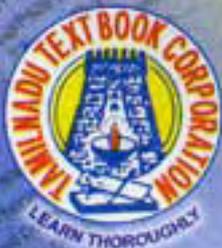
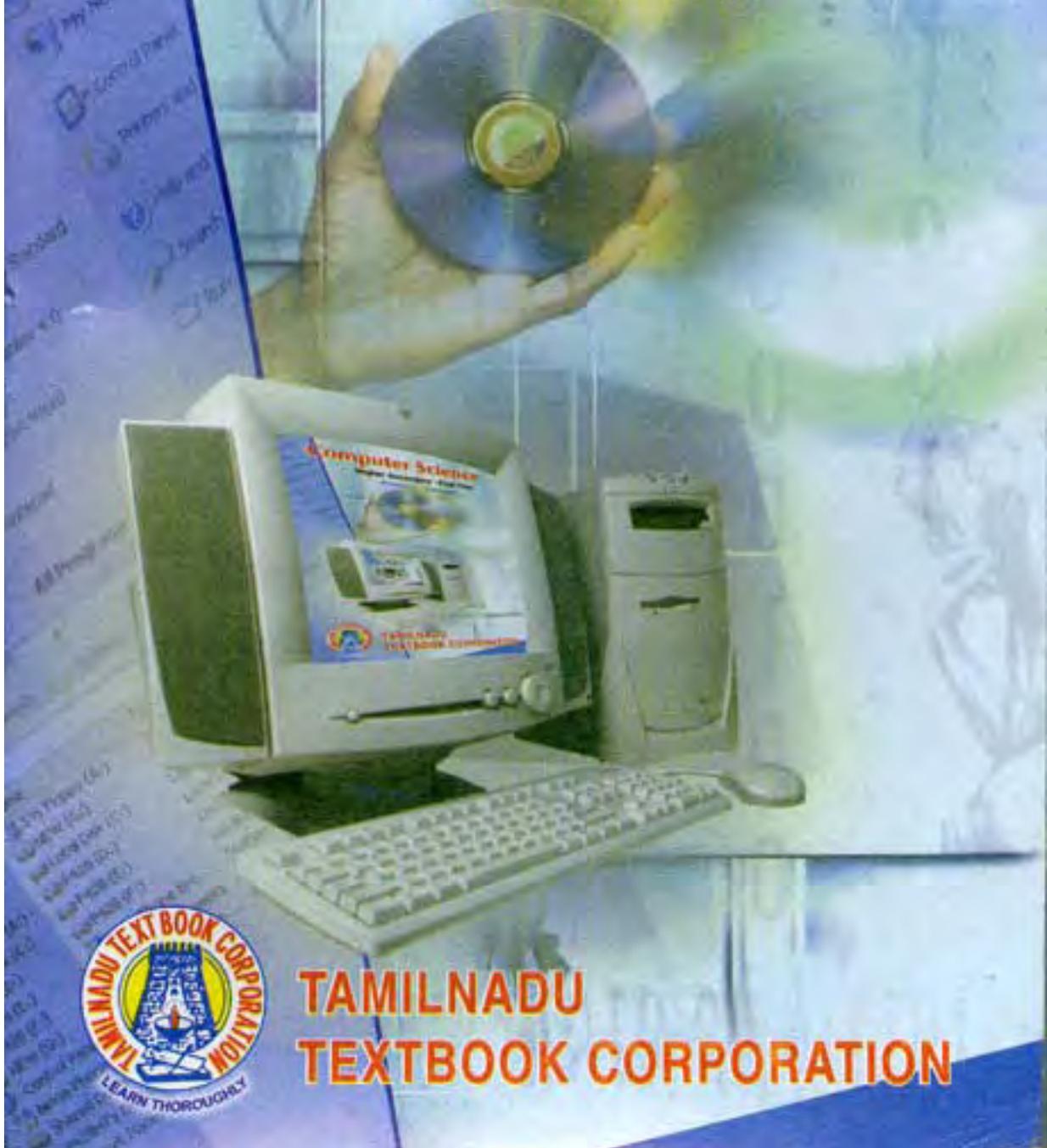


# Computer Science

Higher Secondary - First Year

Volume - I Concepts



**TAMILNADU  
TEXTBOOK CORPORATION**

# **Computer Science**

**PLUS I**

**Volume 1 : Concepts**

**Government of Tamilnadu**

© Government of Tamilnadu  
First Edition – 2005

***Chairman Syllabus Committee***

Dr. Balagurusamy E, Vice Chancellor, Anna University, Chennai

***Co-Ordinator Textbook Writing***

Dr. Sankaranarayanan V, Director, Tamil Virtual University, Chennai

***Authors***

Dr. Elango S, Government Arts College, Nandanam, Chennai

Dr. Jothi A, Former Professor, Presidency College, Chennai

Mr. Malaiarasu P , Government Arts College, Nandanam, Chennai

Dr. Ramachandran V, Anna University, Chennai

Dr. Rhymend Uthariaraj V, Anna University, Chennai

***Reviewers***

Dr. Gopal T V, Anna University, Chennai

Dr. Krishnamoorthy V, Crescent Engineering College, Chennai

***Copy-Editor***

Ms. Subha Ravi, Director, M/s Digiterati Consultancy Pvt. Ltd,  
Chennai

***Cover Design***

Mr. Madan, Free Lance Graphics Designer

***Price Rs. :***

This book has been prepared by the Directorate of School Education on behalf of the Government of Tamilnadu
---

This book has been printed on 70 G.S.M. Paper
---

## FOREWORD

A computer allows users to store and process information quickly and automatically. A computer is a programmable machine. It allows the user to store all sorts of information and then 'process' that information, or data, or carry out actions with the information, such as calculating numbers or organizing words.

These features of computer make it a valuable tool in the hands of users. Computers make life easy. Users can focus their attention on solving more complex problems leaving out the routine activities that can be computerized. The creative abilities of the users can thus be used more effectively. The users have to utilize this powerful tool for the benefit of individuals, organizations, nations and the world.

Computers cannot do things on their own. The users must understand the ways in which problems can be solved with computers. This volume contains the basic concepts required for you to become a user of the computer. This volume

1. Introduces the key components of a computer system (hardware, software, data)
2. Familiarizes students with how computers work through an introduction to number systems
3. Presents the basic concepts of various logic gates that make a computer
4. Gives a broad view of how technology is improving communications through the use of electronic mail and the Internet.

No previous computer related experience is required to understand the concepts contained in this volume.

The field of computers is fast changing. It is the understanding of the basic concepts that will help the users in adjusting to the rapid changes. Without the conceptual basis, the user will find it very difficult to take advantage of the advances in the field of computer science.

Knowing the basic concepts will help the users quickly understand new developments on their own. Hence, the students must focus on understanding the contents of this volume.

The authors, reviewers and editors of this volume have taken great care in ensuring the accuracy of the contents. The presentation is lucid with many illustrations.

I wish the budding computer scientists a fruitful experience with the powerful tool called computer for the rest of their careers.

**(E BALAGURUSAMY)**

Vice Chancellor, Anna University, Chennai  
Chairman Syllabus Committee

# CONTENTS

<b>Chapter 1</b>	<b>INTRODUCTION TO COMPUTERS</b>	<b>1</b>
1.1	History of Computers	1
1.2	Data, Information and Program	8
1.3	Hardware and Software	10
1.4	Types of Computers	15
	Summary	21
	Exercises	22
<b>Chapter 2</b>	<b>NUMBER SYSTEMS</b>	<b>25</b>
2.1	Introduction	25
2.2	Bits and Bytes	26
2.3	Decimal Number System	27
2.4	Binary Number System	28
2.5	Hexadecimal Number System	29
2.6	Decimal to Binary Conversion	30
2.7	Conversion of fractional decimal to binary	34
2.8	Conversion of Decimal to Hexadecimal	35
2.9	Octal Representation	36
2.10	Representation of signed numbers	37
2.11	Binary Arithmetic	42
2.12	Boolean Algebra	48
	Exercises	61
<b>Chapter 3</b>	<b>COMPUTER ORGANIZATION</b>	<b>64</b>
3.1	Basic Components of a Digital Computer	64
3.2	Central Processing Unit	68
3.3	Arithmetic and Logic Unit – ALU	72
3.4	Memory Unit	74
3.5	Input and Output Devices	78
	Summary	96
	Exercises	98

<b>Chapter 4</b>	<b>WORKING PRINCIPLE OF DIGITAL LOGIC</b>	<b>101</b>
4.1	Logic Gates	101
4.2	Conversion of Boolean Function	115
4.3	Half Adder	122
4.4	Full Adder	124
4.5	The Flip-Flop	127
4.6	Electronic Workbench	131
	Summary	151
	Exercises	152
<b>Chapter 5</b>	<b>OPERATING SYSTEMS</b>	<b>155</b>
5.1	Introduction	155
5.2	Major Features of the Operating System	160
5.3	Most Desirable Characters of the Operating System	162
	Summary	168
	Exercises	169
<b>Chapter 6</b>	<b>COMPUTER COMMUNICATIONS</b>	<b>171</b>
6.1	Introduction	171
6.2	Network	171
6.3	Some Important Reasons for Networking	171
6.4	Applications of Network	172
6.5	Benefits of Network	172
6.6	Types of Network	173
6.7	Network Topology	174
6.8	Basics of Networking	176
6.9	Common Network Services	177
6.10	Co-Ordinating Data Communication	179
6.11	Forms of Data Transmission	180
6.12	Modem	181
6.13	Data Transfer Rate	182

6.14	Transmission Mode	182
6.15	Internet	183
6.16	Communication Protocol	184
6.17	Who Governs the Internet ?	184
6.18	Future of Internet	185
6.19	Uses of Internet	185
6.20	Getting Connected to Internet	187
6.21	Popular Uses of the Web	190
6.22	Intranet and Extranet	191
	Exercises	191

# CHAPTER 1

## INTRODUCTION TO COMPUTERS

### 1.1 History of Computers

#### 1.1.1 Introduction

A computer is a tool and partner in every sphere of human life and activity. Computers are bringing many changes in industry, government, education, medicine, scientific research, law, social service and even arts like music, movies and paintings. The areas of application of computers are confined only by the limitation on creativity and imagination.

What is a computer? A child might define a computer to be an instrument capable of producing a combined effect of radio, movie and television. This definition is close but still does not visualize the power and capabilities of a computer.



**Fig. 1.1 Computer**

A computer is an electronic machine, capable of performing basic operations like addition, subtraction, multiplication, division, etc. The computer is also capable of storing information, which can

be used later. It can process millions of instructions in a few seconds and at the same time with high accuracy. Hence a computer can be defined as an automatic electronic machine for performing calculations or controlling operations that are expressible in numerical or logical terms. Computers are very accurate and save time by performing the assigned task very fast. They don't get bored.

Humans have always needed to perform arithmetic like counting and adding. During the pre-historic period, they counted either on their fingers or by scratching marks on the bones and then with the help of stone, pebble and beads. The early civilization had witnessed men develop number systems to keep track of the astronomical cycles, businesses, etc. The word 'computing' means 'an act of calculating'. After the invention of the manual calculating tools, the concept of using 'electronic gadgets' for computations were introduced which gave birth to the computers. The evolution of computers has passed through a number of stages before reaching the present state of development. During the early development period, certain machines had been developed and a brief note of them is given below.

### 1.1.2 Early History

#### 2500 BC – The Abacus

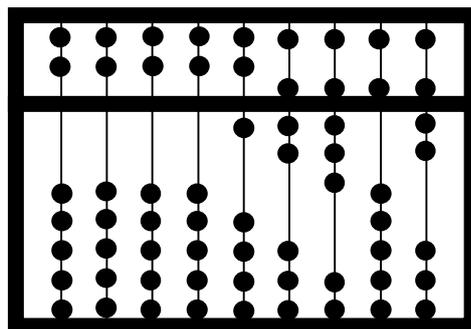
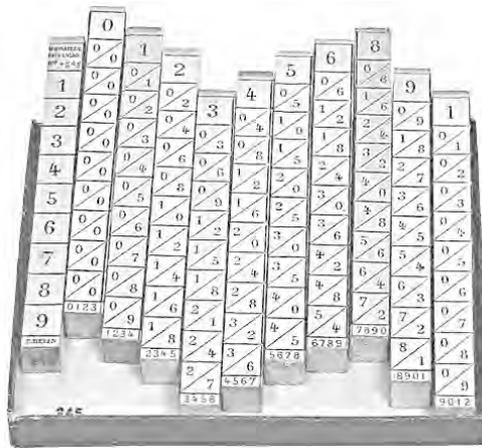


Fig. 1.2 Abacus

Abacus is the first known calculating machine used for counting. It is made of beads strung on cords and is used for simple arithmetic calculations. The cords correspond to positions of decimal digits. The beads represent digits. Numbers are represented by beads close to the crossbar. Abacus was mainly used for addition and subtraction and later for division and multiplication.

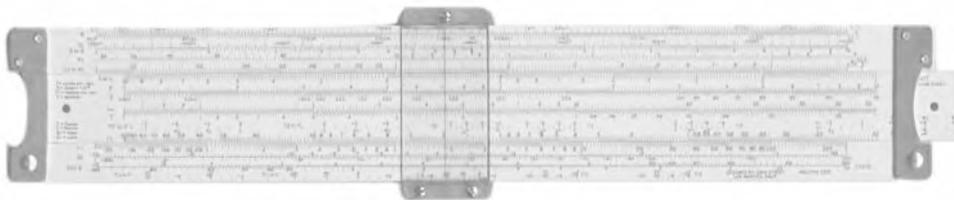
### 1614 AD – Napier’s Bones



**Fig. 1.3 Napier’s Bones**

The Napier’s Bones was invented by John Napier, a Scottish mathematician as an aid to multiplication. A set of bones consisted of nine rods, one for each digit 1 through 9 and a constant rod for the digit ‘0’. A rod is similar to one column of a multiplication table.

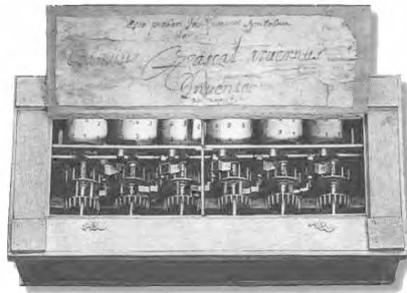
### 1633 AD – The Slide Rule



**Fig. 1.4 The Slide Rule**

The Slide Rule was invented by William Oughtred. It is based on the principle that actual distance from the starting point of the rule is directly proportional to the logarithm of the numbers printed on the rule. The slide rule is embodied by the two sets of scales that are joined together, with a marginal space between them. The suitable alliance of two scales enabled the slide rule to perform multiplication and division by a method of addition and subtraction.

### **1642 AD – The Rotating Wheel Calculator**



**Fig. 1.5 The Rotating Wheel Calculator**

The Rotating Wheel Calculator was developed by a French philosopher, Blaise Pascal, using simple components such as gears and levers. This is a predecessor to today's electronic calculator. He was inspired by the computation work of his father's job and devised the model. He was only 19 years old, when he devised this model.

### **1822 AD – The Difference Engine**



**Fig. 1.6 The Difference Engine**

The Difference Engine was built by Charles Babbage, British mathematician and engineer which mechanically calculated mathematical tables. Babbage is called the father of today's computer.

### **1890 AD - Hollerith Tabulating Machine**



**Fig. 1.7 Hollerith Tabulating Machine**

A tabulating machine using punched cards was designed by Herman Hollerith and was called as the Hollerith Tabulating Machine. This electronic machine is able to read the information on the punched cards and process it electronically.

### **1.1.3 Generation of Computers**

The evolution of electronic computers over a period of time can be traced effectively by dividing this period into various generations. Each generation is characterized by a major technological development that fundamentally changed the way computers operated. These helped to develop smaller, cheaper, powerful, efficient and reliable devices. Now you could read about each generation and the developments that led to the current devices that we use today.

### **First Generation - 1940-1956: Vacuum Tubes**

The first generation of computers used vacuum tubes for circuitry and magnetic drums for memory. They were large in size, occupied a lot of space and produced enormous heat.

They were very expensive to operate and consumed large amount of electricity. Sometimes the heat generated caused the computer to malfunction. First generation computers operated only on machine language. Input was based on punched cards and paper tape, and output was displayed on printouts. First generation computers could solve only one problem at a time.



**Fig. 1.8 Vacuum Tube**

The Universal Automatic Computer (UNIVAC) and the Electronic Numerical Integrator And Calculator (ENIAC) are classic examples of first-generation computing devices.

### **Second Generation - 1956-1963: Transistors**

The second generation of computers witnessed the vacuum tubes being replaced by transistors. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, energy-efficient and more reliable than their first-generation counter parts. The transistors also generated considerable heat that

sometimes caused the computer to malfunction. But it was a vast improvement over the vacuum tube. Second-generation computers used punched cards for input and printouts for output.

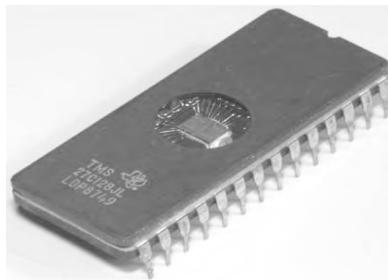


**Fig. 1.9 Transistor**

Second-generation computers moved from the use of machine language to assembly languages, which allowed programmers to specify instructions in words. High-level programming languages were also being developed at this time, such as early versions of COBOL and FORTRAN. The computers stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology.

### **Third Generation - 1964-1971 : Integrated Circuits**

The development of the integrated circuit left its mark in the third generation of computers. Transistors were made smaller in size and placed on silicon chips, which dramatically increased the speed and efficiency of computers.

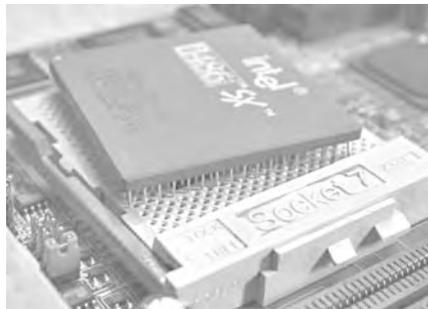


**Fig. 1.10 Integrated Circuit**

In this generation, keyboards and monitors were used instead of punched cards and printouts. The computers were interfaced with an operating system which allowed to solve many problems at a time.

#### **Fourth Generation - 1971-Present : Microprocessors**

The microprocessor brought forth the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip.



**Fig. 1.11 Microprocessor**

As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet.

#### **Fifth Generation - Present and Beyond: Artificial Intelligence**

Fifth generation computing devices, based on artificial intelligence, are still in their developmental stage. Fifth generation computers will come close to bridging the gap between computing and thinking.

### **1.2 Data, Information and Program**

Computer is a tool for solving problems. Computers accept instructions and data, perform arithmetic and logical operations and

produce information. Hence the instructions and data fed into the computer are converted into information through processing.



**Fig. 1.12 Data, Processing and Information**

Basically data is a collection of facts from which information may be derived. Data is defined as an un-processed collection of raw facts in a manner suitable for communication, interpretation or processing.

Hence data are

- Stored facts
- Inactive
- Technology based
- Gathered from various sources.

On the other hand **information** is a collection of facts from which conclusions may be drawn. Data that has been interpreted, translated, or transformed to reveal the underlying meaning. This information can be represented in textual, numerical, graphic, cartographic, narrative, or audiovisual forms.

Hence information is

- Processed facts
- Active
- Business based
- Transformed from data.

**Algorithm** is defined as a step-by-step procedure or formula for solving a problem i.e. a set of instructions or procedures for solving a problem. It is also defined as a mathematical procedure that can

usually be explicitly encoded in a set of computer language instructions that manipulate data.

A computer **program** (or set of programs) is designed to systematically solve a problem. For example, a problem to calculate the length of a straight line joining any two given points.

The programmer must decide the program requirements, develop logic and write instructions for the computer in a programming language that the computer can translate into machine language and execute. Hence, problem solving is an act of defining a problem, understanding the problem and arriving at workable solutions.

In other words, problem solving is the process of confronting a novel situation, formulating connection between the given facts, identifying the goal of the problem and exploring possible methods for reaching the goal. It requires the programmer to co-ordinate previous experience and intuition in order to solve the problem.

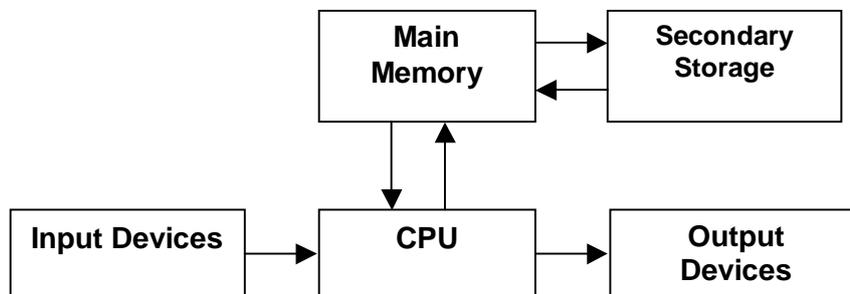
## **1.3 Hardware and Software**

### **1.3.1 Introduction**

A computer system has two major components, hardware and software. In practice, the term hardware refers to all the physical items associated with a computer system. Software is a set of instructions, which enables the hardware to perform a specific task.

### **1.3.2 Computer Hardware**

A computer is a machine that can be programmed to accept data (*input*), and process it into useful information (*output*). It also stores data for later reuse (*storage*). The *processing* is performed by the hardware. The computer hardware responsible for computing are mainly classified as follows:



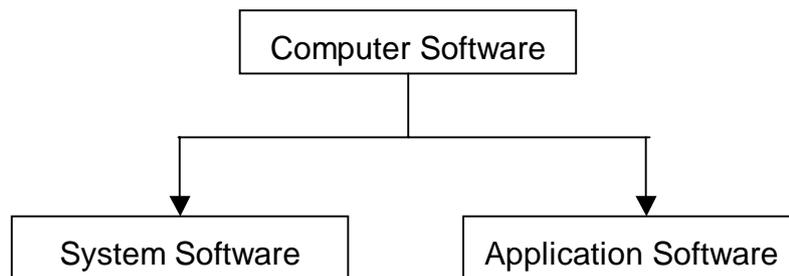
**Fig. 1.13 Computer Hardware**

- **Input devices** allows the user to enter the program and data and send it to the processing unit. The common input devices are keyboard, mouse and scanners.
- The **Processor**, more formally known as the *central processing unit* (CPU), has the electronic circuitry that manipulates input data into the information as required. The central processing unit actually executes computer instructions.
- **Memory** from which the CPU fetches the instructions and data is called main memory. It is also called as primary memory and is volatile in nature.
- **Output devices** show the processed data – information – the result of processing. The devices are normally a monitor and printers.
- **Storage** usually means secondary *storage*, which stores data and programs. Here the data and programs are permanently stored for future use.

The hardware devices attached to the computer are called peripheral equipment. **Peripheral equipment** includes all input, output and secondary storage devices.

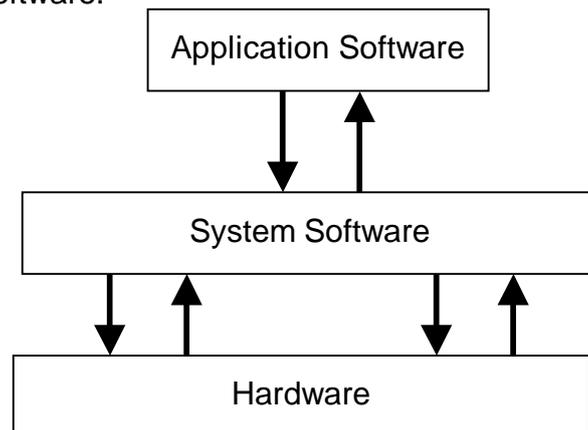
### 1.3.3 Computer Software

Software refers to a program that makes the computer to do something meaningful. It is the planned, step-by-step instructions required to turn data into information. Software can be classified into two categories: System Software and Application Software.



**Fig. 1.14 Software Categories**

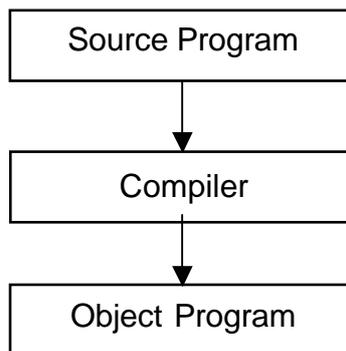
System software consists of general programs written for a computer. These programs provide the environment to run the application programs. System software comprises programs, which interact with the hardware at a very basic level. They are the basic necessity of a computer system for its proper functioning. System software serves as the interface between hardware and the user. The operating system, compilers and utility programs are examples of system software.



**Fig. 1.15 System Software**

The most important type of system software is the operating system. An operating system is an integrated set of specialized programs that is used to manage the overall operations of a computer. It acts like an interface between the user, computer hardware and software. Every computer must have an operating system to run other programs. DOS (Disk Operating System), Unix, Linux and Windows are some of the common operating systems.

The compiler software translates the source program (user written program) into an object program (binary form). Specific compilers are available for computer programming languages like FORTRAN, COBOL, C, C++ etc. The utility programs support the computer for specific tasks like file copying, sorting, linking a object program, etc.



**Fig. 1.16 Compiler**

An Application Software consists of programs designed to solve a user problem. It is used to accomplish specific tasks rather than just managing a computer system. Application software are inturn, controlled by system software which manages hardware devices.

Some typical examples are : railway reservation system, game programs, word processing software, weather forecasting programs. Among the application software some are packaged for specific tasks. The commonly used Application Software packages are word processor, spread sheet, database management system and graphics.

One of the most commonly used software package is **word processing software**. Anyone who has used a computer as a word processor knows that it is far more than a fancy typewriter. The great advantage of word processing over a typewriter is that you can make changes without retyping the entire document. The entire writing process is transformed by this modern word processing software. This software lets you create, edit, format, store and print text and graphics. Some of the commonly used word processors are Microsoft Word, WordStar, WordPerfect, etc.

Spreadsheet software packages allow the user to manipulate numbers. Repetitive numeric calculations, use of related formulae and creation of graphics and charts are some of the basic tools. This capability lets business people try different combinations of numbers and obtain the results quickly. Lotus1-2-3, Excel, etc. are some of the famous spreadsheet applications.

A database management system is a collection of programs that enable to store, modify and extract information from a database. A database organizes the information internally. Computerized banking system, Automated Teller Machine, Airlines and Railway reservation system etc., are some of the database applications.

Type of Software	Functions	Examples
Word Processors	All personal computers are loaded with word processing software which has the same function as a typewriter for writing letters, preparing reports and printing.	Microsoft Word Word Perfect, Word Star.
Spreadsheet	A table containing text and figures, which is used to Calculations and draw charts	Microsoft Excel, Lotus 1-2-3.
Database Management System	Used for storing, retrieval and Manipulation of Information	Microsoft Access, Oracle.

## 1.4 Types of Computers

### 1.4.1 Introduction

Classification of the electronic computers may be based on either their principles of operation or their configuration. By configuration, we mean the size, speed of doing computation and storage capacity of a computer.

### 1.4.2 Classification based on Principles of Operation

Based on the principles of operation, computers are classified into three types, analog computers, digital computers and hybrid computers.

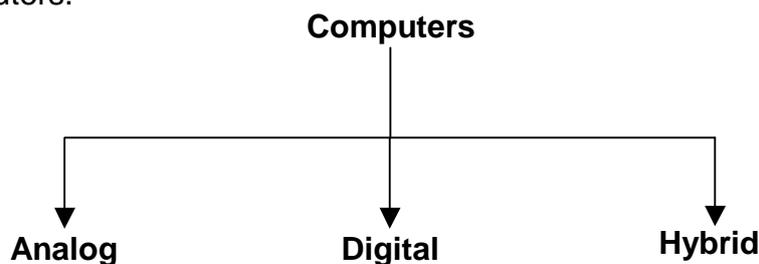


Fig. 1.17 Classification of Computers

#### Analog Computers

Analog Computer is a computing device that works on continuous range of values. The analog computers give approximate results since they deal with quantities that vary continuously. It generally deals with physical variables such as voltage, pressure, temperature, speed, etc.

#### Digital Computers

On the other hand a digital computer operates on digital data such as numbers. It uses binary number system in which there are only two digits 0 and 1. Each one is called a bit. The digital computer

is designed using digital circuits in which there are two levels for an input or output signal. These two levels are known as logic 0 and logic 1. Digital Computers can give the results with more accuracy and at a faster rate.

Since many complex problems in engineering and technology are solved by the application of numerical methods, the electronic digital computer is very well suited for solving such problems. Hence digital computers have an increasing use in the field of design, research and data processing.

Digital computers are made for both general purpose and special purpose. Special purpose computer is one that is built for a specific application. General purpose computers are used for any type of applications. It can store different programs and do the jobs as per the instructions specified on those programs. Most of the computers that we see fall in this category.

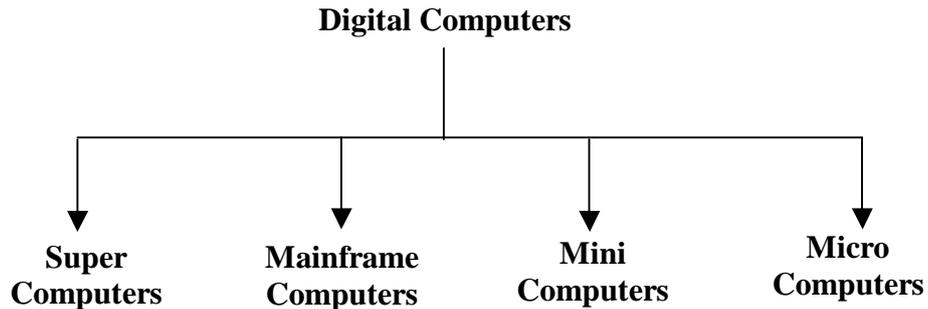
### **Hybrid Computers**

A hybrid computing system is a combination of desirable features of analog and digital computers. It is mostly used for automatic operations of complicated physical processes and machines. Now-a-days analog-to-digital and digital-to-analog converters are used for transforming the data into suitable form for either type of computation.

For example, in hospital's automated intensive care unit, analog devices might measure the patients temperature, blood pressure and other vital signs. These measurements which are in analog might then be converted into numbers and supplied to digital components in the system. These components are used to monitor the patient's vital sign and send signals if any abnormal readings are detected. Hybrid computers are mainly used for specialized tasks.

### 1.4.3 Classification of Computers based on Configuration

Based on performance, size, cost and capacity, the digital computers are classified into four different types : Super computers, Mainframe computers, Mini computers and Micro computers.



**Fig. 1.18 Classification of Digital Computers**

#### **Super Computers**

The mightiest computers but at the same time, the most expensive ones are known as super computers. Super computers process billions of instructions per second. In other words, super computers are the computers normally used to solve intensive numerical computations. Examples of such applications are stock analysis, special effects for movies, weather forecasting and even sophisticated artworks.

#### **Mainframe Computers**

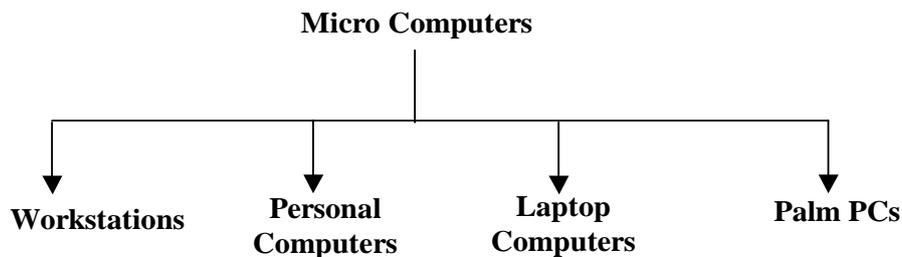
Mainframe computers are capable of processing data at very high speeds – hundreds of million instructions per second. They are large in size. These systems are also expensive. They are used to process large amount of data quickly. Some of the obvious customers are banks, airlines and railway reservation systems, aerospace companies doing complex aircraft design, etc.

## Mini Computers

The mini computers were developed with the objective of bringing out low cost computers. They are lower to mainframe computers, in terms of speed and storage capacity. Some of the hardware features available in mainframes were not included in the mini computer hardware in order to reduce the cost. Some features which were handled by hardware in mainframe computers were done by software in mini computers. Hence the performance of mini computer is less than that of the mainframe. However, the mini computer market has diminished somewhat as buyers have moved towards less expensive but increasingly powerful personal computers.

## Micro Computers

The invention of microprocessor (single chip CPU) gave birth to the micro computers. They are several times cheaper than mini computers.



**Fig. 1.19 Classification of Micro Computers**

The micro computers are further classified into workstation, personal computers, laptop computers and still smaller computers.

Although the equipment may vary from the simplest computer to the most powerful, the major functional units of the computer system remain the same : input, processing, storage and output.

### **Workstations**

Workstations are also desktop machines mainly used for intensive graphical applications. They have more processor speed than that of personal computers.



