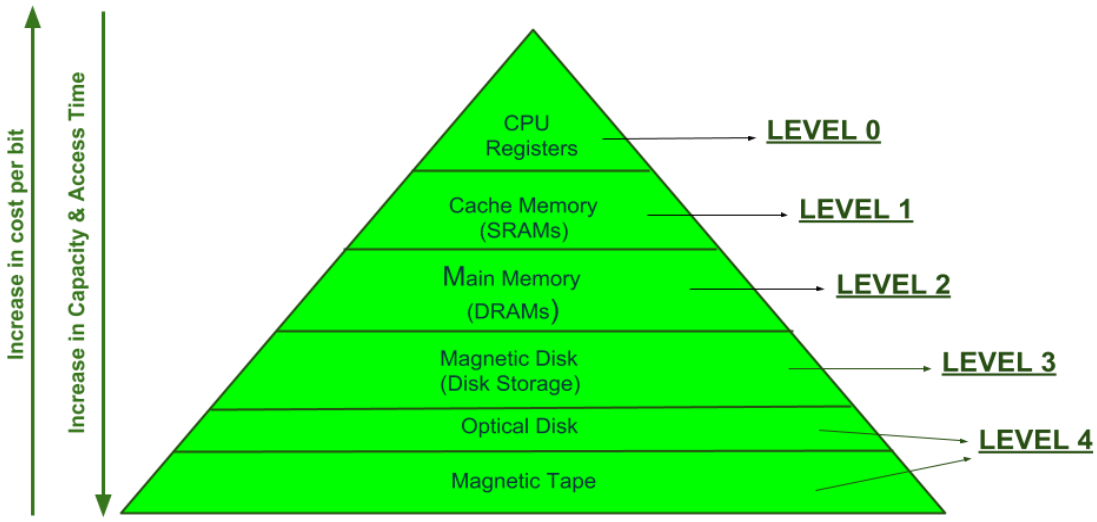
Unit-4: Memory Organization

Memory Hierarchy: The memory in a computer can be divided into five hierarchies based on the speed as well as use. The processor can move from one level to another based on its requirements. The five hierarchies in the memory are registers, cache, main memory, magnetic discs, and magnetic tapes. The first three hierarchies are volatile memories which mean when there is no power, and then automatically they lose their stored data. Whereas the last two hierarchies are not volatile which means they store the data permanently.

Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references. The figure below clearly demonstrates the different levels of memory hierarchy:



MEMORY HIERARCHY DESIGN

This Memory Hierarchy Design is divided into 2 main types:

External Memory or Secondary Memory –

Comprising of Magnetic Disk, Optical Disk, and Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

Internal Memory or Primary Memory –

Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

We can infer the following characteristics of Memory Hierarchy Design from above figure:

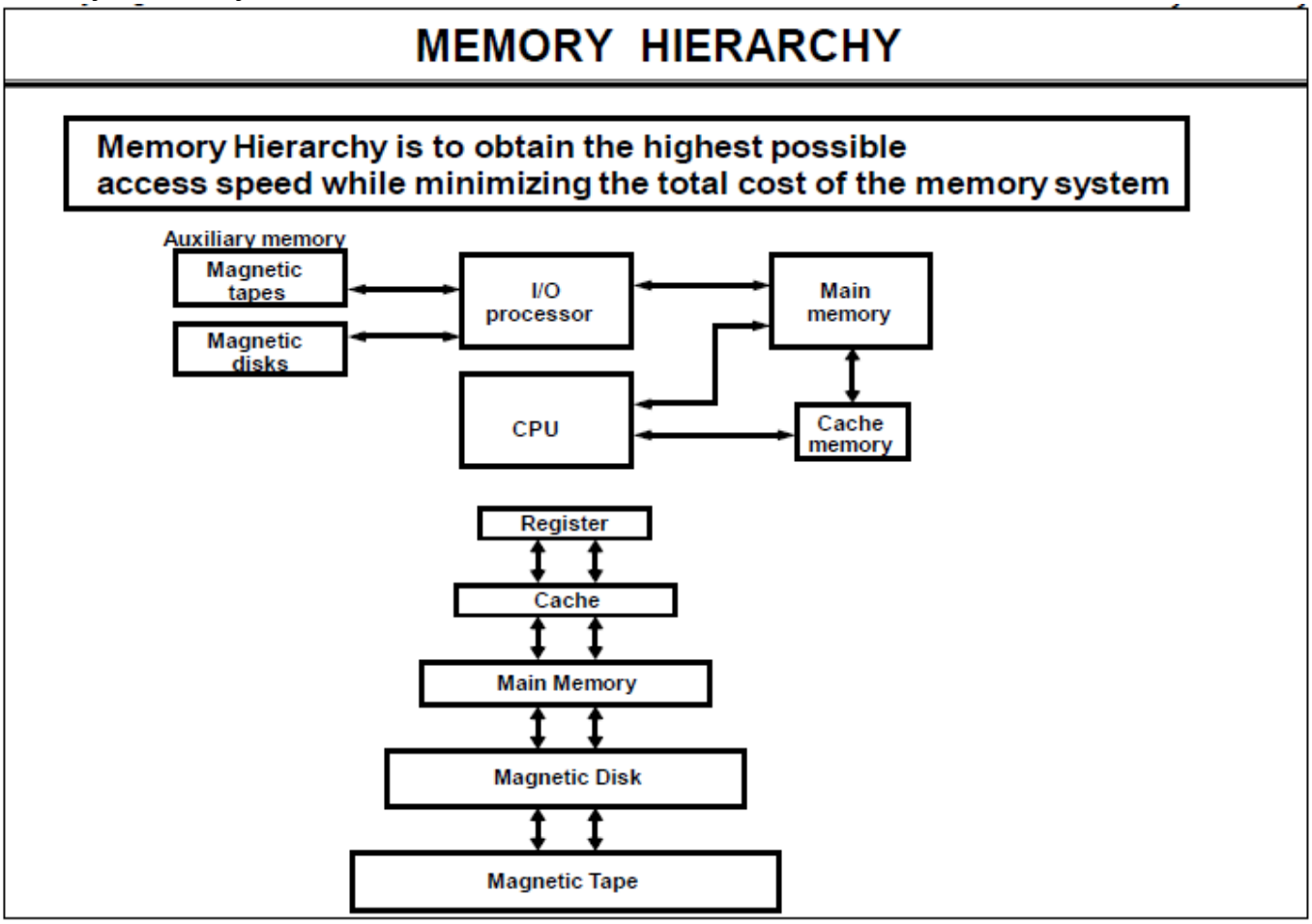
Capacity:
It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.

