

**CO END-SEM ASSIGNMENT**

**ON**

**Cache Simulation**

**By**

**ROBIN GARG**

**2019092**

**SEC-A**

## Documentation

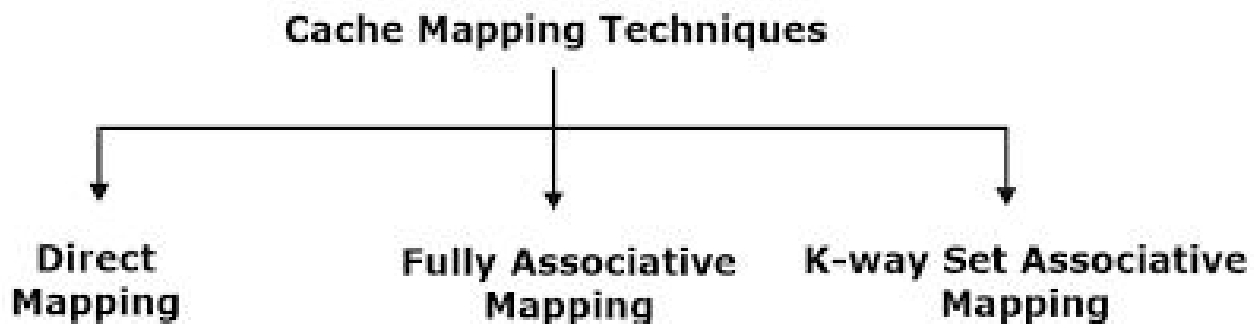
# Cache Mapping

Cache memory mapping is the way in which we map or organise data in cache memory, this is done for efficiently storing the data which then helps in easy retrieval of the same.

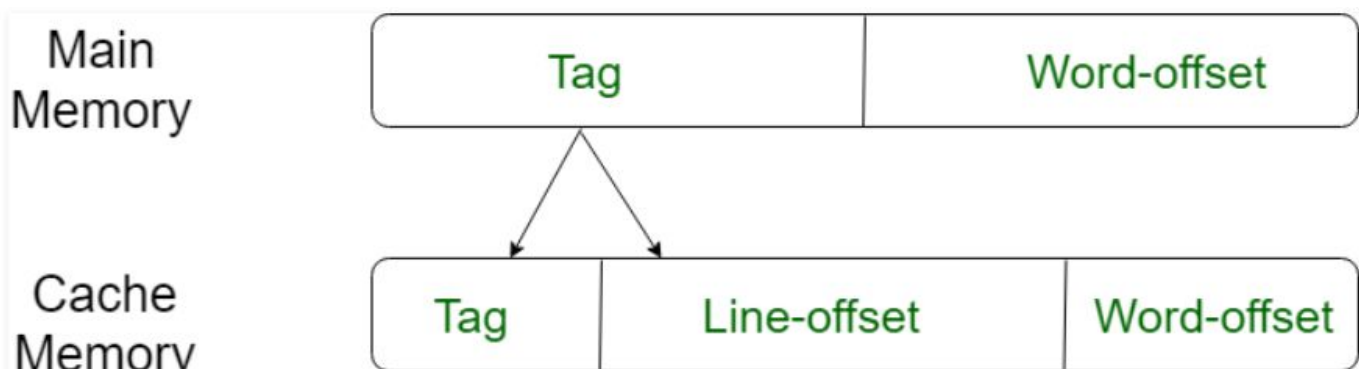
Cache is mapped (written with data) every time the data is to be used by the CPU this is done with the help of cache write policies such as Write through, Write Back, etc.