**Fig. 1.20 Workstation**

Workstations use sophisticated display screens featuring high-resolution colour graphics. Workstations are used for executing numeric and graphic intensive applications such as Computer Aided Design (CAD), simulation of complex systems and visualizing the results of simulation.

### **Personal Computers**



**Fig. 1.21 Personal Computer**

Today the personal computers are the most popular computer systems simply called PCs. These desktop computers are also known as home computers. They are usually easier to use and more affordable than workstations. They are self-contained desktop computers intended for an individual user. Most often used for word processing and small database applications.

### **Laptop Computers**



**Fig. 1.22 Laptop Computer**

Laptop computers are portable computers that fit in a briefcase. **Laptop computers**, also called **notebook computers**, are wonderfully portable and functional, and popular with travelers who need a computer that can go with them.

### **Getting Smaller Still**



**Fig. 1.23 Personal Digital Assistants**

**Pen-based computers** use a pen like stylus and accept handwritten input directly on a screen. Pen-based computers are also called **Personal Digital Assistants (PDA)**. Special engineering and hardware design techniques are adopted to make the portable, smaller and light weight computers.

## Summary

- A computer is an electronic machine, capable of performing basic operations like addition, subtraction, multiplication, division, etc.
- Abacus is the first known calculating machine used for counting.
- The Rotating Wheel Calculator was developed by Blaise Pascal, which is a predecessor to today's electronic calculator.
- Charles Babbage is called as the father of today's computer.
- The first generation of computers used vacuum tubes for circuitry and magnetic drums for memory.
- The second generation of computers witnessed the vacuum tubes being replaced by transistors.
- The third generation computer used the integrated circuits.
- The microprocessor brought forth the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip.
- Data is a collection of facts from which information may be derived.
- Information is a collection of facts from which conclusions may be drawn.
- Algorithm is defined as a step-by-step procedure or formula for solving a problem
- A computer program (or set of programs) is designed to systematically solve a problem.
- A computer system has two major components, hardware and software.
- The *processing* is performed by the hardware.
- Software refers to a program that makes the computer to do something meaningful and classified as System Software and Application Software
- System software consists of general programs written for a computer.
- An Application Software consists of programs designed to solve a user problem.

- Analog Computer is a computing device that works on continuous range of values.
- A digital computer operates on digital data such as numbers.
- A hybrid computing system is a combination of desirable features of analog and digital computers.
- Super computers process billions of instructions per second.
- Mainframes are capable of processing data at very high speeds – hundreds of million instructions per second.
- The mini computers were developed with the objective of bringing out low cost computers.
- The invention of microprocessor (single chip CPU) gave birth to the micro computers.
- The micro computers are further classified into workstation, personal computers, laptop computers and still smaller computers.

## Exercises

### I. Fill in the blanks

- 1) \_\_\_\_\_ is considered to be the father of today's computer.
- 2) \_\_\_\_\_ invented the Slide Rule.
- 3) The first generation of computers used \_\_\_\_\_ for circuitry and \_\_\_\_\_ for memory.
- 4) Integrated circuits were used in \_\_\_\_\_ generation of computers.
- 5) \_\_\_\_\_ refers to the physical items associated with a computer system.
- 6) The hardware devices attached to the computer are called \_\_\_\_\_.
- 7) \_\_\_\_\_ refers to programs that make the computer to do some thing.
- 8) Software can be classified into \_\_\_\_\_ and \_\_\_\_\_ software.
- 9) An \_\_\_\_\_ is an integrated set of specialized programs that is used to manage the overall operations of a computer.

- 10) The \_\_\_\_\_ translates the whole source program into an object program.
- 11) A \_\_\_\_\_ allows users to quickly and efficiently store, organize, retrieve, communicate and manage large amounts of information.
- 12) \_\_\_\_\_ computers are useful in solving differential equation and integration.
- 13) The digital computers are classified into \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_.
- 14) \_\_\_\_\_ is the planned step-by-step instruction required to turn data into information.
- 15) \_\_\_\_\_ is the raw material that is given to a computer for processing.
- 16) A \_\_\_\_\_ computer accepts handwritten input on a screen.
- 17) Raw data is processed by the computer into \_\_\_\_\_.
- 18) PC refers to \_\_\_\_\_.
- 19) \_\_\_\_\_ software allows to create, edit, format, store and print text and graphics.
- 20) The word computing means \_\_\_\_\_.

## **II. State whether the following are true or false**

- 1) The concept of using 'electronic brains' gave birth to computers.
- 2) The most powerful personal computers are known as super computers.
- 3) Blaise Pascal developed the tabulating machine using punched cards.
- 4) Herman Hollerith designed the difference engine.
- 5) Compilers translate higher level language into machine language.
- 6) Word processing is a type of task-oriented software.
- 7) Fifth generation computing devices is based on artificial intelligence.
- 8) The input devices accept data and send them to the processor.
- 9) A hybrid computing system is a combination of desirable features of analog and digital computers.
- 10) The personal computers are sometime called the home computers.

**III. Answer the following.**

- 1) What is a computer?
- 2) What is the name of the machine developed by Charles Babbage?
- 3) What are peripheral devices?
- 4) Define 'Data'.
- 5) Define 'Information'.
- 6) What do you mean by an algorithm?
- 7) What is a word processor software?
- 8) What is an operating System?
- 9) What is an analog computing system?
- 10) What is a lap-top computer?

**IV. Answer the following in detail.**

- 1) Discuss the various computer generations along with the key characteristics of the computer of each generation.
- 2) What is the relationship between software and hardware?
- 3) Write in detail about computer software and their categories.
- 4) Discuss the important features and uses of micro, mini, mainframe and super computers.

## **CHAPTER 2**

### **NUMBER SYSTEMS**

#### **2.1 Introduction**

There are several kinds of data such as, numeric, text, date, graphics, image, audio and video that need to be processed by a computer. The text data usually consist of standard alphabetic, numeric, and special characters. The graphics data consist of still pictures such as drawings and photographs. Any type of sound, including music and voice, is considered as audio data. Video data consist of motion pictures. The data has to be converted into a format that the computer understands. Data can be classified into two forms, analog data and digital data. Analog data can have any value within a defined range and it is continuous. Sound waves, telephone signals, temperatures and all other signals that are not broken into bits are examples of analog data. Digital data can be represented by a series of binary numbers and it is discrete.

The Arithmetic and Logic Unit (ALU) of the computer performs arithmetic and logical operations on data. Computer arithmetic is commonly performed on two different types of numbers, integer and floating point. As the hardware required for arithmetic is much simpler for integers than floating point numbers, these two types have entirely different representations. An integer is a whole number and the floating-point number has a fractional part. To understand about how computers store data in the memory and how they handle them, one must know about bits and bytes and the number systems.

Bits and bytes are common computer jargons. Both the main memory (Random Access Memory or RAM) and the hard disk capacities are measured in terms of bytes. The hard disk and memory capacity of a computer and other specifications are described in terms of bits and bytes. For instance, a computer may be described as having a 32-bit Pentium processor with 128 Megabytes of RAM and hard disk capacity of 40 Gigabytes.

## 2.2 Bits and Bytes

A numbering system is a way of representing numbers. The most commonly used numbering system is the decimal system. Computer systems can perform computations and transmit data thousands of times faster in binary form than they can use decimal representations. It is important for every one studying computers to know how the binary system and hexadecimal system work.

A bit is small piece of data that is derived from the words “**binary digit**”. Bits have only two possible values, 0 and 1. A binary number contains a sequence of 0s and 1s like 10111. A collection of 8 bits is called as a byte. With 8 bits in a byte, we can represent 256 values ranging from 0 to 255 as shown below:

0	=	0000 0000
1	=	0000 0001
2	=	0000 0010
3	=	0000 0011
		.....
		.....
		.....
254	=	1111 1110
255	=	1111 1111

Bytes are used to represent characters in a text. Different types of coding schemes are used to represent the character set and numbers. The most commonly used coding scheme is the American Standard Code for Information Interchange (ASCII). Each binary value between 0 and 127 is used to represent a specific character. The ASCII value for a blank character (blank space) is 32 and the ASCII value of numeric 0 is 48. The range of ASCII values for lower case alphabets is from 97 to 122 and the range of ASCII values for the upper case alphabets is 65 to 90.

Computer memory is normally represented in terms of Kilobytes or Megabytes. In metric system, one Kilo represents 1000, that is,  $10^3$ . In binary system, one Kilobyte represents 1024 bytes, that is,  $2^{10}$ . The following table shows the representation of various memory sizes.

Name	Abbreviation	Size (Bytes)
Kilo	K	$2^{10}^*$
Mega	M	$2^{20}$
Giga	G	$2^{30}$
Tera	T	$2^{40}$
Peta	P	$2^{50}$
Exa	E	$2^{60}$
Zetta	Z	$2^{70}$
Yotta	Y	$2^{80}$

\*Read as 2 power10.

In a 2GB (Gigabytes) storage device (hard disk), totally 21,47,483,648 bytes can be stored. Nowadays, databases having size in Terabytes are reported; Zetta and Yotta size databases are yet to come.

### 2.3 Decimal Number System

In our daily life, we use a system based on digits to represent numbers. The system that uses the decimal numbers or digit symbols 0 to 9 is called as the decimal number system. This system is said to have a base, or radix, of ten. Sequence of digit symbols are used to represent numbers greater than 9. When a number is written as a sequence of decimal digits, its value can be interpreted using the positional value of each digit in the number. The positional number system is a system of writing numbers where the value of a digit depends not only on the digit, but also on its placement within a number. In the positional number system, each decimal digit is weighted relative to its position in the number. This means that each digit in the number is multiplied by ten raised to a power

corresponding to that digit's position. Thus the value of the decimal sequence 948 is:

$$948_{10} = 9 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$$

Fractional values are represented in the same manner, but the exponents are negative for digits on the right side of the decimal point. Thus the value of the fractional decimal sequence 948.23 is:

$$948.23_{10} = 9 \times 10^2 + 4 \times 10^1 + 8 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$$

In general, for the decimal representation of

$$\mathbf{X} = \{ \dots \mathbf{x}_2 \mathbf{x}_1 \mathbf{x}_0 . \mathbf{x}_{-1} \mathbf{x}_{-2} \mathbf{x}_{-3} \dots \},$$

the value of  $\mathbf{X}$  is

$$\mathbf{X} = \sum_i \mathbf{x}_i 10^i \quad \text{where } i = \dots, 2, 1, 0, -1, -2, \dots$$

## 2.4 Binary Number System

Ten different digits 0 – 9 are used to represent numbers in the decimal system. There are only two digits in the binary system, namely, 0 and 1. The numbers in the binary system are represented to the base two and the positional multipliers are the powers of two. The leftmost bit in the binary number is called as the most significant bit (MSB) and it has the largest positional weight. The rightmost bit is the least significant bit (LSB) and has the smallest positional weight.

The binary sequence  $10111_2$  has the decimal equivalent:

$$\begin{aligned} 10111_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 0 + 4 + 2 + 1 \\ &= 23_{10} \end{aligned}$$

The decimal equivalent of the fractional binary sequence can be estimated in the same manner. The exponents are negative powers of two for digits on the right side of the binary point. The binary equivalent of the decimal point is the binary point. Thus the decimal value of the fractional binary sequence  $0.1011_2$  is:

$$\begin{aligned} 0.1011_2 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 0.5 + 0 + 0.125 + 0.0625 \\ &= 0.6875_{10} \end{aligned}$$

## 2.5 Hexadecimal Number System

Hexadecimal representation of numbers is more efficient in digital applications because it occupies less memory space for storing large numbers. A hexadecimal number is represented using base 16. Hexadecimal or Hex numbers are used as a shorthand form of binary sequence. This system is used to represent data in a more compact manner. In the hexadecimal number system, the binary digits are grouped into sets of 4 and each possible combination of 4 binary digits is given a symbol as follows:

0000 = 0	1000 = 8
0001 = 1	1001 = 9
0010 = 2	1010 = A
0011 = 3	1011 = B
0100 = 4	1100 = C
0101 = 5	1101 = D
0110 = 6	1110 = E
0111 = 7	1111 = F

Since 16 symbols are used, 0 to F, the notation is called hexadecimal. The first ten symbols are the same as in the decimal system, 0 to 9 and the remaining six symbols are taken from the first six letters of the alphabet sequence, A to F. The hexadecimal sequence  $2C_{16}$  has the decimal equivalent:

$$\begin{aligned} 2C_{16} &= 2 \times 16^1 + C \times 16^0 \\ &= 32 + 12 \\ &= 44_{10} \end{aligned}$$

The hexadecimal representation is more compact than binary representation. It is very easy to convert between binary and hexadecimal systems. Each hexadecimal digit will correspond to four binary digits because  $2^4 = 16$ . The hexadecimal equivalent of the binary sequence  $110010011101_2$  is:

$$\begin{array}{cccc} 1100 & 1001 & 1101 & = & C9D_{16} \\ C & 9 & D & & \end{array}$$

## 2.6 Decimal to Binary Conversion

To convert a binary number to a decimal number, it is required to multiply each binary digit by the appropriate power of 2 and add the results. There are two approaches for converting a decimal number into binary format.

### 2.6.1 Repeated Division by 2

Any decimal number divided by 2 will leave a remainder of 0 or 1. Repeated division by 2 will leave a string of 0s and 1s that become the binary equivalent of the decimal number. Suppose it is required to convert the decimal number  $M$  into binary form, dividing  $M$  by 2 in the decimal system, we will obtain a quotient  $M_1$  and a remainder  $r_1$ , where  $r_1$  can have a value of either 0 or 1.

$$\text{ie., } M = 2 * M_1 + r_1 \quad r_1 = 0 \text{ or } 1$$

Next divide the quotient  $M_1$  by 2. The new quotient will be  $M_2$  and the new remainder  $r_2$ .

$$\text{ie., } M_1 = 2 * M_2 + r_2 \quad r_2 = 0 \text{ or } 1$$

$$\begin{aligned} \text{so that } M &= 2 (2 * M_2 + r_2) + r_1 \\ &= 2^2 M_2 + r_2 * 2^1 + r_1 * 2^0 \end{aligned}$$

Next divide the quotient  $M_2$  by 2. The new quotient will be  $M_3$  and the new remainder  $r_3$ .

$$\begin{aligned}
 \text{i.e., } M_2 &= 2 * M_3 + r_3 \\
 \text{so that } M &= 2 (2 * (2 * M_3 + r_3) + r_2) + r_1 \\
 &= 2^2(2 * M_3 + r_3) + r_2 * 2^1 + r_1 * 2^0 \\
 &= 2^3 M_3 + r_3 * 2^2 + r_2 * 2^1 + r_1 * 2^0
 \end{aligned}$$

The above process is repeated until the quotient becomes 0, then

$$M = 1 * 2^k + r_k * 2^{k-1} + \dots + r_3 * 2^2 + r_2 * 2^1 + r_1 * 2^0$$

**Example:**

Convert  $23_{10}$  into its equivalent binary number.

	Quotient	Remainder
23/2	11	1 (LSB) 
11/2	5	1
5/2	2	1
2/2	1	0
1/2	0	1 (MSB)

To write the binary equivalent of the decimal number, read the remainders from the bottom upward as:

$$23_{10} = 10111_2$$

The number of bits in the binary number is the exponent of the smallest power of 2 that is larger than the decimal number. Consider a decimal number 23. Find the exponent of the smallest power of 2 that is larger than 23.

$$\begin{aligned}
 16 &< 23 < 32 \\
 2^4 &< 23 < 2^5
 \end{aligned}$$

Hence, the number 23 has 5 bits as 10111. Consider another example.

Find the number of bits in the binary representation of the decimal number 36 without actually converting into its binary equivalent.

The next immediate large number than 36 that can be represented in powers of 2 is 64.

$$\begin{aligned}32 < 36 < 64 \\ 2^5 < 36 < 2^6\end{aligned}$$

Hence, the number 36 should have 6 bits in its binary representation.

### 2.6.2 Sum of Powers of 2

A decimal number can be converted into a binary number by adding up the powers of 2 and then adding bits as needed to obtain the total value of the number. For example, to convert  $36_{10}$  to binary:

- a. Find the largest power of 2 that is smaller than or equal to 36

$$36_{10} > 32_{10}$$

- b. Set the 32's bit to 1 and subtract 32 from the original number.

$$36 - 32 = 4$$

- c. 16 is greater than the remaining total. Therefore, set the 16's bit to 0

- d. 8 is greater than the remaining total. Hence, set the 8's bit to 0

- e. As the remaining value is itself in powers of 2, set 4's bit to 1 and subtract 4

$$4 - 4 = 0$$

Conversion is complete when there is nothing left to subtract. Any remaining bits should be set to 0. Hence

$$36 = 100100_2$$

The conversion steps can be given as follows:

$$\begin{array}{r}
 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \qquad \qquad \qquad 36 - 32 = 4 \\
 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 0 \ 1 \qquad \qquad 4 - 4 = 0 \\
 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 0 \ 1 \ 0 \ 0 \quad 36_{10} = 100100_2
 \end{array}$$

**Example:**

Convert  $91_{10}$  to binary using the sum of powers of 2 method.

The largest power of 2 that is smaller than or equal to 91 is 64.

$$\begin{array}{r}
 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \qquad \qquad \qquad 91 - 64 = 27 \qquad \qquad 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 1 \qquad \qquad 91 - (64 + 16) = 11
 \end{array}$$

(Since  $32 > 27$ , set the 32's bit 0 and  $16 < 27$ . set the 16's bit 1)

$$\begin{array}{r}
 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 1 \ 1 \qquad \qquad 91 - (64 + 16 + 8) = 3
 \end{array}$$

$$\begin{array}{r}
 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \qquad \qquad 91 - (64 + 16 + 8 + 2) = 1
 \end{array}$$

$$\begin{array}{r}
 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \qquad \qquad 91 - (64 + 16 + 8 + 2 + 1) = 0
 \end{array}$$

Hence  $91_{10} = 1011011_2$

## 2.7 Conversion of fractional decimal to binary

The decimal fractions like  $1/2$ ,  $1/4$ ,  $1/8$  etc., can be converted into exact binary fractions. Sum of powers method can be applied to these fractions.

$$0.5_{10} = 1 * 2^{-1} = 0.1_2$$

$$0.25_{10} = 0 * 2^{-1} + 1 * 2^{-2} = 0.01_2$$

$$0.125_{10} = 0 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} = 0.001_2$$

The fraction  $5/8 = 4/8 + 1/8 = 1/2 + 1/8$  has the binary equivalent:

$$\begin{aligned} 5/8 &= 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} \\ &= 0.101_2 \end{aligned}$$

Exact conversion is not possible for the decimal fractions that cannot be represented in powers of 2. For example,  $0.2_{10}$  cannot be exactly represented by a sum of negative powers of 2. A method of repeated multiplication by 2 has to be used to convert such kind of decimal fractions.

The steps involved in the method of repeated multiplication by 2:

- Multiply the decimal fraction by 2 and note the integer part. The integer part is either 0 or 1.
- Discard the integer part of the previous product. Multiply the fractional part of the previous product by 2. Repeat the first step until the fraction repeats or terminates.

The resulting integer part forms a string of 0s and 1s that become the binary equivalent of the decimal fraction.

**Example:**

	Integer part
$0.2 * 2 = 0.4$	0
$0.4 * 2 = 0.8$	0
$0.8 * 2 = 1.6$	1
$0.6 * 2 = 1.2$	1
$0.2 * 2 = 0.4$	0



(Fraction repeats, the product is the same as in the first step)

Read the integer parts from top to bottom to obtain the equivalent fractional binary number. Hence  $0.2_{10} = 0.00110011..._2$

## 2.8 Conversion of Decimal to Hexadecimal

Decimal numbers' conversion to hexadecimal is similar to binary conversion. Decimal numbers can be converted into hexadecimal format by the sum of weighted hex digits method and by repeated division by 16. The sum of weighted hex digits method is suitable for small decimal numbers of maximum 3 digits. The method of repeated division by 16 is preferable for the conversion of larger numbers.

The exponent of the smallest power of 16 that is greater than the given decimal number will indicate the number of hexadecimal digits that will be present in the converted hexadecimal number. For example, the decimal number 948, when converted into hexadecimal number has 3 hexadecimal digits.

$$(16^3 = 4096) > 948 > (16^2 = 256)$$

Hence, the hexadecimal representation of 948 has 3 hex digits.

The conversion process is as follows:

$$\begin{array}{r} 16^2 \quad 16^1 \quad 16^0 \\ 3 \qquad \qquad \qquad 948 - (3 * 256) = 180 \end{array}$$

$$\begin{array}{r} 16^2 \quad 16^1 \quad 16^0 \\ 3 \quad B \quad \quad 948 - (3 * 256 + 11 * 16) = 4 \end{array}$$

$$\begin{array}{r} 16^2 \quad 16^1 \quad 16^0 \\ 3 \quad B \quad 4 \quad 948 - (3 * 256 + 11 * 16 + 4) = 0 \end{array}$$

Hence,  $948_{10} = 3B4_{16}$

The steps involved in the repeated division by 16 to obtain the hexadecimal equivalent are as follows:

- Divide the decimal number by 16 and note the remainder. Express the remainder as a hex digit.
- Repeat the process until the quotient is zero

**Example:**

Process	quotient	remainder
$948 / 16 =$	59	4 (LSB)
$59 / 16 =$	3	11 (B)
$3 / 16 =$	0	3 (MSB)
$948_{10} = 3B4_{16}$		

## 2.9 Octal Representation

An octal number is represented using base 8. Octal representation is just a simple extension of binary and decimal representations but using only the digits 0 to 7. To convert an octal number to a decimal number, it is required to multiply each octal digit by the appropriate power of 8 and add the results.

### Example

What is the decimal value of the octal number  $711_8$ ?

$$7 * 8^2 + 1 * 8^1 + 1 * 8^0 = 457_{10}$$

The steps involved in the repeated division by 8 to obtain the octal equivalent are as follows:

- Divide the decimal number by 8 and note the remainder. Express the remainder as an octal digit.
- Repeat the process until the quotient is zero

What is the octal representation of the decimal number  $64_{10}$ ?

	Quotient	Remainder
64/8	8	0 (LSB)
8/8	1	0
1/8	0	1 (MSB)

↑

Hence  $64_{10} = 100_8$

## 2.10 Representation of signed numbers

If computers represent non-negative integers (unsigned) only, the binary representation is straightforward, as we had seen earlier. Computers have also to handle negative integers (signed). The normal convention that is followed to distinguish between a signed and unsigned number is to treat the most significant (leftmost) bit in the binary sequence as a sign bit. If the leftmost bit is 0, the number is positive, and if the leftmost bit is 1, the number is negative.

### 2.10.1 Sign+magnitude representation

The simplest form of representing a negative integer is the sign+magnitude representation. In a sequence of n bits, the leftmost bit is used for sign and the remaining n-1 bits are used to hold the magnitude of the integer. Thus in a sequence of 4 bits,

$$\begin{aligned}0100 &= +4 \\1100 &= -4\end{aligned}$$

As there are several drawbacks in this representation, this method has not been adopted to represent signed integers. There are two representations for 0 in this approach.

$$\begin{aligned}0000 &= +0_{10} \\1000 &= -0_{10}\end{aligned}$$

Hence it is difficult to test for 0, which is an operation, performed frequently in computers. Another drawback is that, the addition and subtraction require a consideration of both the sign of the numbers and their relative magnitude, in order to carry out the required operation. This would actually complicate the hardware design of the arithmetic unit of the computer. The most efficient way of representing a signed integer is a 2's-complement representation. In 2's complement method, there is only one representation of 0.

### 2.10.2. 2's-complement representation

This method does not change the sign of the number by simply changing a single bit (MSB) in its representation. The 2's-complement method used with -ve numbers only is as follows:

- a. Invert all the bits in the binary sequence (ie., change every 0 to 1 and every 1 to 0 ie., 1's complement)
- b. Add 1 to the result

This method works well only when the number of bits used by the system is known in the representation of the number. Care should be

taken to pad (fill with zeros) the original value out to the full representation width before applying this algorithm.

**Example:**

In a computer that uses 8-bit representation to store a number, the wrong and right approaches to represent -23 are as follows:

**Wrong approach:**

The binary equivalent of 23 is 10111.  
Invert all the bits => 01000  
Add 1 to the result => 01001  
Pad with zeros to make 8-bit pattern => 00001001 => +9

**Right approach:**

The binary equivalent of 23 is 10111  
Pad with zeros to make 8-bit pattern => 00010111  
Invert all the bits => 11101000  
Add 1 to the result => 11101001 => -23

**2.10.3 Manual method to represent signed integers in 2's complement form**

This is an easier approach to represent signed integers. This is for -ve numbers only.

Step 1: Copy the bits from right to left, through and including the first 1.

Step 2: Copy the inverse of the remaining bits.

**Example 1:**

To represent -4 in a 4-bit representation:

The binary equivalent of the integer 4 is 0100

As per step1, copy the bits from right to left, through and including the first 1 => 100

As per step2, copy the inverse of the remaining bits => 1 100 => -4

### **Example 2:**

To represent -23 in a 8-bit representation:

The binary equivalent of 23 is 00010111

As per step 1:           1

As per step 2: 11101001 => -23

### **2.10.4 Interpretation of unsigned and signed integers**

Signed number versus unsigned number is a matter of interpretation. A single binary sequence can represent two different values. For example, consider a binary sequence  $11100110_2$ .

The decimal equivalent of the above sequence when considered as an unsigned integer is:

$$11100110_2 = 230_{10}$$

The decimal equivalent of the sequence when considered as a signed integer in 2's complement form is:

$$11100110_2 = -26_{10} \text{ (after 2's complement and add negative sign).}$$

When comparing two binary numbers for finding which number is greater, the comparison depends on whether the numbers are considered as signed or unsigned numbers.

**Example:**

X = 1001

Y = 0011

Is (X > Y) /\* Is this true or false? \*/

It depends on whether X and Y are considered as signed or unsigned.

If X and Y are unsigned:

X is greater than Y

If X and Y are signed:

X is less than Y.

### 2.10.5 Range of unsigned and signed integers

In a 4-bit system, the range of unsigned integers is from 0 to 15, that is, 0000 to 1111 in binary form. Each bit can have one of two values 0 or 1. Therefore, the total number of patterns of 4 bits will be  $2 \times 2 \times 2 \times 2 = 16$ . In an n-bit system, the total number of patterns will be  $2^n$ . Hence, if n bits are used to represent an unsigned integer value, the range is from 0 to  $2^n - 1$ , that is, there are  $2^n$  different values.

In case of a signed integer, the most significant (left most) bit is used to represent a sign. Hence, half of the  $2^n$  patterns are used for positive values and the other half for negative values. The range of positive values is from 0 to  $2^{n-1} - 1$  and the range of negative values is from  $-1$  to  $-2^{n-1}$ . In a 4-bit system, the range of signed integers is from  $-8$  to  $+7$ .

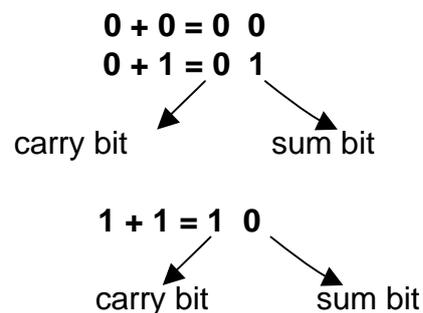
## 2.11 Binary Arithmetic

Digital arithmetic usually means binary arithmetic. Binary arithmetic can be performed using both signed and unsigned binary numbers.

### 2.11.1 Binary Addition – Unsigned numbers

When two digits are added, if the result is larger than what can be contained in one digit, a carry digit is generated. For example, if we add 5 and 9, the result will be 14. Since the result cannot fit into a single digit, a carry is generated into a second digit place. When two bits are added it will produce a sum bit and a carry bit. The carry bit may be zero.

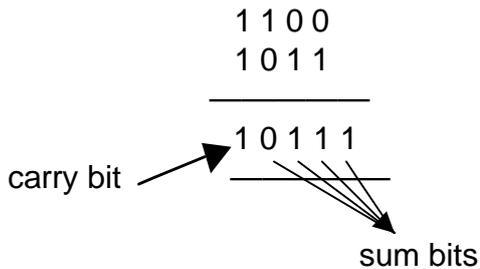
**Example:**



The sum bit is the least significant bit (LSB) of the sum of two 1-bit binary numbers and the carry bit holds the value of carry (0 or 1) resulting from the addition of two binary numbers.

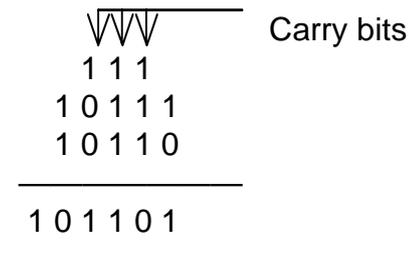
**Example 1:**

Calculate the sum of the numbers, 1100 and 1011:



**Example 2:**

Calculate 10111 + 10110



In unsigned binary addition, the two operands are called augend and addend. An augend is the number in an addition operation to which another number is added. An addend is the number in an addition operation that is added to another.

**2.11.2 Binary addition – signed numbers**

Signed addition is done in the same way as unsigned addition. The only difference is that, both operands must have the same number of magnitude bits and each must have a sign bit. As we have already seen, in a signed number, the most significant bit (MSB) is a sign bit while the rest of the bits are magnitude bits. When the number is negative, the sign bit is 1 and when the number is positive, the sign bit is 0.



If the result of the operation is negative, we get a negative number in 2's complement form. In some cases, there is a carry bit beyond the end of the word size and this is ignored.

**Example 3:**

Add  $-4_{10} + 4_{10}$ . Use 4-bit system.

$$\begin{array}{r}
 1\ 1\ 0\ 0 \quad (-4 \text{ in } 2\text{'s complement form}) \\
 0\ 1\ 0\ 0 \quad (+4) \\
 \hline
 1\ 0\ 0\ 0\ 0 = 0 \\
 \hline
 \end{array}$$

In the above example, the carry bit goes beyond the end of the word and this can be ignored. In this case both operands are having different signs. There will be no error in the result. On any addition, the result may be larger than can be held in the word size being used and this would result in overflow.

**The overflow condition is based on the rule:**

If two numbers are added and if they are either positive or negative, then overflow occurs if and only if the result has the opposite sign.

**Example 4:**

Add  $(-7_{10}) + (-5_{10})$  using the word size 4.

$$\begin{array}{r}
 1\ 0\ 0\ 1 \quad (-7 \text{ in } 2\text{'s complement form}) \\
 1\ 0\ 1\ 1 \quad (-5 \text{ in } 2\text{'s complement form}) \\
 \hline
 1\ 0\ 1\ 0\ 0 \quad (\text{The result is wrong}) \\
 \hline
 \end{array}$$

In the above example both operands are negative. But the MSB of the result is 0 that is the result is positive (opposite sign) and hence overflow occurs and the result is wrong.

### 2.11.3 Binary Subtraction

Subtrahend and minuend are the two operands in an unsigned binary subtraction. The minuend is the number in a subtraction operation from which another number is subtracted. The subtrahend is the number that is subtracted from another number. Simple binary subtraction operations are as follows:

$$\begin{array}{rcl}
 0 - 0 & = & 0 \\
 1 - 0 & = & 1 \\
 1 - 1 & = & 0 \\
 10 - 1 & = & 1
 \end{array}$$

When subtracting 1 from 0, borrow 1 from the next most significant bit (MSB). When borrowing from the next most significant bit, if it is 1, replace it with 0. If the next most significant bit is 0, you must borrow from a more significant bit that contains 1 and replace it with 0 and all 0s up to that point become 1s.

#### Example 1:

Subtract 1101 – 1010

$$\begin{array}{r}
 \phantom{0}01 \quad \leftarrow \text{borrow} \\
 1101 \quad (\text{minuend}) \\
 -1010 \quad (\text{subtrahend}) \\
 \hline
 0011 \\
 \hline
 \end{array}$$

When subtracting the 2<sup>nd</sup> least significant bit (1 in the subtrahend) from 0 (in the minuend), a 1 is borrowed from the more significant bit (3<sup>rd</sup> bit from right in the minuend) and hence  $10 - 1 = 1$ . The 3<sup>rd</sup> least significant bit is made as 0.

**Example 2:**

Subtract  $1000 - 101$

0 1 1	after borrowing, the minuend will become
1 0 0 0	0 1 1 10
- 1 0 1	1 0 1 (subtrahend)
0 0 1 1	difference as per the basic operations for subtraction

To subtract one number (subtrahend) from another (minuend), take the 2's complement of the subtrahend and add it to the minuend.

