CS224
Section No.: 1
Spring 2020
Lab No.: 6
Maryam Shahid
21801344

Part 1

2. a)

Instruction	Iteration No.				
	1	2	3	4	5
lw $t1, 0xA4($0)	Compulsory	Hit	Hit	Hit	Hit
lw $t2, 0xAC($0)	Compulsory	Hit	Hit	Hit	Hit
lw $t3, 0xA8($0)	Hit	Hit	Hit	Hit	Hit

b)

main memory size: 2^{32} bits
instruction length: $log_2(2^{32})$ = 32 bits
cache is directly mapped since n = 1 which has 8 words
size of block: 8 words
byte offset: 2 bit, set: 2 bit, block offset: 1 bit, tag: 32 – (2+2+1) = 27 bit

Total cache contains: (1 + 27 + 32 + 32) x 4 = **368 bits**

c) **1** (2:1) multiplexer, **2** equality comparators, **2** AND gates and **1** OR gate

3. a)

Instruction	Iteration No.				
	1	2	3	4	5
lb $t1, 0xA4($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lb $t2, 0xAC($0)	Compulsory	Capacity	Capacity	Capacity	Capacity
lb $t3, 0xA8($0)	Capacity	Capacity	Capacity	Capacity	Capacity

b)
main memory size: 2^{32} bits
instruction length: $log_2(2^{32})$ = 32 bits
cache is 2 since n = 2 which has 2 words as each block has 1
size of block: 1 word
byte offset: 2 bit, set: 0 bit, block offset: 0 bit, tag: 32 – (2) = 30 bit

Total cache contains: (1+30+32+1+30+32) x 1 = **126 bits**

c) **1** (2:1) multiplexer, **2** equality comparators, **2** AND gates, **1** OR gate.

