

# General Explanation

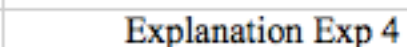
---

- **Advanced protocols (MESI, MOSI, MOESI, MOESIF) with either one or both of exchange state and ownership state always perform better than MSI.** This is because these extra states lead to **lesser coherence misses and better \$-to-\$ transfer.**
- **MESI will always perform either similar to (experiment 1) or better than MSI (experiment 3, 7).** This is because of the **Exclusive** state introduced in MESI. Exclusive state overcomes the drawback of MSI that each read-write sequence incurs 2 bus transactions. This is a huge setback for programs that have little data sharing. Exclusive state distinguishes between a cache block stored in multiple caches and a cache block stored in a single cache. This means that coherence misses reduces as well. This is clearly evident in experiment 3 where because of introduction of E state MESI, MOESI and MOESIF give better performance than MOSI since experiment 3 supports little sharing of data.
- **MOESI will always perform either similar to (experiment 3) or better than MESI (experiment 2).** This is because of the **Owned** state introduced in MOESI. Owned state overcomes the drawback of MESI in that processor in this new state “Owned” can provide modified data to other processors without or even before writing it to the main memory, also popularly known as dirty sharing. This leads to better \$-to-\$ transfer since memory based transfers reduce significantly.
- **MOESIF always performs better than MOESI because of significant increase in \$-to-\$ transfers** as evident all experiments, particularly experiment 6, where all variables are constant except \$-to-\$ transfers. This happens due to **Forward** state in MOESIF where the block in F state elected as the forwarder of data takes advantage of \$ to \$ transfers when there are multiple clean copies of data.

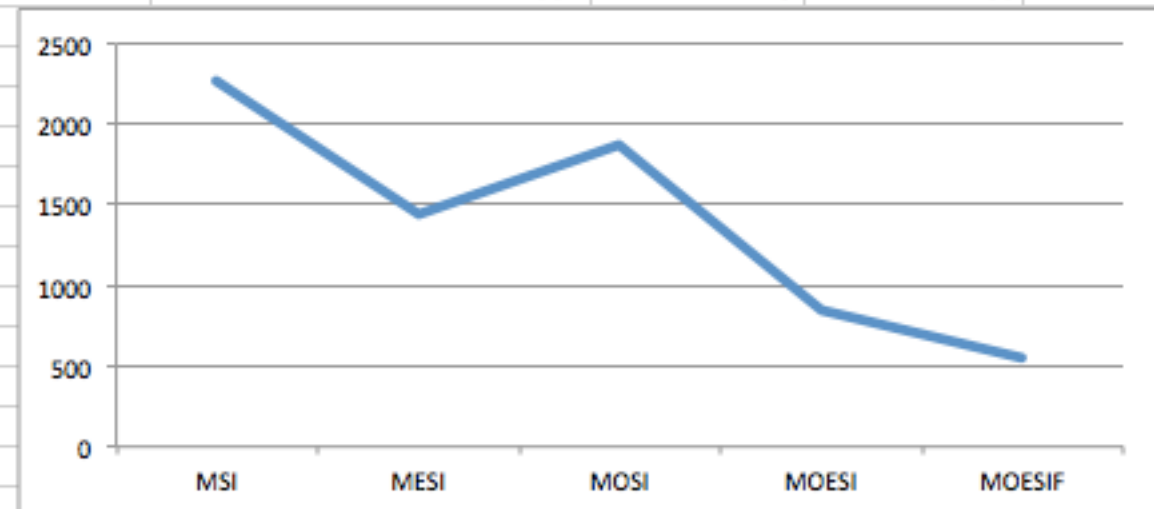


### Explanation Exp 3

- MESI, MOESI and MOESIF performs better than MSI and MOSI. This is because of the Exclusive state introduced in MESI, MOESI and MOESIF Exclusive state overcomes the drawback of MSI that each read-write sequence incurs 2 bus transactions. This is a huge setback for programs that have little data sharing. Exclusive state distinguishes between a cache block stored in multiple caches and a cache block stored in a single cache. This means that coherence misses reduces as well. This is clearly evident in experiment 3 where because of introduction of E state MESI, MOESI and MOESIF give better performance than MOSI since experiment 3 supports little sharing of data.