**Example 3:**

Subtract  $(+2) - (+7)$  using 4-bit system

	0 0 1 0 (+2)
	0 1 1 1 (+7)
	1 0 0 1 (-7 in 2's complement form)
	0 0 1 0 (2)
+	1 0 0 1 (-7)
	1 0 1 1 (-5)

#### Example 4:

Subtract  $(-6) - (+4)$  using 4 bit system

$$\begin{array}{r} \text{Minuend} \quad \quad \quad -6 \quad 1 \ 0 \ 1 \ 0 \\ \text{2's complement of the Subtrahend} \ -4 \quad 1 \ 1 \ 0 \ 0 \\ \hline \quad \quad \quad \quad \quad \quad 1 \ 0 \ 1 \ 1 \ 0 \\ \hline \end{array}$$

Both numbers are represented as negative numbers. While adding them, the result will be : 10110. As the word size is 4, the carry bit goes beyond the end of the word and the result is positive as the MSB is 0. This case leads to overflow and hence the result is wrong. The overflow rule works in subtraction also.

## 2.12 Boolean algebra

Boolean algebra is a mathematical discipline that is used for designing digital circuits in a digital computer. It describes the relation between inputs and outputs of a digital circuit. The name Boolean algebra has been given in honor of an English mathematician George Boole who proposed the basic principles of this algebra. As with any algebra, Boolean algebra makes use of variables and operations (functions). A Boolean variable is a variable having only two possible values such as, **true or false**, or as, **1 or 0**. The basic logical operations are **AND**, **OR** and **NOT**, which are symbolically represented by dot, plus sign, and by over bar / single apostrophe.

Example:

$$\begin{array}{l} \mathbf{A \quad AND \quad B \quad = \quad A \cdot B} \\ \mathbf{A \quad OR \quad B \quad = \quad A + B} \\ \mathbf{NOT \quad A \quad = \quad A' \quad (or \quad \overline{A})} \end{array}$$

A Boolean expression is a combination of Boolean variables, Boolean Constants and the above logical operators. All possible operations in Boolean algebra can be created from these basic logical operators. There are no negative or fractional numbers in Boolean algebra.

The operation AND yields true (binary value 1) if and only if both of its operands are true. The operation OR yields true if either or both of its operands are true. The unary operation NOT inverts the value of its operand. The basic logical operations can be defined in a form known as Truth Table, which is a list of all possible input values and the output response for each input combination.

### 2.12.1 Boolean operators (functions)

#### AND operator

The AND operator is defined in Boolean algebra by the use of the dot (.) operator. It is similar to multiplication in ordinary algebra. The AND operator combines two or more input variables so that the output is true only if all the inputs are true. The truth table for a 2-input AND operator is shown as follows:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

The above 2-input AND operation is expressed as:  $Y = A \cdot B$

#### OR operator

The plus sign is used to indicate the OR operator. The OR operator combines two or more input variables so that the output is true if at least one input is true. The truth table for a 2-input OR operator is shown as follows:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

The above 2-input OR operation is expressed as:  $Y = A + B$

### NOT operator

The NOT operator has one input and one output. The input is either true or false, and the output is always the opposite, that is, the NOT operator inverts the input. The truth table for a NOT operator where A is the input variable and Y is the output is shown below:

A	Y
0	1
1	0

The NOT operator is represented algebraically by the Boolean expression:  $Y = \overline{A}$

Example: Consider the Boolean equation:

$$D = A + (\overline{B} \cdot C)$$

D is equal to 1 (true) if A is 1 or if  $(\overline{B} \cdot C)$  is 1, that is, B = 0 and C = 1. Otherwise D is equal to 0 (false).

The basic logic functions AND, OR, and NOT can also be combined to make other logic operators.

## NAND operator

The NAND is the combination of NOT and AND. The NAND is generated by inverting the output of an AND operator. The algebraic expression of the NAND function is:

$$Y = \overline{A \cdot B}$$

The NAND function truth table is shown below:

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$$A \text{ NAND } B = \text{NOT } (A \text{ AND } B)$$

## NOR operator

The NOR is the combination of NOT and OR. The NOR is generated by inverting the output of an OR operator. The algebraic expression of the NOR function is:

$$Y = \overline{A + B}$$

The NOR function truth table is shown below:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$$A \text{ NOR } B = \text{NOT } (A \text{ OR } B)$$

## 2.12.2 Laws of Boolean algebra

Boolean algebra helps to simplify Boolean expressions in order to minimize the number of logic gates in a digital circuit. You will study about logic gates in the forthcoming chapter. This chapter focuses on the theorems of Boolean algebra for manipulating the Boolean expressions in order to simplify them.

### Boolean Identities

#### Laws of Complementation

The term complement simply means to change 1s to 0s and 0s to 1s.

Theorem 1 : If  $A = 0$ , then  $\overline{A} = 1$

Theorem 2 : If  $A = 1$ , then  $\overline{A} = 0$

Theorem 3 : The complement to complement of A is A itself.

$$\overline{\overline{A}} = A$$

#### Basic properties of AND operator

Theorem 4 :  $A \cdot 1 = A$

If A equals 0 and the other input is 1, the output is 0.

If A equals 1 and the other input is 1, the output is 1.

Thus the output is always equal to the A input.

Theorem 5 :  $A \cdot 0 = 0$

As one input is always 0, irrespective of A, the output is always 0.

Theorem 6 :  $A \cdot A = A$

The output is always equal to the A input.

Theorem 7 :  $A \cdot \bar{A} = 0$

Regardless of the value of A, the output is 0.

### **Basic properties of OR operator**

Theorem 8 :  $A + 1 = 1$

If A equals 0 and the other input is 1, the output is 1.

If A equals 1 and the other input is 1, the output is 1.

Thus the output is always equal to 1 regardless of what value A takes on.

Theorem 9 :  $A + 0 = A$

The output assumes the value of A.

Theorem 10 :  $A + A = A$

The output is always equal to the A input.

Theorem 11 :  $A + \bar{A} = 1$

Regardless of the value of A, the output is 1.

### **2.12.3 Simplification of Boolean expressions**

Before seeing the important theorems used in the simplification of Boolean expressions, some Boolean mathematical concepts need to be understood.

## Literal

A literal is the appearance of a variable or its complement in a Boolean expression.

## Product Term

A product term in a Boolean expression is a term where one or more literals are connected by AND operators. A single literal is also a product term.

Example:  $\overline{AB}$ ,  $AC$ ,  $\overline{A}\overline{C}$ , and  $\overline{E}$  are the product terms.

## Minterm

A minterm is a product term, which includes all possible variables either complemented or uncomplemented. In a Boolean expression of 3 variables,  $x$ ,  $y$ , and  $z$ , the terms  $xyz$ ,  $\overline{x}yz$ , and  $\overline{x}\overline{y}z$  are minterms. But  $\overline{x}y$  is not a minterm. Minterm is also called as a standard product term.

## Sum term

A sum term in a Boolean expression is a term where one or more literals are connected by OR operators.

Example:  $A + \overline{B} + D$

## Maxterm

A maxterm is a sum term in a Boolean expression, which includes all possible variables in true or complement form. In a Boolean expression of 3 variables,  $x$ ,  $y$ , and  $z$ , the terms  $x + y + z$ , and  $x + \overline{y} + \overline{z}$  are the maxterms. Maxterm is also called as standard sum term.

### Sum-of-products (SOP)

A sum of products expression is a type of Boolean expression where one or more product terms are connected by OR operators.

Example:  $\overline{A} + A B + \overline{A} \overline{B} \overline{C}$

In an expression of 3 variables, A, B, and C, the expression  $ABC + A B \overline{C} + \overline{A} B \overline{C}$  is also called as a canonical sum or sum of standard product terms or sum of minterms.

### Product-of-sums (POS)

Product of sums is a type of Boolean expression where several sum terms are connected by AND operators.

Example:  $(A + B) (\overline{A} + \overline{B}) (\overline{A} + B)$

A canonical product or product of standard sum terms is a product of sums expression where all the terms are maxterms. The above example is a canonical product in a Boolean expression of two variables A and B.

### Theorem 12: Commutative Law

A mathematical operation is commutative if it can be applied to its operands in any order without affecting the result.

Addition and multiplication operations are commutative.

Example:

$$A + B = B + A$$

$$AB = BA$$

Subtraction is not commutative:

$$A - B \neq B - A$$

There is no subtraction operation in Boolean algebra.

### **Theorem 13: Associative Law**

A mathematical operation is associative if its operands can be grouped in any order without affecting the result. In other words, the order in which one does the OR operation does not affect the result.

$$(A + B) + C = A + (B+C) = (A + C) + B$$

Similarly, the order in which one does the AND operation does not affect the result.

$$(AB)C = A(BC) = (AC)B$$

### **Theorem 14: Distributive Law**

The distributive property allows us to distribute an AND across several OR functions.

Example:

$$A(B+C) = AB + AC$$

The following distributive law is worth noting because it differs from what we would find in ordinary algebra.

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

The simplest way to prove the above theorem is to produce a truth table for both the right hand side (RHS) and the left hand side (LHS) expressions and show that they are equal.

A	B	C	BC	LHS	A+B	A+C	RHS
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

### Minimum Sum of Products

A minimum sum of products expression is one of those Sum of Products expressions for a Boolean expression that has the fewest number of terms.

Consider the following Boolean Expression:

$$\bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A \bar{B} C + A B C$$

Using Associativity Law

$$= (\bar{A} B \bar{C} + \bar{A} B C) + (A \bar{B} \bar{C} + A \bar{B} C) + A B C$$

$$= \bar{A} B(\bar{C}+C) + A \bar{B}(\bar{C}+C) + ABC$$

Using Theorem 11

$$= \bar{A} B (1) + A \bar{B} (1) + ABC$$

Using Theorem 4

$$= \bar{A} B + A \bar{B} + ABC$$

The above expression is in the minimum sum of products form. The given Boolean expression can be rewritten as follows using theorem 10.

$$\begin{aligned}
 & \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} \bar{C} + A \bar{B} C + A B C + A \bar{B} C \quad (A \bar{B} C + A \bar{B} C = A \bar{B} C) \\
 & = (\bar{A} B \bar{C} + \bar{A} B C) + (A \bar{B} \bar{C} + A \bar{B} C) + (A B C + A \bar{B} C) \\
 & = \bar{A} B (\bar{C} + C) + A \bar{B} (\bar{C} + C) + A C (B + \bar{B}) \\
 & = \bar{A} B + A \bar{B} + A C
 \end{aligned}$$

The same Boolean expression can be simplified into many minimum sum of products form.

Examples:

**Simplify the following Boolean Expression**

$$\bar{A} B \bar{C} + \bar{A} B C$$

$$\text{Let } x = \bar{A} B \quad \text{and } y = \bar{C}$$

The above Boolean expression becomes

$$x y + x \bar{y}$$

$$= x(y + \bar{y})$$

$$= x = \bar{A} B$$

**Prove that  $A + \bar{A} B = A + B$**

According to Distributive Law

$$A + \bar{A} B = (A + \bar{A})(A + B) = 1 \cdot (A + B) = A + B$$

### Simplify the following Boolean Expression

$$\begin{aligned}\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} \\ &= \bar{A}\bar{C}(\bar{B} + B) + \bar{A}BC + A\bar{B}\bar{C} \\ &= \bar{A}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} \\ &= \bar{A}(\bar{C} + BC) + A\bar{B}\bar{C} \\ &= \bar{A}(\bar{C} + B)(\bar{C} + C) + A\bar{B}\bar{C} \\ &= \bar{A}(\bar{C} + B) + A\bar{B}\bar{C} \\ &= \bar{A}\bar{C} + \bar{A}B + A\bar{B}\bar{C} \quad (\text{one minimal form})\end{aligned}$$

In the given Boolean Expression, if the second and third terms are grouped, it will give

$$\begin{aligned}\bar{A}\bar{B}\bar{C} + (\bar{A}B\bar{C} + \bar{A}BC) + A\bar{B}\bar{C} \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}B(\bar{C} + C) + A\bar{B}\bar{C} \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}B + A\bar{B}\bar{C} \\ &= \bar{B}\bar{C}(\bar{A} + A) + \bar{A}B \\ &= \bar{B}\bar{C} + \bar{A}B \quad (\text{most minimal form})\end{aligned}$$

#### 2.12.4 DeMorgan's Theorems

**Theorem 15:**  $\overline{A + B} = \bar{A}\bar{B}$

**Theorem 16:**  $\overline{AB} = \bar{A} + \bar{B}$

The above identities are the most powerful identities used in Boolean algebra. By constructing the truth tables, the above identities can be proved easily.

**Example:**

Given Boolean function  $f(A,B,C,D) = D\bar{A}B + A\bar{B} + D A C$ , Find the complement of the Boolean function

$$\bar{f}(A,B,C,D) = \overline{D\bar{A}B + A\bar{B} + D A C}$$

Apply DeMorgan's Law (theorem 15)

$$= \overline{(D\bar{A}B)} \overline{(A\bar{B})} \overline{(D A C)}$$

Apply DeMorgan's Law (theorem 16)

$$= (\bar{D} + A + \bar{B})(\bar{A} + B)(\bar{D} + \bar{A} + \bar{C})$$

In the above problem, the given Boolean function is in the sum of products form and its complement is in the product of sums form.

The DeMorgan's theorem says that any logical binary expression remains unchanged if we,

- ◆ change all variables to their complements
- ◆ change all AND operations to OR operations
- ◆ change all OR operations to AND operations
- ◆ take the complement of the entire expression

A practical operational way to look at DeMorgan's theorem is that the inversion of an expression may be broken at any point and the operation at that point replaced by its opposite ( i.e., AND replaced by OR or vice versa).

The fundamentals of numbering systems, including examples showing how numbering systems work, converting values between one numbering system and another, and performing simple types of binary arithmetic have been covered in this chapter. Boolean algebra has been introduced and Boolean identities and the laws of Boolean algebra are explained with examples. The identities and the theorems are used in the simplification of Boolean expressions. The pictorial representation of the Boolean operators, that is, logic gates and the design of logic circuits are discussed in Chapter 4.

## EXERCISES

### I. Fill in the blanks

1. The term bit stands for \_\_\_\_\_ .
2. The radix of an octal system is \_\_\_\_\_ and for the hexadecimal system is \_\_\_\_\_ .
3. The range of unsigned integers in an n-bit system is from \_\_\_\_\_ to \_\_\_\_\_ .
4. The synonyms LSB and MSB stand for \_\_\_\_\_ , \_\_\_\_\_ and \_\_\_\_\_ .
5. In binary addition, the operands are called as \_\_\_\_\_ and \_\_\_\_\_ .
6. In binary subtraction, the operands are called as \_\_\_\_\_ and \_\_\_\_\_ .
7. The binary representation of the decimal number 5864 is \_\_\_\_\_ and the hexadecimal representation of the same number will be \_\_\_\_\_ .
8. The 2's complement of 0 is \_\_\_\_\_ .

9. The arithmetic operations in a digital computer are performed using the radix \_\_\_\_\_, \_\_\_\_\_
10. One byte equals \_\_\_\_\_ number of bits.
11. One million bytes are referred to as MB and one billion bytes are referred to as \_\_\_\_\_
12. The exponent of the smallest power of 2 that is larger than 68 is \_\_\_\_\_ and hence the number 68 has \_\_\_\_\_ binary digits in its binary equivalent.

## II. Review questions

1. Convert the following decimal numbers into their equivalent binary, octal and hexadecimal numbers.
  - a. 512
  - b. 1729
  - c. 1001
  - d. 777
  - e. 160
2. Write  $-27_{10}$  as an 8-bit 2's complement number.
3. Add the signed numbers  $+15_{10}$  and  $+36_{10}$ . Write the operands and the sum as 8-bit binary numbers.
4. Write the largest positive and negative numbers for an 8-bit signed number in decimal and 2's complement notation.
5. Do the following signed binary arithmetic operations.
  - a.  $10_{10} + 15_{10}$
  - b.  $-12_{10} + 5_{10}$
  - c.  $14_{10} - 12_{10}$
  - d.  $(-2_{10}) - (-6_{10})$
6. Convert the following binary numbers to decimal numbers
  - a.  $1011_2$
  - b.  $101110_2$
  - c.  $1010011_2$

7. Convert the following binary numbers into hexadecimal numbers
- a.  $101_2$       b.  $11010_2$       c.  $111101000010_2$
8. Convert the following hexadecimal numbers to binary numbers
- a.  $F2_{16}$       b.  $1A8_{16}$       c.  $39EB_{16}$
9. Convert the following hexadecimal numbers to decimal numbers
- a.  $B6_{16}$       b.  $5E9_{16}$       c.  $CAFE_{16}$
10. Do the following binary arithmetic.
- a.  $11011001 + 1011101$       b.  $101110 - 1011$
11. Convert the following decimal numbers to binary using sum of powers of 2 method
- a.  $41_{10}$     b.  $77_{10}$       c.  $95_{10}$
12. Using the theorems stated in Boolean algebra, prove the following
- a.  $A + AB = A$       b.  $(A + B)(A + C) = A + BC$
13. Simplify the following Boolean expressions
- a.  $\overline{A} \overline{B} \overline{C} + \overline{A} B \overline{C} + A \overline{B} \overline{C}$
- b.  $\overline{A} \overline{B} \overline{C} + \overline{A} \overline{B} C + A \overline{B} \overline{C} + A \overline{B} C$
14. Using DeMorgan's theorems, simplify the following Boolean expressions
- a.  $\overline{A} \overline{C} + \overline{B+C}$       b.  $\overline{(\overline{AC}) + B} + C$
15. Draw the truth table of the Boolean Expression
- $$\overline{(\overline{A} + \overline{B} + \overline{C})}$$

## **CHAPTER 3**

### **COMPUTER ORGANIZATION**

#### **3.1 Basic Components of a Digital Computer**

##### **3.1.1 Introduction**

Computers are often compared to human beings since both have the ability to accept data, store, work with it, retrieve and provide information. The main difference is that human beings have the ability to perform all of these actions independently. Human beings also think and control their own activities. The computer, however, requires a program (a predefined set of instructions) to perform an assigned task. Human beings receive information in different forms, such as eyes, ears, nose, mouth, and even sensory nerves. The brain receives or accepts this information, works with it in some manner, and then stores in the brain for future use. If information at the time requires immediate attention, brain directs to respond with actions. Likewise the Central Processing Unit (CPU) is called the brain of the computer. It reads and executes program instructions, performs calculations and makes decisions.

##### **3.1.2 Components of a Digital Computer**

A computer system is the integration of physical entities called hardware and non-physical entities called software. The hardware components include input devices, processor, storage devices and output devices. The software items are programs and operating aids (systems) so that the computer can process data.

##### **3.1.3 Functional Units of a Computer System**

Computer system is a tool for solving problems. The hardware should be designed to operate as fast as possible. The software (system software) should be designed to minimize the amount of idle

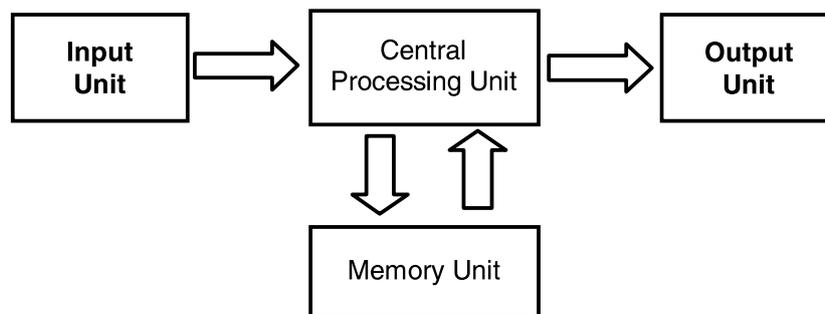
computer time and yet provide flexibility by means of controlling the operations. Basically any computer is supposed to carry out the following functions.

- Accept the data and program as input
- Store the data and program and retrieve as and when required.
- Process the data as per instructions given by the program and convert it into useful information
- Communicate the information as output

Based on the functionalities of the computer, the hardware components can be classified into four main units, namely

- Input Unit
- Output Unit
- Central Processing Unit
- Memory Unit

These units are interconnected by minute electrical wires to permit communication between them. This allows the computer to function as a system. The block diagram is shown below.



**Fig. 3.1 : Functional Units of a Computer System**

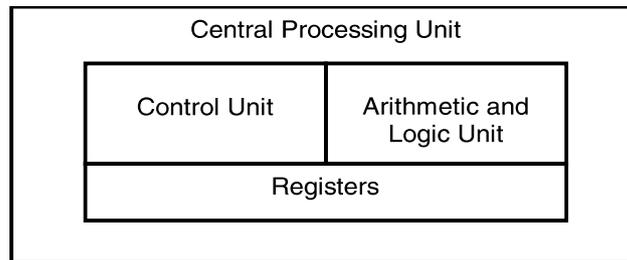
## Input Unit

A computer uses input devices to accept the data and program. Input devices allow communication between the user and the computer. In modern computers keyboard, mouse, light pen, touch screen etc, are some of the input devices.

## Output Unit

Similar to input devices, output devices have an interface between the computer and the user. These devices take machine coded output results from the processor and convert them into a form that can be used by human beings. In modern computers, monitors (display screens) and printers are the commonly used output devices

## Central Processing Unit



**Fig. 3.2. Central Processing Unit**

CPU is the brain of any computer system. It is just like the human brain that takes all major decisions, makes all sorts of calculations and directs different parts of the computer function by activating and controlling the operation. It consists of arithmetic and logic units, control unit and internal memory (registers). The control unit of the CPU coordinates the action of the entire system. Programs (software) provide the CPU, a set of instruction to follow and perform a specific task. Between any two components of the computer system, there is a pathway called a **bus** which allows for the data transfer between them.

Control unit controls all the hardware operations, ie, those of input units, output units, memory unit and the processor. The arithmetic and logic units in computers are capable of performing addition, subtraction, division and multiplication as well as some logical operations. The instructions and data are stored in the main memory so that the processor can directly fetch and execute them.

### **Memory Unit**

In the main memory, the computer stores the program and data that are currently being used. In other words since the computers use the stored program concept, it is necessary to store the program and data in the main memory before processing.

The main memory holds data and program only temporarily. Hence there is a need for storage devices to provide backup storage. They are called secondary storage devices or auxiliary memory devices. Secondary storage devices can hold more storage than main memory and is much less expensive.

#### **3.1.4 Stored Program Concept**

All modern computers use the stored program concept. This concept is known as the Von – Neumann concept due to the research paper published by the famous mathematician John Von Neuman. The essentials of the stored program concept are

- the program and data are stored in a primary memory (main memory)
- once a program is in memory, the computer can execute it automatically without manual intervention.
- the control unit fetches and executes the instructions in sequence one by one.
- an instruction can modify the contents of any location inThe stored program concept is the basic operating principle for every computer.

## 3.2 Central Processing Unit

### 3.2.1 Functions of a Central Processing Unit

The CPU is the brain of the computer system. It performs arithmetic operations as well as controls the input, output and storage units. The functions of the CPU are mainly classified into two categories :

- Co – ordinate all computer operations
- Perform arithmetic and logical operations on data

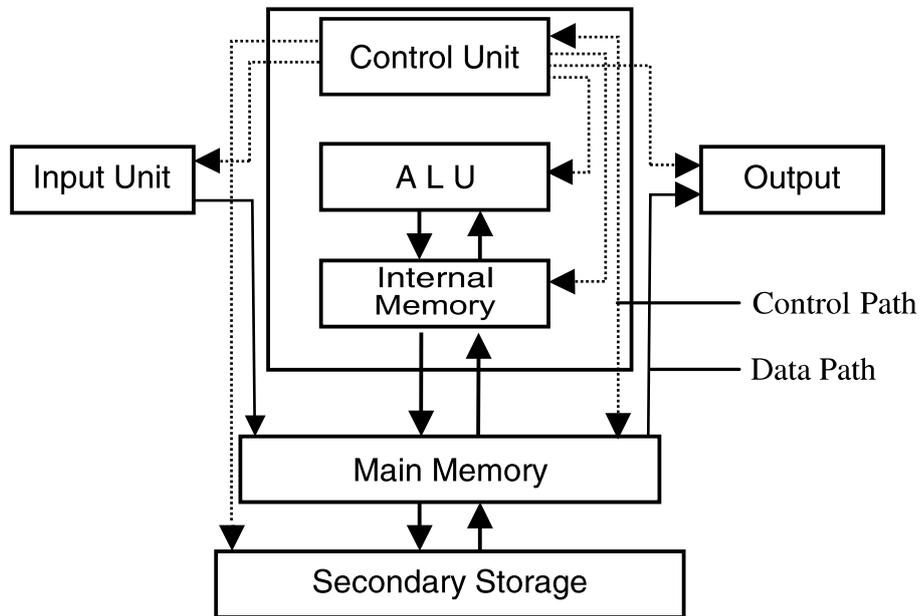
The CPU has three major components.

- Arithmetic and Logic Unit
- Control Unit
- Registers (internal memory)

**The arithmetic and logic unit (ALU)** is the part of CPU where actual computations take place. It consists of circuits which perform arithmetic operations over data received from memory and are capable of comparing two numbers.

The **control unit** directs and controls the activities of the computer system. It interprets the instructions fetched from the main memory of the computer, sends the control signals to the devices involved in the execution of the instructions.

While performing these operations the ALU takes data from the temporary storage area inside the CPU named **registers**. They are high-speed memories which hold data for immediate processing and results of the processing.



**Fig. 3.3 : Functions of a CPU**

### 3.2.2 Working with Central Processing Unit

The CPU is similar to a calculator, but much more powerful. The main function of the CPU is to perform arithmetic and logical operations on data taken from main memory. The CPU is controlled by a list of software instructions. Software instructions are initially stored in secondary memory storage device such as a hard disk, floppy disk, CD-ROM, or magnetic tape. These instructions are then loaded onto the computer's main memory.

When a program is executed, instructions flow from the main memory to the CPU through the bus. The instructions are then decoded by a processing unit called the instruction decoder that interprets and implements the instructions. The ALU performs specific operations such as addition, multiplication, and conditional tests on the data in its registers, sending the resulting data back to the main memory or storing it in another register for further use.

To understand the working principles of CPU, let us go through the various tasks involved in executing a simple program. This program performs arithmetic addition on two numbers. The algorithm of this program is given by

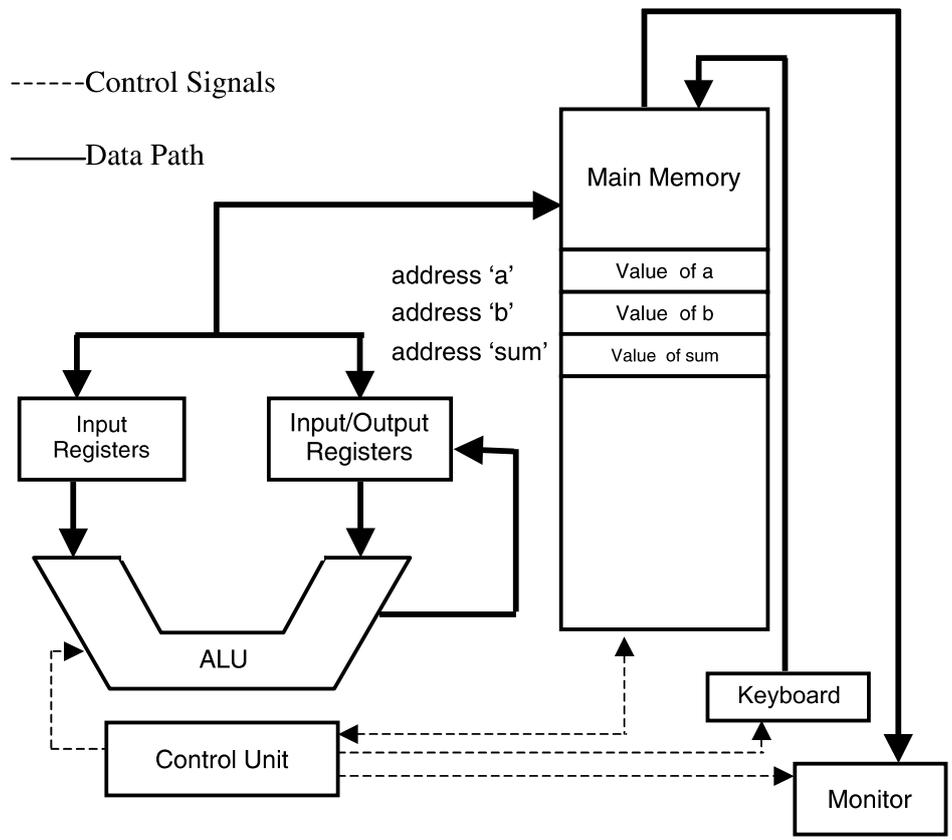
- (i) input the value of a
- (ii) input the value of b
- (iii)  $\text{sum} = a + b$
- (iv) output the value of sum

This program accepts two values from the keyboard, sums it and displays the sum on the monitor. The steps are summarized as follows :

1. The control unit recognizes that the program (set of instructions) has been loaded into the main memory. Then it begins to execute the program instructions one by one in a sequential manner.
2. The control unit signals the input device (say keyboard) to accept the input for the variable 'a'.
3. The user enters the value of 'a' on the keyboard.
4. The control unit recognizes and enables to route the data (value of a) to the pre-defined memory location (address of 'a').
5. The steps 2 to 4 will be repeated for the second input 'b'. The value of 'b' is stored in the memory location (address of 'b').
6. The next instruction is an arithmetic instruction. Before executing the arithmetic instruction, the control unit enables to send a copy of the values stored in address of 'a' and address of 'b' to the internal registers of the ALU and signals the ALU to perform the sum operation.
7. The ALU performs the addition. After the computation, the control unit enables to send the copy of the result back to the memory (address of 'sum').

8. Finally, the result is displayed on the monitor. The control unit enables to send the copy of the values of the address of 'sum' to the monitor (buffer) and signals it. The monitor displays the result.
9. Now this program execution is complete.

The data flow and the control flow of CPU during the execution of this program is given as,



**Fig. 3.4 : Working Principles of a CPU**

### 3.3 Arithmetic and Logic Unit - ALU

The ALU is the computer's calculator. It executes arithmetic and logical operations. The arithmetic operations include addition, subtraction, multiplication and division. The logical operation compares numbers, letters and special characters. The ALU also performs logic functions such as AND, OR and NOT.

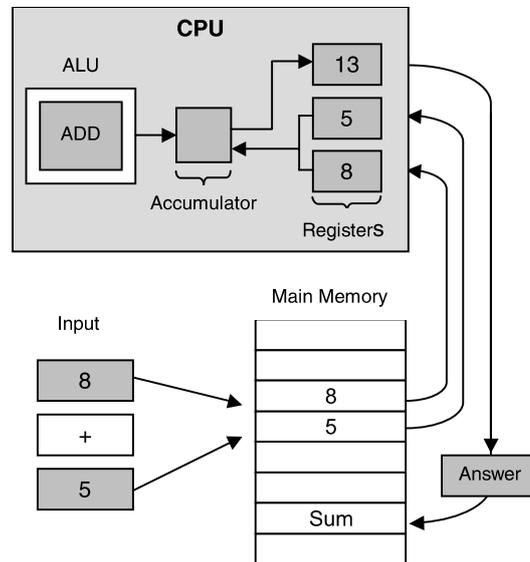
The ALU functions are directly controlled by the control unit. The control unit determines when the services of the ALU are needed, and it provides the data to be operated. The control unit also determines what is to be done with the results.

#### 3.3.1 Arithmetic Operations

Arithmetic operations include addition, subtraction, multiplication, and division. While performing these operations, the ALU makes use of the registers. Data to be arithmetically manipulated are copied from main memory and placed in registers for processing. Upon completion of the arithmetic operation, the result can be transferred from the register to the main memory. In addition to registers, the arithmetic unit uses one or more adders that actually perform arithmetic operations on the binary digits.

The arithmetic operation in adding two numbers can be demonstrated through following steps :

- Step 1** : The numbers (5 and 8) to be added up are put into two separate memory locations.
- Step 2** : The control unit fetches the two numbers from their memory locations into the data registers.
- Step 3** : The arithmetic unit looking at the operator (+) uses the accumulator and adds the two numbers.
- Step 4** : The ALU stores the result (13) in memory buffer register.
- Step 5** : Then the control unit stores the result into a user desired memory location, say 'sum'.



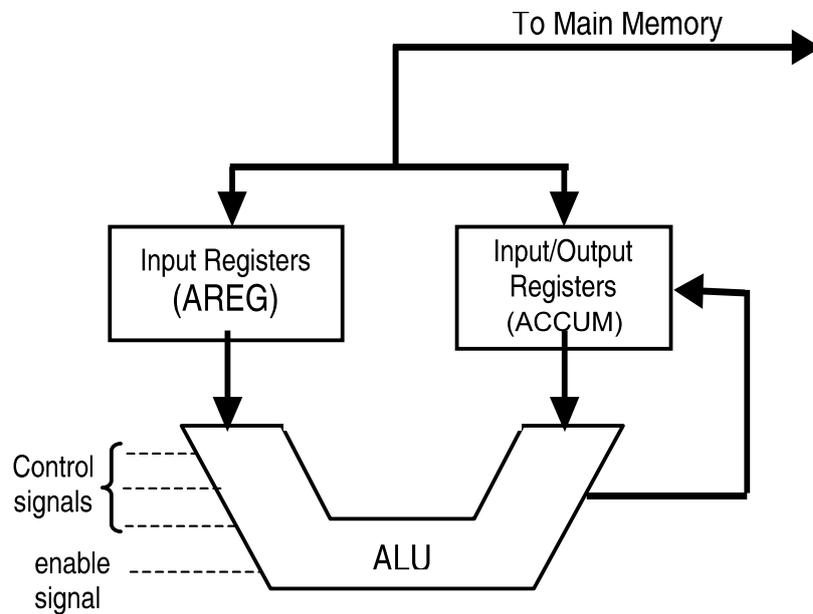
**Fig. 3.5 Arithmetic Logic Unit**

### 3.3.2 Logical Operations

The importance of the logic unit is to make logical operations. These operations include logically comparing two data items and take different actions based on the results of the comparison.

