Lab Experiment Report of Lab 6

CS224
Section No: 5
Spring 2021
Lab No: 6
Full Name/Bilkent ID: Esad İsmail Tök/21801679

First Experiment Report With Matrix Dimension (N) = 75

- In this experiment the matrix dimension (N) is decided to be 75.
- Therefore the matrix size is: 75 x 75 = 5625.

Part 2-a)

- *Table 1* is a table that contains the miss rates and number of misses for the row-major average calculation of a 75x75 matrix with 5 different block sizes and 5 different cache sizes.

Block Size (Words) ——— **Cache Size (Bytes)**	**4**	**8**	**16**	**32**	**64**
512 (0.5 KB)	Number of Misses: 1431 ——— Miss Rate: 24%	Number of Misses: 717 ——— Miss Rate: 12%	Number of Misses: 361 ——— Miss Rate: 6%	Number of Misses: 183 ——— Miss Rate: 3%	Number of Misses: 94 ——— Miss Rate: 2%
1024 (1 KB)	Number of Misses: 1431 ——— Miss Rate: 24%	Number of Misses: 717 ——— Miss Rate: 12%	Number of Misses: 361 ——— Miss Rate: 6%	Number of Misses: 183 ——— Miss Rate: 3%	Number of Misses: 94 ——— Miss Rate: 2%
2048 (2 KB)	Number of Misses: 1431 ——— Miss Rate: 24%	Number of Misses: 717 ——— Miss Rate: 12%	Number of Misses: 361 ——— Miss Rate: 6%	Number of Misses: 183 ——— Miss Rate: 3%	Number of Misses: 94 ——— Miss Rate: 2%
4096 (4 KB)	Number of Misses: 1431 ——— Miss Rate: 24%	Number of Misses: 717 ——— Miss Rate: 12%	Number of Misses: 361 ——— Miss Rate: 6%	Number of Misses: 183 ——— Miss Rate: 3%	Number of Misses: 94 ——— Miss Rate: 2%
8192 (8 KB)	Number of Misses: 1431 ——— Miss Rate: 24%	Number of Misses: 717 ——— Miss Rate: 12%	Number of Misses: 361 ——— Miss Rate: 6%	Number of Misses: 183 ——— Miss Rate: 3%	Number of Misses: 94 ——— Miss Rate: 2%

Table 1: Miss rate and number of misses for direct mapped cache for "Row-Major" average

First Experiment Report With Matrix Dimension (N) 75

- *Table 2* is a table that contains the miss rates and number of misses for the column-major average calculation of a 75x75 matrix with 5 different block sizes and 5 different cache sizes.

Block Size (Words) / Cache Size (Bytes)	4	8	16	32	64
512 (0.5 KB)	Number of Misses: 5650 / Miss Rate: 95%	Number of Misses: 5640 / Miss Rate: 95%	Number of Misses: 5634 / Miss Rate: 95%	Number of Misses: 5631 / Miss Rate: 94%	Number of Misses: 5630 / Miss Rate: 94%
1024 (1 KB)	Number of Misses: 3375 / Miss Rate: 57%	Number of Misses: 5446 / Miss Rate: 91%	Number of Misses: 5634 / Miss Rate: 95%	Number of Misses: 5631 / Miss Rate: 94%	Number of Misses: 5630 / Miss Rate: 94%
2048 (2 KB)	Number of Misses: 2747 / Miss Rate: 46%	Number of Misses: 3929 / Miss Rate: 66%	Number of Misses: 5471 / Miss Rate: 92%	Number of Misses: 5631 / Miss Rate: 94%	Number of Misses: 5630 / Miss Rate: 94%
4096 (4 KB)	Number of Misses: 2744 / Miss Rate: 46%	Number of Misses: 3929 / Miss Rate: 66%	Number of Misses: 4518 / Miss Rate: 76%	Number of Misses: 5385 / Miss Rate: 90%	Number of Misses: 5630 / Miss Rate: 94%
8192 (8 KB)	Number of Misses: 1487 / Miss Rate: 25%	Number of Misses: 784 / Miss Rate: 13%	Number of Misses: 431 / Miss Rate: 7%	Number of Misses: 2315 / Miss Rate: 39%	Number of Misses: 5454 / Miss Rate: 92%

Table 2: Miss rate and number of misses for direct mapped cache for "Column-Major" average

First Experiment Report With Matrix Dimension (N) 75

- *Figure 1* is a graph that shows block size versus miss rate of 5 different cache sizes for the row-major average calculation of a 75x75 matrix.