4.

```
       .text
menu:
      la $a0, msgOption1
      li $v0, 4
      syscall

      la $a0, msgOption2
      li $v0, 4
      syscall

      la $a0, msgOption3
      li $v0, 4
      syscall

      la $a0, msgOption4
      li $v0, 4
      syscall

      la $a0, exitOption
      li $v0, 4
      syscall

      la $a0, chooseOption
      li $v0, 4
      syscall

      li $v0, 5
      syscall

      beq $v0, 1, createMatrixSizeN
      beq $v0, 2, displayElement
      beq $v0, 3, rowMajorSum
      beq $v0, 4, columnMajorSum
      beq $v0, 5, exit

      j menu

createMatrixSizeN:
      li $v0, 4
      la $a0, promptForN
      syscall

      li $v0, 5
      syscall

      move $s0, $v0
      mul $s2, $s0, $s0
      mul $a0, $s2, 4
      li $v0, 9
      syscall

      move $s1, $v0
      jal fillMatrix
      j menu

fillMatrix:
      addi $sp, $sp, -12
      sw $ra, 0($sp)
      sw $s1, 4($sp)
      sw $s2, 8($sp)
      li $t1, 1

writeElements:
      sw $t1, 0($s1)
      addi $s1, $s1, 4
      addi $t1, $t1, 1
      sle  $t3, $t1, $s2
      beq $t3, 1, writeElements

writingDone:
      lw $s2, 8($sp)
      lw $s1, 4($sp)
      lw $ra, 0($sp)
        addi $sp, $sp, 12
      jr $ra

displayElement:
      la $a0, enterRowNo
      li $v0, 4
      syscall

      li $v0, 5
      syscall
      move $t4, $v0

      la $a0, enterColNo
      li $v0, 4
      syscall

      li $v0, 5
      syscall
      move $t5, $v0

      addi $t4, $t4, -1
      mul $t4, $t4, $s0
      mul $t4, $t4, 4
      addi $t5, $t5, -1
      mul $t5, $t5, 4
      add $t4, $t4, $t5
      add $t3, $t4, $s1

      la $a0, displayMsg
      li $v0, 4
      syscall

      lw $a0, 0($t3)
      li $v0, 1
      syscall

      j menu

rowMajorSum:
      move $t0, $s1
      mul $t1, $s0, $s0
      li $t2, 0

rowMajorLoop:
      lw $a0, ($t0)
      add $t2, $t2, $a0

      addi $t0, $t0, 4
      addi $t1, $t1, -1
      bgt $t1, $0, rowMajorLoop

      la $a0, rowResult
      li $v0, 4
      syscall

      move $a0, $t2
      li $v0, 1
      syscall

      j menu

columnMajorSum:
      li $t0, 1
      li $t1, 1
      move $s2, $0
      move $s3, $s1

columnMajorLoop:
      add  $s4, $t0, -1
      mul  $s5, $s0, 4
      mul  $s4, $s4, $s5

      add  $s6, $t1, -1
      mul  $s6, $s6, 4
      add  $s4, $s4, $s6

      add  $s3, $s3, $s4
      lw   $a0, ($s3)
      add  $s2, $s2, $a0

      addi $t0, $t0, 1
      move $s3, $s1
      addi $s7, $t0, -1
      bne  $s7, $s0, columnMajorLoop
      li   $t0, 1
      addi $t1, $t1, 1
      addi $t3, $t1, -1
      bne  $t3, $s0, columnMajorLoop

      la $a0, colResult
      li $v0, 4
      syscall

      addi $a0, $s2, 0
      li $v0, 1
      syscall

      j menu

exit:
      li $v0, 10
      syscall

           .data
msgOption1:  .asciiz "\n1. Enter the matrix size in terms of its
dimensions (N)"
msgOption2:  .asciiz "\n2. Display desired elements of the matrix by its
row and column number"
msgOption3:  .asciiz "\n3. Obtain summation of matrix elements by row-
major (row by row) summation"
msgOption4:  .asciiz "\n4. Obtain summation of matrix elements by column-
major (column by column) summation"
exitOption:  .asciiz "\n5. Exit"
chooseOption:.asciiz "\nPlease choose one of the above: "
promptForN:  .asciiz "\nEnter N for dimension of matrix: "
enterRowNo:  .asciiz "\nEnter row number: "
enterColNo:  .asciiz "Enter column number: "
displayMsg:  .asciiz "Element in the given row/column is: "
rowResult:   .asciiz "Row-major summation: "
colResult:   .asciiz "Column-major summation: "
```

Part 2

Report for Matrix Size 1: 50 x 50

a) **Row-major addition**

	Block Size (words)				
Cache Size (bytes)	**8**	**16**	**32**	**64**	**128**
512	Miss rate: 11% No of misses: 336	Miss rate: 6% No of misses: 168	Miss rate: 3% No of misses: 86	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 23
1024	Miss rate: 11% No of misses: 335	Miss rate: 6% No of misses: 167	Miss rate: 3% No of misses: 85	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 23
2048	Miss rate: 11% No of misses: 334	Miss rate: 6% No of misses: 167	Miss rate: 3% No of misses: 84	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 23
4096	Miss rate: 11% No of misses: 334	Miss rate: 6% No of misses: 167	Miss rate: 3% No of misses: 84	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 23
8192	Miss rate: 11% No of misses: 334	Miss rate: 6% No of misses: 167	Miss rate: 3% No of misses: 84	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 23

