**Name : ABHISHEK JAIN**

**GTID 902985939**

**6100 ACA**

# PROJECT 3 REPORT

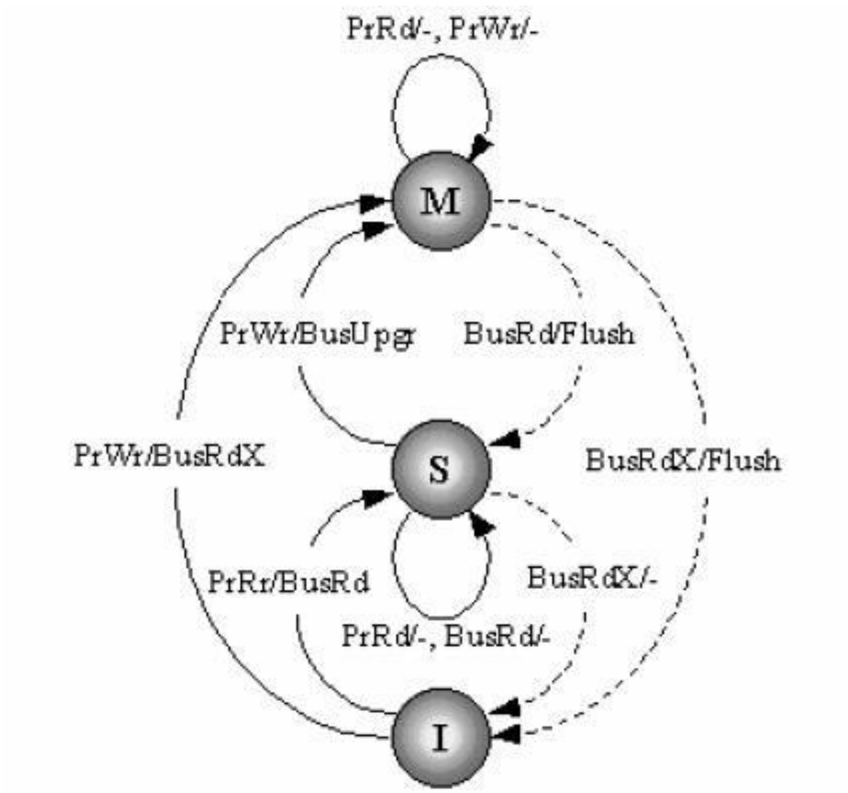
**PART I Validation**

The different protocols namely  MI, MSI, MOSI, MESI, MOESI and MOESIF were implemented Their outputs were then compared to the outputs provided by the instructor. Once the outputs were successfully validated, the next step was to carry out the required experiments.

