Review Questions:

5.2

In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored continuously.

```
for (I=0; I<8000; I++)
   for (J=0 ; J<8 ; J++)
   A[ I ] [J ]=B[J ] [0 ]+A [J ] [ I ] ;
```

5.2.1 [5] <5.1> How many 32-bit integers can be stored in a 16-byte cache line?

Ans: 4

5.2.2 [5] <5.1> References to which variables exhibit temporal locality? Hint: A variable (in this scenario) exhibits temporal locality if its current value will be in the cache at the beginning of each iteration of the inner loop.

Ans:

Since elements within the same row are stored contiguously, the element order is

A[0][0], A[0][1], A[0][2], A[0][3], A[0][4], A[0][5], A[0][6], A[0][7], A[1][0], A[1][1]…..

The variables that are accessed are

A[0] [0]=B[0] [0]+A [0] [0] ;

A[0] [1]=B[1] [0]+A [1] [0] ;

A[0] [2]=B[2] [0]+A [2] [0] ;

A[0] [3]=B[3] [0]+A [3] [0] ;

A[0] [4]=B[4] [0]+A [4] [0] ;

A[0] [5]=B[5] [0]+A [5] [0] ;

A[0] [6]=B[6] [0]+A [6] [0] ;

A[0] [7]=B[7] [0]+A [7] [0] ;

A[1] [0]=B[0] [0]+A [0] [1] ;

A[1] [1]=B[1] [0]+A [1] [1] ;

A[1] [2]=B[2] [0]+A [2] [1] ;

A[1] [3]=B[3] [0]+A [3] [1] ;

A[1] [4]=B[4] [0]+A [4] [1] ;

A[1] [5]=B[5] [0]+A [5] [1] ;

A[1] [6]=B[6] [0]+A [6] [1] ;

A[1] [7]=B[7] [0]+A [7] [1] ;

…………….

Access pattern for A[I][J] is A[0][0] A[0][1] A[0][2] A[0][3] A[0][4] A[0][5] A[0][6] A[0][7] A[1][0]…

⇨ Spatial locality

Access pattern for B[J][0] is B[0][0] B[1][0] B[2][0] B[3][0] B[4][0] B[5][0] B[6][0] B[7][0] B[0][0]….

⇨ Temporal locality

Access pattern for A[J][I] is A[0][0] A[1][0] A[2][0] A[3][0] A[4][0] A[5][0] A[6][0] A[7][0] A[0][1]A[1][1]…

⇨ None

The variables with temporal locality are I, J, B[J][0]

5.2.3 [5] <5.1> References to which variables exhibit spatial locality? Hint: A variable (in this scenario) exhibits spatial locality if it is part of a block that will be in the cache at the beginning of each iteration of the inner loop.

Ans: The variables with spatial locality are A[I][J]

5.2.4 [10] <5.1> How many 16-byte cache lines are needed to store all 32-bit matrix elements being referenced?

Ans: Moved to backup problems

Locality is affected by both the reference order and data layout. The same computation can also be written below in Matlab, which differs from C by contiguously storing matrix elements within the same column.

```
for 1=1:8000
   for J=1:8
   A(I,J)=B(J,0)+A(J,I) ;
   end
end
```

5.2.5 [5] <5.1> References to which variables exhibit temporal locality?

Ans: I, J, B(J, 0)

5.2.6 [5] <5.1> References to which variables exhibit spatial locality?

Ans: A(J, I), B(J, 0)

Exercise 5.3

Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

1,134, 212, 1, 135,213, 162, 161,2,44,41,221

5.3.1 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Binary address: 00000001_2, 10000110_2, 11010100_2, 1_2, 10000111_2, 11010101_2, 10100010_2, 10100001_2, 10_2, 101100_2, 101001_2, 11011101_2

Tag is the binary address right shift 4 bits. For example, the tag of 10000110_2 is 1000

Index is the binary address mod 16. For example, the index of 10000110_2 is 0110_2

	1	134	212	1	135	213	162	161	2	44	41	221
Index	0000	0001	0101	0000	0001	0101	1000	1000	0000	1011	1010	0111
H or M	M	M	M	H	H	H	M	H	H	M	M	M

5.3.2 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Ans: Moved to backup problems

5.3.3 [20] <5.2, 5.3> You are asked to optimized cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

Ans: Moved to backup problems

Exercise 5.3 Part B moved to backup problems

Exercise 5.4

For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
31-10	9-4	3-0

5.4.1 [5] <5.2> What is the cache line size (in words)?

