Unit – IV

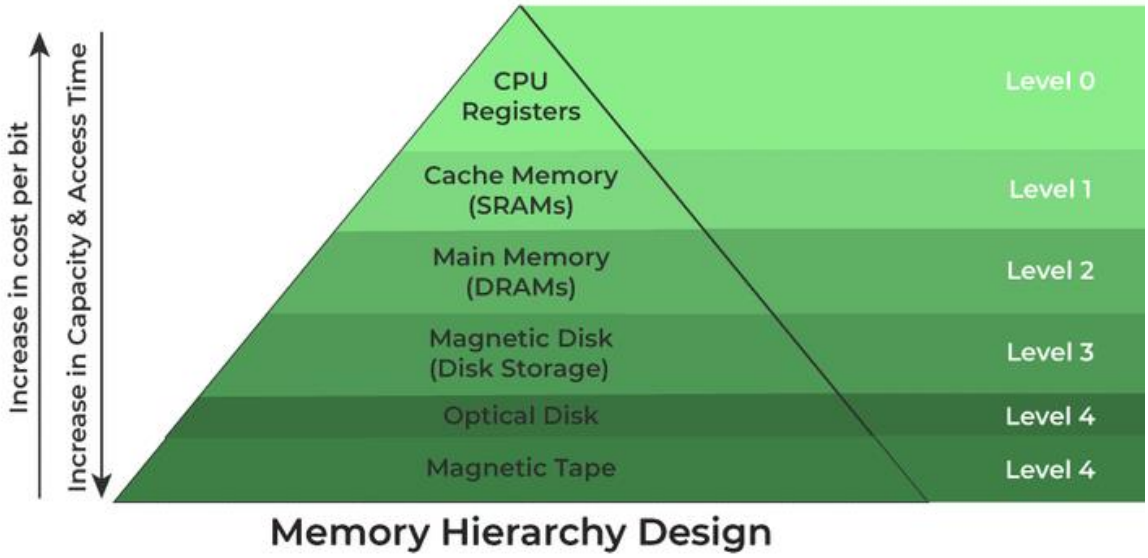
Why Memory Hierarchy is Required in the System?

Memory Hierarchy is one of the most required things in Computer Memory as it helps in optimizing the memory available in the computer. There are multiple levels present in the memory, each one having a different size, different cost, etc. Some types of memory like cache, and main memory are faster as compared to other types of memory but they are having a little less size and are also costly whereas some memory has a little higher storage value, but they are a little slower. Accessing of data is not similar in all types of memory, some have faster access whereas some have slower access.

Types of Memory Hierarchy

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 an I/O Module.
- **Internal Memory or Primary Memory:** Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.



Memory Hierarchy Design

Memory Hierarchy Design

1. Registers

Registers are small, high-speed memory units located in the CPU. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 64 bits.

2. Cache Memory

Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory. Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

3. Main Memory

Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

Types of Main Memory

- **Static RAM:** Static RAM stores the binary information in flip flops and information remains valid until power is supplied. It has a faster access time and is used in implementing cache memory.
- **Dynamic RAM:** It stores the binary information as a charge on the capacitor. It requires refreshing circuitry to maintain the charge on the capacitors after a few milliseconds. It contains more memory cells per unit area as compared to SRAM.

4. Secondary Storage

Secondary storage, such as hard disk drives (HDD) and solid-state drives (SSD), is a non-volatile memory unit that has a larger storage capacity than main memory. It is used to store data and instructions that are not currently in use by the CPU. Secondary storage has the slowest access time and is typically the least expensive type of memory in the memory hierarchy.

5. Magnetic Disk

Magnetic Disks are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material. The Magnetic disks work at a high speed inside the computer and these are frequently used.

6. Magnetic Tape

Magnetic Tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backup of data. In the case of a magnetic tape, the access time for a computer is a little slower and therefore, it requires some amount of time for accessing the strip.

Characteristics of Memory Hierarchy

- **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 a Memory Hierarchy design, the speed gap increased between the CPU registers and Main Memory due to a 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.

1. Primary Memory

It is also known as the main memory of the computer system. It is used to store data and programs or instructions during computer operations. It uses semiconductor technology and hence is commonly called semiconductor memory. Primary memory is of two types:

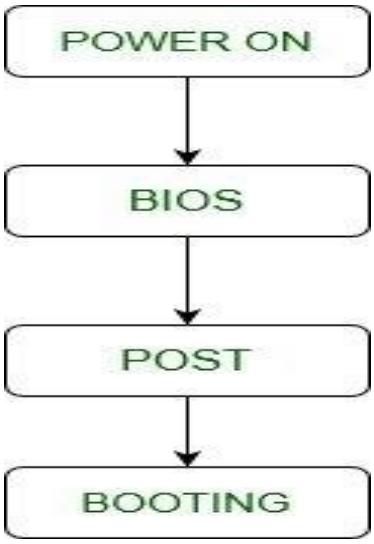
- **RAM (Random Access Memory):** It is a volatile memory. Volatile memory stores information based on the power supply. If the power supply fails/ interrupted/stopped, all the data and information on this memory will be lost. RAM is used for booting up or start the computer. It temporarily stores programs/data which has to be executed by the processor. RAM is of two types:
    - **S RAM (Static RAM):** S RAM uses transistors and the circuits of this memory are capable of retaining their state as long as the power is applied. This memory consists of the number of flip flops with each flip flop storing 1 bit. It has less access time and hence, it is faster.
    - **D RAM (Dynamic RAM):** D RAM uses capacitors and transistors and stores the data as a charge on the capacitors. They contain thousands of memory cells. It needs refreshing of charge on capacitor after a few milliseconds. This memory is slower than S RAM.
- **ROM (Read Only Memory):** It is a non-volatile memory. Non-volatile memory stores information even when there is a power supply failed/ interrupted/stopped. ROM is used to store information that is used to operate the system. As its name refers to read-only memory, we can only read the programs and data that is stored on it. It contains some electronic fuses that can be programmed for a piece of specific information. The information stored in the ROM in binary format. It is also known as permanent memory. ROM is of four types:
    - **MROM (Masked ROM):** Hard-wired devices with a pre-programmed collection of data or instructions were the first ROMs. Masked ROMs are a type of low-cost ROM that works in this way.
    - **PROM (Programmable Read Only Memory):** This read-only memory is modifiable once by the user. The user purchases a blank PROM and uses a PROM program to put the required contents into the PROM. Its content can’t be erased once written.
    - **EPROM (Erasable Programmable Read Only Memory):** EPROM is an extension to PROM where you can erase the content of ROM by exposing it to Ultraviolet rays for nearly 40 minutes.
    - **EEPROM (Electrically Erasable Programmable Read Only Memory):** Here the written contents can be erased electrically. You can delete and reprogrammed EEPROM up to 10,000 times. Erasing and programming take very little time, i.e., nearly 4 -10 ms(milliseconds). Any area in an EEPROM can be wiped and programmed selectively.

