6. Memory Organization

6.1. Computer system overview

The memory unit is an essential component in any digital computer since it is needed for storing programs and data. A very small computer with a limited application may be able to fulfill its intended task without the need of additional capacity, but the general purpose computer required more memory to store the programs and data. There is not much of memory to accommodate all programs used in a typical computer, also not all accumulated information is used by processor at the same time. Therefore, it is more economical to use low-cost storage devices to serve as a backup for storing the information that is not currently used by the CPU. The memory unit that communicated directly with the CPU is called the *main memory*. Devices that provide backup storage are called *auxiliary memory*. The most common auxiliary memory devices used in computer system are magnetic disk and tapes, used for storing system programs, large data files and other back up information. Only programs that are currently needed by the processor reside in main memory. All other information is stored in auxiliary memory and transferred to main memory when needed.

The memory system can be classified into different categories according to the characteristics as follows:

1 **Location:** Internal (e.g. processor registers, main memory, cache). External (e.g. optical disk, magnetic disk, tapes).

2 **Capacity:** Number of words, bytes, for internal memory this is typically expressed in terms of bytes, or word, common capacity lengths are 8, 16, and 32. External memory capacity is typically expressed in terms of bytes.

3 **Unit of transfer:** For internal memory, the unit of transfer is equal to the number of electrical lines into and out of the memory module. This may be equal to the word length, but is often larger, such as 64, 128, or 256 bits. To clarify this point, consider three related concepts for internal memory:

   i. **Word:** The "natural" unit of organization of memory. The size of the word is typically equal to the number of bits used to represent an integer and to the instruction length.

   ii. **Addressable units:** In some systems, the addressable unit is the word. However, many systems allow addressing at the byte level.

   iii. **Unit of transfer:** For main memory, this is the number of bits read out of or written into memory at a time. The unit of transfer need not equal a word or an addressable unit. For external memory, data are often transferred in much larger units than a word, and these are referred to as blocks.

4. **Access method:** it refers to the access method used for data and information. It includes
    i. **Sequential access:** Memory is organized into units of data, called records. Access must be made in a specific linear sequence. Thus, the time to access an arbitrary record is highly variable.
    ii. **Direct access:** individual blocks or records have a unique address based on physical location. Access is accomplished by direct access to reach general vicinity plus sequential searching, counting, or waiting to reach the final location. Again, access time is variable.
    iii. **Random access:** Each addressable location in memory has a unique, physically wired-in addressing mechanism. The time to access a given location is independent of the sequence of prior accesses and is constant. Thus, any location can be selected at random and directly addressed and accessed.
    iv. **Associative:** This is a random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously. Thus, a word is retrieved based on a portion of its contents rather than its address.
5. **Performance: three** performance parameters are
    1. **Access time (latency):** For random-access memory, this is the time it takes to perform a read or write operation, that is, the time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use. For non-random-access memory, access time is the time it takes to position the read–write mechanism at the desired location.
    2. **Memory cycle time:** This concept is primarily applied to random-access memory and consists of the access time plus any additional time required before a second access can commence. This additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively.
    3. **Transfer rate:** This is the rate at which data can be transferred into or out of a memory unit. For random-access memory, it is equal to 1/ (cycle time). For non-random-access memory, the following relationship holds:

$$T_N = T_A + \frac{n}{R}$$

Where, TN: Average time to read or write N bits

TA: Average access time

n: Number of bits

R: Transfer rate, in bits per second (bps)

6. **Physical type:** Semiconductor, magnetic, optical, magneto-optical.
7. **Physical characteristics:** Volatile/ non- volatile, Erasable/ non Erasable
8. **Organization:** By *organization* is meant the physical arrangement of bits to form words.

6.2. The memory hierarchy

The design constraints on a computer's memory can be summed up by three questions, how much? How fast and how expensive?

- Faster access: greater cost per bit.
- Greater capacity: smaller cost per bit.
- Greater capacity: slower access time.

The hierarchy shows the way how the memory comp

hierarchy the following occurs:

a) Decrease cost per bit.
b) Increasing capacity.
c) Increasing access time.
d) Decreasing frequency of access by processor.

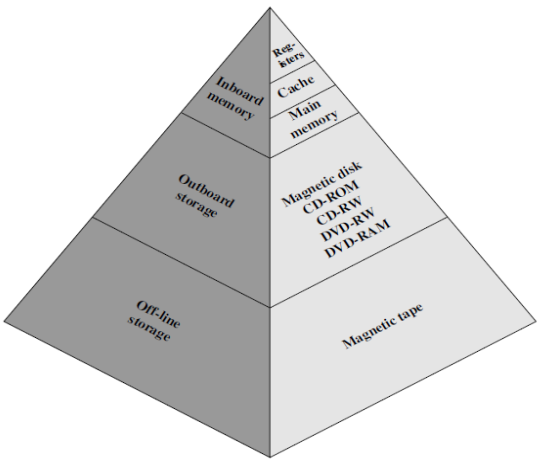


Figure 6.1: Memory hierarchy

6.3. Semiconductor main memory

In earlier computers, the most common form of random-access storage for computer main memory employed an array of doughnut-shaped ferromagnetic loops referred to as cores. It stored data by using induction (hysteresis loop) method. 1 core = 1-bit storage. Today, the use of semiconductor chips for main memory is almost universal. The characteristics of those include it store 1 bit or 1 or 0 and are capable of being read and written. The basic element of semiconductor memory is called cell. Figure below shows the basic cell of a semiconductor memory.

Figure below depicts the operation of a memory cell. Most commonly, the cell has three functional terminals capable of carrying an electrical signal. The select terminal, as the name suggests, selects a memory cell for a read or write operation. The control terminal indicates read or write. For writing, the other terminal provides an electrical signal that sets the state of the cell to 1 or 0. For reading, that terminal is used for output of the cell's state.

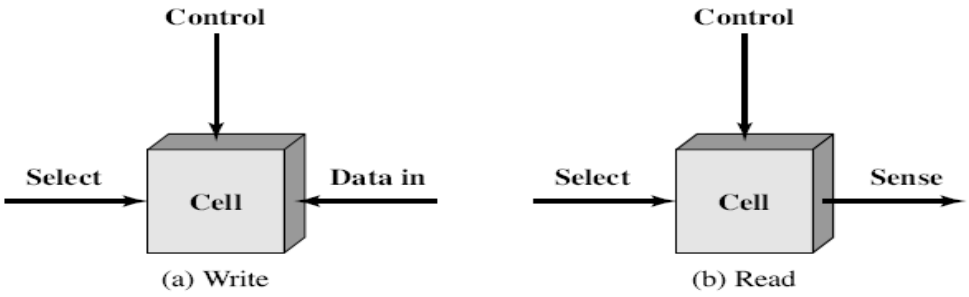


Figure 6.2: Memory cell operation

6.3.1. Random Access Memory (RAM)

The characteristics of RAM is to both read and write data from memory easily and rapidly, both reading and writing are accomplished through the use of electrical signals. RAM is that it is volatile. A RAM must be provided with a constant power supply. If the power is interrupted, then the data are lost.

6.3.1.1. Types of RAM

a) Dynamic Random Access Memory(DRAM):

A dynamic RAM (DRAM) is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0. Because capacitors have a natural tendency to discharge, dynamic RAM require 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 shows the typical DRAM structure for an individual cell that stores 1 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, allowing or stopping the current flow.

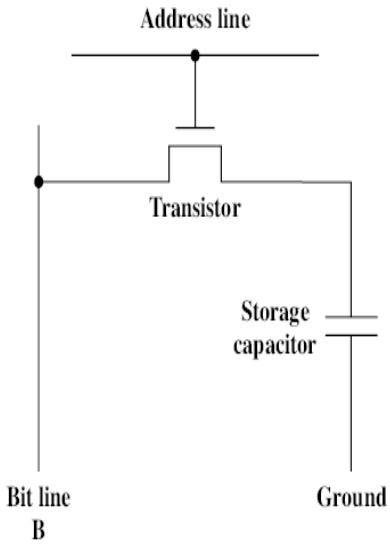


Figure 6.3: Single DRAM cell

For the write operation, 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 a 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 readout 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. Figure shows the circuit of one DRAM cell.

b) Static Random Access Memory (SRAM):

A static RAM (SRAM) is a digital device that uses the same logic elements used in the processor. In a SRAM, binary values are stored using traditional flip-flop logic-gate configurations (see Chapter 20 for a description of flip-flops). A static RAM will hold its data as long as power is supplied to it.

Figure shows a typical SRAM structure for an individual cell. Four transistors (T1, T2, T3, and T4) are cross connected in an arrangement that produces a stable logic state. In logic state 1, point C1 is high and point C2 is low; in this state, T1 and T4 are off and T2 and T3 are on. In logic state 0, point C1 is low and point C2 is high; in this state, T1 and T4 are on and T2 and T3 are off. Both states are stable as long as the direct current (dc) voltage is applied. Unlike the DRAM, no refresh is needed to retain data. As in the DRAM, the SRAM address line is used to open or close a switch. The address line controls two transistors (T5 and T6). When a signal is applied to this line, the two transistors are switched on, allowing a read or write operation. For a write operation, the desired bit value is applied to line B, while its complement is applied to line. This forces the four transistors (T1, T2, T3, and T4) into the proper state. For a read operation, the bit value is read from line B.

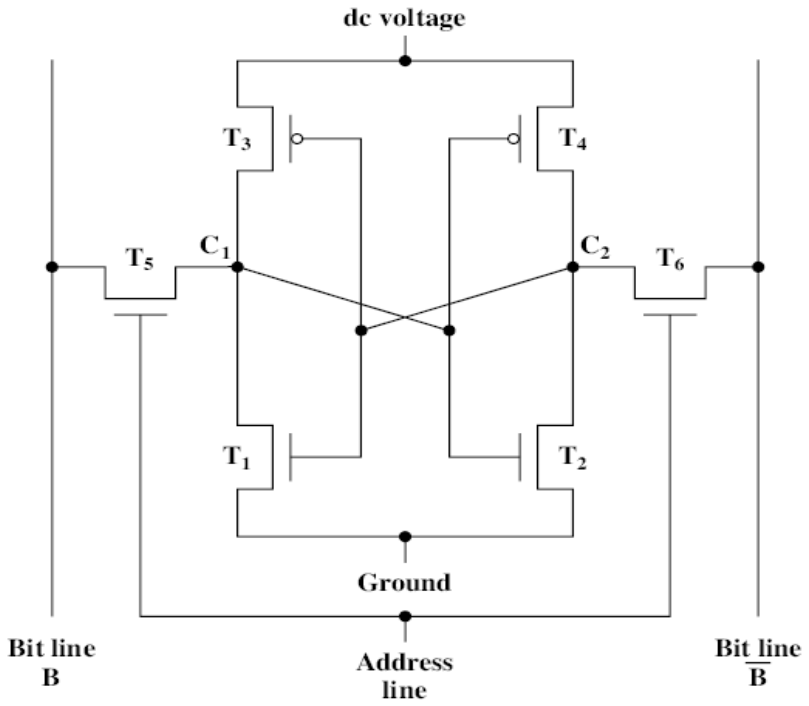


Figure 6.4: SRAM cell

** suppose logic 1 is stored then C1 is high and C2 is low. To read B & $\overline{B}$ are pre charged to Vdd before address line is allowed to be high. Now for logic high combination T2 & T3 will be on and T1 & T4 will be off. The value from line $\overline{B}$ will be drained and B will be retained to dc voltage so B=1 & $\overline{B}$ =0. For opposite case C1=0 & C2=1.

SRAM VERSUS DRAM: Both static and dynamic RAMs are volatile, i.e. power must be continuously supplied to the memory to preserve the bit values. A dynamic memory cell is simpler and smaller than a static memory cell. Thus, a DRAM is more dense (smaller cells more cells per unit area) and less expensive than a corresponding SRAM. On the other hand, a DRAM requires the supporting refresh circuitry.