Access Time:
It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.

Performance:
Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.

Cost per bit:
As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

Memory Hierarchy:



- The memory unit that communicates 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 systems are magnetic disks and tapes. They are used for storing system. Programs, large data files, and other backup information.
- Only programs and data currently needed by the processor reside in main memory. All other information is stored in auxiliary memory and transferred to main memory when needed.
- A special very-high·speed memory called a **cache** is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate.
- The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic.

- CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory.
- A technique used to compensate for the mismatch in operating speeds is to employ an extremely fast, small cache between the CPU and main memory whose access time Is dose to processor logic dock cycle time.
- The cache is used for storing segments of programs currently being executed in the CPU and temporary data frequently needed in the present calculations. By making programs and data available at a rapid rate, it is possible to increase the performance rate of the computer.
- While the I/O processor manages data transfers between auxiliary memory and main memory, the cache organization is concerned with the transfer of information between main memory and CPU.
- *The reason for having two or three levels of memory hierarchy is economics. As the storage capacity of the memory increases, the cost per bit for storing binary information decreases and the access time of the memory becomes longer. The auxiliary memory has a large storage capacity, is relatively inexpensive, but has low access speed compared to main memory. The cache memory is very small, relatively expensive, and has very high access speed. Thus as the memory access speed increases, so does its relative cost.*
- The overall goal of using a memory hierarchy is to obtain the highest-possible average access speed while minimizing the total cost of the entire memory system.
- The transfer from auxiliary to main memory is usually done by means of direct memory access of large blocks of data.
- The typical access time ratio between cache and main memory is about 1 to 7. For example, a typical cache memory may have an access time of 100 ns, while main memory access time may be 700 ns. Auxiliary memory average access time is usually 1000 times that of main memory.

➢ Main Memory

➢ *RAM-Random Access Memory*

- The main memory is the central storage unit in a computer system.
- Primary memory holds only those data and instructions on which computer is currently working.
- It has limited capacity and data is lost when power is switched off.
- These memories are not as fast as registers.
- The data and instruction required to be processed reside in main memory.
- The principal technology used for the main memory is based on semiconductor integrated circuits.

- Integrated circuit RAM chips are available in two possible operating modes, *static RAM and dynamic RAM.*
- Static RAM: Data is stored in transistors and requires a constant power flow. The stored information remains valid as long as power is applied to the unit. Because of the continuous power, SRAM doesn't need to be refreshed to remember the data being stored. SRAM is called static as no change or action i.e. refreshing is not needed to keep the data intact. It is used in cache memories.
- Dynamic RAM: Data is stored in capacitors. Capacitors that store data in DRAM gradually discharge energy, no energy means the data has been lost. So, a periodic refresh of power is required in order to function. DRAM is called dynamic as constant change or action (change is continuously happening) i.e. refreshing is needed to keep the data intact. It is used to implement main memory.
- Refreshing is done by cycling through the words every few milliseconds to restore the decaying charge.

The difference between SRAM and DRAM are as follows:

SRAM	DRAM
It stores information as long as the power is supplied.	It stores information as long as the power is supplied or a few milliseconds when power is switched off.
Transistors are used to store information in SRAM.	Capacitors are used to store data in DRAM.
Capacitors are not used hence no refreshing is required.	To store information for a longer time, contents of the capacitor need to be refreshed periodically.
SRAM is faster compared to DRAM.	DRAM provides slow access speeds.
It does not have a refreshing unit.	It has a refreshing unit.
These are expensive.	These are cheaper.
SRAMs are low-density devices.	DRAMs are high-density devices.
In this bits are stored in voltage form.	In this bits are stored in the form of electric energy.
These are used in cache memories.	These are used in main memories.
Consumes less power and generates less heat.	Uses more power and generates more heat.

➢ *ROM-Read Only Memory*

- ROM, which stands for read only memory, is a semiconductor memory device or storage medium that stores information permanently.
- It is called read only memory as we can only read the programs and data stored on it but cannot write on it.
- It is restricted to reading words that are permanently stored within the unit.
- In view of this it is used where data needs to be stored permanently, even when the power is removed - many memory technologies lose the data once the power is removed.
- This type of semiconductor memory technology is widely used for storing programs and data that must survive when a computer or processor is powered down. For example the BIOS of a computer will be stored in ROM.
- *For example, when you start your computer, the screen does not appear instantly. It takes time to appear as there are startup instructions stored in ROM which are required to start the computer during the booting process. The work of the booting process is to start the computer. It loads the operating system into the main memory (RAM) installed on your computer. The BIOS program, which is also present in the computer memory (ROM) is used by the microprocessor of the computer to start the computer during the booting process. It allows you to open the computer and connects the computer with the operating system.*
- *ROM is also used to store Firmware, which is a software program which remains attached to the hardware or programmed on a hardware device like a keyboard, hard drive, video cards, etc. It is stored in the flash ROM of a hardware device. It provides instructions to the device to communicate and interact with other devices.*
- The process of loading the data in the ROM is known as programming. The way in which ROM is programmed further classifies it.

1. Programmable Read Only Memory (PROM):

- PROM is a blank version of ROM. It is manufactured as blank memory and programmed after manufacturing.
- We can say that it is kept blank at the time of manufacturing. You can purchase and then program it once using a special tool called a programmer.
- To write data onto a PROM chip; a device called PROM programmer or PROM burner is used. The process or programming a PROM is known as burning the PROM. Once it is programmed, the data cannot be modified later, so it is also called as one-time programmable device.

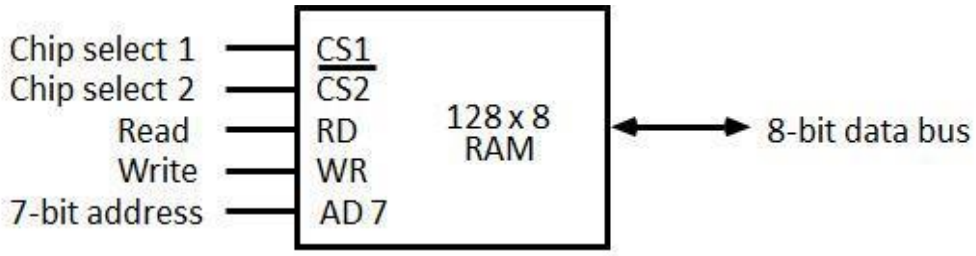
2. Erasable and Programmable Read Only Memory (EPROM):

- This is an Erasable Programmable Read Only Memory. These semiconductor devices can be programmed and then erased at a later time. This is normally achieved by exposing the semiconductor device itself to ultraviolet light. The memory-erasing time lies between 10 to 30 minutes.