Introduction of Basic Input Output System (BIOS)

When a computer system is turned on it requires a series of initialization and test before the user can work on it. This process is called booting.

Basic Input Output System (BIOS):
It provides a set of instructions and is responsible for booting the computer. The BIOS performs all the test needed at startup time. These tests are collectively known as Power On Self-Test (POST). The computer contains hardware like keyboard, monitor, disk drives, etc. there functioning requires interfacing with the operating system. The BIOS provides drivers for basic hardware like keyboard and monitor, mouse, etc. The operating system provides hardware for printer, modem, etc. Drivers for some hardware may not be available in the operating system hence these have to be explicitly installed by the user.

Power On Self-Test (POST):
POST consists of a series of diagnostic test that runs automatically when a user turns on the computer. The actual test may differ depending on the configuration of the BIOS. However, the usual test includes testing of the RAM, keyboard and the disk drives. If these tests are successful the computer boots itself and loads the operating system but if these tests are unsuccessful, the computer reports the errors through a series of beeps to draw the operator's attention and finally an error message is displayed on the monitor.



Location Of BIOS:
BIOS is typically placed in a chip known as Read Only Memory(ROM) that comes with the computer. This ensures that the BIOS will always be available even if the hard disk is formatted or replaced. However, in many cases the content of the ROM is transferred to the RAM when the system is started. This is because the RAM allows quicker access as compared to the ROM. Copying of the content of the ROM to the RAM is known as shadowing.

Auxiliary Memory

An Auxiliary memory is known as the lowest-cost, highest-capacity and slowest-access storage in a computer system. It is where programs and data are kept for long-term storage or when

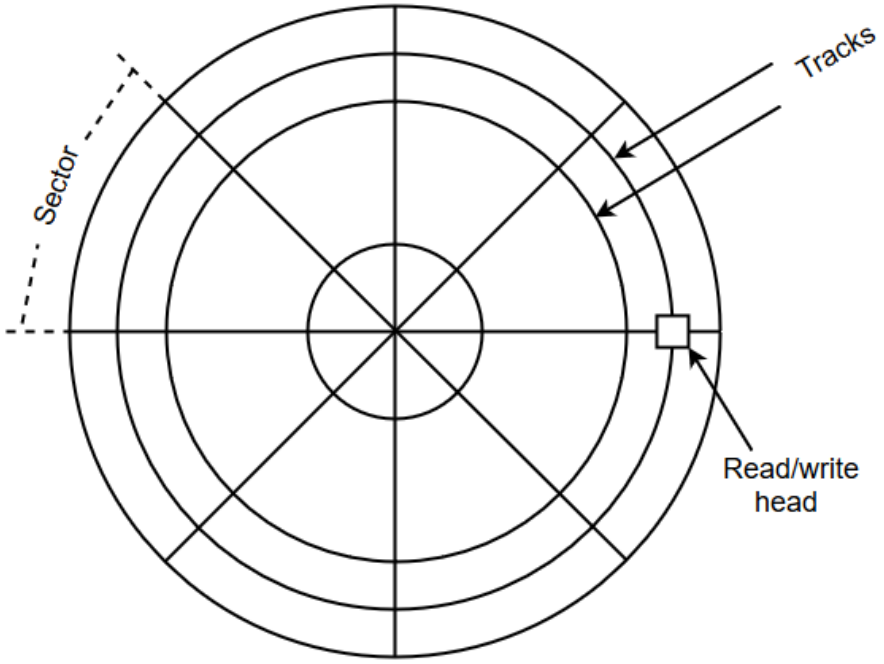
not in immediate use. The most common examples of auxiliary memories are magnetic tapes and magnetic disks.

Magnetic Disks

A magnetic disk is a type of memory constructed using a circular plate of metal or plastic coated with magnetized materials. Usually, both sides of the disks are used to carry out read/write operations. However, several disks may be stacked on one spindle with read/write head available on each surface.

The following image shows the structural representation for a magnetic disk.

Magnetic disks



- The memory bits are stored in the magnetized surface in spots along the concentric circles called tracks.
- The concentric circles (tracks) are commonly divided into sections called sectors.

Magnetic Tape

Magnetic tape is a storage medium that allows data archiving, collection, and backup for different kinds of data. The magnetic tape is constructed using a plastic strip coated with a magnetic recording medium.

The bits are recorded as magnetic spots on the tape along several tracks. Usually, seven or nine bits are recorded simultaneously to form a character together with a parity bit.

