6
MEMORY UNIT

6.0. INTRODUCTION

Every computer system consists of a variety of devices to store the instructions and data required for its operation. The instructions and data are stored in the memory unit of the computer. The memory components of a computer system can be divided into following main groups:

1. **Main Memory (Primary Memory):** It is characterised by the fact that locations in main memory can be directly accessed by the CPU instruction set. This is used for program and data storage during computer operation.

2. **Secondary Memory (Auxiliary or Backup):** It is much larger in size but much slower than main memory and is used for storing system programs and large data files which are continually not required by CPU.

3. **Internal Processor Memory:** This usually comprises a small set of high speed registers used as working registers for temporary storage of instructions and data.

6.1. NEED FOR MEMORY

The computer executes the instructions one after the other. Thus, it is necessary to have a memory in which all the instructions and data can be stored first and can be taken one after other later for execution. For some periodical works such as the preparation of pay bill, pay-roll program is used repeatedly and the data are required to store to avoid to feed them again and again. Thus, the memory is needed for the following purposes:

- To store the program and data during execution.
- To store the program for repetitive use.
- To store the data for future/periodical use.
- The store the results of execution.

6.2. MAIN (PRIMARY) MEMORY

The sole function of the memory unit is to store program and data. Physically it consists of various storage locations each of which can accommodate, a certain number of bits which is either 0 or 1. The smallest unit of information that can be stored by a digital computer is called Bit (Binary digit). A collection of 4 bits is called a NIBBLE. A combination of eight bits is called a Byte. If we wish to store the character A, a byte is required. A number of bits/bytes are grouped together to form a WORD (Fig. 6.1). A word is a set of bits processed as a unit. The fixed number of bits that a word can store is called the *word length* which varies from computer to computer. Thus, if we term the CPU of some computer as eight bit, then it means that the basic adder unit of that computer can add/subtract simultaneously two eight bit numbers and the word length of that computer is also eight bit. The word size of PDP 11/70 is 16 bits, VAX 11/750 is 32 bits while CDC Cyber has 64 bits word.

Though most computers use fixed length words, computers with variable length words also exist.

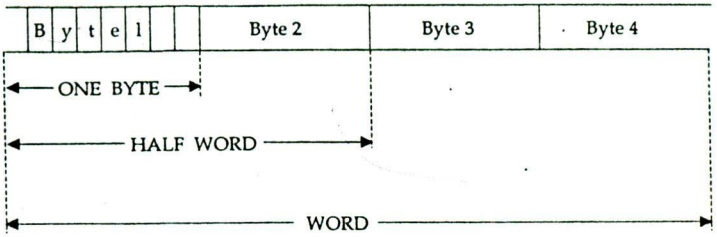


Fig. 6.1

The size of the memory is often measured in terms of total number of storage locations in it and is expressed in kilobytes (KB or K) for smaller computers and megabytes (MB or M) for the larger machines. Kilo means one thousand in the metric system but the computer industry uses K to represent 1024 ($=2^{10}$). Thus, 1 KB computer storage has exactly 1024 storage locations from 0000 to 1023. A microcomputer with 64 KB storage locations has approximately 64,000 (or exactly 65, 536) storage locations of memory. However, such differences are frequently disregarded in order to simplify descriptions of storage capacity. Thus, a megabyte (MB) is roughly 1 million (or exactly 2^{20}) bytes of storage, while a gigabyte (GB) is roughly 1 billion bytes and a terabyte represents about 1 trillion bytes.

At present two kinds of main memory are commonly used in modern computers: (*i*) Semiconductor memory, (*ii*) Magnetic memory.

6.2.1. Memory Access

Instructions or data are written into or read from the memory one word at a time. Reading from the memory or writing into the memory is usually called **Memory Access.** Each word has a unique address used to identify the location which allows direct access to it. The address of memory locations is a binary number.

For example, consider a memory which consists of 8 memory locations. These eight locations are having the addresses 000, 001, 010, 011, 100, 101, 110 and 111 (Fig. 6.2).

Memory locations	Addresses
Word 0	000
Word 1	001
Word 2	010
Word 3	011
Word 4	100
Word 5	101
Word 6	110
Word 7	111

Fig. 6.2

Thus, in this case, the address of each location is a three bit binary number. In other words, 3 bits are required to address the memory having 8 unique locations. If a memory has 16 locations, the memory address is a four bit number. The memory address and capacities are shown in Table 6.1.

Table 6.1

Memory Capacity	*No. of bits needed for the address*
1 KB	10 bits (1 KB = 2^{10} bytes)
64 KB	16 bits (64 KB = 2^{16} bytes)
1 MB	20 bits (1MB = 2^{20} bytes)

6.2.2. Volatile and Non-volatile Memory

A memory is said to be volatile if the stored information is destroyed when power goes off. A non-volatile memory is one that retains its contents even after power failure. Most semiconductor memories are volatile whereas most magnetic memories such as disk, drum, ferrite core are non-volatite.

6.2.3. Destructive and Non-destructive Memory

If the contents of a storage location is lost when it is read, the memory is called destructive memory. Thus, to save the contents of the memory, it has to be rewritten after every read operation. In non-destructive memory, the contents are retained even after reading operation. Magnetic core memory is destructive read out type and IC, magnetic drum, disk and tape are non-destructive read out type memory. In such system, data must be stored in a special register so that it may be immediately restored at the addressed location through a write operation.

6.2.4. Access Time, Random and Serial Access Memories

The performance of a memory device is primarily determined by the rate at which information, i.e., one word from the memory is read/written. This is termed as *access time* of the memory. If locations may be accessed in any order and access time is independent of the location being accessed, the memory is termed as **Random Access Memory (RAM)**. Ferrite core and semiconductor memories are usually of this type memories. Where storage locations can be accessed only in certain predetermined sequences are called **Serial Access Memories**. Magnetic-type units and bubble memories employ serial access. RAMs are categorised as static RAMs and dynamic RAMs.

Static RAMs : The term static refers to the ability of a memory cell to store data, as long as power supply continues, regardless of how long it has been since the cell was written into. The contents written into memory cell will remain in that cell indefinitely so long as power is on. Static RAM is made of large number of flip-flops on IC. They have the disadvantages of being costlier and having lower packing density.

Dynamic RAMs: In case of dynamic RAMs even if power continues the data stored in the memory cells will be lost. In dynamic RAM, each bit is stored as charged in a tiny capacitor inside the IC. Large number of capacitors are fabricated on a silicon crystal using advanced techniques. The presence of charge in a capacitor indicates the corresponding bit to be 1 and 0 otherwise. The charges stored on the capacitor will be discharged slowly. Thus, the capacitors have to be periodically (may be a few millisecond) recharged called **refreshing.** In the process of refreshing, the information is read from the cell and written back in the same position. This means D RAM chips need some external refreshing circuits which makes it difficult, but D RAM chips are preferred compared to S RAM chips because of their smaller size.

6.2.5. ROM, PROM, EPROM and EEPROM

There are some random access memories in which data are permanently recorded during fabrication itself. This memory only allows information stored in it to be read and it would not permit any writing or modification. This is referred to as **Read Only Memory**

(ROM). The widest use of ROM is best illustrated by the capabilities it typically provides in microcomputers. All microcomputers have at least one ROM unit that contains a small program referred to as the "bootstrap loader". This program is automatically copied into primary storage when a computer is powered up and being to execute. This program is responsible for reading in a copy of the operating system from a designated secondary storage device.

PROM: In some special types of ROMs, the user can also write with the help of some program using special writing circuits. The user can store permanent programs, data or any kind of information. This type of ROM chip is known as **Programmable Read Only Memory (PROM)** chip. PROMs are once programmable, i.e., the user can write his information in a PROM only once and they cannot be altered. The operations that have been slowly carried out by software can be converted into microprogram fused into a PROM chip. Once they are in hardware form, these tasks can usually be executed in a fraction of time previously required by the software.

EPROM: A PROM in which the facility of erasing the stored contents is available is called **Erasable Programmable Read Only Memory (EPROM),** i.e., EPROM is an erasable PROM. EPROMs are used to store programs which are permanent but need updating. Exposing the chip to ultraviolet light, the stored program can be erased and programmed again to record different information.

EEPROM: It stands for Electronically Erasable and Programmable Read Only Memory. In EEPROM, instead of erasing the entire contents of the chip, its contents may be erased blockwise (each block containing few bytes to few kilobytes) by electrical method and also can be programmed electrically either partly or fully.

6.2.6. Magnetic Core Storage

A magnetic core is a small ring of ferro-magnetic materials, about 2 mm in diameter. Such cores are strung in a wire-like beads. If an electrical current is passed through the wire, a magnetic field is set up around the wire and core becomes magnetised and remains so even on stoppage of the current. If a current in the reverse direction is passed through the wire, the direction of magnetisation changes. Thus, the two states can be used to represent 0 and 1 of a binary system. A large number of cores are arranged in grid patterns of a maze of planes. A pair of perpendicular wires pass through each core for magnetizing

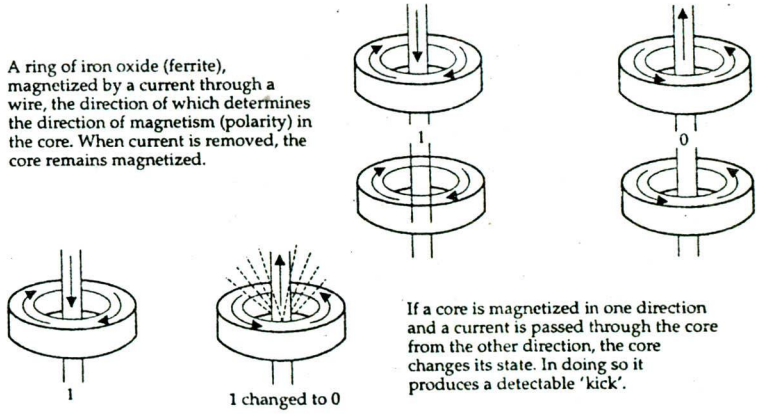


Fig. 6.3

it by an electric current through these wires. Only half the amount of current is needed to magnetize a core at the intersection of the wires. This principle is used to select a particular core in the grid for magnetization to the required state. The state of any core can be detected by using another wire called the *sense* wire. The inhibit wire which also run through the ring prevents a combination of conditions occurring which would de-magnetize the ring altogether.

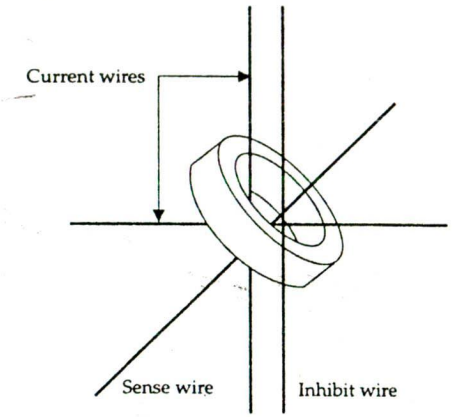


Fig. 6.4 : Single element of ferrite core

If one half-select current $1/2\ I_m$ is applied to line X_1 and one half-select current $1/2\ I_m$ is applied to line Y_1, then the core, which is threaded by both lines X_1 and Y_1, will have a total of $1/2\ I_m + 1/2\ I_m = I_m$ passing through it, and it will switch states. The remaining cores which are threaded by X_1 or Y_1 will each receive only $1/2\ I_m$, and they will therefore not switch states. The designation X_1Y_1 is called the address of the core since it specifies its location which is shown in Fig. 6.5.

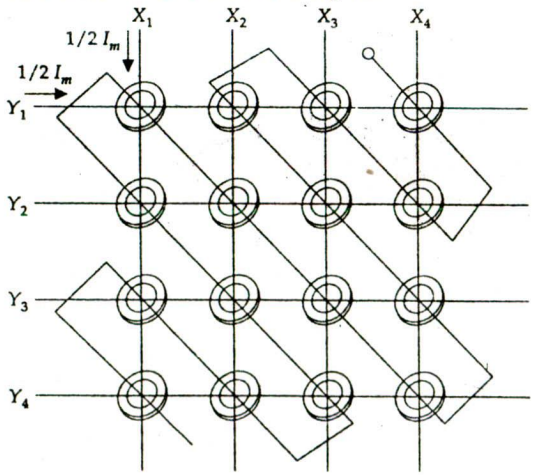


Fig. 6.5 : Magnetic core memory

When electric current or power is switched off, magnetic cores retain their state. Such memories are referred to as non-volatile memories. Core memory has a Destructive Read Operation/Out (DRO). This is so because a read operation magnetizes the cores in the opposite direction. Hence, a read is immediately followed by a 'rewrite' operation in order to restore the contents. The magnetic cores are stacked in parallel planes for the actual storage of data.

Core memories have been popular as they are quite fast, durable and non-volatile but the new storage devices that appeared in 1970s offered even faster performance at a lower cost, and hence the popularity of cores quickly faded.

6.2.7. Semiconductor Storage

Virtually all computers made today have semiconductor elements in their memory. The typical semiconductor memory consists of a rectangular array of memory cells. The basic memory cell is transistor or a circuit capable of storing charge and is used to store 1 bit of information. Metal Oxide Semiconductor (MOS) technology is used for making a computer's main memory chips. Chip is a term used to refer to a semiconductor memory device, having a silicon base. The electronic circuit which consists of a collection of different components like transistors, capacitors, resistors is known as integrated circuit and this is fabricated on a single chip to form a memory device.

The following points summarize the important characteristics of core and semiconductor memories:

1. Core memory has to be assembled manually, whereas MOS memory can be produced mechanically. Hence, MOS memory is cheaper than core memory.
2. MOS memory is available on a miniaturised chip and is therefore much smaller in size than core memory.
3. Core memory has a destructive read operation, while MOS memory has a non-destructive read operation. This enhances the speed of MOS (chip) memory.
4. MOS memory is volatile, whereas core memory is non-volatile.
5. Core memory is available only in large blocks whereas MOS memory is available in wide variety of sizes up to 64 K bits.
6. Core memory has limited range of cycle and access time whereas MOS memory has wide range of cycle and access time and faster than core memory.

Semiconductor memory chips are used in modern computers for low cost and compact size, improved reliability and speed of operation, and increased packing density.

6.2.7.1. Bipolar Semiconductor Technology

Bipolar semiconductor memory is used to provide high-speed buffer storage sections in the CPU. Bipolar memories have two dual emitter bipolar transistors coupled to form a bistable circuit or flip-flop. These chips are faster than even MOS chips but more expensive. These are used as 'Cache' (pronounced as cash) memory, which is a scratch pad to temporarily store 'very active' data and instructions to speed up processing. The cache memory is placed in between the CPU and main memory. The size of cache memory is usually very small (2K bytes — 20K bytes) and has a fast access time (15-40 nanosecond) and is available only on large computers.

Although no one knows what technology will dominate in the future, one is sure that future storage devices will be smaller, faster and cheaper than today's devices.

6.3. SECONDARY (AUXILIARY OR BACK UP) MEMORY

The primary memory is costly, not a permanent storage media and has limited capacity. The secondary memory is the storage other than main memory, and is used for storing large data files, system programs and the like which are not continuously required by the CPU. It also serves as an overflow memory when the capacity of the main memory

is exceeded. This is much larger in capacity but slower than main memory. Information in secondary storage is neither directly accessible to the processing unit, nor to other input-output devices. As such all data must be routed through the main storage.

The secondary storage devices, more commonly in use these days, are as follows:

Magnetic Tape
Magnetic Disk
Floppy Disk
Magnetic Bubble
Optical Disk

They have the following common characteristics : (*i*) Non-volatile storage, (*ii*) Mass storage, (*iii*) Cost efficiency, and (*iv*) Lack of direct processing capability.

Comparison of Primary and Secondary Storages

		Primary Storage	*Secondary Storage*
1.	Cost	Most expensive	Less expensive than primary storage
2.	Capacity	Limited	Nearly limited
3.	Access time	In billionth of a second	In millionth of a second
4.	Processing	Directly accessible to the processing unit	Data must be routed through the primary storage
5.	Media of storing	Semiconductor magnetic core	Magnetic tape Magnetic disk
6.	Location	Within CPU	Outside but connected to CPU

There are two basic types of secondary storage:

1. Sequential access storage: This unit is distinguished by the fact that to read one particular record in the field, all records preceding it must also be read. Media such as magnetic tapes are known as sequential access storage devices.

2. Direct access storage: The media where an individual record can be located and read immediately without reading any other records is known as direct access storage devices. This means that an element of data or instructions can be directly retrieved by selecting and using any of the locations on the storage media. This also means that each storage position (*a*) has a unique address, (*b*) can be individually accessed in approximately the same length of time. Magnetic disk devices are frequently called direct access storage devices (DASD). Magnetic bubble and other devices have a combination of direct access and sequential access properties.

6.3.1. Magnetic Tape

Magnetic tape is a Sequential Access Storage Device (SASD). It is one of the most popular medium for storing very large volume of information. The tape is made up of plastic material (mylar) coated (only one side) with iron oxide magnetizable material. This metallic oxide gets magnetized easily and can retain magnetism permanently. A tape reel may contain 2400 feet of tape or less. The tape vary from 1/2 inch to 1 inch in width and it is divided into horizontal rows called channels or tracks. Most common number of tracks on a tape are 7 or 9. In 9-track magnetic tapes, eight tracks are used for recording data in EBCDIC or ASCII formats and ninth track is used for recording the parity bit for error checking (Fig. 6.6).

In order to read or write information on tape, the tape reel is mounted on a tape drive. A tape drive has following four primary components:

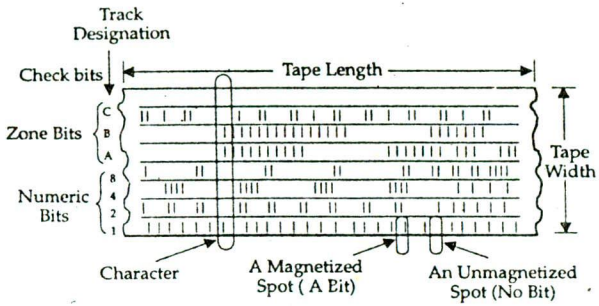


Fig. 6.6 : Data coded on 9-track tape

1. Tape reel holders, one for the supply tape and the other for the take-up spool.
2. A tape drive mechanism.
3. Read, write and erase heads which are physically combined.
4. Tape reservoirs to ensure clean and even tape movement.

