UNIT 13 INTRODUCTION TO COMPUTER

Structure

13.1 INTRODUCTION

We have been using computers for various purposes in day-to-day life. The initial application of computers involved data processing, calculations, data presentations etc. This could be done by the computers, which were working as self-standing or stand-alone machines. When the computer was programmed with relevant instructions to perform the task, it was able to carry on those operations relentlessly. However, the advances in technology brought in the concept of communication between computers. This is commonly known as **computer networking**.

The ability of computers to 'converse' with each other has changed the application areas of computer drastically. It has thrown the whole gamut of Information and Communication Technology (ICT) open to the common users at an affordable price. You use the Internet to communicate with your friends via e-mail or retrieve data about weather in any part of the world at any given time or reserve railway/airlines /movie tickets at the click of a mouse or buy books or grocery using online shopping facilities. Please remember that all this computer-based communication takes place using digital signal transfer (the sound, image files are converted into their binary equivalent and stored in computer memory).

Another use of computers is to control various communication systems. You will appreciate that the complicated process of cellular mobile call handling, for example, cannot be managed without the aid of computers. Similarly, most of the communication systems like TV/Radio transmission, telephone exchanges, and satellite communication are controlled by computers. These computers may be stand-alone or networked.

In this unit you will learn about the basic hardware and software involved in making a computer work. In Sec. 13.2 we discuss the architecture of computers and a process of instruction execution in the computer. A brief review of evolution of computers

from microcomputers to supercomputers is taken in Sec. 13.3. In Sec. 13.4 we discuss the various types of computer software. The categories of computer language are explained in Sec. 13.5. The software necessary to achieve networking of computers is discussed in Sec. 13.6. In Sec. 13.7 we take a brief overview of the use of computer in control of modern communication systems.

Objectives

After studying this unit, you should be able to:

- describe von Neumann architecture of computer;
- classify the computers based on their capacity and application;
- list various peripheral devices;
- describe the process of instruction execution in a computer;
- discuss various types of computer software;
- spell out merits and demerits of different generations of computer languages;
- describe the software necessary for achieving network connectivity; and
- discuss some applications of computer in modern communication systems.

13.2 WORKING OF COMPUTER

To start with, let us take a brief overview of computer basics: what is the construction of a basic computer system and how does a computer execute any task given in the form of an instruction?

13.2.1 Von Neumann Architecture

Computer is defined in the Oxford dictionary as "An automatic electronic apparatus for making calculations or controlling operations that are expressible in numerical or logical terms".

The definition clearly categorises computer as an electronic apparatus although the first computers were mechanical and electro-mechanical apparatus. The definition is also pointing towards the two major areas of computer application viz. data processing and computer assisted controls/operations. The most significant characteristic pointed out in this definition is the fact that the computer can perform only those operations/ calculations, which can be expressed in logical or numerical terms.

The concept of computer is quite old. A hypothetical machine was defined in 1935-6 by Alan Turing, which was used for computability theory proofs. This is known as a **Turing machine**. It consists of an infinitely long 'tape' with symbols written at regular intervals. The tape may be infinite in one direction only, with the understanding that the machine will halt if it tries to move in other direction. All computer instruction sets, high level languages and computer architectures, including parallel processors, can be shown to be equivalent to a Turing Machine and thus equivalent to each other in the sense that any problem that one can solve, any other can solve given sufficient time and memory. In 1947 von Neumann proposed a computer architecture with centralised control unit. This rudimentary stored programme computer is referred to as **von Neumann architecture**. Most of today's computer designs are based on this architecture.

Von Neumann proposed that there should be a unit performing arithmetic and logical operation on the data. This unit is termed as **Arithmetic Logic Unit (ALU)**. One of the ways to provide instructions to such computer will be by connecting various logic components in such a fashion that they produce the desired output for a given set of inputs. The process of connecting various logic components in specific configuration to achieve desired results is called **programming**. This programming done by

providing instructions within hardware by various connections is termed as **hard-wired**. But this is a very inflexible process of programming.

Let us now have a look at the general configuration for programmable arithmetic and logical functions. For this, it is necessary to have a control signal which directs the ALU to perform a specific arithmetic or logic function on the data. By changing the control signal the desired function can be performed on the data. Any operation, which needs to be performed on the data, then, can be obtained by providing a set of control signals. Thus, for a new operation it is enough to change the set of control signals.

But, how can these control signals be supplied? Let us try to answer this from the definition of a programme. A programme consists of a sequence of steps. Each of these steps requires certain arithmetic or logical or input/output operations to be performed on the data. Therefore, each step may require a new set of control signals. Is it possible for us to provide a unique code for each set of control signals? Well, the answer is yes. But what do we do with these codes? It is necessary to have a hardware segment, which accepts a code and generates control signals in the form of voltage levels. The unit, which interprets a code to generate respective control signals is termed as **Control Unit (CU)**. Now, this machine is quite flexible, as we only need to provide a new sequence of codes (programme) for a new task to be performed.

A schematic of the von Neumann architecture of a computer is shown in Fig. 13.1. The Arithmetic Logic Unit (ALU) and the Control Unit (CU) together are termed as the **Central Processing Unit (CPU)**. The CPU is the most important component of a computer's hardware. The control unit interprets instructions and produces the respective control signals. The ALU performs the arithmetic operations such as addition, subtraction, multiplication and division, and the logical operations such as: "Is $A = B$?" (where A and B are both numeric or alphanumeric data) etc.

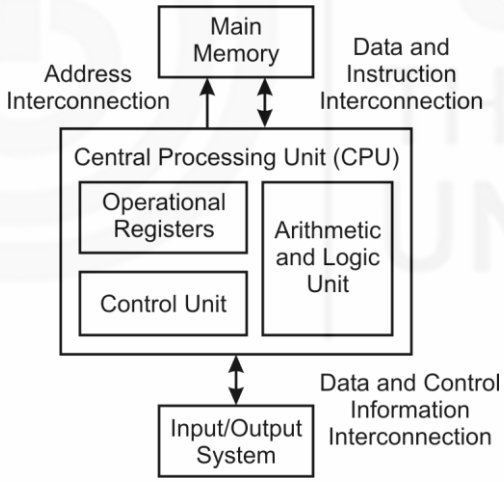

Fig. 13.1: Von Neumann architecture of a computer

All the arithmetic and logical operations are performed within the CPU in the special storage areas called **registers** shown in the figure. The size of the register is one of the important considerations in determining the processing capabilities of the CPU. Register size refers to the amount of information (number of bits) that can be held in a register at a time for processing. The larger the register size, the faster may be the speed of processing. CPU's processing power is measured in Million Instructions per Second (MIPS). The performance of the CPU was measured in milliseconds (one thousandth of a second) on the first-generation computers, in microseconds (one millionth of a second) on second-generation computers, in nanoseconds (one billionth of a second) on third-generation computers, and is measured in picoseconds (one

1000th of a nanosecond) or femtoseconds (one 1000th of a picosecond) in the later generations.