6.3.1.2. Module organization

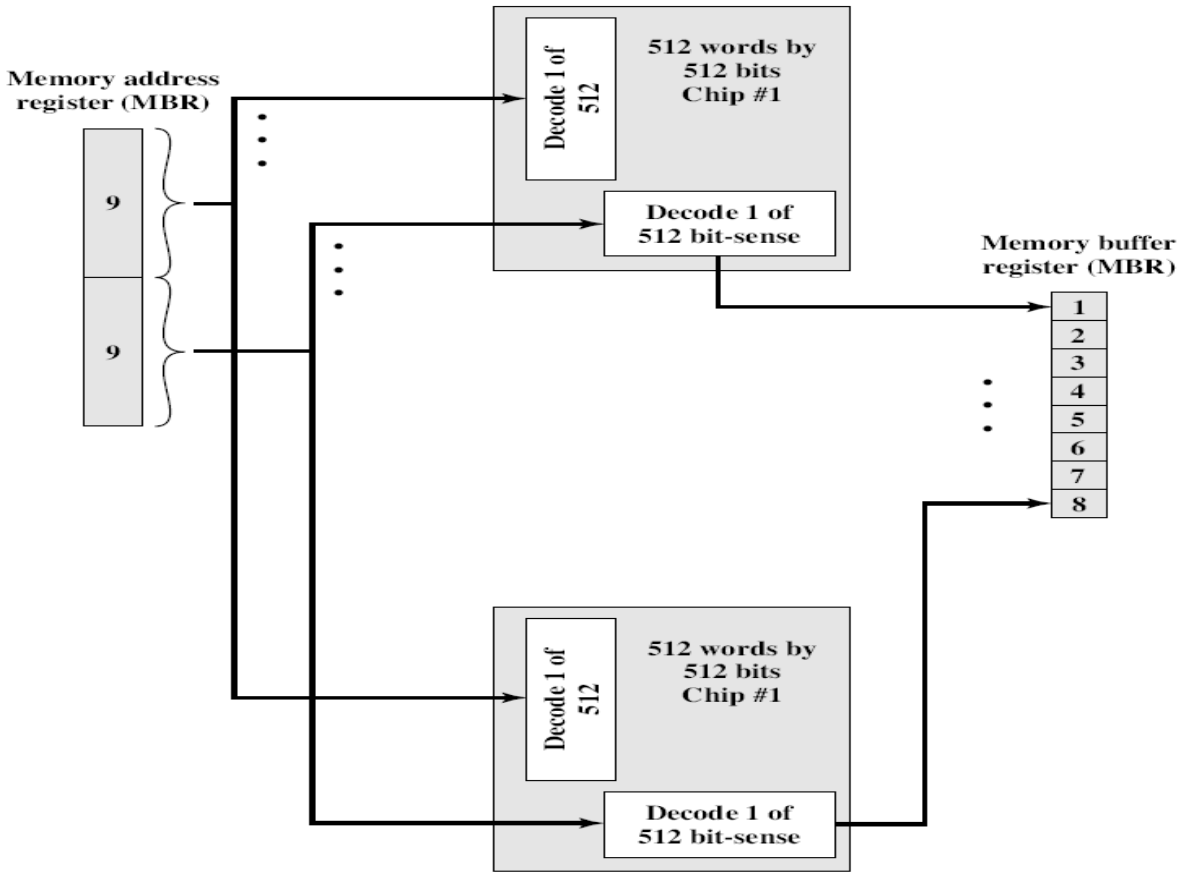


Figure 5.5: 256-KByte Memory Organization

If a RAM chip contains only 1 bit per word, then clearly we will need at least number of chips equal to the number of bits per word. As an example, Figure 5.5 shows how a memory module consisting of 256K 8-bit words could be organized. For 256K words, an 18-bit address is needed and is supplied to the module from some external source (e.g., the address lines of a bus to which the module is attached). The address is presented to 8 256K 1-bit chips, each of which provides the input/output of 1 bit. This organization works as long as the size of memory equals the number of bits per chip. In the case in which larger memory is required, an array of chips is needed.

6.3.1.3. Advanced DRAM organization

a. Synchronous DRAM (SDRAM)

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 of 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 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. Figure 5.6 shows the internal logic of IBM's 64-Mb SDRAM [IBM01], which is typical of SDRAM organization.

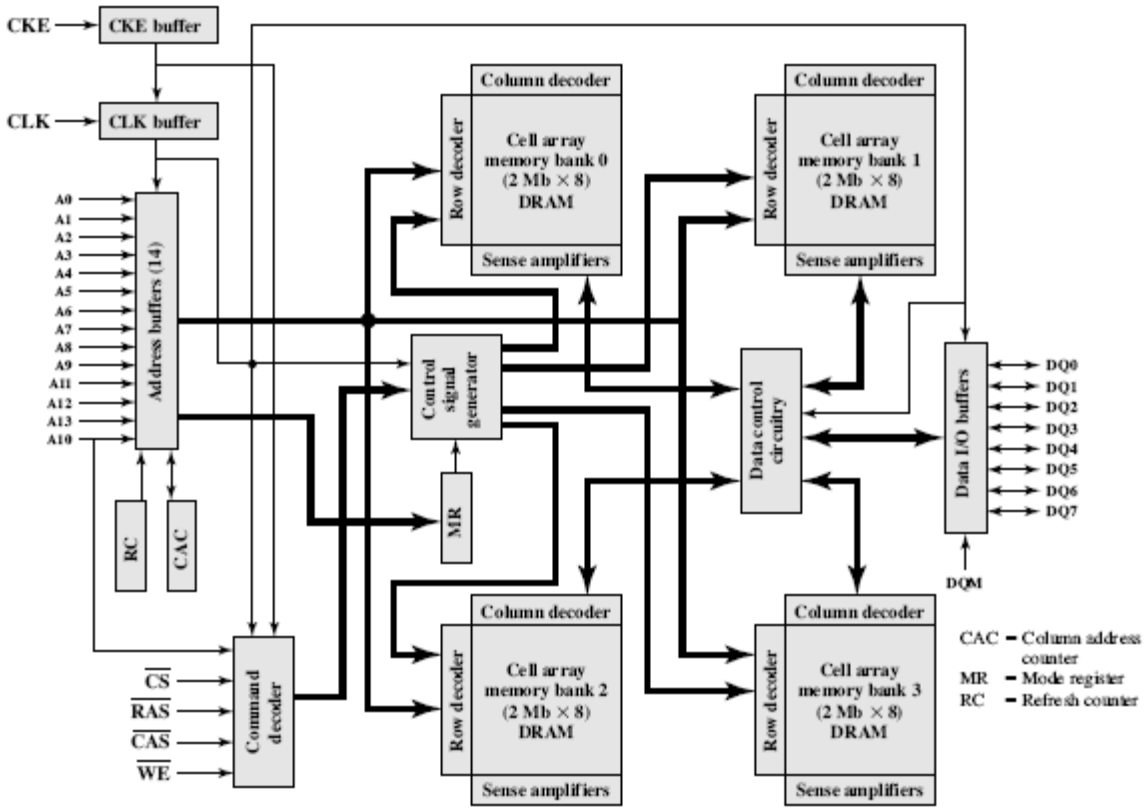


Figure 6.6: Synchronous DRAM

b. Rambus DRAM (RDRAM)

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 can 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. What makes this speed possible is the bus itself, which defines impedances, clocking, and signals very

precisely. RDRAM 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.

Figure illustrates the RDRAM layout. The configuration consists of a controller and a number of RDRAM modules connected via a common bus. The controller is at one end of the configuration, and the far end of the bus is a parallel termination of the bus lines. The bus includes 18 data lines (16 actual data, two parity) cycling at twice the clock rate; that is, 1 bit is sent at the leading and following edge of each clock signal. There is a separate set of 8 lines (RC) used for address and control signals. There is also a clock signal that starts at the far end from the controller propagates to the controller end and then loops back. 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. The remaining bus lines include a reference voltage, ground, and power source.

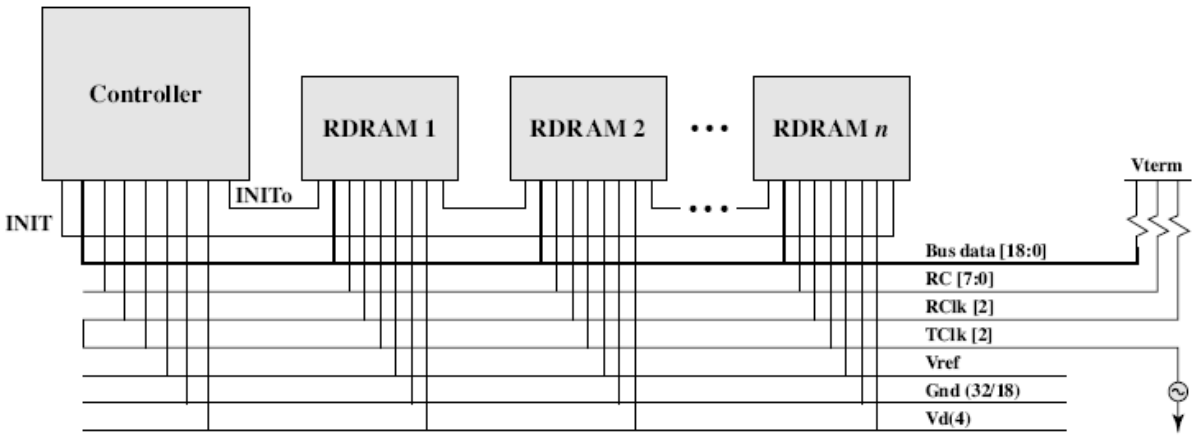


Figure 6.7: RDRAM Structure

c. DDRSRAM

SDRAM is limited by the fact that it can only send data to the processor once per bus clock cycle. Double-data-rate SDRAM can send data twice per clock cycle, once on the rising edge of the clock pulse and once on the falling edge of clock pulse. Theoretically, a DDR module can transfer data at a clock rate in the range of 200 to 600 MHz, a DDR2 module transfers at a clock rate of 400 to 1066 MHz and a DDR3 module transfers at a clock rate of 800 to 1600 MHz. In practice, somewhat smaller rates are achieved.

d. Cache DRAM

Cache DRAM integrates a small SRAM cache (16 Kb) on a generic DRAM chip. It can be sued in two ways:

a) as a true cache consisting a number of 64-bit line, more effective for ordinary random access to memory

b) SRAM on the CDRAM can also be used as a buffer to support the serial access to a block of data.

6.3.2. Read Only Memory (ROM)

Read Only Memory (ROM) is a nonvolatile, permanent data storage that cannot be changed. While it is possible to read a ROM many times, it is not possible to write new data into it. An important application of ROMs is microprogramming.

6.3.2.1. Types of ROM:

a. Read Only Memory(ROM):

A ROM is created like any other integrated circuit chip, with the data actually wired into the chip as part of the fabrication process. This presents two problems:

- The data insertion step includes a relatively large fixed cost, whether one or thousands of copies of a particular ROM are fabricated.
- There is no room for error. If one bit is wrong, the whole batch of ROMs must be thrown out.

b. Programmable ROM (PROM):

- May be written once.
- Writing process is performed electrically may be performed by supplier or customer.
- Special equipment required writing or programming process.
- Useful for small production runs.

c. Erasable PROM(EPROM):

- Optically erasable by exposing to UV light for 20 minutes.
- Read and write done electrically.
- All storage cell must be erased to write again.
- Can be erased, write, read multiple times and can hold data virtually indefinitely.

d. Electrically EPROM(EEPROM):

- Can be written many times while remaining in the system.
- Does not have to be erased first.