A 9-track tape drive system has 9 read/write heads. In a vertical column, all the 9 bits are written/read simultaneously (Fig. 6.7).

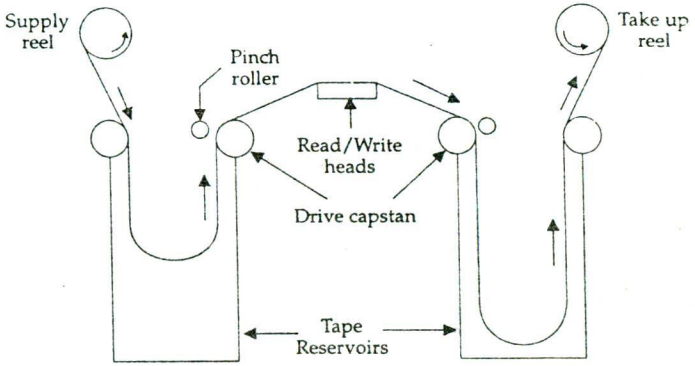
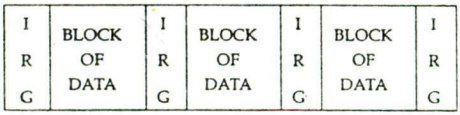


Fig. 6.7 : Schematic diagram of a magnetic tape drive

The recording density, namely, the number of bits per inch (bpi) for a single track is 800, 1600, 3200 or 6250 bits. Since 8 bits (excluding parity bit) are written across the width of the tape in a single recording position, a density of 1600 bpi along 1 inch of magnetic tape provides storage of 1600 characters. A magnetic tape reel of 2400 feet has the potential storage capacity of about 46 million characters which gets reduced about 20 million characters for the requirements of inter block gaps.

Usually, a tape contains a lot of files, while we wish to read or write only a part of a file, not even entire file, at a time. Therefore, a file is usually divided into records. A tape drive will usually read or write one record and stop, move the tape forward or backward to the next desired record, read or write it and stop and so on. The motor used to rotate the reels needs sometime to start or stop the movement of the tape because of physical inertia. Therefore, some blank space must be left between the records. This blank space between records is called the **Inter Record Gap (IRG)**. The length of the IRG usually varies from 0.5 inch to 0.75 inch.

The IRGs occupy more space than the data. To decrease the percentage of tape used by gaps and improve the utilization of the tape, the concept of gathering together many records into a block is used. A group of records is read or written together as a block with no gaps between them, gaps are left only between blocks and are now called **Inter Block Gap (IBG)**. The number of records in a block is called the *blocking factor.*



(*a*) Block with Inter Record Gap

(*b*) Block with three records

Fig. 6.8

A file mark is used to identify the beginning of a file. A file mark is a specially coded record usually preceded by a gap longer than IBG. There is also a similar marker at the end of the usable tape, known as end of reel marker. The record following the file mark is usually known as header label or a file identifier to identify the tape contents and to store other control information. At the end of the tape there is a trailer label record which usually contains the number of records in the file.

Transfer rates (the average number of characters per second between two functional units, a tape drive and a CPU) on different tape drives range from 2,00,000 to 12,50,000 characters per second.

Magnetic tape comes in the form of tape reels and cartridges for mainframes and minicomputers, and small cassettes or cartridges for microcomputers. Mainframe magnetic tape reels and cartridges can hold up to a billion bytes. Small cartridges can store over 100 megabytes.

Advantages of magnetic tape: There are following advantages of magnetic tape usage:

1. It is a relatively inexpensive storage medium.
2. It has a large storage capacity.
3. It is compact and can easily be stored on library racks.
4. Old records can be erased and the tape can be used over and over again.
5. A reel of tape is also convenient way of carrying information from one place to another.

Disadvantages of magnetic tape: In addition to above advantages, there are certain disadvantages, which are as follows:

1. Records can be accessed only serially, which necessitates the reading of previous records until the one desired is reached and, hence, it takes long access time.
2. Updated information cannot be written back on the same location on the same tape. It must be written to a different tape, thus necessitating an additional tape deck.
3. Specks of dust and uncontrolled humidity or temperature levels can cause tape-reading errors. Moreover, tapes and reel containers must be stored in a dust-free environment and labelled.

4. Data transmission is slow compared to disks.
5. Too much operator's time would be required to load and unload tapes.

6.3.2. Magnetic (Hard) Disk

Magnetic disks are the most popular medium for direct-access secondary storage. A magnetic disk is made of metallic film, called Platters, and coated with ferro-magnetic material. Each disk surface is sub-divided into concentric circles called **tracks**. Data are organised into tracks. Each track on the disk has the same total storage capacity, recording density of data is higher on tracks nearer the centre and smaller than those near the outer edge. Each track is further subdivided into sectors, each sector provides a fixed storage capacity. Disks are available in a variety of standard sizes measured by diameter (e.g., 14 inch, 8 inch, 5¼ inch). A hardware device which rotates the disks and facilitates the reading/writing of data is known as a **disk drive.** The disk is mounted on a vertical shaft which rotates at a high and constant speed. An access mechanism moves the read/write head to the desired record to provide direct access. Both surfaces of the disk are available for storage and each surface has a read/write head.

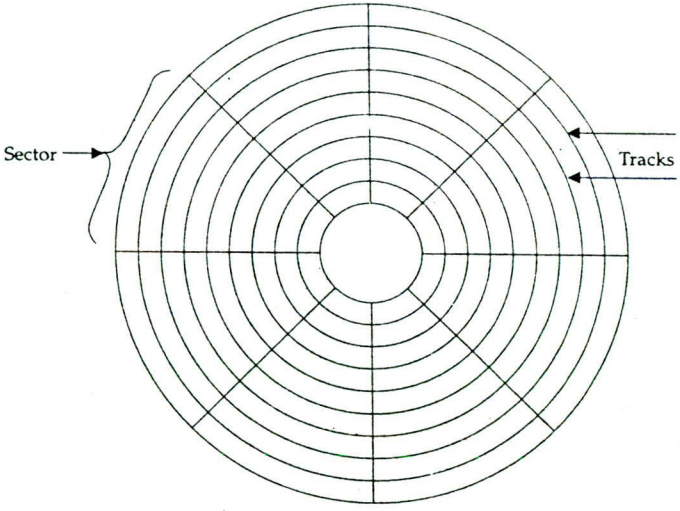


Fig. 6.9 : Magnetic disk

Disk pack: A disk pack is a collection of individual disks packed vertically one over the other. 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 because these surfaces tend to collect dust and other forms of contaminations. Thus, disk pack with 11 individual disks has 20 storage surfaces and each surface has a read/write head. The total number of bytes that can be stored in a disk pack = number of useable surfaces × number of tracks per surface × number of sectors per track × number of bytes per sector.

The disk drive consists of a motor to rotate the disk pack about its axis at a speed of about 3600 revolutions per minute. Thus, all the disks of a disk pack move simultaneously in the same direction and at equal speed. The drive also has a set of magnetic heads mounted on arms. There is enough room in between the spinning disks to allow

access arms with read/write heads to move to any track/sector of any surface. Access arms move in unison. Hence, when access arm is over a particular track on a particular sector, all the other access arms are also over the same track on the remaining surfaces. Consequently, data can be read from or written on multiple tracks simultaneously. This organisation of tracks at one position of the read/write access arms actually resembles a cylinder.

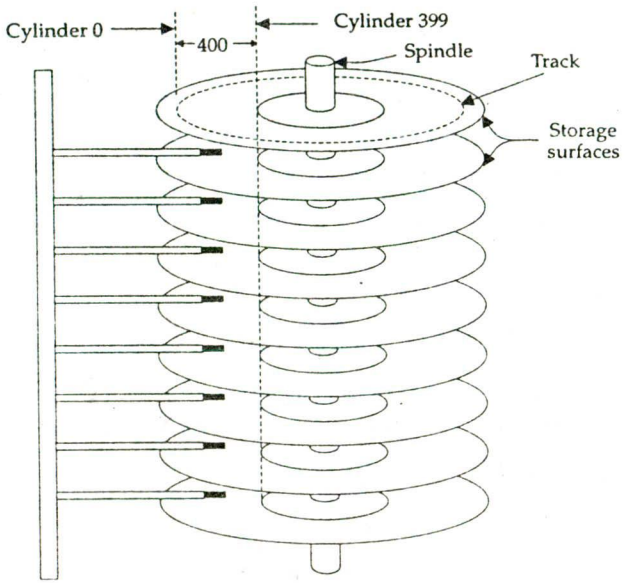


Fig. 6.10

Since a cylinder data can be accessed simultaneously, data are stored cylinder by cylinder when entered sequentially, that is, all the tracks in one cylinder are filled before proceeding to the next cylinder. In order to retrieve a record stored in a device, formatted by cylinder, requires the following steps:

1. A program must identify the physical address of the record on the disk. The address includes the cylinder number, the surface number and the sector number.
2. The access arms then position over the appropriate cylinder (cylinder number and track number are the same in this scheme), the head over the specified surface is activated, and the data are read from the designated sector as it spins under the head. To read data from a disk pack (or to write data on it) requires certain amount of time and is called access time. This is composed of the following components:

Seek time: The time required by the access arm to reach the specified cylinder is known as the *seek time*. The seek time depends upon the position where the arm assembly was at the time read/write command was received by the controller. The disk controller, depending upon the instructions it receives from the computer, positions the read/write heads to the specified location. The common average seek time is of 20-30 millisecond. The maximum seek time is the time taken by the head assembly to reach the innermost cylinder from the outermost cylinder or *vice-versa*.

Rotational delay(latency time): The time required for the rotating disk pack is to bring the correct sector to a position under the read/write heads. This depends on the speed with which the disk is rotating which is usually at 3600 r.p.m. An average rotational delay is of 8-15 millisecond.

The sum of average latency and seek time is known as the average access time.

Transfer rate: It represents the speed with which data can be transferred from the disk to CPU. This transfer rate depends on the speed of rotation, density of the recorded data and the length of the record to be transferred. A peak transfer rate of 1.2 M bytes per second to 2.5 M bytes per second is common. The cylinder method is very effective in cases where a large number of records need to be processed in sequence after the first record is retrieved.

Disk units are classified into two types:

1. Removable disk units
2. Fixed disk storage units.

A removable disk unit can be detached from the drive and transported. It usually has only one read/write head per disk surface. The IBM disk unit 3330 is a removable disk pack.

A fixed disk storage unit is non-removable and placed in a completely sealed system and is termed as **Winchester disk** pack. As the disks and heads are in a sealed unit, no dust or foreign particles can get on the disk surface. Winchester head weighs 0.25 gm and floats above the surface. The surface is lubricated so that light weight head take off and lend on the surface smoothly if power fails. The in-contact, start/stop capability has largely eliminated the head crash problems encountered with ordinary disks.

Most Winchester disk drives use 14", 8", 5¼" or 3½" disks. 14" disk packs are used in mainframe/microcomputer system. The storage capacities vary from about 5 megabytes to very large capacities. It requires less power, generates less heat, provides a faster access and offers significant reliability.

6.3.3. Floppy Disk (Diskettes)

A floppy disk is a popular form of auxiliary storage. The disk is made of very thin plastic material (mylar) and is coated with a layer of magnetic recording oxide and are normally

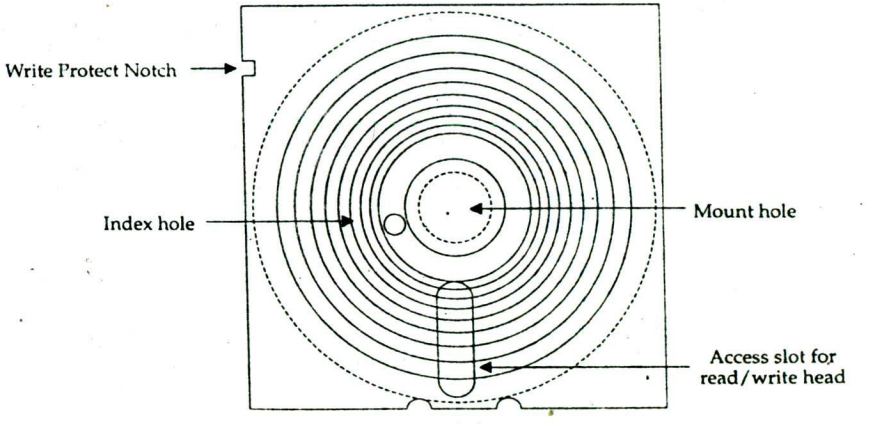


Fig. 6.11 : A diskette

coated on both sides. As the material used is not a hard plate but a flexible tape it is called a floppy disk. The disk is thin and circular and permanently enclosed in a plastic jacket. The jacket is used with diskette to save it from physical damage and dust particles. The disks are available in the sizes of 3½ inch, 5¼ inch and 8 inch in diameter, out of which commonly used disk for small computer system is 5¼ inch. There are four types of diskettes: (*a*) single-sided-single density, (*b*) single-sided-double density, (*c*) double-sided-single density, (*d*) double-sided-double density.

A circular cut at the centre of the disk is meant for the drive spindle which clamps on to the disk through an opening in the plastic jacket. The oval shape cut on the cover is the window for read/write head to contact disk. We cannot write any data on the disk, if this notch is covered, but the disk can be read whether the notch is covered or not. Index hole is provided to mark the position of the first sector in a track. The beginning of the other disk sector is done by either soft sectoring or hard sectoring. In the soft sectoring technique, the disk has only one index hole which mark the beginning of the first sector. The drive then computes the location and position of each subsequent sector by a special program. In the hard sectoring technique, a ring of holes is punched on special track so that each hole marked the beginning of a different sector. Sector's capacity cannot be increased or decreased, but such disks are not used much today. Soft and hard sectored disk can be identified by the number of wholes.

Disk formats: In order that data recorded on the disk may be located or written into an allotted slot, it is necessary to organize the disk into tracks and tracks into sectors. There are two standards which are in common use with 5¼ inch disks having 40 tracks and 80 tracks. The outside track is numbered 0 0 and the inside track is numbered 3 9 (or 79 in the case of the 80 track standard). It is difficult to give precise figure for storage capacity, speed, etc., as this depends on the manufacturer, the size of the floppy disc, whether single or double-sided discs are used, whether double density encoding is used, and so on. However, typical storage capacities are :

(*a*) 5¼" floppy (single-sided)	200 K bytes
(*b*) 5¼" floppy (double-sided and double density)	800 K bytes
(*c*) 8" floppy (double-sided and double density)	2 M bytes

Disk formatting: A floppy cannot be used immediately after its purchase from the market. It has to be formatted before we start recording data on a disk. This is done with the help of a program known as **formatter**. In this process, the concentric magnetic tracks are imposed on the disk and the sectors are marked out and numbered.

A flopply drives are compact units about 18" square by 6" deep with a slot for the user to insert the floppy disk. The floppy disk along with its envelope (jacket) is slipped into the drive mechanism. The mechanism holds the envelope and the flexible disk is rotated inside the envelope by the drive mechanism. Track movement and positioning of read/write head is controlled by a servo-mechanism.

Advantages of floppy disks are:

1. It is relatively cheap and can be reused many times.
2. It is easy to carry around and can be sent from one place to another by mail, or can be exchanged among users.
3. Besides its use as a peripheral memory, the floppy disk is also generally used as a medium for data preparation. Bulk data to be input to a computer is stored in the floppy disk. It is also useful for storing master files, operating systems and application programs, particularly with PC.

Disadvantages include:

1. Floppy disks are to be handled carefully.
2. They are easily damaged by heat, dust, humidity, etc.

6.3.4. Optical Disk

The latest development in secondary storage is the optical disk. The system consists of a rotating disk which is coated with highly reflective material. The data are written by focusing high power laser beam on the surface of the spinning disk in the form of small pits and land (smooth surface). The storage capacity of optical disks is tremendous in comparison to magnetic disks and the storage cost is very low. There are three types of optical disk.

1. CD-ROM (Compact Disk-Read Only Memory): CD-ROMs are manufactured like gramophone records. Each disk contains 16,000 tracks per inch, and each is approximately 4½ inch in diameter. First, a master disk is prepared. By moulding a special plastic to the master disk, CD-ROMs are produced. The disk is written once only during manufacture in the form of small pits and lands. Once written, it cannot be erased. In order to read, a low power laser beam is focused on the disk surface. Smooth surface reflects more light and pitted area reflects less light. The reflected light is sensed by a detector. The variation in light is converted to electrical signals. The major limitation of the disks is that they are read-only devices. A disk normally can hold over 6000 MB of data.

2. WORM (Write Once Read Many) disk: The optical disk which can be written once by the user is called WORM. Once the data are stored on the disk, it can never be erased or changed without physically destroying the disk. This makes WORM storage ideal for backup and archiving applications that require files never be altered. WORM disks do not require the special mastering procedures of CD-ROMs. Data from video scanners, keyboards, optical character recognition equipment, and other devices can be recorded on WORM disk. Access time of WORM disk is greater than magnetic disk but less than microfilm. Normally, 12-inch, 8-inch, or 5¼ inch WORM disks are stored in jukeboxlike systems. A jukebox is a rack system that allows several hundred optical platters to be mounted into a rack that is connected to a central controller for access and reading. The result is hundreds of giga-bytes, even terabytes (TB), on online storage within a single system.

3. Erasable optical disk : This uses both laser and magnetic head to read and write the data. The data stored on the disk can be erased and rewritten. Most erasable disks are recorded using lasers to heat magnetized areas coated with various metals. The magnetism provides polarity in the sections, which can then be read with another laser. Data are erased by shooting an even more powerful laser at the disk, which reverses the magnetism. These disks may be used as alternatives to traditional magnetic disks or for backup.

