Important Issues

Cache coherency:

- its definition
- the hardware support
- how invalidation-based protocols work
- how coherency protocols match or take advantage of the MP design
- how directories work

Adding to our knowledge:

- a 4th type of miss (coherency misses)
- a 3rd locality (processor)
- a 2nd application of snooping (bus-based coherency protocol)
- a 2nd use of sub-block placement

Important Issues

Anything in red or green:

- 2 bus protocols
- inclusion property
- UMA vs. NUMA
- role of local, home, remote nodes
- bus vs. multipath
- snooping vs. directory
- snooping in a coherency protocol vs. snooping in Tomasulo's algorithm
- false sharing: why it occurs, what makes it worse, how to fix

Apply What You Know

A different 4th state:

- what triggers state transitions
- what are the state changes, given a sequence of memory operations

A protocol that isn't based on invalidations:

- what triggers state transitions
- what are the state changes, given a sequence of memory operations

Apply What You Know

Example:

Assume you have a 4-state, write-invalidate protocol, in which three of the states are those used in the baseline 3-state protocol we studied in class and the fourth state is a new one, called *private clean*. A private clean state means that there is only one cached copy of the data, and that it is a read-only copy (i.e., it has thesame value as its backup in memory). Using this new 4-state coherency protocol, fill in the state values for a single cache block in each of the processors (P0, P1, P2), for each of the memory operations listed in the first column. For this question, you can assume the multiprocessor is bus-based.

Operations	P0	P1	P2
Initially	invalid	invalid	invalid
P1: loads B			
P2: loads B			
P0: stores B			
P1: loads B			
P1: stores B			