UNIT 1 THE MEMORY SYSTEM

Structure **Page Nos.**

1.0 INTRODUCTION

In the previous Block, we have touched upon the basic foundation of computers, which include concepts on von Neumann machine, instruction, execution, the digital data representation and logic circuits. In this Block we will define some of the most important component units of a computer, which are the memory unit and the input-output units. In this unit we will discuss various components of the memory system of a computer system. Computer memory is organised into a hierarchy to minimise cost. Also, it does not compromise the overall speed of access. Memory hierarchy include cache memory, main memory and other secondary storage technologies. In this Unit, we will discuss the main memory, the secondary memory and high-speed memories such as cache memory, and the memory system of microcomputer.

1.1 OBJECTIVES

After going though this Unit, you will be able to:

- describe the key characteristics of the memory system;
- distinguish among various types of random access memories;
- describe the latest secondary storage technologies;
- describe the importance of cache memory and other high-speed memories; and
- describe the different memory chips of micro computers.

1.2 THE MEMORY HIERARCHY

Memory in a computer system is required for storage and subsequent retrieval of the instructions and data. A computer system uses a variety of devices for storing these instructions and data that are required for its operation. Normally we classify the information to be stored into two basic categories: Data and Instructions. But what is a memory system?

"The storage devices along with the algorithm or information on how to control and manage these storage devices constitute the memory system of a computer."

A memory system is a very simple system, yet it exhibits a wide range of technology and types. The basic objective of a computer system is to increase the speed of computation. Likewise the basic objective of a memory system is to provide fast, uninterrupted access by the processor to the memory such that the processor can operate at the speed it is expected to work.

But does this kind of technology where there is no speed gap between processor and memory speed exist? The answer is yes, it does. Unfortunately as the access time (time taken by CPU to access a location in memory) becomes less the cost per bit of memory becomes higher. In addition, normally these memories require power supply till the information needs to be stored. Both these things are not very convenient, but on the other hand the memories with smaller cost have very high access time that will result is in slower operation of the CPU. Thus, the cost versus access time anomaly has led to a hierarchy of memories where we supplement fast memories with larger, cheaper, slower memories. These memory units may have very different physical and operational characteristics; therefore, the memory system is very diverse in type, cost, organisation, technology and performance. This memory hierarchy will work only if the frequency of access to the slower memories is significantly less than the faster memories. The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory accessible to the high speed registers and processing logic. Figure 1 illustrates the components of a typical memory system.

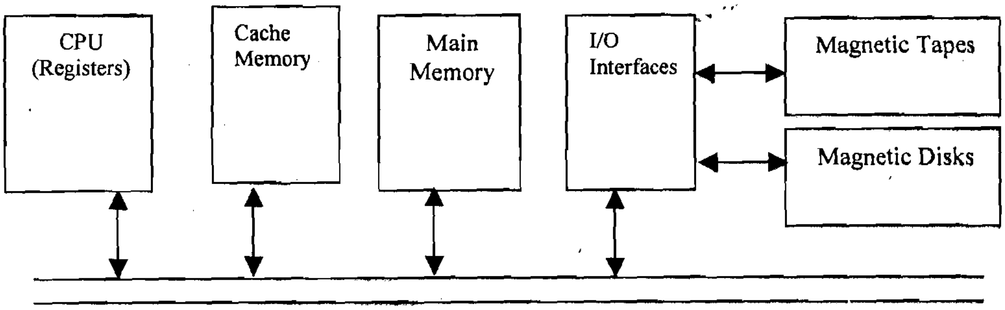


Figure 1: The Memory Hierarchy (Block Diagram)

A typical storage hierarchy is shown in Figure 1 above, Although Figure 1 shows the block diagram, it includes the storage hierarchy:

Register
Cache memory
Main memory
Secondary Storage and
Mass Storage.

As we move up the hierarchy, we encounter storage elements that have faster access time, higher cost per bit stored, and slower access time as a result of moving down the hierarchy. Thus, cache memory generally has the fastest access time, the smallest storage capacity, and the highest cost per bit stored. The primary memory (main memory) falls next in the storage hierarchy list. On-line, direct-access secondary storage devices such as magnetic hard disks make up the level of hierarchy just below the main memory. Off-line, direct-access and sequential access secondary storage devices such as magnetic tape, floppy disk, zip disk, WORM disk, etc. fall next in the storage hierarchy. Mass storage devices, often referred to as archival storage, are at the bottom of the storage hierarchy. They are cost-effective for the storage of very large quantities of data when fast access time is not necessary.

Please note two important points here:

- The size of the memory increases as we move down the hierarchy.
- The quantum of data that is transferred between two consecutive memory layers at a time also increases as we go from a higher to lower side. For example, from main memory to Cache transfer one or few memory words are accessed at a time, whereas in a hard disk to main memory transfer, a block of about 1 Megabyte is transferred in a single access. You will learn more about this in the later sections of the unit.

Let us now discuss various forms of memories in the memory hierarchy in more details.

1.3 RAM, ROM, DRAM, FLASH MEMORY

RAM (Random Access Memory)

The main memory is a random access memory. It is normally organised as words of fixed length. The length of a word is called word length. Each of these memory words has an independent address and each has the same number of bits. Normally the total numbers of words in memory are some power of 2. Some typical memory word sizes are 8 bits, 16 bits, 32 bits etc. The main memory can be both read and written into, therefore it is called read-write memory.

The access time and cycle time in RAMs are constant and independent of the location accessed. How does this happen? To answer this, let us first discuss how a bit can be stored using a sequential circuit. The logic diagram of a binary cell is shown in Figure 2a:

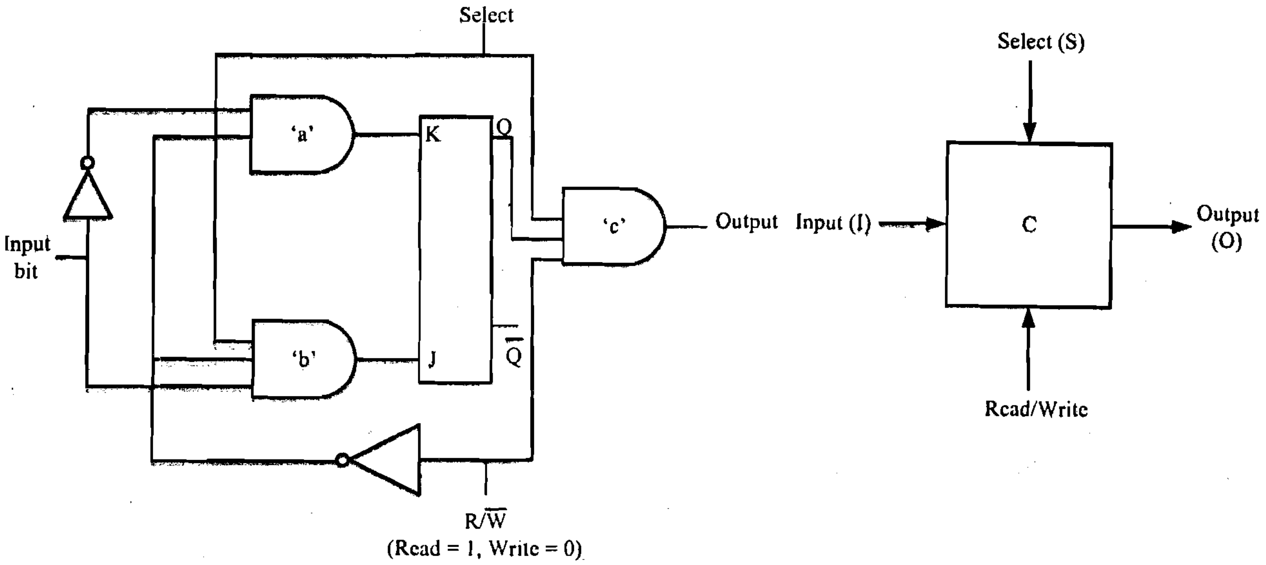


Figure 2 (a): **Logic Diagram of RAM cell**

The construction shown in Figure 2(a) is made up of one JK flip-flop and 3 AND gates. The two inputs to the system are one input bit and read/write signal. Input is fed in complemented form to AND gate 'a'. The read/write signal has a value of 1 if it is a read operation. Therefore, during the read operation the AND gate 'c' has the read/write input as 1. Since AND gate 'a' and 'b' have 0 read/write input, and if the chip is selected i.e. this cell is currently being selected, then output will become equal to the state of flip-flop. In other words the data value stored in flip-flop has been read. In write operation only 'a' and 'b' gates get a read/write value of 1 and they set or clear the JK flip-flop depending on the data input value. If the data input is 0, the flip-flop will go to clear state and if data input is 1, the flip-flop will go to set state. In effect, the input data is reflected in the state of the flip-flop. Thus, we say that the input data has been stored in flip-flop or binary cell.

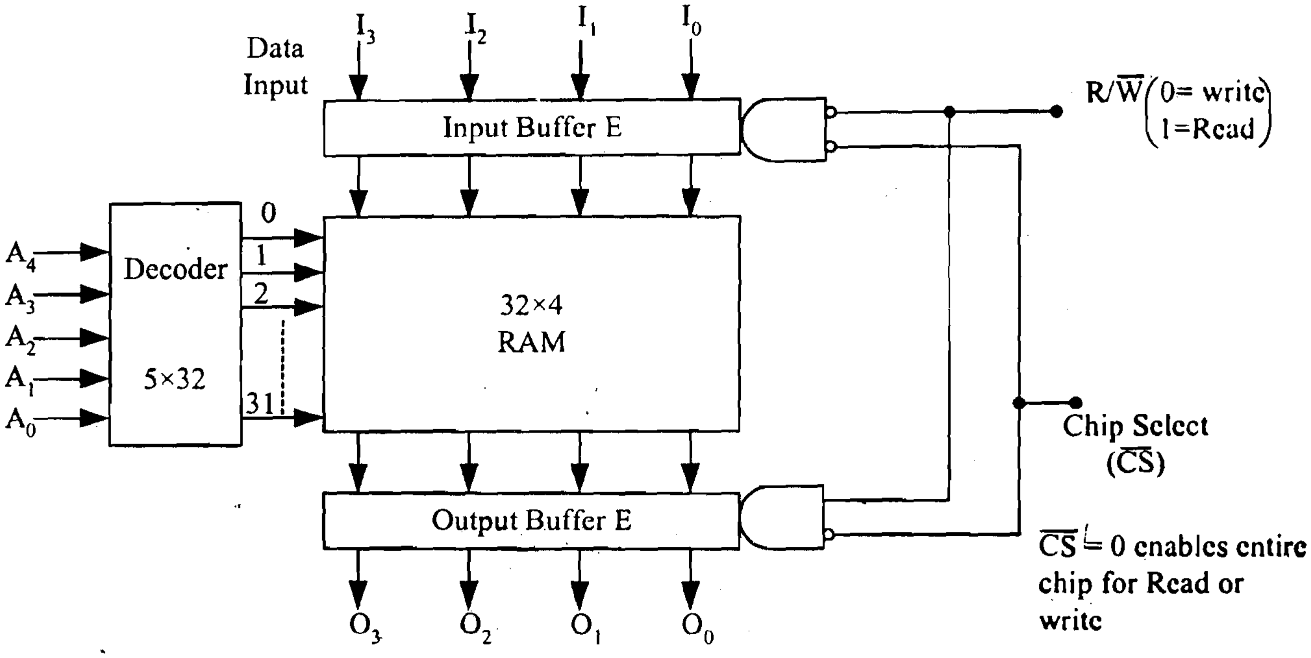


Figure 2 (b):**Internal Organisation of a 32 × 4 RAM**

A 32 × 4 RAM means that this RAM has 32 words, 5 address lines ($2^5 = 32$), and 4 bit data word size. Please note that we can represent a RAM using 2^A×D, where A is the number of address lines and D is the number of Data lines. Figure 2 (b) is the extension of the binary cell to an integrated 32 × 4 RAM circuit where a 5 × 32 bit decoder is used. The 4 bit data inputs come through an input buffer and the 4-bit data output is stored in the output buffer.

A chip select ($\overline{CS}$) control signal is used as a memory enable input. When CS = 0 that is $\overline{CS}$ = 1, it enables the entire chip for read or write operation. A R/W signal can be used for read or write operation. The word that is selected will determine the overall output. Since all the above is a logic circuit of equal length that can be accessed in equal time, thus, the word RAM.

DRAM (Dynamic Random Access Memory)

RAM technology is divided into two technologies: dynamic and static. A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors. The presence or absence of charge on capacitor is interpreted as binary 1 or 0. Because capacitors have a natural tendency to discharge, dynamic RAM requires periodic charge refreshing to maintain data storage. The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.

Figure 3(a) is a typical DRAM structure for an individual cell that stores one bit. The address line is activated when the bit value from this cell is to be read or written. The transistor acts as a switch that is closed (allowing current to flow) if a voltage is applied to the address line and open (no current flows) if no voltage is present on the address line.

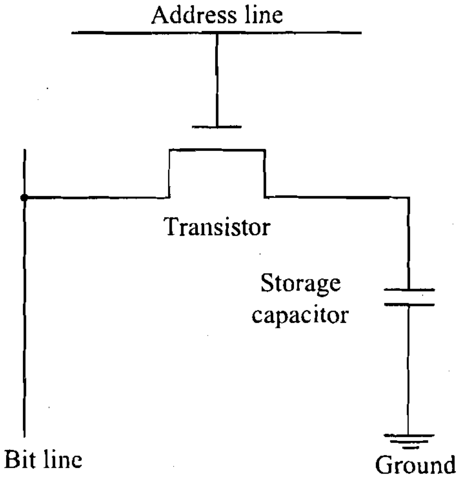


Figure 3(a): DRAM Cell

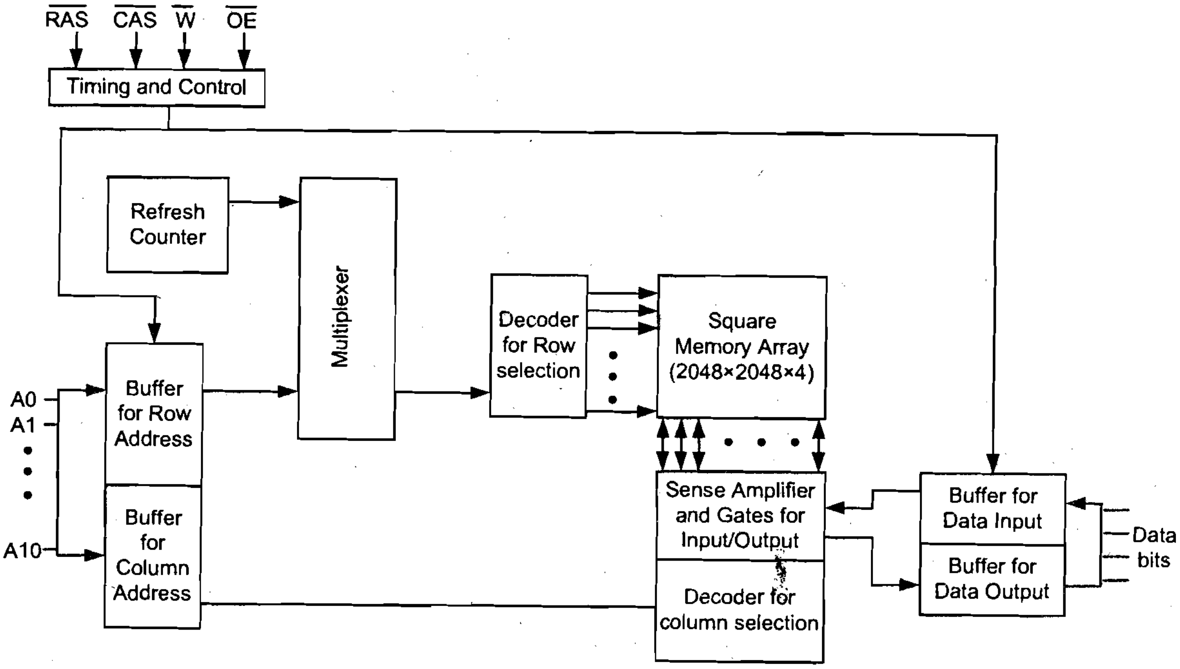

RAS
CAS
W
OE
Timing and Control
Refresh Counter
Multiplexer
Decoder for Row selection
Square Memory Array (2048×2048×4)
A0
A1
A10
Buffer for Row Address
Buffer for Column Address
Sense Amplifier and Gates for Input/Output
Decoder for column selection
Buffer for Data Input
Buffer for Data Output
Data bits

Figure 3(b): Typical 16 Megabit DRAM

For the write operation (please refer to Figure 3 (a), a voltage signal is applied to the bit line; a high voltage represents 1, and a low voltage represents 0. A signal is then applied to the address line, allowing a charge to be transferred to the capacitor.

For the read operation, when the address line is selected, the transistor turns on and the charge stored on the capacitor is fed out onto a bit line and to the sense amplifier. The sense amplifier compares the capacitor voltage to a reference value and determines if the cell contains logic 1 or logic 0. The read out from the cell discharges the capacitor, which **must be restored** to complete the operation.

Although the DRAM cell is used to store a single bit (0 or 1), it is essentially an analog device. The capacitor can store any charge value within a range; a threshold value determines whether the charge is interpreted as 1 or 0.

Organisation of DRAM Chip

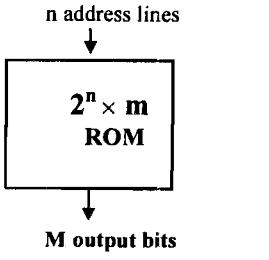
The Figure 3(b) is a typical organisation of 16 mega bit DRAM. It shows a typical organisation of 2048 × 2048 × 4 bit DRAM chip. The memory array in this organisation is a square array that is (2048 × 2048) words of 4 bits each.

Each element, which consists of 4 bits of array, is connected by horizontal row lines and vertical column lines. The horizontal lines are connected to the select input in a row, whereas the vertical line is connected to the output signal through a sense amplifier or data in signal through data bit line driver. Please note that the selection of input from this chip requires:

- Row address selection specifying the present address values A0 to A10 (11 address lines only). For the rows, it is stored in the row address buffer through decoder.
- Row decoder selects the required row.
- The column address buffer is loaded with the column address values, which are also applied to through A0 to A10 lines only. Please note that these lines should contain values for the column.
- This job will be done through a change in external signal $\overline{\text{RAS}}$ (Row address Strobe) because this signal is high at the rising edge of the clock.
- $\overline{\text{CAS}}$ (Column address Strobe) causes the column address to be loaded with these values.
- Each column is of 4 bits, that is, those require 4 bit data lines from input/output buffer. On memory write operation data in bit lines being activated while on read sense lines being activated.
- This chip requires 11 address lines (instead of 22), 4 data in and out lines and other control lines.
- As there are 11 row address lines and 11 column address lines and each column is of 4 bits, therefore, the size of the chip is $2^{11} \times 2^{11} \times 4 = 2048 \times 2048 \times 4 = 16$ mega bits. On increasing address lines from 11 to 12 we have $2^{12} \times 2^{12} \times 4 = 64$ mega bits, an increase of a factor of 4. Thus, possible sizes of such chips may be 16K, 256K, 1M, 4M, 16M, and so on.
- Refreshing of the chip is done periodically using a refresh counter. One simple technique of refreshing may be to disable read-write for some time and refresh all the rows one by one.

ROM (Read-Only Memory)

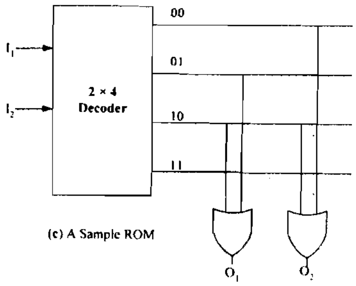
A ROM is essentially a memory or storage device in which a fixed set of binary information is stored. A block diagram of ROM is as shown in Figure 4(a). It consists of n input lines and m output lines. Each bit combination of the input variables is called an ***address***. Each bit combination that comes out of the output lines is called a ***word***. The number of bits per word is equal to the number of output lines m. The number of distinct addresses possible with n input variables is $2^{n.}$



(a) **ROM Block diagram**

Input		**Output**	
I_1	I_2	O_1	O_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

(b) Truth table



(c) A Sample ROM

Figure 4: ROM

A ROM is characterised by the number of words (2^n) and the number of bits (m) per word. For example, a 32 × 8 ROM which can be written as $2^5 \times 8$ consists of 32 words of 8 bit each, which means there are 8 output lines and 32 distinct words stored in the unit. There are only 5 input lines because $32 = 2^5$and with 5 binary variables, we can specify 32 addresses.

A ROM is basically a combinational circuit and can be constructed as shown in Figure 4(c). On applying an Input $I_1 = 0$, $I_2 = 0$, the 00 line of the decoder is selected and we will get $O_1 = 0$ and $O_2 = 1$; on applying $I_1 = 0$ and $I_2 = 1$ we will get $O_1 = 1$ AND $O_2 = 0$. This same logic can be used for constructing larger ROMs.

ROMs are the memories on which it is not possible to write the data when they are on-line to the computer. They can only be read. This is the reason why it is called read-only memory (ROM). Since ROM chips are non-volatile, the data stored inside a ROM are not lost when the power supply is switched off, unlike the case of a volatile RAM chip. ROMs are also known as permanent stores.

The ROMs can be used for storing micro-programs, system programs and subroutines. ROMs are non-volatile in nature and need not be loaded in a secondary storage device. ROMs are fabricated in large numbers in a way where there is no room for even a single error. But, this is an inflexible process and requires mass production. Therefore, a new kind of ROM called PROM was designed which is also non-volatile and can be written only once and hence the name Programmable ROM(PROM). The supplier or the customer can perform the writing process in PROM electrically. Special equipment is needed to perform this writing operation. Therefore, PROMs are more flexible and convenient than ROMs.

The ROMs / PROMs can be written just once, but in both the cases whatever is written once cannot be changed. But what about a case where you read mostly but write only very few times? This led to the concepts of read mostly memories and the best example of these are EPROMs (Erasable PROMs) and EEPROMs (Electrically Erasable PROMs).

The EPROMs can be read and written electrically. But, the write operation is not simple. It requires erasure of whole storage cells by exposing the chip to ultra violet light; thus bringing them to the same initial state. Once all the cells have been brought to same initial state, then the EPROM can be written electrically. EEPROMs are becoming increasingly popular, as they do not require prior erasure of previous contents. However, in EEPROMS the writing time is considerably higher than the reading time. The biggest advantage of EEPROM is that it is non-volatile memory and can be updated easily, while the disadvantages are the high cost and at present they are not completely non-volatile and the write operation takes considerable time. But all these advantages are disappearing with growth in technology. In general, ROMs are made of cheaper and slower technology than RAMs.

Flash Memory

This memory is another form of semiconductor memory, which was first introduced in the mid-1980. These memories can be reprogrammed at high speed and hence the name flash. This is a type of non-volatile, electronic random access memory.

Basically this memory falls in between EPROM and EEPROM. In flash memory the entire memory can be erased in a few seconds by using electric erasing technology. Flash memory is used in many I/O and storage devices. *Flash memory is also used to store data and programming algorithms in cell phones, digital cameras and MP3 music players.*

Flash memory serves as a hard drive for consumer devices. Music, phone lists, applications, operating systems and other data are generally stored on stored on flash chips. *Unlike the computer memory, data are not erased when the device is turned off.*

There are two basic kinds of flash memory:

Code Storage Flash made by Intel, AMD, Atmel, etc. It stores programming algorithms and is largely found in cell phones.

Data Storage Flash made by San Disk, Toshiba, etc. It stores data and comes in digital cameras and MP3 players.

The feature of semiconductor memories are summarised in the Figure 5.

Memory	**Category**	**Erasure**	**Write Mechanism**	**Volatility**
Random-access Memory (RAM)	Read–write memory	Electrically, byte level	Electrically	Volatile
Read –only Memory (ROM)	Read–only memory	Not possible	Masks	Non-volatile
Programmable ROM (PROM)	Read–only memory	Not possible	Electrically	Non-volatile
Erasable PROM (EPROM)	Read-mostly memory	UV light chip level	Electrically	Non-volatile
Electrically Erasable (EEPROM)	Read-mostly memory	Electrically, byte level	Electrically	Non-volatile
Flash memory	Read-mostly memory	Electrically, block level	Electrically	Non-volatile

Figure 5: Features of Semiconductor Memories

1.4 SECONDARY MEMORY AND CHARACTERISTICS

It is desirable that the operating speed of the primary storage of a computer system be as fast as possible because most of the data transfer to and from the processing unit is via the main memory. For this reason, storage devices with fast access times, such as semiconductors, are generally used for the design of primary storage. These high-speed storage devices are expensive and hence the cost per bit of storage is also high for a primary storage. *But the primary memory has the following limitations*:

a) **Limited capacity:** The storage capacity of the primary storage of today 's computers is not sufficient to store the large volume of data handled by most of the data processing organisations.

b) **Volatile:** The primary storage is volatile and the data stored in it is lost when the electric power is turned off. However, the computer systems need to store data on a permanent basis for several days, months or even several years.

The result is that an additional memory called secondary storage is used with most of the computer systems. Some popular memories are described in this section.

1.4.1 Hard Disk Drive

This is one of the components of today's personal computer, having a capacity of the order of several Giga Bytes and above. A magnetic disk has to be mounted on a disk drive before it can be used for reading or writing of information. A disk drive contains all the mechanical, electrical and electronic components for holding one or more disks and for reading or writing of information on it. That is, it contains the central shaft on which the disks are mounted, the access arms, the read/write head and the motors to rotate the disks and to move the access arms assembly. Now-a-days, the disk drive assembly is packed in very small casing although having very high capacity. Now let us know about what a magnetic disk is.

Magnetic Disk

A disk is circular platter constructed of nonmagnetic material, called the substrate, coated with a magnetisable material. This is used for storing large amount of data. Traditionally, the substrate has been an aluminium or aluminium alloy material; more recently, glass substrates have been introduced. The glass substrate has a number of benefits, including the following:

- Improvement in the uniformity of the magnetic film surface to increase disk reliability.
- A significant reduction in overall surface to help reduce read-write errors.
- Ability to support lower fly heights.
- Better stiffness to reduce disk dynamics.
- Greater ability to withstand shock and damage.

