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?

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.

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.

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

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?

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

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.

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.

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?

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)?

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

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

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?

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

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>.

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?

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

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

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?

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

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?

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?

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?

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.

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.

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 ?

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?