CS/ECE 757: ADVANCED COMPUTER ARCHITECTURE II
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON

Prof. Mark D. Hill

Midterm Examination I
In-Class
Wednesday, February 24, 2016
Weight: 25%

1:15 minutes.

CLOSED BOOK, etc., but one cheat sheet allowed (two-sided 8.5x11 page).

The exam is *two-sided* and has **EIGHT** pages, including two blank pages at the end.

Plan your time carefully, since some problems make take longer than others.

NAME: ____

ID# ____

Problem Number	**Maximum Points**	**Actual Points**
1	12	
2	12	
3	12	
4	12	
5	12	
Total	60	

Problem 1: Message Passing (12 points)

(a) Message passing libraries, such as MPI, allow parallel programs to be written for parallel execution even on clusters. For best performance, programmers are often advised to avoid small messages. Why?

(b) What other techniques would you recommend for better message-passing performance? Why?

(c) Message passing libraries, such as MPI, often include alternative SEND routines. MPI_SEND, for example, always sends the value of its buffer argument at the time is was invoked even if the sending thread subsequently writes the buffer. On the other hand, MPI_ISEND permits the sending thread's subsequent buffer modifications to affect the contents to be sent. What are the advantages and disadvantages of these two types of SEND?

Problem 2: Synchronization (12 points)

(a) Provide pseudo-code for a `test-and-test-and-set()` implementation of `Lock(L)` and `Unlock(L)` assuming only a `test-and-set()` hardware primitive. Assume that `test-and-set(L)` sets L to 1 and returns the previous value of L.

(b) What is a sense-reversed `Barrier(B)`? Why might it be better than a barrier implementation that does not use sense reversing?

(c) Discuss at least one good option for implementing a barrier on a very large cache-coherence shared-memory machine.

Problem 3: Consistency (12 points)

Consider the following example. Assume memory variables `data1`, `data2`, `flag1`, and `flag2` are initially `0` and `rij` means processor `Pi`'s register `j`. Assume also that all implementations ensure *write atomicity*, i.e., when a processor's write is seen by *any other* processor, it is seen by *all other* processors.

(a) Will this example behave the same with *total store order (TSO)* (or x86) as with sequential consistency? Why or why not? If not, what exactly should a programmer add to ensure a sequentially consistent execution even on TSO hardware. Why?

```
Processor P1          Processor P2          Processor P3

data1 = 3;            L2: r21 = flag1;      L3: r31 = flag2;
data2 = 4;            if (r21==0)goto L2;if (r31==0)goto L3;
flag1 = 1;            flag2 = 2;            r32 = data1;
                                            r33 = data2;
```

(b) Will this example (same as above) behave the same with *example relaxed consistency* (XC) (or similar model) as with sequential consistency? Why or why not? If not, what are the minimum changes a programmer should make to ensure a sequentially consistent execution even on weaker hardware? Why?

```
Processor P1          Processor P2          Processor P3

data1 = 3;            L2: r21 = flag1;      L3: r31 = flag2;
data2 = 4;            if (r21==0)goto L2;if (r31==0)goto L3;
flag1 = 1;            flag2 = 2;            r32 = data1;
                                            r33 = data2;
```

Problem 4: Snooping Coherence (12 points)

Consider implementing snooping coherence in a system with a single-level of write-back caches. This problem asks you to rank the MOESI states (*modified, owned, exclusive, shared,* and *invalid*) in order of their importance and *to justify your answer*. Assume that your protocol must have the invalid state (I).

(a) If you could only implement two states, which state would you choose to implement along with I. Why?

(b) If you could only implement three states, which state would you add to your answer to (a)? Why?

(c) If you could implement only four states which state would you add to your answer to (b)? Why?

(d) What factors would affect whether you would want to add the fifth MOESI state to your answer to (c)?

Problem 5: Miscellaneous (12 points)

(a) Zhang et al. [MICRO 2015] present COUP with a new *update* coherence state (U)? What is the goal of the U state? What operations can a core perform on a block B in state U? What happens when a core block B in state U seeks to read a word in block B?

(b) What is the data parallel programming model advocated by Hillis and Steele [CACM 1986]? How does it differ from the PRAM model?

(c) In Hydra's thread-level speculation (Hammond et al.), say a write X=20 in iteration 2 of a loop is issued before a write X=10 in iteration 1. What is the final value in memory? How it this ensured?

Scratch Sheet 1 of 2 (in case you need additional space for some of your answers)

Scratch Sheet 2 of 2 (in case you need additional space for some of your answers)