# Assignment 4: Computer Architecture (ECGR 4181/5181) Fall 2020

## (due 20th Nov)

**Part 1: Cache Simulation**

When writing complex programs like our simulator, it is useful to design and unit test new components in isolation before integrating them to the main codebase. In part one you will demonstrate your understanding of cache by creating a simple trace-based cache simulator that simulates the performance of a cache for the memory access trace provided in ‘addresses.txt’.

Cache Design Specs:

1. Your cache model should support the following variable design parameters:
   1. Cache size – Overall size of the cache in Bytes
   2. Cache line size – Size of a single cache line (block)
   3. Associativity
      1. None (1-Way) or Direct Mapped
      2. 2-Way
      3. 4-Way
      4. Fully
   4. Lookup time – How long it takes to search the cache for a cache line
2. Implement a pseudo-LRU replacement policy
3. Track and calculate key statistics
   1. Hit rate
   2. Compulsory (cold start) miss rate

Deliverables:

1. Calculate the hit rate and compulsory miss rates for the following simulation parameters (5 pts per simulation for 20 pts total):
   1. Simulation 1:
      1. Cache Size: 256B
      2. Cache Line Size: 32B
      3. Associativity: None
   2. Simulation 2:
      1. Cache Size: 512B
      2. Cache Line Size: 32B
      3. Associativity: None
   3. Simulation 3:
      1. Cache Size: 256B
      2. Cache Line Size: 64B
      3. Associativity: None
   4. Simulation 4:
      1. Cache Size: 256B
      2. Cache Line Size: 32B
      3. Associativity: 4-Way
2. Compare and contrast the performance of the 4 simulations in terms of (i) hit rate, (ii) miss rate, (iii) and total trace simulation times.

For calculating the total trace simulation times in (iii) above, assume that the access time of the cache in simulation ticks can be determined by the equation:  
  
 (Eq 1)  
  
and that the cost of a miss in the cache is the lookup time plus 100 ticks.

Explain how the variations in performance relate to the cache settings (4x5 pts for 20 pts total).

1. Using the method of calculating cache hit and miss access times from (2) above, design a 256B cache that will minimize the overall access time for the trace provided in ‘addresses.txt’ and report its hit rate, compulsory miss rate (20 pts).

**Part 2: Integrating Cache into the System Simulation**

Now that we have a working cache simulation, it is time to integrate it into our system simulation.



**Figure 1**

Figure 1 Design Specs:

1. See Part 2 of Assignment 3 for CPU, Memory, and Membus specs
2. Memory response time: 100 sim ticks
3. I-Cache Specs:
   1. Cache Size: 256B
   2. Cache Line Size: 32B
   3. Associativity: None
   4. Lookup Time: 3 sim ticks
4. D-Cache Specs:
   1. Cache Size: 512B
   2. Cache Line Size: 32B
   3. Associativity: None
   4. Lookup Time: 4 sim ticks

Deliverables:

1. Rerun the simulation from Part 2 of Assignment 3 (without cache) but change the memory access time from 20 simulation ticks to 100 simulation ticks. Calculate the average CPI for each CPU (10 pts).
2. Construct the system shown in Figure 1 and rerun the same multicore simulation. The cache access times should be 3 sim ticks for instruction caches and 4 ticks for data caches. Memory access time should be 100 simulation ticks. Calculate the average CPI for each CPU (10 pts).
3. Assuming that you can only afford 1.5KiB of cache in your system, and that the cache line size must be standard across all caches in the system, design a cache and memory hierarchy that improves the average CPI across both CPUs. Assume that the access times (hit and miss) for the caches follow Step 2 of Part 1 above. Provide your cache parameters, hit rates, miss rates, and average CPI per CPU. (20 pts)
4. BONUS 10 pts will be awarded to individual/group with the best performance improvement.