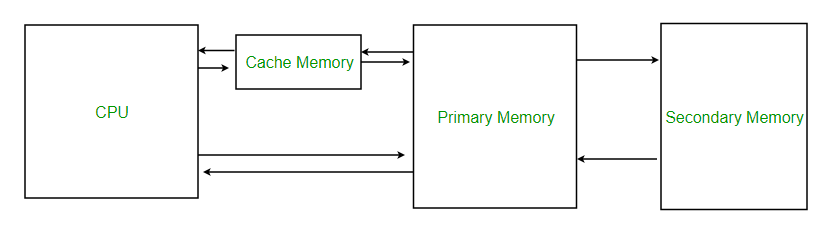
# **Cache Memory in Computer Organization**

Cache Memory is a special very high-speed memory. The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations. There are various different independent caches in a CPU, which store instructions and data. The most important use of cache memory is that it is used to reduce the average time to access data from the main memory.

**Characteristics of Cache Memory**

* Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU.
* Cache Memory holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
* Cache memory is costlier than main memory or disk memory but more economical than CPU registers.
* Cache Memory is used to speed up and synchronize with a high-speed CPU.



*Cache 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 the CPU. The most commonly used register is Accumulator, Program counter, Address Register, etc.
* **Level 2 or Cache memory:** It is the fastest memory that has faster access time where data is temporarily stored for faster access.
* **Level 3 or Main Memory:** It is the memory on which the 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 that is not as fast as the main memory but data stays permanently in this memory.

**Cache Performance**

When the processor needs to read or write a location in the main memory, it first checks for a corresponding entry in the cache.

* If the processor finds that the memory location is in the cache, a [Cache Hit](https://www.geeksforgeeks.org/cache-hits-in-memory-organization/) has occurred and data is read from the cache.
* If the processor does not find the memory location in the cache, a **cache miss** has occurred. For a cache miss, the cache allocates a new entry and copies in data from the main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called **Hit ratio.**

Hit Ratio(H) = hit / (hit + miss) = no. of hits/total accesses

Miss Ratio = miss / (hit + miss) = no. of miss/total accesses = 1 - hit ratio(H)

We can improve Cache performance using higher cache block size, and higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

## Cache Mapping

There are three different types of mapping used for the purpose of cache memory which is as follows:

* Direct Mapping
* Associative Mapping
* Set-Associative Mapping

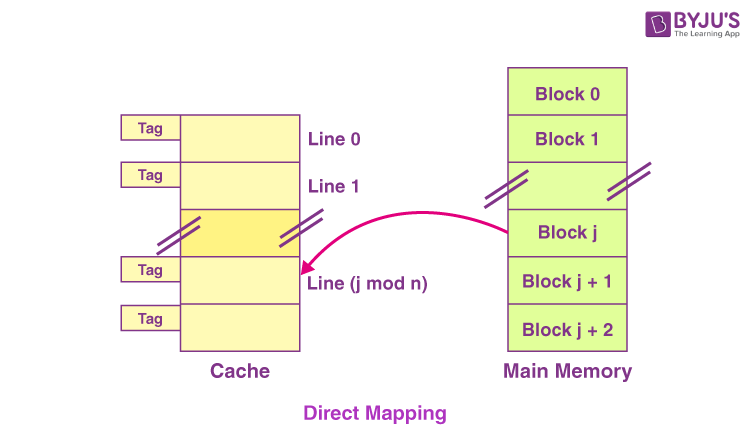
### 1. Direct Mapping

In the case of [direct mapping](https://byjus.com/gate/direct-mapping-notes/), a certain block of the main memory would be able to map a cache only up to a certain line of the cache. The total line numbers of cache to which any distinct block can map are given by the following:

## Cache line number = (Address of the Main Memory Block ) Modulo (Total number of lines in Cache)

#### For example,

* Let us consider that particular cache memory is divided into a total of ‘n’ number of lines.
* Then, the block ‘j’ of the main memory would be able to map to line number only of the cache (j mod n).



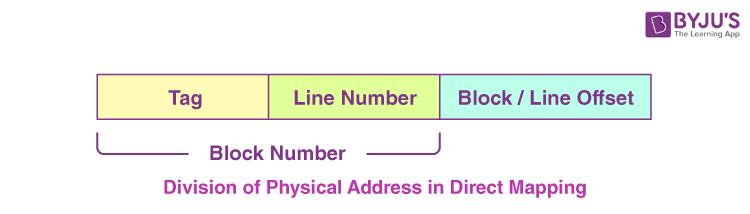
#### The Need for Replacement Algorithm

In the case of direct mapping,

* There is no requirement for a replacement algorithm.
* It is because the block of the main memory would be able to map to a certain line of the cache only.
* Thus, the incoming (new) block always happens to replace the block that already exists, if any, in this certain line.

#### Division of Physical Address

In the case of direct mapping, the division of the physical address occurs as follows:



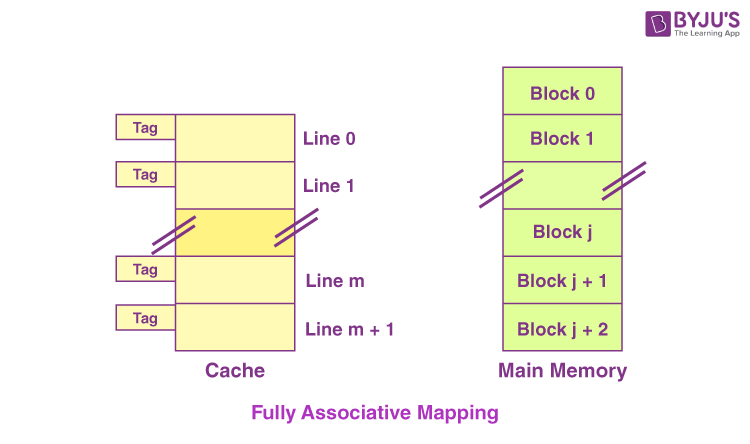
### 2. Fully Associative Mapping

In the case of fully associative mapping,

* The main memory block is capable of mapping to any given line of the cache that’s available freely at that particular moment.
* It helps us make a fully associative mapping comparatively more flexible than direct mapping.

