CS 351 Exam 3, Fall 2012

Your name: ____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This is the only resource you may consult during this exam.
- Include explanations and comments in your answers in order to maximize your partial credit. However, you will be penalized for giving extraneous incorrect information.
- You may use the backs of these pages if you need more space, but make it clear where to find your answer to each question.
- Unless otherwise specified, you do not need to work out the arithmetic on math problems. Just do enough algebra to set up an answer of the form: Answer = [arithmetic expression] [units]

Grade (instructor use only)

	Your Score	**Max Score**
Problem 1: Short answer		30
Problem 2: Cache mappings		20
Problem 3: Cache sizing		10
Problem 4: AMAT with VM		15
Problem 5: Page table mappings		15
Problem 6: Page table sizing		10
Total		100

Problem 1: Short answer (30 points)

a) [5 points] You are thinking about making a new version of your chip that doubles the size of the L2 cache. What are the pros and cons of this choice? Assume that your chip is otherwise identical and you are using the same manufacturing technology.

b) [5 points] For the types of caches we have discussed, is the first access to a particular memory address guaranteed to be a miss? Why or why not?

c) [5 points] When you do a page table or TLB lookup, what information do you know already, and what are you trying to find out? Be specific.

d) [5 points] When a guest OS tries to set the page table base register, how should the VMM respond?

e) [5 points] Describe a machine-code instruction (either real or made up) that would force a VMM to do binary translation instead of direct execution or trap-and-emulate. Explain why binary translation is the only option for this instruction.

f) [5 points] Assume that you have a direct-mapped cache, and addresses *A* and *B* map to the same cache block. Consider the following stream of references:

1. *Load A*
2. *Store A*
3. *Load B*
4. *Store B*
5. *Store B*
6. *Store B*
7. *Load A*

If the cache is write-through, which of these references will cause the cache to send a write to main memory?

What if the cache is write-back?

Problem 2: Cache mappings (20 points)

For each of the following cache configurations, show the final cache state (valid and tag bits only) after the following references. You must draw the entire cache.

LOAD 00011101
LOAD 10110101

Main memory addresses are 8 bits.

A. The cache has:
   32B of data
   4B blocks
   Direct mapping

B. The cache has:
   32B of data
   2B blocks
   2-way set associative mapping

Problem 3: Cache sizing (10 points)

Consider the following memory hierarchy:

- Memory addresses are 32 bits.
- The L1 cache contains 256 KB of data.
- The L1 cache is write-back and 2-way set associative.
- L1 cache blocks are 8B each.

How big is the L1 cache (data and metadata)? Be sure to label your answer with units.

Problem 4: AMAT (15 points)

Assume that a program is executed on a system with the following memory hierarchy and hit rates.

- TLB: 99% hit rate, 1-cycle access time (but can be accessed in parallel with L1)
- L1 cache: 98% hit rate, 1-cycle access time
- L2-cache: 95% hit rate, 5-cycle access time
- Main memory: 100% hit rate, 80-cycle access time.

Page faults are handled by a magical elf who freezes time to bring the data from disk into main memory and update the page table. The elf does not, however, modify the L1 or the TLB. That would be crazy.

A. Once the physical address is known, what is the average memory access time?

B. What is the AMAT when the address translation process is taken into account?

Problem 5: Page table mappings (15 points)

Consider an unrealistically tiny system with 8 pages of physical memory. This system has 3 running processes that each have 4 virtual pages. The processes are not currently sharing any physical pages. There is no TLB.

Show a sample page table for each process. You should make up a set of mappings between virtual and physical addresses that are consistent with the description above and that use the entire physical memory.

PROCESS A:

PROCESS B:

PROCESS C:

If the page size is 32B, how would you translate a request from Process A for address 1001010?

Problem 6: Page table sizing (10 points)

A byte-addressable memory system has:

- 3 protection bits
- 36-bit virtual addresses
- 44-bit physical addresses
- 8 KB pages

A. How many virtual pages does each process have?

B. How many physical pages does the system have?

C. How many bits are needed for each page table entry (data and metadata)? Explain.