Report

Velma Dhatri Reddy
AI21BTECH11030

CODING APPROACH:

I have implemented the Cache Miss Simulator using C++. Input for cache parameters is taken from cache.config and mode and addresses are taken from cache.access. These values are passed to the function cacheAccess. In this function bits required for block offset and tag are calculated and the set and tag are printed. Assuming the cache was empty initially I have initialised the hit to be false which is updated to true if it is found.

I have checked if it is Fully associative (ASSOCIATIVITY = 0) and set associative and direct mapped with (ASSOCIATIVITY != 0)

If there is no more space in the block I have written the conditions for replacement policies (FIFO, RANDOM, LFU). For LFU I have used queue to pop the least recently used.

I have written if conditions for write policies in which WriteBack (WB) updates the cache in R and W mode (when there's a miss) and WriteThrough (WT) updates the cache only in R mode.

TESTING:

I have tested using an example from the lecture. I have tested with few random test cases.

(Not working for few cases of LRUs but rums for all cases on online compilers)