



Cache Project

(2 – Level Cache Implementation)

By:-

SHUBHAM LOHAN
2019275

Working of Code:

In my code I'm using Cache of 2 Level. Cache implementation is completely based on the principle of exclusion policy.

How Cache Data is stored:

All the Data of cache is stored in List.

Detail working of code:

When an address is entered to Read/Write a block

Then there are two cases arises

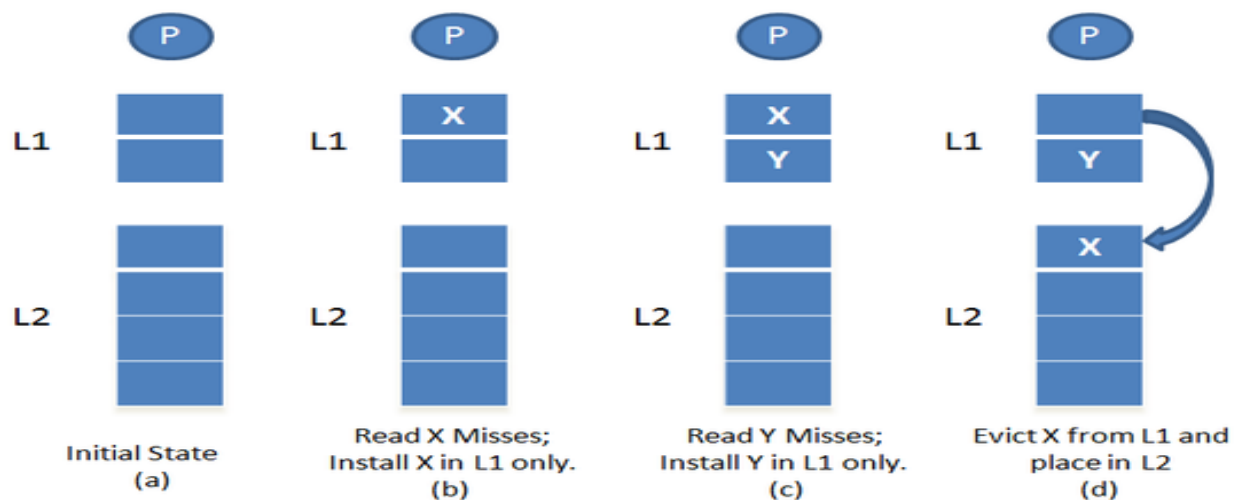
- 1. Data is present in the cache**
- 2. Data is not present in the cache**

If Data is present in the cache:

1. If Data is present in the L1: then it prints cache hit and also print the entire data of cache memory
2. If data is present in L2: Because I'm using exclusion policy, the data present in L2 shift to L1. then it prints cache hit and also print the entire data of cache memory

If Data is not present in the cache:

then the data is load to L1 cache memory and print cache miss and the entire data of cache memory.



Assumptions:

1. If the input address entered don't present in the cache while reading/ writing In the cache then data is load to cache memory.
2. If user enter the address of n bit (satisfying required condition) in the starting Then cache become of n bit.
3. As there is no use of taking input of no of cache Line because it can be easily calculated from the cache size and block size, I'm not taking the input of no of cache Lines.
4. If any word or block is replaced while loading or writing the data then it will go to L2 cache memory if there was not a space available for that block then, I print the whole data of that block no along with block no.
5. Block size should be greater than 1 and in power of 2.

Error Handling:

1. Both cache size and block size should be in power of 2 if the input not in power of 2 then it will be asking the user until he did not enter the correct size.
2. If Block size is greater than the L1 cache memory size then program will terminate with error message.
3. Address should be in binary and greater than the block offset otherwise it will be asking the using until he did not enter the correct address.
4. All the input other than above (for example taking input for reading or writing and input for cache mapping) will take input as long as when the user did not enter the correct input (except for 0 to continue the program)
5. If initially user enter the data of 16 bit and after that if he enters the data of bit not equal to 16 then program throws error.