Magnetic tape units can be halted, started to move forward or in reverse, or can be rewound. However, they cannot be started or stopped fast enough between individual characters. For this reason, information is recorded in blocks referred to as records.

Difference between **Magnetic Tape** and **Magnetic Disk:**

S. No.	Magnetic Tape	Magnetic Disk
1.	Plastic ribbon serves as the primary component of magnetic tape memory.	The metal or plastic circular disk having coating of magnetic oxide serves as the key component of Magnetic disk memory.
2.	The cost of magnetic tape is less.	The cost of magnetic disk is high.
3.	Reliability of magnetic tape is less.	Reliability of magnetic disk is more.
4.	Access time for magnetic tape is more.	Access time for magnetic disk is less.
5.	Data transfer rate for magnetic tape is comparatively less.	Data transfer rate for magnetic disk is more.
6.	Magnetic tape is used for backups.	Magnetic disk is used as a secondary storage.

S. No.	Magnetic Tape	Magnetic Disk
7.	In magnetic tape data accessing rate is slow.	In magnetic disk data accessing rate is high or fast.
8.	In magnetic tape data can’t be updated after fed-up of data.	In magnetic disk data can be updated.
9.	Magnetic tape is more portable.	Magnetic disk is less portable.
10.	Magnetic tape contains reels of tape which is in form of strip of plastic.	Magnetic disk contains round platters which is made up of plastic or metal.
11.	1 track is kept for parity check in magnetic tapes so it is not used for storing data.	In magnetic disk, the topmost surface of top plate and bottommost surface of last plate in a platter are not used for storing data as scratching issue can be there for both surfaces.
12.	In magnetic tape for data recording, magnetic material is coated on only one side of the tape.	While in magnetic disk for data recording, magnetic material is coated on both side of the platters.
13.	Tape drives are used for reading and writing of data from magnetic tapes.	Disk drives are used for reading and writing of data from hard disks.
14.	The storage components of magnetic tapes, are in contact with external devices in tape drives.	Magnetic discs are not touched by any external devices
15.	Hold less data as compared to magnetic disk.	Capacity to hold more data per unit volume. But these need to be kept in a vacuum in order to reduce air friction.
16.	Storing of data takes place in the form of records in magnetic	Storing of data takes place in the form of files, folders or directories on the disk.

S. No.	Magnetic Tape	Magnetic Disk
	tapes that are further organized in blocks.	

Associative Memory

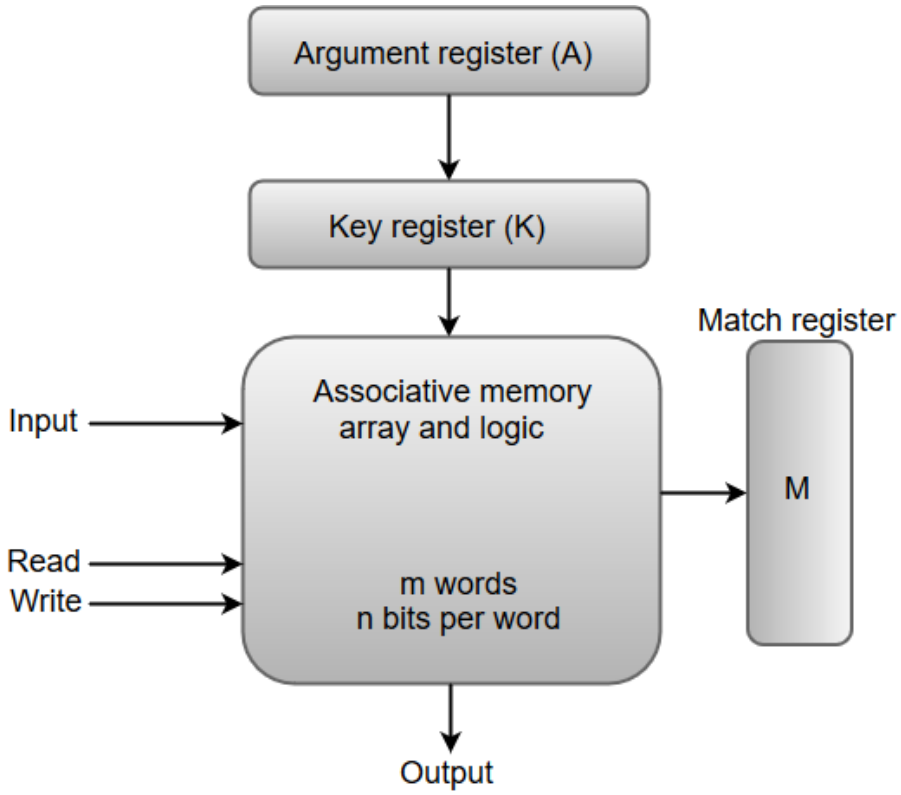
An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location.

Associative memory is often referred to as **Content Addressable Memory (CAM)**.

When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.

On the other hand, when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading.

The following diagram shows the block representation of an Associative memory.



From the block diagram, we can say that an associative memory consists of a memory array and logic for 'm' words with 'n' bits per word.

