Introduction to Computer System

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. In present days most of the computers are digital in nature and we will deal with Digital Computer in this course.

Computer is a digital device, which works on two levels of signal. We say these two levels of signal as High and Low. The High-level signal basically corresponds to some high-level signal (say 5 Volt or 12 Volt) and Low-level signal basically corresponds to Low-level signal (say 0 Volt). This is one convention, which is known as positive logic. There are others convention also like negative logic.

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 knows 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.

Computer Organization and Architecture

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. In this course we will mainly deal with the innovation in computer design.

The task that the computer designer handles is a complex one: Determine what attributes are important for a new machine, and 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.

In this course we will touch upon all those factors and finally come up with the concept how these attributes contribute to build a complete computer system.

Basic Computer Model and different units of Computer

The model of a computer can be described by four basic units in high level abstraction which is shown in figure 1.1. These basic units are:

- Central Processor Unit
- Input Unit
- Output Unit
- Memory Unit

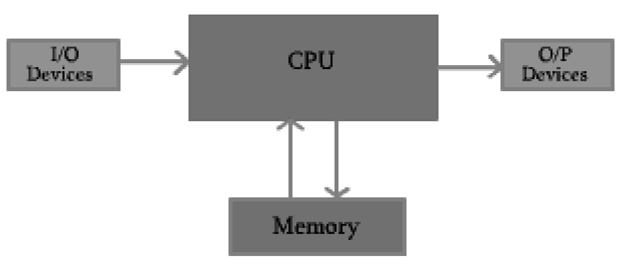


Figure 1.1: Basic Unit of a Computer

Basic Computer Model and different units of Computer

A. Central Processor Unit (CPU):

Central processor unit consists of two basic blocks:

- The program control unit has a set of registers and control circuit to generate control signals.
- The execution unit or data processing unit contains a set of registers for storing data and an Arithmetic and Logic Unit (ALU) for execution of arithmetic and logical operations.

In addition, CPU may have some additional registers for temporary storage of data.

B. Input Unit:

With the help of input unit data from outside can be supplied to the computer. Program or data is read into main storage from input device or secondary storage under the control of CPU input instruction.

Example of input devices: Keyboard, Mouse, Hard disk, Floppy disk, CD-ROM drive etc.

C. Output Unit:

With the help of output unit computer results can be provided to the user or it can be stored in stograge device permanently for future use. Output data from main storage go to output device under the control of CPU output instructions.

Example of output devices: Printer, Monitor, Plotter, Hard Disk, Floppy Disk etc.

D. Memory Unit:

Memory unit is used to store the data and program. CPU can work with the information stored in memory unit. This memory unit is termed as primary memory or main memory module. These are basically semi conductor memories.

There are two types of semiconductor memories -

- Volatile Memory : RAM (Random Access Memory).
- Non-Volatile Memory : ROM (Read only Memory), PROM (Programmable ROM)
  EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM).

Secondary Memory:

There is another kind of storage device, apart from primary or main memory, which is known as secondary memory. Secondary memories are non volatile memory and it is used for permanent storage of data and program.

Example of secondary memories:

Hard Disk, Floppy Disk, Magenetic Tape	------	These are magnetic devices,
CD-ROM	------	is optical device
Thumb drive (or pen drive)	------	is semiconductor memory.

Basic Working Principle of a Computer

Before going into the details of working principle of a computer, we will analyze how computers work with the help of a small hypothetical computer.

In this small computer, we do not consider about Input and Output unit. We will consider only CPU and memory module. Assume that somehow we have stored the program and data into main memory. We will see how CPU can perform the job depending on the program stored in main memory.

P.S. - Our assumption is that students understand common terms like program, CPU, memory etc. without knowing the exact details.

Consider the Arithmetic and Logic Unit (ALU) of Central Processing Unit:

Consider an ALU which can perform four arithmetic operations and four logical operations. To distinguish between arithmetic and logical operation, we may use a signal line,

0 - in that signal, Represents an arithmetic operation and
1 - in that signal, Represents a logical operation.

In the similar manner, we need another two signal lines to distinguish between four arithmetic operations.

The different operations and their binary code are as follows:

Arithmatic		Logical	
000	ADD	100	OR
001	SUB	101	AND
010	MULT	110	NAND
011	DIV	111	NOR

Consider the part of control unit; its task is to generate the appropriate signal at right moment.

There is an instruction decoder in CPU which decodes this information in such a way that computer can perform the desired task

The simple model for the decoder may be considered that there is three input lines to the decoder and correspondingly it generates eight output lines. Depending on input combination only one of the output signals will be generated and it is used to indicate the corresponding operation of ALU.

