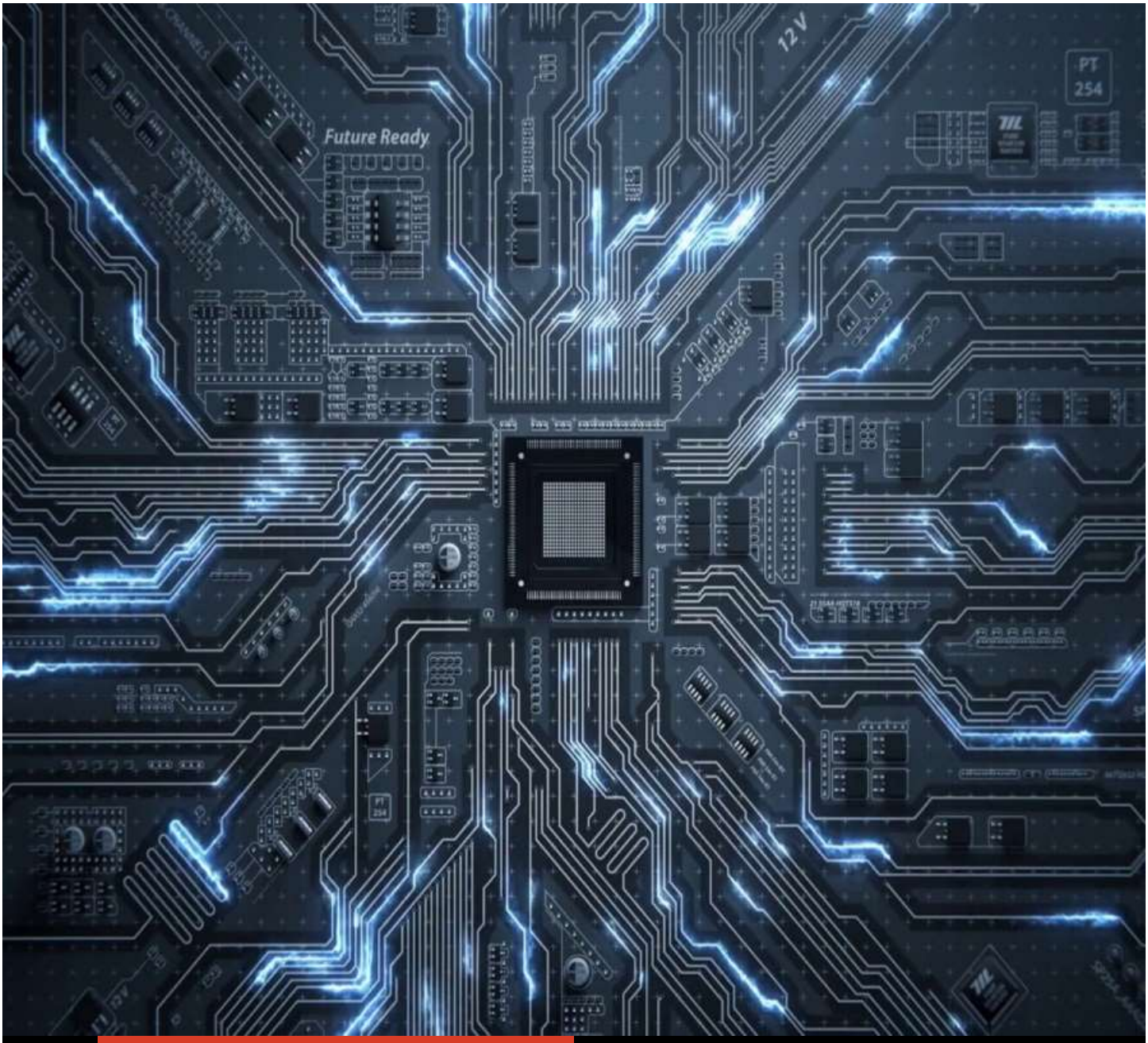


CACHE SIMULATION

AN ANALYSIS PAPER

KARL MATHEW RAMOS 12/4/2022 CDA3101

INTRODUCTION

In this assignment, students were tasked in making their own cache simulation through utilizing a coding language of their choice. The cache simulation would read memory addresses from a file, determine if it is a hit or a miss, and evaluate the hit rate at the end of the simulation. They would have to create tests for three cache types with one of the two corresponding replacement strategies. The three cache types are direct Mapped, N-way associative, and fully associative. The two replacements strategies are least recently used (LRU) and first in, first out (FIFO). In total there should be at least 7 tests which specifically are:

1. *Direct Map (not required for a replacement strategy)*
2. *Fully Associative (FIFO)*
3. *Fully Associative (LRU)*
4. *N-way Set Associative (FIFO)*
5. *N-way Set Associative (LRU)*
6. *A different N-way Set Associative (FIFO)*
7. *A different N-way Set Associative (LRU)*

Students would then have to analyze performance of the hit rate when they test different cache byte sizes and different cache block sizes on the different tests.