3. Electrically Erasable and Programmable Read Only Memory (EEPROM):

- ROM is a type of read only memory that can be erased and reprogrammed repeatedly, up to 10000 times. It is also known as Flash EEPROM as it is similar to flash memory. It is erased and reprogrammed electrically without using ultraviolet light. Access time is between 45 and 200 nanoseconds.

➢ RAM and ROM Chips



Typical RAM chip

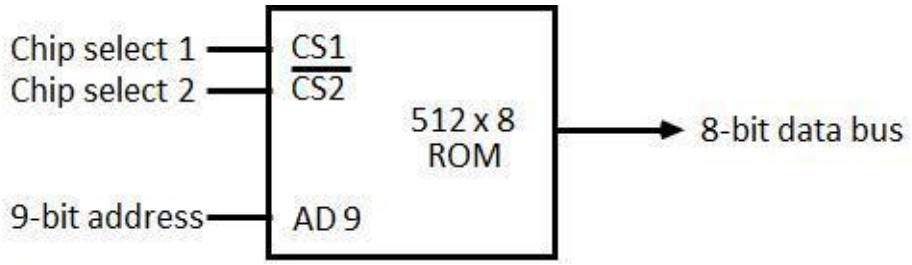
- The capacity of the memory is 128 words of eight bits (one byte) per word. This requires a 7-bit address and an 8-bit bidirectional data bus.

CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	×	×	Inhibit	High-impedance
0	1	×	×	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	×	Read	Output data from RAM
1	1	×	×	Inhibit	High-impedance

- The function table specifies the operation of the RAM chip. The unit is in operation only when CS1 = 1 and $\overline{CS2}$ = 0.
- The bar on top of the second select variable indicates that this input is enabled when it is equal to 0. If the chip select inputs are not enabled, or if they are enabled but the read or

write inputs are not enabled, the memory is inhibited and its data bus is in a high-impedance state.

- When CS1 = 1 and $\overline{CS2}$ = 0, the memory can be placed in a write or read mode.
- When the WR input is enabled, the memory stores a byte from the data bus into a location specified by the address input lines.
- When the RD input is enabled, the content of the selected byte is placed into the data bus. The RD and WR signals control the memory operation as well as the bus buffers associated with the bidirectional data bus.



Typical ROM chip

- The nine address lines in the ROM chip specify any one of the 512 bytes stored in it.
- The two chip select inputs must be CS1 = 1 and $\overline{CS2}$ = 0 for the unit to operate. Otherwise, the data bus is in a high-impedance state.
- There is no need for a read or write control because the unit can only read. Thus when the chip is enabled by the two select inputs, the byte selected by the address lines appears on the data bus.

➢ **Memory Address map of RAM and ROM.**

- The designer of a computer system must calculate the amount of memory required for the particular application and assign it to either RAM or ROM.
- The interconnection between memory and processor is then established from knowledge of the size of memory needed and the type of RAM and ROM chips available.
- The addressing of memory can be established by means of a table that specifies the memory address assigned to each chip.
- To demonstrate with a particular example, assume that a computer system needs 128 bytes of RAM and 512 bytes of ROM.

- The table, called a memory address map, is a pictorial representation of assigned address space for each chip in the system

Component	Hexa address	Address bus 10 9 8 7 6 5 4 3 2 1
RAM 1	0000 - 007F	0 0 0 x x x x x x x
RAM 2	0080 - 00FF	0 0 1 x x x x x x x
RAM 3	0100 - 017F	0 1 0 x x x x x x x
RAM 4	0180 - 01FF	0 1 1 x x x x x x x
ROM	0200 - 03FF	1 x x x x x x x x x

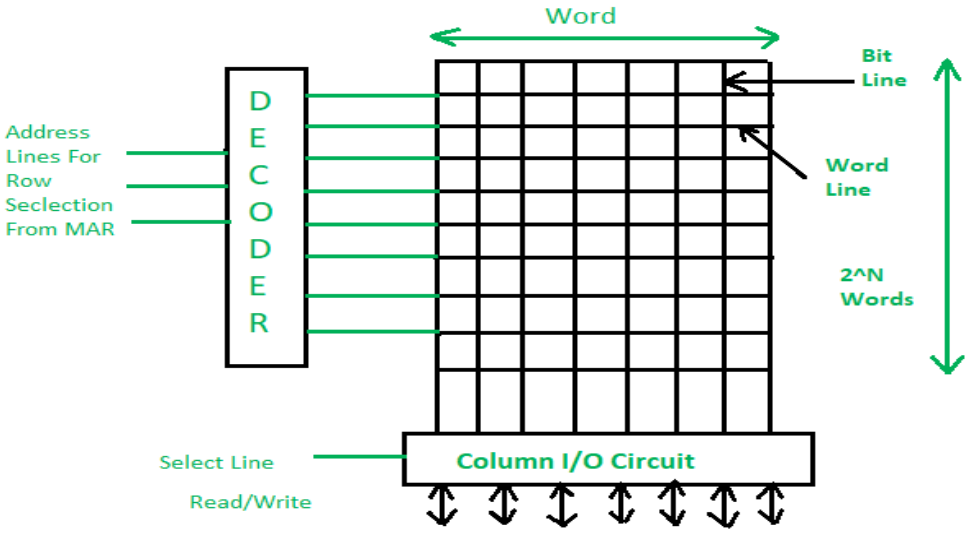
- The component column specifies whether a RAM or a ROM chip is used.
- The hexadecimal address column assigns a range of hexadecimal equivalent addresses for each chip.
- The address bus lines are listed in the third column. Although there are 16 lines in the address bus, the table shows only 10 lines because the other 6 are not used in this example and are assumed to be zero.
- The small x's under the address bus lines designate those lines that must be connected to the address inputs in each chip.
- The RAM chips have 128 bytes and need seven address lines. The ROM chip has 512 bytes and needs 9 address lines.
- The x's are always assigned to the low-order bus lines: lines 1 through 7 for the RAM and lines 1 through 9 for the ROM.
- It is now necessary to distinguish between four RAM chips by assigning to each a different address. For this particular example we choose bus lines 8 and 9 to represent four distinct binary combinations.
- The table clearly shows that the nine low-order bus lines constitute a memory space for ROM equal to 2^9 = 512 bytes.
- The distinction between a RAM and ROM address is done with another bus line. Here we choose line 10 for this purpose. When line 10 is 0, the CPU selects a RAM, and when this line is equal to 1, it selects the ROM

➢ 2D and 2.5D Memory organization

- The **internal structure** of Memory either RAM or ROM is made of memory cells that contain a memory bit. A group of 8 bits makes a byte. The memory is formed in a multidimensional array of rows and columns. In which each cell stores a bit and a complete row contains a word. A memory simply can be divided into this below form.
- 2^n = N, where, n is the no. of address lines and N is the total memory in bytes. There will be 2^n words.

