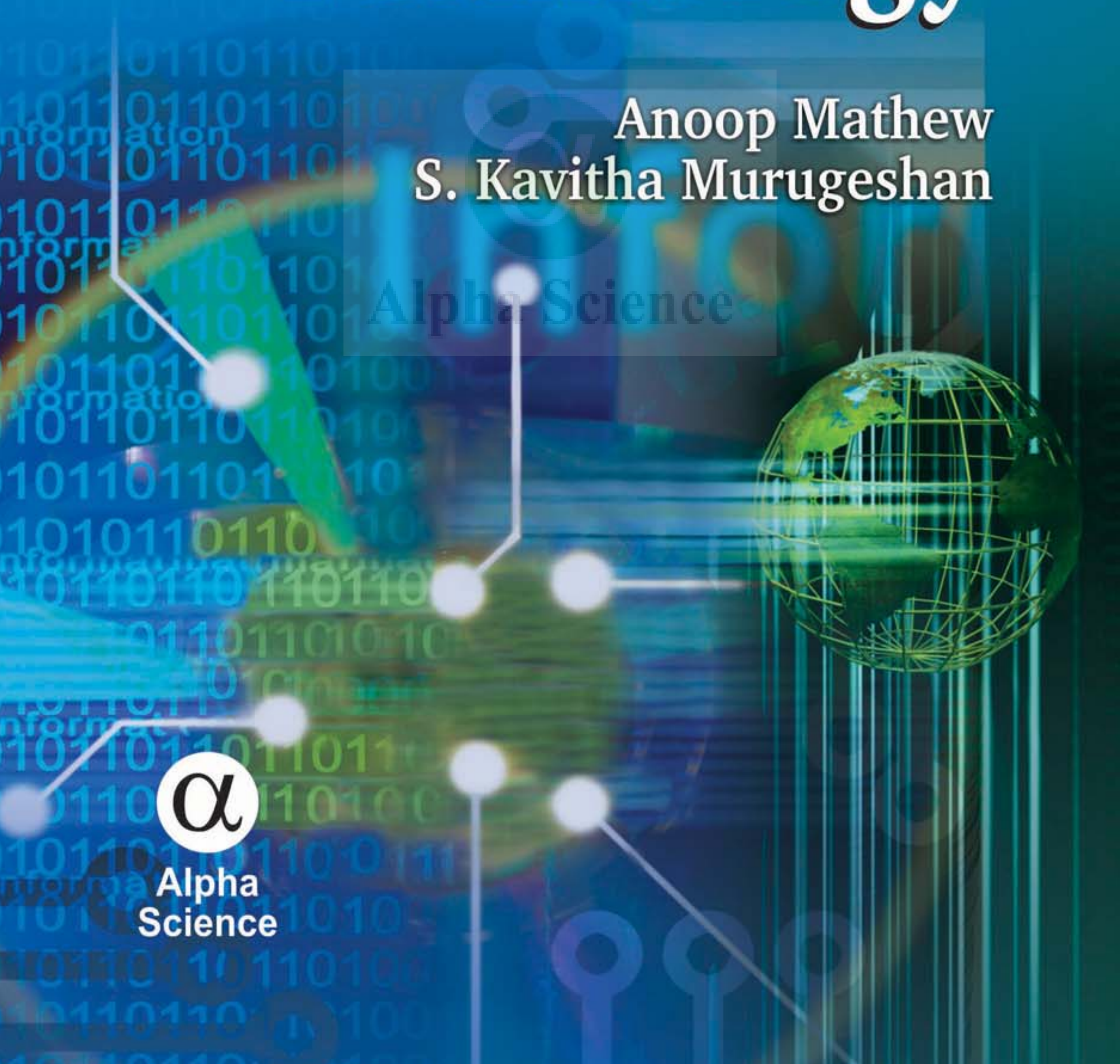
Fundamentals of Information Technology

Anoop Mathew
S. Kavitha Murugeshan



Alpha Science

Fundamentals of

Information Technology

Alpha Science

Alpha Science

Fundamentals of Information Technology

Anoop Mathew
S. Kavitha Murugeshan

Alpha Science International Ltd.
Oxford, U.K.

Fundamentals of Information Technology
236 pgs. | 107 figs. | 09 tbls.

Anoop Mathew
Department of Electronics and Communication Engineering

S. Kavitha Murugeshan
Department of Computer Science and Engineering

Vedavyasa Institute of Technology
Malapuram, Kerala

ALPHA SCIENCE INTERNATIONAL LTD.
7200 The Quorum, Oxford Business Park North
Garsington Road, Oxford OX4 2JZ, U.K.

www.alphasci.com

ISBN 978-1-84265-788-1
E-ISBN 978-1-78332-005-9

Printed in India

PREFACE

There are hundreds of books on Information Technology (IT) ranging from primers to highly theoretical manuscripts. There are a number of textbooks aimed at different readership levels as well. Then why this book?

Purpose: This text addresses a broad audience. It is first and foremost, a text book for technology and engineering students taking a beginning course in Electrical Engineering. It is also written for students taking advanced courses in electrical engineering who need to hone up on some of the basic principles. Finally, mechanical engineers, who simply want to know what this technology is all about – and this number is surprisingly high – will find this book a useful source of comprehensive up-to-date information.

This text is unique in that is responds to the needs and interest of most students seeking to enter the profession and enables them to gain important knowledge whether be it in the theoretical or practical realm. It provides the student with a strong foundation through clear, logical explanation of the basic concepts, augmented by lots of examples, graphical presentations and solution to the problems.

How much to learn: If you hope to become a professional in the information technology field, you will need a strong understanding of the software and semi conductor technology. This knowledge can be acquired only if you have first established a solid background in physics, mathematics and electronics. There is no short cut to this knowledge base. Information technology is a special field, one that integrates knowledge from diverse areas to devise new concepts. As a result, the broader the background and experience you bring to your job, the more valuable you are as a professional. Finally, students interested in becoming IT managers must combine technical knowledge with insight to complex legal regulations and a keen understanding of how a business operates.

To the instructor: It would be inappropriate to tell you how to use this or any text book. Thus, we would like to simply share with you the plan of this book to help you prepare your course syllabus.

Most topics in this book are presented on two levels: "Basics" and "A deeper-look". At the basic level, we introduce the main ideas and the principles behind the technology. The basic sections are necessary to give students with little background in computer science a fundamental understanding of the topic. These sections are presented in a simple manner for the beginning student to grasp the subject matter easily and quickly.

At a deeper-look level, we include a more theoretical, highly detailed discussion of the same material and add new topics. This gives you considerable flexibility based on the technical level of your class and the length of the course. The deeper-look sections have two goals: First, they cover the material in greater depth, thereby involving more theory assuming a stronger background in electronics and computer sciences. Second, they prepare students for further course work that will involve complex theory and the kinds of professional responsibilities they will encounter in the workplace. These sections bring the students to the next level and provide more insight into information technology or computer science. They also provide many topics for student research should this be an integral part of your course.

It should be emphasised that the basic and deeper-look sections are closely related and if the advanced students should feel the need for a "refresher course", we strongly recommend that they reread the introductory section before proceeding to the more complex material.

To sum up, we believe this text book provides you with all the material you need to devise a course that will be suitable for the technical level of your students.

Anoop Mathew
S. Kavitha Murugeshan

ACKNOWLEDGEMENTS

The authors would like to take the opportunity to thank all those individuals who have contributed or helped in some way in the preparation of this text. Particular thanks must go to our management and colleagues of Vedavyasa Institute of Technology for their encouragement and support. Also, the authors would like to thank their respective spouses for their understanding and forbearance shown when the preparation of the book took time that could have been spent with the family.

We are grateful for the many talented engineers that we have been fortunate to work with and learn from over the years. More directly, several friends and colleagues were kind enough to review the initial draft, provide feedback on the content, and bring to our attention details that required correction and clarification. Their thorough reading provided welcome and valuable perspectives on everything ranging from fine technical points to overall flow and style. We are grateful to the reviewers for the comments and suggestions provided. We would like to thank the midshipmen who have taken classes from us. They have been willing or unwilling guinea pigs in testing out many of the ideas and problems in this book.

Anoop Mathew
S. Kavitha Murugeshan

CONTENTS

CHAPTER

1

BASICS OF INFORMATION TECHNOLOGY

1.1 INTRODUCTION

Information Technology refers to the management and use of the information using computer-based tools. It refers to both hardware and software. Mostly, it is the term used to refer to business applications of computer technology rather than scientific applications. Information Technology (IT) can be used potentially in every sector of the economy. Information Technology has been a dynamic sector in many developed economies and India has stood out as a developing country where IT sector has grown rapidly. Information Technology jobs in India are the first choice career for the bright brains who are innovative. Hence, IT jobs in India are also booming with increasing demand for information technology professionals. The IT industry has great scope for people as it provides employment to technical and non-technical graduates and has the capability to generate huge foreign exchange inflow for India. Many countries get benefits in terms of labor costs and business processes. The immense expansion in networking technologies is expected to continue into the next decade also. IT will bring about an improvement in the quality of life as it impacts on global competitiveness. The influence of IT and its applications are spread across various sectors like banking, communication engineering, engineering design, education etc.

1.2 INFORMATION SYSTEM (IS)

An Information System (IS) is typically considered to be a set of interrelated elements or components that can collect (input), manipulate (processes), and disseminate (output) data and information and provide a feedback mechanism to meet an objective.

In today's world, information systems are indispensable to businesses, industries, and organizations, to meet the future challenges. Organizations need ISs to meet the information needs of its employees. More and more business organizations are using information systems to improve the way they conduct business, as discussed below:

- IS, for many businesses, provide an opportunity to do business in a new way and thus gain huge profits and market shares.

- ISs facilitate the acquisition, transformation, and distribution of information. ISs can improve decision making, enhance organizational performance, and help increase profitability.
- The use of information systems to add value to the organization can also give an organization a competitive advantage. It provides significant, long-term benefit to a company over its competition. ISs are used for strategic planning in a competitive world. It allows changing the structure, creating of new services, and improving upon the existing services.
- Damaged or lost data can cause disruptions in normal business activities leading to financial losses, law suits, etc. ISs help an organization to better manage and secure its critical data.
- ISs also improve integration and the work processes.
- Managing and controlling operations of an organization located in different countries becomes almost impossible without the support of an efficient IS.

1.3 CLASSIFICATION OF INFORMATION SYSTEMS

Information systems can be classified broadly as operations support system and management support systems. Classification tree of a typical information system is shown in Fig. 1.1.

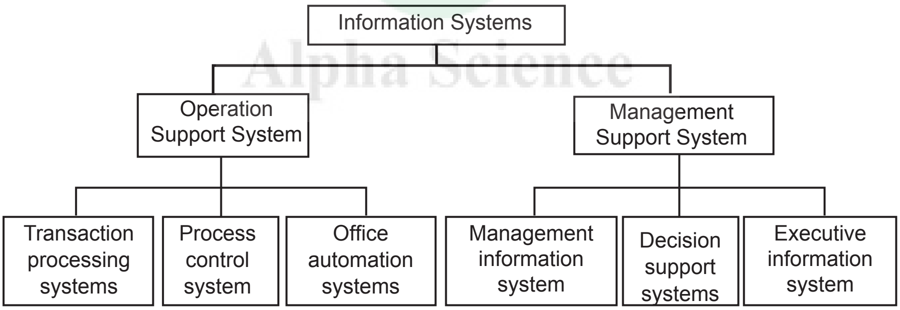


Fig. 1.1 Classification of Information Systems

Operations Support Systems: Operations support systems process data generated by business operations has the following categories:

(i) Transaction processing systems

(ii) Process control systems

(iii) Office automation systems

The features of Operations support systems are:

Transaction Processing Systems: Process business exchanges, Maintain records about the exchanges, Handle routine and critical tasks and Perform simple calculations.

Process Control Systems: Monitor and control industrial processes.

Office Automation Systems: Automate office procedures and enhance office communications and productivity.

Management Support Systems: Management Support Systems provide information and support needed for effective decision making by managers and has the following categories:

(i) Management Information System

(ii) Decision Support Systems

(iii) Executive Information System

The features of Management Support Systems are:

Management Information Systems: Process Routine information for routine decisions, Increased Operational efficiency, Use transaction data as main input, Databases integrate MIS in different functional areas.

Decision Support System: Interactive support for non-routine decisions or problems, End-users are more involved in creating a DSS than an MIS.

Executive Information Systems: Provide critical information tailored to the information needs of executives.

Other categories widely used are:

(a) Expert systems: Expert Systems are knowledge-based systems that provides expert advice and act as expert consultants to the users.

(b) End user computing systems: End user computing systems support the direct, hands on use of computers by end users for operational and managerial applications.

(c) Business information systems: Business information systems support the operational and managerial applications of the basic business functions of a firm.

(d) Strategic information systems: Strategic information systems provide a firm which strategic products, services, and capabilities for competitive advantage.

The main disciplines used in IS field are Computer Science, Political Science, Psychology, Operations Research, Linguistics, Sociology and Organization theory.

1.4 SECURITY ISSUES IN INFORMATION TECHNOLOGY

The advancement of science and technology during the last century makes the basement of new inventions makes the human more inventive and productive and highly secured in Data Communications, Securing Access to resources, to prevent Hacker Attacks.

Computer security is the process of preventing and detecting unauthorized use of your computer. Prevention measures help you to stop unauthorized users (also known as "intruders") from accessing any part of your computer system.

Detection helps you to determine whether or not someone attempted to break into your system, if they were successful, and what they may have done.

We use computers for everything from banking and investing to shopping and communicating with others through email or chat programs. Although you may not consider your communications "top secret," you probably do not want strangers reading your email, using your computer to attack other systems, sending forged email from your computer, or examining personal information stored on your computer (such as financial statements).

Intruders (also referred to as hackers, attackers, or crackers) may not care about your identity. Often they want to gain control of your computer so they can use it to launch attacks on other computer systems. Having control of your computer gives them the ability to hide their true location as they launch attacks, often against high-profile computer systems such as government or financial systems. Even if you have a computer connected to the Internet only to play the latest games or to send email to friends and family, your computer may be a target. Intruders may be able to watch all your actions on the computer, or cause damage to your computer by reformatting your hard drive or changing your data. Also, some software applications have default settings that allow other users to access your computer unless you change the settings to be more secure. Examples include chat programs that let outsiders execute commands on your computer or web browsers that could allow someone to place harmful programs on your computer that run when you click on them.

The methods to ensure Internet security are:

- **Use of firewall:** Network firewalls (whether software or hardware-based) can provide some degree of protection against these attacks.
- **Don't open unknown email attachments:** Before opening any email attachments, be sure you know the source of the attachment.
- **Don't run programs of unknown origin:** Never runs a program unless you know it to be authored by a person or company that you trust.
- **Disable hidden filename extensions:** Windows operating systems contain an option to "Hide file extensions for known file types". The option is enabled by default, but disables this option in order to have file extensions displayed by Windows. After disabling this option, there are still some file extensions that, by default, will continue to remain hidden.
- **Keep all applications, including your operating system, patched:** Vendors will usually release patches for their software when vulnerability has been discovered. Most product documentation offers a method to get updates and patches.
- **Turn off your computer or disconnect from the network when not in use:** Turn off the computer or disconnect its Ethernet interface when you are not using it. An intruder cannot attack your computer if it is powered off or otherwise completely disconnected from the network.
- **Disable Java, JavaScript, and ActiveX if possible:** Be aware of the risks involved in the use of "mobile code" such as ActiveX, Java, and JavaScript.

- **Disable scripting features in email programs:** Because many email programs use the same code as web browsers to display HTML, vulnerabilities that affect ActiveX, Java, and JavaScript are often applicable to email as well as web pages.
- **Make regular backups of critical data:** Keep a copy of important files on removable media such as ZIP disks or recordable CD-ROM disks (CD-R or CD-RW disks). Use software backup tools if available, and store the backup disks somewhere away from the computer.
- **Make a boot disk in case your computer is damaged or compromised:** To aid in recovering from a security breach or hard disk failure, create a boot disk on a floppy disk which will help when recovering a computer after such an event has occurred.

REVIEW QUESTIONS

1. Define Data, Information and Knowledge.
2. Explain the characteristics of Information System.
3. Types of Information System.
4. Explain major categories of Operations support systems.
5. Explain Transaction processing systems.
6. Explain Process control systems.
7. Explain Office autom ation systems.
8. Explain Major categories of Management Support Systems.
9. Explain Management Information System.
10. Explain Decision Support Systems.
11. Explain Executive Information System.
12. Explain the need for security in IT.
13. What is a Firewall?
14. Explain the need for efficient information system.

CHAPTER

2

FUNDAMENTALS OF COMPUTER ENGINEERING

2.1 INTRODUCTION

Logically speaking, the first computers were people. That is, electronic computers (and the earlier mechanical computers) were given this name because they performed the work that had previously been assigned to people. "Computer" was originally a job title: it was used to describe those human beings whose job it was to perform the repetitive calculations required to compute such things as navigational tables, tide charts, and planetary positions for astronomical almanacs.

The abacus was an early aid for mathematical computations. Its only value is that it aids the memory of the human performing the calculation. A skilled abacus operator can work on addition and subtraction problems at the speed of a person equipped with a hand calculator (multiplication and division are slower). The abacus is often wrongly attributed to China. In fact, the oldest surviving abacus was used in 300 B.C. by the Babylonians. The abacus is still in use today, principally in the Far East.

In 1642, Blaise Pascal invented the Pascaline as an aid for his father who was a tax collector. Pascal built 50 of this gear-driven one-function calculator (it could only add) but couldn't sell many because of their exorbitant cost and because they really weren't that accurate. In 1801, a Frenchman Joseph Marie Jacquard invented a power loom that could base its weave (and hence the design on the fabric) upon a pattern automatically read from punched wooden cards, held together in a long row by rope. Descendents of these punched cards have been in use ever since 2000.

By 1822, the English mathematician Charles Babbage proposed a steam driven calculating machine the size of a room, which he called the Difference Engine. This machine would be able to compute tables of numbers, such as logarithm tables. By promoting their commercial and military navies, the British government had managed to become the earth's greatest empire. But in that time frame, the British government was publishing a seven volume set of navigation tables which came with a companion volume of corrections which showed that the set had over 1000 numerical errors. It was hoped that Babbage's machine could eliminate errors in these types of tables. But construction of Babbage's Difference Engine proved exceedingly difficult and the project soon became the most expensive government funded project up to that point in English history.

2.2 CHARACTERISTICS OF COMPUTER

A computer is an electronic device that stores, retrieves, and processes data, and can be programmed with instructions. It is composed of hardware and software components, and can exist in a variety of sizes and configurations. The actual machinery wires, transistors, and circuits are called hardware; the instructions and data are called software.

The increasing popularity of computer has proved that it is very powerful and useful machine. The power and usefulness of this popular machine is mainly due to the following features:

(a) **Speed:** Computer can work at a very high speed. The time taken by the computer to execute instruction is very fast. Its processing speed is measured in a fraction of a second.

(b) **Accuracy:** Though the computer does its work with a very high speed, it does not make any mistake unless the user gives it a wrong instruction. Computers are 100% accurate. But if we feed wrong data to the computer, it returns the same wrong output or information called GIGO (Garbage in garbage out).

(c) **Diligence:** Computers can work for many hours continuously without taking any rest and without decreasing its speed, accuracy and efficiency. It is free from tiredness, lack of concentration, fatigue etc.

(d) **Versatile:** Computer is a versatile machine which can do varieties of task such as simple calculation to a complex and logical operation. It is used in various fields for various purposes.

(e) **Storage:** Computer has mass storage section where we can store large volume of data for future work. Such data are easily recall from the secondary storage devices like floppy disk (FDD), hard disk (HDD), compact disk (CD), etc.

(f) **Automatic:** Once the instruction to do any work is given to the computer, the computer does its work automatically.

(g) **Reliability:** Since computer can do its work very fast, without making any mistake and without taking rest and is able to store data for future use, it is a very reliable or trustworthy machine.

2.3 CAPABILITIES AND LIMITATIONS

Based on the characteristics, the common capabilities of computer are:

1. Can solve complex calculations quickly which takes a long time to solve manually.
2. Capable of handling and processing large calculations at a single time.
3. All Electronic Items have some form of computing functions.
4. It works faster than a man.
5. Computer and man are capable in giving data and information in the form of communication.

Limitations of computer are:

1. **Programmed by Human:** Though computer is programmed to work efficiently, fast and accurately but it is programmed by human beings to do so. Without a program, computer cannot perform any task. A program is a set of instructions. Computer only follows these instructions. If the instructions are not accurate, the working of computer will not accurate.
2. **Thinking:** The computer cannot think itself. The concept of artificial intelligence shows that the computer can think. But still this concept is dependent on set of instructions provided by the human beings.
3. **Self-care:** Computer cannot care itself like a human. A computer is dependent still to human beings and it's environment for this purpose.
4. **Retrieval of memory:** Computer can retrieve data very fast but this technique is linear. A human being's mind does not follow this rule.
5. **Feelings:** One of the main limits in the computer is of feeling. A computer can not feel about some like a human. A computer cannot meet human in respect of relations. Human can feel, think and caring but a computer machine itself cannot. A computer cannot take place of human because computer is always dependent of human.

2.4 PROBLEM-SOLVING USING COMPUTERS

Problem solving is the process of logically breaking down a problem into three major phases:

1. Stating clearly *what the problem is*?
2. Defining *how the problem can be solved* in a logical sequence of steps?
3. Verifying that the *solution really does solve the stated problem.*

When humans solve problems by themselves, they use a vast array of knowledge and understanding to perform these problem solving steps.

1. Clearly define the problem.
2. Analyze the problem and formulate a method to solve it.
3. Describe the solution in the form of an algorithm.
4. Draw a flowchart of the algorithm.
5. Write the computer program.
6. Compile and run the program (debugging).
7. Test the program (debugging).
8. Interpretation of results.

When humans use a computer to solve a problem, the solution must be coded in a programming language which, as we noted above, is much simpler than the natural language human use with one another. Further, if we think of the human and the computer as a problem solving team, the computer is definitely the *junior partner* in the team effort; specifically:

1. The human does all of the problem statement phase of problem solving.
2. The human then outlines a solution in the form of an *algorithm*, which can be thought of as a high-level program.
3. The human then encodes the algorithm in a computer programming language.
4. The computer compiles (i.e., translates) the program into an electronic form that can be executed directly by the computer hardware; this form is called the *machine code*.
5. The computer runs the machine code to produce an answer.
6. The human validates that the answer is correct, but writing testing programs.

2.5 GENERATIONS OF COMPUTERS – PAST, PRESENT AND FUTURE

Charles Babbage is known as the father of computers because of his immense contribution to the world of programming. His idea was soon developed into a programmable computer that could calculate and print logarithmic tables with huge precision. But there were many practical problems and the progress was slow as below.

2.5.1 First Generation (1940-1956)

The first computers used vacuum tubes for circuitry and magnetic drums for memory, and were often enormous, taking up entire rooms. They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions.

First generation computers relied on machine language, the lowest-level programming language understood by computers, to perform operations, and they could only solve one problem at a time. Input was based on punched cards and paper tape, and output was displayed on printouts.

The UNIVAC and ENIAC computers are examples of first-generation computing devices. During the World War, the U.S military had a demand for fast computers that could perform extremely complex calculations and weather predictions in minutes. This was when the ENIAC was built, by a partnership between University of Pennsylvannia and the U.S government. After the landmark "Von Neumann Architecture" was introduced it considerably increased the speed of the computer since it used only one memory. The UNIVAC was the first commercial computer delivered to a business client, the U.S. Census Bureau in 1951.

Limitations of First Generation Computer:

1. They used valves or vacuum tubes as their main electronic component.
2. They were large in size, slow in processing and had less storage capacity.
3. They consumed lots of electricity and produced lots of heat.

4. Their computing capabilities were limited.
5. They were not so accurate and reliable.
6. They used machine level language for programming.
7. They were very expensive.

2.5.2 Second Generation (1956-1963)

Transistors invented in 1947 by William Shockley and Walter Brattain in the Bell Laboratories of the U.S.A. replaced vacuum tubes and ushered in the second generation of computers. The transistor was far superior to the vacuum tube, allowing computers to become smaller, faster, cheaper, more energy-efficient and more reliable than their first-generation predecessors. Though the transistor still generated a great deal of heat that subjected the computer to damage, it was a vast improvement over the vacuum tube. Second-generation computers still relied on punched cards for input and printouts for output.

Second-generation computers moved from cryptic binary machine language to symbolic, or 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. These were also the first computers that stored their instructions in their memory, which moved from a magnetic drum to magnetic core technology.

2.5.3 Third Generation (1964-1971)

The transistors were found to damage the more sensitive parts of a computer since they generated a lot of heat. In 1958, the IC (Integrated Circuit) was invented by Jack Kilby. This revolutionized computing, since all the electronic components were on a single semiconductor chip made of silicon, drastically reducing the size of the computer. Also, new operating systems were developed, which allowed the running of many applications simultaneously. The development of the integrated circuit was the hallmark of the third generation of computers. Transistors were miniaturized and placed on silicon chips, called semiconductor, which drastically increased the speed and efficiency of computers.

Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system, which allowed the device to run many different applications at one time with a central program that monitored the memory. Computers for the first time became accessible to a mass audience because they were smaller and cheaper than their predecessors.

2.5.4 Fourth Generation (1971-Present)

The microprocessor brought the fourth generation of computers, as thousands of integrated circuits were built onto a single silicon chip. This made the computers smaller, more portable and much faster than before. The Intel 4004 chip, developed in 1971, located all the components of the computer-from the central processing unit and memory to input/output controls-on a single chip.

In 1981, IBM introduced its first computer for the home user, and in 1984 Apple introduced the Macintosh. Microprocessors also moved out of the realm of desktop computers and into many areas of life as more and more everyday products began to use microprocessors. As these small computers became more powerful, they could be linked together to form networks, which eventually led to the development of the Internet. Fourth generation computers also saw the development of Graphical user Interface, the mouse and handheld devices.

2.5.5 Fifth Generation (Present and Beyond) Artificial Intelligence

Fifth generation computing devices include the present day computers and the ones that are being developed, based on artificial intellingence, are still in development, though there are some applications, such as 'Voice recognition' which is a software that is used to recognize the user's voice and respond to it, that are being used today. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation and molecular and nanotechnology will radically change the face of computers in years to come. The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

Table 2.1 Generation of Computers and the components used

Generations of computers	Component used
First Generation	Vacuum tubes
Second Generation	Transistors
Third Generation	Integrated Circuits (IC)
Fourth Generation	Very Large Scale Integrated Circuits(VLSI)
Fifth Generation	Ultra Scale Integrated Circuits (ULSI) Micro Processor (SILICON CHIP)

2.6 MOORE'S LAW

The law is named after Intel co-founder Gordon E. Moore, who observed that the number of transistors on integrated circuits doubles approximately every two years. As a result, the scale gets smaller and smaller. For decades, Intel has met this formidable challenge through investments in technology and manufacturing resulting in the unparalleled silicon expertise that has made Moore's Law a reality.

In a universe where smaller is better, Intel's current process technology the most advanced silicon process in volume production anywhere in the world prints individual lines smaller than a virus and 1,000 times thinner than a human hair and manufactures microprocessors with some features as thin as five atomic layers. As transistor counts climb so does the ability to increase device complexity and integrate many capabilities onto a chip. The cumulative impact of these spiraling increases in capability power the economy and the Internet, running everything from digital phones and PCs to stock markets and spacecraft, and enable today's information-rich, converged digital world.

2.7 CLASSIFICATION OF COMPUTERS

Computers are often classified based on:

(a) Principle of Operation

(b) Configuration and Size

(c) Purpose

2.7.1 Classification based on Principle of Operation

Based on the principle of operation, computers are classified into:

- Analog Computers
- Digital Computers
- Hybrid Computers

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

Digital Computers: A digital computer operates on digital data. 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 more accurate and faster results. Digital computer is well suited for solving complex problems in engineering and technology. Hence, digital computers have an increasing use in the field of design, research and data processing.

Hybrid Computers: A hybrid computer combines the 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.

2.7.2 Types of Computers based on Configuration and Size

There are four different types of computers when we classify them based on their performance and capacity.

- Super Computers
- Mainframe Computers
- Mini Computers
- Micro Computers

Super Computers: Super computers are the best in terms of processing capacity and also the most expensive ones. These computers can process billions of instructions per second. Normally, they will be used for applications which require intensive numerical computations such as stock analysis, weather forecasting etc. Other uses of supercomputers are scientific simulations, (animated) graphics, fluid dynamic calculations, nuclear energy research, electronic design, and analysis of geological data (e.g. in petrochemical prospecting).

Mainframe Computers: Mainframe computers can also process data at very high speeds i.e., hundreds of million instructions per second and they are also quite expensive. Normally, they are used in banking, airlines and railways etc for their applications.

Mini-Computers: Mini computers are lower to mainframe computers in terms of speed and storage capacity. They are less expensive than mainframe computers. Some of the features of mainframes will not be available in mini-computers. Hence, their performance also will be less than that of mainframes.

Micro-Computers: The invention of microprocessor (single chip CPU) gave birth to the much cheaper micro computers. They are further classified into: Desktop Computers, Laptop Computers and Handheld Computers (PDAs).

2.7.3 Classification of Computers based on Purpose

The following are the types of computers classified based on purpose of its use.

- General purpose computer
- Special purpose computer

General purpose Computer: General purpose computers are designed to perform a range of multiple tasks. In other words, general purpose computers are used for any type of applications. They can store different programs and do the jobs as per the instructions specified on those programs. Most of the computers that we see today are general purpose computers. They have the ability to store numerous programs, but lack in speed and efficiency.

Special purpose Computer: Special purpose computer is one that is built for a specific application. Specific purpose computers are designed to handle a specific problem or to perform a specific task.

2.8 PARALLEL COMPUTING

Parallel computing is used to solve a problem using more than one computer or a computer with more than one processor. Parallel computing offers certain advantages like faster computation, fault tolerance and larger amount of memory availability.

Parallel computers can be classified into two, viz; shared memory multiprocessor and distributed memory multicomputer.

2.8.1 Shared Memory Multiprocessor

In the mid-1980s, shared-memory multiprocessors were pretty expensive and pretty rare. Now, as hardware costs are dropping, they are becoming commonplace. Home computer operating systems are providing the capability to use more than one processor to improve system performance. Rather than specialized resources locked away in a central computing facility, these shared-memory processors are often viewed as a logical extension of the desktop. These systems run the same operating system as the desktop and many of the same applications from a workstation will execute on these multiprocessor servers. Shared-memory multiprocessors have a significant advantage over other multiprocessors because all the processors share the same view of the memory.

These processors are also described as uniform memory access (also known as UMA) systems. This designation indicates that memory is equally accessible to all processors with the same performance. The popularity of these systems is not due simply to the demand for high performance computing. These systems are excellent at providing high throughput for a multiprocessing load, and function effectively as high-performance database servers, network servers, and Internet servers. Within limits, their throughput is increased linearly as more processors are added.

2.8.2 Distributed Memory Multicomputer

A distributed-memory multicomputer system is modelled in Fig. 2.1. The system consists of multiple computers, often called nodes, interconnected by a message-passing network. Each node is an autonomous computer consisting of a processor, local memory, and sometimes attached disks or I/O peripherals.

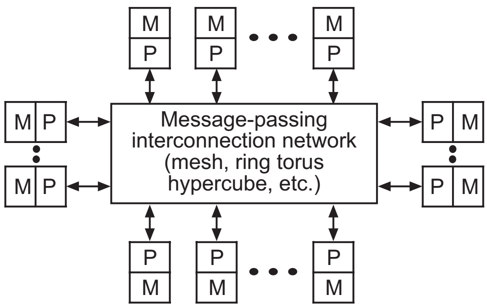


Fig. 2.1 Generic model of a message-passing multicomputer

2.9 DISTRIBUTED SYSTEM

A distributed system refers to physically separate computers working together. This certainly is the ideal form of a distributed system, where the "implementation detail" of building a powerful system out of many simpler systems is entirely hidden from the user. Famous Information Technology expert and writer Andrew Tannenbaum defines distributed system as "A distributed system is a collection of independent computers that appear to its users as a single coherent system."

2.9.1 Characteristics of Distributed Systems

- Cheaper and easier to build lots of simple computers
- Easier to add power incrementally
- Machines may necessarily be remote, but system should work together
- Higher availability - one computer crashes, others carry on working
- Better reliability - store data in multiple locations
- More security - each piece easier to secure to right level

2.9.2 Basic Problems and Challenges of a Distributed System

A number of challenges those are inherent in the distributed systems: Transparency, Scalability, Dependability, Performance, Flexibility, Extensibility, Openness and Interoperability.

Transparency: Transparency is the concealment from the user and the application programmer of the separation of the components of a distributed system (i.e., a single image view). Transparency is a strong property that is often difficult to achieve. There are a number of different forms of transparency including the following:

- **Access Transparency:** Local and remote resources are accessed in same way.
- **ıocation Transparency:** Users are unaware of the location of resources.
- **Migration Transparency:** Resources can migrate without name change.
- **Replication Transparency:** Users are unaware of the existence of multiple copies of resources.
- **Failure Transparency:** Users are unaware of the failure of individual components.
- **Concurrency Transparency:** Users are unaware of sharing resources with others.

Scalability: Scalability is important in distributed systems, and in particular in wide area distributed systems, and those expected to experience large growth. According to Neuman, a system is scalable if it can handle the addition of users and resources without suffering a noticeable loss of performance or increase in administrative complexity. Adding users and resources causes a system to grow. This growth has three dimensions:

Size: A distributed system can grow with regards to the number of users or resources (e.g., computers) that it supports. As the number of users grows the system may become overloaded (for example, because it must process too many user requests). Likewise, as the number of resources managed by the system grows, the administration that the system has to perform may become too overwhelming for it.

Geography: A distributed system can grow with regards to geography or the distance between nodes. An increased distance generally results in greater communication delays and the potential for communication failure. Another aspect of geographic scale is the clustering of users in a particular area. While the whole system may have enough resources to handle all users, when they are all concentrated in a single area, the resources available there may not be sufficient to handle the load.

Administration: As a distributed system grows, its various components (users, resources, nodes, networks, etc.) will start to cross administrative domains. This means that the number of organizations or individuals that exert administrative control over the system will grow. In a system that scales poorly with regards to administrative growth this can lead to problems of resource usage, reimbursement, security, etc.

Dependability: Although distributed systems provide the potential for higher availability due to replication, the distributed nature of services means that more components have to work properly for a single service to function. Hence, there are more potential points of failure and if the system architecture does not take explicit measures to increase reliability, there may actually be a degradation of availability. Dependability requires consistency, security, and fault tolerance.

Performance: Any system should strive for maximum performance, but in the case of distributed systems this is a particularly interesting challenge, since it directly conflicts with some other desirable properties. In particular, transparency, security, dependability and scalability can easily be detrimental to performance.

Flexibility: A flexible distributed system can be configured to provide exactly the services that a user or programmer needs. A system with this kind of flexibility generally provides a number of key properties.

Extensibility: Extensibility allows one to add or replace system components in order to extend or modify system functionality.

Openness: Openness means that a system provides its services according to standard rules regarding invocation syntax and semantics. Openness allows multiple implementations of standard components to be produced. This provides choice and flexibility.

Interoperability: Interoperability ensures that systems implementing the same standards (and possibly even those that do not) can interoperate.

2.10 PERSONAL COMPUTER (PC)

A personal computer comprises a Monitor (VDU), a keyboard, Disk Drive (s), printer and Central Processing Unit (CPU). The CPU of PC has a mother board with several chips mounted on a circuit board. The major components of the circuit board are Microprocessor, Random Access Memory (RAM), Read Only Memory (ROM) chips and other supporting circuits.

Microprocessor: The Microprocessor chip is like a brain of human being which contains circuits and registers to perform arithmetic, logic and control functions. i.e it contains ALU & Control Unit (CU). These chips will be able to retrieve data from the input output devices, store, manipulate and process a byte of data at a time. There is an address bus which is built into these chips to determine the storage locations (of RAM) of the data and the instructions of the program.

Over the years, different microprocessors were developed and the first in the series is INTEL 8080. The other processors are 8088, 80286, 80386, 80486, Pentium I, II, III and Pentium 4.

Microprocessors generally are categorized in terms of the maximum number of binary bits in the data they process. Over time, five standard data widths have evolved for microprocessors: 4-bit, 8-bit, 16-bit, 32-bit, 64-bit. During a single cycle of the CPU processor, a 16-bit program can send 16 bits of instructions to the CPU. A 32-bit program can send twice as many (32-bits) of instructions during a single cycle. Therefore, 32-bit programs are often able to run faster than 16-bit programs but because the speed of a program is affected by so many other factors, a 32-bit program is unlikely to be twice as fast as a 16-bit program. Table 2.2 lists some common microprocessors that are available in the market.

Note that the 8086 has data bus width of 16-bit, and it is able to address 1Megabyte of memory. It is important to note that 80286, 80386, 80486, and Pentium-Pentium4 microprocessors are upward compatible with the 8086 Architecture. This mean that 8086/8088 code will run on the 80286, 80386, 80486, and Pentium Processors, but the reverse in not true if any of the new instructions are in use. Beside to the general-purpose microprocessors, these families involve another type called special-purpose microprocessors that used in embedded control applications. This type of embedded microprocessors is called microcontroller. The 8051, 8048, 80186, 80C186XL are some examples of microcontrollers.

Table 2.2 Some common Microprocessors

Type	Data bus width	Memory size
Intel family: 8085	8	64K
8086	16	1M
80286	16	16M
80386EX, 80386DX	16 , 32	64M, 4G
80486DX4	32	4G + 16K cache
Pentium	64	4G + 16K cache
PentiumIII, Pentium-4	64	64G+32K L1 cache + 256 L2 cache
Motorola family: 6800	8	64K
68060	64	4G + 16K cache

REVIEW QUESTIONS

1. Explain the characteristics of a computer.
2. Briefly explain the demerits of digital computer.
3. Explain the classification of computers based on generation.
4. Explain the classification of computers based on performance.
5. Explain the classification of computers based on purpose.
6. Explain the difference between Shared memory multiprocessor and Distributed memory multicomputer.

CHAPTER

3

COMPUTER ARCHITECTURE

3.1 INTRODUCTION

In this chapter, we discuss the architecture of a typical computer. The study of Computer Architecture is the study of the organization and interconnection of components of computer systems. Computer architects construct computers from basic buildings blocks such as memories, arithmetic units and buses. From those building blocks the Computer Architecture can construct any number of different types of computers.

3.2 ANATOMY OF DIGITAL COMPUTERS

The block diagram of a typical digital computer is shown in Fig. 3.1. The computer system consists of three units:

- **Input device**: Reads information from input media and enters to the computer in a coded form.
- **Central Processing Unit (CPU):** Central processing Unit consists of three main units.
  - (a) **Memory unit:** Stores program and data.
  - (b) **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical functions.
  - (c) **Control Unit:** Interprets program instructions and controls the input and output devices.
- **Output device:** decodes information and presents it to the user.

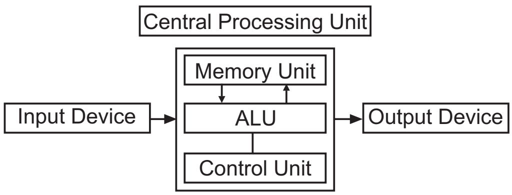


Fig. 3.1 Block Diagram of Digital Computer

3.3 COMPUTER ARCHITECTURES

The two widely accepted and popular computer architectures are Harward architecture and Von Neumann Architecture.

3.3.1 Harvard Architecture

The Harvard architecture is computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark-I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had data storage entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator; the processor could not boot itself. Block diagram representation of Harvard Architecture is shown in Fig. 3.2.

Today, most processors implement such separate signal pathways for performance reasons but actually implement Modified Harvard architecture, so they can support tasks such as loading a program from disk storage as data and then executing it.

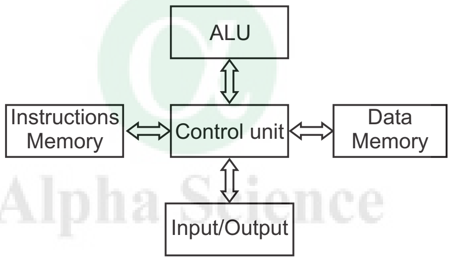


Fig. 3.2 Harvard architecture

3.3.2 Von Neumann Architecture

Von Neumann architecture is named after the mathematician and early computer scientist John Von Neumann. A brief discussion on Neumann architecture is given in this section.

In the Von Neumann Architecture, a computer consists of a Central Processing Unit (CPU), memory and Input/Output (I/O) devices and is shown in Fig. 3.3. The programs are stored in the memory and CPU fetches instructions one at a time from the memory and executes it. CPU consists of Control Unit (CU) and Arithmetic and Logic Unit (ALU).

Thus, the instructions are executed sequentially which is a slow process. Neumann machines are called control flow computer because instructions are executed sequentially as controlled by a program counter. To increase the speed, parallel processing of instructions have been developed in which serial CPU's are connected in parallel to solve a problem. Even in parallel computers, the basic building blocks are Neumann processors. The Von Neumann architecture is a design model for a stored-program digital computer that uses a processing unit and a single separate storage structure to hold both instructions and data.

Instructions and data have to be fetched in sequential order (known as the Von Neumann Bottleneck), limiting the operation bandwidth. It is mostly used to interface the external memory.

Central Processing Unit (CPU): Figure 3.3 shows the block diagram representation of Von Neumann system. The central processing unit executes the programs and controls the operation of all the hardware. Powerful computers may have several processors handling different tasks, although there will need to be one central processing unit controlling the flow of instructions and data through the subsidiary processors.

The central processing unit (CPU) is the portion of a computer system that carries out the instructions of a computer program, to perform the basic arithmetical, logical, and input/output operations of the system. The CPU plays a role somewhat analogous to the brain in the human body. The form, design and implementation of CPUs have changed dramatically since the earliest examples, but their fundamental operation remains much the same.

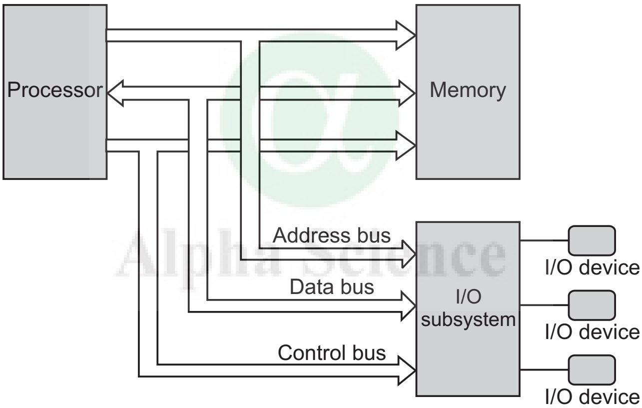


Fig. 3.3 Von Neumann System

Two typical components of a CPU are the arithmetic logic unit (ALU), which performs arithmetic and logical operations, and the control unit (CU), which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

The CPU performs the following four steps for each instruction:

1. The control unit fetches (gets) the instruction from memory.
2. The control unit decodes the instruction (decides what it means) and directs that the necessary data be moved from memory to the arithmetic/logic unit. Time taken for these first two steps together is called instruction time, or I-time.
3. The arithmetic/logic unit executes the arithmetic or logical instruction. That is, the ALU is given control and performs the actual operation on the data.

4. The arithmetic/logic unit stores the result of this operation in memory or in a register. Time taken for Steps 3 and 4 together are called execution time, or E-time.

3.4 BUS ORGANIZATION

It is a digital interconnection mechanism and it allows two or more functional units to transfer data. It can connect processor to Memory or I/O devices. A basic parallel data transfer bus organization scheme is shown in Fig. 3.4.

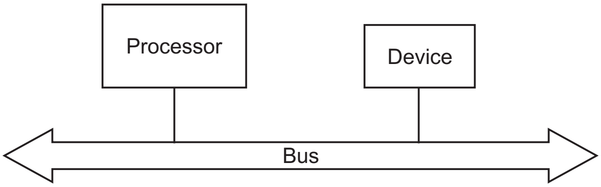


Fig. 3.4 Bus organization

Parallel data transfer

In this scheme, multiple bits can transfer at the same time. The bus is usually passive because it does not contain many electronic components and attached devices handle communication using parallel wires. Physically, bus can be connected by wires on circuit board. There are three separate wires for buses which can do separate functions and they are control bus, address bus and data bus.

The basic operations of buses are: Read (fetch) and Write (Store).

Read operation: An address is placed on the address bus and the control bus to signal *fetch* operation. The control bus places a WAIT signal till the read operation completes.

Write operation: *A*n address is placed on the address bus and keep data item on the data bus, control indicates the signal to store operation and when the operation is complete, control bus will indicate the operation complete signal.

3.5 INSTRUCTION PROCESSING

Computers store the program in memory. This means that the individual program instructions must be represented as sequences of binary digits. Since the instructions are the way in which we communicate the algorithm to the computer, they form a language. This binary language (the language of the machine) is called machine language.

A program stored in the memory provides instructions to the CPU to perform a specific action. This action can be a simple addition. It is function of the CPU to fetch the program instructions from the memory and execute them.

1. The CPU contains a number of registers to store information inside the CPU temporarily. Registers inside the CPU can be 8-bit, 16-bit, 32-bit or even 64-bit depending on the CPU.

2. The CPU also contains Arithmetic and Logic Unit (ALU). The ALU performs arithmetic (add, subtract, multiply, divide) and logic (AND, OR, NOT) functions.
3. The CPU contains a program counter also known as the Instruction Pointer to point the address of the next instruction to be executed.
4. Instruction Decoder is a kind of dictionary which is used to interpret the meaning of the instruction fetched into the CPU. Appropriate control signals are generated according to the meaning of the instruction.

Basically, each different operation, such as adding together two numbers, is assigned a binary number called the operation code (or opcode). Since we must specify what values are to be added, there is also an address part to the instruction, indicating where, in memory, the value(s) resides. As part of the design of a computer processor, the kinds of instructions to provide and the numbers for the opcodes must be chosen. Different processors use different codes, so processors' machine languages are different. This is why, when we buy software, we must be sure we are buying it for the correct processor. To make life a bit easier, processor designers (e.g., Intel, Motorola) maintain a level of consistency within processor families so all Intel Pentium chips have basically the same language, although later chips might have a larger vocabulary. This is called upward compatibility, that is, later processors can understand the machine language of earlier processors in the same family, but not necessarily vice versa.

The control unit is responsible for following the computer program and directing the other components. Since the CPU must remember some things temporarily, it contains a few special pieces of memory called registers. One of these registers (called the instruction address register—IAR), is used by the control unit to sequence through the instructions of the program and another (the instruction register—IR) is used to hold the current instruction. Typically, the arithmetic/logic unit also has a number of registers to hold intermediate results. The CPU is connected to memory by a set of wires called a bus along which data, in the form of electrical current, can flow. The control unit also has a set of control wires to each of the other components to direct them. This organization is shown in Fig. 3.5 (the memory addresses and contents are written in decimal for convenience).

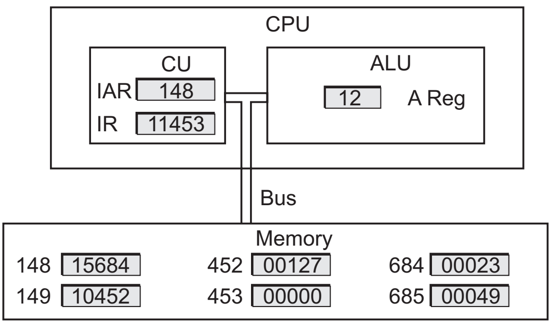


Fig. 3.5 Instruction processing

The basic process that the control unit follows is called the machine cycle. The machine cycle consists of three phases: fetch, decode, and execute, which the control unit repeats from the time the power is turned on until the system is shut down. The instruction fetch begins with the control unit consulting the IAR and directing the memory to send the contents of the indicated memory location along the bus.

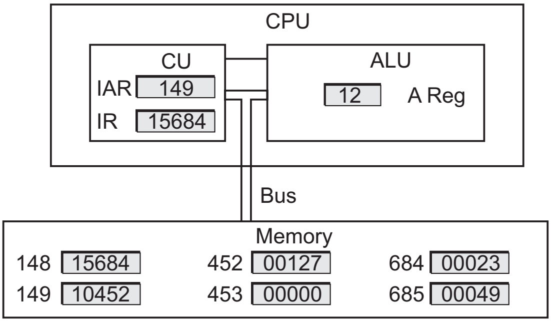


Fig. 3.6 Instruction fetch

The control unit then saves this instruction in the IR and increases the value of the IAR by one, to sequence to the next instruction. The result of the fetch is shown in Fig. 3.6 Now the control unit performs the instruction decode. Basically, within the IR, it divides the instruction up into its opcode (here we will consider the first two decimal digits) and address (the next three digits) parts. It then sends the appropriate signal to other components to tell them what to do next. In this case, let's assume that 15 is the opcode for add.

The control unit sends a signal to the ALU indicating an add operation and a signal to the memory to send the contents of the indicated address (684) along the bus. This is shown in Figure 3.7.

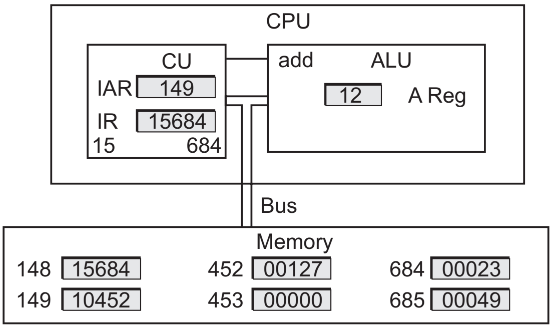


Fig. 3.7 Instruction decode

The final phase is the execute phase. Here the components perform the operations indicated by the control unit. In this case, the memory sends the content (23) of address 684 along the bus. As the ALU receives this value, it adds it to the contents of the register giving the value 35. The result is shown in Fig. 3.8. Now the cycle begins again with the fetch of the instruction at address 149.

To ensure that all of the hardware components work together (that is, know when to look for an instruction from the control unit), they are synchronized by the system clock. The system clock is a crystal that emits electrical pulses at a specific frequency. Each component counts the pulses and knows when to look for control signals.

The clock speed thus controls the timing (the speed) of the machine cycle (some specific number of pulses per cycle), and ultimately the speed of the computer itself.

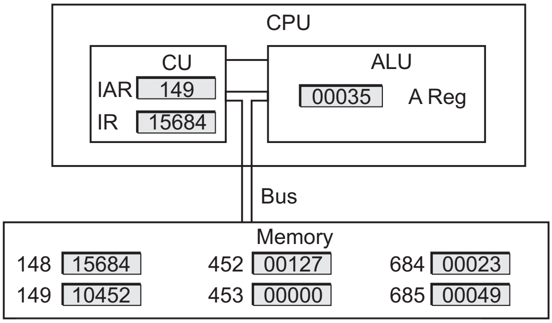


Fig. 3.8 Instruction execute

Clock speeds are measured in megahertz (MHz, million cycles per second) and so CPU speeds are described in megahertz (e.g., a 400MHz Pentium II). Actually, the clock speed is only useful in comparing chips of the same model such as two Pentium II, but it doesn't tell the entire story when comparing different models or different chip families.

3.6 MICROCOMPUTER

The integrated Circuit (IC) chip containing the CPU is called the microprocessor and the entire computer including the microprocessor, memory and I/O is called a microcomputer. The internal organization of a microcomputer is shown in Fig. 3.9.

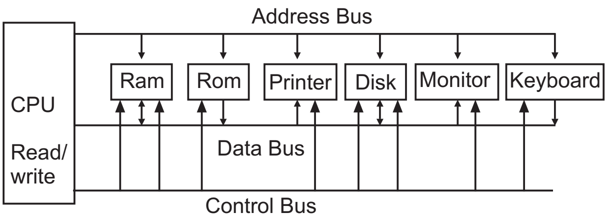


Fig. 3.9 Internal Organization of a Microcomputer

The CPU is connected to memory and I/O devices through a strip of wires called a bus. The bus inside a computer carries information from place to place. In every computer there are three types of busses:

1. **Address Bus:** The address bus is used to identify the memory location or I/O device the processor intends to communicate with. The width of the Address Bus rages from 20 bits (8086) to 36 bits for (Pentium II).

2. **Data Bus:** Data bus is used by the CPU to get data from / to send data to the memory or the I/O devices. The width of a microprocessor is used to classify the microprocessor. The size of data bus of Intel microprocessors vary between 8-bit (8085) to 64-bit (Pentium).
3. **Control Bus:** How can we tell if the address on the bus is memory address or an I/O device address? This where the control bus comes in. Each time the processor outputs an address it also activates one of the four control bus signals: Memory Read, Memory Write, I/O Read and I/O Write. The address and control bus contains output lines only, therefore it is unidirectional, but the data bus is ***bidirectional.***

3.6.1 Memory used in Microcomputers

There two types of memory used in microcomputers:

- *RAM* (**Random Access Memory/ Read-Write memory)** is used by the computer for the temporary storage of the programs that is running. Data is lost when the computer is turned off. So known as ***volatile*** memory.
- *ROM* (**Read Only Memory**) the information in ROM is permanent and not lost when the power is turned off. Therefore, it is called nonvolatile memory. Note that RAM is sometimes referred as primary storage, where magnetic/optical disks are called secondary storage.

3.7 COMPLEX INSTRUCTION SET COMPUTERS (CISC)

CISCs have instructions which are quite complex and of variable length. They use a relatively small number of registers, and are capable of accessing memory locations directly. Complex instructions are sequenced in microcode in modern CISC processors. Example for CISC is x86 processors.

One of the primary advantages of CISC system is that the compiler has to do very little work to translate a high-level language statement into assembly. Because the length of the code is relatively short, very little RAM is required to store instructions. The emphasis is put on building complex instructions directly into the hardware.

3.8 REDUCED INSTRUCTION SET COMPUTER (RISC)

RISC, or Reduced Instruction Set Computer is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures.

Certain design features have been characteristic of most RISC processors as follows:

- **One Cycle Execution Time:** RISC processors have a CPI (Clock Per Instruction) of one cycle. This is due to the optimization of each instruction on the CPU and a technique called pipelining.

