Memory Hierarchy:-

The Computer memory hierarchy looks like a pyramid structure which is used to describe the differences among memory types. It separates the computer storage based on hierarchy.

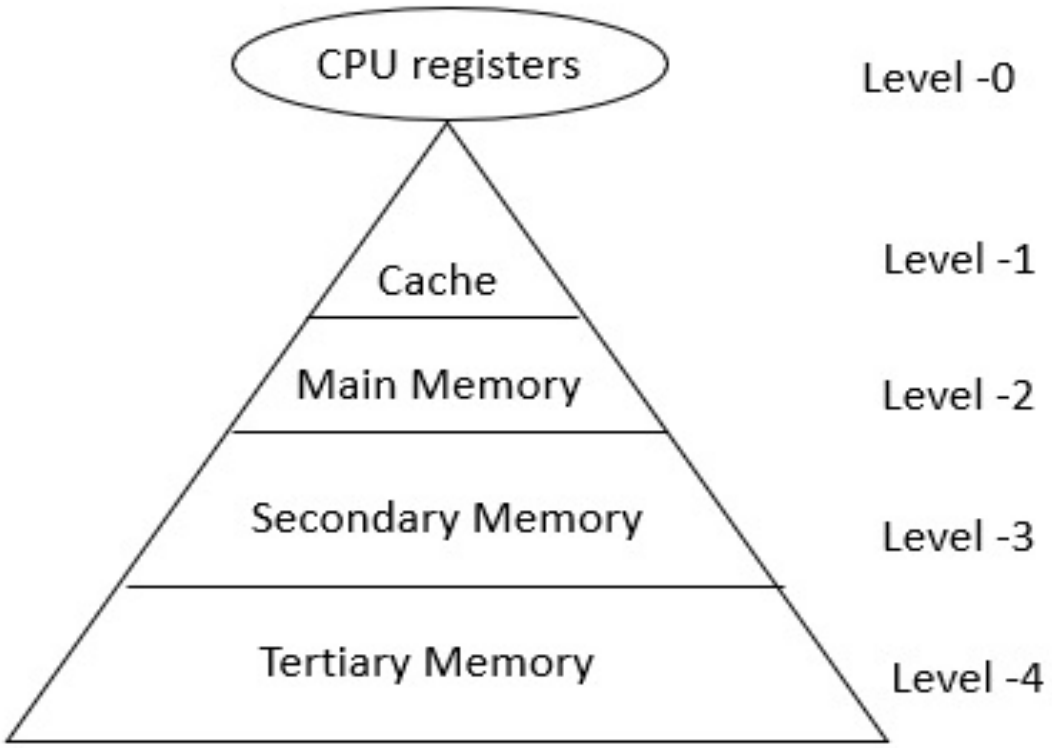
Level 0: CPU registers

Level 1: Cache memory

Level 2: Main memory or primary memory

Level 3: Magnetic disks or secondary memory

Level 4: Optical disks or magnetic types or tertiary Memory



In Memory Hierarchy the cost of memory, capacity is inversely proportional to speed. Here the devices are arranged in a manner Fast to slow that is form register to Tertiary memory.

Let us discuss each level in detail:

Level-0 − Registers

The registers are present inside the CPU. As they are present inside the CPU, they have least access time. Registers are most expensive and smallest in size generally in kilobytes. They are implemented by using Flip-Flops.

Level-1 – Cache(S RAM)

Cache memory is used to store the segments of a program that are frequently accessed by the processor. It is expensive and smaller in size generally in Megabytes and is implemented by using static RAM.

Level-2 − Primary or Main Memory (D RAM)

It directly communicates with the CPU and with auxiliary memory devices through an I/O processor. Main memory is less expensive than cache memory and larger in size generally in Gigabytes. This memory is implemented by using dynamic RAM.

Level-3 − Secondary storage

Secondary storage devices like Magnetic Disk are present at level 3. They are used as backup storage. They are cheaper than main memory and larger in size generally in a few TB.

Level-4 − Tertiary storage

Tertiary storage devices like magnetic tape are present at level 4. They are used to store removable files and are the cheapest and largest in size (1-20 TB).

Let us see the memory levels in terms of size, access time, and bandwidth.

Level	Register	Cache	Primary memory	Secondary memory
Bandwidth	4k to 32k MB/sec	800 to 5k MB/sec	400 to 2k MB/sec	4 to 32 MB/sec
Size	Less than 1KB	Less than 4MB	Less than 2 GB	Greater than 2 GB
Access time	2 to 5nsec	3 to 10 nsec	80 to 400 nsec	5ms
Managed by	Compiler	Hardware	Operating system	OS or user

Associative Memory:-

An associative memory can be treated as a memory unit whose saved information can be recognized for approach by the content of the information itself instead of by an address or memory location. Associative memory is also known as Content Addressable Memory (CAM).

