# Project 1 Report

By: Jason Rodriguez
EEL 4768C Section 00382

Part A

Figure 1 shows a bar graph that shows how different cache sizes can affect the miss rate. With the exception of cache size, every other parameter is kept constant for each test (4 way associativity, LRU replacement policy, and write-back). As we can see in figure 1, larger cache reduces the miss ratio. Meaning that misses occur less frequently with larger cache sizes, assuming every other parameter stays the same. Which makes sense. Larger cache sizes means more sets and more index bits. The index bits determines which set will the data go into the cache. With more sets means more spots for each data to occupy. Since there's many more requests with XSBench than MiniFE, of course it'll have a higher miss ratio. However, we can see that the trends for both files are similar in that they are both decreasing at higher cache sizes.

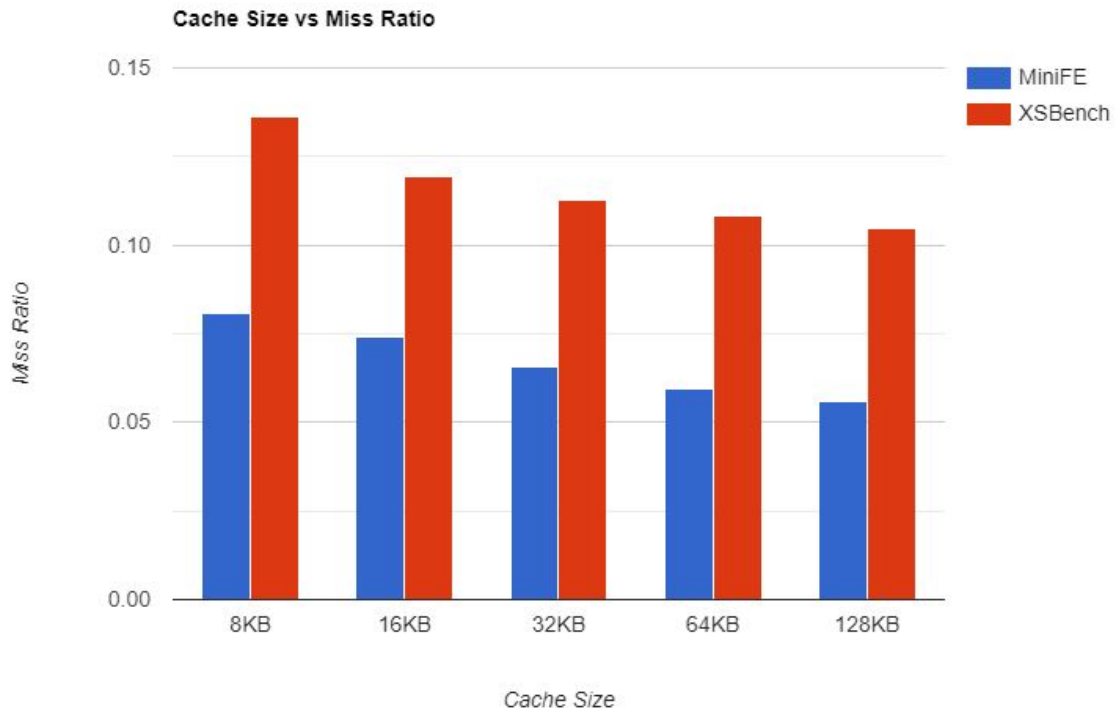


Figure 1

Part B

Figures 2 and 3 are graphs that plots memory accesses for both write-through and write-back policies respectively. Aside from the write-back policies, all other parameters are the same as the previous part. Going over the extreme changes first, write memory accesses are much higher for the write-through policy than it is for the write-back one. In fact, write memory accesses values for the write-back policy is nearly insignificant compared to the other values in the same graph. Compared to write-through, read memory accesses are about as comparable for both graphs. The

reason as to why the drastic difference in writing memory accesses is how they handle cache hits/misses. With write-back, data is only written to memory when a dirty bit is detected in a cache block. Dirty bits are added in any write request, regardless if it's a cache miss or hit. However, dirty bits are only removed when a read request is made and it's a cache miss. This means that with low miss rates, we'll have low write to memory accesses. As opposed to write through, which will write to memory whenever a cache miss occurs on a write request. An interesting property that is worth noting is that the write to memory accesses values stay the same for both MiniFE and XSBench after increasing associativity.