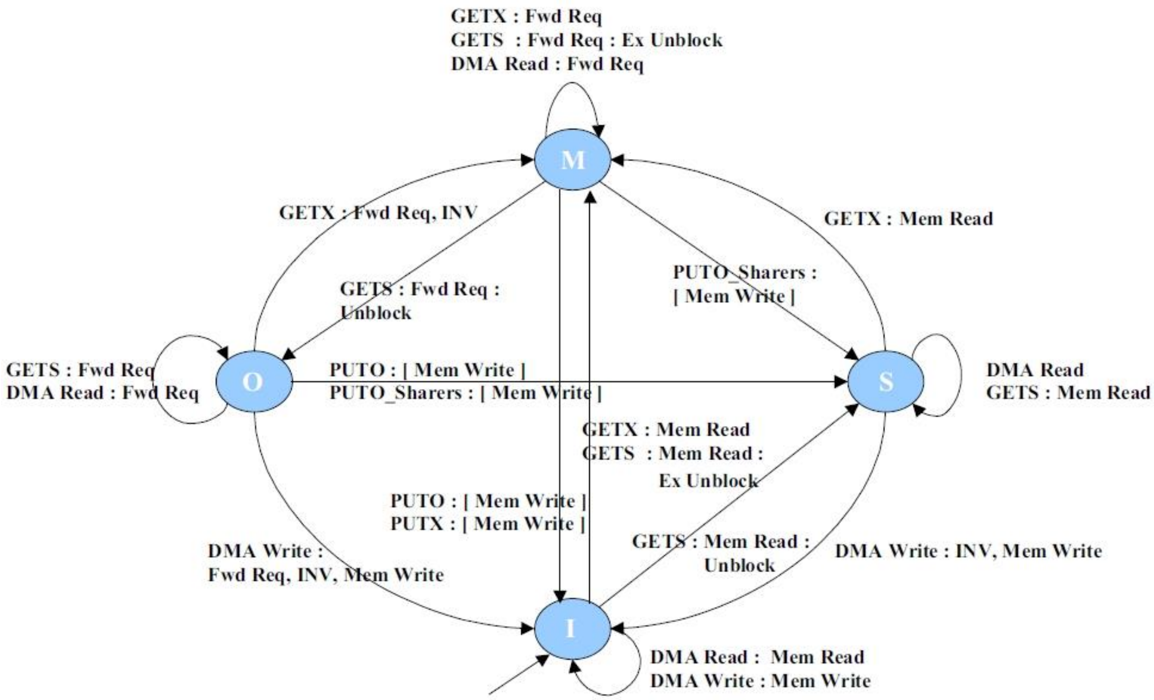
**PART II Experiments**

The different Protocols were run for all the experiments and the results were obtained. The graphs for Runtime and cache to Cache transfers for all Protocols was plotted as it was felt that these were the two parameters which were most suited to comparing the different protocols and their efficiency

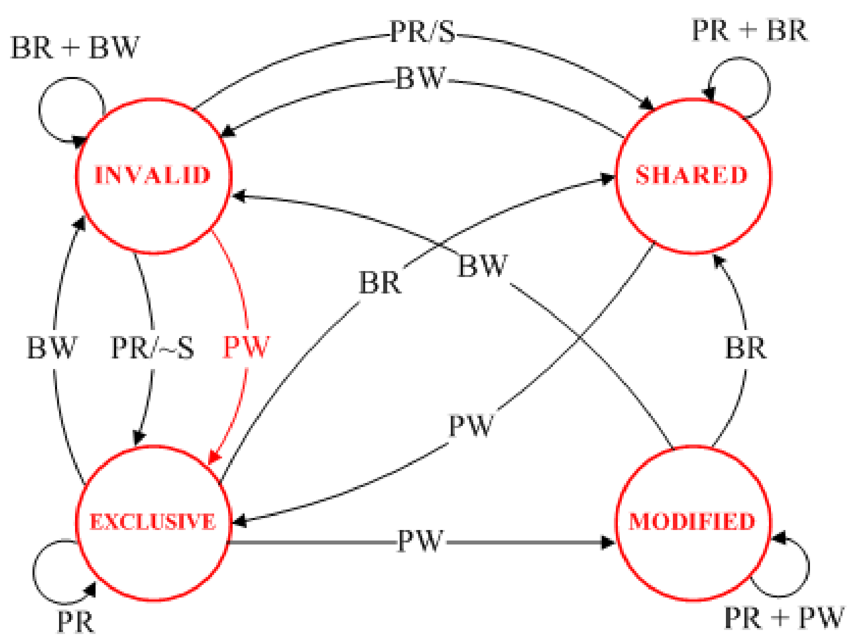
The protocols for which the experiments were run are 1) MSI



