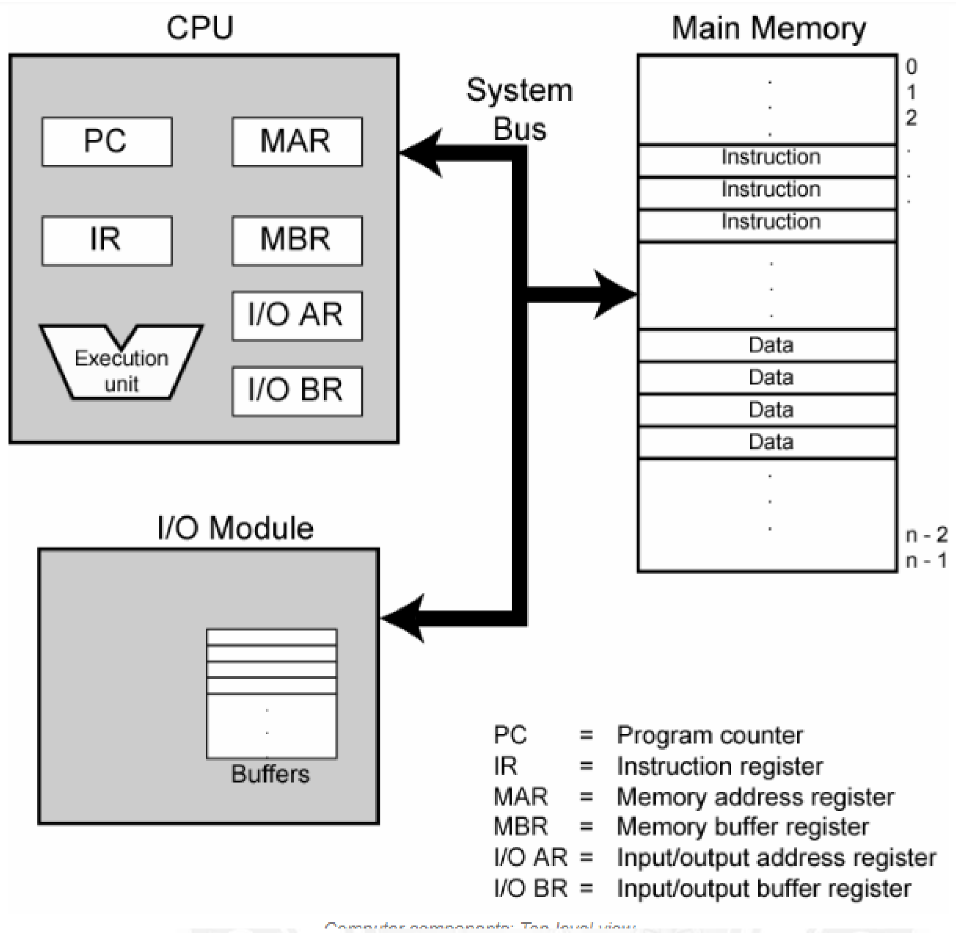


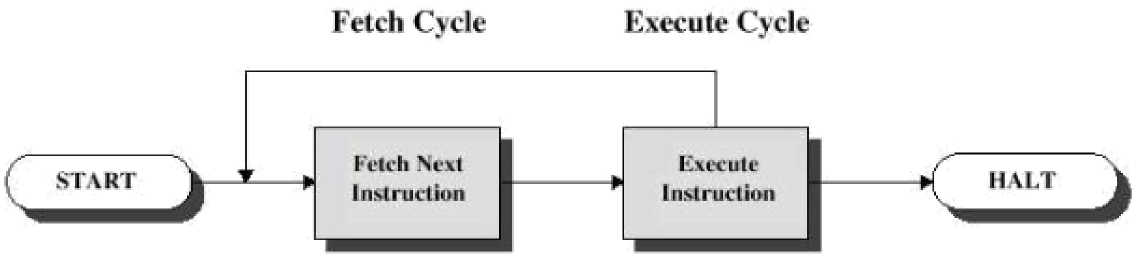

**Fig: Computer Components Top Level View**

- One of the processor's functions is to exchange data with memory. For this purpose, it typically makes use of two internal (to the processor) registers:
- A memory address register (MAR), which specifies the address in memory for the next read or write;
- A memory buffer register (MBR), which contains the data to be written into memory or which receives the data read from memory.
- Similarly, an I/O address register (I/OAR) specifies a particular I/O device.
- An I/O buffer register (I/OBR) is used for the exchange of data between an I/O module and the processor.
- A memory module consists of a set of locations, defined by sequentially numbered addresses. Each location contains a bit pattern that can be interpreted as either an instruction or data.

- An I/O module transfers data from external devices to processor and memory, and vice versa. It contains internal buffers for temporarily holding data until they can be sent on.

