ECE/CS 570 – High Performance Computer Architecture

Instructor: Prof. Lizhong Chen

Homework #2

Due: Monday, May 13 at 10:30am, Hard copy

1. Consider a shared-memory multiprocessor that consists of three processor/cache units and where cache coherence is maintained by an MSI protocol. Table 2 shows the access sequence taken by three processors to the same block but to different variables (A, B, C) in that block.

Table 1 Timing and traffic parameters for protocol actions (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

Table 2

	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?

(c) Determine the fraction of essential traffic resulting from the access sequence using the parameters in Table 1, and assuming that the block size is 64 bytes.

2. We 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, as shown in Figure 1. 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 and with the protocol according to Figure 2. 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 detailed in Table 3, and the block size is 32 bytes. Note that after a missing block is installed (50 cycles) in the cache, another cycle is needed to read again (which is now a hit).

Table 3 Timing and traffic parameters for protocol actions (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

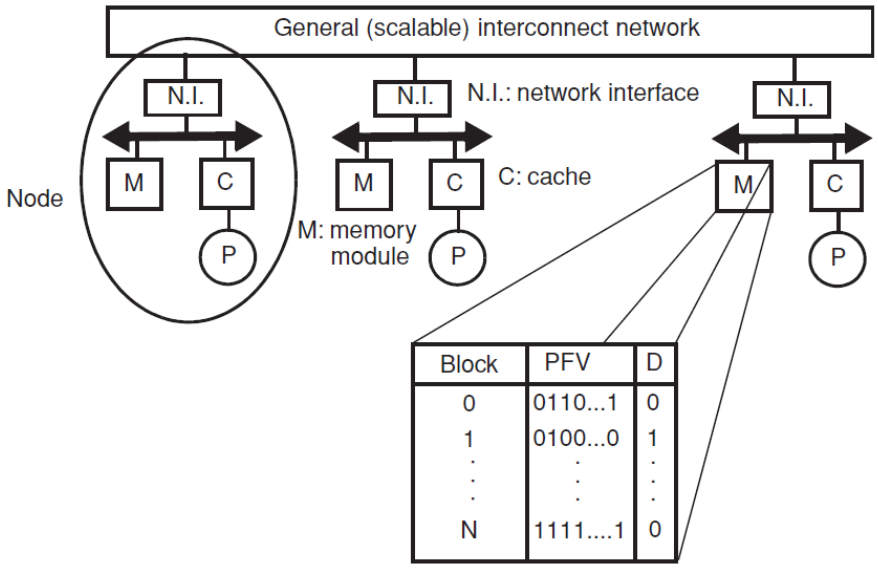

Figure 1 Hardware structures for the presence-flag vector (PFV) directory protocol.