Magnetic Read and Write Mechanisms

Data are recorded on and later retrieved from the disk via a conducting coil named the head; in many systems there are two heads, a read head and a write head.
During a read or write operation, the head is stationary while the platter rotates beneath it.

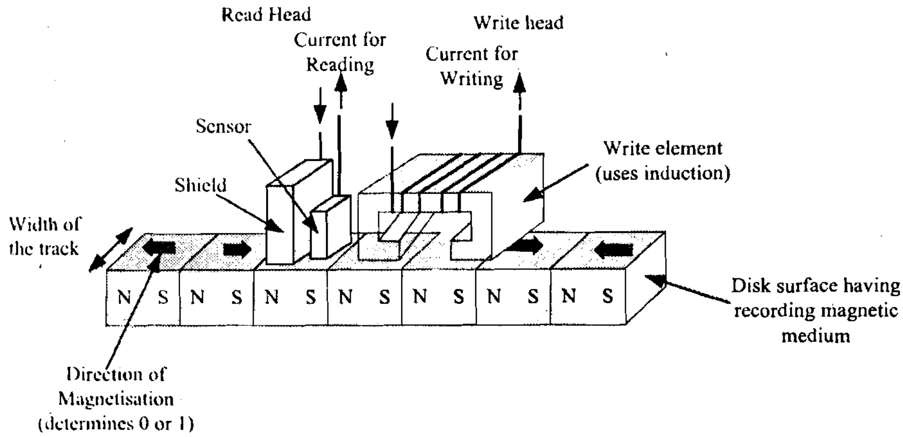

Read Head
Current for Reading
Write head
Current for Writing
Sensor
Shield
Write element (uses induction)
Width of the track
N S N S N S N S N S N S N S
Disk surface having recording magnetic medium
Direction of Magnetisation (determines 0 or 1)

Figure 6: Read /write Heads

The write mechanism is based on the fact that electricity flowing through a coil produces a magnetic field. Pulses are sent to the write head, and magnetic patterns are recorded on the surface below, with different patterns for positive and negative currents. The write head itself is made of easily magnetisable material and is in the shape of a rectangular doughnut with a gap along one side and a few turns of conducting wire along the opposite side (Figure 6). An electric current in the wire induces a magnetic field across the gap, which in turn magnetizes a small area of the recording medium. Reversing the direction of the current reverses the direction of the magnetization on the recording medium.

The traditional read mechanism is based on the fact that a magnetic field moving relative to a coil produces an electrical current in the coil. When the surface of the disk passes under the head, it generates a current of the same polarity as the one already recorded. The structure of the head for reading is in this case essentially the same as for writing and therefore the same head can be used for both. Such single heads are used in floppy disk systems and in older rigid disk systems.

Data Organization and Formatting

The head is a relatively small device capable of reading from or writing to a portion of the platter rotating beneath it. This gives rise to the organization of data on the platter in a concentric set of rings, called tracks; each track is of the same width as the head. There are thousands of tracks per surface.

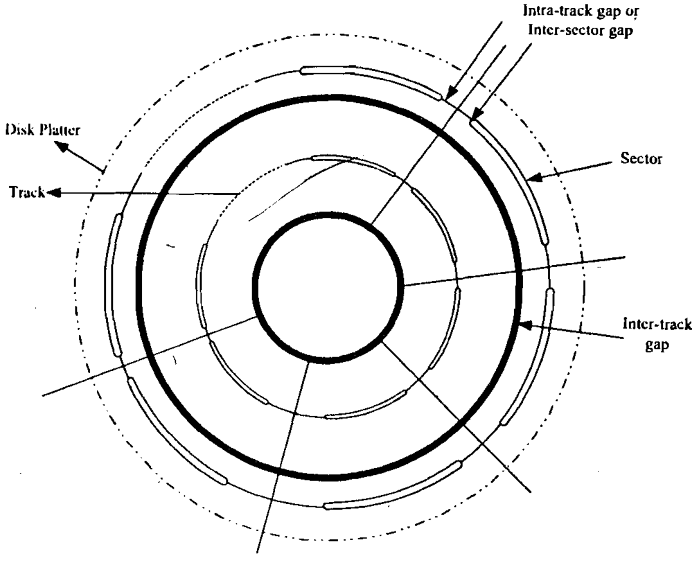


Figure 7: Layout of Magnetic Disk

Figure 7 depicts this data layout. Adjacent tracks are separated by gaps. This prevents, or at least minimizes, errors due to misalignment of the head. Data are transferred to and from the disk in sectors. To identify the sector position normally there may be a starting point of a track and a starting and end point of each sector. But the question is how is a sector of a track recognised? A disk is formatted to record control data on it such that some extra data are stored on it for identical purpose. This control data is accessible only to the disk drive and not to the user. Please note that in Figure 7 as we move away from the centre of a disk the physical size of the track is increasing. Does it mean we store more data on the outside tracks? No. A disk rotates at a constant angular velocity. But, as we move away from centre the liner velocity is more than the liner velocity nearer to centre. Thus, the density of storage of information decreases as we move away from the centre of the disk. This results in larger physical sector size. Thus, all the sectors in the disk store same amount of data.

An example of disk formatting is shown in Figure 8. In this case, each track contains 30 fixed–length sectors of 600 bytes each. Each sector holds 512 bytes of data plus control information useful to the disk controller. The ID field is a unique identifier or address used to locate a particular sector. The SYNC byte is a special bit pattern that delimits the beginning of the field. The track number identifies a track on a surface. The head number identifies a head, because this disk has multiple surfaces. The ID and data fields each contain an error-detecting code.

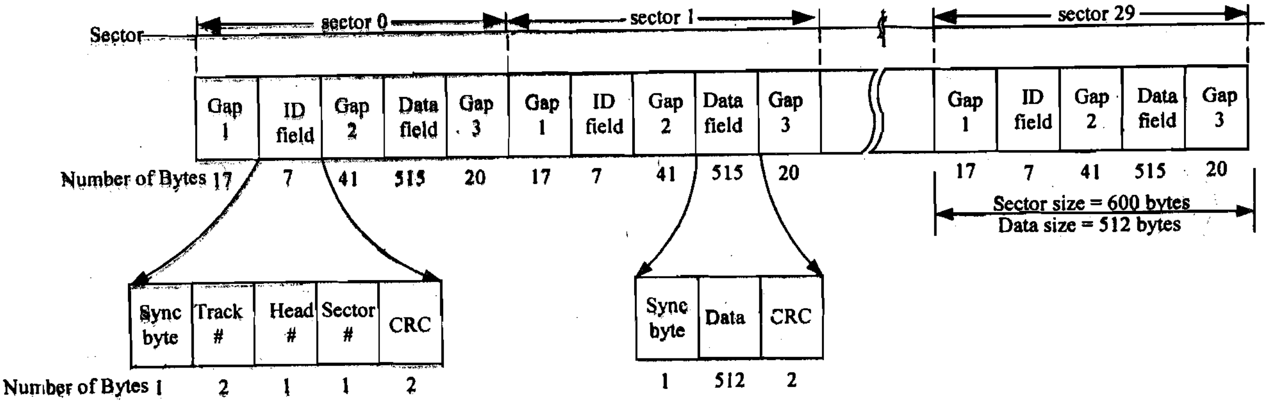


Figure 8: A typical Track Format for Winchester Disk

Physical Characteristics

Figure 9 lists the major characteristics that differentiate among the various types of magnetic disks. First, the head may either be fixed or movable either respect to the radial direction of the platter. In a fixed-head disk, there is one read-write head per track. All of the heads are mounted on a rigid arm that extends across all tracks; such systems are rare today. In a movable-head disk, there is only one read-write head. Again, the head is mounted on an arm. Because the head must be able to be positioned above any track, the arm can be extended or retracted for this purpose.

Head Motion	Platters
Fixed head (one per track) Moveable head (one per surface)	Single platter Multiple platter
Disk Portability	**Head mechanism**
Non-removable Removable disk	Disk Contact (floppy) Fixed gap Aerodynamic gap (Winchester)
Sides	
Single sided Double sided	

Figure 9: Physical characteristics of Disk Systems

The disk itself is mounted in a disk drive, which consists of the arm, a shaft that rotates the disk, and the electronics needed for input and output binary data. A non-removable disk is permanently mounted in the disk drive; the hard disk in a personal computer is a non-removable disk. A removable disk can be removed and replaced with another disk. The advantage of the latter type is that unlimited amounts of data are available with a limited number of disk systems. Furthermore, ZIP cartridge disks are examples of removable disks. Figure 10 shows other components of the disks.

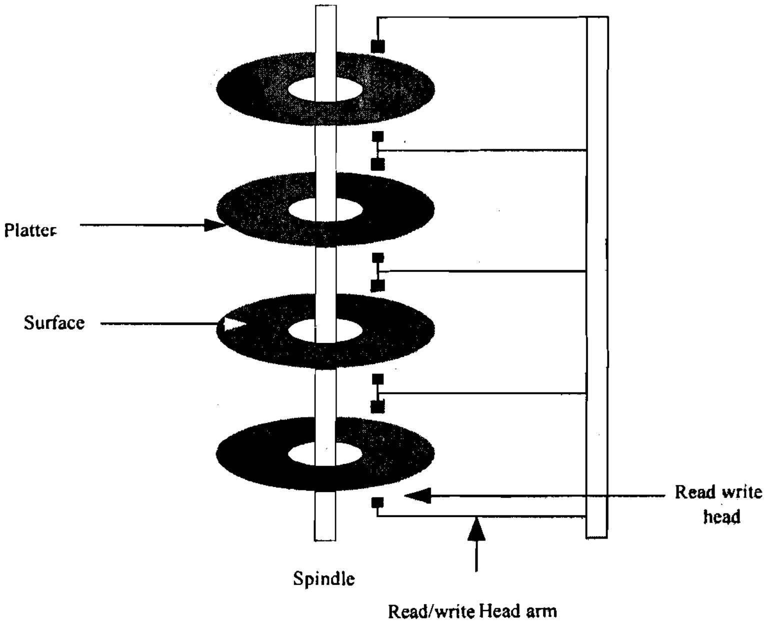


Figure 10: The Disk Components

The head mechanism provides a classification of disks into three types. Traditionally, the read-write head has been positioned at a fixed distance above the platter, allowing an air gap. At the other extreme is a head mechanism that actually comes into physical contact with the medium during a read or write operation. This mechanism is used with the floppy disk, which is a small, flexible platter and the least expensive type of disk.

To understand the third type of disk, we need to comment on the relationship between data density and the distance of head from the surface. The head generates or senses an electromagnetic field of sufficient magnitude to write and read properly. The narrower the head is, the closer it must be to the platter surface to function. A narrower head means narrower tracks and therefore greater data density, which is desirable. However, the closer the head is to the disk, the greater are the risks of errors from impurities or imperfections.

To push the technology further, the Winchester disk was developed. Winchester heads are used in sealed drive assemblies that are almost free of contaminants. They are designed to operate closer to the disk's surface than conventional rigid disk heads, thus allowing greater data density. The head is actually an aerodynamic foil that rests lightly on the platter's surface when the disk is motionless. The air pressure generated by a spinning disk is enough to make the foil rise above the surface. The resulting non-contact system can be engineered to use narrower heads that operate closer to the platter's surface than conventional rigid disk heads.

Accessing the Disk Data

Disks operate is semi-random mode of operation and normally are referenced block wise. The data access time on a disk consists of two main components:

- **Seek time:** Time to position the head on a specific track. On a fixed head disks it is the time taken by the electronic circuit to select the required head while in movable head disks it is the time required to move the head to a particular track.
- **Latency time:** This is the time required by a sector to reach.below the read/write head. On an average it is half of the time taken for a rotation by the disk.

In addition to the seek and latency time, the time taken to transfer a (read/write) block of words can be considered but normally it is too small in comparison to latency and seek time and in general the disk access time is considered to be the sum of seek time and latency time. Since access time of disks is large, therefore it is advisable to read a sizeable portion of data in a single go and that is why the disks are referenced block wise. In fact, you will find that in most of the computer system, the input/output involving disk is given a very high priority. The basic reason for such priority is the latency time that is needed once the block which is to be read passes below the read-write head; it may take time of the order of milliseconds to do that again, in turn delaying the Input /Output and lowering the performance of the system.

1.4.2 Optical Memories

In 1983, one of the most successful consumer products of all times was introduced: the compact disk (CD) digital audio system. This CD was a non-erasable disk that could store more than 60 minutes of audio information on one side. The huge commercial success of this CD enabled the development of low-cast optical-disk storage technology that has revolutionised computer data storage. A variety of optical-disk systems has been introduced. We briefly review each of these.

Compact Disk ROM (CD-ROM)

Both the audio CD and the CD-ROM (compact disk read-only memory) share a similar technology. The main difference is that CD-ROM players are more rugged and have error correction devices to ensure that data are properly transferred from disk to computer. Both types of disk are made the same way. The disk is formed from a resin, such as polycarbonate. Digitally recorded information (either music or computer data) is imprinted as a series of microscopic pits on the surface of the polycarbonate. The pitted surface is then coated with a highly reflective surface, usually aluminium. This shiny surface is protected against dust and scratches by a topcoat of clear acrylic. Finally, a label can be silk-screened onto the acrylic.