➢ **2D Memory organization –**

- In 2D organization, memory is divided in the form of rows and columns (Matrix). Each row contains a word, now in this memory organization, there is a decoder.
- A decoder is a combinational circuit that contains n input lines and 2^n output lines.
- One of the output lines will select the row which address is contained in the MAR and the word which is represented by that row that will get selected and either read or write through the data lines.

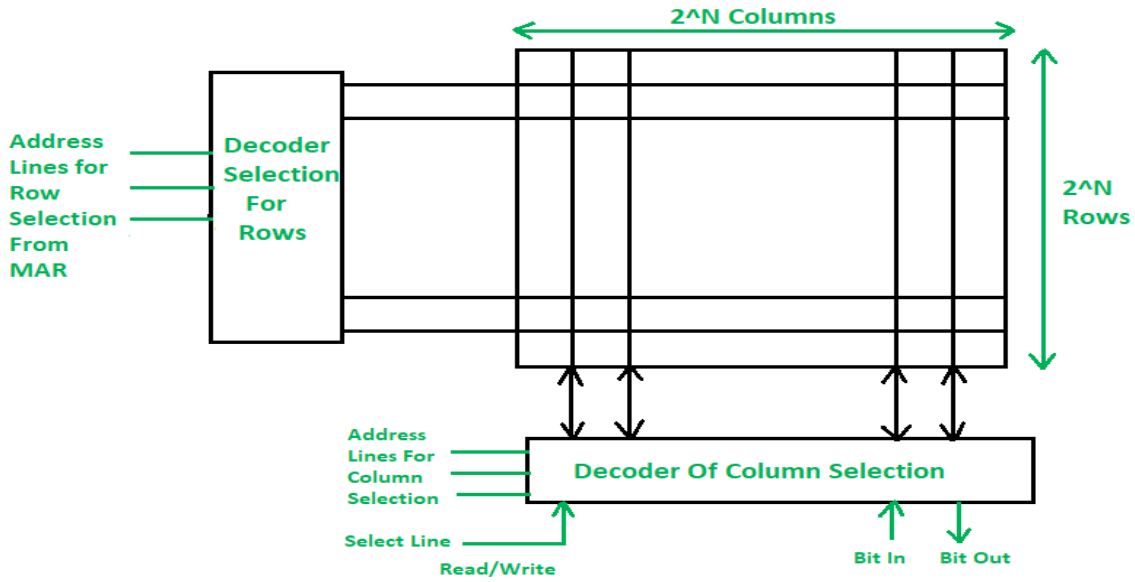


2D Memory Organization

➢ **2.5D Memory organization**

- In 2.5D Organization the scenario is the same but we have two different decoders one is column decoder and another is row decoder.
- Column decoder used to select the column and row decoder is used to select the row. The address from the MAR will go in decoders' input.

- Decoders will select the respective cell through the bit outline, the data from that location will be read or through the bit in line data will be written at that memory location.



2.5D Memory Organization

➢ Read and Write Operations –

- If the select line is in Read mode then the Word/bit which is represented by the MAR that will be coming out to the data lines and get read.
- If the select line is in write mode then the data from memory data register (MDR) will go to the respective cell which is addressed by the memory address register (MAR).
- With the help of the select line the data will get selected where the read and write operations will take place.

➢ Comparison between 2D & 2.5D Organizations –

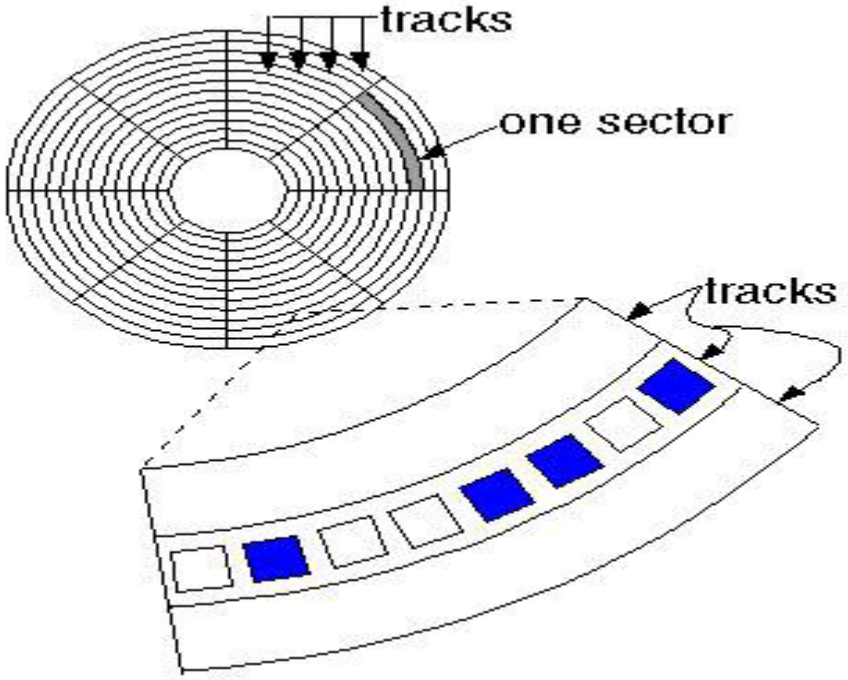
- In 2D organization hardware is fixed but in 2.5D hardware changes.
- 2D Organization requires more no. of Gates while 2.5D requires less no. of Gates.
- 2D is more complex in comparison to the 2.5D Organization.
- Error correction is not possible in the 2D organization but in 2.5D error correction is easy.
- 2D is more difficult to fabricate in comparison to the 2.5D organization.