Table 1.1 - direct-mapped 50x50 matrix, row major addition

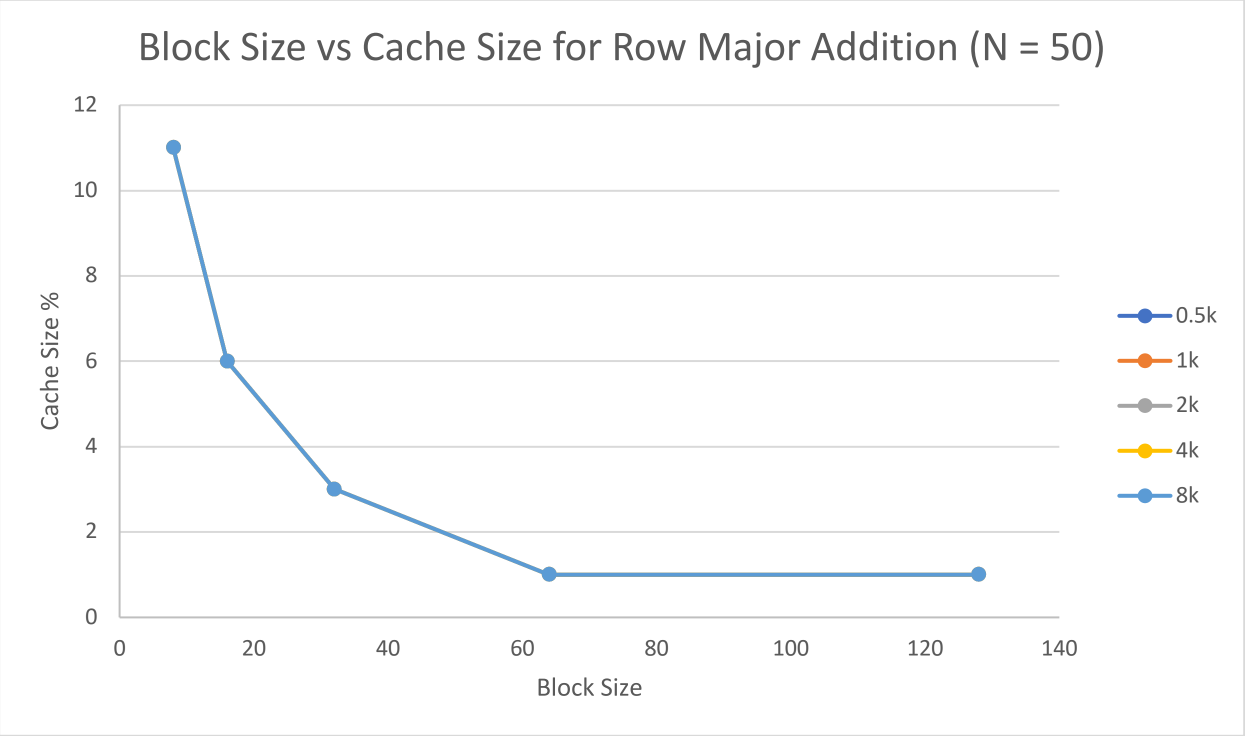

Figure 1.1 – Graph for direct-mapped 50x50 matrix, row major addition

Column Major Addition

	Block Size (words)				
Cache Size (bytes)	**8**	**16**	**32**	**64**	**128**
512	Miss rate: 81% No of misses: 2532	Miss rate: 83% No. of misses: 2571	Miss rate: 83% No of misses: 86	Miss rate: 71% No of misses: 41	Miss rate: 53% No of misses: 23
1024	Miss rate: 54% No of misses: 1633	Miss rate: 81% No. of misses: 2569	Miss rate: 83% No of misses: 2581	Miss rate: 71% No of misses: 41	Miss rate: 53% No of misses: 23
2048	Miss rate: 47% No of misses: 1117	Miss rate: 81% No of misses: 1722	Miss rate: 83% No of misses: 2551	Miss rate: 67% No of misses: 1957	Miss rate: 34% No of misses: 1023
4096	Miss rate: 37% No of misses: 1117	Miss rate: 46% No of misses: 1077	Miss rate: 54% No of misses: 1637	Miss rate: 67% No of misses: 1957	Miss rate: 34% No of misses: 1023
8192	Miss rate: 21% No of misses: 940	Miss rate: 23% No of misses: 957	Miss rate: 32% No of misses: 927	Miss rate: 27% No of misses: 784	Miss rate: 17% No of misses: 433