Ans. $2^{(4-2)}$=4

5.4.2 [5] <5.2> How many entries does the cache have?

Ans: 2^6=64

5.4.3 [5] <5.2> What is the ratio between total bits required for such a cache implementation over the data storage bits? Don’t forget to include the valid bit

Valid bit + tag bit + data bit has 1+22+128 =151 bit.

The ratio is 151/128=1.179

Starting from power on, the following byte-addressed cache references are recorded.

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.4.4 [10] <5.2> How many blocks are replaced?

Ans:Binary addresses are

000000000000_2(=0) 000000000100_2 000000010000_2 (=16) 000010000100_2(=132)
000011101000_2 000010100000_2 (=160) 010000000000_2 (=1024) 000000011110_2(=30)
000010001100_2(=140) 110000011100_2(=3100) 000010110100_2(=180) 100010000100_2(=2180)

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180
Line ID	0	0	1	8	14	10	0	1	8	1	11	8

Hit/miss	M	H	M	M	M	M	M	H	H	M	M	M
Replace	N	N	N	N	N	N	Y,	N	N	Y	N	Y

5.4.5 [10] <5.2> What is the hit ratio?

3/12= 0.25

5.4.6 [20] <5.2> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

<Index, tag, data>:

<00000_2, 0001_2, mem[1024]>

<000001_2, 0011_2, mem[3100]>

<001000_2, 0010_2, mem[2180]>

<001010_2, 0000_2, mem[160]>

<001011_2, 0000_2, mem[180]>

<001110_2, 0000_2, mem[232]>

Exercise 5.5 Moved to backup problems

Exercise 5.6 Part B

Cache block size (B) can affect both miss rate and miss latency. Assuming the following miss rate table, assuming a 1-CPI machine with an average of 1.35 references (both instruction and data) per instruction. Help find the optimal block size given the following miss rates for various block sizes.

	8	16	32	64	128
A	8%	3%	1.8%	1.5%	2%
B	Removed				

5.6.4 [10] <5.2> What is the optimal block size for a miss latency of 20 xB cycles?

Miss latency is 20xB

Block size 8: CPI: 1+ (20*8)*8%=13.8

Block size 16: 1 + (20*16)*3%=10.6

Block size 32: 1+ (20*32)*1.8%=12.52

Block size 64: 1+ (20*64)*1.5%=20.2

Block size 128: 1+ (20*128)*2%=52.2

Therefore, block size 16 is the optimal

5.6.5 [10] <5.2> What is the optimal block size for a miss latency of 24+B cycles?

Block size 8: 1+ (24+8)*8%=3.56

Block size 16: 1 + (24+16)*3%=2.2

Block size 32: 1+ (24+32)*1.8%=2.008

Block size 64: 1+ (24+64)*1.5%=2.32

Block size 128: 1+ (24+128)*2%=4.04

Therefore, block size 32 is the optimal

5.6.6 [10] <5.2> For constant miss latency, what is the optimal block size?

Block size 8: 1+ X*8%=1+0.08X

Block size 16: 1 + X*3%=1+0.03X

Block size 32: 1+ X*1.8%=1+0.018X

Block size 64: 1+ X*1.5%=1+0.015X

Block size 128: 1+ X*2%=1+0.02X

Therefore, block size 64 is the optimal

Exercise 5.7

In this exercise, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional capacity. Assume that main memory accesses takes 70 ns and that memory accesses are 36% of all instructions. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

	L1 size	L1 miss rate	L1 hit time
P1	1 KB	11.4%	0.62ns
P2	2KB	8.0%	0.66ns

5.7.1 [5] <5.3> Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

Ans: P1: 1/0.62ns= 1.61Ghz, P2=1/0.66ns =1.52Ghz

5.7.2 [5) <5.3> What is the AMAT for P1 and P2?

Ans: P1:0.62 + 11.4% *70 = 8.6ns (= 8.6ns / 0.62 = 13.87 cycles)

P2:0.66 + 8.0%*70 = 6.26ns = (= 6.26/0.66 =9.48 cycles)

5.7.3 [5) <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for Pl and P2? Which processor is faster?