A von Neumann machine has only a single path between the main memory and control unit (CU). This constraint is referred to as von Neumann bottleneck. Several other architectures have been suggested for modern computers.

Now, how can the instructions and data be put into the computers? An external environment supplies the instruction and data, therefore, an input module is needed. The main responsibility of input module will be to put the data in the form of signals that can be recognised by the system. Similarly, we need another component, which will report the results in proper format and form. This component is called output module. These components are referred together as **input/output (I/O) devices** or **peripheral devices**. Most common input/output devices are keyboard, mouse, monitor and printer. In addition, to transfer the information, the computer system internally needs the system interconnections called **Bus** which carries data and addresses.

Are these two components sufficient for a working computer? No, because input devices can bring instructions or data only sequentially and a programme may not be executed sequentially as jump instructions are normally encountered in programming. In addition, more than one data elements may be required at a time. Therefore, a temporary storage area is needed in a computer to store temporarily the instructions and the data. This component is referred to as memory. It was pointed out by von Neumann that the same memory can be used for storing data and instructions. In such cases the data can be treated as data on which processing can be performed, while instructions can be treated as data, which can be used for the generation of control signals.

The memory unit stores all the information in a group of memory cells, also called memory locations, as binary digits. Each memory location has a unique address and can be addressed independently. The contents of the desired memory locations are provided to the CPU by referring to the address of the memory location. The amount of information that can be held in the main memory is known as memory capacity. The capacity of the main memory is measured in kilobytes (kB) or Megabytes (MB). One kilobyte stands for 2^{10} bytes, which are 1024 bytes (or approximately 1, 000 bytes). A megabyte stands for 2^{20} bytes, which is approximately little over one million bytes (1,048,576 bytes to be precise). Since the memory on the computer chip is limited, in order to provide extended memory, devices like hard discs are placed within the computer and media like floppy discs, compact discs (CD), and magnetic tapes are used as external storage (memory) devices.

The basic function performed by a computer is the execution of a programme. A programme is a sequence of instructions, which operates on a data to perform certain tasks. In the computers data is represented in binary form by using two symbols 0 and 1, which are called binary digits or **bits**. But the data which we deal in practice consists of numeric data and characters such as decimal digits 0 to 9, alphabets A to Z, arithmetic operators (e.g. +,–, etc.), relations operators (e.g. =, > , etc.), and many other special characters (e.g.;@,{,], etc.). In general, computers use eight bits to represent a character internally. This allows up to 256 different items to be represented uniquely. This collection of eight bits is called **a byte**. Thus, one byte is used to represent one character internally. Most computers use two bytes or four bytes to represent numbers (positive and negative) internally. Another term, which is commonly used in the computer, is a **word**. A word may be defined as a unit of information, which a computer can process, or transfer at a time. A word, generally, is equal to the number of bits transferred between the central processing unit and the main memory in a single step. It may also be defined as the basic unit of storage of

integer data in a computer. Normally, a word may be equal to 8, 16, 32 or 64 bits. The terms like 32-bit computer, 64-bit computers etc. basically point out the word size of the computer.

The von Neumann machine uses stored programme concept, i.e. the programme and data are stored in the same memory unit. The computers prior to this idea used to store programmes and data on separate memories. Entering and modifying these programmes were very difficult as they were entered manually by setting switches. In this machine each memory location can be addressed separately.

The peripheral and external memory devices can be connected to a computer via connecting pins called **ports**. The ports are usually of two types: **serial port**, which carries data between the computer and the device in serial fashion. Then in order to construct a 'word' or address out of these serial bits (or vice versa) there is a serial to parallel (or parallel to serial) conversion unit. Another type of port is called **parallel port** (or Centronic port) which carries data in parallel fashion. Here there are many number of pins conducting the 'word' or address simultaneously. Obviously, this communication is faster since the whole word gets transferred simultaneously. Usually printers are connected to the parallel port. Modern computers have a **universal serial bus (USB)** port which allows connection of any peripheral having USB compatibility. It is even possible to connect more than one peripheral device to a USB port, which can be used as per their priority.

SAQ 1

Spend 5 Min.

State whether following statements are true or false.

i) A byte is equal to 8 bits and can represent a character internally.

ii) Von Neumann architecture specifies different memory for data and instructions. The memory which stores data, is called data memory; and that, which stores instructions, is called instruction memory.

iii) In von Neumann architecture each bit of memory can be accessed independently.

iv) A programme is a sequence of instructions designed for achieving a task/goal.

v) One MB is equal to 1024 kB.

13.2.2 Execution of Instruction

The main aspect in programme execution is the execution of an instruction. You must be wondering as to: (a) how are the instructions supplied to the computer? and (b) how are they interpreted and executed? We will answer these questions now.

Execution of instructions in von Neumann machine is carried out in a sequential fashion (unless explicitly altered by the programme itself) from one instruction to the next. The most basic von Neumann machine in the solid state form is the microprocessor, about which you will learn in the next section. In the following, we will discuss the instruction execution process in a typical microprocessor.

The programme, which is to be executed, is a set of instructions, stored in the memory in the form of **machine language** which comprises of '0's and '1's. The central processing unit (CPU) executes the instructions of the programme to complete a task. The instruction execution takes place in the CPU registers.

Let us first discuss few typical registers which are generally available in the machines. These registers are:

Memory Address Register (MAR): It specifies the address of memory location from which data or instruction is to be accessed (for read operation) or to which the data is to be stored (for write operation).**Memory Buffer Register** (MBR): This register

contains the data to be written in the memory (for write operation) or it receives the data from the memory (for read operation).

Programme Counter (PC): It keeps track of the instruction which is to be executed next, after the execution of the on-going instruction.

Please do not confuse *personal computer* also often referred to as PC with *programme counter*.

Instruction Register (IR): Here the instructions are loaded before their execution.

Accumulator (AC): Stores the data temporarily.

