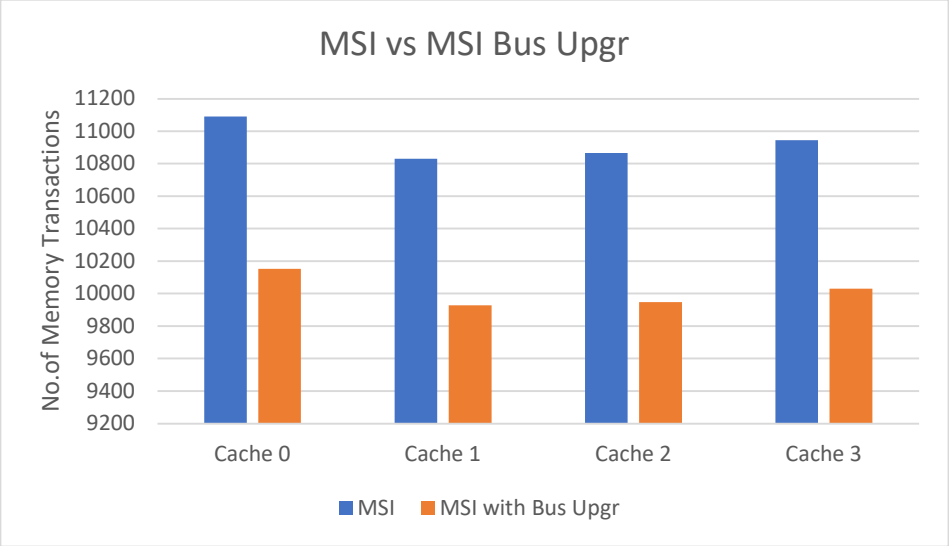
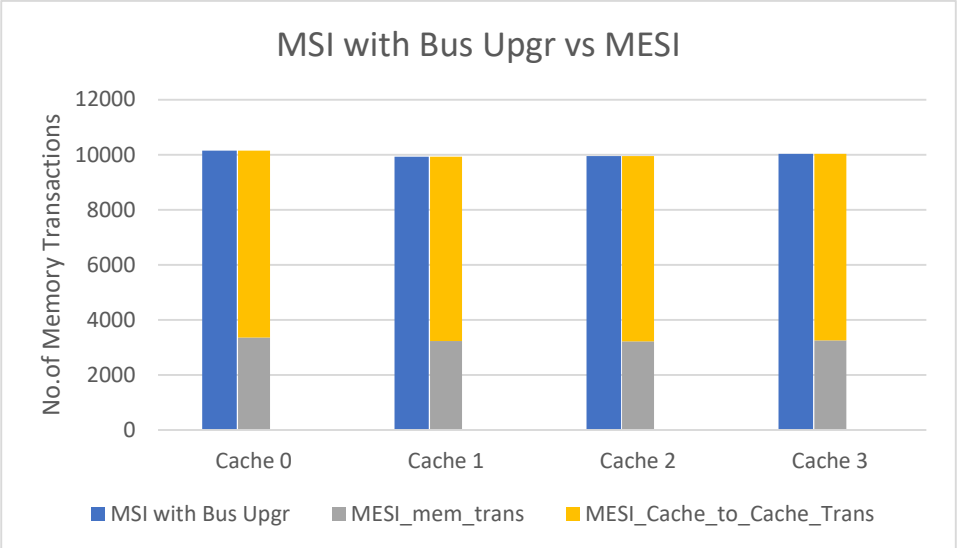
Project 2. Coherence Protocols

1. MSI vs MSI with Bus Upgr:



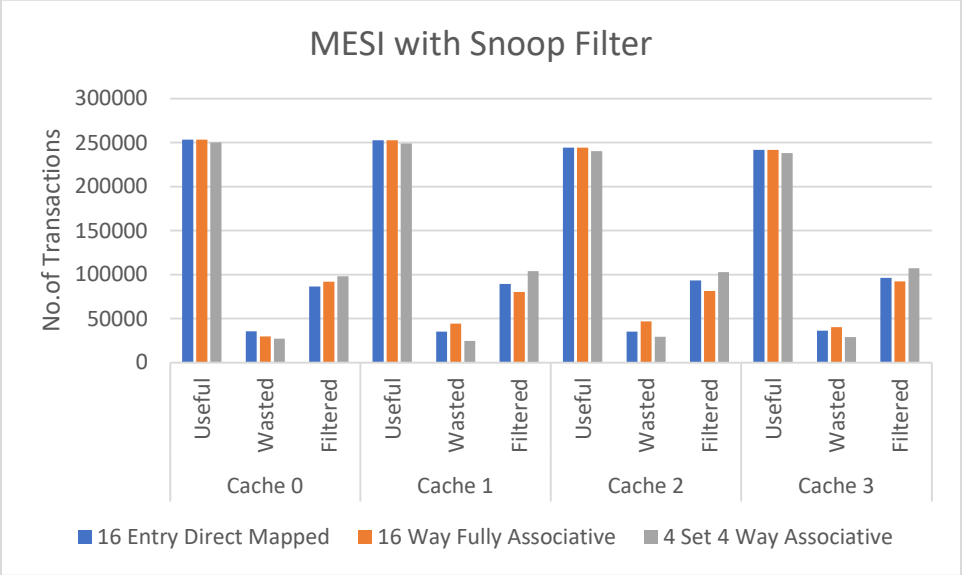
The MSI with bus upgrade has about 8% less memory transaction than MSI because the bus upgrade bus transaction prevents redundant memory transactions. For example, if the a cache line is in shared state and the processor tries to write to this line, then in MSI, the cache controller issues BusRdx upon seeing which, memory controller fetches the block from memory and supplies it on the bus. However, this is not required since the processor already has the block. The BusUpgr transaction differentiates I→M and S→M thereby reducing redundant memory transactions.

2. MSI with Bus Upgr vs MESI



The total number of memory transactions in both MSI BusUpgr and MESI are equal but about 67% of these transactions are cache to cache transactions in case of MESI. So, although the number of transactions are equal, MESI would be faster.

3. MESI Snoop Filter



Cache Config used:

a. 16 Entry Direct mapped filter: Cache Size: 1024, Associativity: 1, Block Size: 64
b. 16 Way fully associative filter: Cache Size: 1024, Associativity: 16, Block Size: 64
c. 4 Set 4 Way associative filter: Cache Size: 4096, Associativity: 4, Block Size: 256

#Useful Transactions: a = b > c

#Wasted Transactions: b > a > c (except in Cache 0)

#Filtered Transactions: c > b > a (except in Cache 0)

Although the number of useful transactions is less compared to configs “a” and “b” (not significantly less), filter config “c” provides best performance by providing maximum filtering and minimum wasted transactions. This is because “c” has minimal conflict and capacity misses compared to other configurations.