Adam Tecle

The implementation of this cache simulator was guided by two prevailing factors: the time I could afford to spend on the assignment, and effectiveness of my methods. The time constraint posed limitations on the functionality the interface could provide the user. As a result, only a write through direct-mapped cache with a FIFO replacement policy is supported. While writing this program, I deemed the effectiveness of a particular method, that is, the ability of a piece of code to do what it was intended to, as the most important attribute to my design. Although it may have compromised quality and efficiency, given the time constraint and my limited knowledge of the C language, I was satisfied to create solutions that worked, if nothing else.

To implement a the cache I created an array of structures, each structure holding the tag, valid bit, and set index, all initially set to zero. As I read through the trace file, I extracted the relevant memory addresses, computed its tag, set bits, and block offset according to the user parameters, and accessed the cache array to determine a hit or a miss. Because it is direct-mapped, the FIFO replacement policy, did not require me to keep track of which line in the set was put in place first, given that there is only one line. As for the write through policy, I simply incremented a counter each time I replaced any data.