#### For Example

Let us consider the scenario given as follows:



Here, we can see that,

* Every single line of cache is available freely.
* Thus, any main memory block can map to a line of the cache.
* In case all the cache lines are occupied, one of the blocks that exists already needs to be replaced.

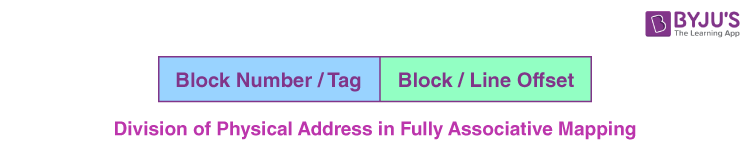
#### The Need for Replacement Algorithm

In the case of fully associative mapping,

* The replacement algorithm is always required.
* The replacement algorithm suggests a block that is to be replaced whenever all the cache lines happen to be occupied.
* So, replacement algorithms such as LRU Algorithm, FCFS Algorithm, etc., are employed.

#### Division of Physical Address

In the case of fully associative mapping, the division of the physical address occurs as follows:



### 3. K-way Set Associative Mapping

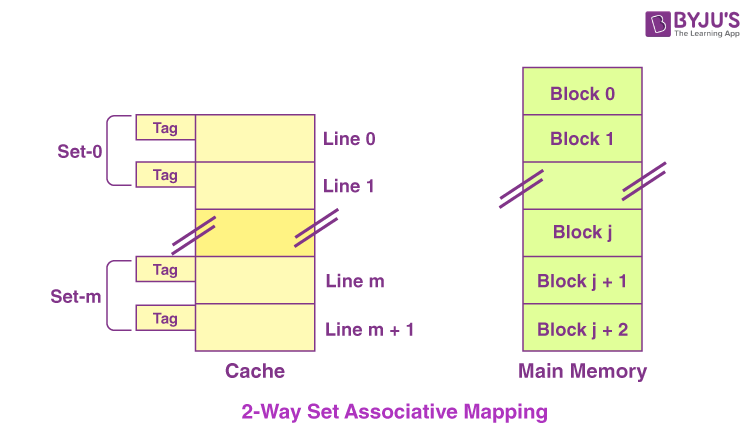
In the case of k-way set associative mapping,

* The grouping of the cache lines occurs into various sets where all the sets consist of k number of lines.
* Any given main memory block can map only to a particular cache set.
* However, within that very set, the block of memory can map any cache line that is freely available.
* The cache set to which a certain main memory block can map is basically given as follows:

Cache set number = ( Block Address of the Main Memory ) Modulo (Total Number of sets present in the Cache)

### For Example

Let us consider the example given as follows of a two-way set-associative mapping:



In this case,

* k = 2 would suggest that every set consists of two cache lines.
* Since the cache consists of 6 lines, the total number of sets that are present in the cache = 6 / 2 = 3 sets.
* The block ‘j’ of the main memory is capable of mapping to the set number only (j mod 3) of the cache.
* Here, within this very set, the block ‘j’ is capable of mapping to any cache line that is freely available at that moment.
* In case all the available cache lines happen to be occupied, then one of the blocks that already exist needs to be replaced.

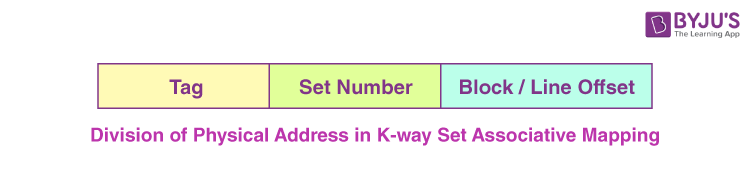
#### The Need for Replacement Algorithm

In the case of k-way set associative mapping,

* The k-way set associative mapping refers to a combination of the direct mapping as well as the fully associative mapping.
* It makes use of the fully associative mapping that exists within each set.
* Therefore, the k-way set associative mapping needs a certain type of replacement algorithm.

#### Division of Physical Address

In the case of fully k-way set mapping, the division of the physical address occurs as follows:



#### Special Cases

* In case k = 1, the k-way set associative mapping would become direct mapping. Thus,

Direct Mapping = one-way set associative mapping

* In the case of k = The total number of lines present in the cache, then the k-way set associative mapping would become fully associative mapping.

## Advantages of Cache Memory

* Cache Memory is faster in comparison to main memory and secondary memory.
* Programs stored by Cache Memory can be executed in less time.
* The data access time of Cache Memory is less than that of the main memory.
* Cache Memory stored data and instructions that are regularly used by the CPU, therefore it increases the performance of the CPU.

## Disadvantages of Cache Memory

* Cache Memory is costlier than primary memory and secondary memory.
* Data is stored on a temporary basis in Cache Memory.
* Whenever the system is turned off, data and instructions stored in cache memory get destroyed.
* The high cost of cache memory increases the price of the Computer System.