## 1.2 Instruction Execution

- A program to be executed by a processor consists of a set of instructions stored in memory.
- Instruction processing consists of two steps: The processor reads ( *fetches* ) instructions from memory one at a time and executes each instruction.
- Program execution consists of repeating the process of instruction fetch and instruction execution.
- The processing required for a single instruction is called an *instruction cycle.*




**Instruction cycle:**

The two steps are referred to as the *fetch stage* and the *execute stage.*

1. At the beginning of each instruction cycle, the processor fetches an instruction from memory.
2. Typically, the program counter (PC) holds the address of the next instruction to be fetched. PC value is incremented after each instruction fetch.
3. The fetched instruction is loaded into the instruction register (IR). The instruction contains bits that specify the action. The processor interprets the instruction and performs the required action.

In general, these actions fall into four categories:

- **Processor-memory:** Data may be transferred from processor to memory or from memory to processor.

• **Processor-I/O:** Data may be transferred to or from a peripheral device by transferring between the processor and an I/O module.

• **Data processing:** The processor may perform some arithmetic or logic operation on data.

• **Control:** An instruction may specify that the sequence of execution be altered.

Figure illustrates a partial program execution, showing the relevant portions of memory and processor registers. The program fragment shown adds the contents of the memory word at address 940 to the contents of the memory word at address 941 and stores the result in the latter location.

**Steps:**

1. The PC contains 300, the address of the first instruction. This instruction (the value 1940 in hexadecimal) is loaded into the IR and the PC is incremented.

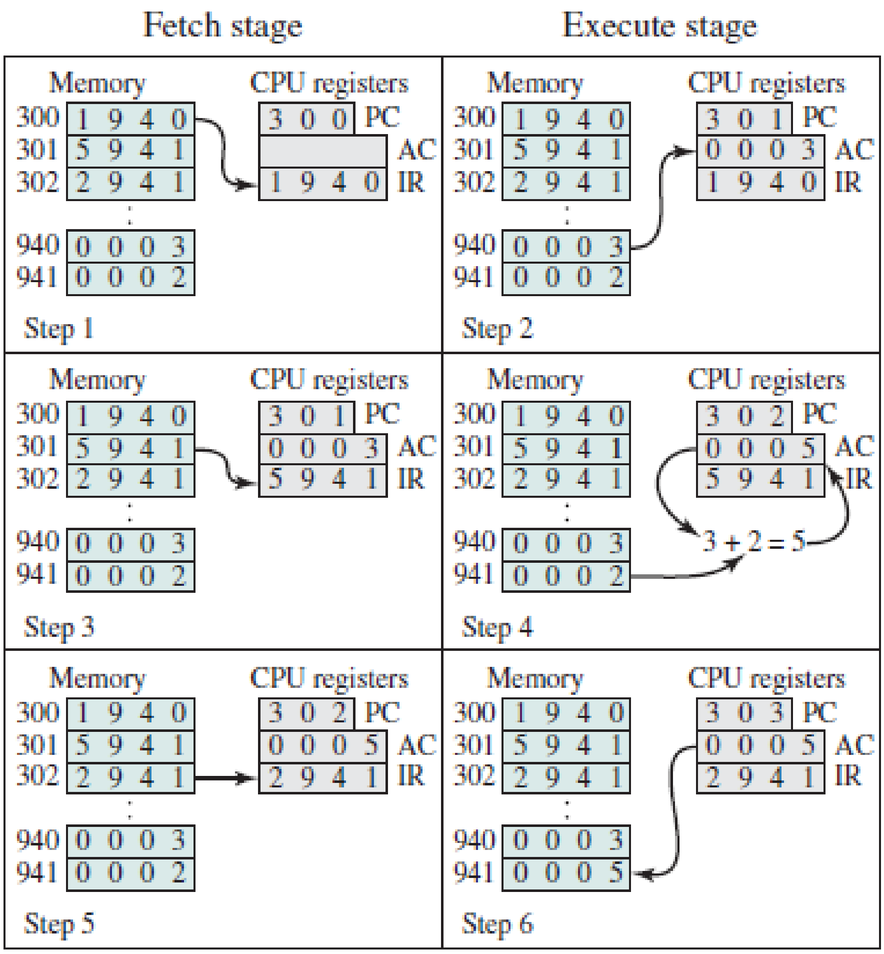



**Figure 1.4** **Example of Program Execution** (contents of memory and registers in hexadecimal)

2. The first 4 bits (first hexadecimal digit) in the IR indicate that the AC is to be loaded from memory. The remaining 12 bits (three hexadecimal digits) specify the address, which is 940.

3. The next instruction (5941) is fetched from location 301 and the PC is incremented.

4. The old contents of the AC and the contents of location 941 are added and the result is stored in the AC.

5. The next instruction (2941) is fetched from location 302 and the PC is incremented.

6. The contents of the AC are stored in location 941.