ASSIGNMENT – 3

Q1: Implement the classical *test-and-set* atomic instruction using *load-linked/store-conditional* instruction pair.

Q2: A system consists of two processors, each having its own private L1 cache of 4 blocks. The variables accessed from these L1 caches are as follows:

Processor P1 - L1 Cache		**Processor P2 - L1 Cache**	
Block Number	Variables	Block Number	Variables
B0	X, Y	B0	X, A
B1	Z	B1	B
B2	W	B2	W
B3	U, V	B3	U, C

The actions by the processors are given below. For each action, identify whether it leads to a *true* or *false* sharing miss or a hit. Assume that initially all of the blocks are in *shared* state.

1. Processor P1 updates the value of X.
2. Processor P2 updates the value of A.
3. Processor P2 updates the value of X.
4. Processor P1 reads the value of X.
5. Processor P2 reads the value of W.
6. Processor P1 reads the value of Z.
7. Processor P2 updates the value of B.
8. Processor P1 updates the value of U.
9. Processor P2 updates the value of C.
10. Processor P1 reads the value of U.

Q3: A system has two processor chips, each containing two cores. The cores have a private L1 cache and share an L2 cache which is on the chip. There is a bus that interconnects the L1 caches to the L2 cache on the same chip. The two chips have distributed memory which is attached to them directly. The two chips are connected on a point-to-point link. The state of the memory locations is as shown in the figure below. For each of the following CPU operations, give the set of actions that are done and the final state in the L1 caches, directory in L2 and directory in memory for that memory location. As shown in the figure, the L1 caches have 2 blocks and L2 caches have 4 blocks. The distributed memory also has 4 blocks. All the caches are direct- mapped. So, a block in memory location 3, is in the block (3 mod 2) for L1 cache and (3 mod 4) in L2 cache. Thus, the location 220, which is in block 2 in the memory is mapped to location 0 in L1 cache and block 2 in L2 cache.

For each operation, assume that the state it starts with is the initial state given below.

1. P1, Core 0 issues a write to address 102 with data 52.
2. P1, Core 1 issues a read to address 210.
3. P1, Core 0 issues a read to address 120.
4. P0, Core 0 issues a write to 100 with data 24.
5. P0, Core 1 issues a write to address 300 with data 62.

P0, Core 0 L1 cache

	State	Addr	Data
B0	M	100	40
B1	S	102	32

P0, Core 1 L1 cache

	State	Addr	Data
B0	S	220	08
B1	S	102	32

P1, Core 0 L1 cache

	State	Addr	Data
B0	S	220	08
B1	S	102	32

P1, Core 1 L1 cache

	State	Addr	Data
B0	I	100	32
B1	M	300	12

P0 L2 cache

	State	Addr	Data	Sharers
B0	DM	100	32	P00
B1	DS	102	32	P00,P01
B2	DS	220	08	P00
B3	DI			

P1 L2 cache

	State	Addr	Data	Sharers
B0	DI	100		
B1	DS	102	32	P10
B2	DS	220	08	P11
B3	DM	300	48	

Memory 0

Addr	State	Sharers	Data
100	DM	P0	32
102	DS	P0,P1	32
120	DI		13
300	DM	P1	48

Memory 1

Addr	State	Sharers	Data
200	DI		20
210	DI		18
220	DS	P0,P1	08
320	DI		64