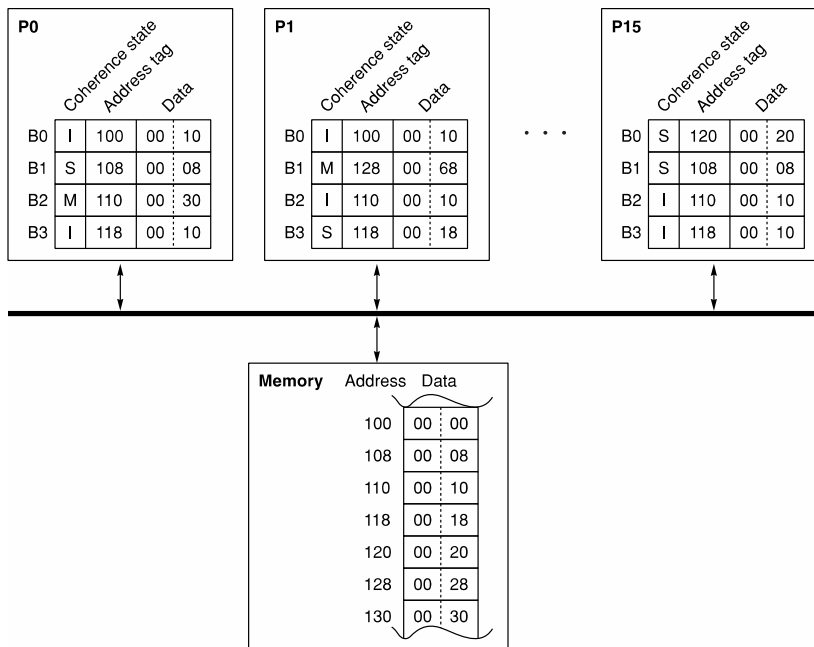


55:132/22C:160, HPCA  
 Spring, 2011  
 Sixth (and final) Homework Assignment

Due Date: Tuesday, May 3, In class.

The simple, bus-based multiprocessor illustrated below represents a commonly implemented symmetric shared-memory architecture. Each processor has a single, private cache with coherence maintained using the three state snooping coherence protocol discussed in lecture and the text. Each cache is direct-mapped, with four blocks each holding two words. To simplify the illustration, the cache address tag contains the full address and each word shows only two hex characters, with the least significant word on the right. The coherence states are denoted M, S, and I for Modified, Shared, and Invalid. (State M corresponds to the “Exclusive” state in the FSM used in lecture)



Bus-based snooping multiprocessor

1. For each part of this exercise below, assume the initial cache and memory state shown above. Each part of this exercise specifies a sequence of one or more CPU operations of the form:

P#: <op> <address> [ <-- <value> ]

where P# designates the CPU (e.g., P0), <op> is the CPU operation (e.g., read or write), <address> denotes the memory address, and <value> indicates the new word to be assigned on a write operation.

Treat each action below as independently applied to the initial state as given above. What is the resulting state (i.e., coherence state, tags, and data) of the caches and memory after the given action? Show only the blocks that change, for example, P0.B0: (I, 120, 00 01) indicates that CPU P0's block B0 has the final state of I, tag of 120, and data words 00

and 01. Also, what value is returned by each read operation?

- a. P15: read 118
- b. P15: write 100 <-- 48
- c. P15: write 118 <-- 80
- d. P15: write 108 <-- 80
- e. P15: read 110
- f. P15: read 128
- g. P15: write 110 <-- 40

2. Many snooping coherence protocols have additional states, state transitions, or bus transactions to reduce the overhead of maintaining cache coherency.

A common protocol optimization is to introduce an Owned state (usually denoted O). The Owned state behaves like the Shared state, in that nodes may only read Owned blocks. But it behaves like the Modified state, in that nodes must supply data on other nodes' read and write misses to Owned blocks. A read miss to a block in either the Modified or Owned states supplies data to the requesting node and transitions to the Owned state. A write miss to a block in either state Modified or Owned supplies data to the requesting node and transitions to state Invalid. This optimized MOSI protocol only updates memory when a node replaces a block in state Modified or Owned. Draw new protocol diagrams with the additional state and transitions.

3. Some applications read a large data set first, then modify most or all of it. The base MSI coherence protocol will first fetch all of the cache blocks in the Shared state, and then be forced to perform an invalidate operation to upgrade them to the Modified state. The additional delay has a significant impact on some workloads.

An additional protocol optimization eliminates the need to upgrade blocks that are read and later written by a single processor. This optimization adds the Exclusive (E) state to the protocol, indicating that no other node has a copy of the block, but it has not yet been modified. A cache block enters the Exclusive state when a read miss is satisfied by memory and no other node has a valid copy. CPU reads and writes to that block proceed with no further bus traffic, but CPU writes cause the coherence state to transition to Modified. Exclusive differs from Modified because the node may silently replace Exclusive blocks (while Modified blocks must be written back to memory). Also, a read miss to an Exclusive block results in a transition to Shared, but does not require the node to respond with data (since memory has an up-to-date copy).

Draw new protocol diagrams for a MESI protocol that adds the Exclusive state and transitions to the base MSI protocol's Modified, Shared, and Invalidate states. (Note that the state that is called "Exclusive" in the lecture notes and figures 4.6 and 4.7 in the text is actually the M (Modified) state in this protocol.)