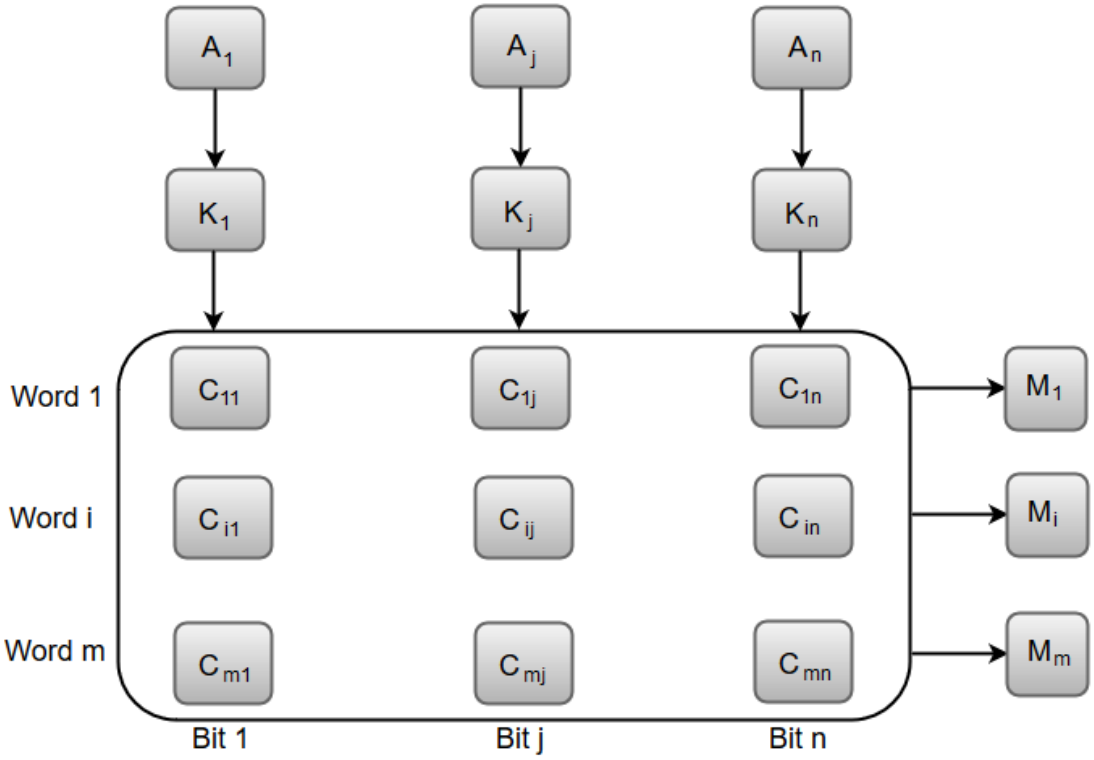
The functional registers like the argument register **A** and key register **K** each have **n** bits, one for each bit of a word. The match register **M** consists of **m** bits, one for each memory word.

The words which are kept in the memory are compared in parallel with the content of the argument register.

The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1's, then the entire argument is compared with each memory word. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

The following diagram can represent the relation between the memory array and the external registers in an associative memory.

Associative memory of m word, n cells per word:



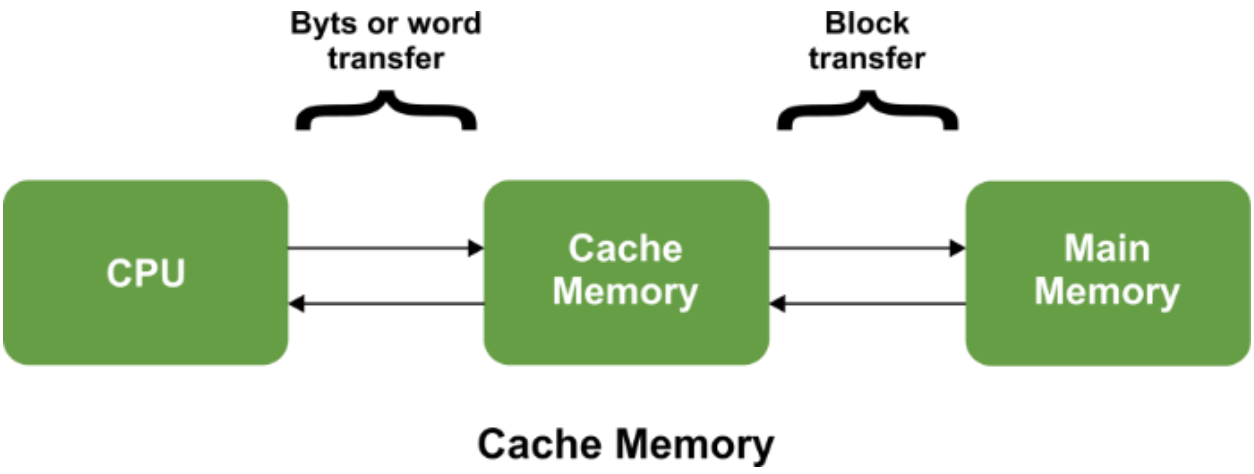
The cells present inside the memory array are marked by the letter C with two subscripts. The first subscript gives the word number and the second specifies the bit position in the word. For instance, the cell C_{ij} is the cell for bit **j** in word **i**.

A bit A_j in the argument register is compared with all the bits in column **j** of the array provided that $K_j = 1$. This process is done for all columns **j** = 1, 2, 3....., n.

If a match occurs between all the unmasked bits of the argument and the bits in word **i**, the corresponding bit M_i in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M_i is cleared to 0.

