Q1. Consider the following code running in two cores. For this question we will use the original MSI protocol.

(1) LW x1, 0(x5)
(2) LW x2, 0(x6)
(3) SW x3, 0(x6)
(4) SW x2, 0(x5)
(5) LW x1, 0(x6)

Do not optimize or re-order the code. The addresses in x5 and x6 map to different cache lines. Assume the following execution sequence:

A.1, A.2, B.1, B.2, A.3, B.3, B.4, B.5 (A and B are the two cores).

A.1 means Core A executes instruction 1, A.3 means Core A executes instruction 3 and so on

a. Write a table such as the one shown below, with the states of the cache lines at every step. Assume they start with an invalid states initially.

b. Which of these load/store requests cause cache to cache transfers of data?

Core and instruction	State of x5 cache line in Core A	State of x5 cache line in Core B	State of x6 cache line in Core A	State of x6 cache line in Core B
A.1 means Core A executes instruction 1				
A.2				
Etc. Add more rows here in your answer script				

Solution 1:

Core and instruction	State of x5 cache line in Core A	State of x5 cache line in Core B	State of x6 cache line in Core A	State of x6 cache line in Core B
A.1 means Core A executes LW x5	S	I	I	I
A.2 Lw x6	S	I	S	I
B1 Lw x5	**S**	**S**	S	I
B2 Lw x6	S	S	**S**	**S**
A3 Sw x6	S	S	M	I
B3 Sw x6	S	S	I	M
B4 Sw x5	I	M	I	M
B5 Lw x6	I	M	I	M

B1 and B2 – the highlighted ones cause cache to cache transfers

Q2: Repeat the same with MESI and MOESI. Which of these protocols reduce the number of bus accesses?