### 3.3.3 Functional Description

Some of the basic functions performed by the ALU are - add, subtract, logical AND, logical OR, shift left and shift right on two's complement binary numbers. The inputs to be calculated are stored in the input register (AREG) and the input / output register (ACCUM) for add, AND and OR functions. The shift left and shift right functions operate on the value in the ACCUM



**Fig. 3.6 : Functional Description of ALU**

The above figure illustrates the functional level block diagram of the ALU. The control unit controls the operations of the ALU by giving appropriate control signals to select a specific function and then enable the operation after the data are fed into the registers. The enable bit is made 1 after the data to be operated are transferred from main memory.

### **3.4 Memory Unit**

Memory units are the storage areas in a computer. The term “memory” usually refers to the main memory of the computer, whereas, the word “storage” is used for the memory that exists on disks, CDs, floppies or tapes. The main memory is usually called a physical memory which refers to the ‘chip’ (Integrated Circuit) capable of holding data and instruction.



**Fig. 3.7 Memory Unit**

There are different types of memory. They are Random Access Memory (RAM), Read Only Memory (ROM), Programmable Read-Only Memory (PROM), Erasable Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM).

### **Random Access Memory - RAM**

RAM is the most common type of memory found in the modern computers. This is really the main store and is the place where the program gets stored. When the CPU runs a program, it fetches the program instructions from the RAM and carries them out. If the CPU needs to store the results of the calculations it can store them in RAM. When we switch off a computer, whatever is stored in the RAM gets erased. It is a volatile form of memory.

### **Read Only Memory - ROM**

In ROM, the information is burnt (pre-recorded) into the ROM chip at manufacturing time. Once data has been written into a ROM chip, it cannot be erased but you can read it. When we switch off the computer, the contents of the ROM are not erased but remain stored permanently. ROM is a non-volatile memory. ROM stores critical programs such as the program that boots the computer.

## **Programmable Read Only Memory - PROM**

PROM is a memory on which data can be written only once. A variation of the PROM chip is that it is not burnt at the manufacturing time but can be programmed using PROM programmer or a PROM burner. PROM is also a non-volatile memory.

## **Erasable Programmable Read Only Memory - EPROM**

In EPROM, the information can be erased and reprogrammed using a special PROM – programmer. EPROM is non-volatile memory. A EPROM differs from a PROM in that a PROM can be written to only once and cannot be erased. But an ultraviolet light is used to erase the contents of the EPROM.

## **Electrically Erasable Programmable Read Only Memory - EEPROM**

EEPROM is a recently developed type of memory. This is equivalent to EPROM, but does not require ultraviolet light to erase its content. It can be erased by exposing it to an electrical charge. It is also non-volatile in nature. EEPROM is not as fast as RAM or other types of ROM. A flash memory is a special type of EEPROM that can be erased and reprogrammed.

The main memory must store many data items and have some way of retrieving them when they are needed. The memory can be compared to the boxes at a post office. Each box-holder has a box with a unique number which is called its address. This address serves to identify the box. The memory has a number of locations in its store. Each location in a memory has a unique number called its memory address. This serves to identify it for storage and retrieval.

Operations on memories are called *reads* and *writes*, defined from the perspective of a processor or other device that uses a memory: a write instruction transfers information from other device to

memory and a read instruction transfers information from the memory to other devices. A memory that performs both reads and writes is often called a RAM, *random access memory*. Other types of memories commonly used in systems are read-only memory.

### **Data Representation**

The smallest unit of information is a single digit called a 'bit' (**binary digit**), which can be either 0 or 1. The capacity of a memory system is represented by a unit called a *byte*, which is 8 bits of information. Memory sizes in modern systems range from 4MB (megabytes) in small personal computers up to several billion bytes (gigabytes, or GB) in large high-performance systems.

The performance of a memory system is defined by two different measures, the *access time* and the *memory cycle time*. Access time, also known as *response time* or *latency*, refers to how quickly the memory can respond to a read or write request. Memory cycle time refers to the minimum period between two successive requests.

The following terminology is used while discussing hierarchical memories:

- The registers (internal memory) are used to hold the instruction and data for the execution of the processor. Eventually the top of the hierarchy goes to the registers.
- The memory closest to the processor is known as a *cache*. It is a high speed memory that is much faster than the main memory.
- The next is the main memory which is also known as the *primary memory*.
- The low end of the hierarchy is the *secondary memory*.

The secondary memory is the memory that supplements the main memory. This is a long term non-volatile memory. It is external to the system nucleus and it can store a large amount of programs and data. The CPU does not fetch instructions of a program directly from the secondary memory. The program should be brought into the main memory from the secondary memory before being executed.

The secondary memory is cheaper compared to the main memory and hence a computer generally has limited amount of main memory and large amount of secondary memory.

### **3.5 Input and Output Devices**

The main function of a computer system is to process data. The data to be processed by the computer must be input to the system and the result must be output back to the external world.

#### **3.5.1 Input Devices**

An input device is used to feed data into a computer. For example, a keyboard is an input device. It is also defined as a device that provides communication between the user and the computer. Input devices are capable of converting data into a form which can be recognized by computer. A computer can have several input devices.

#### **Keyboard**

The most common input device is the keyboard. Keyboard consists of a set of typewriter like keys that enable you to enter data into a computer. They have alphabetic keys to enter letters, numeric keys to enter numbers, punctuation keys to enter comma, period, semicolon, etc., and special keys to perform some specific functions. The keyboard detects the key pressed and generates the corresponding ASCII codes which can be recognized by the computer.



**Fig. 3.8 Keyboard**

### **Mouse**

Mouse is an input device that controls the movement of the cursor on the display screen. Mouse is a small device, you can roll along a flat surface. In a mouse, a small ball is kept inside and touches the pad through a hole at the bottom of the mouse. When the mouse is moved, the ball rolls. This movement of the ball is converted into signals and sent to the computer. You will need to click the button at the top of the mouse to select an option. Mouse pad is a pad over which you can move a mouse. Mouse is very popular in modern computers.



**Fig. 3.9 Mouse**

## Scanner

Scanner is an input device that allows information such as an image or text to be input into a computer. It can read image or text printed on a paper and translate the information into a form that the computer can use. That is, it is used to convert images (photos) and text into a stream of data. They are useful for publishing and multi-media applications.



**Fig. 3.10 Scanner**

## Bar Code Reader

The barcode readers are used in places like supermarket, bookshops, etc. A bar code is a pattern printed in lines of different thickness. The bar-code reader scans the information on the bar-codes and transmits to the computer for further processing. The system gives fast and error-free entry of information into the computer.



**Fig.3.11 Bar Code and Reader**

## Digital Camera

The digital camera is an input device mainly used to capture images. The digital camera takes a still photograph, stores it and sends it as digital input to the computer. It is a modern and popular input device.



**Fig. 3.12 Digital Camera**

## Touch Sensitive Screen

Touch Sensitive Screen is a type of display screen that has a touch-sensitive panel. It is a pointing device that enables the user to interact with the computer by touching the screen. You can use your fingers to directly touch the objects on the screen. The touch screen senses the touch on the object (area pre-defined) and communicate the object selection to the computer.



**Fig. 3.13 Touch Sensitive Screen**

## Magnetic Ink Character Recognition (MICR)



Fig. 3.14 MICR Cheque

MICR is widely used by banks to process cheques. Human readable numbers are printed on documents such as cheque using a special magnetic ink. The cheque can be read using a special input unit, which can recognize magnetic ink characters. This method eliminates the manual errors. It also saves time, ensures security and accuracy of data.

## Optical Character Recognition (OCR)

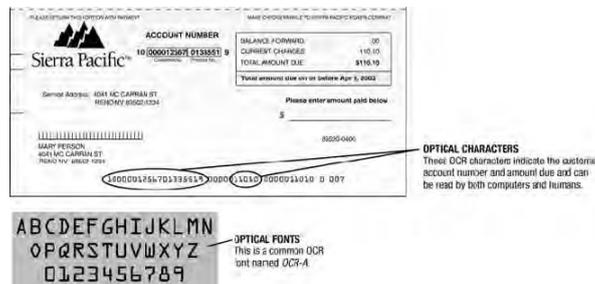
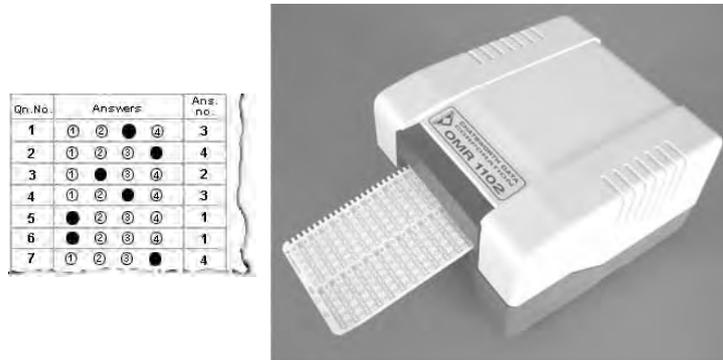


Fig. 3.15 OCR Sheet

The OCR technique permits the direct reading of any printed character like MICR but no special ink is required. With OCR, a user can scan a page from a book. The computer will recognize the characters in the page as letters and punctuation marks, and stores. This can be edited using a word processor.

## Optical Mark Reading and Recognition (OMR)



**Fig. 3.16 OMR Reader**

In this method special pre-printed forms are designed with boxes which can be marked with a dark pencil or ink. Such documents are read by a reader, which transcribes the marks into electrical pulses which are transmitted to the computer. They are widely used in applications like objective type answer papers evaluation in which large number of candidates appear, time sheets of factory employees etc.

### Light Pen

A light pen is a pointing device shaped like a pen and is connected to a monitor. The tip of the light pen contains a light-sensitive element which, when placed against the screen, detects



**Fig. 3.17 Light Pen**

the light from the screen enabling the computer to identify the location of the pen on the screen. Light pens have the advantage of 'drawing' directly onto the screen, but this can become uncomfortable, and they are not accurate.

### **Magnetic Reader**

Magnetic reader is an input device which reads a magnetic strip on a card. It is handy and data can be stored and retrieved. It also provides quick identification of the card's owner.

All the credit cards, ATM cards (banks), petro cards, etc. stores data in a magnetic strip which can be read easily by the magnetic reader.



**Fig. 3.18 Magnetic Reader**

### **Smart Cards**

This input device stores data in a microprocessor embedded in the card. This allows information, which can be updated, to be stored on the card. These data can be read and given as input to the computer for further processing. Most of the identification cards use this method to store and retrieve the vital information.



**Fig. 3.19 Smart Card Reader**

## Notes Taker

Notes taker is a device that captures natural handwriting on any surface onto a computer. Using an electronic pen, the **notes taker** displays the user's handwritten notes, memos or drawings on the computer, and stores the image for future use.



**Fig. 3.20 Notes Taker**

## Microphone

Microphone serves as a voice input device. It captures the voice data and input to the computer. Using the microphone along with speech recognition software can offer a completely new approach to input information into your computer.

Speech recognition programs, although not yet completely exact, have made great strides in accuracy as well as ease of use. The voice-in or speech recognition approach can almost fully replace the keyboard and mouse. Speech recognition can now open the computer world to those who may have been restricted due to a physical handicap. It can also be a boon for those who have never learned to type.



**Fig. 3.21 Microphone**

### 3.5.2 Output Devices

Output is anything that comes out of a computer. An output device is capable of presenting information from a computer. There are many output devices attached with the computers. But the monitors and printers are commonly used output devices.

#### Monitors

Monitor is a commonly used output device, sometimes called as display screen. It provides a visual display of data. Monitors are connected with the computer and are similar in appearance to a television set.



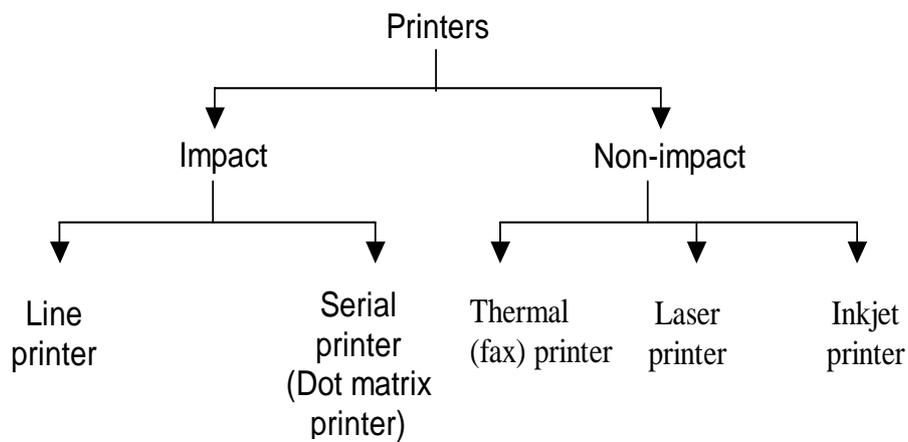
**Fig. 3.22 Monitor**

Initially there were only monochrome monitors. But gradually, we have monitors that display colour. Monitors display images and text. The smallest dot that can be displayed is called a pixel (picture element) The resolution of the screen improves as the number of pixels is increased. Most of the monitors have a 4 : 3 width to height ratio. This is called 'aspect ratio'.

The number of pixels that can be displayed vertically and horizontally gives the resolution of the monitor. The resolution of the monitor determines the quality of the display. Some popular resolutions are 640 x 480 pixels, 800 x 600 pixels and 1024 x 768 pixels. A resolution of 1024 x 768 pixels will produce sharper image than 640 x 480 pixels.

## Printers

Printer is an output device that prints text or images on paper or other media (like transparencies). By printing you create what is known as a 'hard copy'. There are different kinds of printers, which vary in their speed and print quality. The two main types of printers are impact printers and non-impact printers.



**Fig. 3.23 Types of Printers**

Impact printers include all printers that print by striking an ink ribbon. Impact printers use a print head containing a number of metal pins which strike an inked ribbon placed between the print head and the paper. Line printers, dotmatrix printers are some of the impact printers.

### Characteristics of Impact Printers

- Ø In impact printers, there is physical contact with the paper to produce an image.
- Ø Due to being robust and low cost, they are useful for bulk printing.

- Ø Impact printers are ideal for printing multiple copies (that is, carbon copies) because they can easily print through many layers of paper.
- Ø Due to its striking activity, impact printers are very noisy.
- Ø Since they are mechanical in nature, they tend to be slow.
- Ø Impact printers do not support transparencies.

Non-impact printers are much quieter than impact printers as their printing heads do not strike the paper. Non-impact printers include laser printers, inkjet printers and thermal printers.

### **Characteristics of Non-Impact Printers**

- Ø Non-impact printers are faster than impact printers because they have fewer moving parts.
- Ø They are quiet than impact printers because there is no striking mechanism involved.
- Ø They possess the ability to change typefaces automatically.
- Ø These printers produce high-quality graphics
- Ø These printers usually support the transparencies
- Ø These printers cannot print multipart forms because no impact is being made on the paper.

### **Line Printer**

Line printers are high-speed printers capable of printing an entire line at a time. A line printer can print 150 lines to 3000 lines per minute. The limitations of line printer are they can print only one font, they cannot print graphics, the print quality is low and they are noisy to operate. But it can print large volume of text data very fast compared to the other printers. It is also used to print on multipart stationaries to prepare copies of a document.



**Fig. 3.24 Line Printer**

### **Dot Matrix Printer**

The most popular serial printer is the dot matrix printer. It prints one line of 8 or 14 points at a time, with print head moving across a line. They are similar to typewriters. They are normally slow. The printing speed is around 300 characters per second. It uses multipart stationaries to prepare copies of a document.



**Fig. 3.25 Dot Matrix Printer**

### **Thermal Printer**

Thermal printers are printers that produce images by pushing electrically heated pins against special heat-sensitive paper. They are inexpensive and used widely in fax machines and calculators.



**Fig. 3.26 Thermal Printer**

Thermal printer paper tends to darken over time due to exposure to sunlight and heat. So the printed matters on the paper fade after a week or two. It also produces a poor quality print.

### **Laser Printers**

Laser printers use a laser beam and dry powdered ink to produce a fine dot matrix pattern. It can produce very good quality of graphic images. One of the chief characteristics of laser printers is their resolution – how many dots per inch (dpi) they lay down. The available resolutions range from 300 dpi at the low end to around 1200 dpi at the high end.



**Fig. 3.27 Laser Printer**

## Inkjet Printers

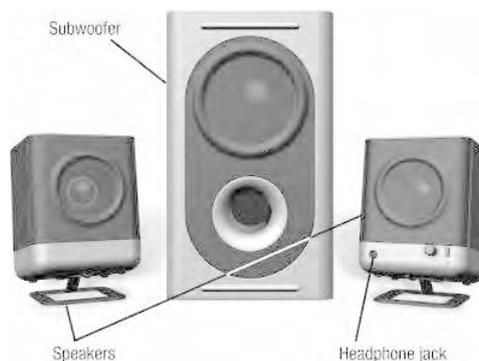
Inkjet printers use colour cartridges which combine magenta, yellow and cyan inks to create colour tones. A black cartridge is also used for crisp monochrome output. Inkjet printers work by spraying ionizing ink at a sheet of paper. Magnetized plates in the ink's path direct the ink onto the paper in the described shape.



**Fig. 3.28 Inkjet Printer**

## Speakers

The computer can also give produce voice output(audio data). Speaker serves as a voice output device. Using speakers along with speech synthesizer software, the computer can provide voice output. Voice output has become very common in many places like airlines, banks, automatic telephone enquiry system etc. Users can also hear music/songs using the voice output system.



**Fig. 3.29 Speakers**

## **Plotters**

Apart from the output devices like printers, plotters are also used to produce graphical output. Although printer output is very convenient for many purposes, the user needs to present the information graphically in order to understand its significance.

### **3.5.3 Storage Devices**

The computer may need to store data, programs etc. in a computer readable medium. This is called the secondary storage. Secondary storage is also called backup storage. Secondary storage can be used to transmit data to another computer either immediately or a latter time. This provides a mechanism for storing a large amount of data for a long period of time. Some of the commonly used storage devices are hard disks, magnetic tapes, floppy disks and CD-ROM.

To understand the physical mechanism of secondary storage devices one must have knowledge of magnetism, electronics and electro mechanical systems. The average time required to reach a storage location and obtain its contents is called its *access time*. In electromechanical devices with moving parts such as disks and tapes, the access time consists of a seek time required to position the read write head to a location and transfer time required to transfer the data to or from the device.

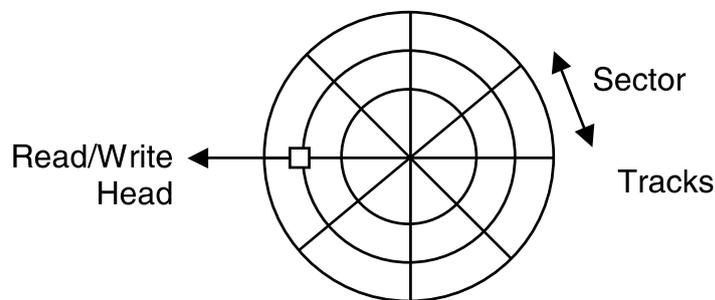
#### **Hard Disk**

Hard disk is a magnetic disk on which you can store computer data. The hard disk is a direct-access storage medium. This means you can store and retrieve data randomly.

Disk storage systems are essentially based on magnetic properties. The magnetic disk consists of high speed rotating surfaces coated with a magnetic recording medium. The rotating surface of the disk is a round flat plate. When writing data, a write

head magnetizes the particles on the disk surface as either north or south poles. When reading data, a read head converts the magnetic polarisations on the disk surface to a sequence of pulses. The read and write heads are generally combined into a single head unit. There may be more than one read/write head.

Data is arranged as a series of concentric rings. Each ring (called a track) is subdivided into a number of sectors, each sector holding a specific number of data elements (bytes or characters).



**Fig. 3.30 A track subdivided into sectors**

The smallest unit that can be written to or read from the disk is a sector. Once a read or write request has been received by the disk unit, there is a delay involved until the required sector reaches the read/write head. This is known as rotational latency, and on average is one half of the period of revolution.

The **storage capacity** of the disk is determined as **(number of tracks \* number of sectors \* bytes per sector \* number of read/write heads)** Thus, the data is stored as magnetized spots arranged in concentric circles (tracks) on the disk. Each track is divided into sectors. The arrangement of tracks and sectors on a disk is known as its 'format'.

High data rates demand that the disk rotates at a high speed (about 3,600 rpm). As the disk rotates read/write heads move to the correct track and fetch the desired data.



**Fig. 3.31 Hard Disk Drive**

The storage capacity of a hard disk can be Gigabytes (GB), i.e. thousands of Megabytes of information.

#### **Magnetic Tape**

A recording medium consisting of a thin tape with a coating of a fine magnetic strip, used for recording digital data. The tape itself is a strip of plastic coated with a magnetic recording medium.



**Fig. 3.32 Magnetic Tape Reader**

Bits are recorded as magnetic spots on the tape along several tracks. Usually, seven or nine bits are recorded simultaneously to form a character together with a parity bit. Read /write heads are mounted one in each track so that data can be recorded and read as a sequence of characters.

Data is stored in frames across the width of the tape. The frames are grouped into blocks or records which are separated from other blocks by gaps. Magnetic tape is a serial access medium, similar to an audio cassette, and so data cannot be randomly located. This characteristic has prompted its use in the regular backing up of hard disks.

### **Floppy Disk**



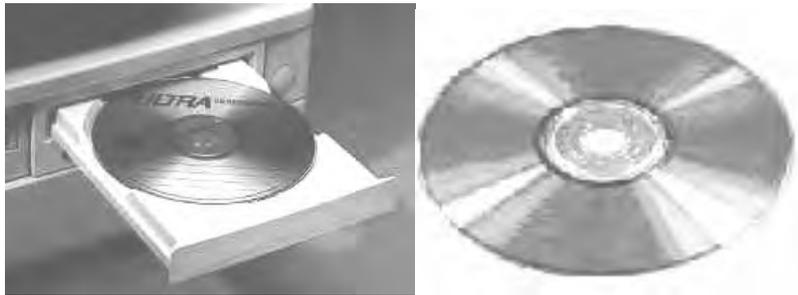
**Fig. 3.33 Floppy Disk**

The floppy drive uses a thin circular disk for data storage. It is a soft magnetic disk. It is a thin magnetic-coated disk contained in a flexible or semi-rigid protective jacket. The disk rotates at 360rpm. A read/write head makes physical contact with the disk surface. Data is recorded as a series of tracks subdivided into sectors.

The floppy disks are usually 3.5" in size. However, older floppy disks may be in use; these would be 5.25" in size or even 8" in size. A 3.5" floppy disk can hold 1.44 MB of data. Once data is stored on a floppy disk it can be 'write protected' by clicking a tab on the disk. This prevents any new data being stored or any old data being erased. Disk drives for floppy disks are called floppy drives. Floppy disks are slower to access than hard disks and have less storage capacity. It is less expensive and are portable. It can be accessed randomly.

## Optical Disk

Optical disks are a storage medium from which data is read and to which it is written by lasers. The optical disk is a random access storage medium; information can be easily read from any point on the disk. CD-ROM stands for Compact Disk - Read Only Memory.



**Fig. 3.34 Compact Disk**

It is now possible to have CD-ROMs where tracks of information can be written onto them by the user. These are called read/write CD-ROMs and these are becoming a popular and cheap method for storage.

## Summary

- \* Computers are often compared to human beings since both have the ability to accept data, store, work with it, retrieve and provide information.
- \* A computer system is the integration of physical entities called hardware and non-physical entities called software.
- \* The hardware components include input devices, processor, storage devices and output devices.

- \* The software items are programs and operating aids so that the computer can process data.
- \* A computer uses input devices to accept the data and program.
- \* In modern computers, monitors and printers are the commonly used output devices.
- \* CPU is the brain of any computer system. It consists of arithmetic and logic units, control unit and internal memory (registers).
- \* Control unit controls all the hardware operations, ie, those of input units, output units, memory unit and the processor.
- \* The arithmetic and logic units in computers are capable of performing addition, subtraction, division and multiplication as well as some logical operations.
- \* In the main memory, the computer stores the program and data that are currently being used.
- \* All modern computers use the stored program concept. This concept is due to John Von Neuman.
- \* The smallest unit of information is a single digit called a 'bit' (**binary digit**), which can be either 0 or 1.
- \* The secondary memory is the memory that supplements the main memory. This is a long term non-volatile memory.
- \* The most common input device is the keyboard.

- \* Mouse is an input device that controls the movement of the cursor on the display screen.
- \* Monitor is a commonly used output device.
- \* Some of the commonly used storage devices are hard disks, magnetic tapes, floppy disks and CD-ROM.

## Exercises

### I. Fill in the blanks

- 1) A computer system is the interpretation of physical entities called \_\_\_\_\_ and non-physical entities called \_\_\_\_\_
- 2) The computer uses \_\_\_\_\_ devices to accept data and program.
- 3) CPU stands for \_\_\_\_\_
- 4) ALU stands for \_\_\_\_\_
- 5) RAM stands for \_\_\_\_\_
- 6) ROM stands for \_\_\_\_\_
- 7) The stored program concept is conceived by \_\_\_\_\_
- 8) Main memory is also known as \_\_\_\_\_ memory.
- 9) The performance of the memory system is defined by \_\_\_\_\_ time and \_\_\_\_\_ time.
- 10) \_\_\_\_\_ supplements the main memory.
- 11) \_\_\_\_\_ is popular input device for GUI application.
- 12) \_\_\_\_\_ is a input device mainly used to capture images.
- 13) Monitor is a commonly used output unit, sometimes called as \_\_\_\_\_
- 14) The smallest dot that can be displayed on the monitor is called a \_\_\_\_\_
- 15) Printers can be classified into \_\_\_\_\_ and \_\_\_\_\_ printers.

### II. State whether the following are True or False

- 1) The operating system is a software.
- 2) Keyboard is an output device.
- 3) Touch sensitive screen is an input device.

- 4) Main memory is a non-volatile memory.
- 5) ALU performs arithmetic and logical operations.
- 6) Registers are a part of secondary storage.
- 7) Bar code reader is an output device.
- 8) Light pen is an input device.
- 9) Inkjet printers are impact printers.
- 10) CD – ROM stands for Compact Disk – Read Only Memory.

### **III. Answer the following**

- 1) How are the human being and the computers are related?
- 2) What are the components of the digital computer?
- 3) What are the functional units of a computer system?
- 4) Write the essentials of the stored program concept.
- 5) Write the main functions of the central processing unit.
- 6) What are the different types of main memory?
- 7) Define memory read and memory write operations.
- 8) What do you mean by memory access time?
- 9) What is the advantage of EEPROM over EPROM?
- 10) When do we use ROM?
- 11) What is an input device?
- 12) List few commonly used input devices.
- 13) What is an output device?
- 14) List few commonly used output devices.
- 15) What is a storage device?
- 16) List few commonly used storage devices.
- 17) What is the role of ALU?
- 18) What is a control unit?
- 19) What are registers?
- 20) What is a bus?

### **IV. Answer the following in detail.**

- 1) Describe in detail the various units of the Central Processing Unit.
- 2) Explain the working principle of CPU with an example.
- 3) Briefly explain various types of memory.

- 4) List a few commonly used input / output devices and explain them briefly.

## **V. Project**

- 1) List out the sequence of activities in executing the following program steps.
  - (i) input the value of a
  - (ii) input the value of b
  - (iii) multiply  $c = a * b$
  - (iv) output the value c
- 2) Describe a configuration for a personal computer system by identifying the input, output, processing and storage devices and their specifications.

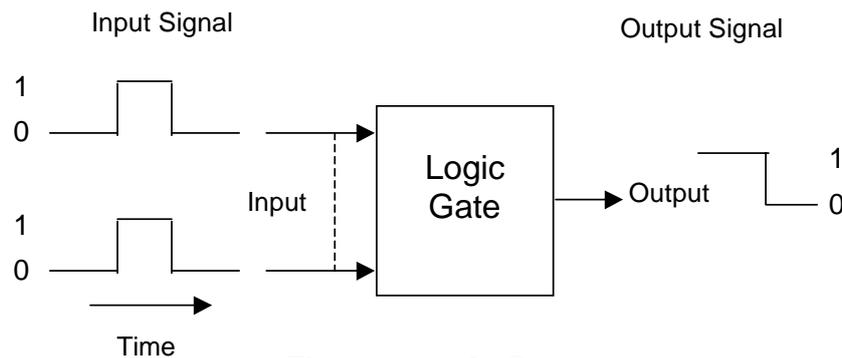
## CHAPTER 4

### WORKING PRINCIPLE OF DIGITAL LOGIC

#### 4.1 Logic Gates

A logic gate is an elementary building block of a digital circuit. It is a circuit with one output and one or more inputs. At any given moment, logic gate takes one of the two binary conditions *low* (0) or *high* (1), represented by different voltage levels.

A voltage level will represent each of the two logic values. For example +5V might represent a logic 1 and 0V might represent a logic 0.



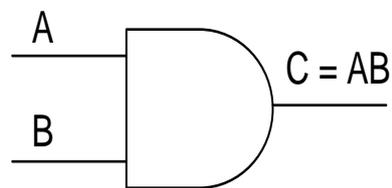
**Fig. 4.1 Logic Gate**

This diagram which represents a logic gate accept input signals (two or more) and produces an output signal based on the type of the gate. The input signal takes values '1' or '0'. The output signal also gives in the value '1' or '0'.

There are three fundamental logic gates namely, AND, OR and NOT. Also we have other logic gates like NAND, NOR, XOR and XNOR. Out of these NAND and NOR gates are called the universal gates, because the fundamental logic gates can be realized Through them. The circuit symbol and the truth table of these logic gates are explained here.

## AND Gate

The AND gate is so named because, if 0 is called “false” and 1 is called “true,” the gate acts in the same way as the logical “AND” operator. The output is “true” only when both inputs are “true”, otherwise, the output is “false”. In other words the output will be 1 if and only if both inputs are 1; otherwise the output is 0. The output of the AND gate is represented by a variable say C, where A and B are two and if input boolean variables. In boolean algebra, a variable can take either of the values ‘0’ or ‘1’. The logical symbol of the AND gate is



**Fig. 4.2 Logic symbol of AND Gate**

One way to symbolize the action of an AND gate is by writing the boolean function.

$$C = A \text{ AND } B$$

In boolean algebra the multiplication sign stands for the AND operation. Therefore, the output of the AND gate is

$$C = A \cdot B \quad \text{or}$$

simply  $C = AB$

Read this as “C equals A AND B”. Since there are two input variables here, the truth table has four entries, because there are four possible inputs : 00, 01, 10 and 11.

For instance, if both inputs are 0,

$$\begin{aligned}
 C &= A \cdot B \\
 &= 0 \cdot 0 \\
 &= 0
 \end{aligned}$$

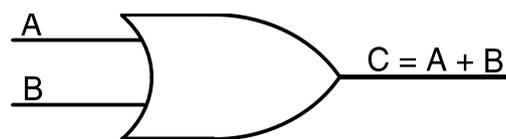
The truth table for AND Gate is

Input		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

**Table 4.1 Truth Table for AND Gate**

### OR Gate

The *OR gate* gets its name from the fact that it behaves like the logical inclusive “OR”. The output is “true” if either or both of the inputs are “true”. If both inputs are “false,” then the output is “false”. In other words the output will be 1 if and only if one or both inputs are 1; otherwise, the output is 0. The logical symbol of the OR gate is



**Fig. 4.3 Logic symbol of OR Gate**

The OR gate output is

$$C = A \text{ OR } B$$

We use the + sign to denote the OR function. Therefore,

$$C = A + B$$

Read this as “C equals A OR B”.