Cache Memory

The data or contents of the main memory that are used frequently by CPU are stored in the cache memory so that the processor can easily access that data in a shorter time. Whenever the CPU needs to access memory, it first checks the cache memory. If the data is not found in cache memory, then the CPU moves into the main memory.
Cache memory is placed between the CPU and the main memory. The block diagram for a cache memory can be represented as:



Cache Memory

The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

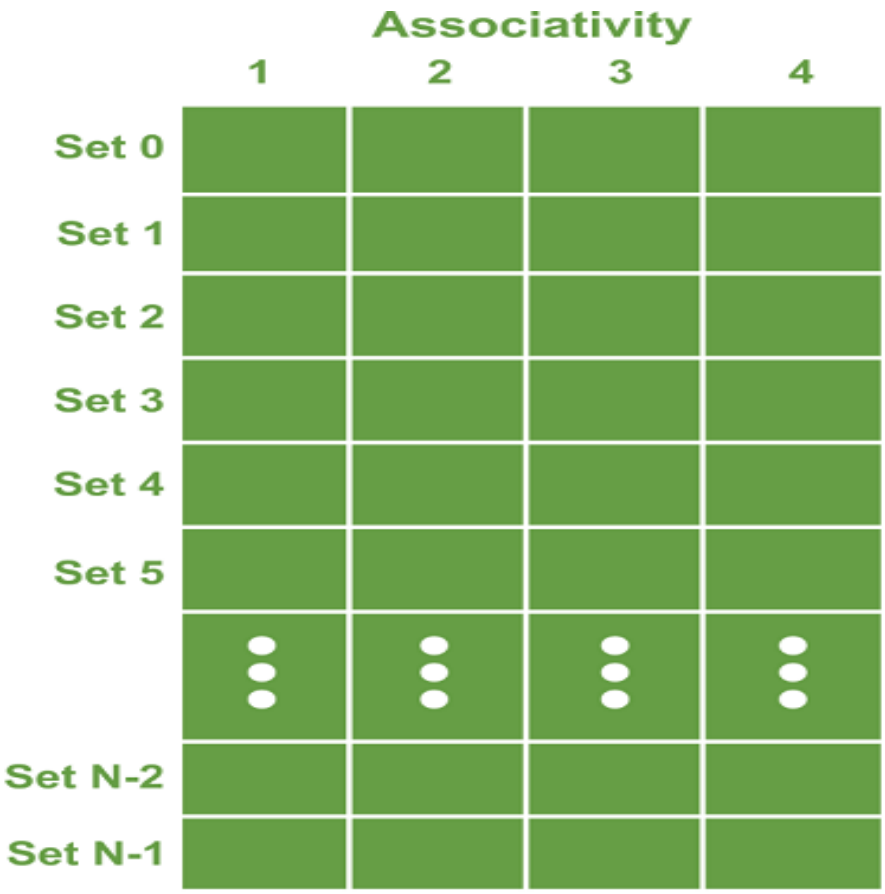
Cache memory is organized as distinct set of blocks where each set contains a small fixed number of blocks.

The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

Cache memory is organized as distinct set of blocks where each set contains a small fixed number of blocks.

The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

Cache memory is organized as distinct set of blocks where each set contains a small fixed number of blocks.



Logical organisation of a 4-way set associate cache

As shown in the above sets are represented by the rows. The example contains N sets and each set contains four blocks. Whenever an access is made to cache, the cache controller does not search the entire cache in order to look for a match. Rather, the controller maps the address to a particular set of the cache and therefore searches only the set for a match.

If a required block is not found in that set, the block is not present in the cache and cache controller does not search it further. This kind of cache organization is called set associative because the cache is divided into distinct sets of blocks. As each set contains four blocks the cache is said to be four way set associative.

The basic operation of a cache memory 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 one just accessed is then transferred from main memory to cache memory. The block size may vary from one word (the one just accessed) to about 16 words adjacent to the one just accessed.

- The performance of the 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 the 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.

Levels of memory:

Level 1
It is a type of memory in which data is stored and accepted that are immediately stored in CPU. Most commonly used register is accumulator, Program counter, address register etc.
Level 2
It is the fastest memory which has faster access time where data is temporarily stored for faster access.
Level 3
It is memory on which computer works currently. It is small in size and once power is off data no longer stays in this memory.
Level 4
It is external memory which is not as fast as main memory but data stays permanently in this memory.

What is virtual memory?

Virtual memory is a memory management technique where secondary memory can be used as if it were a part of the main memory. Virtual memory is a common technique used in a computer's operating system (OS).

Virtual memory uses both hardware and software to enable a computer to compensate for physical memory shortages, temporarily transferring data from random access memory (RAM) to disk storage. Mapping chunks of memory to disk files enables a computer to treat secondary memory as though it were main memory.

Today, most personal computers (PCs) come with at least 8 GB (gigabytes) of RAM. But, sometimes, this is not enough to run several programs at one time. This is where virtual memory comes in. Virtual memory frees up RAM by swapping data that has not been used recently over to a storage device, such as a hard drive or solid-state drive (SSD).

Virtual memory is important for improving system performance, multitasking and using large programs. However, users should not overly rely on virtual memory, since it is considerably slower than RAM. If the OS has to swap data between virtual memory and RAM too often, the computer will begin to slow down -- this is called thrashing.

Virtual memory was developed at a time when physical memory -- also referenced as RAM -- was expensive. Computers have a finite amount of RAM, so memory will eventually run out when multiple programs run at the same time. A system using virtual memory uses a section of the hard drive to emulate RAM. With virtual memory, a system can load larger or multiple programs running at the same time, enabling each one to operate as if it has more space, without having to purchase more RAM.

Virtual memory

Virtual memory uses both hardware and software to operate. When an application is in use, data from that program is stored in a physical address using RAM. A memory management unit (MMU) maps the address to RAM and automatically translates addresses. The MMU can, for example, map a logical address space to a corresponding physical address.

If, at any point, the RAM space is needed for something more urgent, data can be swapped out of RAM and into virtual memory. The computer's memory manager is in charge of keeping track of the shifts between physical and virtual memory. If that data is needed again, the computer's MMU will use a context switch to resume execution. While copying virtual memory into physical memory, the OS divides memory with a fixed number of addresses into either page files or swap files. Each page is stored on a disk, and when the page is needed, the OS copies it from the disk to main memory and translates the virtual addresses into real addresses.