2) MOSI



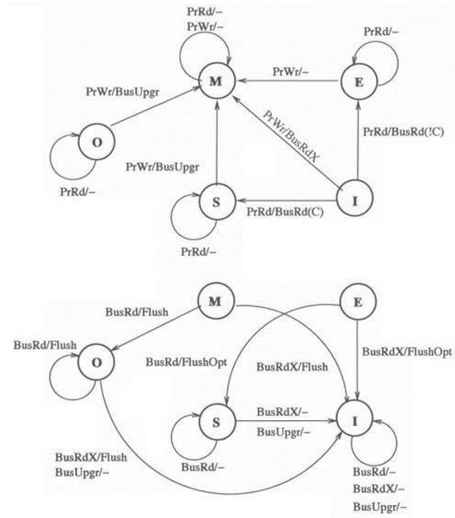

3) MESI



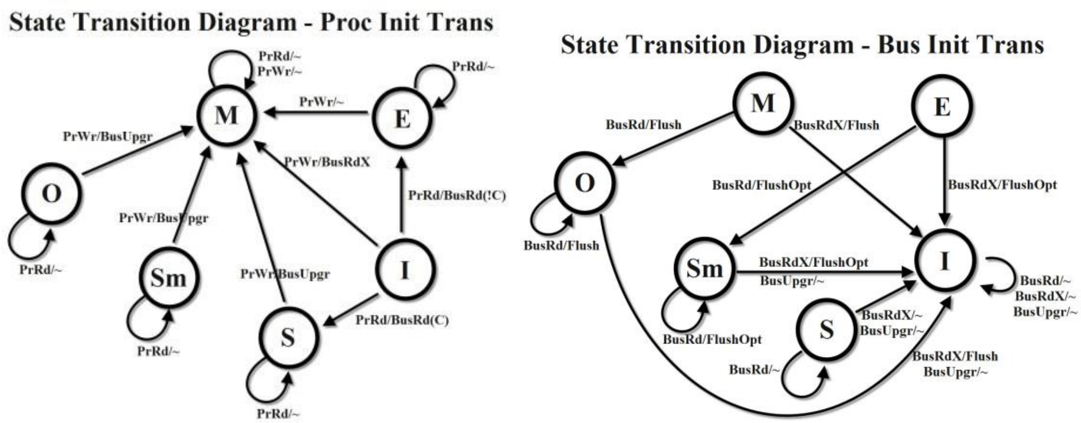

PR = processor read
PW = processor write
S/~S = shared/NOT shared

BR = observed bus read
BW = observed bus write

4) MOESI




5) MOESIF

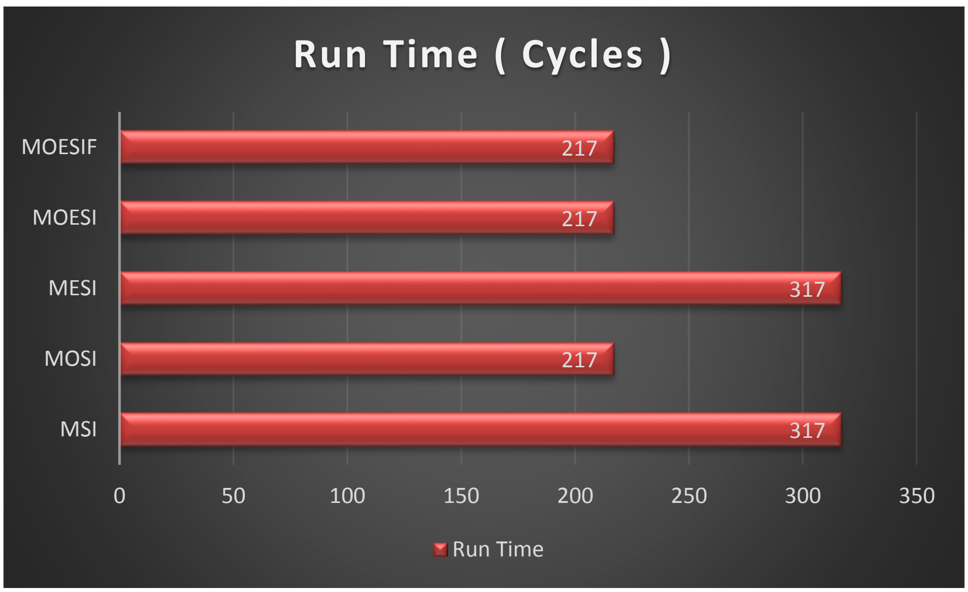


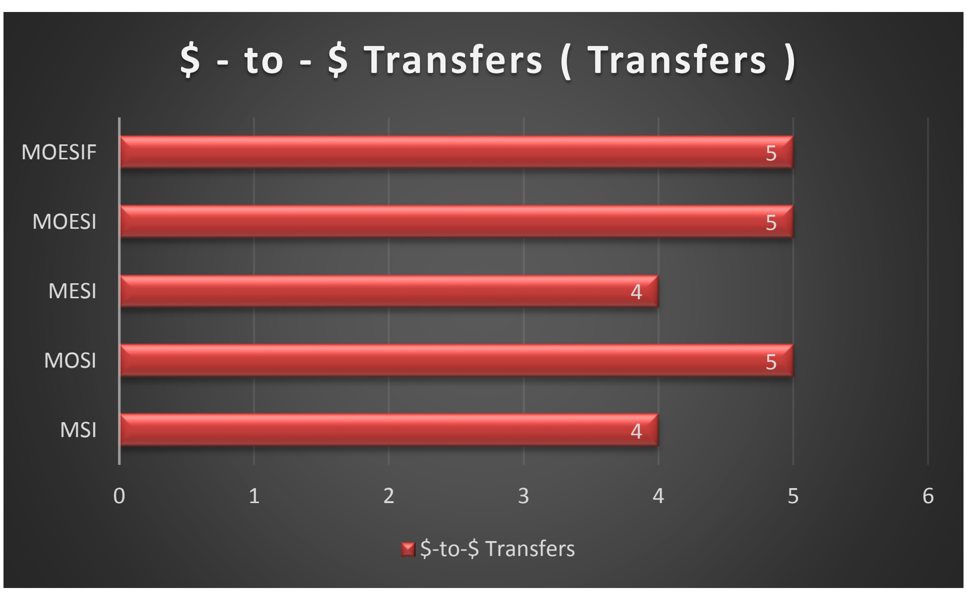

*Here SM is F

**1) Experiment 1**

**Experiment 1 was run for a 4 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 317 | 217 | 317 | 217 | 217 |
| **Cache Misses** | 7 | 7 | 7 | 7 | 7 |
| **Cache Accesses** | 12 | 12 | 12 | 12 | 12 |
| **Silent Upgrades** | 0 | 0 | 0 | 0 | 0 |
| **$-to-$ Transfers** | 4 | 5 | 4 | 5 | 5 |



Run Time for Experiment 1 is minimum for MOSI, MOESI and MOSIEF ( 217 ) whereas for the others it is 317.which can be attributed to the fact that , there is an additional Owner State available in each of these protocols which ensures that the Cache which has the Block in O state can directly transfer the block to any other processor requesting that data instead of going to memory and getting it which leads to fewer Memory accesses and hence lower miss penalty.

As seen from the Graph of $ - to - $ Transfers, MOSI, MOESI and MOESIF have more Cache to Cache Transfers ( 5 ) than MSI and MESI ( 4 ).