The simplest model of instruction processing can be a two-step process. The CPU reads (fetches) instructions (codes) from the memory one at a time, and then executes or performs the operation specified by this instruction. The instruction fetch has to be carried out for all the instructions. It involves reading of an instruction from a memory location to the CPU. The execution of this instruction may involve several operations depending on the nature of the instruction. The processing needed for a single instruction (fetch and execution) is referred to as *instruction cycle*.

For fetch cycle, a typical CPU uses a programme counter (PC). PC keeps track of the instruction which is to be fetched next. The fetched instruction is in the form of a binary code and is loaded into an instruction register (IR) in the CPU. The CPU interprets the instruction and does the required action. In general, these actions can be divided into following three categories:

a. **Data Transfer:** From the CPU to memory or from memory to CPU, or from CPU to I/O or I/O to CPU.

b. **Data Processing:** A logic or arithmetic operation performed by the CPU on the data.

c. **Sequence Control:** This action may require alteration of sequence of execution. For example, an instruction from location 100 H, on execution may specify that the next instruction should be fetched from location 200 H. On execution of such an instruction the Programme Counter that was having location value 101 H (the next instruction to be fetched in case where memory word is equal to register size) will be modified to contain a location value 200 H.

The data and address are usually expressed in hex code and indicated by a suffix H. You have already learnt in the course on Electrical Circuits and Electronics (PHE-10) regarding conversion from binary to hex code.

Execution of an instruction may involve any combination of these actions. Let us understand the process with an example.

Example 1:

Let us assume a machine which has a 16 bit long instructions. Each instruction of the machine consists of two components: (a) operation code (op-code) and (b) address of the operand in memory.

If the op-code is 4 bits long, the remaining 12 bits are for the address of the operand as shown in Fig. 13.2a. The memory is divided into words of 16 bits. Fig. 13.2b shows a data format, which caters for the sign bit.

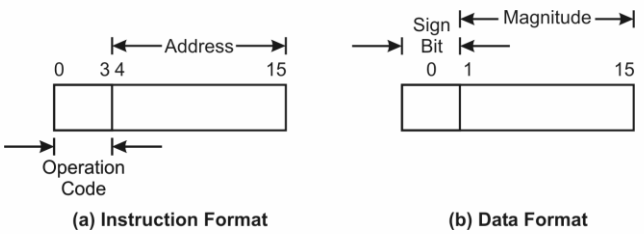

Fig. 13.2: a) Instruction; and b) data format for a typical machine

The machine can have $2^4 = 16$ possible operation codes. For example, let us assume operation with codes:

0001 as "Load the accumulator with the content of memory"
0010 as "Store the current value of Accumulator in the memory"
0011 as "Add the value from memory to the Accumulator"

This machine can address $2^{12} = 4096$ memory words directly. Remember that in this case the programme counter is of 12 bits to accommodate address of the memory where the programme resides.

Let us assume that three consecutive instructions to be executed are

Instruction code				Operation desired
0001	0111	0110	0000	(Load accumulator with content at memory location 760 H)
0011	0111	0110	0001	(Add value from memory 761 H to accumulator)
0010	0111	0110	0000	(Store content of AC in the memory location 760 H)

The hexadecimal notation for these instructions is:

1 7 6 0 H
3 7 6 1 H
2 7 6 0 H

Let us assume that these instructions are stored in three consecutive memory locations 101 H, 102 H and 103 H and the PC contains 101 H, which is the address of first of these instructions as shown in Fig. 13.3a.

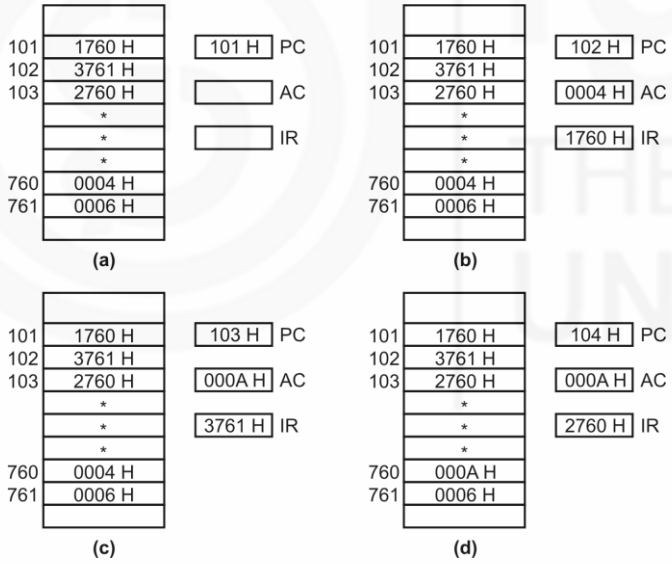

Fig 13.3: Memory and register content during execution of instructions: a) before execution; b) after first instruction execution; c) after second instruction execution; and d) after final execution

The PC contains value 101 H, so on the next instruction cycle the address stored in PC is passed to MAR (not shown here), which in turn helps in accessing the memory location 101 H and brings its content in MBR which in turn passes it on to IR. The PC is incremented to contain 102 H now. The IR has the value 1760 H that is decoded as "Load the content of address 760 H in the accumulator". (Refer to Fig. 13.3b). Thus, the accumulator register is loaded with the content of location 760 H that is 0004 H.

Now the instruction 101 H execution is complete, and the next instruction at 102 H (indicated by PC) is fetched and PC is incremented to 103 H. This instruction is

3761 H which implies "add the content of memory location 761 H to the accumulator". Therefore, accumulator will now contain the sum of its earlier value and the value stored in memory location 761 H. (Refer to Fig. 13.3c)

On execution of the instruction at memory location 103 H, PC becomes 104 H; the accumulator results are stored in location 760 H and IR still contains the third instruction. This state is shown in Fig. 13.3d.

You should note that the execution of the instructions in the above example required only data transfer and data processing operations. All the instructions in the example required one memory reference during their execution. Does an instruction require more than one memory reference?

Well, yes! One such instruction is ADD *A*, *B*. The execution cycle of this instruction may consist of steps such as:

- Decode the instruction, which is ADD.
- Read the contents of memory location *A* into the CPU.
- Read the contents of memory location *B* into the CPU. (Here, we are assuming that the CPU has at least two registers for storing memory location contents. The contents of location *A* and location *B* need to be written in two different registers).
- Add the values of the two above registers.
- Write back the result from the register containing the sum (in the CPU) to the memory location *B*.

