University of Central Florida

Department of Computer Science

CDA 5106: Fall 2020

Machine Problem 1: Cache Design, Memory Hierarchy Design

by

Parker Scott

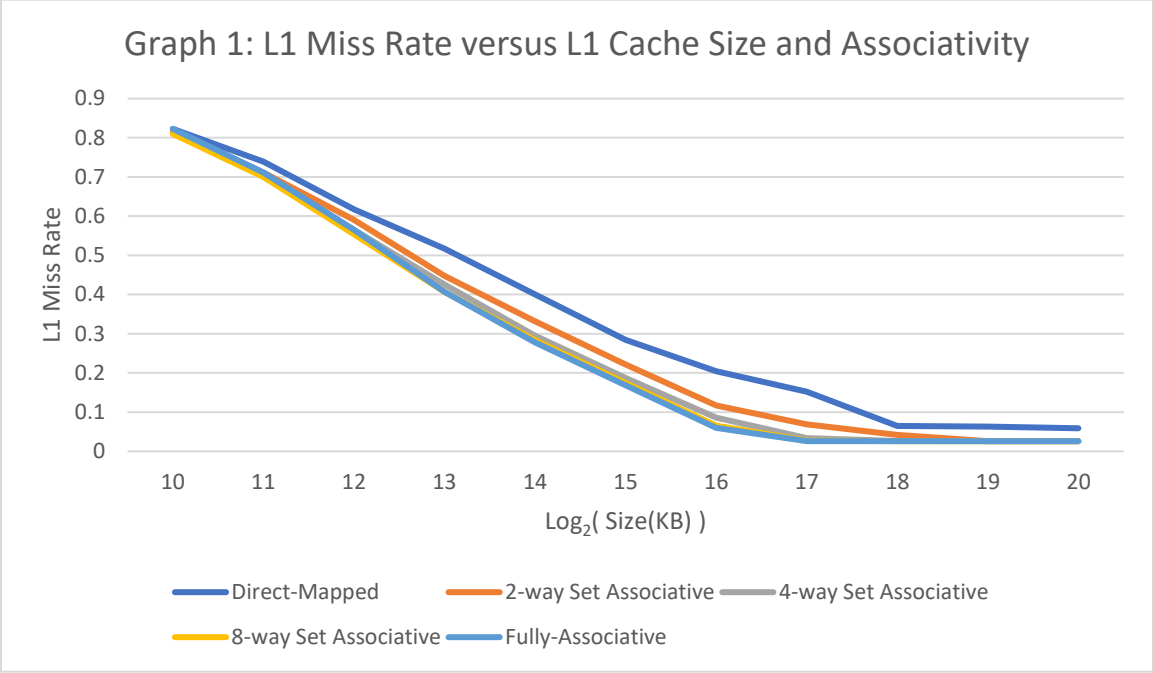
Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."

Student's electronic signature: ___Parker Scott_____
(sign by typing your name)

Machine Problem 1: Cache Design, Memory Hierarchy Design

Experiment 1

Graph 1 plotted L1 miss rate on the y-axis versus $\log_2$(size of the L1 cache) for 5 different associativities: Direct-Mapped, 2-way set associative, 4-way set associative, 8-way set associative, and Fully-associative.



Graph 1 depicts a decline in the L1 miss rate for all associativities as the size of the cache increases. The decline of the L1 miss rate hits a support level at around a cache size of 64-128 KB and doesn't decrease significantly as that size increases. For a given associativity, increasing cache size will lower the L1 miss rate until a floor at the 64-128 KB size. As well, as associativity increases for a given cache size, the L1 miss rate will decrease with 8-way associativity and fully-associative showing similar results at the lowest miss rate levels.

The compulsory miss rate is based on the number of blocks to miss on in a cold start, so therefore as the cache size increases, there will be more compulsory misses due to more blocks being cold and having to be brought into the cache. Therefore, this would be equal to:

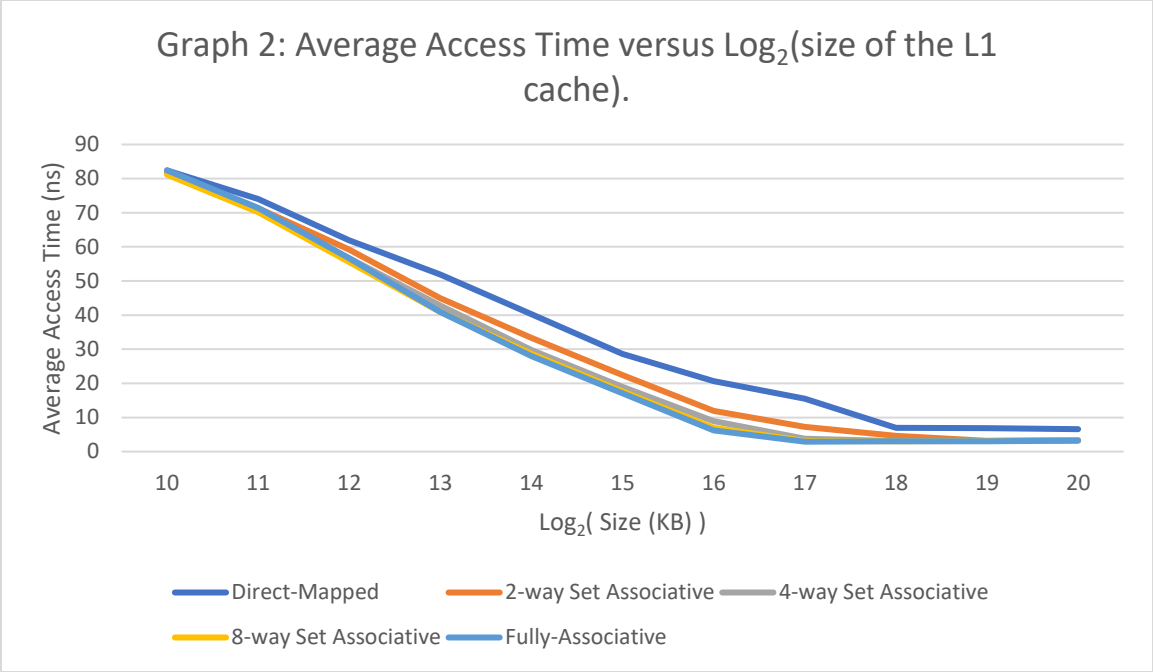
$$Compulsory\ Miss\ Rate = \frac{Num\ of\ Sets * Associativity}{Reads + Writes}$$

Therefore, for a 32 block-size cache with a size of 1024 and associativity of 4 and ran against the GCC Benchmark, we could estimate a compulsory miss rate of around .00032.

The conflict miss rates would be the comparison of a specific associativity's miss rate against a fully-associative version. Therefore, using the data from graph 1 and averaging the differences for each cache size and associativity versus the fully-associative miss rate, the conflict miss rates are .0733, .0257, .0061, -.0016, and 0 for Direct-Mapped, 2-way set associative, 4-way set associative, 8-way set associative, and Fully-associative, respectively. The 8-way set associative yielded less misses than a fully-associative; therefore, the conflict miss rate might be 0 instead of a negative number.

Experiment 2

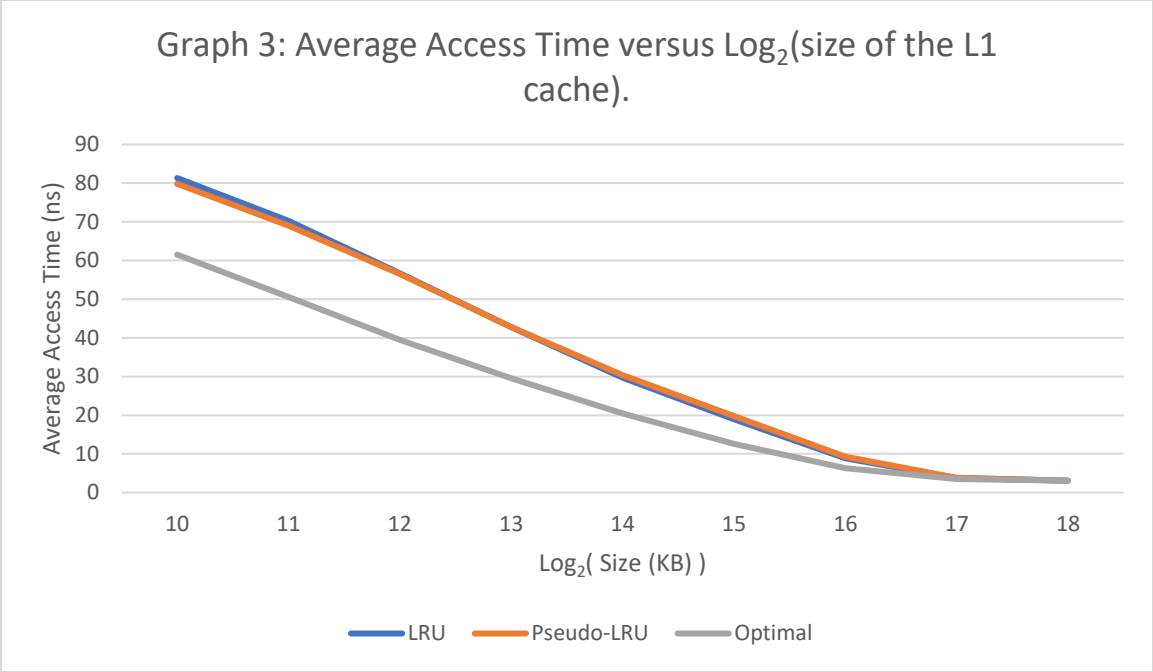
Graph 2 plotted Average Access Time versus $\log_2$(size of the L1 cache) for 5 different associativities: Direct-Mapped, 2-way set associative, 4-way set associative, 8-way set associative, and Fully-associative.



