A Computer Organisation article on

How Computers Work

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

Contents

1 Introduction **2**

2 Computer Functions and Components **3**
2.1 The execution of a program . . . 3
2.2 The System Bus . . . 4

3 The Cache Memory **4**

4 The Internal Memory **4**
4.1 Semiconductor Main Memory . . . 4
4.1.1 The Dynamic RAM (DRAM) . . . 5
4.1.2 The Static RAM (SRAM) . . . 5
4.1.3 Read Only Memory (ROM) . . . 5

5 The CPU **5**
5.1 Indirect Cycle . . . 5
5.2 Data Flow . . . 5
5.3 Instruction Pipelining . . . 6
5.3.1 Resource Hazard . . . 6
5.3.2 Data Hazard . . . 6
5.3.3 Control Hazard . . . 6

6 The Control Unit **7**
6.1 The Fetch Cycle . . . 7
6.2 Functions of control Unit . . . 7

7 The Input/Output **8**
7.1 I/O Interface . . . 8
7.2 I/O Transfer Modes . . . 8
7.2.1 Programmed I/O . . . 8
7.2.2 Interrupt Driven I/O . . . 8
7.2.3 Direct Memory Access . . . 8

8 Conclusion **9**

9 References **9**

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,

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

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.

5.1 Indirect Cycle

An indirect cycle occurs when the addressing mode is not direct. At the time of instruction fetch, the instruction field is checked if there is any indirect addressing in the field. If there is any indirect addressing, the correct procedure is followed and finally the address of the instruction is loaded.

5.2 Data Flow

During the fetch cycle, an instruction is read from memory. The PC contains the address of the next instruction to be fetched. This address is moved to the MAR and placed on the address bus. The control unit requests a memory read, and the result is placed on the data bus and copied into the MBR and then moved to the IR. Meanwhile, the PC is incremented by 1, preparatory for the next

fetch. After the fetch cycle is over, th instruction register is referenced to seek the operand. If the operand address could be obtained by indirect addressing, then that is performed.

The fetch and indirect cycles are simple and predictable. The execute cycle takes many forms; the form depends on which of the various machine instructions is in the IR. This cycle may involve transferring data among registers, read or write from memory or I/O, and/or the invocation of the ALU. An idle processor is a waste of money and power. Utilizing every second is important when dealing with a computer system. Therefore, a new term called **instruction pipelining** is introduced.

5.3 Instruction Pipelining

Let me give an analogy of pipelining. When inside a manufacturing plant, different stages of the factory work at different levels. Let us take an example of automobile manufacturing industry. Stage i of the company might be designing the frame of the car and stage j may then white wash it. In the meantime, stage i will not stay idle and will start working on the next car frame. This is the essence of pipelining. Similar things happen in microprocessors. They don't sit idle. You can think of microprocessor as a stage which does not remain idle even after its work is over. It hires the next incoming task. This way a computer's processor saves time.

There are six stages in an instruction pipeline.

1. **FI** - Fetch Instruction
2. **DI** - Decode Instruction
3. **CO** - Calculate Opearands
4. **FO** - Fetch Operands
5. **EI** - Execute Instruction
6. **WO** - Write Operand

There can be a case that an execute instruction evaluates to a branch instruction and then the pipeline hazards start to occur. These resource hazards include,

5.3.1 Resource Hazard

A resource hazard occur when two or more instructions on the same pipeline content for the same resource.

5.3.2 Data Hazard

When there is a conflict in the location of data in the pipeline, a data hazard occurs.

5.3.3 Control Hazard

A control hazard, also known as a branch hazard, occurs when the pipeline makes the wrong decision on a branch prediction and therefore brings instructions into the pipeline that must subsequently be discarded.

Instruction pipelining is an important concept in the study of microprocessors. Study of microprocessors is important because it tells us how the computer's brain work.

6 The Control Unit

With so many things working in a computer together, some master is needed who will take control of what's happening in the system and that's where a control unit comes. Instructions are executed using micro-operations like fetch, indirect and execute. The control unit outputs signals to the processor which actually controls the operations. It makes the data to move from one register to another register and activates specific ALU operations. The input to the control units include inputs from instruction register that contains the opcode of the instruction, flag inputs which determine the state of the CPU after previous executions and the control signals from the control bus.