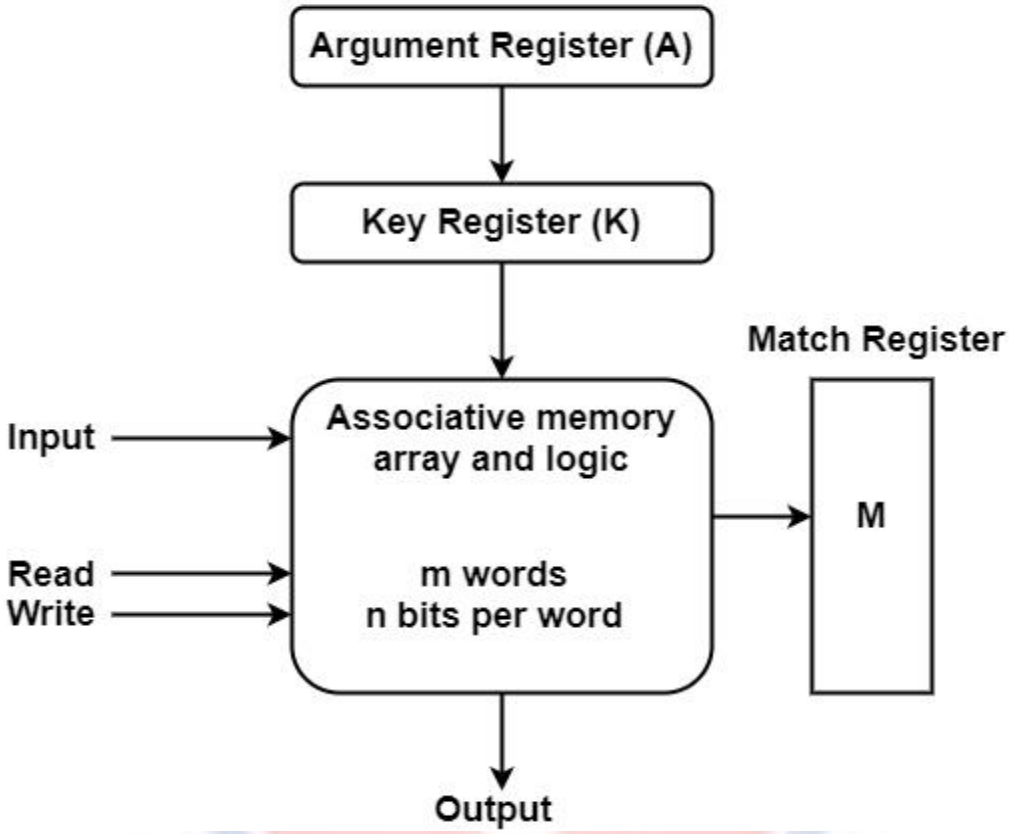
The block diagram of associative memory is shown in the figure. It includes a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word.

The match register M has m bits, one for each memory word. Each word in memory is related in parallel with the content of the argument register.

The words that connect the bits of the argument register set an equivalent bit in the match register. After the matching process, those bits in the match register that have been set denote the fact that their equivalent words have been connected.

Reading is proficient through sequential access to memory for those words whose equivalent bits in the match register have been set.

Block Diagram of Associative Memory



The key register supports a mask for selecting a specific field or key in the argument word. The whole argument is distinguished with each memory word if the key register includes all 1's.

Hence, there are only those bits in the argument that have 1's in their equivalent position of the key register are compared. Therefore, the key gives a mask or recognizing a piece of data that determines how the reference to memory is created.

The following figure can define the relation between the memory array and the external registers in associative memory.

Applications of Associative memory :-

- It can be only used in memory allocation format.

- It is widely used in the database management systems, etc.

Advantages of Associative memory :-

- It is used where search time needs to be less or short.
- It is suitable for parallel searches.
- It is often used to speedup databases.
- It is used in page tables used by the virtual memory and used in neural networks.

