CPEG-323: Introduction to Computer Systems Engineering

Machine Problem #3: Cache Simulation (100 points)

You may work alone, or in groups of up to two.

Due on Dec. 15 (11:59:59pm)

In this MP, you will write be completing a program to simulate a cache for a processor. The simulator will read in a trace of memory references and determine which are cache hits and which are cache misses. We have provided a partial implementation.

Your cache simulator should simulate a cache with the following configuration:

1. 64-byte blocks
2. 2-way set-associative
3. total size: 32 KB
4. write-back

To compile your code (on the linux/unix machines) use the following command:

$ g++ -g -o cache_sim cache.c parse_input.c

The "-g" flag tells the compiler to include debugging information so that you can use a debugger like gdb. The "-o cache sim" command says to create an executable called "cache sim".

The simulator expects a number of command line arguments, as shown:

$ cache_sim <trace_file> [verbose]

- trace file is a file containing the loads and stores the simulator will run.
- verbose is an optional argument. If you include "verbose" as the fifth argument, the code will print out which accesses hit or missed line-by-line.

We provide a trace file, you are strongly encouraged to generate your own.

There are three things that you need to do complete the code:

1. implement the address to tag and address to index functions.
2. implement the find block and update lru function (see the comment in the code).
3. implement the fill cache with block function (see the comment in the code).

When the code completes, it prints out the number of accesses (i.e., the number of loads and stores), how many were hits, and how many accesses caused writebacks. In your implementation of find block and update lru and fill cache with block you need to update the accesses, hits, and writebacks counters appropriately.

Five test cases are provided for you to verify the correctness of your code. We strongly suggest that you first create a number of simple test cases, where you can predict the correct behavior of your cache to make sure the basics are working.

Submission: Please email the following files to cpeg323.udel@gmail.com.

- Modified code: cache.h and cache.c
- Partners.txt containing the names of everyone in your group.