- **Auxiliary Memory :**
- **Magnetic Tape:** Magnetic tapes are used for large computers like mainframe computers where large volume of data is stored for a longer time. In PC also you can use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetic materials that store data permanently. It can be 12.5 mm to 25 mm wide plastic film-type and 500 meter to 1200 meter long which is coated with magnetic material. The deck is connected to the central processor and information is fed into or read from the tape through the processor. It's similar to cassette tape recorder.
  - Magnetic tape is an information storage medium consisting of a magnet sable coating on a thin plastic strip. Nearly all recording tape is of this type, whether used for video with a video cassette recorder, audio storage (reel-to-reel tape, compact audio cassette, digital audio tape (DAT), digital linear tape (DLT) and other formats including 8-track cartridges) or general purpose digital data storage using a computer (specialized tape formats, as well as the abovementioned compact audio cassette, used with home computers of the 1980s, and DAT, used for backup in workstation installations of the 1990s).
  - Magneto-optical and optical tape storage products have been developed using many of the same concepts as magnetic storage, but have achieved little commercial success.
  - **Magnetic Disk:** You might have seen the gramophone record, which is circular like a disk and coated with magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the computer drive. Data is stored on both the surface of the disk. Magnetic disks are most popular for direct access storage device. Each disk consists of a number of invisible concentric circles called tracks. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents one bit and its absence represents zero bit. The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded. For Example-Floppy Disk.
  - The primary computer storage device. Like tape, it is magnetically recorded and can be re-recorded over and over. Disks are rotating platters with a mechanical arm that moves a read/write head between the outer and inner edges of the platter's surface. It can take as long as one second to find a location on a floppy disk to as little as a couple of milliseconds on a fast hard disk. See hard disk for more details. The disk surface is divided into concentric tracks (circles within circles). The thinner the

tracks, the more storage. The data bits are recorded as tiny magnetic spots on the tracks. The smaller the spot, the more bits per inch and the greater the storage.

➢ **Sectors**

- Tracks are further divided into sectors, which hold a block of data that is read or written at one time; for example, READ SECTOR 782, WRITE SECTOR 5448. In order to update the disk, one or more sectors are read into the computer, changed and written back to disk. The operating system figures out how to fit data into these fixed spaces. Modern disks have more sectors in the outer tracks than the inner ones because the outer radius of the platter is greater than the inner radius.



Block diagram of Magnetic Disk

➢ **Optical Disk:** With every new application and software there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the development of optical disk storage medium. Optical disks can be divided into the following categories:

1. Compact Disk/ Read Only Memory (CD-ROM
2. Write Once, Read Many (WORM)
3. Erasable Optical Disk

➢ Associative Memory: Content Addressable Memory (CAM)

- 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 the 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 is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.

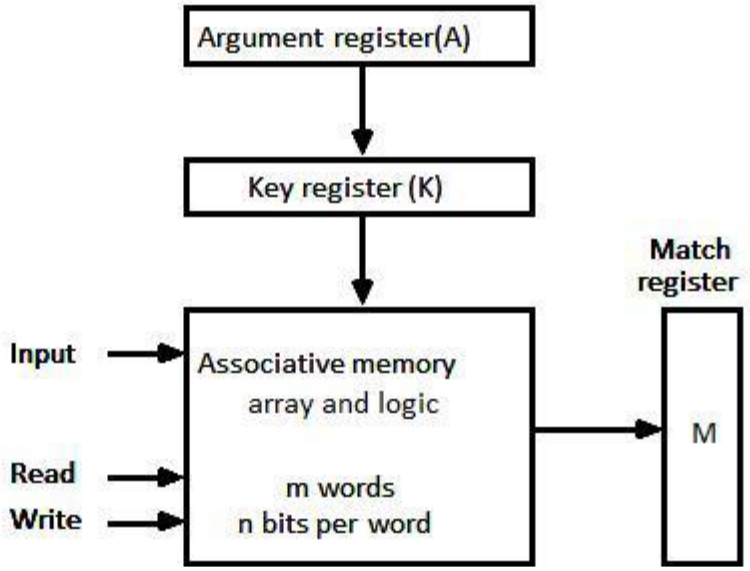


Diagram of an Associative Memory

- It consists of a memory array and logic form 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.

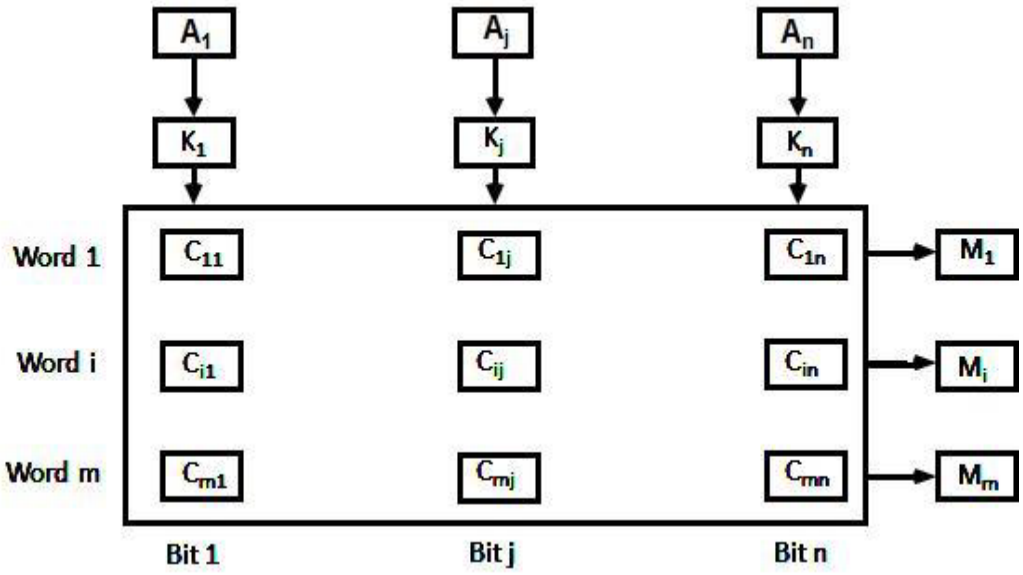
➢ Hardware Organization

- 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 1's.

- Otherwise, only those bits in the argument that have 1st in their corresponding position of the key register are compared.
- Thus the key provides a mask or identifying piece of information which specifies how the reference to memory is made.
- 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 are compared with memory words because K has 1's in these position.