WORM, CD-ROM and erasable optical disks provide vast expenses of storage space that leave hard disks and cartridge tapes far behind. Although all three types of optical drives use a low-power laser to read data from a spinning removable disk, they differ in other technological ways, and each is best suited for different applications.

Table 6.2 : Comparison of Important Secondary Storage Methods

Peripheral Equipment	*Storage Media*	*Primary Functions*	*Speed and Capacity*	*Major Advantages and/or Disadvantages*
Magnetic disk drive	Magnetic disk Disk pack Fixed Disk	Secondary storage (direct access) and input/output	Access time: 10-100 millisecond Data transfer: 200,000 to 5 million bytes per second Capacity: From 10 million to 15 billion bytes per drive	Large capacity, fast direct access storage device (DASD), but relatively expensive

contd.

Floppy disk drive	Magnetic diskette 8,5¼" and 3½" diameters	Secondary storage (direct access) and input/output	Access time: 100-600 millisecond Data transfer: 10,000-30,000 bytes per second Capacity: From 360,000 to several million bytes per disk	Small, inexpensive, and convenient, but slower and smaller capacity than other DASDs
Magnetic tape drive	Magnetic tape reel	Secondary storage (sequential access), input/output, and disk backup	Data transfer: 15,000 to 2 million bytes per second Capacity: Up to one billion bytes per tape reel	Inexpensive, fast transfer rate, only sequential access
Optical disk drive	Optical disk: CD-ROM, WORM, and erasable	Secondary storage (direct access) and archival storage	Access time: 30-200 millisecond Data transfer: 150,000-500,000 bytes per second Capacity: CD-ROM: up to 700 million; WORM: up to 3 billion; erasable: up to one billion bytes	Large capacity, high quality storage of data, text, and images. Primarily a read-only medium

6.3.5. Magnetic Bubble

These storages were introduced in the late 1970s and so far they have not gained much popularity like RAM disks. Magnetic bubble technology lies somewhere between disk and tape technology on the one hand and semiconductor memory on the other. Magnetic bubble storage devices are solid state, i.e., they do not have any moving part, and are not volatile.

The technique depends on the use of a layer of a material in which the preferred direction of the magnetic field is perpendicular to the surface of the layer. Magnetic bubbles are formed by applying magnetic fields to thin sheets of certain magnetic materials, such as garnet crystal. The magnetic fields strengthen some regions in the material and weaken others. The strengthened regions break into isolated cylinders that resemble small positively charged islands surrounded by a sea of negative charges (Fig. 6.12). Data are represented in bubble storage by the presence or absence of bubbles which corresponds to 1 or 0 in the binary code.

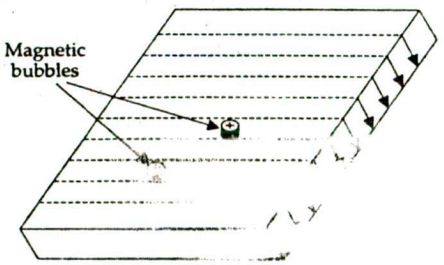


Fig. 6.12 : Formation of magnetic bubbles

Magnetic bubble memories are used as main memory in several microprocessor applications. They are also used as a low-cost alternative to magnetic disks. Bubble chips

are also used in telephone systems that redirect improperly dialed numbers, machine tools and robots.

The bubbles are made to move about in the medium by applying a separate magnetic field in its plane. A pattern of metal shapes is laid down on the surface. These shapes locally modify the inplane magnetic field and produce preferred locations for the bubbles. With a suitable pattern, repeated many times in a straight line, it is possible to make bubbles move along the line of the pattern by making the inplane field rotate. Each computer rotation of the field will make each bubble to the pattern in such a way that bubbles represent the 'one' bits in a stream of data with gaps in the line of bubbles representing 'zero' bits. We can rotate the field until the whole stream is represented by a pattern of bubbles spread out along the line of shapes. When we want to read the data, we start up the rotating field again and observe the bubbles as they reach the other end of the pattern.

6.4. A NOTE ON STORAGE HIERARCHY (AMIE, W '95, W '97)

There is a trade-off among the three key characteristics of storage devices: Cost, capacity and access time. At any given time, a variety of technologies are used to implement storage systems. Across this spectrum of technologies, the following relationships hold:

- Smaller access time, greater cost per bit
- Greater capacity, smaller cost per bit
- Greater capacity greater access time.

A cost effective technique for the design of large computer systems is the use of hierarchy of memory technologies. A typical storage hierarchy ladder is shown in Fig. 6.13.

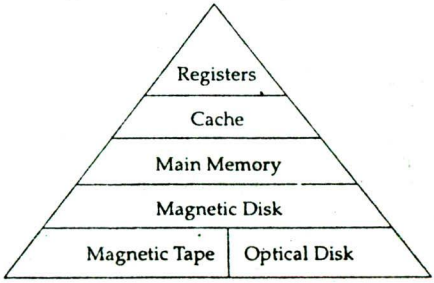


Fig. 6.13 : A typical storage hierarchy ladder

As one goes down the hierarchy, the following conditions occur:

- Decreasing cost per bit
- Increasing capacity
- Increasing access time.

This smaller, more expensive, faster memories are supplemented by larger, cheaper, slower memories.

6.5. REGISTER SECTION (AMIE, W '97)

In order to carry out the various functions by CPU, it has built into it certain basic registers. Registers are the memory of the computer which remain in the CPU itself. A register is a group of binary cells. Since a cell stores one bit of information, it follows that a register with n cells store any discrete quantity of information that contain n bits. Size of the register is related to the word length of the computer. Word length is the number of bits which CPU handles at any instant.

The registers can be of general purpose or special purpose (dedicated). General purpose registers can be used for several functions under program control. They are convenient store for holding data which is currently processed by the ALU, for storing some data temporarily and also for storing the intermediate results. Special purpose registers are dedicated for certain functions only and are generally not under program control.

Although the number of registers varies from computer to computer, there are some registers that are common to all computers. A brief description of these registers are given below:

Accumulator: This register holds the initial data to be operated upon, the intermediate results, and also the final results of processing operations. It is used during the execution of most instructions. The results of arithmetic operations are returned to the accumulator register for transfer to main storage through the memory buffer register. In many computers there are more than one accumulator registers.

Program Counter (PC): It holds the address of the next instruction to be fetched from the memory. Program counter is updated each time it fetches an instruction. It is assumed that the instructions are stored in consecutive memory locations and read and executed in sequence unless a branch instruction is encountered. For a branch instruction, the address part of the branch instruction is transferred to the PC register to become the address of the next instruction.

Memory Address Register (MAR): This register holds the address of the location when an instruction is read out or written in the memory. The address is loaded from PC.

Memory Buffer Register (MBR): The register acts as a buffer between the CPU and the memory. It holds the contents of the memory word read from or written in memory. An instruction word placed in this register is transferred to the IR. A data word placed in this register is accessible for operation with the accumulator register or for transfer to I/O register.

Instruction Register (IR): This is a register into which the instruction that has been fetched from the memory is temporarily stored. It is subsequently decoded and interpreted for the actions to be performed.

Input/Output Register: This register is used to communicate with the input/output devices. All input information such as instructions and data are transferred to this register by an input device. Similarly, all output information to be transferred to an output device are found in this register.

Example 6.1. *The length of a magnetic tape is 3600 ft. Each record has 50 characters and the tape carries 50,000 characters. The tape speed is 100/sec and the blocking factor is 5. How many blocks are there on the tape?*

Solution. Total no. of characters = 50,000

Each record has 50 characters.

Total no. of records = 50,000/50 = 1000

Blocking factor = 5

Total no. of blocks = 1000/5 = 200

Example 6.2. *The length of a magnetic tape is 2400 ft with storage capacity of 1600 bpi with block size of 1600 bytes and inter-record gap of 0.6 inch. Find the amount of data in bytes that can be stored in this tape.*

Solution. Since the storage capacity is 1600 bytes per inch and the length of block = 1" of tape.

To store one block or 1600 bytes of data = Inter record gap + 1" length of tape

= 0.6" + 1" = 1.6"

Hence, 1.6" length of tape can store 1600 bytes.

Hence, 2400 ft length of tape can store $= \frac{1600}{1.6} \times 2400 \times 12$

$= 28800000$ bytes

$= 28.8 \times 10^6$ bytes.

Example 6.3. *A disk pack has 11 plates with 20 read/write heads which can take 400 positions across the disk surface. Each track on a disk surface is divided into 100 sectors of 512 bytes each. Determine*

(a) Number of cylinders in the disk pack

(b) Number of tracks in a cylinder

(c) The volume of data that can be accessed at and the position of the read/write head

(d) The total storage capacity of the disk pack.

Solution. (a) Total no. of tracks on each surface = Number of positions that the read/write head can take = 400

Number of cylinders = Number of tracks on each surface

= 400

(b) Number of tracks in a cylinder = Total number of used surfaces on the disk pack

= 2 × (no. of plates) − 2

= 2 × 11 − 2 = 20

(c) Volume of data at any position of read/write head

= (No. of sectors in each track) × (No. of bytes per sector) × (No. of surfaces used for storing data)

= 100 × 512 × 20 = 1024 × 100 bytes = 100 K bytes

(d) Total storage capacity = No. of tracks × capacity per position of head

= 400 × 100 K bytes

= 40000 K bytes

Example 6.4. *Find the data transfer rate (in bytes/sec) and average access time (in millisecond) for a disk pack with the following details :*

Disk pack capacity	= 200 M bytes
Storage surfaces	= 19
No. of tracks/surfaces	= 700
Drive speed	= 3600 rpm
Average seek time	= 25 ms

Solution. Average access time = Average latency time + Average seek time

Now, average latency time = time to reach the specified sector

$= \frac{1}{2}$ (maximum time taken in one revolution of the disk pack)

$= \frac{1}{2} \times \frac{1}{3600}$ min $= \frac{1}{2} \times \frac{1}{60}$ sec $= \frac{1}{120}$ sec

Average seek time = 25 ms

$= 25 \times 10^{-3}$ sec $= \frac{25}{1000} = \frac{1}{40}$ sec

Hence, average access time $= \frac{1}{120} + \frac{1}{40} = \frac{1}{30}$ sec

$$\text{Data transferred in one second} = \frac{\text{Data stored in one track}}{\text{Time taken to transfer data from one track}}$$

Disk pack capacity = 200 M byte
= 200 × 1024 K bytes
No. of cylinders = Number of tracks per surface = 700
No. of tracks/cylinders = No. of used surfaces of the disk pack
= 19 – 2 = 17

$$\text{Data stored/track} = \frac{200 \times 1024}{700 \times 17} = 17.21 \text{ K bytes}$$

$$\text{Hence, data transferred in bytes/sec} = \frac{17.21}{1/30} = 516 \text{ K bytes/sec.}$$

REVIEW QUESTIONS SET

1. Distinguish between the following terms: BIT, BYTE, NIBBLE, WORD and WORD LENGTH, seek time and latency time. **(AMIE, W '97)**
2. What do you understand by storage hierarchy?
3. Write short notes on the following terms:
   (*a*) Main memory, (*b*) Secondary memory, (*c*) Virtual memory, (*d*) Cache memory.
4. How does one measures the size of a computer memory ?
5. What are advantages of random access memory over sequential access memory?
6. What is semiconductor? Why are semiconductors used for primary memory and not for secondary storage. Summarize the characteristics of core and semiconductor memories.
7. How many address bits will be required to address 16 MB of main memory to a computer.
8. Define (*a*) storage location in memory, (*b*) address of a storage location.
9. Discuss the advantages and disadvantages of (*a*) semiconductor memory over magnetic core memory, (*b*) magnetic tape. **(AMIE, W '97)**
10. Explain the following:
    (*a*) ROM, (*b*) PROM, and (*c*) EPROM.
11. What are static and dynamic RAMs? Discuss their merits, demerits and areas of applications?
12. Draw the block diagram of a 8 bit, 8-word RAM system and explain briefly its operations. **(AMIE, W '95)**
13. What is a magnetic bubble? How are magnetic bubbles formed? Name some of the applications of magnetic bubble memories.
14. What is a memory hierarchy? Name the general classes of storage media that might make up a memory hierarchy.
15. A disk pack has 20 recording surfaces and 400 cylinders. If each track can store 4096 bytes of data, how much data can be stored on 8 such packs. **(Ans. 26214400 bytes)**
16. A floppy disk has 80 tracks, each track is subdivided into 9 sectors. Each sector can store 512 bytes. What is the capacity of the disk in K bytes? **(Ans. 360 K bytes) (AMIE, S '96)**
17. A particular disk drive has one movable head for each of the ten recording surfaces. Each surface has 200 tracks. Each track has the same capacity, namely, 8 blocks of 128 words, each of 24 bits.

    The disks rotate at a speed of 3000 revolutions per minute and the speed with which the head can move averages out at one track per millisecond. Calculate:
    (*a*) The total capacity, in words, of the disk pack, and
    (*b*) The data rate, in bits per second, achieved during the transfer of a block.
    [Ans. (*a*) 2,048,000 words (*b*) 12,28,800 bits per sec (AMIE, W '97)]
18. What are the functions of the following registers in the CPU?
    (*a*) Program Counter (*b*) Instruction Register
    (*c*) Accumulator (d) Memory address register
    (*e*) Memory buffer register. **(AMIE, W '96)**
19. Differentiate between magnetic tape and magnetic disc. **(AMIE, W '97)**
20. A disk pack has 8 surfaces. There are 256 tracks on each surface. If cash track has 80 sectors and each sector is having 400 bytes of capacity find capacity of hard disk.
    (Ans. 65536000) (AMIE, W '98)

7

INPUT/OUTPUT UNITS

7.0. INTRODUCTION

The input and output units of a computer are normally abbreviated as I/O units. Computer system use input devices for data entry purpose to enable a user to communicate with the machine. The output unit transforms the result of the processed data into a form which can be read by people. The input and output devices together are commonly known as peripheral devices. Some devices can only input data into computer like keyboard and some devices can only do output operation like printer. But there are some devices which can do both input and output of data like magnetic tape and magnetic disk. In this chapter, common input/output devices used in computer system are discussed.

7.1. INPUT UNITS

There are a variety of input units which are used by computer. Input devices can broadly be categorized as follows:

(*a*) Paper media
(*b*) Magnetic media
(*c*) Magnetic Ink Character Reader (MICR)
(*d*) Optical media
(*e*) Direct entry devices.

7.1.1. Paper Media

Punched cards: The oldest and most commonly used storage medium has been cards. There are two types of punched cards. One has 80 columns and other has 96 columns.

80 column card: It is 18.8 cm in length, 8.3 cm in width and 0.018 cm in thickness. The card is divided from left to right into 80 columns numbered 1 to 80 and into 12 rows numbered 9 to 0 from bottom to top and 11 and 12 are above row 0. The top three rows 0, 11, 12 are called zone rows and bottom ten rows (0 to 9) are called punch rows. 0 code row is both a zone punch and digit punch row. A digit is represented by a single hole punched in its corresponding rows from 0 to 9. Alphabetic and special character is represented by two holes, one punched in a zone row and one punched in a digit row. The coding system used to represent data in 80 column cards is known as Hollerith code after the name of Dr. Herman Hollerith, who first used punched cards to handle the U.S. census data in 1889.

Each column can represent one character and a maximum of 80 characters can be represented in a single card. Desired data are recorded on the punch card by punching rectangular holes in the columns with the help of a key punch machine which is similar to a typewriter. The characters so represented are also printed at the top of the card in their respective column to verify the correctness of the card puncher.

96 column card: In the late 1960s IBM introduced its System/3, a computer system

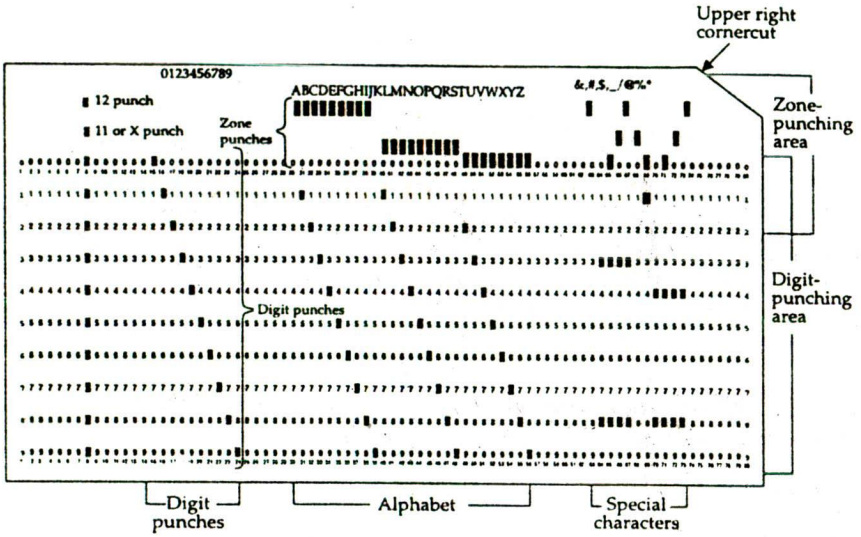


Fig. 7.1 : A 80-column punched card

designed to provide high-speed processing capabilities for the small computer user. Accompanying the System/3 was a newly designed 96 column punch card. This new punched card differs from the 80 column punch card in following areas:

1. It is smaller in size and is square in shape.
2. It is subdivided horizontally into three punching areas or tiers, each running

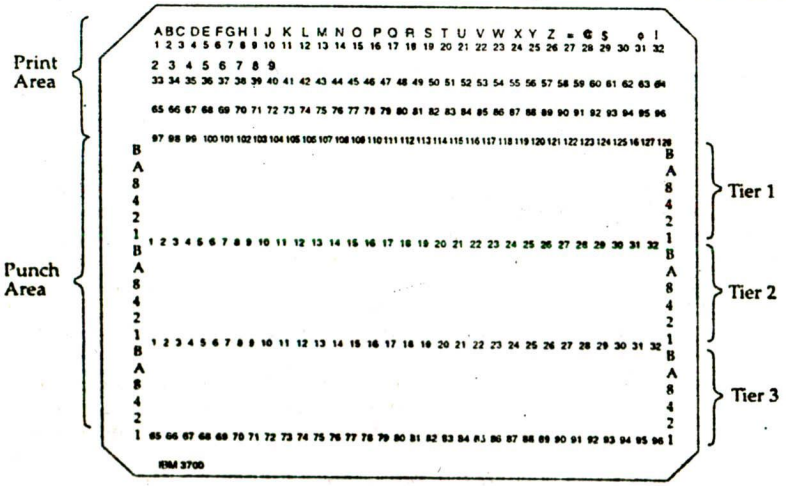


Fig. 7.2 : A 96-column punched card

the length of the card and capable of recording 32 characters of information.
3. Punched holes are round, as opposed to rectangular.
4. There are three rows at the top of the card for printing, one for each tier or punching area.
5. The coding system employed is no longer the Hollerith code but a code based on the BCD code.