- **Pipelining:** A technique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions;
- **Large Number of Registers:** The RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory.

The important observations that motivated designers to consider alternatives to CISC designs were:

- **Simple Instructions:** The designers of CISC architectures anticipated extensive use of complex instructions because they close the semantic gap. In reality, it turns out that compilers mostly ignore these instructions. Several empirical studies have shown that this is the case. One reason for this is that different high-level languages use different semantics. For example, the semantics of the C for loop is not exactly the same as that in other languages. Thus, compilers tend to synthesize the code using simpler instructions.
- **Few Data Types:** CISC tends to support a variety of data structures, from simple data types such as integers and characters to complex data structures such as records and structures. Empirical data suggest that complex data structures are used relatively infrequently. Thus, it is beneficial to design a system that supports a few simple data types efficiently and from which the missing complex data types can be synthesized.
- **Simple Addressing Modes:** CISC designs provide a large number of addressing modes. The main motivations are: (1) To support complex data structures and (2) To provide flexibility to access operands.

Features of RISC: Although CISC allows flexibility, it also introduces problems. First, it causes variable instruction execution times, depending on the location of the operands. Second, it leads to variable-length instructions. For example, the IA-32 instruction length can range from 1 to 12 bytes. Variable instruction lengths lead to inefficient instruction decoding and scheduling.

- **Identical General Purpose Registers:** Allowing any register to be used in any context, simplifying compiler design (although normally there are separate floating point registers).
- **Harvard Architecture Based:** RISC designs are also more likely to feature a Harvard memory model, where the instruction stream and the data stream are conceptually separated; this means that modifying the memory where code is held might not have any effect on the instructions executed by the processor (because the CPU has a separate instruction and data cache), at least until a special synchronization instruction is issued. On the upside, this allows both caches to be accessed simultaneously, which can often improve performance.

REVIEW QUESTIONS

1. Explain the basic units of a digital computer with the help of its block diagram.
2. Explain Harvard architecture.
3. Write a brief note on Von Neumann architecture.
4. Explain in detail about how instructions are being processed in a computer.
5. Distinguish between microcomputer and computer system.

CHAPTER

4

DATA REPRESENTATION AND BINARY ARITHMETIC

4.1 INTRODUCTION

The study of number systems is useful to the student of computing due to the fact that number systems other than the familiar decimal (base 10) number system are used in the computer field.

Digital computers internally use the binary (base 2) number system to represent data and perform arithmetic calculations. The binary number system is very efficient for computers, but not for humans. Representing even relatively small numbers with the binary system requires working with long strings of ones and zeroes.

The hexadecimal (base 16) number system (often called "hex" for short) provides us with a shorthand method of working with binary numbers. One digit in hex corresponds to four binary digits (bits), so the internal representation of one byte can be represented either by eight binary digits or two hexadecimal digits. Less commonly used is the octal (base 8) number system, where one digit in octal corresponds to three binary digits (bits).

In the event that a computer user (programmer, operator, end user, etc.) needs to examine a display of the internal representation of computer data viewing the data in a "shorthand" representation (such as hex or octal) is less tedious than viewing the data in binary representation. The binary, hexadecimal and octal number systems will be discussed in the following sections.

4.2 NUMBER SYSTEMS

The decimal number system that we are all familiar with is a positional number system. The actual number of symbols used in a positional number system depends on its base (also called the radix). The highest numerical symbol always has a value of one less than the base. The decimal number system has a base of 10, so the numeral with the highest value is 9; the octal number system has a base of 8, so the numeral with the highest value is 7, the binary number system has a base of 2, so the numeral with the highest value is 1, etc.

Any number can be represented by arranging symbols in specific positions. You know that in the decimal number system, the successive positions to the left of the decimal point represent units (ones), tens, hundreds, thousands, etc. Put another way, each position represents a specific power of base 10.

The reason for the common use of hexadecimal numbers is the relationship between the numbers 2 and 16. Sixteen is a power of 2 ($16 = 2^4$). Because of this relationship, four digits in a binary number can be represented with a single hexadecimal digit. This makes conversion between binary and hexadecimal numbers very easy, and hexadecimal can be used to write large binary numbers with much fewer digits. When working with large digital systems, such as computers, it is common to find binary numbers with 8, 16 and even 32 digits. Writing a 16 or 32 bit binary number would be quite tedious and error prone. By using hexadecimal, the numbers can be written with fewer digits and much less likelihood of error.

To convert a binary number to hexadecimal, divide it into groups of four digits starting with the rightmost digit. If the number of digits isn't a multiple of 4, prefix the number with 0's so that each group contains 4 digits. For each four digit group, convert the 4 bit binary number into an equivalent hexadecimal digit.

For example: Convert the binary number 10110101 to a hexadecimal number:

Divide into groups for 4 digits	1011 0101
Convert each group to hex digit	B 5
	$= (B5)_{16}$

To convert a hexadecimal number to a binary number, convert each hexadecimal digit into a group of 4 binary digits.

Example: Convert the hex number 374F into binary

	3 7 4 F
Convert the hex digits to binary	0011 0111 0100 1111
	$(0011011101001111)_2$

There are several ways in common use to specify that a given number is in hexadecimal representation rather than some other radix. In cases where the context makes it absolutely clear that numbers are represented in hexadecimal, no indicator is used. In much written material where the context doesn't make it clear what the radix is, the numeric subscript 16 following the hexadecimal number is used. In most programming languages, this method isn't really feasible, so there are several conventions used depending on the language. In the C and C++ languages, hexadecimal constants are represented with a '0x' preceding the number, as in: 0 × 317F, or 0 × 1234, or 0 × AF. In assembler programming languages that follow the Intel style, a hexadecimal constant begins with a numeric character (so that the assembler can distinguish it from a variable name), a leading '0' being used if necessary. The letter 'h' is then suffixed onto the number to inform the assembler that it is a hexadecimal constant. In Intel style assembler format: 371Fh and 0FABCh are valid hexadecimal constants. Note that: A37h isn't a valid hexadecimal constant. It doesn't begin with a numeric character, and so will be taken by the assembler as a variable name. In assembler programming languages that follow the Motorola style, hexadecimal constants begin with a '$' character. So in this case: $371F or $FABC or $01 are valid hexadecimal constants.

4.2.1 The Binary Number System

The same principles of positional number systems we applied to the decimal number system can be applied to the binary number system. However, the base of the binary number system is two, so each position of the binary number represents a successive power of two. From right to left, the successive positions of the binary number are weighted 1, 2, 4, 8, 16, 32, 64, etc. A list of the first several powers of 2 follows:

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$ $2^4 = 16$

$2^5 = 32$ $2^6 = 64$ $2^7 = 128$ $2^8 = 256$ $2^9 = 512$ $2^{10} = 1024$

For reference, the following table shows the decimal numbers 0 through 31 with their binary equivalents:

Decimal	Binary	Decimal	Binary
0	0	16	10000
1	1	17	10001
2	10	18	10010
3	11	19	10011
4	100	20	10100
5	101	21	10101
6	110	22	10110
7	111	23	10111
8	1000	24	11000
9	1001	25	11001
10	1010	26	11010
11	1011	27	11011
12	1100	28	11100
13	1101	29	11101
14	1110	30	11110
15	1111	31	11111

4.2.2 Converting a Binary Number to a Decimal Number

To determine the value of a binary number (1001, for example), we can expand the number using the positional weights as follows:

$$1\ 0\ 0\ 1_2$$

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

Here's another example to determine the value of the binary number 1101010_2:

$1\ 1\ 0\ 1\ 0\ 1\ 0_2$

$0 \times 2^0 = 0 \times 1 = 0$
$1 \times 2^1 = 1 \times 2 = 2$
$0 \times 2^2 = 0 \times 4 = 0$
$1 \times 2^3 = 1 \times 8 = 8$
$0 \times 2^4 = 0 \times 16 = 0$
$1 \times 2^5 = 1 \times 32 = 32$
$0 \times 2^6 = 1 \times 64 = 64$

106_{10}

4.2.3 Converting a Decimal Number to a Binary Number

To convert a decimal number to its binary equivalent, the remainder method can be used. (This method can be used to convert a decimal number into any other base.) The remainder method involves the following four steps:

(1) Divide the decimal number by the base (in the case of binary, divide by 2).

(2) Indicate the remainder to the right.

(3) Continue dividing into each quotient (and indicating the remainder) until the divide operation produces a zero quotient.

(4) The base 2 number is the numeric remainder reading from the last division to the first (if you start at the bottom, the answer will read from top to bottom).

Example 1: Convert the decimal number 99_{10} to its binary equivalent:

Answer: Step-1: Divide 2 into 99. The quotient is 49 with a remainder of 1; indicate the 1 on the right.

Step-2: 2 into 49 (the quotient from the previous division). The quotient is 24 with a remainder of 1, indicated on the right.

Step-3: Divide 2 into 24. The quotient is 12 with a remainder of 0, as indicated

Step-4: Divide 2 into 12. The quotient is 6 with a remainder of 0, as indicated.

Step-5: Divide 2 into 6. The quotient is 3 with a remainder of 0, as indicated.

Step-6: Divide 2 into 3. The quotient is 1 with a remainder of 1, as indicated.

Step-7: Divide 2 into 1. The quotient is 0 with a remainder of 1, as indicated. Since the quotient is 0, stop here.

Divisor	Dividend	Quotient	Remainder
2	1	0	1
2	3	1	1
2	6	3	0
2	12	6	0
2	24	12	0
2	49	24	1
Start Here ⇒ 2	99	49	1

The answer, reading the remainders from top to bottom, is 1100011, so 99_{10} = 1100011.

Example 2: Convert the decimal number 13_{10} to its binary equivalent:

```
                 0
           2  |  1
                 1
           2  |  3
                 3
           2  |  6
Start            6
Here ⇒     2  |  13
```

The answer, reading the remainders from top to bottom, is 1101, so $13_{10} = 1101_2$.

4.2.4 Converting Decimal Fractions to Binary

In this section, a step-by-step method for computing the binary expansion on the right-hand side of the decimal point. We will illustrate the method by converting the decimal value .625 to a binary representation.

Step 1: Begin with the decimal fraction and multiply by 2. The whole number part of the result is the first binary digit to the right of the point.

Because .625 × 2 = 1.25, the first binary digit to the right of the point is a 1. So far, we have .625 = .1??? . . . (base 2).

Step 2 : Next we disregard the whole number part of the previous result (the 1 in this case) and multiply by 2 once again. The whole number part of this new result is the second binary digit to the right of the point. We will continue this process until we get a zero as our decimal part or until we recognize an infinite repeating pattern.

Because .25 × 2 = 0.50, the second binary digit to the right of the point is a 0. So far, we have .625 = .10?? . . . (base 2).

Step 3 : Disregarding the whole number part of the previous result (this result was .50 so there actually is no whole number part to disregard in this case), we multiply by 2 once again. The whole number part of the result is now the next binary digit to the right of the point.

Because 0.50 × 2 = 1.00, the third binary digit to the right of the point is a 1. So now we have .625 = .101?? . . . (base 2).

Step 4 : In fact, we do not need a Step 4. We are finished in Step 3, because we had 0 as the fractional part of our result there.

4.2.5 Infinite Binary Fractions

The method we just explored can be used to demonstrate how some decimal fractions will produce infinite binary fraction expansions. We illustrate by using that method to see that the binary representation of the decimal fraction 1/10 is, in fact, infinite.

Recall our step-by-step process for performing this conversion.

Step 1: Begin with the decimal fraction and multiply by 2. The whole number part of the result is the first binary digit to the right of the point.

Because .1 × 2 = 0.2, the first binary digit to the right of the point is a 0. So far, we have .1 (decimal) = .0??? . . . (base 2).

Step 2: Next we disregard the whole number part of the previous result (0 in this case) and multiply by 2 once again. The whole number part of this new result is the *second* binary digit to the right of the point. We will continue this process until we get a zero as our decimal part or until we recognize an infinite repeating pattern.

Because .2 × 2 = 0.4, the second binary digit to the right of the point is also a 0. So far, we have .1 (decimal) = .00?? . . . (base 2).

Step 3: Disregarding the whole number part of the previous result (again a 0), we multiply by 2 once again. The whole number part of the result is now the next binary digit to the right of the point.

Because .4 × 2 = 0.8, the third binary digit to the right of the point is also a 0. So now we have .1 (decimal) = .000?? . . . (base 2).

Step 4: We multiply by 2 once again, disregarding the whole number part of the previous result (again a 0 in this case).

Because .8 × 2 = 1.6, the fourth binary digit to the right of the point is a 1. So now we have .1 (decimal) = .0001?? . . . (base 2).

Step 5: We multiply by 2 once again, disregarding the whole number part of the previous result (a 1 in this case).

Because .6 × 2 = 1.2, the fifth binary digit to the right of the point is a 1. So now we have .1 (decimal) = .00011?? . . . (base 2).

Step 6: We multiply by 2 once again, disregarding the whole number part of the previous result. Let's make an important observation here. Notice that this next step to be performed (multiply 2. × 2) is exactly the same action we had in step 2. We are then bound to repeat steps 2-5, then return to Step 2 again indefinitely. In other words, we will never get a 0 as the decimal fraction part of our result. Instead we will just cycle through steps 2-5 forever. This means we will obtain the sequence of digits generated in steps 2-5, namely 0011, over and over. Hence, the final binary representation will be.

.1 (decimal) = .00011001100110011 . . . (base 2).

The repeating pattern is more obvious if we highlight it in color as below:

.1 (decimal) = .00011001100110011 . . . (base 2).

4.2.6 Negative Binary Numbers

So far, when we have been looking at binary numbers, we have treated the 1's and 0's as numbers or digits. However, when we include negative numbers in this system, we have to take a slightly different view of binary numbers. We will now treat these 1's and 0's as codes for numbers, where the left most digit will be the sign digit, indicating whether the number is positive or negative. If the left most digit is 0 the number is positive, and if the left most digit is 1 the number is negative.

We will now use this positive/negative convention and apply it to two different ways of representing negative numbers - ones compliment and twos compliment.

4.2.7 One's Compliment

The one's compliment is found by changing the zeros for ones and ones for zeros when forming the negative number. The following is a list of the numbers in ones compliment that can be formed using only four digits (remember that the left most digit is the sign digit).

Decimal Number	Binary	Decimal Number	Binary (1s complement of decimal positive numbers)
0	0000	0	1111
1	0001	–1	1110
2	0010	–2	1101
3	0011	–3	1100
4	0100	–4	1011
5	0101	–5	1010
6	0110	–6	1001
7	0111	–7	1000

4.2.8 Two's Compliment

The two's compliment is obtained by forming the ones compliment first and then adding one.

e.g. $+3 = (0011)_2$

one's compliment: 1100

add 1 : 1101 = –3 in two's complement form.

To go back to +3: –3 = 1101

One's complement : 0010

Add 1 : 0011 = +3 in two's complement.

The following table gives a list of numbers in two's compliment that can be formed using only four digits (remember that the left digit is the sign digit). Once you know how to find negative numbers, subtraction follows quite nicely.

0	0000	0	0000
1	0001	–1	1111
2	0010	–2	1110
3	0011	–3	1101
4	0100	–4	1100
5	0101	–5	1011
6	0110	–6	1010
7	0111	–7	1001

4.2.9 Binary Addition

Adding two binary numbers together is easy, keeping in mind the following four addition rules:

(1) 0 + 0 = 0

(2) 0 + 1 = 1

(3) 1 + 0 = 1

(4) 1 + 1 = 10

After the first two binary counting numbers, 0 and 1, all of the binary digits are used up. In the decimal system, we used up all the digits after the tenth counting number, 9. The same method is used in both systems to come up with the next number: place a zero in the "ones" position and start over again with one in the next position on the left. In the decimal system, this gives ten, or 10. In binary, it gives 10_2, which is read "one-zero, base two."

Consider the following binary addition problems and note where it is necessary to carry the 1:

```
                                1        1 1        1 1 1 1
  1 1 0     1 1     1 0 0      1 1     1 0 1 0     1 1 1 1 1
+ 1 0 1   + 1 0   + 1 0 1    + 0 1   + 1 1 1 1   + 0 1 0 1 1
-------   -----   -------    -----   ---------   -----------
  1 1 1   1 0 1   1 0 0 1    1 0 0   1 0 0 0 1   1 0 1 0 1 0
```

4.2.10 Subtraction Using Complements

Subtraction in any number system can be accomplished through the use of complements. A complement is a number that is used to represent the negative of a given number.

When two numbers are to be subtracted, the subtrahend can either be subtracted directly from the minuend (as we are used to doing in decimal subtraction) or, the complement of the subtrahend can be added to the minuend to obtain the difference. When the lat ter method is used, the addition will produce a high-order (leftmost) one in the result (a "carry"), which must be dropped. This is how the computer performs subtraction: it is very efficient for the computer to use the same "add" circuitry to do both addition and subtraction; thus, when the computer "subtracts", it is really adding the complement of the subtrahend to the minuend.

The ten's complement of any decimal number may be formed by subtracting each digit of the number from 9, then adding 1 to the least significant digit of the number formed.

4.2.11 Binary Subtraction

We will use the complement method to perform subtraction in binary and in the sections on octal and hexadecimal that follow. The use of complemented binary numbers makes it possible for the computer to add or subtract numbers using only circuitry for addition - the computer performs the subtraction of A – B by adding A + (two's complement of B) and then dropping the carried 1.

The steps for subtracting two binary numbers are as follows:

(1) Compute the one's complement of the subtrahend by subtracting each digit of the subtrahend by 1. A shortcut for doing this is to simply reverse each digit of the subtrahend - the 1's become 0's and the 0's become 1's.

(2) Add 1 to the one's complement of the subtrahend to get the two's complement of the subtrahend.

(3) Add the two's complement of the subtrahend to the minuend and drop the high-order 1.

Example 1: Compute $11010101_2 - 1001011_2$

(1) Compute the one's complement of 1001011_2 by subtracting each digit from 1 (note that a leading zero was added to the 7-digit subtrahend to make it the same size as the 8-digit minuend):

1	1	1	1	1	1	1	1	
	– 0	– 1	– 0	– 0	– 1	– 0	– 1	– 1
1	0	1	1	0	1	0	0	

(2) Add 1 to the one's complement of the subtrahend, giving the two's complement of the subtrahend:

1	1	1	1	1	1	1	1
		+	0				
1	0	1	1	0	1	0	1

(3) Add the two's complement of the subtrahend to the minuend and drop the high-order 1, giving the difference:

1	1	1		1		1			
1	1	0	1	0	1	0	1		
	+	1	1	0	1	0	1	0	1
	+	1	0	0	0	1	0	1	0

So $11010101_2 - 1001011_2 = 10001010_2$.

The answer can be checked by making sure that $1001011_2 + 10001010_2 = 11010101_2$.

4.2.12 Multiplication of binary numbers

Similar to addition of binary numbers, there are four cases that we need to look at when multiplying binary numbers.

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Example problem: 1 Multiply 101 by 11

```
    1 0 1
    x 1 1
  -------
    1 0 1
  1 0 1
  -------
  1 1 1 1
```

4.2.13 Binary division

We shall demonstrate binary division with the help of an example as follows:

```
      11
 11 | 1011
      -11
       101
       -11
        10 <—— Remainder
```

To check our answer, we first multiply our divisor 11 by our quotient 11. Then we add its' product to the remainder 10, and compare it to our dividend of 1011.

$$11 \times 11 = 1001 \longleftarrow \text{product of 11 and 11}$$
$$1001 + 10 = 1011 \longleftarrow \text{sum of product and remainder}$$

The sum is equal to our initial dividend, therefore our solution is correct.

4.3 THE OCTAL NUMBER SYSTEM

The same principles of positional number systems we applied to the decimal and binary number systems can be applied to the octal number system. However, the base of the octal number system is eight, so each position of the octal number represents a successive power of eight. From right to left, the successive positions of the octal number are weighted 1, 8, 64, 512, etc. A list of the first several powers of 8 follows:

$8^0 = 1$ $8^1 = 8$ $8^2 = 64$ $8^3 = 512$ $8^4 = 4096$ $8^5 = 32768$

4.3.1 Converting an Octal Number to a Decimal Number

To determine the value of an octal number (367_8, for example), we can expand the number using the positional weights as follows:

$3\ 6\ 7_8$

$$7 \times 8^0 = 7 \times 1 = 7$$
$$6 \times 8^1 = 6 \times 8 = 48$$
$$3 \times 8^2 = 3 \times 64 = 192$$

$$247_{10}$$

4.4 THE HEXADECIMAL NUMBER SYSTEM

The hexadecimal (base 16) number system is a positional number system as are the decimal number system and the binary number system. Recall that in any positional number system, regardless of the base, the highest numerical symbol always has a value of one less than the base. Furthermore, one and only one symbol must ever be used to represent a value in any position of the number.

For number systems with a base of 10 or less, a combination of Arabic numerals can be used to represent any value in that number system. The decimal number system uses the Arabic numerals 0 through 9; the binary number system uses the Arabic numerals 0 and 1; the octal number system uses the Arabic numerals 0 through 7; and any other number system with a base less than 10 would use the Arabic numerals from 0 to one less than the base of that number system.

However, if the base of the number system is greater than 10, more than 10 symbols are needed to represent all of the possible positional values in that number system. The hexadecimal number system uses not only the Arabic numerals 0 through 9, but also uses the letters A, B, C, D, E and F to represent the equivalent of 10_{10} through 15_{10}, respectively.

The same principles of positional number systems we applied to the decimal, binary, and octal number systems can be applied to the hexadecimal number system. However, the base of the hexadecimal number system is 16, so each position of the hexadecimal number represents a successive power of 16. From right to left, the successive positions of the hexadecimal number are weighted 1, 16, 256, 4096, 65536, etc.:

$16^0 = 1$ $16^1 = 16$ $16^2 = 256$ $16^3 = 4096$ $16^4 = 65536$

4.5 BINARY CODED DECIMALS

Another way to represent decimals in binary form is binary coded decimals (BCD's). BCD's are binary numbers that code each decimal digit separately as a four digit binary number.

e.g. 95 = 1001. 0101

Therefore BCD equivalent of 95 is 10010101 (BCD)

The advantage to using BCD's is the size of the number is unlimited with perfect accuracy. However, BCD's have several disadvantages. They use more memory since each decimal digit requires four binary digits and arithmetic operations are difficult to program and significantly slower.

4.6 FLOATING POINT REPRESENTATION OF BINARY NUMBERS

Another form that computers can use to represent numbers (remember that computers have to use some form of binary numbers) is to use scientific notation. This allows all numbers to be represented using the same basic form, with unlimited size. However, there is a slight disadvantage that all numbers cannot be represented with total accuracy. The following example shows how a number can be stored using this representation.

e.g. $104 = 1.04 \times 10^2$ (base 10)

$01101000 = .1101 \times 2^{0111}$ (base 2)

.1101 is called mantissa and 2^{0111} is called Exponent.

Computer Representation:

Exponent	Mantissa
0000 0111	0110100 00000000 00000000
sign digit	sign digit

Note that the exponent and mantissa are represented using two's compliment

4.7 CODES

When we wish to send information over long distances unambiguously it becomes necessary to modify (encoding) the information into some form before sending, and convert (decode) at the receiving end to get back the original information. This process of encoding and decoding is necessary because the channel through which the information is sent may distort the transmitted information. Much of the information is sent as numbers. While these numbers are created using simple weighted-positional numbering systems, they need to be encoded before transmission. The modifications to numbers were based on changing the weights, but predominantly on some form of binary encoding. There are several codes in use in the context of present day information technology, and more and more new codes are being generated to meet the new demands.

Coding is the process of altering the characteristics of information to make it more suitable for intended application. By assigning each item of information a unique combination of 1s and 0s we transform some given information into binary coded form. The bit combinations are referred to as "words" or "code words". In the field of digital systems and computers different bit combinations have different designations.

Bit - a binary digit 0 or 1

Nibble - a group of four bits

Byte - a group of eight bits

Word - a group of sixteen bits; a word has two bytes or four nibbles

Sometimes 'word' is used to designate a larger group of bits also, for example 32 bit or 64 bit words.

We need and use coding of information for a variety of reasons:

- to increase efficiency of transmission,
- to make it error free,
- to enable us to correct it if errors occurred,
- to inform the sender if an error occurred in the received information etc.
- for security reasons to limit the accessibility of information
- to standardise a universal code that can be used by all

Coding schemes have to be designed to suit the security requirements and the complexity of the medium over which information is transmitted. Decoding is the process of reconstructing source information from the encoded information. Decoding process can be more complex than coding if we do not have prior knowledge of coding schemes.

In view of the modern day requirements of efficient, error free and secure information transmission coding theory is an extremely important subject. However, at this stage of learning digital systems we confine ourselves to familiarising with a few commonly used codes and their properties.

We will be mainly concerned with binary codes. In binary coding we use binary digits or bits (0 and 1) to code the elements of an information set. Let n be the number of bits in the code word and x be the number of unique words.

If $n = 1$, then $x = 2$ (0, 1)

$n = 2$, then $x = 4$ (00, 01, 10, 11)

$n = 3$, then $x = 8$ (000,001,010 ...111)

. . . .

$n = j$, then $x = 2^j$

From this we can conclude that if we are given elements of information to code into binary coded format,

$x \leq 2^j$ or $j \geq \log_2 X \geq 3.32 \log_{10} X$

where j is the number of bits in a code word.

For example, if we want to code alphanumeric information (26 alphabetic characters + 10 decimals digits = 36 elements of information),

we require $j \geq 3.32 \log_{10} 36$ $j \geq 5.16$ bits

Since bits are not defined as fractional parts, we take $j = 6$. In other words a minimum six-bit code would be required to code 36 alphanumeric elements of information. However, with a 6-bit code only 36 code words are used out of the 64 code words possible. In this section we consider a few commonly used codes including:

1. Binary coded decimal codes
2. Unit distance codes
3. Error detection codes
4. Alphanumeric codes

4.7.1 Binary Coded Decimal Codes

The main motivation for binary number system is that there are only two elements in the binary set, namely 0 and 1. While it is advantageous to perform all computations on hardware in binary forms, human beings still prefer to work with decimal numbers. Any electronic system should then be able to accept decimal numbers, and make its output available in the decimal form. One method, therefore, would be to:

- convert decimal number inputs into binary form
- manipulate these binary numbers as per the required functions, and
- convert the resultant binary numbers into the decimal form

However, this kind of conversion requires more hardware, and in some cases considerably slows down the system. Faster systems can afford the additional circuitry, but the delays associated with the conversions would not be acceptable. In case of smaller systems, the speed may not be the main criterion, but the additional circuitry may make the system more expensive.

We can solve this problem by encoding decimal numbers as binary strings, and use them for subsequent manipulations. There are ten different symbols in the decimal number system: 0, 1, 2, . . ., 9. As there are ten symbols we require at least 4-bits to represent them in the binary form. Such a representation of decimal numbers is called binary coding of decimal numbers.

As four bits are required to encode one decimal digit, there are sixteen 4-bit groups to select ten groups. This would lead to nearly 30 × 1010 (16C10.10!) possible codes. However, most of them will not have any special properties that would be useful in hardware design. We wish to choose codes that have some desirable properties like:

- ease of coding
- ease in arithmetic operations
- minimum use of hardware
- error detection property
- ability to prevent wrong output during transitions

In a weighted code the decimal value of a code is the algebraic sum of the weights of 1s appearing in the number. Let $(A)_{10}$ be a decimal number encoded in the binary form as $a_3\, a_2\, a_1\, a_0$. Then:

$$(A)_{10} = w_3a_3 + w_2a_2 + w_1a_1 + w_0a_0$$

where w_3, w_2, w_1 and w_0 are the weights selected for a given code, and a_3, a_2, a_1 and a_0 are either 0*s* or 1*s*. The more popularly used codes have the weights as:

w_3	w_2	w_1	w_0
8	4	2	1
2	4	2	1
8	4	–2	–1

The decimal numbers in these three codes are:

Decimal digial	Weights 8 4 2 1	Weights 2 4 2 1	Weights 8 4 —2 —1
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 1 1
2	0 0 1 0	0 0 1 0	0 1 1 0
3	0 0 1 1	0 0 1 1	0 1 0 1
4	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1	1 0 1 1
6	0 1 1 0	1 1 0 0	1 0 1 0
7	0 1 1 1	1 1 0 1	1 0 0 1
8	1 0 0 0	1 1 1 0	1 0 0 0
9	1 0 0 1	1 1 1 1	1 1 1 1

In all the cases only ten combinations are utilized to represent the decimal digits. The remaining six combinations are illegal. However, they may be utilized for error detection purposes.

Consider, for example, the representation of the decimal number 16.85 in Natural Binary Coded Decimal code (NBCD)

$$(16.85)10 = (0001\ \ 0110\ .\ 1000\ \ 0101)_{NBCD}$$

1 6 8 5

There are many possible weights to write a number in BCD code. Some codes have desirable properties, which make them suitable for specific applications. Two such desirable properties are:

1. Self-complementing codes
2. Reflective codes

When we perform arithmetic operations, it is often required to take the "complement" of a given number. If the logical complement of a coded number is also its arithmetic complement, it will be convenient from hardware point of view. In a self-complementing coded decimal number, $(A)_{10}$, if the individual bits of a number are complemented it will result in $(9 - A)_{10}$.

Example: Consider the 2421 code.

- The 2421 code of $(4)_{10}$ is 0100.
- Its complement is 1011 which is 2421 code for $(5)_{10} = (9 – 4)_{10}$.

Therefore, 2421 code may be considered as a self-complementing code. A necessary condition for a self-complimenting code is that the sum of its weights should be 9. A self-complementing code, which is not weighted, is excess-3 code. It is derived from 8421 code by adding 0011 to all the 8421 coded numbers. Another self-complementing code is 631-1 weighted code. Three self-complementing codes are shown in the table below.

Decimal digial	Excess-3 Code	631—1 Code	2421 Code
0	0011	0011	0000
1	0100	0010	0001
2	0101	0101	0010
3	0110	0111	0011
4	0111	0110	0100
5	1000	1001	1011
6	1001	1000	1100
7	1010	1010	1101
8	1011	1101	1110
9	1100	1100	1111

A reflective code is characterized by the fact that it is imaged about the centre entries with one bit changed. For example, the 9's complement of a reflected BCD code word is formed by changing only one its bits. Two such examples of reflective BCD codes are:

Desimal	Code - A	Code - B
0	0000	0100
1	0001	1010
2	0010	1000
3	0011	1110
4	0100	0000
5	1100	0001
6	1011	1111
7	1010	1001
9	1000	0101

The BCD codes are widely used and the reader should become familiar with reasons for using them and their application. The most common application of NBCD codes is in the calculator.

4.7.2 Unit Distance Codes

There are many applications in which it is desirable to have a code in which the adjacent codes differ only in one bit. Such codes are called Unit distance Codes. "Gray code" is the most popular example of unit distance code. The 3-bit and 4-bit Gray codes are:

Decimal	3-Bit Gray	4-Bit Gray
0	000	0000
1	001	0001
2	011	0011
3	010	0010
4	110	0110
5	111	0111
6	101	0101
7	100	0100
8	-	1100
9	-	1101
10	-	1111
11	-	1110
12	-	1010
13	-	1001
15	-	1000

These Gray codes listed here have also the reflective properties. Some additional examples of unit distance codes are:

Decimal digial	UDC-1	UDC-2	UDC-3
0	0000	0000	0000
1	01000	0001	1000
2	1100	0011	1001
3	1000	0010	0001
4	1001	0110	0011
5	1011	11110	0111
6	1111	1111	1111
7	0111	1101	1011
8	0011	1100	1010
9	0001	1000	0010

The most popular use of Gray codes is in the position sensing transducer known as shaft encoder. A shaft encoder consists of a disk in which concentric circles have alternate sectors with reflective surfaces while the other sectors have non-reflective surfaces. The position is sensed by the reflected light from a light emitting diode.

However, there is choice in arranging the reflective and non-reflective sectors. A 3-bit binary coded disk will be as shown in the Fig. 4.1.

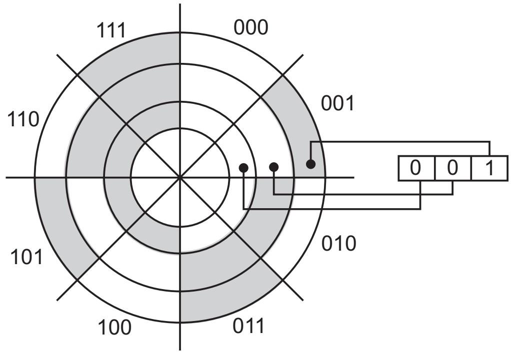


Fig. 4.1 3-Bit binary coded shaft encoder

From this figure we see that straight binary code can lead to errors because of mechanical imperfections. When the code is transiting from 001 to 010, a slight misalignment can cause a transient code of 011 to appear. The electronic circuitry associated with the encoder will receive 001 --> 011 -> 010. If the disk is patterned to give Gray code output, the possibilities of wrong transient codes will not arise. This is because the adjacent codes will differ in only one bit. For example the adjacent code for 001 is 011. Even if there is a mechanical imperfection, the transient code will be either 001 or 011. The shaft encoder using 3-bit Gray code is shown in the Fig. 4.2.

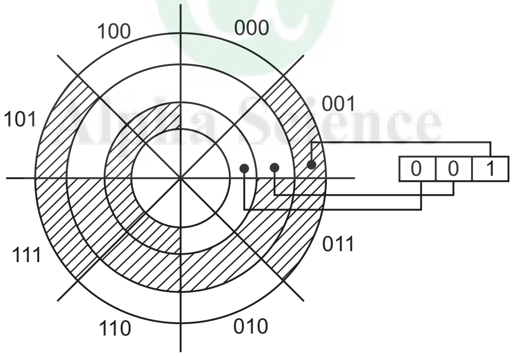


Fig. 4.2 Shaft encoder disk using a 3-bit Gray code

There are two convenient methods to construct Gray code with any number of desired bits. The first method is based on the fact that Gray code is also a reflective code. The following rule may be used to construct Gray code:

- A 1-bit Gray code had code words, 0 and 1.
- The first 2^n code words of an $(n + 1)$-bit Gray code equal the code words of an n-bit Gray code, written in order with a leading 0 appended.
- The last 2^n code words of a $(n + 1)$-bit Gray code equal the code words of an n-bit Gray code, written in reverse order with a leading 1 appended.

However, this method requires Gray codes with all bit lengths less than '*n*' also be generated as a part of generating n-bit Gray code. The second method allows us to derive an n-bit Gray code word directly from the corresponding n-bit binary code word:

- The bits of an n-bit binary code or Gray code words are numbered from right to left, from 0 to n-1.
- Bit i of a Gray-code word is 0 if bits i and $i + 1$ of the corresponding binary code word are the same, else bit i is 1. When $i + 1 = n$, bit n of the binary code word is considered to be 0.

4.7.3 Alphanumeric Codes

When information to be encoded includes entities other than numerical values, an expanded code is required. For example, alphabetic characters (A, B, ...Z) and special operation symbols like +, –, / , *, (,) and other special symbols are used in digital systems. Codes that include alphabetic characters are commonly referred to as Alphanumeric Codes. However, we require adequate number of bits to encode all the characters. As there was a need for alphanumeric codes in a wide variety of applications in the early era of computers, like teletype, punched tape and punched cards, there has always been a need for evolving a standard for these codes.

Alphanumeric keyboard has become ubiquitous with the popularization of personal computers and notebook computers. These keyboards use ASCII (American Standard Code for Information Interchange) code:

b4	b3	b2	b1	b7 b6 b5							
				000	001	010	011	100	101	110	111
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	“	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	,	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N		n	~
1	1	1	1	SI	US	/	?	O	-	o	DEL

Alphanumeric codes like EBCDIC (Extended Binary Coded Decimal Interchange Code) and 12-bit Hollerith code are in use for some applications. However, ASCII code is now the standard code for most data communication networks. Therefore, the reader is urged to become familiar with the ASCII code.

4.8 ERROR DETECTION AND CORRECTING CODES

When data is transmitted in digital form from one place to another through a transmission channel/medium, some data bits may be lost or modified. This loss of data integrity occurs due to a variety of electrical phenomena in the transmission channel. As there are needs to transmit millions of bits per second, the data integrity should be very high. The error rate cannot be reduced to zero. Then we would like to ideally have a mechanism of correcting the errors that occur. If this is not possible or proves to be expensive, we would like to know if an error occurred.

If an occurrence of error is known, appropriate action, like retransmitting the data, can be taken. One of the methods of improving data integrity is to encode the data in a suitable manner. This encoding may be done for error correction or merely for error detection.

A simple process of adding a special code bit to a data word can improve its integrity. This extra bit will allow detection of a single error in a given code word in which it is used, and is called the 'Parity Bit'. This parity bit can be added on an odd or even basis. The odd or even designation of a code word may be determined by actual number of 1's in the data (including the added parity bit) to which the parity bit is added. For example, the S in ASCII code is:

$(S) = (1010011)_{\text{ASCII}}$

S, when coded for odd parity, would be shown as

$(S) = (11010011)_{\text{ASCII with odd parity}}$

In this encoded 'S' the number of 1's is five, which is odd.

When S is encoded for even parity

$(S) = (01010011)_{\text{ASCII with even parity.}}$

In this case the coded word has even number (four) of ones. Thus the parity encoding scheme is a simple one and requires only one extra bit. If the system is using even parity and we find odd number of ones in the received data word we know that an error has occurred. However, this scheme is meaningful only for single errors. If two bits in a data word were received incorrectly the parity bit scheme will not detect the faults. Then the question arises as to the level of improvement in the data integrity if occurrence of only one bit error is detectable.

The improvement in the reliability can be mathematically determined. Adding a parity bit allows us only to detect the presence of one bit error in a group of bits. But it does not enable us to exactly locate the bit that changed. Therefore, addition of one parity bit may be called an error detecting coding scheme. In a digital system detection of error alone is not sufficient. It has to be corrected as well. Parity bit scheme can be extended to locate the faulty bit in a block of information. The information bits are conceptually arranged in a two-dimensional array, and parity bits are provided to check both the rows and the columns.

If we can identify the code word that has an error with the parity bit, and the column in which that error occurs by a way of change in the column parity bit, we can both detect and correct the wrong bit of information. Hence such a scheme is single error detecting and single error correcting coding scheme.

This method of using parity bits can be generalized for detecting and correcting more than 1-bit error. Such codes are called parity-check block codes. In this class known as (n, k) codes, $r(= n - k)$ parity check bits, formed by linear operations on the k data bits, are appended to each block of k bits to generate an n-bit code word.

An encoder outputs a unique n-bit code word for each of the 2^k possible input k-bit blocks. For example a (15, 11) code has $r = 4$ parity-check bits for every 11 data bits. As r increases it should be possible to correct more and more errors. With $r = 1$ error correction is not possible, as such a code will only detect an odd number of errors.

It can also be established that as k increases the overall probability of error should also decrease. Long codes with a relatively large number of parity-check bits should thus provide better performance. Consider the case of (7, 3) code:

Data bits			Code words						
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	1	1
0	1	0	0	1	0	0	1	1	0
0	1	1	0	1	1	1	0	0	1
1	0	0	1	0	0	1	1	0	0
1	0	1	1	0	1	0	0	1	1
1	1	0	1	1	0	1	0	1	0
1	1	1	1	1	1	0	1	0	1

A close look at these indicates that they differ in at least three positions. Any one error should then be correctable since the resultant code word will still be closer to the correct one, in the sense of the number of bit positions in which they agree, than to any other. This is an example of single-error-correcting-code. The difference in the number of positions between any two code words is called the Hamming distance, named after R.W. Hamming who, in 1950, described a general method for constructing codes with a minimum distance of 3. The Hamming distance plays a key role in assessing the error-correcting capability of codes. For two errors to be correctable, the Hamming distance d should be at least 5. In general, for t errors to be correctable, $d \geq 2t + 1$ or $t = [(d - 1)/2]$, where the $[x]$ notation refers to the integer less than or equal to x.

Innumerable varieties of codes exist, with different properties. There are various types of codes for correcting independently occurring errors, for correcting burst errors, for providing relatively error-free synchronization of binary data etc.

4.9 DIGITAL SYSTEMS

The quantities that we deal in digital electronics take only two values viz 0 and 1. The algebra that is linked with such quantities is called the algebra of two values. These values may themselves represent the state of conduction – either ON or OFF of an electronic component. The symbol “1”can be used to represent the conduction state and “0” to represent the state of non conduction. The switch in Fig. 4.3 (a) represents non-conduction and hence the value of A is 0. The value of B is thus 1 in Fig. 4.3 (b).

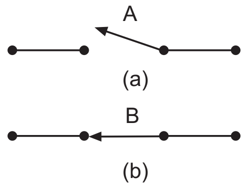


Fig. 4.3 Switch open (indicates logic 0) (b) Switch close (indicates logic 1)

In digital systems, the two states (or values) may be represented by the different voltage levels (+5V and nearly zero). The +5V may be represented by “1” and nearly zero by “0”. It is this logical 1 or 0 that are the values taken as inputs and outputs of gates. Gates may be constructed out of diodes, transistors, or IC’s.

A gate is a device that controls the information which is usually in the binary form. In Fig. 4.4, if switch A AND switch B are both closed the lamp is turned ON. This therefore constitutes something like an AND circuit or AND gate. In Fig. 4.5, the lamp is ON if switch A or switch B OR both are closed and this is called the OR gate. The switches may be closed by some means say an electrical signal.

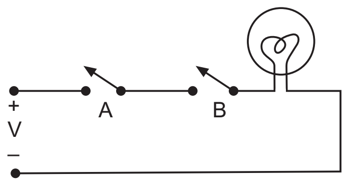


Fig. 4.4 AND equivalent (Switches in series)

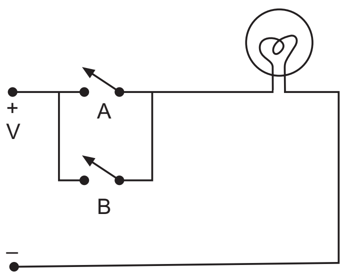


Fig. 4.5 OR equivalent (Switches in parallel)

4.10 SYMBOLS AND TRUTH TABLES OF LOGIC GATES

Each basic logic gate operation is indicated by a symbol and the function by what is called a truth table. The truth table takes into account all possible combinations (or values) for the concerned variable and gives the corresponding outputs. As the number of input increases, the truth table will be bigger as it should indicate all the possibilities. A three input gate has 8 possibilities whereas a 4 input gate has $2^4 = 16$ possibilities. In general, if the there are '*n*' inputs, and then there will be 2^n possibilities.

4.10.1 AND Gate

A AND B operation can be represented as A.B, the dot indicating the AND operation. There could be more inputs when the output is the AND-ed value of all these inputs. Thus J=A.B.C.D., (etc.). The output, J will be zero, if even one of the input variables is zero. The symbol of AND and its truth table is given in Fig. 4.6 and Table-4-1 respectively.

Table 4.1

A	B	J = A.B
0	0	0
0	1	0
1	0	0
1	1	1

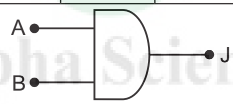


Fig. 4.6 AND gate symbol

4.10.2 OR Gate

A OR B operation can be represented as A + B. There could be more inputs when the output is the OR-ed value of all these inputs. Thus *J* = A.B.C.D., (etc). The output, *J* will be at logic ONE, if any one of the input variables is zero. The symbol of AND and its truth table is given in Fig. 4.7 and Table 4-2 respectively.

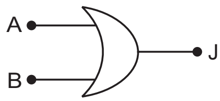


Fig. 4.7 OR gate Symbol

Table 4.2

A	B	J = A + B
0	0	0
0	1	1
1	0	1
1	1	1

4.10.3 NOT Gate

In this gate, the output is the complement (opposite) of the input. Thus if the input is high (1), the output is low (0) and vice versa. The logic symbol and the truth table are all shown in Fig. 12.6 and Table 12-3 respectively.

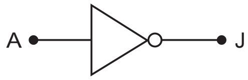


Fig. 4.8 NOT gate symbol

Table 4.3

A	$J = \bar{A}$
0	1
1	0

When A is low, the transistor is cut off and *J* remains high. On the other hand, when A is high, the transistor conducts forcing *J* to go low.

J = Compliment of $A = \bar{A}$ or A'

The NOT gate is also called the inverter circuit. If two such inverters are cascaded, we have a non-inverting circuit, the symbol of which is shown in Fig. 4.9. Non-inverting circuits are used in buffering (isolating) two circuits.

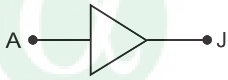


Fig. 4.9 Buffer

4.10.4 Nand Gate – The Universal Building Block

Consider the logic circuit shown in Fig. 4.10. The output of the AND gate is inverted to form what is called the NAND operation. The symbol of the NAND gate is shown in Fig. 4.11 and the truth table is given in Table 4-4.

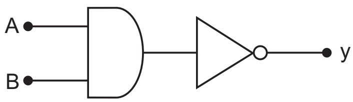


Fig. 4.10 NAND equivalent circuit

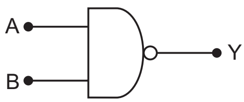


Fig. 4.11 NAND symbol

Table 4.4 Truth Table of NAND gate

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

The interesting property of this gate is that it can be used for building other basic logic gates. This is shown in Fig 4.12. Since the NAND gate is widely used as the unit for building digital systems, it is called the universal building block.

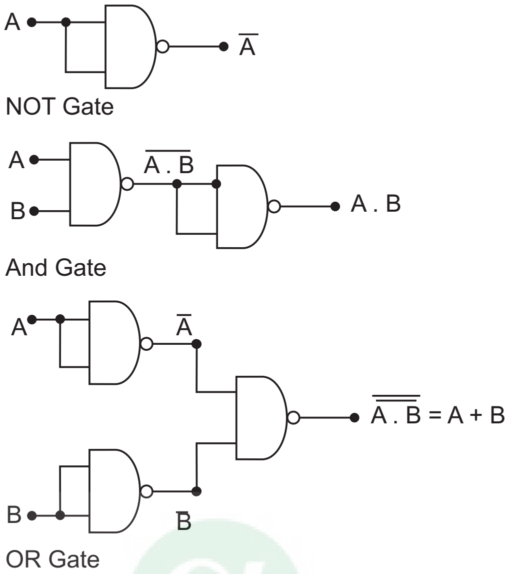


Fig. 4.12 Basic logic gates implementation using NAND

4.10.5 NOR Gate (Universal Gate)

NOR gate is complement of OR gate and its symbol is shown in Fig. 4.13. Its truth table is shown in Table 4-5. It is also a universal building block.

Table 4.5 Truth Table of NOR gate

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

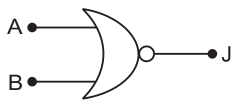


Fig. 4.13 NOR gate symbol

4.10.6 The Exclusive-OR (EX-OR or XOR) Gate

XOR gives a high output only if the two inputs are different. If there are two inputs, then the XOR gives high output if any one of the input is high. If there are three or more inputs, the XOR gate gives a high output if the number of high input is odd. The symbol of XOR gate is shown in Fig. 4.14.

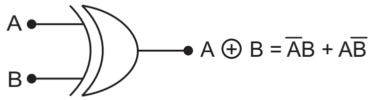


Fig. 4.14 XOR gate

Table 4.6 XOR Truth Table

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

The output equation is

$Y = A \oplus B = A \,.\, \overline{B} + \overline{A} \,.\, B$

Which is Y = A XOR B

The XOR gate can be implemented as given in Fig. 4.15.

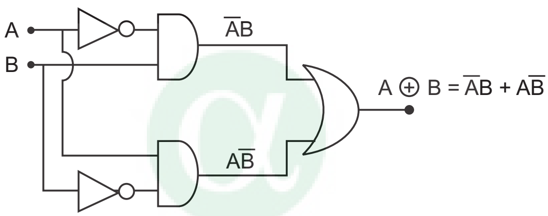


Fig. 4.15 XOR gate implementation using AND and OR gates

The truth table of XOR gate when three inputs are there is given below.

Table 4.7 XOR truth table with 3 inputs

A	B	C	Y
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	0
1	1	1	1

The XNOR (Exclusive NOR) gate is logically equivalent to the XOR gate followed by an inverted (NOT) gate.

Application of XOR gate: XOR gates are ideal to check the even or odd parity of a word. Since XOR gates recognize words with an odd number of 1's, odd parity words produce a high output whereas even parity words produce a low output.

4.11 BOOLEAN LAWS

(a) The Commutative law for addition and multiplication

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

(b) The distributive laws

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

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

It is interesting to note that in ordinary algebra, only the first of the above two distributing laws holds. It is easy to prove the second relationship using the truth table below:

A	B	C	(A + B) . (A + C)	A + (B . C)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

(c) The associative law for addition and multiplication

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

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

(d) Laws of addition

$$A + A = A$$

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + \overline{A} = 1$$

(e) Laws of multiplication

$$A \cdot A = A$$

$$A \cdot 1 = A$$

$$A \cdot 0 = 0$$

$$A \cdot \overline{A} = 0$$

The double inversion rule

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

i.e., complementing a complemented variable gives the original variable itself.

4.12 DE-MORGAN'S THEOREM

De-Morgan's first theorem states that, the complement of a sum of two variables is the product of the complements of the individual variables.

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

De-Morgan's second theorem:

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

These laws could be used to simplify Boolean expressions.

4.13 SIMPLIFICATION OF BOOLEAN EXPRESSIONS

Boolean expressions can be simplified with the help of any or all the theorem or laws that we have discussed. Some typical examples are shown below.

Example: 1. Simplify the Boolean expression $XY + X'Z + YZ$ Where X′ indicates the compliment of X.

Ans: $XY + X'Z + YZ = XY + X'Z + (X + X')YZ$

$$= (XY + XYZ) + (X'Z + X'YZ)$$

$$= XY(1 + Z) + X'Z(1 + Y)$$

$$= XY + X'Z$$

4.14 IMPLEMENTATION OF BOOLEAN EXPRESSIONS USING GATES

Generally, any Boolean expression can be expressed either in Sum of Products (SOP) or Product of Sum (POS) form. [Expressions in the form $Y = AB + BC + C$ are called sum of product and $Y = (A + B)(C + D)$B are called Product of Sum]. Hence all Boolean expression can be implemented with AND, OR and NOT gates.

Since NAND and NOR gates are considered as universal gates, any Boolean expression can be implemented with any of these gates alone. If the function is in SOP form, it can be implemented by NAND gates and in POS form, can be implemented using NOR gates.

1. Implement the following Boolean expressions using logic gates.

   (a) $Y = AB + CD$

   (b) $Y = (\overline{A} + B) . B$

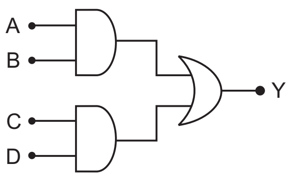
   (c) $Y = A . (B + C)$

   (d) $Y = \overline{A . B + C . D}$

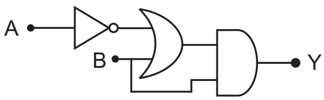
   (e) $(A + B) (B + C) (A + C)$

   Answers:

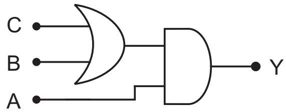
   (a) $Y = AB + CD$



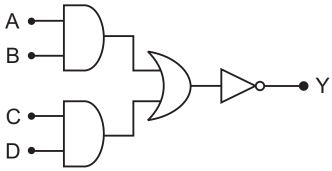
(b) $Y = (\overline{A} + B) \cdot B$



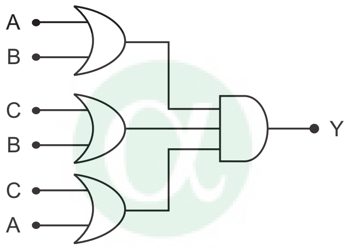
(c) $Y = A \cdot (B + C)$



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

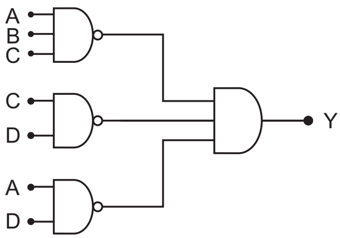


(e) $(A + B)\,(B + C)\,(A + C)$



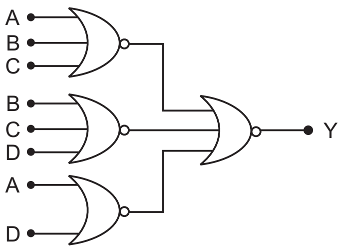
2. Implement the function $Y = ABC + CD + AD$ using *NAND* gates alone.

Answer:



3. Implement the function $Y = (A + B + C)\,(B + C + D)\,(A + D)$ using NOR gates alone.

Answer:



REVIEW QUESTIONS

1. List the main types of number systems.
2. What is the importance of binary number systems in computer science?
3. How negative of a binary number can be represented?
4. What are the needs of coding?
5. List a few commonly used codes.
6. Write short note on BCD code.
7. What is meant by self-complementing codes?
8. Write short note on unit distance code.
9. Explain in detail about error detection and correction codes.
10. Write short notes on:
    a. AND gate
    b. OR gate
    c. NOT gate
    d. XOR gate
    e. NOT gate
11. Give applications of XOR gate.
12. Give the symbol and truth table of the following gates:
    a. AND gate
    b. OR gate
    c. NOT gate
    d. XOR gate
    e. NOT gate
13. Differentiate inverter and buffer.
14. What gates are called as universal gates? Why they called so?
15. Give the truth table and symbol of:
    a. NAND gate
    b. NOR gate
16. State the following Boolean laws:
    a. commutative law
    b. distributive law
    c. associative law
17. State De' Morgans theorem.