Thus, in general, the execution cycle for a particular instruction may involve more than one stage and memory reference. In addition, an instruction may ask for an I/O operation. For such purposes the ports where the I/O devices are connected are allotted addresses just like any other memory location. When a data from such device is required, the address of the device is placed in the instruction instead of the address of a memory location.

Spend 5 Min.

SAQ 2

What will be the steps involved in interchanging the data in memory location *A* and memory location *B*?

After understanding the fundamental operation of a computer, let us now briefly discuss the evolution that has occurred in the computers over the years.

13.3 EVOLUTION OF COMPUTERS

You know that, with the growth in microelectronics, the integrated circuit (IC) technology evolved rapidly. One of the major milestones in this technology was the very large scale integration (VLSI) where thousands of transistors could be integrated on a single chip. The main impact of VLSI was that it was possible to produce a complete CPU or main memory or other similar devices on a single IC chip. This implied that mass production of CPU, memory etc. can be done at a very low cost. Let us take a brief review of the important breakthroughs of VLSI technologies.

Semiconductor Memories: Initially the IC technology was used for constructing processors, but soon it was realised that the same technology can be used for construction of memory. The first memory chip was constructed in 1970 and could hold 256 bits. Although the cost of this chip was initially high, gradually it is going down. The memory capacity per chip has increased as: 1k, 4k, 16k, 64k, 256k and 1M

bytes or even more. These memories can be random access memory (RAM), read only memory (ROM) or erasable, programmable read only memory (EPROM).

Microprocessors: Keeping pace with electronics as more and more components were fabricated on a single chip, fewer chips were needed to construct a single processor. Intel in 1971 achieved the breakthrough of putting all the components on a single chip. The single chip processor is known as a microprocessor. The Intel 4004 was the first microprocessor. It was a primitive microprocessor designed for a specific application. Intel 8080, which came in 1974, was the first general-purpose microprocessor. It was an 8-bit microprocessor. Motorola is another manufacturer in this area. At present 32- and 64-bit general-purpose microprocessors are already in the market. For example Intel 486 is a 32-bit processor, similarly Motorola's 68000 is a 32 bit microprocessor. Pentium, which was developed by Intel in 1993, processes integers as long as 64 bits. The VLSI technology is still evolving more and more powerful microprocessors and more storage space now is being put in a single chip.

In Fig. 13.4 we depict the evolution of Intel microprocessor families, which started with 8-bit chip to develop to latest available 32/64-bit Pentium chips.

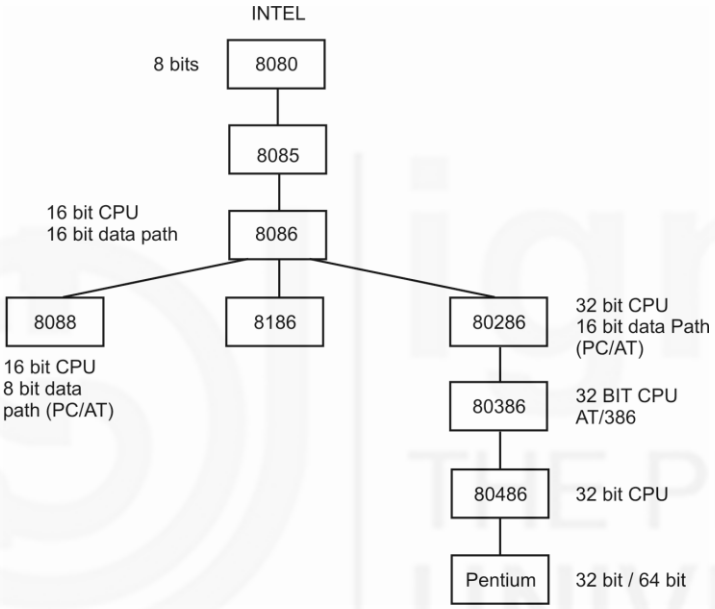

Fig. 13.4: Intel microprocessor families

In the Foundation Course in Science and Technology (FST-1), you have already learnt that computers are broadly classified as micro-computers, mini-computers, mainframe computers and supercomputers. Although with development in technology the distinction between all these is becoming blurred, yet it is important to classify them as it is useful to differentiate the key elements and architecture among the classes.

a. Micro-computers

The CPU of a micro-computer is a microprocessor. The micro-computer originated in late 1970's. The first micro-computers were built around 8-bit microprocessor chips. What do we mean by an 8-bit chip? It means that the chip can retrieve instructions/ data from storage, manipulate, and process an 8-bit data at a time or we can say that the chip has a built-in 8-bit data transfer path. 8088 was a 8/16 bit chip i.e. an 8-bit path was used to move data between chip and primary storage (external path), but processing was done within the chip using a 16-bit path (internal path) at a time. 8086 was a 16/16 bit chip i.e. the internal and external paths both were 16 bit wide. Both these chips could support a primary storage capacity of up to 1 MB.

b. Mini-computer

The term mini-computer originated when it was realised that many computing tasks do not require an expensive contemporary mainframe computer but can be solved by a small, inexpensive computer. Initial mini-computers were 8-bit and 12-bit machines but by 1970's almost all mini-computers were 16-bit machines. The 16-bit mini-computers have the advantage of large instruction set and address field and efficient storage and handling of text, in comparison to lower bit machines. Thus, 16-bit mini-computer was a more powerful machine, which could be used in a variety of applications and could support business applications along with the scientific applications.

With the advancement in technology the speed, memory size and other characteristics developed and the mini-computer was then used for various stand-alone or dedicated applications. The mini-computer was then used as a multi-user system which can be used by various users at the same time. This was actually the first conceptual realisation of computer networking, where the multiple users were connected using hardware. The concept started from the fact that the CPU performed the processing at much faster rates than those required to entre new data and its display. The spare unused time of CPU could be used by other users. Gradually the architectural requirement of mini-computers grew and a 32-bit mini-computer, which was called super-mini, was introduced. The super-mini had more peripheral devices, larger memory and could support more users working simultaneously on the computer in comparison to previous mini-computers.

c. Mainframes

Mainframe computers are generally 32-bit machines or on the higher side. These are suitable for organisations, to manage high volume applications. Few popular mainframe series are DEC, IBM, HP, ICL, MEDHA, Sperry, etc. Mainframes are also used as central host computers in distributed systems. Libraries of application programmes developed for mainframe computers are much larger than those of the micro- or mini-computers because of their evolution over several decades as families of computing. All these factors and many more make the mainframe computers indispensable even with the popularity of micro-computers.

d. Supercomputers

The upper end of the state of the art mainframe machine is the supercomputers. These are among the fastest machines in terms of processing speed and use multiprocessing techniques, where a number of processors are used to solve a problem. There are a number of manufacturers who dominate the market of supercomputers-CRAY (CRAY YMP, CRAY 2), ETA (CDC-ETA 10, ETA 20), IBM 3090 (with vector), NEC (NEC SX-3), Fujitsu (VP Series) and HITACHI (S Series) are some of them. Lately ranges of parallel computing products, which are multiprocessors sharing common buses, have been in use in combination with the mainframe supercomputers. The supercomputers are reaching up to speeds well over 25000 million arithmetic operations per second. India has also developed its indigenous supercomputer PARAM.