Table 1.2 - direct-mapped 50x50 matrix, column major addition

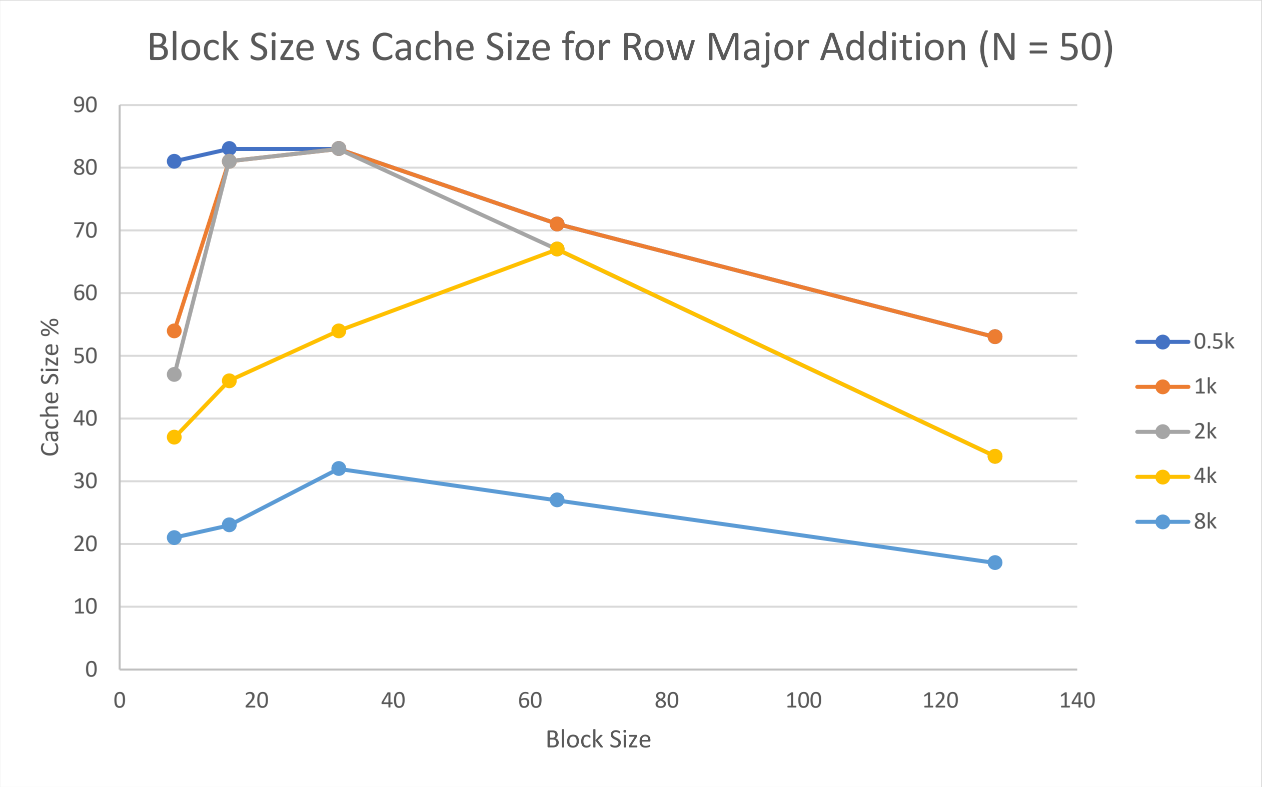

Figure 1.2 - Graph for direct-mapped 50x50 matrix, column major addition

b)

	Cache Type		
Cache size/ Block size	**Direct Mapping**	**Fully Associative LRU**	**Fully Associative Random**
1024/ 64 (good)	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 41	Miss rate: 1% No of misses: 41
1024/ 32 (medium)	Miss rate: 3% No of misses: 85	Miss rate: 3% No of misses: 85	Miss rate: 3% No of misses: 85
1024/ 16 (poor)	Miss rate: 6% No of misses: 167	Miss rate: 6% No of misses: 167	Miss rate: 6% No of misses: 167