- Programming can be done at individual bytes level.
- Write requires longer time than read.
- Used in system for development, personalization and other task requiring unique information to be stored.

e. Flash memory:

- Intermediate between EPROM & EEPROM in both cost and functioning.
- Like EEPROM used electrical erasing technology.
- Fast erasing of entire blocks or just few blocks.
- Higher density than EEPROM because it uses only one transistor per bit.

**Read chip logic from computer Organization and architecture design for performance by William Stallings.

6.4. Auxiliary Memory

6.4.1. Magnetic disk

- The magnetic disk is a metal or plastic platter coated with magnetizable materials.
- Metal or plastic platter is called substrate, traditionally made from aluminum or aluminum alloy materials, recently glass substrate.

Magnetic Read and Write Mechanism

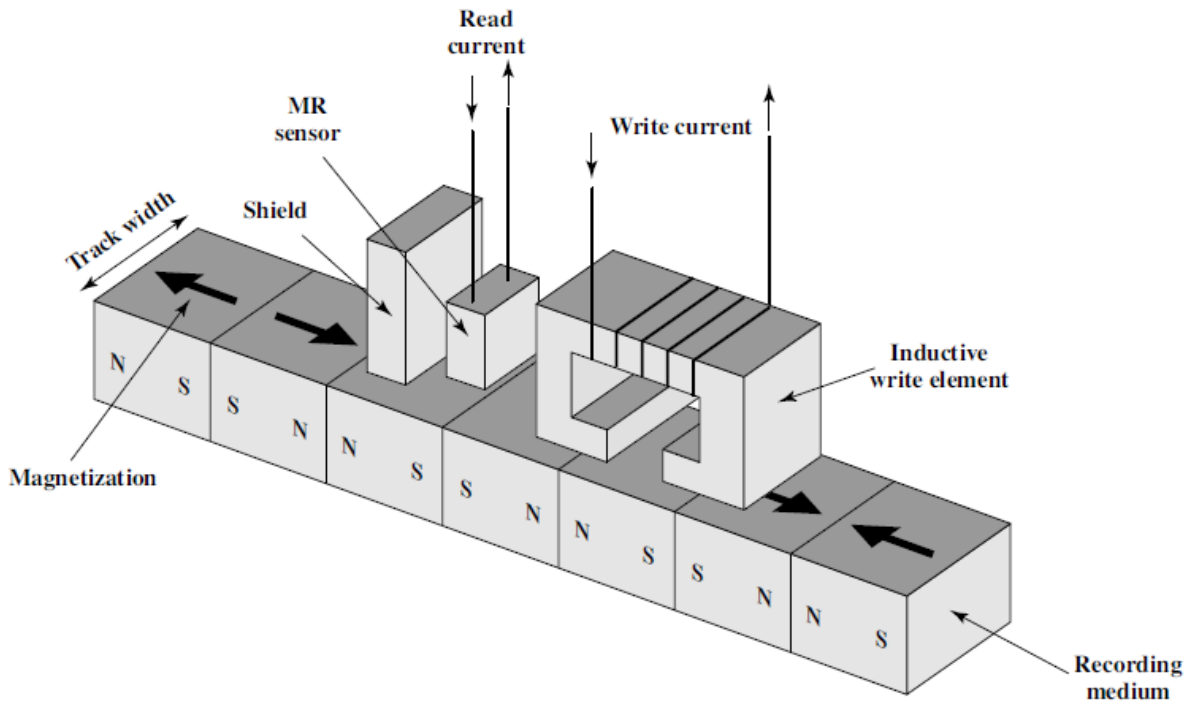


Figure 6.8: inductive wire/ Magnetoresistive Read Head

- Data are recorded and later retrieved from the disk via conducting coil called *head* during the read or write operation the head is stationary while platter rotates beneath it.
- The write mechanism exploits the fact that electricity flowing through a coil produce a magnetic field. For writing data, a pulse of electricity is sent to write head that results in magnetic patterns which are stored on the surface below.
- Write head itself is made from easily magnetizable materials shaped like a rectangular doughnut with a gap on one side and few turns of conducting wire on opposite side.
- An electric current on wire induces a magnetic field that magnetizes small areas of recording medium. Reversing the direction of the currents reverses the direction of magnetization.
- Traditional read mechanism exploits the fact that a magnetic field moving relative to coil produces an electric current in coil. Same way when surface of the disk passes under the head it generates a current of the same polarity as the one already recorded.
- For rigid disk system require a separate read head positioned for convenience close to write head. The read head consists of partially shielded magnetoresistive(MR) sensor. By passing a current through MR sensor, resistance changes are detected as voltage signal. The MR design allows higher frequency operation which equals to greater storage capacity.

Disk characteristics

- Single Vs multiple platters.
- Fixed Vs movable head.
- Removable Vs non removable platters.
- Data access time
  - Seek time: time to position the head over correct track.
  - Rotational latency: wait for the desired sector to come under head.
  - Access time: seek time plus rotational latency.
  - Block transfer time: time to read block of data.
- Disk capacity= M*T*P*S bytes

  S=bytes stored/sector

  P=sector per track

  T=track per surface

  M= no of surface

6.4.2. Magnetic tapes

- Magnetic tape system uses same technique for reading and recording as disk system.

- The medium is flexible polyester tape coated with magnetizable material.
- Tape is similar to home tape recorder system.

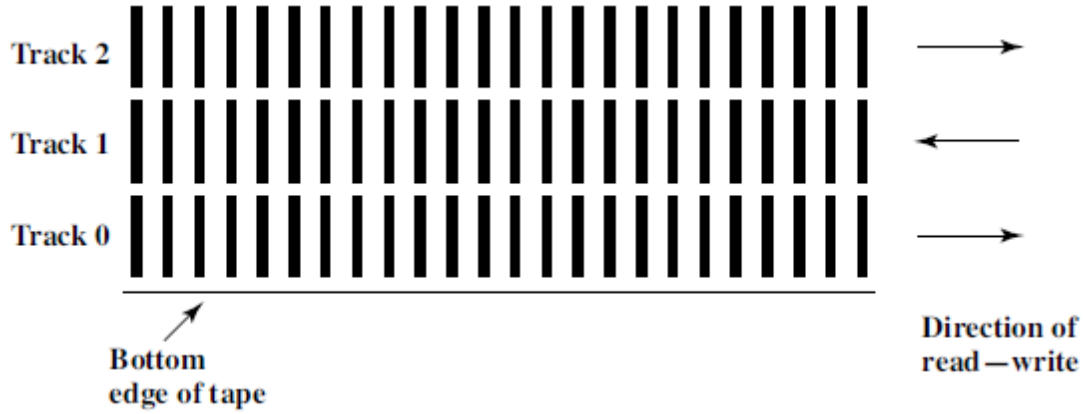


figure 6.9: Read write of tracks in magnetic tapes

- Data on tapes are structured as a number of parallel tracks running lengthwise.
- Earlier system used 9 tracks, later 18 or 36 tracks.
- Modern system use serial recording in which data are laid out as a sequence of bits along each track.
- The typical recording technique used in serial tapes is called serpentine recording.
- In this technique the first set of bits is recorded along the whole length of the tapes.
- When the end of the tape is reached the head is repositioned to record new track and again recorded on its whole length in opposite direction. This process continues back and forth for the whole length of the tapes.
- To increase the speed some read write head is capable of writing on a number of adjacent tracks simultaneously.

6.4.3. Optical memory

6.4.3.1. Compact Disk

The CD (Compact Disk) is a non-erasable disk. Different optical disk is CD-audio, CD-ROM, CD-R, DVD etc.

Compact disk

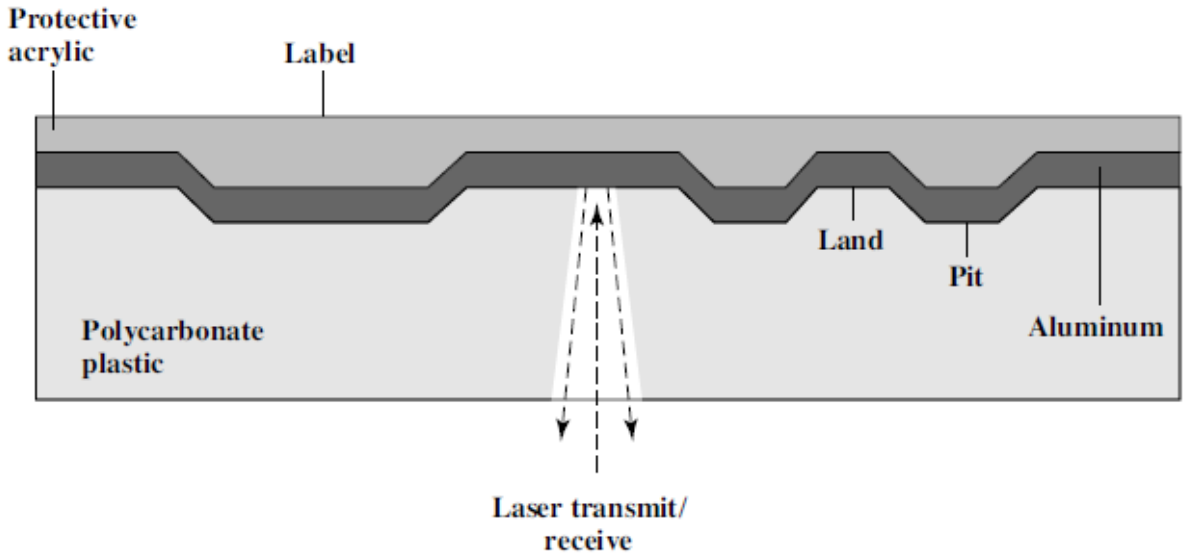


Figure 6.10: CD operation

- The disk is formed from resign, such as polycarbonate.
- Digitally recorded information is imprinted as a series of microscopic pits on the surface of the polycarbonate.
- This is done first of all with a finely focused high intensity laser to create a master disk. The master is used in turn to make a die to stamp out copies of only poly carbonate.
- The pitted surface is then coated with a highly reflective usually aluminum or gold. The shiny surface is protected against dust and scratches by a top coat of acrylic.
- Information is retrieved from CD or CD-ROM by a low powered laser housed in an optical disk player or drive unit.
- The intensity of the light reflected by the surface changes as it encounters pits or lands.
- If the laser beam falls on a pit which has a somewhat rough surface the light scatters and a low intensity is reflected back from surface.
- A land is smooth surface which reflects back at higher intensity.
- The change between pits and land is detected by a photo sensor and converted into a digital signal.
- The beginning or end of pit represents a 1, when no change in elevation occurs between intervals 0 is recorded.

6.4.3.2. CD-Rewritable

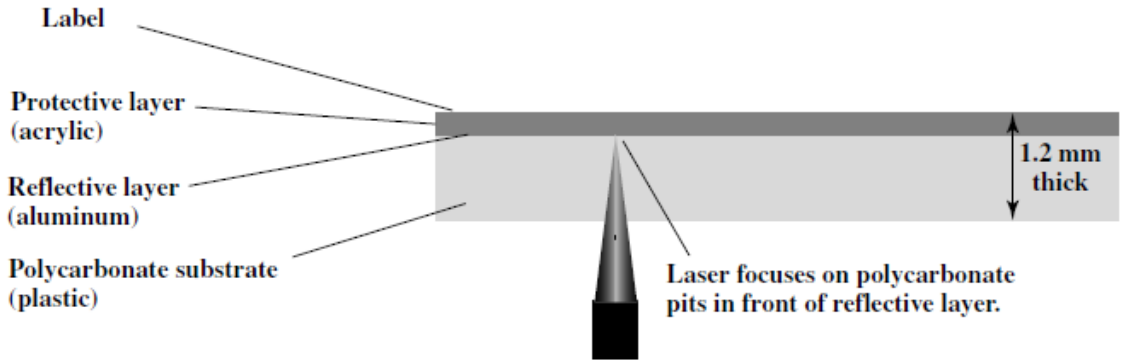
- CD-RW optical disk can be repeatedly written and overwritten. Although a number of approaches have been tried the only pure optical approach that has proved attractive is called phase change.

