ENGN 2910A Homework 09 (60 points) – Due Date: Dec 5th 2013

Professor: Sherief Reda

School of Engineering, Brown University

1. [from Debois – 20 points] Assume that a shared-memory multiprocessor using private caches connected to a shared bus uses an MSI cache protocol to maintain cache coherence. The time it takes to carry out various protocol actions are listed in the table below. While a read and write hit take only a single cycle, a read request takes 40 cycles as it has to bring the block from the next level of the cache hierarchy. A bus upgrade request takes less time as it does not involve a block transfer but rather invalidates other shared copies. This action consists of transferring the request on the bus and making a snoop action in each cache; the time for the latter is also given in the table.

Let R_i/M and W_i/M denote a read and write operations, respectively by processor/cache unit i to block M. For the following sequence:

$$R1/X, R2/X, R3/Y, R4/X, W1/X, R2/X, R3/Y, R4/X.$$

Determine how many cycles it takes to carry out all memory requests under the assumption that snoop actions get a higher priority than processor read/write requests from the same unit; they have to wait until the snoop action is done. Assume the tag directory is not duplicated.

B is the block size

Request type	Time to carry out protocol action	Traffic
Read hit	1 cycle	N/A
Write hit	1 cycle	N/A
Read request serviced by next level	40 cycles	6 bytes + B
Read request serviced by private cache	20 cycles	6 bytes + B
Read-exclusive request serviced by next level	40 cycles	6 bytes + B
Read-exclusive request serviced by private cache	20 cycles	6 bytes + B
Bus upgrade/update request	10 cycles	10 bytes
Ownership request	10 cycles	6 bytes
Snoop action	5 cycles	N/A

2. [from Debois – 20 points] Assume a shared-memory multiprocessor with a number of processor/private cache units connected by a shared single-transaction bus. Our baseline cache coherence protocol is a MSI protocol, but we want to investigate what performance gain can be achieved by adding an exclusive state to make it a MESI protocol. We want to determine how many cycles it takes to execute a sequence of accesses with the same assumptions and notations as in the previous problem with a MSI and a MESI protocol. Consider the following sequences of accesses by the processors:

$R1/X, W1/X, W1/X, R2/X, W2/X, W2/X, R3/X, W3/X, W3/X, R4/X, W4/X, W4/X$

a. (14 points) Suppose that a transition from state E to state M brings no access cost. How many cycles does it take to execute the access sequence under MSI vs. MESI? Assume the access costs for the protocol transactions as given in the Table of problem 1.

b. (7 points) Compare the traffic generated by the MSI and MESI protocols counted in bytes transferred assuming that B = 32 bytes.

3. [from Debois – 10 points] Consider a shared-memory multiprocessor that consists of three processor/cache units and where cache coherence is maintained by a MSI protocol. The given table shows the access sequence taken by three processors to the same block but to different variables (A, B, C) in that block.

	Processor 1	Processor 2	Processor 3
1	R_A		
2		R_B	
3			R_C
4	W_A		
5			R_C
6		R_A	
7	W_B		
8			R_A
9			R_B

a. Classify the misses with respect to cold, true sharing, and false sharing misses.

b. Which of the misses could be ignored and still guarantee that the execution is correct?

4. [from Debois] Consider a scalable implementation of a shared-memory multiprocessor using a set of nodes that each contains a processor, a private cache, and a portion of the memory. Cache coherence is maintained using a directory cache protocol, where the directory uses a presence-flag vector associated with each memory block to keep track of which nodes have copies of that block. The time it takes to process a directory request at the home and a remote node is 50 cycles. Further, the latency and traffic of all consistency-induced requests and responses are given in the following table, and the block size is 32 bytes.

a. Determine the number of cycles needed to handle a cache miss when the home node is the same as the requesting node and the memory copy is clean. Also determine the amount of traffic (in bytes) caused by the coherence transaction.

B is the block size

Request type	Time to carry out protocol action	Traffic
Read hit	1 cycle	N/A
Write hit	1 cycle	N/A
BusRd	20 cycles	6 bytes
RemRd	20 cycles	6 bytes
RdAck	40 cycles	6 bytes
Flush	100 cycles	6 bytes + B
InvRq	20 cycles	6 bytes
InvAck	20 cycles	6 bytes
UpgrAck	20 cycles	6 bytes

b. Determine the number of cycles needed to handle a cache miss when the home node is the same as the requesting node and the memory copy is dirty. Also determine the amount of traffic (in bytes) caused by the coherence transaction.

c. Determine the number of cycles needed to handle a cache miss when the home node is different from the requesting node and the memory copy is clean. Also determine the amount of traffic (in bytes) caused by the coherence transaction.

d. Determine the number of cycles needed to handle a cache miss when the home node is different from the requesting node, the memory node, the memory copy is dirty, and the remote node is the same as the home node. Also determine the amount of traffic (in bytes) caused by the coherence transaction.

e. Determine the number of cycles needed to handle a cache miss when the home node is different from the requesting node, the memory copy is dirty, and the remote node is different from the home node (and of course different from the requesting node). Also determine the amount of traffic (in bytes) caused by the coherence transaction.