Table 1.3

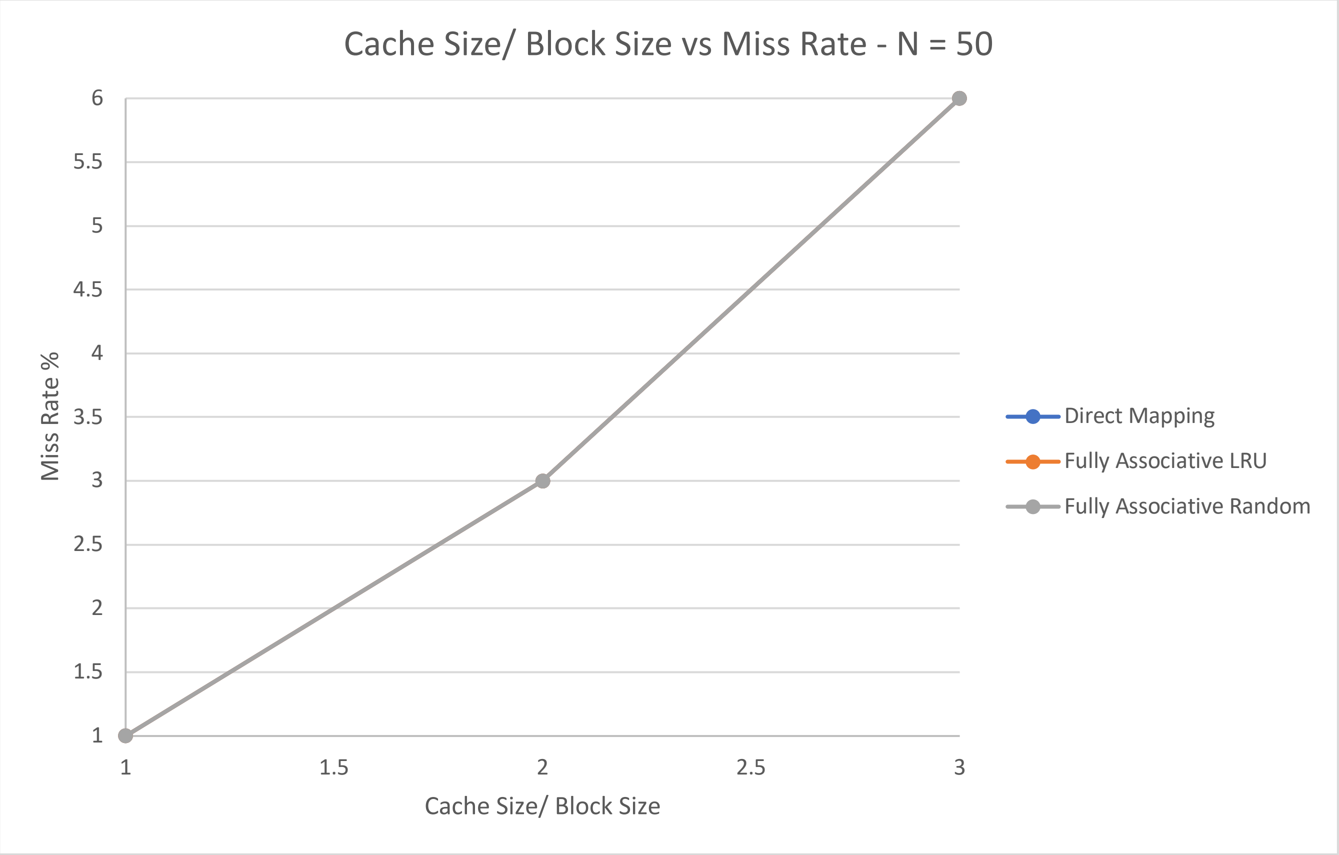

Figure 1.3

c)

