University of Central Florida

Department of Computer Science

CDA 5106: Spring 2020

Machine Problem 1: Cache Design, Memory Hierarchy Design

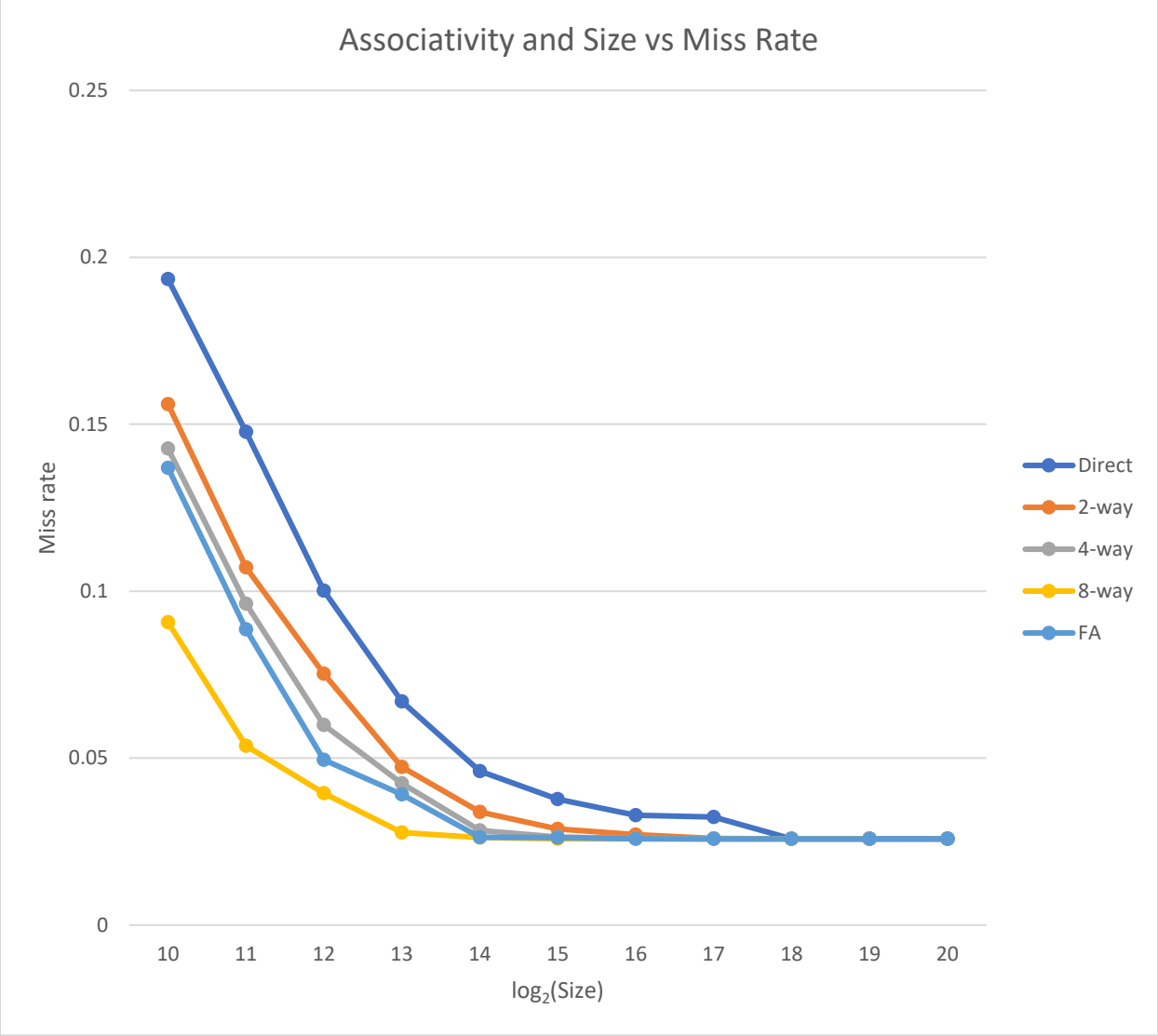
by

William Nicholas Chitty

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

Student's electronic signature: William Nicholas Chitty
(sign by typing your name)