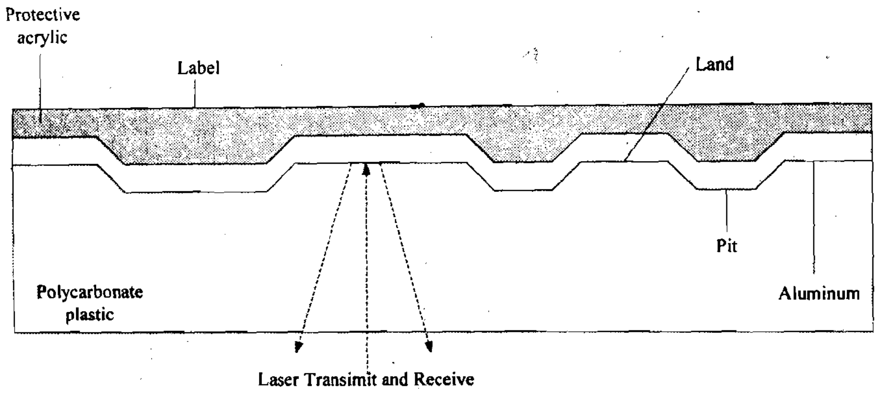
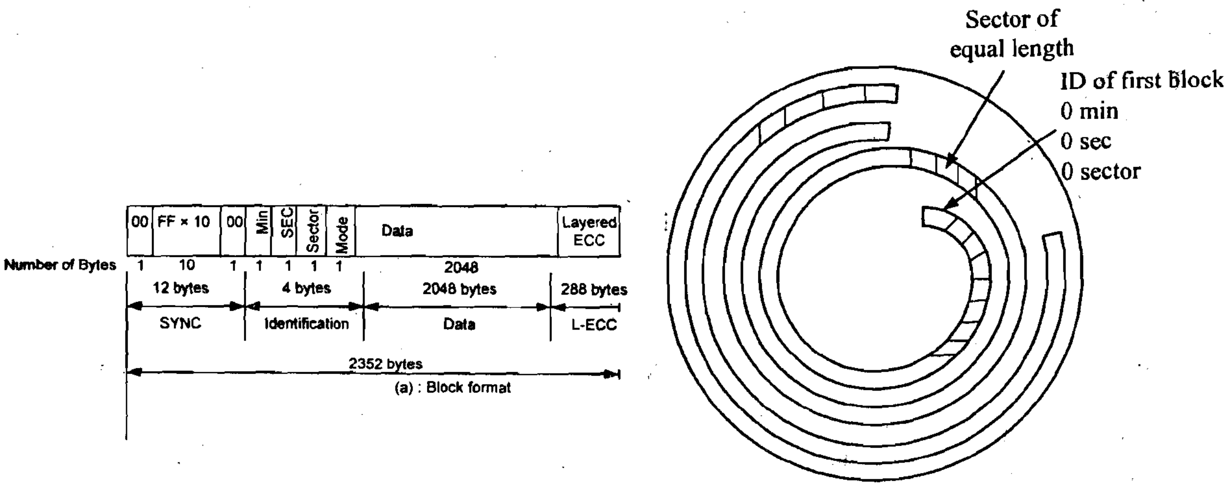


Figure 11: The CD Surface and Operation

Information is retrieved from a CD or CD-ROM by a low-powered laser housed in an optical-disk player, or drive unit. The laser shines through the clear polycarbonate while a motor spins the disk past it (Figure 11). The intensity of the reflected light of the laser changes as it encounters a pit. Specifically, if the laser beam falls on a pit, the light scatters and a low intensity is reflected back to the source. The areas between pits are called ***lands***. A land is a smooth surface, which reflects back at a higher intensity. The change between pits and lands is detected by a **photo sensor** and converted into a digital signal. The sensor tests the surface at regular intervals. The beginning or end of a pit represents a 1; when no change in elevation occurs between intervals, a 0 is recorded.

Data on the CD-ROM are organised as a sequence of blocks. A typical block format is shown in Figure 12 (a). It consists of the following fields:

- **Sync:** The sync field identifies the beginning of a block. It consists of a byte of all 0s, 10 bytes of all 1s, and bytes of all 0s.
- **Header:** The header contains the block address and the mode byte. Mode 0 specifies a blank data field; mode 1 specifies the use of an error-correcting code and 2048 bytes of data; mode 2 specifies 2336 bytes of user data with no error correcting code.
- **Data:** User data.
- **Auxiliary:** Additional user data in mode 2. In mode 1, this is a 288-byte error correcting code.



(a) Block Format

(b) CD-ROM Layout

Figure 12: Block Format and Disk Layout on CD-ROM

But what is the Min (Minute), Sec (Second) and Sector fields in the Header field? The sectors of CD-ROM are not organised like the sectors in hard disks (Please refer Figure 12(b)). Rather, they are all equal length segments. If we rotate the CD drive at constant speed the linear velocity of disk surface movement will be higher at the outer side than that of the centre portions. To offset this liner speed gap, either we store less data on the outer sectors or we reduce the speed of rotation while reading outer tracks. The CD follows the later approach, that is, instead of moving the CD drive at constant velocity, it is rotated at variable velocity. The speed or rotation of disk reduces as we move away from the centre such that the sector's can be read in constant time. This method of reading is called Constant Liner Velocity (CLV).

CD-ROM is appropriate for the distribution of large amounts of data to a large number of users. CD-ROMs are a common medium these days for distributing information. Compared with traditional hard disks, *the CD-ROM has three advantages*:

1. Large data/information storage capability.
2. The optical disk together with information stored on it can be mass replicated inexpensively, unlike a magnetic disk. The database on a magnetic disk has to be reproduced by copying data from one disk to second disk, using two disk drives.
3. The optical disk is removable, allowing the disk itself to be used for archival storage. Most magnetic disks are non-removable. The information on non-removable magnetic disks must first be copied on tape before the disk drive / disk can be used to store new information.

The disadvantages of CD- ROM are as follows:

1. It is read-only and cannot be updated.
2. It has an access time much longer than that of a magnetic disk drive (as it employs CLV), as much as half a second.

Compact Disk Recordable (CD-R)

To accommodate applications in which only one or a small number of copies of a set data is needed, the write-once read-many CD, known as the CD Recordable (CD-R), has been developed. For CD-R a disk is prepared in such a way that it can be subsequently written once with a laser beam of modest intensity. Thus, with a somewhat more expensive disk controller than for CD-ROM, the customer can write once as well as read the disk.

The CD-R medium is similar to but not identical to that of a CD or CD-ROM. For CDs and CD-ROMs, information is recorded by the pitting of the surface of the medium, which changes reflectivity. For a CD-R, the medium includes a dye layer. The resulting disk can be read on a CD-R drive or a CD-ROM drive.

The CD-R optical disk is attractive for archival storage of documents and files. It provides a permanent record of large volumes of user data.

Compact Disk Rewritable (CD-RW)

The CD-RW optical disk can be repeatedly written and overwritten, as with a magnetic disk. Although a number of approaches have been tried, the only pure optical approach that has proved attractive is called phase change. The phase change disk uses a material that has two significantly different reflectivities in two different phase states. There is an amorphous state, in which the molecules exhibit a random orientation and which reflects light poorly; and a crystalline state, which has a smooth surface that reflects light well. A beam of laser light can change the material from one phase to the other. The primary disadvantage of phase change optical disks is that the material eventually and permanently loses its desirable properties. Current materials can be used for between 500,000 and 1,000,000 erase cycles.

The CDRW has the obvious advantage over CD-ROM and CD-R that it can be rewritten and thus used as a true secondary storage. As such, it competes with magnetic disk. A key advantage of the optical disk is that the engineering tolerances for optical disks are much less severe than for high-capacity magnetic disks. Thus, they exhibit higher reliability and longer life.

Digital Versatile Disk (DVD)

With the capacious digital versatile disk (DVD), the electronics industry has at last found an acceptable replacement for the videotape used in videocassette recorders (VCRs) and, more important for this discussion, replace the CD-ROM in personal computers and servers. The DVD has taken video into the digital age. It delivers movies with impressive picture quality, and it can be randomly accessed like audio CDs, which DVD machines can also play. Vast volumes of data can be crammed onto the disk, several times as much as a CD-ROM. With DVD's huge storage capacity and vivid quality, PC games will become more realistic and educational software will incorporate more video.

1.4.3 CCDs, Bubble Memories

Charge-coupled Devices (CCDs)

CCDs are used for storing information. They have arrays of cells that can hold charge packets of electron. A word is represented by a set of charge packets, the presence of each charge packet represent the bit-value 1. The charge packets do not remain stationary and the cells pass the charge to the neighbouring cells with the next clock pulse. Therefore, cells are organized in tracks with a circuitry for writing the data at the beginning and a circuitry for reading the data at the end. Logically the tracks (one for each bit position) may be conceived as loops since the read circuitry passes the information back to the write circuit, which then re-creates the bit values in the track unless new data is written to the circuit.

These devices come under the category of semi-random operation since the devices must wait till the data has reached the circuit for detection of charge packets. The access time to these devices is not very high. At present this technology is used only in specific applications and commercial products are not available.

Magnetic Bubble Memories

In certain material such as garnets on applying magnetic fields certain cylindrical areas whose direction of magnetization is opposite to that of magnetic field are created. These are called magnetic bubbles. The diameter of these bubbles is found to be in the range of 1 micrometer. These bubbles can be moved at high speed by applying a parallel magnetic field to the plate surface. Thus, the rotating field can be generated by an electromagnetic field and no mechanical motion is required.

In these devices deposition of a soft magnetic material called Perm alloy is made as a predetermined path, thus making a track. Bubbles are forced to move continuously in a fixed direction on these tracks. In these memories the presence of a bubble represents a 1 while absence represents a 0 state. For writing data into a cell, a bubble generator to introduce a bubble or a bubble annihilator to remove a bubble, are required. A bubble detector performs the read operation. Magnetic bubble memories having capacities of 1M or more bits per chip have been manufactured. The cost and performance of these memories fall between semi-conductor RAMs and magnetic disks.

These memories are non-volatile in contrast to semi-conductor RAMs. In addition, since there are no moving parts, they are more reliable than a magnetic disk. But these memories are difficult to manufacture and difficult to interface with in conventional processors. These memories at present are used in specialized applications, e.g., as a secondary memory of air or space borne computers, where extremely high reliability is required.

Check Your Progress 1

1. **State True or False:** T/F

   a) Bubble memories are non-volatile. □

   b) The disadvantage of DRAM over static RAM is the need to refresh the capacitor charge every few milliseconds. □

   c) Flash memory is a volatile RAM. □

2. **Fill in the blanks:**

   a) The EPROM is ___________ erasable and _____________ programmable.

   b) __________ memory requires a rechargeable cycle in order to retain its information.

   c) Memory elements employed specifically in computer memories are generally ___________ circuits.

3. Differentiate among RAM, RÔM, PROM and EPROM.

   ....

4. What is a flash memory? Give a few of its typical uses.

   ....

5. A memory has a capacity of 4K × 8

   (a) How many data input and data output lines does it have?

   (b) How many address lines does it have?

   (c) What is the capacity in bytes?

   ....

   ....

6. Describe the internal architecture of a DRAM that stores 4K bytes chip size and uses a square register array. How many address lines will be needed? Suppose the same configuration exists for an old RAM, then how many address lines will be needed?

   ....

   ....

7. How many RAM chips of size 256K × 1 bit are required to build 1M Byte memory?

   ....

   ....

1.5 RAID AND ITS LEVELS

Researchers are constantly trying to improve the secondary storage media by increasing their capacity, performance, and reliability. However, any physical media has a limit to its capacity and performance. When it gets harder to make further improvements in the physical storage media and its drive, researchers normally look for other methods for improvement. Mass storage devices are a result of such improvements, which researchers have resorted to for larger capacity secondary devices. The idea is to use multiple units of the storage media being used as a single secondary storage device. One such attempt in the direction of improving disk performance is to have multiple components in parallel. Some basic questions for such systems are:

How are the disks organised?
May be as an array of disks.

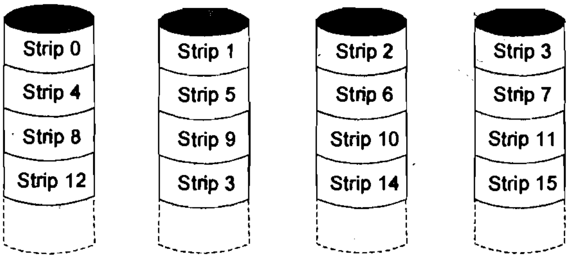
Can separate I/O requests be handled by such a system in parallel?
Yes, but only if the disk accesses are from separate disks.

Can a single I/O request be handled in parallel?
Yes, but the data block requested should be available on separate disks.

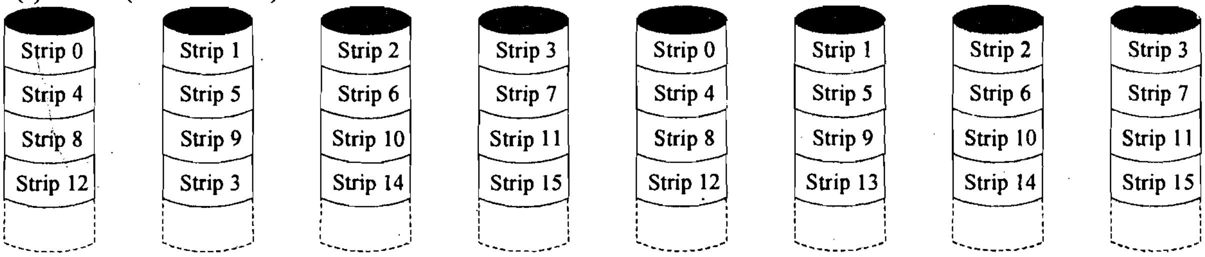
Can this array of disks be used for increasing reliability?
Yes, but for that redundancy of data is essential.

