

Lecture-1

Computer technology has made incredible improvement in the past half century. In the early part of computer evolution, there were no stored-program computer, the computational power was less and on the top of it the size of the computer was a very huge one.

Today, a personal computer has more computational power, more main memory, more disk storage, smaller in size and it is available in affordable cost. This rapid rate of improvement has come both from advances in the technology used to build computers and from innovation in computer design.

The task that the computer designer handles is a complex one: Determine what attributes are important for a new machine, then design a machine to maximize performance while staying within cost constraints. This task has many aspects, including instruction set design, functional organization, logic design, and implementation. While looking for the task for computer design, both the terms computer organization and computer architecture come into picture.

It is difficult to give precise definition for the terms **Computer Organization and Computer Architecture**. But while describing computer system, we come across these terms, and in literature, computer scientists try to make a distinction between these two terms.

Computer architecture refers to those parameters of a computer system that are visible to a programmer or those parameters that have a direct impact on the logical execution of a program. Examples of architectural attributes include the instruction set, the number of bits used to represent different data types, I/O mechanisms, and techniques for addressing memory.

Computer organization refers to the operational units and their interconnections that realize the architectural specifications. Examples of organizational attributes include those hardware details transparent to the programmer, such as control signals, interfaces between the computer and peripherals, and the memory technology used.

Representation of Basic Information

The basic functional units of computer are made of electronics circuit and it works with electrical signal. We provide input to the computer in form of electrical signal and get the output in form of electrical signal.

There are two basic types of electrical signals, namely, analog and digital. The **analog signals** are continuous in nature and **digital signals** are discrete in nature.

The electronic device that works with continuous signals is known as analog device and the electronic device that works with discrete signals is known as digital device.

Since Computer is a digital electronic device, we have to deal with two kinds of electrical signals. But while designing a new computer system or understanding the working principle of computer, it is always difficult to write or work with 0V or 5V. To make it convenient for understanding, we use some logical value, say, LOW (L) - will represent 0V and HIGH (H) - will represent 5V.

Computer is used to solve mainly numerical problems. Again, it is not convenient to work with symbolic representation. For that purpose, we move to numeric representation.

In this convention, we use 0 to represent LOW and 1 to represent HIGH.

0 means LOW

1 means HIGH

To know about the working principle of computer, we use two numeric symbols only namely 0 and 1. All the functionalities of computer can be captured with 0 and 1 and its theoretical background corresponds to two valued boolean algebra.

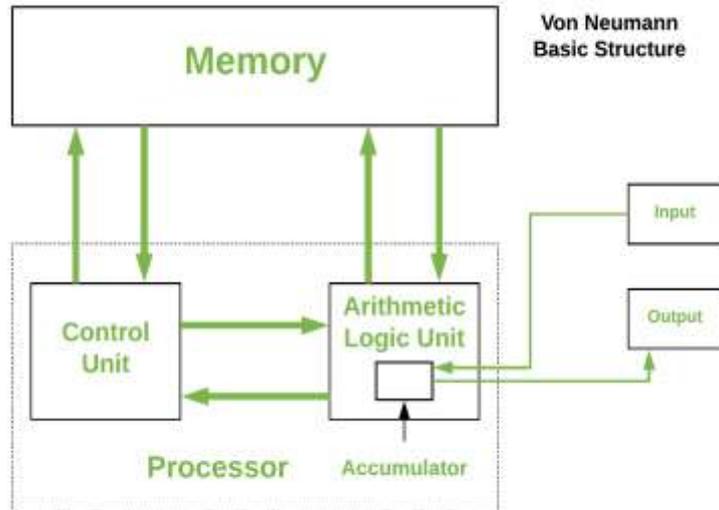
With the symbol 0 and 1, we have a mathematical system, which is known as binary number system. Basically, binary number system is used to represent the information and manipulation of information in computer. This information is basically strings of 0s and 1s. The smallest unit of information that is represented in computer is known as **Bit (Binary Digit)**, which is either 0 or 1. **Four bits** together is known as **Nibble**, and Eight bits together is known as **Byte**.

Main Memory Organization: Stored Program The present-day digital computers are based on stored-program concept introduced by Von Neumann. In this stored-program concept, programs and data are stored in separate storage unit called memories.

1. **Fixed Program Computers** – Their function is very specific and they couldn't be programmed, e.g. Calculators.
2. **Stored Program Computers** – These can be programmed to carry out many different tasks, applications are stored on them, hence the name.

The modern computers are based on a stored-program concept introduced by John Von Neumann. In this stored-program concept, programs and data are stored in a separate storage unit called memories and are treated the same. This novel idea meant that a computer built with this architecture would be much easier to reprogram.