Graph 1



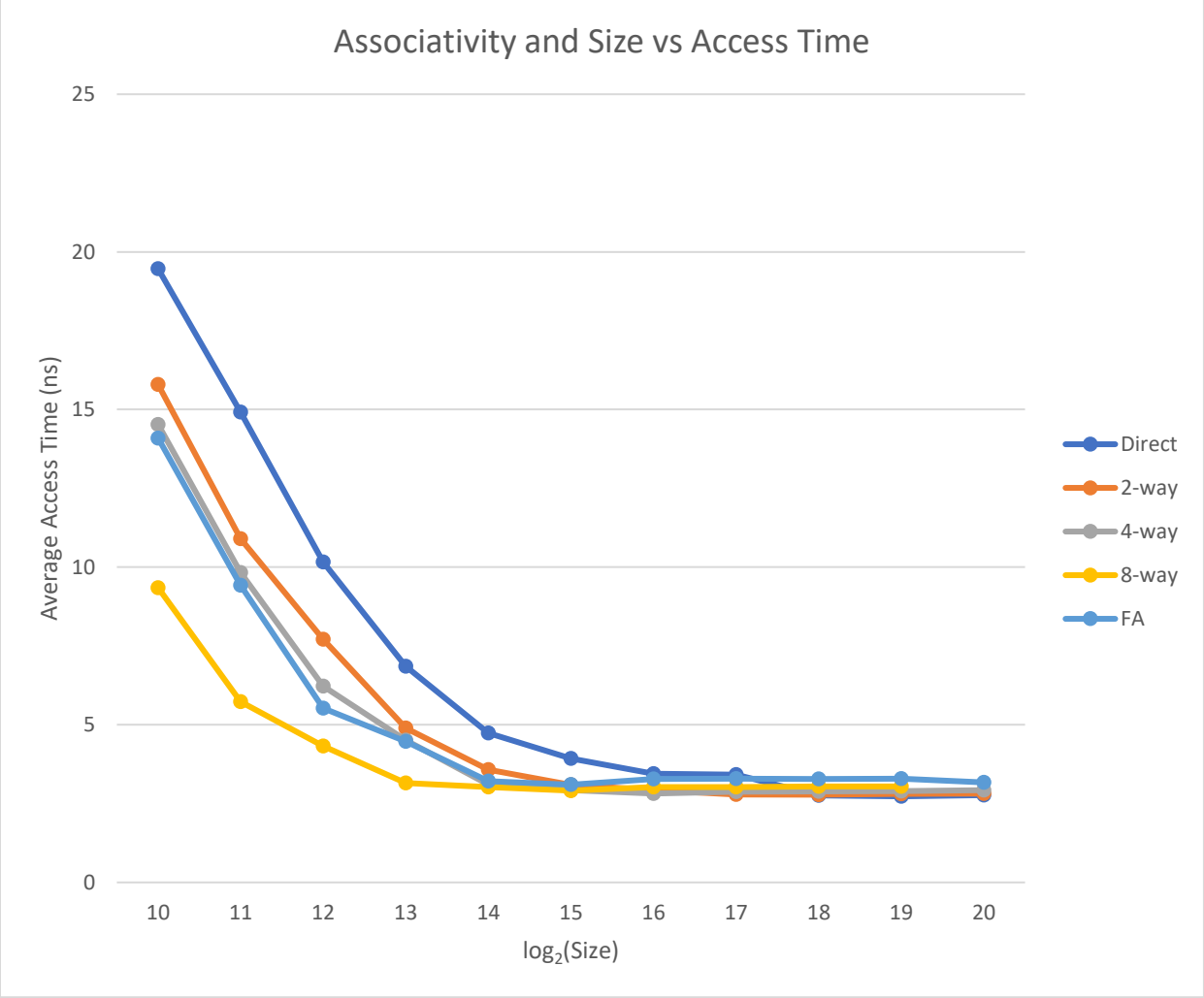
Discussion

For any associativity, increasing the size of the cache leads to a lower miss rate which is expected, there is more room in the cache for new each new memory block to be put for later. For any given size, increasing the associativity also decreases miss rate except for the fully associative case. The fully associative case starts higher than 8-way associativity but still reaches the minimum as cache size increases.

The compulsory miss rate is due to blocks that must occur as they are the first references to the block, based on the graph this would appear to be around .025.

