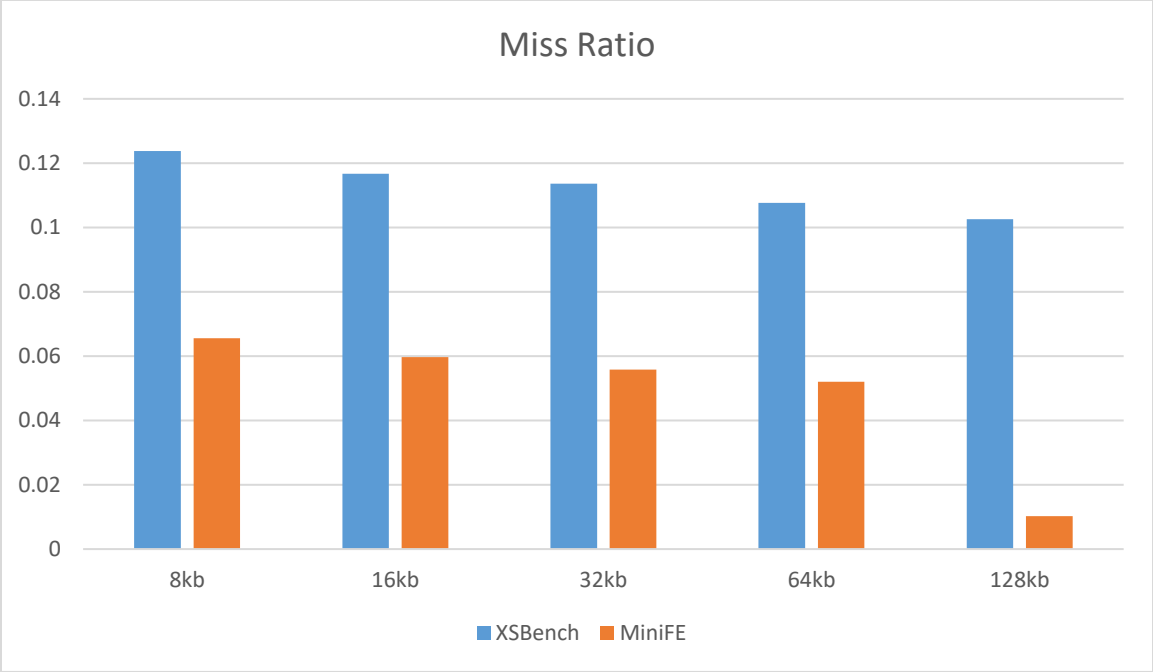
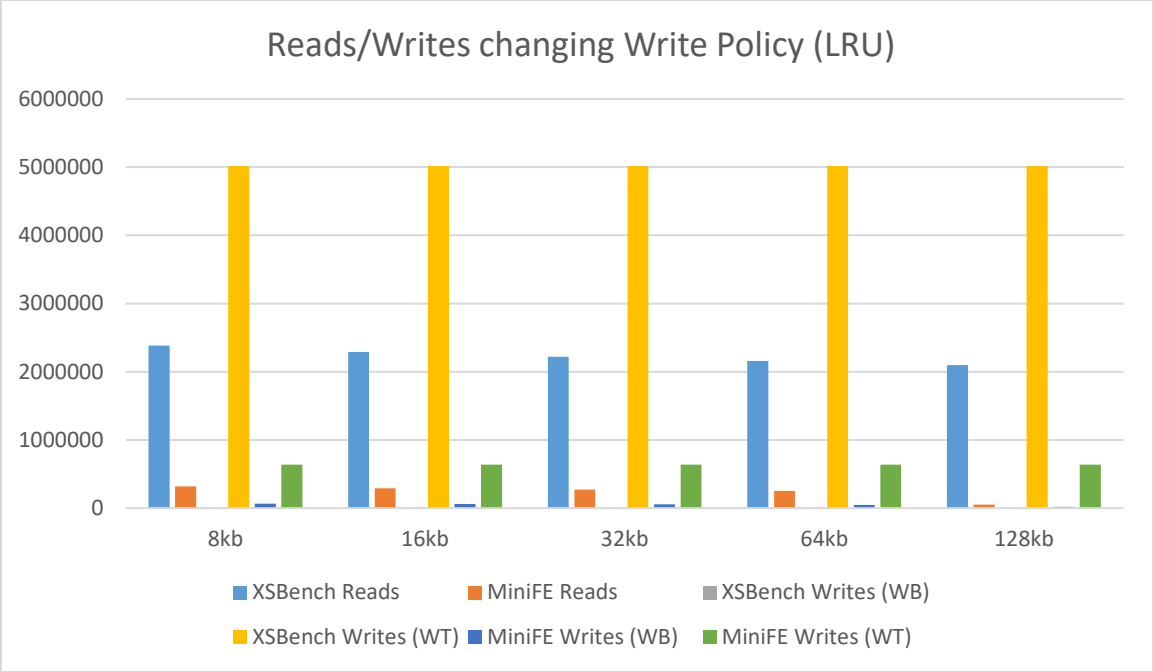
Ty Abbott – ty937650