- The phase change disk uses two materials with different reflectivities.
- The amorphous state-in which molecules exhibit random orientation that reflects light poorly and crystalline state which has a smooth surface that reflects light well.
- A beam of laser can be used to change the material from one phase to other. i.e. erasing.
- The primary disadvantage of phase change optical disk is that the material eventually and permanently loses its desirable properties.

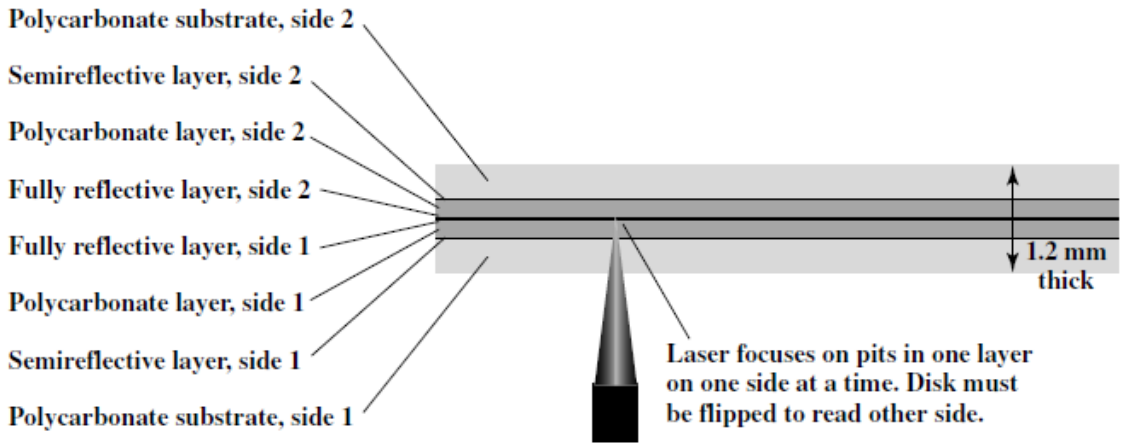
6.4.3.3. Digital Versatile Disk(DVD)

- The DVD's greater capacity is due to three differences than from CD's.

1. Bits are packed more closely on DVD. The spacing between loops of a spiral on CD is 1.64 μm & minimum distance between pits along spiral is 0.834 μm but for DVD loop spacing of 0.74 μm & minimum distance between pits 0.4 μm. capacity is 4.7 GB.
2. The DVD employs a second layer of pits and lands on top of the first layer. It has a semi-reflective layer on top and by adjusting focus the laser drive can read each layer separately. Almost doubles the capacity to 8.5GB.
3. The DVD-ROM can be two sided that could bring total capacity up to 17 GB.



(a) CD-ROM–Capacity 682 MB



(b) DVD-ROM, double-sided, dual-layer–Capacity 17 GB

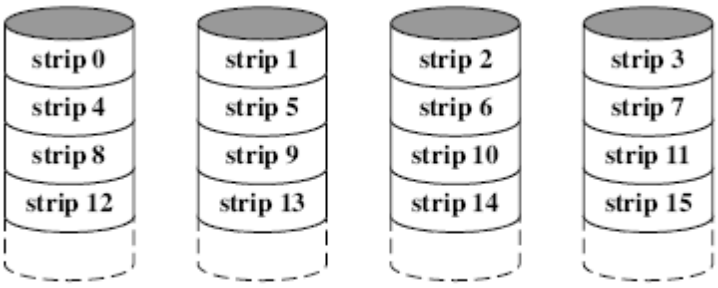
Figure 6.11: CD-ROM and DVD-ROM

6.4.4. RAID (Redundant Array of Independent Disks)

- Disk drive performance has not kept pace with improvements in other parts of the system.
- Limited in many cases by electro-mechanical transport means.
- This lead to the development of array of disks that operate independently and in parallel.
- With multiple disk separate I/O request can be handled in parallel and block of data can be accessed in parallel if distributed across multiple disks.
- With many parallel disk operating as if they are a single unit, redundancy techniques can be used to guard data loss in unit.
- There are 0-6 levels of RAID schemes.

RAID 0

- No redundancy techniques are used.
- Data is distributed over all disk in array.
- Data is divided into strips for actual storage, the divided data strips are stored in separate disks.
- RAID 0 for high data transfer capacity.
  - The strips of data can be accessed in parallel.
- RAID 0 for high I/O request rate.
  - If in case of multiple number of I/O request, if data is divided along the disks then more I/O request can be handled at a time.
- Can support low response time by having the block transfer size equal to strip, i.e. support multiple strip transfer in parallel.



(a) RAID 0 (Nonredundant)

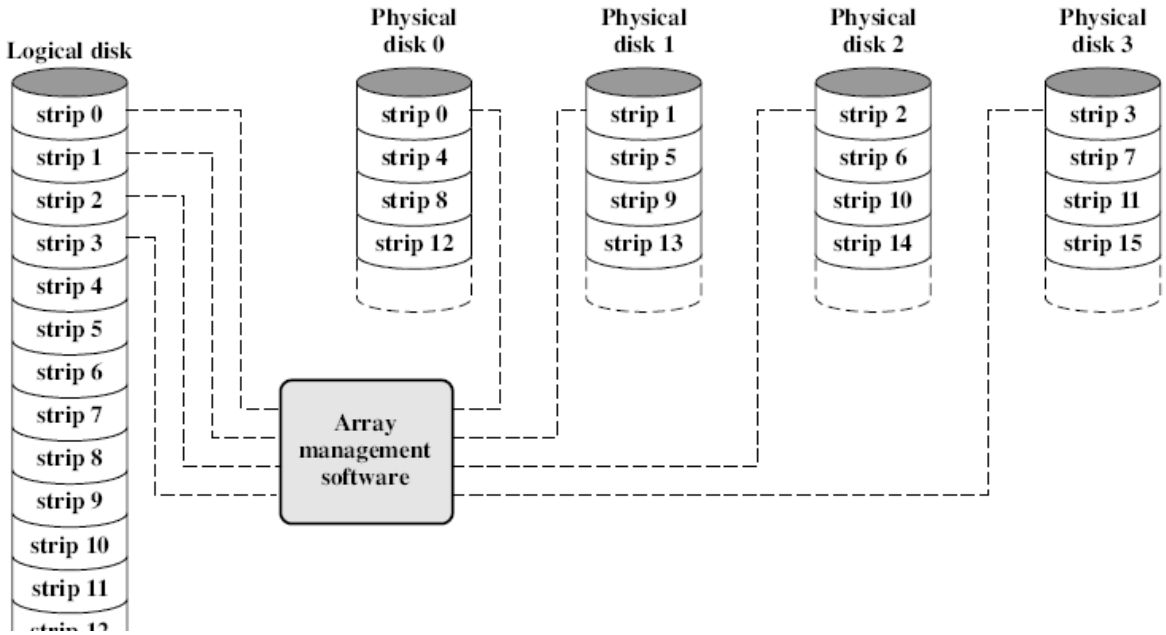
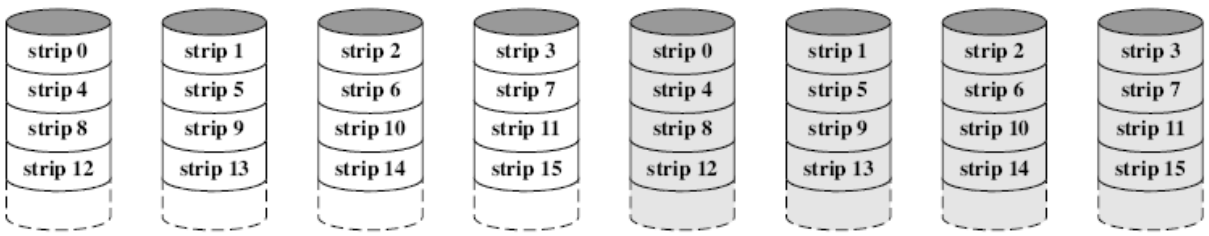


figure 6.12: data mapping for a RAID 0 array

RAID 1

- All disks are mirrored-duplicated, data is stored on a disk and its mirror disks.
- Read from either the disk or its mirror.
- Write must be done to both the data disk and mirror.
- Faulty recovery is easy, and can use data from mirror disk.
- Expensive.
- RAID 1 can achieve high I/O request rate if the bulk of the request are reads (because reading involves read from single disk where as any write operation require write to both disk)

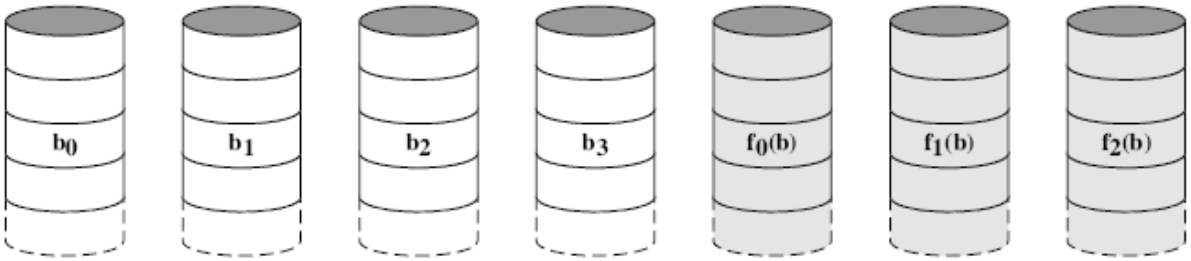


(b) RAID 1 (Mirrored)

RAID 2

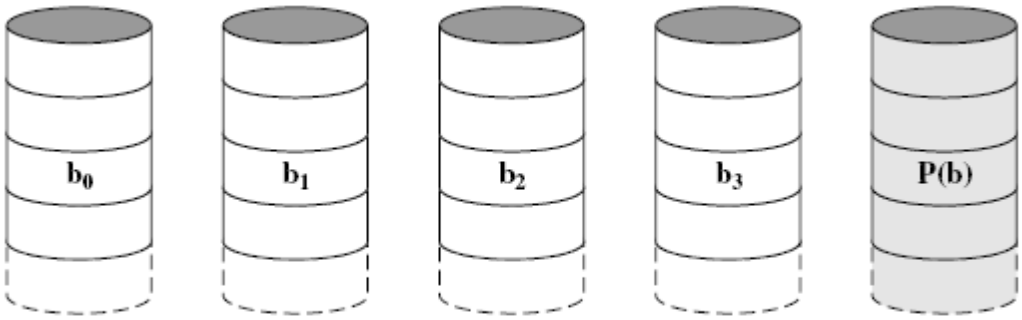
- All disks are used for every access. The spindles(disks) are synchronized so that each disk head is in the same position on each disk at any given time.
- Data strips are very small often as small as a single byte or word.
- An error correcting code are stored in the corresponding bit position on multiple parity disk (or additional disk).

- Typically hamming codes is used which is able to correct single bit error and detect double bit errors.
- RAID 2 require fewer disk than RAID 1, but still costlier.
- The number of additional disk is proportional to log of number of data disk.
- on single read, all disks are simultaneously accessed, the requested data and associated error correcting code are delivered to array controller.
- If any single bit error occurs, then controller can recognize and correct the error instantly.
- On single write, all data disk and additional disk must be accessed for write operation.