Supercomputers are mainly being used for number crunching problems such as weather forecasting, computational fluid dynamics, remote sensing, image processing, biomedical applications, etc.

Spend 5 Min.

SAQ 3

i) What is a general-purpose machine?

ii) What are the advantages of IC technology over discrete components?

The processor, memory and peripheral devices comprise the *hardware* part of the computer. However, all these parts function only because of the software incorporated in the computers. Let us now take a brief look at computer software.

13.4 COMPUTER SOFTWARE

Computer software consists of sets of instructions that provide the raw arithmetic and logical capabilities to the hardware units to perform. Computer software can be broadly classified into two categories: **Systems Software** and **Application Software.**

Today, there are many languages available for developing programme software. These languages are designed keeping in mind some specific areas of applications. Thus, some of the languages may be good for writing system programme/software and some other for application software.

i) **System Programming Languages:** System programmes are designed to make the computer easier to use. An example of system software is an operating system, which consists of many programmes for controlling input/output devices, memory, processor etc. To write an operating system, the programmer needs instructions to control the computer's circuitry (hardware part). For example, instructions that move data from one location of storage to a register of the processor. Today C-language is widely used to develop system software.

ii) **Application Programming Language:** Application programmes are designed for specific computer applications such as payroll processing, inventory control, word processing etc. To write programmes for pay-roll processing or other applications, the programmer does not need to control the basic circuitry of a computer. Instead the programmer needs instructions that make it easy to input data, produce output, do calculations and store and retrieve data. Programming languages that are suitable for such application programmes support these instructions but not necessarily the types of instructions needed for development of system programmes.

There are two main categories of application programmes: business programmes and scientific application programmes. Most programming languages are designed to be good for one category of application but not necessarily for the other, although there are some general-purpose languages that support both types. Business applications are characterised by processing of high volume data but call for simple calculations. Languages which are suitable for business programme development must support high volume input, output and storage but need not support complex calculations. On the other hand, programming languages that are designed for writing scientific programmes contain very powerful instructions for calculations but rather poor instructions for input, output etc. Among traditionally used programming languages, COBOL (Commercial Business Oriented Programming Language) is more suitable for business applications whereas FORTRAN (Formula Translation Language) and C-language are more suitable for scientific applications. Before we discuss more about language let us briefly look at the categories of software viz. system and application software.

13.4.1 System Software

Operating System

An operating system (OS) is the most important system software and is a must to operate a computer system. An operating system manages computer's resources very effectively, takes care of scheduling multiple jobs for execution and manages the flow of data and instructions between the input/output units and the main memory.

Operating system became a part of computer software with the second-generation computers. Since then operating systems have undergone several revisions and modifications in order to achieve a better utilisation of computer resources. Advances in the field of computer hardware have also helped in the development of more efficient operating systems.

Language Translator

A language translator is a system software which translates a computer programme written by a user into a machine understandable form. We will discuss about the two prominent translators viz. the Compiler and the Interpreter in the next section.

Utilities

Utility programmes are those which are very often requested by many application programmes. A few examples are:

1. SORT/MERGE for sorting large volumes of data and merging them into a single sorted list.

2. Transfer programme for transforming contents from one storage medium to another, e.g. disk to tape, tape to disk, etc.

Special Purpose Software

Special purpose programmes are those which extend the capability of operating systems to provide specialised services to application programmes. A few examples are:

1. Spreadsheet software like LOTUS, VISICALS, etc.

2. Data management software like dBASE III, Unify, etc.

These programmes can also be used as stand-alone application programmes.

13.4.2 Application Software

Application software is written to enable the computer to solve a specific data processing task. There are two categories of application software:

- pre-written software packages, and
- user application programmes.

A number of powerful application software packages, which do not require significant programming knowledge while using them, have been developed. These are easy to learn and use as compared to the programming languages. Although these packages can perform many general and special functions, there are applications where these packages are not found adequate. In such cases, the application programme is written to meet the exact requirement. A user application programme may be written using one of these packages or a programming language. Some important categories of software packages available are:

- Data Base Management Software
- Word Processing, Desktop Publishing (DTP) and Presentation Software
- Graphics Software
- Data Communication Software
- Statistical and Operational Research Software.

SAQ 4

Spend 4 Min.

Explain the following terms in one or two sentences each: (a) Operating Systems, and (b) Database Management Software.

13.5 CATEGORIES OF LANGUAGES

You can choose any language for writing a programme according to the need. But a computer executes programmes only after they are represented internally in binary form (sequences of 1s and 0s). Programmes written in any other languages must be translated to the binary representation of the instructions before they can be executed by the computer. A programme written for a computer may be in one of the following categories of languages.

13.5.1 Machine Language

This is a sequence of instructions written in the form of binary numbers consisting of 1s, 0s to which the computer responds directly. You have used this language in Example 1. The machine language was initially referred to as code, although now the term code is used more broadly to refer to any programme text.

As you have already learnt, an instruction prepared in any machine language will have at least two parts. The first part is the Command or Operation, which tells the computer, what function, is to be performed. All computers have an operation code for each of its functions. The second part of the instruction is the operand or it tells the computer where to find or store the data that has to be manipulated.

Just as hardware is classified into generations based on technology, computer languages also have a generation classification based on the level of interaction with the machine. Machine language is considered to be the first generation language.

Advantage of Machine Language

It is faster in execution since the computer directly starts executing it.

Disadvantage of Machine Language

It is difficult to understand and develop a programme using machine language. Anybody going through this programme for checking will have a difficult task understanding what will be achieved when this programme is executed. Nevertheless, the computer hardware recognises only this type of instruction code.

