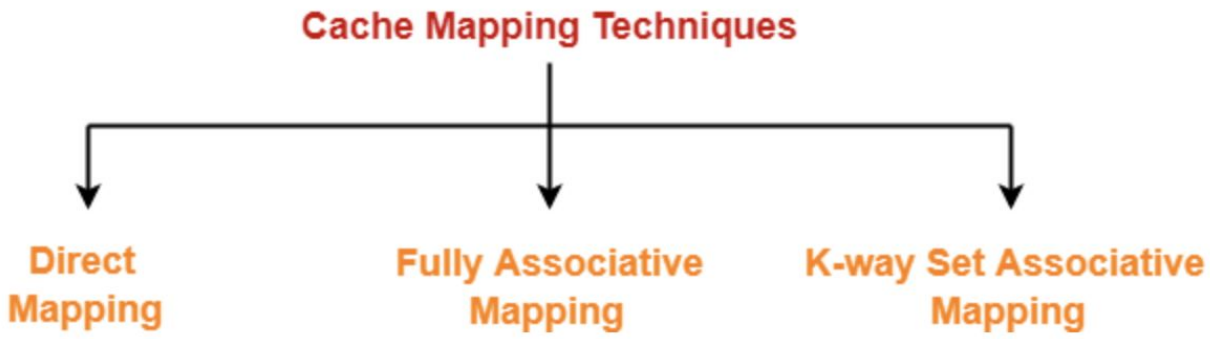
Computer Organisation

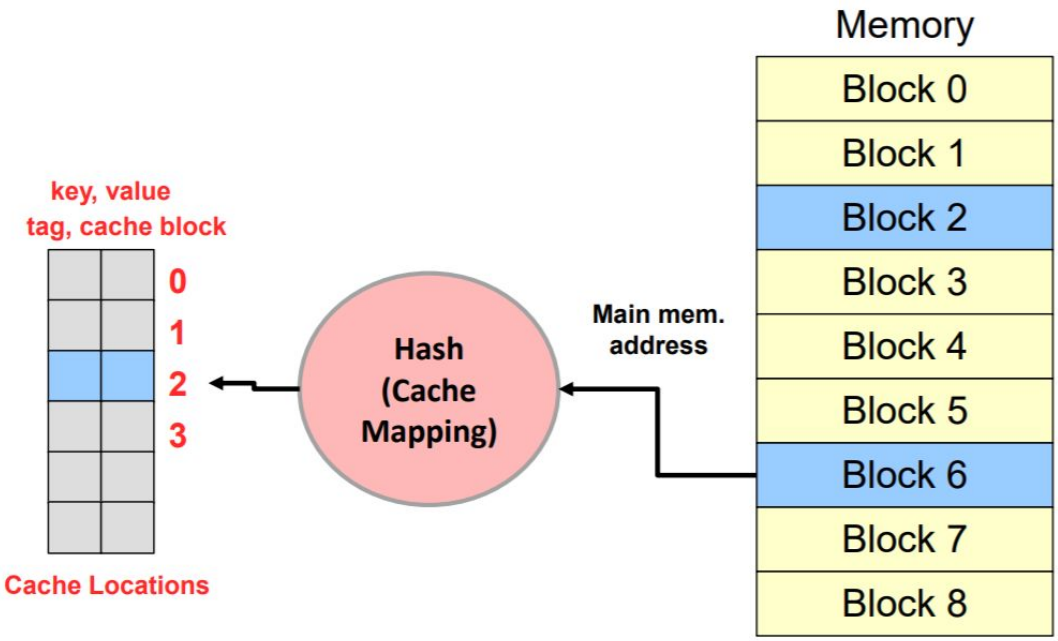
Cache Assignment

Made by-Utkarsh Dubey
(2019213)

Types of Cache Mapping Techniques:

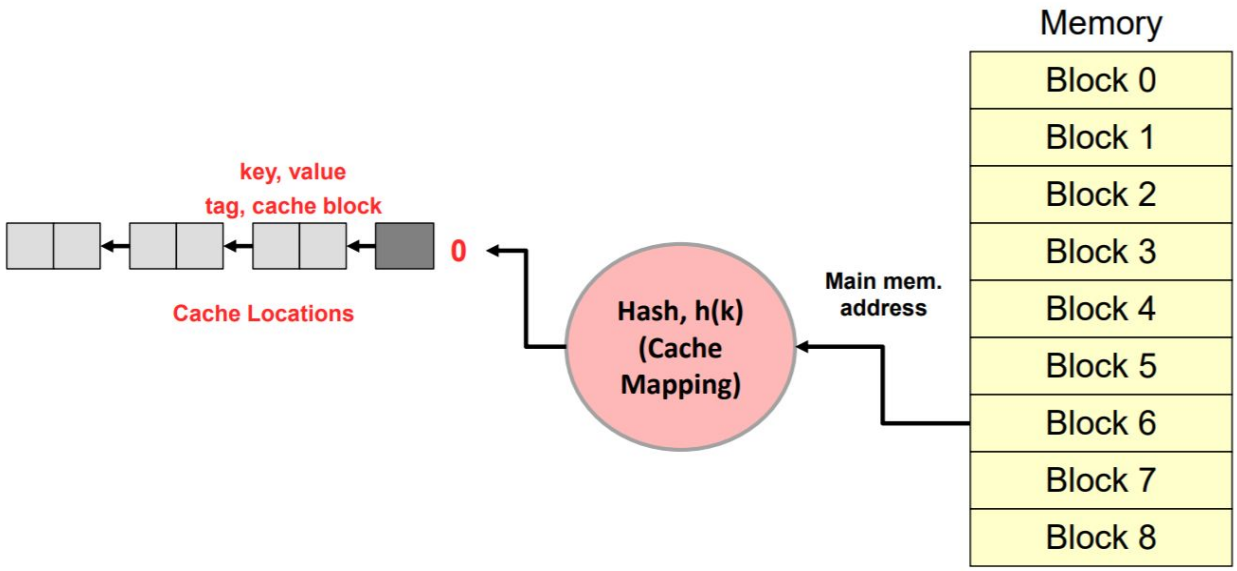


1- Direct Mapping



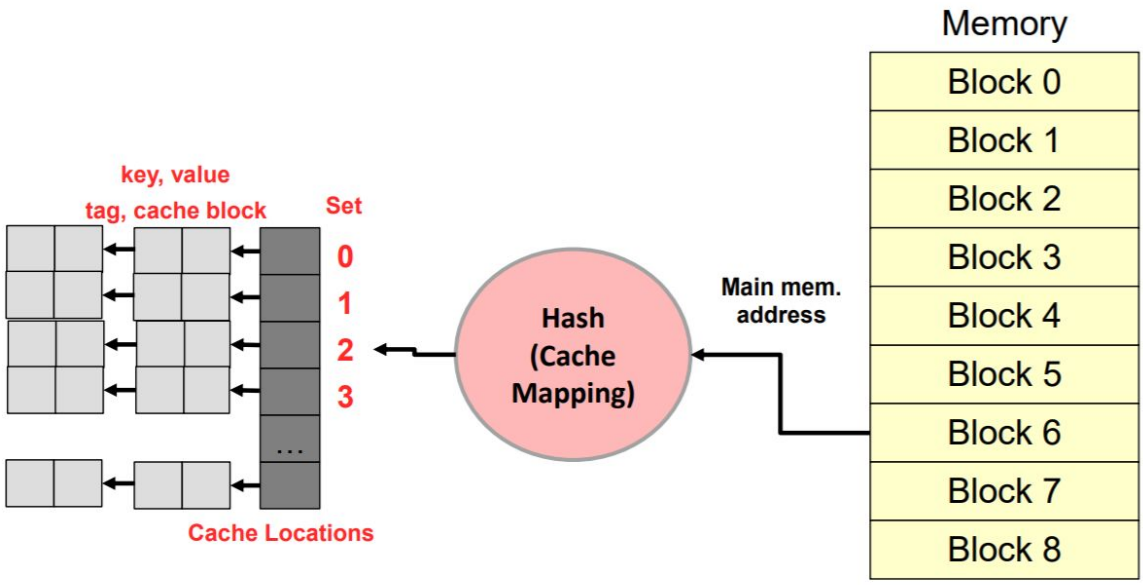
- In this type of mapping, each block of memory maps to exactly one fixed line of cache.
- Fixed-line for a particular block is found using the formula -
    - Cache line = Block number % total cache lines

2- Fully Associative



- In this type of mapping, any block can go to any cache line.
- Once the cache is full, new entries can be accommodated using certain algorithms like:
    - FIFO
    - LRU