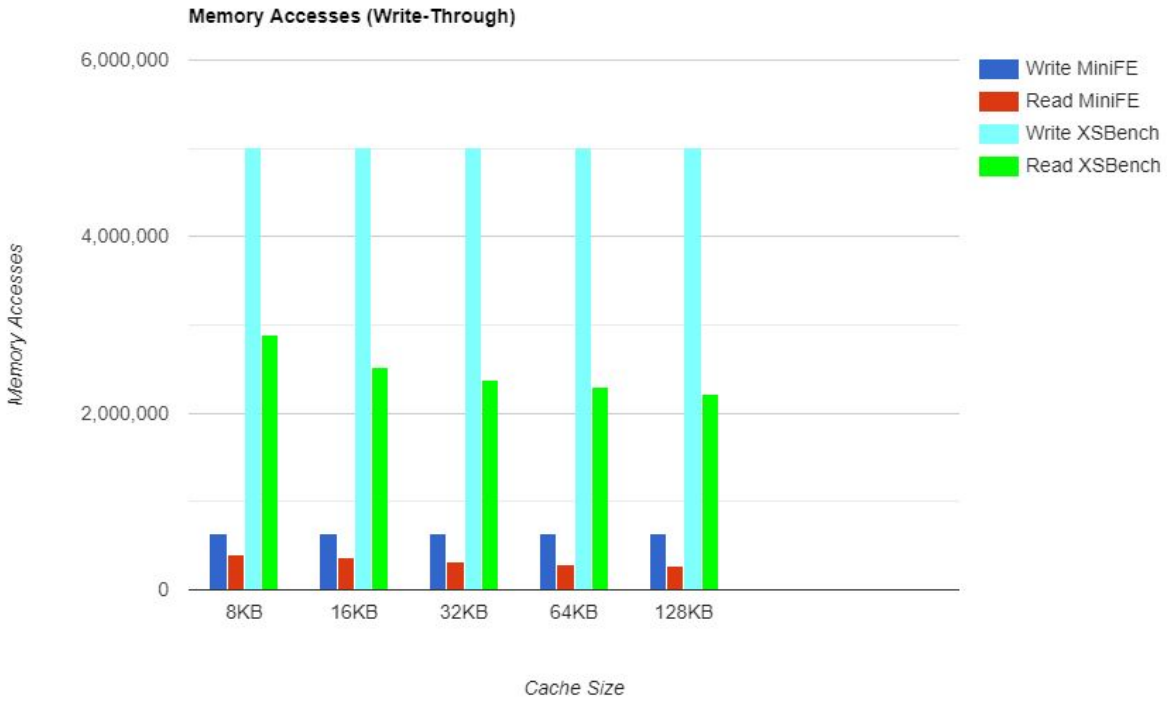


Figure 2

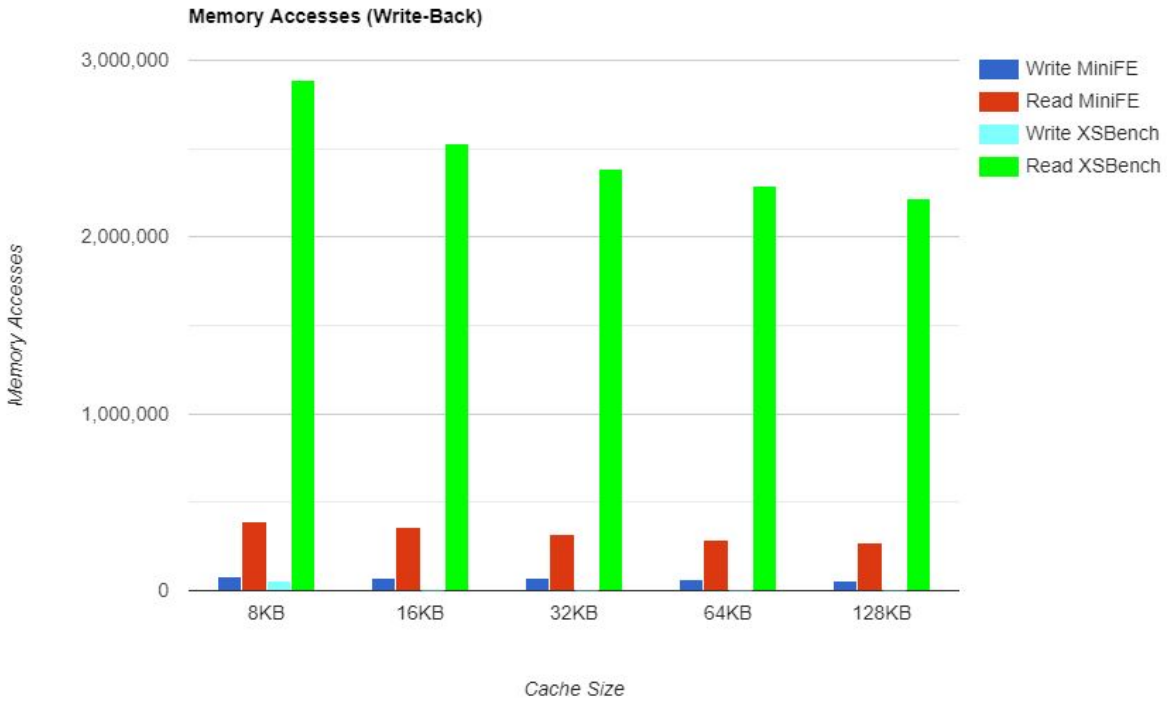


Figure 3

Part C

Figure 4 shows a bar graph that shows how different associativity values affect the miss ratio with the LRU replacement policy. Every other parameter such as cache size (32KB) and write-back policy (write-back) are kept the same throughout the test. Out of all the associativity values presented in the figure, 1-way (direct mapped) associativity has the highest miss ratio for both test files. This is because every cache entry is considered a set. Since there's 32KB and every block is 64B, there's 512 sets/entry. We are not taking advantage of spatial locality, so the comparator is only looking at one tag contained in the set, which can lead to a higher miss ratio. Beyond 2 way associativity, we can see that the miss ratio levels for both test files. We can see the diminishing returns (potentially higher miss penalty  that is happening with higher associativity. However, if one takes a closer look at the figures, we can see that after 8 way associativity, there's a slight increase for miss ratio for the MiniFE file while XSBench's miss ratio is about the same (although the actual data shows decreases that is too negligible to be noticed on the graph).