The cards are read by a unit called card reader. Card readers are electromechanical devices. Cards are loaded into a card reader by putting them in an input hopper. They are then extracted and passed through a reading mechanism and deposited in an output stacker. The read station usually contains light sensitive cells which recognise the pattern of holes in the cards and passes corresponding signals to the computer. The card reader speeds vary from 300 to 2000 cards per minute.

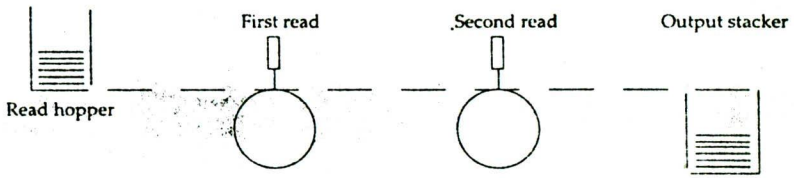


Fig. 7.3 : Mechanism of a card reader

Advantages of Card

- Cards are less expensive than the other media.
- In case of card deck, insertion, deletion or modification is performed by removing individual cards. This is not easy in case of magnetic tape, disk or a punched paper tape.
- Cards are manually readable.
- They can be prepared off-line.

Disadvantages of Card

- Cards are not reusable.
- Because of low density (it refers to the number of characters that can be stored in a given physical space) card files are bulky. Hence, cards are not suitable for the storage of large volume of data.
- Cards are easily damageable.
- Cards depend upon slow mechanical devices for reading and punching.

Punched paper tape: The tape consists of a long roll of paper about one inch width. Characters are recorded on the paper tape by punching holes (round) across its width. There may be 5 or 8 vertical punches in a column to represent a character. The former is referred to as a 5-channel (track) tape while the latter is 8-channel tape.

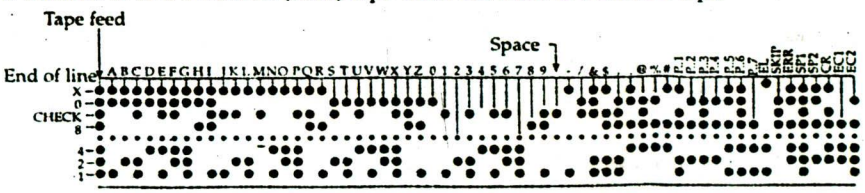


Fig. 7.4 : 8-channel tape

Paper tape reader reads the information punched into a paper tape and converts the coded information into electrical signals that the CPU can accept. As the tape passes from the sensing station, the holes are sensed by photoelectric cells. Typical speeds of paper tape readers range from 1000 to 2000 characters per second.

Although paper tapes are cheaper than cards and less bulky, but they are rarely used today. The major disadvantages are: (*a*) punched paper tape tends to tear easily; (*b*) the data stored on them is held sequential and, hence, amendments are slow and inconvenient; and (*c*) the data reading from them is a very slow process compared to the internal speeds of computers.

7.1.2. Magnetic Media

The disadvantages associated with magnetic media have led to the development of several off-line devices for direct recording. Those in wide use are:

(*a*) Key-to-tape system,

(*b*) Key-to-disk and key-to-diskette system.

Key-to-tape system: A key-to-tape device consists of a keyboard, a buffer store, and a magnetic tape unit. It is used as follows:

1. Reading from a source document, an operator keys a block of data into the unit's memory. Any conscious errors can be corrected by backspacing and re-keying.
2. When the operator is satisfied with a block of data, it is written to the magnetic tape.
3. Data can be verified on the same machine by re-winding the tape, re-reading its data into the memory, and keying it again. Discrepancies are signalled and the operator sorts out the errors in order to ensure that the correct data are written to the tape.
4. Verified data can then be input to the computer from the magnetic tape.

It can be seen that there are several advantages to be gained from using key-to-tape equipment:

1. Encoding data onto magnetic tape is fast because little mechanical movement of the medium is required.
2. The process is quiet; this must inprove operator's productivity in the long run.
3. Error correction is simple and quick.
4. Verification can be carried out using the same machine.
5. Magnetic tape can be read into the computer quickly.
6. As magnetic tape is a re-usable medium, its long-term cost will be low.
7. Where transactions need to be kept in machine-sensible form for security purposes, magnetic tape takes up little storage space.
8. Special input devices are not necessary; the computer's existing tape units can be used.

Because of the problems of loading magnetic tape into an encoding unit, key-to-cassette systems have been developed. These are the same in principle as key-to-tape system, but use a shorter reel of tape in a plastic housing; loading and handling is, therefore, much simpler. Before input to the computer, the data from several cassettes is combined onto one reel of magnetic tape.

Key-to-disk and key-to-diskette system: Key-to-disk and key-to-diskette system have a disk drive and a diskette drive, respectively, along with a small processor, keyboard and visual display unit.

The diskette on which information is to be recorded is inserted in the specified slot of the machine. The correct position of the diskette in the slot is indicated by a 'click' sound. When the machine is switched on, the disk is rotated at a constant speed of about 360 revolutions per minute by a motor.

A data entry operator records data from the source document on the diskette by depressing appropriate keys held temporarily in the buffer memory in the coded form till all the 80 characters are entered. The operator can view the data on the display unit and correct it by re-entering the correct data which overwrites the incorrect data in buffer memory. The operator presses a control key when the data is ready to be entered on the diskette. Then data are recorded on the diskette along a circular track as a series of magnetized spots. When a track is full, the recording head automatically moves to the next concentric track. Chapter 6 provides more details about the diskette.

7.1.3. Optical Media

Optical devices eliminate manual transcription; hence they are sometimes referred to as source data automation. The devices under this category are:

(*a*) Optical Mark Reader (OMR)
(*b*) Optical Character Reader (OCR)
(*c*) Special Terminals like Point-of-Sale terminal.

Optical Mark Reader (OMR): It is an input device which is capable of recognising a prescribed type of mark made by pencil or pen. Special reprinted forms are designed with boxes which can be marked with a dark pencil or ink. Each box is annotated distinctly so that the user clearly understands what response he is making. The recognition of marks involves focusing a light on the page being scanned and detecting reflected light pattern from the marks. The document reader then transcribes the marks into electrical pulse which are transmitted into the computer. This type of scanner is used for evaluating the objective type answer papers in which a candidate is advised to put a mark with soft pencil at a specified spot. The advantage of this method is that information is entered at its source and no further transcription is required and, hence, increases reliability of data. The disadvantages are that it needs accurate alignment of printing on forms and the need for good quality expensive paper. The form cannot be redesigned frequently because reprinting of forms will be expensive.

Optical Character Recognition (OCR): These input devices are capable to read different shapes of mark and complete set of alphanumeric characters. In the operation of the OCR reader, a mechanical drum is used to rotate documents past an optical scanning station. A light source and lens system can distinguish the patterns of the character. These patterns are converted into electrical pulses and are compared with the stored pattern for all possible characters until an exact match is found, thus identifying the character. When the generated signal does not match with any of the characters that have been stored, the inputed character is rejected. Today, however, optical readers are manufactured which can recognise handwritten characters, with a rejection rate of less than one per cent of the total number of documents scanned. This method of input is ideally suited to documents which can be used as turnaround documents. Gas and electricity bills are good examples. The bills are printed with all the information necessary for re-input to the system in an OCR font. If the customer pays the amount stated on the bill, then the portion of the bill with the OCR data on it can be returned for direct input to the system.

Since optical readers read the source documents directly, they eliminate the bottleneck of having a person transfer the data from the source document onto a computer input medium. Input document is readable by both machine and human being. OCR also has following disadvantages:

- Printing for OCR must meet high standards, and this is expensive.
- OCR is economical only when a large number of documents are to processed.
- Only certain types of printed or handwritten characters are recognised.

Bar-Code Readers-Point-of-Sale (POS): This method uses a number of bars (lines of varying thickness and spacing between them to indicate the desired information. Bar codes are used particularly by the retail trade for labelling goods and by supermarkets for labelling shelves and in stock control. They are also used for numbering books in public libraries so that when a book is borrowed/returned, it can be recorded using a computer. An optical-bar reader can read such bars and convert them into electrical pulses to be processed by a computer.

The most widely known bar code, known as the universal product code (UPC), consists of a block of vertical lines that vary in width. These bars are detected as ten digits. The first five digits are assigned by a central agency to identify the manufacturers and the next five digits are assigned by the manufacturer himself to each of the items in his product line. The digit 0 on the left of the bar indicates that the product is a grocery item. Similarly, the digit 3 stands for drugs and digit 9 for books.

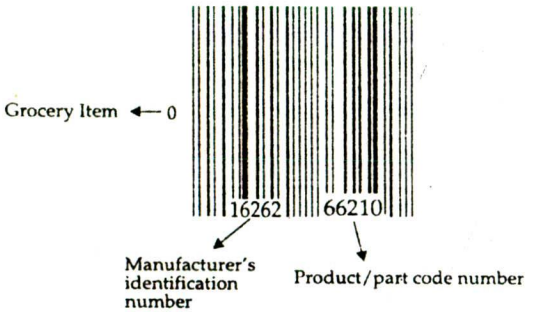


Fig. 7.5 : An example of an optical bar code

A system which is used to capture data concerning sales in a retail establishment, such as a supermarket, is known as point-of-sale (POS) system.

The point-of-sale terminal often contains an optical-bar reader. It is an on-line terminal connected to a computer for processing. As each product passes the scanner, the POS terminal identifies the product, determines the current price and prepares the listing of the purchases, showing the item numbers, their description and prices. This information is usually displayed on a lighted panel. In addition, the device also adds salex tax, etc., where applicable, and prints a cash receipt. The terminal then transmits the data to the central computer where it is used for inventory control, sales analysis and allied purposes.

Some advantages of POS data collection are as follows :

1. It eliminates the preparation of an intermediary data medium, e.g., a floppy disk.
2. It eliminates data transcription errors and saves money.
3. Processing errors caused by illegible sales slips may be reduced.
4. The inventory disbursements data required for inventory control are collected as a natural part of the sales transaction.

7.1.4. Magnetic Ink Character Reader (MICR)

MICR is actually a combination of magnetic and paper media. It is widely used by banks to process tremendous volume of cheques being written each day. In these systems, characters are printed using special ink which can be magnetized so that, after being subjected to a magnetic field, they can be read and decoded. The first major use of MICR was in

the banks in the United States. The reading station is used to sense and identify the magnetic characters as they pass through. When the characters enter the reading station they are magnetized by the write head. As the magnetized characters pass the reading head, they induce electrical signals in it to identify the characters. Because each character has a rigidly defined shape, the signals produced in the read head are unique and can be easily coded into a form suitable for transmission to the computer. The speed of MICR range from 750-2500 documents per minute (approximately 15,000 to 50,000 characters per minute). Information can be read directly into the computer's memory.

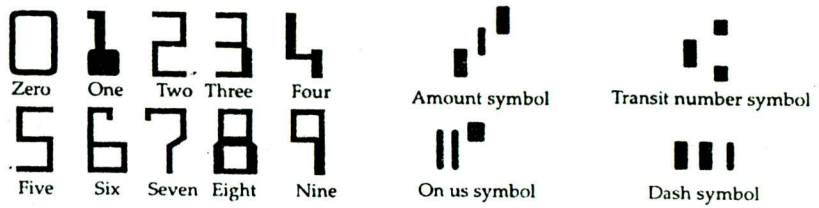


Fig. 7.6: Standard symbols of the MICR code

In banks, which use the MICRs, the cheque number, the number identifying the bank, and the customer's account number are preprinted on the bottom of the cheque in magnetic ink. Once a customer presents a cheque, it can be read using a special input unit which can recognize magnetic ink characters. This method eliminates the need to manually enter data from cheques and record it on a floppy disk.

MICR has following advantages:

1. The characters are easily recognizable to the human being.
2. There is high accuracy in reading the characters by the input device.
3. The input devices are reasonable in cost.
4. MICR coding is flexible since documents of varying sizes, thicknesses, and widths can be used without hampering the processing capability of the reading or sorting equipment.

A major disadvantage is that the number of characters are limited to 10 digits and 4 special characters. Hence, the system is unsuitable for general purpose data processing. No alphabetic characters are available.

7.1.5. Direct Data Entry Devices

On-line devices are the means of direct and interactive communication between the users and the computers.

Visual Display Terminals (VDT)

A video display terminal or a video unit consists of a keyboard and a display device in the form of cathode ray tube (CRT). Keyboard is used to enter commands and data to computer. As each character is typed on keyboard, it is displayed on CRT screen. A cursor (a small arrow, underline or a small square which can be moved horizontally or vertically to indicate the position of a character) moves on the screen with the control of keys. A letter is printed at the position where the cursor is located on the screen. After one line is completed, all the lines will move up by one line and top line will disappear. Normally a VDT has capacity of 24 lines, each line accommodating 80 characters. A VDT can be used both as on line and off-line terminal.

The terminals can be classified as under:

(*a*) Non-intelligent (dumb) terminals;

(*b*) Smart terminals;
(*c*) Intelligent terminals.

Dumb terminals: These terminals have a keyboard for input, a means of communication with the CPU, a printer or a screen to receive input. They usually have neither processing capability nor storage capacity. They merely send and receive; all processing is performed by the CPU to which they are connected.

Smart terminals: Smart terminals contain a microprocessor and internal storage. Special function keys are incorporated in the keyboard. These terminals have data editing capability and the ability to store input data prior to sending it to (or receiving it from) the CPU. These terminals are non-programmable by users.

Intelligent terminals: These terminals consist of a microprocessor chip that are user programmable. Small tasks can be performed by intelligent terminals without taking any help from larger CPU. Floppy disks are provided for storage while processing small jobs. These terminals have ability to guard against input errors, to check data for logical consistency and help users make printed copies or floppy disk backups of computer input. Most modern on-line workstations today use terminals with at least some intelligence.

7.1.6. Pointing Devices

Each of the following devices permits the user to select something on CRT screen by pointing to it. Therefore, these devices are called pointing devices.

Mouse

The mouse is a pointing input device that is used with Video Display Terminal (VDT) and Personal Computer system. It is a small box with a round ball on the bottom and one or more buttons on the top. The mouse is attached to a terminal or microcomputer by a cable. It is held in one hand and moved across a flat surface.

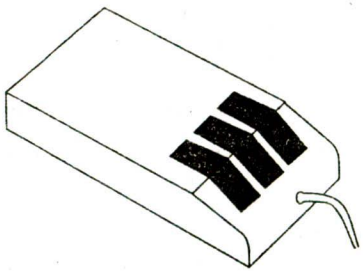


Fig. 7.7: Mouse

A mouse enables the user to manipulate the cursor on the screen. When a user rolls the mouse across a flat surface, such as a desk, the screen cursor moves in the direction of the mouse's movement. If the user rolls the mouse forward and to the right, the cursor moves up and to the right on the screen. Such movements enable user to point to menu of his choice on the screen. With a click of the mouse's button, the user communicates his choice to the computer. With proper software, a mouse can also be used to draw pictures on the screen and edit test. Mouse is very popular in the medium computer which use windows and other graphical user interface applications.

Light Pens

A light pen is a pointing device used in conjunction with a VDU and is basically just a single photocell at the end of a pen-like want (replacing the nib, as it were). The user

presses his light pen against the screen of the VDU and when he is sure that he has the position he wants, he presses a button on the pen, or presses on the pen tip, causing a switch to sense the pressure. This switch signals to the computer that the position of the pen on the screen must now be worked out. The computer first passes a horizontal band of light from the top of the screen to the bottom. Somewhere in its travel it will cross the photocell at the top of the light pen and a signal will pass to the computer indicating that the photocell has seen the band of light. The time delay between the computer starting the band and the detection of light by the photocell indicates to the computer how far down the screen the pen is placed. Similarly, a vertical band passed from left to right enables the computer to detect how far across the screen the light pen is placed in this way, the computer can determine the coordinates of the place at which the pen is pointing.

The light pen enables the user to select options from a menu. The user indicates his choice by touching light pen against a desired option of the menu.

The light pen is also useful for graphics work. The users at a computer aided design (CAD) terminal can draw directly on the screen with the pen. The user can select different colours and line thickness, and can reduce or enlarge drawings and can add or erase lines.

Voice Input Systems

Voice data entry terminals allow the direct entry of data into a computer system by verba communication from a human operator. A typical configuration consists of one or mor portable voice recognition units, microphones and a CRT terminal for visual displa of spoken input. The user speaks into microphone that is attached to a digitizer— device that converts the analog sound of the user's voice to digital data, storable i computer memory. Most voice input systems require training the computer to reco nize a limited vocabulary of standard words for each systems user. A word is reco nised only when a choice match is found. A voice recognition system can be used i work situations where both hands of workers are engaged in the job and he needs perform data entry into the computer. At the present time, the majority of voice inp systems are used with microcomputers. Speaker-independent voice recognition system which allow a computer to understand a voice it has never heard before, are still in th development stage.

