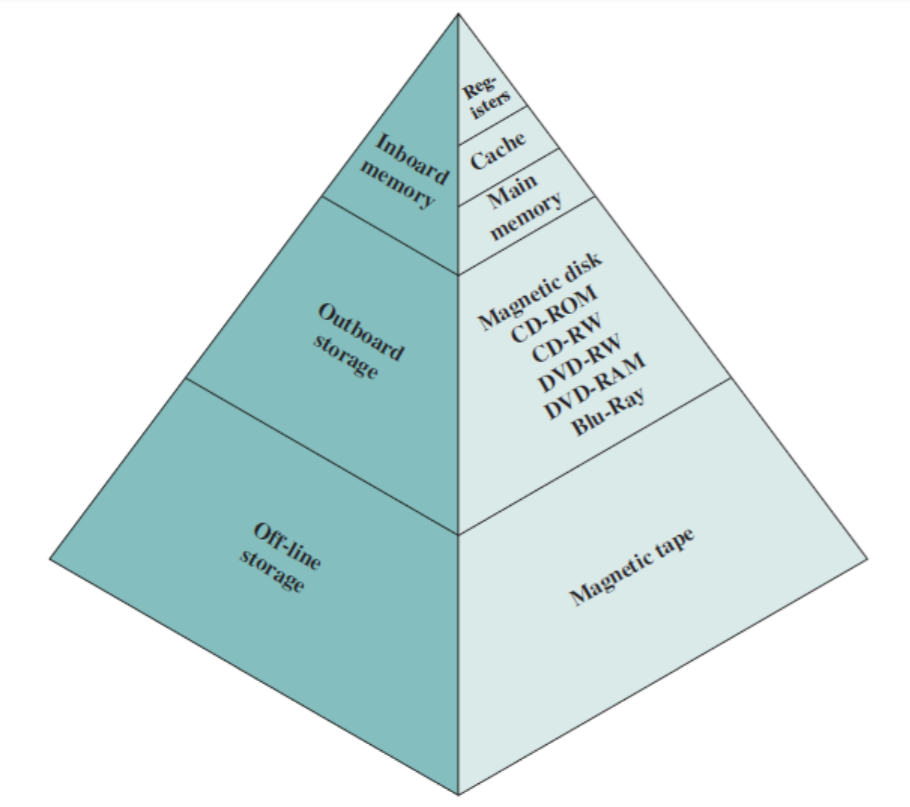
1. The memory hierarchy design in a computer system mainly includes different storage devices. Most of the computers were inbuilt with extra storage to run more powerfully beyond the main memory capacity



The memory hierarchy characteristics mainly include the following.

**Performance**

Previously, the designing of a computer system was done without memory hierarchy, and the speed gap among the main memory as well as the CPU registers enhances because of the huge disparity in access time, which will cause the lower performance of the system. So, the enhancement was mandatory. The enhancement of this was designed in the memory hierarchy model due to the system’s performance increase.

**Ability**

The ability of the memory hierarchy is the total amount of data the memory can store. Because whenever we shift from top to bottom inside the memory hierarchy, then the capacity will increase.

**Access Time**

The access time in the memory hierarchy is the interval of the time among the data availability as well as request to read or write. Because whenever we shift from top to bottom inside the memory hierarchy, then the access time will increase

**Cost per bit**

When we shift from bottom to top inside the memory hierarchy, then the cost for each bit will increase which means an internal Memory is expensive compared with external memory.

**Memory Hierarchy Design**

The memory hierarchy in computers mainly includes the following.

**Registers**

Usually, the register is a static RAM or SRAM in the processor of the computer which is used for holding the data word which is typically 64 or 128 bits. The program counter register is the most important as well as found in all the processors. Most of the processors use a status word register as well as an accumulator. A status word register is used for decision making, and the accumulator is used to store the data like mathematical operation. Usually, computers like complex instruction set computers have so many registers for accepting main memory, and RISC- reduced instruction set computers have more registers.

**Cache Memory**

Cache memory can also be found in the processor, however rarely it may be another IC (integrated circuit) which is separated into levels. The cache holds the chunk of data which are frequently used from main memory. When the processor has a single core then it will have two (or) more cache levels rarely. Present multi-core processors will be having three, 2-levels for each one core, and one level is shared.

