EECS 570 *Final Exam*

Winter 2015

Name: _____ unique name: _____

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

_____

Scores:

#	Points
1	**/ 21**
2	**/ 32**
3	**/ 12**
4	**/ 35**
Total	**/**

NOTES:

- Closed book, closed notes.
- Calculators are allowed, but no PDAs, Portables, Cell phones, etc.
- Don't spend too much time on any one problem.
- You have about 90 minutes for the exam (avg. 22 minutes per problem).
- There are 11 pages in the exam (including this one). Please ensure you have all pages.
- **Be sure to show work and explain what you've done when asked to do so.**

1) Chip Multiprocessing/Multithreading [21 points]
(Thanks to Babak Falsafi for several of these questions).

a) Briefly describe each of the following. State how each form of multithreading improves performance. [9 points]

i) switch-on-miss multithreading

ii) fine-grain multithreading

ii) simultaneous multithreading

b) Would OLTP throughput increase, decrease, or not change when using a simultaneously multithreaded core as compared to a superscalar core? **Explain.** [6 points]

c) What is the key shortcoming in superscalar that speculatively-threaded designs (e.g., MultiScalar) address? **Explain**. [6 points]

2) Interconnection networks [32 points]

a) Briefly state an advantage and a disadvantage of source routing. [6 points]

b) Briefly state an advantage and a disadvantage of the flattened butterfly topology as compared to a mesh with the same number of nodes.[6 points]

For parts (c) through (f), consider a 16-node 2D torus interconnect. Suppose there are 64-bit wide links in each direction between adjacent routers, as well as 64-bit incoming and outgoing links to each CPU. The interconnect is clocked at 1 GHz, and the link latency is a single cycle. All network messages include a 64-bit header and 64 bytes of message payload (e.g., a 64-byte cache block). The routers use the conventional 5-stage architecture (BW, RC, VA, SA, ST) discussed in class. The network uses credit-based flow control. Assume that credit updates are generated in the same cycle as switch traversal and take a cycle to process when received.

c) How many flits are there in each network message? [3 points]

d) How many buffers are required for each virtual channel to avoid stalls due to a lack of flow control credits for flits that encounter no contention? Explain your reasoning. [5 points]

e) Suppose we wish to use a deterministic minimal routing algorithm on this topology. Briefly describe a strategy to make this routing algorithm deadlock free . [4 points]

f) Consider a uniform random traffic pattern, that is, one where every node transmits a packet to every other node (including itself) with equal probability. What is the average number of hops a message must traverse? [4 points]

g) Describe a scenario where non-minimal routing may be advantageous over minimal routing. [4 points]

3) GPUs [12 points]

a) Briefly describe how the strategy for dealing with long-latency memory accesses differs in a GPU from an out-of-order CPU. [6 points]

b) Why is memory management currently more difficult in applications that use a GPU to accelerate computational kernels? [6 points]

4) Memory Consistency [35 points]

a) List two optimizations (hardware or software) enabled by the DRF0 programming model that are disallowed under (non-speculative implementations of) SC. [4 points]

b) Briefly define store atomicity. [3 points]

c) Propose a mechanism to achieve store atomicity in a processor with an **update-based** coherence protocol and an unordered point-to-point interconnect (i.e., one in which messages may be arbitrarily reordered). [4 points]

c) Assume the following program segments are executed on three processors of a multiprocessor machine. Initially before execution, all variables are equal to 0. *(Thanks to Prof. Narayanasamy for this question).* [4 points]

P1	**P2**	**P3**
A=1	u = A	v = B
	B = 1	w = A

i) What are the possible final states for u, v and w under a sequentially consistent model?

ii) What are the final states that are NOT possible under SC?

d) Conventional high-performance out-of-order processors, such as the MIPS R10K, allow load instructions in the re-order buffer to execute speculatively out of order with respect to one another and with respect to store instructions. (i) How do out-of-order processors like the R10K ensure sequentially consistent execution? (ii) What hardware structures are required? (iii) Briefly describe a scenario where a misspeculation occurs. (Be sure your answer addresses all three parts (i)-(iii)). [10 points]

e) Rather than using complex hardware to detect memory ordering misspeculations, one simple way to enforce sequential consistency with high performance is to execute each load instruction twice: first, execute loads out-of-order as their addresses are calculated (as in your answer to part (c)), and then a second time using the naive method (as in your answer to part (a)). If both load operations return the same value, then sequential consistency has been maintained (i.e., the early out-of-order load did not race with a conflicting store). This approach is called "value-based memory ordering."

However, the downside of value-based memory ordering is that, because all loads execute twice, the L1 cache access bandwidth is doubled. Fortunately, several filtering mechanisms can identify loads that do not need to be re-executed, because they could not possibly have caused a memory ordering violation; these filtered loads are executed only once.

Propose a design for such a filtering method. In your answer, be sure to include: [10 points]

- A description of which loads your filtering mechanism will not re-execute
- An explanation of why the filtered loads do not need to be re-executed (i.e., how do you know they couldn't have exposed a memory ordering violation.)
- An argument that your mechanisms filters out a substantial number of load re-executions (you don't need to filter a majority, but you should argue you are capturing some kind of common case).
- A diagram and description of any new hardware structures you add and/or description of new software or compiler support you require