Assignment 5
Due December 2, 2011

You may type or write your answers by hand. If you write by hand, make sure it is clearly presented. **Do not use pencils but ink**. Please put your name and student ID clearly on the submitted assignment.

You may make use of reasonable assumptions of your own for data that might be missing in the problem texts, provided that they are explicitly and clearly stated.

You may submit a partial answer to a problem. The grading will account for this.

Question 0, Feedback (1 pt extra credit)

How many hours did you spend working on this homework assignment?

Question 1 (10 pts) Snooping Coherence Protocol Transitions

Consider the bus-based multiprocessor illustrated in Figure 1. Each processor has a single, private cache. 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. Coherence is maintained using the snooping coherence protocol illustrated in Figure 4.7 of Hennessy and Patterson (page 215). The coherence states are denoted M, S, and I for Modified, Shared, and Invalid.

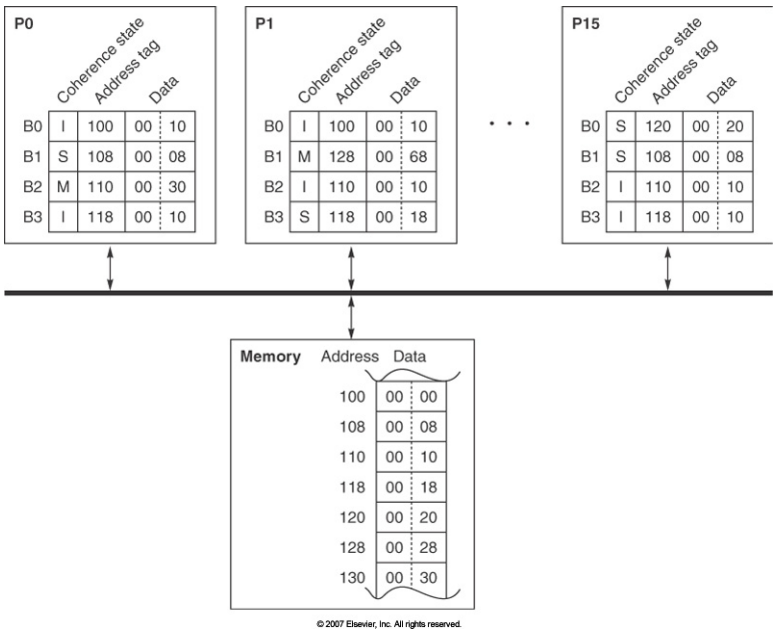

Figure 1: Initial state of a bus-based snooping multiprocessor.

For each part of this problem, assume the initial cache and memory state as illustrated in Figure 1. Each part of this problem specifies a CPU operation in 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 in Figure 1. 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: write 128
g. P15: write 110 <-- 40

Question 2 (10 pts) Coherence Protocol Performance

Now consider the performance of the multiprocessor in Figure 1 under the coherence protocol in Figure 4.7 of Hennessy and Patterson. Assume that:

- CPU read and write hits generate no stall cycles,
- CPU read and write misses generate N_{memory} and N_{cache} stall cycles if satisfied by memory and cache, respectively,
- CPU write hits that generate an invalidate incur $N_{invalidate}$ stall cycles, and
- a write back of a block, either due to a cache replacement or due to a cache supplying a block in response to another processor's request, incurs an additional $N_{writeback}$ stall cycles.

Consider the two implementations with different performance characteristics in Table 1.

Table 1: Snooping coherence latencies

Parameter	Implementation 1	Implementation 2
N_{memory}	100	100
N_{cache}	70	130
$N_{invalidate}$	15	15
$N_{writeback}$	10	10

Now consider the following sequence of operations assuming the initial cache state in Figure 1. For simplicity, assume that the second operation begins after the first completes. For example, consider:

P1: read 110
P15: read 110

Under Implementation 1, the first read generates 80 stall cycles because the read is satisfied by P0's cache. P1 stalls for 70 cycles while it waits for the block, and P0 stalls for 10 cycles while it writes the block back to memory in response to P1's request. Thus the second read by P15 generates 100 stall cycles because its miss is satisfied by memory. Thus this sequence generates a total of 180 stall cycles.

For the following sequences of operations, how many stall cycles are generated by each implementation?

a. P15: read 120
   P15: read 128
   P15: read 130
b. P15: read 118
   P15: write 110 <-- 48
   P15: write 130 <-- 78
c. P15: read 110
   P15: read 108
   P15: read 130
d. P15: read 100
   P15: write 108 <-- 48
   P15: write 128 <-- 78