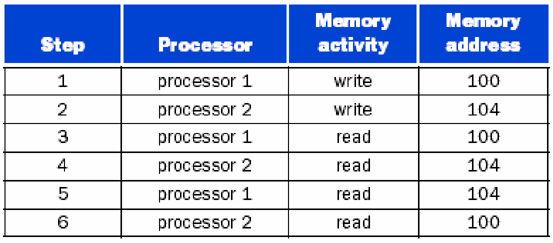
CEG 4131 Assignment 3

MESI cache coherency protocol

As you might imagine, there are many variations on cache coherence that are much more complicated than the simple Write Back- Write Invalidate model. The one found on the Pentium 4 and many other microprocessors is called **MESI**, a write-invalidate protocol whose name is an acronym for the four states of the protocol: Modified, Exclusive, Shared, Invalid. Modified and Invalid are the same as with Write Back- Write Invalidate protocol. The Shared state of Write Back- Write Invalidate protocol is divided, depending on whether there are multiple copies (Shared state) or there is just one (Exclusive state). In either case, memory has an up-to-date version of the data. This extra Exclusive state means there is only one copy of the block, so a write hit doesn't need to invalidate. A write hit to a Shared block of MESI model requires an invalidation, since there may be multiple copies.

Problem 1

Count the number of transactions on the bus for the following sequence of activities involving shared data. Assume that both processors use write-back caches, write-update cache coherency, and a block size of one word. Assume that all the words in both caches are clean.



Step	Processor	Memory activity	Memory address
1	processor 1	write	100
2	processor 2	write	104
3	processor 1	read	100
4	processor 2	read	104
5	processor 1	read	104
6	processor 2	read	100

Problem 2

Two processors require access to the same line of data from data memory. Processors have a cache and use the MESI protocol. Initially both caches are empty.
Figure bellow depicts the consequence of a read of line x by Processor P1. If this is the start of a sequence of accesses, draw the subsequent figures for the following sequence:

1. P2 reads x.
2. PI writes to x (for clarity, label the line in PI's cache x').
3. PI writes to x (label the line in PI's cache x").
4. P2 reads x.

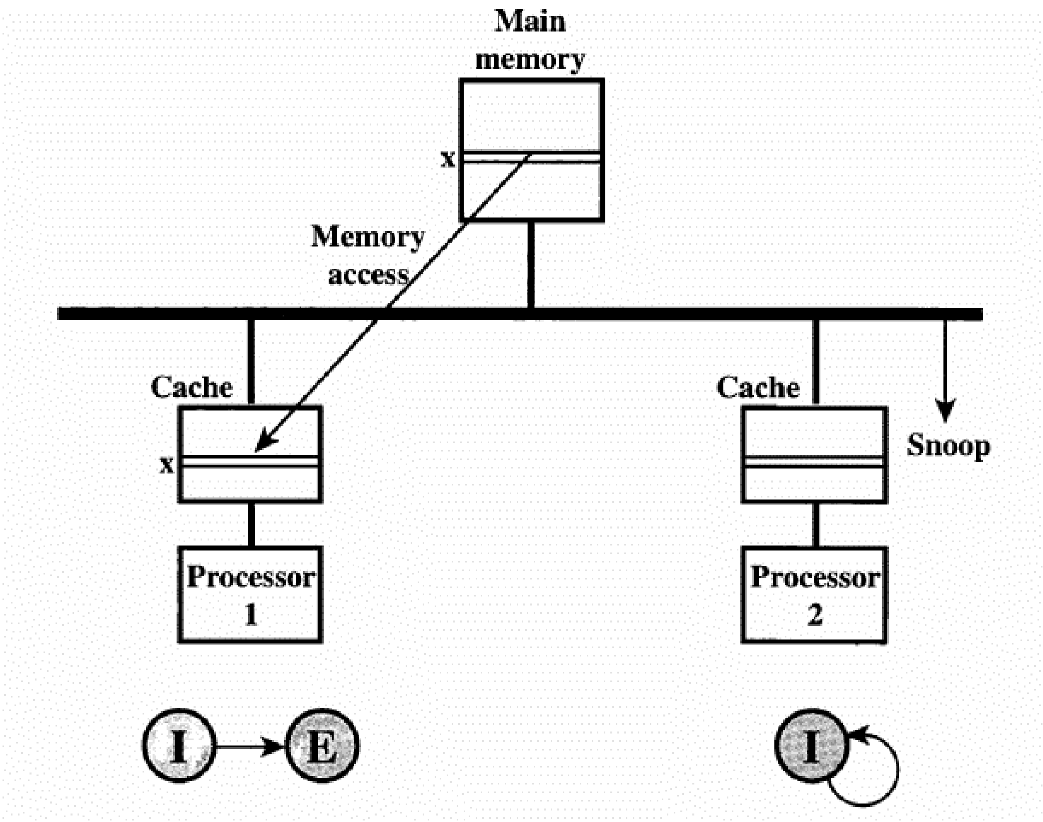
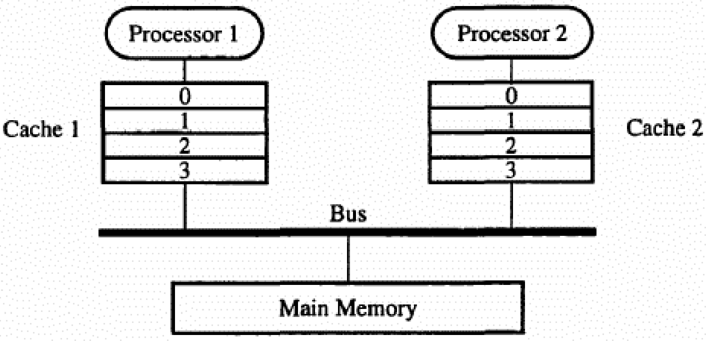


Figure 1 Processor 1 reads line x.