Graph 2 depicts a decrease in the Average Access Time in nanoseconds for all associativities as the size of the cache increases until around 128-256 KB where it levels off or increases as the size increases to 512 KB and 1 MB. As well, as associativity increases, the average access time decreases until about 8-way set associativity which yields similar results to fully-associative. However, the configuration which yields the lowest average access time is the fully-associative, 128 KB size cache yielding 2.904486 ns average access time. This tells us that there is an optimal average access time around cache sizes 128 KB to 256 KB for each associativity level.

Experiment 3

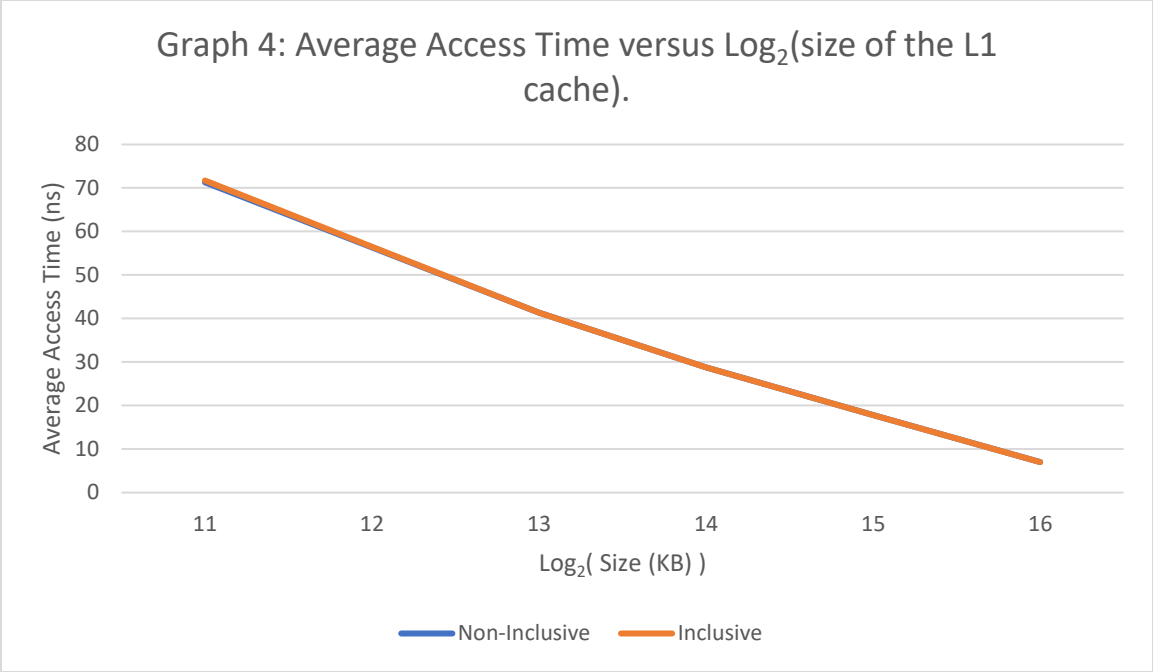
Graph 3 plotted Average Access time versus $\log_2$(size of the L1 cache) for 3 different replacement policies: LRU, Pseudo-LRU, and Optimal.



Graph 3 indicates that all replacement policies decrease in average access time as the size of the cache increases. Similar to the previous experiments, it reaches an optimal zone around 128 KB which leads to a very minimal decrease in average access time as compared to 256 KB. As expected, optimal yields the lowest average access time consistently for all size caches. As well, Pseudo-LRU yields a lower average access time than LRU for smaller cache sizes but crosses over to become more time costly than LRU with a cache size of 8 KB and higher.

Experiment 4

Graph 4 plotted Average Access time versus $\log_2$(size of the L2 cache) for inclusion and non-inclusive properties.



Graph 4 shows a decline in average access time as the size of the cache increases for both non-inclusive and inclusive properties. Non-inclusive has a slightly lower average access time for cache sizes 2-8 KB; however, they then align in time for any cache size larger than 8 KB. This shows that as the cache size increases inclusivity is less likely to affect average access time since there would be less chances of eviction in the L2 cache as the size of it increases.