PROBLEMS

1. Simplify the expression $(AB)'(A' + B)(B' + B)$

   Ans: A′

2. Simplify: $(A + C)(AD + AD') + AC + C$

   Ans: A + C

3. Simplify: $A'(A + B) + (B + AA)(A + B')$

   Ans: A + B

4. Implement the following Boolean functions using logic gates:

   a. $Y = ABC + C + BA$

   b. $Y = C'A + B'C + D$

   c. $Y = (A'B + C)(C' + A)(B' + A)$

   d. $Y = (A'BC' + C)(B' + A)$

5. Implement $Y = AB + C'A + DA$ using NAND gates alone.

6. Implement $Y = A' + CB + DA$ using NAND gates alone.

CHAPTER

5

COMPUTER HARDWARE AND PERIPHERALS

5.1 INTRODUCTION

The computer will be of no use unless it is able to communicate with the outside world. Input/Output devices are required for users to communicate with the computer. In simple terms, input devices bring information INTO the computer and output devices bring information OUT of a computer system. These input/output devices are also known as peripherals since they surround the CPU and memory of a computer system. In this section, we will discuss about a few input/output and connectivity devices and technologies.

5.2 INPUT DEVICES

Devices used to provide data and instructions to the computer are called input devices. Some important input devices are Key board, Mouse, Scanner, MICR, Web camera, Microphone etc.

5.2.1 Keyboard

A computer keyboard is an input device that connects to the computer and sends signals between that and the computer. In general, a keyboard uses switches and circuits to translate a person's keystrokes into a signal a computer can understand. Using a keyboard, a person can type a document, use keystroke shortcuts, access menus, play games and perform a variety of other tasks. Keyboards can have different keys depending on the manufacturer, the system they're designed for, and whether they are attached to a desktop computer or part of a laptop. But for the most part, these keys, also called keycaps, are the same size and shape from keyboard to keyboard. They're also placed at a similar distance from one another in a similar pattern, no matter what language or alphabet the keys represent.

Most keyboards have between 80 and 110 keys, including: Typing keys, A numeric keypad, Function keys and Control keys. The typing keys include the letters of the alphabet, generally laid out in the same pattern used for typewriters. According to legend, this layout, known as QWERTY for its first six letters, helped keep mechanical typewriters' metal arms from colliding and jamming as people typed. Some people question this story — whether it's true or not, the QWERTY pattern had long been a standard by the time computer keyboards came around.

Keyboards can also use a variety of other typing key arrangements. The most widely known is Dvorak, named for its creator, August Dvorak. The Dvorak layout places all of the vowels on the left side of the keyboard and the most common consonants on the right. The most commonly used letters are all found along the home row. The home row is the main row where you place your fingers when you begin typing. People who prefer the Dvorak layout say it increases their typing speed and reduces fatigue. Other layouts include ABCDE, XPERT, QWERTZ and AZERTY. Each is named for the first keys in the pattern. The QWERTZ and AZERTY arrangements are commonly used in Europe.

The numeric keypad is a more recent addition to the computer keyboard. As the use of computers in business environments increased, so did the need for speedy data entry. Since a large part of the data was numbers, a set of 17 keys, arranged in the same configuration found on adding machines and calculators, was added to the keyboard.

In 1986, IBM further extended the basic keyboard with the addition of function and control keys. Applications and operating systems can assign specific commands to the function keys. Control keys provide cursor and screen control. Four arrow keys arranged in an inverted T formation between the typing keys and numeric keypad move the cursor on the screen in small increments. Other common control keys include: Home, End, Insert, Delete, Page Up, Page Down, Control (Ctrl), Alternate (Alt) and Escape (Esc).

The Windows keyboard adds some extra control keys: two Windows or Start keys, and an Application key. Apple keyboards, on the other hand, have Command (also known as "Apple") keys. A keyboard developed for Linux users features Linux-specific hot keys, including one marked with "Tux" the penguin - the Linux logo/mascot.

Keyboard connection: As you type, the processor in the keyboard analyzes the key matrix and determines what characters to send to the computer. It maintains these characters in its memory buffer and then sends the data.

Many keyboards connect to the computer through a cable with a PS/2 or USB (Universal Serial Bus) connector. Laptops use internal connectors. Regardless of which type of connector is used, the cable must carry power to the keyboard, and it must carry signals from the keyboard back to the computer.

Wireless keyboards, on the other hand, connect to the computer through infrared (IR), radio frequency (RF) or Bluetooth connections. IR and RF connections are similar to that in a TV remote control. Regardless of which sort of signal they use, wireless keyboards require a receiver, either built in or plugged in to the USB port, to communicate with the computer. Since they don't have a physical connection to the computer, wireless keyboards have an AC power connection or use batteries for power.

Whether it's through a cable or wireless, the signal from the keyboard is monitored by the computer's keyboard controller. This is an integrated circuit

(IC) that processes all of the data that comes from the keyboard and forwards it to the operating system. When the operating system (OS) is notified that there is data from the keyboard, it checks to see if the keyboard data is a system level command. A good example of this is Ctrl-Alt-Delete on a Windows computer, which reboots the system. Then, the OS passes the keyboard data on to the current application.

The application determines whether the keyboard data is a command, like Alt-f, which opens the File menu in a Windows application. If the data is not a command, the application accepts it as content, which can be anything from typing a document to entering a URL to performing a calculation. If the current application does not accept keyboard data, it simply ignores the information. This whole process, from pressing the key to entering content into an application, happens almost instantaneously.

Electronics of a keyboard: A keyboard is a lot like a miniature computer. It has its own processor and circuitry that carries information to and from that processor. A large part of this circuitry makes up the key matrix.

The key matrix is a grid of circuits underneath the keys. In all keyboards (except for capacitive models, which we'll discuss in the next section), each circuit is broken at a point below each key. When you press a key, it presses a switch, completing the circuit and allowing a tiny amount of current to flow through. The mechanical action of the switch causes some vibration, called bounce, which the processor filters out. If you press and hold a key, the processor recognizes it as the equivalent of pressing a key repeatedly.

When the processor finds a circuit that is closed, it compares the location of that circuit on the key matrix to the character map in its read-only memory (ROM). A character map is basically a comparison chart or lookup table. It tells the processor the position of each key in the matrix and what each keystroke or combination of keystrokes represents. For example, the character map lets the processor know that pressing the a key by itself corresponds to a small letter "a," but the Shift and a keys pressed together correspond to a capital "A."

A computer can also use separate character maps, overriding the one found in the keyboard. This can be useful if a person is typing in a language that uses letters that don't have English equivalents on a keyboard with English letters. People can also set their computers to interpret their keystrokes as though they were typing on a Dvorak keyboard even though their actual keys are arranged in a QWERTY layout. In addition, operating systems and applications have keyboard accessibility settings that let people change their keyboard's behaviour to adapt to disabilities.

Keyboards use a variety of switch technologies. Capacitive switches are considered to be non-mechanical because they do not physically complete a circuit like most other keyboard technologies. Instead, current constantly flows through all parts of the key matrix. Each key is spring-loaded and has a tiny plate attached to the bottom of it. When you press a key, it moves this plate closer to the plate below it. As the two plates move closer together, the amount of current flows through the matrix changes. The processor detects the change and interprets it as a key press for that location. Capacitive switch keyboards are expensive, but they have a longer life than any other keyboard. Also, they do not have problems with bounce since the two surfaces never come into actual contact.

All of the other types of switches used in keyboards are mechanical in nature. Each provides a different level of audible and tactile response — the sounds and sensations that typing creates. Mechanical key switches include: Rubber dome, Membrane, Metal contact and Foam element.

Rubber dome switches are very common. They use small, flexible rubber domes, each with a hard carbon center. When you press a key, a plunger on the bottom of the key pushes down against the dome, and the carbon center presses against a hard, flat surface beneath the key matrix. As long as the key is held, the carbon center completes the circuit. When the key is released, the rubber dome springs back to its original shape, forcing the key back up to its at-rest position. Rubber dome switch keyboards are inexpensive, have pretty good tactile response and are fairly resistant to spills and corrosion because of the rubber layer covering the key matrix.

Rather than having a switch for each key, membrane keyboards use a continuous membrane that stretches from one end to another. A pattern printed in the membrane completes the circuit when you press a key. Some membrane keyboards use a flat surface printed with representations of each key rather than keycaps. Membrane keyboards don't have good tactile response, and without additional mechanical components they don't make the clicking sound that some people like to hear when they're typing. However, they're generally inexpensive to make.

Metal contact and foam element keyboards are increasingly less common. Metal contact switches simply have a spring-loaded key with a strip of metal on the bottom of the plunger. When the key is pressed, the metal strip connects the two parts of the circuit. The foam element switch is basically the same design but with a small piece of spongy foam between the bottom of the plunger and the metal strip, providing a better tactile response. Both technologies have good tactile response, make satisfyingly audible "clicks," and are inexpensive to produce. The problem is that the contacts tend to wear out or corrode faster than on keyboards that use other technologies. Also, there is no barrier that prevents dust or liquids from coming in direct contact with the circuitry of the key matrix.

5.2.2 Computer Mouse

The main goal of any mouse is to translate the motion of your hand into signals that the computer can use. Let's take a look inside a track-ball mouse to see how it works. A **ball** inside the mouse touches the desktop and rolls when the mouse moves.

Two rollers inside the mouse touch the ball. One of the rollers is oriented so that it detects motion in the X direction, and the other is oriented 90 degrees to the first roller so it detects motion in the Y direction. When the ball rotates, one or both of these rollers rotate as well. The rollers each connect to a **shaft**, and the shaft spins a **disk** with holes in it. When a roller rolls, its shaft and disk spin.

On either side of the disk there is an **infrared LED** and an **infrared sensor**. The holes in the disk break the beam of light coming from the LED so that the infrared sensor sees pulses of light. The rate of the pulsing is directly related to the speed of the mouse and the distance it travels. An **on-board processor chip** reads the pulses from the infrared sensors and turns them into binary data that the computer can understand. **The logic section of a mouse is dominated by an encoder chip, a small processor that reads the pulses coming from the infrared sensors and turns them into bytes sent to the computer. You can also see the two buttons that detect clicks (on either side of the wire connector).** The chip sends the binary data to the computer through the mouse's cord. In this **opto-mechanical** arrangement, the disk moves mechanically, and an optical system counts pulses of light. On this mouse, the ball is 21 mm in diameter. The roller is 7 mm in diameter. The encoding disk has 36 holes. So if the mouse moves 25.4 mm (1 inch), the encoder chip detects 41 pulses of light.

Each encoder disk has two infrared LEDs and two infrared sensors, one on each side of the disk (so there are four LED/sensor pairs inside a mouse). This arrangement allows the processor to detect the disk's **direction of rotation**. There is a piece of plastic with a small, precisely located hole that sits between the encoder disk and each infrared sensor. This piece of plastic provides a window through which the infrared sensor can "see." The window on one side of the disk is located slightly higher than it is on the other -- one-half the height of one of the holes in the encoder disk, to be exact. That difference causes the two infrared sensors to see pulses of light at slightly different times. There are times when one of the sensors will see a pulse of light when the other does not, and vice versa.

Most mice on the market today use a USB connector to attach to your computer. USB is a standard way to connect all kinds of peripherals to your computer, including printers, digital cameras, keyboards and mice. Some older mice, many of which are still in use today, have a PS/2 type connector. Instead of a PS/2 connector, a few other older mice use a serial type of connector to attach to a computer.

Optical Mice: Developed by Agilent Technologies and introduced to the world in late 1999, the optical mouse actually uses a tiny camera to take thousands of pictures every second. It is able to work on almost any surface without a mouse pad; most optical mice use a small, red light-emitting diode (LED) that bounces light off that surface onto a complimentary metal-oxide semiconductor (CMOS) sensor. In addition to LEDs, a recent innovation is laser-based optical mice that detect more surface details compared to LED technology. This results in the ability to use a laser-based optical mouse on even more surfaces than an LED mouse.

The CMOS sensor sends each image to a digital signal processor (DSP) for analysis. The DSP detects patterns in the images and examines how the patterns have moved since the previous image. Based on the change in patterns over a sequence of images, the DSP determines how far the mouse has moved and sends the corresponding coordinates to the computer. The computer moves the cursor on the screen based on the coordinates received from the mouse. This happens hundreds of times each second, making the cursor appear to move very smoothly.

Optical mice have several benefits over track-ball mice:

- No moving parts in it hence less wear and a lower chance of failure.
- There's no way for dirt to get inside the mouse and interfere with the tracking sensors.
- Increased tracking resolution means a smoother response.
- They don't require a special surface, such as a mouse pad.

A number of factors affect the accuracy of an optical mouse. One of the most important aspects is the **resolution**. The resolution is the number of pixels per inch that the optical sensor and focusing lens "see" when you move the mouse. Resolution is expressed as dots per inch (dpi). The higher the resolution, the more sensitive the mouse is and the less you need to move it to obtain a response.

Most mice have a resolution of 400 or 800 dpi. However, mice designed for playing electronic games can offer as much as 1600 dpi resolution. Some gaming mice also allow you to decrease the dpi on the fly to make the mouse less sensitive in situations when you need to make smaller, slower movements.

Usually, corded mice have been more responsive than wireless mice. Other factors that affect quality include:

- **Size of the optical sensor** - larger is generally better, assuming the other mouse components can handle the larger size. Sizes range from 16 x 16 pixels to 30 x 30 pixels.
- **Refresh rate** - it is how often the sensor samples images as you move the mouse. Faster is generally better, assuming the other mouse components can process them. Rates range from 1500 to 6000 samples per second.
- **Image processing rate** - is a combination of the size of the optical sensor and the refresh rate. Again, faster is better and rates range from 0.486 to 5.8 mega pixels per second.
- **Maximum speed** - is the maximum speed that you can move the mouse and obtain accurate tracking. Faster is better and rates range from 16 to 40 inches per second.

Wireless Mice: Wireless mouse offers a solution to the menace of wires that connect the mouse to the computer. Most wireless mice use radio frequency (RF) technology to communicate information to the computer. Being radio-based, wireless mouse requires two main components: a transmitter and a receiver. The transmitter is housed in the mouse. It sends an electromagnetic (radio) signal that encodes the information about the mouse's movements and the buttons you click. The receiver, which is connected to the computer, accepts the signal, decodes it and passes it on to the mouse driver software and the computer's operating system.

The receiver can be a separate device that plugs into the computer, a special card that place in an expansion slot, or a built-in component. Unlike infrared technology, which is commonly used for short-range wireless communications such as television remote controls, RF devices do not need a clear line of sight between the transmitter (mouse) and receiver. Just like other types of devices that use radio waves to communicate, a wireless mouse signal can pass through barriers such as a desk or monitor.

RF technology provides a number of additional benefits for wireless mice. These include: RF transmitters require low power and can run on batteries; RF components are inexpensive and light weight.

Pairing and Security: In order for the transmitter in the mouse to communicate with its receiver, they must be paired. This means that both devices are operating at the same frequency on the same channel using a common identification code. A channel is simply a specific frequency and code. The purpose of pairing is to filter out interference from other sources and RF devices.

To protect the information the mouse transmits to the receiver, most wireless mice include an **encryption scheme** to encode data into an unreadable format. Some devices also use a **frequency hopping** method, which causes the mouse and receiver to automatically change frequencies using a predetermined pattern. This provides additional protection from interference and eavesdropping.

Bluetooth Mice: One of the RF technologies that wireless mice commonly use is Bluetooth. Bluetooth technology wirelessly connects peripherals such as printers, headsets, keyboards and mice to Bluetooth-enabled devices such as computers and personal digital assistants (PDAs). Since a Bluetooth receiver can accommodate multiple Bluetooth peripherals at one time, Bluetooth is also known as a **personal area network** (PAN). Bluetooth devices have a range of about 33 feet (10 meters).

RF Mice: The other common type of wireless mouse is an RF device that operates at 27 MHz and has a range of about 6 feet (2 meters). More recently, 2.4 GHz RF mice have hit the market with the advantage of a longer range -- about 33 feet (10 meters) and faster transmissions with less interference. Multiple RF mice in one room can result in **cross-talk**, which means that the receiver inadvertently picks up the transmissions from the wrong mouse. Pairing and multiple channels help to avoid this problem.

Typically, the RF receiver plugs into a USB port and does not accept any peripherals other than the mouse (and perhaps a keyboard, if sold with the mouse). Some portable models designed for use with notebook computers come with a compact receiver that can be stored in a slot inside the mouse when not in use.

5.2.3 Trackball

A trackball is a pointing device consisting of a ball held by a socket containing sensors to detect a rotation of the ball about two axes—like an upside-down mouse with an exposed protruding ball. The user rolls the ball with the thumb, fingers, or the palm of the hand to move a pointer. Compared with a mouse, a trackball has no limits on effective travel; at times, a mouse can reach an edge of its working area while the operator still wishes to move the screen pointer farther. With a trackball, the operator just continues rolling. Figure 5.1 shows the line drawing of a typical track-ball.

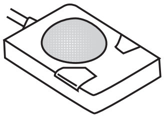


Fig. 5.1 Track-ball

The advantage of trackballs over mice is that the trackball is stationary so it does not require much space to use it. In addition, you can place a trackball on any type of surface, including your lap. For both these reasons, trackballs are popular pointing devices for portable computers.

Large trackballs are sometimes seen on computerized special-purpose workstations, such as the radar consoles in air-traffic control room or sonar equipment on a ship or submarine. Modern installations of such equipment may use mice instead, since most people now already know how to use one. However, military mobile anti-aircraft radars and submarine sonars tend to continue using trackballs, since they can be made more durable and fit for fast emergency use. Large and well made ones allow easier high precision work, for which reason they may still be used in these applications (where they are often called "tracker balls") and in computer-aided design.

5.2.4 Touch Pad

A touchpad (or trackpad) is a pointing device featuring a actile sensor, a specialized surface that can translate the motion and position of a user's fingers to a relative position on screen. Touchpads are a common feature of laptop computers, and are also used as a substitute for a mouse where desk space is scarce. Because they vary in size, they can also be found on personal digital assistants (PDAs) and some portable media players. Wireless touchpads are also available as detached accessories.

Touchpads operate in one of several ways, including capacitive sensing and conductance sensing. The most common technology used as of 2010 entails sensing the capacitive virtual ground effect of a finger, or the capacitance between sensors. Capacitance-based touchpads will not sense the tip of a pencil or other similar implement. Gloved fingers may also be problematic.

Some touchpads and associated device driver software may interpret tapping the pad as a click, and a tap followed by a continuous pointing motion (a "click-and-a-half") can indicate dragging. Tactile touchpads allow for clicking and dragging by incorporating button functionality into the surface of the touchpad itself. To select, one presses down on the touchpad instead of a physical button. To drag, instead performing the "click-and-a-half" technique, one presses down while on the object, drags without releasing pressure and lets go when done. Touchpad drivers can also allow the use of multiple fingers to facilitate the other mouse buttons (commonly two-finger tapping for the center button).

5.2.5 Light Pen

A light pen, also called a selector pen, is a computer input device in the form of a light-sensitive wand used in conjunction with a computer's CRT display. It allows the user to point to displayed objects or draw on the screen in a similar way to a touchscreen but with greater positional accuracy

A light pen is fairly simple to implement. Just like a ight gun, a light pen works by sensing the sudden small change in brightness of a point on the screen when the electron gun refreshes that spot. By noting exactly where the scanning has reached at that moment, the X,Y position of the pen can be resolved. This is usually achieved by the light pen causing an interrupt, at which point the scan position can be read from a special register, or computed, from a counter or timer. The pen position is updated on every refresh of the screen.

5.2.6 Pointing Stick

The pointing stick (trademarked by IBM as the TrackPoint and by Synaptics as the TouchStyk) is an isometric joystick used as a pointing device (compare especially touchpad and rackball). It was invented by research scientist Ted Selker. It is present on many brands of laptops, including enovo's line of ThinkPad laptops, Toshiba Tecra laptops etc.It has also been used on computer mice and on some desktop keyboards (as an integrated pointing device).

The pointing stick operates by sensing applied force (hence it is also known as an isometric joystick), by using a pair of resistive strain gauges. The velocity of the pointer depends on the applied force. On a QWERTY keyboard, the stick is embedded between the "G", "H" and "B" keys, and the mouse buttons are placed just below the Spacebar. The mouse buttons can be operated right-handed or left-handed due to their placement below the keyboard along the centerline.

5.2.7 Touch Screen

A touchscreen is an electronic visual display that can detect the presence and location of a touch within the display area. The term generally refers to touching the display of the device with a finger or hand. Touch-screens can also sense other passive objects, such as a stylus. Touch screens are common in devices such as game consoles, all-in-one computers, tablet computers, and smartphones.

The touchscreen has two main attributes. First, it enables one to interact directly with what is displayed, rather than indirectly with a pointer controlled by a mouse or touchpad. Secondly, it lets one do so without requiring any intermediate device that would need to be held in the hand (other than a stylus, which is optional for most modern touch screens). Such displays can be attached to computers, or to networks as terminals. They also play a prominent role in the design of digital appliances such as the personal digital assistant (PDA), satellite navigation devices, mobile phones, and video games.

There are four types of touch screens viz; Capacitive, Infrared, Resistive, SAW (surface acoustic wave) and Guided Wave with Resistive and Capacitive being the most widely used types for Industrial Applications. All of these technologies have their own distinct characteristics, both advantageous and with limitations.

Capacitive touch screen technology is recommended for use in KIOSK applications that require a "finger touch". It will not operate with either a gloved hand or with a mechanical stylus. It is made of glass, which makes it extremely durable and scratch resistant. This glass overlay has a coating that stores the charge deposited over its surface electrically. Capacitive touch screens operate using oscillator circuits that are located in each corner of the glass overlay and measure the capacitance of the area be "touched". Depending on where the person touches the overlay, the oscillators will vary in frequency. Then a touch screen controller measures the frequency variations to ascertain the coordinates of the person's touch. When used with flat panel displays, capacitive offers drift-free stable performance that is not susceptible to deterioration over time. A capacitive touch screen is impervious to grease, dirt and water, which makes it ideal for frequent use.

Resistive touch screen technology is recommended for use in POS (Point of Sale): Grocery Stores, Hotels, Restaurants and Retail Stores; Industrial Applications: MMI (Man Machine Interface), Machine and Process Control; Portable Devices; Personal Information Management Systems; Transportation Solutions; Medical Solutions: Equipment, Instrumentation and Patient Monitoring Systems. It generally uses a display overlay composed of layers, each with a conductive coating on the interior surface. Special separator "dots" are distributed evenly across the active area and separate the conductive interior layers. The pressure from using either a mechanical stylus or finger produces an internal electrical contact at the "action point" which supplies the controller with vertical and horizontal analog voltages for data input. To reduce parallax for older "curved" CRT applications only, resistive touch screens are generally spherical to match the curvature of the CRT (true flat resistive touch overlays are also available for TFT flat panels and/or CRTs). Our resistive touch screens are anti-glare to reduce reflective shine intensity, which will slightly diffuse the light output throughout the screen. Resistive technology offers tremendous versatility in that activation can be initiated by; a gloved hand, fingernail, mechanical stylus or an ungloved finger.

SAW touch screen technology is suggested for use in ATMs, Amusement Parks, Banking and Financial Applications, Gaming Environments, Industrial Control Rooms, and KIOSK. SAW touch cannot be used within NEMA environments, as the technology cannot be gasket sealed. It has excellent durability that allows it to continue working if scratched since the overlay for the touch sensor is a solid glass display. The disadvantage to this glass overlay is that is breakable and won't work in wash down conditions. The waves are spread across the screen by bouncing off reflector arrays along the edges of the overlay. The waves are detected by two “receivers”. The acoustic wave weakens when the user touches the glass with their finger, gloved hand or soft stylus. The coordinates are then determined by the controller circuitry that measures the time at which the amplitude declines. It is the only technology that can produce a Z-coordinate axis response. SAW technology offers drift-free stable performance that is always precise. SAW offers superior image clarity, resolution, and high light transmission.

Infrared touch screen technology is based on “legacy” technology and is becoming increasingly replaced by Resistive or Capacitive touch systems. Over the years, Infra-red bezels have proven to be a very reliable technology for use in ATMs, Food Service and Preparation, KIOSK, Medical Instrumentation, Process Control Systems, and Transportation Tracking applications. It does not incorporate any sort of “overlay” that could inhibit screen clarity or brightness, but instead, uses a special bezel of LEDs (light emitting diodes) along with diametrically opposing phototransistor detectors which surround the glass of the of the display surface. The controller circuitry scans the screen with an invisible lattice of infra-red light beams just in front of the surface that directs a sequence of pulses to the LED’s. It then detects information at the location where the LEDs have become interrupted by a stylus or finger. The infrared frame housing the transmitters can impose design constraints on operator interface products.

5.2.8 Bar Code Reader

A barcode reader (or barcode scanner) is an electronic device for reading printed barcodes. Like a flatbed scanner, it consists of a light source, a lens and a light sensor translating optical impulses into electrical ones. Additionally, nearly all barcode readers contain decoder circuitry analyzing the barcode’s image data provided by the sensor and sending the barcode’s content to the scanner’s output port.

5.2.9 Scanner

In computing, an image scanner—often abbreviated to just scanner—is a device that optically scans images, printed text, handwriting, or an object, and converts it to a digital image. Common examples found in offices are variations of the desktop (or flatbed) scanner where the document is placed on a glass window for scanning. Hand-held scanners, where the device is moved by hand, have evolved from text scanning “wands” to 3D scanners used for industrial design, reverse engineering, test and measurement, orthotics, gaming and other applications. Mechanically driven scanners that move the document are typically used for large-format documents, where a flatbed design would be impractical.

Modern scanners typically use a charge-coupled device (CCD) or a Contact Image Sensor (CIS) as the image sensor, whereas older drum scanners use a photomultiplier tube as the image sensor. A rotary scanner, used for high-speed document scanning, is another type of drum scanner, using a CCD array instead of a photomultiplier. Other types of scanners are planetary scanners, which take photographs of books and documents, and 3D scanners, for producing three-dimensional models of objects.

Another category of scanner is digital camera scanners, which are based on the concept of reprographic cameras. Due to increasing resolution and new features such as anti-shake, digital cameras have become an attractive alternative to regular scanners. While still having disadvantages compared to traditional scanners (such as distortion, reflections, shadows, low contrast), digital cameras offer advantages such as speed, portability and gentle digitizing of thick documents without damaging the book spine.

5.2.10 Joystick

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. A joystick, also known as the control column, is the principal control device in the cockpit of many civilian and military aircraft, either as a center stick or side-stick. It often has supplementary switches to control various aspects of the aircraft's flight.

Joysticks are often used to control video games, and usually have one or more push-buttons whose state can also be read by the computer. A popular variation of the joystick used on modern video game consoles is the analog stick. Joysticks are also used for controlling machines such as cranes, trucks, underwater unmanned vehicles, wheelchairs, surveillance cameras and zero turning radius lawn mowers. Miniature finger-operated joysticks have been adopted as input devices for smaller electronic equipment such as mobile phones.

5.3 OUTPUT DEVICES

5.3.1 Displays

Displays are one of the common output devices. The visual interface between you and your computer is known as the display. In desktop computers, an external display is often called a monitor. A Cathode-Ray Tube (CRT) monitor resembles a television set without the tuning or volume controls. A liquid-crystal-display (LCD) is lightweight and thin. This type of display is used in notebook and portable computers. It has become increasingly popular for desktop computers because the technology has improved and has become somewhat less expensive.

Showing detail: The image resolution of a computer display is important. This is the extent to which it can show detail: the better the resolution, the sharper the image. Resolution can be specified in terms of dot size or dot pitch. This is the diameter, in millimetres (mm), of the individual elements in the display—the "smallest unbreakable pieces."

A good display has a dot pitch that is a small fraction of a millimeter. A typical CRT has a dot pitch of 0.25 or 0.26 mm. The smaller the number, the higher the resolution and, all other factors being equal, the crisper the image in an absolute sense. Image resolution can be specified in a general sense as a pair of numbers, representing the number of pixels (picture elements) the screen shows horizontally and vertically.

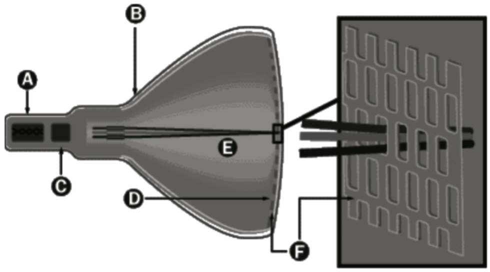
For a particular screen size, the greater the number of pixels the unit can display, the crisper the image. In personal computers, typical displays have either 800 × 600 resolution (800 pixels wide by 600 pixels high) or 1024 × 768 resolution. A few can work up to 1600 × 1200. Older computers often work at 640 × 480. Screen sizes are given in terms of diagonal measure; a popular size is the so-called 17-inch CRT (which actually has a viewable display measure of about 15.5 inches). This will work quite well at 800 x 600 pixels. For higher pixel dimensions, a larger screen is preferable, such as 19 or even 21 inch, but CRTs that size are rather expensive, can easily take up most of the surface of your desk, and can weigh 50 to 100 pounds. A small "15-inch" CRT (with viewable display measure of 13 to 14 inches) is sufficient if you intend to work at 640 x 480 pixels.

A high-end display is crucial for doing graphics work, when using the Internet, in remote-control robotics, and in computer games. Besides these practical advantages, a sharp display is more pleasant to work with than a marginal one. Along with memory and hard drive capacity, the display is one of the most important parts of a computer from a user-friendliness standpoint. On the job, long hours at a computer can get tedious even if the machine is perfect. An inadequate display can give rise to eye strain and headaches and can also degrade the quality and accuracy of work done by people using the computer. Main display technologies like CRT, plasma and LCD are discussed in the following sections.

5.3.1.1 *Cathode Ray Tube (CRT) Monitors*

In a cathode ray tube, the "cathode" is the heated filament. The heated filament is in a vacuum created inside a glass "tube." The "ray" is a stream of electrons generated by an electron gun that naturally pour off a heated cathode into the vacuum. Since the anode is positive, it attracts the electrons pouring off the cathode. This screen is coated with phosphor, an organic material that glows when struck by the electron beam. A CRT monitor contains millions of tiny red, green, and blue phosphor dots that glow when struck by an electron beam that travels across the screen to create a visible image. Fig. 5.2 shows the cross-sectional view of a CRT.

There are three ways to filter the electron beam in order to obtain the correct image on the monitor screen: shadow mask, aperture grill and slot mask. These technologies also impact the sharpness of the monitor's display.



A – Cathode B – Conductive coating C – Anode
D – Phosphor-coated screen E – Electron beams F - Shadow mask

Fig. 5.2 Cross-sectional view of a CRT

Shadow-mask: A shadow mask is a thin metal screen filled with very small holes. Three electron beams pass through the holes to focus on a single point on a CRT displays' phosphor surface. The shadow mask helps to control the electron beams so that the beams strike the correct phosphor at just the right intensity to create the desired colours and image on the display. The unwanted beams are blocked or "shadowed."

Aperture-grill: Monitors based on the Trinitron technology, which was pioneered by Sony, use an aperture-grill instead of a shadow-mask type of tube. The aperture grill consists of tiny vertical wires. Electron beams pass through the aperture grill to illuminate the phosphor on the faceplate. Most aperture-grill monitors have a flat faceplate and tend to represent a less distorted image over the entire surface of the display than the curved faceplate of a shadow-mask CRT. However, aperture-grill displays are normally more expensive.

Slot-mask: A less-common type of CRT display, a slot-mask tube uses a combination of the shadow-mask and aperture-grill technologies. Rather than the round perforations found in shadow-mask CRT displays, a slot-mask display uses vertically aligned slots. The design creates more brightness through increased electron transmissions combined with the arrangement of the phosphor dots.

Dot pitch: Dot pitch is an indicator of the sharpness of the displayed image. It is measured in millimeters (mm), and a smaller number means a sharper image.

- In a shadow-mask CRT monitor, you measure dot pitch as the diagonal distance between two like-colored phosphors. Some manufacturers may also cite a horizontal dot pitch, which is the distance between two-like colored phosphors horizontally.
- The dot pitch of an aperture-grill monitor is measured by the horizontal distance between two like-colored phosphors. It is also sometimes are called stripe pitch.
- The smaller and closer the dots are to one another, the more realistic and detailed the picture appears. When the dots are farther apart, they become noticeable and make the image look grainier. Unfortunately, manufacturers are not always upfront about dot pitch measurements, and you cannot

necessarily compare shadow-mask and aperture-grill CRT types, due to the difference in horizontal and vertical measurements.

Refresh Rate: In monitors based on CRT technology, the refresh rate is the number of times that the image on the display is drawn each second. If your CRT monitor has a refresh rate of 72 Hertz (Hz), then it cycles through all the pixels from top to bottom 72 times a second. Refresh rates are very important because they control flicker, and you want the refresh rate as high as possible. This is especially important with larger monitors where flicker is more noticeable. Recommendations for refresh rate and resolution include 1280 × 1024 at 85 Hertz or 1600 × 1200 at 75 Hertz.

Multiple Resolutions: Because a CRT uses electron beams to create images on a phosphor screen, it supports the resolution that matches its physical dot (pixel) size as well as several lesser resolutions. For example, a display with a physical grid of 1280 rows by 1024 columns can obviously support a maximum resolution of 1280 × 1024 pixels. It also supports lower resolutions such as 1024 × 768, 800 × 600, and 640 × 480.

5.3.1.2 *Liquid crystal displays (LCD's)*

Liquid crystal displays (LCD's) offer several advantages over traditional cathode-ray tube displays. LCD's are flat, and they use only a fraction of the power required by CRT's. They are easier to read and more pleasant to work with for long periods of time than most ordinary video monitors. One should also know that there are several tradeoffs, such as limited view angle, brightness, and contrast, not to mention high manufacturing costs.

Today's LCD's come mostly in two flavors; passive and active. The less expensive passive matrix displays trade off picture quality, view angle, and response time with power requirements and manufacturing costs. Active matrix displays have superior picture quality and viewing characteristics, but need more power to run and are much more expensive to fabricate.

Liquid crystal is a special state of material that certain kinds of matter can enter into under the right conditions. The molecules in solids exhibit both positional and orientational order, in other words the molecules are constrained to point only certain directions and to be only in certain positions with respect to each other. In liquids, the molecules do not have any positional or orientational order; the direction the molecules point and positions are random.

Active matrix (TFT) LCDs: A display made with active matrix TFT technology is a LCD that has a separate and independent transistor for each pixel on the display. Having a transistor at each pixel means that the current that triggers pixel illumination can be smaller and therefore it may be switched on and off more quickly. The pixel emits colored light which passes through a RGB (Red, Green, Blue) filter. The result of this is a screen with a very small response time and direct pixel addressing is also possible which makes the TFT LCD suitable for video and fast graphic applications.

Passive matrix LCDs: In passive matrix LCD, each colour pigment is turned on or off by supplying current through a timed operation method to electrodes oriented in length and width. TFT LCDs may be noted for their outstanding screen image, but passive matrix LCDs certainly offer superior economy.

5.3.1.3 *Comparison between LCD and CRT Monitors*

LCD Monitors:

- Require less power - Power consumption varies greatly with different technologies. CRT displays are somewhat power-hungry, at about 100 watts for a typical 19-inch display. The average is about 45 watts for a 19-inch LCD display. LCDs also produce less heat.
- Smaller and weigh less - An LCD monitor is significantly thinner and lighter than a CRT monitor, typically weighing less than half as much. In addition, you can mount an LCD on an arm or a wall, which also takes up less desktop space.
- More adjustable - LCD displays are much more adjustable than CRT displays. With LCDs, you can adjust the tilt, height, swivel, and orientation from horizontal to vertical mode. As noted previously, you can also mount them on the wall or on an arm.
- Less eye strain - Because LCD displays turn each pixel off individually, they do not produce a flicker like CRT displays do. In addition, LCD displays do a better job of displaying text compared with CRT displays.

CRT Monitors:

- Less expensive - Although LCD monitor prices have decreased, comparable CRT displays still cost less.
- Better colour representation - CRT displays have historically represented colours and different gradations of colour more accurately than LCD displays. However, LCD displays are gaining ground in this area, especially with higher-end models that include colour-calibration technology.
- More responsive - Historically, CRT monitors have had fewer problems with ghosting and blurring because they redrew the screen image faster than LCD monitors. Again, LCD manufacturers are improving on this with displays that have faster response times than they did in the past.
- Multiple resolutions - If you need to change your display's resolution for different applications, you are better off with a CRT monitor because LCD monitors don't handle multiple resolutions as well.
- More rugged - Although they are bigger and heavier than LCD displays, CRT displays are also less fragile and harder to damage.

5.3.1.4 *Plasma Displays*

Cathode ray tubes produce crisp, vibrant images, but they are bulky. As an alternative to CRT, plasma flat panel displays came into market. These displays have wide screens, comparable to the largest CRT sets, but they are only about 6 inches (15 cm) thick. The basic idea of a plasma display is to illuminate tiny, colored fluorescent lights to form an image. Each pixel is made up of three fluorescent lights - a red light, a green light and a blue light. Just like a CRT television, the plasma display varies the intensities of the different lights to produce a full range of colours.

Figure 5.3 shows the cross-sectional view of a plasma display. Nowadays, Xenon and Neon gas are used in the manufacturing of plasma displays. The xenon and neon gas in a plasma television is contained in hundreds of thousands of tiny cells positioned between two plates of glass. Long electrodes are also sandwiched between the glass plates, on both sides of the cells. The address electrodes sit behind the cells, along the rear glass plate. The transparent display electrodes, which are surrounded by an insulating dielectric material and covered by a magnesium oxide protective layer, are mounted above the cell, along the front glass plate.

Both sets of electrodes extend across the entire screen. The display electrodes are arranged in horizontal rows along the screen and the address electrodes are arranged in vertical columns. As you can see in the diagram below, the vertical and horizontal electrodes form a basic grid. To ionize the gas in a particular cell, the plasma display's computer charges the electrodes that intersect at that cell. It does this thousands of times in a small fraction of a second, charging each cell in turn.

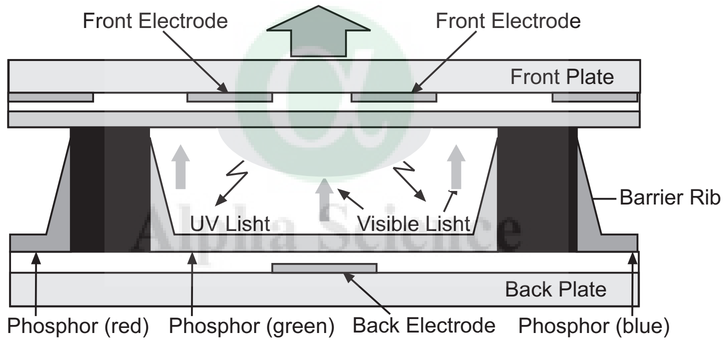


Fig. 5.3 Cross-sectional view of plasma display panel

When the intersecting electrodes are charged (with a voltage difference between them), an electric current flow through the gas in the cell. The current creates a rapid flow of charged particles, which stimulates the gas atoms to release ultraviolet photons. The released ultraviolet photons interact with phosphor material coated on the inside wall of the cell. Phosphors are substances that give off light when they are exposed to other light. When an ultraviolet photon hits a phosphor atom in the cell, one of the phosphor's electrons jumps to a higher energy level and the atom heats up. When the electron falls back to its normal level, it releases energy in the form of a visible light photon.

5.3.2 Printers

A computer printer is an electromechanical device that produces hard copy (text and images on paper). The most common printers are dot-matrix, thermal, inkjet, and laser.

Dot matrix printers is the least expensive, in terms of both the purchase price and the long-term operating cost. Dot-matrix printers produce fair print quality for most manuscripts, reports, term papers, practical purposes, LED printers are equivalent to laser printers, The laser blinks rapidly while it scans a cylindrical drum. The drum has special properties that cause it to attract the printing chemical, called toner, in some places but not others, creating an image pattern that will ultimately appear on the paper. A sheet of paper is pulled past the drum and also past an electrostatic charger. Toner from the drum is attracted to the paper. The image thus goes onto the paper, although it has not yet been permanently fused, or bonded, to the paper. The fuser, a hot pair of roller/squeezers, does this job, completing the printing process.

The main advantage of laser printers is their excellent print and graphics quality. The image resolution of a laser printer ranges from about 300 dots per inch (DPI) for older units to 1200, 2400, or even more DPI in state-of-the-art machines. Another asset of laser printers is that they make almost no noise. Yet another good feature is that this type of printer is relatively fast. The best highend machines can produce a couple of dozen copies per minute.

5.3.3 Plotters

A plotter is a graphics printer that uses a pen or pencil to draw images. Plotters differ from printers in that plotters use continuous lines to create images while printers use a collection of dots. Like printers, plotters are connected to computers and are used to produce complex images and text. However, plotters are much slower than printers because of the mechanical motion necessary to draw detailed graphics using continuous lines. Architects and product designers use plotters for technical drawings and computer-aided design purposes since plotters have the ability to create large images on oversized sheets of paper. Additionally, many garment and sign manufacturers use cutting plotters in which the plotter's pen is replaced with a sharp razorblade.

A plotter works closely with a computer's imaging software to produce a final picture or object. The first step in using a plotter is to enter the appropriate coordinates for where you want the image to appear on the paper. Modern software allows the user to accomplish this goal very easily by drawing lines and images with the imaging software. Once the schematics for the image are complete, the computer downloads the coordinates to the plotter, which interprets the code and calculates the most efficient path for the pen and paper. Newer plotters use one of two programming languages: Hewlett-Packard's HPGL2 or Houston Instruments' DMPL. Early plotters contained two separate pens, one that moved vertically and one that moved horizontally. These plotters were limited in terms of the complexity of the images they could produce as well as the speed at which they could operate. Modern plotters use a sliding roller, which moves the paper against a stationary pen. A plotter pen is usually a hollow fiber rod with a sharpened end. The ink supply runs through the center of the rod and dispenses through the sharpened tip. The paper moves horizontally and vertically against the pen until the drawing is complete.

Some manufacturers create cutting plotters by replacing a plotter's pen with a knife. The cutting plotter may also contain a pressure control device that regulates how firmly the knife presses down on the material. Many cutting plotters operate by moving the cutter's knife rather than the material itself. However, vinyl sign cutters and other cutting plotters that work with flexible material continue to use the sliding roller featured in pen plotters. Manufacturers who wanted cutting plotters used to purchase and overhaul pen plotters, but many companies have begun producing cutting plotters that can be purchased directly. Cutting plotters are useful in manufacturing a wide variety of items, including customized signs, billboards, posters and vinyl sheets.

5.3.4 Fax

Fax (short for facsimile), sometimes called tele-copying, is the telephonic transmission of scanned printed material (both text and images), normally to a telephone number connected to a printer or other output device. The original document is scanned with a fax machine (or a tele-copier), which processes the contents (text or images) as a single fixed graphic image, converting it into a bitmap, and then transmitting it through the telephone system. The receiving fax machine reconverts the coded image, printing a paper copy. Before digital technology became widespread, for many decades, the scanned data was transmitted as analog.

Although businesses usually maintain some kind of fax capability, the technology has faced increasing competition from Internet-based alternatives. Fax machines still retain some advantages, particularly in the transmission of sensitive material which, if sent over the Internet unencrypted, may be vulnerable to interception, without the need for telephone tapping. In some countries, because electronic signatures on contracts are not recognized by law while faxed contracts with copies of signatures are, fax machines enjoy continuing support in business.

In many corporate environments, standalone fax machines have been replaced by fax servers and other computerized systems capable of receiving and storing incoming faxes electronically, and then routing them to users on paper or via an email (which may be secured). Such systems have the advantage of reducing costs by eliminating unnecessary printouts and reducing the number of inbound analog phone lines needed by an office.

5.4 FIREWIRE (IEEE1394 HIGH PERFORMANCE SERIAL BUS)

FireWire allows users to connect storage devices and other peripherals to host computers, giving unparalled flexibility in data capture, storage, transportation, and backup capabilities. FireWire allows devices to be hot swapped from one computer to another. "IEEE1394 High Performance Serial Bus" is the official name for this technology, which is also known with the commercial trademark FireWire, a trademark owned by Apple.

A single 1394 port can be used to connect up 63 external devices. In addition to its high speed, 1394 also supports isochronous data — delivering data at a guaranteed rate. This makes it ideal for devices that need to transfer high levels of data in real-time, such as video devices. Like USB, 1394 supports both Plug-and-Play and hot plugging, and also provides power to peripheral devices. IEEE-1394 is usuable in both tree (similar to star) and daisy-chain topologies, or a combination of the two. In any case, cable length is limited to about 15 feet between devices.

Firewire Operation: FireWire allows the attachment of external devices to host computers through a cable which is composed of the following wiring components:

- Serial Data Pairs, of which there are two, giving a total of 4 wires.
- Power, which is generally somewhere between 8 to 24 volts DC.
- Ground, which is a lead that provides a current return for the Power line.
- Shield, which helps prevent the emission of Radio Frequency Interference from the FireWire cable.

The Power and Ground lines must be present to allow FireWire devices to be bus powered. For instance, portable FireWire drives usually run off of bus power, meaning that they will function when attached to a powered FireWire host. Other types of FireWire devices, such as Desktop FireWire drives, usually (but not always) require a separate power supply. As a result, they do not utilize any power from the Power/Ground pair provided by the host FireWire port. They obtain their power from an independent power supply. Manufacturers are not required to provide Power and Ground within the FireWire host. Two different types of commonly used FireWire connectors have been defined for FireWire 400 usage. One of these types is the more common six pin connector, while the other type is a physically smaller connector which omits Power and Ground. Obviously, bus powered FireWire devices will not work when attached to a FireWire host which does not provide bus power.

Types of Firewire Cables:

- **6-Pin 6 Wire Cables:** The 6 conductor type has two separately shielded twisted pairs for data and two power wires in an overall shielded cable with 6-pin connectors on either side.
- **4-Pin 4 Wire Cables:** The 4 wire cable uses two separately shielded data cables without power wires in an overall shielded cable with 4-pin connectors on either end.
- **6-Pin to 4-Pin Cables:** This cable uses either type of actual cable, with a 6-pin connector on one side, and a 4-pin connector on the other side of the cable.

Firewire Host Adaptors: Fig. 5.4 shows a FireWire host adapter with 2 external FireWire ports, 1 internal FireWire port and a PCI-Express interface. The terms "external" and "internal" refer to whether the port is actually accessible from outside the PC enclosure (visible from the outside).

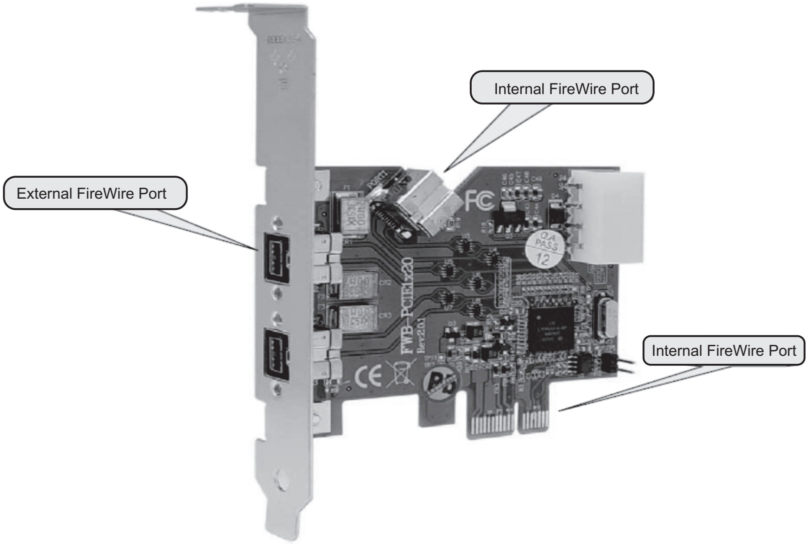


Fig. 5.4 Firewire host interface adaptor

Architecture: The following reference from the 1394-2008 standard, describes the 1394 "Module Architecture". The serial bus architecture is defined in terms of nodes. A node is an addressable entity, which can be independently reset and identified. More than one node may reside on a single module, and more than one unit may reside in a single node, as illustrated in Fig. 5.5.

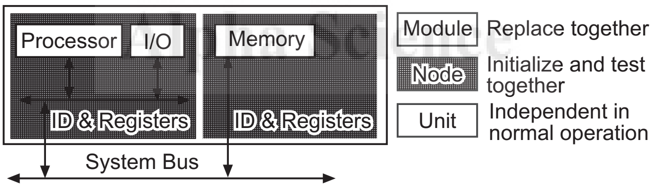


Fig. 5.5 Module architecture

Each module consists of one or more nodes, which are independently initialized and configured. Note that modules are a physical packaging concept and nodes are a logical addressing concept. A module is a physical device, consisting of one or more nodes that share a physical interface. In normal operation, a module is not visible to software.

A node is a logical entity with a unique address. It provides an identification ROM and a standardized set of control registers, and it can be reset independently. In practice the terms "1394 device" and "1394 node" are interchangeable.

FireWire Applications:

- **Robotic Control:** Such systems use FireWire cameras as "environmental sensors". The computer "sees" the environment through FireWire cameras, performs image analysis and provides movement instructions to robotic "hands" that need to interact with the environment (pick up objects, place objects into new positions, etc).

- **Automated Optical Inspection:** Such systems use FireWire cameras to take "photos" of products manufactured in automated assembly lines (e.g. PC motherboards, PC adapters, cell phones, etc) and examine whether the said artifacts appear as if they have been constructed properly (i.e. they have no visible discrepancies).
- **Medical Imaging:** Such systems might use FireWire cameras to create 3-dimensional models of a patient's face or body. These models are then used in various ways through the medical procedures.
- **Filming:** Such systems might use arrays of FireWire cameras to create special 3-dimensional visual recordings used in special visual effects.
- Security Surveillance: FireWire cameras are used to monitor places of interest for security reasons.
- **Storage:** High performance external hard disks for storage and backup.
- **Communication Systems:** FireWire is used as an internal local network in data centers for high speed server-to-server communications.
- **Audio & Pro-Audio:** Specialized audio applications like amplifier/speaker control, audio channel routing/mixing, audio stream delivery for theatre systems and concerts, etc, are done with FireWire.
- **Set-Top Box:** By FCC regulation all set-top boxes in the USA are required to have a functional FireWire port to permit recording of digital content.
- **Digital Camcorders:** Many digital camcorders provide a FireWire port to allow easy connectivity to the user's personal computer.
- **Commercial Aviation:** Such systems use FireWire as a high-capacity local network that can also provide the "Quality of Service" (QoS) required for on demand video streaming in in-flight entertainment systems.
- **Military:** FireWire is being used as a reliable, high-capacity local network that carries control information and sensor data all over a military aircraft, helicopter or vehicle.
- **Automotive:** FireWire is making strong efforts to get established as the "in car" communications network for modern and future cars, where services like on demand video, TV, music will be available per passenger seat.

5.5 UNIVERSAL SERIAL BUS (USB)

Universal Serial Bus (USB) is an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication and power supply between computers and electronic devices. USB is not a true bus, meaning only the root hub sees every signal on the bus. This implies there is no method to monitor upstream communications from a down stream device.

USB was designed to standardize the connection of computer peripherals, such as keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters to personal computers, both to communicate and to supply electric power. USB has effectively replaced a variety of earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices. The symbol of USB is shown in Fig. 5.6.

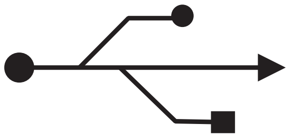


Fig. 5.6 USB symbol

The design architecture of USB is asymmetrical in its topology, consisting of a host, a multitude of downstream USB ports, and multiple peripheral devices connected in a tiered-star topology. Additional USB hubs may be included in the tiers, allowing branching into a tree structure with up to five tier levels. A USB host may implement multiple host controllers and each host controller may provide one or more USB ports. Up to 127 devices, including hub devices if present may be connected to a single host controller. USB devices are linked in series through hubs. One hub is known as the root hub which is built into the host controller. A typical USB system layout is shown in Fig. 5.7.

A physical USB device may consist of several logical sub-devices that are referred to as device functions. A single device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). This kind of device is called composite device. An alternative for this is compound device in which each logical device is assigned a distinctive address by the host and all logical devices are connected to a built-in hub to which the physical USB wire is connected.

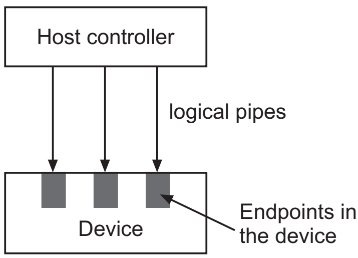


Fig. 5.7 USB system layout

USB endpoints actually reside on the connected device: the channels to the host are referred to as pipes USB device communication is based on pipes (logical channels). A pipe is a connection from the host controller to a logical entity, found on a device, and named an endpoint. Because pipes correspond 1-to-1 to endpoints, the terms are sometimes used interchangeably. A USB device can have up to 32 endpoints, though USB devices seldom have this many endpoints. An endpoint is built into the USB device by the manufacturer and therefore exists permanently, while a pipe may be opened and closed.

There are two types of pipes: stream and message pipes. A message pipe is bidirectional and is used for control transfers. Message pipes are typically used for short, simple commands to the device, and a status response, used, for example, by the bus control pipe number 0. A stream pipe is a unidirectional pipe connected to a unidirectional endpoint that transfers data using an isochronous, interrupt, or bulk transfer:

- Isochronous transfers: at some guaranteed data rate (often, but not necessarily, as fast as possible) but with possible data loss (e.g., real-time audio or video).
- Interrupt transfers: devices that need guaranteed quick responses (bounded latency) (e.g., pointing devices and keyboards).
- Bulk transfers: large sporadic transfers using all remaining available bandwidth, but with no guarantees on bandwidth or latency (e.g., file transfers).

An endpoint of a pipe is addressable with a tuple (device_address, endpoint_number) as specified in a TOKEN packet that the host sends when it wants to start a data transfer session. If the direction of the data transfer is from the host to the endpoint, an OUT packet (a specialization of a TOKEN packet) having the desired device address and endpoint number is sent by the host. If the direction of the data transfer is from the device to the host, the host sends an IN packet instead. If the destination endpoint is a uni-directional endpoint whose manufacturer's designated direction does not match the TOKEN packet (e.g., the manufacturer's designated direction is IN while the TOKEN packet is an OUT packet), the TOKEN packet will be ignored. Otherwise, it will be accepted and the data transaction can start. A bi-directional endpoint, on the other hand, accepts both IN and OUT packets.