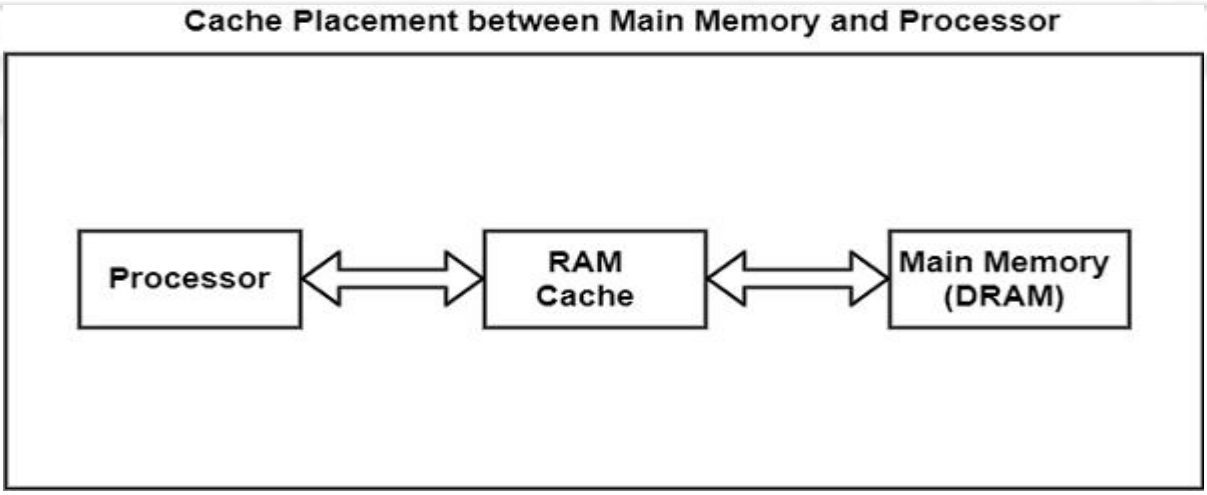
Disadvantages of Associative memory :-

- It is more expensive than RAM.
- Each cell must have storage capability and logical circuits for matching its content with external argument.

Cache Memory:-

The data or contents of the main memory that are used generally by the CPU are saved in the cache memory so that the processor can simply create that information in a shorter time. Whenever the CPU requires to create memory, it first tests the cache memory. If the data is not established in cache memory, so the CPU transfers into the main memory.

Cache memory is located between the CPU and the main memory. The block diagram for a cache memory can be represented as –

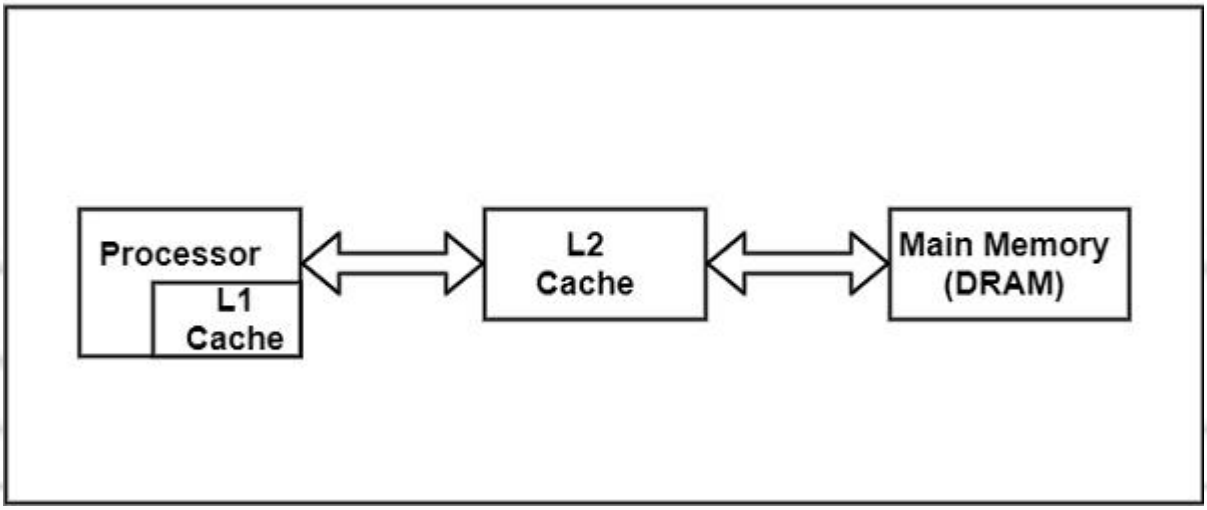


The concept of reducing the size of memory can be optimized by placing an even smaller SRAM between the cache and the processor, thereby creating two levels of cache. This new cache is usually contained inside the processor. As the new cache is put inside the processor, the wires connecting the two become very short, and the interface circuitry becomes more closely integrated with that of the processor.

These two conditions together with the smaller decoder circuit facilitate faster data access. When two caches are present, the cache within the processor is referred to as a level 1 or L1 cache. The cache between the L1 cache and memory is referred to as a level 2 or L2 cache.

