**Cache Simulation**

My simulation is built using C++ and uses an unordered map of strings of another unordered map of strings of a linked list queue data structure to store the tags. The first unordered map contains keys to the set number that links to a map of tags that links to a node representing the position in the linked list queue. I used unordered maps to make it easier to find nodes in the middle of the queue to rearrange them when a hit occurred for the LRU strategy. To implement the FIFO strategy the linked list operated as normal and did not rearrange on a hit thus keeping the order the same. To implement the LRU strategy the node was searched in the map then removed to be readded at the top of the queue to preserve the order of least recently used.

My parameters were the number of sets per cache, number of blocks in a set, number of bytes in a block, and strategy for replacement. I used these parameters as they are all the parameters needed to calculate the hit rate of all three cache types and replacement types. The first parameter is used to calculate the number of set bits to be allocated for set-associative and direct mapping. The second parameter is used to calculate the max size of each set of blocks when filling the cache. The third is used to calculate the offset bits of the address. Finally, the last parameter lets the user choose what replacement strategy they want to use.

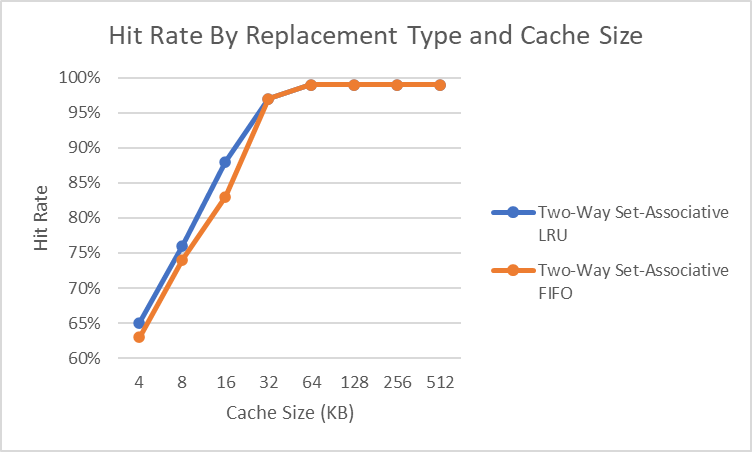
Results:

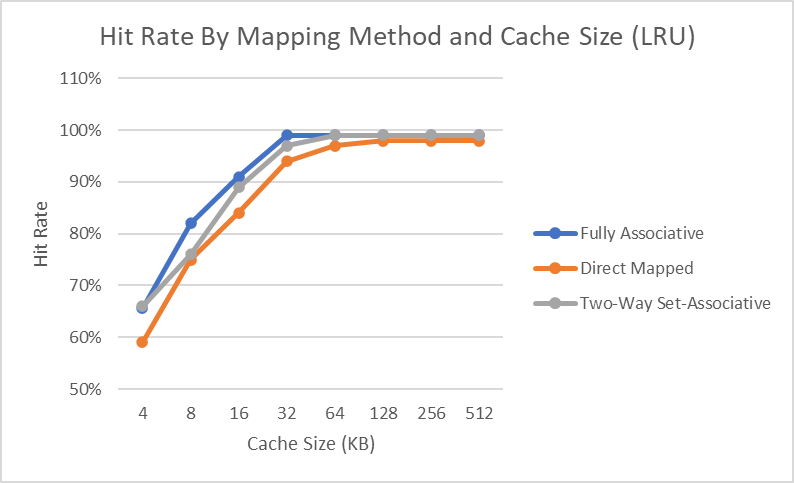
Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated





Conclusions:

My results have shown that as you increase the cache size the hit rate increases as well.

It also shows the best hit rate in descending order for cache design is fully associative and then set-associative then direct mapped.

Finally, my results show that the hit rates for both fully associative and set-associative increase when using the LRU strategy over the FIFO strategy.

Additional insights:

The fully associative design starts off the same hit rate as set-associative for the number of sets started at one and gradually increased and because set-associative is the same as fully associative when there is only one set it makes sense that they start the same in the last plot.

The hit rates never hit 100% just 99% which makes sense it is impossible to have 100% as the first access is always a cold miss.