One such industrial standard, which exists for multiple-disk database schemes, is termed as RAID, i.e., Redundant Array of Independent Disks. The basic characteristics of RAID disks are:

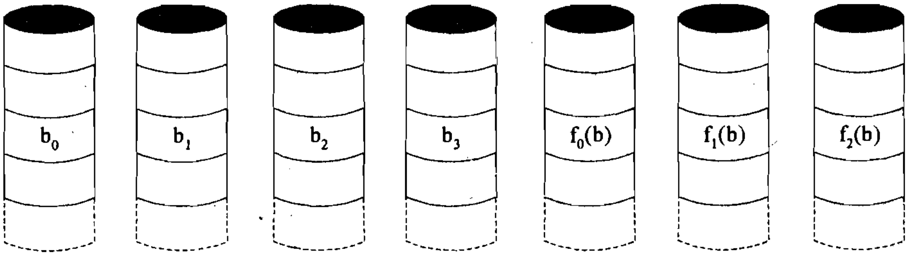
- Operating system considers the physical disks as a single logical drive.
- Data is distributed across the physical disks.
- In case of failure of a disk, the redundant information (for example, the parity bit) kept on redundant disks is used to recover the data.



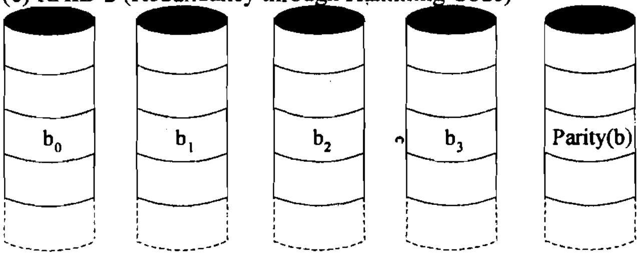
(a) RAID 0 (Non-redundant)



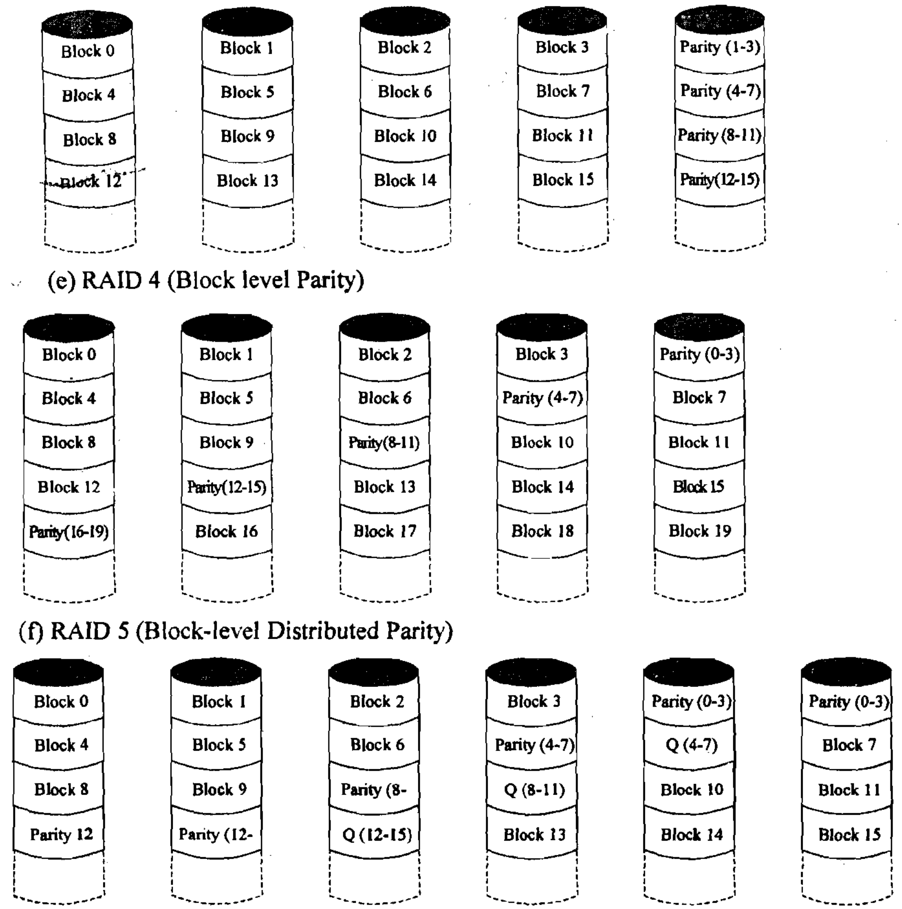
(b) RAID 1 (Mirrored)



(c) RAID 2 (Redundancy through Hamming Code)



(d) RAID 3 (Bit-interleaved parity)



(e) RAID 4 (Block level Parity)

(f) RAID 5 (Block-level Distributed Parity)

(g) RAID 6 (Dual Redundancy)

Figure 13: The RAID levels

The term RAID was coined by researchers at University of Berkley. In their paper the meaning of RAID was Redundant Array of Inexpensive Disks. However, later the term Independent was adopted instead of Inexpensive to signify performance and reliability gains.

RAID has been proposed at various levels, which are basically aimed to cater for the widening gap between the processor and on-line secondary storage technology.

The basic strategy used in RAID is to replace the large capacity disk drive with multiple smaller capacity disks. The data on these disks is distributed to allow simultaneous access, thus improving the overall input/output performance. It also allows an easy way of incrementing the capacity of the disk. Please note that one of the main features of the design is to compensate for the increase in probability of failure of multiple disks through the use of parity information. The seven levels of RAID are given in Figure 13 shown above. Please note that levels 2 and 4 are not commercially offered.

In RAID technologies have two important performance considerations:

- The Data Transfer Rate
- Input/Output Request rate

High data transfer rate is dependent on:

- Path between individual disks and Memory.
- Fulfilment of an I/O request by multiple disks of disk array. Thus, increasing the transfer rate.

Input/Output request rate is defined as the time taken to fulfil an I/O request. This is an important consideration while dealing with transaction processing systems like Railway reservation system. A high I/O request rate can be obtained through distributing data requests on multiple disks.

The features of these RAID levels are:

RAID Level	Category	Features	I/O Request Rate (Read /write)	Data Transfer Rate (Read /write)	Typical Application
0	Striping	a) The disk is divided into strips, maybe a block, a sector or other unit. b) Non-redundant.	Large strips: Excellent	Small strip: Excellent	Applications requiring high performance for non-critical data
1	Mirroring	a) Every disk in the array has a mirror disk that contains the same data. b) Recovery from a failure is simple. When a drive fails, the data may still be recovered from the second drive.	Good / fair	Fair /fair	System drives; critical files
2	Parallel Access	a) All member disks participate in the execution of every I/O request by synchronising the spindles of all the disks to the same position at a time. b) The strips are very small, often a single byte or word. c) Redundancy via hamming code which is able to correct single-bit errors and detect double-bit errors.	Poor	Excellent	Commercially not useful.
3	Parallel Access	a) Employs parallel access as that of level 2, with small data strips. b) A simple parity bit is computed for the set of data instead of an error-correcting code in case a disk fails.	Poor	Excellent	Large I/O request size application, such as imaging CAD
4	Independent access	a) Each member disk operates independently, thus enabling fulfilment of separate input/output requests in parallel. b) Data strip is large and bit by bit parity strip is created for bits of strips of each disk. c) Parity strip is stored on a separate disk.	Excellent/ fair	Fair / poor	Commercially not useful.
5	Independent access	a) Employs independent access as that of level 4 and distributes the parity strips across all disks. b) The distribution of parity strips across all drives avoids the potential input/output bottleneck found in level 4.	Excellent / fair	Fair / poor	High request rate read intensive, data lookup
6	Independent access	a) Also called the P+Q redundancy scheme, is much like level 5, but stores extra redundant information to guard against multiple disk failures. b) P and Q are two different data check algorithms. One of the two is the exclusive-or calculation used in level 4 and 5. The other one is an independent data check algorithm.	Excellent/ poor	Fair / poor	Application requiring extremely high availability

1.6 THE CONCEPTS OF HIGH SPEED MEMORIES

Why are high-speed memories needed? Is the main memory not a high-speed memory? The answer to the second question is definitely "No", but why so? For this, we have to go to the fundamentals of semiconductor technology, which is beyond the scope of the Unit. Then if the memories are slower, then how slow are they? On an average it has been found that the operating speed of main memories lack by a factor of 5 to 10 than that of the speed of processors (such as CPU or Input / Output Processors).

In addition, each instruction requires several memory accesses (it may range from 2 to 7 or even more sometimes). If an instruction requires even 2 memory accesses, even then almost 80% of the time of executing an expression, processors waits for memory access.

The question is what can be done to increase this processor-memory interface bandwidth? There are four possible answers to the question. These are:

a) Decrease the memory access time; use a faster but expensive technology for main memory.

b) Access more words in a single memory access cycle. That is, instead of accessing one word from the memory in a memory access cycle, access more words.

c) Insert a high-speed memory termed as Cache between the main memory and processor.

d) Use associative addressing in place of random access.

Hardware researchers are taking care of the first point. Let us discuss some high speed memories that are in existence at present.

1.6.1 Cache Memory

Cache memory is an extremely fast, small memory between CPU and main memory whose access time is closer to the processing speed of the CPU. It acts as a high-speed buffer between CPU and main memory and is used to temporarily store currently active data and instructions during processing. Since the cache memory is faster than main memory, the processing speed is increased by making data and instructions needed in present processing available in the cache

The obvious question that arises is how the system can know in advance which data and instruction are needed in present processing so as to make it available beforehand in the cache. The answer to this question comes from a principle known as ***locality of reference***. According to this principle, during the course of execution of most programs, memory references by the processor, for both instructions and data, tend to cluster. That is, if an instruction is executed, there is a likelihood of the nearby instruction being executed soon. Locality of reference is true not only for reference to program instruction but also for references to data. As shown in Figure 14, the cache memory acts as a small, fast-speed buffer between the processor and main memory.

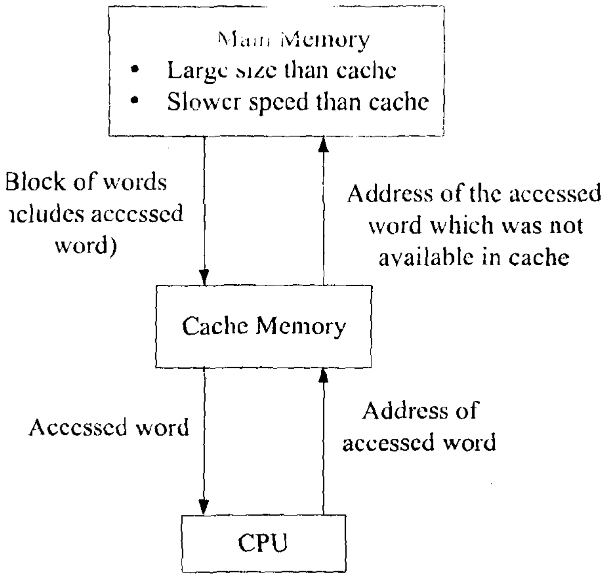


Figure 14: Cache Memory Operation

It contains a copy of a portion of main memory contents. When a program is running and the CPU attempts to read a word of memory (instruction or data), a check is made to determine if the word is in the cache. If so, the word is delivered to the CPU from the cache. If not, a block of main memory, consisting of some fixed number of words including the requested word, is read into the cache and then the requested word is delivered to the CPU. Because of the feature of locality of reference, when a block of memory word is fetched into the cache to satisfy a single memory reference, it is likely that there will soon be references to other words in that block. That is, the next time the CPU attempts to read a word, it is very likely that it finds it in the cache and saves the time needed to read the word from main memory.

Many computer systems are designed to have two separate cache memories called ***instruction cache*** and ***data cache***. The instruction cache is used for storing program instruction and the data cache is used for storing data. This allows faster identification of availability of accessed word in the cache memory and helps in further improving the processor speed. Many computer systems are also designed to have multiple levels of caches (such as level one and level two caches, often referred to as **L1 and L2 caches**). L1 cache is smaller than L2 cache and is used to store more frequently accessed instruction/data as compared to those in the L2 cache.

The use of cache memory requires several design issues to be addressed. Some key design issues are briefly summarised below:

1. **Cache Size:** Cache memory is very expensive as compared to the main memory and hence its size is normally kept very small. It has been found through statistical studies that reasonably small caches can have a significant impact on processor performance. As a typical example of cache size, a system having 1 GB of main memory may have about 1 MB of cache memory. Many of today's personal computers have 64KB, 128KB, 256KB, 512KB, or 1 MB of cache memory.