For instance, if both the inputs are 1

$$C = A + B = 1 + 1 = 1$$

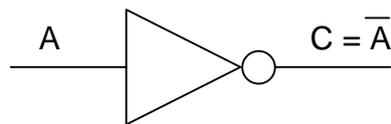
The truth table for OR gate is

Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

**Table 4.2 Truth Table for OR Gate**

### **NOT Gate**

The NOT gate, called a logical inverter, has only one input. It reverses the logical state. In other words the output C is always the complement of the input. The logical symbol of the NOT gate is



**Fig. 4.3 Logic symbol of NOT Gate**

The boolean function of the NOT gate is

$$C = \text{NOT } A$$

In boolean algebra, the overbar stands for NOT operation. Therefore,

$$C = \bar{A}$$

Read this as “C equals NOT A” or “C equals the complement of A”.

$$\begin{aligned} &\text{If A is 0,} \\ &C = \overline{0} = 1 \end{aligned}$$

On the otherhand, if A is 1,

$$C = \overline{1} = 0$$

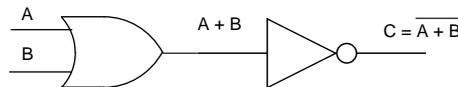
The truth table for NOT gate is

Input	Output
A	C
1	0
0	1

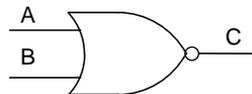
**Table 4.3 Truth Table for NOT Gate**

NOR Gate

The NOR gate circuit is an OR gate followed by an inverter. Its output is “true” if both inputs are “false” Otherwise, the output is “false”. In other words, the only way to get ‘1’ as output is to have both inputs ‘0’. Otherwise the output is 0. The logic circuit of the NOR gate is



**Fig. 4.5 Logic Circuit of NOR Gate**



**Fig. 4.6 Logic symbol of NOR Gate**

The output of NOR gate is

$$C = \overline{A + B}$$

Read this as “C equals NOT of A OR B” or “C equals the complement of A OR B”.

For example if both the inputs are 0,

$$C = (\overline{0 + 0}) = \overline{0} = 1$$

The truth table for NOR gate is

Input		Output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Table 4.4 Truth Table for NOR Gate

### Bubbled AND Gate

The Logic Circuit of Bubbled AND Gate

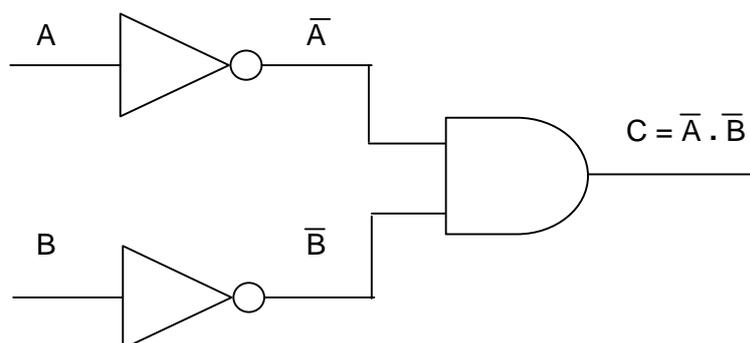
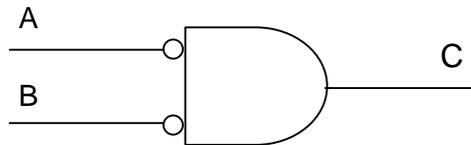


Fig. 4.7 Logic Circuit of Bubbled AND Gate

In the above circuit, invertors on the input lines of the AND gate gives the output as

$$C = \bar{A} \cdot \bar{B}$$

This circuit can be redrawn as the bubbles on the inputs, where the bubbles represent inversion.



**Fig. 4.8 Logic Symbol of Bubbled AND Gate**

We refer this as bubbled AND gate. Let us analyse this logic circuit for all input possibilities.

If A = 0 and B = 0                       $C = (\bar{0} \cdot \bar{0}) = 1 \cdot 1 = 1$

If A = 0 and B = 1                       $C = (\bar{0} \cdot \bar{1}) = 1 \cdot 0 = 0$

If A = 1 and B = 0                       $C = (\bar{1} \cdot \bar{0}) = 0 \cdot 1 = 0$

If A = 1 and B = 1                       $C = (\bar{1} \cdot \bar{1}) = 0 \cdot 0 = 0$

Here the truth table is

Input		Output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

**Table 4.5 Truth Table for Bubbled AND Gate**

You can see that, a bubbled AND gate produces the same output as a NOR gate. So, you can replace each NOR gate by a bubbled AND gate. In other words the circuits are interchangeable.

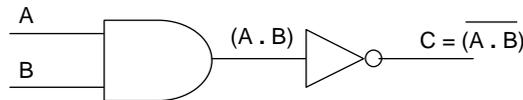
Therefore

$$\overline{(A + B)} = \bar{A} \cdot \bar{B}$$

which establishes the De Morgan's first theorem.

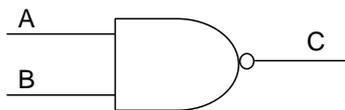
### NAND Gate

The *NAND gate* operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "AND" followed by inversion. The output is "false" if both inputs are "true", otherwise, the output is "true". In other words the output of the NAND gate is 0 if and only if both the inputs are 1, otherwise the output is 1. The logic circuit of NAND gate is



**Fig. 4.9 Logic Circuit of NAND Gate**

The logical symbol of NAND gate is



**Fig. 4.10 Logic Symbol of NAND Gate**

The output of the NAND gate is

$$C = \overline{(A \cdot B)}$$

Read this as "C equals NOT of A AND B" or "C equals the complement of A AND B".

For example if both the inputs are 1

$$C = \overline{(1 \cdot 1)} = \bar{1} = 0$$

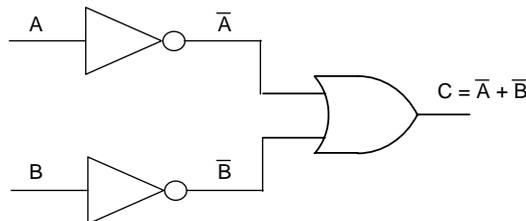
The truth table for NAND gate is

Input		Output
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

**Table 4.6 Truth Table for NAND Gate**

### Bubbled OR Gate

The logic circuit of bubbled OR gate is

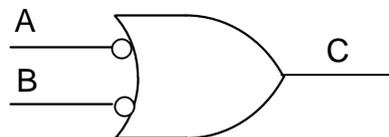


**Fig. 4.11 Logic Circuit of Bubbled OR Gate**

The output of this circuit can be written as

$$C = \bar{A} + \bar{B}$$

The above circuit can be redrawn as the bubbles on the input, where the bubbles represents the inversion.



**Fig. 4.12 Logic Symbol of Bubbled OR Gate**

We refer this as bubbled OR gate. The truth table for the bubbled OR is

Input		Output
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

**Table 4.7 Truth Table for Bubbled OR Gate**

If we compare the truth tables of the bubbled OR gate with NAND gate, they are identical. So the circuits are interchangeable.

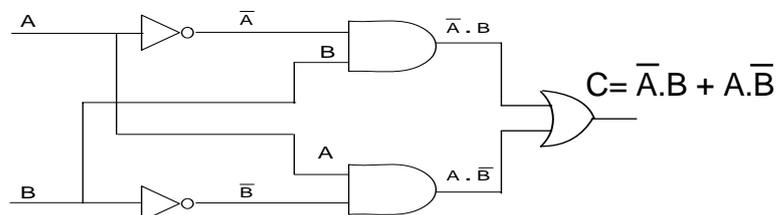
Therefore

$$\overline{(A \cdot B)} = \bar{A} + \bar{B}$$

which establishes the De Morgan's second theorem.

### XOR Gate

The *XOR (exclusive-OR) gate* acts in the same way as the logical "either/or." The output is "true" if either, but not both, of the inputs are "true." The output is "false" if both inputs are "false" or if both inputs are "true." Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same. The logic circuit of XOR gate is



**Fig. 4.13 Logic Circuit of XOR Gate**

The output of the XOR gate is

$$C = \bar{A} B + A \bar{B}$$

The truth table for XOR gate is

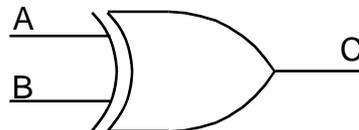
Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

**Table 4.8 Truth Table for XOR Gate**

In boolean algebra, exclusive-OR operator is  $\oplus$  or “encircled plus”.

Hence,  $C = A \oplus B$

The logical symbol of XOR gate is



**Fig. 4.14 Logic Symbol of XOR Gate**

### **XNOR Gate**

The *XNOR (exclusive-NOR) gate* is a combination XOR gate followed by an inverter. Its output is “true” if the inputs are the same, and “false” if the inputs are different. In simple words, the output is 1 if the input are the same, otherwise the output is 0. The logic circuit of XNOR gate is



**Fig. 4.15 Logic Circuit of XNOR Gate**

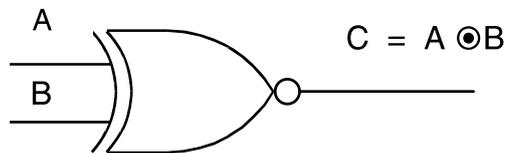
The output of the XNOR is NOT of XOR

$$\begin{aligned}
 C &= \overline{A \oplus B} \\
 &= \overline{\bar{A}.B + A.\bar{B}} \\
 &= AB + \bar{A}\bar{B} \quad \text{(Using De Morgan's Theorem)}
 \end{aligned}$$

In boolean algebra,  $\odot$  or “included dot” stands for the XNOR.

Therefore,  $C = A \odot B$

The logical symbol is



**Fig. 4.16 Logic Symbol of XNOR Gate**

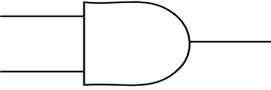
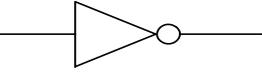
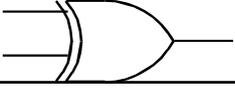
The truth table for XNOR gate is

Input		Output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

**Table 4.9 Truth Table for XNOR Gate**

Using combinations of logic gates, complex operations can be performed. In theory, there is no limit to the number of gates that can be arranged together in a single device. But in practice, there is a limit to the number of gates that can be packed into a given physical space. Arrays of logic gates are found in digital integrated circuits.

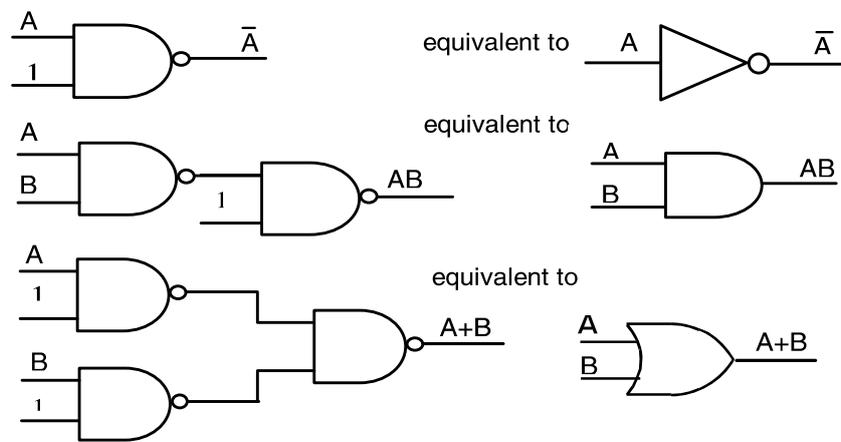
The logic gates and their corresponding truth tables are summarized in the following table.

Logical Gates	Symbol	Truth Table															
AND		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	A B	0	0	0	0	1	0	1	0	0	1	1	1
A	B	A B															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
OR		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A+B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	A+B	0	0	0	0	1	1	1	0	1	1	1	1
A	B	A+B															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
NOT		<table border="1"> <thead> <tr> <th>A</th> <th><math>\bar{A}</math></th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	$\bar{A}$	0	1	1	0									
A	$\bar{A}$																
0	1																
1	0																
NAND		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>AB</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	AB	0	0	1	0	1	1	1	0	1	1	1	0
A	B	AB															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
NOR		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A+B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	A+B	0	0	1	0	1	0	1	0	0	1	1	0
A	B	A+B															
0	0	1															
0	1	0															
1	0	0															
1	1	0															
XOR		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A+B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	A+B	0	0	0	0	1	1	1	0	1	1	1	0
A	B	A+B															
0	0	0															
0	1	1															
1	0	1															
1	1	0															
XNOR		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	A B	0	0	1	0	1	0	1	0	0	1	1	1
A	B	A B															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

**Table 4.10 summary of Logic Gates**

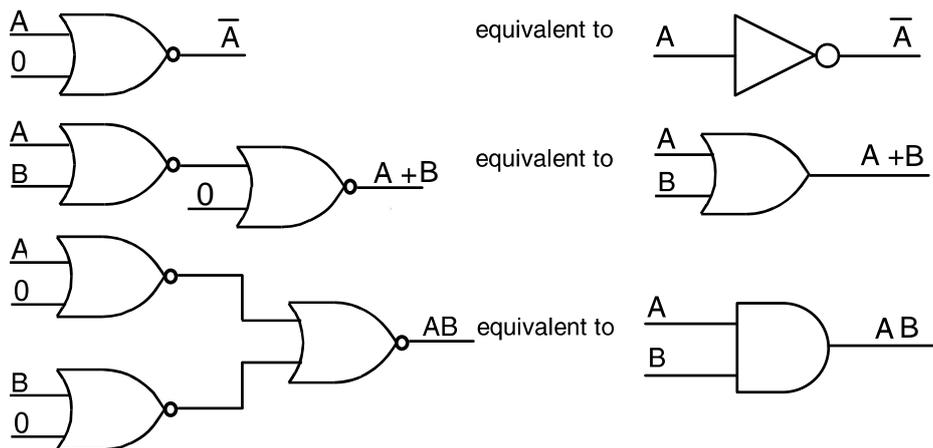
## Universality of NAND and NOR gates

We know that all boolean functions can be expressed in terms of the fundamental gates namely AND, OR and NOT. In fact, these fundamental gates can be expressed in terms of either NAND gates or NOR gates. NAND gates can be used to implement the fundamental logic gates NOT, AND and OR. Here A and B denote the logical states (Input).



**Fig. 4.17 Universality of NAND Gates**

NOR gates can also be used to implement NOT, OR and AND gates.



**Fig. 4.18 Universality of NOR Gates**

## 4.2 Conversion of Boolean Function

To build more complicated boolean functions, we use the *AND*, *OR*, and *NOT* operators and combine them together. Also, it is possible to convert back and forth between the three representations of a boolean function (equation, truth table, and logic circuit). The following examples show how this is done.

### Converting a Boolean Equation to a Truth Table

Truth table lists all the values of the boolean function for each set of values of the variables. Now we will obtain a truth table for the following boolean function

$$D = (A \cdot B) + \bar{C}$$

Clearly,  $D$  is a function of three input variables  $A$ ,  $B$ , and  $C$ . Hence the truth table will have  $2^3 = 8$  entries, from 000 to 111. Before determining the output  $D$  in the table, we will compute the various intermediate terms like  $A \cdot B$  and  $\bar{C}$  as shown in the table below.

For instance, if  $A = 0$ ,  $B = 0$  and  $C = 0$   
then

$$\begin{aligned} D &= (A \cdot B) + \bar{C} \\ &= (0 \cdot 0) + \bar{0} \\ &= 0 + \bar{0} \\ &= 0 + 1 \\ &= 1 \end{aligned}$$

Here we use the hierarchy of operations of the boolean operators NOT, AND and OR over the parenthesis.

The truth table for the boolean function is

Input			Intermediate		Output
A	B	C	$A \cdot B$	$\overline{C}$	D
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

### Converting a Boolean Equation to a Logic Circuit

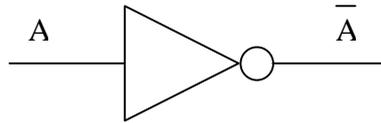
The boolean function is realized as a logic circuit by suitably arranging the logic gates to give the desired output for a given set of input. Any boolean function may be realized using the three logical operations NOT, AND and OR. Using gates we can realize boolean function.

Now we will draw the logic circuit for the boolean function.

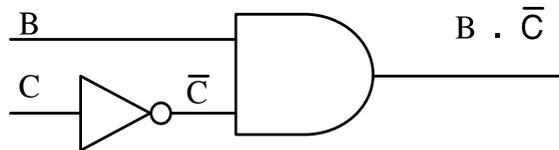
$$E = \overline{A} + (B \cdot \overline{C}) + \overline{D}$$

This boolean function has four inputs A, B, C, D and an output E. The output E is obtained by ORing the individual terms given in the right side of the boolean function. That is, by ORing the terms  $\overline{A}$ ,  $(B \cdot \overline{C})$  and  $\overline{D}$ .

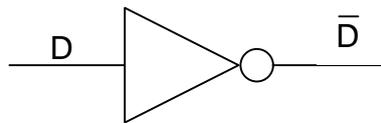
The first term  $\bar{A}$ , which is the complement of the given input A, is realized by



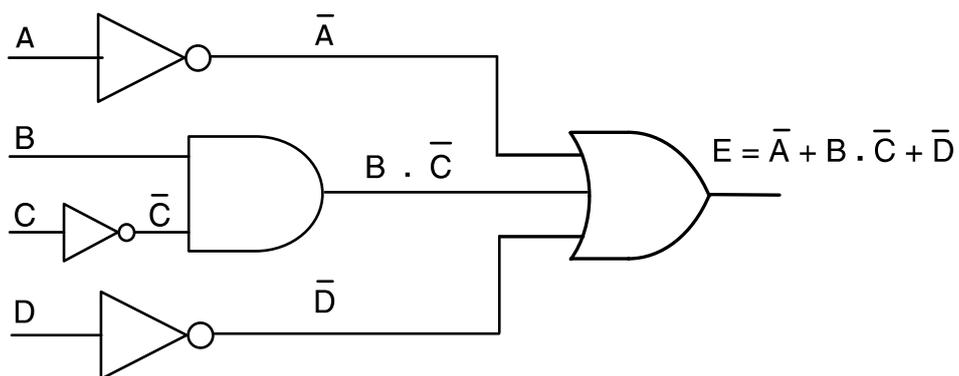
The second term is  $(B \cdot \bar{C})$ . Here the complement of C is AND with B. The logic circuit is realized by



The third term, which is the complement of D is realized by

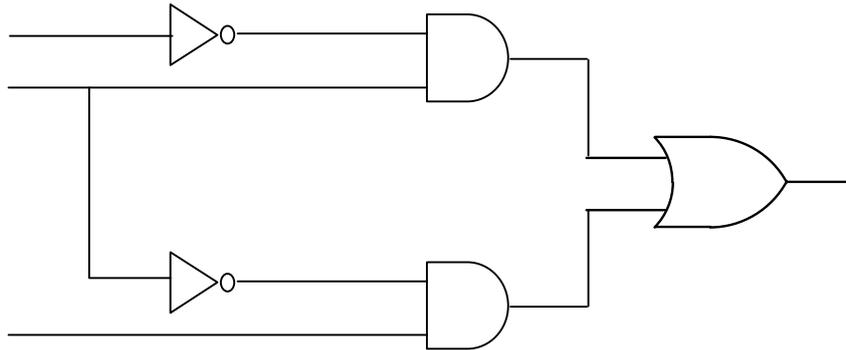


The output D is realized by ORing the output of the three terms. Hence the logic circuit of the boolean equation is

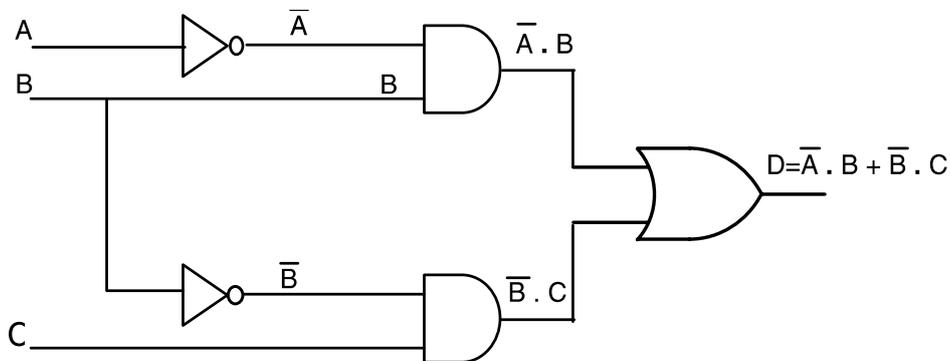


## Converting a Logic Circuit to a Boolean Function

As a reversal operation, the realization of the logic circuit can be expressed as a boolean function. Let us formulate an expression for the output in terms of the inputs for the given the logic circuit



To solve this, we simply start from left and work towards the right, identifying and labeling each of the signals that we encounter until we arrive at the expression for the output. The labeling of all the signals is shown in the figure below. Let us label the input signals as A, B, C and the output as D.



Hence the boolean function corresponding to the logic circuit can be written as

$$D = \bar{A} \cdot B + \bar{B} \cdot C$$

## Converting a Truth Table to a Boolean Function

There are many ways to do this conversion. A simplest way is to write the boolean function as an OR of minterms. A minterm is simply the ANDing of all variables, and assigning bars (NOT) to variables whose values are 0.

For example, assuming the inputs to a 4-variable boolean function as A, B, C, and D the minterm corresponding to the input 1010 is :  $A \cdot \bar{B} \cdot C \cdot \bar{D}$ . Notice that this minterm is simply the AND of all four variables, with B and D complemented. The reason that B and D are complemented is because for this input,  $B = D = 0$ . As another example, for the input 1110, only  $D = 0$ , and so the corresponding minterm is  $A \cdot B \cdot C \cdot \bar{D}$ .

For example let us write a boolean function for the following truth table

Input			Output
A	B	C	D
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

To do this problem, we first circle all of the rows in the truth table which have an output  $D = 1$ . Then for each circled row, we write the corresponding minterm. This is illustrated in the table below.

Input			Output	
A	B	C	D	Minterms
0	0	0	1	$\bar{A} \cdot \bar{B} \cdot \bar{C}$
0	0	1	0	
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$
0	1	1	0	
1	0	0	1	$A \cdot \bar{B} \cdot \bar{C}$
1	0	1	0	
1	1	0	1	$A \cdot B \cdot \bar{C}$
1	1	1	0	

Finally, the boolean expression for D is obtained by ORing all of the minterms as follows:

$$D = (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot \bar{C}) + (A \cdot \bar{B} \cdot \bar{C}) + (A \cdot B \cdot \bar{C})$$

### Design of Logic Circuit

There are many steps in designing a logic circuit. First, the problem is stated (in words). Second, from the word description, the inputs and outputs are identified, and a block diagram is drawn. Third, a truth table is formulated which shows the output of the system for every possible input. Fourth, the truth table is converted to a boolean function. Fifth, the boolean function is converted to a logic circuit diagram. Finally, the logic circuit is built and tested.

Let us consider the design aspects of a 2-input /single output system which operates as follows: The output is 1 if and only if precisely one of the inputs is 1; otherwise, the output is 0.

**Step 1: Statement of the problem .** Given above.

**Step 2: Identify inputs and outputs.** It is clear from the statement of the problem that we need two inputs, say A and B, and one output, say C. A block diagram for this system is



**Step 3: Formulate truth table.** The truth table for this problem is given below. Notice that the output is 1 if only one of the inputs is 1. otherwise the output is '0'.

Input		Output	
A	B	C	
0	0	0	
0	1	1	
1			
1			

Input		Output	Minterm
A	B	C	
0	0	0	
0	1	1	A B
1	0	1	A B
1	1	0	

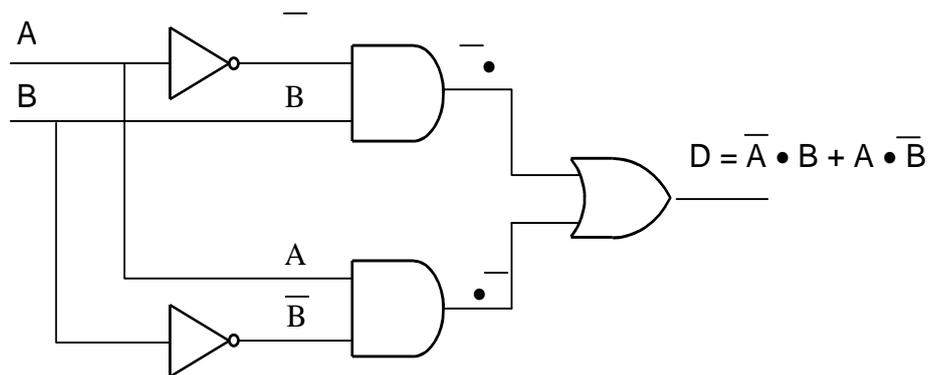
**Step 4: Convert the truth table to minterms for the rows in the truth table where the output is '1'**

By ORing the minterms, we obtain the boolean function corresponding to the truth table as

$$D = (\bar{A} \bullet B) + (A \bullet \bar{B})$$

### Step 5: Realization of the Boolean function into a Logic Circuit Diagram.

The logic circuit diagram corresponding to this boolean function is given below.



### 4.3 Half Adder

The circuit that performs addition within the Arithmetic and Logic Unit of the CPU are called adders. A unit that adds two binary digits is called a half adder and the one that adds together three binary digits is called a full adder.

A half adder sums two binary digits to give a sum and a carry. This simple addition consists of four possible operations.

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

The first three operations produce a single digit sum, while the fourth one produces two digit sum. The higher significant bit in this operation is called a carry. So the carry of the first three operations are '0', where the fourth one produces a carry '1'.

The boolean realization of binary addition is shown in the truth table. Here A and B are inputs to give a sum S and a carry C.

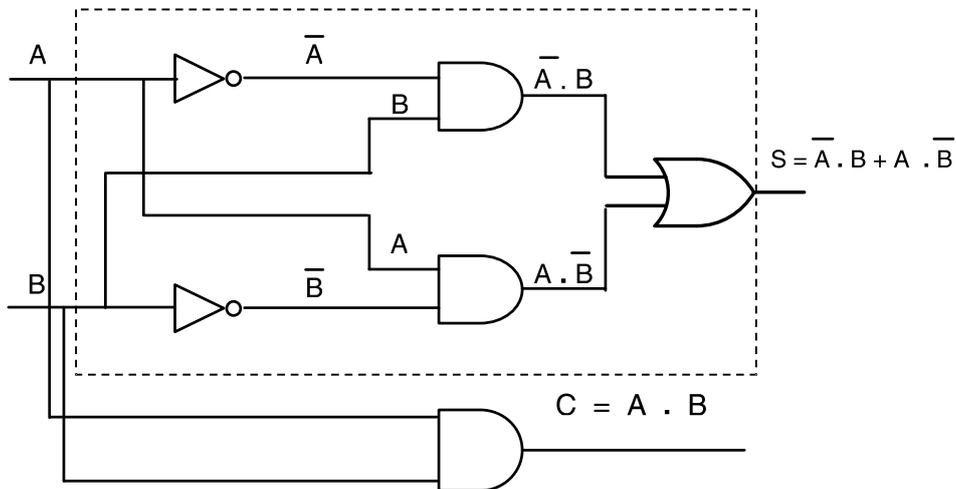
Input		Sum	Minterms Of S	Carry	Minterms of C
A	B	S		C	
0	0	0	-	0	
0	1	1	$\bar{A}.B$	0	
1	0	1	$A.\bar{B}$	0	
1	1	0		1	$A.B$

The boolean functions corresponding to the sum and carry are

$$S = \bar{A} . B + A . \bar{B}$$

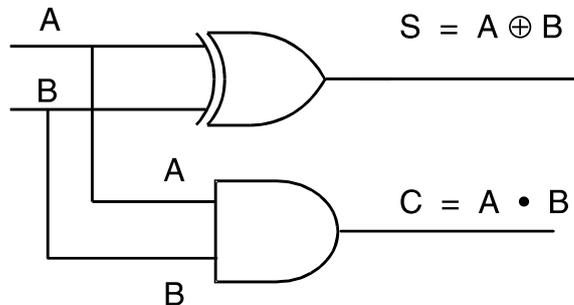
$$C = A . B$$

Which can be realized using logic circuit as,



**Fig. 4.19 Logic Circuit of Half Adder**

which is further simplified as



In a half adder, an AND gate is added in parallel to the XOR gate to generate the carry and sum respectively. The 'sum' column of the truth table represents the output of the XOR gate and the 'carry' column represents the output of the AND gate.

#### 4.4 Full Adder

A half adder logic circuit is a very important component of computing systems. As this circuit cannot accept a carry bit from a previous addition, it is not enough to fully perform additions for binary number greater than 1. In order to achieve this a full adder is required.

A full adder sums three input bits. It consists of three inputs and two outputs. Two of the inputs represent the two significant bits to be added and the third one represents the carry from the previous significant position.

Here A, B referred as the inputs, C1 as carry input from the previous stage, C2 as carry output and S as sum.

Input			Output	
A	B	C <sub>1</sub>	C <sub>2</sub>	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

**Table 4.12 Truth Table for Full Adder**

The carry bit  $C_2$  is 1 if both  $A$  and  $B$  are 1, or exactly one of  $A$  and  $B$  is 1 and the input carry,  $C_1$  is 1. The sum bit  $S$  is 1 if there exist odd number of '1's of the three inputs.  $S$  is the XOR of the three inputs. Hence, the full adder can be realized as shown below.

By ORing the minterms of the full adder truth table, the sum and carry can be written as

$$S = \bar{A} \bar{B} C_1 + \bar{A} B \bar{C}_1 + A \bar{B} \bar{C}_1 + A B C_1$$

$$C_2 = \bar{A} B C_1 + A \bar{B} C_1 + A B \bar{C}_1 + A B C_1$$

Consider

$$\begin{aligned} (A \oplus B) \oplus C_1 &= (\bar{A} B + A \bar{B}) \oplus C_1 \\ &= \overline{(\bar{A} B + A \bar{B})} C_1 + (\bar{A} B + A \bar{B}) \bar{C}_1 \\ &= (\overline{\bar{A} B} \overline{A \bar{B}}) C_1 + (\bar{A} B + A \bar{B}) \bar{C}_1 \\ &= ((A + \bar{B})(\bar{A} + B)) C_1 + (\bar{A} B + A \bar{B}) \bar{C}_1 \end{aligned}$$

$$= \cancel{A\bar{A}C_1} + ABC_1 + \cancel{\bar{A}B\bar{C}_1} + \cancel{B\bar{B}C_1} + \bar{A}B\bar{C}_1 + A\bar{B}\bar{C}_1$$

(Since,  $A\bar{A} = B\bar{B} = 0$ )

$$= S$$

**Also**

$$C_2 = \bar{A}B C_1 + A\bar{B} C_1 + AB\bar{C}_1 + ABC_1$$

$$= (\bar{A}B + A\bar{B})C_1 + AB(C_1 + \bar{C}_1)$$

$$= (A \oplus B) C_1 + AB$$

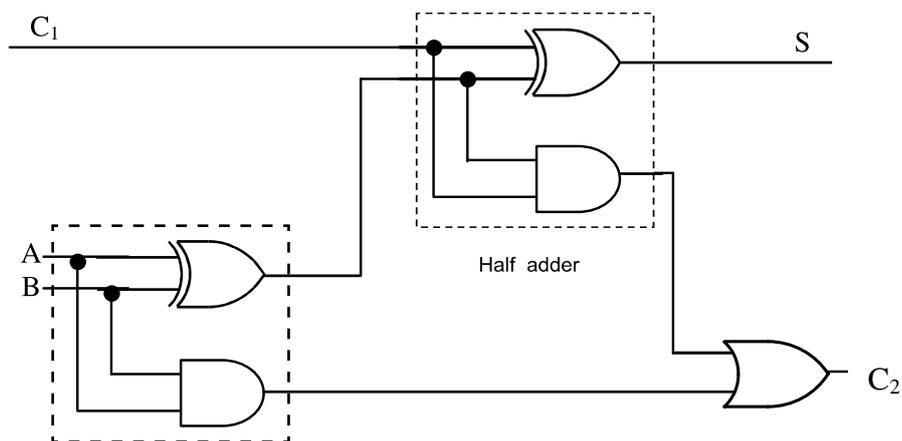
**Hence**

$$S = (A \oplus B) \oplus C_1 \text{ and}$$

$$C_2 = (A \oplus B) C_1 + AB$$

To realize the full adder we need two 2-input XOR, two 2-input AND gates and a 2-input OR gate.

Hence the full adder can be realized as.



**Fig. 4.20 Logic Circuit of Full Adder**

Notice that the full adder can be constructed from two half adders and an OR gate.

If the logic circuit outputs are based on the inputs presented at that time, then they are called combinational circuit. The half adder and full adder circuits are the examples for the combinational circuits. On the other hand, if the logic circuit outputs are based on, not only the inputs presented at that time, but also the previous state output, then they are called sequential circuits.

There are two main types of sequential circuits. A synchronous sequential circuit is a system whose output can be defined from its inputs at discrete instant of time. The output of the asynchronous sequential circuit depends upon the order in which its input signals change at any instance of time. The flip-flop circuit is an example of sequential circuit.

#### **4.5 The Flip-Flop**

A flip flop is a circuit which is capable of remembering the value which is given as input. Hence it can be used as a basic memory element in a memory device. These circuits are capable of storing one bit of information.