However, the process of swapping virtual memory to physical is rather slow. This means using virtual memory generally causes a noticeable reduction in performance. Because of swapping, computers with more RAM are considered to have better performance.

Types of virtual memory

A computer's MMU manages virtual memory operations. In most computers, the MMU hardware is integrated into the central processing unit (CPU). The CPU also generates the virtual address space. In general, virtual memory is either paged or segmented.

Paging divides memory into sections or paging files. When a computer uses up its available RAM, pages not in use are transferred to the hard drive using a swap file. A swap file is a space set aside on the hard drive to be used as the virtual memory extension for the computer's RAM. When the swap file is needed, it is sent back to RAM using a process called page swapping. This system ensures the computer's OS and applications do not run out of real memory. The maximum size of the page file can be 1 ½ to four times the physical memory of the computer.

The virtual memory paging process uses page tables, which translate the virtual addresses that the OS and applications use into the physical addresses that the MMU uses. Entries in the page table indicate whether the page is in RAM. If the OS or a program does not find what it needs in RAM, then the MMU responds to the missing memory reference with a page fault exception to get the OS to move the page back to memory when it is needed. Once the page is in RAM, its virtual address appears in the page table.

Segmentation is also used to manage virtual memory. This approach divides virtual memory into segments of different lengths. Segments not in use in memory can be moved to virtual memory space on the hard drive. Segmented information or processes are tracked in a segment table, which shows if a segment is present in memory, whether it has been modified and what its physical address is. In addition, file systems in segmentation are only made up of segments that are mapped into a process's potential address space.

Segmentation and paging differ as a memory model in terms of how memory is divided; however, the processes can also be combined. In this case, memory gets divided into frames or pages. The segments take up multiple pages, and the virtual address includes both the segment number and the page number.

Other page replacement methods include first-in-first-out (FIFO), optimal algorithm and least recently used (LRU) page replacement. The FIFO method has memory select the replacement for a page that has been in the virtual address for the longest time. The optimal algorithm method selects page replacements based on which page is unlikely to be replaced after the longest amount of time; although difficult to implement, this leads

to less page faults. The LRU page replacement method replaces the page that has not been used for the longest time in the main memory.

How to manage virtual memory

Managing virtual memory within an OS can be straightforward, as there are default settings that determine the amount of hard drive space to allocate for virtual memory. Those settings will work for most applications and processes, but there may be times when it is necessary to manually reset the amount of hard drive space allocated to virtual memory -- for example, with applications that depend on fast response times or when the computer has multiple hard disk drives (HDDs).

When manually resetting virtual memory, the minimum and maximum amount of hard drive space to be used for virtual memory must be specified. Allocating too little HDD space for virtual memory can result in a computer running out of RAM. If a system continually needs more virtual memory space, it may be wise to consider adding RAM. Common OSes may generally recommend users not increase virtual memory beyond 1 ½ times the amount of RAM.

Managing virtual memory differs by OS. For this reason, IT professionals should understand the basics when it comes to managing physical memory, virtual memory and virtual addresses.

RAM cells in SSDs also have a limited lifespan. RAM cells have a limited number of writes, so using them for virtual memory often reduces the lifespan of the drive.

What are the benefits of using virtual memory?

The advantages to using virtual memory include:

- It can handle twice as many addresses as main memory.
- It enables more applications to be used at once.
- It frees applications from managing shared memory and saves users from having to add memory modules when RAM space runs out.
- It has increased speed when only a segment of a program is needed for execution.
- It has increased security because of memory isolation.
- It enables multiple larger applications to run simultaneously.
- Allocating memory is relatively inexpensive.
- It does not need external fragmentation.
- CPU use is effective for managing logical partition workloads.
- Data can be moved automatically.

- Pages in the original process can be shared during a fork system call operation that creates a copy of itself.

In addition to these benefits, in a virtualized computing environment, administrators can use virtual memory management techniques to allocate additional memory to a virtual machine (VM) that has run out of resources. Such virtualization management tactics can improve VM performance and management flexibility.

Virtual memory (virtual RAM) vs. physical memory (RAM)

When talking about the differences between virtual and physical memory, the biggest distinction commonly made is to speed. RAM is considerably faster than virtual memory. RAM, however, tends to be more expensive.

When a computer requires storage, RAM is the first used. Virtual memory, which is slower, is used only when the RAM is filled.

Users can actively add RAM to a computer by buying and installing more RAM chips. This is useful if they are experiencing slowdowns due to memory swaps happening too often. The amount of RAM depends on what is installed on a computer. Virtual memory, on the other hand, is limited by the size of the computer's hard drive. Virtual memory settings can often be controlled through the OS.

In addition, RAM uses swapping techniques, while virtual memory uses paging. While physical memory is limited to the size of the RAM chip, virtual memory is limited by the size of the hard disk. RAM also has direct access to the CPU, while virtual RAM does not.

What is Memory Mapping?

Memory mapping or ***mmap()*** is a function call in an Operating system like Unix. It is a low-level language, and it helps to directly map a file to the currently executing process's own memory address space. Which could optimize File I/O operations, inter-process communication, etc.

Types of Mapping

1. **File Mapping:** It will map the File to the process virtual memory

- **Read-Only Mappings:** The File is mapped into memory only for the purpose of reading. If we try to write it will end up in a segmentation fault.
- **Read-Write Mappings:** The file is mapped into memory for both reading and writing. ie, it helps with file manipulation.

2. Anonymous mapping: It is a memory-mapped region where there is no connection with any file/device. Or we can say it is used for dynamic allocation of memory within a program.

- **Private Anonymous Mapping:** This is an area of memory that does not have a file associated with it. It is commonly used for dynamic allocation. Custom memory allocators are examples of this type of mapping.

