Name:

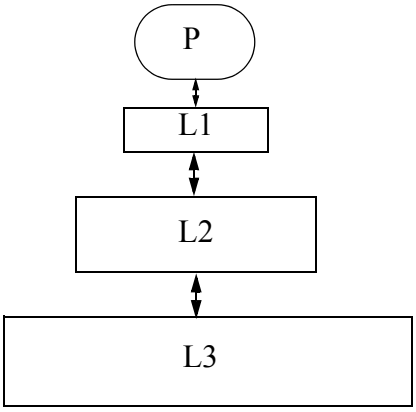
EE666 Mid-Term #2

Please answer all questions. You have 110 minutes starting from now.
Please write clearly and concisely. **Please do not** dump core!

Multi-Level Cache Hierarchy (20 points)

1. In a snoopy bus implementation, is it desirable to maintain inclusion in the cache hierarchy? Why? (5 points)

2. Assume a multi-level cache hierarchy with inclusion depicted in the diagram. Assume a simple 3-state MSI protocol with ReadCached, WriteCached (or dirty), and Invalid states. For a given block X, describe all possible state combinations in the various levels. [Hint, a block may be in either of the states or it may simply not exist in the caches.] (10 points)



Name:

3. For the same hierarchy, what would be your answer to question 2 if the caches did not maintain inclusion? (10 points)

Name:

Livelock, Deadlock, and Starvation (15 points)

1. In shared-memory multiprocessors, there is a phenomenon known as the Window of Vulnerability (WOV) which may result in a livelock. Describe a scenario either in DSM or SMP where WOV indeed results in a livelock. (5 points)

2. Describe a scenario where cache misses in SMP will result in a deadlock. (5 points)

3. Describe a scenario where cache misses in SMP result in starvation. (5 points)

Name:

Name:

Memory Consistency Models (20points)

1. What are the two requirements for memory operations to satisfy sequential consistency? For each requirement, name two ways in which hardware or software may violate the requirement. (10 points)

2. Assume the code fragment below. Assume a single issue out-of-order processor. Assume that remote cache misses take 100 processor cycles and cache hits and ALU operations take a single cycle. Assume that both stores miss and the loads hit in the cache. How long does it take to execute the code fragment for a naïve SC system (i.e., one in which the system naively enforces the model constraints for every load and store)? How long does it take to execute the code fragment for a system implementing Weak Ordering? Explain your answer with the details of how you account for the cycles (10 points)

```
1. store   R1, [R2]
2. load    [R3], R1
3. load    [R4], R2
4. add     R1, R2, R3
5. mul     R1, R2, R5
6. add     R3, R5, R3
7. store    R3, [R4]
```

Name:

Name:

Allocation, Mapping, Transfer and Access Control (20 points)

1. Compare and contrast data mapping, storage allocation, and transfer in each of CC-NUMA, Simple COMA, and page-based software DSM. Describe one advantage and one disadvantage of each system with regards to mapping, allocation, and transfer (10 points)

Name:

2. Compare and contrast access control mechanisms in each of CC-NUMA, Simple COMA, and page-based software DSM. Describe what the minimum hardware requirements are for implementing access control in each system (10 points)

Name:

Directory-Based Systems (15 points)

1. What are full-map directories? What are limited-pointer directories? Name two types of limited-pointer directories and describe how they implement coherence.

Name:

Scalable Architectures (5 points)

1. Name the fundamental protection mechanism (available in all commodity systems) that Princeton SHRIMP relies on to implement protected communication with no software overhead in the common case.

Application-Specific Protocols (5 points)

1. Describe the key motivation behind application-specific coherence protocols. How is a shared-memory application with such a protocol different from a message-passing application?

Name: