A Computer Organisation article on

How Computers Work

By: Himanshu Sharma **Roll Number:** 1610110149 **ECE 3rd Yr.**

Contents

1 Introduction

Computers have become an important part of our life in the modern era. From scientific endeavour to the personal computer revolution, computers are found everywhere. Computers, as we see them today are a result of a prcoess that continues even today to make them better day by day. Early computers were just designed to make complex computations easy. In that sense, a simple calculator is also a computer. But today's computers can do even more. With deveopment in the VLSI technology, today's computer can do what not, it can stream videos, music and even high end quality games. Considering such an important machine, we should atleast know how it functions, and this is what this article aims to provide the reader.

Every computer is made up of various electronic components that work together on a single hardware. The first generation of computers include *ENIAC*. It was the first general purpose digital computer that was designed to solve complex computational problems. It occupied large space and had a weight of 30 tons. It consumed 140 kilowatts of power when it used to run since it heavily depended on the vacuum tubes. The next computer in the sequence of history was the *VON NEUMANN MACHINE*. The Von Neumann machine provided an additional benefit of the **stored program concept**. Imagine this, the ENIAC if shuts down due to some power issues while it is still operating, it would loose all the data with which it started. The Von Neumann machine stores this data beforehand so that feeding it back to the machine manually is not required. In 1946, Neumann started designing another computer which used stored program concept, called the *IAS*. The IAS computer had the following features.

1. ALU - Arithmetic and Logic Unit
2. Control Unit
3. Main Memory
4. Input/Output (I/O)

This basic set of features is followed even today in our computers. Thus almost all the computers today can be technically called the Von Neumann Machines.

Computers evolve based on the technology and thus give rise to a classification based on the generation of the technology used.

1. First Generation - Used vacuum tubes
2. Second Generation - Used transistors
3. Third Generation - Used Small and Medium Scale Integration
4. Fourth Generation - Used Large Scaled Integration
5. Fifth Generation - Very Large Scale Integration
6. Sixth Generation - Ultra Large Scale Integration

From third generation onwards, the computers saw the use of integrated circuits. An example is the *IBM System/360* and *DEC PDP-8*. Microelectronics is the main role player in the computer industry. It opened doors for more complex designs of the computers. This field of electronics made possible a transition from a concept to design - the Microprocessor. The entire computer industry cannot imagine computers without this now. All applications today in which computer finds itself, like image processing, simulation, computer vision, speech recognition, signal processing, etc are made possible by this microprocessor only.

We will now see how a machine with so many features like this actually works. We will see this both on logical and hardware front in this article.

2 Computer Functions and Components

As already discussed, a computer consists of a CPU, control unit, I/O and main memory. These four basic elements of the computer are connected in a fashion such that they are able to execute set of instructions called a program. Executing a program is the main function of any computer system. And this can be achieved in two ways - by hardware or by software. When we say that a computer is programmed via hardware, what we actually mean is that the set of instructions are solved using physical hardware devices like flip-flops, latches, etc which are designed by the designer himself. The advantage of a computer working like this is that the instructions are executed fast but the problem is that changing a program can be cumbersome. Imagine changing dozens of wires and switches just to change a set of parameters in the program. We can solve this problem by using software programming which does not use any hardware but then we are at a loss of speed. It must be noted that both hardware and software mode of programming will give the same results but their way of working and execution speeds are different.

When it comes to reporting the results of the execution, some sort of interface is required. And this is where an I/O comes into the picture. There must also be some place where these results could be stored in the system and that's where the main memory plays its role. A control unit on the other hand decides what data will flow to the CPU and the CPU performs the execution. Most of the times it is the CPU and the main memory which have the interaction with each other. A CPU frequently requests data from the main memory and dumps data into it after execution and therefore to meet these demands, two internal registers called the **Memory Buffer Register** (MBR) and the **Memory Address Register** (MAR) are used by the CPU which contain the data that is to be written into the memory or data which needs to be read from the memory. Similarly, an **I/O Address Register** (I/OAR) is required which addresses the I/O device which is right now being serviced by the processor.