(c) RAID 2 (Redundancy through Hamming code)

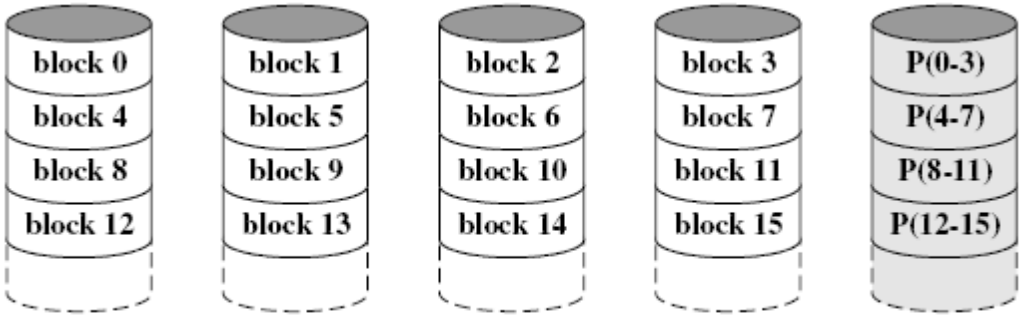
RAID 3



(d) RAID 3 (Bit-interleaved parity)

- RAID 3 is organized in a similar fashion to RAID 2.
- RAID 3 requires only a single redundant disk.
- RAID 3 employs parallel access with data distributes in small strips.
- A simple parity bit is computed for the set of individual bits in the same position on all data disk.
- Is a drive fails, the parity drive is accessed and data is reconstructed from remaining device, once the failed drive is replaced the missing data can be restored on the new drive and operation resumed.

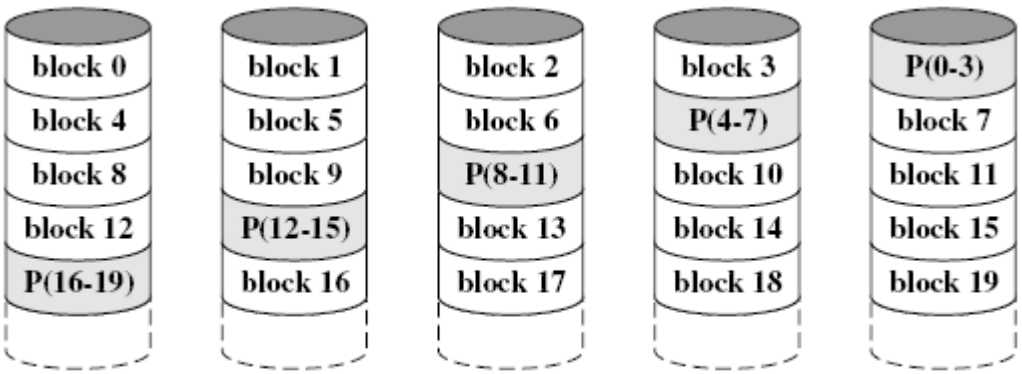
RAID 4



(e) RAID 4 (Block-level parity)

- RAID 4 make use of an independent access techniques, so that separate I/O request can be satisfied in parallel.
- Data striping is used and the strips are relatively large.
- A bit by bit parity strip is calculated across corresponding strips on each data and the parity bits are stored in the corresponding strip on parity disk.
- It involves a write penalty when an I/O Write request of small size is performed, for every write, the parity strip must also be recalculated and written. Thus one write on the disk needs two physical disk access. Thus the parity drive always written to and can cause a bottle neck.

RAID 5

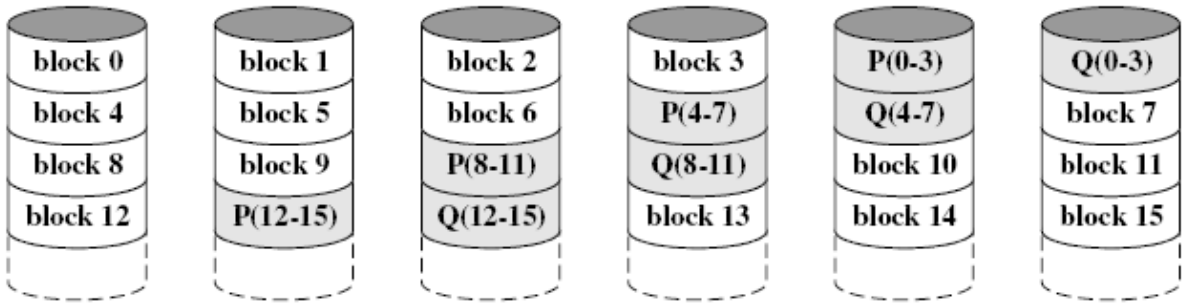


(f) RAID 5 (Block-level distributed parity)

- Organized in similar fashion to RAID 4, the difference is that RAID 5 distributes the parity strips across all disk.
- Strips arranged in round robin scheme.
- The distribution of parity strips across all drives avoids the potential I/O bottle neck found in RAID 4.

**a round robin is an arrangement of choosing all elements in a group equally I some order (circular order), simple way of thinking of round robin is that taking turns. The name of the algorithm comes from round-robin principle.

RAID 6



(g) RAID 6 (Dual redundancy)

- In RAID 6 two different parity calculation are carried out and stored in separate blocks on different disk.
- RAID 6 requires N+2 disk where N is the data disk.
- P & Q are two different check algorithm, one of the two is exclusive or calculation and other is an independent data check algorithm.
- This makes possible to generate data even if two disk fails.
- Advantage extremely high data availability.
- Disadvantage substantial write penalty because each write has written in two parity blocks.

6.4.5. Flash memory

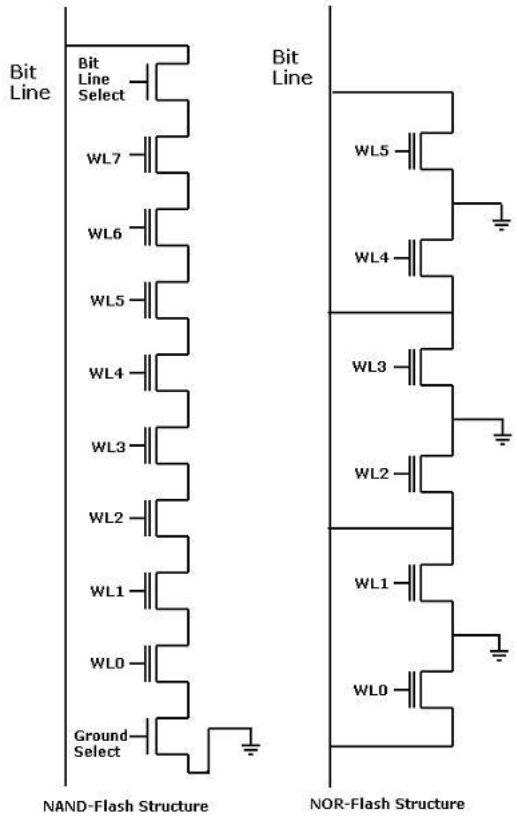
Flash memory technology is a mix of EPROM and EEPROM technologies. The term flash was chosen because if large chunk of memory could be erased at one time. The name therefore distinguishes flash device from EEPROMS where each byte is erased individually.

The more common elementary flash cell consists of one transistor with floating gate similar to an EPROM cell. The electrical functionality of the flash memory cell is similar to that of an EPROM or EEPROM, electrons are trapped onto the floating gate. These electrons modify the threshold voltage od the storage transistor.

Architecture

As other semiconductors the flash memory chip size is the major contributor to the cost of the device. Some of the array structure used are NOR, NAND.

NOR cell: The NOR architecture is currently the most popular flash architecture. It is commonly used in EPROM and EEPROM design aside from active transistor the largest contributor to area in the cell array is the metal to diffusion contact. NOR architecture requires one contact per two cells, which consumes the most of all flash area. The electrons trapping is floating gate is done by hot electrons injection. Electrons are removed by Fowler-Nordheim tunneling.

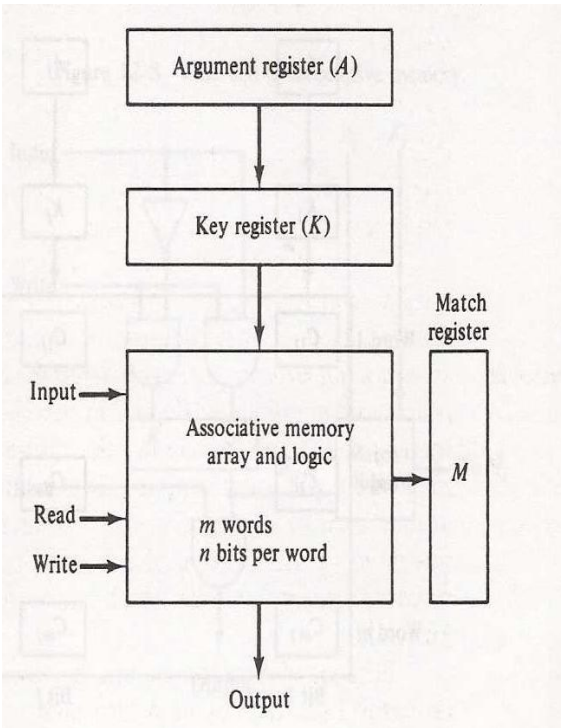


NAND cell

To reduce the cell area, the NAND configuration was developed. The NAND configuration is considerably is that when a cell is read, the sense amplifier sees a weak signal than the on NOR configuration since several transistors in series are used. The weak signal slows down the speed of the read circuitry which can be overcome by operating in series access method. This memory will not be competitive for random access applications.

6.5. Associative memory

Many data processing applications require the search of items in a table stored in memory. An assembler program searches the symbol address table in order to extract the symbol's binary equivalent. The established way to search a table is to store all items where they can be addressed in sequence. The search procedure is strategic for choosing a sequence of address reading the content of memory at each address and comparing the information read with the item being searched until a match occurs. The number of access to memory



depends upon on the location of the item and efficiently of search algorithm.

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of data itself rather than by an address. A memory unit accessed by content is called an associative memory or Content Addressable Memory(CAM). This type of memory accessed simultaneously and in parallel on the basis of the data content rather than by specific address or location. When word is written in an associative memory no address is given and word are stored in empty unused location. When a word is to be read from an associative memory, the content of word or part of the word is specified. The memory allocates 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. Searches can be done on entire word or specified word. An associative memory is more expensive than random access memory because each cell must have storage capacity as well as logic circuit for matching its content with external argument. For this reason, associative memory is used in application where the search time is very critical and must be very short.

6.5.1. Hardware organization

The block diagram of an associative memory is shown in figure. It consists of a memory array and logic for m words with n bits per word. The argument registers A and key register k each have n bits one for each bit word. The matching register M has m bits, one for each memory word in memory is compared in parallel with the content of the argument register and **set** the corresponding match register if there is a match. Reading is accomplished by 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 comparison is only done to those bits in the argument that have 1's in their corresponding position of the key register.