Ans: P1: 1 + (1.36 * 0.114 * 70/0.62) =18.5

P2: 1+ (1.36*0.08*70/0.66) = 12.54. Therefore P2 is faster

For the next three problems, we will consider the addition of an L2 cache to P1 to presumably make up for its 1imited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

L2 size	L2 miss rate	L2 hit time
512 KB	98%	3.22ns

5.7.4 [10] <5.3> What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

Ans: L2 global miss rate = 0.114 *0.98 = 0.11172

0.62ns + 11.4%*3.22ns + 0.11172* 70 = 8.81ns

8.81ns/0.62 = 14.21 cycles

Worse

5.7.5 [5] <5.3> Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

Ans: 1+ 1.36*(0.114*3.22/0.62+0.11172*70/0.62) =18.96

Exercise 5.8

This exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.2. For these exercises, refer to the table of address streams shown in Exercise 5.3.

a.	1,134,212,1,135,213,162,161,2,44,41,221

5.8.1 [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a three-way set associative cache with two-word blocks and a total size of 24 words. Use LRU replacement. For each reference identify the index bits, the tag bits, the block offset bits, and if it is a hit or a miss.

Ans:

Address		Tag	Index	Block offset	H/M
Decimal	Binary				
1	00000001	00000	00	1	M
134	10000110	10000	11	0	M
212	11010100	11010	10	0	M
1	00000001	00000	00	1	H
135	10000111	10000	11	1	H
213	11010101	11010	10	1	H
162	10100010	10100	01	0	M
161	10100001	10100	00	1	M
2	00000010	00000	01	0	M
44	00101100	00101	10	0	M
41	00101001	00101	00	1	M
221	11011101	11011	10	1	M

	Block 0	Block 1	Block 2
Set 00	0,1	160,161	40,41
Set 01	162,163	2,3	
Set 10	212,213	44,45	220,221
Set 11	134,135		

5.8.2 [10] <5.3> Using the references from Exercise 5.3, show the final cache contents for a fully associative cache with one-word blocks and a total size of 8 words. Use LRU replacement. For each reference, identify the index bits, the tag bits, and if it is a hit or a miss.

Ans:

Address		Tag	H/M
decimal	binary		

1	00000001	00000001	M
134	10000110	10000110	M
212	11010100	11010100	M
1	00000001	00000001	H
135	10000111	10000111	M
213	11010101	11010101	M
162	10100010	10100010	M
161	10100001	10100001	M
2	00000010	00000010	M
44	00101100	00101100	M
41	00101001	00101001	M
221	11011101	11011101	M

Block No.	Block 0
0	221
1	44
2	41
3	135
4	213
5	162
6	161
7	2

Multilevel caching is an important technique to overcome the 1imited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters:

Base CPI, no memory stalls	Processor speed	Main memory access time	First-level cache miss rate per instruction	Second level cache direct mapped speed	Global miss rate with second level, direct	Second level cache eight way set associative speeds	Global miss rate with second-level cache eight way set associativity
2	3 GHZ	125 ns	5%	15 cycles	3.0%	25 cycles	1.8%

5.8.4 [10] <5.3> Calculate the CPI for the processor in the table using: 1) only a first level cache, 2) a second level direct-mapped cache, and 3) a second level eight-way set associative cache. How do these numbers change if main memory access time is doubled? If it is cut in half ?

Base CPI: 2.0

Memory miss cycles: 125 cycles/(1/3) ns/clock = 375 clock cycles

1. Total CPI: 2.0 + 375 × 5% = 20.75/39.5/11.375 (normal/double/half)

2. Total CPI: 2.0 + 15 × 5% + 375 × 3% = 14/25.25/8.375 (normal/double/half)

3. Total CPI: 2.0 + 25 × 5% + 375 × 1.8% = 10/16.75/6.625 (normal/double/half)

5.8.5 [10] <5.3> It is possible to have an even greater cache hierarchy than two levels. Given the processor above with a second level, direct-mapped cache, a designer wants to add a third level cache that takes 50 cycles to access and will reduce the global miss rate to 1.3%. Would this provide better performance? In general, what are the advantages and disadvantages of adding a third level cache?

Base CPI: 2.0

Memory miss cycles: 125 cycles/(1/3) ns/clock = 375 clock cycles

Total CPI: 2.0 + 15 × 5% + 50 × 3% + 375 × 1.3% = 9.125

This would provide better performance, but may complicate the design of the processor. This could lead to: more complex cache coherency, increased cycle time, larger and more expensive chips.