3- K-way Set Associative Mapping



- In this type of mapping, Cache is divided into sets of size k each, and certain blocks are allowed to be loaded into certain sets only.
- The formula used to find the set number is:
  - Set number = Block number % k
- Once the cache is full, new entries can be accommodated using certain algorithms like:
  - FIFO
  - LRU

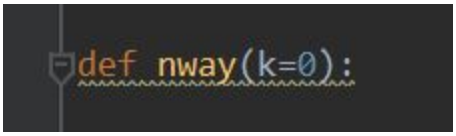
The layout of the program:

- Language used - Python
- The program contains mainly 4 sections:
  - Main

  - def direct()

  - def associative()

  - def nway(k)



Assumptions:

- Proper inputs should be provided by the user.
- The address provided by the user should be in binary.
- The input address should be of 16/32/64 bit depending on the choice of the user.

The algorithm used for replacement:

LRU (Least Recently Used) has been used in order to implement the replacement when the cache is full.

Explaining the different section:

Main:

This section is mainly used to initialize the code and construct a menu-driven program and ultimately calling all different sections.

```
print("Welcome to the cache stimulation program")
print()
print("Choose the type of mapping you want to work with")
print("1- Direct Mapping")
print("2- Associative Memory")
print("3- n-way set Associative Memory")
print("press the corresponding number in order to continue")
choice=int(input())
flag=0
if(choice==1):
    flag=direct()
elif(choice==2):
    flag=associative()
elif(choice==3):
    print("Enter the value of n")
    k=int(input())
    flag=nway(k)
```

def direct():

This section is coded in order to implement the direct mapping technique. Further in this section, certain options are given like read and write.
And all the calculations are done using the inputted address like calculating tag, offset, line number for inputting into the cache.

```
print("Choose what you want to do-")
print("1- Read from the cache")
print("2- Write into cache")
print("3- Leave current type of Mapping")
print("press the corresponding number in order to continue....")

choice=int(input())
```

def nway(k):

This section is coded in order to implement k-way set associative mapping. It takes in an argument k representing the size of each set in which the cache will be divided.
Further same options are provided like direct section and the same inputs have been taken but in this address is used in order to calculate tag, offset and set number in which the given address can be pushed.

```
setnumber = int(address[-(blockpower + setpower):-blockpower], 2)
```

def associative():

This section is coded similar to the nway(k) just the difference being that here k will be equal to the number of lines present in cache hence only one possible set will be created and the further implications will be same

```
number = lines
setpower = int(log(lines // number, 2))
```

INPUT/OUTPUT

- The program starts by giving out the option to choose the type is mapping which is to be implemented.

```
Welcome to the cache stimulation program

Choose the type of mapping you want to work with
1- Direct Mapping
2- Associative Memory
3- n-way set Associative Memory
press the corresponding number in order to continue
1
```

- Further entering the possible mapping user will be asked to enter the cache size and number of cache lines.

```
1
Enter the size of cache
16
Enter the number of lines in cache
4
```

- But on entering the n-way mapping you will be asked to enter an additional n.

```
Choose the type of mapping you want to work with
1- Direct Mapping
2- Associative Memory
3- n-way set Associative Memory
press the corresponding number in order to continue
3
Enter the value of n
2
```

- After entering the cache size and number of lines read and write options will be provided

```
Choose what you want to do-
1- Read from the cache
2- Write into cache
3- Leave current type of Mapping
press the corresponding number in order to continue....
```

- Then on choosing the option read you will be asked the address and on choosing the option write you will be asked to enter the address as well as the data and output will be shown depending on whether it a read miss, read hit, write hit, write miss or a replacement.

```
Choose what you want to do-
1- Read from the cache
2- Write into cache
3- Leave current type of Mapping
press the corresponding number in order to continue....
1
Enter the address
0000000010100101
Read miss
Given address not present in the cache
```

```
Choose what you want to do-
1- Read from the cache
2- Write into cache
3- Leave current type of Mapping
press the corresponding number in order to continue....
2
Enter the address
0000110100000101
Enter the data
1101001010111010
write miss
writing in set: 1 and in line: 3
```

References:

http://ee.usc.edu/~redekopp/cs350/slides/Ch9_Caching.pdf
https://www.gatevidyalay.com/