A numerical example:

A	101	111100	
K	111	000000	
Word1	100	111110	no match
Word2	101	000001	match

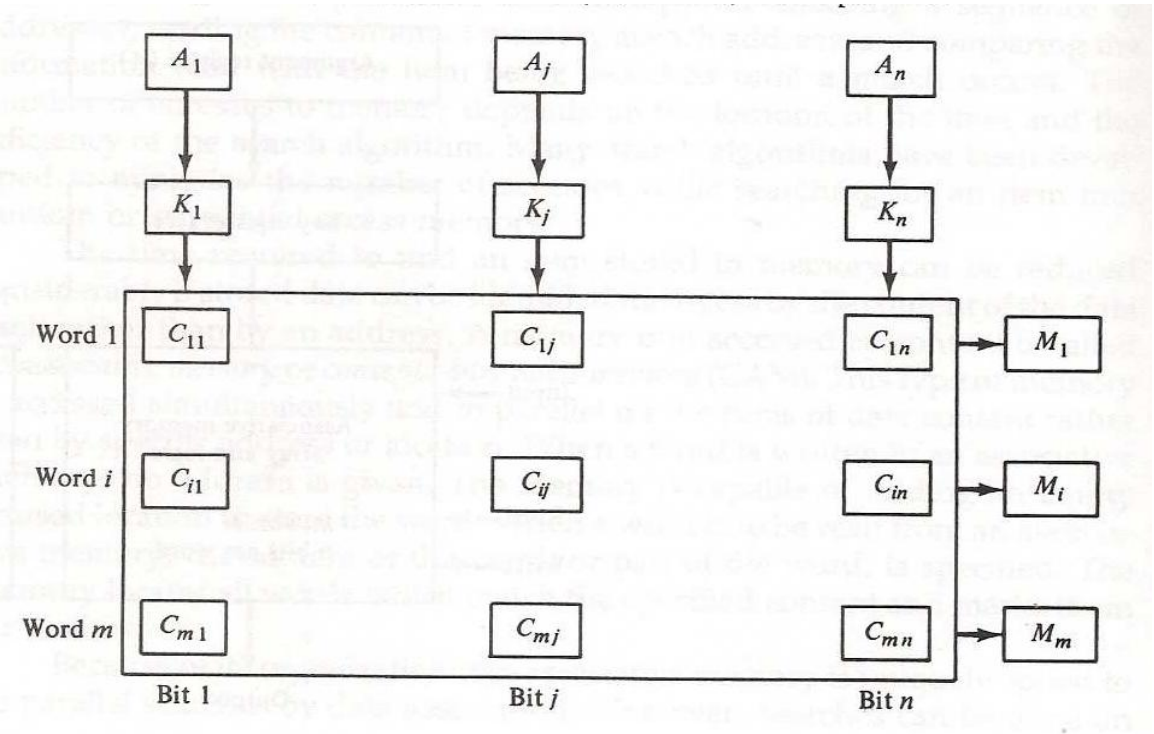


Figure: associative memory of m words, n cells per word

The relation between the memory array and external register in an associative memory is shown above. The cells in the array are marked by letter C with two subscripts the first subscript gives the word number and the second specified the bit position in the word. A_j in the augment register is compared with all bits in column j of the array provided that k_j =1. This is done for all columns j=1,2...n. if a match occurs between all the unmasked bits of the argument and bits in word i, the corresponding bits M_i in the match register is set 1. If one or more unmasked bits if the argument and word do not match M_j is cleared to 0. The internal organization of a typical cell c_{ij} is as shown in figure below. It consists of flip-flops storage elements F_{ij} and circuits for writing reading and matching the cell. The input bit is transferred into the storage cell during a write operation. The bit stored is read out during read operation. The match logic compares the content of the storage cells with the corresponding unmasked bit if the argument and provides an output for the decision logic that sets the bits in M.

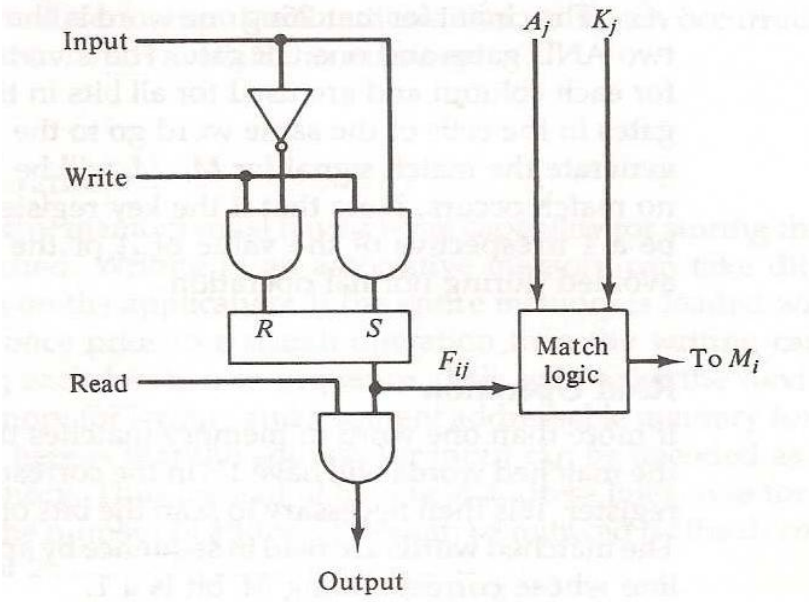


Figure: One cell of associative memory

6.5.2. Match logic

The match logic for each word can be derived from the comparison algorithm for two binary numbers. First we neglect the key bits and compare the argument in A with the bits stored in the cells of words. Word i is equal to the argument in A if $A_j=F_{ij}$ for j=1,2 ...n. two bits are equal if they are both 1 0r 0. The equality of two bits can be expressed logically by the Boolean function.

$$x_i = A_j F_{ij} + A'_j F'_{ij}$$

Where $x_i=1$, if the pair of bits in position j are equal.

The condition for setting the corresponding match bits M_i to 1. The Boolean function for this condition is $M_i = x_1 x_2 x_3 \dots x_n$

And constitute the AND operations of all pairs of matched bits in a word.

Now for key bits, the requirement is that if $k_j = 0$, the corresponding bits of A_j and F_{ij} needs no comparison, only when $k_j = 1$, must they be compared. This requirement is achieved by Oring each item with k'_j

Thus,

$$x_j + k'_j = \begin{cases} x_j & if\ k_j = 1 \\ 1 & if\ k_j = 0 \end{cases}$$

A term $(x_j + k'_j)$ will be in the 1 state if it's pairs of bits is not compared. This is necessary because each termed is ANDed with all other terms so that an output of 1 will have no effect. The comparison of the bits has an effect only when $k_j = 1$.

The match logic for word i in an associative memory can now be expressed by the following Boolean expressing.

$$M_i = (x_1 + K'_1) + (x_2 + K'_2) + \cdots + (x_n + K'_n)$$

Each term in the expression will be equal to 1, if the corresponding k_j=0, if k_j=1, the term will be either 0 or 1 depending on the value of x_j. A match will occur and M_j will be 1, if all terms are equal to 1.

$M_j = \prod_{j=1}^{n} = (A_j F_{ij} + A'_j F'_{ij} + k'_j)$

6.5.3. Read Write logic

Read operations

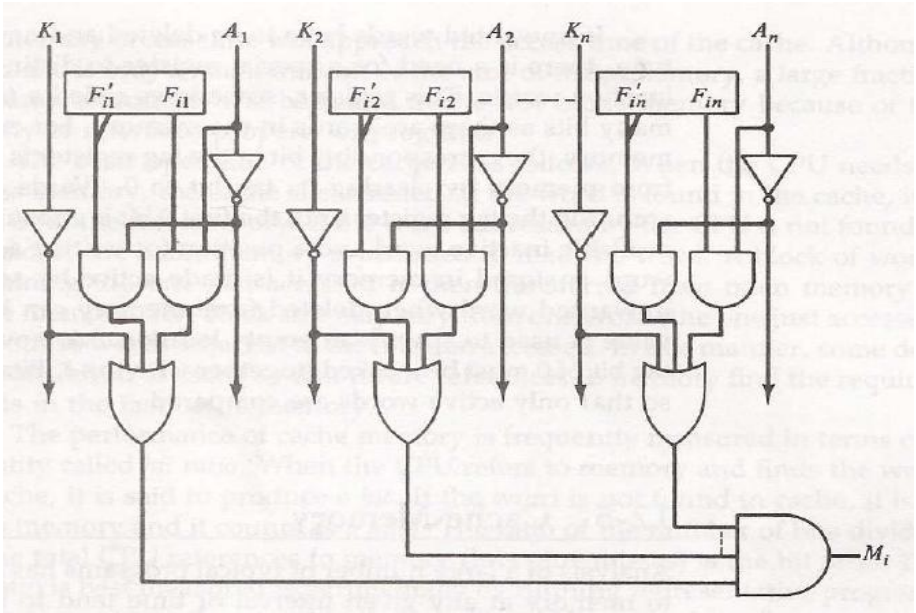


Figure: Match logic for one word of associative memory

If more than one word in memory matches the unmasked argument field, all the matched words will have 1's in the corresponding bit position of the match register. It is then necessary to scan the bits of the match register one at a time. The matched word is read in sequence by applying a read signal to each word line whose M_j bit is 1.

In most application the associative memory stores a table with no two identical items under a given key. In this case only one word may match the unmasked argument field. By connecting output M_j directly to the read line in the same word position, the content of the match word will be presented automatically at the output lines and no special read command signal is needed. Furthermore, if we exclude words having zero content, an all zero output will indicate the no match occurred and that the searched item is not available in memory.

Note that if the key register contains all 0's and outputs M_j will be 1, irrespective of the value of A or the word. This occurrence must be avoided during normal operation.

Write operation

Writing in an associative memory can take different forms, depending on the application. If the entire memory is loaded with new information at once prior to a search operation, then the writing can be done by addressing each location in sequence. This will make the device random access memory for reading. The advantage here is the address for input can be decoded as in random access memory. Thus instead of having m address line one for each word in memory, the number of address line can be reduced by the decoder to d lines where $m = 2^d$

If unwanted words have to be deleted and new words inserted one at a time, there is a need for a special register to distinguish between active and inactive words. This register sometimes called tag register. For active word stored corresponding tag register is 1, a word is deleted by clearing the tag bit to 0. After the new word is stored in memory in memory it is made active by setting its tag bit to 1.

6.6. Cache memory

A cache memory is a small very fast memory that retains copies of recently used information from main memory. It operates transparent to the programmer automatically decoding which values to be kept and which to be overwrite.

The processor operates at high clock rates but the memory is slow so it requires cache. The overall system performance depends on the successful access from cache. An access to an item which is in cache is called a *hit* and an access to an item which is not in the cache is called a *miss*.