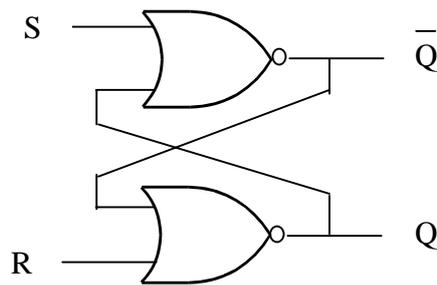
##### **Basic flip-flops**

A flip-flop circuit can be constructed using either two NOR gates or two NAND gates.

A common example of a circuit employing sequential logic is the flip-flop, also called a bi-stable gate. A simple flip-flop has two stable states. The flip-flop maintains its states indefinitely until an input pulse called a trigger is received. If a trigger is received, the flip-flop outputs change their states according to defined rules, and remain in those states until another trigger is received.

## Flip – Flop Circuit using NOR Gates

By cross-coupling two NOR gates, the basic operation of a flip-flop could be demonstrated. In this circuit the outputs are fed back again to inputs.



**Fig. 4.21 Flip Flop Circuit using NOR Gates**

The flip-flop circuit has two outputs, one for the normal value  $Q$  and another for the complement value  $\bar{Q}$ . It also has two inputs  $S$  (set) and  $R$  (reset). Here, the previous output states are fed back to determine the current state of the output.

The NOR basic flip-flop circuit operates with inputs normally at '0' unless the state of the flip-flop has to be changed.

As a starting point, we assume  $S = 1$  and  $R = 0$ . This makes  $\bar{Q} = 0$ . This  $\bar{Q} = 0$  is again given along with  $R = 0$  to make  $Q = 1$ .

ie. when  $S = 1$  and  $R = 0$  make  $Q = 1$  and  $\bar{Q} = 0$ .

When the input  $S$  returns to '0', the output remains the same, because the output  $Q$  remain as '1' and  $\bar{Q}$  as '0'.

ie. when  $S = 0$  and  $R = 0$  make  $Q = 1$  and  $\bar{Q} = 0$  after  $S = 1$  and  $R = 0$ .

In a similar manner the reset input changes the output  $Q = 0$  and  $\bar{Q} = 1$ .

We assume  $S = 0$  and  $R = 1$ . This make  $Q = 0$ . This  $Q = 0$  is again given along with  $S = 0$  to make  $\bar{Q} = 1$ .

ie. when  $S = 0$  and  $R = 1$  make  $Q = 0$  and  $\bar{Q} = 1$ .

When the reset input returns to 0, the outputs do not change, because the output  $Q$  remains as '0' and  $\bar{Q}$  as '1'.

ie. when  $S = 0$  and  $R = 0$  make  $Q = 0$  and  $Q = 1$  after  $S = 0$  and  $R = 1$ .

This can be tabulated as

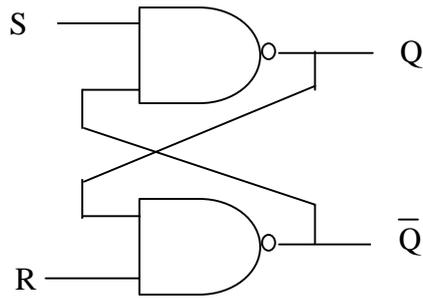
S	R	Q	$\bar{Q}$	
1	0	1	0	
0	0	1	0	(after $S = 1$ and $R = 0$ )
0	1	0	1	
0	0	0	1	(after $S = 0$ and $R = 1$ )

When '1' is applied to both  $S$  and  $R$ , the outputs  $Q$  and  $\bar{Q}$  become 0. These facts violate the output  $Q$  and  $\bar{Q}$  are the complements of each other. In normal operations this condition must be avoided.

Thus a basic flip-flop has two useful states. When  $Q = 1$  and  $\bar{Q} = 0$ , it is called as set state. When  $Q = 0$  and  $\bar{Q} = 1$ , it is called as reset state.

### Flip – Flop Circuit using NAND Gates

In a similar manner one can realize the basic flip-flop by cross coupling two NAND gates.



**Fig. 4.22 Flip Flop Circuit using NAND Gates**

The corresponding truth table is given as

S	R	Q	$\bar{Q}$	
1	0	0	1	
1	1	0	1	(after S =1 and R = 0)
0	1	1	0	
1	1	1	0	(after S =0 and R = 1)

The NAND basic flip-flop circuit operates with inputs normally at '1' unless the state of the flip-flop has to be changed. A momentary '0' to the input S gives  $Q = 1$  and  $\bar{Q} = 0$ . This makes the flip-flop to set state. After the input S returns to 1, a momentary '0' to the input R gives  $Q = 0$  and  $\bar{Q} = 1$ . This makes the flip-flop to reset state.

When both the inputs become 0, ie.,  $S = 0$  and  $R = 0$ , both the outputs become 1. This condition must be avoided in the normal operation.

There are several kinds of flip-flop circuits, with designators such as D, T, J-K, and R-S. Flip-flop circuits are interconnected to form the logic gates that comprise digital integrated circuits (ICs) such as memory chips and microprocessors.

## **4.6 Electronic Workbench**

### **4.6.1 Introduction**

Electronic workbench is a simulation tool for electronic circuits. It allows to design and analyze circuits without using actual instruments. The workbench's click-and-drag operation make editing a circuit fast and easy. It is a windows compatible tool and follows the normal conventions of windows. The completed circuit can be simulated within the workbench by the use of simulated test and measurement equipment which are wired into the circuit.

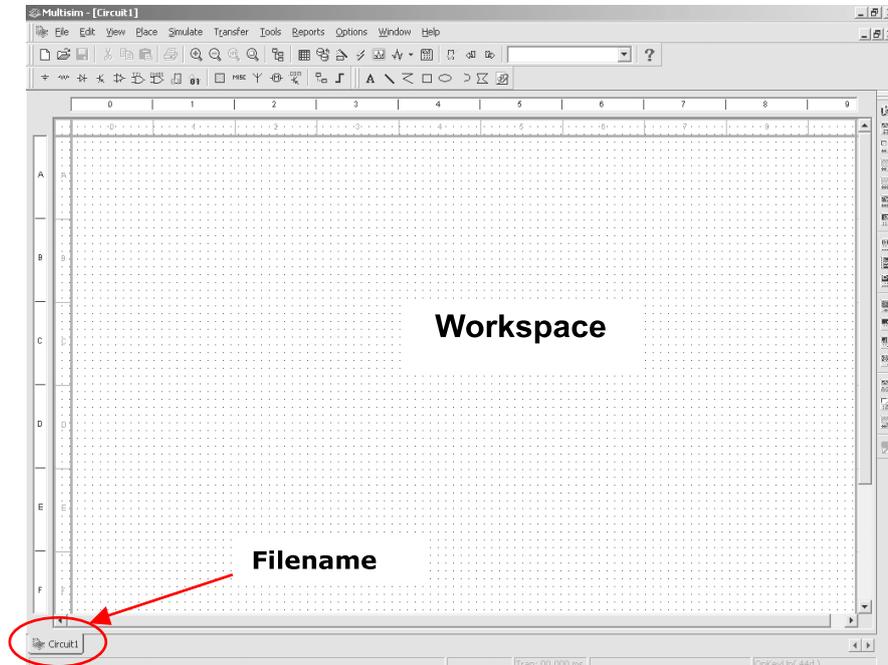
The circuit diagrams and the output of the simulated circuits can be printed or exported to other tools such as word processors for inclusion in other documents. The electronic workbench can be used for analogue, digital or mixed mode circuits. MultiSim is a electronic workbench which is used for design and analysis of circuits. It offers a single, easy-to-use graphical interface for the design needs. It also provides the advanced functionality you need to design from specifications.

### **4.6.2 Objectives and Expectations**

- § Even though the Multisim is designed with much functionality, we use the tool to help us get familiar with the basic digital features.
- § To learn how to build and test some simple digital logic gates using Multisim.
- § To know how to build and simulate simple combinational logic circuits using the logic converter.

### **4.6.3 Building a Simple Digital logic Gate**

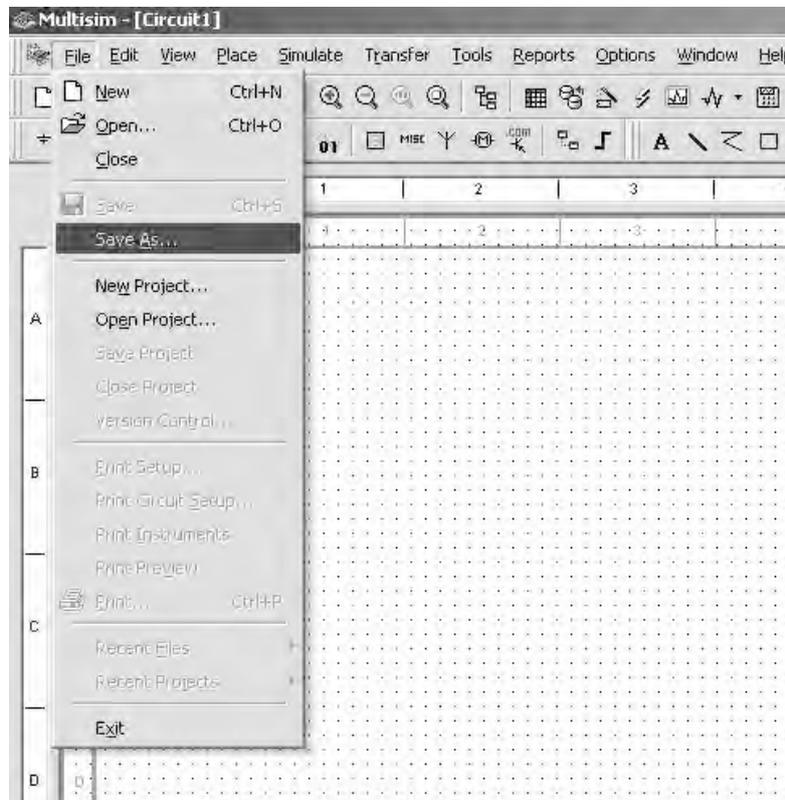
1. Start MultiSim from the **Programs** group in the **Start** menu. The main screen is as shown in the fig. 4.23



**Fig. 4.23 MultiSim Main Screen**

2. Assign a name to your file by following the steps given below.

Choose **File > Save As**, under the **File Name** box to save the new workspace as **circuit1**. Press **OK**. The filename will then appear in the lower left hand corner of the workspace as shown in the fig. 4.24.



**Fig. 4.24 File Menu**

3. Click on the **Place** menu, then click on the **Component**, which shows the list of components in the component library. A ghost image of the component appears on the circuit window showing exactly where the component is placed. One can use the appropriate database, group and component drop-down list to select the desired component.

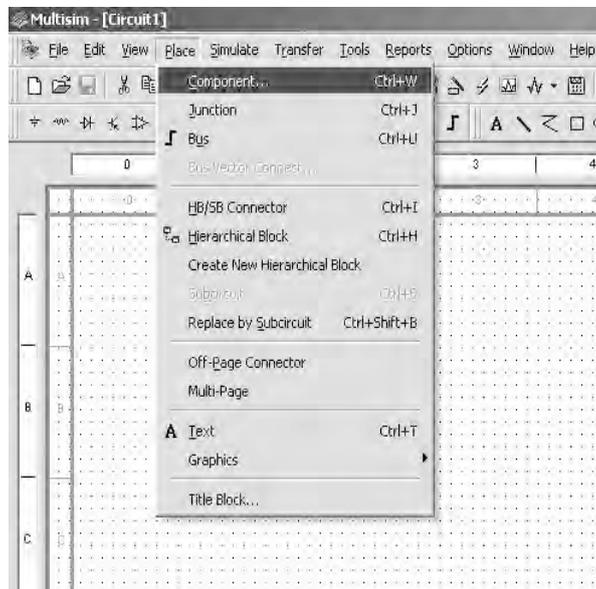


Fig. 4.25 Place Menu

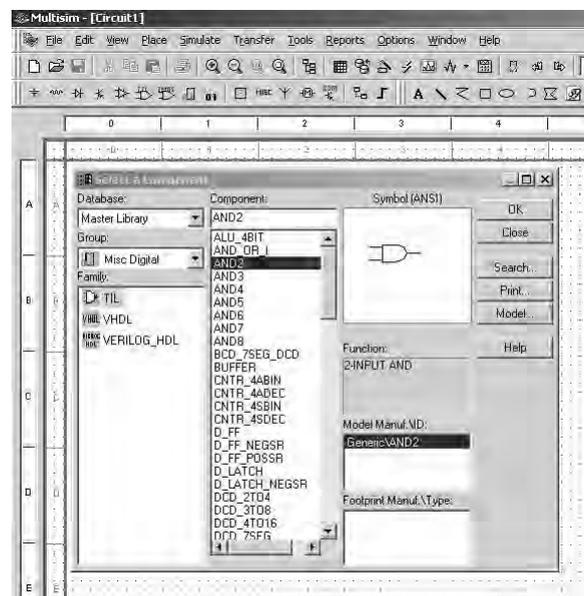
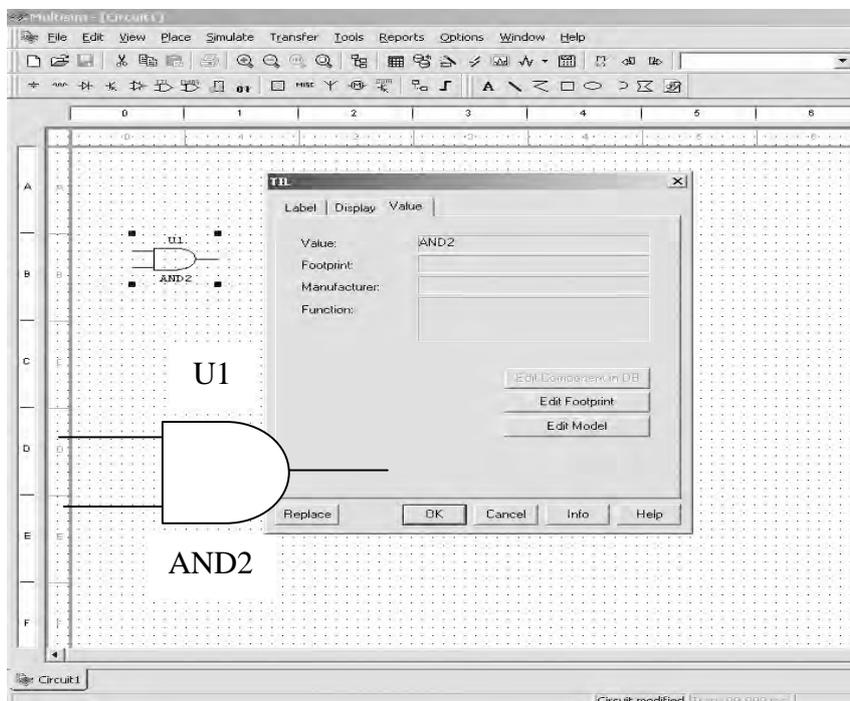


Fig. 4.26 Component Window

4. Click on any one of the logic gates available in the 'select a component' dialog box and drag it to place it in the workspace. Double click on the logic gate to change the properties of it and label the gate.

To get help, just click on the logic gate using the left mouse button and choose help.



**Fig. 4.27 Selecting a Component**

#### **4.6.4 Building a Simple Combinational Logic Circuit**

Here the general steps to build a circuit by dragging the components like gates, wires, etc. are discussed.

## Placing Components

Components like logic gates are placed in the workspace by

- i) Selecting **component** from **Place** menu.
- ii) Right clicking on the workspace and select **place components** or **Ctrl+W**.

## Selecting Components

Click on the component to highlight it .Once highlighted the options in the 'Circuit ' menu will apply to that component.

To select several components drag a box around them. All selected components will be highlighted.

## Copying Components

To copy a component select it and then choose '**Copy**' from the 'Edit' menu (or <cntrl> C). Then select paste (or <cntrl> V). Copies of the component will appear in the middle of the drawing area. They can then be moved to their required positions.

## Modifying components

- i) Select the components and then choose the available options from the 'CIRCUIT' menu.
- ii) Double click on the component. A window will be opened which gives the parameters for the component which can be customized. Change the values as required and then click on 'Accept'.

## Moving Components

To move components on the drawing select it and drag it to a new position and drop it. Any wires connected to it will be dragged with it.

## **Deleting Components**

Select components and then select 'Delete' or 'Cut' from the 'EDIT' menu or press delete. You will be asked to confirm the delete.

## **Building the circuit**

### **Placing interconnecting wires**

Click on the end of the leg of the component. Drag the wire directly to the leg of the next component and then drop it. The wire will then route itself neatly.

## **Printing the Circuit**

Select 'Print' from the File Menu and then select the item(s) to print from the dialogue box that is opened.

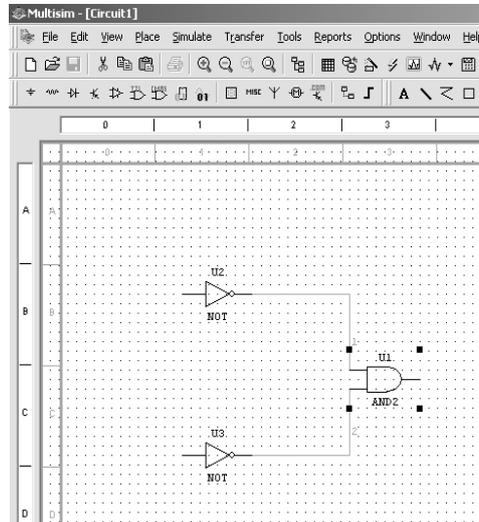
## **Saving the Circuit**

To save a new circuit or rename an old one, select 'Save As' from the File Menu and complete the details in the dialogue box that opens. To save an existing circuit select 'Save' from the 'File' Menu

## **Opening an Existing Circuit**

Select 'Open' from the File Menu and then enter the filename in the dialogue box that opens.

Consider a logical circuit by connecting two NOT gates with an AND gate. Place the NOT gate and AND gate on the workspace by selecting the respective components from the 'select a component' dialog box. Copy NOT gate. Drag the leg of the components and wire them as shown in fig. 4.28.



**Fig. 4.28 Construction of a Logic Circuit**

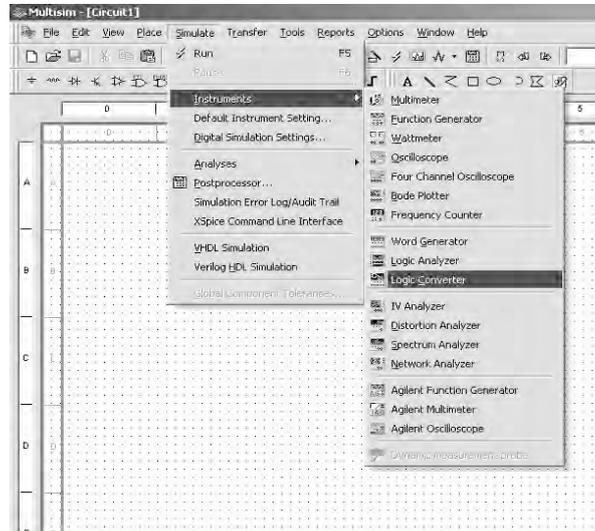
#### 4.6.5 The Logic Converter

MultiSim provides a tool called the **logic converter** from the **instrument bin**. With the **logic converter**, one can carry out several conversions of a logic circuit representation:

- |                       |                                 |
|-----------------------|---------------------------------|
| 1. Logic Circuit      | ➔ Truth Table                   |
| 2. Truth Table        | ➔ Boolean Expression            |
| 3. Truth Table        | ➔ Simplified Boolean Expression |
| 4. Boolean Expression | ➔ Truth Table                   |
| 5. Boolean Expression | ➔ Logic Circuit                 |
| 6. Boolean Expression | ➔ NAND-gate only logic circuit  |

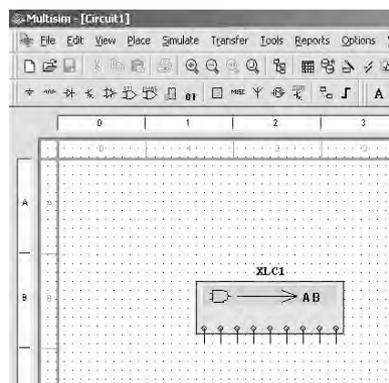
One can connect it to a digital circuit to derive the truth table or Boolean expression the circuit represents. One can also use it to produce a logic circuit from a truth table or boolean expression.

To open the **logic converter**, click on the **simulate** menu, **instruments** and then **logic converter** as shown in fig. 4.29.



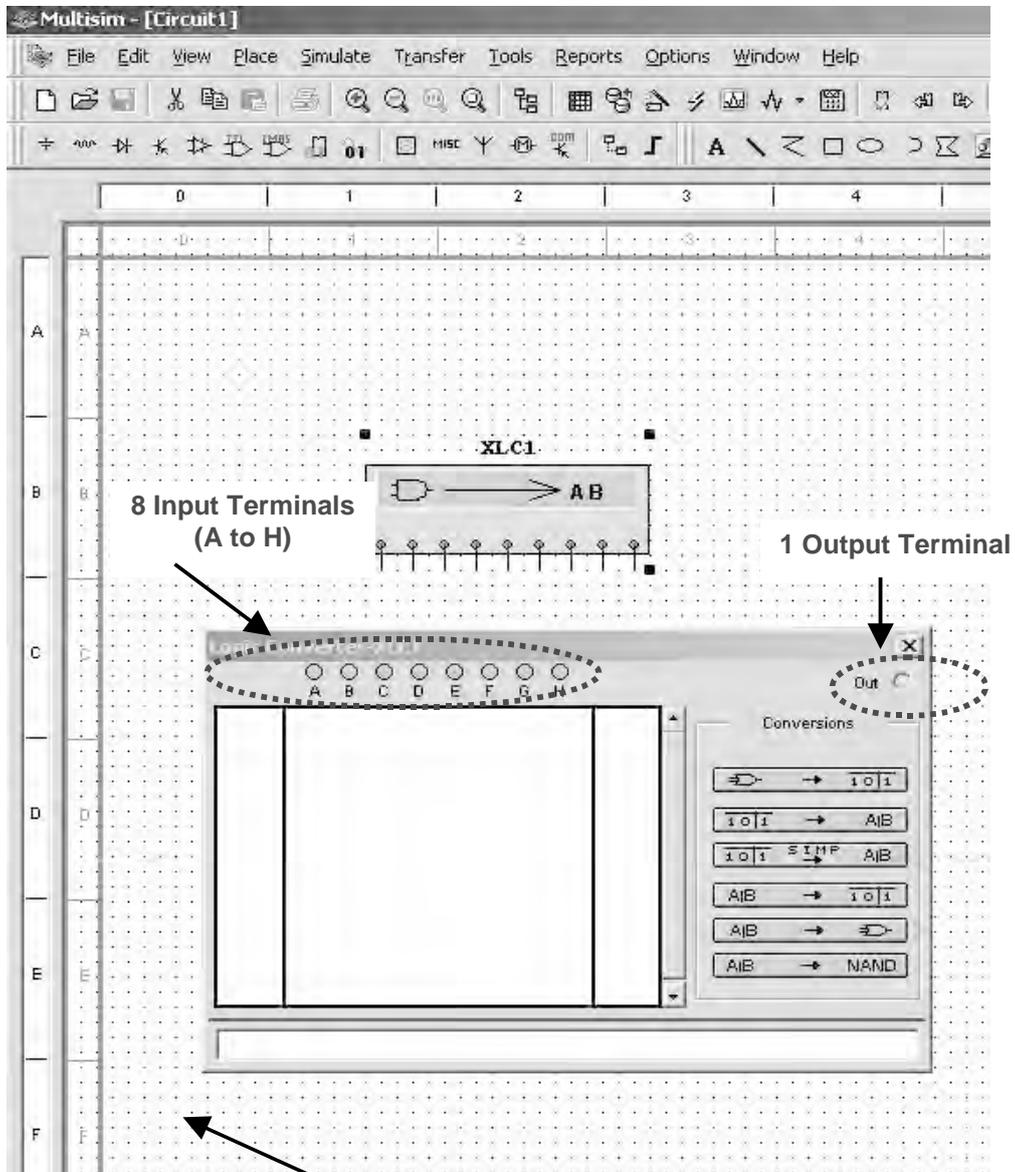
**Fig. 4.29 Logic Converter**

The fig. 4.30 shows the **logic converter** that you have placed in the workspace. The little circles in the bottom are the inputs and output terminals. There are 8 input terminals and 1 output terminal available for use. That means if you want to use the **logic converter** to carry out the conversions, an individual circuit is limited to 8 inputs and 1 output.



**Fig. 4.30 Logic Converter on the Workspace**

To use the **logic converter**, double-click on the **logic converter icon**. Your workspace should now look like the fig. 4.31.

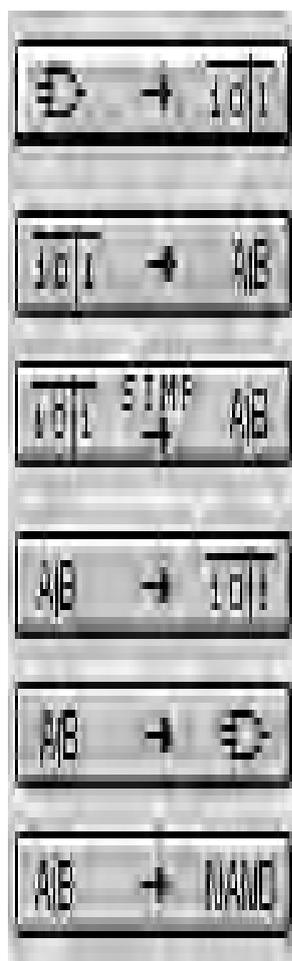


Boolean expression goes here

**Fig. 4.31 View of Logic Converter**

Notice that the 8 input terminals are labeled as A to H and the output terminal is labeled as 'Out'.

The converting options available in the logic converter are as follows:



**Descriptions:**

⇔ Convert Logic Circuit to Truth Table

⇔ Convert Truth Table to Boolean Expression

⇔ Convert Truth Table to Simplified Boolean Expression

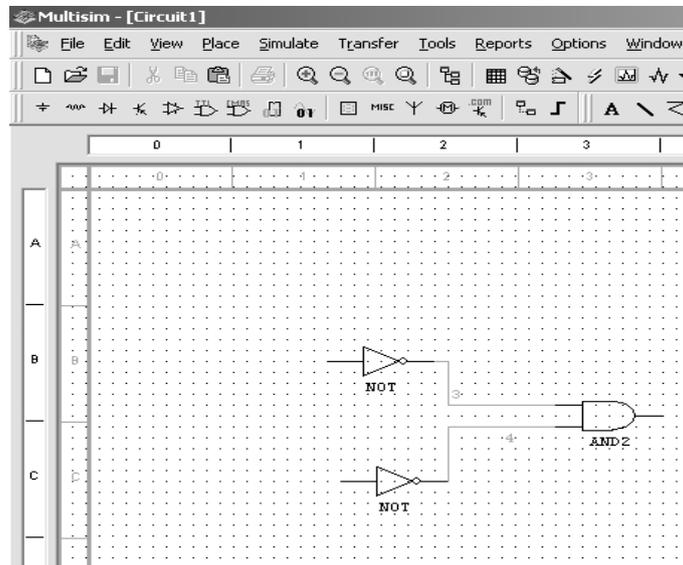
⇔ Convert Boolean Expression to Truth Table

⇔ Convert Boolean Expression to Logic Circuit

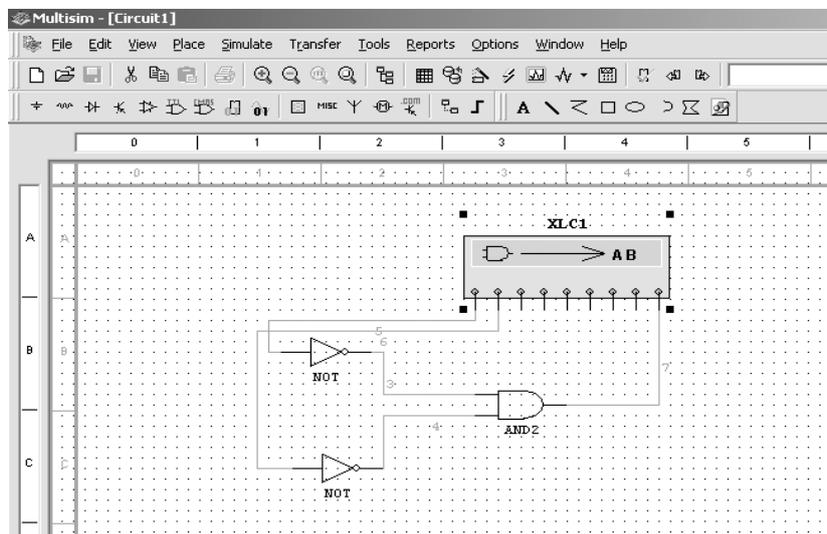
⇔ Convert Boolean Expression to NAND-gate only logic circuit

**4.6.6 Converting a logic circuit To a truth table**

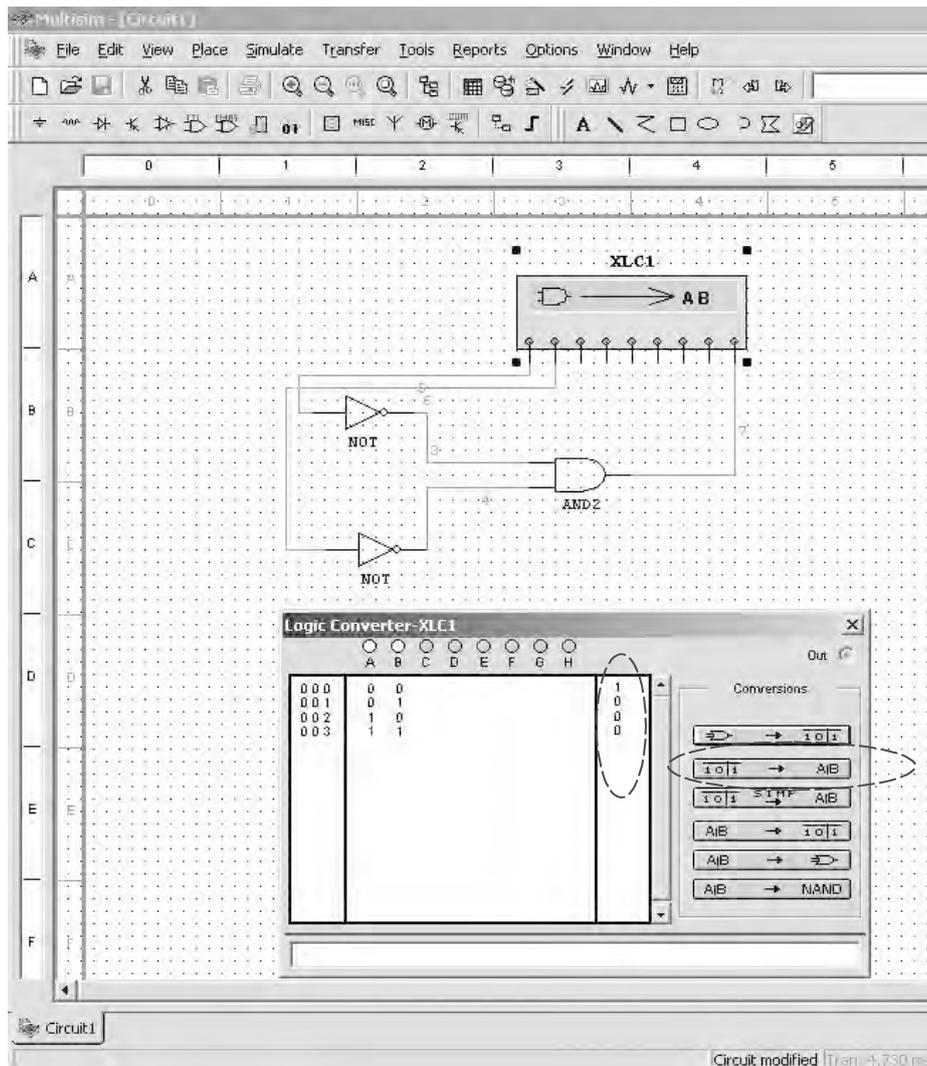
1. Construct the circuit by drawing the components (logic circuit).



2. Connect the inputs of your circuit to the input terminals on the logic converter and single output of the circuit to the output terminal on the logic converter as shown below.



3. Click the logic circuit  $\rightarrow$    $\rightarrow$   $\overline{101}$  You should get the truth table for the logic circuit in the logic converter.