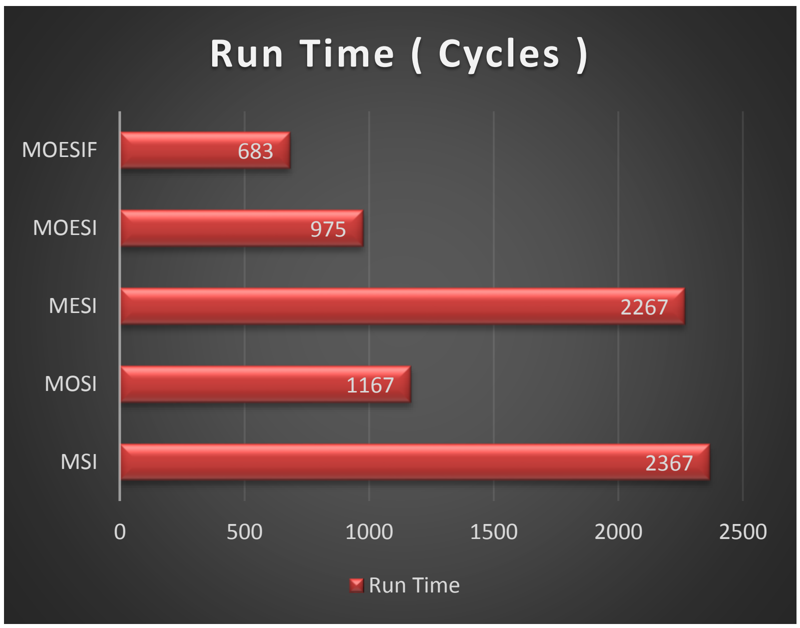
The most ideal protocol for experiment 1 would be the one that incurs least hardware cost and gives optimum performance. Hence it can be concluded that for experiment 1 MOSI would the most ideal as it has lesser states to implement when compared to others, thus incurring less cost and at the same time succeeds in giving optimum performance.

**Conclusion : For experiment 1, the MOSI protocol would be the most apt protocol.**

**2) Experiment 2**

**Experiment 2 was run for a 4 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 2367 | 1167 | 2267 | 975 | 683 |
| **Cache Misses** | 30 | 30 | 30 | 31 | 34 |
| **Cache Accesses** | 104 | 104 | 104 | 104 | 104 |
| **Silent Upgrades** | 0 | 0 | 1 | 1 | 1 |
| **$-to-$ Transfers** | 7 | 19 | 8 | 22 | 28 |




As can be seen from the graph, MOESIF has the least runtime ( 683 )

For MSI and MESI, the Run time is higher as they do not have the O State and hence cannot do Cache to Cache Transfers. MESI Can only do Cache to Cache Transfers when the block is in E State and hence it does not reduce the run time significantly.

In MOSI, the O state allows Cache to Cache Transfers and it is seen that the property of O state that allows a Block that is in O state to remain in O State until it is invalidated improves the performance by reducing the runtime.

In MOESI, since both the E State and O State are present the runtime is further reduced.

IN MOESIF, there is an additional state, the Forwarder state which is quite similar to O (only difference is it is for clean data) and allows cache to cache transfer. This state helps in reducing the runtime to a minimum.

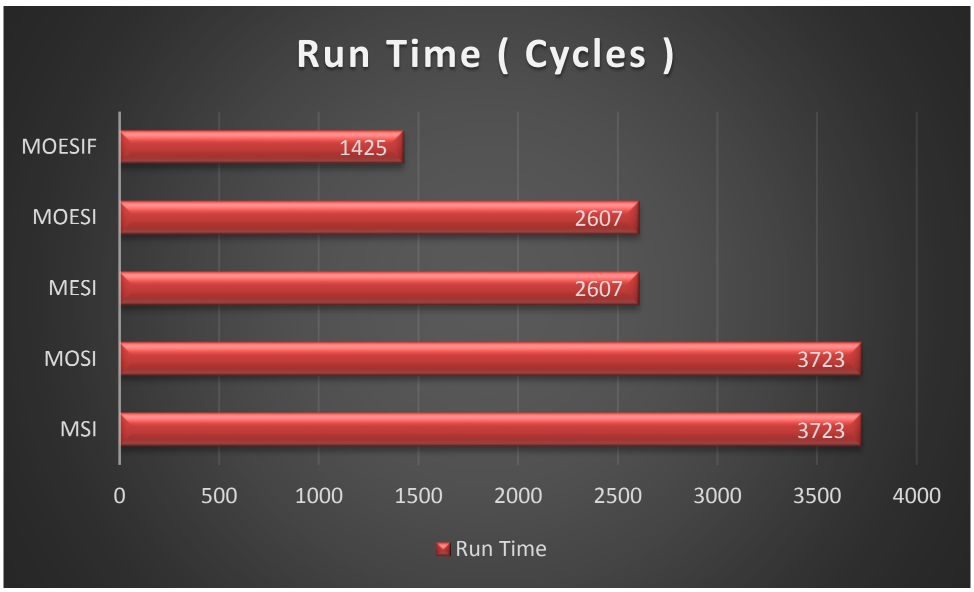
Hence the best protocol for Experiment 2 is MOESIF Protocol.