**Main Memory**

The main memory in the computer is nothing but, the memory unit in the CPU that communicates directly. It is the main storage unit of the computer. This memory is fast as well as large memory used for storing the data throughout the operations of the computer. This memory is made up of RAM as well as ROM.

**Magnetic Disks**

The magnetic disks in the computer are circular plates fabricated of plastic otherwise metal by magnetized material. Frequently, two faces of the disk are utilized as well as many disks may be stacked on one spindle by read or write heads obtainable on every plane. All the disks in computer turn jointly at high speed. The tracks in the computer are nothing but bits which are stored within the magnetized plane in spots next to concentric circles. These are usually separated into sections which are named as sectors.

**Magnetic Tape**

This tape is a normal magnetic recording which is designed with a slender magnetizable covering on an extended, plastic film of the thin strip. This is mainly used to back up huge data. Whenever the computer requires to access a strip, first it will mount to access the data. Once the data is allowed, then it will be unmounted. The access time of memory will be slower within magnetic strip as well as it will take a few minutes for accessing a strip.

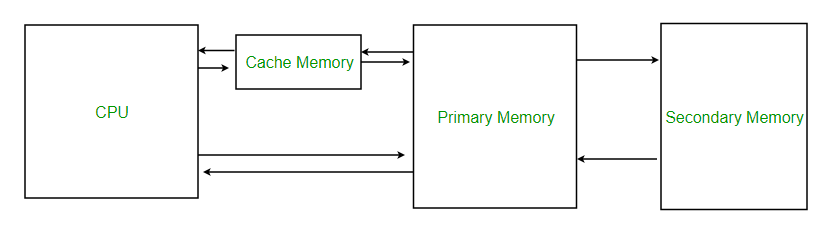
**Advantages of Memory Hierarchy**

The need for a memory hierarchy includes the following.

* Memory distributing is simple and economical
* Removes external destruction
* Data can be spread all over
* Permits demand paging & pre-paging
* Swapping will be more proficient

2. **Cache Memory** is a special very high-speed memory. It is used to speed up and synchronizing with high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU registers.

Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed. Cache memory is used to reduce the average time to access data from the Main memory.



**Levels of memory:**

* **Level 1 or Register –**  
  It is a type of memory in which data is stored and accepted that are immediately stored in CPU. Most used register is accumulator, Program counter, address register etc.
* **Level 2 or Cache memory –**  
  It is the fastest memory which has faster access time where data is temporarily stored for faster access.
* **Level 3 or Main Memory –**  
  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 or Secondary Memory –**  
  It is external memory which is not as fast as main memory but data stays permanently in this memory.

1. There are three different types of mapping used for the purpose of cache memory which are as follows: Direct mapping, Associative mapping, and Set-Associative mapping. These are explained below.

**Direct Mapping –**  
The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. or  
In Direct mapping, assigne each memory block to a specific line in the cache. If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory. Direct mapping`s performance is directly proportional to the Hit ratio.

