Design Experiment

CS 2200
Mani Japra – 902958199

- Astar.trace

For this trace, the cache design that best minimizes the average access time is one in which the the cache size is of 15 (C = 15), associativity is of 5 (B = 5), and the block size is of 3 (S = 3). This is the case because of the 501,468 accesses, there are a very small number of read and write misses in this design. With that being said, the cost of accessing memory is reduced because of this fact and it in turn reduces the overall average access time of the cache. Furthermore, with a miss rate of only 0.108274, this cache design allows the trace to retain an average access time of 12.827411 ns. As opposed to other designs with a smaller cache size or a smaller block size, this cache design allows the AAT to remain significantly lower than the other combinations.

- Bzip2.trace

In the case of this trace, the cache design that best minimizes the average access time is one in which the cache size is of 15 (C = 15), associativity is of 5 (B = 5), and the block size is of 3 (S = 3). This cache design allows the average access time to be about a whopping 2.307614 ns. Comparing this design to the direct mapped cache, it retains an incredible 3 times faster average access time. The only other cache design that has a sub 3.5 ns average access time is the 4 way associative cache. In this particular design, the cache size and the associativity are a bit smaller but the block size is bigger. The only problem is that this design has a much larger number of write-backs which wouldn't be a very good design.

- Mcf.trace

For the Mcf.trace, the cache design that best minimizes the average access time is one in which the associativity is as high as possible. By applying various combinations as well as looking at the fully associative, 4 way associative, and direct mapped cache, it seems as if the Mcf.trace works with a smaller average access time as long as the associativity is high. By changing the cache size and block size thereafter, it showed no significant change in AAT for the Mcf.trace thus concluding that as long as the S is as large as possible, the AAT will be the smallest (in nanoseconds).

- Perlbench.trace

For the Perlbench.trace, the cache design that best minimizes the average access time is one in which the structure is has a large cache size, with a large associativity, and a rather small block size. Based on the simulator, this cache design allows the trace to have the fewest write backs as well as the fewest misses in relation to the amount of accesses. With a miss rate of 0.043233, this cache design boasts an average access time of 6.323261 nanoseconds which is the smallest AAT I could render through the simulator.