2. **Block Size:** Block size refers to the unit of data (few memory words) exchanged between cache and main memory. As the block size increases from very small to larger size, the hit ratio (fraction of times that referenced instruction/data is

found in cache) will at first increase because of the principle of locality since more and more useful words are brought into the cache. However, the hit ratio will begin to decrease as the block size further increases because the probability of using the newly fetched words becomes less than the probability of reusing the words that must be moved out of the cache to make room for the new block. Based on this fact, the block size is suitably chosen to maximise the hit ratio.

3. **Replacement Policy:** When a new block is to be fetched into the cache, another may have to be replaced to make room for the new block. The replacement policy decides which block to replace in such a situation. Obviously, it will be best to replace a block that is least likely to be needed again in the near future.

4. **Write Policy:** If the contents of a blo :k in the cache are altered, then it is necessary to write it back to main meı nory before replacing it. The write policy decides when the altered words of a b'ock are written back to main memory. At one extreme, an updated word of a blo :k is written to the main memory as soon as such updates occur in the block. At the other extreme, all updated words of the block are written to the main memory only when the block is replaced from the cache. The latter policy minimises overheads of memory write operations but temporarily leaves main memory in an inconsistent (obsolete) state.

1.6.2 Cache Organisation

Cache memories are found in almost all latest computers. They are very useful for increasing the speed of access of information from memory. Let us look into their organisation in more detail in this section.

The fundamental idea of cache organisation is that by keeping the most frequently accessed instructions and data in the fast cache memory; hence the average memory access time will approach the access time of the cache.

The basic operation of the cache is as follows. When the CPU needs to access memory, the cache is examined. If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word. A block of words is then transferred from main memory to cache memory.

The performance of cache memory is frequently measured in terms of a quantity called ***hit ratio***. When the CPU refers to the main memory and finds the word in cache, it is said to produce a ***hit***. If the word is not found in cache, it is in the main memory and it counts as a ***miss***. The ratio of the number of hits divided by the total CPU references to memory is the hit ratio.

The average memory access time of a computer system can be improved considerably by use of a cache. For example, if memory read cycle takes 100 ns and a cache read cycle takes 20 ns, then for four continuous references, the first one brings the main memory contents to cache and the next three from cache.

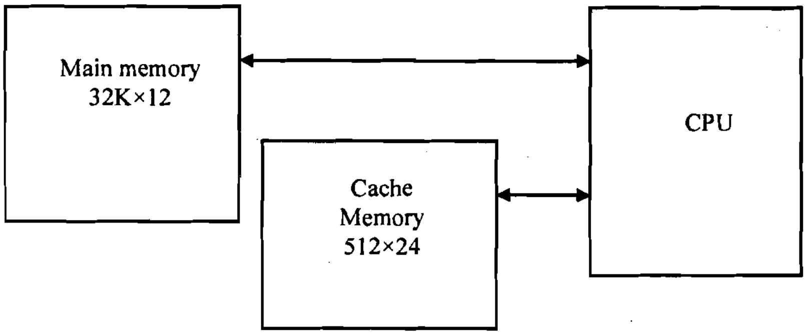
The time taken with cache = (100+20) + (20×3)
(For the first read operation) (For the last three read operations)
= 120 + 60
= 180 ns
Time taken without cache = 100×4 = 400 ns

Thus, the closer are the reference, the better is the performance of cache.

The basic characteristic of cache memory is its fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. The transformation of data from main memory to cache memory is referred to as a mapping process. The mapping procedure for the cache organization is of three types:

1. Associative mapping
2. Direct mapping
3. Set-associative mapping

Let us consider an example of a memory organization as shown in Figure 15 in which the main memory can store 32K words of 12 bits each and the cache is capable of storing 512 blocks (each block in the present example is equal to 24 bits, which is equal to two main memory words) at any time.



Size of main memory address (Given word size of 12 bits) = 32 K words = 2^{15} words

$\Rightarrow$ 15 bits are needed for address

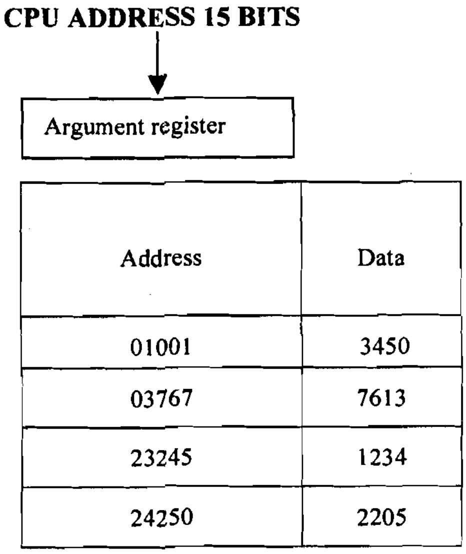
Block Size of Cache = 2 Main Memory Words

Figure 15: Cache Memory

For every word stored in cache, there is a duplicate copy in the main memory. The CPU communicates with both memories. It first sends a 15 bits ($32K = 2^5 \times 2^{10} = 2^{15}$) address to cache. If there is a hit, the CPU uses the relevant 12 bits data from 24 bit cache data. If there is a miss, the CPU reads the block containing the relevant word from the main memory. So the key here is that a cache must store the address and data portions of the main memory to ascertain whether the given information is available in the cache or not. However, let us assume the block size as 1 memory word for the following discussions.

Associative Mapping

The most flexible and fastest cache organization uses an associative memory which is shown in Figure 16. The associative memory stores both the address and data of the memory word. This permits any location in cache to store any word from the main memory. The address value of 15 bits is shown as a five-digit octal number and its corresponding 12 bits word is shown as a five digit octal number. A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address. If the address is found, the corresponding 12 bits data is read and sent to the CPU. If no matches are found, the main memory is accessed for the word. The address-data pair is then transferred to the associative cache memory. This address checking is done simultaneously for the complete cache in an associative way.



Address	Data
01001	3450
03767	7613
23245	1234
24250	2205

(All numbers are in octal)

Figure 16: Associative Mapping Cache

Direct Mapping

In the general case, there are 2^k words in cache memory and 2^n words in the main memory. The n-bits memory address is divided into two fields: k bits for the index field and (n - k) bits for the tag field.

The direct mapping cache organization uses the n-bit address to access the main memory and k-bit index to access the cache. The internal organization of the words in the cache memory is as shown in Figure 17. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access the cache.

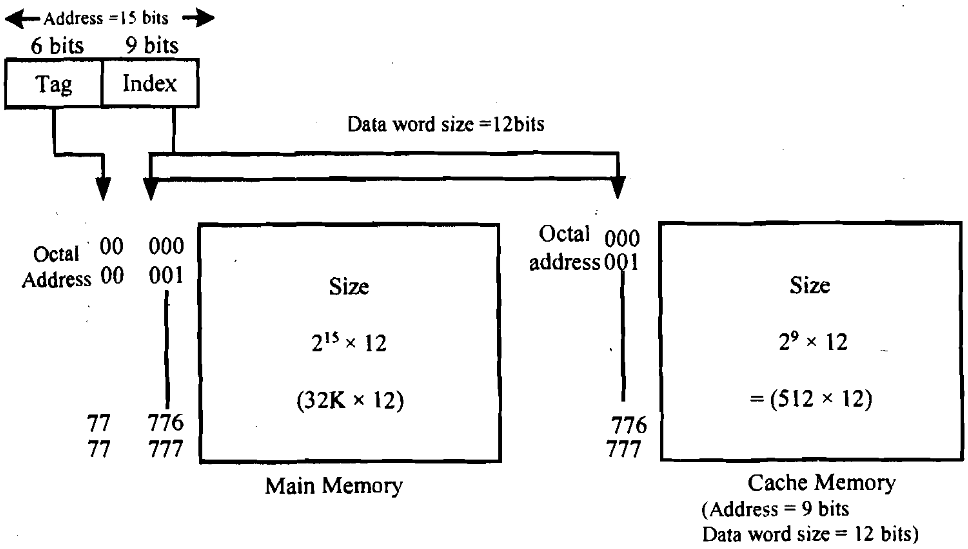


Figure 17: Addressing Relationship for Main Memory and Cache

The tag field of the CPU address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache. If there is no match, there is a miss and the required word is read from the main memory.

Let us consider a numerical example shown-in Figure 18. The word at address zero is at present stored in the cache (index = 000, tag = 00, data = 1456). Suppose that the CPU wants to access the word at address 02000. The index address is 000, so it is used to access the cache. The two tags are then compared. The cache tag is 00 but the address tag is 02, which does not produce a match. Therefore, the main memory is accessed and the data word 4254 is transferred to the CPU. The cache word at index address 000 is then replaced with a tag of 02 and data of 4254.

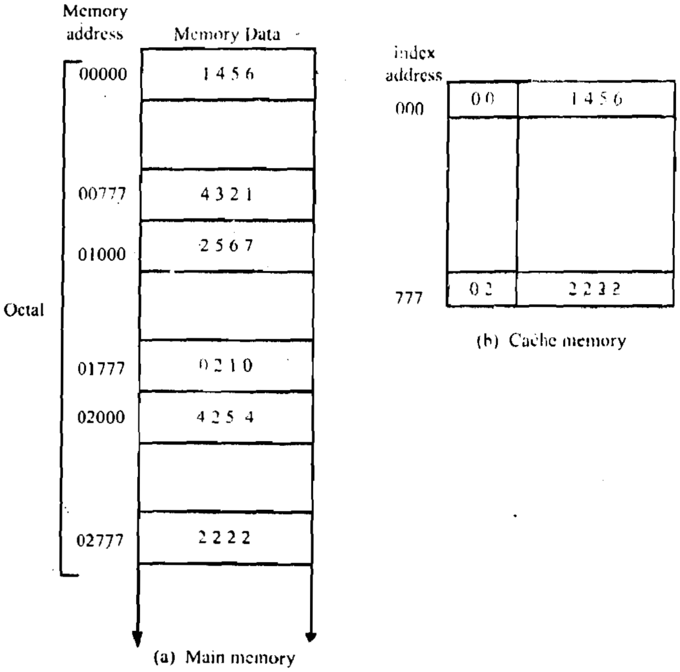


Figure 18: Direct Mapping Cache Organisation

Set-Associative Mapping

A third type of cache organization called set-associative mapping is an improvement on the direct mapping organization in that each word of cache can store two or more words of memory under the same index address. Each data word is stored together with its tag and the number of tag data items in one word of cache is said to form a set.

Let us consider an example of a set-associative cache organization for a set size of two as shown in the Figure 19. Each index address refers to two data words and their associated tags. Each tag requires six bits and each data word has 12 bits, so the word length of cache is 2(6+12) = 36 bits. An index address of nine bits can accommodate 512 words. Thus, the size of cache memory is 512 × 36. In general, a Set-Associative cache of set size K will accommodate K-words of main memory in each word of cache.

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Figure 19: Two-Way Set-Associative Mapping Cache

Write Policy: The data in cache and main memory can be written by processors or input/output devices. The main problems associated in writing with cache memories are:

a) The contents of cache and main memory can be altered by more than one device. For example, CPU can write to caches and input/output module can directly write to the main memory. This can result in inconsistencies in the values of the cache and main memory.

b) In the case of multiple CPUs with different cache, a word altered in one cache automatically invalidate the word in the other cache.

The suggested techniques for writing in system with caches are:

(a) **Write through:** Write the data in cache as well as main memory. The other CPUs - Cache combination has to watch with traffic to the main memory and make suitable amendment in the contents of cache. The disadvantage of this technique is that a bottleneck is created due to large number of accesses to the main memory by various CPUs.

(b) **Write block:** In this method updates are made only in the cache, setting a bit called Update bit. Only those blocks whose update bit is set is replaced in the main memory. But here all the accesses to the main memory, whether from other CPUs or input/output modules, need to be from the cache resulting in complex circuitry.

(c) **Instruction Cache:** An instruction cache is one which is employed for accessing only the instructions and nothing else. The advantage of such a cache is that as the instructions do not change we need not write the instruction cache back to memory, unlike data storage cache.

1.6.3 Memory Interleaving

In this method, the main memory is divided into 'n' equal-size modules and the CPU has separate Memory Address Register and Memory Base register for each memory module. In addition, the CPU has 'n' instruction register and a memory access system. When a program is loaded into the main memory, its successive instructions are stored in successive memory modules. For example if n=4 and the four memory modules are M_1, M_2, M_3, and M_4 then 1st instruction will be stored in M_1, 2nd in M_2, 3rd in M_3, 4th in M_4, 5th in M1, 6th in M_2 and so on. Now during the execution of the program, when the processor issues a memory fetch command, the memory access system creates n consecutive memory addresses and places them in the Memory Address Register in the right order. A memory read command reads all the 'n' memory modules simultaneously, retrieves the 'n' consecutive instructions, and loads them into the 'n' instruction registers. Thus each fetch for a new instruction results in the loading of 'n' consecutive instructions in the 'n' instruction registers of the CPU. Since the instructions are normally executed in the sequence in which they are written, the availability of N successive instructions in the CPU avoids memory access after each instruction execution, and the total execution time speeds up. Obviously, the fetch successive instructions are not useful when a branch instruction is encountered during the course of execution. This is because they require the new set of 'n' successive instructions, overwriting the previously stored instructions, which were loaded, but some of which were not executed. The method is quite effective in minimising the memory-processor speed mismatch because branch instructions do not occur frequently in a program.