In our simple model, we use three storage units in CPU,

Two -- for storing the operand and
one -- for storing the results.

These storage units are known as register.

But in computer, we need more storage space for proper functioning of the Computer.

Some of them are inside CPU, which are known as register. Other bigger chunk of storage space is known as primary memory or main memory. The CPU can work with the information available in main memory only.

To access the data from memory, we need two special registers one is known as Memory Data Register (MDR) and the second one is Memory Address Register (MAR).

Data and program is stored in main memory. While executing a program, CPU brings instruction and data from main memory, performs the tasks as per the instruction fetch from the memory. After completion of operation, CPU stores the result back into the memory.

Main Memory Organization

Main memory unit is the storage unit; there are several locations for storing information in the main memory module.

The capacity of a memory module is specified by the number of memory location and the information stored in each location.

A memory module of capacity 16 X 4 indicates that, there are 16 location in the memory module and in each location, we can store 4 bit of information.

We have to know how to indicate or point to a specific memory location. This is done by address of the memory location.

We need two operations to work with memory.

READ Operation: This operation is to retrieve the data from memory and bring it to CPU register
WRITE Operation: This operation is to store the data to a memory location from CPU register

We need some mechanism to distinguish these two operations READ and WRITE.

With the help of one signal line, we can differentiate these two operations. If the content of this signal line is

0, we say that we will do a READ operation; and if it is
1, then it is a WRITE operation.

To transfer the data from CPU to memory module and vice-versa, we need some connection. This is termed as DATA BUS.

The size of the data bus indicates how many bit we can transfer at a time. Size of data bus is mainly specified by the data storage capacity of each location of memory module.

We have to resolve the issues how to specify a particular memory location where we want to store our data or from where we want to retrieve the data.

This can be done by the memory address. Each location can be specified with the help of a binary address.

If we use 4 signal lines, we have 16 different combinations in these four lines, provided we use two signal values only (say 0 and 1).

To distinguish 16 locations, we need four signal lines. These signal lines use to identify a memory location is termed as ADDRESS BUS. Size of address bus depends on the memory size. For a memory module of capacity of 2^n location, we need *n* address lines, that is, an address bus of size *n*.

We use a address decoder to decode the address that are present in address bus

As for example, consider a memory module of 16 location and each location can store 4 bit of information

The size of address bus is 4 bit and the size of the data bus is 4 bit
The size of address decoder is 4 X 16.

There is a control signal named R/W.

If R/W = 0, we perform a READ operation and
if R/W = 1, we perform a WRITE operation

If the contents of address bus is 0101 and contents of data bus is 1100 and R/W = 1, then 1100 will be written in location 5.

If the contents of address bus is 1011 and R/W=0, then the contents of location 1011 will be placed in data bus.

Memory Instruction

We need some more instruction to work with the computer. Apart from the instruction needed to perform task inside CPU, we need some more instructions for data transfer from main memory to CPU and vice versa. In our hypothetical machine, we use three signal lines to identify a particular instruction. If we want to include more instruction, we need additional signal lines.

Instruction	Code	Meaning
1000	LDAI imm	Load register A with data that is given in the program
1001	LDAA addr	Load register A with data from memory location addr
1010	LDBI imm	Load register B with data
1011	LDBA addr	Load register B with data from memory location addr
1100	STC addr	Store the value of register C in memory location addr
1101	HALT	Stop the execution
1110	NOP	No operation
1111	NOP	No operation

With this additional signal line, we can go up to 16 instructions. When the signal of this new line is 0, it will indicate the ALU operation. For signal value equal to 1, it will indicate 8 new instructions. So, we can design 8 new memory access instructions.

We have added 6 new instructions. Still two codes are unused, which can be used for other purposes. We show it as NOP means No Operation.

We have seen that for ALU operation, instruction decoder generated the signal for appropriate ALU operation.

Apart from that we need many more signals for proper functioning of the computer. Therefore, we need a module, which is known as control unit, and it is a part of CPU. The control unit is responsible to generate the appropriate signal.

As for example, for LDAI instruction, control unit must generate a signal which enables the register A to store in data into register A.

