CS 3339
Computer Architecture
Fall 2014

Project 5

Due: *Thursday, November 13 at 3:00 PM*

All project files have to be submitted using TRACS. Please follow the instructions at http://tracsfacts.its.txstate.edu/trainingvideos/submitassignment/submitassignment.htm. Note that files are only submitted if TRACS indicates a successful submission. See the end of this handout for what files need to be submitted. This project can be done individually. You have to be able to explain all code that you submit.

5.1 MIPS Data Cache Statistics

Augment your interpreter from Project 2 such that it simulates a data cache. No new code is provided for this project. The data cache should have the following parameters: associativity = 5 ways, replacement policy = round robin (do not look for invalid entries), write policy = write-allocate and write-back, block size = 32 bytes, sets = 8. All blocks in the cache are initially invalid. Since this is a data cache, only loads and stores access it, instruction fetches do not. You only need to model the behavior of the cache, so you do not actually have to store any data in it. Only simulate the valid, tag, and dirty bits as well as the round-robin replacement policy. Count the number of accesses, misses, and write backs. Then calculate the hit ratio. Note that all dirty blocks have to be written back at the end of the program before the statistics are computed.

I recommend you write three functions, one that is called before the first instruction to initialize the cache, one that is called after the last instruction to print the cache statistics, and one that is called from every lw and sw instruction to model the cache accesses.

The following is the expected result for *sssp.mips*. Your output must match this format verbatim.

```
CS3339 -- MIPS Cache Simulator
running sssp.mips

 7  1

program finished at pc = 0x400440  (449513 instructions executed)

accesses: 197484
misses: 11281
writebacks: 4960
hit ratio: 94.3%
```

Code Requirements
- Make sure your code compiles with gcc and runs on zeus.cs.txstate.edu.
- Make sure your code is well commented.
- Make sure your code does not produce unwanted output such as debugging messages.

- Make sure your code's runtime is not excessive.
- Make sure your code is correctly indented and uses a consistent coding style.
- Make sure your code does not exceed array bounds.
- Make sure your code does not include unused variables, unreachable code, etc.

File Submission
- Delete all files that you do not need anymore such as core and *.o files.
- Make sure your code complies with the above code requirements before you submit it.
- Any special instructions or comments to the grader should be included in a "README" file.
- Upload your final **`cache.c`** file (all code you write has to go into this file) and the optional **`README.txt`** file onto TRACS. Both files have to be text files, not PDF etc.
- Upload each file separately and do not compress them.
- Do not submit any unnecessary files (e.g., provided or generated files).

You can submit your file(s) as many times as you want before the deadline. Only the last submission will be graded. Be sure to submit at least once before the deadline.

November 6, 2014