Cache Mapping Techniques

Cache mapping is a technique that is used to bring the main memory content to the cache or to identify the cache block in which the required content is present. In this article we will explore cache mapping, primary terminologies of cache mapping, cache mapping techniques I.e., direct mapping, set associative mapping, and fully associative mapping. Let's start exploring the topic "Cache Mapping".

Primary Terminologies

Some primary terminologies related to cache mapping are listed below:

- **Main Memory Blocks:** The main memory is divided into equal-sized partitions called the main memory blocks.
- **Cache Line:** The cache is divided into equal partitions called the cache lines.
- **Block Size:** The number of bytes or words in one block is called the block size.
- **Tag Bits:** Tag bits are the identification bits that are used to identify which block of main memory is present in the cache line.
- **Number of Cache Lines:** The number of cache lines is determined by the ratio of cache size divided by the block or line size.
- **Number of Cache Set:** The number of cache sets is determined by the ratio of several cache lines divided by the associativity of the cache.

What is Cache Mapping?

Cache mapping is the procedure in to decide in which cache line the main memory block will be mapped. In other words, the pattern used to copy the required main memory content to the specific location of cache memory is called cache

mapping. The process of extracting the cache memory location and other related information in which the required content is present from the main memory address is called as cache mapping. The cache mapping is done on the collection of bytes called blocks. In the mapping, the block of main memory is moved to the line of the cache memory.

Need for Cache Mapping

Cache mapping is needed to identify where the cache memory is present in cache memory. Mapping provides the cache line number where the content is present in the case of cache hit or where to bring the content from the main memory in the case of cache miss.

Important Points Related to Cache Mapping

Some important points related to cache mappings are listed below.

- The number of bytes in main memory block is equal to the number of bytes in cache line i.e., the main memory block size is equal to the cache line size.
- Number of blocks in cache = Cache Size / line or Block Size
- Number of sets in cache = Number of blocks in cache / Associativity
- The main memory address is divided into two parts i.e., main memory block number and byte number.

Cache Mapping Techniques

There are three types of cache mappings namely:

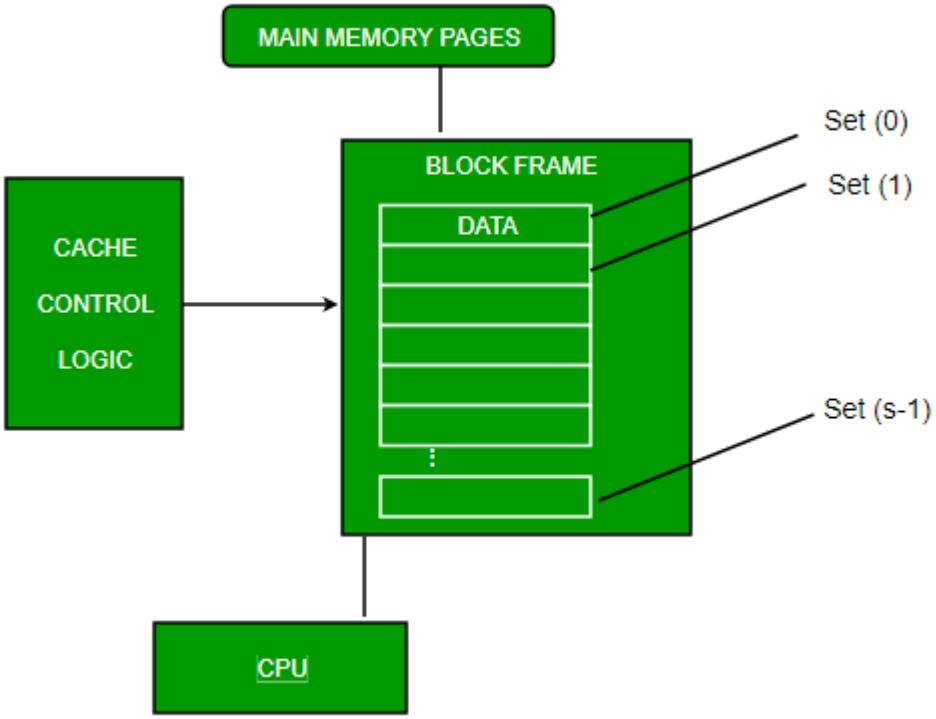
- Direct Mapping
- Fully Associative Mapping
- Set Associative Mapping

Direct Mapping

In direct mapping physical address is divided into three parts i.e., Tag bits, Cache Line Number and Byte offset. The bits in the cache line number represents the cache line in which the content is present whereas the bits in tag are the identification bits that represents which block of main memory is present in cache. The bits in the byte offset decides in which byte of the identified block the required content is present.

Tag	Number of Cache Lines	Byte Offset

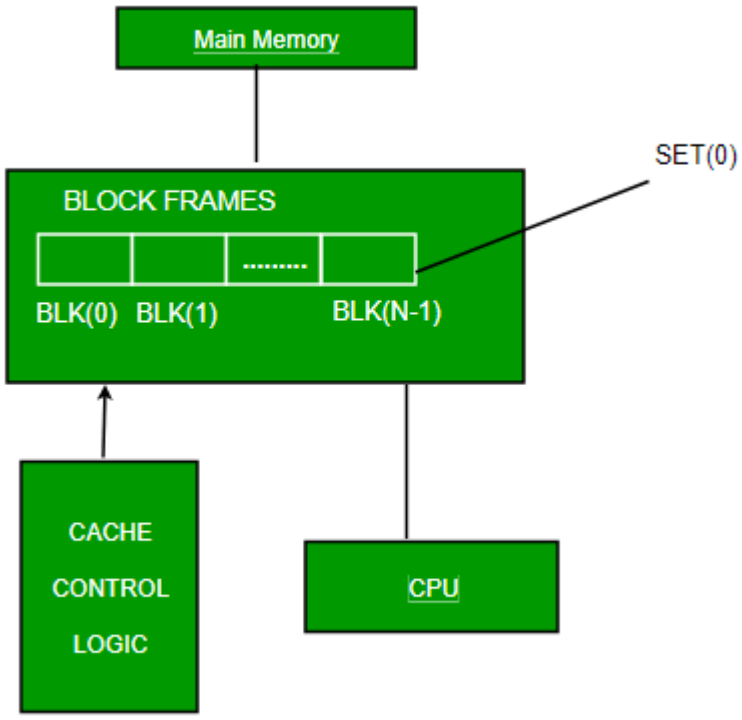
Cache Line Number = Main Memory block Number % Number of Blocks in Cache



Fully Associative Mapping

In fully associative mapping address is divided into two parts i.e., Tag bits and Byte offset. The tag bits identify that which memory block is present and bits in the byte offset field decides in which byte of the block the required content is present.

Tag	Byte Offset



Set Associative Mapping

In set associative mapping the cache blocks are divided in sets. It divides address into three parts i.e., Tag bits, set number and byte offset. The bits in set number decides that in which set of the cache the required block is present and tag bits identify which block of the main memory is present. The bits in the byte offset field gives us the byte of the block in which the content is present.

Tag	Set Number	Byte Offset

Cache Set Number = Main Memory block number % Number of sets in cache