

# Computer Systems Architecture

## Homework 4

### Cache Simulator

In this HW, you will build a simple simulator in either **C** or **C++** that reads a memory trace and simulates the action of a '**write-through with write-allocate**' cache.

The replacement policy is **LRU** (Least Recently used) when all the blocks in the the set are valid.

The configuration of the cache is as follows:

**Size:** 4096KB, **Block Size:** 16B, **Associativity:** Two-way, **Address Size:** 32 Bits.

This replacement works as follows in a two-way associative cache:

First way in a set is MRU and Second way is LRU. When the first block comes in it is stored in MRU position. When the next block comes, the old block is moved to LRU position and the new block is stored in MRU position. When the third blocks comes, block in LRU position is replaced by the block currently in MRU position and the new block occupies MRU position.

The valid bit for all the blocks is set to 0 initially. This bit changes to 1 when this block is occupied.

You don't need a dirty bit, since it is not a write-back cache.

Each trace is a series of lines, each listing a hexadecimal memory address followed by R or W to indicate a read or a write.

Example:

0041f7a0 R

13f5e2c0 R

31348900 W

At the end of the simulation, i.e. all traces have been read and simulated, print out the following stats in following manner:

**Total traces:**

**Reads:**

**Writes:**

**Cache hits:**

**Cache misses:**

**Submission:**

Submit files with following naming convention:

Executable file:

**memsim.** Make sure it works on openlab server. This executable has one command line argument i.e. the trace file. Therefore it must be able to run simply as:

`$/memsim <input_trace_file_name>`

Source file(s): All together in a zip/tar/tar.gz file.

**NOTE: Do NOT compress Executable and Source file(s) into zip, tar, etc. together.**

**Test Cases:**

1. <https://drive.google.com/open?id=19dw3BDAYZex2eYaZyA3kvNVKFnJhAu7d>
2. <https://drive.google.com/open?id=1pK3BoWTPsVKZ9oeXQDybBPwHkPkyA-DJ>