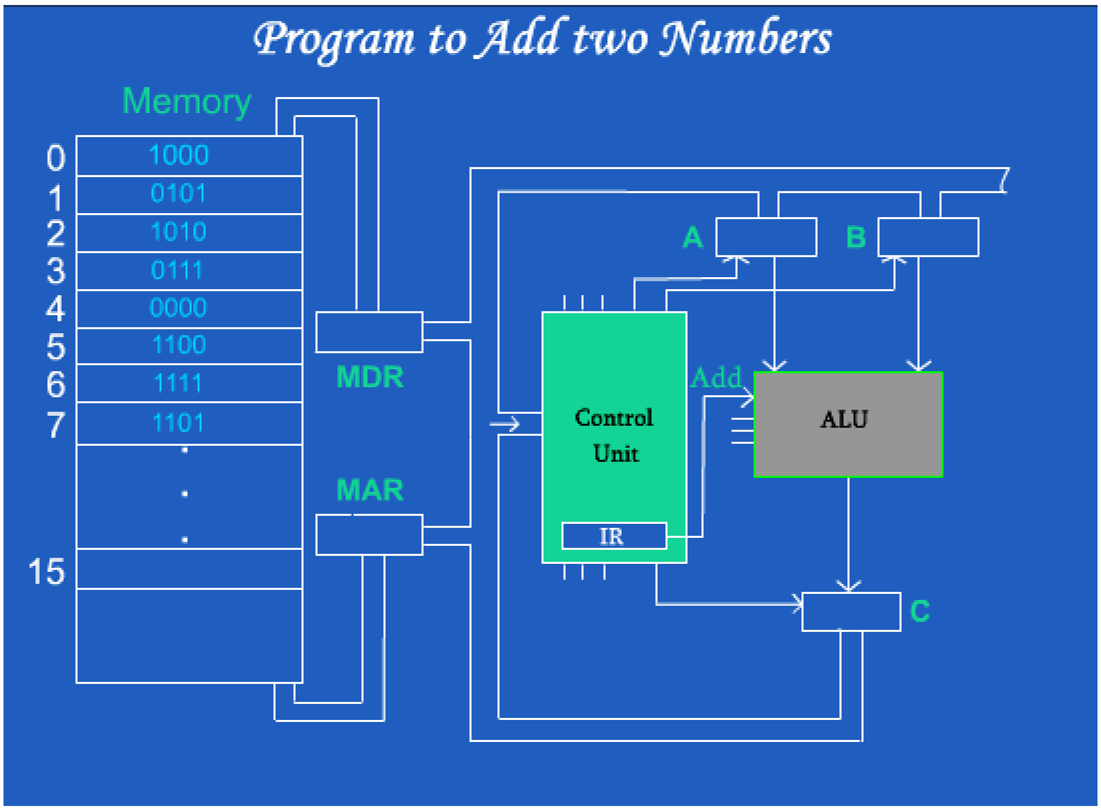
One major task is to design the control unit to generate the appropriate signal at appropriate time for the proper functioning of the computer.

Consider a simple problem to add two numbers and store the result in memory, say we want to add 7 to 5.

To solve this problem in computer, we have to write a computer program. The program is machine specific, and it is related to the instruction set of the machine.

For our hypothetical machine, the program is as follows

Instruction	Binary	HEX	Memory Location
LDAI 5	1000 0101	8 5	(0, 1)
LDBI 7	1010 0111	A 7	(2, 3)
ADD	0000	0	(4)
STC 15	1100 1111	C F	(5, 6)
HALT	1101	D	(7)



Consider another example, say that the first number is stored in memory location 13 and the second data is stored in memory location 14. Write a program to Add the contents of memory location 13 and 14 and store the result in memory location 15.

Instruction	Binary	HEX	Memory Location
LDAA 13	1000 0101	8 5	(0, 1)
LDBA 14	1010 0111	A 7	(2, 3)
ADD	0000	0	(4)
STC 15	1100 1111	C F	(5, 6)
HALT	1101	D	(7)

One question still remains unanswered: How to store the program or data to main memory. Once we put the program and data in main memory, then only CPU can execute the program. For that we need some more instructions.

We need some instructions to perform the input tasks. These instructions are responsible to provide the input data from input devices and store them in main memory. For example instructions are needed to take input from keyboard.

We need some other instructions to perform the output tasks. These instructions are responsible to provide the result to output devices. For example, instructions are needed to send the result to printer.

We have seen that number of instructions that can be provided in a computer depends on the signal lines that are used to provide the instruction, which is basically the size of the storage devices of the computer.

For uniformity, we use same size for all storage space, which are known as register. If we work with a 16-bit machine, total instructions that can be implemented are 216.

The model that we have described here is known as Von Neumann Stored Program Concept. First we have to store all the instruction of a program in main memory, and CPU can work with the contents that are stored in main memory. Instructions are executed one after another.

We have explained the concept of computer in very high level abstraction by omitting most of the details.