Computer Design and Organization

Assignment #6

Due: Monday December 9

The purpose of this last assignment is to sharpen your understanding of snoopy cache coherence protocols. Once again, you can do this homework by yourself or with a partner with whom you have not worked with before.

1. Problem 6.5. (this should not be too difficult since a solution is in the slides!). Explain what happens on the sequence:

	Processor P1	Processor P2	Processor P3
Time 1	Read A	-	-
Time 2	-	Read A	-
Time 3	-	-	Read A
Time 4	Read B	-	-
Time 5	-	Write A	-
Time 6	-	-	Read A
Time 7	-	Read B	-
Time 8	-	-	Read B

   where A and B are two different memory lines mapping to the same cache line (assume that the caches are direct-mapped and therefore a miss to B forces A to be replaced).

   Of particular interest are your choices for:

   - Who produces the value of A at time 2 for P2?
   - When is the dirty value of A written back to memory? (Note the same value can be written back more than once but not necessarily in this example.)

2. Problem 6.6

3. In this exercise, you have to complete the description of the following snoopy cache coherence protocol.

   There are 3 possible states for a cache block:

   - *Valid Exclusive*. The cache with the block in this state has the only copy of the block and memory contents for that block are consistent with the cache's copy.
   - *Shared*. It is possible that other caches have a copy; Memory contents for that block are consistent with the caches copy.
   - *Dirty*. The cache with the block in this state has the only copy of the block and memory contents for that block are NOT consistent with the cache's copy.

   Note that there is no Invalid state. For this protocol to function, the bus has an extra control line called *MShare* that is raised by the cache controllers when they detect that a bus transaction is for some block present in their cache.

The basic idea is that when writing occurs to a Shared block, the writing is transmitted to other blocks and to main memory.

The protocol is as follows on a Write hit, say in cache X.

- If the block is in state *Dirty*, the write hit proceeds without any other action.
- If the block is in state *Valid Exclusive*, the write hit proceeds and the state becomes *Dirty*.
- If the block is in state *Shared*, the block is written to main memory. Other caches snoop on this transaction. If a copy of the block exists in other caches (at least one) all these caches grab a copy of the block and raise the *MShare* line. If X sees that the *MShare* line has been raised, it keeps the block in state *Shared*. If the *MShare* line has not been raised, the state of the block in cache X becomes *Valid Exclusive*.

Your assignment is to complete the protocol for Read miss and Write miss. Do not worry about what happens to the block that is replaced (evidently, it will have to be written back to memory only if it is in state *Dirty*).