N-way set associative Cache Set Size	Cache Size/ Block Size		
	1024/ 64	**1024/ 32**	**1024/ 16**
4	Miss rate: 1% No of misses: 41	Miss rate: 3% No of misses: 85	Miss rate: 6% No of misses: 167
6	Miss rate: 1% No of misses: 41	Miss rate: 3% No of misses: 85	Miss rate: 6% No of misses: 167
16	Miss rate: 1% No of misses: 41	Miss rate: 3% No of misses: 85	Miss rate: 6% No of misses: 167

Table 1.4

Report for Matrix Size 2: 100 x 100

a) **Row-major addition**

	Block Size (words)				
Cache Size (bytes)	**8**	**16**	**32**	**64**	**128**
512	Miss rate: 12% No of misses: 1271	Miss rate: 6% No of misses: 633	Miss rate: 3% No of misses: 324	Miss rate: 2% No of misses: 157	Miss rate: 1% No of misses: 83
1024	Miss rate: 12% No of misses: 1271	Miss rate: 6% No of misses: 633	Miss rate: 3% No of misses: 324	Miss rate: 2% No of misses: 157	Miss rate: 1% No of misses: 83
2048	Miss rate: 12% No of misses: 1271	Miss rate: 6% No of misses: 633	Miss rate: 3% No of misses: 324	Miss rate: 2% No of misses: 157	Miss rate: 1% No of misses: 83
4096	Miss rate: 12% No of misses: 1271	Miss rate: 6% No of misses: 633	Miss rate: 3% No of misses: 324	Miss rate: 2% No of misses: 157	Miss rate: 1% No of misses: 83
8192	Miss rate: 12% No of misses: 1271	Miss rate: 6% No of misses: 633	Miss rate: 3% No of misses: 324	Miss rate: 2% No of misses: 157	Miss rate: 1% No of misses: 83

