Lab-6 (Cache Miss Simulator)
CO21BTECH11004

Approach: -

Input: -

- Read the input file line by line.
- Store input as a vector of strings.
- For address, make another vector of strings containing the address in binary form.

Classes used: -

- Desc:- This class is made to configure the cache. It is constructed by using input from file cache.config.
- Cache:- This class contains arrays to store tags, valid bit, dirty bit, LRU counter, and FIFO counter, which are used to determine cache hit/miss based on input from caches.access.

For each line in access.txt, the following is done: -

- Read
    - Find offset, index, tag.
    - Find the memory location in the cache by comparing tags in the index's set.
    - If not, there is a miss.
        - If the set is not full, find the first empty location and bring in the cache.
        - If the set is full, find the corresponding line by policy and bring in the cache.
    - If present, there is a hit.
- Write back (write allocate) has the same idea as read.
- Write through (not allocate)
    - If present, there is a hit.
    - If not, there is a miss
        - Do nothing (Don't bring in cache).

Output: - Conversion of address from binary (pad with 0 if required) to hex, then print.

Code File: -

- cacheMissSimulator.cpp
- Compile the file. While running executable give input files names (like argc/argv parameters)

Testing the code: -

- Test cases given in assignment.
- Other simple test cases, on corner cases.