A control unit (CU) (or controller, same thing) is a piece of hardware that manages the activities of peripherals (separate devices attached to the computer, such as monitors, hard drives, printers, etc.) Control units found on personal computers are usually contained on a single printed circuit board. The control unit acts as a sort of "go-between," executing transfers of information between the computer's memory and the peripheral. Although the CPU (central processing unit-the "big boss" in the computer) gives instructions to the controller, it is the control unit itself that performs the actual physical transfer of data. The control unit fetches one or more new instructions from memory (or an instruction cache), decodes them and dispatches them to the appropriate functional units to be executed. The control unit is also responsible for setting the latches in various data paths that ensure that the instructions are performed on the correct operand values stored in the registers. In a CISC processor, the control unit is a small processor in its own right that executes microcode programs stored in a region of rom that prescribe the correct sequence of latches and data transfers for each type of macroinstruction. A RISC processor does away with microcode and most of the complexity in the control unit, which is left with little more to do than decode the instructions and turn on the appropriate functional units.

6.1 The Fetch Cycle

The fetch cycle of a control unit include 4 registers.

1. Memory Address Register - Connected to the address bus.
2. Memory Buffer Register - Connected to the data bus.
3. Program Counter - Holds the address for the next instruction to be fetched.
4. Instruction Register - Holds last instruction fetched.

Address from the PC is transferred to data bus and then it is copied to the MBR. After that, data is moved from MBR to IR and MBR is now free for further data fetches.

6.2 Functions of control Unit

1. Regulate transfers of information between memory and I/O. Fetches and decodes instructions from microprograms.
2. Responsible for correct instruction execution between a processor's many sub-units.
3. Control unit converts received information into sequence of control signals, and transfer to computer processor.
4. It controls data flow inside the computer processor.

7 The Input/Output

The devices which are used to input the data and the programs in the computer are known as "Input Devices". or Input device can read data and convert them to a form that a computer can use. Output Device can produce the final product of machine processing into a form usable by humans. It provides man to machine communication. Peripherals connected to a computer need special communication links for interfacing with CPU.

7.1 I/O Interface

Interface is a shared boundary btween two separate components of the computer system which can be used to attach two or more components to the system for communication purposes.

Peripherals connected to a computer need special communication links for interfacing with CPU. In computer system, there are special hardware components between the CPU and peripherals to control or manage the input-output transfers. These components are called input-output interface units because they provide communication links between processor bus and peripherals. They provide a method for transferring information between internal system and input-output devices.

7.2 I/O Transfer Modes

7.2.1 Programmed I/O

Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program. Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.

7.2.2 Interrupt Driven I/O

n the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly. This problem can be overcome by using interrupt initiated I/O. In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt. After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

7.2.3 Direct Memory Access

Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as DMA. In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.

Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.

8 Conclusion

A computer system is a wonderful machine. No one could have ever imagined that human race could have built something like this in the early 1900s, a machine that could perform complex task in the blink of an eye. As we discussed above, a computer system is made up of;

1. Memory
2. Processor
3. Control Unit
4. I/O interface

Within memory, we saw how different kinds of memories are there, and here we have only discussed about cache and the internal memory and not the external memory. We discussed about cache blocks and DRAM and SRAM concepts and why DRAM is an analog memory device whereas SRAM is digital. Capacitor charge leak was also described in brief. These memory elements make a computer work efficiently because they store dynamic data when the computer is running, like, when you play games on computer, temporary information is stored in RAM. We also discussed about ROM and how it is useful for storing computer information which might be required for, say, booting up the system, etc.

In CPU, we discussed how an idle processor is a waste of resources and how a computer keeps its proccessor continuously busy by using pipelining. We also saw about indirect cycle and how data flow occurs inside a CPU.

Then we discussed about the control unit and how it manages the control of the system and finally we ended this article by discussing about the I/O devices which actually give meaning to a computer because if they are not there, then there is no means to interact with that dumb machine.

9 References

Computer Organisation and Architecture, William Stallings, 9th Edition