Computer Architecture and Engineering

CS152 Quiz #5

April 29, 2010

Professor Krste Asanovic

Name:____ANSWER KEY_____

This is a closed book, closed notes exam.

80 Minutes

10 Pages

Notes:

- **Not all questions are of equal difficulty, so look over the entire exam and budget your time carefully.**
- **Please carefully state any assumptions you make.**
- **Please write your name on every page in the quiz.**
- **You must not discuss a quiz's contents with students who have not yet taken the quiz. If you have inadvertently been exposed to the quiz prior to taking it, you must tell the instructor or TA.**
- **You will receive no credit for selecting multiple-choice answers without giving explanations if the instructions ask you to explain your choice.**

Writing name on each sheet	**_____**	**1 Point**
Question 1	**_____**	**18 Points**
Question 2	**_____**	**15 Points**
Question 3	**_____**	**20 Points**
Question 4	**_____**	**26 Points**
TOTAL	**_____**	**80 Points**

NAME:_____

Problem Q5.1: Sequential Consistency — 18 points

In this problem, we consider the implementation of sequential consistency (SC) in a cache-coherent multiprocessor that uses a snoopy bus. A processor can intervene in a bus transaction by asserting the *retry* signal, causing the processor that initiated the request to attempt it again later. The bus supports only one outstanding cache miss at a time for the entire system.

Each processor is equipped with a non-blocking data cache that supports *hit-under-miss*: while the cache is processing a miss, accesses that hit in the cache can still proceed. The cache does not support *miss-under-miss*, so it will block in the event of a second miss. A processor sends loads and stores to its cache in program order.

Problem Q5.1.A — 10 points

Without further modifications, the hit-under-miss scheme can lead to violations of sequential consistency. Carefully explain why. Additionally, provide a short pseudocode sequence for two processors that could result in a sequentially inconsistent execution. Indicate whether each data memory access in your example is a cache hit or miss.

Hit-under-miss enables younger memory accesses that hit in the cache to be made visible before an older memory access that misses in the cache. Another processor could see these accesses out of program order, which violates SC.

Assume that variables X and Y start out as 0. Because the write to Y may proceed before the write to X, the following sequence of accesses could result in r1 = 1, r2 = 0, which is clearly a violation of SC.

Processor 0	Hit/Miss	Processor 1	Hit/Miss
___X=1_____	___M___	___r1=Y_____	___M___
___Y=1_____	___H___	___r2=X_____	___H___

NAME:_____

Problem Q5.1.B **8** points

Fortunately, it is possible to make the hit-under-miss scheme appear to be sequentially consistent by leveraging the snoopy bus. Describe how to implement SC while maintaining hit-under-miss.

Reordering only violates SC if other processors can observe it. The only way for e.g. P1 to observe P0's accesses is via a bus transaction. So, if P1 initiates a bus transaction while P0 is servicing a miss that has been hit-under, P0 can tell P1 to retry until the miss has been resolved. This way, the miss and all subsequent hits are made visible at the same time.