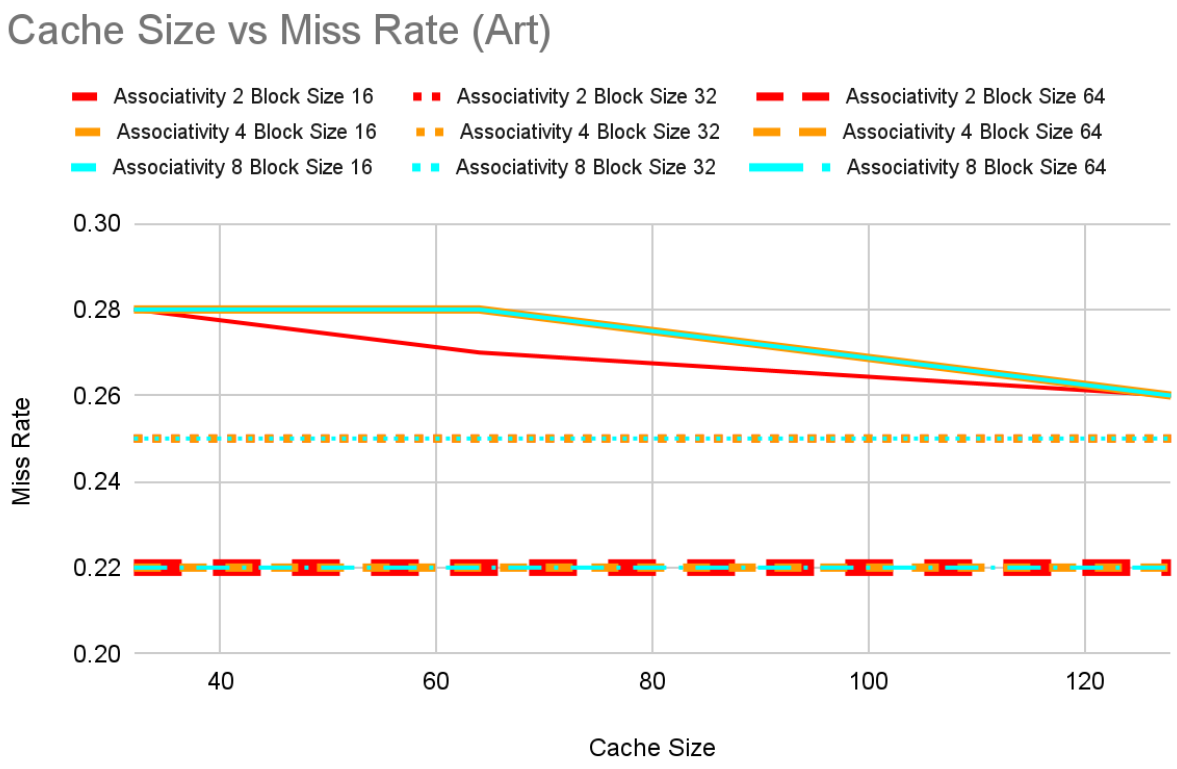
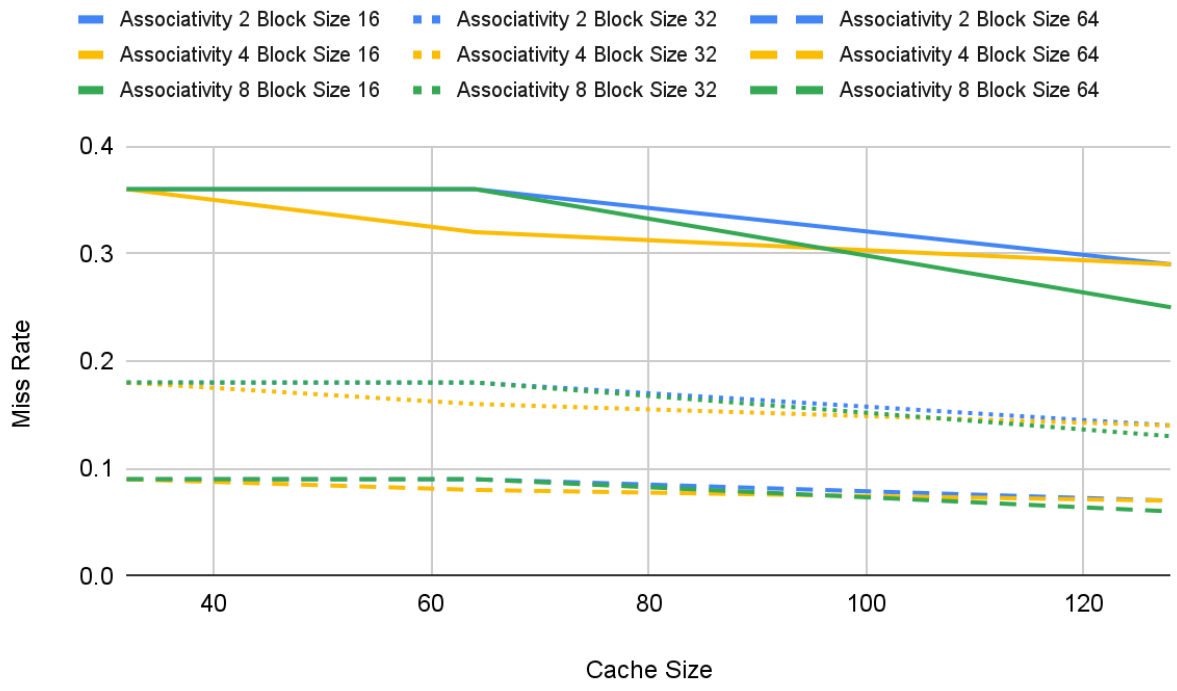
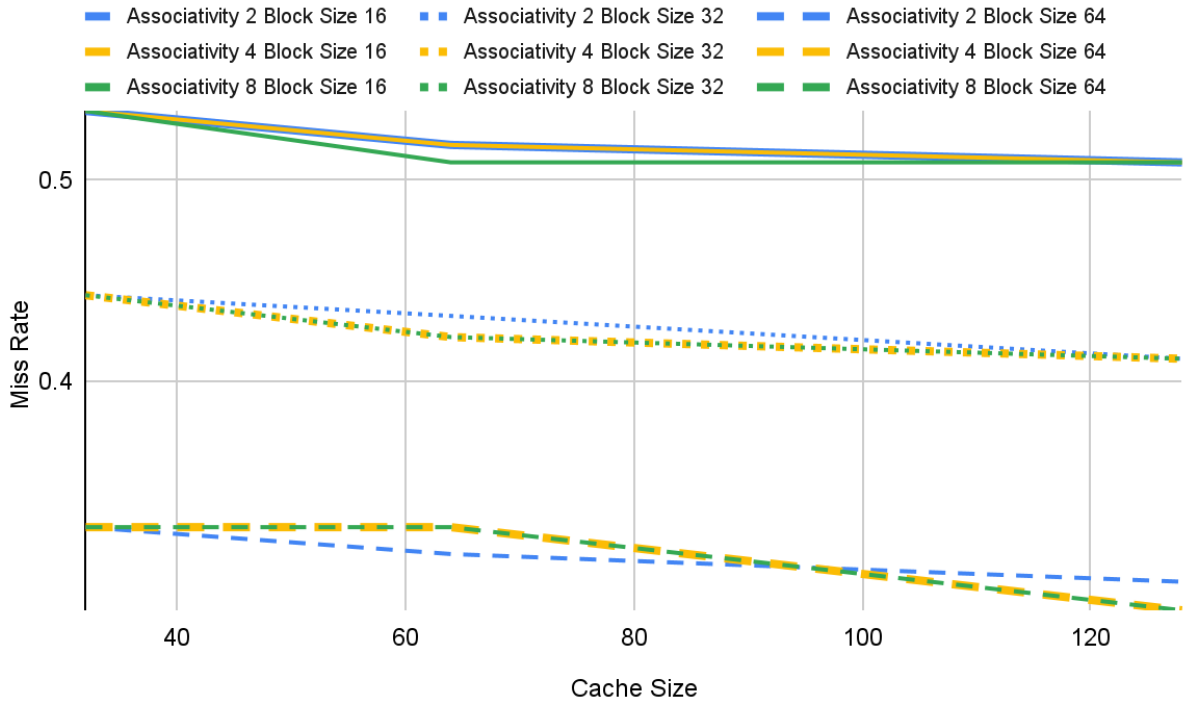
Michael Starks



