EECS 570 Final Exam - SOLUTIONS
Winter 2012

Name: ____________________________________ unique name: _______________

Sign the honor code:

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

Scores:

<table>
<thead>
<tr>
<th>#</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/ 20</td>
</tr>
<tr>
<td>2</td>
<td>/ 20</td>
</tr>
<tr>
<td>3</td>
<td>/ 24</td>
</tr>
<tr>
<td>4</td>
<td>/ 18</td>
</tr>
<tr>
<td>5</td>
<td>/ 18</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>/ <strong>100</strong></td>
</tr>
</tbody>
</table>

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. 18 minutes per problem).
- There are 8 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) Speculative Multithreading [20 points]

a) How does speculative multithreading differ from transactional memory? [6 points]

Speculative multithreading uses speculative parallel execution of regions of a sequential program, and maintains the sequential semantics of the overall execution. Transactional memory provides isolated execution of code regions in an explicitly parallel program.

b) Describe two key roles the compiler plays in the Multiscalar design. [8 points]

1) The compiler identifies spawn points for speculative threads

2) The compiler identifies the register def and use masks for each thread, and indicates the last write to each register within each thread.

c) What is the purpose of the address resolution buffer (ARB) in the Multiscalar system. [6 points]

The ARB checks for memory access ordering violations among threads. In the event of an ordering violation, speculative threads are squashed and re-executed. In effect, it acts as a load-store queue for the entire multiscalar execution.
2) Ring Interconnects [20 points]

a) What is the average latency in a unidirectional ring with N nodes? [4 points]

Latency - under uniform random traffic, average message traverses half-way around the ring, hence # hops is N/2.

b) Suppose we make the ring bidirectional. How does this change your answers for part (a). [4 points]

Latency is now N/4.

c) Consider an 8-node bidirectional ring that uses an oblivious routing protocol.

i) What is the worst-case traffic pattern for this topology? Provide the permutation function dest = π(src) for the traffic pattern. (A permutation function provides the destination node id for each message as a function of the source node id). [4 points]

π(s)= s + 3 mod 8

ii) Describe an oblivious routing algorithm that achieves the best possible throughput for this traffic pattern. [4 points]

5/8 traffic goes the short way around the ring, 3/8 goes the long way.

iii) What is that throughput (expressed as a fraction of the maximum link bandwidth)? [4 points]

γ_{max} = 5/8 * 3 hops the short way, 3/8 * 5 hops the long way; = 15/8 on all links
Φ(s)= b / γ_{max} = 8/15 = 0.53
3) **Routing & Flow Control [24 points]**

a) Can a network with virtual cut-through routing deadlock? Explain why or why not? [6 points]

Yes. Whether or not the network can deadlock is a property of the routes packets may take; it is possible to construct a circular dependency among buffers whether a network uses store-and-forward, virtual cut-through, or wormhole routing. Consider the case of a square network with 4 nodes and only one buffer per node. If all buffers are occupied and all packets want to traverse clockwise, we have a deadlock.

b) State an advantage of on-off flow control relative to credit-based flow control. [6 points]

On-off flow control generates fewer flow-control related messages; credit-based flow control generates credit messages as each flit is transmitted; on-off flow control generates flow control updates only when high/low water marks are crossed.
c) Consider a 16-node 2D torus topology that uses minimal deterministic routing. Assume the routers are optimized with lookahead routing, speculative virtual channel allocation, and bypassing when input buffers are empty. Further assume that the link between routers is pipelined and takes 2 cycles to traverse. What is the longest a flit can take to traverse the network if it is injected by itself (i.e., it encounters no congestion). Be sure to include time spent in the router at the ingress and egress nodes (i.e., a flit starts in the first stage of the router at the ingress node, and arrives in the cycle after switch traversal at the egress node). [6 points]

The packet must traverse 4 links and 5 routers. Each router requires 2 cycles, and each link require 2, for a total of 18 cycles.

d) State an advantage and a disadvantage of deflection (a.k.a hot potato) routing. [6 points]

Advantage: can route around congestion. Deadlock free provided there is enough total buffering for the maximum # of outstanding packets and some mechanism guarantees eventual delivery

Disadvantage: packets follow non-minimal routes. Complex to implement.
4) **GPUs [18 points]**

a) Why is a branch that is 95% taken and 5% not taken problematic under the SIMT threading model of GPUs? [6 points]

   A branch that is taken only 5% of the time will frequently cause a group of threads to break into warps where only a single thread executes a different path than the remaining threads.

b) How does a GPU hide the long latency of accesses to global memory? [6 points]

   While waiting for a memory access, the GPU switches to execute a different group of threads.

c) Briefly explain why CUDA code optimized for one GPU might run inefficiently when executed on the next generation GPU. [6 points]

   The number of threads, number of registers, and size of the various GPU storage areas tends to change each generation, which requires re-optimization of the CUDA code to decide how the work should be divided among threads and thread groups.
5) Memory Consistency [18 points]

a) Suppose the following loads and stores are executed on a system that implements the Release Consistency (RC_sc) memory model. The numbers in parenthesis indicate [4 points]

Load X
(1) Load.acquire Lock
(2)
Store Y
(3) Store.release Lock
(4)
Store Z

For each of the three accesses to X, Y, and Z, list all of the numbered points in the execution where those operations can legally be executed. (In other words, indicate if it is legal to re-order each of those three instructions such that they occur at each of the numbered points relative to the other operations).

Load X: (1) (2) (3)

Store Y: (2) (3)

Store Z: (2) (3) (4)

b) Briefly explain why store atomicity is difficult to maintain in update-based coherence protocols. [4 points]

To get store atomicity in an update protocol, you need to use two-phase commit to ensure that the new value has been communicated to all processors before any processor may read the value.
c) Describe the design of a store buffer for a processor that implements a relaxed memory consistency model like Sun’s RMO (i.e., it uses fence instructions to enforce ordering). Draw a picture of your design. Be sure to state (i) under what circumstances a store instruction may drain from the store buffer into the cache hierarchy, and (ii) how fence instructions are implemented. [10 points]