And this writing of data is done again with the help of certain techniques

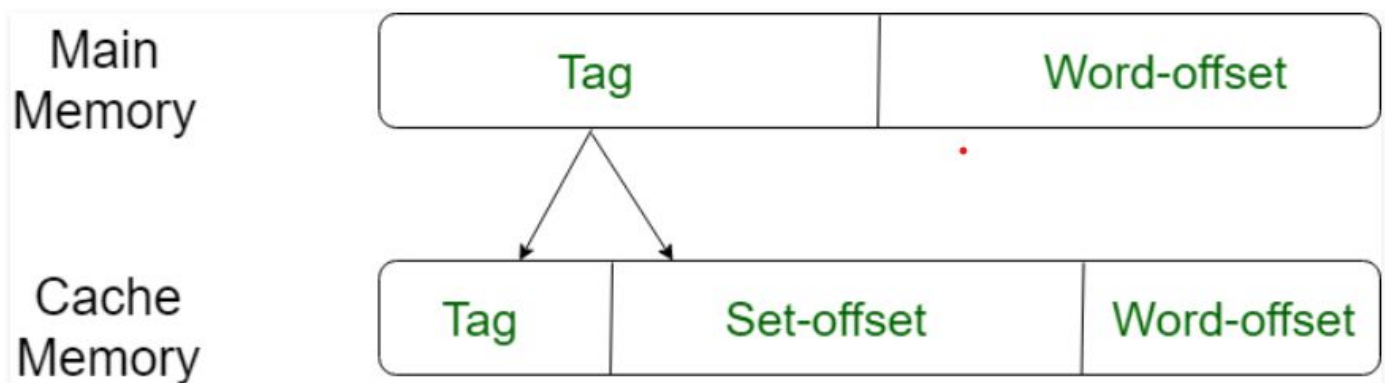
There are three different types of Cache Mapping:



- 1. Direct Mapping:** In direct mapping we map each block of main memory to only single cache line. So when there is another block needed to add in the same cache line we need to delete the previous block. In this we break address of block into a tag and line offset which gives the Cache line number.



- 2. Fully Associative Mapping:** In this type of mapping, we can map each block of main memory to any cache line. This uses CAM hardware to search in each cache line parallelly. CAM is very expensive hardware and direct mapping loses the capability to load multiple block.
- 3. N- way Set Associative Mapping:** We have N-way set associative mapping to tackle the disadvantage of both direct and fully associative mapping. 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.



We needed to simulate all three types of cache mapping. I have used JAVA language to simulate mappings in my program. Below are my main functions to map-cache by all types.



```
// METHOD FOR DIRECT MAPPING  
static void Direct()  
{
```



```
// METHOD FOR ASSOCIATIVE MAPPING  
static void Associative()  
{
```



```
METHOD FOR N-WAY SET ASSOCIATIVE  
static void nassociativeread(String S,int b)  
{
```

How to run the simulation:

★ All the numerical inputs are in power of 2.

First, we need to enter the Number of Cache lines and Block Size of Cache.

```
Command Prompt - java Robin_2019092_FinalAssignment.java  
Input Number of Cache Lines  
128_
```

```
Input Block Size  
16_
```

Then we'll specify the type of mapping we are going to use

```
Command Prompt - java Robin_2019092_FinalAssignment.java  
What do you wanna do:  
Presss 1 for Direct Mapping  
Press 2 for Associative Mapping  
Press 3 for n-way Associative Mapping  
Press 4 for Exit  
3
```

I have chosen n-way associative mapping.

Now I need to enter the value of n.

```
Command Prompt - java Robin_2019092_FinalAssignment.java
Enter the n for Associative mapping:
2_
```

Now a common window will prompt to whose either read or write:

```
Command Prompt - java Robin_2019092_FinalAssignment.java
What do you wanna do:
Presss 1 for read
Press 2 for write
Press 3 for exit
2
```

★ To write into the cache we'll need the address of memory location which can be either of lengths 16,32,64.

I'm entering a 16-bit address and writing my name into that location.

```
Command Prompt - java Robin_2019092_FinalAssignment.java
Input the address to be readed
0000100000011100
Address Not Found
The Address has been added to Cache
_
```

The address was not present in the cache so it is now added to the cache.

```
Command Prompt - java Robin_2019092_FinalAssignment.java
Input the address to be readed
0000100000011100
Address Not Found
The Address has been added to Cache

Input the Data to be added
Robin
Given data is added to the Cache
The New value in the address is:  Robin
```

The value of data is being updated at the correct offset location.

Now, When the blocks of a set are filled replacement will take place according to LRU(least recently used) method and a new block will be added to the set.

```
Command Prompt - java Robin_2019092_FinalAssignment.java
Input the address to be readed
0000000000011010
Address Not Found

The Cache block is being replaced by new Block following the LRU method
The Address has been added to Cache
Input the Data to be added
Section-A
Given data is added to the Cache
The New value in the address is:  Section-A
```

★P.S.- The user is required to provide all the inputs in the specified format only to run the program correctly.

## REFERENCES

<https://www.geeksforgeeks.org/cache-memory-in-computer-organization/>