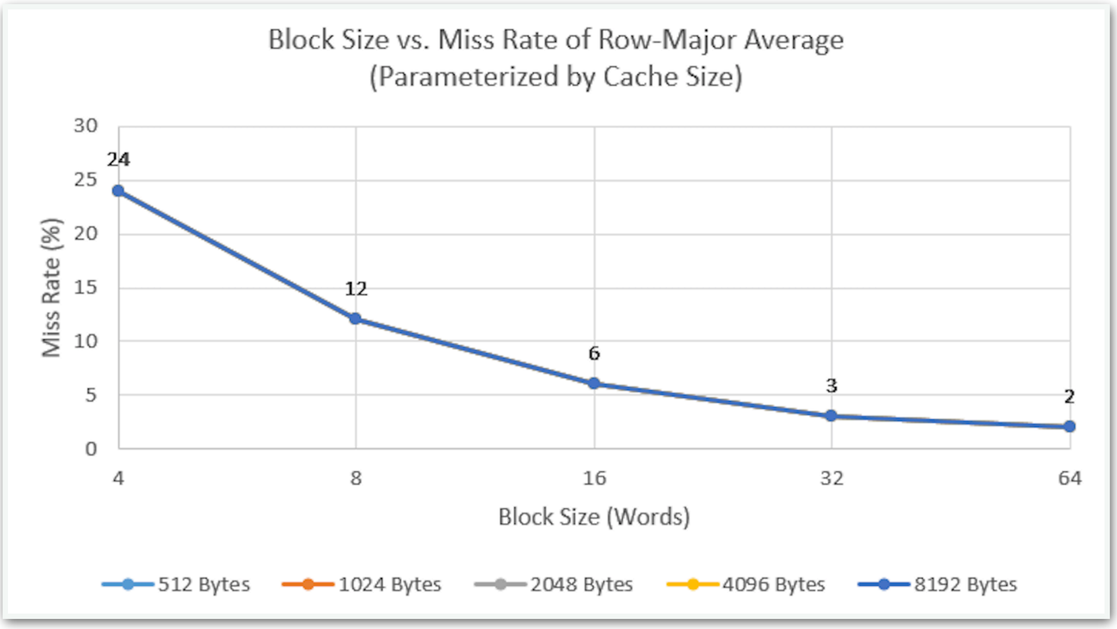

Figure 1: Block Size vs. Miss Rate for Row-major Average of 75x75 Matrix with 5 Different Cache Sizes

- *Figure 2* is a graph that shows block size versus miss rate of 5 different cache sizes for the column-major average calculation of a 75x75 matrix.

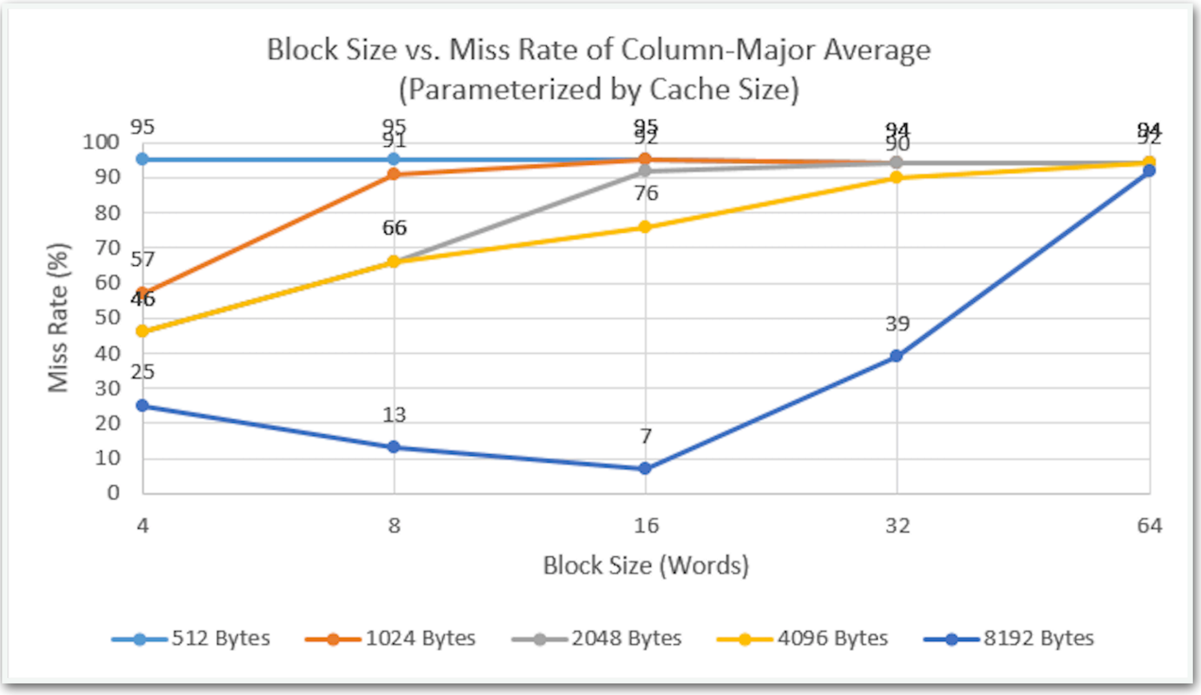

Figure 2: Block Size vs. Miss Rate for Column-major Average of 75x75 Matrix with 5 Different Cache Sizes

First Experiment Report With Matrix Dimension (N) = 75

Part 2-b)

- *Table 3* includes all 9 values that are required for each configuration pairs in column-major average calculation.

	Good Hit Rate Cache Size: 8192 Block Size: 16	**Medium Hit Rate Cache Size: 2048 Block Size: 8**	**Poor Hit Rate Cache Size: 1024 Block Size: 16**
Direct Mapped	Number of Misses: 431 ——————— Miss Rate: 7%	Number of Misses: 3929 ——————— Miss Rate: 66%	Number of Misses: 5634 ——————— Miss Rate: 95%
Fully Associative (LRU)	Number of Misses: 431 ——————— Miss Rate: 7%	Number of Misses: 5640 ——————— Miss Rate: 95%	Number of Misses: 5634 ——————— Miss Rate: 95%
Fully Associative (Random)	Number of Misses: 772 ——————— Miss Rate: 13%	Number of Misses: 3034 ——————— Miss Rate: 51%	Number of Misses: 5597 ——————— Miss Rate: 94%

Table 3: Miss rate and number of misses for poor, medium, and good hit rates with different cache implementations

- *Figure 3* is a graph that shows the plots for the values recorded in *Table 3*. Configuration Index indicates whether the configuration is Good Hit Rate, Medium Hit Rate, or Poor Hit Rate.
- Index 0 => Good Hit Rate
- Index 1 => Medium Hit Rate
- Index 2 => Poor Hit Rate

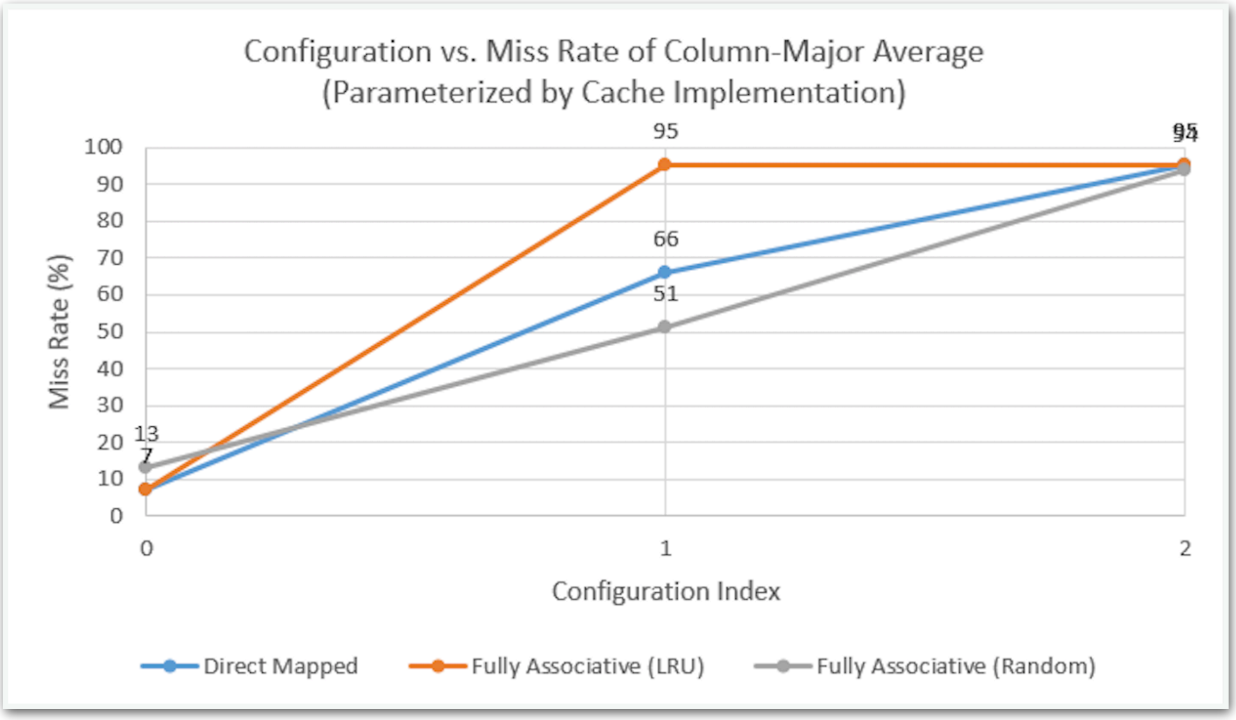

Figure 3: Configuration Index vs. Miss Rate for Column-major Average of 75x75 Matrix with 3 Different Cache Implementations

First Experiment Report With Matrix Dimension (N) = 75

Analysis of Part 2-b)

- In terms of Good Hit Rate, it can be seen that switching from direct mapped cache implementation to fully associative (LRU) implementation do not provide any benefit. It is because the cache is large enough to map needed memory locations in direct mapped cache and there is not so much conflict misses; therefore using a fully associative cache to decrease the conflict misses does not provide any benefit. Switching to Fully Associative Random cache again cannot provide us any benefit but in this time the random replacement policy also makes the cache remove some random locations that might be used later and therefore increases the miss rate because of compulsory misses.

- In terms of Medium Hit Rate, it can be seen that switching from direct mapped cache implementation to fully associative (LRU) implementation increases the miss rate. It is because the cache this time is smaller and the block size is also smaller then Good Hit Rated cache and since it is now fully associative cache all the memory locations are mapped into the same set and since the cache is not that big and block size is smaller, therefore whenever we are done with one column and try to start with next column in the column-major average calculation (in which we turn back to the nearby location of the beginning of previous column that is brought by spatial locality) we lost the consecutive array location that was previously came by spatial locality because of the LRU replacement and it causes us to have more compulsory and capacity misses. Capacity misses are also important factor here because there is only 1 set and the cache is not large enough to hold each matrix element. Switching to Fully Associative Random cache provides a benefit for us this time and decreases the miss rate. It is because in this implementation replacement is done randomly, therefore this time when we come to a nearby location of an array location we do not lost it by the LRU replacement but instead we lost a random location.

- In terms of Poor Hit Rate, it can be seen that switching from direct mapped cache implementation to fully associative (LRU) implementation do not provide any benefit. The hit rate was already poor and fully associative cache do not provide any change for us. It is because the cache size is too small this time and the cache misses we suffer from is not because of conflict misses but because of capacity misses. Switching to Fully Associative (LRU) cache can only decrease the rate of conflict misses however the problem in this case is not conflict misses but capacity misses. Therefore we cannot see any change between the implementations. Similarly, switching to Fully Associative Random cache provides very little benefit and decreases the miss rate by 1% and it is because the replacement policy not always replaces the block that we try to reach this time but it makes it random. However, again, since the cache is not that large, even this policy cannot help us in terms of the miss rate. The only way we can considerably increase the cache performance is to increase the cache size for this case.

First Experiment Report With Matrix Dimension (N) = 75

Part 2-c)

Medium Hit Rate
Cache Size: 2048
Block Size: 8

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	50%	36%	15%	5%
Miss Rate	50%	64%	85%	95%
Number of Misses	2982	3791	5084	5640

Table 4: Includes the rates of different set associative caches that is based on Medium Hit Rate Configuration

- According to the *Table 4,* the best performance is obtained with the 2-way set associative cache implementation because in that implementation the hit rate is greater than the others and the miss rate is less than the others.
- Switching from direct mapped cache to the 2-way set associative cache implementation reduced the miss rate from 66% to 50%. And provide us a significant improvement however when we keep increasing the N (the number of blocks in a set) we cannot get any improvement and the best N for the Medium Hit Rate configuration is 2.

Good Hit Rate
Cache Size: 8192
Block Size: 16

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	93%	93%	93%	93%
Miss Rate	7%	7%	7%	7%
Number of Misses	431	431	431	431

Table 5: Includes the rates of different set associative caches that is based on Good Hit Rate Configuration

First Experiment Report With Matrix Dimension (N) = 75

- According to the *Table 5,* all of the different N-way set associative cache implementations give the same miss and hit rates and there is no best option for "N" in terms of hit rate in the case of Good Hit Rate configuration.
- Switching from direct mapped cache to the 2-way set associative cache implementation does not provide us any benefit. It was the same case for the fully associative cache implementation for the good hit rate configuration as well. It is because the cache is large enough to map needed memory locations to unique locations and this reduces the possibility of conflict misses. In terms of preference, 2-way associative cache should be used among the N-way associative caches because all the implementations have the same hit rates but 2-way associative cache has less number of hardware components, therefore it is preferable.

Poor Hit Rate
Cache Size: 1024
Block Size: 16

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	5%	5%	5%	5%
Miss Rate	95%	95%	95%	95%
Number of Misses	5634	5634	5634	5634

Table 6: Includes the rates of different set associative caches That is based on Poor Hit Rate Configuration

- According to the *Table 6,* all of the different N-way set associative cache implementations give the same miss and hit rates and there is no best option for "N" in terms of hit rate in the case of Poor Hit Rate configuration.
- Switching from direct mapped cache to the 2-way set associative cache implementation does not provide us any benefit. It was the same case for the fully associative cache implementation for the poor hit rate configuration as well. It is because the cache is too small and has considerable large block size compared to its total size, therefore when a new word comes with the LRU replacement policy large block of words are removed and this miss rate is caused by capacity misses rather than conflict misses, therefore increasing the associativity cannot help us to reduce the miss rate. In terms of preference, 2-way associative cache should be used among the N-way associative caches because all the implementations have the same hit rates but 2-way associative cache has less number of hardware components, therefore it is preferable.

Second Experiment Report With Matrix Dimension (N) = 150

- In this experiment the matrix dimension (N) is decided to be 150.
- Therefore the matrix size is: 150 x 150 = 22500.

Part 2-a)

Table 7 is a table that contains the miss rates and number of misses for the row-major average calculation of a 150x150 matrix with 5 different block sizes and 5 different cache sizes.

Block Size (Words) / Cache Size (Bytes)	4	8	16	32	64
512 (0.5 KB)	Number of Misses: 5649 / Miss Rate: 25%	Number of Misses: 2826 / Miss Rate: 12%	Number of Misses: 1415 / Miss Rate: 6%	Number of Misses: 710 / Miss Rate: 3%	Number of Misses: 357 / Miss Rate: 2%
1024 (1 KB)	Number of Misses: 5649 / Miss Rate: 25%	Number of Misses: 2826 / Miss Rate: 12%	Number of Misses: 1415 / Miss Rate: 6%	Number of Misses: 710 / Miss Rate: 3%	Number of Misses: 357 / Miss Rate: 2%
2048 (2 KB)	Number of Misses: 5649 / Miss Rate: 25%	Number of Misses: 2826 / Miss Rate: 12%	Number of Misses: 1415 / Miss Rate: 6%	Number of Misses: 710 / Miss Rate: 3%	Number of Misses: 357 / Miss Rate: 2%
4096 (4 KB)	Number of Misses: 5649 / Miss Rate: 25%	Number of Misses: 2826 / Miss Rate: 12%	Number of Misses: 1415 / Miss Rate: 6%	Number of Misses: 710 / Miss Rate: 3%	Number of Misses: 357 / Miss Rate: 2%
8192 (8 KB)	Number of Misses: 5649 / Miss Rate: 25%	Number of Misses: 2826 / Miss Rate: 12%	Number of Misses: 1415 / Miss Rate: 6%	Number of Misses: 710 / Miss Rate: 3%	Number of Misses: 357 / Miss Rate: 2%

Table 7: Miss rate and number of misses for direct mapped cache for "Row-Major" average

Second Experiment Report With Matrix Dimension (N) = 150

- *Table 8* is a table that contains the miss rates and number of misses for the column-major average calculation of a 150x150 matrix with 5 different block sizes and 5 different cache sizes.

Block Size (Words) ——————— Cache Size (Bytes)	4	8	16	32	64
512 (0.5 KB)	Number of Misses: 22525 ——————— Miss Rate: 99%	Number of Misses: 22515 ——————— Miss Rate: 99%	Number of Misses: 22509 ——————— Miss Rate: 99%	Number of Misses: 22506 ——————— Miss Rate: 99%	Number of Misses: 22505 ——————— Miss Rate: 99%
1024 (1 KB)	Number of Misses: 20749 ——————— Miss Rate: 91%	Number of Misses: 22515 ——————— Miss Rate: 99%	Number of Misses: 22509 ——————— Miss Rate: 99%	Number of Misses: 22506 ——————— Miss Rate: 99%	Number of Misses: 22505 ——————— Miss Rate: 99%
2048 (2 KB)	Number of Misses: 13325 ——————— Miss Rate: 58%	Number of Misses: 21364 ——————— Miss Rate: 94%	Number of Misses: 22509 ——————— Miss Rate: 99%	Number of Misses: 22506 ——————— Miss Rate: 99%	Number of Misses: 22505 ——————— Miss Rate: 99%
4096 (4 KB)	Number of Misses: 5725 ——————— Miss Rate: 25%	Number of Misses: 11061 ——————— Miss Rate: 48%	Number of Misses: 21984 ——————— Miss Rate: 96%	Number of Misses: 22506 ——————— Miss Rate: 99%	Number of Misses: 22505 ——————— Miss Rate: 99%
8192 (8 KB)	Number of Misses: 5725 ——————— Miss Rate: 25%	Number of Misses: 11061 ——————— Miss Rate: 48%	Number of Misses: 18686 ——————— Miss Rate: 82%	Number of Misses: 20592 ——————— Miss Rate: 90%	Number of Misses: 22505 ——————— Miss Rate: 99%

Table 8: Miss rate and number of misses for direct mapped cache for "Column-Major" average

Second Experiment Report With Matrix Dimension (N) = 150

- *Figure 4* is a graph that shows block size versus miss rate of 5 different cache sizes for the row-major average calculation of a 150x150 matrix.

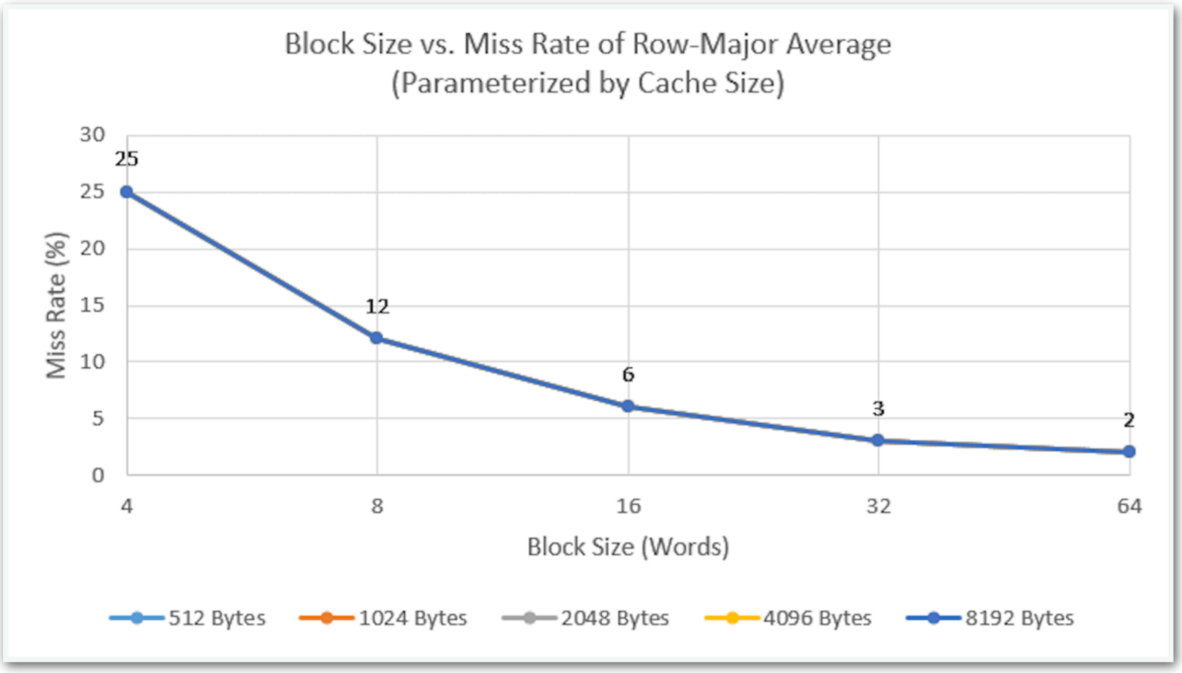

Figure 4: Block Size vs. Miss Rate for Row-major Average of 150x150 Matrix with 5 Different Cache Sizes

- *Figure 5* is a graph that shows block size versus miss rate of 5 different cache sizes for the column-major average calculation of a 150x150 matrix.

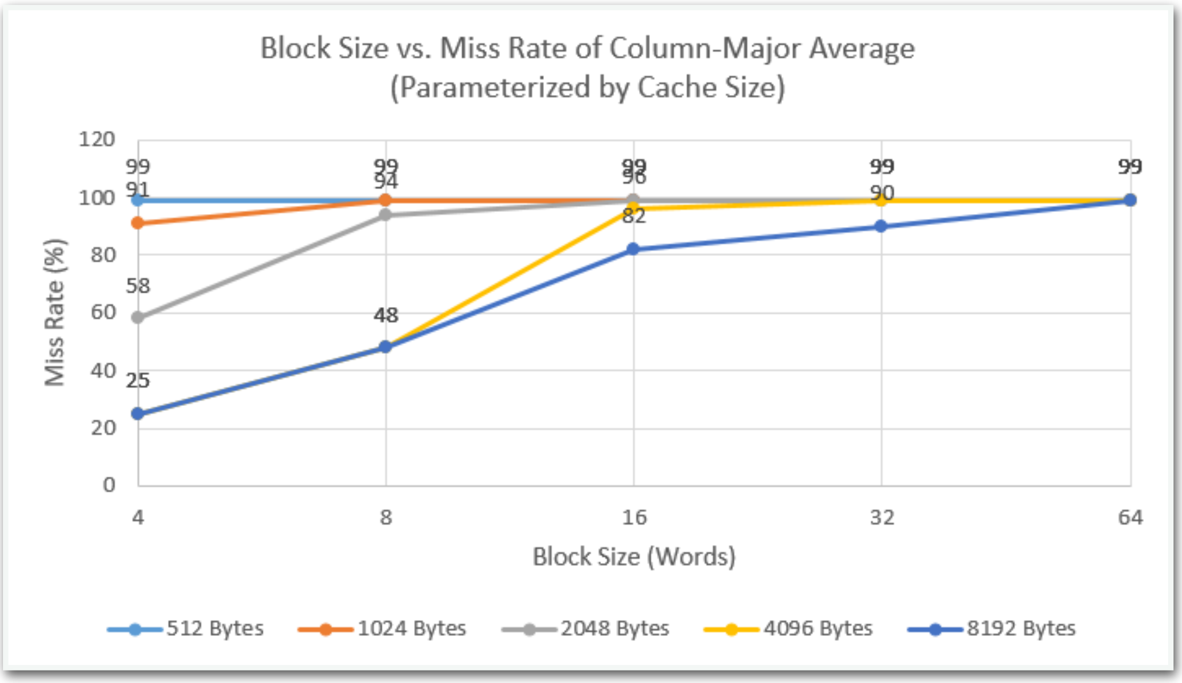

Figure 5: Block Size vs. Miss Rate for Column-major Average of 150x150 Matrix with 5 Different Cache Sizes

Second Experiment Report With Matrix Dimension (N) = 150

Part 2-b)

- *Table 9* includes all 9 values that are required for each configuration pairs in column-major average calculation.

	Good Hit Rate Cache Size: 4096 Block Size: 4	**Medium Hit Rate Cache Size: 2048 Block Size: 4**	**Poor Hit Rate Cache Size: 1024 Block Size: 16**
Direct Mapped	Number of Misses: 5725 ——————— Miss Rate: 25%	Number of Misses: 13325 ——————— Miss Rate: 58%	Number of Misses: 22509 ——————— Miss Rate: 99%
Fully Associative (LRU)	Number of Misses: 5725 ——————— Miss Rate: 25%	Number of Misses: 22525 ——————— Miss Rate: 99%	Number of Misses: 22509 ——————— Miss Rate: 99%
Fully Associative (Random)	Number of Misses: 9205 ——————— Miss Rate: 40%	Number of Misses: 14662 ——————— Miss Rate: 64%	Number of Misses: 22508 ——————— Miss Rate: 99%

Table 9: Miss rate and number of misses for poor, medium, and good hit rates with different cache implementations

- *Figure 6* is a graph that shows the plots for the values recorded in *Table 9*. Configuration Index indicates whether the configuration is Good Hit Rate, Medium Hit Rate, or Poor Hit Rate.
- Index 0 => Good Hit Rate
- Index 1 => Medium Hit Rate
- Index 2 => Poor Hit Rate

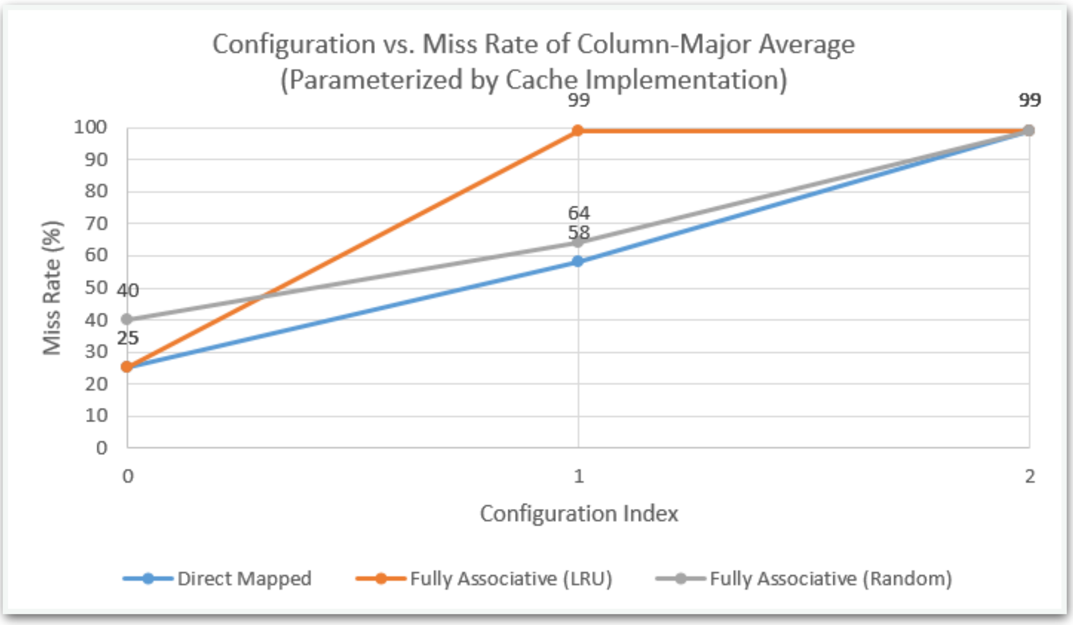

Figure 6: Configuration Index vs. Miss Rate for Column-major Average of 150x150 Matrix with 3 Different Cache Implementations

Second Experiment Report With Matrix Dimension (N) = 150

Analysis of Part 2-b)

- In terms of Good Hit Rate, it can be seen that switching from Direct Mapped Cache implementation to Fully Associative (LRU) Cache implementation does not provide us any benefit or harm. It simply does not change the miss rate. It is because the Direct Mapped cache in here is large enough to keep conflict misses as low as possible. Therefore switching to Fully Associative (LRU) implementation, which we do that to decrease the conflict misses, does not provide any benefit to us. On the other hand switching to the Fully Associative (Random) cache implementation significantly decreases our cache performance by increasing the miss rate. It is because this time the replacement policy is random and now instead of replacing the least recently used block, the cache replaces random blocks to put new blocks and it causes us to lose the words that we bring with a block by spatial locality which we want to use later. Therefore conflict misses increase.

- In terms of Medium Hit Rate, it can be seen that switching from Direct Mapped Cache implementation to Fully Associative (LRU) Cache implementation reduces our cache performance by increasing the miss rate. It is because our cache size here is smaller than the one in Good Hit Rate. Fully Associative (LRU) implementation has only 1 set and it has all blocks in it, therefore since the cache is not large enough to hold all the matrix elements there occurs capacity misses. Each time we come back a new column in our matrix we lost the words that were brought by previous block because of capacity miss and we need to retrieve It back and then lose other words that came with blocs and so on. Therefore even if we prevent conflict misses we encounter with compulsory and capacity misses in this implementation. On the other hand switching to the Fully Associative (Random) cache implementation also increases miss rate but it does not increase the miss rate as much as Fully Associative (LRU) does. It is because with the random replacement policy we do not lose the words in blocks consecutively each time we try to reach the next column in our matrix but this time the replacement is random and we can make use of spatial locality better than LRU replacement.

- In terms of Poor Hit Rate, it can be seen that switching from Direct Mapped Cache implementation to Fully Associative (LRU) Cache implementation does not change our cache performance because it does not change the miss rate or hit rate. It is because the cache size is considerably small this time and the miss rate was already considerably high in direct mapped implementation as well. The reason for that is not the conflict misses but it is capacity misses therefore switching to the Fully Associative (LRU) implementation does not provide us any benefit since it cannot prevent the capacity misses. Switching to the Fully Associative (Random) cache implementation also cannot help us to reduce the miss rate and the cache performance does not change again. The miss rate again is too high because of the capacity misses and changing the replacement policy from LRU to Random does not change anything. The only way we can improve the cache performance in this case would be increasing the cache size.

Second Experiment Report With Matrix Dimension (N) = 150

Part 2-c)

Medium Hit Rate
Cache Size: 2048
Block Size: 4

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	46%	23%	11%	1%
Miss Rate	54%	77%	89%	99%
Number of Misses	12303	17601	20421	22525

Table 10: Includes the rates of different set associative caches that is based on Medium Hit Rate Configuration

- According to the *Table 10,* the best performance is obtained with the 2-way set associative cache implementation because in that implementation the hit rate is greater than the others and the miss rate is less than the others.
- Switching from direct mapped cache to the 2-way set associative cache implementation reduced the miss rate from 58% to 54%. And provide us an improvement. However as we increase the N (the number of blocks in a set) the cache performance become worse since the miss rate increases gradually. Therefore the best N for the Medium Hit Rate configuration is 2.

Good Hit Rate
Cache Size: 4096
Block Size: 4

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	75%	75%	75%	75%
Miss Rate	25%	25%	25%	25%
Number of Misses	5725	5725	5725	5725

Table 11: Includes the rates of different set associative caches that is based on Good Hit Rate Configuration

Second Experiment Report With Matrix Dimension (N) = 150

- According to the *Table 11,* all of the different N-way set associative caches have the same cache performance and all have the hit rate of %75. Therefore there is no best option for the Good Hit Rate configuration in terms of hit rate.
- Switching from direct mapped cache to the 2-way set associative cache implementation does not change anything in terms of hit rate and miss rate. Similarly increasing the "N" (the number of blocks in a set) also does not provide us any change. It was the same case when we switch from Direct Mapped cache to Fully Associative (LRU) cache as well. The reason is that since we have a reasonably large cache size, we are already keeping the conflict misses at minimum possible rate in direct mapped cache therefore adding associativity does not provide us any benefit. However since increasing "N" adds more hardware to the implementation, it is preferable to choose 2 for the value of "N" if we need to use N-way set associative implementation.

Poor Hit Rate
Cache Size: 1024
Block Size: 16

N-way Set Associativity / Rates	2	4	8	16
Hit Rate	1%	1%	1%	1%
Miss Rate	99%	99%	99%	99%
Number of Misses	22509	22509	22509	22509

Table 12: Includes the rates of different set associative caches that is based on Poor Hit Rate Configuration

- According to the *Table 12,* all of the different N-way set associative caches have the same cache performance and all have the hit rate of %1. Therefore there is no best option for the Poor Hit Rate configuration in terms of hit rate.
- Switching from direct mapped cache to the 2-way set associative cache implementation does not change anything in terms of hit rate and miss rate. Similarly increasing the "N" (the number of blocks in a set) also does not provide us any change. It was the same case when we switch from Direct Mapped cache to Fully Associative (LRU) cache and Fully Associative (Random) cache implementations as well. It is because, the cache is too small and the reason behind the large miss rate is capacity misses rather than conflict misses, therefore increasing the associativity does not provide us any change in terms of miss rate or hit rate. However since increasing "N" adds more hardware to the implementation, it is preferable to choose 2 for the value of "N" if we need to use N-way set associative implementation.