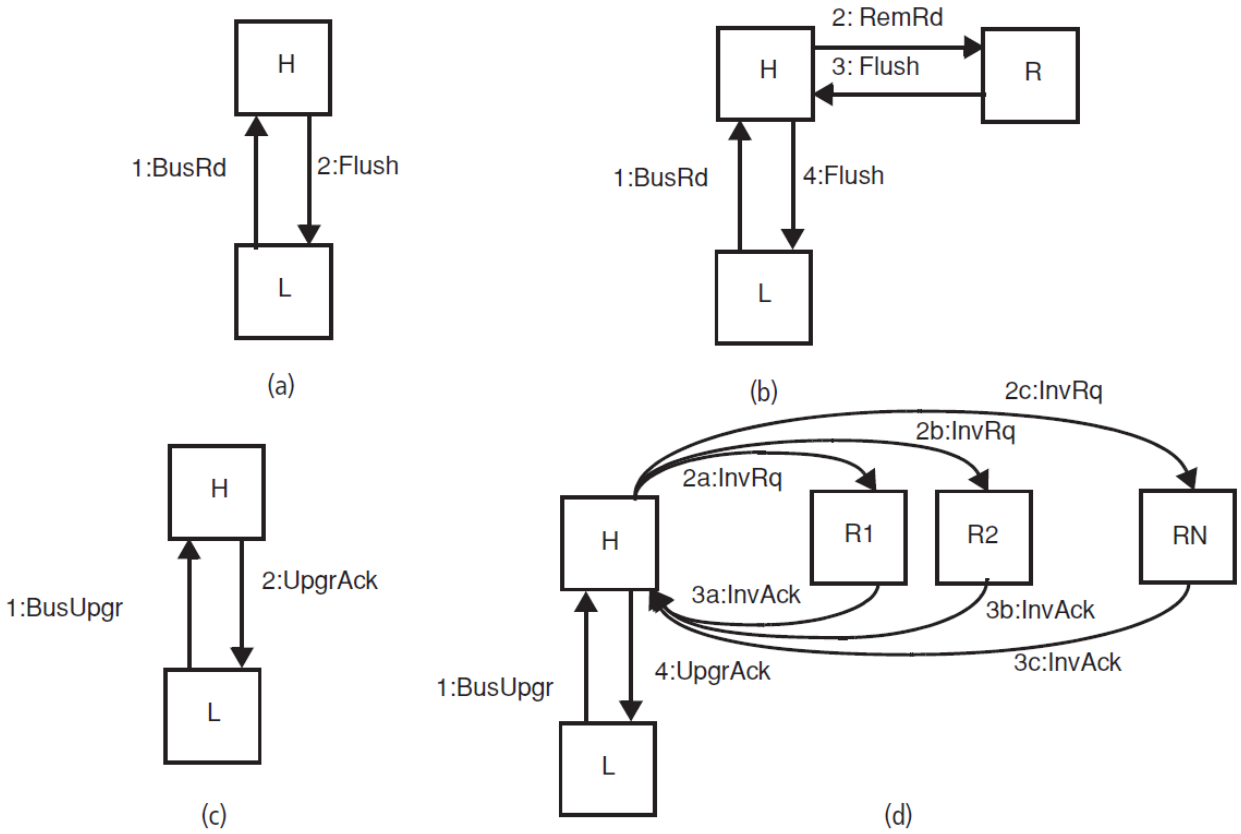

Figure 2 Coherence transactions in the presence-flag vector protocol. (a) BusRd on a clean miss; (b) BusRd on a dirty miss; (c) BusUpgr with single copy at home; (d) BusUpgr with multiple copies.

(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) 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 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.

3. Consider the following program, assuming that A and B are variables in memory initialized to 0, and R1, R2, and R3 are registers:

P1	P2	P3
A=1	R1=A	R2=B
	B=1	R3=A

Please For all Possible R1, R2, R3 outcome combinations, and which outcome(s) is/are not sequentially consistent and why?

4. Consider a bidirectional k-ary n-mesh (i.e., between two neighboring nodes there are two unidirectional channels in the opposite direction). Please calculate the following quantities and show your steps in detail.

(a) The total number of network channels. Count each direction separately if the channels are bidirectional.

(b) Bisection bandwidth (defined as the number of uni-directional channels in the bisection cut).

(c) Maximum channel load, assuming uniform traffic and perfect load balancing across all minimal paths.

5. Prove whether or not the following routing algorithms based on prohibiting dimensional turns are suitable to be used as escape paths for 2D meshes by analyzing whether they are both connected and deadlock-free. Explain your answer. The routing algorithms are expressed with the following abbreviations: W = west, E = east, N = north, and S = south.

(Hint: You may wish to refer to the Turn Model algorithm to prove your answer by drawing a directed graph that depicts dependencies between channels and verifying the channel dependency graph is free of cycles.)

a. Allowed turns are from W to N, E to N, S to W, and S to E.

b. Allowed turns are from W to S, E to S, N to E, and S to E.

c. Allowed turns are from W to S, E to S, N to W, S to E, W to N, and S to W.

d. Allowed turns are from S to E, E to S, S to W, N to W, N to E, and E to N.