13.5.2 Assembly Language

When we employ symbols (letter, digit or special characters) for the operation part, the address part and other parts of the instruction code, this representation is called an assembly language programme. This is considered to be the second generation language.

Machine and Assembly language are referred to as low-level languages since the coding for a problem is at the individual instruction level.
Each machine has got its own assembly language, which is dependent upon the internal architecture of the processor.

An **Assembler** is a translator, which takes its input in the form of an assembly language programme and produces machine language code as its output.

The following programme is an example of an assembly language programme for

adding two numbers *X* and *Y* and storing the result in some memory location.

```
LDA, 7          ;    Load register A with 7
LDB, 10         ;    Load register B with 10
ADD A, B        ;    A ← A + B
LD (100), A     ;    Save the result in the location 100
HALT            ;    Halt process
```

From this programme, it is clear that usage of mnemonics (in our example LD, ADD, HALT are mnemonics) has improved the readability of our programme significantly.

An assembly language programme cannot be executed by a machine directly as it is not in a binary form. An assembler is needed in order to translate an assembly language programme into the object code executable by the machine.

Advantage of Assembly Language

Writing a program in assembly language is more convenient than in machine language. Instead of binary sequence, as in machine language, it is written in the form of symbolic instructions. Therefore, it gives better readability than the machine language programme.

Disadvantages of Assembly Language

Assembly language (programme) is specific to a particular machine architecture. Assembly languages are designed for specific makes and models of microprocessors. It means that assembly language programmes written for one processor will not work on a different processor if it is architecturally different. That is why the assembly language programme is *not portable*.

Assembly language programme is not as fast as machine language since it has to be first translated into machine (binary) language code.

13.5.3 High-Level Language

You must have already heard about the programming languages such as COBOL, FORTRAN, BASIC, PASCAL, JAVA, C etc. They are called high-level programming languages. The time and cost of creating machine and assembly language was quite high. And this was the prime motivation for the development of high-level languages. The programme shown below is written in BASIC to obtain the sum of two numbers

```
10    LET   X     =    7
20    LET   Y     =    10
30    LET   SUM   =    X + Y
40    PRINT SUM
50    END
```

The high-level source programme must be translated first into the form that the machine can understand. This is done by a software called **compiler** which takes the source code as input and produces an output in the machine language code of the machine on which it is to be executed.

During the process of translation, the compiler reads the source programme statement-wise and checks the syntax (grammatical) errors. If there is any error, the computer generates a print-out of the errors it has detected. This action is known as **diagnostics.**

There is another type of software, which also does the translation. This is called an **interpreter**. The compiler and interpreter have different approaches to translation. Table 13.1 lists the differences between a Compiler and an Interpreter.

Table 13.1: Comparison of compiler and interpreter

Compiler	Interpreter
1. Scans the entire programme first and then translates it into machine code.	Translates the programme line by line.
2. Converts the entire programme to machine code; when all the syntax errors are removed, execution takes place.	Each time the programme is executed every line is checked for syntax error and then converted to equivalent machine code.
3. Slow for debugging (removal of mistakes from a programme).	Good for fast debugging.
4. Execution time is less.	Execution time is more.

Advantage of High-level Programming Language: There are four main advantages of high-level programming languages. These are:

i) **Readability:** Programmes written in these languages are more readable than assembly and machine language.
ii) **Portability**: Programmes could be run on different machines with little or no change. We can, therefore, exchange software leading to creation of programme libraries.
iii) **Easy debugging**: Errors could easily be removed (debugged).
iv) **Easy Software development**: Software could be easily developed. Commands of programming language are similar to natural languages (English).

13.5.4 Fourth Generation Language

The fourth generation of programming languages is not as clearly defined, as are the other earlier generations. Most people feel that a fourth generation language, commonly referred to as 4GL is a high level language that requires significantly fewer instructions to accomplish a task than a third generation language does. Thus, a programmer should be able to write a programme faster in 4GL than in a third generation language.

Most third generation languages are procedural languages. This means that the programmer must specify the steps, that is the procedure the computer has to follow in a programme. By contrast, most fourth generation languages are non-procedural languages. The programmer does not have to give the details of procedure in the programme but instead, specifies what is wanted. For example, assume that a programmer needs to display some data on a screen, such as the address of a particular employee (SANTOSH) from the personnel file. In a procedural language, the programmer would have to write a series of instructions in the following steps:

Step 1 : Get a record from the personnel file
Step 2 : If this is the record for SANTOSH, display the address
Step 3 : If this is not the record for SANTOSH, go to Step 1.

In a non-procedural language (4GL), however, the programmer would write a single instruction that says:

Get the address of SANTOSH from personnel file.

Major fourth generation languages are used to get information from files and data bases, as in the above example and to display or print the information. Fourth generation languages are mostly machine independent. Usually they can be used on more than one type of computer. They are mostly used for office automation or business applications, but not for scientific programmes. Some fourth generation languages are designed to be easily learnt and used by end users.

Spend 5 Min.

SAQ 5

State whether the following statements are True or False.

(a) Assembly language programmes are machine independent.
(b) Machine language programmes are machine independent.
(c) High level languages are machine independent.
(d) All programmes are machine independent.
(e) 4th generation languages are dependent on the databases they use.

So far we have discussed the working of stand-alone computers. They require operating systems to perform the job. The peripheral devices require special software called '**drivers**' for their operation. To establish the connectivity with other computers, special software called **Networking Software** is required.

13.6 NETWORKING SOFTWARE

Networking support is typically provided by two software components: High-Level Networking Software, and Network Driver Software.

High-level Networking Software:

High-level Networking Software provides end-user-oriented functions. This is the software that the end user perceives. Some types of high-level networking software subsystems, especially in the personal computer environment, are called Network Operating Systems (NOS).

Network Driver Software:

Network Driver Software provides an interface between the high-level networking software and the particular Network Interface Card (NIC) that is being used for physical implementation of computer network communication. Like the NIC itself, the driver software is generally transparent to the end user.

13.6.1 Network Operating System (NOS)

It is an operating system which includes software to communicate with other computers via a network and manages network resources.

A network operating system (NOS) causes a collection of independent computers to act as one system. A network operating system is analogous to a desktop operating system like DOS or OS/2, except that it operates over more than one computer. Like DOS, a network operating system works behind the scenes to provide services for users and application programmes. But instead of controlling the pieces of a single computer, a network operating system controls the operation of the network system, including who uses it, when they can use it, what they have access to, and which network resources are available.