Endpoints are grouped into interfaces and each interface is associated with a single device function. An exception to this is endpoint zero, which is used for device configuration and which is not associated with any interface. A single device function composed of independently controlled interfaces is called a composite device. A composite device only has a single device address because the host only assigns a device address to a function.

When a USB device is first connected to a USB host, the USB device enumeration process is started. The enumeration starts by sending a reset signal to the USB device. The data rate of the USB device is determined during the reset signaling. After reset, the USB device's information is read by the host and the device is assigned a unique 7-bit address. If the device is supported by the host, the device drivers needed for communicating with the device are loaded and the device is set to a configured state. If the USB host is restarted, the enumeration process is repeated for all connected devices.

The host controller directs traffic flow to devices, so no USB device can transfer any data on the bus without an explicit request from the host controller. In USB 2.0, the host controller polls the bus for traffic, usually in a round-robin fashion. The throughput of each USB port is determined by the slower speed of either the USB port or the USB device connected to the port.

High-speed USB 2.0 hubs contain devices called transaction translators that convert between high-speed USB 2.0 buses and full and low speed buses. When a high-speed USB 2.0 hub is plugged into a high-speed USB host or hub, it will operate in high-speed mode. The USB hub will then either use one transaction translator per hub to create a full/low-speed bus that is routed to all full and low speed devices on the hub, or will use one transaction translator per port to create an isolated full/low-speed bus per port on the hub.

Because there are two separate controllers in each USB 3.0 host, USB 3.0 devices will transmit and receive at USB 3.0 data rates regardless of USB 2.0 or earlier devices connected to that host. Operating data rates for them will be set in the legacy manner.

USB 1: The Universal Serial Bus Specification Revision 1.0 was released in January 1996. USB capability first became available on PCs with the release of Windows 95's. The USB support in these versions was limited and buggy, and there weren't many USB peripherals available, so use of USB was limited in this era. The situation improved with the release of Windows 98 in June 1998. By this time, many more vendors had USB peripherals available, and USB began to take hold as a popular interface.

USB 2: As USB gained in popularity and PCs became more powerful, demand grew for a faster bus speed. April 2000 saw the release of the Universal Serial Bus Specification Revision 2.0, which added high speed at 480 Mbps. High speed made USB more attractive for peripherals such as printers, disk drives, and video cameras. Windows added support for USB 2.0 in Windows XP SP2. The USB 2.0 specification replaced USB 1. USB 2.0 is backwards compatible with USB 1.x. In other words, USB 2.0 devices can use the same connectors and cables as 1.x devices, and a USB 2.0 device works when connected to a PC that supports USB 1.x or USB 2.0, except for a few devices that function only at high speed and thus require USB 2.0 support.

When USB 2.0 devices first became available, there was confusion among users about whether all USB 2.0 devices supported high speed. To reduce confusion, the USB-IF released naming and packaging recommendations that emphasize speed and compatibility rather than USB version numbers. A product that supports high speed should be labelled "Hi-Speed USB," and messages on the packaging might include Fully compatible with Original USB and Compatible with the USB 2.0 Specification. For products that support low or full speed only, the recommended messages on packaging are Compatible with the USB 2.0 Specification and Works with USB and Hi-Speed USB systems, peripherals and cables.

USB 3: The Universal Serial Bus 3.0 Specification Revision 1.0 was released in November 2008, with the first USB 3.0 device-controller hardware expected to follow about a year later. USB 3.0 defines a new dual-bus architecture with two physical buses that operate in parallel. USB 3.0 provides a pair of wires for USB 2.0 traffic and additional wires to support the new Super-Speed bus at 5 Gbps. Super-Speed offers a more than 10× increase over USB

2.0's high speed. Plus, unlike USB 2.0, Super-Speed has a pair of wires for each direction and can transfer data in both directions at the same time. USB 3.0 also increases the amount of bus current devices can draw and defines protocols for more aggressive power saving and more efficient transfers. USB 3.0 is backwards compatible with USB 2.0. USB 3.0 hosts and hubs support all four speeds. USB 2.0 cables fit USB 3.0 receptacles.

Power: The USB 1.x and 2.0 specifications provide a 5 V supply on a single wire from which connected USB devices may draw power. The specification provides for no more than 5.25 V and no less than 4.75 V (5 V±5%) between the positive and negative bus power lines. For USB 3.0, the voltage supplied by low-powered hub ports is 4.45–5.25 V.

A unit load is defined as 100 mA in USB 2.0, and 150 mA in USB 3.0. A device may draw a maximum of 5 unit loads (500 mA) from a port in USB 2.0; 6 (900 mA) in USB 3.0. There are two types of devices: low-power and high-power. A low-power device draws at most 1 unit load, with minimum operating voltage of 4.4 V in USB 2.0, and 4 V in USB 3.0. A high-power device draws at most the maximum number of unit loads permitted by the standard. Every device functions initially as low-power but the device may request high-power and will get it if the power is available on the providing bus.

Some devices, such as high-speed external disk drives, require more than 500 mA of current and therefore may have power issues if powered from just one USB 2.0 port: erratic function, failure to function, or overloading/damaging the port. Such devices may come with an external power source or a Y-shaped cable that has two USB connectors (one for power+data, the other for power only) to be plugged into a computer. With such a cable, a device can draw power from two USB ports simultaneously.

A bus-powered hub initializes itself at 1 unit load and transitions to maximum unit loads after it completes hub configuration. Any device connected to the hub will draw 1 unit load regardless of the current draw of devices connected to other ports of the hub (i.e. one device connected on a four-port hub will draw only 1 unit load despite the fact that more unit loads are being supplied to the hub).

A self-powered hub will supply maximum supported unit loads to any device connected to it. In addition, the V_{BUS} will present 1 unit load upstream for communication if parts of the Hub are powered down.

Communication: During USB communication data is transmitted as packets. Initially, all packets are sent from the host, via the root hub and possibly more hubs, to devices. Some of those packets direct a device to send some packets in reply.

After the sync field, all packets are made of 8-bit bytes, transmitted least-significant bit first. The first byte is a Packet Identifier (PID) byte. The PID is actually 4 bits; the byte consists of the 4-bit PID followed by its bitwise complement. This redundancy helps detect errors. (Note also that a PID byte contains at most four consecutive 1 bits, and thus will never need bit-stuffing, even when combined with the final 1 bit in the sync byte. However, trailing 1 bit in the PID may require bit-stuffing within the first few bits of the payload.)

Handshake packets: Handshake packets consist of a PID byte, and are generally sent in response to data packets. The three basic types are ACK, indicating that data was successfully received, NAK, indicating that the data cannot be received and should be retried, and STALL, indicating that the device has an error condition and will never be able to successfully transfer data until some corrective action (such as device initialization) is performed.

USB 2.0 added two additional handshake packets, NYET which indicates that a split transaction is not yet complete. A NYET packet is also used to tell the host that the receiver has accepted a data packet, but cannot accept any more due to buffers being full. The host will then send PING packets and will continue with data packets once the device ACK's the PING. The other packet added was the ERR handshake to indicate that a split transaction failed. The only handshake packet the USB host may generate is ACK; if it is not ready to receive data, it should not instruct a device to send any.

Token packets: Token packets consist of a PID byte followed by 2 payload bytes: 11 bits of address and a 5-bit CRC. Tokens are only sent by the host, never a device. IN and OUT tokens contain a 7-bit device number and 4-bit function number (for multifunction devices) and command the device to transmit DATAx packets, or receive the following DATAx packets, respectively.

An IN token expects a response from a device. The response may be a NAK or STALL response, or a DATAx frame. In the latter case, the host issues an ACK handshake if appropriate. An OUT token is followed immediately by a DATAx frame. The device responds with ACK, NAK, NYET, or STALL, as appropriate.

SETUP operates much like an OUT token, but is used for initial device setup. It is followed by an 8-byte DATA0 frame with a standardized format.

Every millisecond (12000 full-bandwidth bit times), the USB host transmits a special SOF (start of frame) token, containing an 11-bit incrementing frame number in place of a device address. This is used to synchronize isochronous data flows. High-bandwidth USB 2.0 devices receive 7 additional duplicate SOF tokens per frame, each introducing a 125 μs "microframe" (60000 high-bandwidth bit times each).

USB 2.0 added a PING token, which asks a device if it is ready to receive an OUT/DATA packet pair. The device responds with ACK, NAK, or STALL, as appropriate. This avoids the need to send the DATA packet if the device knows that it will just respond with NAK.

USB 2.0 also added a larger 3-byte SPLIT token with a 7-bit hub number, 12 bits of control flags, and a 5-bit CRC. This is used to perform split transactions. Rather than tie up the high-bandwidth USB bus sending data to a slower USB device, the nearest high-bandwidth capable hub receives a SPLIT token followed by one or two USB packets at high bandwidth, performs the data transfer at full or low bandwidth, and provides the response at high bandwidth when prompted by a second SPLIT token.

Data packets: A data packet consists of the PID followed by 0–1,023 bytes of data payload (up to 1,024 in high bandwidth, at most 8 at low bandwidth), and a 16-bit CRC.

There are two basic forms of data packet, DATA0 and DATA1. A data packet must always be preceded by an address token, and is usually followed by a handshake token from the receiver back to the transmitter. The two packet types provide the 1-bit sequence number required by Stop-and-wait ARQ. If a USB host does not receive a response (such as an ACK) for data it has transmitted, it does not know if the data was received or not; the data might have been lost in transit, or it might have been received but the handshake response was lost. To solve this problem, the device keeps track of the type of DATAx packet it last accepted. If it receives another DATAx packet of the same type, it is acknowledged but ignored as a duplicate. Only a DATAx packet of the opposite type is actually received.

When a device is reset with a SETUP packet, it expects an 8-byte DATA0 packet next. USB 2.0 added DATA2 and MDATA packet types as well. They are used only by high-bandwidth devices doing high-bandwidth isochronous transfers which need to transfer more than 1024 bytes per 125 μs microframe (8,192 kB/s).

PRE packet: Low-bandwidth devices are supported with a special PID value, PRE. This marks the beginning of a low-bandwidth packet, and is used by hubs which normally do not send full-bandwidth packets to low-bandwidth devices. Since all PID bytes include four 0 bits, they leave the bus in the full-bandwidth K state, which is the same as the low-bandwidth J state. It is followed by a brief pause during which hubs enable their low-bandwidth outputs, already idling in the J state, then a low-bandwidth packet follows, beginning with a sync sequence and PID byte, and ending with a brief period of SE0. Full-bandwidth devices other than hubs can simply ignore the PRE packet and its low-bandwidth contents, until the final SE0 indicates that a new packet follows.

5.6 SOUND CARD

A sound card is a device installed in the computer that processes information from a computer program, such as a media player, and then sends a signal to the computer's speakers that can hear. Some motherboards have sound capabilities built into them, but computers often have a dedicated sound card, which usually improves the quality of the sound the computer can produce.

A duplex sound card is a sound card capable of both playing and recording sounds at the same time. Most current sound cards are duplex sound cards. A sound card is rectangular in shape with numerous contacts on the bottom of the card and multiple ports on the side for connection to audio devices such as speakers. The sound card installs in a PCI slot on the motherboard. Since the motherboard, case and peripheral cards are designed with compatibility in mind, the side of the sound card fits just outside the back of the case when installed, making its ports available for use. Most sound cards have ports for a joystick, speaker, microphone and an auxiliary device. Still other cards may have inputs and outputs designed for more advanced tasks such as audio editing and professional audio output.

5.7 GRAPHICS CARD

A graphics card (also called a display card, graphics board, display adapter or graphics adapter) is an expansion card which generates a feed of output images to a display. Most video cards offer various functions such as accelerated rendering of 3D scenes and 2D graphics, MPEG-2/MPEG-4 decoding, TV output, or the ability to connect multiple monitors.

Video hardware can be integrated into the motherboard or (as with more recent designs) the CPU, but all modern motherboards provide expansion ports to which a video card can be attached. In this configuration, it is sometimes referred to as a video controller or graphics controller. Modern low-end to mid-range motherboards often include a graphics chipset manufactured by the developer of the motherboard. This graphics chip usually has a small quantity of embedded memory and takes some of the system's main RAM, reducing the total RAM available. This is usually called integrated graphics or on-board graphics, and is usually low in performance and undesirable for those wishing to run 3D applications. A dedicated graphics card on the other hand has its own RAM and Processor specifically for processing video images, and thus offloads this work from the CPU and system RAM.

Components: A modern video card consists of a printed circuit board on which the following components are mounted.

Graphics Processing Unit: A GPU is a dedicated processor optimized for accelerating graphics. The processor is designed specifically to perform floating-point calculations, which are fundamental to 3D graphics rendering and 2D picture drawing. The main attributes of the GPU are the core clock frequency, which typically ranges from 250 MHz to 4 GHz and the number of pipelines (vertex and fragment shaders), which translate a 3D image characterized by vertices and lines into a 2D image formed by pixels. Modern GPUs are massively parallel, and fully programmable. Their computing power is orders of magnitude greater than that of CPUs for certain kinds of operations. This has led to the emergence of general-purpose computing on graphics processing units.

Heat Sink: A heat sink is mounted on high performance graphics cards. A heat sink spreads out the heat produced by the graphics processing unit evenly throughout the heat sink and unit itself. The heat sink commonly has a fan mounted as well to cool the heat sink and the graphics processing unit.

Video BIOS: The video BIOS or firmware contains the basic program, which is usually hidden, that governs the video card's operations and provides the instructions that allow the computer and software to interact with the card. It may contain information on the memory timing, operating speeds and voltages of the graphics processor, RAM, and other information. It is sometimes possible to change the BIOS (e.g. to enable factory-locked settings for higher performance), although this is typically only done by video card over-clockers and has the potential to irreversibly damage the card.

Video memory: The memory capacity of most modern video cards ranges from 128 MB to 8 GB. Since video memory needs to be accessed by the GPU and the display circuitry, it often uses special high-speed or multi-port memory, such as VRAM, WRAM, SGRAM, etc. Around 2003, the video memory was typically based on DDR technology. During and after that year, manufacturers moved towards DDR2, GDDR3, GDDR4 and GDDR5. The effective memory clock rate in modern cards is generally between 400 MHz and 3.8 GHz. Video memory may be used for storing other data as well as the screen image, such as the Z-buffer, which manages the depth coordinates in 3D graphics, textures, vertex buffers, and compiled shader programs.

RAMDAC: The RAMDAC, or Random Access Memory Digital-to-Analog Converter, converts digital signals to analog signals for use by a computer display that uses analog inputs such as Cathode ray tube (CRT) displays. The RAMDAC is a kind of RAM chip that regulates the functioning of the graphics card. Depending on the number of bits used and the RAMDAC-data-transfer rate, the converter will be able to support different computer-display refresh rates. With CRT displays, it is best to work over 75 Hz and never under 60 Hz, in order to minimize flicker. (With LCD displays, flicker is not a problem.) Due to the growing popularity of digital computer displays and the integration of the RAMDAC onto the GPU die, it has mostly disappeared as a discrete component. All current LCDs, plasma displays and TVs work in the digital domain and do not require a RAMDAC.

5.8 BLU-RAY

Blu-ray Disc is the ultimate next generation optical packaged media format. The name "Blu-ray" came from the fact that the laser beam which reads the data from the new discs is blue instead of red, which is used for current DVDs and CDs. This new blue laser is at the heart of Blu-ray Disc Technology (i.e. blue ray of light) and has a shorter wavelength than a red laser, which makes it possible to read data with greater precision. Blu-Ray offering the following key advantages:

- Large capacity (25 GB single layer – 50 GB dual layer) – with over 5 times the amount of content possible with current DVDs, particularly well suited for high definition feature films with extended levels of interactive features.
- Pristine picture quality with full 1080p high definition resolution (1920 × 1080p).
- Best audio possible with crystal clear master quality audio via high definition audio codecs (it can be better than theater sound), up to 7.1 channels of audio.
- Enhanced Interactivity – seamless menu navigation, exciting new high def bonus features, BD-Live internet-connected capabilities
- Broadest industry support – ensuring more choice for consumers in the marketplace.
- Hardware is backward compatible with DVDs, allowing continued enjoyment of existing DVD libraries.

- Robustness of Disc – new breakthroughs in hard coating technologies mean that Blu-ray Disc offers the strongest resistance to scratches and finger prints on a bare disc the same size as existing CD and DVD (i.e. no cartridge is required).

DVD is meant to a 5 to 10 times increase in storage capacity compared to CD, Blu-ray Disc represents an increase over DVD capacity by 5 to 10 times. This is due, among other reasons, to the usage of a blue instead of a red laser and improved lens specifications, allowing for a much smaller focus laser beam which enables the recording of much smaller and higher density pits on the disc. Hence, more precision and ultra high storage densities are made possible.

Due to the fact that the data layer on a Blu-ray Disc is placed much "closer" to the laser lens than in DVD, there is less distortion resulting in significantly improved tolerances. The Blu-ray Disc system uses the same 12 cm and 8 cm disc sizes to support backward-compatibility with your CDs and DVDs and other optical disc formats. Subsequently, all Blu-ray Disc hardware products are backward-compatible.

BD-Live: BD-Live is a recently developed Blu-ray feature that enables you to access content via your internet-connected Blu-ray player. BD-Live allows you to download a variety of up-to-date content (e.g., refreshed previews, ringtones, exclusive special features and events), and is an entirely new experience for interactive content. BD-Live is available on all Profile 2.0 enabled Blu-ray players.

Blu-ray Disc supports all the traditional DVD audio codecs, Dolby Digital, DTS, Linear PCM, but also adds more advanced codecs, Dolby TrueHD, DTS-MA and multi-channel LPCM uncompressed audio.

BD+: A technology which is unique to Blu-ray, BD+ is an additional, optional layer of content protection that is independent from AACS, the fundamental content protection technology for Blu-ray Disc. It provides a way for content providers to stop playback of BD+ protected titles on players deploying known "hacks" that attempt to defeat the player's security system. On the other hand, if known hacks are not found on a given model of BD player, then playback can proceed undisturbed. This approach is very consumer friendly because it does not target innocent users who happen to own a known compromised model.

ROM mark: Another technology unique to Blu-ray, ROM Mark is a hidden undetectable mark digitally imprinted by a licensed replicator which can only be read by a licensed BD player or computer drive. A ROM Mark will exist on movies, games and other data discs to help prevent mass production piracy and sale of unauthorized copies. It will be a unique identifier on each pressed disc that can link it back to the very machine that produced it. This will allow customs enforcement to determine between real and counterfeit copies of copyrighted works. Special players will be able to check the code on the disc, where it should be and where it is detected.

Blu-ray Disk types:

- BD-ROM - read-only format for software, games and movie distribution.
- BD-R - write-once recordable format for HDTV recording and PC data storage.
- BD-RE - rewritable format for HDTV recording and PC data storage.

Disc format: The recording area on the disc should be separated into an area for the metadata and database files, and a different area for real-time data recordings as shown in Fig. 5.8. The files recorded in the area for metadata and database files can be read with a fewer number of seeks, reducing the response time during PlayList editing and menu display, resulting in greatly improved system response.

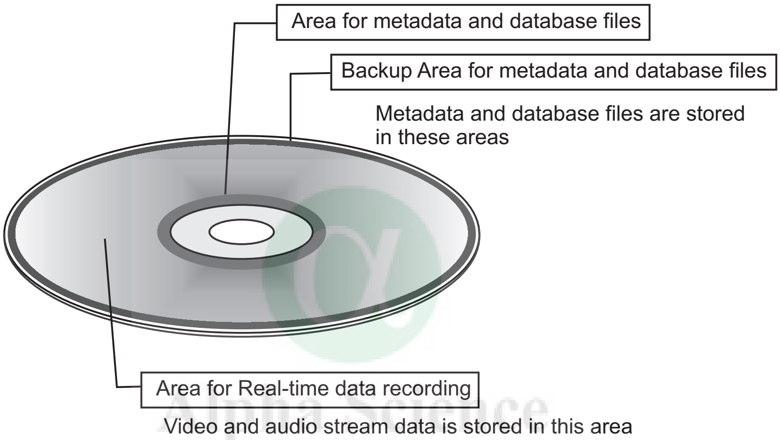


Fig. 5.8 Recording area for different file types

An additional concern for rewritable media is its cyclability, which is the number of times any area of the disc can be overwritten and depends on the physical properties of the media. If a recording system frequently rewrites only a particular area, that particular area will wear out before the rest of the media. This can occur when a recording system continually uses and reuses areas near the inner perimeter, instead of spreading new recordings across the radius of the disc. Circular recording methods solve this problem by using a system which uses the free spaces on the disc uniformly.

Continuous Data Supply by Expanses: When recording, deleting or editing operations are performed repeatedly, small areas of empty space will occur across the disc. These small areas can be used to record a new Real-Time file, which results in a single Real-Time file composed of many small extents scattered across the disc. A group of these extents, each of which is recorded on contiguous logical sectors, is called an Expanse1 (Fig. 5.9). The Expanse is conceptually a contiguous area to be read, and may include small areas in which Real-Time data is not recorded.

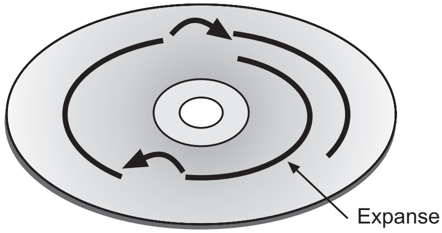


Fig. 5.9 Expanse

When a single Real-Time file is made up of several expanses, the file is read by jumping from one expanse to the next and reading the expanses in order. However, when jumping from one expanse to another, the disc rotation speed needs to be changed and the optical pickup needs to be moved to a different radius on the disc. Although data cannot be retrieved from the disc during this interval, the decoding/playback of video/audio data must continue without interruption. So to prevent any breaks occurring during playback, a mechanism is required to absorb the effects of the access intervals that occur between expanses. To prevent interruption in video/audio playback while reading data from the disc, the buffer memory must not be emptied of data before readout from the next expanse becomes possible. This requirement for continuous supply of data is necessary to insure seamless playback. Therefore the minimum expanse size is defined such that the buffer memory does not become empty when jumping from one expanse to another on the disc. The Blu-ray UDF file system specifications specify a larger minimum expanse size than Blu-ray BDFS to enable the design of cost effective portable players, as well as simultaneous recording and playback recorders.

5.9 BLUETOOTH

Bluetooth is a wireless protocol utilizing short-range communications technology facilitating data transmissions over short distances from fixed and/or mobile devices, creating wireless personal area networks (PANs). The intent behind the development of Bluetooth was the creation of a single digital wireless protocol, capable of connecting multiple devices and overcoming issues arising from synchronization of these devices. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics.

Principle of operation: Bluetooth radios operate in the unlicensed 2.4GHz Industrial, Scientific, and Medical application (ISM) frequency range. This frequency is already widely used by devices such as microwave ovens, baby monitors, cordless telephones, and 802.11b/g wireless networking devices In order to avoid interference from these devices; Bluetooth uses a technology called spread spectrum frequency hopping. Spread spectrum frequency hopping changes the transmission frequency up to 1600 times per second across 79 different frequencies. As a result, interference on any one of those frequencies will only last a fraction of a second. (Franklin 4) This, coupled with the limited range of Bluetooth radio transmitters, results in a robust signal that is highly tolerant of other devices sharing the same frequency.

Bluetooth devices automatically attempt to communicate whenever one device comes within range of another. Bluetooth devices discover each other and initiate communication via inquiry and page transmissions. If device A would like to connect to device B, it will broadcast a page message progressively across the Bluetooth frequency range. Before device B establishes a connection, it will wait in standby mode listening for page and inquiry messages. In order for the page message to succeed in establishing communications, device A must know device B's Bluetooth address and system clock settings. If device A does not have device B's address, it will broadcast an inquiry message requesting this information. Device B will respond with the correct information and communication is then initiated with a page message.

Bluetooth devices have the ability to form ad hoc networks. The topology of these networks is both temporary and random. An ad hoc network of two or more Bluetooth devices is called a piconet. When two Bluetooth devices initiate a connection, they automatically determine if one device needs to control the other. Generally, the device that initiates the communication assumes the role of master and exercises certain controls over the other members of the piconet which are known as slaves. Upon establishing a piconet, the slave devices synchronize their frequency hopping sequence and system clock with that of the master in order to maintain their connection. A master device can have up to seven slaves. A slave in one piconet can also be the master in another, thus allowing piconets to overlap and interact forming what is known as a scatternet.

Classification of Bluetooth

Class	Maximum Permitted Power mW(dBm)	Range (approximate)
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

Bluetooth versions: Versions 1.0 and 1.0B had many problems, and manufacturers had difficulty making their products interoperable. Versions 1.0 and 1.0B also included mandatory Bluetooth hardware device address (BD_ADDR) transmission in the connecting process (rendering anonymity impossible at the protocol level), which was a major setback for certain services planned for use in Bluetooth environments.

Bluetooth 1.1:

- Ratified as IEEE Standard 802.15.1-2002.
- Many errors found in the 1.0B specifications were fixed.
- Added support for non-encrypted channels.
- Received Signal Strength Indicator (RSSI).

Bluetooth 1.2: This version is backward-compatible with 1.1 and the major enhancements include the following:

- Faster Connection and Discovery
- Adaptive frequency-hopping spread spectrum (AFH), which improves resistance to radio frequency interference by avoiding the use of crowded frequencies in the hopping sequence.
- Higher transmission speeds in practice, up to 721 kbit/s, as in 1.1.
- Extended Synchronous Connections (eSCO), which improve voice quality of audio links by allowing retransmissions of corrupted packets, and may optionally increase audio latency to provide better support for concurrent data transfer.
- Host Controller Interface (HCI) support for three-wire UART.
- Ratified as IEEE Standard 802.15.1-2005.

Bluetooth 2.0: This version is backward-compatible with 1.1. The main enhancement is the introduction of an Enhanced Data Rate (EDR) for both data (ACL) and voice (eSCO) packets. The nominal signalling rate of EDR is about 3 megabits per second, although the practical data transfer rate is 2.1 megabits per second. This additional throughput is obtained by using a different modulation scheme for radio transmission of the data payload. Standard or Basic Rate transmission uses the Gaussian Frequency Shift Keying (GFSK) method, while EDR uses a combination of GFSK and Phase Shift Keying (PSK).

According to the 2.0 specification, EDR provides the following benefits:

- Three times faster transmission speed — up to 10 times in certain cases (up to 2.1 Mbit/s).
- Lower power consumption through a reduced duty cycle.
- Simplification of multi-link scenarios due to more available bandwidth.

Bluetooth 2.1: Bluetooth Core Specification Version 2.1 is fully backward-compatible with 1.1. This specification includes the following features:

- Extended inquiry response: provides more information during the inquiry procedure to allow better filtering of devices before connection. This information includes the name of the device, a list of services the device supports, as well as other information like the time of day, and pairing information.
- Sniff subrating: reduces the power consumption when devices are in the sniff low-power mode, especially on links with asymmetric data flows. Human interface devices (HID) are expected to benefit the most, with mouse and keyboard devices increasing the battery life by a factor of 3 to 10. It lets devices decide how long they will wait before sending keep alive messages to one another. Previous Bluetooth implementations featured keep alive message frequencies of up to several times per second. In contrast, the 2.1 specification allows pairs of devices to negotiate this value between them to as infrequently as once every 5 or 10 seconds.

- Encryption Pause Resume: enables an encryption key to be refreshed, enabling much stronger encryption for connections that stay up for longer than 23.3 hours (one Bluetooth day).
- Secure Simple Pairing: radically improves the pairing experience for Bluetooth devices, while increasing the use and strength of security. It is expected that this feature will significantly increase the use of Bluetooth.

Near Field Communication (NFC) cooperation: Automatic creation of secure Bluetooth connections when NFC radio interface is also available. For example, a headset should be paired with a Bluetooth 2.1 phone including NFC just by bringing the two devices close to each other (a few centimeters). Another example is automatic uploading of photos from a mobile phone or camera to a digital picture frame just by bringing the phone or camera close to the frame.

Bluetooth 3.0: The next version of Bluetooth after v2.1, code-named Seattle has many of the same features, but is most notable for plans to adopt ultra-wideband (UWB) radio technology. This will allow Bluetooth use over UWB radio, enabling very fast data transfers of up to 480 Mbit/s, while building on the very low-power idle modes of Bluetooth.

Connection setup: Any Bluetooth device will transmit the following information on demand:

- Device name.
- Device class.
- List of services.
- Technical information, for example, device features, manufacturer, Bluetooth specification used, clock offset.

Any device may perform an inquiry to find other devices to connect to, and any device can be configured to respond to such inquiries. However, if the device trying to connect knows the address of the device, it always responds to direct connection requests and transmits the information shown in the list above if requested. Use of device services may require pairing or acceptance by its owner, but the connection itself can be initiated by any device and held until it goes out of range. Some devices can be connected to only one device at a time, and connecting to them prevents them from connecting to other devices and appearing in inquiries until they disconnect from the other device.

Every device has a unique 48-bit address. However these addresses are generally not shown in inquiries. Instead, friendly Bluetooth names are used, which can be set by the user. This name appears when another user scans for devices and in lists of paired devices.

Most phones have the Bluetooth name set to the manufacturer and model of the phone by default. Most phones and laptops show only the Bluetooth names and special programs that are required to get additional information about remote devices.

Pairing: Pairs of devices may establish a trusted relationship by learning (by user input) a shared secret known as a passkey. A device that wants to communicate only with a trusted device can cryptographically authenticate the identity of the other device. Trusted devices may also encrypt the data that they exchange over the airwaves so that no one can listen in. The encryption can, however, be turned off, and passkeys are stored on the device file system, not on the Bluetooth chip itself. Since the Bluetooth address is permanent, a pairing is preserved, even if the Bluetooth name is changed. Pairs can be deleted at any time by either device. Devices generally require pairing or prompt the owner before they allow a remote device to use any or most of their services. Some devices, such as mobile phones, usually accept OBEX business cards and notes without any pairing or prompts.

Certain printers and access points allow any device to use its services by default, much like unsecured Wi-Fi networks. Pairing algorithms are sometimes manufacturer-specific for transmitters and receivers used in applications such as music and entertainment. Bluetooth 2.1 has an optional "touch-to-pair" feature based on NFC. By simply bringing two devices into close range (around 10cm), pairing can securely take place without entering a passkey or manual configuration.

Bluetooth Security: Bluetooth security is based on three critical services: authentication, authorization, and encryption. The authentication service is tasked with ensuring that a device seeking a connection is indeed who it claims to be. Authorization is the process that determines whether or not a requesting device is allowed access to specific information or services. Encryption helps to ensure confidentiality by protecting private data from being viewed by unintended recipients.

Bluetooth devices can be set in one of three different security modes. In security mode 1, no security measures are utilized. Any other Bluetooth device can access the data and services of a device in security mode 1. Security mode 2 enacts security measures based on authorization. In this mode, different trust levels can be defined for each of the services offered by the device. Security mode 3 requires both authentication and encryption.

Authentication: Bluetooth authenticates devices using secret keys called link keys. There are two different types of link keys: unit keys and combination keys. A device that uses a unit key uses the same link key for all of its connections. A combination key is specific to one pair of Bluetooth devices. Link keys can be generated either dynamically or through a process called pairing. When a device is configured to generate link keys dynamically, it requires the user to enter the passkey each time a connection is established.

Pairing, on the other hand, generates a long-term, stored link key that allows for the simple automated connections that are the hallmark of the Bluetooth specification. In order to pair two devices, the user will set both devices in pairing mode and will then enter a shared passkey. This passkey is then used to generate an initialization key. The initialization key is based on the Bluetooth addresses of the two devices, a random number and the passkey. This initialization key is then used to authenticate each device as well as in the creation of the link key. Finally, the link key is stored locally on each device for future authentication. After the pairing process has completed, the devices will "automatically and transparently authenticate and perform encryption of the link."

Bluetooth authentication is based on a challenge-response process and can be both unidirectional and mutual. The authentication process uses the E1 algorithm which is based on the SAFER+ block cipher. If device A were seeking to be authenticated by device B, it would begin the process by sending its 48 bit Bluetooth address to device B. Device B will then issue a 128 bit random number-based challenge to device A. At this point, both devices will compute an authentication response which is a function of the E1 algorithm and is based on device A's Bluetooth address, the random number challenge issued by device B, and the previously established link key. Device A will then transmit its authentication response and Device B will compare it to its own calculations. If the two agree, then the device is authenticated. If the authentication responses do not match, then the connection is refused. Once the authentication process has completed, device B generates a new random number for its next authentication session.

Authorization: Authorization is the process by which a Bluetooth device determines whether or not another device is allowed access to a particular service. Authorization incorporates two important Bluetooth security concepts: trust relationships and service security levels. Authorization is dependent on authentication as the authentication process establishes the device identity which is used to determine access.

The Bluetooth specification allows for three different levels of trust between devices: trusted, untrusted, and unknown. If device A has a trusted relationship with device B, then device B is allowed unrestricted access to device A. If device B is untrusted, then device B has been previously authenticated, but its access to services on device A is restricted by service security levels. An unknown device has not been authenticated and it is considered untrusted. Service security levels control access to a device's services on a per service basis. The first security service level requires both authentication and authorization in order to grant access to a service. In other words, the identity of the requesting device has to be confirmed and the requesting device has to be granted specific permission to access the service. The second level of service security requires authentication only. At this security level, the identity of the requesting device need only be judged genuine in order to be granted access to the service. The third level requires encryption only. At this level, access to the service will be granted to any device that is encrypting its communications. The last level is open to all devices. An example of a use for this security level would be if a user wanted to grant unrestricted access to a business card stored on the device while restricting access to other, more sensitive services.

Encryption: Bluetooth strives to maintain confidentiality by offering a 128 bit encryption service. By encrypting its transmissions, a Bluetooth device ensures that only a recipient with the proper decryption key can view the data. Bluetooth's encryption uses an algorithm called E0. A device's encryption key is based on its link key. This simplifies the key generation process as both the sender and receiver have shared secret information upon which to key their encryption. Bluetooth's encryption service has three different modes. In mode 1, no encryption is performed. In mode 2, communication with individual devices is encrypted, but broadcast traffic is not. In mode 3, all communications are encrypted.

In addition to reducing interference, Bluetooth's limited range and spread spectrum frequency hopping help to ensure confidentiality by reducing the possibility of eavesdropping.

REVIEW QUESTIONS

1. What is the purpose of input/output devices in a computer system?
2. Name any five input devices.
3. What is the principle of operation of a computer keyboard?
4. Explain how a compute mouse works.
5. List various types of computers.
6. Write a brief note on track-ball.
7. Explain in detail about various types of touch screens.
8. Explain in detail about CRT monitors with the help of necessary diagrams.
9. Compare LCD and CRT monitors.
10. Write a brief note on different types of printers.
11. Write note on Firewire applications.
12. Explain in detail about USB.
13. What is the difference between USB 1 and USB 2.
14. Write short notes on the following:
    - (a) Graphic card.
    - (b) Sound card
    - (c) Blu-ray
    - (d) Bluetooth

CHAPTER

6

COMPUTER SOFTWARE

6.1 INTRODUCTION

In this chapter, your attention will be drawn to understand the 'computer software' and functions within computer systems. Software plays a vital role in computer systems by creating a bridge between the computer hardware and computer users.

We are aware that computers need instruction to carryout tasks, and it cannot perform itself without instructions. When instructions are given to computers in a manner that computers can understand then computers can execute those instructions accordingly. Such instructions to carryout certain task or tasks that are pre-designed when made available in a computer readable format are called programmes. Such programmes are called computer software. These software can be stored in the secondary storage devices and make use of them as and when required. Computer software, or just software, is a collection of computer programs and related data that provides the instructions for telling a computer what to do and how to do it.

In other words, software is a set of programs, procedures, algorithms and its documentation concerned with the operation of a data processing system. Program software performs the function of the program it implements, either by directly providing instructions to the computer hardware or by serving as input to another piece of software. The term was coined to contrast to the old term hardware (meaning physical devices). In contrast to hardware, software "cannot be touched". Software is also sometimes used in a more narrow sense, meaning applicatin software only. Sometimes the term includes data that has not traditionally been associated with computers, such as film, tapes, and records.

6.2 TYPES OF SOFTWARES

Software includes all the various forms and roles that digitally stored data may have and play in a computer (or similar system), regardless of whether the data is used as code for a CPU, or other interpreter, or whether it represents other kinds of information. Software thus encompasses a wide array of products that may be developed using different techniques such as ordinary programming languages, scripting languages, microcode, or an FPGA configuration.

Practical computer systems divide software systems into three major classes: system software, programming software and application softwares. The organization of computer software is shown in Fig. 6.1.

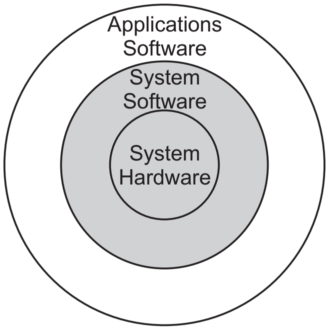


Fig. 6.1 Organization of computer organization

6.2.1 System Software

System software manages the computer system and consists primarily of the operating system. The operating system (OS) is a set of programs that manage the resources of the computer. When the computer is first turned on, it is the operating system that gets things started and presents a user interface that allows the user to choose what she/he wishes to do. The control unit starts fetching instructions from a special kind of memory called read-only memory (ROM). This memory is non-volatile and comes from the manufacturer loaded with a program called the bootstrap loader. This is a simple program that starts loading the operating system from the hard disk into RAM and then instructs the control unit to start fetching instructions of the operating system. The operating system then checks out the system to make sure all components are functioning correctly and presents the user interface. This interface is the so-called desktop, which mimics an office desktop and consists of pictures called icons that symbolize the hard drive, file folders, and programs themselves. When the user indicates that she/he wishes to do word processing, the operating system loads the designated program into memory and then instructs the control unit to fetch instructions from it.

The operating system typically assists the application programs in doing common tasks such as reading from disk or drawing on the screen. It also keeps track of where files are located on the disk and handles the creation and deletion of files. When the user asks a word processing program such as Word to open a file, Word, in turn, asks the operating system to locate the file and load it into memory. When the user is editing the file, Word is simply modifying the copy in memory. This is why, if you don't save the file and your computer crashes or there is a power failure then, you will lose what you have done.

Finally, when the user asks Word to save the file, Word requests this operation of the operating system. When the user quits Word, it instructs the control unit to continue fetching instructions from the operating system, which can then go on to a different task. When the user shuts down the computer, the operating system makes sure everything that must be remembered is written to disk and then shuts down.

6.2.2 Programming Software

Programming software includes tools in the form of programs or applications that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs such as compilers, debuggers, interpreters, linkers, and text editors, that can be combined together to accomplish a task, much as one might use multiple hand tools to fix a physical object. Programming tools are intended to assist a programmer in writing computer programs, and they may be combined in an integrated development environment (IDE) to more easily manage all of these functions.

6.2.3 Application Software

Application programs work with the operating system to apply the computer to specific tasks. The kinds of application programs available are only limited by programmers' imagination and, of course, market conditions. We have already mentioned one of the most common application programs—word processing programs such as Microsoft Word or Corel WordPerfect. These are designed primarily for creating text documents.

Other applications include spreadsheets (as found in Microsoft Excel or Corel Quatro Pro), for doing numerical calculations such as balancing a checkbook and database systems (such as Microsoft Access, Corel Paradox, or Oracle), for keeping track of interrelated data such as student registration and grade information at a university.

Although, complex in their own right, application programs are written to require little knowledge of computer science on behalf of the user. Rather, the user must have significant domain knowledge, that is, knowledge of the area in which the program is applied.

6.3 COMPUTER ALGORITHM

An algorithm is a set of instructions, sometimes called a procedure or a function, that is used to perform a certain task. This can be a simple process, such as adding two numbers together, or a complex function, such as adding effects to an image. For example, in order to sharpen a digital photo, the algorithm would need to process each pixel in the image and determine which ones to change and how much to change them in order to make the image look sharper.

Most computer programmers spend a large percentage of their time creating algorithms. The goal is to create efficient algorithms that do not waste more computer resources (such as RAM and CPU time) than necessary. This can be difficult, because an algorithm that performs well on one set of data may perform poorly on other data.

A poorly written algorithm can cause programs to run slowly and even crash. Therefore, software updates are often introduced, for "improved stability and performance."

6.4 SOFTWARE DEVELOPMENT

Development of software (sometimes called software engineering) involves the analysis of a problem and the design and development of a computer program to apply the computer to that problem. A computer program is an algorithm expressed in a special notation called a programming language and an algorithm is a sequence of steps to achieve a specific task. To be effective, an algorithm must cover all the possibilities that might occur. It must be expressed unambiguously so that it is clear what must be done. The process must also terminate, that is, it cannot go on forever. When we develop programs, we must keep these requirements in mind.

Before a software system can be developed, what is required must be clearly understood. Systematic approach to develop a software involves various phases and they are discussed in the following sections.

Analysis phase: to develop a requirements specification that clearly indicates what is (and sometimes what is not) required of the system.

Design phase: is the phase in software development in which decisions are made about how the software system will be implemented in a programming language.

Coding phase: is the phase of software development in which the classes defined in the design phase are implemented in a programming language.

Testing phase: is the phase of software development in which the implemented classes are executed, individually and in groups, to determine whether they meet the specifications.

Debugging phase: is the phase of software development in which it is determined why the class(es) fail and the problem is corrected.

Production phase: is the phase of software development in which the developed system has been tested and debugged and is made available to the user community.

Maintenance phase: is the phase of software development in which bugs detected in the field are corrected and new features are analyzed and implemented.

6.5 PROGRAM PREPARATION

Once an algorithm has been developed in a high-level programming language, a number of steps must be completed to produce the desired executable code. This is called the edit-compile-link-execute cycle, consisting of four steps.

Step-1: The first step is edit. Here the programmer uses a special program called a program editor (similar to a word processor, but designed for programming languages instead of natural languages) to type in, correct, and save a source (high-level language) program.

Step-2: In the compile phase, a compiler is used to translate the program into object code. Often, the program hasn't been correctly expressed and contains errors in grammar known as syntax errors. If the compiler detects a syntax error, the programmer uses the editor to correct it and then recompiles the program.

Step-3: When the program is free of syntax errors, the linker is used to link the generated object code with library code. If a link error occurs, perhaps because a name has been mistyped, the programmer re-edits the source program, recompiles, and relinks.

Step-4: Once the program is successfully linked, the program is executed to test that it does what is desired. The program may try to do things that are unreasonable (such as divide a number by zero), or it might execute but produce incorrect results. These situations are called execution errors, logic errors, or bugs and must be corrected, resulting in the source program being re-edited, recompiled, relinked, and finally executed again.

This cycle of edit-compile-link-execute continues until the programmer is satisfied that the resulting code works as desired. Since most real-world programs typically are composed of many separately developed pieces of code, the cycle begins again with another piece, and so on until the entire software system is completed.

Today, most programmers use software development environments or interactive development environments (IDEs) to perform the edit-compile-link-execute cycle. The IDE allows the system to be developed as a number of separately created pieces called files. When the programmer has modified one or more pieces, the IDE determines which pieces must be compiled and linked so that the system can be tested. This means that the programmer may not be aware of the complete cycle as it is occurring.

6.6 COMPUTER LANGUAGES

We generally use natural language such as English to express algorithms to other people. But English statements are often ambiguous and rely upon the listener's common sense and world knowledge. Since computers have no common sense, it is necessary to be unambiguous. For that reason, natural languages are not used for programming, but rather specially designed computer programming languages are used instead.

Like computers themselves, computer programming languages have evolved through a number of generations. At the beginning, programmers wrote their programs in machine language and each operation was written as a separate instruction as a sequence of binary digits. These early languages are known as the first-generation languages or machine language. Machine language is the native language of the machine. Since digital computers use 0 and 1 as their alphabet, machine language naturally uses 1s and 0s to encode the instructions.

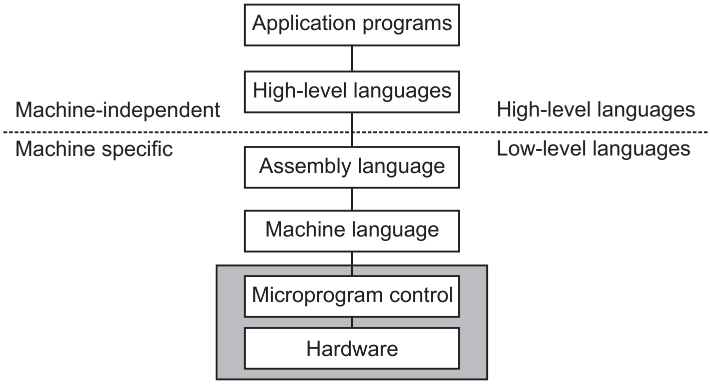


Fig. 6.2 Hierarchy of programming languages

One level up, there is the assembly language as shown in Fig. 6.2. Writing long series of 0's and 1's in machine language was tedious. It was decided that the computer itself could help things if a program could be written that would automatically convert an algorithm written in a symbolic notation into machine language. Each operation (opcode) was given a name and the operands (addresses) were expressed as a combination of names and simple arithmetic operations. These second-generation languages were called assembly languages. In a second-generation language or assembly language, each operation (opcode) is represented by a name and the operands (addresses) are expressed as a combination of names and simple arithmetic operations. Each assembly language instruction still corresponds to one machine operation. Each assembly language instruction still corresponds to one machine operation; the difference from machine language is the use of symbols for the opcodes and addresses.

High-level languages are the computer languages which are relatively closer to the natural language of the humans. Hence these languages must be translated to machine language using either a compiler or a interpreter. Examples for high-level languages are C, C++, FORTRAN etc.

Since the computer does not understand assembly language, running the assembly-language program requires two phases:

1. translation of the assembly program into machine language (called assembly).
2. running of the resulting machine language program (called execution).

The entire process is described in Fig. 6.3. The cylinders represent information stored on disk. The rectangles indicate a machine-language program being executed by the CPU. In the assembly phase, a program called an assembler reads the assembly-language program, and then produces and stores an equivalent machine-language program. In the execution phase, the resulting machine-language program is loaded into memory and executed, reading its data and producing its results. Once the program has been assembled (phase 1), it can be executed (phase 2) any number of times. In fact, the assembler itself may have been originally written in an assembly language and translated into machine language by another assembler.

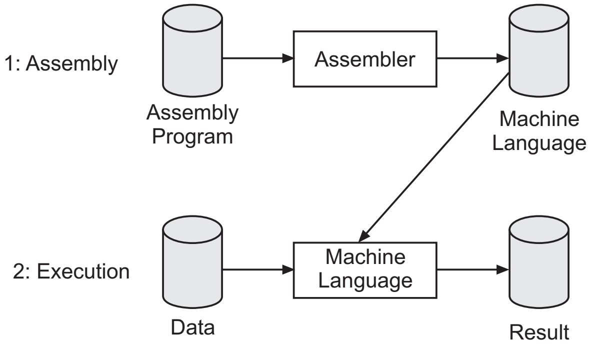


Fig. 6.3 Executing an assembly-language program

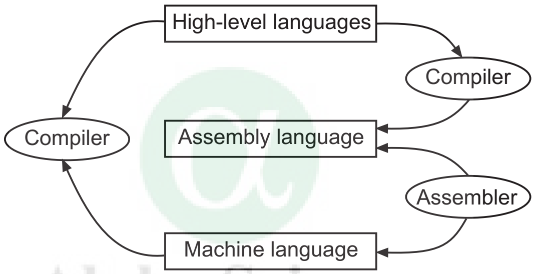


Fig. 6.4 Translation of various levels of languages

Translation of higher-level languages into machine language is done by assemblers and compilers. A compiler can translate a high-level language program directly into the machine language, or it can produce the equivalent assembly language. The scheme is shown in Fig. 6.4. An assembler is the program that reads an assembly language program and produces and stores an equivalent machine language program.

6.7 PROGRAMMING

Programming languages are human-engineered languages developed to convey instructions to machines; they are based on rules of syntax and semantics. Thousands of different programming languages have been developed, used, and discarded. The struggle for survival among programming languages is ruthless. Languages arise, are found wanting, and are soon replaced by more clever or more flexible offspring.

Initially, the evolution of programming languages was driven by a quest for efficient translation of human language to machine code. This produced languages with high levels of abstraction, which hid the hardware and used representations that are convenient and comfortable to human programmers. At some point, another critical aspect of language design became managing the complexity of programs. As programs became larger and more sophisticated, developers realized that some language types were easier to support in large systems. This has lead to greater use of objected-oriented and event driven programming languages.

One of the most fundamental ways programming languages are characterized is by programming paradigm. A programming paradigm provides the programmer's view of code execution. Broadly, programming paradigms can be classified into two - Procedural Programming and Object Oriented programming (OOP).

6.7.1 Procedural Programming

Procedural programming can sometimes be used as a synonym for imperative programming (specifying the steps the program must take to reach the desired state), but can also refer to a programming paradigm, derived from structured programming, based upon the concept of the procedure call. Some good examples of procedural programs are the Linux Kernel, Git, Apache Server, and Quake III Arena.

Procedural programming is also being called as linear programming since code is executed from the top of the file to the bottom. Procedural programming often emphasizes writing code that is as concise as possible and coding directly for the end result. In other words, most procedural programming uses targeted groups of functions that immediately address the problem at hand -- usually nothing more, and nothing less. In most situations, this gets you extremely efficient and high-performance applications. One of the downsides to this approach is a lack of maintainability. If the project grows large enough, the developer or developers could end up having to maintain a large number of individual functions, and in some cases, the logic of different functions can become confusingly similar.

Procedural programming specifies a list of operations that the program must complete to reach the desired state. Each program has a starting state, a list of operations to complete, and an ending point. This approach is also known as imperative programming. Integral to the idea of procedural programming is the concept of a procedure call.

Procedures, also known as functions, subroutines, or methods, are small sections of code that perform a particular function. A procedure is effectively a list of computations to be carried out. The main drawback of procedural programming is that it breaks down when problems get very large. There are limits to the amount of detail one can cope with. Furthermore, some kinds of problem are most easily modelled by non-procedural techniques. A procedural solution may be natural, but a non-procedural one may be more direct.

Structured Programming Languages: Structured programming is a special type of procedural programming. It provides additional tools to manage the problems that larger programs were creating. Structured programming requires that programmers break program structure into small pieces of code that are easily understood. It also frowns upon the use of global variables and instead uses variables local to each subroutine. One of the well known features of structural programming is that it does not allow the use of the GO TO statement. It is often associated with a "top-down" approach to design. The top-down approach begins with an initial overview of the system that contains minimal details about the different parts. Subsequent design iterations then add increasing detail to the components until the design is complete. The most popular structured programming languages include C, Ada, and Pascal.

6.7.2 Object Oriented Programming

In this section, we will have a theoretical approach on Object Oriented programs. Hence, readers are requested to go through some standard text books which are dealing with any of the Object Oriented language such as Java, C# etc.

Object-**O**riented **P**rogramming (OOP) uses a different set of programming languages than procedural programming languages like *C, Pascal*, etc. Everything in OOP is grouped as self sustainable "objects". Hence, you gain re-usability by means of four main object-oriented programming concepts.

In order to clearly understand the object orientation, let's take your "*hand*" as an example. The "*hand*" is a class. Your body has two objects of type hand, named left hand and right hand. Their main functions are controlled/ managed by a set of electrical signals sent through your shoulders (through an interface). So the shoulder is an interface which your body uses to interact with your hands. The hand is a well architected class. The hand is being re-used to create the left hand and the right hand by slightly changing the properties of it.

An object can be considered a "*thing*" that can perform a set of **related** activities. The set of activities that the object performs defines the object's behaviour. For example, the hand can grip something or a *Student* (object) can give the name or address. In pure *OOP* terms an object is an instance of a class.

A *class* is simply a representation of a type of object. It is the blueprint/ plan/template that describe the details of an object. A class is the blueprint from which the individual objects are created. *Class* is composed of three things: a name, attributes, and operations.

In the software world, though you may not have realized it, you have already used classes. For example, the *TextBox* control, you always used, is made out of the *TextBox* class, which defines its appearance and capabilities. Each time you drag a *TextBox* control, you are actually creating a new instance of the *TextBox* class.

A software system may consist of many classes. But in any case, when you have many, it needs to be managed. Think of a big organization, with its work force exceeding several thousand employees (let's take one employee as a one class). In order to manage such a work force, you need to have proper management policies in place. Same technique can be applies to manage classes of your software system as well. In order to manage the classes of a software system, and to reduce the complexity, the system designers use several techniques, which can be grouped under four main concepts named Encapsulation, Abstraction, Inheritance, and *Polymorphism*. These concepts are the four main gods of *OOP* world and in software term, they are called four main Object Oriented Programming (*OOP*) Concepts.

Encapsulation: The encapsulation is the inclusion within a program object of all the resources need for the object to function - basically, the methods and the data. In OOP the encapsulation is mainly achieved by creating classes, the classes expose public methods and properties. The class is kind of a container or capsule or a cell, which encapsulate the set of methods, attribute and properties to provide its indented functionalities to other classes. In that sense, encapsulation also allows a class to change its internal implementation without hurting the overall functioning of the system. That idea of encapsulation is to hide how a class does it but to allow requesting what to do.

In order to modularize/define the functionality of a one class, that class can uses functions/properties exposed by another class in many different ways. According to Object Oriented Programming there are several techniques, classes can use to link with each other and they are named association, aggregation, and composition.

There are several other ways that an encapsulation can be used, as an example we can take the usage of an interface. The interface can be used to hide the information of an implemented class.

Association: Association is a relationship between two classes. It allows one object instance to cause another to perform an action on its behalf. Association is the more general term that define the relationship between two classes, where as the aggregation and composition are relatively special.

Abstraction and Generalization: Abstraction is an emphasis on the idea, qualities and properties rather than the particulars (a suppression of detail). The importance of abstraction is derived from its ability to hide irrelevant details and from the use of names to reference objects. Abstraction is essential in the construction of programs. It places the emphasis on what an object is or does rather than how it is represented or how it works. Thus, it is the primary means of managing complexity in large programs.