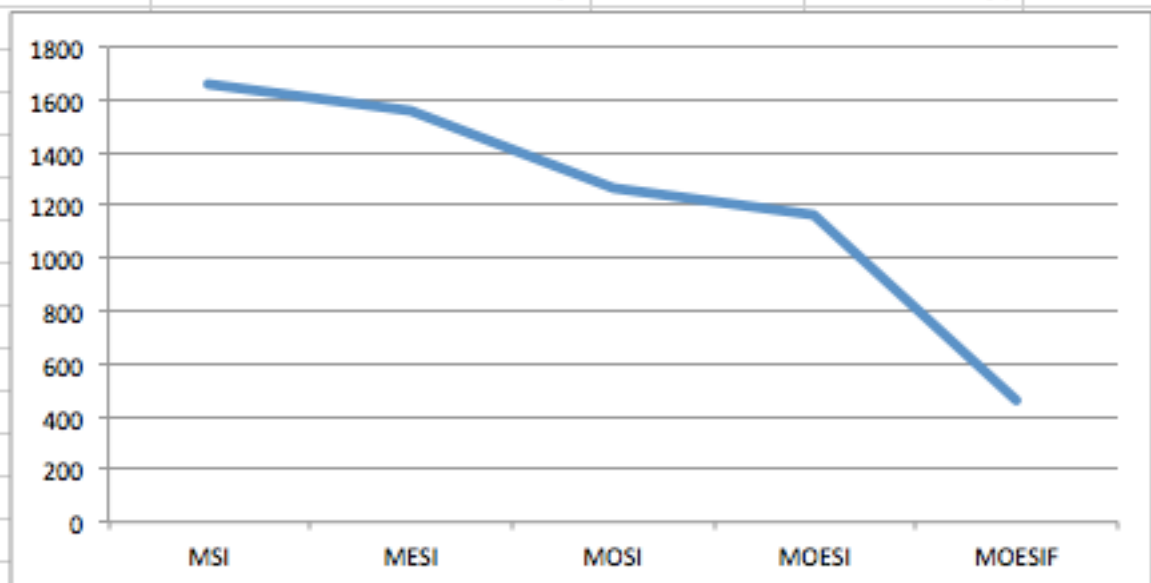


Similar to experiment 3 but also Forward state where the block in F state elected as the forwarder of data takes advantage of S to S transfers when there are multiple clean copies of data.



|              |  |      |    |    |   |    |   |
|--------------|--|------|----|----|---|----|---|
| Experiment 5 |  |      |    |    |   |    | 8 |
| MSI          |  | 1661 | 21 | 37 | 0 | 5  | 8 |
| MESI         |  | 1561 | 21 | 37 | 0 | 6  | 8 |
| MOSI         |  | 1261 | 21 | 37 | 0 | 9  | 8 |
| MOESI        |  | 1161 | 21 | 37 | 0 | 10 | 8 |
| MOESIF       |  | 461  | 21 | 37 | 0 | 17 | 8 |

Explanation Exp 5  
Everything remaining constant, \$-to-\$ transfer due to states such Owned and Exclusive leads to faster execution.



|              |  |      |    |     |    |    |    |
|--------------|--|------|----|-----|----|----|----|
| Experiment 6 |  |      |    |     |    |    | 16 |
| MSI          |  | 7775 | 87 | 747 | 0  | 12 | 16 |
| MESI         |  | 4925 | 62 | 747 | 25 | 15 | 16 |
| MOSI         |  | 6975 | 87 | 747 | 0  | 20 | 16 |
| MOESI        |  | 4125 | 62 | 747 | 25 | 23 | 16 |
| MOESIF       |  | 3125 | 62 | 747 | 25 | 33 | 16 |

Explanation Exp 6  
Results similar to experiment 3, 7