The conflict miss rate is due to there not being enough space in the cache and thus two blocks get mapped to the same set and block as each other. For direct associativity this appears to be around .17 (the value for the smallest cache size minus the estimated compulsory miss rate). 2-way associativity has a conflict miss rate just over .13. Full associative and 4-way associativity have a conflict mis rate around .11. 8-way associative has a conflict miss rate around .06.

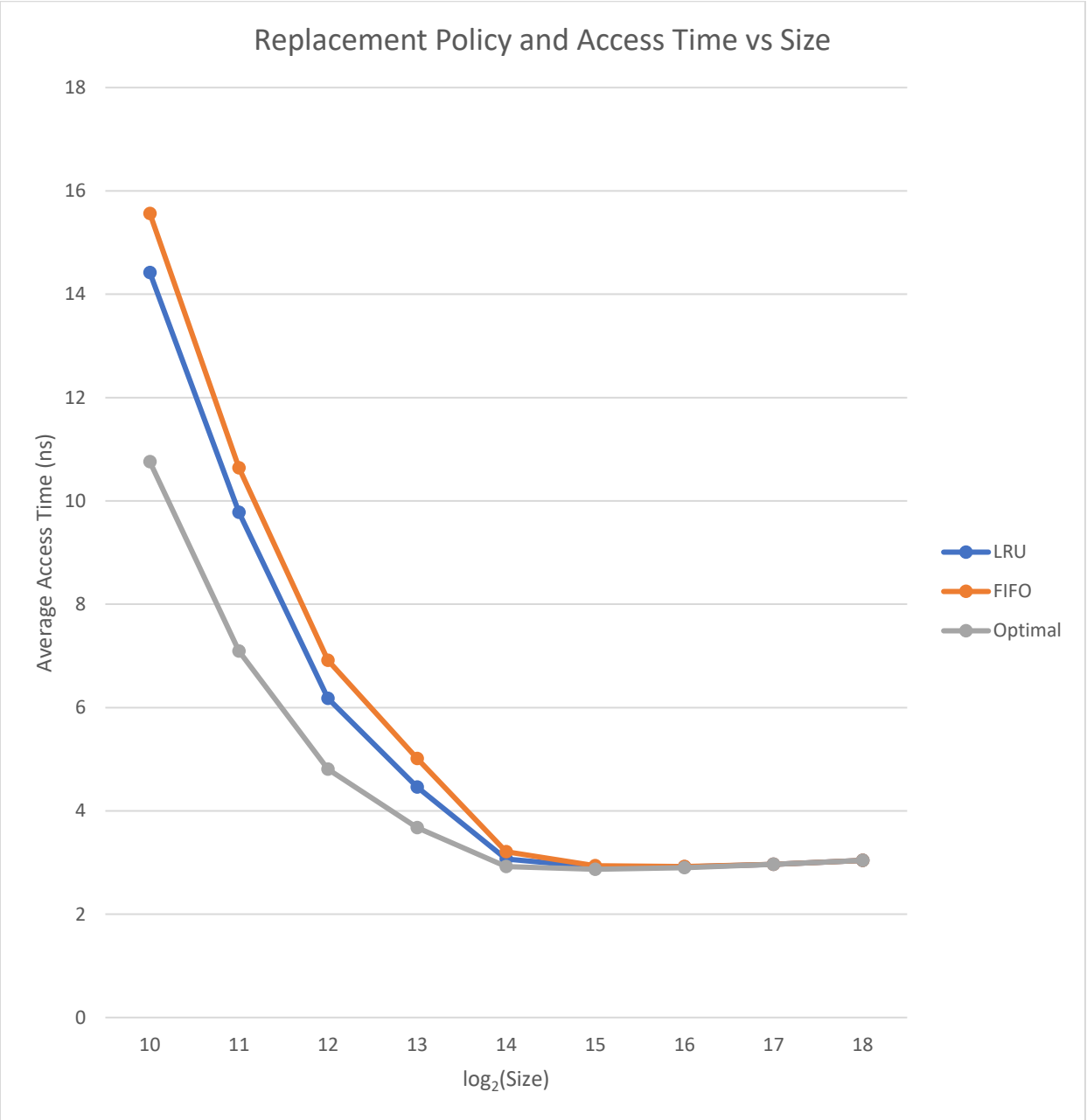
Graph 2



Discussion

The graph shows almost a halving in access time in each doubling of size which is also expected as there is room in the cache for double the blocks meaning there should be expected to be half as many misses. The best configuration based on the graph appears to be direct mapped at around 256kB in size. An educated guess as to why this is because this gives the direct-map cache enough space for every memory block and has a lower hit time than the other associativities which makes sense as there is no need to "search" through the blocks of a set for the appropriate tag.

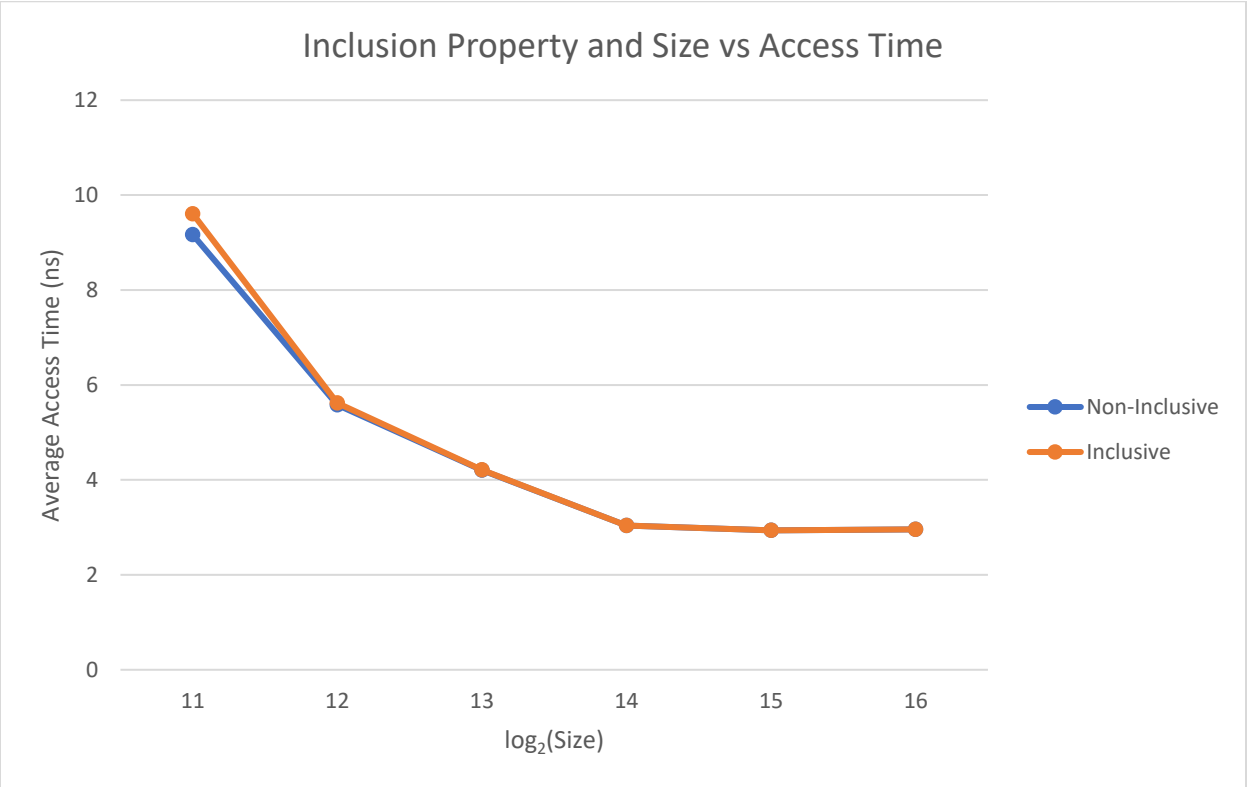
Graph 3



Discussion

The graph falls drastically with increases in cache size up to about 16kB, at this point it looks like the cache is large enough where the only misses are compulsory. Optimal, as expected, performs better than the other cache types throughout the entire graph. FIFO performs the worst as the heuristic for replacing the first in is typically bad on the premise that we might have a lot of blocks of data to read or write from and just because it was in first does not mean it will not get used the soonest. Based on the slight rise in AAT after 16kB, the optimal strategy with 16kB seems to be the best configuration.

Graph 4



Discussion

Both of these lines look very similar, as with the other graphs, they both decline exponentially. Consulting the raw data, an inclusive cache with an AAT of 2.93ns at 32kB performed better by .01ns than non-inclusive so therefore this is the better configuration.