CS 351 Exam 3 May 15, 2017

Name: _____

Rules and Hints

- You may use one handwritten 8.5 × 11″ cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- You may write your answers in the form *[mathematical expression][units]*. There is no need to actually do the arithmetic.

Grade

	Your Score	**Max Score**
Problem 1: Cache: short answer		16
Problem 2: Average memory access time		10
Problem 3: Cache tracing		12
Problem 4: Virtual memory: short answer		8
Problem 5: Virtual memory: tracing		10
Problem 6: Disks and RAID		28
Problem 7: Parallelism: short answer		16
Total		**100**

Problem 1: Cache: short answer (16 points)

Part A (4 points)

Caches: why?

Part B (4 points)

Define the term *cold start* miss (also known as *compulsory* miss). Is there any way to reduce the number of compulsory misses?

Part C (4 points)

Assuming that your L2 cache is larger than your L1 cache, is it possible for the L2 cache to have a higher miss rate? Explain.

Part D (4 points)

You have a program that processes a large amount of data. For each piece of data, your program reads it several times in quick succession, writes it once, and never touches it again. Would a write-back or write-through cache perform better for this program, and why?

Problem 2: Average memory access time (10 points)

You have a memory system with an L1 cache whose access time is 1 processor cycle, an L2 cache whose access time is 4 processor cycles, an L3 cache whose access time is 15 processor cycles, and a main memory whose access time is 300 processor cycles. What is the average memory access time of a program that has an 80% hit rate in each level of cache?

Problem 3: Cache tracing (12 points)

You have a 32-byte direct-mapped cache with 4-byte blocks. Assuming that the cache is initially empty, give the hit rate of the following sequence of accesses, and show the final state of the cache. The spaces are just for readability.

```
1100 0110
1101 0000
1100 1001
0010 1011
1001 1100
0101 0011
1001 0010
1101 1001
1101 0111
1101 1000
```

Problem 4: Virtual memory: short answer (8 points)

Part A (4 points)

Virtual memory: why?

Part B (4 points)

For a typical program you'd write in CS 215 or 315, would you expect its TLB hit rate to be higher or lower than its L1 hit rate? Explain.

Problem 5: Virtual memory: tracing (10 points)

You have a process with the following (flat) page table:

Index	Valid	PPN
0	1	12
1	1	13
2	1	14
3	1	15

Part A (4 points)

If virtual addresses are 8 bits, how big is each page?

Part B (6 points)

Translate the virtual address 1011 1001 to its physical counterpart. Don't worry about leading zeroes for the physical address.

Problem 6: Disks and RAID (28 points)

For this problem, you are dealing with disks whose seek time is 9 ms, rotation speed is 7200 rpm, and transfer rate is 150 MB/s.

Part A (8 points)

How long does an 8KB write take, and how many can you do per second with one disk?

Part B (4 points)

If your workload consists mostly of 8KB writes, would flash be a better solution than hard disk? Explain.

Part C (16 points)

You buy an array of 8 of these disks and fill each one with data (JBOD). What is the latency of an 8 KB write in this configuration, and what is the throughput for the array?

Repeat for the following configurations. Include any extra disks you need to buy for redundancy in your performance calculations.

- RAID 0
- RAID 1
- RAID 5

Problem 7: Parallelism: short answer (16 points)

Part A (8 points)

You are designing a snoopy cache coherence protocol (the kind discussed in class). For each of the following accesses made by Processor 1, explain what information (if any) Processor 1 might need to send to other processors to preserve cache coherence, and what kind of information it might need to receive from them.

- Read hit
- Write hit
- Read miss
- Write miss

Part B (4 points)

Give an example of data-parallel C/C++ code that might not be well-suited to running on a GPU. (Partial credit for just providing data-parallel code.)

Part C (4 points)

A certain program takes 100 seconds to run on a single core, 60 seconds on 2 cores, and 40 seconds on 4 cores. What is its efficiency on 2 and 4 cores?

.