7.2. OUTPUT UNITS

Like input units, output devices are instruments of interpretation and communicatic between human and computers. These devices take machine coded output results fro the processor and convert them into a form that can be used by users. Sometimes outp is recorded directly on magnetic media, such as magnetic tapes, magnetic disks or dis ettes, which may be used subsequently as input to some other program. Output m also be obtained on a paper or microfilm or it may be presented on a display termin

Computers or microprocessor-based systems are now widely used for automatic co trol applications in industry and other commercial organizations. In such cases the co puter outputs electrical signals which are sent directly for control purposes.

A wide range of equipment and media are available for outputs, the actual cho depends on the following considerations:

- Suitability to application
- Speed at which output is required
- Whether a printed version is required
- Volume of data
- Cost benefit of a particular system.

This unit begins with a discussion of printers and printed output. Also discussed in this unit are microfilm output and audio output.

Output, understood by humans, can be in the form of (*a*) hard copy, and (*b*) soft copy.

Hard copy: It is output on paper and can be read immediately or stored and read later. This is a relatively stable and permanent form of output.

Soft copy: It is usually a screen-displayed output. It is a transient form of output and is lost when the computer is turned off.

Hard copy devices: Hard copy devices can be broadly grouped into two categories:

(*a*) Printer; and

(*b*) Plotter.

7.2.1. Printers

Printers are the primary output devices used to prepare permanent documents for human use. Printing output is called hard copy output. Printers which are used with computers are classified as follows:

(*a*) Character printers;

(*b*) Line printers; and

(*c*) Page printers.

A character printer prints one character of the text at a time. A line printer prints one line at a time and a page printer prints one page of the text at a time.

The printers have been classified as to how they print. There is another classification which depends on the technology used in their manufacture. Printing technology classifies the printer into two categories:

(*a*) Impact printers; and

(*b*) Non-impact printers.

Impact printers make physical contact with paper while printing, whereas non-impact printers transfer information to paper without physical contact. Line printers, dot matrix printers and daisy wheel printers are all impact printers as they strike the paper surface. Laser printers, xerographic printers, ink jet printers and electrothermal printers are non-impact printers. Non-impact printers are normally faster than impact printers and have more reliability since they use fewer movable parts in printing.

Character Printers

Character printers are printers which print one character at a time and are used with microcomputers and personal computers. Common examples of impact type of character printers are dot-matrix and daisy-wheel printers and non-impact type of character printers are ink-jet, thermal and electrostatic printers.

Impact Type Character Printers

Dot-matrix printers · Dot-matrix printer has a printing head with needles (pins) on it. The printing pins are arranged one below the other in a single column. Each pin can be independently activated electronically. When a pin is activated it prints a small dot on the paper. Thus, in a dot matrix printer, each letter is formed with a series of dots. A typical dot-matrix printer uses a 5×7 (i.e., the formation of a character is shown using 5-dot rows and 7-dot column) or 7×9 dot matrix formats to represent each character. A normal 7×9 dot printer has speeds from 50 to 600 per second. Many dot-matrix printers are bidirectional, i.e., they can print one line from left to right and next time from right to left.

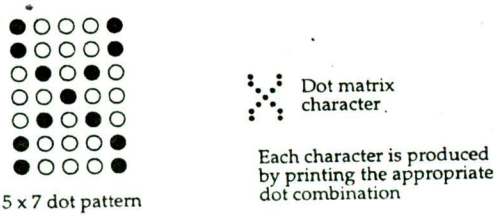


Fig. 7.8

The print resolution of dot-matrix printers is usually not as clear as that of typewriters because dot-matrix printers depend upon the human eye to connect the dots and recognize symbols. Dot-matrix printers are comparatively inexpensive and one of the fastest types of character-at-a time printers available. For these reasons, they are often used with microcomputers. The abilities to use variable type font (the term font is used to refer to a character set of a printer) and variable line densities are also important. Some of the advanced form of these printers are also capable of printing graphics and a few have multicolour facility.

Daisy wheel printers: The daisy wheel is a flat disk. The disk has a set of spokes each having a single character embossed at the tip. There is only one strike hammer. Like typewriters, an inked ribbon is kept between the paper and the hammer. The spin wheel is revolved continuously and brings the desired character between the ribbon and the hammer. The print hammer strikes the ribbon against the paper making an impression of the character on the paper. These are too expensive and very slow. Their speed range from 20 to 80 characters per second. The mechanism of daisy wheel printer is shown in Fig. 7.9.

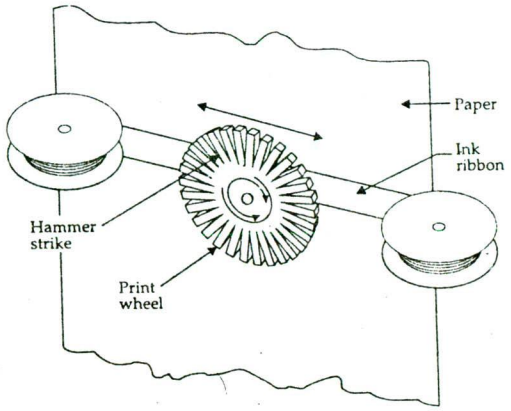


Fig. 7.9: Daisy wheel printer mechanism

Non-impact Character Printers

This type of printers use thermal, electrostatic, chemical and ink-jet technologies. They produce much less noise and print high quality characters.

Ink-jet printers: The printer head is a specially designed nozzle. It sprays small droplets of ink at high speed. Droplets of ink are electrically charged after leaving a nozzle. The droplets are then guided to the proper position on the paper by electrically

charged deflection plates. In this type of printers the continuous-stream of ink-jet approach is used. In another type of ink-jet printers, the drop-on-demand ink-jet approach is used. The ink drops are produced when needed. Multiple nozzles are used in these types of printers.

The high resolutions of ink-jet printers make them suitable for letter-quality printing. Many of them can be programmed to print unusual symbols. Coloured printing is also possible. The speed (40-300 cps) and resolution of these printers are as high as that of laser printers at a lower cost. But the droplet generator is highly sensitive to dust and, hence, the maintenance is difficult.

Thermal printer: Thermal printer produces heat to produce characters in dot-matrix form on special sensitised paper. To print a character the printing head is moved first to the correct character position. Then the heating element for the desired character is turned on. The print head is moved to the next character position after they turned off. The cost of thermal paper has prevented thermal printers from becoming commercially viable.

The new thermal type printer uses ribbons which hold ink in a wax binder. The pins forming a character pattern are heated and pressed against the ribbon. In this process, the wax melts and the ink gets transferred to the paper.

Electrostatic printers: The electrostatic printers also need specially prepared paper. The electrostatic approach applies an electronic field to a sensitised paper. The process creates charged spots on the paper, which attract ink particles when the paper is passed through a toner (powdered ink) solution.

Impact Line Printers

A line printer prints a whole line at a time. A typical line printer produces 2000 lines per minute. They are used with mini, micro or mainframe computers.

There are three types of line printers: (*i*) Drum printers, (*ii*) Chain printers, and (*iii*) Band printers.

Drum printer: A drum printer consists mainly of a cylindrical drum. The characters to be printed are embossed on its surface. The codes of all the characters to be printed on one line — are transmitted from the memory of the computer to a storage unit in the printer. The printer drum is rotated at a high speed. A set of print hammers — one for each character in a line — are mounted in front of the drum. A character is printed by the striking of the appropriate hammer against the embossed character on the surface. A carbon ribbon and paper are interposed between the hammer and the drum. As the drum

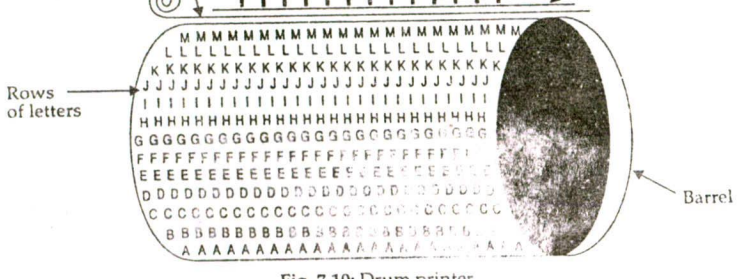


Fig. 7.10: Drum printer

rotates, the hammer waits until the character to be printed appears in front of it. Thus, the drum would have to complete one full revolution for a line to be printed. The movement of the drum and striking of the hammer must synchronize accurately. Otherwise, the printing will not be uniform. As printer drums are expensive, they cannot be often changed.

Chain printer: In the chain printer, the print characters are mounted on a chain which moves horizontally in front of the paper. There is one print hammer for each column on the paper. The chain may contain several sets of the print character. As a character to be printed passes in front of each position on the paper, where it is to be printed, the magnetically controlled hammer behind the paper presses the paper against the type set to print the character (using an inked ribbon which is placed between the chain and the paper). Thus, moving one character set of the chain past the paper is sufficient to print a line. In normal printers, the chain can be removed so that more than one type fonts (e.g., italic or bold face) and a set of special characters may be used.

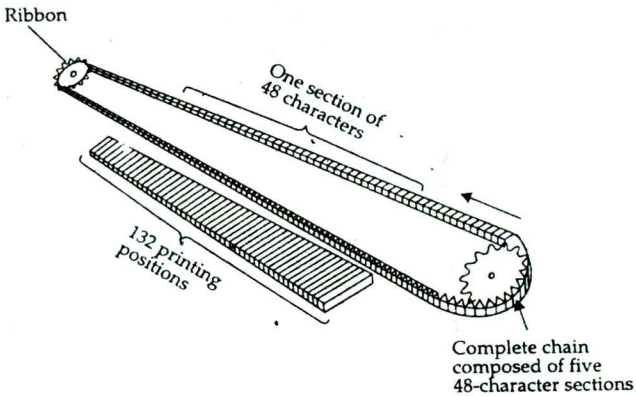


Fig. 7.11: Chain printer

Band printers: A band printer works much the same way as a chain printer except that the print characters are embossed on a metallic bond instead of a print chain. The print band cycles continuously on a track. Hammers strike the ribbon and the paper against the character to print the character. Some printers can print up to 3000 lines/min.

Non-impact Page Printers

Laser printers: These are based on laser technology and print one page at a time. It employs a revolving drum, which has electrostatic charges on its surface. The signals are sent from the computer to a laser beam. The laser beam is deflected by a revolving mirror, so that the laser beam can scan the surface of the revolving drum. The laser exposed areas attract toner (an electrostatic sensitive black powder). At the next stage, the drum transfer the toner to the paper. The paper then moves to a fusing station where the toner is permanently fused on the paper with heat or pressure. Thereafter a cleaning process is used to clean off excess ink and the drum gets ready to print the next page.

Low speed laser printer used with PCs provides output ranging from 6 to 12 pages per minute. High speed laser printers producing up to 300 pages per minute are manufactured for mini and larger computers. Laser printers are an example of a popular printing method for producing high quality printed output.

7.2.2. Other Forms of Output Devices

Computer Output on Microfilm (COM): COM is basically an output device that records computer output information in microscopic photograph. The recording technology consists of a microfilm recorder that receives information that are normally stored on magnetic tapes. Recorder may also receive the information directly from the CPU. The recorder in turn project the characters of output information on to a CRT screen. A high speed camera, inbuilt into the system, now takes a picture of the displayed information at speeds of up to 32,000 lines per minute. The standard photographic rolled films of 16 mm or 35 mm are used. A COM recorder is expensive, and a high volume of workload is needed to justify its cost. The technology is suited for organisations in which vast computer printed outputs present a storage problem, banks and insurance companies to name a few.

Microfiche: A microfiche (Fiche is a French word meaning card) is a 4" × 6" sheet of film that can store about 700 A4 size pages of information. A special microfiche reader is required to read stored information on microfiches. Microfiche is easier to mail between locations.

Voice output: Voice output refers to computer output that is translated into spoken language. Voice output devices range from mainframe audio-response units to speech synthesizer microprocessors. 'Talking' chips are currently used to provide computerized speech for toys, games, automobiles and a variety of other consumer, commercial and industrial uses. In automobile, drivers are warned of open doors or low oil levels; in banking, bank customers use touch-tone telephones to obtain their account balances; in telephones, callers are informed that a number has been changed, etc.

REVIEW QUESTIONS SET

1. What do you understand by 'Peripheral'? Explain with examples. Discuss the function of input and output devices.
2. Describe a punched card inasmuch detail as possible. Why are punched cards not used much in data transcription any more?
3. Explain the difference between smart, dumb and intelligent terminals. **(AMIE, W '96)**
4. What is an optical scanner? Describe Optical Character Reader, Optical Mark Reader and Optical Bar Code Reader.
5. What does MICR stand for? Discuss MICR with its area of application. What are the advantages and disadvantages of MICR?
6. What are point-of-sale terminals? Describe their advantages and disadvantages.
7. Discuss the functions of a mouse, light pen, joy stick and track ball.
8. What are the different types of printers? Why these are called hard copy devices? What is the difference between impact type and non-impact type printers?
9. Distinguish between character printers, line printers and page printers.
10. What is a laser printer? Discuss the working principle of laser printers.
11. Write short notes on the following:
    (*a*) Dot-matrix printer
    (*b*) Ink-jet printer
    (*c*) Page printer.
12. What are microfilm and microfiche? Discuss their working principles and applications.

8

CLASSIFICATION OF PROGRAMMING LANGUAGES

8.0. INTRODUCTION

A language is a system of communication. A programming language consists of all the symbols, characters, and usage rules that permit people to communicate with computer. Three types of programming languages are available. They are:

1. Machine language (known as low-level langauge).
2. Assembly (or symbolic) language.
3. Procedure - oriented language (known as high level language).

8.1. MACHINE LANGUAGE

The computer understands nothing but 0s and 1s. Thus, the most basic method to program a computer is to fed the string of binary coded instructions to the computer. The language in which binary code is used to write a program is known as machine language (low level language). Since it is the most basic form of programming and, hence, explicit instructions are to be given to the machine to perform every operation. An instruction prepared in machine language will have at least two parts: a Operation Code (OP CODE) for each of its functions like add, multiply, etc., and the OPERAND, which tells the computer where to find or store the data or other instructions that are to be manipulated.

Fig. 8.1: Structure of a machine language instruction

The number of operands in an instruction varies among computers. As the machine code is represented in binary digits, the programmers are expected to know all the operation codes, and the addresses and length of operands that could be used in that particular computer. Because machine code is determined by each CPU hardware design, machine languages are said to be machine-dependent.

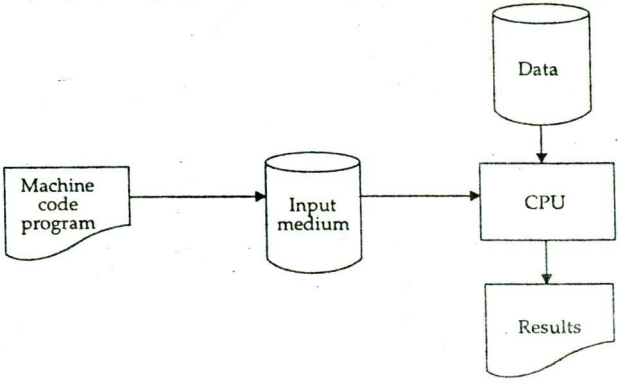


Fig. 8.2: The machine language program run

Advantages

The programs written in machine language can be executed very fast by the computer. This is

mainly because machine instructions are directly understood by the CPU and no translation or compilation of the program is needed.

Machine languages make efficient use of storage-language instructions and their storage in computer memory can be controlled by manipulating the individual bits.

Disadvantages

- Writing program in machine language is tedius, time-consuming and highly error-prone.
- Writing program in machine language requires a high level of programming skill.
- Machine languages are machine-dependent (that means a program that has been developed for a particular machine cannot be run on another machine).
- It is difficult to correct or modify machine language programs. Checking machine instructions to locate errors is same as writing them initially. Any modification in machine language program results in a series of changes. If a program is to be modified by adding or deleting some instructions, then addresses of all subsequent instructions are to be changed. The change causes further modifications in addresses of the operands.

8.2. ASSEMBLY LANGUAGE

It is the language in which programmer uses some form of name or level to refer to specific storage locations, i.e., binary address by symbolic address and instructions (OP CODE) are expressed in mnemonic codes such as ADD, MOVE, etc. Assembly language software (known as assembler) first translates the specified operation code symbol and symbols address into its machine language equivalent, because the only language understood by the machine is the machine language.

Advantages

- Operation codes used in machine language are replaced by mnemonics which are easier to remember.
- It is not required to keep the track of memory locations as is required in the machine language. The memory addresses are replaced by the variable names.
- While writing programs in assembly language, fewer errors are made, and those are easier to find.
- It is easier to modify because of the use of mnemonics are symbolic field names.
- Insertions and deletions in the program are easy.
- The human effort required to write a program is much less as compared to that needed in writing of a machine language program.

Disadvantages

- An assembly language program cannot be executed on small-sized computers.
- Assembly languages are machine-dependent as each instruction in the symbolic language is translated into exactly one machine language. It means that they are designed for the specific make and model of the processor being used.
- Coding in assembly language is time-consuming.

8.3. HIGH LEVEL LANGUAGE

The language in which symbols and words are similar to those of ordinary arithmetic and English and are independent of the computer of which the program is to be used, is known as high level language.

Hundreds of languages have been evolved along these lines, but none is regarded

as universal. The widespread languages are BASIC, FORTRAN, COBOL, PASCAL, PL/1, ALGOL and

Advantages **(AMIE, S '96)**

- High level languages are machine-independent.
- They are easier to learn than assembly languages.
- They require less time to write.
- The writing of program in these languages does not require the knowledge of the internal structure of a computer.
- They provide better documentation.
- They are easier to maintain.
- Modifications, if required, written in these languages are easy and straight forward.