While abstraction reduces complexity by hiding irrelevant detail, generalization reduces complexity by replacing multiple entities which perform similar functions with a single construct. Generalization is the broadening of application to encompass a larger domain of objects of the same or different type. Programming languages provide generalization through variables, parameterization, generics and *polymorphism*. It places the emphasis on the similarities between objects. Thus, it helps to manage complexity by collecting individuals into groups and providing a representative which can be used to specify any individual of the group.

Abstraction and generalization are often used together. Abstracts are generalized through parameterization to provide greater utility. In parameterization, one or more parts of an entity are replaced with a name which is new to the entity. The name is used as a parameter. When the parameterized abstract is invoked, it is invoked with a binding of the parameter to an argument.

Abstract class: Abstract classes, which declared with the abstract keyword, cannot be instantiated. It can only be used as a super-class for other classes that extend the abstract class. Abstract class is the concept and implementation gets completed when it is being realized by a subclass. In addition to this a class can inherit only from one abstract class (but a class may implement many interfaces) and must override all its abstract methods/properties and may override virtual methods/properties.

Interface: The Interface separates the implementation and defines the structure, and this concept is very useful in cases where you need the implementation to be interchangeable. Apart from that an interface is very useful when the implementation changes frequently. Some say you should define all classes in terms of interfaces, but I think recommendation seems a bit extreme.

Interface can be used to define a generic template and then one or more abstract classes to define partial implementations of the interface. Interfaces just specify the method declaration (implicitly public and abstract) and can contain properties (which are also implicitly public and abstract). Interface definition begins with the keyword interface. An interface like that of an abstract class cannot be instantiated.

If a class that implements an interface does not define all the methods of the interface, then it must be declared abstract and the method definitions must be provided by the subclass that extends the abstract class. In addition to this an interfaces can inherit other interfaces.

Difference between a Class and an Interface: A *class* and an *interface* are two different types (conceptually). Theoretically a *class* emphasis the idea of encapsulation, while an *interface* emphasis the idea of abstraction (by suppressing the details of the implementation). The two poses a clear separation from one to another. Therefore, it is very difficult or rather impossible to have an effective meaningful comparison between the two, but it is very useful and also meaningful to have a comparison between an interface and an abstract class.

Difference between an Interface and an Abstract class: There are quite big differences between an *interface* and an *abstract class*, eventhough both looks similar.

- Interface definition begins with a keyword interface so it is of type interface.
- Abstract classes are declared with the abstract keyword so it is of type class.
- Interface has no implementation, but they have to be implemented.
- Abstract class's methods can have implementations and they have to be extended.
- Interfaces can only have method declaration (implicitly public and abstract) and fields (implicitly public static).
- Abstract class's methods can't have implementation only when declared abstract.

- Interface can inherit more than one interface.
- Abstract class can implement more than one interface, but can inherit only one class.
- Abstract class must override all abstract method and may override virtual methods.
- Interface can be used when the implementation is changing.
- Abstract class can be used to provide some default behaviour for a base class.
- Interface makes implementation interchangeable.
- Interface increase security by hiding the implementation.
- Abstract class can be used when implementing framework.
- Abstract classes are an excellent way to create planned inheritance hierarchies and also to use as non-leaf classes in class hierarchies.

Abstract classes let you define some behaviors; they force your subclasses to provide others. For example, if you have an application framework, an abstract class can be used to provide the default implementation of the services and all mandatory modules such as event logging and message handling etc. This approach allows the developers to develop the application within the guided help provided by the framework.

Polymorphism: Polymorphism is a generic term that means 'many shapes'. More precisely *Polymorphism* means the ability to request that the same operations be performed by a wide range of different types of things. In OOP, the *polymorphism is* achieved by using many different techniques named method overloading, operator overloading and method overriding.

6.8 OPERATING SYSTEMS

An operating system manages and coordinates the functions performed by the computer hardware, including the CPU, input/output devices, secondary storage devices and communication and network equipments. Operating systems are the most important program that runs on a computer. Every general-purpose computer must have an operating system to run other programs. Operating systems performs basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk and controlling peripheral devices such as disk drives and printers.

The operating system software must keep track of each hardware resource, determine who gets what, determine when the user will have access to the resource, allocate how much of the resource the user will be given and terminate access at the end of use period.

The primary purpose of an operating system is to maximise the productivity of a computer system by operating it in the most efficient manner and minimizing the amount of human intervention required. An operating system also simplifies the job of computer programmers, since it includes programmes that perform common input/output and storage operation and other standard processing functions.

Many operating systems are designed as a collection of programme modules which can be organised in combination with various capabilities around a central module or kernel. Examples of popular micro computer operating system are Windows, DOS, OS/2, for PCs and Mac OS for apple computers etc. an example of an operating system for a main frame is MVS .

6.8.1 Functions of an Operating System

Even the simplest operating system in a microcomputer or mainframe performs a number of resource management tasks or functions. These functions include job management, batch processing, on-line processing, data management, virtual storage and input/output management.

Job management: Job management software manages the jobs waiting to be processed. It recognizes the job, identifies their priorities, determines whether the appropriate main memory and secondary storage capability they require is available and schedules and finally runs each job at the appropriate moment.

Batch processing: System software is available to support the different methods of processing a job. With batch processing, the most basic method, data are accumulated and processed in groups.

On-line processing: In on-line processing, data are processed instantaneously. Using an on-line system the request for information will be instantly acknowledged by the online software and the appropriate steps will be taken to access the central database and return the requested information to the terminal from which the request was made. All of these steps take less than a few seconds at the most. Most online operating system have multi-user multi-tasking capabilities.

Data management: In the process of managing the resources of the computer system, operating system software also manages he storage and retrieval of data. As system software handles many of the details associated with this process, such details are not a primary concern for users or programmers writing application programmes.

Virtual Storage: Operating systems also manages the allocation of main memory to specific jobs. Some operating systems have a feature called virtual storage. With this software it is possible to increase the capacity of main memory without actually increasing its size. This is accomplished by breaking a job into sequences of instructions, called pages or segments and keeping only a few of these in main memory at a time; the reminder are kept on secondary storage devices. As a result, relatively large jobs can be processed by a CPU that in fact contains a relatively small memory.

Input/output management: Operating systems also mange the input to and output from a computer system. This applies to the flow of data among computers, terminals, and other devices such as printers. Application programs use the operating system extensively to handle input and output devices as needed.

6.8.2 Classification of Operating Systems

Operating systems can be classified as follows:

- **Multi-user OS:** Multi-user operating systems allow two or more users to run programme at the same time. Examples are MVS, UNIX etc. Another term for multi user is time sharing.
- **Multi-processing OS:** Multi-processing refers to a computer system's ability to support more than one process (programme) at the same time. Multi-processing operating systems enable several programs to run concurrently MVS and UNIX are two of the most widely used multi-processing systems, but there are many others including OS/2 for high-end PCs. Multi-processing also refers to the utilization of multiple CPUs in a single computer systems. This is also called parallel processing.
- **Multi-tasking OS:** This allows more than one program to run concurrently. It is the ability to execute more than one task at the same time (a task being a programme). The terms multi tasking and multi-processing are often use interchangeably, although multiprocessing some times implies that more than one CPU is involved. In multitasking only one CPU is involved but it switches from one program to another so quickly that it gives the appearance of executing all the programs at the same time. There are two basic types of multitasking: pre-emtive and cooperative.

  In pre-emptive multitasking, the operating system parcels out CPU time slices to each program. In cooperative multi-tasking each program can control the CPU for as long as it needs it. If a program is not using the CPU, however, it can allow another program to use it temporarily. OS/2, Windows NT, Windows 95, Amiga operating system and UNIX uses preemtive multitasking, whereas Microsoft Windows 3.x and the MultiFinder (for Mac computer) use cooperative multitasking.
- **Multi threading OS:** This allows different parts of a single program to run concurrently. Multi threading is the ability of an operating system to execute different parts of a program, called threads, simultaneously.
- **Real time operating systems (RTOS):** These are the OS that responds to input immediately. This category includes operating systems designed substantially for the purpose of controlling and monitoring external activities with timing constrains. They are used for such tasks as navigation, in which the computer must react to a steady flow of new information without interruption. Most general purpose operating system like, windows or UNIX are not real time because they can take a few seconds or even minutes to react. Examples for RTOS are BLMX from national semi-conductors, CCP from IBM, FADOS etc.

6.9 HISTORY AND EVOLUTION OF LINUX

In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues. They developed a new operating system, which was Simple and elegant, Written in the C programming language instead of in assembly code and able to recycle code. The Bell Labs developers named their project "UNIX."

The code recycling features were very important UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language, C. This language was especially developed for creating the UNIX system. Using this new technique, it was much easier to develop an operating system that could run on many different types of hardware. UNIX has been a popular OS for more than two decades because of its multi-user, multi-tasking environment, stability, portability and powerful networking capabilities.

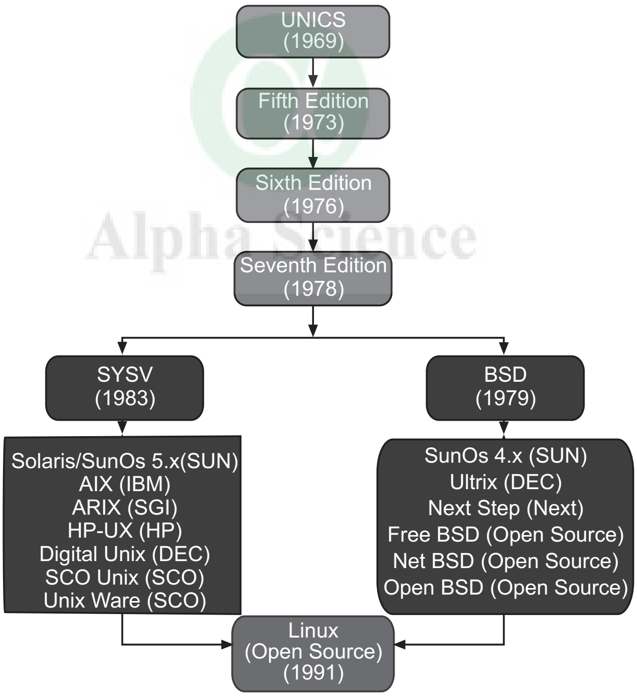


Figure 6.5 History of Linux

6.9.1 Properties of Linux

- **Linux is free:** The license commonly used is the GNU Public License (GPL). The license says that anybody who may want to do so, has the right to change Linux and eventually to redistribute a changed version, on the one condition that the code is still available after redistribution.

- **Linux is portable to any hardware platform:** A vendor who wants to sell a new type of computer and who doesn't know what kind of OS his new machine will run (say the CPU in your car or washing machine), can take a Linux kernel and make it work on his hardware, because documentation related to this activity is freely available.
- **Linux system expects to run without rebooting all the time:** This property allows for Linux to be applicable also in environments where people don't have the time or the possibility to control their systems night and day.
- **Linux is secure and versatile:** The security model used in Linux is based on the UNIX idea of security, which is known to be robust and of proven quality. But Linux is not only fit for use as a fort against enemy attacks from the Internet: it will adapt equally to other situations, utilizing the same high standards for security. Your development machine or control station will be as secure as your firewall.
- **Linux is scalable:** From a Palmtop with 2MB of memory to a petabyte storage cluster with hundreds of nodes: add or remove the appropriate packages and Linux fits all. You don't need a supercomputer anymore, because you can use Linux to do big things using the building blocks provided with the system. If you want to do little things, such as making an operating system for an embedded processor or just recycling your old 486, Linux will do that as well.
- **The Linux OS and most Linux applications have very short debug-times:** Linux has been developed and tested by thousands of people, both errors and people to fix them are usually found rather quickly. It sometimes happens that there are only a couple of hours between discovery and fixing of a bug.

6.9.2 Architecture of the Linux Operating System

The main components of a Linux OS are listed below:

- **Kernel:** The Linux kernel includes device driver support for a large number of PC hardware devices (graphics cards, network cards, hard disks etc.), advanced processor and memory management features.
- **Shells and GUIs:** Linux supports two forms of command input: through textual command line shells similar to those found on most UNIX systems and through graphical interfaces (GUIs) such as the KDE and GNOME window managers.
- **System Utilities:** Virtually every system utility that you would expect to find on standard implementations of UNIX has been ported to Linux. These system utilities are designed to be powerful tools that do a single task extremely well (e.g. grep finds text inside files while wc counts the number of words, lines and bytes inside a file). Users can often solve problems by interconnecting these tools instead of writing a large monolithic application program.

6.9.3 Application Programs of Linux

Linux OS typically come with several useful application programs as standard. Examples include the emacs editor, xv (an image viewer), gcc (a C compiler), g++ (a C++ compiler), xfig (a drawing package), latex (a powerful typesetting language) and soffice (StarOffice, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files).

6.10 MS-DOS (MICROSOFT DISK OPERATING SYSTEM)

MS-DOS (Microsoft Disk Operating System) is an operating system introduced by Microsoft Corp. DOS is considered generally as a command-line OS much similar to UNIX but it gives more versatility in operation. In 1975, Gates and Allen form a partnership called Microsoft. Like most start-ups, Microsoft begins small, but has a huge vision—a computer on every desktop and in every home. In June 1980, Gates and Allen hire Gates' former Harvard classmate Steve Ballmer to help run the company. The next month, IBM approaches Microsoft about a project code-named "Chess." In response, Microsoft focuses on a new operating system—the software that manages, or runs, the computer hardware and also serves to bridge the gap between the computer hardware and programs, such as a word processor. It's the foundation on which computer programs can run. They name their new operating system "MS-DOS." When the IBM PC running MS-DOS ships in 1981, it introduces a whole new language to the general public. Typical DOS commands are given in the following section.

Table 6.1 Simple commands in MS-DOS

dir	lists the contents of a folder
cd	changes folder
cd ..	parent folder
md or mkdir	creates a new folder
deltree	deletes a folder and all sub-folders
copy, xcopy	copies a file
move	moves a file
del	deletes a file
type	displays the contents of a file
type |m ore	displays file contents page by page, pausing after each page
help	help for the given command
print	prints the given file
attrib (–/+r, –/+a, –/+s, –/+h)	changes a file's attributes (–deactivates, + activates, r: read-only, a: archive, s: system, h: hidden file)
format	formats the given drive
label	assigns a drive name to a drive
ver	gives the version number

Changing the default drive: To change the default drive, type the desired drive letter followed by a carriage return. DOS will respond with a prompt that gives the desired drive as the default.

Directories:

To obtain a listing of the files on the default drive (a directory), give the command:

A>dir

To a listing of the files on another drive, type:

A>dir a:

to see, for example, which files are on the A disk.

Specifying a filename:

Most DOS commands require you to specify one or more filenames. Valid filenames are composed of up to eight characters and are optionally followed by an extension of up to three characters. The filename and extension are separated by a period. Some examples of DOS filenames are:

sst.exe

demo.cmd

Copying files:

To copy a file (named oldfile) to a new file named newfile, enter:

A>copy oldfile newfile

The file newfile will have exactly the same contents as the file oldfile. As always, DOS assumes oldfile is on the default drive and also places newfile on the default drive.

Abbreviating filenames in DOS: DOS allows you to use a simple scheme of wild cards to refer to multiple filenames without having to type out the entire list of filenames that you want. There are two wild cards, the question mark and the asterisk. A question mark matches any character while an asterisk matches any string of characters. For example, 's?t' matches 'sat', 'sst', and 'st'. Similarly, 's*t' would match each of the previous names as well as 'short', 's123.45t'. The abbreviation '*.sav' would match any filename with the extension '.sav' (i.e., any SST system file), while '*' (or, equivalently, '*.*') would match any filename.

To obtain a listing of all the files on the A drive with the extension '.sav', type:

B>dir a:*.cmd

To copy all files from A to B, you could use the command:

C>copy a:*.* b:

SST uses a similar scheme of wild cards to abbreviate variable names.

Listing the contents of a file on the monitor: To display the contents of a text file on the monitor, you can use the DOS type command which has the following syntax:

A>type filename

Subdirectories: DOS allows you to organize disks into subdirectories—groupings of files in a tree structure. For most purposes a subdirectory on a disk is used as if it were a separate disk.

To create a subdirectory named sst, type:

C>mkdir sst

The subdirectory sst now exists. There are two ways to access files in the subdirectory. First, you can make the subdirectory the default directory in the same way that you make a disk drive the default drive. This is done by changing directories:

C>cd sst

You are now located within the sst subdirectory. To obtain a listing of files in the sst subdirectory, give the command:

C>dir

Unless you specify otherwise, DOS assumes that you only want a listing of files in the default subdirectory--which is whatever subdirectory you happen to be located in at the time you issue a command. To return to the main (or root) directory, give the command:

C>cd \

The backslash ('\') is DOS's symbol for the top directory.

The other way to access files in a subdirectory is to provide DOS with a path telling it how to find the file you are interested in. If you are in the root directory and you would like to see the contents of a file in the sst subdirectory, try:

C>type sst\filename

Thus, it is not necessary to change directories to access a file in another directory.

Specifying pathnames can become rather tedious and DOS provides a facility for specifying which directories are to be searched for programs with the path command. If you type:

C>path \;\sst

DOS will search the root directory (\) and the sst subdirectory (\sst) when it looks for programs to execute. With the above path command, it is possible to execute sst, for example, without being located in the sst subdirectory.

The AUTOEXEC.BAT File: Every time you start your computer, DOS looks for a file called autoexec.bat on the default drive and, if it finds this file, it executes the commands there.

6.10.1 History and Evolution of MS-Windows

MS-DOS is effective, but also proves difficult to understand for many people. There has to be a better way to build an operating system. The following section discusses the development of MS-Windows in its chronological order.

1982–1985 Introducing Windows 1.0: Microsoft works on the first version of a new operating system. Interface Manager is the code name and is considered as the final name, but Windows prevails because it best describes the boxes or computing "windows" that are fundamental to the new system. Windows is announced in 1983, but it takes a while to develop.

There were drop-down menus, scroll bars, icons, and dialog boxes that make programs easier to learn and use. You're able to switch among several programs without having to quit and restart each one. Windows 1.0 ships with several programs, including MS-DOS file management, Paint, Windows Writer, Notepad, Calculator, and a calendar, card file, and clock to help you manage day-to-day activities. There's even a game—Reversi.

1987–1992: Windows 2.0–2.11: On December 9, 1987 Microsoft releases Windows 2.0 with desktop icons and expanded memory. With improved graphics support, you can now overlap windows, control the screen layout, and use keyboard shortcuts to speed up your work. Some software developers write their first Windows–based programs for this release.

Windows 2.0 is designed for the Intel 286 processor. When the Intel 386 processor is released, Windows/386 soon follows to take advantage of its extended memory capabilities. Subsequent Windows releases continue to improve the speed, reliability, and usability of the PC. In 1988, Microsoft becomes the world's largest PC software company based on sales. Computers are starting to become a part of daily life for some office workers.

1990–1994: Windows 3.0–Windows NT: On May 22, 1990, Microsoft announces Windows 3.0, followed shortly by Windows 3.1 in 1992. Taken together, they sold 10 million copies in their first 2 years.

When Windows NT releases on July 27, 1993, Microsoft meets an important milestone: the completion of a project begun in the late 1980s to build an advanced new operating system from scratch. Unlike Windows 3.1, however, Windows NT 3.1 is a 32-bit operating system, which makes it a strategic business platform that supports high-end engineering and scientific programs.

1995–2001 Windows 95: On August 24, 1995, Microsoft releases Windows 95, selling a record-setting 7 million copies in the first five weeks. It's the most publicized launch Microsoft has ever taken on. Windows 95 has built-in Internet support, dial-up networking, and new Plug and Play capabilities that make it easy to install hardware and software.

1998–2000 Windows 98, Windows 2000, Windows Me: Windows 98 released on June 25, 1998, as the first version of Windows designed specifically for home PCs. With Windows 98, you can find information more easily on your PC as well as the Internet. Other improvements include the ability to open and close programs more quickly, and support for reading DVD discs and universal serial bus (USB) devices. Another first appearance is the Quick Launch bar, which lets you run programs without having to browse the Start menu or look for them on the desktop.

More than just the upgrade to Windows NT Workstation 4.0, Windows 2000 Professional is designed to replace Windows 95, Windows 98, and Windows NT Workstation 4.0 on all business desktops and laptops. Built on top of the proven Windows NT Workstation 4.0 code base, Windows 2000 adds major improvements in reliability, ease of use, Internet compatibility, and support for mobile computing, support for a wide variety of new Plug and Play hardware, including advanced networking and wireless products, USB devices, IEEE 1394 devices, and infrared devices.

2001–2005 Windows XP: On October 25, 2001, Windows XP is released with a redesigned look and feel that's centered on usability and a unified Help and Support services center. It's available in 25 languages. From the mid-1970s until the release of Windows XP, about 1 billion PCs have been shipped worldwide. It's both fast and stable. Navigating the Start menu, taskbar, and Control Panel are more intuitive. Awareness of computer viruses and hackers increases, but fears are to a certain extent calmed by the online delivery of security updates. Consumers begin to understand warnings about suspicious attachments and viruses. There's more emphasis on Help and Support.

Windows XP has several editions during these years: Windows XP 64-bit Edition (2001), Windows XP Media Center Edition (2002) and Windows XP Tablet PC Edition (2002).

2006–2008 Windows Vista: Windows Vista is released in 2006 with the strongest security system yet. User Account Control helps prevent potentially harmful software from making changes to your computer. In Windows Vista Ultimate, BitLocker Drive Encryption provides better data protection for your computer, as laptop sales and security needs increase. Windows Vista also features enhancements to Windows Media Player as more and more people come to see their PCs as central locations for digital media.

2009 – Today Windows 7 and counting: By the late 2000s, the wireless world has arrived. When Windows 7 is released in October 2009, laptops are outselling desktop PCs and it's common to get online at public wireless hotspots like coffee shops. Wireless networks can be created at the office or at home.

Windows 7 includes mabt featyres, such, as new ways to work with windows—Snap, Peek, and Shake. Windows Touch makes its debut, enabling you to use your fingers to browse the web, flip through photos, and open files and folders. You can stream music, videos, and photos from your PC to a stereo or TV.

6.11 OFFICE SUITES

In computing, an office suite, sometimes called an office software suite or productivity suite is a collection of productivity programs intended to be used by knowledge workers . The components are generally distributed together, have a consistent user interface and usually can interact with each other, sometimes in ways that the operating system would not normally allow.

Existing office suites contain wide range of various components. Most typically, the base components include: Word processor, Spreadsheet and Presentation program. Less common components of office suites include: Database, Graphics suite (raster graphics editor) vector graphics editor, image viewer), Desktop publishing software, formula editor, diagramming software, Email client, Prihect management software and so on.

6.11.1 Word Processors

A word processor is a computer application used for the production (including composition, editing, formatting, and possibly printing) of any sort of printable material. Word processing software evolved from the needs of writers as word processing software gives users an extensive set of tools for working with text. It is used to create all kinds of documents, from simple notes and memos, to brochures, resumes, and long reports that might otherwise be prepared on a typewriter.

The key advantage of a word processor is its ability to make changes easily, such as correcting spelling, adding, deleting, and relocating text. Once created, the document can be printed quickly and accurately and saved for later modifications. A number of softwares are available for word processing like Microsoft Word, Lotus WordPro or WordPerfect etc. Although these are expensive and sophisticated products, many of the facilities of word processing are also to be found in free software, like Microsoft WordPad, which comes as part of the Microsoft Windows standard installation.

Modern word processors display documents in graphics mode, thus enhancing the WYSIWYG function (What You See Is What You Get); this means that the way the text and images are displayed on the screen is the way they will appear when printed. Their formatting features include font changes, page layout, and paragraph indention. They also check spelling, find synonyms, incorporate graphics, perform calculations, and so on. The basic concept of word processing software is that one can type the text into the computer, instead of onto paper. Then errors can be corrected, text can be rearranged and all sorts of modifications can be made. It can then be seen on the screen and if we are satisfied with it, then only take a print-out on paper. It is a four steps process-type, store, edit, and print. A word processor is software that enables us to write, edit, format, and print text. As we type on a keyboard, the words appear on the screen to see. Mistakes can be corrected easily before printing the text on paper. Words, sentences, and even entire paragraphs can be moved by special commands. Nothing is printed until we are satisfied with the results.

Getting Started with Microsoft Word: Before getting started with Microsoft Word, we need to locate and open it from the computer desktop. On computer, we should be able to see a MS Word icon on the desktop. From the computer desktop:

1. Double-click on the MS Word icon or Go to the Start menu if the MS Word icon is not on the desktop.
2. Click ►Start ►Programs ►Microsoft Word

   *MS Word will open a blank page called Document 1.

Structure of Microsoft Word Window: When you open the word a blank document opens. You can start typing on it directly. But for performing various other options you have to learn about the different components in the window and the options under these bars.

1. **Title Bar:** Title bar shows the name of the document and is situated in the top of the window application.
2. **Menu Bar:** Menu bar contains the various commands under the various topics to perform some special tasks. Menu bar is located under the title bar.

3. **Standard Toolbar:** Toolbar is nothing more than the shortcut of the Menu options to execute or perform the menu options in the easiest and fastest way. Standard toolbar contains the commands for the Word's most frequently used commands.
4. **Formatting Toolbar:** Formatting toolbar contains the various commands related to the formatting of the document.
5. **Ruler:** From Ruler margins and indents can be adjusted in the easier way and it also provides measurement for the page formatting.
6. **Cursor:** Cursor is Word pointer, which tells that from or on the cursor position you can perform your action.
7. **Drawing Toolbar:** Drawing toolbar has the various commands related to the graphics and the drawing options.
8. **Status Bar:** It displays the positioning of the cursor, displays the status of some important keys of keyboard, and shows the messages for the toolbar button when a mouse points to it, displays messages for menu option when a menu option is selected or pointed out by a user.
9. **View Buttons:** View buttons are shortcuts of various views in the View Menu.
10. **ScrollBars:** There are two types of scrollbars
    - **i.** Horizontal Scrollbar
    - **ii.** Vertical Scrollbar

    Working of both the scrollbar is to scroll the document in either direction.

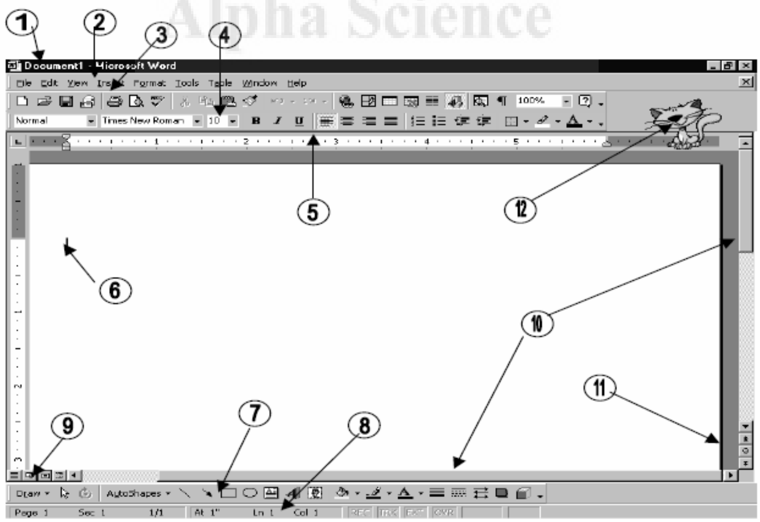


Fig. 6.6 Different components of a MS Word window

11. **Document Navigator:** Document Navigator allows navigating the document in different types of objects and is activated when clicked on the ball type button on the vertical scroll bar.

12. **Office Assistant:** Office assistant provides you the online help, real-time tips while working.

Working with Word Document - File Related Operations: When WORD is opened, you will be presented with a blank page. All what you have to do is start writing. Of course you don't have to worry about overflowing the text line because the program will take care of breaking it up appropriately. After writing your first document you have to save it; if you don't and turn off the computer what you have written will be cancelled and nothing will remain. However the program asks you what you want to do of the document the moment you shut off WORD without having saved it first. If you intend to keep the document you will be prompted to tell the computer where you want to save it. By default it will be saved as a file in a folder called My Documents. Give a name to the document and choose the file's destination: it will be saved with a .DOC extension; this is to tell Windows that this document is a WORD file. From now on the document you have written will be available for you to do whatever you like: you can open it again and modify it, or print it, or send it via e-mail or fax. We can do all file or document related operations from FILE menu.

Open a New File: To open a new file, Click on the File menu. Then Click on the new option or you can directly press the Ctrl+N from the keyboard. Select the Blank document from the General Tab from the following dialog box and then press OK button.

Save a Document: For saving a document, Click on the File►Save option. OR Press Ctrl+S OR Press the Save tool from standard toolbar.

Formatting a Document: Once the text of a document has been typed, it is required to be formatted. MS-Word provides Format menu to give a meaningful format of your document, so that a user can read and understand the document in an easy manner. Following are the options available in the Format menu.

Text Formatting: Format of font i.e. the font type, font size, font style, font Color, character spacing etc. can be changed by going in the option Font of Format menu.

Changing of Fonts: The word font means the way of writing characters. To change the font-using Font dialog box gives you the opportunity to see the result within the dialog box before applying it on the text. Other benefit of using the font dialog box is that you can apply several effects within the same dialog box like font color, bold or font size etc. Select the desired text to change the font. Select the Font option from Format menu. A window will appear on the screen. Scroll through the Font list and click the font that you want. Select the font color, underline style and Effects of your choice. Select the font style according to the text. Click on Ok button.

Paragraphs Formatting - Alignment of the Paragraph: Alignment refers to the way the right and left edges of a paragraph line up along the right and left margins of your document. Word gives you four alignment choices - left, centered, right and justified.

Page Setup: Click ►File ►Page Setup

A Page Setup window will appear. From the Page setup option one can setup the page layout according to the job. You can adjust the different margins or apply different options from the margin tab To change the orientation of your document to landscape (sideways): From the Paper size tab you can set the length or width of the page. When you click on the Paper size tab the following window will appear.

1. Click on the Paper Size tab
2. Click in the circle besides Landscape
3. Click ►OK

Find, Replace And Go To Options:

Find: Some times while working in document you need to find a particular text. To find a particular text click on the Edit menu, click Find or Ctrl+F. Then you will see the window shown below. In the Find what box, enter the text that you want to search for. Select the direction of searching from Search list box. Select any other options that youMatch case: To find the characters that are a word by themselves and are not a part of another word.

Replace: If you have to replace a word in the document with another word you can use find and replace command. On the Edit menu, click Replace or Ctrl+H. Then you will find a window.

6.11.2 Spreadsheet Software

Spreadsheet helps to prepare data in an organized, orderly and meaningful fashion. Spreadsheet finds its major contribution in creating reports, using formulas and performing calculations. It is best suited for scientific and statistical analysis. Spreadsheet can also be used to prepare Profit and Loss accounts, Balance sheet and other Tax statements. It can be used for preparing analytical reports including statistical analysis, forecasting and regression analysis. Good looking and attractive charts can be created which depict data in clearer and meaningful fashion. Spreadsheet can also used to create relationships between different types of data. Spreadsheet can do all the work of a full-fledged word-processor but it lacks the advanced features of desktop publishing. It also supports the high level features of object linking and embedding which means data from word processor can be safely and easily put and linked with data in spreadsheet and vice versa is also true. MS Excel is one of the popular spreadsheet software.

An electronic spreadsheet is a relational grid of columns and rows used to store and manipulate numeric information by means of a computer. The grid appears on the display screen, and data is stored in the memory of computer.

Structure of Worksheet in MS Excel: To start Excel, follow the following steps:

1. Click on the Start button at windows taskbar.
2. Select Program option.
3. Select Microsoft Excel option.

4. Selecting this option will open up Microsoft Excel and the main screen will appear.

Let us discuss the structure of Worksheet and Functions of Various Components of Microsoft Excel from the main screen.

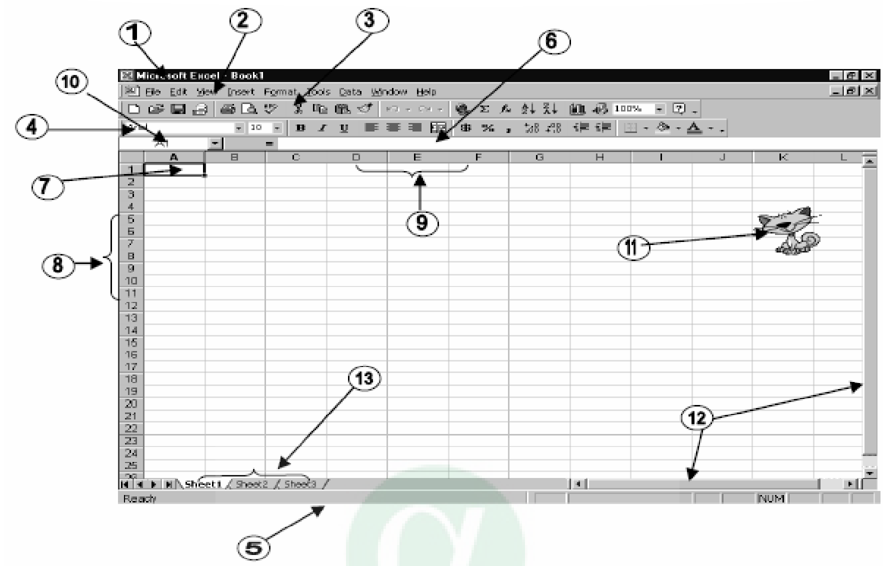


Fig. 6.7 Functions of various components of Microsoft Excel

1. **Title Bar:** Title bar shows the name of the document and situated in the top of the window application.
2. **Menu Bar:** Menu bar contains the various commands under the various topics to perform some special tasks. Menu bar is located under the title bar.
3. **Standard Toolbar:** Toolbar is nothing more than the shortcut of the Menu options to execute or perform the menu options in the easiest and faster way. Standard toolbar contains the commands for the Excel's most frequently used commands.
4. **Formatting Toolbar:** Formatting toolbar contains the various commands related to the formatting of the document.
5. **Status Bar:** It displays the positioning of the cursor, displays the status of some important keys of keyboard, shows the messages for the toolbar button when a mouse points to it, display messages for menu option when a menu option is selected or pointed out by a user.
6. **Formula Bar:** Formula bar shows you the formula instead of the result whenever you click on a cell in which you have a formula.
7. **Rows:** Rows are combination of all the horizontal cells joined together in single horizontal line. An Excel workbook has maximum 65,536 rows in a worksheet.

8. **Columns:** Columns are combination of all the vertical cells joined together in single vertical line. An Excel workbook has maximum 256 columns in a worksheet
9. **Cell:** A Cell is an intersection point of row and column. It is used to store data. There are 65536 × 256 = 16777216 cells in a single worksheet.
10. **Name Box:** This box shows the name or the address of the currently active cell.
11. **Office Assistant:** Office assistant provides you the online help, real-time tips while working.
12. **Scroll Bars:** There are two types of scroll bars
    - **i.** Horizontal Scrollbar
    - **ii.** Vertical Scrollbar

    Working of both the scroll bar is to scroll the Worksheet in the either directions.
13. **Worksheets:** As name suggests it is the working area in which you can work. Each worksheet of a single workbook has its own identity they are separate from other worksheets. The maximum limit of the worksheet in a workbook is 255.

Creation of Spreadsheet:

To open a new workbook, follow the steps:

1. Click on the File Menu.
2. Select Option New.
3. Click the OK Button. Now you will get a fresh new workbook to work on.

To Save a Workbook:

For saving a document, following are the steps:

Click Save option on the File OR Press Ctrl+S OR Press the Save Button from standard toolbar. If you are saving the file for the first time, the Save As… window will appear. Choose the appropriate folder from Save in combo box. Write the proper file name in the File name text box. And then press the Save button.

Opening of an Existing Workbook:

To open an existing workbook, the following steps are required:

1. Select the Open option from the File menu OR Press Ctrl+O OR Click on the Open tool from the standard toolbar. Then the open dialog box will appear.
2. Select the appropriate folder from Look in combo box.
3. Select the required file from the file window.
4. Click on open button on the right hand side OR Press Enter.

Closing of Workbook: To close an already opened workbook just choose the Close option from file menu but keep it in your mind that only the current window or workbook will close because Microsoft Excel works in MDI (Multi document interface) environment unlike notepad which works in SDI(Single document interface).

Page Setup: From the Page setup option, one can setup the page layout according to his work. For using the Page setup option, you have to perform the following steps:

1. Click on the Page setup option from the file menu. Then a window will appear.
2. Select the page Orientation from Portrait or Landscape.
3. Setting the Adjust to setting will Reduces or enlarges the printed worksheet.
4. Select the Adjust to check box, and then enter a percentage number in the % normal size box.

Print Option: For taking the printout you have to select the print option of the file menu. After selecting the print option from file menu the window given below will appear.

Creating Formulas: After entering the data into the worksheet, calculations are performed with that data. This is done using formulas, the true power of a spreadsheet program. For creating the formula, syntax is required which describe the sequence of character used in a formula. The syntax of a formula begins with an equal sign (=) in Excel and is followed by a combination of values, operators and cell references.

Operators: Operators specify the type of operation that has to be performed on the parts of a formula. Basically we use three types of operators.

1. Arithmetic Operators: To perform mathematical operation on numeric values and thereby produces numeric results.

2. Comparison Operators: To compare the values or labels in two cells. The result of this formula is either TRUE or FALSE.

3. Text Operators: To join one or more text values into a single combined text value. The ampersand (&) (which is the only text operator) is used to join one or more labels into a single combined label. For example, = D2& " "&F3 This formula combines the labels in cells D2 and F3.

When arithmetic operators are used in formulas, spreadsheets calculate the results using the rules of precedence followed in mathematics. The order is as follows:

1. Exponentiation (^)
2. Negation (-)
3. Multiplication and Division (*,/)
4. Addition and Subtraction (+, –)

Otherwise, operators are evaluated from left to right. In order to change the order of precedence to suit your needs, you add parentheses around any part of the formula that you want to be calculated first.

Creating Functions: Spreadsheets come with many built formulas, called functions that perform specialized calculations automatically. We can include these functions in our own formulas. Some functions are quite simple, such as the COUNT function (to count the number of values in a range of cells). Many functions however are very complex.

To insert a function you have to follow the following steps.

1. Click on the cell in which you want to insert the function.
2. Select the option Function from insert menu.
3. Above dialog box will appear.
4. Select the desired function and click on OK button

Using Various Functions in MS-Excel: The leading spreadsheets come with hundreds of these functions. Here we are discussing some of the most commonly used ones with the help of examples:

1. Date and Time Functions
   i. DATE: Represents the date number that represents a particular date.

      Syntax: DATE(year, month, day)

      Example: = date(2005, 04, 17) MS-EXCEL returns 4/17/05

   ii. DATEVALUE: Returns the serial number of the date represented by date_text.

      Syntax: DATEVALUE(date_text)

      Example: =DATEVALUE("2005/04/17") MS-EXCEL returns 38459

   iii. DAY: Returns the day of a date, which is represented by a serial number. The day is given as an integer ranging from 1 to 31.

      Syntax: DAY(serial_number)

      Argument Type: Number(Date number)

      Return Type: Number

      Example: =DAY(38459) MS-EXCEL returns 17

   iv. MONTH: Returns the month of a date, which is represented by a serial number. The month is given as an integer ranging from 1 to 12.

      Syntax: MONTH(serial_number)

      Argument Type: Number(Date number)

      Return Type: Integer

      Example: =MONTH(38459) MS-EXCEL returns 4

2. Financial Functions
   i. FV: Returns the future value of an investment based on periodic, constant payments and a constant interest rate.

      Syntax: FV(rate, nper, pmt, pv, type)

      Argument Type: Number, Number, Number, Number,

      Return Type: Number

Example: =FV(0.005, 18, –500, –2000, 0) MS-EXCEL returns $11,580.75

ii. PMT: Calculates the payment for a loan based on constant payments and a constant interest rate.

Syntax: PMT(rate, nper, pv, fv, type)

Argument Type: Number, Number, Number, Number, Number

Return Type: Number

Example: =PMT(8%/12,10,10000,0,1) MS-EXCEL returns ($1,030.16)

3. Math Functions

i. ABS: Returns the absolute value of a number.

Syntax: ABS(Number)

Argument Type: Number

Return Type: Number

Example: =ABS(5)

MS-EXCEL returns 5 Type: =ABS(-5) MS-EXCEL returns 5

ii. EXP: Returns e raised to the power of a number e.g. e^x. The constant e=2.71828182845904, the base of the natural logarithm.

Syntax: EXP(number)

Argument Type: Number

Return Type: Number

Example: =EXP(1) MS-EXCEL returns 2.718281828

Example: =EXP(2) MS-EXCEL returns 7.389056099

Data Management:

Data menu of the Excel provides various commands, which you can apply to your data.

Sorting of Data: To sort the data you have to perform the following steps.

1. Select the columns which you want to sort.
2. Choose Sort option from Data menu. You will find out the following dialog box on the screen.
3. Choose the Column on which you want to do the sorting. You can select more than one column.
4. Select the Ascending or descending option.
5. Select the Header row if there is any header row in the table.
6. Click on the OK button.

Graphics on Spreadsheet:

Inserting Images, Auto shapes and Word Art, Inserting Images from Clip Gallery:

To insert an image from Clip Gallery you have to follow these steps:

1. Move the cursor on the cell where you want to insert the image.
2. Select Clip art option from Insert->Picture menu.
3. Click the picture category from which you want to insert the picture.
4. If you know what type of image you are looking for, type a descriptive word or two in the Search for Clips text box, and press Enter. Excel displays the images that most closely match your keywords.
5. When you find out the image that you want to use, click it.
6. Clicking on image will display a small toolbar, from that toolbar click the insert button.This will insert a clipart in your worksheet.

Spreadsheets can also be applicable in presenting the graphics and managing the database. For the future, one can expect spreadsheets to offer more special features for the advanced user while simplifying frequently used tasks for beginners and advanced users alike.

6.11.3 Presentation Software

A presentation program (also called a presentation graphics program) is a computer software package used to display information, normally in the form of a slide show. It typically includes three major functions: an editor that allows text to be inserted and formatted, a method for inserting and manipulating graphic images and a slide-show system to display the content. Since MS Power Point is one of the popular presentation softwares, the following discussion is meant for it.

Steps to create a typical presentation:

1. First, create a new presentation.
   - File > New > Blank Presentation, select Title Layout.
   - Type in your title, then your name, your institution/program and date underneath.
   - You can change the orientation of your slides by selecting File > Page setup…
2. Then you can choose a background design.
   - Either select Format menu or Right-click mouse on slide then Apply Design, select design (e.g. Whirpool.pot). Note that some backgrounds have preset animation.
   - You can make your own background as well. Right-click on the slide, or Format > Background > Select the pull-out menu > Fill Effects.., and choose solid color or patterns.
   - You can also use your own image (JPEG, TIFF, or GIF) to create a custom background. At Fill Effects, select Picture > Select Pictures...> find your image file. PowerPoint will automatically fit your image to the slide size; however, you should make sure that the image size is proportional to your slide to avoid distorted background.

- The background color for a preset scheme can be changed: Right-click or Format > Color Scheme > Custom> select Title text > Change Color… > select color (e.g. light yellow).

3. Now that you have a good background, go on to add new slides.
   - On the slide after which you want to add a new one, select Insert > New Slides, choose appropriate layout (e.g. Bulleted List).
   - Type in appropriate text into fields.
   - To demote or promote a paragraph (increase or decrease indent in a bullet list), you can either use the left and right arrows on your toolbar or use key shortcuts: Tab to demote, Shift-Tab to promote.
4. There are four ways you can enter graphics into your presentation.
   - Clip Arts from PowerPoint: Insert > Picture > Clip Art.
   - Your own picture files (GIF, TIFF, JPEG, or BMP): Insert > Picture > From File.
   - Inserting Excel Graph: You can select the graph in Excel, copy it (Ctrl-C) and then paste it in the PowerPoint slide (Ctrl-V) or Paste as Picture (Edit > Paste Special > Picture). It is recommended (from experience) that you paste as a picture to avoid formatting problems you may have later on.
   - You can add drawings on slide using the drawing tools on the bottom toolbar (See attached tutorial on Drawing in PowerPoint).
   - To move the graphics, click on the image and drag to a new position. For finer adjustments (not snapping on grid), you can hold down Alt key while moving the object. You can also select the object (click on it), then move it using the arrow keys. The Alt key can be used in this case as well. To move an object horizontally or vertically, hold down the Shift key while moving to make sure it only moves in those directions.
   - To resize an object, first select the object, then click on one of the white squares that appear along the side or corner of the image, then drag the square. To resize an image proportionally, hold down the Shift key while re-sizing. To re-size about the center of the image, hold down Ctrl key while re-sizing.
   - To re-size text fields, first click on the field so that the box is highlighted. Next, click on the frame surrounding the field to select the text field. Then, follow similar procedure to resize the text box.
   - You can flip or rotate any PowerPoint generated drawing using the drawing toolbar on the bottom. If the icon is missing, go to Draw > Rotate or Flip.
   - In order to use these commands on an image, you must first change the image into Microsoft Office Drawing Object by un-grouping, then re-grouping the picture: Select image > Right-click > Grouping > Ungroup/Regroup.

5. For adding equation into slide: Insert > Object > Select Microsoft Equation. Click outside equation to return to slide. You can change the color within the equation box by right-clicking on the object > Format Object… > then select Fill color or add a frame around the equation by selecting the Line color.

6. There are two ways to change your font size and type.

   - highlight the text you want to change, and go to Format > Font on the upper toolbar
   - To change the font style on every slide at once, use the Slide Master. Go to View > Master > Slide Master, and make any appropriate changes. You can go back to your slides by going to View > Slides. Anything that you change here will be applied on every slide except the title slide. To change the format of your title slide, go to View > Master > Title Master.

7. The bullet style can be changed in a similar manner (from each slide or Slide Master) by going to Format > Bullet…

8. You can enter date, institution name and slide number on the bottom of all the slides: View > Header and Footer…

9. To Add some punch into your presentation, you can apply special effects.

   - To add animation effects into your text and/or objects (include graphics): Select object to be animated, then right-click and select Custom Animation. You can make your bulleted list appear one by one from different locations as you click on the screen during presentation.
   - You can also use preset animation: select text or object > Slide Show > Preset Animation. Be aware that most of these animations have been programmed with sound effects.
   - To apply slide transition effects: Slide Show > Slide Transition. These effects refer to the way in which a new slide appears.

10. If you want to change slide order, use Slide Sorter view (View > Slide Sorter). In this view, you can drag one more more slides to a new position by selecting them and dragging them, or you can delete slide(s) by selecting and pressing Backspace.

11. There are two ways in which PowerPoint lets you practice the presentation.

    - Full Screen with no timing: Slide Show > View Show
    - Full Screen timed: Slide Show > Rehearse Timings
    - After practicing with timing, be sure to disable automatic advancing of slides (PowerPoint can save timings from your practice and automatically advance slides at those timings). Go to Slide Show > SetUp Show…, and click on Advance slides, Manually.

12. To navigate your way through the slides during presentation:
    - To go to next slide: Click on mouse, or press Right arrow, Down arrow, or Spacebar.
    - To go to previous slide: press Left arrow or Up arrow.
13. PowerPoint also allows you to turn your presentation into html format so that it can be viewed on the web.
    - Go to File > Save as HTML… >, and a window will appear. This window guides you through the different settings you can choose for the file.
    - Choose a layout, image style, button style, and fill out some other information. When you type in a folder and click on Finish, it will create a new folder containing everything you need to publish your presentation on web. To put the presentation on the web, make sure you up-load all files in the folder.
    - The presentation is saved so that each slide is an image. Because of this, some details may be lost during the conversion between Microsoft Drawing or Text Object and the graphic file.

6.12 WEB BROWSERS

Web browser is a software application used to locate, retrieve and also display content on the World Wide Web, including Web pages, images, video and other files. As a client/server model, the browser is the client run on a computer that contacts the Web server and requests information. The Web server sends the information back to the Web browser which displays the results on the computer or other Internet-enabled device that supports a browser.

Today's browsers are fully-functional software suites that can interpret and display HTML Web pages, applications, JavaScript, AJAX and other content hosted on Web servers. Many browsers offer plug-ins which extend the capabilities of a browser so it can display multimedia information (including sound and video), or the browser can be used to perform tasks such as videoconferencing, to design web pages or add anti-phishing filters and other security features to the browser.

Also, there are a number of browsers that are designed to access the Web using a mobile device. These mobile browsers ("Microbrowser") are optimized to display Web content on smaller mobile device screens and to also perform efficiently on these devices which have far less computing power and memory capacity as Desktop or laptop computers. Mobile browsers are typically "stripped down" versions of Web browsers and offer fewer features in order to run well on mobile devices.

Browsers translates the basic HTML (Hypertext Mark Up Language) code that allows us to see images, text videos and listen to audios on websites, along with hyperlinks that let us travel to different web pages. The browser gets in contact with the web server and requests for information. The web server receives the information and displays it on the computer.

There are different browsers for various things you do on the internet. There's a text based browser. With a text based browser you are only allowed to see text. Graphical material are not displayed. Hyperlinks are accessed by typing it in through the keyboard. An example of a text based browser is LYNX. There's a graphical browser and that allows the user to see all types of multimedia.

There are several types of browsers but the one thing they have in common is that they carry out the same function; they transfer hypertext. Here are some familiar common web browsers:

- Mozilla FireFox
- Netscape Navigator
- Microsoft Internet Explorer

Every browser features a toolbar that allows you to perform various functions like:

- Go back to the first page you started on the internet which is called Home.
- Book your favorite websites
- Print content you find interesting on web pages
- Check your web history, like the websites you visited in the past
- You can go forward and backwards to see the previous sites you viewed

There are two basic ways that you can navigate through the browser:

- You can click on links on different web pages. Those links will lead you to other web pages.
- If you know the URL or the address of the website you can type it in write in the browser's box all the way at the top. There's no need to type in the: http:// part when inserting the address because the browser automatically places it in. Then you have to click enter.

Once you click enter you have to wait for the page to load. While it's loading you will see the browser icon. Usually a little box will appear at the bottom with bars. It's called a status bar. When all the bars are filled you will know that the page has finished loading. So every time you click or a link or enter a URL your browser will request the information from the web server. If your browser returns to the page with a error message the address you typed in could be wrong on you may not be connected to the internet.

Web pages can look different in various types of browser. It's always important to download the latest version of your browser. Today most web pages are designed to be viewed in updated browsers. By keeping up with the most up-to-date version of your web browser you will be able to get the best of the web for everything that you need.

Cookies: An HTTP cookie is a packet of information sent by a server to a World Wide Web browser and then sent back by the browser each time it accesses that server. HTTP cookies are used for user authentication, user tracking, and maintaining user-specific information (preferences, electronic shopping cart, etc.)

Cookies have been of concern for Internet privacy, since they can be used for tracking the browsing of a user. As a result, they have been subject to legislation in various countries such as the United States, as well as the European Union. Cookies have also been criticized because the identification of users they provide is not always accurate and because they can be used for network attacks.

Several Web sites also use cookies for personalization based on users' preferences. Sites that require authentication often use this feature, which is however also present on site not requiring authentication. Cookies are also used to track users across a Web site. Third-party cookies and Web bugs, explained below, also allows for tracking across multiple sites. Tracking within a site is typically done to the aim of producing usage statistics, while tracking across sites is typically used by advertising companies to produce anonymous user profiles, which is then used to target advertising (deciding which advertising image to show) based on the user profile.

6.13 DATA BASE MANAGEMENT SYSTEMS (DBMS)/ DATA BASE MANAGEMENT SOFTWARE

DBMS software was developed to handle the problems of maintaining and integrating large volumes of data on computers. It extends the ability to organize collections of data stored in computer and provides features that help in finding subsets of the data selected through specified criteria. Data Base Management Software (DBMS) is a collection of programs that enables the user to store, modify, and extract information from a database.

6.13.1 Evolution of Database Technology

A database is an organized collection of facts. In other words, it is collection of information arranged and presented to serve an assigned purpose. An example of a database is a dictionary, where words are arranged alphabetically.

In order to keep database updated, we may need to perform operations like adding information, removing information, editing existing information, etc. Take the case of a telephone diary, in which you note down the names, addresses and phone numbers of your friends. This is also a type of database. If you make a new friend, you note the information about him/her in the diary, which is equivalent to adding data to a database. If address of your friend changes then you note the new address in your diary, which means editing a database. If you want to send a letter to your friend then you look into the diary to locate the address. This is called searching a database.

6.13.2 Traditional File Concepts and Environment

The traditional file-oriented approach to information processing has for each application a separate master file and its own set of personal files. COBOL language supported these file-oriented applications. It was used for developing applications such as of payroll, inventory, and financial accounting. However, in general an organization needs flow of information across these applications also and this requires sharing of data, which is very difficult to implement in the traditional file approach. In addition, a major limitation of file-based approach is that the programs are dependent on the files and the files are dependent upon the programs. These file-based approaches, which came into being as the first commercial applications of computers, suffered from the following significant disadvantages:

Data Redundancy: In a file system, if an information is needed by two distinct applications, then it may be stored in two or more files. For example, the particulars of an employee may be stored in payroll and leave record applications separately. Some of this information may be changing, such as the address, the pay drawn, etc. It is therefore quite possible that while the address in the master file for one application has been updated the address in the master file for second application may have not been. Sometimes, it may not be easy to find that in how many files the repeating items such as the address has occurred. The solution, therefore, is to avoid this data redundancy by storing the address at just one place physically, and making it accessible to all applications.

Program/Data Dependency: In the traditional file oriented approach if a data field (attribute) is to be added to a master file, all such programs that access the master file would have to be changed to allow for this new field that would have been added to the master record. This is referred to as data dependence.

Lack of Flexibility: Since the data and programs are strong coupled in a traditional system, most information retrieval requests would be limited to well anticipated and pre-determined. The system would normally be capable of producing scheduled records and queries that it has been programmed to create. In the fast moving and competent business environment of today, apart from such regularly scheduled records, there is a need for responding to un-anticipatory queries and some kind of investigative analysis that could not have been envisaged professionally.

6.13.3 Advantages of DBMS

(a) It represents complex relationships among different data items.

(b) Keeps a tight control on data redundancy.

