# CS 351 Exam 3 Mon. 5/16/2016

## 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.
- You may use extra scratch paper if you need more space, but make it clear where to find your answer to each question.

### Grade

| | **Your Score** | **Max Score** |
|---|---|---|
| *Problem 1:* Short answer | | 13 |
| *Problem 2:* Virtual memory | | 19 |
| *Problem 3:* RAID | | 24 |
| *Problem 4:* Parallelism | | 12 |
| *Problem 5:* Cache | | 32 |
| **Total** | | **100** |

# Problem 1: Short answer (13 points)

## Part A (1 point)

*+1 extra credit point if entire class gets it right.*

Third time's the charm: how many bits are in a byte?

## Part B (6 points)

For each of the following, state whether it is (a) on the processor chip; (b) in RAM; (c) on disk, or (d) unclear. If you answer (d), explain.

**TLB:**

**L2 cache:**

**Page table:**

## Part C: AMAT (6 points)

What is the average memory access time of the memory system detailed below? Assume 25% of instructions are loads, 0% are stores, and that none of these accesses causes a page fault.

- L1 instruction cache: 99% hit rate, 1-cycle access time
- L1 data cache: 95% hit rate, 1-cycle access time
- Unified L2 cache: 90% hit rate, 5-cycle access time
- Main memory: 250-cycle access time

# Problem 2: Virtual memory (19 points)

Answer the following questions about a system that can accommodate 16 GB of RAM and has 32-bit virtual addresses and a page size of 4 KB.

## Part A: Virtual pages (4 points)

How many virtual pages will each process have?

## Part B: Physical pages (4 points)

How many physical pages will the system have?

## Part C: Page table organization (7 points)

What is the minimum size of a page table entry? What is the minimum size of a page table?

## Part D: Translation (4 points)

Explain, as specifically as possible, how you would translate the virtual address 0xcfa548b2 to a physical address.

## Problem 3: RAID (24 points)

Consider a disk with the following characteristics:

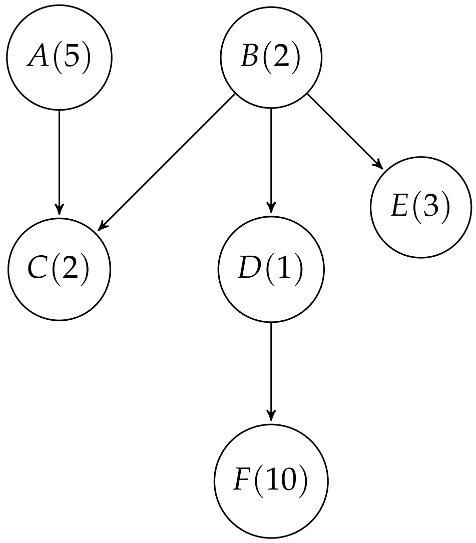
- 5 ms average seek time
- 5200 rpm
- Transfer rate of 100 MB/s

Fill in the table below. To make the table fit, I have abbreviated small write as ”SW” and a 5 MB write as ”LW” (for large).

| | SW latency (ms) | SW throughput (writes/s) | LW latency (ms) |
|---|---|---|---|
| RAID 0 | | | |
| RAID 1 | | | |
| RAID 5 | | | |

## Problem 4: Parallelism (12 points)

Consider the task graph below, where an arrow from X to Y means that Y cannot start executing until X completes. The amount of time (in some unspecified units) that each task takes is in parentheses. The individual tasks (nodes) cannot be parallelized.



### Part A (4 points)

What is the minimum number of threads necessary to obtain the maximum possible performance from this code? List an optimal allocation of tasks (graph nodes) to threads.

### Part B (5 points)

What is the maximum possible *speedup* of a parallel version, compared to executing all of the tasks in some sequential order?

## Part C (4 points)

What is the *efficiency* for the number of threads you identified in Part A?

# Problem 5: Cache (32 points)

You are designing a hilariously tiny memory system with 8-bit memory addresses and 16B of data in the first-level cache.

## Part A (16 points)

You have decided to make the cache direct-mapped, and you are deciding between 1B and 4B blocks. Draw both options (including metadata).

## Part A continued

Give a sequence of references (as many as necessary) that will have a higher hit rate in the cache with 4B blocks. Show the final state of both caches in your diagram on the previous page.

Give a sequence of references (as many as necessary) that will have a higher hit rate in the cache with 1B blocks.

## Part B (16 points)

You have reconsidered your decisions in Part A. Now you are sure you want 2-byte blocks, and the only question is whether to have a direct-mapped or a 2-way set associative cache. Draw both options (including metadata).

## Part B continued

Given the following sequence of references, show the final state of each cache in your Part A drawing. You may want to make extra notes to maximize partial credit in case of a mistake.

0000 0001
0000 0011
1110 0011
0000 0000
1110 0011
1010 0011

If these references are all writes, how many bytes will the direct-mapped version of this cache send to memory if it is write-back? What about write-through?

[blank]