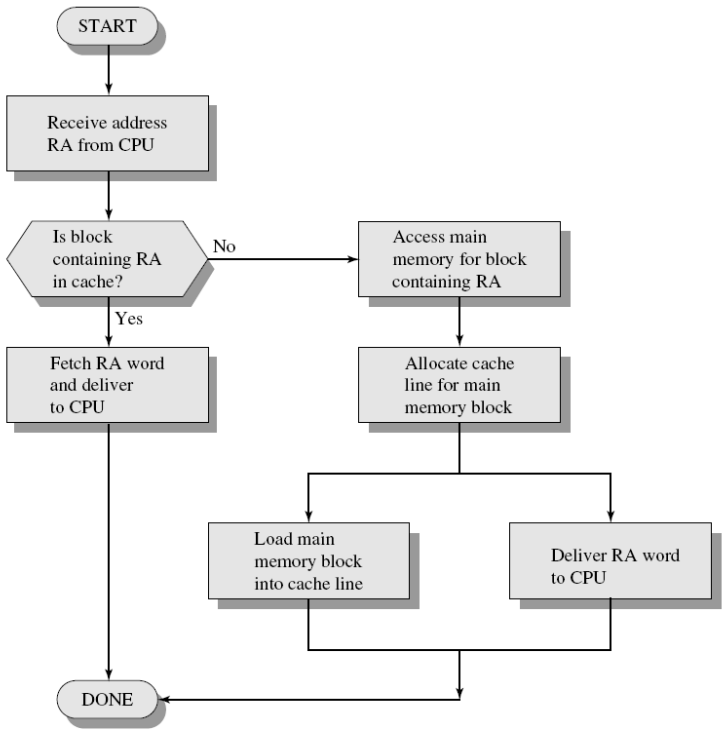


Figure: shows the flow chart of read operation in cache

Typical cache organization

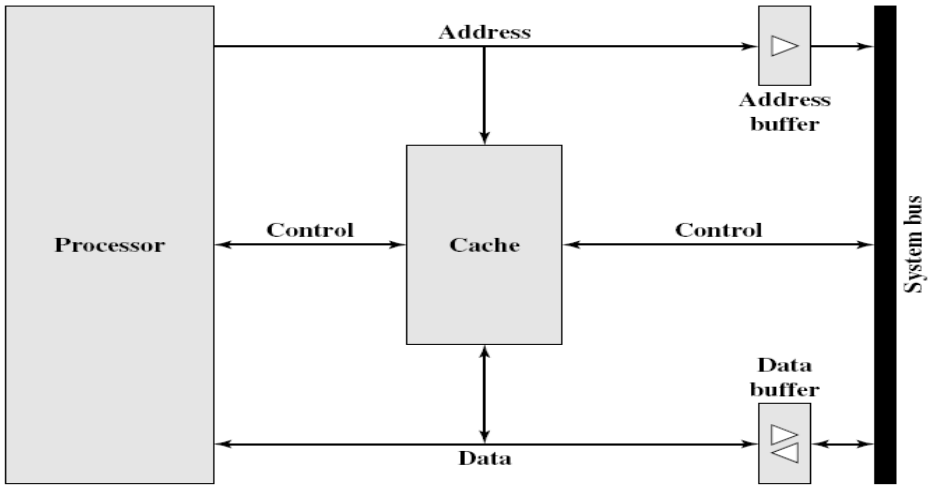


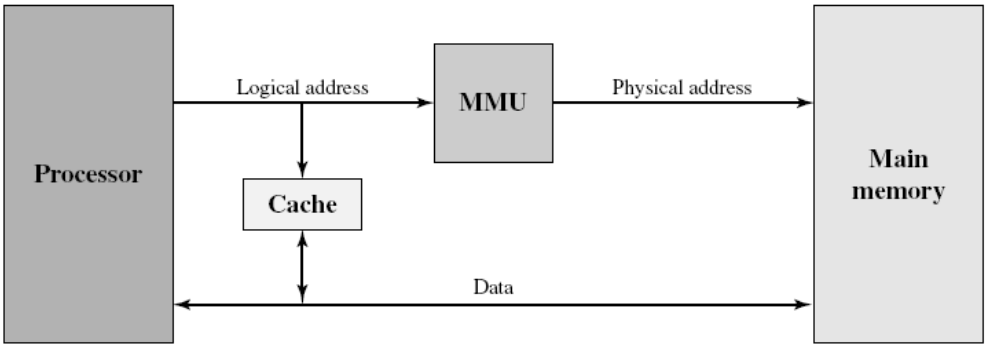
Figure: Typical cache organization

Figure shows the cache organization. In this organization, the cache connects to the processor via data, control, and address lines. The data and address lines also attach to data and address buffers, which attach to a system bus from which main memory is reached. When a cache hit occurs, the data and address buffers are disabled and communication is only between processor and cache, with no system bus traffic. When a cache miss occurs, the desired address is loaded onto the system bus and the data are returned through the data buffer to both the cache and the processor.

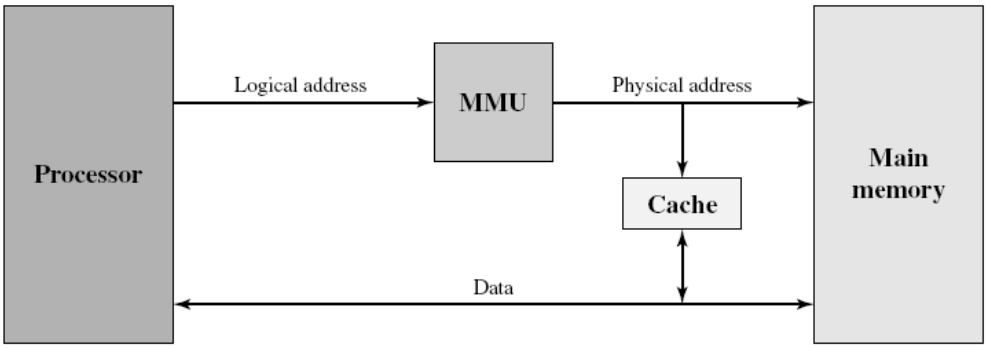
6.6.1. Elements of cache design:

a. Cache address:

A virtual memory is a facility that allows programs to address memory from a logical point of view, without regard to the amount of main memory physically available. When virtual memory is used, the address fields of machine instructions contain virtual addresses. For reads to and writes from main memory it needs a hardware memory management unit (MMU) translates each virtual address into a physical address in main memory. A **logical cache**, also known as a **virtual cache**, stores data using **virtual addresses**. The processor accesses the cache directly, without going through the MMU. A physical cache stores data using main memory **physical addresses** as shown in figure for logical address the cache is inserted in between the processor and MMU and foe physical address between MMU and main memory.



(a) Logical cache



(b) Physical cache

One obvious advantage of the logical cache is that cache access speed is faster than for a physical cache, because the cache can respond before the MMU performs an address translation. The disadvantage has to do with the fact that a single virtual address can be used for many spaces (physical address) for different application. So either it needs to completely flush the cache memory or add extra bits to identify the exact location.

b. Cache size:

For faster operation of the system we want higher cache size but as the size increases the cost per bit also increases, so there is a tradeoff between the size and speed. Hence is the designers wish how much cache to be used. Following table shows the different cache sizes used for different systems.

Processor	Type	Year of Introduction	L1 Cache[a]	L2 Cache	L3 Cache
IBM 360/85	Mainframe	1968	16 to 32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128 to 256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256 to 512 KB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 KB to 1 MB	2 MB
IBM S/390 G4	Mainframe	1997	32 kB	256 KB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTA[b]	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 KB	4 MB
SGI Origin 2001	High-end server	2001	32 kB/32 kB	4 MB	—
Itanium 2	PC/server	2002	32 kB	256 KB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1 MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24–48 MB

c. Mapping function:

For fast memory access there should be no time lost in searching for words in the cache for this mapping is used. The transformation of data from main memory to cache memory is referred to as mapping process. There are three types of mapping:

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

To explain the concept of mapping first consider a specific example of a memory organization the main memory can store 32k words of 12 bits each. The cache is capable of storing 512 of these words at any given time. for every word there is a duplicate copy on main memory. The CPU first sends 15-bit address to cache if there is a hit, the CPU accepts 12-bit data from cache if there is a miss the CPU read the word from main memory and the word is then transferred to cache.

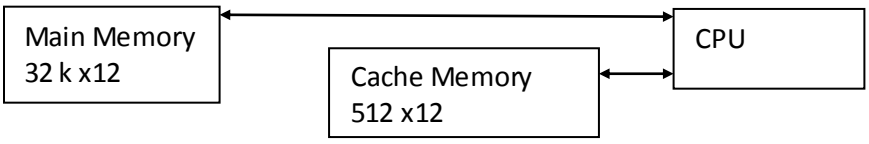
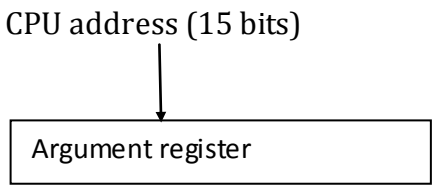


Figure: example of cache memory

1. Associative mapping:



Address	Data
01000	3450
02777	6710
22345	1234

Figure: associative mapping

- The fastest and most flexible cache organization.
- Stores both address and data of memory word.
- The address value of 15 bits is shown as a five-digit octal number and its 12-bit word is shown as a four-digit octal number.
- A CPU address of 15 bits is placed in the argumented register and the associative memory is searched for matching address, if address is found the corresponding 12 bit is read and sent to CPU.
- If no match occurs the main memory is accessed for word also the address data pair must be placed in cache.
- If cache is full then what value to be displaced is determined by replacement algorithms

2. Direct mapping:

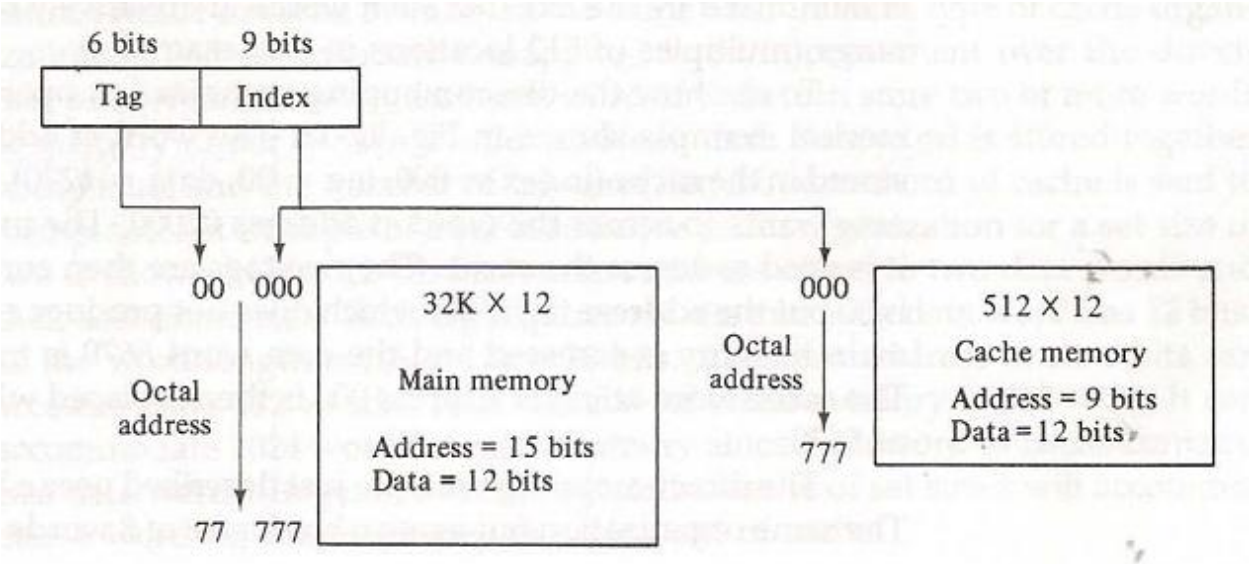


Figure: address relationship between main memory and cache

- The CPU address is divided into two parts:
  a. Index field
  b. Tag field

  Here in example 9 bits as index and 6 bits as tag.
- The number of bits in the index field is equal to the number of address bits required to access cache memory.
- When CPU generates a memory request the index field is used as the address to access the cache.
- Then tag field in of CPU is compared with the tag in the word read from the cache, if tag match then there is a hit and desired data word is in cache. If no match, then miss and required word is read from main memory.
- It also stores the value and tag in cache or replaces the old ones.