(c) Enforces user defined rules to ensure the integrity of data in a table form.

(d) Maintains data dictionary for the storage of information pertaining to data fields and data manipulation.

(e) Ensures that data can be shared across all applications.

(f) Enforces data access authorization.

(g) Has an automatic intelligent backup and recovery procedure of data.

(h) Has different interfaces through which users can manipulate data.

6.13.4 Types of Data Models

There are five types of data models depending upon the relationships between entities:

- Relational data model
- Hierarchical data model
- Network data model
- Object-oriented data model
- Distributed data model
- Relational Data Model

1. **Relational data model:** The relational data model was first proposed by Dr. E. F. Codd in 1970. In relational data model, the entities and their relationships are represented with two dimensional tables. The data is represented in a two dimensional table which is called a relational model of the data. In relational data model, such type of table is know as relation. But to avoid confusion between relation and relationship between entities, sometimes is called table.
2. **Hierarchical Data Model:** Hierarchical Tree Structure is the key concept of Hierarchical Data Modal. A hierarchical tree structure is made up of nodes and branches. The highest node of hierarchical tree structure is called root. Nodes at lower levels (Second, third etc.) are known as dependent nodes. A node is a collection of data attributes describing the entity at that node. A hierarchical data model is one that organizes data in hierarchical tree structure.
3. **Network Data Model:** Any item in a network structure can be linked to any other item. In this structure a node at lower level can be linked to more than one nodes at a level higher to it. A data model that interconnects the entities of an enterprise in network structure is known as network data model.
4. **Object Oriented Data Model (OODM):** Object: An object is defined as an entity with a set of predefined operations and data items that manipulate and access with in it or out side it.
5. **Distributed Data Model (DDM):** In a distributed database system, database is stored in several computers. The computers in a distributed system communicate and exchange data among one other using leased lines, telephone lines or other means of communication. These computers do not share memory or clock. Each of the computers in a distributed system participates in the execution of transactions. These computers can also be known as sites or nodes.

6.14 GRAPHICS SOFTWARE

The graphics Software offers the use of computers for manipulating (creating, editing, viewing, storing, retrieving and printing) images, designs, drawings, pictures, graphs and anything else that can be drawn in the traditional manner.

It is used to convert the data in a graphic form on the screen as well as on the paper using printers or plotters.

Now-a-days business organizations use presentation graphics application software packages which help in creating professional looking visual aids such as computer images, paper printouts, or photo transparencies for an audience. These are specialized type of graphics software to present data in the form of charts and graphs. These packages facilitate the users to quickly convert tabular data to graphical form.

Sophisticated graphs and complex drawings can be drawn in multicolor by pressing a few keys of computer keyboard. The user can orchestrate images and even sounds to produce automated “slide shows”. A graphics software package can also be used in text publishing. A desktop publishing software includes graphics to produce pictures and other forms of graphics in the text where needed. One more advantage of graphic software is the use of multimedia, a type of application that incorporates images, text, sound, computer animation and video and the programs that create multimedia are known as multimedia authoring software.

Another important application of graphics software is Computer Aided Design (CAD) software. CAD software is often used by the architects and engineers to draw designs of buildings or products before the actual construction or manufacturing gets under way.

Engineering is such an area where graphics software contributes a lot in the form of engineering drawings and different types of graphs.

6.15 DATA COMMUNICATION SOFTWARE

Data communication software is that software which put one computer in touch with others and provides facility of transferring data from one machine to other. In this way more than one computer using a modem may communicate with each other with the help of this communication software. Network Software is another type of data communication software which allows users to share data by creating a group of connected computers i.e. network of computers. Most popular network communication software, which are almost considered as a necessity for business computing are Novell Netware and Microsoft’s Windows NT.

Other data communication software may include fax software for computers to work as a fax machine and the software distributed by the online service providers, such as CompuServe, America Online, and Microsoft Network.

REVIEW QUESTIONS

1. Define computer software.
2. Classify computer software based on their types.
3. Write a short note on system software.
4. What is meant by programming software?
5. How application software differ from programming software?
6. Explain in detail about various steps involved in software development.
7. Distinguish High-level languages and Machine languages.
8. What is meant by Procedural programming?
9. Explain the concept of Object Oriented Programming.
10. What are the functions of an operating system?
11. Classify operating systems.
12. What are the properties of Linux?
13. Write short note on architecture of the Linux Operating System.
14. List a few commonly used office suites.
15. What is the use of spread sheet software?
16. Discuss briefly about the working principle of web browsers.
17. Explain in detail about Data Base Management Software.
18. What is the purpose of graphics software?

EXERCISE QUESTIONS

1. Create a MS Word file and insert a picture of a computer and type the working of computer using the following formatting conditions:

   Font size: 12 Font colour: Black Spacing: 1.5 Alignment: Left.
2. Create a presentation using MS Power Point to take a seminar on "Office Suites". (Resources can be taken from this chapter itself).

CHAPTER

7

MEMORIES

7.1 BASIC CONCEPTS

Memory is an essential part of any computation system. It is used to store both the computation instructions and the data. Logically, memory can be viewed as a collection of sequential locations, each with a unique address as its label and capable of storing information. Accessing memory is accomplished by supplying the address of the desired data to the device.

Memory devices can be categorized according to their functionality and fall into two major categories viz, Read-Only-Memory (ROM), and write-and-read memory or Random-Access Memory (RAM). There is also another subcategory of ROM: mostly read but sometimes write memory or flash ROM memory. Within the RAM category there are two types of memory devices differentiated by storage characteristics, static RAM (SRAM) and dynamic RAM (DRAM) respectively. DRAM devices need to be refreshed periodically to prevent the corruption of their contents due to charge leakage. SRAM devices, on the other hand, do not need to be refreshed.

Both SRAM and DRAM are volatile memory devices, which means that their contents are lost if the power supply is removed from these devices. Non-volatile memory, the opposite of volatile memory, retains its contents even when the supply power is turned off. All current ROM devices, including mostly read sometimes write devices are non-volatile memories. Except for a very few special memories, these devices are all interfaced in a similar way. When an address is presented to a memory device, and sometimes after a control signal is strobed, the information stored at the specified address is retrieved after a certain delay. This process is called a memory read. This delay, defined as the time taken from address valid to data ready, is called memory read access time. Similarly, data can be stored into the memory device by performing a memory write. When writing, data and an address are presented to the memory device with the activation of a write control signal. There are also other control signals used to interface. For example, most of the memory devices in packaged chip format has a chip select (or chip enable) pin. Only when this pin is asserted, the particular memory device gets active. Once an address is supplied to the chip, internal address decoding logic is used to pinpoint the particular content for output. Because of the nature of the circuit structure used in implementing the decoding logic, a memory device usually needs to recover before a subsequent read or write can be performed. Therefore, the time between subsequent address issues is called cycle time. Cycle time is usually twice as long as the access time. There are other timing requirements for memory devices. These timing parameters play a very important role in interfacing the memory devices with computation processors. In many situations, a memory device's timing parameters affect the performance of the computation system greatly.

Some special memory structures do not follow the general accessing scheme of using an address. Two of the most frequently used are content addressable memory (CAM), and first-in-first-out (FIFO) memory. Another type of memory device, which accepts multiple addresses and produces several results at different ports, is called multi-port memory. There is also a type of memory that can be written in parallel, but is read serially. It is referred to as video RAM or VDRAM since they are used primarily in graphic display applications.

On the system level, we must organize the memory to accomplish different tasks and to satisfy the need of the program. In a computation system, addresses are supplied by the CPU to access the data or instruction. With a given address width a fixed number of memory locations may be accessed. This is referred to as the memory address space. Some processing systems have the ability to access another separate space called input/output (I/O) address space. Others use part of the memory space for I/O purposes. This style of performing I/O functions is called memory–mapped I/O. The memory address space defines the maximum size of the directly addressable memory that a computation system can access using memory type instructions. For example, a processor with address width of 16-b can access up to 64 K different locations (memory entries), whereas a 32-b address width can access up to 4 Giga-byte different locations. However, sometimes we can use indirection to increase the address space. The method used by 80×86 processors provides an excellent example of how this is done. The address used by the user or programmer in specifying a data item stored in the memory system is called a logical address. The address space accessed by the logical address is named logical address space. However, this logical address may not necessarily be used directly to index the physical memory. We called the memory space accessed by physical address the physical address space. When the logical space is larger than the physical space, then memory hierarchy is required to accommodate the difference of space sizes and store them in a lower hierarchy. In most of the current computing systems, a hard-disk is used as this lower hierarchy memory. This is termed virtual memory system. This mapping of logical address to physical address could be either linear or nonlinear. The actual address calculation to accomplish the mapping process is done by the CPU and the memory management unit (MMU). Thus far, we have not specified the exact size of a memory entry. A commonly used memory entry size is one byte.

For historical reasons, memory is organized in bytes. A byte is usually the smallest unit of information transferred with each memory access. Wider memory entries are becoming more popular as the CPU continues to grow in speed and complexity. There are many modern systems that have a data width wider than a byte. A common size is a double word (32b) in current desktop. As a result, memory in bytes is organized in sections of multi-bytes. However, due to need for backward compatibility, these wide data-path systems are also organized to be byte addressable.

The maximum width of the memory transfer is usually called memory word length, and the size of the memory in bytes is called memory capacity. Since there are different memory device sizes, the memory system can be populated with different sized memory devices.

7.2 MEMORY DEVICE ORGANIZATION

Physically, within a memory device, cells are arranged in a two-dimensional array with each of the cell capable of storing one bite of information. This matrix of cells is accessed by specifying the desired row and column addresses. The individual row enable line is generated using an address decoder while the column is selected through a multiplexer. There is usually a sense amplifier between the column bit line and the multiplexer input to detect the content of the memory cell it is accessing.

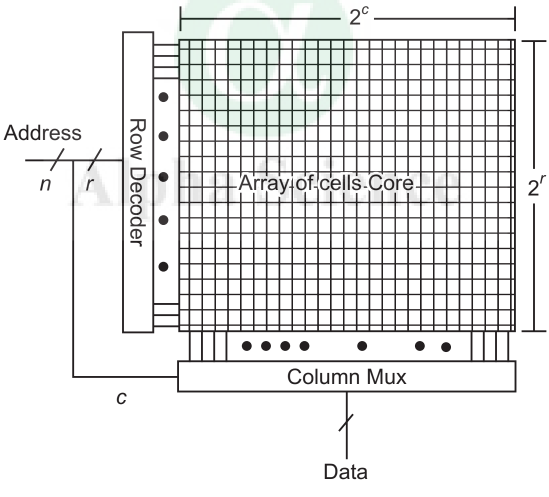


Fig. 7.1 Generic memory structure

Figure 7.1 illustrates this general memory cell array described with r bit of row address and 'c' bit of column address. With the total number of $r + c$ address bits, this memory structure contains $2r + c$ number of bits. As the size of memory array increases, the row enable lines, as well as the column bit lines, become longer. To reduce the capacitive load of a long row enable line, the row decoders, sense amplifiers, and column multiplexers are often placed in the middle of divided matrices of cells, as illustrated in Fig. 7.2. By designing the multiplexer differently we are able to construct memory with different output width, for example, ×1, ×8, ×16, etc.

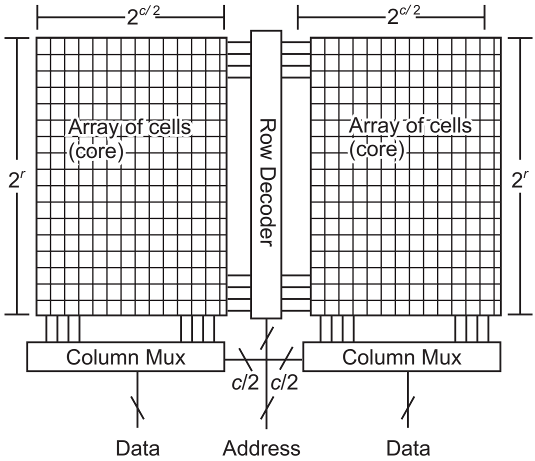


Fig. 7.2 Divided memory structure

According to the functionality and characteristics of memory, we may divide memory devices into two major categories: ROM and RAM. We will describe these different types of devices in the following sections.

7.3 READ-ONLY MEMORY (ROM)

In many systems, it is desirable to have the system level software (for example, Basic Input/Output System (BIOS) stored in a read-only format, because these types of programs are seldom changed. Many embedded systems also use read-only memory to store their software routines because these programs also are never changed during their lifetime, in general. Information stored in the read-only memory is permanent. It is retained even if the power supply is turned off. The memory can be read out reliably by a simple current sensing circuit without worrying about destroying the stored data. Figure 7.3 shows the general structure of a read-only memory (ROM). The effective switch position at the intersection of the word-line/bit-line determines the stored value. This switch could be implemented using different technologies resulting in different types of ROMs. Main types of ROMs are Masked Read Only Memory or simply ROM, Programmable ROM, Erasable Programmable ROM (EPROM), Electrically Erasable ROM (EEROM), and Flash EEPROM.

7.3.1 Masked Read-Only Memory (ROM)

The most basic type of this read-only-memory is called masked ROM, or simply ROM. It is programmed at manufacturing time using fabrication processing masks. ROM can be produced using different technologies, bipolar, complementary metal oxide semiconductor (CMOS), *n*-channel metal oxide semiconductor (*n*MOS), *p*-channel metal oxide semiconductor (*p*MOS),etc. Once they are programmed there are no means to change their contents. Moreover, the programming process is performed at the factory.

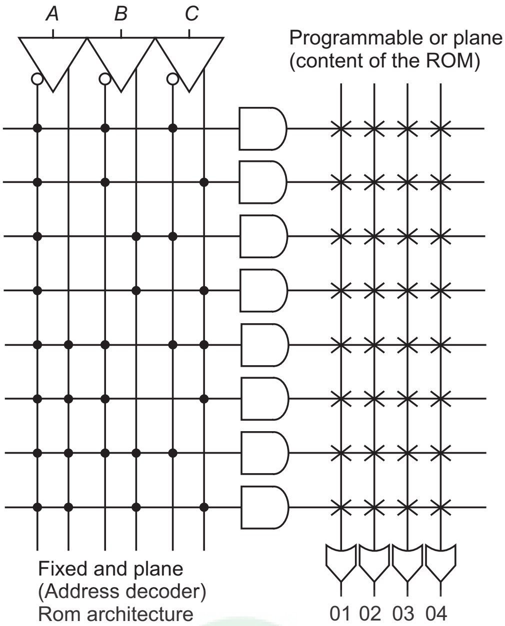


Fig. 7.3 General structure of a ROM (an 8 × 4 ROM)

7.3.2 Programmable Read-Only Memory (PROM)

Some read-only memory is one-time programmable, but it is programmable by the user at the user's own site. This is called programmable read-only memory (PROM). It is also often referred to as write once memory (WOM). PROMs are based mostly on bipolar technology, since this technology supports it very well. Each of the single transistors in a cell has a fuse connected to its emitter. This transistor and fuse make up the memory cell. When a fuse is blown, no connection can be established when the cell is selected using the row line. Thereby a zero is stored. Otherwise, with the fuse intact, a logic one is represented. The programming is done through a programmer called a PROM programmer or PROM burner. Figure 7.4 illustrates the structure of a bipolar PROM cell and its cross section when fabricated.

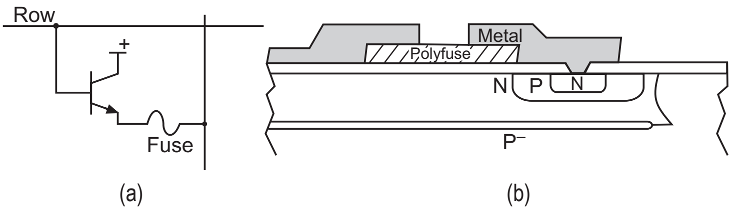


Fig. 7.4 Bipolar PROM: (a) bipolar PROM cell, (b) cross section of a bipolar PROM cell.

7.4 RANDOM ACCESS MEMORY (RAM)

RAM stands for random-access memory. It is really read-write memory because ROM is also random access in the sense that given a random address the corresponding entry is read. RAM can be categorized by content duration. A static RAM's contents are always retained, as long as power is applied. A DRAM, on the other hand, needs to be refreshed every few milliseconds. Most RAMs by themselves are volatile, which means that without the power supply their content will be lost. All of the ROMs mentioned in the previous section are non-volatile. RAM can be made non-volatile by using a backup battery.

7.4.1 Static Random Access Memory (SRAM)

Figure 7.5 shows various SRAM memory cells (6T, 5T, 4T). The six transistor (6T) SRAM cell is commonly used SRAM. The crossed coupled inverters in a SRAM cell retain the information indefinitely, as long as the power supply is on since one of the pull-up transistors supplies current to compensate for the leakage current. During a read, bit and bit-bar line are pre-charged while the word enable line is held low. Depending on the content of the cell, one of the lines is discharged slightly causing the pre-charged voltage to drop when the word enable line is strobed. This difference in voltage between the bit and bit-bar lines is sensed by the sense amplifier, which produces the read result. During a write process, one of the bit/bit-bar lines is discharged, and by strobing the word enable line the desired data is forced into the cell before the word line goes away.

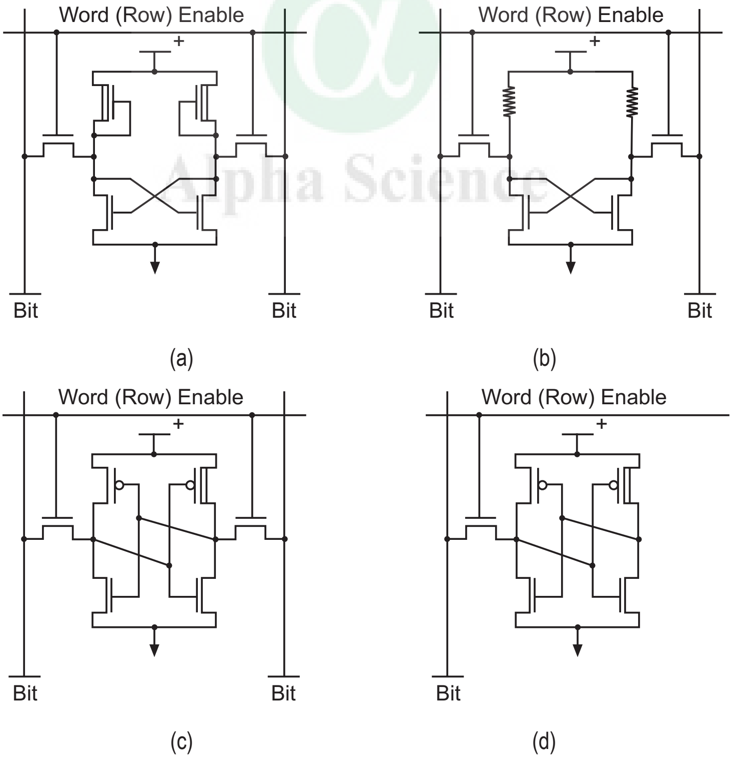


Fig. 7.5 Different SRAM cells: (a) six-transistor SRAM cell with depletion transistor load (b) four transistor SRAM cell with poly resistor load, (c) CMOS six-transistor SRAM cell, (d) five-transistor SRAM cell.

There is also a special type of SRAM cell used in computers to implement registers. These are called multiple port memories. In general, the contents can be read by many different requests at the same time. Figure 7.6 shows a dual-read port single-write port SRAM cell.

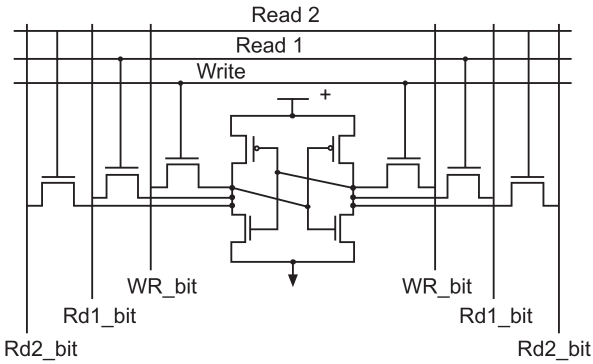


Fig. 7.6 Multi-ported CMOS SRAM cell (shown with 2-read and 1-write)

7.4.2 Dynamic Random Access Memory (DRAM)

The main disadvantage of SRAM is in its size since it takes six transistors (or at least four transistors and two resistors) to construct a single memory cell. Thus, the DRAM is used to improve the capacity. Another difference between DRAMs and SRAMs is in the number of address pins needed for a given size RAM. SRAM chips require all address bits to be given at the same time. DRAMs, however, utilize time-multiplex address lines. Only half of the address bits are given at a given time. They are divided by rows and columns. An extra control signal is thus required. This is the reason why DRAM chips have two address strobe signals: row address strobe (RAS) and column address strobe (CAS).

There are different DRAM cell designs. There is the four-transistor DRAM cell, three-transistor DRAM cell, and the one-transistor DRAM cell. Figure 7.7 shows the corresponding circuits for these cells. Data writing is accomplished in a three-transistor cell by keeping the RD line low (see Fig. 7.7(b)) while strobing the WR line with the desired data to be written is kept on the bus. If a 1 is desired to be stored, the gate of T2 is charged turning on T2. This charge remains on the gate of T2 for a while before the leakage current discharges it to a point where it cannot be used to turn on T2. When the charge is still there, a read can be performed by pre-charging the bus and strobing the RD line. If a 1 is stored, then both T2 and T3 are on during a read, causing the charge on the bus to be discharged. The lowering of voltage can be picked up by the sense amplifier. If a zero is stored, then there is no direct path from the bus to ground, thus the charge on the bus remains. To further reduce the area of a memory cell, a single transistor cell is often used. Figure 7.7 (c) shows the single transistor cell with a capacitor. Usually, two columns of cells are mirror images of each other to reduce the layout area.

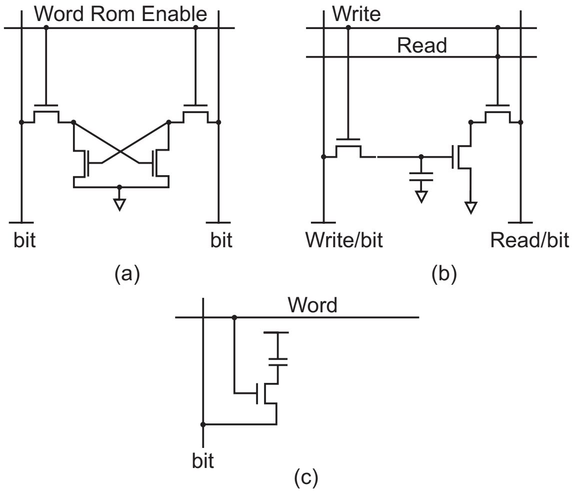


Fig. 7.7 Different DRAM cells: (a) four-transistor DRAM cell, (b) three-transistor DRAM cell, (c) one-transistor DRAM cell

7.5 MEMORY EXPANSION

In many memory applications, the required memory capacity, that is, the number of words and/or word size, cannot be satisfied by a single available memory IC chip. Therefore, several similar chips have to be combined to provide the desired number of words and/or word size.

Expanding word size: If it is required to have a memory of word size 'n' and the word size of the available memory chip is 'N' (n>N), then a number of similar chips can be combined together to achieve the desired word size. The number of chips required is an integer, next higher to the value n/N. These chips are to be connected in the following way:

- Connect the corresponding address lines of each chip individually, i,e., the first address pin of all the chips are to be connected together and it becomes the first pin of the overall memory. Similarly, connect other address lines together.
- Connect the READ input of each IC together and it becomes the read input for the overall memory. Similarly, connect the WRITE and CHIP SELECT inputs.

Now, the number of data input/output lines will be equal to the product of the number of chips used and the word size of each chip.

Expanding Word Capacity: Memory chip can be combined together to produce a memory, with the desired number of locations. To obtain a memory of capacity of '*m*' words, using the memory chip with 'M' words each, the number of chip required is an integer next higher to the value m/M. These chips are to be connected in the following way:

- Connect the corresponding address lines of each chip individually.

- Connect the READ input of each chip together. Similarly, connect the WRITE input.
- Use a decoder of proper size and connect each of its outputs to one of the CHIP SELECT terminals of the memory chips. For example, if eight chips are to be connected, a 3-line-to -8 line decoder is required to select one of the eight chips at any one time.

7.6 MAGNETIC BUBBLE MEMORY (MBM)

Magnetic bubble memory technology has advanced considerably since the concept was introduced by Bell Telephone Laboratories in 1967. Research indicated that small cylindrical magnetic domains, which are called magnetic bubbles, can be formed in single-crystal thin films of synthetic ferrites or garnets when an external magnetic field is applied perpendicularly to the surface of the film. These bubbles can be moved laterally through the film by using a varying magnetic field. These characteristics of magnetic bubbles make them ideally suited for serial storage of data bits; the presence or absence of a bubble in a bit position is used to define the logic state. Since the diameter of a bubble is so small (as little as a tenth of a micrometer), many thousands of data bits can be stored in a single bubble-memory chip. In 1977, Texas Instruments was the first to market a 92,304-bit bubble memory. This bubble memory is much like magnetic tape or magnetic disc memory storage in that it is nonvolatile meaning that the data is retained even when power is no longer applied to the chip. Since bubble memories are a product of solid-state technology (there are no moving parts), they have higher reliability than tape or disc storage and do not require any preventive maintenance. In addition, the bubble memory is small and lightweight and is, therefore, an excellent choice for compact designs and portable applications.

Functional operation of bubble memories: The basic bubble-memory package contains the bubble-memory chip, magnetic field coils, and permanent magnets as shown in Fig. 7.8. A rotating magnetic field created by two mutually perpendicular coils causes the data in the form of magnetic bubbles to move serially through the magnetic field in a manner similar to data in a semiconductor shift register. Two permanent magnets provide non-volatility and allow for the stable existence of magnetic-bubble domains. Interfacing circuits that are compatible with standard TTL devices complete the memory module to allow a convenient building-block concept for the nonvolatile memory system.

The chip is composed of a nonmagnetic crystalline substrate upon which a thin crystalline magnetic epitaxial film is grown. Only certain materials exhibit the properties necessary to form magnetic bubbles and these include ortho-ferrites, hexagonal ferrites, synthetic garnets, and amorphous metal films. Among these, the synthetic garnets have the best combination of the desired properties. Synthetic garnets support the formation of small magnetic bubbles that allow high-density data storage. The bubbles are highly mobile and are stable over a fairly wide range of temperatures.

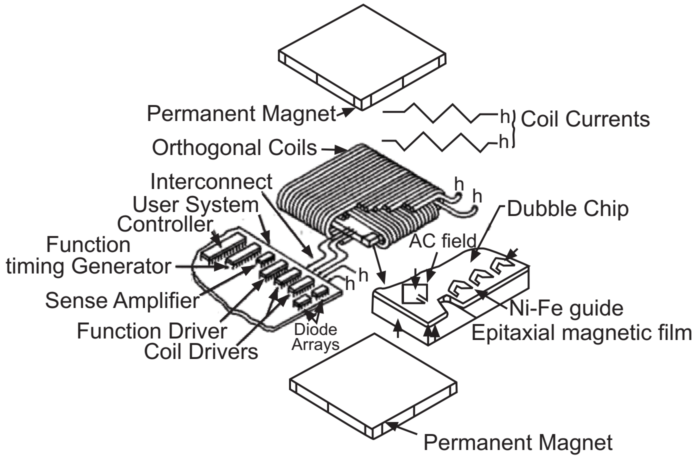


Fig. 7.8 Exploded view of magnetic bubble memory

The material chosen for the substrate depends on several factors. The crystalline structure should be compatible with that of the magnetic film, it should have nearly the same coefficient of expansion, and it should be nonmagnetic. The most-used garnet substrate with these properties is gadolinium gallium garnet (GGG). The magnetic film grown on this substrate has a crystalline structure that will allow the formation of magnetic domains (bubbles) in a plane perpendicular to the substrate. Without the influence of an external magnetic field, these magnetic domains form random serpentine patterns of equal area, minimizing the total magnetic energy of the magnetic film. The magnetic field of the serpentine domains tends to line up primarily along a single axis (the "easy" axis) that is perpendicular to the plane of the film. If an external magnetic field is applied, its energy tends to expand domains polarized in the direction of the field and to shrink those polarized opposite to the field until they become small cylinders embedded in a background of opposite magnetization. Viewed on end, these cylinders have the appearance of small circles or bubbles with diameters from 2 to 30 micrometers. Increasing the field further causes the bubble to collapse or to be "annihilated". The external field provides a bias that makes the bubbles stable. This bias, being a static field, can be readily provided by permanent magnets with no expenditure of power.

Before bubbles can be shifted through the magnetic film, they must be generated in accordance with input data. Bubbles are generated by locally altering the bias field with a magnetic field produced by a pulse of current through a microscopic one-turn metalized loop. This loop is located on a secondary layer immediately above the magnetic film on the surface of the chip. Given a current of the correct amplitude and polarity through the one-turn loop, a localized vertical magnetic field opposite to that of the permanent magnets is produced. This localized field establishes a domain wall inversion in the magnetic film resulting in bubble creation.

Once a bubble has been created, a method is then required to move the bubble domain along a predetermined path. This is accomplished by the deposition of chevron-shaped patterns of a soft magnetic material on the chip surface above the magnetic epitaxial film. When magnetized sequentially by a magnetic field rotating in the same plane, these chevron propagation patterns set up magnetic polarities that attract the bubble domain and establish motion. Fig. 7.9 shows the various polarities at different positions of the rotating magnetic field. In actual practice the rotating in-plane magnetic field is implemented by applying a two-phase alternating current to the two coils.

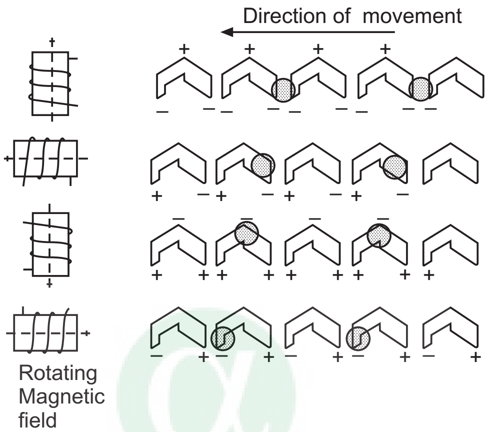


Fig. 7.9 Movement of magnetic bubbles under patterns of conductive chevrons

One possible implementation for the magnetic bubble memory is a long shift register. As shown in Fig. 7.10 the bubbles would shift under the influence of the rotating magnetic field following the path determined by the placement of chevron patterns. Even though this approach offers the simplest design and interface control, it suffers a major disadvantage of having the slowest access time. The reason for this is that after a data bit is entered or written it must circulate through the entire shift register before it can be retrieved or read. Another problem with this single loop design is that a single fault in the shift register structure produces a defective bubble memory chip. This results in a low processing yield and a high cost to the consumer.

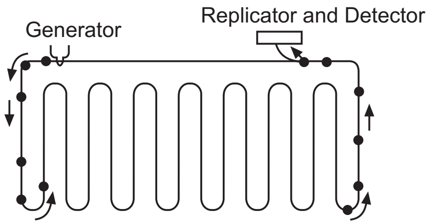


Fig. 7.10 MBM architecture using a single loop

For these reasons, Texas Instruments has chosen the major-minor loop architecture, which offers a dramatic improvement in access time. As shown in Fig. 7.11, during a write operation (data entry), data is generated one bit at a time in the major loop. The data is then transferred in parallel to the minor loops where it circulates until the next time data is to be read out of the memory.

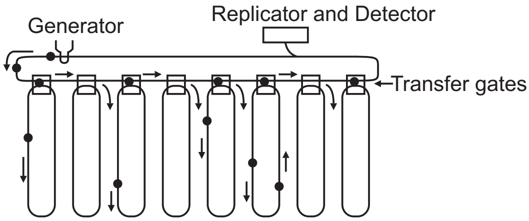


Fig. 7.11 MBM architecture using major and minor loops

During a write operation data are introduced into the major loop by pulses of current through the hairpin loop of the generator. The major loop is essentially a unidirectional circular shift register from which data can be transferred in parallel to the minor loops. Thus a block of data is entered in the major loop and shifted until the first data bit is aligned with the most remote minor loop. At that time, each parallel transfer element receives a current pulse that produces a localized magnetic field causing the transfer of all the bubbles in the major loop to the top bit position of the corresponding minor loop. Once data is written into the magnetic bubble memory, new data may be written only by first removing the old data by doing a destructive read. In this operation bubbles are transferred from the minor loops and annihilated by running them into the Permalloy guard rail that usually surrounds bubble devices.

During a read operation the data block to be accessed in the minor loops is rotated until it is adjacent to the major loop. At this time the data block is transferred in parallel to the major loop. The block of data is them serially shifted to the replicator where the data stream is duplicated. The duplicated data takes the path to the magneto-resistive detector element. The presence of a bubble in the detector lowers the resistance resulting in a corresponding increase in detector current, which can be detected via a sense amplifier. The original data stream remaining in the major loop is rotated and transferred back into the minor loops thus saving the data for further operations.

The magnetic-bubble-memory devices are fabricated using fine geometries that make the manufacture of perfect devices a difficult task. In order to increase production yields and achieve correspondingly lower costs, redundant minor loops on the bubble-memory chip allow some loops to be defective. Defective loops are determined at final test and a map of these loops is supplied to the end user so that the defective loops can be avoided in the final memory system. This redundancy of minor loops can be handled in several ways. The map could be written into a software program that would direct data to be stored only to the perfect minor loops, but this would require a unique software package for each memory system. Alternatively, the map could be stored in the MBM (magnetic-bubble memory) itself with some risk of being written over with new data. The recommended approach is to store the map in a programmable read-only memory (PROM). Each bit in a page of data would then be written to the MBM or read from it in accordance with the contents of the PROM, thus preventing data bits from the defective minor loops from mingling with valid data. Of course all this requires control circuitry in addition to that necessary for the timing and control of the alternating current in the field coils, the transfer of data to and from the minor loops, and the replication and detection of the magnetic bubbles.

Interfacing with bubble memories: The magnetic-bubble memory requires accurate current pulses for generate, replicate, and transfer operations, an interface circuit called a function driver is needed to convert the digital input control signals to the required current pulses. Also, the two field coils each require a triangular current drive 90 degrees out of phase with each other. This requirement is satisfied with another set of interface circuits (coil drivers and diode array) that is driven with digital input signals. The output signal amplitude of the MBM is relatively small, about 3 millivolts. For this to be useful in a system, the output is converted to standard TTL levels with the use of a set of interface circuits (RC networks and sense amplifiers). The block diagram in Figure 7.12 shows the connection of all these interface circuits as a memory module. This modular building block promotes efficient construction of mass memories.

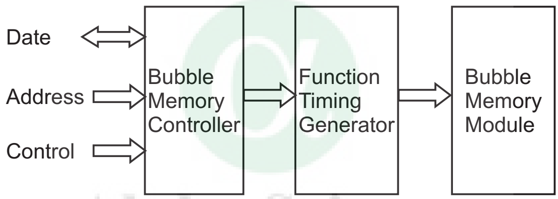


Fig. 7.12 Block diagram of bubble memory interface circuit

The control and timing signals for the memory module are derived from the function-timing generator. This integrated circuit provides input timing control to the function driver, coil drivers, and sense amplifier on a per-cycle basis. The function-timing generator provides control signals to the memory module. These signals provide control for five basic operations: generate, replicate, annihilate, transfer-in, and transfer-out. The function-timing generator also initiates the rotating magnetic field and precisely synchronizes the timing of other control signals with this field.

The time at which a particular data bit is detected in the MBM may not exactly match the time at which it is needed in the system. The sense amplifier not only increases the voltage level of the detected data, but also provides temporary storage of the data bits in a circuit called a D-type flip-flop. The sense amplifier receives a control input from the function timing generator to transfer the detected data into the internal flip-flop. In addition, the function-timing generator provides the control signals necessary to put the existing data in a known position during a power shut down. When the system is turned on again, the stored data can then be accurately located and retrieved.

In a typical system the major computing and data processing is done by a microprocessor. To provide a convenient interface from the microprocessor to the MBM system, a custom controller is needed for the read, write, and memory-addressing operations. The TMS5502/TMS9916 MBM controller responds to commands from the microprocessor system and sends control signals to the function timing generator necessary to access a page (or pages) of data. The controller maintains page-position information, handles serial-parallel data conversion between the bubble memory and the microprocessor and generates the control signals to the function-timing generator to perform read and write operations while handling the redundancy of the minor loops.

Advantages of bubble memories: The future growth of distributed process systems will be greatly impacted by magnetic-bubble memories. These microprocessor-based systems demand high-density mass storage at low cost. Magnetic-bubble memories satisfy all of these requirements with definite advantages over the existing magnetic storage technologies. MBM's advantages over moving-head disks or floppy disks are low access time (the time ncessary to retrieve the desired data), small physical size, low user entry cost, no maintenance, and higher reliability.

The advantages of MBM's over random-access memories (RAM's) are non-volatility, potentially lower price per bit, and more bits per chip. The RAM has the advantage of much better access time, higher transfer rate, and simpler interfacing.

The main MBM advantages are the low price disks for the low-end user, non-volatile semiconductor memories, and high-density storage in a small physical space. Because magnetic bubble memories are a solid-state, nonvolatile technology, they are ideally suited for portable applications as well as providing memory for traditional processing systems. Industrial applications include memory for numerical control machines and various types of process control. Solid-state bubble memories are more reliable in harsh environments; they are affected much less by shock, vibration, dirt, and dust than electromechanical magnetic memories. Innovative new products include data terminals, calculators, word processing, voice storage, and measurement equipment.

7.7 SURFACE STORAGE DEVICES

7.7.1 Magnetic Disk

A magnetic disk is a thin, circular metal plate/platter coated on both sides with a magnetic material. A disk pack consists of a number of these disks, three or more, mounted about half -an -inch apart from each other on a central shaft which rotates at a speeds of 2,400 or more revolutions per minute (rpm). Thus all disk of a disk pack move simultaneously in the same direction and at equal speed. Magnetic disks are the most popular medium for direct - access secondary storage.

In the disk pack, information is stored on both the surfaces of each disk plate except the upper surface of the top plate and lower surface of the bottom plate which are not used. As shown in the Fig. 7.13, each disk consists of a number of invisible concentric circles called tracks. A set of corresponding tracks in the entire surface is called a cylinder (Fig. 7.14) thus a disk pack having 10 disk plates will have 18 recording surfaces and hence it will have 18 tracks per cylinder. Each track is further subdivided into sectors (Fig. 7.15).

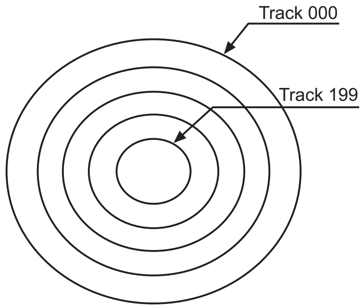


Fig. 7.13 Tracks on a magnetic disk

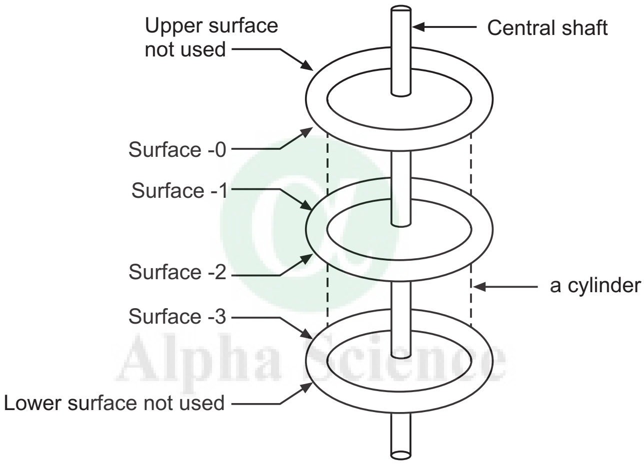


Fig. 7.14 Illustration of cylinders of a magnetic disk

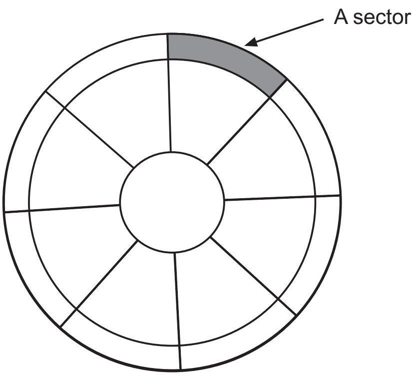


Fig. 7.15 Sectors of a disk

Information are recorded on the tracks of a disk surface in the form of invisible tiny magnetic spots. The presence of a magnetized spot represents a 1 bit and its absence represent a 0 bit. a standard binary code, usually 8- bit EBCDIC, is used for recording data in some systems, the outer tracks contains more bits than the inner tracks, because the circumference of the outer track is greater than the inner track .however in most of the systems, each track contains the same number of characters which means that the outer track of the disk are less densely packed with characters than those towards the centre.

The information stored on a disk can be read many times without affecting the stored data so the reading operation is non-destructive. But the writing of a new data, erases the previous data stored at that location of the disk. The data stored on the magnetic disk remains indefinitely until they are erased and reused at a future time.

The more disk surfaces of a particular disk pack has, the greater storage capacity. But the storage capacity of a disk system also depends on the tracks per inch of the surface and the bits per inch of the track.

The total number of bytes that can be stored in a disk pack = Number of cylinders x tracks per cylinder x sectors per tracts x bytes per sectors.

Data are recorded on the tracks of a spinning disk surface and read from the surface by one or more read/write heads. There are two basic types of disk systems viz, moving-head and fixed-head. The moving-head system consists of one read/write head for each disk surface mounted on an access arm which can be moved in and out (Fig. 7.16). So in this case, each read/write head moves horizontally across the surface of the disk, so that it is able to access each track individually. Each usable surface of the disk pack has its own head and all the heads move together. Information stored on the tracks which constitute a cylindrical shape through the disk packs are therefore accessed simultaneously.

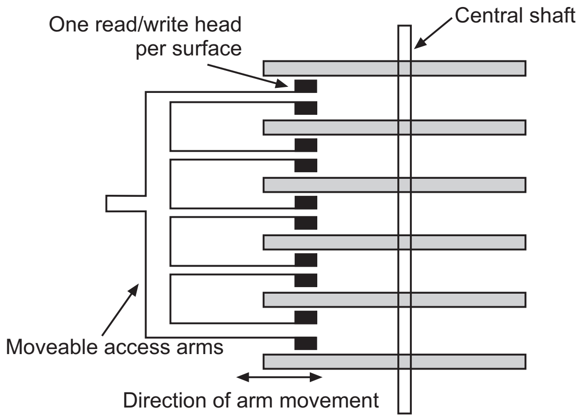


Fig. 7.16 Vertical cross section of a moving-head disk system

In the fixed-head system, the access arm is non-movable. A large number of read/write heads are distributed over the disk surfaces, one head for each track (Fig. 7.17). As a result, no head movement is required and therefore information is accessed more quickly. However, because of the space required for the additional read/write heads, fixed head disks have less capacity and cost more byte per data stored than moving head disks.

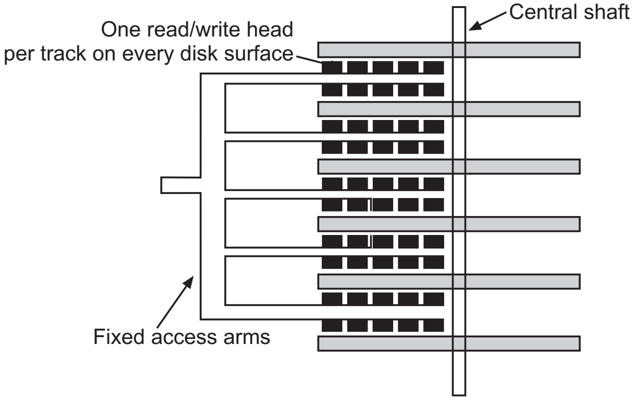


Fig. 7.17 Vertical cross section of a fixed-head disk system

7.7.2 CD-ROM

Large storage devices can also be implemented using optical means. The familiar compact disk (CD), used in audio systems, was the first practical application of this technology. Soon after, the optical technology was adapted with the computer environment to provide high-capacity read-only storage referred to as CD-ROM.

The technology exploited the possibility of using digital representation of analog sound signals. The first version was designed to hold up to 75 minutes, which requires a total of about 3×10^9 bits (3 Giga bits) of storage. Since then, higher – capacity devices have been developed. A video CD is capable of storing a full length movie. This requires approximately in the order of magnitude more bit storage capacity that that of audio CDs. Multimedia CDs are also suitable for storing large amounts of computer data.

The optical technology that is used for CD systems is based on a LASER light source. A LASER beam is directed on to the surface of the spinning disk. The grooves in the surface are arranged along the tracks of the disks. They reflect the focused beam towards a photo detector, which detects the stored binary patterns.

The LASER emits the coherent light beam that is sharply focused on the surface of the disk. Coherent light consists of synchronized waves that have the same wavelength. If a coherent light beam is combined with another beam of the same kind, and the beams are in phase, then the result will be a brighter beam. But, if the waves of the two beams are 180 degrees out of phase, they will cancel each other. Thus, if a photo detector is used to detect the beams, it will detect a bright spot in the first case and a dark spot in the second case.

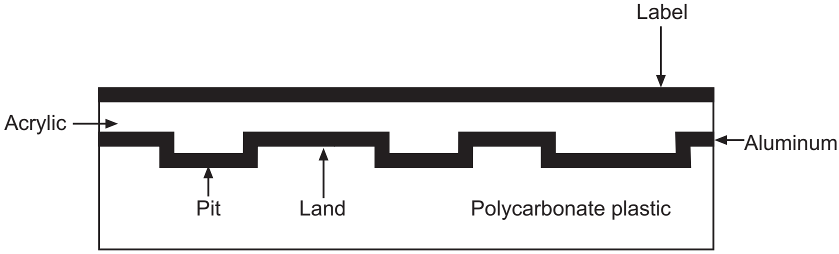


Fig. 7.18 Cross-section of an optical disc

A cross-section of a small section of a CD is shown in Fig. 7.18. The bottom layer is poly-carbonate plastic, which functions as a clear glass base. The surface of this plastic is programmed to store data by indenting it with pits. The un-indented parts are called lands. A thin layer of reflecting aluminium material is placed on the top of a programmed disk. The aluminium is covered by a protective acrylic. Finally, the top-most layer is deposited and stamped with a label. The total thickness of the disk is 1.2 mm. Almost all of it is contributed by the poly-carbonate plastic. The other layers are very thin.

The LASER source and the photo detector are positioned below the poly carbonate plastic. The emitted beam travels through this plastic, reflects off the aluminium layer, and travel back towards the photo detector. Note that from the LASER side, the pits actually appear as bumps with respect to the lands.

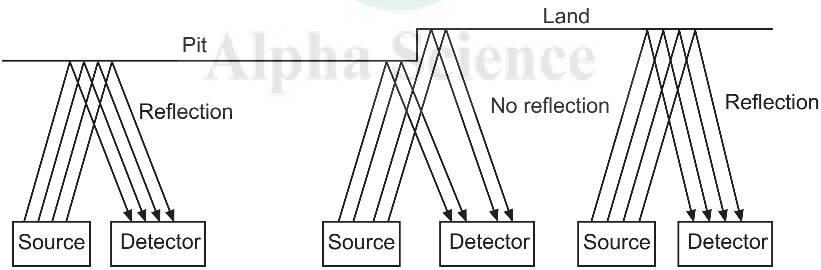


Fig. 7.19 Transition from pit to land

Figure 7.19. shows what happens as the LASER beam scans across the disk and encounters a transition from a pit to a land. Three different positions of the LASER source and the detector are shown, as could occur when the disk is rotating. When the light reflects solely from the pit, or solely from the land, the detector will see the reflected beam as bright spot. But, a different situation arises when the beams moves through the edge where the pit changes to the land, and vise versa. The pit is reduced to one quarter of the wavelength of the light. Thus, the reflected wave from the pit will be 180 degrees out of phase with the wave reflected from the land, cancelling each other. Hence, at the pit-land and land-pit transitions the detector will not see a reflected beam and will detect a dark spot.

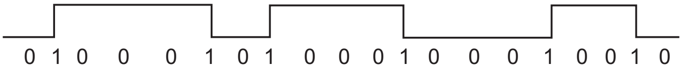


Fig. 7.20 Stored binary pattern

Figure 7.20 depicts several transitions between lands and pits. If each transition, detected as a dark spot is taken to denote the binary value 1 and the flat portions represents zeros, then the detected binary pattern will be as shown in the figure. This pattern is not a direct representation of the stored data. CDs use a complex encoding scheme to represent data. Each byre to data is represented by a 14-bit code, which provides considerable error detection capacity. We will not delve into details of this code.

The pits are arranged along tracks on the surface of the disk. Actually, there is just one physical track, spiralling from the middle of the disk toward the outer edge. But, it is customary to refer to each circular path spanning 360 degrees as a separate track, which is analogous to the terminology used for magnetic disks. The CD is 120 mm in diameter. There is a 15 mm hole in the center. Data are stored on tracks that cover the area from 25 mm radius to 58 mm radius. The space between the tracks is 1.6 microns. Pits are 0.5 microns wide and 0.8 to 3 microns long.

Stored data are organized on CD-ROM tracks in the form of blocks that are called sectors. There are several different formats for a sector. One format, known as mode-1, uses 2352 byte sectors. There is a 16 byte header that contains a synchronization field used to detect the beginning of the sector and addressing information used to identify the sector. This is followed by 2048 bytes of stored data. At the end of the sector, there are 288 bytes used to implement the error correcting scheme. The number of sectors per track is variable; there are more sectors on the longer outer tracks.

Error detection and correction is done at more than one level. Each byte of stored information is encoded using a 14-bit code that has some error correcting capability. This code can correct single bit errors. Errors that occur in short burst, affecting several bits, are detected and corrected using the error checking bits at the end of the sector.

CD-ROM drives operate at a number of different rotational speeds. The basic speed, known as 1X, is 75 sectors per second. This provides a data rate of 153600 bytes/s (150 Kbytes/s), using the mode-1 format. Note that the speed of the drive affects only the data transfer rate but not the storage capacity of the disk. Higher speed CD-ROM drives are identified in relation with the basic speed. Thus, a 40X CD-ROM has a data transfer rate that is 40 times higher than that of the 1X CD-ROM.

7.8 SPECIAL MEMORIES

7.8.1 Flash Memory and Memory Sticks

Flash memory is rewritable memory chip that holds its contents without power. It is used for easy and fast information storage in cell phones, digital cameras, and video game consoles. Flash memory works much faster than traditional EEPROMs because it writes data in chunks, usually 512 bytes in size, instead of 1 byte at a time. One of the most common uses of flash memory is for the basic input/output system (BIOS) of a computer. The task of the BIOS it to make sure that all the other devices, i.e., hard drives, ports, microprocessor, etc., function together and properly.

A memory stick is a device that uses flash memory. It was first introduced in 1998. A memory stick records various types of digital content, allows sharing of content among a variety of digital products, and it can be used for a broad range of applications. Generally, all memory stick media are preformatted by the manufacturer and they are readily available for immediate use. The manufacturer provides reformatting instructions in the event that the user wishes to reformat the memory stick at a later date. It is also possible to transfer data from a memory stick to a PC through a memory stick USB reader/writer. The advantages of flash memory over a hard disk are that flash memory is noiseless, provides faster access, smaller size and weight, and has no moving parts. However, the big advantage of a hard disk is that the cost per megabyte for a hard disk is considerably cheaper, and the its capacity is substantially higher than that of a flash memory device.

7.8.2 Cache Memory

Cache memory is essentially a fast storage buffer in the microprocessor of a computer. Caching refers to the arrangement of the memory subsystem in a typical computer that allows us to do our computer tasks more rapidly. Thus, the main purpose of a cache is to accelerate the computer while keeping the price of the computer low. Modern computers are bundled with both L1 and L2 caches and these terms are explained below.

Let us consider the case where in a typical computer the main memory (RAM) access time is 20 nanoseconds and the cycle time of the microprocessor is 0.5 nanosecond. Without cache memory, the microprocessor is forced to operate at the RAM's access time of 20 nanoseconds. Now, let us assume that a microprocessor is build with a small amount of memory within in addition to the other components, and that memory has an access time of 5 nanosecond, the same as the microprocessor's cycle time. This memory is referred to as Level 1 cache or L1 cache for short. Let us also suppose that, in addition to L1 cache, we install memory chips on the motherboard that have an access time of 10 nanoseconds. This memory is referred to as Level 2 cache or L2 cache for short.

Some microprocessors have two levels of cache built right into the chip. In this case, the motherboard cache becomes Level 3 cache, or L3 cache for short. Cache can also be built directly on peripherals. Modern hard disks come with fast memory, hardwired to the hard disk. This memory is controlled by the hard disk controller. Thus, as far as the operating system is concerned, these memory chips are the disk itself. When the computer asks for data from the hard disk, the hard disk controller checks into this memory before moving the mechanical parts of the hard disk (which is very slow compared to memory). If it finds the data that the computer asked for in the cache, it will return the data stored in the cache without actually accessing data on the disk itself, saving a lot of time.

7.8.3 Virtual Memory

Virtual memory refers to a scheme where the operating system frees up space in RAM to load a new application. Virtual memory can be thought of as an alternative form of memory caching. For instance, let us suppose that the RAM is full with several applications open at a particular time.

Without virtual memory, we will get a message stating that the RAM is full and we cannot load another application unless we first close one or more already loaded applications. However, with virtual memory, the operating system looks at RAM for files that have not used recently, and copies them onto the hard disk. This creates space for another application to be loaded. The copying from RAM to the hard disk occurs automatically, and the user is unaware of this operation. Therefore, to the user it appears that the computer has unlimited RAM space. Virtual memory provides the benefit that hard disk space can be used in lieu of a large amount of RAM since the hard disk costs less. The area of the hard disk that stores the RAM image is called a page file. It holds pages of RAM on the hard disk, and the operating system moves data back and forth between the page file and RAM. On Windows platforms, page files have the SWP extension.