I chose to code the simulation in C++ coding language.

DESCRIPTION OF TESTS

The Cache Size, Block Size, and Set Number Associativity

In these tests, we were encouraged to test different values for the cache size, block size, and set number associativity. For my testing, I chose a standard block size of 64 for all my tests as the controlled variable to provide comparisons with different cache sizes. The cache sizes I chose were 512,1024,2048,4096, and 8192 because the number 512 was our given starting value and I multiplied by 2 to get the rest of the values. The set numbers I chose for the n-way associativity are 2 and 4 because I get an error if I try to do a 6 or 8. I chose the gcc.trace file for the tests.

Table of Parameter values for Each Test

In the following diagram, I have provided the parameters in a table for each test.

Block Size	Byte Size	Cache Method
64	512	Direct Map
64	512	Fully Associative FIFO
64	512	Fully Associative LRU
64	512	2-Way Set Associative FIFO
64	512	2-Way Set Associative LRU
64	512	4-Way Set Associative FIFO
64	512	4-Way Set Associative LRU

Block Size	Byte Size	Cache Method
64	1024	Direct Map
64	1024	Fully Associative FIFO
64	1024	Fully Associative LRU
64	1024	2-Way Set Associative FIFO
64	1024	2-Way Set Associative LRU
64	1024	4-Way Set Associative FIFO
64	1024	4-Way Set Associative LRU

Block Size	Byte Size	Cache Method
64	2048	Direct Map
64	2048	Fully Associative FIFO
64	2048	Fully Associative LRU
64	2048	2-Way Set Associative FIFO
64	2048	2-Way Set Associative LRU
64	2048	4-Way Set Associative FIFO
64	2048	4-Way Set Associative LRU

Block Size	Byte Size	Cache Method
64	4096	Direct Map
64	4096	Fully Associative FIFO
64	4096	Fully Associative LRU
64	4096	2-Way Set Associative FIFO
64	4096	2-Way Set Associative LRU
64	4096	4-Way Set Associative FIFO
64	4096	4-Way Set Associative LRU

Block Size	Byte Size	Cache Method
64	8192	Direct Map
64	8192	Fully Associative FIFO
64	8192	Fully Associative LRU
64	8192	2-Way Set Associative FIFO
64	8192	2-Way Set Associative LRU
64	8192	4-Way Set Associative FIFO
64	8192	4-Way Set Associative LRU

RESULTS

In the following diagram, I have provided tables of the hit rates to their corresponding test with different cache sizes. The block size is 64 for all tests as mentioned before.

Size	Direct Map Hit Rates
512	78.5403
1024	83.1627
2048	88.9568
4096	94.8028
8192	96.3736

Size	Full Associative FIFO Hit Rates
512	84.7637
1024	92.1015
2048	95.7439
4096	97.8722
8192	98.7075

Size	Full Associative LRU Hit Rates
512	85.0598
1024	92.603
2048	96.2729
4096	98.3416
8192	98.9703