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

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

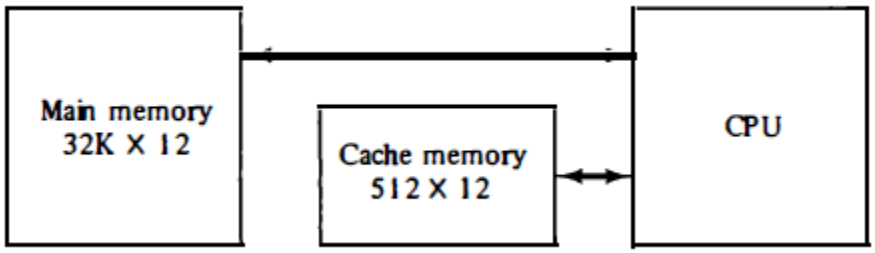


Associative memory of *m* word, n cells per word

- The relation between the memory array and external registers in an associative memory is shown in the above figure.
- The cells in the array are marked by the letter C with two subscripts.
- The first subscript gives the word number and the second specifies the bit position in the word. Thus cell Cij is the cell for bit j in words i.
- A bit Aj in the argument register is compared with all the bits in column j of the array provided that Kj =1.
- This is done for all columns j = 1, 2... n.
- If a match occurs between all the unmasked bits of the argument and the bits in word i, the corresponding bit Mi in the match register is set to 1.
- If one or more unmasked bits of the argument and the word do not match, Mi is cleared to 0.

➢ Cache Memory

- If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, thus reducing the total execution time of the program. Such a fast small memory is referred to as a cache memory.



- It is placed between the CPU and main memory. The cache memory access time is less than the access time of main memory by a factor of 5 to 10. The cache is the fastest component in the memory hierarchy and increasing the speed of CPU components.
- The fundamental idea of cache organization is to keeping the most frequently accessed instructions and data in the fast cache memory, 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 is found in the cache, it is read from the fast memory.
- 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 containing the one just accessed is then transferred from main memory to cache memory.

➢ ***Performance:*** The performance of cache memory is frequently measured in terms of a quantity called hit ratio.

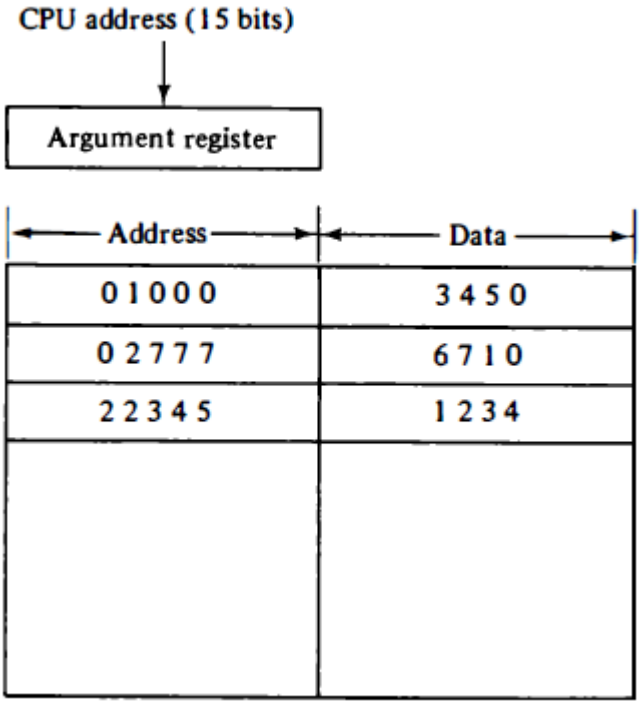
- When the CPU refers to memory and finds the word in cache, it is said to produce a hit. If the word is not found in cache, it is in main memory and it counts as a miss. The ratio of the number of hits divided by the total CPU references to memory (hits plus misses) is the hit ratio.

➢ ***Mapping:*** The transformation of data from main memory to cache memory is referred to as a mapping process. Three types of mapping procedures are of practical interest when considering the organization of cache memory:

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

1. Associative mapping:

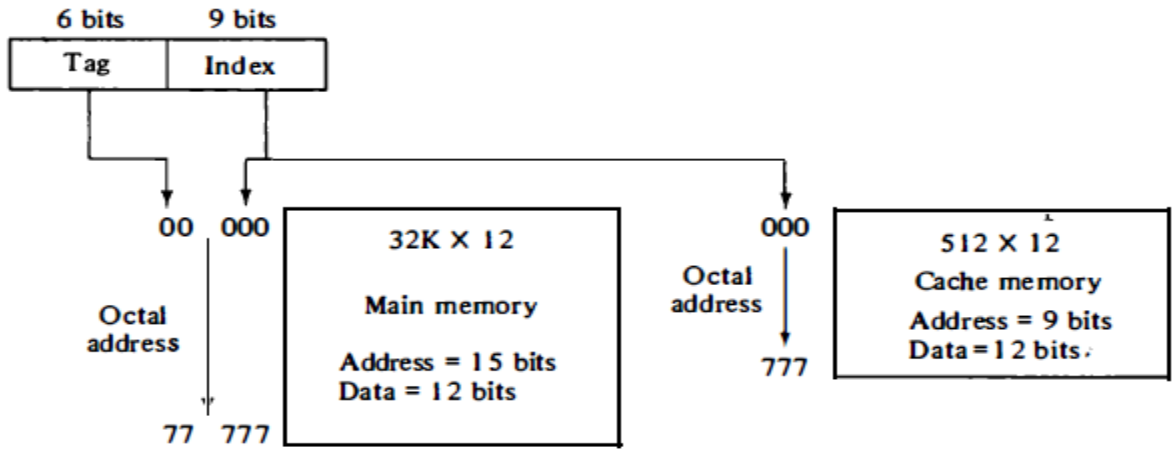
- The fastest and most flexible cache organization uses an associative memory.
- The associative memory stores both the address and content (data) of the memory word.
- This permits any location in cache to store any word from main memory.
- The diagram shows three words presently stored in the cache. The address value of 15 bits is shown as a five-digit octal number and its corresponding 12 -bit word is shown as a four-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-bit data is read and sent to the CPU.
- If no match occurs, the main memory is accessed for the word. The address-data pair is then transferred to the associative cache memory.
- If the cache is full, an address--data pair must be displaced to make room for a pair that is needed and not presently in the cache.
- This constitutes a first-in first-one (FIFO) replacement policy.