Table 2.1 - direct-mapped 100x100 matrix, row major addition

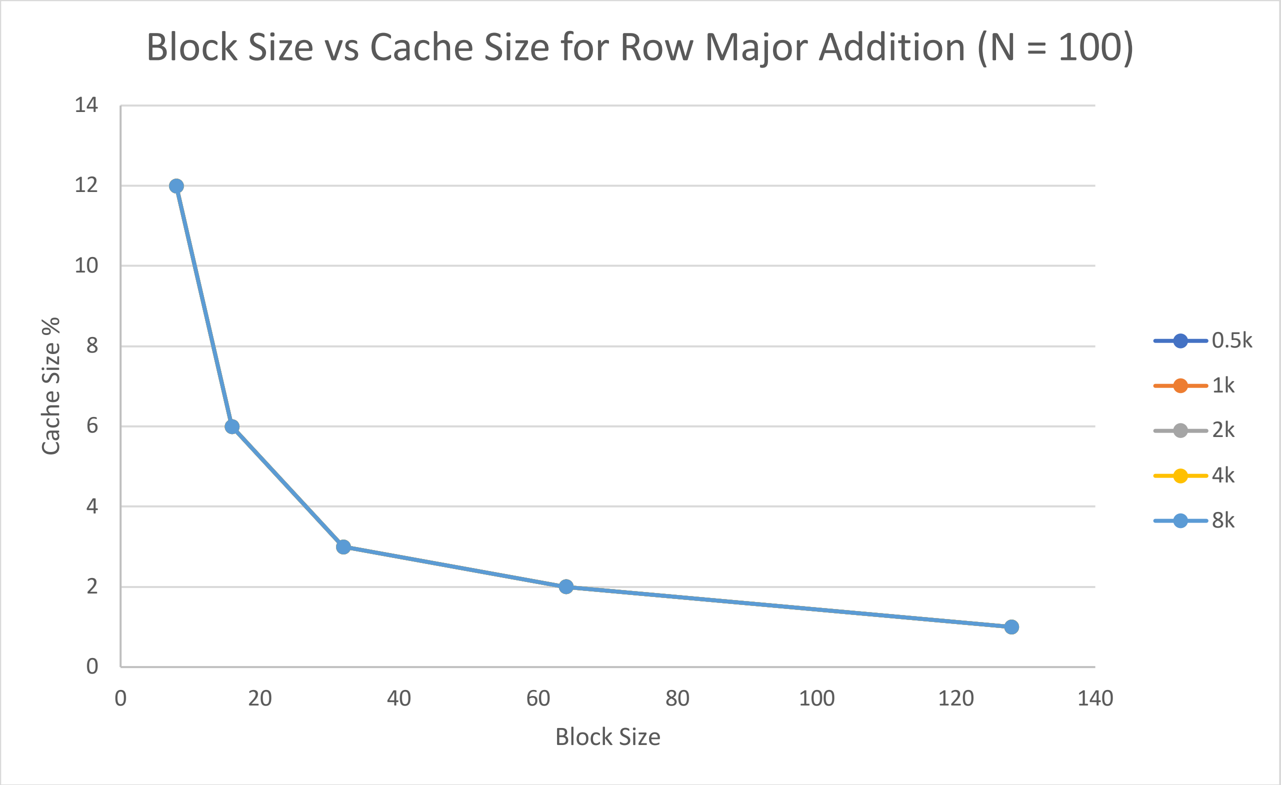

Figure 2.1 - Graph for direct-mapped 100x100 matrix, row major addition

Column Major Addition

	Block Size (words)				
Cache Size (bytes)	**8**	**16**	**32**	**64**	**128**
512	Miss rate: 95% No of misses: 10123	Miss rate: 95% No. of misses: 10122	Miss rate: 95% No of misses: 10111	Miss rate: 76% No of misses: 7847	Miss rate: 78% No of misses: 7848
1024	Miss rate: 95% No of misses: 10122	Miss rate: 95% No. of misses: 10122	Miss rate: 95% No of misses: 10111	Miss rate: 76% No of misses: 7847	Miss rate: 78% No of misses: 7848
2048	Miss rate: 77% No of misses: 8085	Miss rate: 95% No. of misses: 8012	Miss rate: 95% No of misses: 10108	Miss rate: 76% No of misses: 7847	Miss rate: 78% No of misses: 7848
4096	Miss rate: 66% No of misses: 8578	Miss rate: 74% No. of misses: 8983	Miss rate: 95% No of misses: 10108	Miss rate: 76% No of misses: 7847	Miss rate: 78% No of misses: 7848
8192	Miss rate: 41% No of misses: 8723	Miss rate: 73% No of misses: 8997	Miss rate: 87% No of misses: 10106	Miss rate: 76% No of misses: 7847	Miss rate: 78% No of misses: 7848