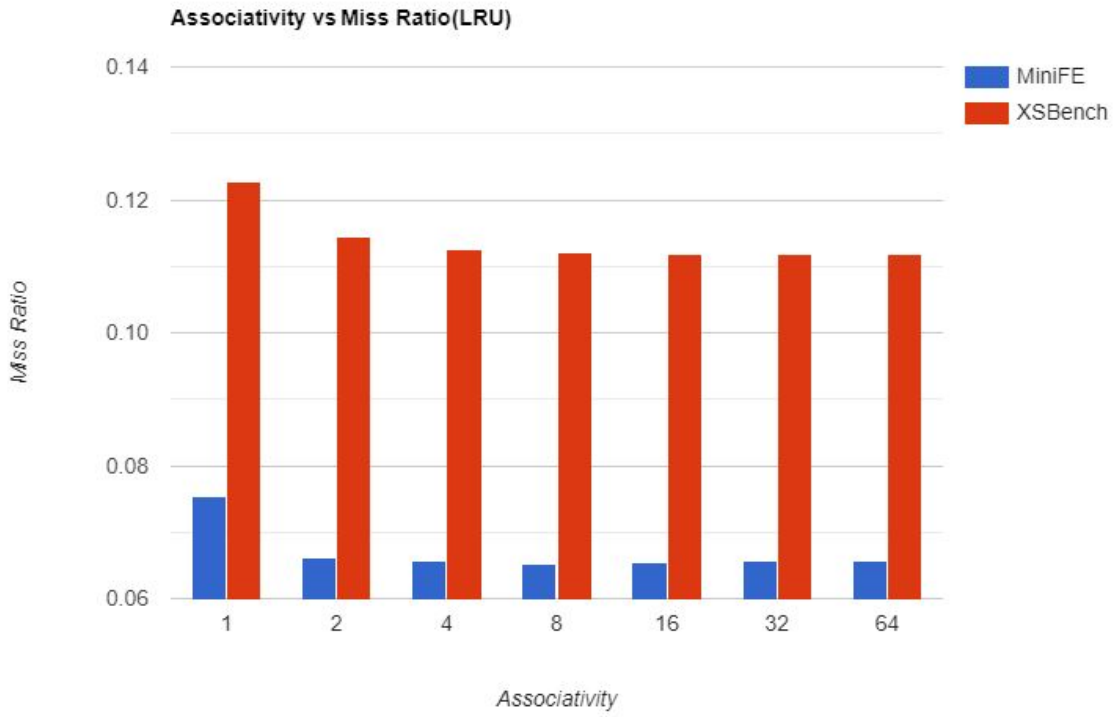


Figure 4

Part D

This part is just like the previous one, but we're using the FIFO replacement policy. We can immediately see that miss ratios for each associativity are higher than their LRU counterparts for both files. Again, we're seeing that after 1 way associativity, the miss ratio starts leveling towards some value but at a higher value compared to LRU. Also, we're seeing that the miss ratio is slightly raised in associativity higher than 8 way. In this instance, we can see that LRU is the preferred replacement policy when considering lowering the miss ratio. However, the difference between the two aren't too drastic. These trends apply to both MiniFE and XSBench, with MiniFE having lower miss ratios due to the relatively low requests.

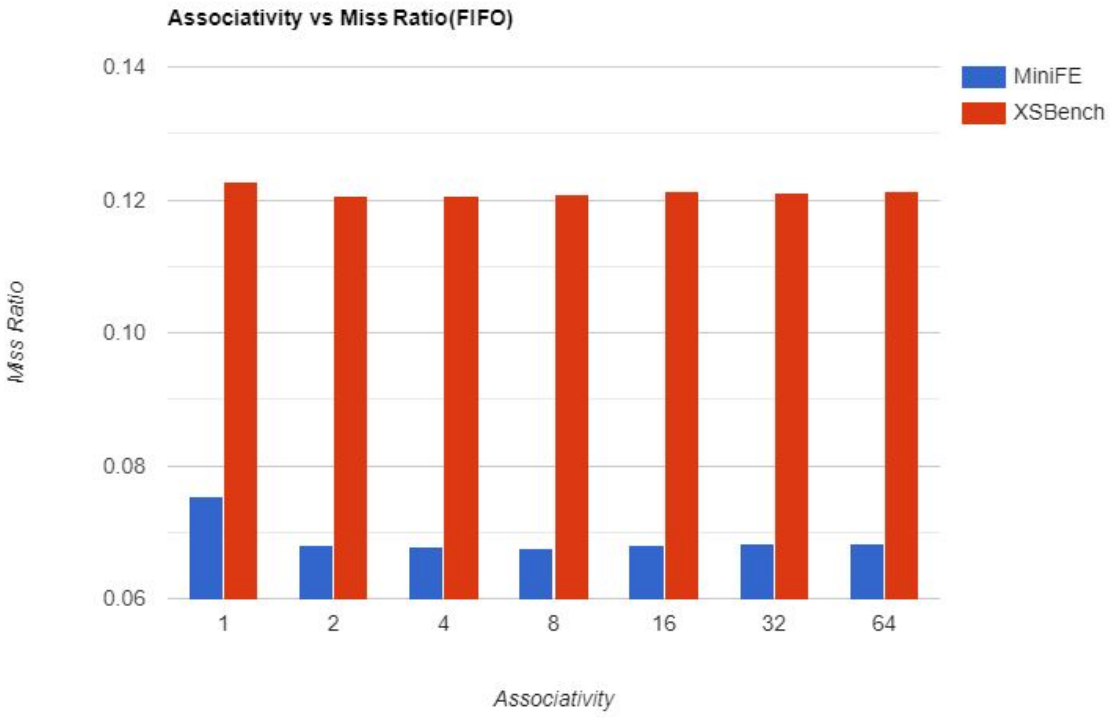
Associativity vs Miss Ratio(FIFO)
0.14
0.12
0.10
0.08
0.06
Miss Ratio
1
2
4
8
16
32
64
Associativity
MiniFE
XSBench

Figure 5