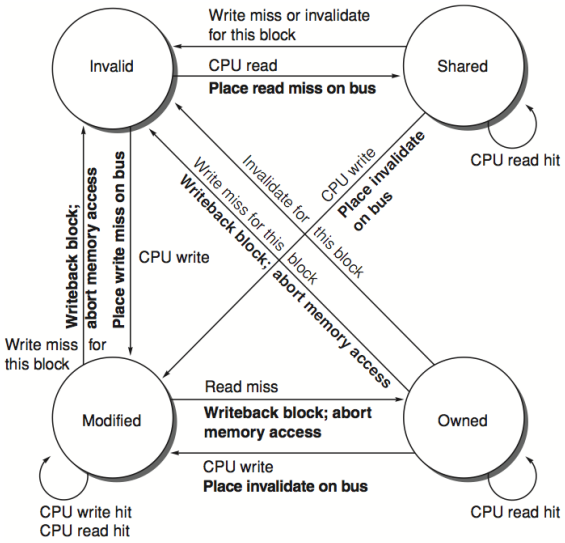
CSE 490/590 Computer Architecture
Homework 3

1. (Textbook Exercise 4.3, p.267) Many snooping coherence protocols have additional states, state transitions, or bus transactions to reduce the overhead of maintaining chache coherency. In Implementation 1 of Exercise 4.2, misses are incurring fewer stall cycles when they are supplied by cache than when they are supplied by memory. Some coherence protocols try to improve performance by increasing the frequency of this case.

   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 hehaves 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.

   Answer:



2. (Textbook Exercise 4.17, p.278) Directory protocols are more scalable than snooping protocols because they send explicit request and invalidate messages to those nodes that have copies of a block, while snooping protocols broadcase all requests and invalidates to all nodes. Consider the 16-processor system illustrated in Figure 4.42 and assume that all caches not shown have invalid blocks. For each of the sequences below, identify which nodes receive each request and invalidate.

   a. P0: write $110 <--$ 80
   b. P0: write $108 <--$ 88
   c. P0: write $118 <--$ 90
   d. P0: write $128 <--$ 98

   Answer:

   a. No messages, hits in P0's cache
   b. Send invalidate to P15
   c. Send invalidate to P1
   d. Send fetch/invalidate to P1

3. (Textbook Example p.219) Assume that words x1 and x2 are in the same cache block, which is in the shared state in the caches of both P1 and P2. Assuming the following sequence of events, identify each miss as a true sharing miss, a false sharing miss, or a hit. Any miss that would occur if the block size were one word is designated a true sharing miss.

Time	P1	P2
1	Write x1	
2		Read x2
3	Write x1	
4		Write x2
5	Read x2	

Answer: Please refer to the textbook.

4. Consider the following sequences of pseudo-instructions. In the following, r is a register and x is a memory location. Assume that the data inside of x contains 0 initially.

P1	P2	P3
st x, 1	r = ld x	st x, 6
r = ld x	r = add r, 1	r = ld x
r = add r, r	st x, r	r = add r, r
st x, r	r = ld x	st x, r
	r = add r, r	
	st x, r	

Can x hold the value of 4 after all execution is done? or 5? or 6?

Answer:

Value 4: Yes
(Sequence: P1's 1st, P2's 1st - 3rd, P1's 2nd - 3rd, P2's 4th - 6th, P3's 1st-4th, & P1's 4th)

Value 5: No because all results must be even

Value 6: Yes
(Sequence: P3's 1st - 4th, P1's 1st - 4th, P2's 1st - 6th)

5. Assume that you have a two-processor SMP machine with write-through caches. Consider the following two thread running concurrently on the two processors.

P1	P2
st x, 1	ld y, r1
st y, 11	st y', r1
	ld x, r1
	st x', r1

Under what circumstance(s) can the execution of these threads satisfy sequential consistency?

Answer: One possible scenario: P1 & P2 have never access x & y before; and the execution sequence is P1's 1st - 2nd & P2's 1st - 4th.