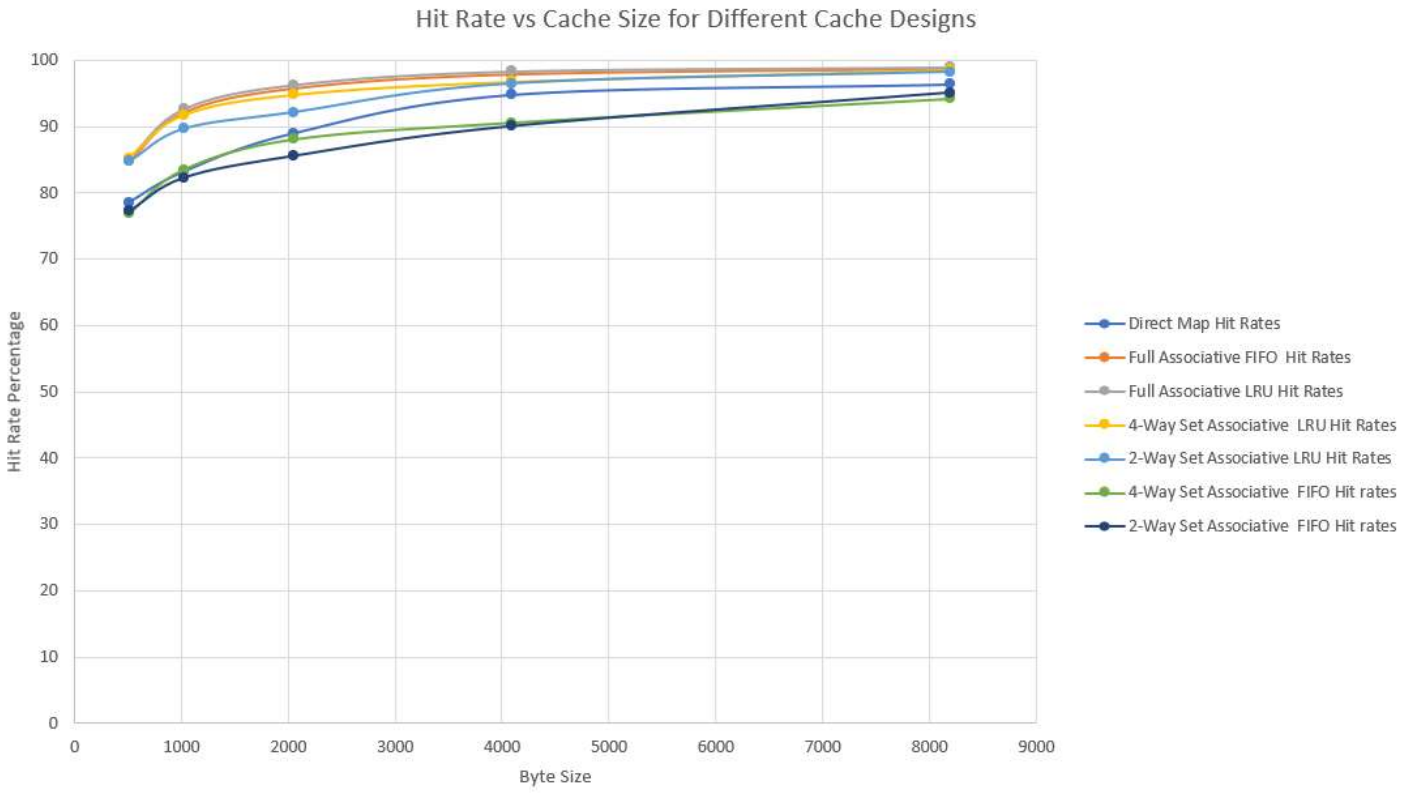
Size	4-Way Set Associative LRU Hit Rates
512	85.2411
1024	91.7283
2048	94.791
4096	96.7764
8192	98.5797

Size	2-Way Set Associative LRU Hit Rates
512	84.7215
1024	89.669
2048	92.16
4096	96.4994
8192	98.2603

Size	4-Way Set Associative FIFO Hit rates
512	76.8829
1024	83.5244
2048	88.0746
4096	90.6109
8192	94.204

Size	2-Way Set Associative FIFO Hit rates
512	77.3226
1024	82.3318
2048	85.5891
4096	90.055
8192	95.0596

Next is a comparison plot graph with all the tests. The x variable is the cache size while the y variable is the hit rate. Each cache design was colored and labelled as stated on the right of the diagram.



CONCLUSIONS

In general, direct mapped should have the lowest performance in comparison to the rest. However, in the results, the hit rates are not the lowest. The direct mapped test seems to yield results between fully associative and n-way associative tests. This creates a question of whether the implementation was incorrect.

The fully associative tests yield the highest hit rates in comparison to the other tests. This is to be expected because only the tag number is being checked, unlike the other tests where there is one more attribute to be checked to have access to the tag number. However, this comes at the cost of the process being expensive.

In general, n-way set associative should be in between direct mapped and fully associative because n-way associative is a combination of those two. When the number in the set associative gets higher, the higher the hit rate will become. However, the hit rates in the n-way associative tests seem to be all over the place. Only 4-way set associative LRU and 2-way set associative LRU were able to meet the expectation. Meanwhile, the FIFO methods for 2-way and 4-way were located below the direct map method. This is rather unusual especially since their hit rates are much lower than direct map hit rates and are not grouped together one after the other.

In general, the different replacement strategies are not distinct where one is better than the other. Usually, they are close to each other when it comes to hit rates. In the results, this is true when it comes to fully associative hit rates but not in the n-way set associative hit rates. Unexpectedly, the FIFO and LRU methods are gapped between each other for 2-way and 4-way set associative tests.

The results showed that the higher the cache size the higher the hit rate will become. This is true because it decreases the number of times a miss will occur in the cache. This also means if the cache size was lower the lower the hit rate would become.

In conclusion, hit rates are affected by the cache size and the type of cache design that was used.