Disadvantages

Lower efficiency : As the programs written in high level languages need a compiler which is loaded into the main memory of the computer and, thus, occupies enough of memory space, take more time to run. Hence, a program written in assembly language or machine language is more efficient than the one written in high level language.

Lack of flexibility : Because the automatic features of high level languages always occur and are not under the control of the programmer, they are less flexible than assembly language. An assembly language provides programmers access to all the special features of the machine they are using.

The most common high level languages and their acronyms with application area are as follows :

Acronym	*Expanded Form*	*Main Applicaton Areas*
BASIC	Beginners All-purpose Symbolic Instruction Code	Scientific problems, games, puzzles and business.
FORTRAN	FORmula TRANslation	Scientific problems particularly solving numerical problems.
PL/1	Programming Language	General purpose.
ALGOL	ALGOrithmic Language	Scientific problems expressing stepwise solution to problems.
APL	A Programming Language	Scientific problems.
COBOL	COmmon Business Oriented Language	Commercial and business with good file handling facilities.

An example of machine, assembly and high level language is given.

Machine Language

OP CODE (Function)	*OPRAND (Operand Address)*	*Meaning*
00100	000001111	Store the number in store position 000001111 in the accumulator.
00101	000001100	Add the number in 000001100 to the quantity in the accumulator.
10000	000001101	Move the quantity in the accumulator to store in position 000001101.

Assembly Language

LDR	R1, X	Load the value stored at location X into Register 1.
ADD	R1, Y	Add the value stored at location Y to the value in Register 1.
STORE	R1, Z	Store the value in Register 1 at location Z.

High Level Language

Z = X + Y (BASIC OR FORTRAN)
ADD X TO Y GIVING Z (COBOL)

Source Program

A program written in assembly or high level language is known as *source program.*

Object Program

Any program not written in machine language has to be translated before it is executed by the computer. Object program is a translation of source program into machine language program.

8.4 TRANSLATOR (COMPILER, INTERPRETER, ASSEMBLER)

Translator is a computer program that performs the task of converting a program written in one programming language into a program in another programming language. Translators of programming language are mainly classified into two groups depending on the nature of the source language accepted by them.

Compiler

Compiler is a translator that accepts instructions written in high level language and compiles each instruction into machine language producing a complete program in machine language known as object program. This phase is known as compilation phase. All the testing of the source program as regards the correct format of the instructions is performed at this stage and errors (syntax errors), if detected, are printed out as messages to the programmer to enable him to correct his program. These error messages are called *compiled time diagnostic error messages.* Once the program is compiled successfully, one could use this compiled program next time. It will not have to compile again unless some change is needed. Since compilation is slow process, this will save much time when one avoids compiling programs. The compiler resides on a disc on other mass-storage media. When the compiler is needed, it is called and is capable of translating source programs written in only one high level language. Thus, a Fortran compiler cannot be used to translate a Cobol source program (Fig. 8.3).

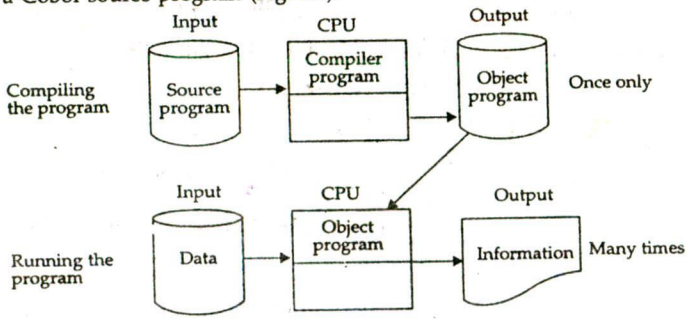


Fig. 8.3: Compilation and execution

Interpreter

The development and rapid spread of the video keyboard terminal opened a way of communicating with a computer by direct interaction. Compiled languages do not fit into this scheme very easily. Interpreter is a one kind of translator that translates and executes each source language statement written in high level language before translating and executing next one but does not produce an object program. For example, suppose the following three lines in BASIC are typed :

```
LET   A  =  6
LET   B  =  2 + A
PRINT B
```

Immediately after the first line is entered, computer creates a storage area, namely, A and stores the value 6 in it. Immediately after the second line is entered, computer creates a storage area, namely, B and stores the value of 8 in B. Immediately after the third line is entered, the computer prints the value 8 on terminal's screen.

BASIC, APL, PROLOG and LISP are mainly interpreted languages. PASCAL, FORTRAN, COBOL, PL/1, C, ADA are mainly compiled languages.

A compiler or an interpreter is itself a program written in some language called host language. The three languages involved in compiler are: source, object and host which are often different. A Fortran compiler might be written in PL/1.

Comparison between Compiler and Interpreter **(AMIE, S '96)**

The compiler translates entire program chunk at a time while an interpreter translates single line at a time. Thus, compiler produces object program but interpreter does not.

Use of an interpreter can save core space since the interpreter program itself is quite small in size. It also eliminates the need to store the machine language version of the program whereas compiler takes more space because of its long program and there is a need to store machine language version of the program.

Use of an interpreter gives rapid and direct feedback. Modification or adding something can be done immediately. Compilers more or less reveal the situation that one do-not get any feedback until the whole program is compiled and processed. For any modification the program is to be compiled again.

With interpreter, program statements that are used multiple times must be translated each time they are executed and its speed is very slow but with compiler one can save much time using the compiled program. The object code, thus obtained, is permanently saved for future use and is used every time the program is executed. Thus, repeated compilation is not necessary for repeated execution of a program. Figure 8.4 shows the comparison between compiler and interpreter.

Interpreter	Compiler
1. Translates the program line by line.	Translates the entire program at a time.
2. Requires less main memory.	Requires more main memory.
3. Each time the program is executed, every line is checked for syntax and then converted to equivalent machine code.	Converts the entire program to machine-code, when all the syntax errors are removed, and executes the object code directly.
4. Source program and the interpreter are required for execution.	Neither source nor the compiler are required for execution.
5. Good for fast debugging and at testing stage.	Slow for debugging and testing.
6. Execution time is more.	Execution time is less.

Fig. 8.4

Assembler

The assembler, like the compiler, is a translator that converts assembly language written program into the machine language program. The resulting program can be executed only when the assembly process is completed. It also generates diagnostic error messages.

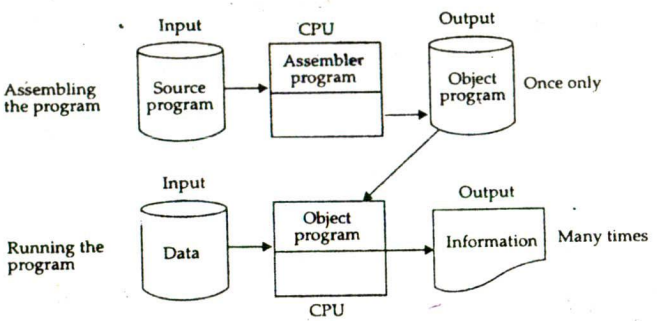


Fig. 8.5 : Assembly and execution

8.5. PROCESS OF COMPILATION

A compiler is a translator program which converts a high level language program into an equivalent machine language program that can be executed. The major steps involved in compilation process are:

(*a*) Lexical analysis,

(*b*) Syntax analysis, and

(*c*) Code generation.

Lexical analysis: The lexical analysis phase reads the source program and separates all characters of the source program into meaningful groups called the tokens. The tokens in the source program are typically the names of the data items (identifiers) defined by the programmer, operators (such as *, > , etc); reserved words (or keywords) (such as IF and WHILE); and punctuation symbols (such as parentheses and semicolons). A token has two parts:

- the token type, and
- the token value.

The token type indicates the kind of entity the token represents. Examples of token types are identifiers, constants are labels. The token value represents the value of the token as it appears in the source program.

Let us consider the following high level statement(s) to understand how lexical analysis is performed by the computer.

If (sum > 30)

Max = 100;

The lexical analyser reads the above statement and produces the following sequence of token.

Token Type	*Token Value*	*Line Number*
reserved word	If	1
punctuation	(	
identifier	Sum	

operator	>	
constant	30	
punctuation	)	
identifier	Max	2
operator	=	
constant	100	
punctuation	;	

Syntax analysis : The syntax analysis of a source program performed by the compiler ascertains whether the grammatical rules of the language have been followed or not. If these rules have been followed correctly then the source program is converted into machine language and loaded into main memory of the computer for execution.

The output of the lexical analyser is the input of syntax analyser. The syntax analyser

- receives the tokens one by one,
- checks if the tokens occur in permissible sequences, and
- groups the token into syntactic structures following the rules of the source language. This activity is called 'parsing' and the syntax analyser is called 'Parser'.

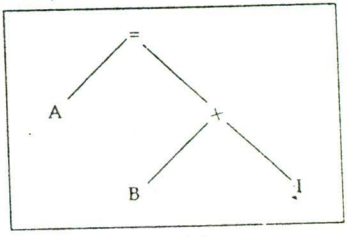


Fig. 8.6

Output of the parsing step is a representation of the syntactic structure of a statement. A convenient presentation is in the form of a syntax tree. For example, the statement A = B + 1 could be represented by the syntax tree of Fig. 8.6. Syntax errors like missing operators/operands, etc., would be pointed out during syntax analysis.

Code generation: The code generation phase produces the appropriate machine language statements from the structures produced by the syntax analyser. The code generation involves

- allocation of memory space for the data items used in the program, and
- generation of machine language instructions.

8.6 DEBUGGING

The error in a computer program is termed as bug. A bug in the program can abort a computer run and/or produce absurd output. The process of location and removal of errors is called debugging. The errors present in a source program are in general of three types, namely:

(*a*) Syntax errors;

(*b*) Execution errors; and

(*c*) Logical errors.

Syntax errors: These errors occur due to the violation of the rules (syntax) of the language like incorrect punctuations, invalid symbolic names, misspelled keywords, illegal statements, etc. These errors get detected at the compilation stage and is, therefore also referred to as the compilation errors. The syntax errors are the easiest to locate and correct since they are identified by error messages indicating the line number in which they occur and the cause of the error. Execution will begin until all compilation errors have been eliminated.

Execution errors: Syntax errors usually prevent execution of the program from starting. Other kinds of errors may allow execution to begin but cause it to terminate prema

turely or producing wrong results. Execution errors are detected during execution and may be referred to as runtime errors. Examples of execution errors include :

- Attempts to divide by zero.
- Infinite loops causing no output.
- Taking square root of logarithm of a negative number.
- Using an array that exceeds the space reserved for it.

It is difficult to locate the specific cause of execution errors in a long program or loop.

Logical errors: Logical errors arise from faulty programming logic as, for example, the attempt to execute (*a*) invalid sequence of instructions, (*b*) a valid instruction but with invalid data. These errors are difficult to locate. A logical program error can be detected by going through the flow chart and by running programs on some sample data, for which the answer is known.

8.7. PROGRAM DESIGN

The important techniques which are useful in designing programs are as follows:

(*a*) Modular programming,

(*b*) Structured programming, and

(*c*) Top-down and bottom-up design.

8.7.1. Modular Programming

When a program becomes very long and complex, it becomes very difficult for the programmer to design, test and debug such a program. Therefore, a long and complex program is split into a number of smaller programs known as modules. A module in itself is a complete program which can be designed, tested and debugged separately. Every module is designed to perform specific task. Some of the modules may be broken into submodules, which in turn may contain submodules of their own and so on. The technique of writing programs in modules is called modular programming.

The characteristics of a module are:

1. A module is designed to perform a single task.
2. A module is self-contained and independent of other modules, i.e., a module can be compiled, tested and debugged separately without the intervention of the other modules.
3. A module has only one entry and one exit point.
4. A module can call other modules.

The modular design approach has the following advantages over a non-modular one :

1. Program readability is improved. This in turn reduces the time needed for debugging and maintaining the program.
2. Programmer productivity is increased because it is easier to design, code and test the program one module at a time than all at once.
3. The various program modules can be designed and/or coded by different programmers, thus, providing the possibility of earlier completion.
4. In some cases, a single module can be used at more than one place in the program. This reduces the total amount of codes within the program.
5. Modules performing common programming tasks (such as sorting) can be used in more than one program. These utility modules reduce both design and coding time.

The modular design approach suffers from the following drawbacks:

1. Modular programs often need more memory space and extra time.
2. It may be difficult task to integrate the various modules written by different programmers into a single program.
3. Testing and debugging of separate modules may prove to be time-consuming.

8.7.2. Structured Programming

A method of writing program in a certain systematic way is known as structured programming. A structured program is highly readable, easily debugged, easily maintained and can be developed quickly.

The structured programming is based on the use of the following techniques:

(*a*) Control strucutre,

(*b*) Modular programming, and

(*c*) Top-down approach in program design.

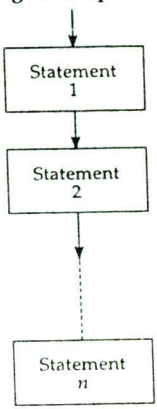


Fig. 8.7

Control Structure

There are three basic control structures:

1. Sequential structure,
2. Loop structure, or repetition structure, and
3. Decision structure, or selection structure.

The sequential structure consists of services of instructions statements of which are executed consecutively in sequence.

A loop control structure or repetition control structure contains a block of code that is executed repeatedly till the condition is satisfied. There are two fundamental types. DO while (pre-test) and DO until (post-test). The exit condition for a DO while loop is tested before entering the body of the loop; and of a DO until loop is tested after executing the body of the loop (Figs. 8.8 and 8.9).

A decision control structure or selection control structure consisting of a test condition and one or more blocks of code. The results of the test determine which of these blocks are executed. A decision structure is called **IF Then** or **IF Then Else** or **case structure** (Fig. 8.10).

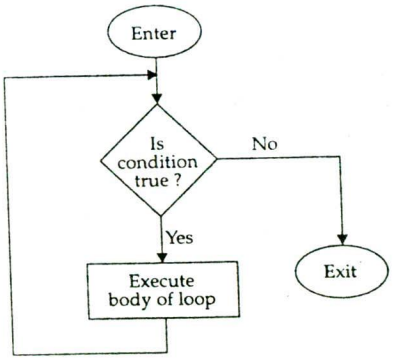


Fig. 8.8: DO while loop

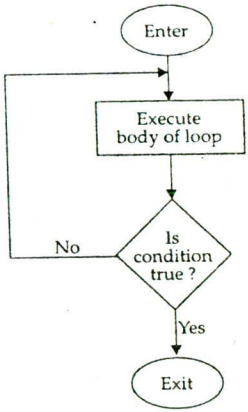


Fig. 8.9: DO until loop

8.7.3. Top-down and Bottom-up Design

In top-down approach, the program is first considered as a whole and is then sub-divided hierarchially into less complex, smaller and easily manageable form until a stage is reached when further breakdown will serve no useful purpose.

The disadvantage of top-down design is that the overall system design may not take

good advantages of the hardware. Bottom-up approach for a program design is just the reverse of the top-down approach. Lowest level sub-tasks are designed first and then combined/linked to form a complete program.

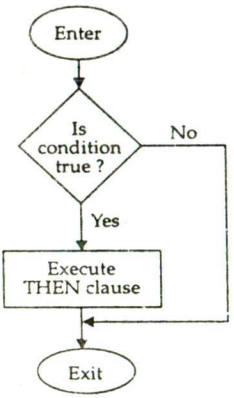


Fig. 8.10: IF THEN structure

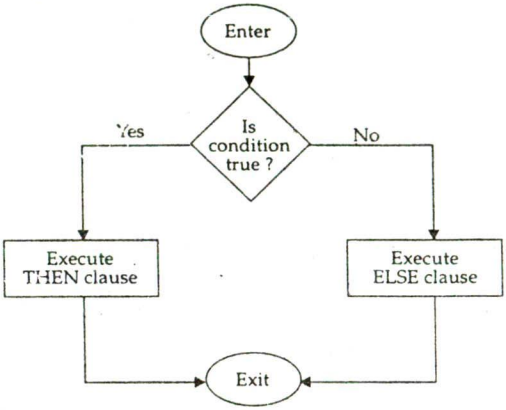


Fig. 8.11: IF THEN ELSE structure

8.8 PROGRAM DOCUMENTATION

Program documentation means writing the explanation about the program in the form of comments and remarks placed at various places in the program. It indicates what functions are performed by the program, and how these functions are carried out. It helps users to understand, maintain and modify the program. The sentences used in documentation are not executed because the compiler overlooks them. The documentation included in programs in different common languages are as follows:

(*a*) **Fortran:** A documentation must be preceded by writing C or * in the first column.
(*b*) **Cobol:** An asterisk * in column 7 make a documentation line.
(*c*) **PL/I:** A documentation line is enclosed in /*and */ at left and right hand sides respectively.
(*d*) **Pascal:** A documentation line is enclosed in a pair of braces { and }.

REVIEW QUESTIONS SET

1. What do you understant by low-level and high-level languages? Explain with examples.
2. Compare the merits and demerits of assembly languages and high-level languages.
3. Discuss modular programming, structured programming and top-down and bottom-up design techniques.
4. Explain the terms : (*a*) Source program, and (*b*) Object program. Which of these is executed by a computer? **(AMIE, W '97)**
5. Explain the term structured programming.
6. What are the advantages and limitations of high level languages? **(AMIE, S '96)**
7. What are the main differences between a compiler and an interpreter? **(AMIE, S '96, W '97)**
8. Write short notes on: (*a*) Compiler, (*b*) Interpreter, (*c*) Assembler.
9. What is machine language? What are the advantages and disadvantages of it?
10. In the content of a compiler, what do you understand by the following?
    (*a*) Logical analysis (*b*) Syntax analysis
    (*c*) Semantic analysis (*d*) Intermediate code generation **(AMIE, W '96)**
11. What are assembly languages? What are its advantages over machine languages?
12. What is meant by program documentation? How is documentation included in programs in different languages you have studied? **(AMIE, W '93)**

9

OPERATING SYSTEMS

9.0. INTRODUCTION

In early computer system, operators monitored computer operations. They used to load the program translations and program cards which were read by translating program. The output of these programs were written into a tape. Operators used to load these tapes and put the program into memory and start it. The data cards were loaded by the operators at this stage and the output devices were made ready by the operators. Operating system (OS) eliminated the necessity of human interaction at each stage, i.e., OS effectively isolates hardware from the user.