Problem 3
Deduce and explain protocols at slides 4 and 5 of the file CEG4131_shared_memory_additional.ppt. Compare these protocols to MESI.

Problem 4

Problem 7.3 Consider a dual-processor (P1 and P2) system using write-back private caches and a shared memory, all connected to a common contention bus. Each cache has four block frames labeled below as 0, 1, 2, 3.



The shared memory is divided into eight cache blocks as 0, 1, ..., 7. To maintain cache coherence, the system uses a three-state (RO, RW, and invalid) snoopy protocol based on the write-invalidate policy described in Fig. 7.12b.

Assume the same clock drives the processors and the memory bus. Within each cycle, any processor can submit a request to access the bus. In case of simultaneous bus requests from both processors, the request from P1 is granted and P2 must wait one or more cycles to access the bus.

In all cases, the bus allows only one transaction per cycle. Once a bus access is granted, the transaction must be completed before the next request is granted. When there is no bus contention, memory-access events from each processor may require one to two cycles to complete, as specified below separately:

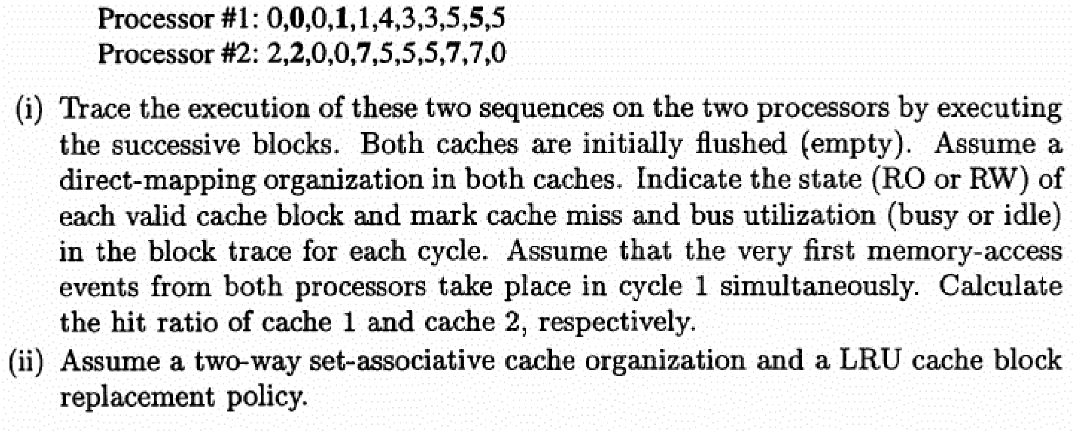
- Read-hit in cache requires one cycle and no bus request at all.
- Read-miss in cache requires two cycles without contention: one for block fetch and one for CPU read from cache.
- Write-hit requires one cycle for CPU write and bus invalidation simultaneously.
- Write-miss requires two cycles: one for block fetch and bus invalidation, and one for CPU write.
- Replacement of a dirty block requires one cycle to update memory via the bus.

(a) In the case of bus contention, one additional cycle is needed for bus arbitration in all the above cases except a read-hit.

(i) Show how to map the eight cache blocks to four cache block frames using a direct-mapping cache organization.

(ii) Show how to map the eight cache block frames using a two-way set-associative cache organization.

(b) Consider the following two asynchronous sequences of memory-access events, where boldface numbers are for write and the remaining are for read.



Processor #1: 0,**0**,0,**1**,1,4,3,3,5,**5**,5
Processor #2: 2,**2**,0,0,**7**,5,5,5,**7**,7,0

(i) Trace the execution of these two sequences on the two processors by executing the successive blocks. Both caches are initially flushed (empty). Assume a direct-mapping organization in both caches. Indicate the state (RO or RW) of each valid cache block and mark cache miss and bus utilization (busy or idle) in the block trace for each cycle. Assume that the very first memory-access events from both processors take place in cycle 1 simultaneously. Calculate the hit ratio of cache 1 and cache 2, respectively.

(ii) Assume a two-way set-associative cache organization and a LRU cache block replacement policy.

Problem 5

A common communication pattern in scientific programs is to consider the nodes as elements of a two-dimensional array and then have communication to the nearest neighbor in a given direction. (This pattern is sometimes called *NEWS communication*, standing for north, east, west, and south, the directions on the compass.) Map an eight-by-eight array onto the 64 nodes in each topology, and assume every link of every interconnect is the same speed. How long does it take each node to send one message to its northern neighbor and one to its eastern neighbor? Ignore nodes that have no northern or eastern neighbors.

Problem 6

Draw Omega and crossbar switch network with 8 processing nodes that allows unidirectional connection among these processing nodes.