At a basic level, the NOS allows network users to share files and peripherals such as disks and printers. Most NOSs do much more. They provide data integrity and

security by keeping people out of certain resources and files. They have administrative tools to add, change, and remove users, computers, and peripherals from the network. They have troubleshooting tools to tell the network managers what is happening on the network. They have internetworking support to tie multiple networks together.

It manages multiple requests (inputs) concurrently and provides the security necessary in a multi-user environment. It may be a completely self-contained operating system, such as NetWare, UNIX and Windows NT, or it may require an existing operating system in order to function.

In case of client/server network, to be discussed in the next unit, one piece of the network operating system resides in each client machine and another resides in each server. It allows the remote drives on the server to be accessed as if they were local drives on the client machine. It allows the server to handle requests from the client to share files and applications as well as network devices such as printers, faxes and modems. In a peer-to-peer network, the network operating system allows each station to be both client and server.

Along with file and print services, a network operating system may also include directory services and a messaging system as well as network management and multiprotocol routing capabilities.

In networks of PCs, NetWare is the most widely used network operating system. Windows NT, Windows 95/98, UNIX are also examples.

a. NetWare

It is a family of network operating systems from Novell that support DOS, Windows, OS/2 and Macintosh clients. UNIX client support is available from third parties. NetWare is the largest installed base of computer network operating systems.

NetWare is a stand-alone operating system that runs in the server. Its hard disks are formatted with the NetWare format, and although DOS and Windows applications reside in the server, they cannot be run in the server. All programmes that run on a NetWare server are typically written in C language and must be compiled using Novell libraries into executable files known as NetWare Loadable Modules (NLMs).

b. UNIX

Pronounced *yoo-niks*, it is a multi-user, multitasking operating system that is widely used as the master control programme in workstations and especially servers. Myriads of commercial applications run on UNIX servers, and most Websites run under UNIX. There are many versions of UNIX, and, except for the PC world, where Windows dominates, almost every hardware vendor offers it either as its primary or secondary operating system. Sun has been singularly instrumental in commercialising UNIX with its Solaris OS (formerly SunOS). HP, SCO, IBM and Digital have also been major UNIX vendors and promoters.

UNIX is written in C. Both UNIX and C were developed by AT&T and freely distributed to government and academic institutions, causing it to be ported to a wider variety of machine families than any other operating system. As a result, UNIX became synonymous with *open systems*.

The softwares are normally divided into proprietary (e.g. Windows) and free software systems like LINUX. The source code of the free systems is made public and anybody can use it or modify it.

UNIX is made up of the kernel, file system and shell (command line interface). The major shells are the Bourne shell (original), C shell and Kern shell. The UNIX vocabulary is exhaustive with more than 600 commands that manipulate data and text in every way conceivable.

c. Windows NT

Windows New Technology is an advanced 32-bit operating system from Microsoft for Intel x86 and Alpha CPUs. Introduced in 1993, NT does not use DOS, it is a self-contained operating system that runs 16-bit and 32-bit Windows applications as well as DOS applications.

There are actually two versions of Windows NT: Windows NT Server, designed to act as a server in networks, and Windows NT Workstation for stand-alone or client workstations.

Features of NT include peer-to-peer networking, pre-emptive multitasking, multithreading, multiprocessing, fault tolerance and support for the Unicode character set. NT provides extensive security features and continually tests the validity of application requests even after the application has been opened.

Windows NT supports 2GB of virtual memory for applications and 2GB for its own use. Windows NT and Windows NT Workstation are the first and second releases of the client version. Windows NT Advanced Server (NTAS) and Windows NT Server (NTS) are first and second releases of the server version, which support symmetric multiprocessing (SMP) and provide transaction processing for hundreds of online users.

13.6.2 High-Level Networking Software Systems

a. PPP

Protocol is a standard set of rules according to which all the computers in a network exchange information, so that it is interpreted and understood by all.

Short for Point-to-Point Protocol, this is a method of connecting a computer to the Internet. PPP is more stable than the older SLIP protocol and provides error-checking features. It is a data link protocol that provides dial-up access over serial lines. It can run on any full-duplex link from normal telephone lines to ISDN to high-speed lines. Developed by the Internet Engineering Task Force in 1991, it has become popular for Internet access.

Over ISDN, PPP uses one 64 kbps *B* channel for transmission. The Multilink PPP protocol (MP, MPPP or MLPPP) bridges two or more *B* channels for higher-speed operation.

b. SLIP

Serial Line Internet Protocol (SLIP) is commonly used to gain access to the Internet as well as to provide dial-up access between two computer networks. SLIP transmits data over any serial link (dial up or private lines).

c. FTP

File Transfer Protocol is used to transfer files over a computer network (Internet, UNIX, etc.) It includes functions to log on to the network, list directories and copy files. It can also convert between the ASCII and EBCDIC character codes. FTP operations can be performed by typing commands at a command prompt or via an FTP utility running under a graphical interface such as Windows. FTP transfers can also be initiated from within a Web browser by entering the URL preceded by ftp:// as you will learn later in Unit 15.
Unlike e-mail programmes in which graphics and programme files have to be "attached," FTP is designed to handle binary files directly and does not add the overhead of encoding and decoding the data.

d. TELNET

A terminal emulation protocol is commonly used on the Internet and other networks. It allows a user at a terminal or computer to log onto a remote computer and run a programme. Telnet was originally developed for ARPAnet and is an inherent part of the TCP/IP communications protocol, about which you will be learning in the next unit.

ARPAnet, a protocol developed by Advanced Research Projects Agency is the world's first package switching network and the progenitor of Internet.

e. SMTP

Simple Mail Transfer Protocol (SMTP) is the standard e-mail protocol on the Internet. It is a protocol that defines the message format and the message transfer agent (MTA), which stores and forwards the mail. SMTP was originally designed for only ASCII text, but MIME and other encoding methods enable programme and multimedia files to be attached to e-mail messages.

f. POP

The Post Office Protocol (POP) provides a standard mechanism for retrieving e-mails from a remote server. This is useful when the recipient of the mail is not permanently connected to the Internet. The POP server stores the recipient's mails and delivers them to user whenever s/he connects to the Internet.

g. SNMP

Simple Network Management Protocol is a widely-used network monitoring and control protocol. Data is passed from SNMP agents, which are hardware and/or software processes reporting activity in each network device to the workstation console used to oversee the network. The agents return information contained in a MIB (Management Information Base), which is a data structure that defines what is obtainable from the device and what can be controlled (turned off, on etc.). Originating in the UNIX community, SNMP has become a widely used protocol on all major platforms.