2.1 The execution of a program

The basic function of a computer is to execute a program. This is done in two macro steps, basically, a *fetch* and an *execute*. These macro-steps are followed like a cycle for each statement in a program. In the instruction fetch, the value of the processor register, **program counter** (PC) is loaded into the memory which points to the location of the instruction. After the instruction is loaded, the value of the program counter is incremented by 1. The basic procedure followed is,

1. **if** - Instruction Fetch - Read the instruction from the memory and load it into the processor.
2. **iod** - Instruction Operand Decoding - Decide the type of operation to be performed.
3. **oac** - Operand Address Calculation - Find the location of operand from the memory, if required.
4. **of** - Operand Fetch - Fetch the operand once the location is finalized.
5. **do** - Do Operation - Once the operation is loaded, perform the desired operation.
6. **os** - Operation Store - Write the result into memory.

If there are interrupts while some procedure is under way, then an **Interrupt Service Routine** (ISR) is initiated which ranks the interrupt based on its priority. Interrupts occur mainly because different I/O device request processor attention at any time whenever they require it and the only way to handle is the ISR.

2.2 The System Bus

The system bus is an important part of any computer system because it acts like a highway for different components of the system. A comon analogy is that the system bus is like an interstate highway with the components being the states and the signal flow on the wires like traffic. The system bus is not the only kind of the bus possible, other buses being the *expansion bus* and the *high-speed bus*. Main memory, CPU and the Cache are connected to each other via system bus. Any data that flows between any two of them needs to use the system bus.

3 The Cache Memory

We will now see an important part of a computer system called the Cache Memory. After studying how the execution of program actually happens briefly, we are here at discussing this important part of the computer system. Cache memory is a combination of access time of high speed, less expensive memory and memory with large capacity with less speed. For the processor, cache is the favourite location to look for something. If found, the data is directly transferred to it and if not found, the main memory transfer to the cache memory is requested. The main memory contains 2^n addressable words with each word having n bit address. For mapping puposes to the cache, these 2^n words are divided into K blocks such that there are $M = \frac{2^n}{K}$ sets of lines in each block. The cache conatins m blocks where $m << M$, where m is called *lines*.

There are different mapping functions wherein, the data in the main memory is mapped to cache. These include,

1. Direct Mapping
2. Associative Mapping
3. Set-associative mapping

In direct mapping, the block of main memory is directly mapped to the cache lines. Associative mapping overcomes the disadvantage of direct mapping by permitting each main memory block to be loaded into any line of the cache.

4 The Internal Memory

Internal memory is that part of system's memory which can store small chunks of data when the system is running. This memory is able to store the data only when power is supplied to the system. The two most popular types of internal memories are **Random Access Memory** (RAM) and **Read Only Memory** (ROM).

We should note one thing, the cache is used for data transfer from internal memory to the processor. In this way, the data which is actually stored in the internal memory, gets transferred to the processor via temporary memory called cache memory.

4.1 Semiconductor Main Memory

As the name suggests, the semiconductor main memory is the memory made up of semiconductor devices. The basic element of this type of memory is called the *cell*. A single cell has two stable states, either a 0 or 1 and they can be written into or read from. Data can be written or read from the cells using control signals. A semiconductor main memory is also sometimes referred to as a core.

4.1.1 The Dynamic RAM (DRAM)

The dynamic RAM is made up of electonic devices like capacitors (mainly) and capacitors which are governed by the laws of analog electronics have a tendency to leak charge with time given by the following equation,

$$q(t) = q_o(1 - e^{-t/\tau})$$

where, τ is the time constant measured in seconds. The word *dynamic* used here refers to this decay of charge tendency in the capacitor. A high voltage is represented by logic 1, and a low voltage is represented by logic 0. Logic 1 takes in all the voltages above a threshold voltage and logic 0 takes in all the voltages into its consideration which have a value lower than the threshold voltage. Although we are storing 1 and 0, but DRAM is in itself an analog device because it uses capacitors.

4.1.2 The Static RAM (SRAM)

The static RAM on the other hand is digital in nature because it uses digital devices like flip-flops. It will also hold the data as long as power is supplied to it.

4.1.3 Read Only Memory (ROM)

The ROM, as the name suggests, is a memory device that can be only used to read from and its contents cannot be altered. It does not require any power source to maintain the data. Designing the ROM is a part of fabrication process based on hardwire concept and the data is emmbedded into the ROM at the time of manufacturing process itself. There are three types of ROMs.

1. Programmable ROM (PROM)
2. Erasable PROM (EPROM)
3. Electrically EPROM (EEPROM)

A ROM is an essential part of the computer system because it contains that type of information about the system which should not be altered or played with, like the bootup information which makes possible the system bootup.

5 The CPU

The CPU is that popular part of any computer system that enables it to actually work. When we say how a computer works, we are unconsciously thinking of CPU itself only. The CPU functions by executing machine instructions.