Practically, all operating systems include a virtual memory manager to enable the user to configure virtual memory manually in the event that we work with applications that are speed critical and our computer has two or more physical hard disks. Speed may be a serious concern since the read/write speed of a hard drive is much slower than RAM, and the technology of a hard drive is not designed for accessing small pieces of data at a time. If our system has to rely too heavily on virtual memory, we will notice a significant performance drop. Our computer must have sufficient RAM to handle everything we need to work with. Then, the only time we may sense the slowness of virtual memory would be when there is a slight pause when we are swapping tasks. In this case, the virtual memory allocation is set properly. Otherwise, the operating system is forced to constantly swap information back and forth between RAM and the hard disk, and this occurrence is known as thrashing, this, can make our computer operate at very slow speed.

7.8.4 Scratch Pad Memory

A scratch pad memory is a usually high speed internal register used for temporary storage of preliminary data or notes. It is a region of reserved memory in which programs store status data. Scratch pad memory is fast SRAM memory that replaces the hardware managed cache and may be used to transfer variables from one task to another.

7.8.5 Charged-Coupled Device (CCD)

A CCD is an electrical device that is used to create images of objects, store information (analogous to the way a computer stores information), or transfer electrical charge (as part of larger device). It receives as input light from an object or an electrical charge. The CCD takes this optical or electronic input and converts it into an electronic signal - the output. The electronic signal is then processed by some other equipment and/or software to either produce an image or to give the user valuable information.

Refer Fig. 7.21. A CCD chip is a metal oxide semiconductor (MOS) device. This means that its base, which is constructed of a material is a conductor under certain conditions, is topped with a layer of a metal oxide. In the case of the CCD, usually silicon is used as the base material and silicon dioxide is used as the coating. The final, top layer is also made of polysilicon.

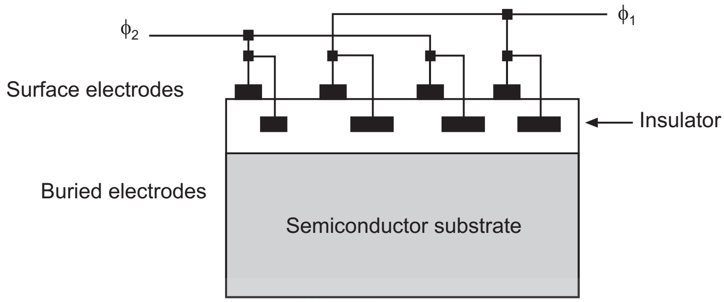


Fig. 7.21 CCD structure

This silicon that forms the base and the top layer is doped with, or made to contain, a small amount of some other material. Doping endows materials with special properties that can be exploited through different electrical means. A material with extra electrons is called an *n*-type material. In a *n*-type material, the doping element has five available electrons, so it makes the four usual bonds, but it has an extra electron left over. It is important to note that these materials are all neutral, and that extra electrons or extra holes in this case do not make the materials charged but merely come from what is left over or needed for a neutral atom to form four bonds.

Upon application of the right stimulus, the movement of the hole can be directed. This is one of the fundamental keys to the operation of the CCD. An electron is repelled by negative charge and is attracted by positive charge. A hole, however, is repelled by the positive charge and is attracted by negative charge. As we can control the motion of electrons by applying different electrical fields or charges in the vicinity, so we can control the motion of holes.

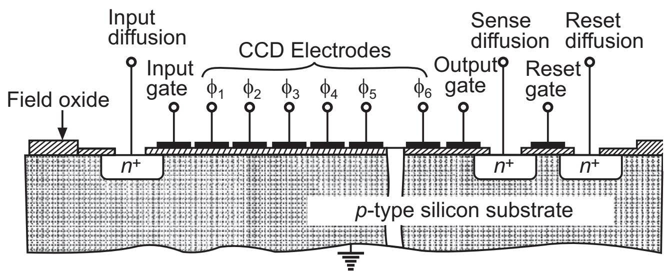


Fig. 7.22 CCD system with input and output circuitry

The diagrammatic representation of a CCD system with input and output circuitry is shown in Fig. 7.22. In the figure, n^+ denotes heavily doped *n*-region. The importance of applying voltages to the depletion region (called biasing the *p-n* junction) is that it precisely allows us to control applied current through any *p-n* material. When the *p-n* junction is reversed-biased, an only infinitesimal amount of applied current can flow, which for all practical purposes is zero. This corresponds to the “off” state. When the *p-n* junction is forward-biased, current easy flows through the junction (because the smaller electric field does not impede the flow of charges as much). In fact, by plotting a graph of applied voltage versus current flow - an I-V Curve - we can see that the dependence of current flow on applied voltage across the junction.

In a CCD, the electrical field at different parts of the surface is controlled by an array or matrix of electrodes; these electrodes are called the gates. (CCD arrays can be either one-dimensional or two-dimensional, but here we will consider the one-dimensional array in detail, and then apply that information to understand the two-dimensional array). This array of electrodes biases each small part of the surface differently, which allows any flow of charge on the CCD to be controlled.

The surface of the CCD is further broken down into smaller regions called pixels, or picture elements. The concept of a pixel in a CCD is shown in Fig. 7.23. This name pixel is appropriate because they represent a single “grain” of the imaged object (just like you can see that your TV images appear to be made up of tiny “grains”). The array of electrodes apply a positive potential, $+V_g$ to two-thirds of each pixel, thus forward-biasing that portion of the pixel. Let’s represent the first third of the pixel by (1, the second third by (2, and the last third by (3. So, (1 and (3 are at a positive potential of $+V_g$, and (2 is at a lower potential, V_s.

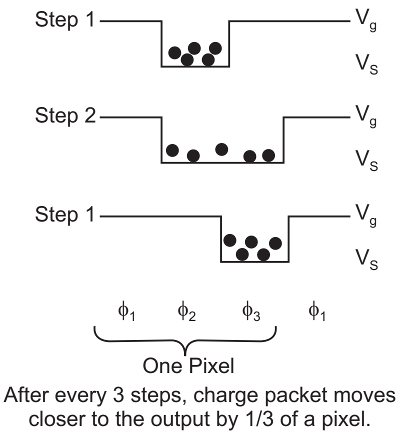


Fig. 7.23 Concept of a pixel in a CCD

When light or photons of high enough energy strike the surface, electrons are usually liberated from the surface. (The quantum efficiency or the ratio of electrons liberated per incident photon is about 0.70-0.80). For every electron liberated, a hole is created simply by the act of the electron leaving. Thus, incident photons create electron-hole pairs. The hole, being effectively positive, is repelled by the applied positive potential (1 and (3, and eventually escapes into the base of the chip. The electron, however, is captured in the nearest potential well (2. The more light incident on a pixel, the more electrons captured in the potential wells. Thus, differences in the intensity of incoming light are "recorded" by the number of electrons collected in each potential well. So now the challenge is to extract information from these "electron-collecting bins" (which may also be thought of as tiny capacitors). To do this, the charge packets (the collection of electrons in each well) must be transferred to another device for data processing. This is accomplished by sequentially changing the applied voltage at the three parts of each pixel. First, the level of the potential barrier (V_3) closest to the data processing device is lowered to the same potential as (2. This causes the electrons to divide between the two wells. The primary mechanism for this electron diffusion is induced self-drift from Coulomb repulsions, which acts to separate the charge. Then, the potential of (2 is raised over a finite time interval (corresponding to the diffusion rate of the electrons from (2 to (3) so that (2 now becomes a potential barrier. The remaining charge is transferred from (2 to (3 by these changes in potential. (1 is maintained at constant potential during this entire process to keep the charge packets separate from one another. Now, each charge packet in the row has moved over one-third of a pixel closer to the data processing device. This cycle is repeated over and over in fractions of a second to transfer all the charge off the chip to a detector which usually uses a load resistance to measure the amount of charge collected in each "bin". This is how the three-phase CCD works. The term charge-coupling in charged-coupled device comes from the coupling of electrical potentials.[4] A two-dimensional CCD is composed of channels, or rows along which charge is transferred. Each channel is essentially a one-dimensional CCD array. Charge is prevented from moving sideways by channel stops, which are the narrow barriers between the closely spaced channels of the CCD.

CCDs are used in a variety of different imaging devices. Linear imaging devices (LIDs) employing CCDs are typically used in facsimile machines, photocopiers, mail sorters, and bar code readers, in addition to being used for aerial mapping and reconnaissance. Area imaging devices (AIDs) employing CCDs are used for closed-circuit television cameras, video cameras, and vision systems for robotics and as film for astronomical observations. CDs are also used in drug discovery in combination with DNA arrays in a process called Multi-Array Plate Screening (MAPS).

Non-imaging applications of CCDs include signal processing, memory applications, video integration, transversal filtering, and MTI filtering for radar. Again, non-imaging applications fall under the categories of either signal processing or delay line applications.

REVIEW QUESTIONS

1. Write short note on basic memory organization.
2. Explain in detail about Read Only Memories.
3. Classify ROMs.
4. Classify RAMs.
5. Explain in detail about the principle of operation of Dynamic Random Access Memory (DRAM).
6. What is the operating principle of magnetic bubble memory? Explain.
7. What are the advantages of magnetic bubble memories?
8. Explain how read and write operations are performed in a CD-ROM?
9. Write short note on flash memories.
10. What is the role of cache memory in computer systems?
11. How virtual memory improves the performance of a computer system?

CHAPTER

8

DATA COMMUNICATION

8.1 INTRODUCTION

The distance over which data moves within a computer may vary from a few thousandths of an inch, as is the case within a single IC chip, to as much as several feet along the backplane of the main circuit board. Over such small distances, digital data may be transmitted as direct, two-level electrical signals over simple copper conductors. Except for the fastest computers, circuit designers are not very concerned about the shape of the conductor or the analog characteristics of signal transmission.

Frequently, however, data must be sent beyond the local circuitry that constitutes a computer. In many cases, the distances involved may be enormous. Unfortunately, as the distance between the source of a message and its destination increases, accurate transmission becomes increasingly difficult. This results from the electrical distortion of signals traveling through long conductors, and from noise added to the signal as it propagates through a transmission medium. Although some precautions must be taken for data exchange within a computer, the biggest problems occur when data is transferred to devices outside the computer's circuitry. In this case, distortion and noise can become so severe that information is lost.

Data Communications concerns the transmission of digital messages to devices external to the message source. "External" devices are generally thought of as being independently powered circuitry that exists beyond the chassis of a computer or other digital message source. As a rule, the maximum permissible transmission rate of a message is directly proportional to signal power and inversely proportional to channel noise. It is the aim of any communications system to provide the highest possible transmission rate at the lowest possible power and with the least possible noise.

8.2 DIGITAL DATA TRANSMISSION

The digitization starts with the conversion of analog voice signals into a digital format. An analog can be converted into a digital signal of equal quality if the analog signal is sampled at a rate that corresponds to at least twice the signal's maximum frequency. A technique for converting an analog signal to a digital form is PCM, which requires three operations:

1. **Sampling:** This operation converts the continuous analog signal into a set of periodic pulses, the amplitudes of which represent the instantaneous amplitudes of the analog signal at the sampling instant. Thus, the process of sampling involves reading of input signals at discrete points in time. Hence, the sampled signals consist in electrical pulses, which vary in accordance with the amplitude of the input signal. In accordance with the Nyquist sampling rate, an analog signal of bandwidth B Hz must be sampled at a rate of at least 1/2B to preserve its wave shape when reconstructed.
2. **Quantizing:** This technique is the process of representing the continuous amplitude of the samples by a finite set of levels. If V quantizer levels are employed to represent the amplitude range it take the $\log_2$ V bit to code each sample. In voice transmission 256 quantized levels are employed, hence each sample is coded using $\log_2 256 = 8$ bits and thus, the digital bit rate is 8,000 × 8 = 64,000 *b/s*. Thus, the process of quantization introduces distortion into the signal, making the received voice signals raspy and hoarse. This type of distortion is known as quantization noise, which is only present during speech. When a large number of quantization steps, each of ΔS volts, are used to quantize a signal having an rms signal level Srms, the signal-to-quantization noise ratio is given by:

   $$S_{qn} = (S_{rms})^2/(\Delta S)^2/12$$

   A large number of bits are necessary to provide an acceptable signal-to-quantization noise ratio throughout the dynamic amplitude range. Some analysis of speech signals shows that smaller amplitude levels have a much higher probability of occurrence than high levels.
3. **Coding:** This solution protects message signals from impairment by adding redundancy to the message signal. Another important approach in digital coding of analog signals is Differential PCM. The difference between the analog input signals and their approximation at the sampling instant is quantized into voltage (V) levels and the output of the encoder is coded into log2 V bits. In such a way, it combines the simplicity of Delta Modulation and the multilevel quantizing feature of PCM and in many applications can provide good reproduction of analog signals comparable to PCM, with a considerable reduction in the digital bit rate.

Quite often we have to send digital data through analog transmission media such as a telephone network. In such situations it is essential to convert digital data to analog signal. Basic approach is shown in Fig. 8.1. This conversion is accomplished with the help of special devices such as modem (modulator-demodulator) that converts digital data to analog signal and vice versa. Since modulation involves operations on one or more of the three characteristics of the carrier signal, namely amplitude, frequency and phase, three basic encoding or modulation techniques are available for conversion of digital data to analog signals. The three techniques, referred to as amplitude shift keying (ASK), frequency shift keying (FSK) and phase shift keying (PSK) and those will be discussed later in this chapter.

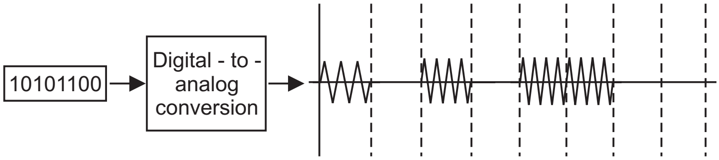


Fig. 8.1 Conversion of digital data to analog signal

8.3 TRANSMISSION MEDIA

Transmission media can be defined as physical path between transmitter and receiver in a data transmission system.

- **Guided:** Transmission capacity depends critically on the medium, the length, and whether the medium is point-to-point or multipoint (e.g. LAN). Examples are co-axial cable, twisted pair, and optical fibre.
- **Unguided:** provides a means for transmitting electro-magnetic signals but do not guide them. Example: wireless transmission.

Characteristics and quality of data transmission are determined by medium and signal characteristics. For guided media, the medium is more important in determining the limitations of transmission. While in case of unguided media, the bandwidth of the signal produced by the transmitting antenna and the size of the antenna is more important than the medium. Signals at lower frequencies are omni-directional (propagate in all directions). For higher frequencies, focusing the signals into a directional beam is possible. These properties determine what kind of media one should use in a particular application. In this lesson we shall discuss the characteristics of various transmission media, both guided and unguided.

8.3.1 Guided Transmission Media

In this section we shall discuss about the most commonly used guided transmission media such as twisted-pair of cable, coaxial cable and optical fiber.

8.3.1.1 *Twisted Pair*

In twisted pair technology, two copper wires are strung between two points. The two wires are typically “twisted” together in a helix to reduce interference between the two conductors as shown in Fig.8.2. Twisting decreases the cross-talk interference between adjacent pairs in a cable. Typically, a number of pairs are bundled together into a cable by wrapping them in a tough protective sheath.

Twisted cables can carry both analog and digital signals. Actually, they carry only analog signals. However, the “analog” signals can very closely correspond to the square waves representing bits, so we often think of them as carrying digital data.

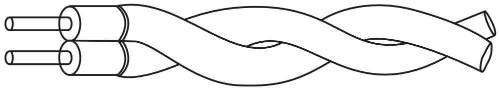


Fig. 8.2 Twisted cable

Data rate determined by wire thickness and length. In addition, shielding to eliminate interference from other wires impacts signal-to-noise ratio, and ultimately, the data rate.

Characteristics: Twisted-pair can be used for both analog and digital communication. The data rate that can be supported over a twisted-pair is inversely proportional to the square of the line length. Maximum transmission distance of 1 km can be achieved for data rates up to 1 Mb/s. For analog voice signals, amplifiers are required about every 6 km and for digital signals, repeaters are needed for about 2 km. To reduce interference, the twisted pair can be shielded with metallic braid. This type of wire is known as Shielded Twisted-Pair (STP) and the other form is known as Unshielded Twisted-Pair (UTP).

8.3.1.2 *Base band Coaxial*

With "coax", the medium consists of a copper core surrounded by insulating material and a braided outer conductor as shown in Fig. 8.3. The term base band indicates digital transmission (as opposed to broadband analog).

Fig. 8.3 Co-axial cable

Physical connection consists of metal pin touching the copper core. There are two common ways to connect to a coaxial cable:

1. **Vampire taps,** is a metal pin is inserted into the copper core. A special tool drills a hole into the cable, removing a small section of the insulation and a special connector is screwed into the hole. The tap makes contact with the copper core.
2. **With a T-junction**, the cable is cut in half and both halves connect to the T-junction. A T-connector is analogous to the signal splitters used to hook up multiple TVs to the same cable wire.

Characteristics: Co-axial cable has superior frequency characteristics compared to twisted-pair and can be used for both analog and digital signalling. In baseband LAN, the data rates lies in the range of 1 KHz to 20 MHz over a distance in the range of 1 Km. Co-axial cables typically have a diameter of 3/8". Coaxial cables are used both for *baseband* and *broadband* communication. For broadband CATV application coaxial cable of 1/2" diameter and 75 Ω impedance is used. This cable offers bandwidths of 300 to 400 MHz facilitating high-speed data communication with low bit-error rate. In broadband signalling, signal propagates only in one direction, in contrast to propagation in both directions in baseband signalling. Broadband cabling uses either dual-cable scheme or single-cable scheme with a head-end to facilitate flow of signal in one direction. Because of the shielded, concentric construction, co-axial cable is less susceptible to interference and cross talk than the twisted-pair. For long distance communication, repeaters are needed for every kilometer or so. Data rate depends on physical properties of cable, but 10 Mbps is typical.

8.3.1.3 *Broadband Coaxial*

The term broadband refers to analog transmission over coaxial cable. (Note, however, that the telephone folks use broadband to refer to any channel wider than 4 kHz). The technology generally uses analogue signalling and is used in cable television.

Total available spectrum typically divided into smaller channels of 6 MHz each. That is, to get more than 6MHz of bandwidth, you have to use two smaller channels and somehow combine the signals. Requires amplifiers to boost signal strength; because amplifiers are one way, data flows in only one direction. Two types of broadband coaxial systems have emerged:

1. **Dual cable systems** use two cables, one for transmission in each direction: One cable is used for receiving data and second cable used to communicate with head-end. When a node wishes to transmit data, it sends the data to a special node called the head-end. The head-end then resends the data on the first cable. Thus, the head-end acts as a root of the tree, and all data must be sent to the root for redistribution to the other nodes.
2. **Midsplit** systems divide the raw channel into two smaller channels, with each sub channel having the same purpose as above.

8.3.1.4 *Fiber Optics*

In fiber optic technology, the medium consists of a hair-width strand of silicon or glass, and the signal consists of pulses of light. For instance, a pulse of light means “1”, lack of pulse means “0”. It has a cylindrical shape and consists of three concentric sections: the core, the cladding, and the jacket as shown in Fig. 8.4.

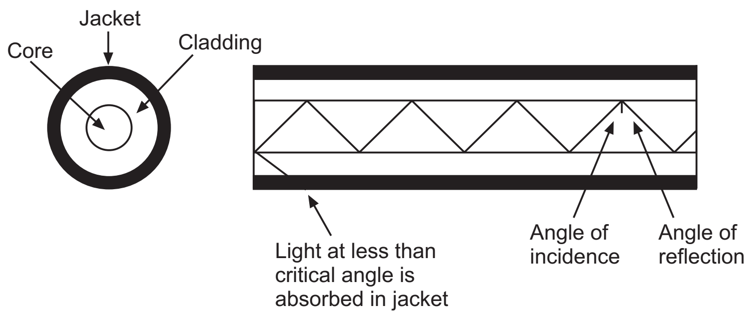


Fig. 8.4 Optical Fiber

The core, innermost section consists of a single solid dielectric cylinder of diameter d1 and of refractive index n1. The core is surrounded by a solid dielectric cladding of refractive index n2 that is less than n1. As a consequence, the light is propagated through multiple total internal reflection. The core material is usually made of ultra pure fused silica or glass and the cladding is either made of glass or plastic. The cladding is surrounded by a jacket made of plastic. The jacket is used to protect against moisture, abrasion, crushing and other environmental hazards.

Three components of an optical fibre communication system are:

1. **Fibre medium:** carries light pulses for tremendous distances (e.g., 100s of kilometers) with virtually no signal loss.
2. **Light source:** typically a Light Emitting Diode (LED) or laser diode. Running current through the material generates a pulse of light.
3. **A photo diode light detector,** which converts light pulses into electrical signals.

Advantages:

1. Very high data rate, low error rate. 1000 Mbps (1 Gbps) over distances of kilometers common. Error rates are so low they are almost negligible.
2. Difficult to tap, which makes it hard for unauthorized taps as well. This is responsible for higher reliability of this medium.
3. Much thinner (per logical phone line) than existing copper circuits. Because of its thinness, phone companies can replace thick copper wiring with fibers having much more capacity for same volume. This is important because it means that aggregate phone capacity can be upgraded without the need for finding more physical space to hire the new cables.
4. Not susceptible to electrical interference (lightning) or corrosion (rust).
5. Greater repeater distance than coax.

Disadvantages:

1. Difficult to tap: It really a point-to-point technology. In contrast, tapping into coax is trivial. No special training or expensive tools or parts are required.
2. One-way channel: Two fibers needed to get full duplex (both ways) communication.

Optical fibers are available in two varieties; *Multi-Mode Fiber* (*MMF*) and *Single-Mode Fiber* (*SMF*). For multi-mode fiber the core and cladding diameter lies in the range 50-200μm and 125-400μm, respectively. Whereas in single-mode fiber, the core and cladding diameters lie in the range 8-12μm and 125μm, respectively. Single-mode fibers are also known as Mono-Mode Fiber. Moreover, both single-mode and multi-mode fibers can have two types; *step index* and *graded index*. In the former case the refractive index of the core is uniform throughout and at the core cladding boundary there is an abrupt change in refractive index. In the later case, the refractive index of the core varies radially from the centre to the core-cladding boundary from n1 to n2 in a linear manner. Fig. 8.5 shows the optical fibre transmission modes.

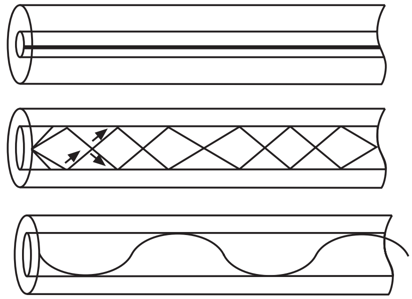


Fig. 8.5 Schematics of three optical fiber types, (a) Single-mode step-index, (b) Multi-mode step-index, and (c) Multi-mode graded-index

Characteristics: Optical fiber acts as a dielectric waveguide that operates at optical frequencies (10^{14} to 10^{15} Hz). Three frequency bands centered around 850,1300 and 1500 nanometers are used for best results. When light is applied at one end of the optical fiber core, it reaches the other end by means of total internal reflection because of the choice of refractive index of core and cladding material (n1 > n2). The light source can be either light emitting diode (LED) or injection laser diode (ILD). These semiconductor devices emit a beam of light when a voltage is applied across the device. At the receiving end, a photodiode can be used to detect the signal-encoded light. Either PIN detector or APD (Avalanche photodiode) detector can be used as the light detector.

In a multi-mode fiber, the quality of signal-encoded light deteriorates more rapidly than single-mode fiber, because of interference of many light rays. As a consequence, single-mode fiber allows longer distances without repeater. For multi-mode fiber, the typical maximum length of the cable without a repeater is 2km, whereas for single-mode fiber it is 20km.

Uses of Fibre: Because of greater bandwidth (2Gbps), smaller diameter, lighter weight, low attenuation, immunity to electromagnetic interference and longer repeater spacing, optical fiber cables are finding widespread use in long-distance telecommunications. Especially, the single mode fiber is suitable for this purpose. Fiber optic cables are also used in high-speed LAN applications. Multi-mode fiber is commonly used in LAN.

- Long-haul trunks-increasingly common in telephone network (Sprint ads)
- Metropolitan trunks-without repeaters (average 8 miles in length)
- Rural exchange trunks-link towns and villages
- Local loops-direct from central exchange to a subscriber (business or home)
- Local area networks-100Mbps ring networks.

8.3.2 Unguided Transmission

Unguided transmission is used when running a physical cable (either fiber or copper) between two end points is not possible. For example, running wires between buildings is probably not legal if the building is separated by a public street.

Infrared signals typically used for short distances (across the street or within same room). Microwave signals commonly used for longer distances (10's of km). Sender and receiver use some sort of dish antenna as shown in Fig. 8.6.

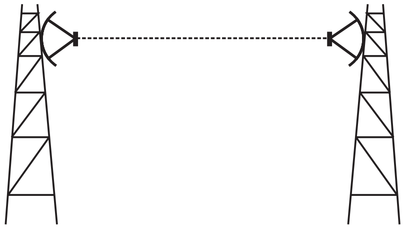


Fig. 8.6 Communication using Terrestrial Microwave

Disadvantages

1. Weather interferes with signals. For instance, clouds, rain, lightning, etc., may adversely affect communication.
2. Radio transmissions are easy to tap. A big concern for companies worried about competitors stealing plans.
3. Signals bouncing off of structures may lead to out-of-phase signals that the receiver must filter out.

8.3.2.1 *Satellite Communication*

Satellite communication is based on ideas similar to those used for line-of-sight. A communication satellite is essentially a big microwave repeater or relay station in the sky. Microwave signals from a ground station is picked up by a transponder, amplifies the signal and rebroadcasts it in another frequency, which can be received by ground stations at long distances as shown in Fig. 8.7.

To keep the satellite stationary with respect to the ground based stations, the satellite is placed in a geostationary orbit above the equator at an altitude of about 36,000 km. As the spacing between two satellites on the equatorial plane should not be closer than 40, there can be 360/4 = 90 communication satellites in the sky at a time. A satellite can be used for point-to-point communication between two ground-based stations or it can be used to broadcast a signal received from one station to many ground-based stations as shown in Fig. 8.8. Number of geo-synchronous satellites limited (about 90 total, to minimize interference). International agreements regulate how satellites are used, and how frequencies are allocated. Weather affects certain frequencies. Satellite transmission differs from terrestrial communication in another important way: One-way *propagation delay* is roughly 270 ms. In interactive terms, propagation delay alone inserts a 1 second delay between typing a character and receiving its echo.

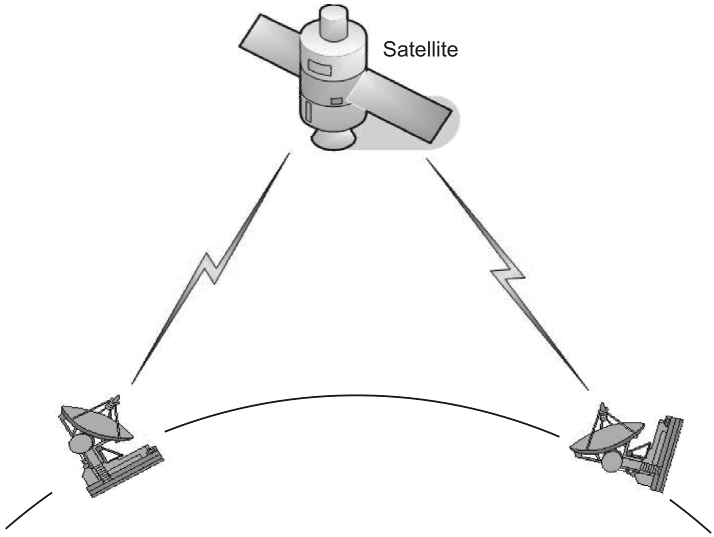


Fig. 8.7 Satellite Microwave Communication: point-to-point

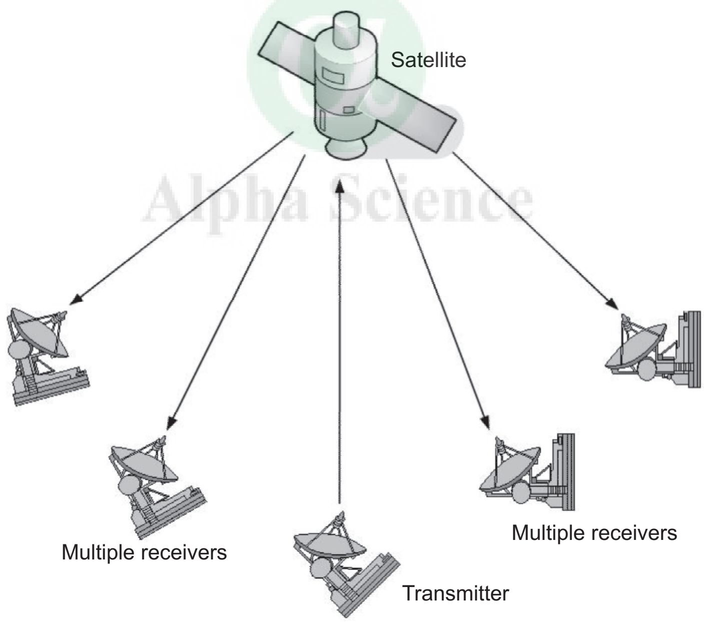


Fig. 8.8 Satellite Microwave Communication: Broadcast links

Characteristics: Optimum frequency range for satellite communication is 1 to 10 GHz. The most popular frequency band is referred to as 4/6 band, which uses 3.7 to 4.2 GHz for down link and 5.925 to 6.425 for uplink transmissions. The 500 MHz bandwidth is usually split over a dozen transponders, each with 36 MHz bandwidth. Each 36 MHz bandwidth is shared by time division multiplexing. As this preferred band is already saturated, the next highest band available is referred to as 12/14 GHz. It uses 14 to 14.5GHz for upward transmission and 11.7 to 12.2 GHz for downward transmissions. Communication satellites have several unique properties. The most important is the long communication delay for the round trip (about 270 ms) because of the long distance (about 72,000 km) the signal has to travel between two earth stations. This poses a number of problems, which are to be tackled for successful and reliable communication.

Another interesting property of satellite communication is its broadcast capability. All stations under the downward beam can receive the transmission. It may be necessary to send encrypted data to protect against piracy.

8.4 DIGITAL MODULATION TECHNIQUES

In this section, various digital modulation techniques, including those employing constant and non-constant envelope signalling formats are presented. Commonly used modulation schemes, such as Phase Shift Keying (PSK), Amplitude Shift Keying (ASK) and Frequency Shift Keying (FSK) and several coded modulation techniques are studied.

A constellation diagram is a representation of a signal modulated by a digital modulation scheme such as quadrature amplitude modulation or phase-shift keying. It displays the signal as a two-dimensional scatter diagram in the complex plane at symbol sampling instants. In a more abstract sense, it represents the possible symbols that may be selected by a given modulation scheme as points in the complex plane. Measured constellation diagrams can be used to recognize the type of interference and distortion in a signal.

By representing a transmitted symbol as a complex number and modulating a cosine and sine carrier signal with the real and imaginary parts (respectively), the symbol can be sent with two carriers on the same frequency. They are often referred to as quadrature carriers. A coherent detector is able to independently demodulate these carriers. This principle of using two independently modulated carriers is the foundation of quadrature modulation. In pure phase modulation, the phase of the modulating symbol is the phase of the carrier itself.

As the symbols are represented as complex numbers, they can be visualized as points on the complex plane. The real and imaginary axes are often called the in phase, or I-axis and the quadrature, or Q-axis. Plotting several symbols in a scatter diagram produces the constellation diagram. The points on a constellation diagram are called constellation points. They are a set of modulation symbols which comprise the modulation alphabet.

Digital modulation is a process of transforming a digital symbol to a signal suitable for transmission via a communication channel. According to this process the information is carried by a sinusoidal signal:

$$s(t) = -a(t)\sin[2\pi f(t)t + \varphi(t)]$$

where: amplitude $a(t)$, frequency $f(t)$ and phase $\varphi(t)$. Thus, by encoding the information in one of the parameters, keeping the other two unchanged, three basic modulation schemes exist:

- **Amplitude Shift Keying (ASK):** The digital symbol is encoded in the amplitude of the signal. A simple example of ASK is the on/off modulation with digital 1,0 represented by signal with amplitude A and no signal, respectively.
- **Frequency Shift Keying (FSK):** The signal with carrier frequency f_1 represents digital 1 and the signal with carrier frequency f_2 denotes digital 0. A simple FSK modulator consists of a switch and two oscillators with frequencies f_1 and f_2.
- **Phase Shift Keying (PSK):** The information is encoded in the phase changes of the transmitted signal.

Advanced modulation schemes can be derived by combining basic modulation schemes to comply with transmission requirements, for example higher bandwidth efficiency. The waveforms of the three above basic schemes are shown in Fig. 8.9. Two basic criteria exist to compare the performance of the modulation schemes, namely, power efficiency and bandwidth efficiency.

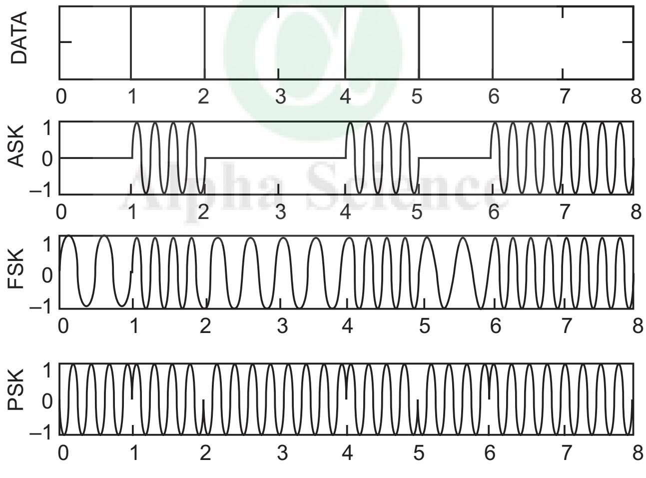


Fig. 8.9 Waveforms for amplitude, frequency and phase modulations

Power efficiency: Power efficiency is defined as the ratio of the required bit energy to noise Power Spectral Density (PSD) determined for a certain Bit Error Probability (BEP) over a Gaussian channel. Normally, a Bit Error Rate (BER) equal to 10^{-5} is used to determine the power efficiency. Moreover, the symbol energy and symbol error rate are applied to define power efficiency.

Bandwidth efficiency: Bandwidth efficiency is defined as the number of bits per second that can be transmitted within one Hertz of the system bandwidth. The calculated values of bandwidth efficiency depend on the definition of the system bandwidth.

8.4.1 Phase Shift Keying

PSK is a modulation scheme whereby the phase of the transmitted signal is modulated by the data stream. A few of the PSK variants are discussed below.

Binary phase shift keying: In BPSK the phase of the carrier is shifted between two positions that are 180 degrees apart. On an I/Q diagram, the I channel has two different values as it is depicted in Fig. 8.10. There are two possible locations in the state diagram, so a binary one or zero can be sent.

The symbol rate is one bit per symbol, Es — Eb, where Es is the average signal energy and Eb is the average bit energy. Special attention should be paid to the subtle difference between the ASK and BPSK modulations. In BPSK the baseband signal takes the values of $+A$ and $-A$ instead of $+A$ and zero, as in ASK. Thereby, the phase of the output signal, for one and zero, is reversed rather than turned on or off.

The envelope of a BPSK signal is:

$$s(t) = \sum_{n=-\infty}^{n=\infty} g_n p(t - nT_s)$$

where $p(t)$ is a unit amplitude shaping pulse of duration Ts and g_n is the BPSK symbol in the nth symbol interval $nTs \leq t < (n + 1)\ Ts$ given by:

$$g_n = Aexp\{j\varphi_n\}$$

In the above equation, A is the constant BPSK symbol envelope and φ^n is the phase of the nth symbol.

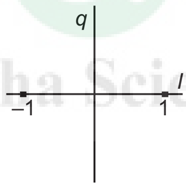


Fig. 8.10 BPSK constellation diagram

8.4.1.1 *Quadrature phase shift keying*

In QPSK the phase of the carrier is shifted between four positions that are 90 degrees apart. On an I/Q diagram, the I channel has four different values, as it is depicted in Fig. 8.11.

There are four possible locations in the state diagram, so all possible binary combinations of one and zero can be sent. The symbol rate is two bit per symbol, $Es = 2Eb$.

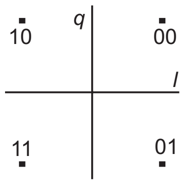


Fig. 8.11 QPSK constellation diagram

8.4.1.2 *Offset QPSK*

An equivalent modulation signalling, which is widely employed in satellite communications, is OQPSK (or staggered QPSK). OQPSK is a form of QPSK wherein the I and Q channels are misaligned with respect to one another by half a symbol time interval, $Ts/2 = T_b$, This signalling has the advantage over the QPSK that the abrupt phase change of 180^0 is eliminated. Hence, the phase trajectories passing through the origin of the constellation diagram are avoided. This property makes the OQPSK signal less sensitive to nonlinear amplifier impairments as compared to QPSK. The complex OQPSK envelope is:

$$S(t) = S1(t) + jS_Q(t)$$

$$S1(t) = \sum_{n=-\infty}^{n=\infty} gI_n p(t - nT_s)$$

$$S_Q(t) = \sum_{n=-\infty}^{n=\infty} gQ_n p(t - nT_s - T_s/2)$$

where g_{1n} and g_{Qn} are the I and Q data symbols for the n^{th} transmission interval that take on equiprobable ±1 values.

8.4.1.3 *M-ary PSK*

In M-ary PSK (M-PSK) the phase of the carrier is shifted between M positions that are: $\varphi = (2i - 1)(□/M)$, $i = 1, 2, \ldots M$ radians apart. On an I/Q diagram the I channel has M different values. There are M possible locations in the state diagram, so all possible binary combinations of one and zero can be sent. In Fig. 8.12, the constellation diagram of 8-PSK modulation scheme is depicted.

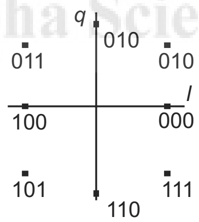


Fig. 8.12 8-PSK constellation diagram

8.4.2 Amplitude Shift Keying (ASK)

This scheme can be accomplished simply by the on-off gating of a continuous carrier. The simplest ASK technique is to represent one binary level (binary 1) by a single signal of fixed amplitude and the other level (binary 0) by switching off the signal. The absence of the signal for one of the binary levels has the disadvantage that if fault conditions exist it could be misinterpreted as received data. Waveform for ASK, using different amplitude signals for the logic levels, are an alternative method to prevent this disadvantage. As with speech telephony circuits, the upper side band and carrier may be suppressed to reduce the bandwidth requirement and concentrate the available power on the signal containing the information.

8.4.3 Frequency Shift Keying (FSK)

This technique may be used whereby the carrier frequency has one value for a 1 bit and another for a 0 bit. The main difficulty in the use of this FM technique is that the gap between the frequencies used must be increased as the modulation rate increases. More exactly, for a restricted channel bandwidth, especially using in-band supervisory signalling, there is a limit to the maximum bit rate that is possible with this technique.

8.5 SWITCHING TECHNIQUES

When there are many devices, it is necessary to develop suitable mechanism for communication between any two devices. One alternative is to establish point-to-point communication between each pair of devices using mesh topology. However, mesh topology is impractical for large number of devices, because the number of links increases exponentially ($n(n-1)/2$), where n is the number of devices) with the number of devices. A better alternative is to use switching techniques leading to switched communication network. In the switched network methodology, the network consists of a set of interconnected nodes, among which information is transmitted from source to destination via different routes, which is controlled by the switching mechanism. A basic model of a switched communication is shown in Fig. 8.13. The end devices that wish to communicate with each other are called stations. The switching devices are called nodes. Some nodes connect to other nodes and some are to connected to some stations. Key features of a switched communication network are given below:

- Network Topology is not regular.
- Uses FDM or TDM for node-to-node communication.
- There exist multiple paths between a source-destination pair for better network reliability.
- The switching nodes are not concerned with the contents of data.
- Their purpose is to provide a switching facility that will move data from node to node until they reach the destination.

The switching performed by different nodes can be categorized into the following three types: Circuit Switching, Packet Switching and Message Switching.

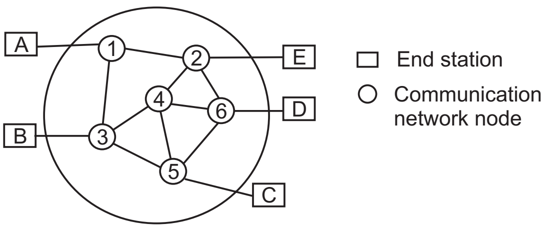


Fig. 8.13 Basic model of a switched communication network

8.5.1 Message Switching

It has been explained earlier that switching plays a very important role in telecommunication networks. It enables any two users to communicate with each other. Voice being a very vital medium of human communication, telephone was invented. It permitted long distance voice communication. The need of a user to talk to a desired person out of many persons on a real time basis lead to the concept of establishing a direct path between the caller and the called users. Circuit switching was conceived to be an appropriate technique for the purpose. Telephone systems use circuit switching largely to date because it serves the purpose very well. However, a major drawback of circuit switching is the requirement of a dedicated path between the calling and the called parties. This means reserving resources like the chain of switches and transmission media over the entire path. This is practically a waste of resources.

The concept of "store-and-forward" of data was emerged under this situation. The data packets have been transmitted to the proceeding node and it will be stored there till further path to the destination is cleared.

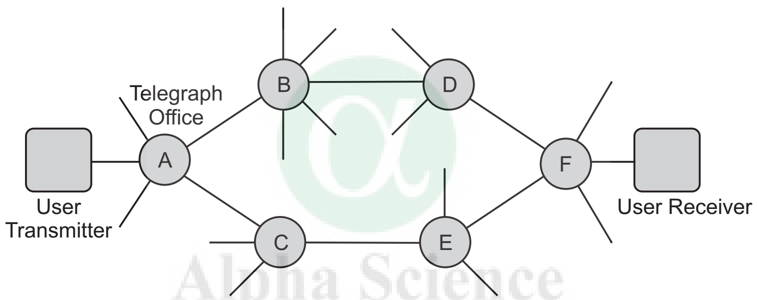


Fig. 8.14 Illustration of message switching

Refer Fig. 8.14. The data packet from user destination is being transmitted to the node-A and it stored in its buffer till the channel either from node-A to node-B or from node-A to node-C is available. Assume that channel to node-B is available and the data packet from node-A has been transmitted to node-B. When the channels from B to D, D to F and F to user receiver are available, the data packet is being transmitted.

8.5.2 Packet Switching

A possible solution for the above problem is to fragment the long messages into small size units, known as packets. It is these packets that are transmitted instead of the single long message. This method is slightly different from Message switching and is called Packet switching. Figure 8.15 shows a message broken down into small sized packets P_1, $P_2 \ldots P_5$.

These packets are now transmitted over the network in the same manner as the messages in message switching. The packets are stored and forwarded at every node. Obviously every packet now has to have the source and destination addresses. Even in message switching repeated transmission of addresses at every node consumes network bandwidth. In packet switching the overhead/ wastage is more because every packet is now required to carry the addresses on their head. So with the user message in a packet the header is to be transmitted also. From this point of view network bandwidth consumed is maximum in packet switching and minimum in circuit switching.

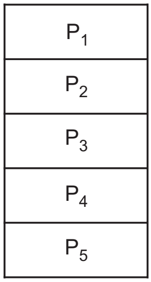


Fig. 8.15 Packets

Packets of the same message are launched into the network in parallel over different available forward links at a node. These packets would travel through different paths to arrive at the destination. This simultaneous transmission of packets over different paths results in further improvement of the link utilization compared to the message switching. Another advantage is that no link is engaged for a long time since the packets are of smaller size than the single message. This permits better sharing of the links amongst multiple users. However the scheme just discussed has two major drawbacks. Firstly, the packets of the same message travelling through different paths may arrive at the destination at different times due to different delays encountered in different paths. Thus the packets may arrive out of order. In order to deliver them to the destination, they need to be ordered which requires extra processing and so more delay. They need to be given sequence numbers for reordering them. The sequence number increases the overhead and requires more network bandwidth. Secondly, some of the paths may not be very good and some packets may get lost. This worsens the quality. To improve the quality, they require retransmission which in turn requires more processing time and more bandwidth. In spite of these drawbacks the packet switching more favoured. In fact for computer communication and network packet communication was the choice. Basic reasons for this choice were: 1. the computer traffic being (at least then) being mostly text is non real time and 2. the computer data traffic is highly bursty in nature. Considering these features it becomes obvious that circuit switching was not the right kind of switching. Message switching can do the job but for better line utilization packet switching is preferable. Thus computer networks use packet switching.

8.5.3 Circuit Switching Technique

Communication via circuit switching implies that there is a dedicated communication path between the two stations. The path is a connected through a sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Circuit switching is commonly used technique in telephony, where the caller sends a special message with the address of the callee (i.e. by dialling a number) to state its destination. It involved the following three distinct steps, as shown in Fig. 8.16.

Circuit Establishment: To establish an end-to-end connection before any transfer of data. Some segments of the circuit may be a dedicated link, while some other segments may be shared.

Data transfer: Transfer data is from the source to the destination. The data may be analog or digital, depending on the nature of the network. The connection is generally full-duplex.

Circuit disconnect:

- Terminate connection at the end of data transfer.
- Signals must be propagated to deallocate the dedicated resources.

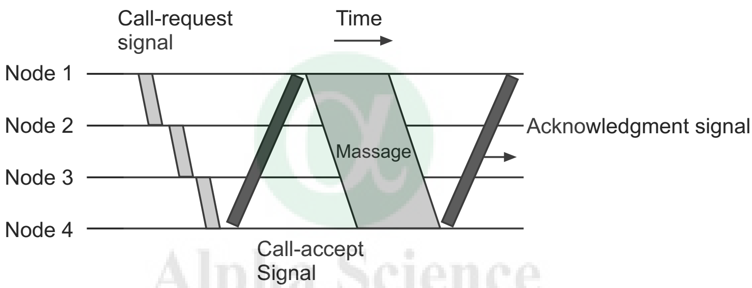


Fig. 8.16 Circuit Switching technique

Thus the actual physical electrical path or circuit between the source and destination host must be established before the message is transmitted. This connection, once established, remains exclusive and continuous for the complete duration of information exchange and the circuit becomes disconnected only when the source wants to do so.

Switching Node: Let us consider the operation of a single circuit switched node comprising a collection of stations attached to a central switching unit, which establishes a dedicated path between any two devices that wish to communicate.

Major elements of a single-node network are summarized below:

- Digital switch: That provides a transparent (full-duplex) signal path between any pair of attached devices.
- Network interface: That represents the functions and hardware needed to connect digital devices to the network (like telephones).
- Control unit: That establishes, maintains, and tears down a connection.

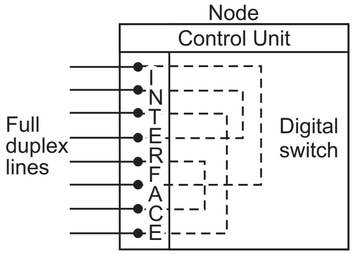


Fig. 8.17 Schematic diagram of a switching node

The simplified schematic diagram of a switching node is shown in Fig. 8.17. An important characteristic of a circuit-switch node is whether it is blocking or non-blocking. A blocking network is one, which may be unable to connect two stations because all possible paths between them are already in use. A non-blocking network permits all stations to be connected (in pairs) at once and grants all possible connection requests as long as the called party is free. For a network that supports only voice traffic, a blocking configuration may be acceptable, since most phone calls are of short duration. For data applications, where a connection may remain active for hours, non-blocking configuration is desirable.

Circuit switching uses any of the three technologies: Space-division switches, Time-division switches or a combination of both. In Space-division switching, the paths in the circuit are separated with each other spatially, i.e. different ongoing connections, at a same instant of time, uses different switching paths, which are separated spatially. This was originally developed for the analog environment, and has been carried over to the digital domain. Some of the space switches are crossbar switches, Multi-stage switches (e.g. Omega Switches). A crossbar switch is shown in Fig. 8.18. Basic building block of the switch is a metallic cross-point or semiconductor gate that can be enabled or disabled by a control unit.

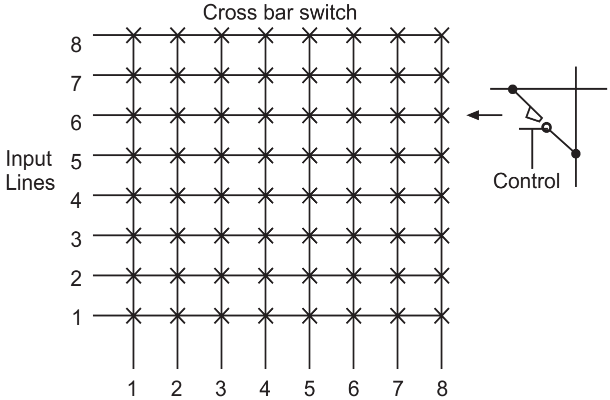


Fig. 8.18 Schematic diagram of a crossbar switch

Limitations of crossbar switches:

- The number of cross-points grows with the square of the number of attached stations.
- Costly for a large switch.
- The failure of a cross-point prevents connection between the two devices whose lines intersect at that cross-point.
- The cross-points are inefficiently utilized.
- Only a small fraction of cross-points are engaged even if all of the attached devices are active.

Some of the above problems can be overcome with the help of multistage space division switches. By splitting the crossbar switch into smaller units and interconnecting them, it is possible to build multistage switches with fewer cross-points.

REVIEW QUESTIONS

1. Explain the concept of digital data transmission.
2. What is meant by quantizing?
3. Distinguish between guided and unguided transmission media.
4. Explain in detail about various types of guided transmission media.
5. What are the characteristics of coaxial cables?
6. Write the advantages and disadvantages of optical fibres.
7. Write short note on satellite communication.
8. Explain in detail about various digital modulation techniques.
9. Write short note on Quadrature phase shift keying.
10. Write short notes on the following:
    - Message switching
    - Packet switching
    - Circuit switching
11. What are the limitations of crossbar switches?

CHAPTER

9

COMPUTER NETWORKS

9.1 INTRODUCTION

A computer network is a group of computers interconnected to interact each other. More specific, a computer network is an infrastructure which provides connectivity to multiple autonomous computer systems in order to communicate (e.g., e-mail) and share resources. These resources include hardware, such as storage media and peripherals (e.g., hard disks, printers, etc.), software (i.e., computer programs), and information/data. Based on the number of computer connected to the network and the geographical area the network is covering, computer networks are broadly classified into three, viz; Local Area Network (LAN), Metropolitan Area network (MAN) and Wide Area Network (WAN). Among these, most common are the local area network (LAN) and the wide area network (WAN).

9.2 LOCAL AREA NETWORKS (LAN)

Local Area Networks (LAN), are privately owned networks within a single building or campus of up to a few kilometres in size. They are widely used to connect personal computers and workstations in company office and factories to share resources (e.g. printers) and exchange information.

Various topologies are possible for broadcast LANs. Fig. 9.1 shows two of them. In a bus (i.e., a linear cable) network, at any instant at most one machine is the master and is allowed to transmit. All other machines are required to refrain from sending. An arbitration mechanism is needed to resolve conflicts when two or more machines want to transmit simultaneously. The arbitration mechanism may be centralized or distributed. Computers on an Ethernet (IEEE 802.3, popularly called Ethernet) can transmit whenever they want to; if two or more packets collide, each computer just waits a random time and tries again later.

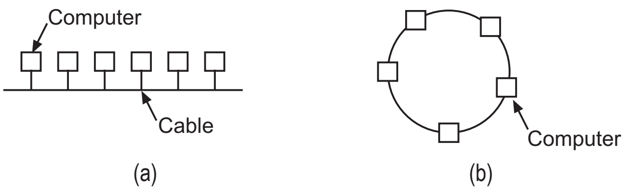


Fig. 9.1 Two broadcast networks. (a) Bus (b) Ring

A second type of broadcast system is a ring. In a ring, each bit propagates around own its own, not waiting for the rest of the packet to which it belongs. Typically, each bit circum navigates the entire ring in the time it take to transmit a few bits, often before he complete packet has ever been transmitted.

Broadcast network can be further divided into static and dynamic, depending on how the channel is allocated. A typical static allocation would be to divide time into discrete intervals and use around round robin algorithm, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to broadcast during its allocated slot, so more systems attempt to allocate the channel dynamically (that is on demand).

Dynamic allocation methods for a common channel are either centralized or decentralized. In the centralized channel allocation method, there is a central entity. It might do this by accepting requests and making a decision according to some internal algorithm. In the decentralized channel allocation method, there is no central entity; each machine must decide for itself whether to transmit.

9.3 METROPOLITAN AREA NETWORKS (MAN)

Metropolitan Area Network covers a city. The best known example for MAN is cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over the air television reception. A MAN might look something like a system shown in Fig. 9.2. In this figure we see both television signals and internet being fed into the centralized head-end for subsequent distribution to subscriber's home.

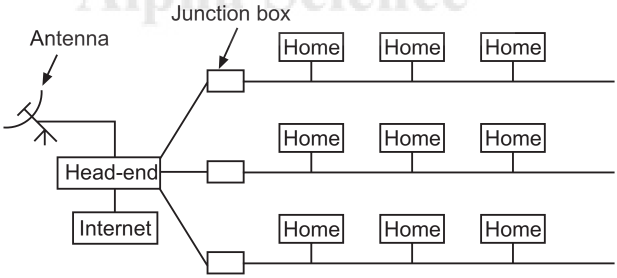


Fig. 9.2 A MAN based on cable TV

9.4 WIDE AREA NETWORKS (WAN)

Wide Area Networks (WAN) spans a large geographical area, often a country or a continent. It contains a collection of machines intended for running user (i.e. application) programs. These machines are called as hosts. The hosts are connected by a communication sub-net. The hosts are owned by the customers (example: personal computers), where as the communication subnet is typically owned and operated by a telephone company or internet service provider. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speakers to listeners.