The figure shows the placement of L1 and L2 cache in memory.

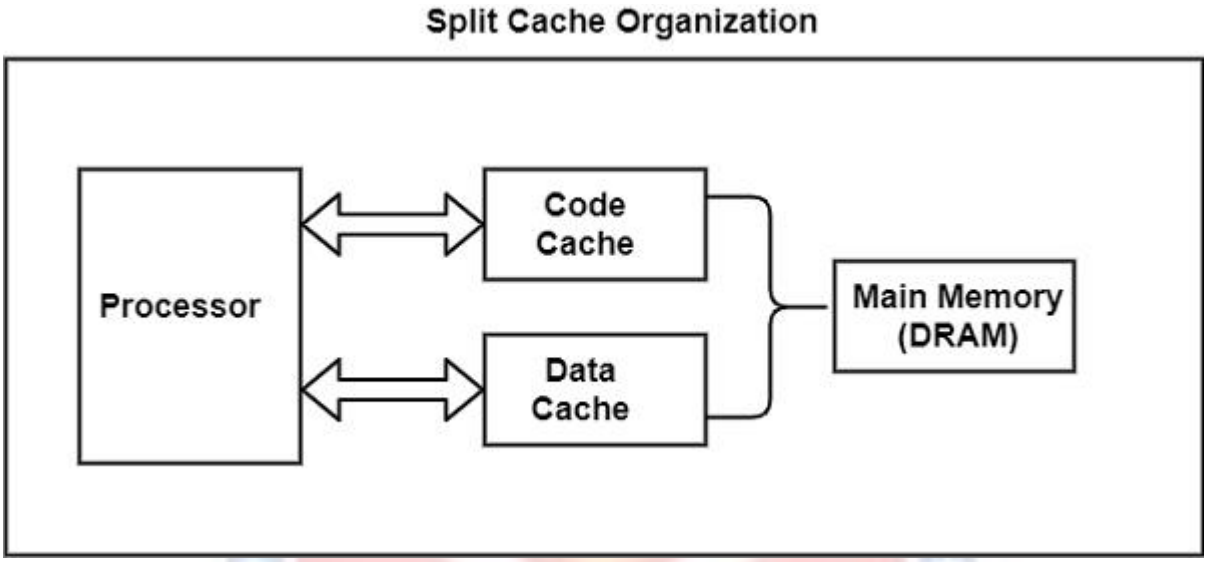
L1 and L2 Cache Placement



The split cache, another cache organization, is shown in the figure. Split cache requires two caches. In this case, a processor uses one cache to store code/instructions and a second cache to store data.

This cache organization is typically used to support an advanced type of processor architecture such as pipelining. Here, the mechanisms used by the

processor to handle the code are so distinct from those used for data that it does not make sense to put both types of information into the same cache.



The success of caches depends upon the principle of locality. The principle proposes that when one data item is loaded into a cache, the items close to it in memory should be loaded too.

If a program enters a loop, most of the instructions that are part of that loop are executed multiple times. Therefore, when the first instruction of a loop is being loaded into the cache, it loads its bordering instructions simultaneously to save time. In this way, the processor does not have to wait for the main memory to provide subsequent instructions.