Table 2.2 - direct-mapped 100x100 matrix, column major addition

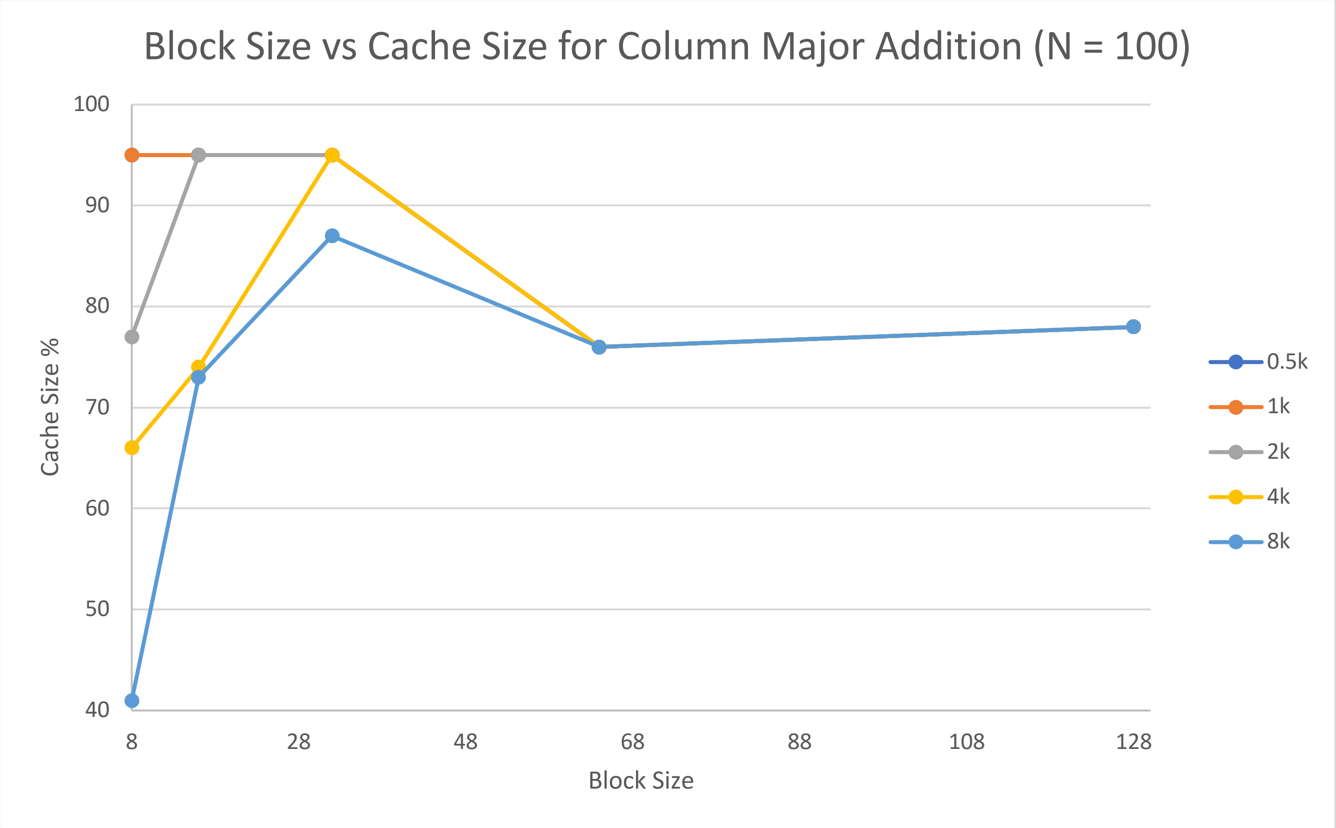

Figure 2.2 - Graph for direct-mapped 100x100 matrix, column major addition

b)

	Cache Type		
Cache size/ Block size	**Direct Mapping**	**Fully Associative LRU**	**Fully Associative Random**
1024/ 64 (good)	Miss rate: 2% No of misses: 157	Miss rate: 2% No of misses: 157	Miss rate: 2% No of misses: 157
1024/ 32 (medium)	Miss rate: 3% No of misses: 324	Miss rate: 3% No of misses: 324	Miss rate: 3% No of misses: 324
1024/ 16 (poor)	Miss rate: 6% No of misses: 633	Miss rate: 6% No of misses: 633	Miss rate: 6% No of misses: 633

Table 2.3

c)

N-way set associative Cache Set Size	Cache Size/ Block Size		
	1024/ 64	**1024/ 32**	**1024/ 16**
4	Miss rate: 2% No of misses: 157	Miss rate: 3% No of misses: 324	Miss rate: 6% No of misses: 633
6	Miss rate: 2% No of misses: 157	Miss rate: 3% No of misses: 324	Miss rate: 6% No of misses: 633
16	Miss rate: 2% No of misses: 157	Miss rate: 3% No of misses: 324	Miss rate: 6% No of misses: 633

Table 2.4