COE 1541 Introduction to Computer Architecture
Electrical and Computer Engineering

Spring 2019

Project 2: Cache Simulator

Due: 4/12/2019

1. **Introduction**
   In this project, you will build a highly parameterized cache module that can be configured into various types of cache hierarchy. Please form a team of at most two people asap, and inform the TA. You will be given an input stream of cache accesses, and output 1). total execution time in cycles, 2). hit and miss rates of all cache levels.
2. **Description of cache operations**
   You will build a 2-level cache hierarchy defined by a given setup configuration. The input to this cache hierarchy is a stream of memory accesses with the format of, "op address", e.g.:

   $$r\ 12345_{10}$$
   $$w\ 65432_{10}$$

   The caches are initially empty, and filled by the input accesses. The cache should contain all the control bits as discussed in class: valid and dirty bits. Use **LRU** as replacement policy. For memory latency, assume it is 100 more cycles than the latency of the last level cache.

   For input memory access streams, first implement a *sequential* version, meaning that the next memory access does not start until the previous access is returned (including misses). Then implement an *access-under-misses* version. In this version, while a cache is waiting for a miss to return, it can continue to serve subsequent requests. As subsequent requests may also miss the cache, there is an upper bound of how many outstanding misses this cache can handle. Implementation wise, you may use a buffer to save those outstanding misses until they return from lower level caches/memory. A cache is single-ported, meaning that it can be probed or filled one at a time. For example, when a miss returns and if the cache needs to be filled, it needs to wait till after the current request finishes probing due to a structure hazard.

3. **Configurable parameters**
   The following parameters should be configurable:
   1) Size in bytes
   2) Access latency in cycles
   3) Block size in bytes
   4) Set associativity
   5) Write policy, including write back, write through and write evict (Evict a block upon a write hit. Write miss turns into write through)

6) Allocation policy, including write allocate and non-write allocate
7) Max number of outstanding misses

4. Inputs

A stream of memory accesses with the aforementioned format

5. Outputs

1) Total completion time in cycles of the input memory stream for both versions discussed above
2) Hit rates at different levels
3) Cache status image of all layers (you can output non-zero entries only)