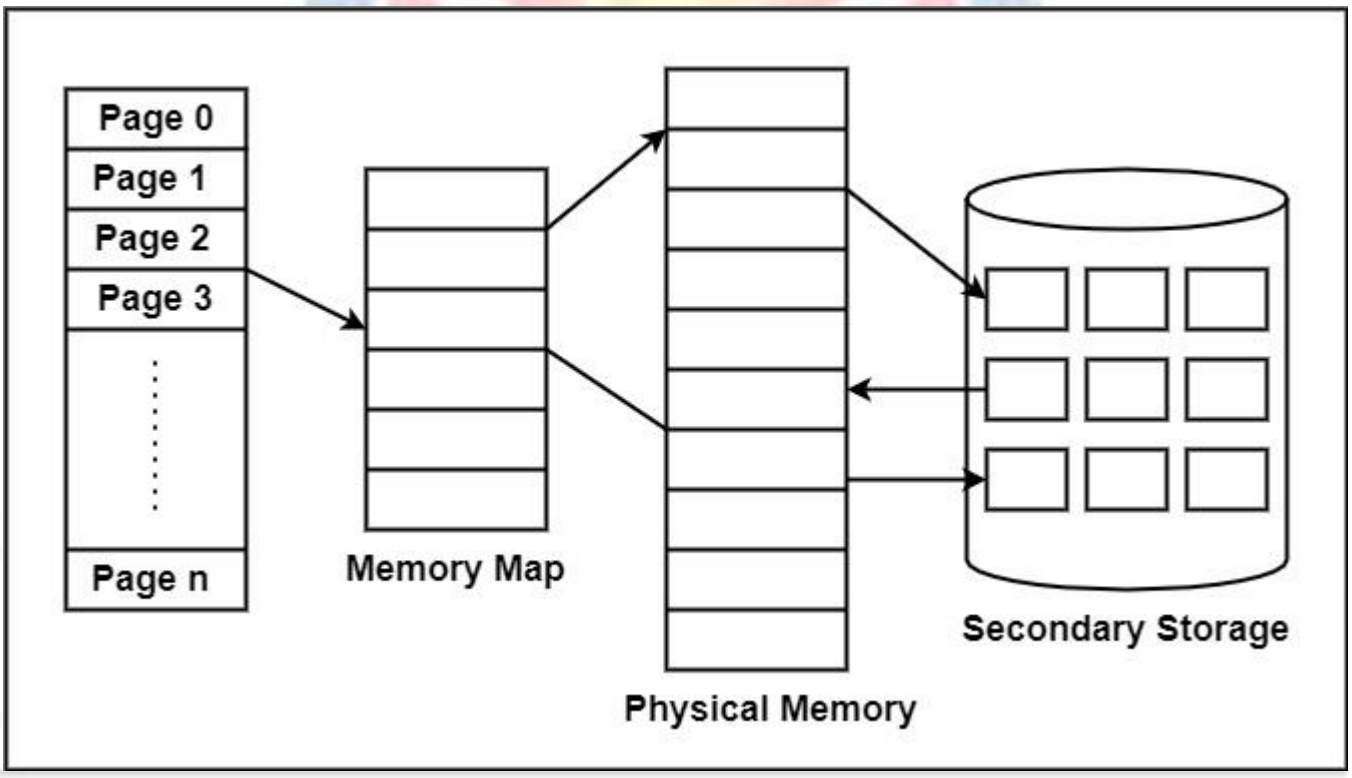
As a result of this, caches are organized in such a way that when one piece of data or code is loaded, the block of neighbouring items is loaded too. Each block loaded into the cache is identified with a number known as a tag.

This tag can be used to find the original addresses of the data in the main memory. Therefore, when the processor is in search of a piece of data or code (hereafter referred to as a word), it only needs to check the tags to see if the word is contained in the cache.

Virtual Memory:-

Virtual memory is the partition of logical memory from physical memory. This partition supports large virtual memory for programmers when only limited physical memory is available.

Virtual memory can give programmers the deception that they have a very high memory although the computer has a small main memory. It creates the function of programming easier because the programmer no longer requires to worry about the multiple physical memory available.



Virtual memory works similarly, but at one level up in the memory hierarchy. A memory management unit (MMU) transfers data between physical memory and some gradual storage device, generally a disk. This storage area can be defined as a swap disk or swap file, based on its execution. Retrieving data from physical memory is much faster than accessing data from the swap disk.

There are two primary methods for implementing virtual memory is as follows

Paging:-

Paging is a technique of memory management where small fixed-length pages are allocated instead of a single large variable-length contiguous block in the case of the dynamic allocation technique. In a paged system, each process is divided into several fixed-size ‘chunks’ called pages, typically 4k bytes in length. The memory space is also divided into blocks of the equal size known as frames.

Advantages of Paging

- There are the following advantages of Paging are −
- In Paging, there is no requirement for external fragmentation.
- In Paging, the swapping among equal-size pages and page frames is clear.
- Paging is a simple approach that it can use for memory management.

Disadvantage of Paging

- There are the following disadvantages of Paging are −
- In Paging, there can be a chance of Internal Fragmentation.
- In Paging, the page table employs more memory.
- Because of Multi-level Paging, there can be a chance of memory reference overhead.

Segmentation:-

The partition of memory into logical units called segments, according to the user’s perspective is called segmentation. Segmentation allows each segment to grow independently, and share. In other words, segmentation is a technique that partition memory into logically related units called a segment. It means that the program is a collection of the segment.

Unlike pages, segments can vary in size. This requires the MMU to manage segmented memory somewhat differently than it would manage paged memory. A segmented MMU contains a segment table to maintain track of the segments resident in memory.

A segment can initiate at one of several addresses and can be of any size, each segment table entry should contain the start address and segment size. Some system allows a segment to start at any address, while other limits the start address. One such limit is found in the Intel X86 architecture, which requires a segment to start at an address that has 6000 as its four low-order bits.

Thank You