Figure 20 illustrates the memory interleaving architecture. The Figure shows a 4- way (n=4) interleaved memory system.

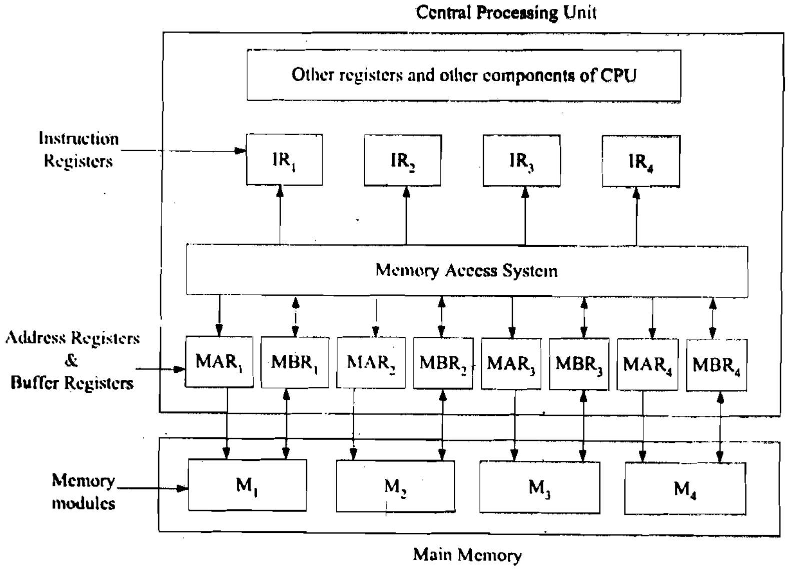


Figure 20: A 4-way Interleaved Memory

1.6.4 Associative Memory

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the contents of the data itself rather than by an address. A memory unit accessed by content of the data is called an **associative memory** or **content addressable memory (CAM)**. This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location. When a word is written in an associative memory, no address is given. The memory is capable of finding an empty unused location to store the word. When a word is to be read from an associative memory, the content of the word, or part of the word, is specified. The memory locates all words, which match the specified content, and marks them for reading.

Because of its organization, the associative memory is uniquely suited to do parallel searches by data association. Moreover, searches can be done on an entire word or on a specific field within a word. An associative memory is more expensive than a random access memory because each cell must have storage capability as well as logic circuits for matching its content with an external argument. For this reason associative memories are used in applications where the search time is very critical and must be very short.

Hardware Organization

The block diagram of an associative memory is shown in Figure 21. It consists of a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word. The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register; the words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched. Reading is accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

The key register provides a mask for choosing a particular field or key in the argument word. The entire argument is compared with each memory word if the key register contains all 1s. Otherwise, only those bits in the argument that have 1s in their corresponding positions of the key register are compared. Thus the key provides a mask or identifying information, which specifies how reference to memory is made.

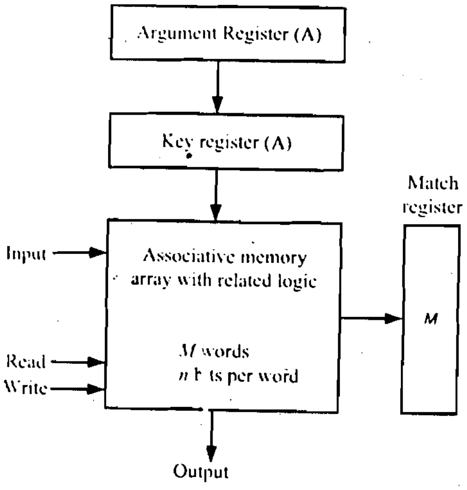


Figure 21: Associative Memory – Block Diagram

To illustrate with a numerical example, suppose that the argument register A and the key register K have the bit configuration shown below. Only the three leftmost bits of a compared with memory words because K has 1's on these positions

A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

Check Your Progress 2

1. What is a RAID? What are the techniques used by RAID for enhancing reliability?

   ....

   ....

   ....

2. **State True or False:** T/F

   a) Interleaved memories are best suited for small loops and large sequential code. □

   b) The principle of locality of reference justifies the use of cache memory. □

   c) High-speed memories are needed to bridge the gap of speed between I/O device and memory. □

   d) Write policy is not needed for instruction cache. □

   e) A replacement algorithm is needed only for associative and set associative mapping of cache. □

3. How can the Cache memory and interleaved memory mechanisms be used to improve the overall processing speed of a Computer system?

....

....

....

4. Assume a Computer having 64 word RAM (assume 1 word = 16 bits) and cache memory of 8 blocks (block size = 32 bits). Where can we find Main Memory Location 25 in cache if (a) Associative Mapping (b) Direct mapping and (c) 2 way set associative (2 blocks per set) mapping is used.

....

....

....

5. How is a given memory word address (memory location 25 as above) located to Cache for the example above for (a) Associative (b) Direct and (c) 2 way set associative mapping.

....

....

....

6. A computer system has a 4K-word cache organised in block set associative manner with 4 blocks per set, 64 words per block. What is the number of bits in the Index and Block Offset fields of the main memory address formula?

....

....

....

1.7 VIRTUAL MEMORY

In a memory hierarchy system, programs and data are first stored in auxiliary or secondary memory. The program and its related data are brought into the main memory for execution. What if the size of Memory required for the Program is more than the size of memory? Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of secondary memory. Each address generated by the CPU goes through an address mapping from the so-called virtual address to a physical address in the main memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory. A Virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations. This is done dynamically, while programs are being executed in the CPU. The translation or mapping is handled automatically by the hardware by means of a mapping table.

Address Space and Memory Space

An address used by a programmer will be called a virtual address, and the set of such addresses the address space. An address in the main memory is called a physical address. The set of such locations is called the memory space. Thus, the address space is the set of addresses generated by programs as they reference instructions and data;

the memory space consists of the actual main memory locations directly addressable for processing.

Consider a computer with a main-memory capacity of 64K words (K=1024). 16-bits are needed to specify a physical address in memory since 64K = 2^{16}. Suppose that the computer has auxiliary memory for storing information equivalent to the capacity of 16 main memories. Let us denote the address space by N and the memory space by M, we then have for this example N = 16 × 64 K = 1024K and M = 64K.

In a multiprogramming computer system, programs and data are transferred to and from auxiliary memory and main memory based on demands imposed by the CPU. Suppose that program 1 is currently being executed in the CPU. Program 1 and a portion of its associated data are moved from secondary memory into the main memory as shown in Figure 22. Portions of programs and data need not be in contiguous locations in memory since information is being moved in and out, and empty spaces may be available in scattered locations in memory.

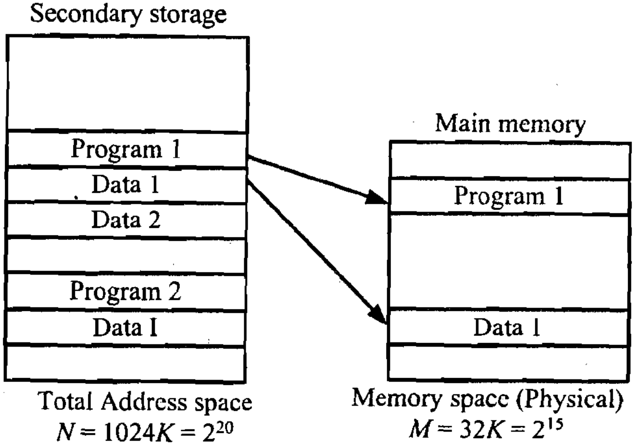


Figure 22: Address and Memory Space in Virtual Memory

In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 16-bits. Thus CPU will reference instructions and data with a 20 bits address, but the information at this address must be taken from physical memory because access to auxiliary storage for individual words will be prohibitively long. A mapping table is then needed, as shown in Figure 23, to map a virtual address of 20 bits to a physical address of 16 bits. The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.

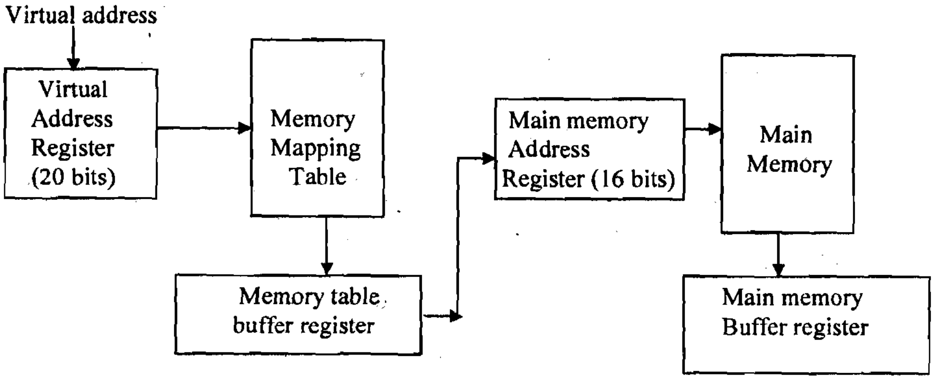

Virtual address
Virtual Address Register (20 bits)
Memory Mapping Table
Memory table buffer register
Main memory Address Register (16 bits)
Main Memory
Main memory Buffer register

Figure 23: Memory table for mapping a virtual table

1.8 THE MEMORY SYSTEM OF MICROCOMPUTER

Till now we have discussed various memory components. But, how is the memory organised in the physical computer? Let us discuss various kinds of memory technologies used in personal computer.

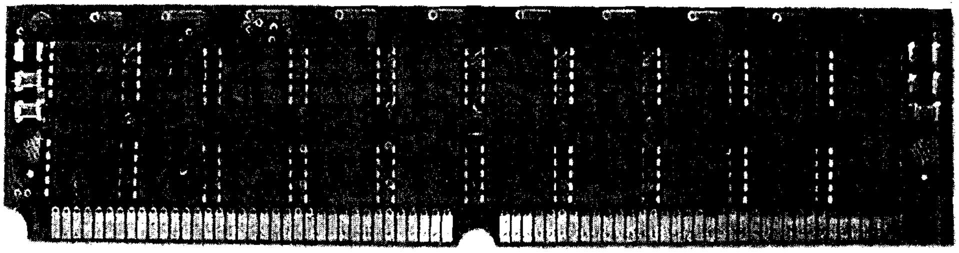
1.8.1 SIMM (Single In-line Memory Modules), DIMM (Dual In line Memory Modules), etc., Memory Chips

From the early days of semiconductor memory until the early 1990s, memory was manufactured, brought and installed as a single chip. Chip density went from 1K bits to 1M bits and beyond, but each chip was a separate unit. Early PCs often had empty sockets into which additional memory chips could be plugged, if and when the purchaser needed them. At present, a different arrangement is often used called SIMM or DIMM.

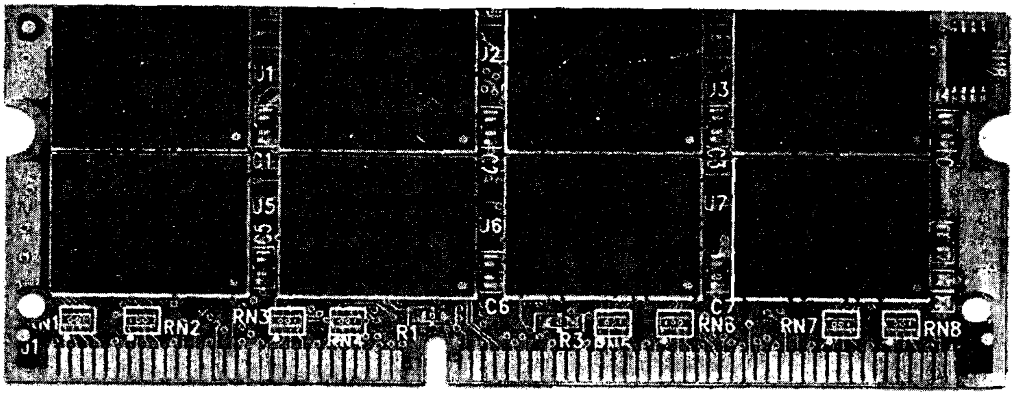
A group of chips, typically 8 to 16, is mounted on a tiny printed circuit board and sold as a unit. This unit is called a SIMM or DIMM depending on whether it has a row of connectors on one side or both sides of the board.

A typical SIMM configuration might have 8 chips with 32 megabits (4MB) each on the SIMM. The entire module then holds 32MB. Many computers have room for four modules, giving a total capacity of 128MB when using 32MB SIMMs. The first SIMMs had 30 connectors and delivered 8 bits at a time. The other connectors were addressing and control. A later SIMM had 72 connectors and delivered 32 bits at a time. For a machine like Pentium, which expected 64-bits at once, 72-connectors SIMMs were paired, each one delivering half the bits needed.