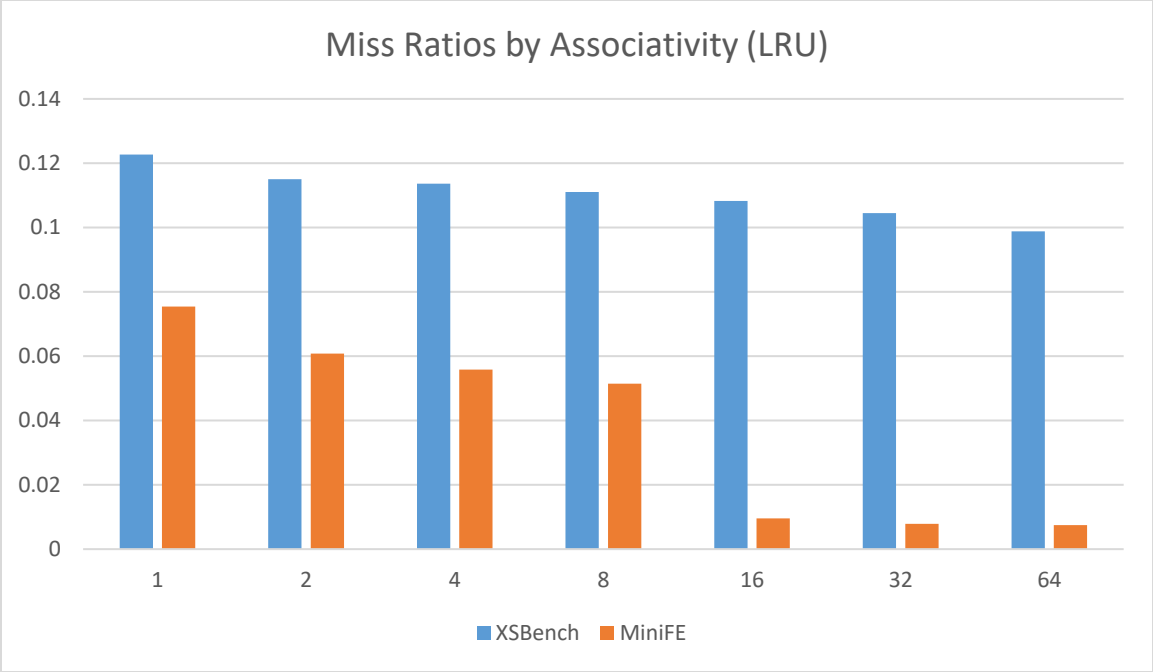
Cache Simulator Report



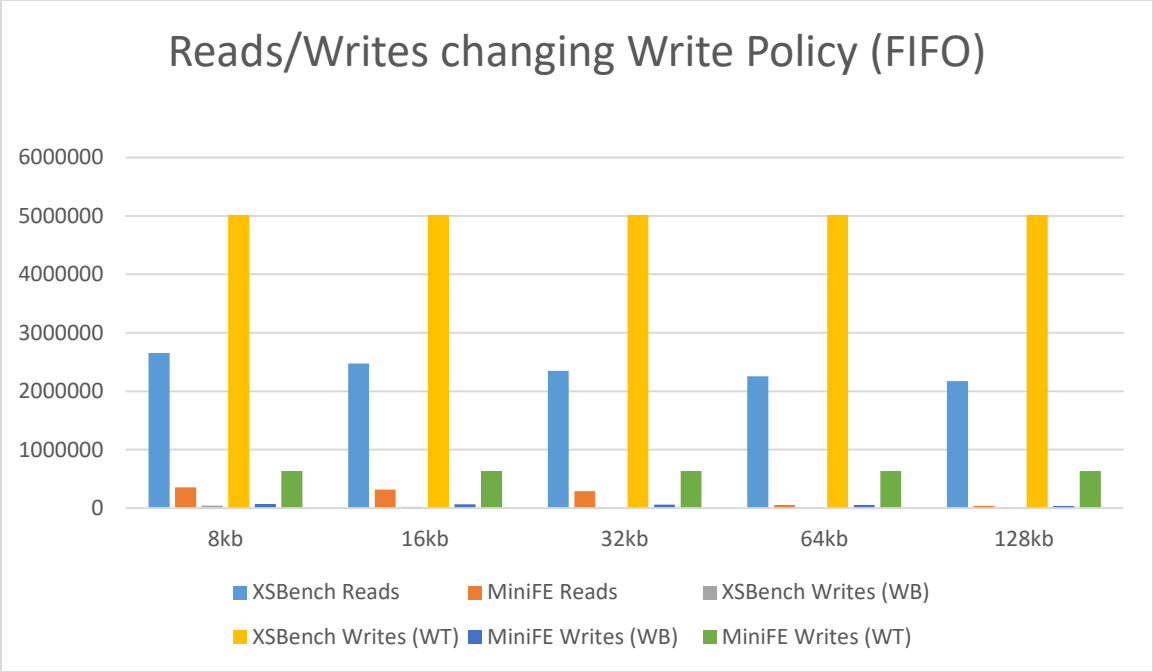
As expected changing the cache size effects the miss ratio inversely. By increasing the size of the cache miss ratio decreases. Using **XSBENCH** starting at 8kb going up to 128kb the miss ratios were as follows: .1238, .1167, .1136, .1077, .1026. These are changes of .7, .3, .6, and .5 roughly. The biggest decrease in miss ratio was between 8kb and 16kb. Whereas, the smallest decrease in miss ratio was between 16kb and 32kb. Using **MiniFE** starting at 8kb going up to 128kb the miss ratios were as follows: .0656, .0597, .0558, .0520, .0102. These are changes of .006, .004, .004, and .04 roughly. The biggest change was between 64kb and 128kb by far. A bigger cache allows for more addresses to be stored, so this is expected.