i = j modulo m

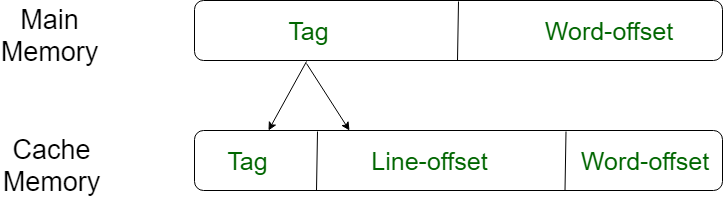
where

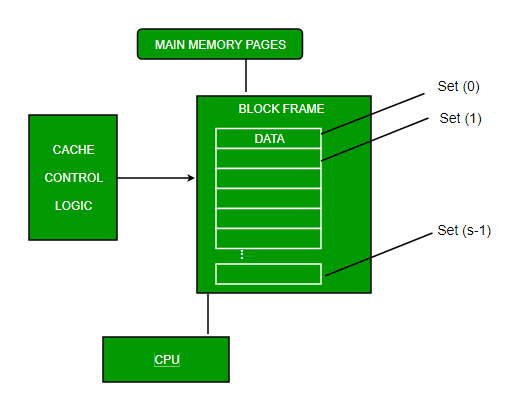
i=cache line number

j= main memory block number

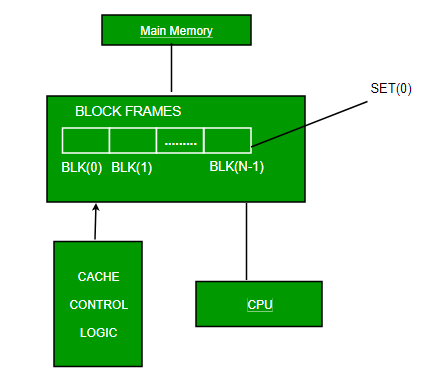
m=number of lines in the cache

For purposes of cache access, each main memory address can be viewed as consisting of three fields. The least significant w bits identify a unique word or byte within a block of main memory. In most contemporary machines, the address is at the byte level. The remaining s bits specify one of the 2s blocks of main memory. The cache logic interprets these s bits as a tag of s-r bits (most significant portion) and a line field of r bits. This latter field identifies one of the m=2r lines of the cache.





**Associative Mapping –**  
In this type of mapping, the associative memory is used to store content and addresses of the memory word. Any block can go into any line of the cache. This means that the word id bits are used to identify which word in the block is needed, but the tag becomes all the remaining bits. This enables the placement of any word at any place in the cache memory. It is the fastest and the most flexible mapping form.



**Set-associative Mapping –**  
This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed. Set associative addresses the problem of possible thrashing in the direct mapping method. It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a ***set***. Then a block in memory can map to any one of the lines of a specific set..Set-associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address. Set associative cache mapping combines the best of direct and associative cache mapping techniques.

In this case, the cache consists of a number of sets, each of which consists of a number of lines. The relationships are

m = v \* k

i= j mod v

where

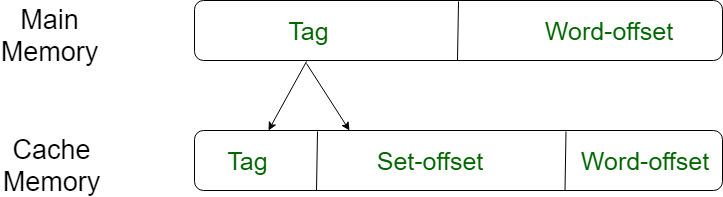
i=cache set number

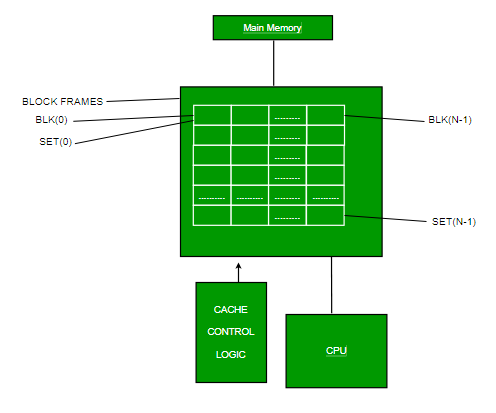
j=main memory block number

v=number of sets

m=number of lines in the cache number of sets

k=number of lines in each set





1. Unlike direct mapping, for associative and set associative mapping, replacement policy is needed. To achieve high speed Hardware implemented algorithm are used.
   * xLeast recently used algorithm (LRU): Most effective algorithm is Least Recently used (LRU). It Replaces that block in the set that has been in the cache longest (not recently used) with new block which is supposed to be required by CPU.To implement this algorithm, each line in the cache is added with a flag bit, this bit is called as the USE bits. LRU is the most popular algorithm as it is the simplest algorithm to implement.

* First in first out (FIFO): ▫ This algorithm replaces the block which is there in cache memory for the longest time. This algorithm is implemented as Round Robin technique.
* Least frequently used (LFU): ▫ Least frequently used replace block which has had fewest hits. ▫ This algorithm is used when the program contained in the main memory contains non repetitive, sequential set of instructions. LFU replaces the block from cache which has experienced fewer references.
* Random: ▫ This method, randomly replaces a line from cache memory and allocates new line from main memory. ▫ There are some simulated results showing random method being slightly inferior to the other abovementioned methods.