Cache Size vs Miss Rate (Swim)



Cache Size vs Miss Rate (mcf)



1) From the data gathered we see that the Total CPI strictly increases with the miss rate. Since hits do not have a penalty, the only way to increase the total CPI is through misses. This is corroborated by the lower Total CPI seen with lower miss rates.
2) The lowest miss rate seems to correlate with the lowest execution time suggesting that the increase in cycle time needed to ensure a lower miss rate is not more detrimental to performance than a cache miss.
3) A lower number of clocks per instruction decreases the execution time. The number of cycles has a higher relative impact on execution time than marginal cycle time.
4) The miss rate of each trace is relative to the size. The increase in miss rate means the change in execution time will not be strictly linear, but not different enough to suggest anything other than an increase in trace size means more discrete memory units will be accessed increasing the miss rate.
5) The best-case scenario seems to be a cache size of 128 an associativity of 2 and a block size of 64. These parameters have the lowest miss rate and belong to the group of lower execution times for their respective traces. Compared to the default cases, these run times are significantly faster.
    a) Art: 7184006.50 (ps)
    b) Mcf: 35194872.00 (ps)
    c) Swim: 3138191.75 (ps)
6) To have the highest clock rate, use a cache size of 128 a block size of 2, and a block size of 64. These parameters have the lowest execution time and the highest clock rate. From the data that I gathered these scenarios are the same.
    a) Art:
        i) Execution Time (ps): 7184006
    b) Mcf:
        i) Execution Time (ps): 35194872
    c) Swim:
        i) Execution Time (ps): 3138191.25