13.6.3 Network Security

Security refers to techniques for ensuring that data stored in a computer cannot be read or compromised. Most security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phrase that gives a user an access to a particular programme or system.

Passwords can be checked by the operating system to prevent users from logging onto the system in the first place, or they can be checked in software, such as DBMSs, where each user can be assigned an individual view (subschema) of the database. Any application programme running in the computer can also be designed to check for passwords. Data transmitted over communications networks can be secured by encryption to prevent eavesdropping.

There are two types of security available for use on the network. The type of security you use depends largely on the type of network and the operating system. Workgroups depend on share-level security while domains employ user-level security.

a. Share-level Security

Share-level security involves assigning a password to resources shared on the network. A user needs a password to access the resource. The same resource can be shared with different permissions and different passwords. The level of access to the resource depends on which password one uses to access it. This allows the resource to be shared as read-only, and the password for this share is given to the users who need

to view the resource. The resource could be then shared as full access. The users that use the password assigned to the full access could delete, change, and read the data. However, this method of security can be difficult to maintain. Users may have to remember several passwords in order to access all the resources needed to perform their jobs.

b. User-level Security

Most networks share data with user-level security. User-level security requires the proper user name and password to access a resource. When resources are shared, permission is granted to certain users or a group of users. Only those user accounts can access the resource. User-level security not only provides higher-level security, it also allows a wide variety of permissions like write, add, change, delete and so on. The added security and flexibility makes user-level security the preferred method for networks of over about ten computers.

Spend 4 Min.

SAQ 6

What are the differences between these two levels of security?

Just as computers communicate with each other via networks, they also help in functioning of other communication systems like telephones, audio-video communication, satellite communication etc.

13.7 COMPUTER IN COMMUNICATION SYSTEMS

All the modern communication systems of present day are invariably aided by computers in some way or the other. These can be stand-alone personal computers used to grab and process images from of a digital camera or wide area networks acting as the backbone of a telecommunication system.

In the telephone systems, the electronic exchanges are controlled entirely by computers. These computers help in performing the switching operation in call handling as well as work as administrative tools to keep track of customer billings, act as voice mail boxes which allow retrieval of messages when the customer wishes. The computers are also used in setting up interactive voice services, which can guide the callers to appropriate help requested by them. The large databases like telephone directories are stored in the computer and can be used as online directory services.

In case of commercial establishments the in-house telephones are generally handled by EPBX (Electronic Private Branch Exchange) systems. These are normally controlled by a personal computer. In mobile telephony the operation of call handling is very complicated and is realisable only with computer aid. Determination of free frequency channel, its allotment for a particular conversation and communicating this allotment to called handset and handing over the call to next base station when signal level goes beyond minimum set level, is all achieved with the help of computers.

The satellite based communication is possible because of geo-stationary satellites placed in their orbits. The entire control of the satellite function (not just its use in communication) is carried out by the computer based master control facility at the earth station. Further, the trans-receiving of satellite communication signals is done by computer-aided systems.

The advancement of software supporting the computer operations allow the technologists to use simulators to construct virtual models of the systems they design. Right from designing the shape and size of the antenna to achieve required radiation pattern to designing very high density large scale integrated circuits (VLSI) to provide

efficient communication ICs, the computers are being used in every field of communication these days.

Let us now summarise the points you learnt in this unit.

13.8 SUMMARY

- Von Neumann architecture defines the structure of a general purpose computer.
- It consists of a Central Processing Unit (CPU) comprising of Arithmetic Logic Unit (ALU), Control Unit (CU) and Operational Registers like Accumulator (AC), Programme Counter (PC) etc. CPU is connected to external memory via data/address bus and this memory is mapped by assigning addresses to each memory unit.
- It is possible to attach many peripheral I/O devices like keyboard, mouse, monitor, printer, floppy and CD drivers etc to a computer.
- Execution of instruction is done by converting it into machine readable language.
- According to the capacity, the computers are classified as micro-computers, mini-computers, mainframe computers and supercomputers.
- Network operating systems provide necessary software support to handle computer networking.
- Many communication systems have computers as the central controller for their functioning.

13.9 TERMINAL QUESTIONS *Spend* 15 *Minutes*

1. Define the following terms:
   (i) Microprocessors
   (ii) Lap-top
   (iii) Supercomputer
2. What is computer software?
3. What is the difference between ftp and telnet?

13.10 SOLUTIONS AND ANSWERS

Self Assessment Questions

1. (i) True; (ii) False; (iii) False; (iv) True; (v)True

2. The steps involved are:

   - Move data from memory location *A* to temporary register 1.
   - Move data from memory location *B* to AC.
   - Move data from AC to location *A.*
   - Move data from temporary register 1 to memory location *B.*

3. i) A machine, which can be used for a variety of applications and is not modelled for specific applications only. Von Neumann machines are general-purpose machines since they can be programmed for any general application, while the a microprocessor based control systems are not general-purpose machines as they are specifically modelled as control systems.

ii) Low cost; increased operating speed; reduction in size of the computers; reduction in power and cooling requirements; and more reliable performance.

4. a) Operating system is a system software for effective management of computer resource. It is a must for operating a computer system.

   b) A software that maintains and integrates large volume of data and provides simpler user interaction.

5. a. False; b. False; c. True; d. False; e. False

6. In share-level security access control to a file, printer or other network resource is based on knowing the password of that resource. Share-level security provides less protection than user-level security, which identifies each person in the organisation.

   In user-level security access control to a file, printer or other network resource is based on username. It provides greater protection than share-level security, because users are identified individually or within a group.

Terminal Questions

1. i) Microprocessor is a complete processor constructed on a single chip using VLSI technology. Intel and Motorola are two popular concerns which are producing popular Microprocessors.

   ii) A very small, battery operated micro-computer, which uses liquid crystal display technology and is very handy to carry.

   iii) The upper end of the state-of- the art mainframe machines with very powerful computational capabilities. These are mainly suitable for very large processing requirements such as weather forecasting.

2. Refer to Sec. 13.4.

3. If ftp, the files from the host machine are transported to the client machine. Apart from the request to send the file and actual file transfer, there is no dialog between the client and host.

   In telnet, you can remote login to the host computer and perform the actions, just as if you are sitting in front of the host machine. Here, it is possible to execute programmes on the host machine as well as modify files stores in the host machine.