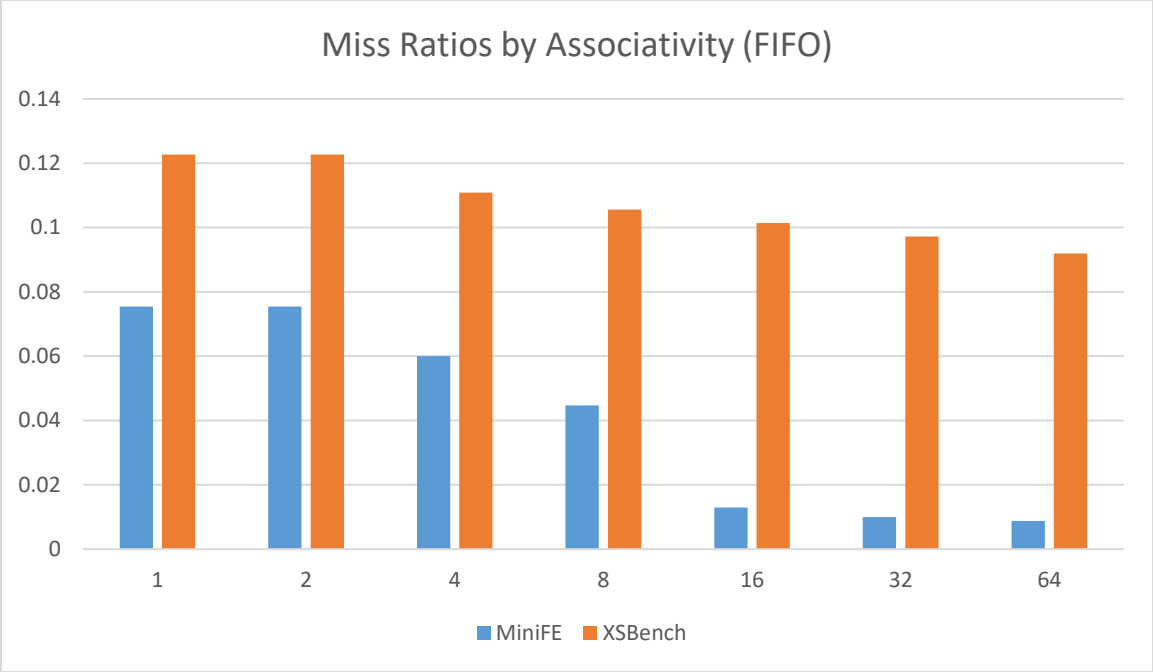
The Memory Reads are unchanged by the Write policy. The Memory Writes are changed drastically however. With Write Through, there is a memory write every time there is a write operation. So no matter the cache size there will always be a fixed number of memory writes. With Write back we see the number of memory writes drops by a lot. The number is so small that it can't be seen on the chart but at 8kb it is about 600, and quickly drops to about 30 for 128kb. The biggest change in performance was 16kb for memory writes.



As expected changing the associativity inversely effects the miss ratio. As I increased the associativity the miss ratio decreased. Using **XSBENCH** I started with an associativity of 1, then increased it starting with 2, 4, 8, 16, 32, and 64. The miss ratios were as follows: .1227, .1150, .1136, .1111, .1083, .1045, .0988. The biggest decrease was between associativity of 1 and 2. For **MiniFE** the change from 1 to 8 was steady but not very substantial. The biggest change was between 8 and 16.



For Reads and Writes with the 2 different write policies the data trends are almost identical between FIFO and LRU. The number of reads and writes is greater across the board for every category except for Write Through, which is the same and is unaffected by any cache technique. The number of memory reads for XSBench are about 10% higher for every cache size across the board for FIFO. For MiniFE this varied more, but the reads were about 30% higher for the cache sizes for FIFO. The number of Writes using writeback was more than double for 8kb and 16kb, whereas the number was many times bigger once you got to 32kb+, because for LRU the number was so small. FIFO is an easier implementation, but across both reads and writes there is a significant performance decrease.



The starting numbers are exactly the same, because if the associativity is 1, it doesn't matter what the replacement policy is. For the rest of the associativities, the miss ratios are almost identical with most Miss ratios being lower for LRU, but in a couple of cases the miss ratio is actually better for FIFO. Like Associativity 8 for MiniFE, and Associativities 8 and 16 for XSBench. All in all, LRU is more consistent than FIFO. With FIFO the miss ratio tends to vary more between associativities.

All in all I would choose a 128kb Cache, with LRU, WB, and an associativity of 16.