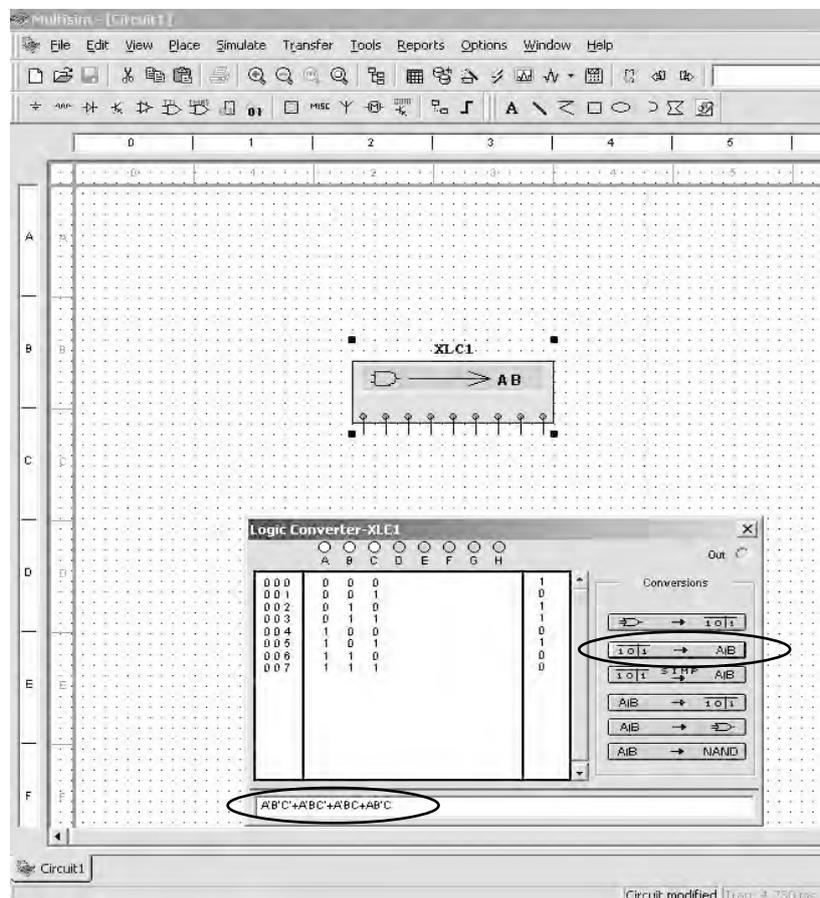


#### 4.6.7 Converting a Truth Table to a Boolean Expression

Once you have a truth table, the logic converter can transform it into a boolean function in the form of an algebraic expression.

**To create a Truth table :**

- Drag a logic converter to the workspace and open it.
- Click the number of inputs you want, from A to H at the top of the logic converter.
- The inputs are present in standard binary count format. In this case select A, B and C.
- The values in the output column are set to '?'. Click the output values once to change as '0' and twice to change as '1'.



Click the 'Truth Table to Boolean Expression' button.

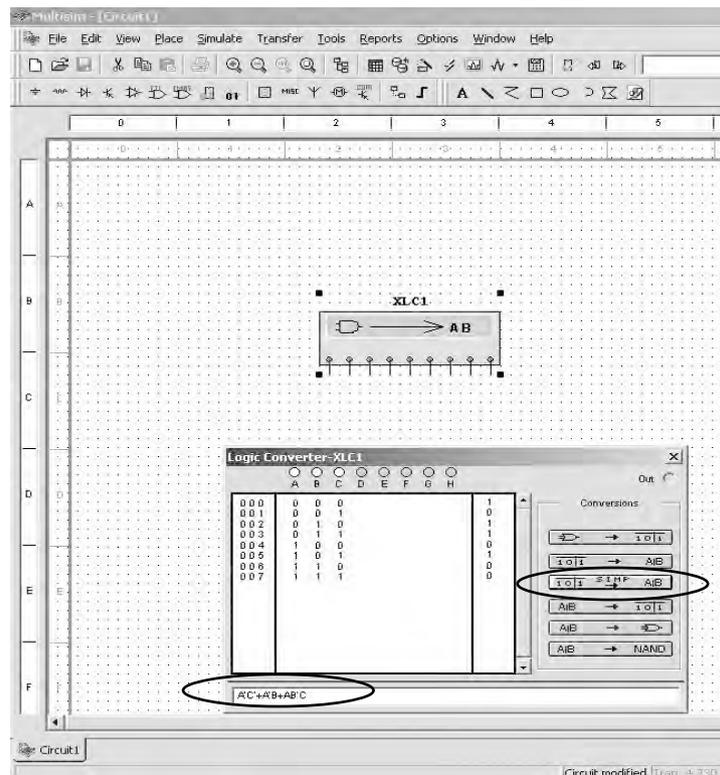
The boolean expression will be displayed at the bottom of the logic converter. In this case:-

$$A'B'C' + A'BC' + A'BC + AB'C$$

Note the primes, representing inversion. A' means NOT A, or  $\bar{A}$ .

#### 4.6.8 Converting a Truth Table to a Simplified Boolean Expression

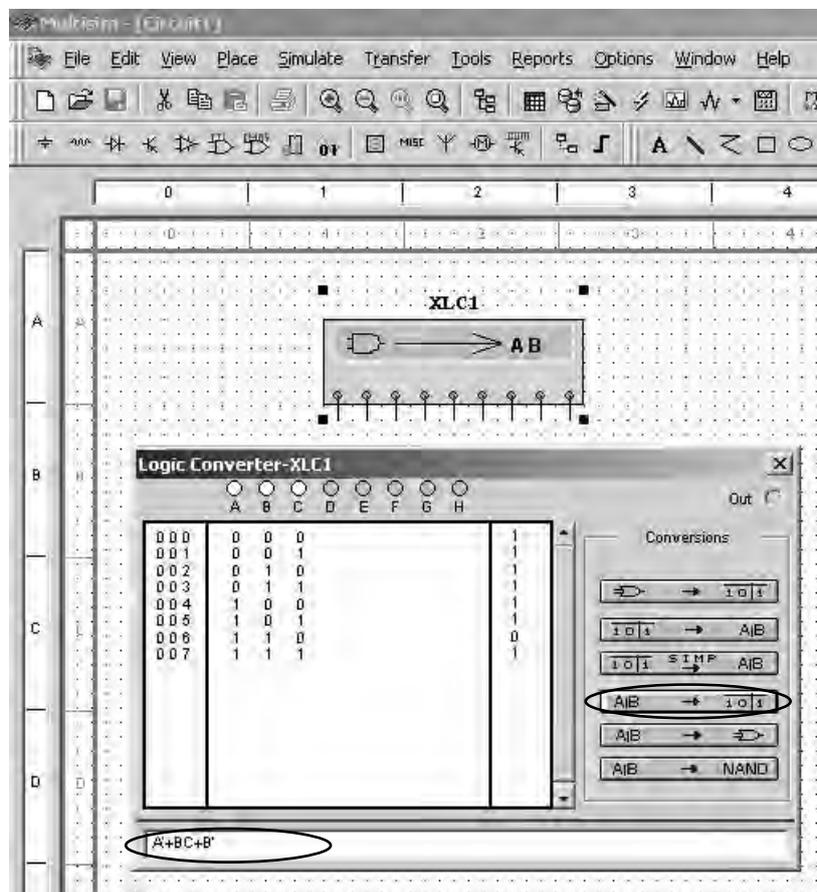
Some expressions can be recalculated in a simpler form. To try to simplify the expression, click the 'simplify' button using the previous truth table. In this case, the expression can be simplified to  $A'C' + A'B + AB'C$



#### 4.6.9 Converting a Boolean Expression to Truth Table

Once you have a boolean expression, the logic converter can transform it into a truth table. Click the 'Boolean Expression to Truth Table' button  after entering the Boolean expression  $A' + BC + B'$  in the logic converter.

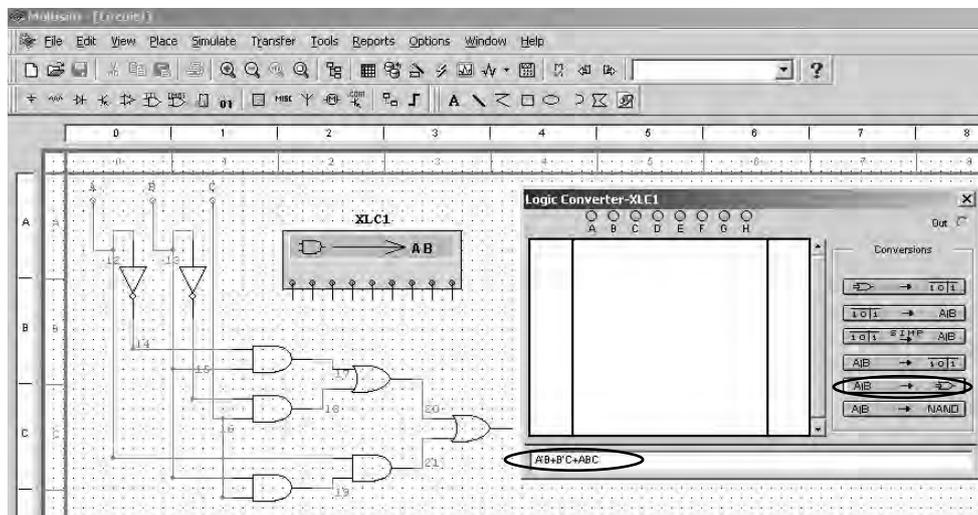
The truth table will be displayed in the logic converter as shown below.



#### 4.6.10 Converting a Boolean Expression to a Circuit

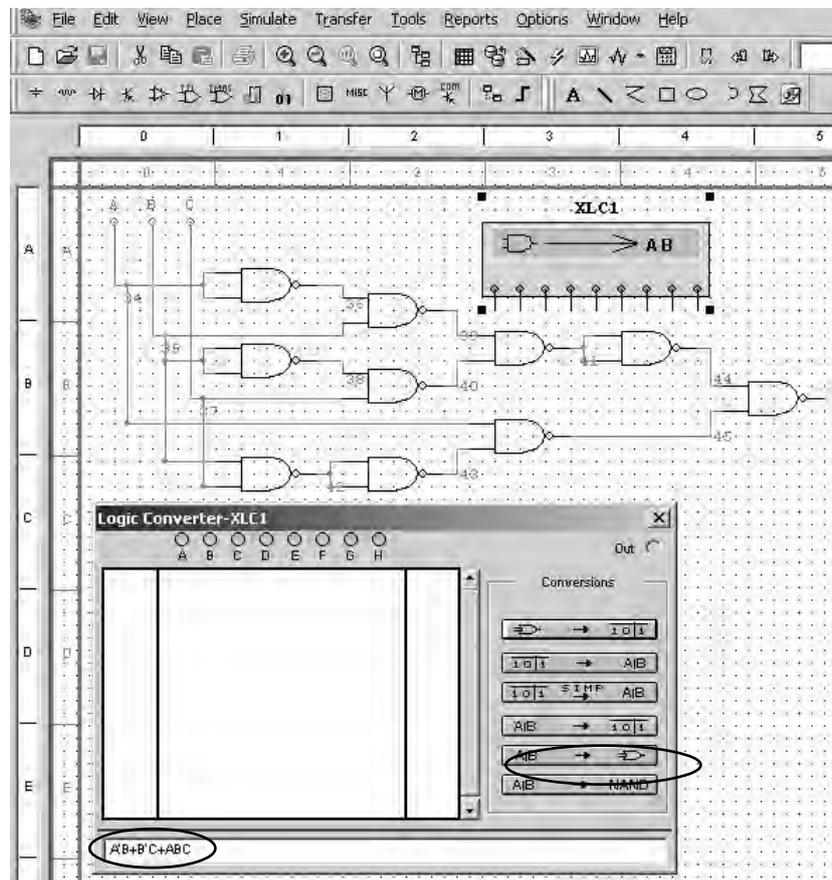
To do this, enter the boolean expression,  $A'B+B'C+ABC$  and click the 'Boolean to Circuit' button 

The resulting circuit will appear in the workspace.



#### 4.6.11 Converting Boolean Expression to NAND-gate only logic circuit

To realize the logic circuit using NAND-gates for the same boolean expression  $A'B+B'C+ABC$ , click . The resulting circuit will appear on the workspace.

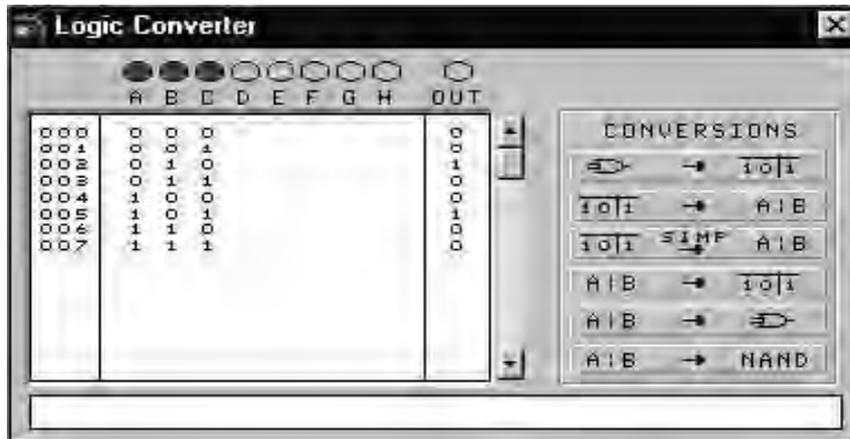


#### 4.6.12 Creating a Circuit from a Truth Table

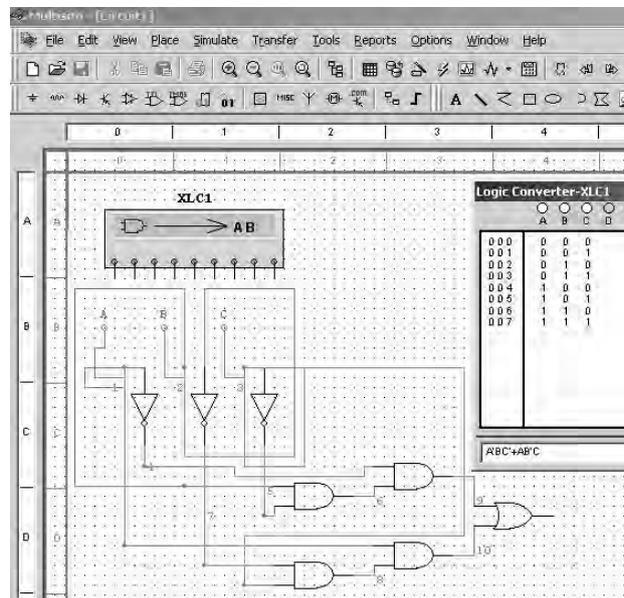
This is the most useful conversion for a circuit designer. Normally, you will have translated the client's specification into a truth table, and have then to produce a logic gate circuit to do the job. This requires two conversions using the logic converter. We will practice using a different problem.

- Create a truth table.
- Click the 'simplify' button to convert the truth table to the simplest Boolean expression ( $A'BC' + AB'C$ )

- Click the 'Boolean to Circuit' button



The resulting circuit will appear, selected, on the workspace. If you want to move it, point to one component and drag the circuit.



The following table summarizes all possible scenarios that one can encounter:

From	To	Options
Truth Table	Boolean Expressions	$\overline{10 1} \rightarrow A B$
	Simplified Boolean Expressions	$\overline{10 1} \xrightarrow{\text{SIMP}} A B$
	Logic Circuit	$\overline{10 1} \rightarrow A B$ $A B \rightarrow \Rightarrow$
	NAND-gate only logic circuit	$\overline{10 1} \rightarrow A B$ $A B \rightarrow \text{NAND}$
Boolean Expressions	Truth Table	$A B \rightarrow \overline{10 1}$
	Logic Circuit	$A B \rightarrow \Rightarrow$
	NAND-gate only logic circuit	$A B \rightarrow \text{NAND}$
	Simplified Boolean Expressions	$A B \rightarrow \overline{10 1}$ $\overline{10 1} \xrightarrow{\text{SIMP}} A B$
Logic Circuit	Truth Table	$\Rightarrow \rightarrow \overline{10 1}$
	Boolean Expressions	$\Rightarrow \rightarrow \overline{10 1}$ $\overline{10 1} \rightarrow A B$
	Simplified Boolean Expressions	$\Rightarrow \rightarrow \overline{10 1}$ $\overline{10 1} \xrightarrow{\text{SIMP}} A B$

**T able 4.13 : Quick Guide to convert a Digital Circuit using the Logic Converter**

## Summary

- ☛ A logic gate is an elementary building block of a digital circuit.
- ☛ There are three fundamental logic gates namely, AND, OR and NOT.
- ☛ we have other logic gates like NAND, NOR, XOR and XNOR.
- ☛ NAND and NOR gates are called the universal gates.
- ☛ The circuit that performs addition within the Arithmetic and Logic Unit of the CPU are called adders.
- ☛ A unit that adds two binary digits is called a half adder.
- ☛ one that adds together three binary digits is called a full adder.
- ☛ A flip flop is a circuit which is capable of remembering the value which is given as input.
- ☛ Electronic workbench is a simulation tool for electronic circuits.
- ☛ MultiSim is a electronic workbench which is used for design and analysis of circuits.

## Exercises

### I. Fill in the blanks

1. In AND gate the output is \_\_\_\_\_ when both the inputs are 'true'.
2. In OR gate the output is \_\_\_\_\_ if both the inputs are 'false'.
3. A \_\_\_\_\_ is an elementary building block of a digital circuit.
4. The NAND gate operates as an AND gate followed by a \_\_\_\_\_ gate.
5. The \_\_\_\_\_ gate circuit is an OR gate followed by an inverter.
6. \_\_\_\_\_ and \_\_\_\_\_ gates are called universal gates.
7. \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ gates are called the fundamental gates.
8. A unit that adds two binary digits is called a \_\_\_\_\_.
9. A full adder can be constructed from two \_\_\_\_\_ and a \_\_\_\_\_ gate.
10. A simple flip flop has \_\_\_\_\_ stable states.

### II. State whether the following are True or False

1. Logic gate has two or more outputs.
2. Logic circuits have two or more outputs.
3. AND is an universal gate.
4. XOR is a fundamental gate.
5. NAND gate can be used to realize OR gate.
6. NOR gate can be used to realize OR gate.
7. Full adder can be realized using two half adders and an OR gate.
8. XOR gate is used to build half adder.
9. XOR gate circuit is an OR gate following by an inventor
10. Flip-flop is used as a storage.

### III. Answer in one or two lines:

1. What is a logic gate?
2. List the fundamental logical gates?
3. Why NAND and NOR gates are called as universal gates?
4. How AND gate can be realized using NOR gate?

5. How OR gate can be realized using NAND gate?
6. Give the truth table of XOR gates for two inputs.
7. What is a half adder?
8. What is a full adder?
9. What is a combinational circuit?
10. What is a sequential circuit?

#### IV. Answer the following

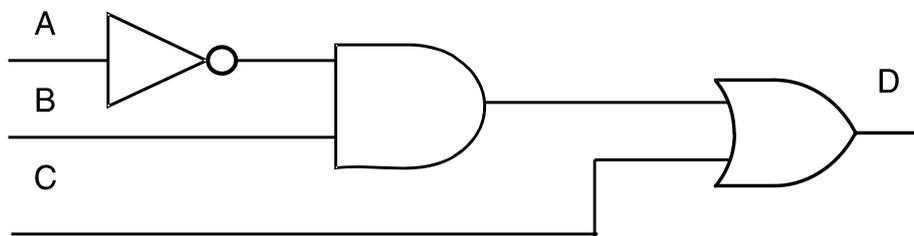
1. Determine the truth table for the following Boolean functions

$$E = \bar{A} + (B \cdot C) + \bar{D}$$

2. Convert the following truth tables to a Boolean equations.

Input			Output
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

3. Convert the following logic circuit to a boolean equation.



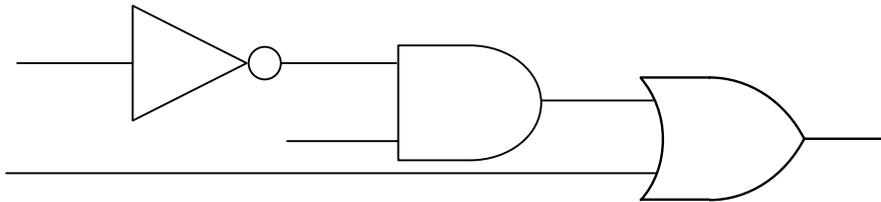
- Realize the boolean function to a logic circuit

$$E = \bar{A} B + \bar{B} C + A B C$$

- Explain the steps involved in designing a logic circuit.
- What are the different types of logic gates? Explain with the help of truth tables and give an example for each gate.

### Project

- Using the logic converter, test the basic logic gates by constructing their truth table.
- Using the logic converter, build and test the following logic circuits and record your simulation results in a truth table.



- Build a logic circuit for the boolean function  $A'C'+A'C+AC'+AC$  using the logic converter.
- Build a logic circuit for the following truth table and bring out its equivalent Boolean expression using logic converter. Also simplify the Boolean expression.

Input			Output
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

# Chapter 5

## OPERATING SYSTEM

### 5.1 Introduction

When a computer is devoid of any software it is just like dead bodies. The computer software and hardware have been intrinsically linked with each other. One of them cannot do anything useful by itself without help from the other. There are two types of software, one is the System Software and the other is the Application Software. System Software looks after the functions of the computer. This is just like involuntary actions controlling involuntary muscles such as digestive system of the animal. System software makes efficient use of the computing resources and normally provides a uniform base for different applications. It is there to operate, control and extend the processing capabilities of computers. Application software helps the user to do his/her work. This is similar to the voluntary actions controlling voluntary muscles like hand etc. Moving a hand is a voluntary action. The Operating System comes under the System Software category. The actual work is undertaken only by the hardware. In order to do useful work on a computer, one has to access the hardware, but one can access the hardware directly, only in the first generation computers. In the subsequent generations of computers direct access is denied. The architecture of the computers at the machine level is primitive and very hard to program. What is the reason for the denial of the access? If the access is not denied, unless the user is hard working, one of the two alternatives can happen.

- 1) The user may be forced to conclude that computer is not for him/her.
- 2) The user may damage the computer hardware.

This leads us to a natural question. Who will access the computer hardware directly if the user is denied such permission? The answer is the Operating System. The Operating system provides so many facilities with which a user comfortably uses their computers. This resembles the life of modern man, who cannot tolerate power cut even for five minutes.

Apart from being complicated, trying to deal with I/O(Input/Output) operations can prove truly frustrating. The average user wants to have a simple high-level abstraction to deal with. If you see an object, a lot of impulses are triggered on inside your brain. At the end you are shown an image of the object. If an average person is asked to explain the activities that happen inside the brain, he/she will not dare to even open his eyes. The brain provides a convenient highly sophisticated abstraction. It merely provides the image of an object without letting people know about the activities that happen inside the brain. In this case eye is an interface between the object and the part of the brain that processes visual data. In a similar fashion the Operating System hides from a person know the complexity of the hardware and allows the user to use the name of the files for reading and writing. The Operating System adds extended capabilities to a machine with which it is easier to program than the underlying hardware. The Operating System manages the resource. Modern computers are highly complex machines. They get more complex day by day. So it is very difficult to manage such a complex system but the Operating System manages the complex system in an efficient way. It provides special routines called device drivers to support the specific behaviour of individual device. In this view the primary task of the Operating System is to keep track of who is using which resource, to grant resource requests, to account for usage and to mediate conflicting requests from different programs and users. The Operating System is to provide an orderly and controlled allocation of resources among the various programs competing for them. The Operating System is the intermediary between the user and computer hardware. When the Operating System was first introduced, the primary goal of the Operating System was mainly to optimize resources. The secondary goal was to make the computer environment, user-friendly. Now providing the user-friendly environment is the main aim of the operating system.

The Operating System acts as the manager of resources such as CPU time, memory space, file storage, I/O devices. Since there may be many conflicting requests, Operating System allocates resources in an optimal manner. That is, Operating System allocates resources in such a manner so as to achieve the maximum best possible result.

The Operating System also provides the means for the proper use of hardware, software and data in the operation of the computer system. The Operating System is like a supervisor in a company providing an excellent environment in which the other people can perform useful work.

Operating System assumes yet another responsibility, that of serving as a control program. A control program controls the execution of user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control I/O devices.

It is hard to define Operating System. There are several definitions for Operating System. One of the definitions is that Operating System is one program running at all times on the computer. The another, somewhat more widely accepted definition is that an Operating System is an interface between the user and hardware.

The Operating System's goals are to:

- 1) execute user programs in a user-friendly atmosphere.
- 2) make the computer system convenient to use.
- 3) optimize computer hardware.

Any Operating system should be easy to use. Now-a-days people are hard pressed for time, so they cannot undergo any training for making use of the Operating System. The idea of using facilities available in the Operating System should be intuitive. The Operating System should allow developing application programs easier. Otherwise people cannot concentrate on the application development; instead, they have to spend lot of time in concentrating on the peculiarities of the Operating System. The Operating System should be portable. That is, the Operating System should run in almost all hardware platforms.

If there is a new version of the Operating System, it should not confuse the people who used the earlier version and also it should run software that ran successfully in earlier versions. The Operating System should provide **data security**; it should not allow one user to write on the

file/files of the other user and thus spoiling the contents of the owner of the file/files.

The Operating System should provide data confidentiality. That is, the Operating System should not allow unauthorised people to access the data of the other people. If this is possible then the scientific and technological institutions and banks will have a nightmarish existence. The vendor who provided the Operating System should undertake the service facility also. The vendor should be accessed easily. Otherwise that Operating System will attain notoriety.

The Operating System should work in a network as well as distributed environment. The Operating System should make system administration more efficient.

The Operating system should provide the help facility. There are people who do not like to get help from the other people; this will prick their self-esteem. Instead they may try to get help from the system. The help facility should mainly concentrate on people like them. There should be different levels of help to satisfy the needs of different levels of users.

According to John Von Neumann architecture application, program and data should reside in main memory. In those days programs dealt mainly with scientific problems. For solving these types of problems, software libraries are created. These libraries complicated the normal user of that time. Therefore, the computer operator job is created. But this prevented the programmer to remove the errors (debug) immediately. Programmers had to wait for nearly six hours for debugging their programs.

### **A brief History of the Operating System.**

In the beginning, programs were run one at a time. In order to use CPU more efficiently, jobs of similar nature were grouped and made to run in batches. But after program execution, the computer operator manually restarted the next program. To avoid the delay due to manual operation, Automatic Job Sequencing mechanism was introduced.

This is called Resident Monitor and this may be the first elementary Operating System. If scientific applications had been used in a computer, the CPU (**C**entral **P**rocessing **U**nit) was busy with the program, but if business problems had been handled, I/O system was busy and the CPU was kept idle.

In order to make the CPU busy, the I/O operations were done in an inexpensive computer, the contents of card reader was transferred in a magnetic tape and which in turn was placed in the computer that performed the business operation. In those days data were stored in cards, called punched cards. Another attempt was made to keep the CPU busy for most of the time. A buffer (Refer chapter 6) was (and still is) allowed to store Input, when the CPU needed the data, buffer sent the data immediately. The next input might be ready in the buffer before the CPU processed the earlier data. When the processing was completed, the CPU sent the output to the buffer. When the data were ready, an interrupt mechanism interrupted the CPU. The CPU having taken the necessary actions and resumed its original work.

At the same time, Direct Memory Access (DMA) mechanism was also created, which allowed transferring data to and from memory without the intervention of the CPU. Spooling (is a way of dealing with dedicated I/O devices in the multiprogramming system.) allowed (and still allows) reading a set of jobs in disk system from the card reader. When printing work had to be undertaken, the print image was copied into the disk system and when conditions were favourable the print image was sent to the printer.

Spooling is superior to the buffer, because in spooling I/O operations can be overlapped with the working of other jobs but that is not possible with the buffer. While executing one job, the OS, reads next job from card reader into a storage area on the disk and outputs printout of previous job from disk to the printer. Spooling allowed the CPU to choose a particular job for execution leading to the concept called the **Job Scheduling**. The job scheduling led to the concept known as the **Multiprogramming**. In multiprogramming, memory is divided into many partitions. Multiprogramming allows many programmers to load their programs in the different partitions. Each programmer is made to believe his/her program is the only program running. Multiprogramming was followed by **Time-sharing** concept. Here the CPU allocated a fixed time for each program. In the next cycle, the program that had been considered earlier was taken once again. This process continued until all the programs were executed.

## 5.2 Major Features of the Operating System

### 5.2.1 Types

As per the number of users, there are two types of the Operating Systems. They are

1. Single user Operating System.
2. Multi-user Operating System.

**Single user Operating System** : At a time, only one user can operate the system. MS Disk Operating System is an example of single user Operating System.

**Multi-user Operating System** : More than one user can operate the same system simultaneously. The multi-user Operating System is based on the concept of time-sharing. Unix is an example of multi-user Operating System.

### 5.2.2 Input/Output

Application software does not allocate or de-allocate the storage area on the disk for different files belonging to various users. If the application software is allowed to allocate or de-allocate, two or more users may try to write on the same sector of disk, resulting in confusion. Even a single user may try to write in some sector, which may contain valuable information. In order to avoid such an awkward situation, only the Operating System is empowered to make such an allocation or de-allocation. This arrangement safeguards the loss of data. Such safeguarding of data is called **Data Security**.

From the above discussion, one may come to the conclusion that the Operating System alone should be empowered to instruct the hardware to write data from memory onto a Pre-specified location on the disk. In fact Input/Output operation code of the Operating System constitute a sizeable code of the Operating System.

The application program is not allowed to read data from the disk. Otherwise any user can access any type of sensitive information. There may not be any secrecy.

For example, banks will have precarious existence. An application program can do all the operations with the exception of input/output operations. When the application program is translated into the machine code, the request for reading or writing will not be translated into the machine code, instead a system call is given. (A set of extended instructions providing an interface between the Operating System and the user programs, is called a **System Call**.) The Operating System will then generate suitable input/output command to the hardware to replace this system call. You cannot fool the system by writing the I/O code in machine language. User code will not be entertained for input/output at any circumstance. This arrangement not only helps in protecting the data integrity, but also, it saves the user from writing a lot of code to execute I/O operations. Thus it saves from reinventing the wheel. It is enough, if the programmer concentrated in logical behaviour of the program. The Operating System does the spadework for the arrival of application program and the Operating System which is in the back ground, when needed comes into forefront and does its work gracefully after that it relegates itself to the background.

### 5.2.3 Printer

Ideally we can expect computers to create truly paperless society. Appropriate networking and Infrastructure must be provided for this. As of today computers consume a lot of papers. Usage of printers is rampant. How does the Operating System help for printing? The Operating System makes the programmer's life much easier as far as the printing work is concerned. Even a small document takes a few minutes to complete the printing work. Now for printing a document, the Operating System first makes a temporary copy of the file and then hands over the control back to the application. Spooling is a way of dealing with dedicated I/O devices in a multiprogramming system. To print a file, a process first generates the entire file to be printed and puts in the spooling directory. (Directory is a container for storing files on other sub-directories).

Then the special process having permission to use printer's special file is allowed to print the files in the directory. If two users print two documents, in the absence of overlapping of the two documents.

This is similar to overlapping of the signals from two radio stations. This unwanted behaviour is completely eliminated by spooling. Each document's print image will be written on to the disk at two different locations of the spool file. While printing, the printer is not allowed to print the original document; instead it is allowed to print the contents of spooler program.

Multiprocessor systems have more than one CPU in close communication with the others. In a tightly coupled system the processors share memory and a clock; communication usually takes place through the shared memory.

### **5.3 Most Desirable characters of the Operating System**

#### **5.3.1 User Interface**

The Operating System should concentrate on the user interface. The only way that you can see this world is through your eyes. Similarly the only way that you can interact with a computer is through the user interface. People may be attracted to the computer in the beginning. If the interface is not user-friendly, only persistent people may continue with their work with the computer. The other people slowly move away from the computer with the intention of not returning to the computer forever. One can judge, from the immense popularity of the GUI (Graphical User Interface) based interface, the importance of well designed well thought interface. The GUI is window based. The vivid colours attract children. Novices are attracted by the help pop up messages. Icons give iterative usage of the particular application.

Now Linux is also available as windows based Operating System. The user interface of the Operating System should be appealing to the senses. The human brain is good in pattern recognition. Human brain learns through association. All these factors should be taken into account, when user interface is improved. In addition to the above, the following points should also be considered when User Interface is designed.

- 1) Interface should be designed in such a manner as to master the interface. As already stated, people are hard pressed for time.
- 2) The speed of response should play a vital role in designing the user interface. The speed of response is nothing but the time taken to execute a particular task.
- 3) The user interface should reduce the number of errors committed by the user. With little practice, the user should be in a position to avoid errors.
- 4) The user interface should be pleasing to the senses. Vivid colours, enchanting music may achieve this.
- 5) The user interface should enable the people to retain this expertise for a longer time.
- 6) The ultimate aim of any product is to satisfy the customer. The user interface should also satisfy the customer.

Interface developers should also take the following considerations into account. Interfaces mainly should satisfy the end users. The other users such as programmers may work even in an unfriendly environment. The interface should not heavily burden the memory of users. Menus, minimal typing work will be an added advantage of the Operating System.