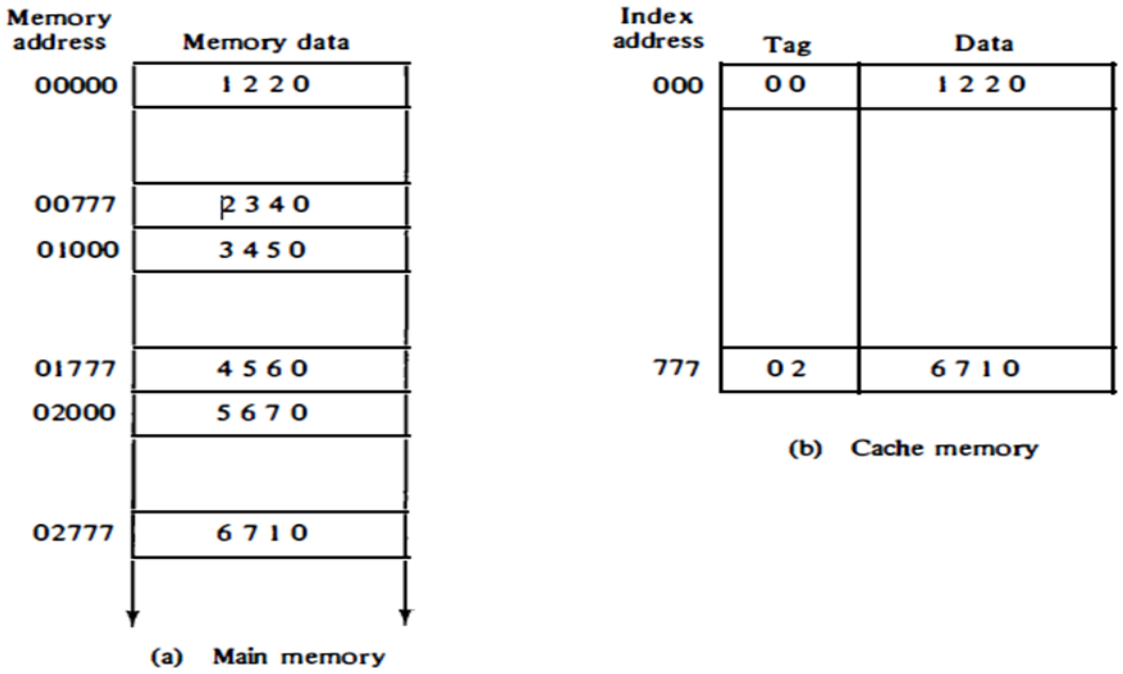
2. Direct Mapping:

- The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the index field and the remaining six bits form the tag field.



Addressing relationships between Main and Cache memories

- The figure shows that main memory needs an address that includes both the tag and the index bits. The number of bits in the index field is equal to the number of address bits required to access the cache memory.
- The internal organization of the words in the cache memory is shown below



Direct mapping cache organization

- In the general case, there are 2^k words in cache memory and 2^n words in main memory.
- The n-bit 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 the k-bit index to access the cache.
- 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.
- 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 main memory.
- It is then stored in the cache together with the new tag, replacing the previous value.

Example: The word at address zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now 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 5670 is transferred to the CPU.
- The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.
- The **disadvantage** of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.

3. Set-associative mapping

- It was mentioned previously that the disadvantage of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.
- Third type of cache organization, called set-associative mapping, is an improvement over 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.

Index	Tag	Data	Tag	Data
000	0 1	3 4 5 0	0 2	5 6 7 0
777	0 2	6 7 1 0	0 0	2 3 4 0

- 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 is 2(6 + 12) = 36 bits.
- The words stored at addresses 01000 and 02000 of main memory are stored in cache memory at index address 000. Similarly, the words at addresses 02777 and 00777 are stored in cache at index address 777.

- When the CPU generates a memory request, the index value of the address is used to access the cache. The tag field of the CPU address is then compared with both tags in the cache to determine if a match occurs.

➢ **Write-through and Write-back cache write method.**

Write Through

- The simplest and most commonly used procedure is to update main memory with every memory write operation.
- The cache memory being updated in parallel if it contains the word at the specified address. This is called the *write-through* method.
- This method has the advantage that main memory always contains the same data as the cache.
- This characteristic is important in systems with direct memory access transfers. It ensures that the data residing in main memory are valid at all times so that an I/O device communicating through DMA would receive the most recent updated data.

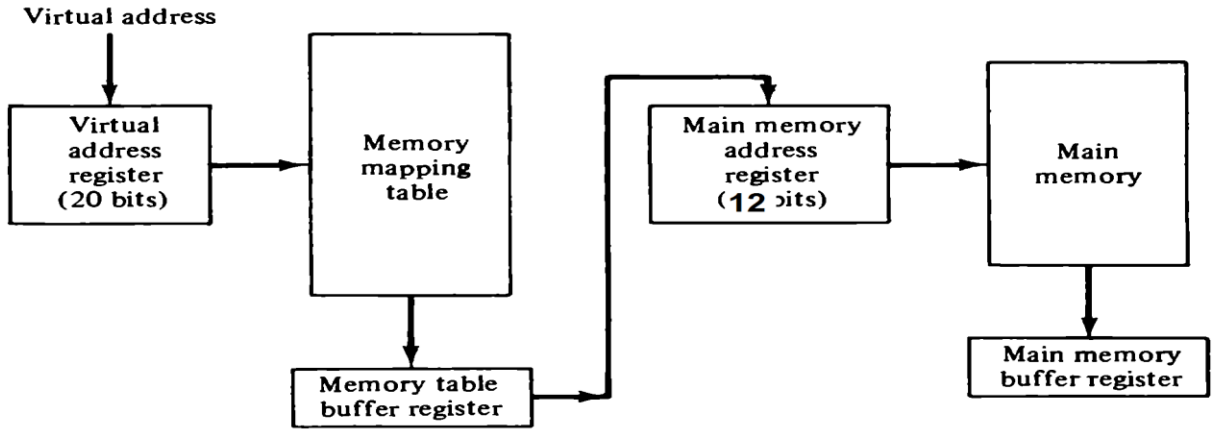
➢ ***Write-Back (Copy-Back)***

- The second procedure is called the write-back method.
- In this method only the cache location is updated during a write operation.
- The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory.
- The reason for the write-back method is that during the time a word resides in the cache, it may be updated several times.
- However, as long as the word remains in the cache, it does not matter whether the copy in main memory is out of date, since requests from the word are filled from the cache.
- It is only when the word is displaced from the cache that an accurate copy need be rewritten into main memory.

