CS 351 Exam 3, Fall 2011

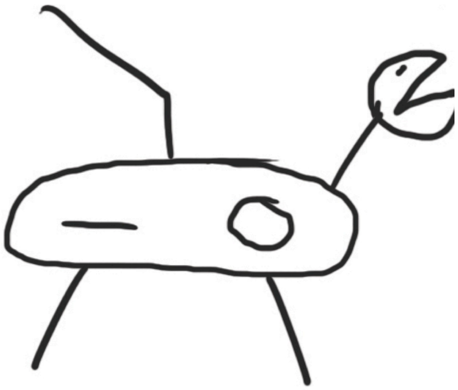
Your name: ____

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 (muxen use only)

	Your Score	Max Score
Problem 1		20
Problem 2		20
Problem 3		25
Problem 4		15
Problem 5		2-
Total		100



Problem 1: Average memory access time (20 points)

Assume a program with the following instruction mix: 40% integer instructions, 30% load and store instructions, 20% branches and jumps, and 10% floating-point instructions.

Assume that the program is executed on a system with the following memory hierarchy and hit rates. This system does NOT have virtual memory.

- Level 1 instruction cache: 98% hit rate, 1-cycle access time.
- Level 1 data cache: 95% hit rate, 1-cycle access time.
- Level 2 combined cache: 90% hit rate, 4-cycle access time.
- Main memory: 100% hit rate, 80-cycle access time.

---

a) How many cycles does the average instruction spend accessing the memory subsystem?

b) For this program (and most others), the instruction cache has a higher hit rate than the Level 1 data cache. Why do you think this is?

c) If this system had virtual memory and no TLB, how would your answer from part (a) change? (Be quantitative – give an exact increase or decrease.)

Problem 2: Cache sizing (20 points)

A byte-addressable machine with 32-bit memory addresses has a cache with the following properties:

- 8-byte cache blocks
- 32 KB of data in the cache
- 2-way set-associative
- Write-back

A. How many blocks are in this cache?

B. On average, how many bits of metadata are required per block? Explain.

C. How many BYTES is the entire cache (data and metadata)?

D. 1 KB = 2^X B. What is X? (+1 if everyone gets this right)

Problem 3: Page table sizing (25 points)

A byte-addressable memory system has:

- 3 protection bits
- A 32-entry TLB
- 32-bit virtual addresses
- 48-bit physical addresses
- 4 KB pages

The system can run up to 512 processes at once.

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.

D. If the TLB is NOT flushed on a context switch, how many bits are needed for each TLB entry? Explain.

E. How many PAGES will each process's page table occupy?

Problem 4: Cache mappings (15 points)

For each of the following caches, show the cache contents after memory references to addresses 01010110 and 10110001. Include any necessary metadata. The main memory has 8-bit addresses.

Be sure you show at LEAST the valid bit and tag of each block.

A. The cache has:
   - 8B of data (total)
   - 1-byte blocks
   - Direct mapping

B. [5 points] The cache has:
- 4-byte blocks
- Direct mapping
- 16B of data (total)

C. [5 points] The cache has:
- 2-byte blocks
- 2-way set associativity
- 16B of data (total)

Problem 5: Virtualization (20 points)

A. Give an example of a machine code instruction that would be virtualized using each of the following techniques. Your instruction does not have to exactly match an instruction on a real architecture, but you should explain how your instruction works and why the given virtualization technique is the right one.
   1. Directly execute the instruction on the hardware

   2. Trap-and-emulate; when the instruction causes a fault, it will trap, and the VMM can execute code that simulates its effect.

   3. Binary translation; the instruction cannot be safely executed on the hardware and must be translated into one or more safer instructions.

B. Although running on a VMM should not affect the correctness of guest OS code, it will often affect the performance. Explain how the performance of a memory reference might be dramatically different if an OS is running as a virtualized guest.