### **5.3.2 Memory Management**

The Operating System should provide memory management techniques also. Any error in the user program should not be allowed to spoil the entire memory. So the Operating System divides the main memory into user memory and reserved memory. If some errors creep into the user program then only user memory may be affected however the reserved memory is always in an unaffected condition.

User memory is divided into many partitions to accommodate various jobs. Therefore the number of jobs accommodated cannot exceed the number of partitions. Naturally the size of the user program should be less than that of the available main memory. This is like cutting the feet to the size of the shoe (if the size of the shoe is inadequate).the Operating System provides virtual (imaginary) memory to include the entire program.

Operating System should manage the devices also. It is not uncommon for several processes to run simultaneously. In order to achieve successful operation, the processes should effectively communicate with each other. This interprocess communication is made possible by the Operating System.

### **5.3.3 Process management**

Process management undertakes the allocation of processors to one program. The Operating System controls the jobs submitted to the system (CPU). Several algorithms are used to allocate the job to the processor. Algorithm is a step-by-step method to solve a given problem.

- 1.FIFO.
- 2.SJF
- 3.Round Robin.
- 4.Based on Priority.

#### **FIFO (First In First Out)**

This algorithm is based on queuing. Suppose you are standing in a queue to get your notebook corrected from your teacher. The student who stands first in the queue gets his/her notebook corrected first and leaves the queue. Then the next student in the queue gets it corrected and so on. This is the basic methodology of the FIFO algorithm.

Now, let us deal with this FIFO a little more technically. The process (A process is basically a program in execution) that enters the queue first is executed first by the CPU, then the next and then the next and so on. The processes are executed in the order in which they enter the queue.

### **SJF (Shortest Job First:)**

This algorithm is based on the size of the job.

Take two jobs A and B.

A = 5 kilo bytes

B = 8 kilo bytes

Kilo literally means 1000 but here kilo means 1024. A byte consists of eight bits. A bit can store either TRUE (1) or FALSE (0).

First the job A will be assigned processor time after which B gets its turn.

### **Round Robin**

Jobs are assigned processor time in a circular method. For example take three jobs A, B, C. First the job A is assigned to CPU then job B and after B job C and then again A,B and C and so on.

### **Based On Priority**

In this method each job is assigned a Priority. The higher Priority job is awarded favorable treatment. Take two jobs A and B. Let the priority of A be 5 and priority B be 7.

Job B is assigned to the processor before job A.

The allocation of processors by process management is also known as the CPU Scheduling. The objectives of the CPU Scheduling should be to maximise

- (1) The CPU utilisation.
- (2) The number of jobs done in a unit time (throughput) and to minimise the time taken.

before the execution of the job and to run the job.

Let us consider e-mail, which allows to :-

(1) represent the information electrically

(2) carry information from source to destination

(3) manage the flow of such information. The telecommunication industry provides all the above facilities. The information that may be sent by network may be voice, data, video, fax etc. Web camera unites the parents and their children who are away from each other. Now the size of LAN has grown. You can see a LAN with more than 1000 computers connected to it. Through telecommunication links, LAN can access remote computers. The size and complexity of the network grow day by day. It is not a mean achievement to manage them. The Operating System shoulders the burden (responsibility) of managing the nets.

#### **5.3.4 Security Management**

The biggest challenge to the computer industry is to safeguarding one's data from unauthorized people. The Operating System provides three levels of securities to the user. They are

(1) File access level

(2) System level

(3) Network level

In order to access the files created by other people, you should have the requisite permission. Permissions can either be granted by the creator of the file or by the administrator of the system.

System level security is offered by the password in a multi-user environment. Both windows XP professional and Linux offer the password facility.

Network security is an elusive one. People from all over the world try to provide such a security.

All the above levels of security are provided only by the Operating System.

### **5.3.5 Fault tolerance**

The Operating Systems should be robust. When there is a fault, the Operating System should not crash, instead the Operating System have fault tolerance capabilities.

### **5.3.6 Application Base**

Operating System should provide a solid basis for running many popular applications.

### **5.3.7 Distributed Operating System**

If you want to make use of the Network, you must know the machine address and the variety of services provided by that machine. But Distributed Operating System ensures that the entire network behaves as a single computer. Getting access to the remote resources is similar to access to local resources. The user's job is executed in an idle machine and the result is communicated to the user machine. The user is under the illusion that everything is done only in his/her computer. In a distributed Operating System a user is not aware of multiplicity of machines. The future of the Operating System may be Distributed Operating System since all the computers become a part of one or other network. But the virus attacks discourage people to get connected to the net.

From the above one can appreciate the importance of the Operating System.

## Summary

Software is of two types. They are

1. System Software and
2. Application Software.

Operating System is a system Software that comes under System Software.

Operating System is an intermediary between the user and the hardware.

There are

1. Single user Operating System and
2. Multi-user operating system.

The I/O operations are tedious and they are always maintained by the Operating system. When Application programs want to access the I/O capabilities, they merely substitute with the system call.

Direct Memory Access (DMA) mechanism allows transferring data to and from memory without the intervention of the CPU.

Spooling is superior to buffer. Spooling takes care of the printing work with the printer.

Multiprogramming gives the illusion that many programs run simultaneously.

The desirable characters of the Operating System are

- 1 User Interface
- 2 Memory management
- 3 Process management
- 4 File management
- 5 Networking Capabilities management
- 6 Security Management
- 7 Fault tolerance
- 8 Application Base
- 9 Distributed Operating System.

### **Exercises**

#### **I. Fill in the blanks**

1. The \_\_\_\_\_, \_\_\_\_\_ can access the hardware directly.
2. Operating System is the \_\_\_\_\_ between the user and computer hardware.
3. Operating System comes under \_\_\_\_\_ software.
4. The \_\_\_\_\_, \_\_\_\_\_ is only means by which a user interacts with the computer.

#### **II. State whether the following are True or False**

1. The primary goal of Operating System is to mainly optimize the memory management.
2. There are many types of the Operating Systems.
3. The higher priority jobs get executed faster than those with lower priorities.
4. In Distributed Operating System, the user should know the address of the accessed machine.

### **III. Answer the following**

1. Who will access the computer hardware directly?
2. Define an OS.
3. Explain the different roles taken by the OS.
4. Explain the main functions of the operating system.
5. Explain the process and memory managements.
6. Explain the input / output managed by operating system.
7. Write note on User Interface.
8. List out advantages of the Distributed Operating System over the Network Operating System.
9. Name some of the required features of Operating System.

## CHAPTER 6

### COMPUTER COMMUNICATIONS

#### 6.1 Introduction

Communication is the desire of man. When human voice became inadequate, ancient civilizations devised drum codes and smoke signals to send information to far off distances. These primitive methods have given way to sending messages through electronic pulses. A stand-alone computer communicates very efficiently by connecting it with other computers. Data in a computer is transmitted to another computer located across continents almost instantaneously using telephone, microwaves or radio links. The long distance communication link between a computer and a remote terminal was set up around 1965. Now networking has become a very important part of computing activity.

#### 6.2 Network

A large number of computers are interconnected by copper wire, fiber optic cable, microwave and infrared or through satellite.

A system consisting of connected nodes made to share data, hardware and software is called a Computer Network.

#### 6.3 Some Important Reasons for Networking

- \* **Sharing of resources:** Primary goal of a computer network is to share resources. For example several PCs can be connected to a single expensive line printer.
- \* **Sharing information:** Information on a single computer can be accessed by other computers in the network. Duplication of data file on separate PCs can be avoided.

- \* **Communication:** When several PCs are connected to each other, messages can be sent and received. From a remote location, a mobile salesman can relay important messages to the central office regarding orders. Relevant databases are updated and the business commitments are fulfilled.

#### **6.4 Applications of Network**

The following are the areas where computer networks are employed.

- Electronic data interchange
- Tele-conferencing
- Cellular telephone
- Cable Television
- Financial services, marketing and sales
- Reservation of Airlines, trains, Theatres and buses
- Telemedicine
- ATM
- Internet banking

Several educational institutions, businesses and other organizations have discovered the benefits of computer networks. Users can share data and programmes. They can co-operate on projects to maximize the usage of available expertise and talent.

#### **6.5 Benefits of Network**

- Effective handling of personal communications
- Allowing several users to access simultaneously Important programs and data:
- Making it easy for the users to keep all critical data on shared storage device and safeguard the data.
- Allowing people to share costly equipment.

The computer communication should ensure safe, secure and reliable data transfer.

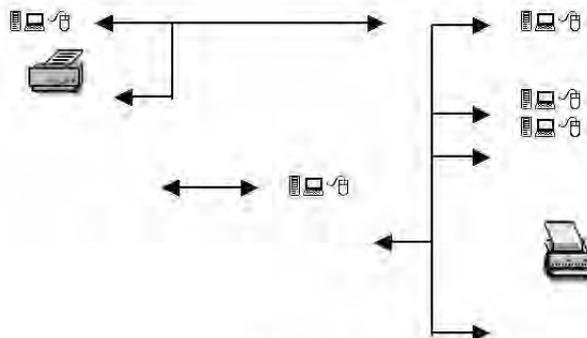
- Safe : The data received is the same as the data sent
- Secure : The data being transferred cannot be damaged either will fully or accidentally.
- Reliable: Both the sender and the receiver knows the status of the data sent. Thus the sender knows whether the receiver got the correct data or not.

## 6.6 Types of Network

The following are the general types of networks used today.

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)

A network connecting systems and devices inside a single building or buildings close to each other is called Local Area Network (LAN) (Fig.6.1). Generally LANs do not use the telephone network. They are connected either by wire or wireless. Wired connection may be using twisted pairs, coaxial cables or Fiber Optic cables. In a wireless LAN, connections may be using infrared or radio waves. Wireless networks are useful when computers are portable. However, wireless network communicates slowly than a wired network.



**Fig. 6.1 Local Area Network**

The number of Computers in the network is between two to several hundreds. LAN is generally used to share hardware, software and data. A computer sharing software package and hard disk is called a file server or network server.

A Network that spans a geographical area covering a Metropolitan city is called Metropolitan Area Network (MAN)

A WAN is typically two or more LANs connected together across a wide geographical area. The individual LANs separated by large distances may be connected by dedicated links, fiber-optic cables or satellite links.

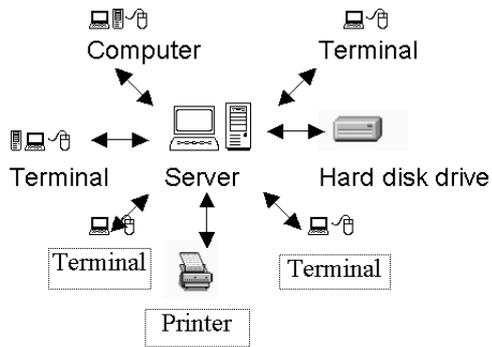
## **6.7 Network Topology**

The network topology is the structure or layout of the communication channels that connects the various computers on the network. Each computer in the network is called a node.

There are a number of factors that determine the topology suitable for a given situation. Some of the important consideration is the type of nodes, the expected performance, type of wiring (physical link) used and the cost.

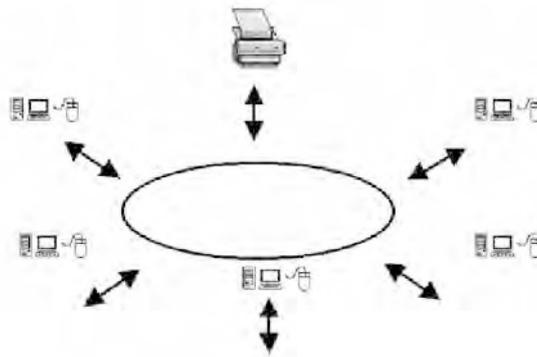
Network can be laid out in different ways. The five common topologies are star, ring, bus, hybrid and FDDI.

**Star Network :** In a star network all computers and other communication devices are connected to a central hub. (Fig.6.2) Such as a file server or host computer usually by a Unshielded Twisted Pair (UTP) cables.



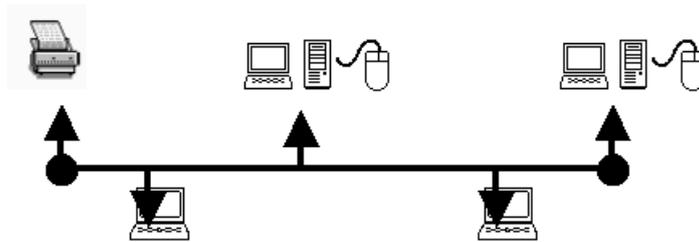
**Fig. 6.2 Star network**

**Ring Network:** In a ring network computers and other communication devices are connected in a continuous loop (Fig. 6.3). Electronic data are passed around the ring in one direction, with each node serving as a repeater until it reaches the right destination. There is no central host computer or server.



**Fig. 6.3 Ring Network**

**Bus Network:** In a bus network all communication devices are connected to a common cable called bus (Fig. 6.4). There is no central computer or server. The data transmission is bidirectional.



**Fig 6.4. Bus Network**

**Hybrid Network:** A hybrid network is a combination of the above three networks suited to the need.

**FDDI Network:** A FDDI network (pronounced as fiddy short for Fiber Distributed Data Interface) is a high-speed network using fiber optic cable. It is used for high tech purposes such as electronic images, high-resolution graphics and digital video. The main disadvantage is its high cost.

## **6.8 Basic Elements in Networking**

All networks require the following three elements

### **1. Network services**

Network services are provided by numerous combinations of computer hardware and software. Depending upon the task, network services require data, input/output resources and processing power to accomplish their goal.

## **2. Transmission media**

Transmission media is the pathway for contacting each computer with other. Transmission media include cables and wireless Technologies that allows networked devices to contact each other. This provides a message delivery path.

## **3. Protocols**

A protocol can be one rule or a set of rules and standards that allow different devices to hold conversations.

### **6.9 Common Network Services**

The following common network services are available.

#### **6.9.1 File Services**

Those are the primary services offered by the computer networks. This improves the efficient storage and retrieval of computer data. The service function includes.

- File transfer –Rapidly move files from place to place regardless of file size, distance and Local operating system.
- File storage and data migration – Increasing amount of Computer data has caused the development of several storage devices. Network applications are well suited to control data storage activity on different storage systems. Some data becomes less used after certain time. For example higher secondary examination result posted on the web becomes less used after a week. Such data can be moved from one storage media (say hard disc of the computer) to another, less expensive media (say an optical disk) is called data migration.

- File update synchronization – Network service keeps track of date and time of intermediate changes of a specific file. Using this information, it automatically updates all file locations with the latest version.
- File archiving – All organizations create duplicate copies of critical data and files in the storage device. This practice is called file archiving or file backup. In case of original file getting damaged, Computer Operator uses the Network to retrieve the duplicate file. File archiving becomes easier and safe when storage devices are connected in the Network.

### **6.9.2 Print services**

Network application that control manage access to printers and fax equipments. The print service function includes

- Provide multiple access (more than one user, use the network) – reduce the number of printers required for the organization.
- Eliminates distance constraints – take a printout at a different location.
- Handle simultaneous requests – queue print jobs reducing the computer time.
- Share specialized equipments-Some printers are designed for specific use such as high-speed output, large size formals or colour prints. Specialised equipments may be costlier or may not be frequently used by the user, when numerous clients are using the network, printer use is optimized.
- Network fax service – Fax service is integrated in the network. The computer in the network sends the digital document image to any location. This reduces the time and paper handling.

### **6.9.3 Message services**

Message services include storing, accessing and delivering text, binary, graphic digitized video and audio data. Unlike file services, message services deal actively with communication interactions between computer users applications, network applications or documents.

### **6.9.4 Application Services**

Application services are the network services that run software for network clients. They are different from file services because they allow computers to share processing power, not just share data.

Data communication is the process of sending data electronically from one location to another. Linking one computer to another permits the power and resources of that computer to be tapped. It also makes possible the updating and sharing of data at different locations.

## **6.10 Co-ordinating Data Communication**

The device that coordinates the data transfer is called Network interface card (NIC). NIC is fixed in the computer and communication channel is connected to it. Ethernet, Arcnet and token ring are the examples for the NIC. Protocol specifies the procedures for establishing maintaining and terminating data transfer.

In 1978, the International Standards organization proposed protocol known as open system interconnection (OSI). The OSI provided a network architecture with seven layers. Fig.6.5 gives the seven layers and the respective functions. This architecture helps to communicate between Network of dissimilar nodes and channels.

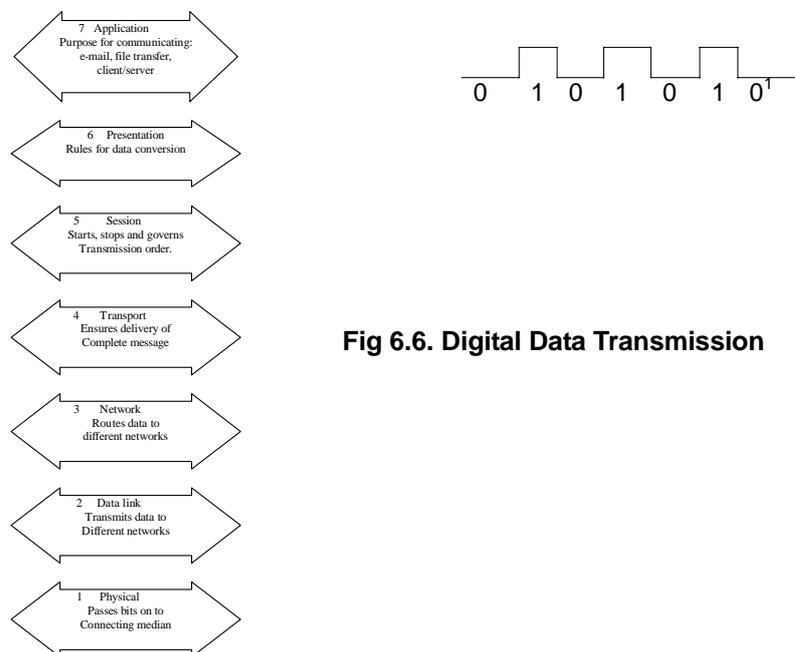
## 6.11 Forms of Data Transmission

Data is transmitted in two forms

1. Analog data transmission
2. Digital data transmission

Analog data transmission is the transmission of data in a continuous waveform. The telephone system, for instance, is designed for analog data transmission. Analog signals are sometimes modulated or encoded to represent binary data.

Digital data transmission is the widely used communication system in the world. The distinct electrical state of 'on' and 'off' is represented by 1 and 0 respectively. Digital data transmission as shown in Fig.6.6 is faster and more efficient than analog. All computers understand and work only in digital forms



**Fig 6.6. Digital Data Transmission**

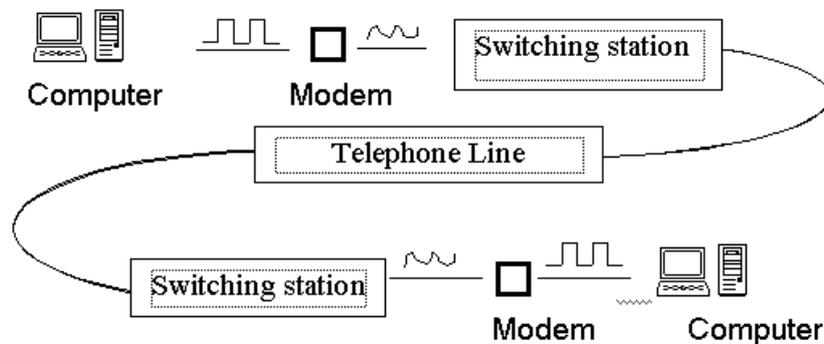
**Fig 6.5. Seven Layers of Protocols**

## 6.12 Modem

Computers at different parts of the world are connected by telephone lines. The telephone converts the voice at one end into an electric signal that can flow through a telephone cable. The telephone at the receiving end converts this electric signal into voice. Hence the receiver could hear the voice. The process of converting sound or data into a signal that can flow through the telephone wire is called modulation.

The reverse process is called demodulation. The telephone instrument contains the necessary circuit to perform these activities. The device that accomplishes **modulation** – **demodulation** process is called a modem. It is known that the electrical and sound signals are analog - which continuously vary with time.

The figure 6.7 shows the relationship of modem to communication link



**Fig . 6.7 Communication Using Modem**

Equipments (DTE) are connected through modem and Telephone line. The modems are the Data Circuit Terminating Equipments (DCE). DTE creates a digital signal and modulates using the modem. Then the signals relayed through an interface. The second modem at the receiving end demodulates into a form that the computer

can accept. A modem that has extra functions such as automatic answering and dialing is called intelligent Modems.

### 6.13 Data Transmission Rate

The speed at which data travel over a communication channel is called the communication rate. The rate at which the data are transferred is expressed in terms of bits per second (bps)

### 6.14 Transmission Mode

When two computers are in communication, data transmission may occur in one of the three modes (Fig.6.8).

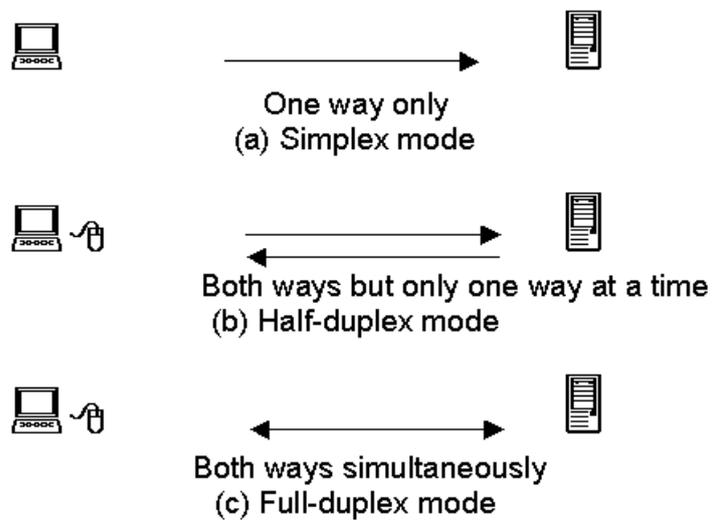


Fig. 6.8 Transmission modes

#### 6.14.1 Simplex mode

In simplex mode, data can be transmitted in one direction as shown in the figure. The device using the simplex mode of transmission can either send or receive data, but it cannot do both. An example is the traditional television broadcast, in which the signal

is sent from the transmitter to the TV. There is no return signal. In other words a TV cannot send a signal to the transmitter.

#### **6.14.2 Half duplex mode**

In Half duplex mode data can be transmitted back and forth between two stations. But at any point of time data can go in any one direction only. This arrangement resembles traffic on a one-lane bridge. When traffic moves in one direction, traffic on the opposite direction is to wait and take their turn. The common example is the walky-talky, wherein one waits for his turn while the other talks.

#### **6.14.3 Full duplex mode**

In full duplex mode a device can simultaneously send or receive data. This arrangement resembles traffic on a two-way bridge, traffic moving on both directions simultaneously. An example is two people on the telephone talking and listening simultaneously. Communication in full duplex mode is faster. Full duplex transmission is used in large computer systems. Products like “Microsoft Net Meeting’ supports such two way interaction

### **6.15 Internet**

Several networks, small and big all over the world, are connected together to form a Global network called the Internet. Today’s Internet is a network of about 50 million or more computers spread across 200 countries. Anyone connected to the Internet can reach, communicate and access information from any other computer connected to it.

Some of the Internet users are

- Students
- Faculty members
- Scientists

- Executives and Corporate members
- Government employees.

The Internet protocol (IP) addressing system is used to keep track of the million of users. Each computer on net is called a host. The IP addressing system uses the letter addressing system and number addressing systems.

### **6.16 Communication Protocol**

Internet is a packet-switching network. Here is how packet-switching works: A sending computer breaks an electronic message into packets. The various packets are sent through a communication network-often by different routes, at different speeds and sandwiched in between packets from other messages. Once the packets arrive at the destination, the receiving computer reassembles the packets in proper sequence. The packet switching is suitable for data transmission. The software that is responsible for making the Internet function efficiently is TCP/IP. TCP/IP is made up of two components. TCP stands for transmission control protocol and IP stands for Internet Protocol.

TCP breaks up the data to be sent into little packets. It guarantees that any data sent to the destination computer reaches intact. It makes the process appear as if one computer is directly connected to the other providing what appears to be a dedicated connection.

IP is a set of conventions used to pass packets from one host to another. It is responsible for routing the packets to a desired destination IP address.

### **6.17 Who Governs The Internet ?**

The Internet as a whole does not have a single controller. But the Internet society, which is a voluntary membership organization, takes the responsibility to promote global information exchange through the Internet technology. Internet Corporation for Assigned Names and

Numbers (ICANN) administers the domain name registration. It helps to avoid a name which is already registered.

### **6.18 Future of Internet**

The popularity of Internet is growing ever since its evolution 20 years ago. This will bring out

- New standard protocol
- International connections
- Consumer civilization
- Data sharing in research and Engineering

### **6.19 Uses of Internet**

The following are some of the popular Internet tools, used by the million of the users.

#### **World Wide Web**

Web is a multimedia portion of the Internet. It consists of an interconnection system of sites or servers all over the world that can store information in the multimedia form. The Multimedia sites include text, animated graph, voice and images.

The World Wide Web is the most graphically inviting and easily navigable section of the Internet. It contains several millions of pages of information. Each page is called a web page. A group of related web pages linked together forms a web site. The first page of the website is called a Home page. The Home page usually contains information about the site and links to other pages on that site. The Fig.6.9 gives the home page of Indian Space Research Organization ( ISRO ).



**Fig. 6.9. Home Page of ISRO**

Every web page has a unique address called the Uniform Resource Locator or URL. The URL locates the pages on the Internet. An example of URL is

[http:// www.country-watch.com/India](http://www.country-watch.com/India)

where http stands for Hypertext Transfer Protocol (HTTP). This protocol is meant for transferring the web files. The www portion of the address stands for “world wide web” and the next part country-watch.com is the domain name. Generally, the domain name will be followed by directory path and the specific document address separated by slashes. Looking for information on the Internet is called surfing or browsing. To browse the Internet, a software called web browser is used. Web browser translates HTML documents of the website and allows to view it on the screen. Examples of web browsers are Internet Explorer and Netscape Navigator. The mouse pointer moves over a underlined or highlighted words and images

change to a hand icon. This is called an hyperlink. This indicates the link to other sites. To go to one of the linked sites, just click the mouse on the hyperlink.

**E-mail** - The World Wide Web is getting a lot of attention due to its main attraction of Electronic mail. Electronic mail is usually used to exchange messages and data files. Each user is assigned an electronic mail box. Using mail services, one can scan a list of messages that can be sent to anyone who has the proper email identification. The message sent to any one resides in the mailbox till it is opened. Many other features of standard mail delivery are implemented in email.

**Usenet News Groups:** Electronic discussion groups. User network abbreviated as usenet is essentially a giant disbursed bulletin board. Electronic discussion groups that focus on specific topic forms, computer forums.

**Mailing list:** Email based discussion groups combining E-mail, news groups and mailing lists send messages on a particular subject. Automatically messages reach the mailbox of that group.

**FTP:** File Transfer Protocol, abbreviated as FTP is used for the net user for transferring files around the world. The transfer includes software, games, photos, maps, music and such other relevant materials.

**Telnet:** Telnet is a protocol that allows the user to connect to a remote computer. This feature is used to communicate a microcomputer with mainframe.

## **6.20 Getting connected to Internet**

To use an Internet in the simplest way, we need

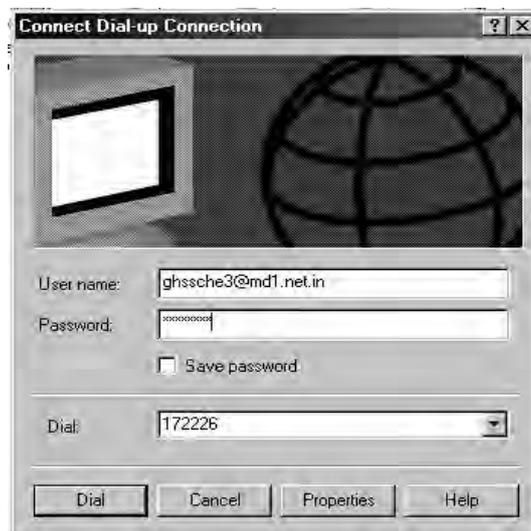
- A Computer
- A Telephone line

- A Modem
- Internet Service Provided or ISP

The ISPs are the companies which allows the user to use the Internet for a price. One has to register with the ISP for an Internet account. ISP provides the following:

- User name - An unique name that identifies the user
- Password - A secret code that prevents other users from using your account
- E-mail address- Unique address that you can send or receive E-mails.
- Access telephone number - Internet users can use this number to connect to the service provider.

Fig.6.10 shows dialog boxes on the computer screen wherein the user name (Govt. Higher Secondary School, Chennai -600 003 abbreviated as a ghssch3), a password (alpha numeric of word length 8 characters appearing as 'x') and access telephone number are entered. By clicking on the dial button, the modem establishes a connection with the ISP.



**Fig.6.10. Dialogue Box for Connecting to the Internet**

There are two ways to look for the information on the Web. If the URL of the website is known, enter it on the address bar (Fig.6.11). If URL is not known, then 'Search Engines' will help us to get the information. Search Engines are tools that allow the user to find specific document through key words or menu choices. Some of the popular Search engines are Yahoo, Lycos, AltaVista, Hotbot , Google and Askjeeves.



**Fig.6.11. Entering the URL**

Internet explorer helps to use the net more effectively with the navigation buttons (Fig.6.12) on the toolbar.



**Fig.6.12. Navigation Buttons**

**1. Back button:** This button helps to go back to the previous link. The small triangle adjacent to it displays a dropdown list of several recently used pages. Instead of pressing the back button several times, select a page from the list.

**2. Forward button:** This is a similar to the back button. One can jump forward by one page or several pages.

**3. Stop button:** After clicking on a link, some times we may realize that the link is not necessary. The click stop button and move back without wasting time.

**4. Refresh button:** Sometimes a page may take longer time or may not load properly. Click on the refresh button, helps reload the page faster.

**5. Home button:** While following the hyperlink, it is very easy to get lost. The home button reverts to the home page of the website.

## **6.21 Popular uses of the web**

**Research:** The web provides research materials from libraries, research institutions, encyclopedia, magazines and newspapers. Some sample sites  
www.encarta.com the Internet Public Library site www.ipl.com and Library of Congress www.loc.gov.

**Chatting:** Some websites proved chat rooms to interact with an individual or a group.

**Free-wares:** Some sites provide free download of software's, tutorials and benchmarks.

**Education online:** Educational institutions offer courses via the web. Student can attend and interact in a class from home using a computer.

**Online services:** Online shopping, online booking for travels and entertainments managing investments are the upcoming areas of Internet that reaches every home.

**Job searches:** The digital revolution is changing everything it touches and the job market is no exception. Several web sites are assisting people in finding internship, jobs and helps companies to fill job

vacancies. There are sites relating to specific job and profession also. Some of these sites charge a fee for the services while others are free.

## **6.22 Intranet and Extranet**

Many organizations have Local Area Network that allows their computers to share files, data, printers and other resources. Sometimes these private network uses TCP / IP and other Internet standard protocols and hence they are called intranet. All the Internet services such as web pages, email, chat; usenet and FTP are provided on the intranet to serve the organization. Creating a web page on the intranet is simple because they use only Word-Processing Software One of the main consideration of the intranet is security. The sensitive company data available on the intranet is protected from the outside world.

Taking intranet technology a few steps forward extranets are useful in the business world. Intranet connecting selected customers, suppliers and offices in addition to the internal personnel, is called extranet. By using extranet business organizations can save telephone charges. For example a can manufacturing company can extend their intranet to their dealers and customers for support and service.

### **EXERCISES**

#### **I. Fill in the blanks:**

1. \_\_\_\_\_ is a typically two or more LANs connected together across a wide geographical area.
2. \_\_\_\_\_ network computers and other communication devices are connected in a continuous loop.
3. In a high speed network \_\_\_\_\_ cables are used.

4. The device that coordinates the data transfer is called \_\_\_\_\_, \_\_\_\_\_
5. The OSI provided a network architecture with \_\_\_\_\_ layers.
6. All computers understand and work only in \_\_\_\_\_ form.
7. \_\_\_\_\_ signals continuously vary with time.
8. Communication in \_\_\_\_\_ mode is faster.
9. \_\_\_\_\_ protocol is used to assist communication between a microcomputer and a mainframe.
10. \_\_\_\_\_ tools, that allow the Internet user to find specific document through keywords or menu choices.

## **II. Answer the following (short answer)**

1. What are the reasons for networking?
2. Mention a few areas, where computer networks are employed
3. What are the elements that a computer communication should ensure?
4. List the general types of networks used today.
5. Explain WAN.
6. How data is transmitted in different forms?
7. What are the transmission modes?
8. What is TCP?
9. What is the role of ICANN?
10. Explain URL.