9.1. WHAT IS OPERATING SYSTEM?

An Operating System (OS) is an organised collection of programs that acts as an interface between machine hardware and users providing users with a set of facilities and maintenance of programs and at the same time controlling the allocation of resources to cause efficient operation. Its prime objective is to improve the performance, and efficiency of a computer system and increase facility, the ease with which a system can be used. An operating system performs the following functions:

1. **Resource management:** Allocation of computer resources such as processor, memory, and I/O devices, the jobs (programs) being executed.
2. **Job management:** Scheduling (selecting) new jobs for execution according to the desired priority.
3. **I/O management:** Managing the flow of data and instructions between the I/O units and the primary memory.
4. **Data management:** Providing data management facilities such as data organisation and retrieval from secondary storage devices.
5. Maintaining security, communication of error, and control messages to the users, human operators, etc.

With an operating system, the computer system can be perceived as a collection of interacting hardware and software elements, functioning at different levels (Fig. 9.1).

Usually, operating systems are too large to be stored in memory at a time. They can be divided into a number of parts. Some portions of the operating system must always be present in the memory which is called **nucleus** or **kernel**. It is also called supervisor or monitor. It performs the basic operations such as starting and terminating user programs, allocations of memory and files, basic input/output operation and in-

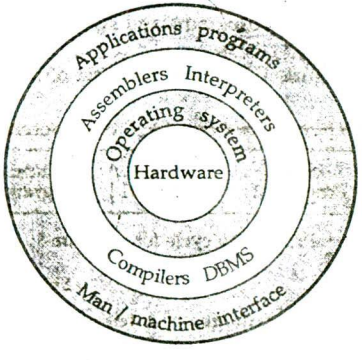


Fig. 9.1 : Functional levels of a computer system

terrupts. Other portions of the operating system which are brought into the memory when needed and removed when not needed, are called **transient programs.**

Bootstrap Program

The operating system programs are stored on disk. When a computer is turned on, the operating system must be brought into the computers' memory from the hard disk memory. The process of reading the operating system programs from disk; loading it in the main memory and executing it is called "booting". The function of the bootstrap program is to perform the booting process. The bootstrap program is permanently stored in the main memory (usually in PROMs). This program is automatically executed when the computer is switched on or *reset*. Once the computer has been booted, the user can enter any command supported by the operating system.

In general, the operating system can be classified as follows:

(*a*) Batch processing,
(*b*) Multiprogramming or concurrent programming,
(*c*) Time-shared multiprogramming,
(*d*) Multiprocessing,
(*e*) Real time, and
(*f*) Network.

9.1.1. Batch Processing

The first operating system, called batch-processing (serial) operating system (OS), was developed for the second generation computers. The data are gathered for a time and collected into a group (or batch) before they are entered into a computer system and processed. The system would process the program one after the other. In this mode, one user has complete control of the machine until his program is completely exhausted. Each program is executed from the beginning to the end without pause. The batch processing OS makes no provision for job scheduling criteria since it can take the jobs in order in which they have been put at the input of the computer system, i.e., first-come-first-serve basis. The batch mode is most suited for programs that are long or require large computing time and remains an efficient approach to use in applications such as preparing bills, processing payroll checks, periodic sales analysis, etc., where it makes sense to accumulate data and then process all the data for a period at once. In this systems, CPU remains idle most of the time because as compared to CPU speed, the speed of I/O devices is quite low. However, the work of mounting and dismounting of tapes for loading of user's programs and system programs such as compilers, etc., is taken over by the operating system.

Figure 9.2 shows utilisation of resources using a batch processing operating system.

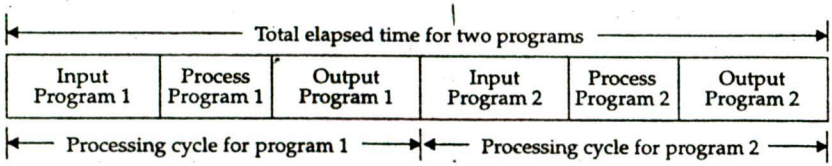


Fig. 9.2 : Batch processing

The advantages of batch processing operating system are as follows:

- User intervention for execution of each job is not required.
- Batch processing reduces the CPU (or computer system) idle time during transition from one job to another, as it does not require operator's intervention.

- Most repetitive kind of job(s), e.g., payroll processing checks, sales analysis are often executed as a "batch".

The disadvantages of batch processing operating system are as follows:

- Batch processing makes each job wait in line at each step and often increases its turnaround time (i.e., time elapsed between job submission and completion in system).
- Batch processing system suffers from under-utilisation of computer resources. When a program is executed, all computer resources are allocated to that program resulting: (*i*) if a program is small, it does not occupy all the memory and, therefore, its expensive memory is wasted, (*ii*) not every program is purely computational. The large commercial type of program normally reads in vast amount of data, performs very little computation and output large amount of data. Thus, when a job is performing I/O operation, the CPU remains idle till the I/O operation is completed. Similarly, if a job is doing some computation, the CPU time is utilised, but the I/O devices lie idle. Thus, the batch processing systems suffer from under-utilisation of computer resources.

9.1.2. Multiprogramming

Batch processing essentially dedicates resources of the computer system to a single program at a time. To keep all units of computer simultaneously busy for most of the time it is desirable to process a number of programs concurrently. A multiprogramming operating system keeps many jobs of different users in the memory at a time, schedules and executes them and provides I/O facilities requested by each user and optimises the use of computer resources. The processor executes a portion of one program, then a portion of another and so on, and then it comes back to continue the computation of first user's job, from where it was earlier suspended for another short burst of computation. This cycling continues indefinitely. When one program is finished it is replaced by another one.

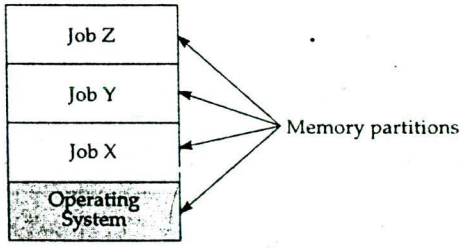


Fig. 9.3

Suppose three programs, X, Y and Z, have been loaded (input) into the main memory from secondary memory, and are ready for processing. Let the CPU take up program X for processing and one of the processing steps in program X involves retrieving a few records from a disk file. Such an I/O request interrupts the CPU, as it has to wait till the desired records are brought into the main memory. In such a situation, a multiprogramming OS will switch CPU from program X to another program, say program Y, based on some criteria instead of idling the CPU (waiting for program X to complete its I/O operations). I/O operations are very slow compared to CPU operations. The CPU continues to process Y till it encounters an I/O request from Y. At that instant of time, CPU switches to X or Z whichever is ready for execution, and this process continues.

Figure 9.4 shows three different states, READY, RUNNING and BLOCKED of a program residing in main memory in case of multiprogramming.

READY : The program is able to use the process or when it is assigned to it.

BLOCKED : The program is waiting for the I/O operation to complete and is not able to utilise the processor at present.

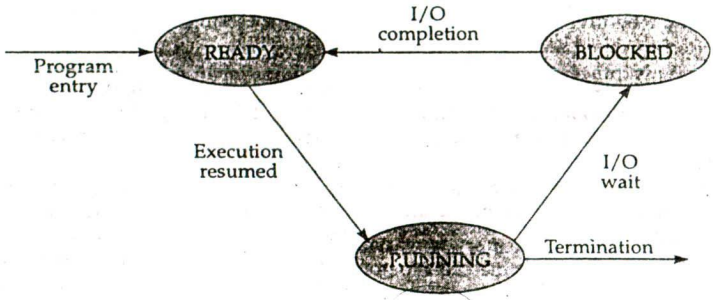


Fig. 9.4 : Three different states of a program

RUNNING: The program is under processing by CPU.

Points worth noting are :

- program entering the system must go initially into a READY state
- program can only enter the RUNNING state via the READY state
- program only (normally) leave the system from the RUNNING state

We say 'normally' in the last line because it is possible for a program to be aborted by the system (in the event of an error, say) or by the user, which could catch the process in the READY or BLOCKED state.

Figure 9.5 shows the utilisation of resources using a multiprogramming system.

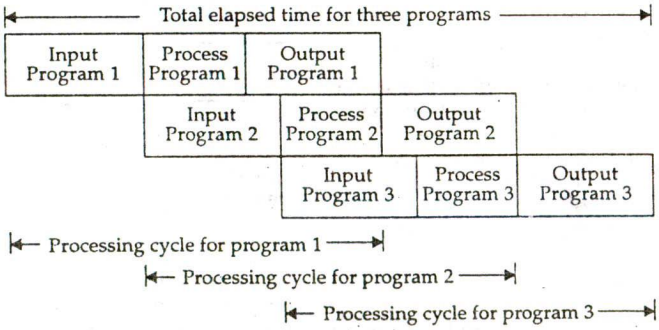


Fig. 9.5 : Overlapped processing

The main advantage of multiprogramming is that the user does not have to wait until the computer finishes running all other programs. A multiprogramming system increases the *throughput* of the computer.

Spooling

This term is an acronym for Simultaneous Peripheral Operations On-Line (SPOOL). In multiprogramming environment, it often becomes necessary that input data from low speed devices are stored temporarily on high-speed secondary storage units to form a queue which can be quickly accessed by the CPU. Output data are also written at high speed onto tape or disk units and form another queue waiting to use slow speed devices such as printer. The technique is known as **spooling**, carried out by means of a spooler program. This is required because of the inherent mismatch between the processing

speeds of the CPU and slow speed input/output devices. Hence, with the aid of spooling technique, the CPU does not have to wait for the slow input/output devices and, hence, it can work at its maximum speed.

9.1.3. Time-sharing

A multiprogramming system increases the throughput of the computer. It is more desirable for an individual user to minimise turnaround time. Time sharing operating systems were evolved to give a quick turnaround time to an individual user in a group of user processing their jobs in a system. In this system several users work on the computer simultaneously. Several terminals are connected to a single CPU and operates on a time-shared basis. The CPU allots a fixed time period (time slices) to each user and serves them in turn. If the work is not completed at the end of its time slice (usually 20 millisecond), it is interrupted and placed on a waiting queue in a READY state permitting another 'ready' program to enter. The CPU switches from one job to another when there is a natural program break or the fixed time period has expired whichever happens earlier. The process is so fast that a user has the illusion that no one else is using the computer. Because of this, time sharing is often called **interactive processing system.**

In such a system the operating system keeps track of ready programs, that is, those programs which are ready for execution. A program which was executing but needs I/O operation enters a 'Wait' state and becomes ready again after completion of the I/O operation. But if it is just forced out because of expiry of time slice then it remains in 'Ready' state.

The next program to be executed is selected from ready one using scheduling algorithm which takes into account many things: priority, CPU time consumed, etc.

The most popular time shared OS readily available on various computers is UNIX.

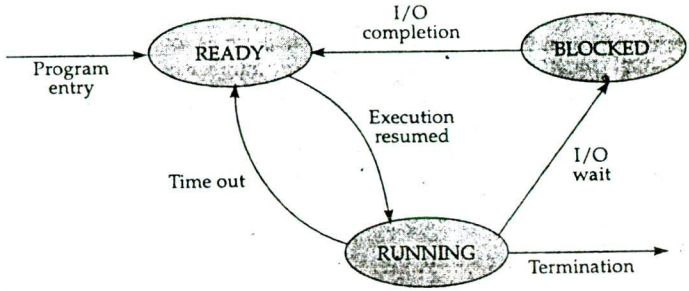


Fig. 9.6 : Three different states of a program

Scheduling

In the multiprogramming environment, at any time, more than one program may be awaiting execution. The operating system selects each program one after another, depending on the availability of CPU. The portion of the operating system that performs this function is called the "scheduler" module. Thus, the function of the scheduler module is to *schedule* the CPU for running multiple programs in an effective manner. The scheduling algorithms are generally classified as :

(*a*) Non-preemptive scheduling, and

(*b*) Preemptive scheduling.

In preemptive scheduling, the CPU is released by a currently running program when any one of the following conditions is satisfied:

- When the time-slice allocated for CPU is over.
- Some other program, having a higher priority than the running program, is ready for execution.

In non-preemptive scheduling, once the CPU is allocated to a particular program, it is not released until the program itself releases it. Usually the program releases the CPU when any one of the following happens:

- The program execution is completed.
- When a program needs to perform some I/O operation, it enters into a 'Wait' state. It stays in the wait state till the I/O operation is complete.
- The program aborts due to some error.

Time-sharing arrangements, however, have to deal with the problems arising from the limitation of many terminals sharing the main memory. To overcome the same, operating system needs the capabilities of virtual storage and the allied schemes of paging and memory swapping. In paging, each program is broken into small sets of instructions, called pages or segments. This makes it possible that only those program pages which are actually required at a particular time in processing to be brought in the primary (the real) storage. The remaining pages can be kept temporarily in online or virtual storage from where these can be retrieved when needed, following a program interrupt. The task of moving program pages between the primary and secondary storage is carried out by the operating system. Such loading and unloading of programs between two stages is known as **swapping.**

Advantages of time-sharing operating system are as follows:

- By using the time-sharing operating system, more than one user can access the system.
- Though many users are using the time-sharing system, each user gets the impression that only he/she (and no one else) is using the computer.

Disadvantages of time-sharing operating system are as follows:

- For small organisations, where only a small number of users use the computer system, time-sharing operating systems may not be cost-effective.
- It is extremely important to have good response time (the time elapsed between the command typed by the user and the computer's reply to that command) in time-sharing systems. As the number of users increase on the system, the system throughput decreases in time-sharing systems.

Virtual Memory

It is the technique for enabling a computer to handle large programs in larger than available storage capacity. Each program instruction and each data item to be used by the programmer must be in the main memory during the time the instruction is executed or the data are being used. If a programmer writes a program which is too large to reside in main memory, the program is segmented so that only part of it is in main memory at a time. The remainder is put into secondary storage (such as magnetic disc). One of the trends in computer organization is to provide hardware and software which automatically perform segmentation and the related task of bringing program and data segments from secondary storage into main memory when needed. This allows the programmer to ignore main memory limitations and to act as if he had unlimited or virtual memory.

9.1.4. Multiprocessing (AMIE, S '96)

A computer system which contains one CPU is called uniprocessor system. A computer system that contains two or more CPUs is called a multiprocessor system (Fig. 9.7). Multiprocessing is the term used to describe a processing approach in which two or more

independent processors are linked together in a coordinated system. In such a system, instructions from different and independent program can be processed at the same instant

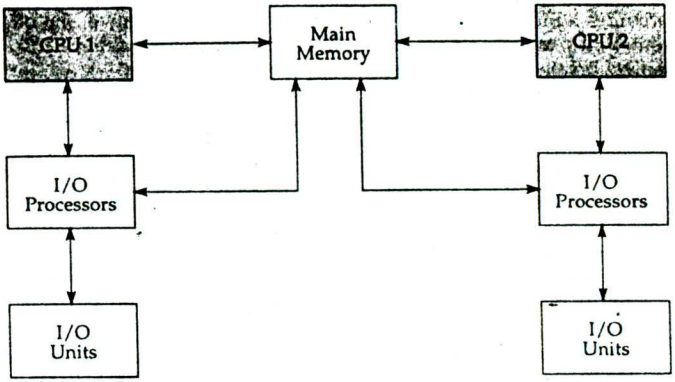


Fig. 9.7 : Basic organisation of a typical multiprocessing system

of time by different processors or the processors may simultaneously execute different instructions from the same program. Thus, in addition to other resources, the processors also become a resource to be managed.

The multiprocessor computers are mainly of two types: Shared memory type and distributed memory type. They are also called tightly coupled and loosely coupled multiprocessor computers, respectively. If the main memory, or major portion thereof, can be directly accessed by all the processors of a multiprocessor system, then the system is referred to as shared memory type. The shared portion of the main memory is called global memory. A small local memory may exist with each processor. An interconnecting network is provided to allow a processor to access the global memory which holds programs and data that are to be shared among the processors. In the distributed memory type, each processor has a large local memory. They may have no global memory or little global memory. A common OS may control all or part of operations of each processor. A related concept with multiprocessor operating systems is distributed operating system. The distributed operating system runs parallel processes with coherent structure of the multiprocessor systems. There is a need of framework for establishing proper communication and cooperation among concurrent processes.

Typical applications are:

- Where some jobs are too large for one CPU to process
- Where it is required to solve a considerable number of problems simultaneously and very speedily
- Where a high degree of protection against breakdowns is required
- In a process control system where a number of minicomputers controlling individual machines are linked to a large computer in charge of the whole process.

It is the job of the operating system to schedule and balance the input, output and processing capabilities of these systems.

The advantages of multiprocessing operating system are listed here:

- It improves the performance of computer systems by allowing parallel processing of segments of programs, which is directly reflected by increased throughput and lowered turnaround time of such systems.

- In addition to the CPU, it also facilitates more efficient utilization of all other devices of the computer system.
- It provides a built-in backup. If one of the CPUs breaks down, the other CPU(s) automatically takes over the complete workload until repairs are made. Thus, a complete breakdown of such system is very rare.

Multiprocessing, however, is not an easy task because of the following reasons:

- A very sophisticated operating system is required to schedule, balance and coordinate the input, output and processing activities of multiple CPUs. The design of such an OS is a time taking job and requires highly skilled computer professionals.
- A large main memory is required for accommodating the sophisticated operating system along with several users programs.
- Such systems are very expensive. In addition to the high initial cost, the regular operation and maintenance of these systems is also a costly affair.