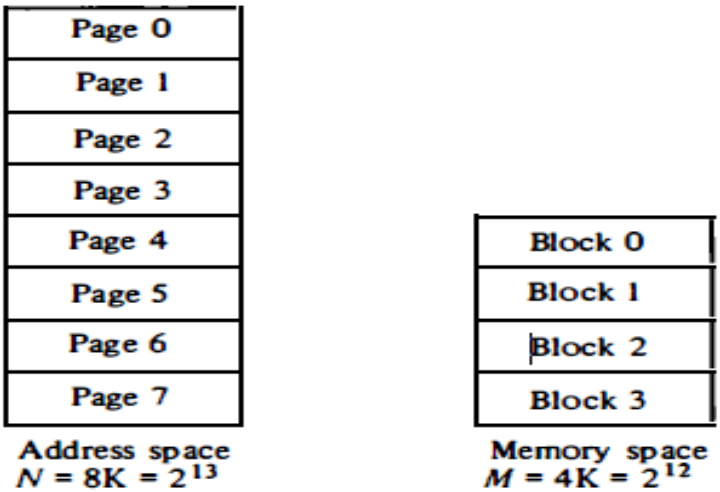
➢ ***Virtual Memory***

- 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.

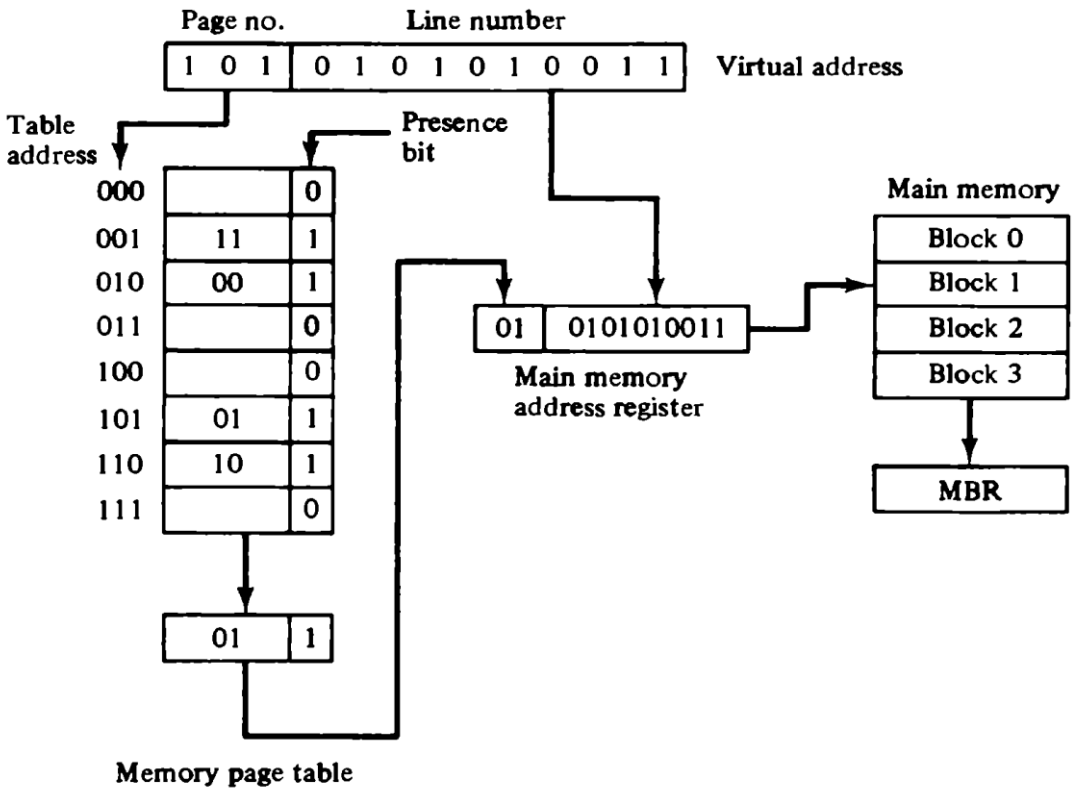
- **Address space**
  - An address used by a programmer will be called a virtual address, and the set of such addresses is known as address space.
- **Memory space**
  - An address in main memory is called a location or physical address. The set of such locations is called the memory space.
  - As an illustration, consider a computer with a main-memory capacity of 32K words (K = 1024). Fifteen bits are needed to specify a physical address in memory since 32K = 2^{15}.
  - Suppose that the computer has available auxiliary memory for storing 2^{20} = 1024K words.
  - Denoting the address space by N and the memory space by M, we then have for this example N = 1024K and M = 32K.
  - 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.
  - The address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits.
  - A table is then needed, as shown below to map a virtual address of 20 bits to a physical address of 15 bits. The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.
  - 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 Mapping Using Pages: The physical memory is broken down into groups of equal size called BLOCKS (or page frame). The groups of address space of same size are called PAGES.
- A page refers to the organization of address space, while a block refers to the organization of memory space.
- For Example: Consider a computer with an address space of 8K and a memory space of 4K. If we split each into groups of 1K words we obtain eight pages and four blocks as shown in Fig. At any given time, up to four pages of address space may reside in main memory in any one of the four blocks.



Address space and memory space split into groups of lK words



- The address in the page table denotes the page number and the content of the word gives the block number where that page is stored in main memory.
- The table shows that pages 1, 2, 5, and 6 are now available in main memory in blocks 3, 0, 1, and 2, respectively.
- A presence bit in each location indicates whether the page has been transferred from auxiliary memory into main memory. A 0 in the presence bit indicates that this page is not available in main memory.
- A call to the operating system is then generated to fetch the required page from auxiliary memory and place it into main memory before resuming computation.

- **Page Replacement: It must decide**

(1) Which page in main memory ought to be removed to make room for a new page.
(2) When a new page is to be transferred from auxiliary memory to main memory.
(3) Where the page is to be placed in main memory.

- **Page Fault:** When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate their position. The program is executed from main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called *page fault*. When page fault occurs, the execution of the present program is suspended until the required page is brought into main memory.

- When a page fault occurs in a virtual memory system, it signifies that the page referenced by the CPU is not in main memory. A new page is then transferred from auxiliary memory to main memory. If main memory is full, it would be necessary to remove a page from a memory block to make room for the new page.
- The policy for choosing pages to remove is determined from the replacement algorithm.
- **FIFO (First-In-First-Out):** Two of the most common replacement algorithms used are the first-in, first-out (FIFO):
- The FIFO algorithm selects for replacement the page that has been in memory the longest time.
- **LRU (least recently used):** The LRU policy is more difficult to implement but has been more attractive on the assumption that the least recently used page is a better candidate for removal than the least recently loaded page as in FIFO.