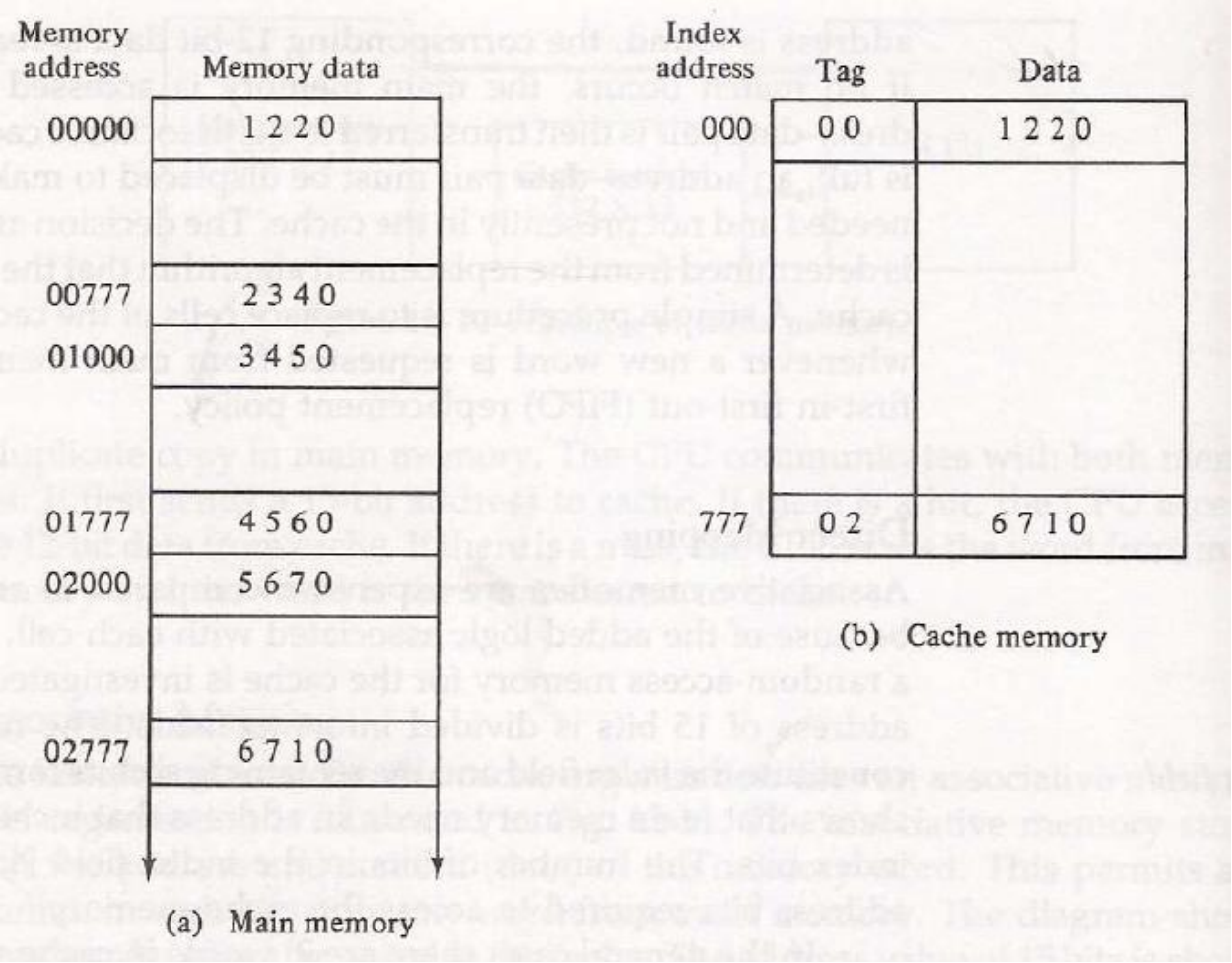


Figure: direct mapping cache organization

Disadvantage:

If hit rate drops considerably if two or more words whose address have the same index but different tags are accessed repeatedly.

** see direct mapping example of how direct mapping done using numerical value from Morris mano book.

3. Set- Associate Mapping
    - The disadvantage of direct mapping is it cannot store two words of same index with different tags.
    - In set associate mapping two or more words of memory under same index address with different tags can be stored.
    - Each data word is stored together with its tag and the number of tag-data item in one word of cache is said to form a set.
    - For searching, CPU sends address first it searches for matched index in cache, after matching index it then matches the tag. For matched tag data is sent to CPU, if no match is found it is searched from main memory

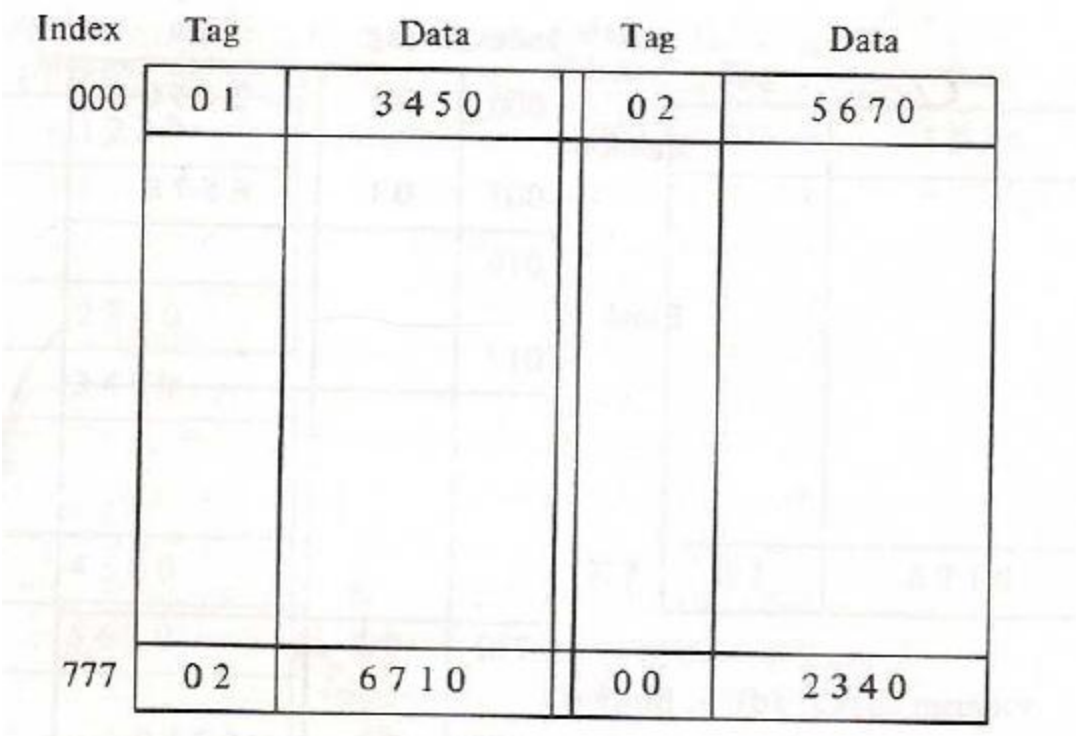


Figure: two way set associative mapping cache.

- The hit ratio will improve as the set size increases because more words with same index but different tags can reside in cache.

6.6.2. Replacement algorithms

The most common replacement algorithms are:

- Random replacement: replacements are done at random.
- First-In-First –Out(FIFO): replaces the item that has been in the set the longest.
- Least Recently Used(LRU): replaces the item that have been least recently used.

Writing into cache:

During read operation the main memory is not involved in transfer. For write operation it can be done in two ways:

a. Write through: in this to update main memory with every memory write operation, the cache memory also is updated in parallel. Advantage of this is that main memory always contains the same data(valid) as cache.

b. Write back: in this only the cache location is updated during write operation. The location is then marked by a flag so that when the word is removed from cache it is copied to main memory. Advantage is that during the time the word resides in the cache it may be updated number of times and will be used from cache only so it needs to be updated in main memory only if it is being removed from cache.

Cache initialization

The cache is initialized when the power is applied to the computer or when the main memory is loaded with a complete set of programs. After initialization the cache is considered to be empty, but in fact it contains some non-valid data. it is customary to include each word with valid bit to indicate whether the word is valid or not.

The cache is initialized by clearing all the valid bits to 0. The valid bit of a particular cache is set to 1 the first time this word is loaded from main memory and stays this way unless it is re initialized. The use of valid bit is that the replacement of data in cache is done for only those data whose valid bit is 0.

*** the cache used can be of single level or multiple levels depending upon the design and performance required.

6.7. Memory Management Hardware

In multiprogramming environment where many programs reside in memory it becomes necessary to move programs and data around the memory to vary the amount of memory in use by a given program and to prevent program from changing other programs. A memory management system is a collection of hardware and software procedures for managing the various programs residing in memory.

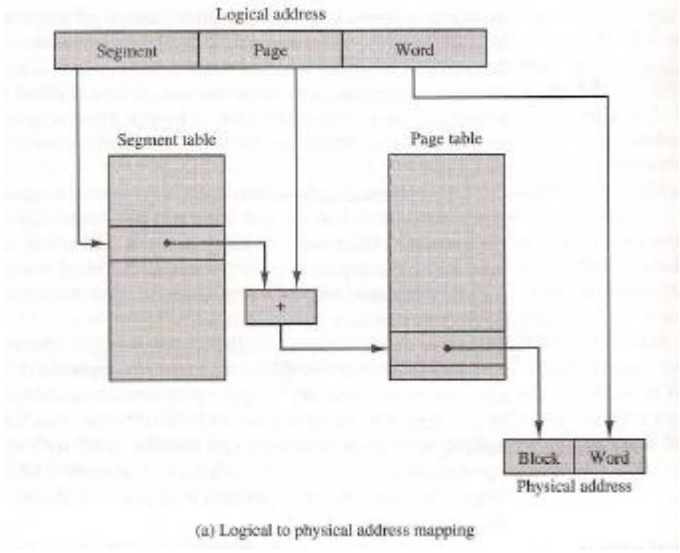
The basic components of a memory management unit are:

1. A facility for dynamic storage location that maps logical memory references into physical memory address.
2. A provision for sharing common programs stored in memory by different users.
3. Protection of information against unauthorized access between users and preventing users from changing operating system function.

    1. The dynamic storage relocation hardware is mapping process similar to paging system. It is more convenient to divide programs and data into logical parts called segments. A segments is a set of logically related instruction or data elements associated with a given name. segments may be generated by programmer or by operating system.

       The address generated by a segmented program is called logical address. Logical address space is associated with variable length segments rather than fixed length pages. The logical address may be equal or differ than actual physical address. Shared programs are placed in a unique segment in each user's logical address space so that a single physical copy can be shared.

Segmented page mapping



(a) Logical to physical address mapping

Consider the logical address as shown in figure. The logical address is partitioned into three fields. The segment field specifies a segment number, the page field specifies the page within the segment and word field gives the specific word within the page.

The mapping of the logical address into a physical address is done by means of two tables a & b. the segment number of the logical address specifies the address for the segment table. The entry in the segment table is pointer for page table base. The page table base is added to the page number given in logical address. The sum produces a pointer address to an entry in the page table. The value found in the page table provides the block number in physical memory.

In this process memory reference from CPU will require three access to memory. One from segment table, one from page table and third from main memory. This will slow the system significantly. Alternative approach will be to use Translation Lookaside Buffer(LTB). The first time a given block is referenced its value together with the corresponding segment and page number are entered into associative memory. Thus mapping process is first attempted by associative search with given segment and page number.

** for numerical example see Morris mano book.

2. The sharing of common programs is an integral part of a multiprogramming system. For example, several users may wish to use compiler that is shared between them. Other systems programs residing in the memory are also shared by all users in a multiprogramming system without having to produce multiple copies, this type of management is done by memory management hardware.

3. Memory protection

Memory protection can be assigned to the physical address or logical address. The protection of memory through the physical address can be done by assigning to each block in memory a number of protection bits. Every time a page is moved from one block to another it would be necessary update the block protection bits. A much better place to apply protection is in logical address space rather than the physical address space. This can be done by including protection information within the segment table or segment register of memory management hardware.

The content of each entry in the segment table or a segment register is called descriptor. A typical descriptor would contain in addition to base address field one or two additional field for protection purpose.

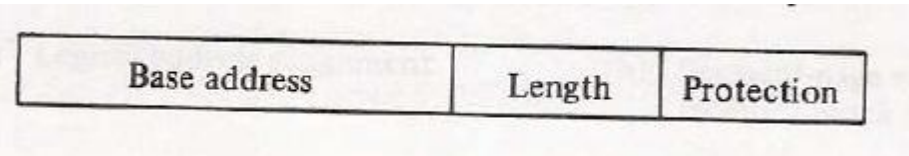


Figure: format of typical segment descriptor

The base address field give the base of the page table address in a segmented page organization or the block base address in segmented register organization. This address is used in mapping from logical to physical address. The length field give the segment size by specifying the maximum number of pages assigned to the segment. A size violation occurs if the page number falls outside the segment length boundary. Thus a given program and its data cannot access memory not assigned to it by the operating system.

The protection field in a segment descriptor specifies the access right available to the particular segment. Some of the access rights that are used for protecting the programs residing in memory are:

1. Full read & write privileges: given to a program when it is executing its own instructions.
2. Write protection: useful for sharing system programs such as utility programs & other library routines.
3. Execution only: protects programs from being copied.
4. System only: occasional user from accessing operation system segments.