Difference between Multiprogramming and Multiprocessing

Multiprogramming is the interleaved execution of two or more processes by a single CPU computer system. On the other hand, multiprocessing is the simultaneous execution of two or more processes by a computer system having more than one CPU.

Multiprogramming involves executing a portion of one program, then a segment of another, etc., in consecutive time periods. Multiprocessing, however, makes it possible for the system to simultaneously work on several program segments of one or more programs.

9.1.5. Real Time Processing

An operating system which supports a real time application is called a real time operating system. Real time operating system are used in environments when a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. A primary objective is to provide quick event-response. In an airline's reservation system, an enquiry is made for reservation on a particular flight. This enquiry is put into a terminal and transmitted through very fast communication lines into the central computer located far away. The enquiry software with support from the operating system executes this request by checking the status of that particular flight and provides an immediate response.

A computer is sometimes expected to control the operation of a physical system without human intervention. Due to any variations of the physical system the computer has to act instantaneously to correct the variations. Any delay in response from the computer would be disastrous. A system software can be programmed to work instantaneously for critical jobs. For example, rockets are controlled by computer. The computer has to react fast to any variation in the physical quantities of the rocket. The right temperature, velocity, acceleration and pressure of the rocket are fed into the computer. The real-time operating system has been programmed inside the computer to control the physical quantities of the rocket, whenever these show any variation.

9.1.6. Network Operating System

An interesting development that began taking place during the mid-1980s is the growth of network of personal computers running NETWORK OPERATING systems. In a network operating system, the users are aware of the existence of multiple computers, and can log into remote machines and copy files from one machine to another. Each machine runs its own local operating system and has its own user (or users).

Network operating systems are not fundamentally different from single-processor operating systems. They obviously need a network interface controller and some low-level software to drive it, as well as programs to achieve remote log in and remote file access, but these additions do not change the essential structure of the operating system.

Some of the common operating systems for micro and large computers are shown in Figs. 9.8 and 9.9.

Operating System	*Description*
Macintosh System Software	Icon-oriented operating system used on Apple Macintosh microcomputers
MS-DOS	The most widely used operating system on IBM-compatible microcomputers
NetWare	The most widely used operating system on local area network (LAN) composed for microcomputers.
OS/2	An operating system designed for use on higher-end IBM and IBM-compatible microcomputers.
PC-DOS	The operating systems used most widely on IBM microcomputers.
UNIX	A multiuser, multitasking operating system used on small computers — popular on RISC-based microcomputers.
Windows	A multitasking operating environment with a graphical user interface used on IBM and IBM-compatible microcomputers.

Fig. 9.8 : Some popular operating systems for microcomputers

Operating System	*Description*
COS, UNICOS	Operating systems used on Cray supercomputers
MCP/AS, OS/1100	Operating systems used on Unisys mainframes
MVS, VM	Operating systems used on IBM mainframes
OS/400	The operating system used most commonly on the IBM AS/400 line of mid-range computers
PICK	A highly portable multiuser, multitasking operating system used on minicomputers and some microcomputers
UNIX	A family of portable multiuser, multitasking operating systems used on minicomputers, microcomputers, and some mainframes

Fig. 9.9 : Some common operating systems for large computing platforms

9.2 POPULAR OPERATING SYSTEMS

MS-DOS Operating System

MS-DOS is a Micro Soft Disk Operating System used to operate with IBM PC compatible machines. The name DOS is used instead of OS, because the program that wake up the OS is designed with the assumption that the files are to be found on the disk.

Following are the main features of MS-DOS:

1. It is a single user, single process operating system.
2. It supports batch processing mode of operation.
3. It provides the file system in a hierarchial organization, i.e., there is a root directory and sub-directories at various intermediate levels and the files at the lowest level.
4. It supports a wide range of system software tools like language compilers, interpreters, editors, etc. It also supports application packages like word processors, spreadsheets, data base management systems and CAD/CAM, etc.

UNIX Operating System

UNIX was developed in early 1970s as a time sharing operating system that lets many users run multiple tasks on one computer. UNIX operating system is so versatile that it is available on various types of computer systems including micro, mini and mainframe computers.

Following are the main features of UNIX:

1. It is a multiuser, multitasking operating system.
2. Its file system is tree-structures (hierarchial) which allows efficient and easy maintenance and implementation.
3. It is written in high level language, C, which lets users modify the kernel and port it to different hardware.

4. It provides tools and utilities for developing more tools and utilities and the application packages.
5. It has facility of networking through modern circuits for electronic mail.

Windows

Windows has made a revolution in the world of software. Windows is a new operating system environment for PCs developed by Microsoft Corporation. It provides graphical interface to users. In this method, the user has little work on the keyboard. The monitor shows everything in the form of pictures and one can easily choose from the options given. The computer can be operated easily with the help of the mouse.

The first version of **Windows** did not work very well. The release of Windows 3.0 enables user to take full advantages of Intel's new 32-bit microprocessor, the 80386, provides the graphical user interface (GUI) to PC users and the ability to load more than one program into memory at a time. It uses point-and-click method to execute commands.

In 1995, Microsoft Corporation released **Windows 95**. It is a 32-bit, preemptive multitasking operating system with a revised GUI.

Windows 98 released in 1998, an improved version of **Windows 95**, which offers better stability, improved Internet connectivity and updated drivers for new peripherals, including DVD-ROM discs.

Windows NT

Microsoft released Windows NT, a 32 bit operating system for PCs in 1993. It has multitasking and multiprocessing capabilities. It is specially designed primarily for powerful workstations and network services.

OS/2

This 32-bit operating system supports multitasking with a point-and-click interface, making it faster than DOS.

Linux

Linux is the fastest-growing operating system for Intel-based personal computers. It includes all the features of UNIX including multitasking, vertical memory, Internet support, multiprocessor support and GUIs.

REVIEW QUESTIONS SET

1. What is an operating system of a computer and what should be its desirable properties?
2. What facilities are provided by an operating system to a user?
3. What do you understand by the term batch processing system?
4. What are the advantages and disadvantages of batch processing operating system?
5. Define multiprogramming. Explain how multiprogramming ensures effective utilisation of main memory and CPU?
6. Explain the term 'Ready', Blocked' and 'Running' in context of multiprogramming.
7. Define the term 'Spooling'.
8. What special hardware features are required in a computer to implement multiprogramming? **(AMIE, W '96)**
9. What is 'Time sharing operating system'? Give an example. **(AMIE, W '94)**
10. What are advantages and disadvantages of time sharing operating system? **(AMIE, W '97)**
11. What do you understand by the term 'Multiprocessing'? What are the advantages of multiprocessing operating system?
12. What are the differences between multiprocessing and multiprogramming? **(AMIE, W '97)**
13. Multiprogramming and time-sharing both involve multiple users in the computers concurrently. What are the basic differences between the two concepts?
14. Explain the term 'Real time processing'. Differentiate between on line and real time processing. **(AMIE, W '97)**
15. Define the terms: (*a*) Throughput, (*b*) Turnaround time, (*c*) Swapping.
16. What are the main features of MS-DOS and UNIX operating system?
17. What hardware support is useful in time-sharing and in multiprocessing? **(AMIE, W '96)**

10

BASIC OPERATIONAL CONCEPTS

10.0. INTRODUCTION

The basic operation in a digital computer is the transfer of information between the various processor level components (CPUs, main memory, I/O or peripheral devices) through a series of wires known as **Bus**. The basic concepts of bus structure and the operations in a digital computer are discussed.

10.1. BUS STRUCTURE

The components of a computer system communicate with one another through a series of wires (lines) known as Bus. The bus may be unidirectional or bidirectional. A bus connected with a unique source and destination is known as a dedicated bus. If n units are interconnected by buses in all possible ways, then the number of dedicated buses required are $n\ (n-1)$. A shared bus is one which can connect one of several sources to one of several destinations. It permits simultaneous transfers between different pairs of devices. The three types of bus are :

1. Data bus
2. Address bus
3. Control bus.

The data lines are used for transmission of data. Hence, the number of such lines in the data bus corresponds to the number of bits in the word. The width of the bus is typically 16 or 32, which, at any instant, represents a 16 or 32 bit binary value.

The address buses are used to specify the source of destination of data. Thus, k bit address bus is required to identify the $2k$ addressable locations in the main memory. Note that this addressing scheme refers both main memory locations and I/O controllers.

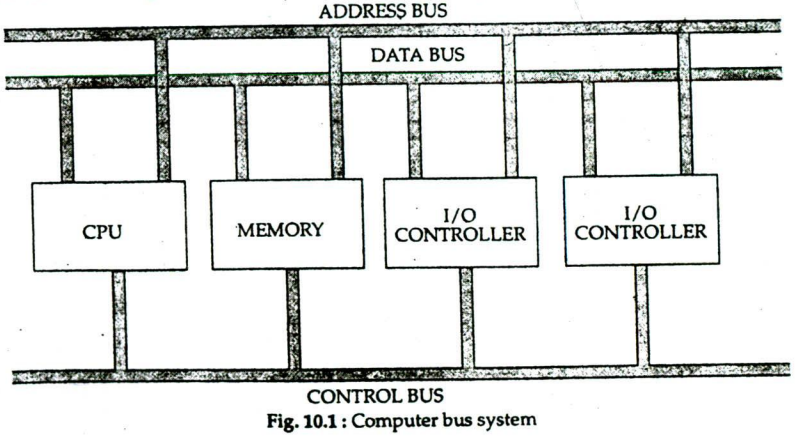


Fig. 10.1 : Computer bus system

Control buses are used to indicate the direction of data transfer and to coordinate the timing of events during a transfer.

The most common arrangement of buses is shown in Fig. 10.1. The I/O controllers are hardware modules used to facilitate communication with I/O devices.

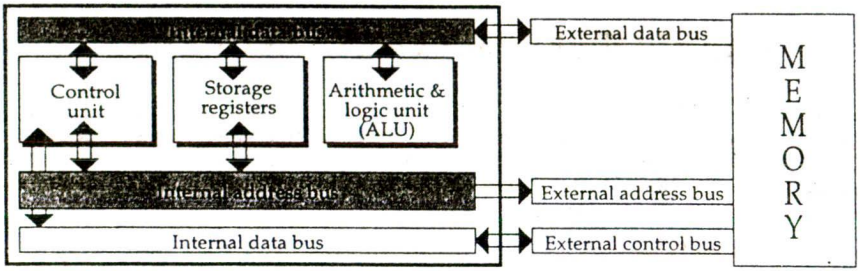


Fig. 10.2 : Interconnection between CPU and memory units

An interconnection between processor and memory units is shown in Fig. 10.2. The data, control and address buses are the equivalent of those described above but are located within CPU and so they are labelled internal buses. They can be viewed as straight extensions of the external buses.

Many bus structures are possible. Two very common types are:

(*i*) Single bus structure, and

(*ii*) Two bus structure.

Figure 10.3 shows a single system of bus (containing data bus, address bus and a few control lines) shared by all the components. Since the bus can be used for only one transfer at a time, it follows that only two units can be actively using the bus at any given time. Bus control lines are used to arbitrate among requesters for use of the bus. The main advantage of the single bus structure is its low cost and flexibility of attaching peripheral devices but it has lower operating speed. This structure is primarily found in small machines, namely, minicomputers and microcomputers.

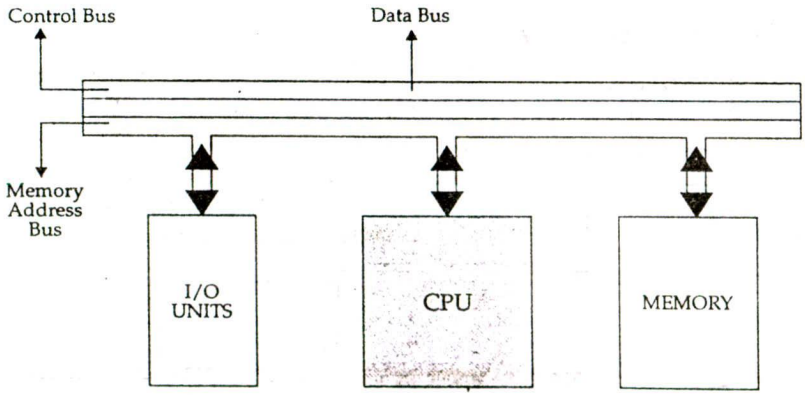


Fig. 10.3 : Single-bus structure

Since communications with I/O devices is a major source of difficulty due to the wide variety of operating characteristics, large computer systems with separate I/O processors frequently use the dual bus system.

10.2. BASIC OPERATIONS

The instructions constituting a program to be executed by a computer are loaded in sequential locations in its main memory. To execute this program, the CPU fetches one instruction at a time and performs the functions specified. Instructions are fetched from successive memory locations except the execution of a branch or a jump instruction.

Major steps in processing an instruction are:

(*i*) Generate the next instruction address
(*ii*) Fetch the instruction
(*iii*) Decode the instruction
(*iv*) Generate the operand address
(*v*) Fetch the operand
(*vi*) Execute the instruction
(*vii*) Store the results
(*viii*) Go to step (*i*) to begin executing the following instruction. Thus, the processing of an instruction has two phases:

1. Instruction Fetch Cycle.
2. Instruction Execution Cycle.

Instruction Fetch Cycle

Program counter (PC) must be initialized to contain the first address of the program stored in memory. When a start is activated, the following sequence is followed:

The contents of PC are transferred into the Memory Address Register (MAR) and a memory read cycle is initiated.

Memory unit sensing the signal, transfer the contents of memory location indicated by MAR to Memory Buffer Register (MBR).

The contents read from memory into MBR is then transferred into Instruction Register (IR).

As soon as the instruction is stored in IR, the operation part and the address part of the instruction is separated. The address part of the instruction is sent to the MAR while its operation part is sent to the control section where it is decoded.

This sequence is called the instruction fetch cycle since it fetches the operation code from memory and places it in the control section. The PC is incremented at the end of instruction cycle. If a branch instruction calls for a transfer to a non-consecutive instruction in the program, the address part of branch instruction is transferred to PC to become the address of the next instruction. The fetch cycle is common to all instructions.

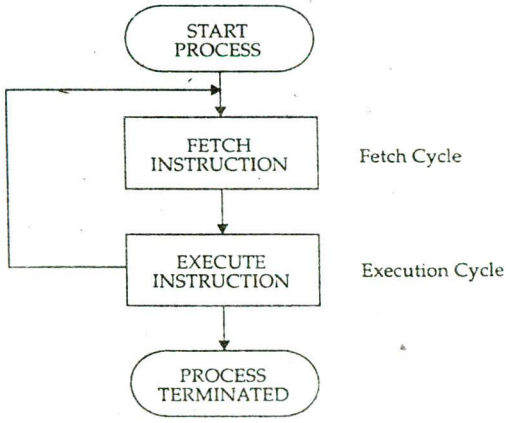


Fig. 10.4 : Simple fetch-execute cycle

The simplest view of processor action is that it consists of a simple fetch-execute cycle, as shown in Fig. 10.4.

Instruction Execution Cycle

Operations as per the decoded instruction is carried out in execution cycle (Fig. 10.5).

If the decoded output is a memory reference instruction, the control unit generates a READ signal which causes the transfer of information from the memory unit whose address is specified by the MAR to the MBR.

The information is transferred from MBR to ALU for performing the required operation, if any, on the information.

If the instruction requires writing information in the memory, the data are first placed in MBR and the control unit generates a write signal, the data in MBR are written in memory in the address specified by MAR.

After completion of the execution cycle, the CPU starts the next instruction cycle and takes the next instruction from the address specified by the PC.

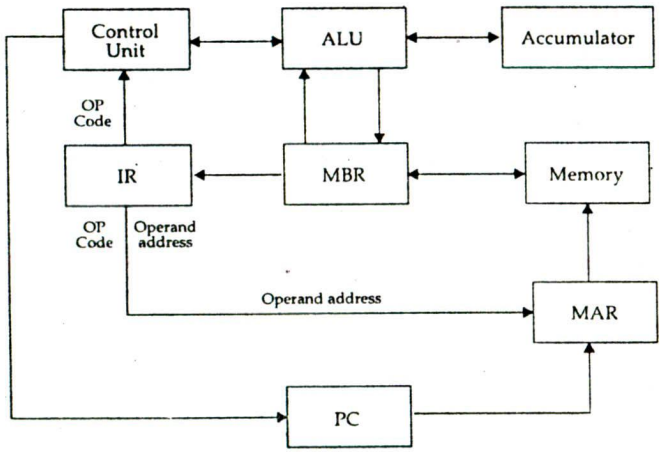


Fig. 10.5 : A processing unit

The simplified sequence of events of execution steps of an addition process may be described as follows:

1. The contents of PC are transferred to MAR and a memory read cycle is initiated.
2. The contents of the location indicated by MAR are transferred to MBR.
3. The opcode of the instruction 'add' and the two operands now in MBR, are transferred to IR.
4. The address of the augend from the first operand part of IR is transferred to MAR.
5. Memory read operation is initiated and the content (augend) of MBR is transferred to accumulator.
6. The address from the second operand part of the IR is transferred to MAR.
7. Memory read operation is initiated.
8. The contents of MBR is transferred to one addend register B (say).
9. The results of arithmetic operation 'add' are produced in accumulator.

REVIEW QUESTIONS SET

1. What is a Bus? What are different buses required in a computer system?
2. Explain with the help of block diagram the working of single bus CPU of a computer. Explain how the following functions are performed for such CPU: **(AMIE, W '97)**
   (*a*) Fetching a word from memory
   (*b*) Storing a word into memory
   (*c*) Performing an arithmetic and logic operation. **(AMIE, S '94)**
3. What are the functions of the following registers in the CPU?
   (*a*) Program Counter
   (*b*) Instruction Register.
   (*c*) Accumulator. **(AMIE, W '96)**
4. Draw and explain the functional block diagram of a digital computer showing the data, address and control bus and their interconnection to the different modules. **(AMIE, W '95)**
5. Describe the sequence of events that take place during the CPU instruction cycle 'Add'. **(AMIE, W '95)**