TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #3, Module: MA5612

Gustavo Ramirez

November 18, 2016

0.1 SOLUTION

In general, the program is structured in the following way (two of the implemented functions are omitted, because they're not relevant to the general structure of the program):

- query_cache [function]: one important detail, on the internal specific structure of this implemented cache, is on the implementation of the LRU characteristic of this cache. For this, each line in each subset is characterized by a number; a 0 represents a highly used line, whereas a 3 (2) represents the least used cache line for 4-way (2-way). For the LRU behaviour of the cache, is also useful to know that, for this particular implementation, all cache lines are initialized to -1, regarding addresses. This way, cache lines with a -1 in the address will be prefered for substitution over those with information in the address field.
- get_set_id: for the full implementation of this function (which returns a set id, given a hexadecimal address in string form), two more functions were implemented, both for parsing between hexadecimal, decimal, binary (some in string format, some in int format).
- main(): first, it's important to notice that the order in which the parameters are specified when executing the program, is irrelevant (i.e. the program can be called like ./cache -s 128 -l 16 -a 2 -f addressfile or ./cache -l 16 -a 2 -s 128 -f addressfile).

After execution (and on hit rates):

- a file named execution_2WAY_cache.txt was generated for the case of executing the 2-way cache
- a file named execution_4WAY_cache.txt was generated for the case of executing the 4-way cache

From the first file, can be seen that a 12.5 % of hit rate was obtained for the 2-way cache, and from the second a 6.25 % for the 4-way cache.