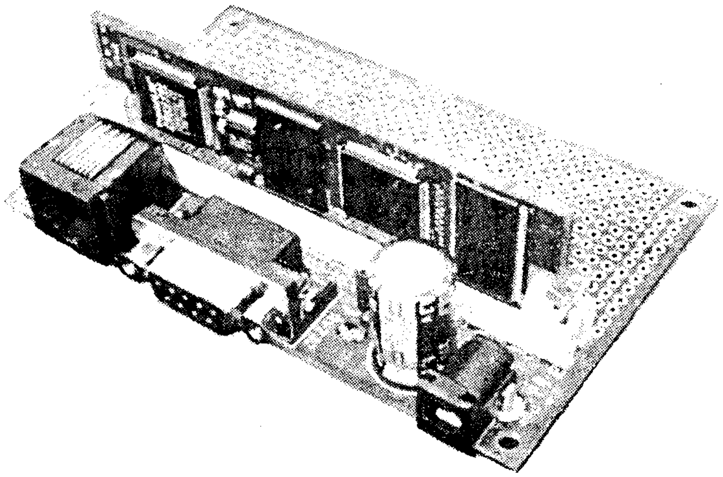
A DIMM is capable of delivering 64 data bits at once. Typical DIMM capacities are 64MB and up. Each DIMM has 84 gold patted connectors on each side for a total of 168 connectors. SIMM and DIMM are shown in Figure 24 (a) and (b) respectively. How they are put on a motherboard is shown in Figure 24 (c).



(a) SIMM



(b) DIMM



(c) DIMM on Motherboard

Figure 24: SIMM & DIMM

1.8.2 SDRAM, RDRAM, Cache RAM Types of Memory

The basic building block of the main memory remains the DRAM chip, as it has for decades. Until recently, there had been no significant changes in DRAM architecture since the early 1970s. The traditional DRAM chip is constrained both by its internal architecture and by its interface to the processor's memory bus. The two schemes that currently dominate the market are SDRAM and RDRAM. A third one, that is Cache RAM, is also very popular.

SDRAM (Synchronous DRAM)

One of the most widely used forms of DRAM is the synchronous DRAM (SDRAM). Unlike the traditional DRAM, which is asynchronous, the SDRAM exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor /memory bus without imposing wait states.

In a typical DRAM, the processor presents addresses and control levels to the memory, indicating that a set of data at a particular location in memory should be either read from or written into the DRAM. After a delay, the access time, the DRAM either writes or reads the data during the access-time delay. The DRAM performs various internal functions, such as activating the high capacitance of the row and column lines, sensing the data and routing the data out through the output buffers. The processor must simply wait through this delay, slowing system performance.

With synchronous access, the DRAM moves data in and out under control of the system clock. The processor or other master issues the instruction and address information, which is latched on to by the DRAM. The DRAM then responds after a set number of clock cycles. Meanwhile, the master can safely do other tasks while the SDRAM is processing the request.

The SDRAM employs a burst mode to eliminate the address setup time. In burst mode, a series of data bits can be clocked out rapidly after the first bit has been accessed. The mode is useful when all the bits to be accessed are in sequence and in the same row of the array as the initial access. In addition, the SDRAM has a multiple-bank internal architecture that improves opportunities for on-chip parallelism.

The mode register and associated control logic is another key feature differentiating SDRAMs from conventional DRAMs. It provides a mechanism to customize the SDRAM to suit specific system needs. The mode register specifies the burst length, which is the number of separate units of data synchronously fed onto the bus. The register also allows the programmer to adjust the latency between receipt of a read request and the beginning of data transfer.

The SDRAM performs best when it is transferring large blocks of data serially, such as for applications like word processing, spreadsheets, and multimedia.

RDRAM (Rambus DRAM)

RDRAM, developed by Rambus, has been adopted by Intel for its Pentium and Itanium processors. It has become the main competitor to SDRAM. RDRAM chips are vertical packages, with all pins on one side. The chip exchanges data with the processor over 28 wires no more than 12 centimeters long. The bus address up to 320 RDRAM chips and is rated at 1.6 GBps.

The special RDRAM bus delivers address and control information using an asynchronous block-oriented protocol. After an initial 480 ns access time, this produces the 1.6 GBps data rate. The speed of RDRAM is due to its high speed Bus. Rather than being controlled by the explicit RAS CAS R/W, and CE signals used in conventional DRAMs an RDAR gets a memory request over the high-speed bus. This request contains the desired address, the type of operation and the number of bytes in the operation.

CDRAM (Cache DRAM)

Cache DRAM (CDRAM), developed by Mitsubishi, integrates a small SRAM cache (16Kb) onto a generic DRAM chip. The SRAM on the CDRAM can be used in two ways. First, it can be used as true cache consisting of a number of 64-bit line. The cache mode of the CDRAM is effective for ordinary random access to memory.

The SRAM on the CDRAM can also be used as a buffer to support the serial access of a block of data. For example, to refresh a bit-mapped screen, the CDRAM can prefetch the data from the DRAM into the SRAM buffer. Subsequent accesses to chip result in accesses solely to the SRAM.

Check Your Progress 3

1 Difference between

a) SDRAM and RDRAM b) SIMM and DIMM

....

....

....

....

2. A Computer supports a virtual memory address space of 1Giga Words, but a physical Memory size of 64 Mega Words. How many bits are needed to specify an instruction address for this machine?

....

....

....

....

1.9 SUMMARY

In this unit, we have discussed the details of the memory system of the computer. First we discussed the concept and the need of the memory hierarchy. Memory hierarchy is essential in computers as it provides an optimised low-cost memory system. The unit also covers details on the basic characteristics of RAMs and different kinds of ROMs. These details include the logic diagrams of RAMs and ROMs giving basic functioning through various control signals. We have also discussed the latest secondary storage technologies such as CD-ROM, DVD-ROM, CD-R, CD-RW etc. giving details about their data formats and access mechanisms.

The importance of high-speed memories such as cache memory, interleaved memory and associative memories are also described in detail. The high-speed memory, although small, provides a very good overall speed of the system due to locality of reference. There are several other concepts such as the memory system of the microcomputer which consists of different types of chips such as SIMM, DIMM and different types of memory such as SDRAM, RDRAM also defined in easy way. The unit also contains details on Virtual Memory. For more details on the memory system you can go through further units.

1.10 SOLUTIONS / ANSWERS

Check Your Progress 1

1. a) True
   b) True
   c) False
2. a) Ultraviolet light, electrically
   b) Dynamic
   c) Sequential
3.

	RAM	**ROM and all types of ROM's**
a)	Volatile Memory	Non – volatile
b)	Faster access time	Slower than RAM
c)	Higher cost per bit storage	Lower than RAM
d)	Random access	Sequential access
e)	Less storage capacity	Higher storage capacity

4. A type of EPROM called EEPROM is known as flash memory used in many I/O and storage devices. It is commonly used memory in embedded systems.

5. (a) Eight, since the word size is 8.

   (b) 4K = 4 × 1024 = 4096 words. Hence, there are 4096 memory addresses. Since 4096 = 2^{12} it requires 12 bits address code to specify one of 4096 addresses.

   (c) The memory has a capacity of 4096 bytes.

6. 4K bytes is actually 4 × 1024 = 4096 bytes and the DRAM holds 4096 eight bit words. Each word can be thought of as being stored in an 8 bit register and there are 4096 registers connected to a common data bus internal to the chip. Since 4096 = $(64)^2$, the registers are arranged in a 64 × 64 array, that is there are 64=2^6 rows and 64=2^6 columns. This requires a 6 × 64 decoder to decode six- address inputs for the row select and a second 6 × 64 decoder to decode six other address

inputs for the column select. Using the structure as shown in Figure 3 (b), it requires only 6 bit address input.

While in the case of an old RAM, the chip requires 12 address lines (Please refer to Figure 2(b)), since 4096 = 2^{12} and there are 4096 different addresses.

7. 1 M Bytes = 2^{20}. 2^3 bits = 2^{23}
   256K × 1 bit = 2^8. 2^{10} bits = 2^{18}

   Hence, total number of RAM chips of size (256K × 1) = $\frac{2^{23}}{2^{18}} = 2^5 = 32$

Check Your Progress 2

1) A disk array known as RAID systems is a mass storage device that uses a set of hard disks and hard disk drives with a controller mounted in a single box. All the disks of a disk array form a single large storage unit. The RAID systems were developed to provide large secondary storage capacity with enhanced performance and enhanced reliability. The performance is based upon the data transfer rate, which is very high rather than taking an individual disk. The reliability can be achieved by two techniques that is mirroring (the system makes exact copies of files on two hard disks) and stripping (a file is partitioned into smaller parts and different parts of the files are stored on different disks).

2. a) True
   b) True
   c) False
   d) True
   e) False

3. The Cache memory is a very fast, small memory placed between CPU and main memory whose access time is closer to the processing speed of the CPU. It acts as a high-speed buffer between CPU and main memory and is used to temporarily store data and instruction needed during current processing. In memory interleaving, the main memory is divided into n number of equal size modules. When a program is loaded in to the main memory, its successive instruction in also available for the CPU, thus, it avoids memory access after each instruction execution and the total time speeds up.

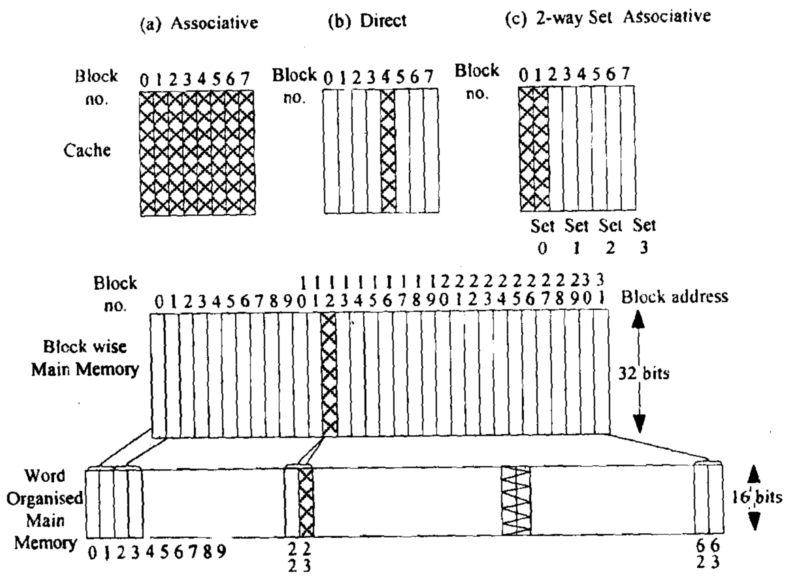
4. Main memory Size = 64 Words
   Main Memory word size = 16 bits
   Cache Memory Size = 8 Blocks
   Cache Memory Block size = 32 words
   ⇒1 Block of Cache = 2 Words of RAM
   ⇒ Memory location address 25 is equivalent to Block address 12.
   ⇒ Total number of possible Blocks in Main Memory = 64/2 = 32 blocks

(a) Associative Mapping: The block can be anywhere in the cache.

(b) Direct Mapping:
Size of Cache = 8 blocks
Location of Block 12 in Cache = 12 modulo 8 = 4

(c) 2 Way set associative mapping:
Number of blocks in a set = 2
Number of sets = Size of Cache in blocks / Number of blocks in a set
= 8 / 2 = 4
Block 12 will be located anywhere in (12 modulo 4) set, that is set 0.

Following figure gives the details of above schemes.



Cache mapping scheme

5. The maximum size of physical memory address in the present machine
= 6 bits (as Memory have 64 words, that is, 2^6)
The format of address mapping diagram for Direct and Set-Associative Mapping:

Physical Memory Word Address		
Block Address		Block Offset
Tag	Index	Block Offset

The address mapping in Direct Mapping:

Memory Address	0	1	1	0	0	1	⇒ Memory Address = 25
Block Address	0	1	1	0	0	1	⇒ Block Address = 12 and . Block offset = 1
Cache Address	0	1	1	0	0	1	⇒ Tag = 1; Index = 4 and . Block offset = 1

Please note that a main memory address 24 would have a block offset as 0.

Address Mapping for 2 way set associative mapping:

Memory Address	0	1	1	0	0	1	⇒ Memory Address = 25
Block Address	0	1	1	0	0	1	⇒ Block Address = 12 and . Block offset = 1
Cache Address	0	1	1	0	0	1	⇒ Tag = 3; Index (Set . Number) = 0 and . Block offset = 1

The Tag is used here to check whether a given address is in a specified set. This cache has 2 blocks per set, thus, the name two way set associative cache. The total number of sets here is 8 / 2 = 4.

For Associative mapping the Block address is checked directly in all location of cache memory.

6. There are 64 words in a block, therefore 4K cache has (4 × 1024)/ 64 = 64 blocks. Since 1 set has 4 blocks, there are 16 sets. 16 sets need 4 bit as $2^4 = 16$ representation. In a set there are 4 blocks. So, the block field needs 2 bits. Each block has 64 words. So the block offset field has 6 bits.
Index Filed is of 4 bits.
Block offset is of 6 bits.

Check Your Progress 3

1. a) SDRAM exchanges data with the processor synchronized to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states.

   A RDRAM module sends data to the controller synchronously to the clock to master, and the controller sends data to an RDRAM synchronously with the clock signal in the opposite direction.

   b) In SIMM, a group of chips, typically 8 to 16, is mounted on a tiny printed circuit board with one side only. While in DIMM, this can be mounted on both sides of the printed circuit board.

   The latest SIMMs and DIMMs are capable of delivering 64 data bits at once. Each DIMM has 84 gold patted connectors on each side for a total of 168 connectors while each SIMM 72 connectors.

2. The virtual address is 1 GB = 2^{30}, thus, 30 bit Virtual address, that will be translated to physical memory address of 26 bits (64 Mega words = 2^{26}).