The basic structure is like,



It is also known as **IAS** (Immediate Access Store) computer and is having three basic units:

1. The Central Processing Unit (CPU)
2. The Main Memory Unit
3. The Input/Output Device

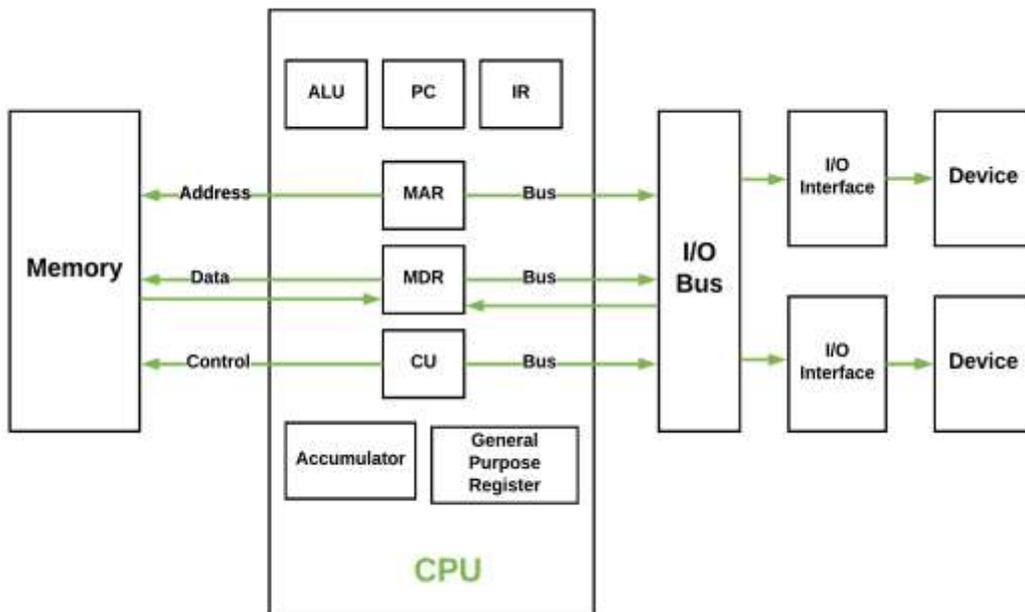
Let's consider them in details.

- **Control Unit –**

A control unit (CU) handles all processor control signals. It directs all input and output flow, fetches code for instructions and controlling how data moves around the system.

- **Arithmetic and Logic Unit (ALU) –**

The arithmetic logic unit is that part of the CPU that handles all the calculations the CPU may need, e.g. Addition, Subtraction, Comparisons. It performs Logical Operations, Bit Shifting Operations, and Arithmetic Operation.



• **Figure – Basic CPU structure, illustrating ALU**

- **Main Memory Unit (Registers) –**
 1. **Accumulator:** Stores the results of calculations made by ALU.
 2. **Program Counter (PC):** Keeps track of the memory location of the next instructions to be dealt with. The PC then passes this next address to Memory Address Register (MAR).
 3. **Memory Address Register (MAR):** It stores the memory locations of instructions that need to be fetched from memory or stored into memory.
 4. **Memory Data Register (MDR):** It stores instructions fetched from memory or any data that is to be transferred to, and stored in, memory.
 5. **Current Instruction Register (CIR):** It stores the most recently fetched instructions while it is waiting to be coded and executed.
 6. **Instruction Buffer Register (IBR):** The instruction that is not to be executed immediately is placed in the instruction buffer register IBR.
- **Input/Output Devices –** Program or data is read into main memory from the *input device* or secondary storage under the control of CPU input instruction. *Output devices* are used to output the information from a computer. If some results are evaluated by computer and it is stored in the computer, then with the help of output devices, we can present it to the user.
- **Buses –** Data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory, by the means of Buses. Types:
 1. **Data Bus:** It carries data among the memory unit, the I/O devices, and the processor.
 2. **Address Bus:** It carries the address of data (not the actual data) between memory and processor.
 3. **Control Bus:** It carries control commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer.

Von Neumann bottleneck –

Whatever we do to enhance performance, we cannot get away from the fact that instructions can only be done one at a time and can only be carried out sequentially. Both of these factors hold back the competence of the CPU. This is commonly referred to as the 'Von Neumann bottleneck'. We can provide a Von Neumann processor with more cache, more RAM, or faster components but if original gains are to be made in CPU performance, then an influential inspection needs to take place of CPU configuration. This architecture is very important and is used in our PCs and even in Super Computers.

- Parallel implementation of program is not allowed due to sequential instruction processing.
- Von Neumann bottleneck – Instructions can only be carried out one at a time and sequentially.
- Risk of an instruction being rewritten due to an error in the program.