CS4110: Computer System Design Lab
Assignment #6
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Submission deadline: 13th Nov, 2015

Cache Coherence Protocol Implementation: Consider a bus-based shared memory two-processor system, with 1KB private cache per processor and 32KB main memory. Caches are organised as direct-mapped and the cache block size is 32B. The data bus width is 32-bit and each datum is accessed one word (i.e., 32 bits) at a time.

Each processor performs only two operations: *read* and *write*. Each processor randomly picks a block, i.e., selects a block out of 1024 blocks in the main memory, and reads the data word-by-word from the beginning to the end of the block. When the entire block is read by a processor, it then writes to a random word location in the block. This process is repeated again and again. When both the processors try to access the bus, only one will be given the access permission to place its transaction on the bus. The main memory can serve only one request from a processor at a time.

One way to minimize the number of coherence transactions placed on the bus is to modify the MSI protocol by considering an additional state, *Exclusive* (E). The new protocol is called MESI protocol. When a block B is brought from the memory to a processor cache (C1), if B is not present in the other processor's cache (C2), we can assign state E to block B, indicating that C1 has the exclusive copy of B. Whenever we perform a write to B, which is in state E, we need not place any invalidation transaction on the bus to inform C2, but we change the state of B from E to M. By adding E state, we can eliminate the unnecessary invalidation transactions on the bus.

Simulate the above memory access pattern and record the frequency of state changes and the number of coherence transactions placed on the bus **for both MSI and MESI protocols**.