- **Shared Anonymous mapping:** Inter-process communication is a type of shared mapping in which multiple processes can have access to and change the data. An example of this is implementing Inter-process communication (IPC).

3. Device Mapping: In this type, Registers, hardware devices like graphic cards, and network adapters can be mapped into the process's address space.

In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when a new page comes in. Page replacement becomes necessary when a page fault occurs and there are no free page frames in memory. However, another page fault would arise if the replaced page is referenced again. Hence it is important to replace a page that is not likely to be referenced in the immediate future. y. If no page frame is free, the virtual memory manager performs a page replacement operation to replace one of the pages existing in memory with the page whose reference caused the page fault. It is performed as follows: The virtual memory manager uses a page replacement algorithm to select one of the pages currently in memory for replacement, accesses the page table entry of the selected page to mark it as "not present" in memory, and initiates a page-out operation for it if the modified bit of its page table entry indicates that it is a dirty page.

Page Fault

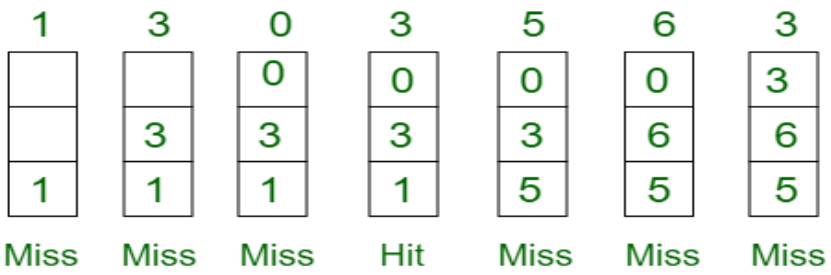
A page fault happens when a running program accesses a memory page that is mapped into the virtual address space but not loaded in physical memory. Since actual physical memory is much smaller than virtual memory, page faults happen. In case of a page fault, Operating System might have to replace one of the existing pages with the newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce the number of page faults.

Page Replacement Algorithms:

1. First In First Out (FIFO): This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

Example 1: Consider page reference string 1, 3, 0, 3, 5, 6, 3 with 3 page frames.Find the number of page faults.

Page reference 1, 3, 0, 3, 5, 6, 3



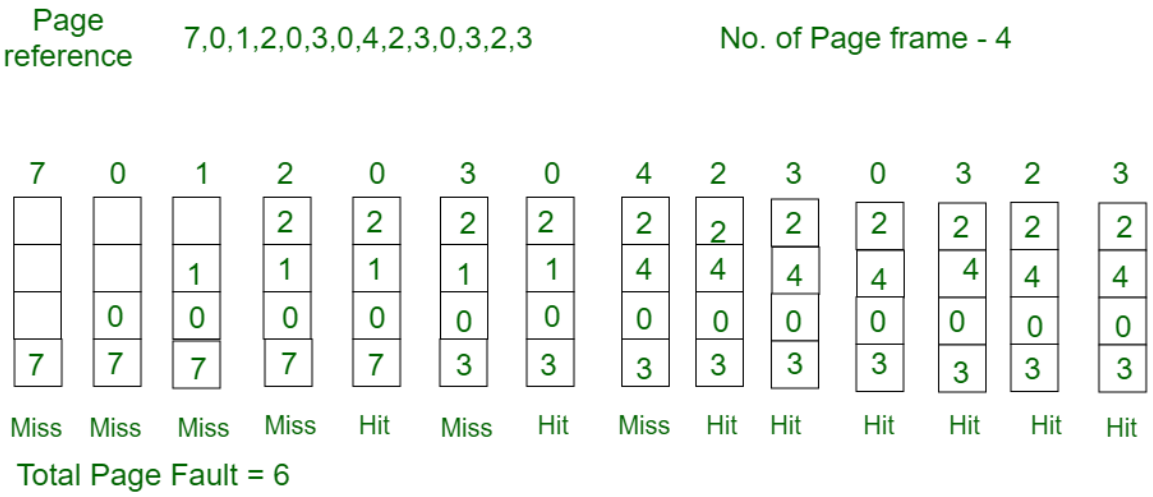
Total Page Fault = 6

Initially, all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> **3 Page Faults.**
when 3 comes, it is already in memory so —> **0 Page Faults.** Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —>**1 Page Fault.** 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —>**1 Page Fault.** Finally, when 3 come it is not available so it replaces 0 **1 page fault.**
Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First Out (FIFO) page replacement algorithm. For example, if we consider reference strings 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4, and 3 slots, we get 9 total page faults, but if we increase slots to 4, we get 10-page faults.
2. Optimal Page replacement: In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.
Example-2: Consider the page references 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3 with 4 page frame. Find number of page fault.

Page reference 7,0,1,2,0,3,0,4,2,3,0,3,2,3 No. of Page frame - 4

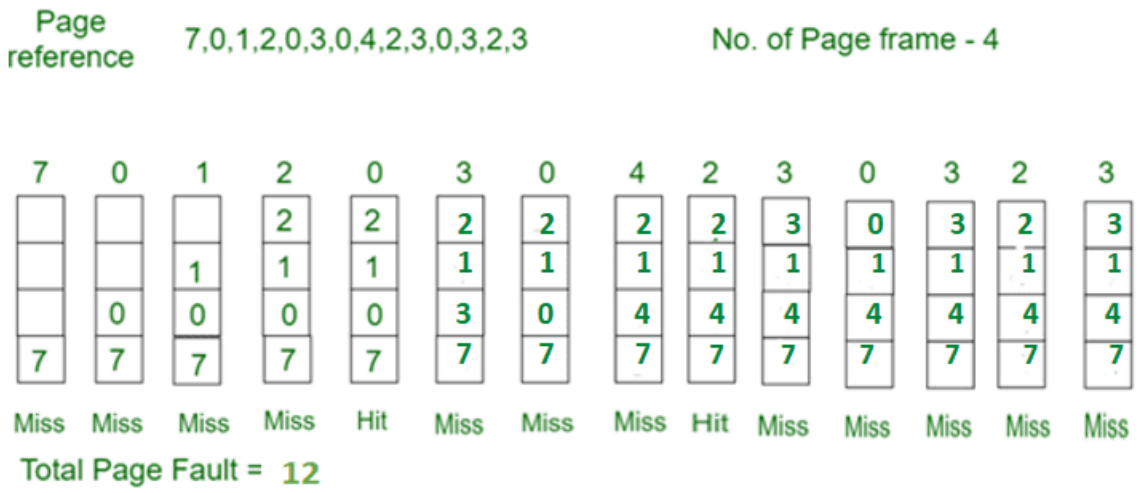
Total Page Fault = 6

Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**
0 is already there so —> **0 Page fault.** when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.—>**1 Page fault.** 0 is already there so —> **0 Page fault.** 4 will takes place of 1 —> **1 Page Fault.**
Now for the further page reference string —> **0 Page fault** because they are already available in the memory.
Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests. The use of Optimal Page replacement is to set up a benchmark so that other replacement algorithms can be analyzed against it.
3. Least Recently Used: In this algorithm, page will be replaced which is least recently used.
Example-3: Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3 with 4 page frames. Find number of page faults.



Here LRU has same number of page fault as optimal but it may differ according to question.

Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**
0 is already their so —> **0 Page fault.** when 3 came it will take the place of 7 because it is least recently used —>**1 Page fault**
0 is already in memory so —> **0 Page fault**.
4 will takes place of 1 —> **1 Page Fault**
Now for the further page reference string —> **0 Page fault** because they are already available in the memory.
4. Most Recently Used (MRU): In this algorithm, page will be replaced which has been used recently. Belady's anomaly can occur in this algorithm.



Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**
0 is already their so–> **0 page fault**
when 3 comes it will take place of 0 because it is most recently used —>**1 Page fault**
when 0 comes it will take place of 3 —>**1 Page fault**
when 4 comes it will take place of 0 —>**1 Page fault**
2 is already in memory so —> **0 Page fault**
when 3 comes it will take place of 2 —>**1 Page fault**
when 0 comes it will take place of 3 —>**1 Page fault**
when 3 comes it will take place of 0 —>**1 Page fault**
when 2 comes it will take place of 3 —>**1 Page fault**
when 3 comes it will take place of 2 —>**1 Page fault**

Flash Memory

Flash memory is secondary memory and so it is not volatile which means it persists the data even if there is not an electrical supply provided. This flash memory works on the principle of EEPROM. EEPROM stands for Electrical Erasable Programmable Read-Only Memory. ROM operation can only one time write and many times read and we can't erase it. But Flash Memory can be erased multiple times and update the data or program integrated into it. So it gives flexibility to the updation of the program but ROM has no such type of feature.

History of Flash Memory

Flash memory is developed by Dr. Masuoka Fujio and his team at Toshiba Corporation in the mid-1980. He was a Japanese engineer. When they discovered the flash memory semiconductor technology get the boom in that time frame. Many devices such as digital cameras, camcorders, MP3 players, and audio/video are developed using flash memory.

How Does Flash Memory Work?

These two are the main steps.

- **Writing the data into flash memory:** Flash memory is made up of small-small memory cells which are made up of floating-gate transistors. All memory cells

are organized in the sequential order called an array of memory cells. Data can be stored to the hardware in the form of 0 and 1 only. So to store the data into the flash memory we should give the electric supply to add the pattern of 0 and 1 into the flash memory and once this pattern is embedded on to the chip then corresponding data is also stored in the flash memory. If we want to erase the data from flash memory then we have to do it by supplying electrical impulse to make all bits of memory cell to 0.

- **Reading the data into flash memory:** By reading the stored 0 and 1 into the cell we can get our data stored in the flash memory. So to read the bit which can contain 0 or 1 we have to apply voltage to the gate of transistor, and whatever the current flow in the circuit is measured and by using that measurement we are good to go to identify the bit present in the cell.

Limitations of Flash Memory

- **Limited lifespan:** Writing onto the flash memory by electrical supply may damage the hardware so it has some limitations to the lifespan of flash memory.
- **Slower write speeds:** Frequency of writing speed so less than RAM and to write the data by using an electrical pulse every time may take more time than RAM.
- **Limited storage capacity:** Flash memory has a high storage density, but lesser than some other memory devices such as HDDs or tape drives.
- **Data corruption:** When we are writing to the flash memory by using electrical pulse and when power supply is cut off accidentally then loss of data will be there.

Benefits of Flash Memory

1. **Large storage capacity:** Flash memory has high memory density so it is able to store a high volume of data.
2. **High speed:** Some flash memory has parallel architecture of memory cells so it has faster speed to read and write operation.
3. **Persistent Data:** Without supply of electricity it persists the data like HDDs.
4. **Low power consumption:** Flash memory don’t have mechanical components like HDD’s so it consumes less amount of power than HDDs.

Applications of Flash Memory

1. **Used in SSDs:** Flash memory is used in SSDs to increase the speed of read/write of operations.
2. **Embedded systems:** Flash memory is used in embedded systems. Examples: digital cameras, camcorders, MP3 players etc.
3. **Smartphones and tablets:** Flash memory is used in smartphones and tablets.
4. **USB drives:** Flash memory is commonly used in USB drives.