In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements. Transmission lines move bits between machines. They can be made of copper wires, optical fibers, or even radio links. Switching elements are specialized computers that can connect three or more transmission lines. When data arrives on an incoming line, the switching elements must choose an outgoing line on which to forward them. These switching computers have been called by various names in the past: the name router is now most commonly used.

In this model shown in Fig. 9.3, each host is frequently connected to a LAN on which a router is present, although, in some cases a host can be connected directly to a router. The collection of communication lines and routers (but not the host) form this subnet.

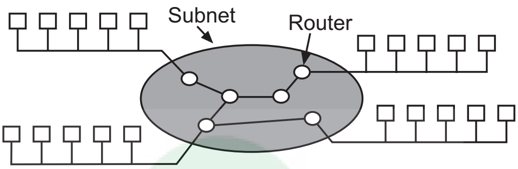


Fig. 9.3 Hosts on LANs and the subnet

In most WANs, a network contains numerous transmission lines, each one connecting a pair of routers. If two routers do not share a transmission line wish to communicate, they must do this indirectly, via other routers. When a data packet is send from one router to another via one or more intermediate routers, the packet is received at each intermediate routers in its entirety, stored there until the required output line is free and then forwarded. A subnet organized according to this principle is called a store and forward or packet switched subnet. Nearly all WANs (except those using satellites) have store and forward subnets. When the packets are small and all the same size they are often called cells.

9.5 NETWORKING COMPONENTS

Networking components are the electronic modules which are use to connect a computer or a computer network to another network or a computer. It is basically hardware. Some of the networking components are discussed in this section.

9.5.1 Network Interface Card (NIC)

A NIC is a hardware board or card that you put into an empty slot in the back of your client computer or server. NIC is the interface between the PC and the physical network connection. This card physically connects to the cable that links your network. As with any other type of adapter card NICs come in ISA, PCMCIA and PCI bus variaties. Fig. 9.4 shows typical Ethernet NIC. Since the NIC contains both BNC and RJ-45 connectors, it is called a combo card.

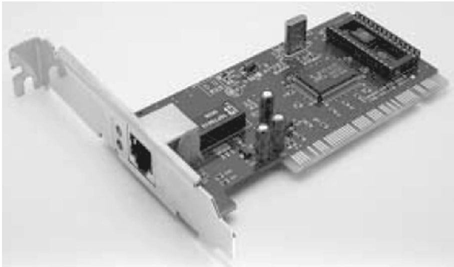


Fig. 9.4 Network interface card

In addition to providing the physical connection to the networks, they also perform the following:

Prepare data: NIC prepare data so that it can transmit through the cable. The card translates data bit back and forth as they go from the computer to the cable and back again.

Address data: Each NIC has its own unique address that it imparts to the data stream. The card provides the data with an identifier. When it goes out on to the net and enables data seeking a particular computer to known where to exit the cable.

Control data flow: The card has RAM on it to help it, place the data so that it doesn't overwhelm the receiving computer on the cable.

Make (and agree on) the connection to another computer: Before it actually sends data, the NIC an electronic dialog with the other PC on the network that wants to communicate. They agree on thing like the maximum size of data groups to be sent. The total maximum size of data (amount), the time interval between data checks the amount of time that will elapse before confirmation that the data has arrived successfully and how much data each card hold before it overflows.

NIC is an especially useful place to implement IP sec. technology. This is the place where end station data is turned into useful security management information where data can be queued in order of priority before transport and where hardware acceleration can be used to the greatest advantage to help facilitate encryption.

An encrypted audio/video stream from a server to its clients provides a good example of the benefits of hardware acceleration. Users would experience much better network performance, if the stream were decrypted on an IP. Instead of via decryption software only hardware acceleration in the NIC can help to improve network performance by accelerating the many math cycle required by encryption and decryption algorithms by offloading the process onto a NIC problems are avoided. Data transfers between the interfaces or nodes takes using these hardware address.

9.5.2 Routers

Routers are internetwork connectivity devices. An internetwork may consists of two or more physical connected independent network. These networks can be of different type.

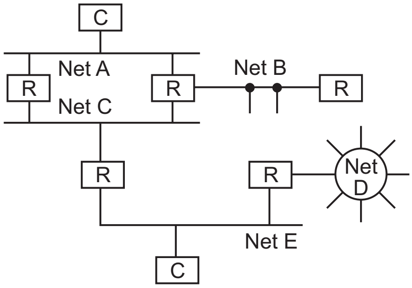


Fig. 9.5 Internet separated by router

For example, they can be ethernet and token ring network. Each network is logically separate and is assigned an address. Routers can use network address to assists efficient delivery of message.

Delivering packets according to logical network address is called routing. Routers performs routing. Routing is the process of finding a path from a source to every destination in the network. Routers are intelligent. They can use algorithms to determine most efficient path for sending a packet to any given network.

Routers can also be used to divide large busy LANs into smaller segments. The protocols like IP, IPX and DDP are used to support routing functions. Router are also employed to connect LAN to wide area network (WAN).

Routers are of two types:

1. **Static routers:** Static routers do not determine paths, but you need to specify them.
2. **Dynamic routers:** Dynamic routers have capacity to determine routes.

9.5.3 Bridges/Switches

Bridges are both hardware and software devices. They can be standalone devices. Most server-software will automatically act as a bridge when a second NIC card is installed. Bridges provides most economical means of interconnecting two or more LANs. Because bridges operate independently of network protocol, they are universal in nature. There are two physical types of bridges.

Local Bridges: They are used where the network is being locally (talking physical location now) segmented. The two segment are physically close together *i.e.*, in same building same floor etc., only one bridge is then required.

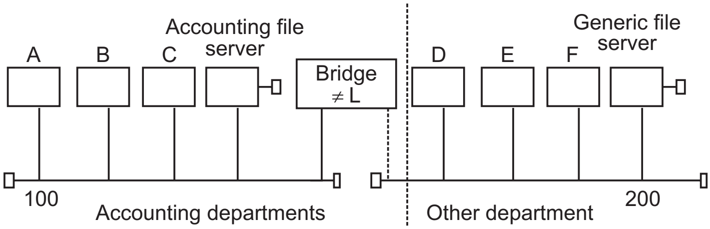


Fig. 9.6 Local bridge requirement

Remote Bridges: They are used in pairs and where the network is remotely segmented. These two segment are physically far apart like for different buildings, different floors, etc. The remote bridges are half of the normal bridge and may use several different communications media in between.

Purposes of bridges:

1. Isolate network by MAC addresses.
2. Manage network traffic by filtering packet.
3. Translate from one protocol to another.

9.5.4 Hubs

Hubs are special repeaters that overcome the electromechanical limitations of a media. Hubs are wiring concentrators. They provide central attachment point for network cabling. Fig. 9.7 shows the diagram of a passive hub.

There are three types of hubs.

Active hubs: Active hubs can amplify and clean up the electronic signals. The process of cleaning up is called signal regeneration. Active hubs consist of electronic components.

Passive hubs: Passive hub is used to combine the signals from several network cable segments. This does not contain any electronic component and do not process the data signal in any way. All devices attached to passive hub receive all the packets that pass through hub.

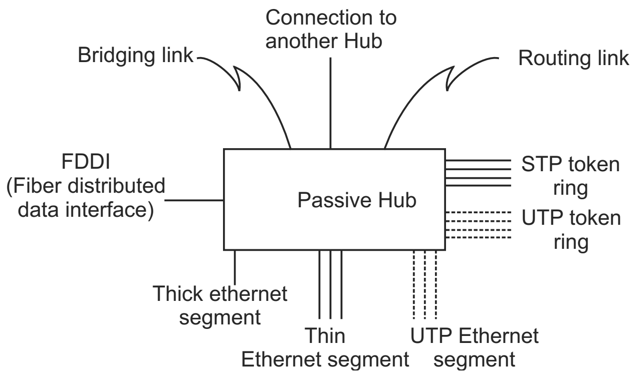


Fig. 9.7 Passive hub

Switching hub: It quickly routes the signal between ports of hub. These can be used in place of routers.

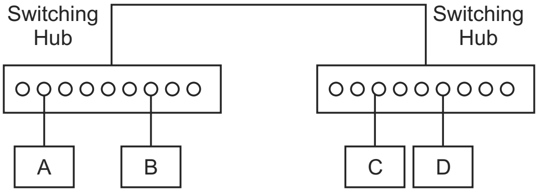


Fig. 9.8 Switching hubs

Consider Fig. 9.8. if A wanted to communicate with B then a dedicated 10 Mbps connection would be established between the two. If C wanted to communicate with D then another dedicated 10 Mbps connection would be established.

9.5.5 Modems

The term 'Modem' is derived from the words, modulator and demodulator. A modem contains a modulator as well as a demodulator. A typical data connect setup using modems is shown in Fig. 9.9.

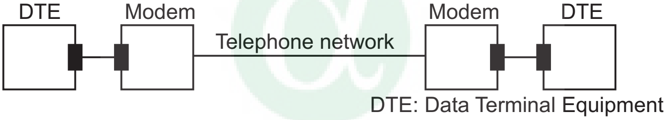


Fig. 9.9 A data circuit implemented using modems

The digital terminal devices which exchange digital signals are called Data Terminal Equipment (DTE). Two modems are always required, one at each end. The modem at the transmitting end converts the digital signal from the DTE into an analog signal by modulating a carrier. The modem at the receiving end demodulates the carrier and handover the demodulated digital signal to the DTE.

The transmission medium between the two modems can be dedicated leased circuit or a switched telephone circuit. In the latter case, modems are connected to the local telephone exchange. Whenever data transmission is required, connection between the modems is established through the telephone exchanges. Modems are also required within a building to connect terminals which are located at distances usually more than 15 metres from the host.

Broadly, a modem comprises a transmitter, a receiver and two interfaces as shown in Fig. 9.10. The digital signal to be transmitted is applied to the transmitter. The modulated carrier which is received from the distant end is applied to the receiver. The digital interface connects the modem to the DTE which generates and receives the digital signals. The line interface connects the modem to the transmission channel for transmitting and receiving the modulated signals. Modems connected to telephone exchanges have additional provision for connecting a telephone instrument. The telephone instrument enables establishment of the telephone connection.

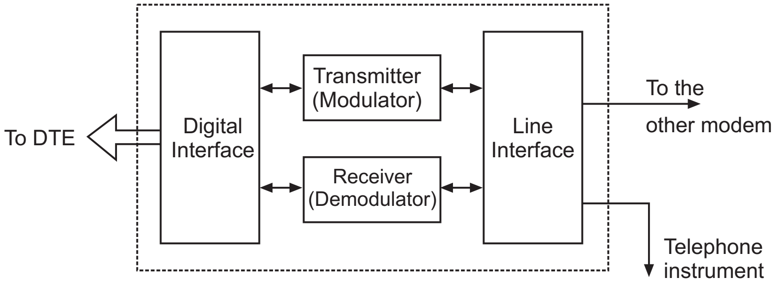


Fig. 9.10 Building blocks of a modem

The transmitter and receive in a modem comprise several signal processing circuits which include a modulator in the transmitter and a demodulator in the receiver.

Types of Modems: Modems are of several types and they can be categorized in a number of ways based on following basic modem features; as shown in Fig. 9.11.

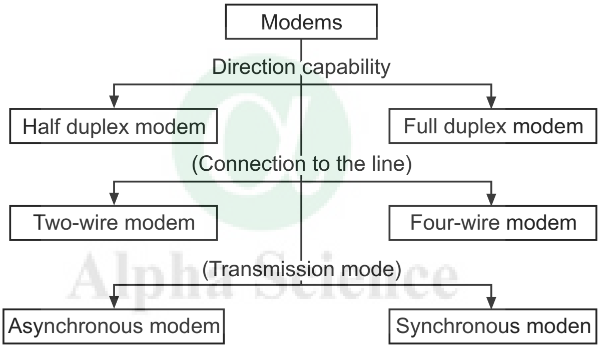


Fig. 9.11 Types of modems

- Directional capability—Half duplex and full duplex modem
- Connection to the line—2-wire modem and 4-wire modem
- Transmission mode—Asynchronous and synchronous modem

Half Duplex Modem: A half duplex modem permits transmission in one direction at a time. If a carrier is detected on the line by the modem, it gives an indication of the incoming carrier to the DTE through a control signal of its digital interface as shown in Fig. 9.12. So long as the carrier is being received, the modem does not give clearance to the DTE to transmit.

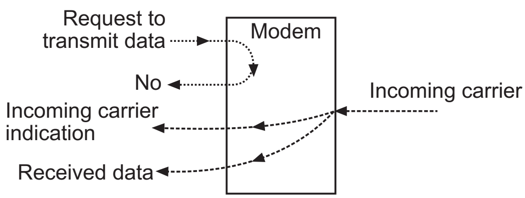


Fig. 9.12 Half duplex modem

Full Duplex Modem: A full duplex modem allows simultaneous transmission in both directions. Thus, there are two carriers on the line, one outgoing and the other incoming as shown in Fig. 9.13.

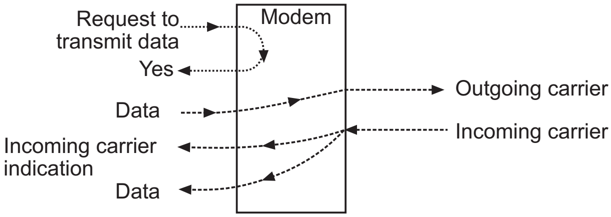


Fig. 9.13 Full duplex modem

2-Wire Modem: In a 2-wire connection only one pair of wires is extended to the subscriber's premises. In this modems with a 2-wire line interface are required. Such modem use the same pair of wires for outgoing and incoming carriers. Half duplex mode of transmission using the same frequency for the incoming and outgoing carriers, as shown in Fig. 9.14. The transmit and receive carrier frequencies can be the same because only one of them is present on the line at a time.

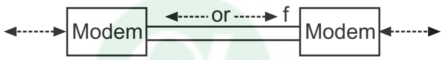


Fig. 9.14 2-Wire half duplex modem

For full duplex mode of operation on a 2-wire connection, it is necessary to have two transmission channels, one for the transmit direction and the other for the receive direction as shown in Fig. 9.15.

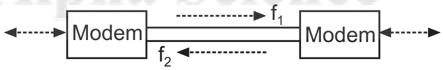


Fig. 9.15 2-Wire full duplex modem

This is achieved by frequency division multiplexing of two different carrier frequencies. These carriers are placed within the bandwidth of the speech channel. A modem transmits data on one carrier and receives data from the other end on the other carrier. A hybrid is provided in the 2-wire modem to couple the line to its modulator and demodulator as shown in Fig. 9.16. There is a special technique which allows simultaneous transmission of incoming and outgoing carriers having the same frequency on the 2-wire transmission medium. Full bandwidth of the speech channel is available to both the carriers simultaneously. This technique is called echo cancellation technique.

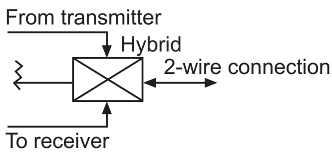


Fig. 9.16 Line Interconnection in a 2-wire full duplex modem

4-Wire Modem: In a 4-wire connection, one pair of wires is used for outgoing carrier and the other is used for the incoming carrier as shown in Fig. 9.17. Full duplex and half-duplex modes of data transmission are possible on a 4-wire connection.

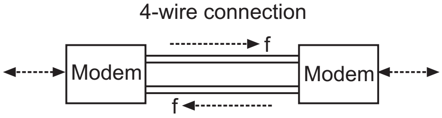


Fig. 9.17 4-Wire modem

Asynchronous Modem: An asynchronous modem can only handle data bytes with start and stop bits. There is no separate timing signal or clock between the modem and the DTE, as shown in Fig. 9.18. The internal timing pulses are synchronized repeatedly to the leading edge of the start pulse.

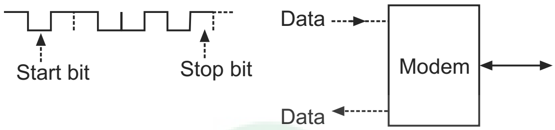


Fig. 9.18 Asynchronous modem

Synchronous Modem: A synchronous modem can handle a continuous stream of data bits but requires a clock signal as shown in Fig. 9.19. The data bits are always synchronized to the clock signal. There are separate clocks for the data bits being transmitted and received. For synchronous transmission of data bits, the DTE can use its internal clock and supply the same to the modem. Else, it can take the clock from the modem and send data bits on each occurrence of the clock pulse. At the receiving end, the modem recovers the clock signal from the received data signal and supplies it to the DTE. High speed modems are provided with scramblers and descramblers for this purpose.

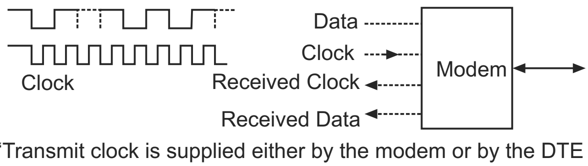


Fig. 9.19 Synchronous modem

Scrambler: A scrambler is incorporated in the modems which operate at data rates of 4800 bps and above. The data stream received from the DTE at the digital interface is applied to the scrambler. The scrambler divides the data stream by the generating polynomial and its output is applied to the encoder.

Descrambler: The decoder output is applied to the descrambler which multiplies the decoder output by the generating polynomial. The unscrambled data is given to the DTE through the digital interface.

9.6 NETWORK PROTOCOLS

A protocol is an agreed–upon format for transmitting data between two devices. The protocol determines the following:

- the type of error checking to be used.
- data compression method; if any
- how the sending device will indicate that it has finished sending a message.
- how the receiving device will indicate that it has received a message.

There are a variety of standard protocols from which programmers can choose. Each has its own particular advantages and disadvantages; for example, some are simpler than the others whle some are more reliable and faster. From a user's point of view, the only interesting aspect about protocols is that your computer or device must support the right ones if you want to communicate with others. The protocol can be implemented either in hardware or in software. Some of the popular protocols are TCP /IP, HTTP, FTP, SMTP, POP, Token–Ring, MNP etc.

9.7 INTRANET AND EXTRANET

Intranet is the generic term for a collection of private computer networks within an organization. An intranet uses network technologies as a tool to facilitate communication between people or work groups to improve the data sharing capability and overall knowledge base of an organization's employees.

Intranets utilize standard network hardware and software technologies like Ethernet, WiFi, TCP/IP, Web browsers and Web servers. An organization's intranet typically includes Internet access but is firewalled, so that its computers cannot be reached directly from the outside.

Many schools and non-profit groups have deployed them, but an intranet is still seen primarily as a corporate productivity tool. A simple intranet consists of an internal email system and perhaps a message board service. More sophisticated intranets include Web sites and databases containing company news, forms, and personnel information. Besides email and groupware applications, an intranet generally incorporates internal Websites, documents, and/or databases.

The business value of intranet solutions is generally accepted in larger corporations, but their worth has proven very difficult to quantify in terms of time saved or return on investment. Intranets are also known as corporate portal or private business network.

An extranet is a computer network that allows controlled access from the outside for specific business or educational purposes. Extranets are extensions to, or segments of, private intranet networks that have been built in many corporations for information sharing and ecommerce.

Most extranets use the Internet as the entry point for outsiders, a firewall configuration to limit access, and a secure protocol for authenticating users.

REVIEW QUESTIONS

1. How computer networks are classified based on number of computers connected to the network?
2. Explain in detail about Local Area Network (LAN).
3. Write short note on Metropolitan Area Network (MAN).
4. What is meant by WAN and distinguish WAN from MAN ?
5. List any 4 networking components.
6. What is the working principle of Network Interfacing Card (NIC) ?
7. Explain the working of Routers.
8. Compare Hubs, Routers and Switches.
9. Explain in detail about the working of MODEM.
10. Classify MODEMs.
11. What is meant by network protocol?
12. Distinguish intranet and extranet?

CHAPTER

10

THE INTERNET

10.1 INTRODUCTION

The Internet is a worldwide system, or network, of computers. It got started in the late 1960s, originally conceived as a network that could survive nuclear war. Earlier it was called ARPAnet, named after the Advanced Research Project Agency (ARPA) of the United States federal government.

When people began to connect their computers into ARPAnet, the need became clear for a universal set of standards, called a protocol, to ensure that all the machines "speak the same language." The modern Internet is such that you can use any type of computer—IBM-compatible, Mac, or other—and take advantage of all the network's resources.

All Internet activity consists of computers "talking" to one another. This occurs in machine language. However, the situation is vastly more complicated than when data goes from one place to another within a single computer. In the Internet (often called simply the Net), data must often go through several different computers to get from the transmitting or source computer to the receiving or destination computer. These intermediate computers are called nodes, servers, hosts, or Internet Service Providers (ISPs). Millions of people are simultaneously using the Net; the most efficient route between a given source and destination can change from moment to moment. The Net is set-up in such a way that signals always try to follow the most efficient route.

If you are connected to a distant computer, the requests you make of it and the data it sends to you are broken into small units called packets. Each packet coming to you has, in effect, your computer's name written on it. But not all packets necessarily travel the same route through the network. Ultimately, all the packets are reassembled into the data you want, say, the infrared satellite image of a hurricane; even though they might not arrive in the same order they were sent.

Architecture of the Internet: Figure 10.1 shows the basic architecture of the Internet. The basic function of the internet system is to communicate (including resource sharing) between a client computer and the internet cloud. Internet cloud comprises large file servers and web servers. The contents in the internet cloud is accessed by the client server using an intermediate agent called Internet Service Provider (ISP). Usually the client has to pay some service charge to the ISP to use their hardware to connect to the internet.

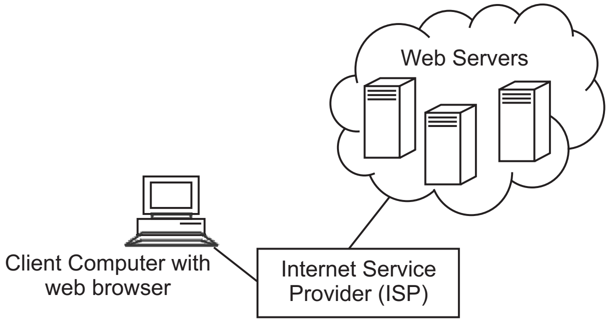


Fig. 10.1 Basic Internet Architecture

Internet conversations: You can carry on a teletype-style conversation with other computer users via the Internet, but takes a bit of getting used to. When done among users within a single service provider, this is called chat. When done among people connected to different service providers, it is called Internet relay chat (IRC). Typing messages to and reading them from other people in real time is more personal than letter writing, because your addressees get their messages immediately. But it's less personal than talking on the telephone, especially at first, because you cannot hear, or make, vocal inflections.

It is possible to digitize voice signals and transfer them via the Internet. This has given rise to hardware and software schemes that claim to provide virtually toll-free long-distance telephone communications. As of this writing, this is similar to amateur radio in terms of reliability and quality of connection. When Net traffic is light, such connections can be good. But when Net traffic is heavy, the quality is marginal to poor. Audio signals, like any other form of Internet data, are broken into packets. All, or nearly all, the packets must be received and reassembled before a good signal can be heard. This takes variable time, depending on the route each packet takes through the Net. If many of the packets arrive disproportionately late, and the destination computer can only "do its best" to reassemble the signal. In the worst case, the signal might not get through at all.

Getting information: One of the most important features of Internet is the fact that it can get you in touch with thousands of sources of information. Data is transferred among computers by means of a file transfer protocol (FTP) that allows the files on the hard drives of distant computers to become available exactly as if the data were stored on your own computer's hard drive, except the access time is slower. You can also store files on distant computers' hard drives. Fig. 10.2 shows how the data have been flowing from a source station to a destination station.

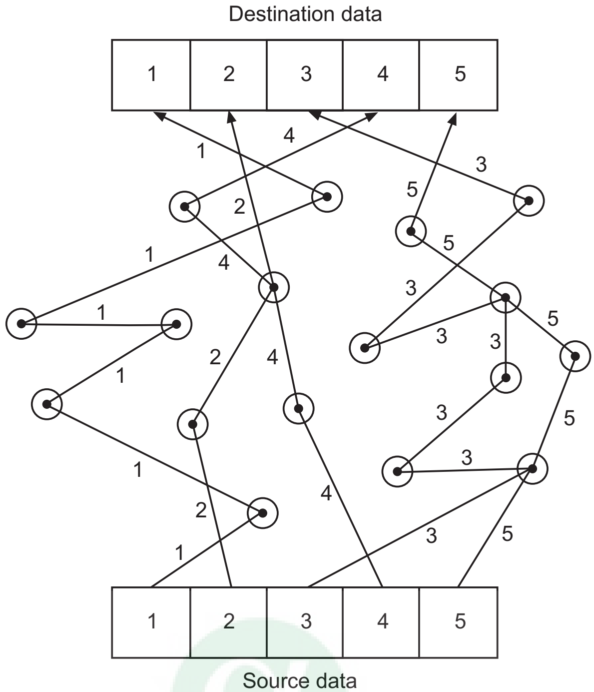


Fig. 10.2 Internet data flows in packets from the source to the destination

The World Wide Web (also called WWW or the Web) is one of the most powerful information servers you will find on-line. Its outstanding feature is hypertext, a scheme of cross-referencing. Certain words, phrases, and images make up so-called links. When you select a link in a Web page or Website (a document containing text and graphics and sometimes also other types of files), your computer is transferred to another document dealing with the same or a related subject. This site will probably also contain numerous links. Before long, you might find yourself "surfing" the Web for hours going from site to site. The word surfing derives from the similarity of this activity to television "channel surfing."

10.2 THE INTERNET PROTOCOL

The Internet operates in a packet-switched mode and Fig. 10.3 shows the protocol stack associated with it. In the figure, we assume the network interface card in all hosts that are attached to an access network communicate with other hosts using the TCP/IP protocol stack. In practice, this is not always the case. Nevertheless, any end system (host) that communicates directly over the Internet does so using the TCP/IP protocol stack.

In general, the various access networks have different operational parameters associated with them in terms of their bit rate, frame format, maximum frame size, and type of addresses that are used. In the TCP/IP protocol stack the network layer protocol is the Internet protocol (IP) and, as we show in Fig. 10.3, in order to transfer packets of data from one host to another, it is the IP in the two hosts, together with the IP in each access gateway and router involved, that perform the routing and other harmonization functions that are necessary.

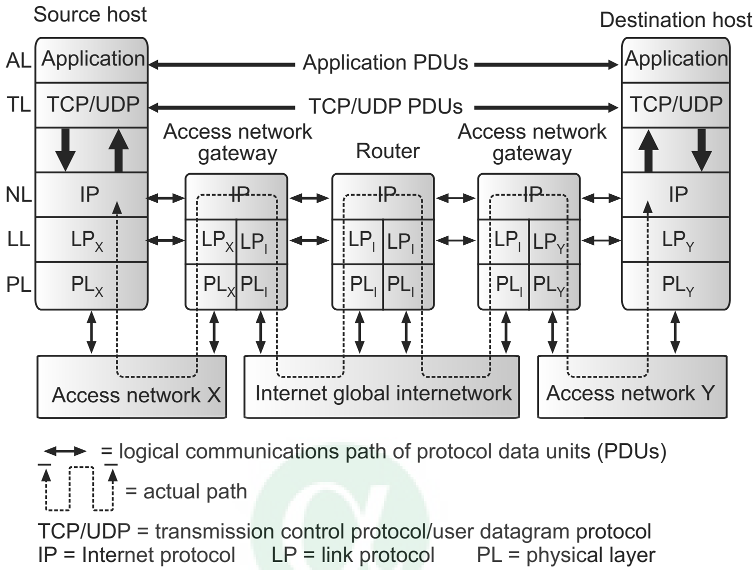


Fig. 10.3 Internet networking components and protocols

The IP in each host (that communicates directly over the Internet) has a unique Internet-wide address assigned to it. This is known as the host's Internet address or, more usually, its IP address. Each IP address has two parts: a network number/identifier (netid) and a host number/identifier (hostid). The allocation of netids is centrally managed by the Internet Corporation for Assigned Names and Numbers (ICANN) and each access network has a unique netid assigned to it. For example, each campus/site LAN is assigned a single netid. The IP address of a host attached to an access network then contains the unique netid of the access network and a unique hostid. As with netids, hostids are centrally allocated but this time by the local administrator of the access network to which the host is attached.

The IP provides a connectionless best-effort service to the transport layer above it which, as we show in the figure, is either the transmission control protocol (TCP) or the user datagram protocol (UDP). Hence when either protocol has a TCP/UDP PDU to transfer, it simply passes the PDU to its local IP together with the IP address of the intended recipient. The (source) IP first adds the destination and source IP addresses to the head of the PDU, together with an indication of the source protocol (TCP or UDP), to form what is known as an IP datagram. The IP then forwards the datagram to its local gateway. At this point the datagram is often referred to as a packet and hence the two terms are used interchangeably.

Each access gateway is attached to an internetwork router and, at regular intervals, the IP in these routers exchange routing information. When this is complete, each router has built up a routing table which enables it to route a packet/datagram to any of the other networks/netids that make up the Internet. Hence, on receipt of a packet, the router simply reads the destination netid from the packet header and uses the contents of its routing table to forward the packet on the path/route through the global internetwork first to the destination internetwork router and, from there, to the destination access gateway. Assuming the size of the packet is equal to or less than the maximum frame size of the destination access network, on receipt of the packet, the destination gateway reads the hostid part of the destination IP address and forwards the packet to the local host identified by the hostid part. The IP in the host then strips off the header from the packet and passes the block of information contained within it – known as the payload – to the peer transport layer protocol indicated in the packet header.

If the size of the packet is greater than the maximum frame size – that is, the maximum transmission unit (MTU) – of the destination access network, the IP in the destination gateway proceeds to divide the block of data contained in the packet into a number of smaller blocks each known as a fragment. Each fragment is then forwarded to the IP in the destination host in a separate packet the length of which is determined by the MTU of the access network. The destination IP then reassembles the fragments of data from each received packet to form the original submitted block of data and passes this to the peer transport layer protocol indicated in the packet header.

10.3 IP DATAGRAMS

The IP is a connectionless protocol and all user data is transferred in the payload part of what is known as a datagram or packet. The header of each datagram contains a number of fields, the formats of which are shown in Fig. 10.4. The version field contains the version of the IP used to create the datagram and ensures that all systems – gateways, routers, and hosts – that process the datagram/packet during its transfer across the Internet to the destination host interpret the various fields correctly. The current version number is 4 and hence the IP is referred to as IP version 4 or simply IPv4.

The header can be of variable length and the intermediate header length (IHL) field specifies the actual length of the header in multiples of 32-bit words. The minimum length (without options) is 5. If the datagram contains options, these are in multiples of 32-bits with any unused bytes filled with padding bytes. Also, since the IHL field is 4 bits, the maximum permissible length is 15.

The type of service (TOS) field allows an application protocol/process to specify the relative priority (precedence) of the application data and the preferred attributes associated with the path to be followed. It is used by each gateway and router during the transmission and routing of the packet to transmit packets of higher priority first and to select a line/route that has the specified attributes should a choice be available. The total length field defines the total length of the initial datagram including the header and payload parts.

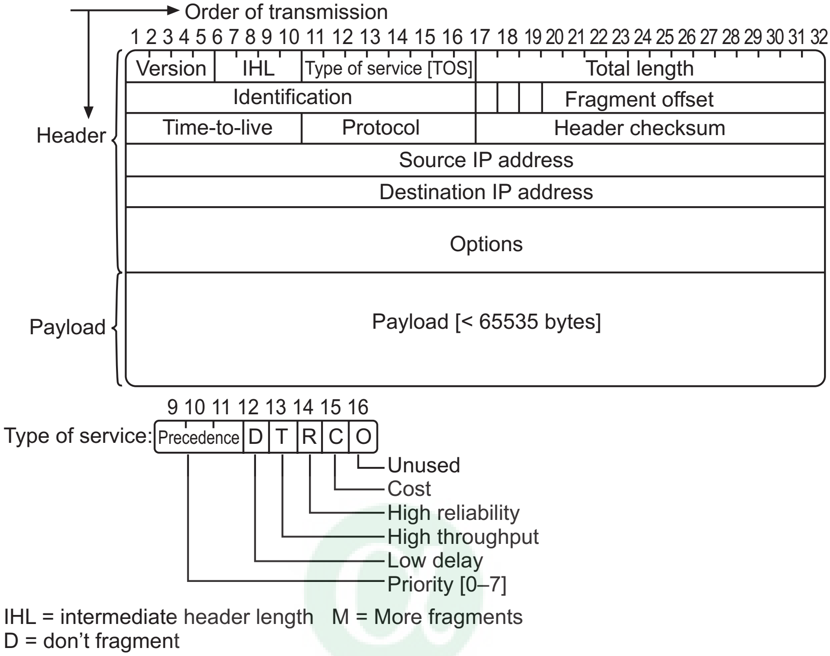


Fig. 10.4 IP datagram/packet format and header fields

Each smaller packet contains the same value in the identification field to enable the destination host to relate each received packet fragment to the same original datagram. The next three bits are known as flag bits of which two are currently used. The first is known as the DO NOT FRAGMENT or D-bit. It is set by a source host and is examined by routers. A set D-bit indicates that the packet should be transferred in its entirety or not at all. The second is known as more fragments or M-bit and this also is used during the reassembly procedure associated with data transfers involving multiple smaller packets/fragments. It is set to 1 in all but the last packet/fragment in which it is set to 0. In addition, the fragment offset is used by the same procedure to indicate the position of the first byte of the fragment contained within a smaller packet in relation to the original packet payload. All fragments except the last one are in multiples of 8 bytes.

The value in the time-to-live field defines the maximum time for which a packet can be in transit across the Internet. The value is in seconds and is set by the IP in the source host. It is then decremented by each gateway and router by a defined amount and, should the value become zero, the packet is discarded. In principle, this procedure allows a destination IP to wait a known maximum time for an outstanding packet fragment during the reassembly procedure. In practice, it is used primarily by routers to detect packets that are caught in loops.

For this reason, therefore, the value is normally a hop count. In this case, the hop count value is decremented by one by each gateway/router visited and, should the value become zero, the packet is discarded.

The value in the protocol field is used to enable the destination IP to pass the payload within each received packet to the same (peer) protocol that sent the data. The header checksum applies just to the header part of the datagram and is a safeguard against corrupted packets being routed to incorrect destinations. It is computed by treating each block of 16 bits as an integer and adding them all together using 1s complement arithmetic. The source address and destination address are the Internet-wide IP addresses of the source and destination host respectively. Finally, the options field is used in selected datagrams to carry additional information relating to:

- **Security:** the payload may be encrypted, for example, or be made accessible only to a specified user group. The security field then contains fields to enable the destination to decrypt the payload and authenticate the sender;
- **Source routing:** if known, the actual path/route to be followed through the Internet may be specified in this field as a list of gateway/router addresses;
- **Loose source routing:** this can be used to specify preferred routers in a path;
- **Route recording:** this field is used by each gateway/router visited during the passage of a packet through the Internet to record its address. The resulting list of addresses can then be used, for example, in the source routing field of subsequent packets;
- **Stream identification:** this, together with the source and destination addresses in the datagram header, enables each gateway/router along the path followed by the packet to identify the stream/flow to which the packet belongs and, if necessary, give the packet precedence over other packets. Examples include streams containing samples of speech or compressed video;
- **Time-stamp:** if present, this is used by each gateway/router along the path followed by the packet to record the time it processed the packet.

10.4 IP ADDRESSES

Each host, gateway and router has a unique Internet-wide IP address assigned to it that comprises a netid and hostid part. In the case of a host/computer, the netid identifies the network to which it is attached and the hostid then identifies the host on this network. In the case of an access gateway and router, however, each network interface of the gateway/router has a different netid assigned to it.

Over the lifetime of the Internet five different schemes have been used for assigning IP addresses. In general, as the number of Internet users has expanded, the aim of each scheme has been to utilize the 32-bit address space in a more efficient way. A brief summary of each scheme is helpful in order to follow their development.

- **Class-based addresses:** this was the first scheme that was used to assign addresses in the early Internet. It involved dividing the overall address space into five address classes: A, B, C, D and E. Each of the first three address classes – A, B and C – then has a defined boundary between the netid and hostid part of the address space. As we shall see, Class D is used for multicasting and class E is reserved.
- **Subnetting:** this was the first approach to utilizing the address space in a more efficient way. In the case of a large campus or company, for example, in the past it was not uncommon to have a number of different types of LAN at the same site, each of which had a different frame format and, more importantly, maximum frame length.
- **Classless addresses:** this is a more recent development and involves exploiting the full address space in a more efficient way than the use of class-based addresses. With this type of addressing, the network part of an IP address can be any number bits rather than being constrained to the fixed class boundaries and, as a result, this leads to a more efficient use of the total address space.
- **Network address translation:** this is the most recent development. The aim of the NAT scheme is for each access network to be allocated just a single IP address and this is then used by all the hosts when communicating outside of their local access network. For communications within the access network, however, every host is assigned its own (private) address.
- **IPv6:** this is a completely new version of IP – the current version is IPv4 –and was developed to overcome the limited address space – and other limitations – of IPv4. Hence it is the ideal solution. In practice, however, at this point in time, IPv4 is still the dominant protocol in the current Internet and the use of IPv6 is being introduced in an incremental way.

10.5 URL AND HTTP

URL stands for Universal Resource Locator. It is the address of a web page. Each page has its own unique web address (URL). This is how your computer locates the web page that you are trying to find.

Uniform Resource Locator (URL) formerly known as Universal Resource Locator, is a technical, Web-related term used in two distinct meanings: In popular usage and many technical documents, it is a synonym for Uniform Resource Identifier (URI); Strictly, the idea of a uniform syntax for global identifiers of network-retrievable documents was the core idea of the World Wide Web. In the early times, these identifiers were variously called "document names", "Web addresses" and "Uniform Resource Locators". These names were misleading, however, because not all identifiers were locators, and even for those that were, this was not their defining characteristic. Nevertheless, by the time the RFC 1630 formally defined the term "URI" as a generic term best suited to the concept, the term "URL" had gained widespread popularity, which has continued to this day. Every URI (and therefore, every URL) begins with the scheme name that defines its namespace, purpose, and the syntax of the remaining part of the URI. Most

Web-enabled programs will try to dereference a URI according to the semantics of its scheme and a context-vbn. For example, a Web browser will usually dereference a http://example.org/ by performing an HTTP request to the host example.org, at the default HTTP port. Dereferencing the URI mailto:bob@ example.com will usually open a "Compose e-mail" window with the address bob@example.com in the "To" field.

In its current strict technical meaning, a URL is a URI that, in addition to identifying a resource [provides] a means of locating the resource by describing its primary access mechanism (e.g., its network location).

Hypertext Transfer Protocol (HTTP) is a communications protocol used to transfer or convey information on intranets and the World Wide Web. Its original purpose was to provide a way to publish and retrieve hypertext pages. Development of HTTP was coordinated by the W3C (World Wide Web Consortium) and the IETF (Internet Engineering Task Force). HTTP is a request/response protocol between a client and a server. The client making an HTTP request - such as a web browser, spider, or other end-user tool - is referred to as the user agent. The responding server - which stores or creates resources such as HTML files and images - is called the origin server. In between the user agent and origin server may be several intermediaries, such as proxies, gateways, and tunnels. HTTP is not constrained to using TCP/IP and its supporting layers, although this is its most popular application on the Internet. Indeed HTTP can be "implemented on top of any other protocol on the Internet, or on other networks. HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used."

Typically, an HTTP client initiates a request by establishing a Transmission Control Protocol (TCP) connection to a particular port on a host. An HTTP server listening on that port waits for the client to send a request message.

Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own, the body of which is perhaps the requested file, an error message, or some other information.

Resources to be accessed by HTTP are identified using Uniform Resource Identifiers (URIs) (or, more specifically, Uniform Resource Locators (URLs)) using the http: or https URI schemes.

10.6 DOMAIN NAME SYSTEM (DNS)

The DNS is a very large distributed database that translates host names (like gurus.com) into IP addresses (like 23.45.67.89) and other information. DNS names are organized into a tree, as in Fig. 10.5.

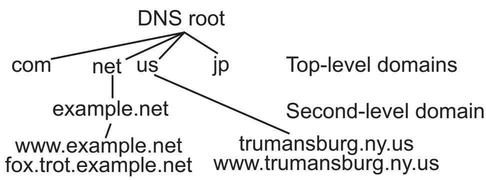


Fig. 10.5 The Domain Name System

All names start at the root, above the set of top-level domains. The root is considered to be at the right end of the name. In a domain name like www.example.net, net is a first-level name within the root, example is a second-level name within net, and www is a third-level name within example. The example.net domain contains another third-level domain called fox.trot. The tree can extend to any number of levels, but in practice is rarely more than four or five levels deep.

The DNS root has a small set of Top-Level Domains (TLDs) that rarely changes. They are listed in Chapter 1, and currently include aero, arpa, biz, com, coop, edu, gov, info, int, mil, museum, name, net, .in and org, along with over 100 two-letter country-name TLDs.

Technically, all top-level domains work the same, although the rules and prices for registering domain names in them vary greatly. Domain data is stored in a set of records. There are many types of records, each storing a specific type of information about the domain. For example, the record that says that a name like www.example.net has an IP address like 10.23.45.67 is called an A (Address) record.

Storing all of the DNS data in one place would not be a good idea since the risk of computer crash, or lose connection with the rest of the Internet cannot be written-off. So, the DNS is divided up into subtrees called zones. Each zone contains the records for a bunch of domains. Large domains may be stored as many zones, so that the network administrators in different places can update the records for the computers in their domains. Smaller domains may be stored in one zone. Records are divided into zones purely for administrative convenience, and the most Internet users never notice which domain information is stored in which zone.

Different zones are stored on various computers called name servers. Most zones have at least two redundant name servers; a few (such as the extremely busy .com zone) have a dozen. A single name server can handle DNS for many zones. The root zone resides on a set of agreed-upon servers dispersed around the world. The root zone contains NS (Name Server) records for each TLD.

10.7 INTERNET SECURITY

Internet security is a branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing.

10.7.1 Types of Security

Network layer security: TCP/IP can be made secure with the help of cryptographic methods and protocols that have been developed for securing communications on the Internet. These protocols include SSL and TLS for web traffic, PGP for email, and IPsec for the network layer security.

IPsec Protocol: This protocol is designed to protect communication in a secure manner using TCP/IP. It is a set of security extensions developed by IETF, and it provides security and authentication at the IP layer by using cryptography. To protect the content, the data is transformed using encryption techniques. There are two main types of transformation that form the basis of IPsec: the Authentication Header (AH) and Encapsulating Security Payload (ESP). These two protocols provide data integrity, data origin authentication, and anti-replay service. These protocols can be used alone or in combination to provide the desired set of security services for the Internet Protocol (IP) layer.

The basic components of the IPsec security architecture are described in terms of the following functionalities:

- Security protocols for AH and ESP
- Security association for policy management and traffic processing
- Manual and automatic key management for the internet key exchange (IKE)
- Algorithms for authentication and encryption

The set of security services provided at the IP layer includes access control, data origin integrity, protection against replays, and confidentiality. The algorithm allows these sets to work independently without affecting other parts of the implementation. The IPsec implementation is operated in a host or security gateway environment giving protection to IP traffic.

Security token: Some online sites offer customers the ability to use a six-digit code which randomly changes every 30-60 seconds on a security token. The key on the security token have mathematical computations built-in and manipulate numbers based on the current time built into the device. This means that every thirty seconds there's only a certain possible array of numbers which would be correct to validate access to the online account. The website that the user is logging into would be made aware of that devices' serial number and therefore would know the computation and correct time built into the device to verify that the number given is in deed one of the handful of six-digit numbers that would work in that given 30-60 second cycle. After the 30-60 seconds the device will present a new random six-digit number which can log into the website.

Electronic mail security (E-mail): Email messages are composed, delivered, and stored in a multiple step process, which starts with the message's composition. When the user finishes composing the message and sends it, the message is transformed into a standard format: an RFC 2822 formatted message. Afterwards, the message can be transmitted. Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transfer agent (MTA) operating on the mail server. The mail client then provides the sender's identity to the server. Next, using the mail server commands, the client sends the recipient list to the mail server. The client then supplies the message. Once the mail server receives and processes the message, several events occur: recipient server identification, connection establishment, and message transmission. Using Domain Name System (DNS) services, the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the recipient mail server(s) and sends the message employing a process similar to that used by the originating client, delivering the message to the recipient(s).

Pretty Good Privacy (PGP): PGP provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption algorithm such 3DES or CAST-128. Email messages can be protected by using cryptography in various ways, such as the following:

- Signing an email message to ensure its integrity and confirm the identity of its sender.
- Encrypting the body of an email message to ensure its confidentiality.
- Encrypting the communications between mail servers to protect the confidentiality of both the message body and message header.

The first two methods, message signing and message body encryption, are often used together; however, encrypting the transmissions between mail servers is typically used only when two organizations want to protect emails regularly sent between each other. For example, the organizations could establish a virtual private network (VPN) to encrypt the communications between their mail servers over the Internet. Unlike methods that can only encrypt a message body, a VPN can encrypt entire messages, including email header information such as senders, recipients, and subjects. In some cases, organizations may need to protect header information. However, a VPN solution alone cannot provide a message signing mechanism, nor can it provide protection for email messages along the entire route from sender to recipient.

Multipurpose Internet Mail Extensions (MIME): MIME transforms non-ASCII data at the sender's site to Network Virtual Terminal (NVT) ASCII data and delivers it to client's Simple Mail Transfer Protocol (SMTP) to be sent through the Internet. The server SMTP at the receiver's side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

Secure/Multipurpose Internet Mail Extensions (S/MIME): S/MIME provides a consistent means to securely send and receive MIME data. S/MIME is not only limited to email but can be used with any transport mechanism that carries MIME data, such Hypertext Transfer Protocol (HTTP).

Message Authentication Code: A Message Authentication Code is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.

10.7.2 Firewalls

A firewall controls access between networks. It generally consists of gateways and filters which vary from one firewall to another. Firewalls also screen network traffic and are able to block traffic that is dangerous. Firewalls act as the intermediate server between SMTP and HTTP connections.

Firewalls impose restrictions on incoming and outgoing packets to and from private networks. All the traffic, whether incoming or outgoing, must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as choke points. Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet.

10.7.3 Types of Firewalls

- **Packet filters:** Packet filters are one of several different types of firewalls that process network traffic on a packet-by-packet basis. Their main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network.
- **Circuit-level gateways:** The circuit-level gateway is a proxy server that statically defines what traffic will be allowed. Circuit proxies always forward packets containing a given port number, provided the port number is permitted by the rules set. This gateway operates at the network level of an OSI model. The main advantage of a proxy server is its ability to provide Network Address Translation (NAT), which can hide the user's IP address from the Internet, effectively protecting all internal information from the Internet.
- **Application-level gateways:** An application-level gateway is a proxy server operating at the TCP/IP application level. A packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.

10.8 MALICIOUS SOFTWARE AND ANTIVIRUS

Commonly, a computer user can be tricked or forced into downloading software onto a computer that is of malicious intent. Such programs are known as malware and come in many forms, such as viruses, Trojan horses, spyware, and worms. Malicious software is sometimes used to form botnets.

Viruses are programs that can replicate their structures or effects by infecting other files or structures on a computer. The common use of a virus is to take over a computer to steal data.

Trojan horse: A Trojan horse (commonly known as a Trojan) is a general term for malicious software that pretends to be harmless so that a user willingly allows it to be downloaded onto the computer.

Spyware: The term spyware refers to programs that surreptitiously monitor activity on a computer system and report that information to others without the user's consent.

Worms: Worms are programs that can replicate themselves throughout a computer network, performing malicious tasks throughout.

Botnet: A botnet is a network of "zombie" computers that have been taken over by a "bot" that performs large-scale malicious acts for the creator of the botnet.

Antivirus: Antivirus programs and Internet security programs are useful in protecting a computer or programmable device from malware. Such programs are used to detect and usually eliminate viruses; however, it is now common to see security suites, containing also firewalls, anti-spyware, theft protection, and so on to more thoroughly protect users.

10.9 NETIQUETTE

Netiquette is the social code of network communication or in other words, netiquette is defined as a philosophy of effective internet communication that utilizes common conventions and norms as a guide for rules and standards. Netiquette is the combination of the words network and etiquette. It describes the culture of the internet. Specifically, it's the culture of how we communicate digitally through networks.

The core rules of netiquette are the regulatory parameters. These are the most agreed upon rules. Netiquette is the social code of the internet because the internet is a network and etiquette is a social code. Internet etiquette is centred around data and its use. The types of data are: text, audio, graphic, and video. Each website supports these types of data to some degree. Since internet use is new and increasing there are new words to accommodate its use, cyber words.

Proper netiquette is positive. Visit positive websites and choose online friends who share positive attitudes for proper netiquette. Live the good life online with positive and peaceful effective communication. Negative communication without a positive message is a bad netiquette.

Each website has its own etiquette because websites have different features, benefits, and users. The rules of netiquette are the common conventions of internet communication as a whole and for specific sites, netiquettes. The Golden Rule of Netiquette is to: do unto others online as you would have done to you.

Each common practice of network communication is a netiquette rule. It takes two people to create a rule of netiquette because it takes two people to create a society according to the definition.

The ten core rules of netiquette should be followed in most circumstances, but everything the situation. Freedom is part of the netiquette philosophy because the internet was created to preserve it. Net Neutrality is a policy that protects an individual's freedom to view content without preference to a given provider.

Good netiquette involves respecting others' privacy and not doing anything online that will annoy or frustrate other people. Three areas where good netiquette is highly stressed are e-mail, online chat, and newsgroups. For example, people that spam other users with unwanted e-mails or flood them with messages have very bad netiquette. You don't want to be one of those people. If you're new to a newsgroup or online chat room, it may help to observe how people communicate with each other before jumping in.

REVIEW QUESTIONS

1. Explain in detail about architecture of the Internet.
2. Describe the procedure for Internet data flow in packets from the source to the destination.
3. What is meant by internet protocol?
4. Explain the concept of IP datagram.
5. Write short note on IP datagram.
6. Write short notes on:
   (a) URL
   (b) HTTP
   (c) Domain Name System (DNS).
7. Explain in detail about types of internet security.
8. What are the functions of a firewall?
9. What are the types of Firewalls?
10. What is the purpose of Antivirus?
11. Explain in detail about the concept of Netiquette.

REFERENCES

1. Smith, A. J., Bibliography and readings on CPU cache memories and related topics. ACM SIGARCH Comput. Architecture News 14(1): 22–42., 1986.
2. Smith, A. J. 1991. Second bibliography on cache memories. ACM SIGARCH Comput. Architecture News 19(4): 154–182.
3. Talluri, M. and Hill, M.D, Surpassing the TLB performance of super-pages with less operating system support, pp. 171–182. In Proc. 6th Int. Symp. on Architectural Support for Programming Languages and Operating Systems, 1994.
4. William S. Davis and T. M. Rajkumar. Operating Systems: A Systematic View. Addison-Wesley, fifth ed. 2000.
5. Raphael A. Finkel, An Operating Systems Vade Mecum. Prentice Hall, second edition, 1988.
6. Ida M. Flynn and Ann McIver McHoes. Understanding Operating Systems. Brooks/Cole, 2000.
7. Al Geist, Adam Beguelin, and Jack Dongarra, Eds. PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing (Scientific and Engineering Computation). MIT Press, 1994.
8. D. Hildebrand. An architectural overview of QNX. Proc. Usenix Workshop on Micro-Kernels and Other Kernel Architectures, pages 113–126, 1992.
9. Lawrence J. Kenah and Simon F. Bate. VAX/VMS Internals and Data Structures. Digital Equipment Corporation, 1984.
10. Michael S. Kogan and Freeman L. Rawson, The design of operating system/2, IBM Journal of Research and Development, 27(2):90–104, June 1988.
11. Samuel J. Leffler, Marshall Kirk McKusick, Michael J.Karels, and John S. Quarterman. 4.3 BSD UNIX Operating System. Addison-Wesley, 1989.
12. Adrian Nye. Xlib Programming Manual. O'Reilly & Associates, third edition, 1992.
13. John K. Ousterhout. Tcl and the Tk Toolkit. Addison-Wesley, 1994.
14. Peter Pacheco. Parallel Programming with MPI. Morgan Kaufmann, 1997.
15. David Pogue and Joseph Schorr. Macworld Macintosh SECRETS, IDG Books Worldwide, 1993.

16. Richard Rashid, Threads of a new system. UNIX Review, pages 37–49, August 1986.
17. David Gelernter, Suresh Jagannathan: *Programming Linguistics*, The MIT Press 1990.
18. Shriram Krishnamurthi: *Programming Languages: Application and Interpretation*. Available online at: http://www.cs.brown.edu/~sk/Publications/Books/ProgLangs/
19. Anuj Batra, Jin-Meng Ho, and Kofi Anim-Appiah, *Proposal for Intelligent BT Frequency Hopping for Enhanced Coexistence*, IEEE 802.15-01/082, January 2001.
20. Code of Federal Regulations, Title 47, Chapter 1, Part 15, Section 247.
21. Oren Eliezer, *Non-Collaborative Mechanisms for the Enhancement of Coexistence Performance*, IEEE 802.15-01/092, January 2001.
22. Oren Eliezer, *Evaluation of Coexistence Performance*, IEEE 802.15-01/091, January 2001.
23. Jie Liang, *Proposal for Non-Collaborative BT Mechanisms for Enhanced Coexistence*, IEEE 802.15-01/026, January 2001.
24. Jie Liang, *Proposal for Collaborative BT and 802.11b MAC Mechanisms for Enhanced Coexistence*, IEEE 802.15-01/080, January 2001.
25. Matthew B. Shoemake, *Proposal for Power Control for Enhanced Coexistence*, IEEE 802.15-01/081, January 2001.
26. Matthew B. Shoemake, *Proposal for Non-collaborative 802.11 MAC Mechanisms for Enhancing Coexistence: Adaptive Fragmentation*, IEEE 802.15-01/083, January 2001.
27. "From DATAR To The FP-6000 Computer". *IEEE Annals of the History of Computing*. IEEE. Retrieved October 15, 2007.
28. Djafar K. Mynbaev et al, *Fiber-Optic Communications Technology,* Pearson Education Inc. 2001.

INDEX