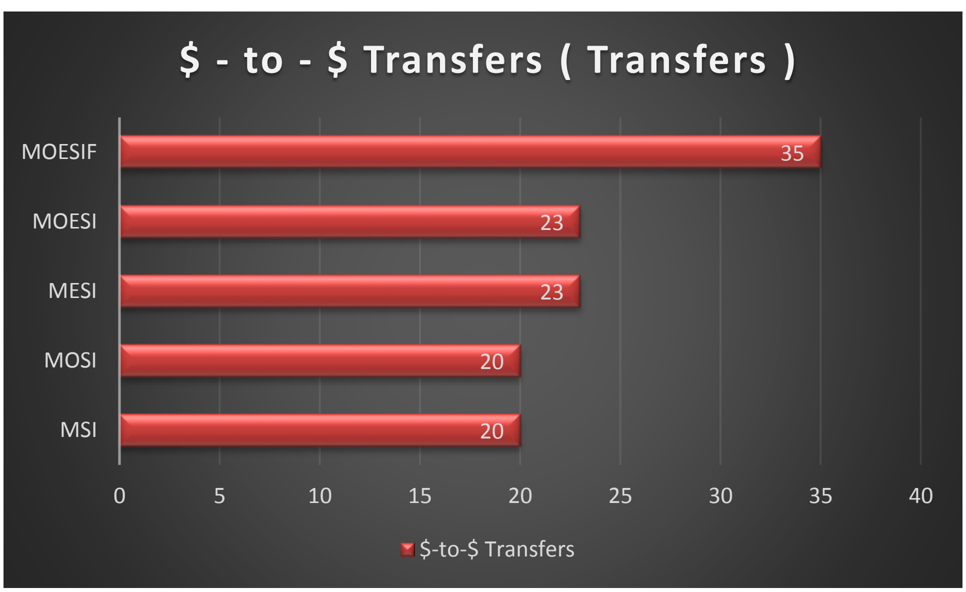
**Conclusion : MOESIF is the best Protocol for Experiment 2 as it gives the least Run Time.**

**3) Experiment 3**

**Experiment 3 was run for an 8 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 3723 | 3723 | 2607 | 2607 | 1425 |
| **Cache Misses** | 56 | 56 | 48 | 48 | 48 |
| **Cache Accesses** | 200 | 200 | 200 | 200 | 200 |
| **Silent Upgrades** | 0 | 0 | 8 | 8 | 8 |
| **$-to-$ Transfers** | 20 | 20 | 23 | 23 | 35 |



As can be seen from the graph, MOESIF has the least runtime ( 1425 )

MSI has a poor runtime as it does not have resources or states to perform Silent Upgrades or Cache to Cache transfers and hence gives a poor runtime value.

For MESI, Run Time is 2607 which is less than that for MOSI ( 3273 ). This can happen when memory blocks are not frequently shared between different processors. Also since the blocks are in E state they allow for cache to cache transfers and incur minimum penalty for the same reason. Also they allow silent upgrades and downgrades thus allowing for an even better runtime.

In MOESI, the runtime further reduces because of Cache to Cache Transfers due to O state and E State.

In MOESIF, the runtime is least due to the extra Forwarder State F. Thus, it has the benefit of Cache to Cache transfers due to E, O and F State.

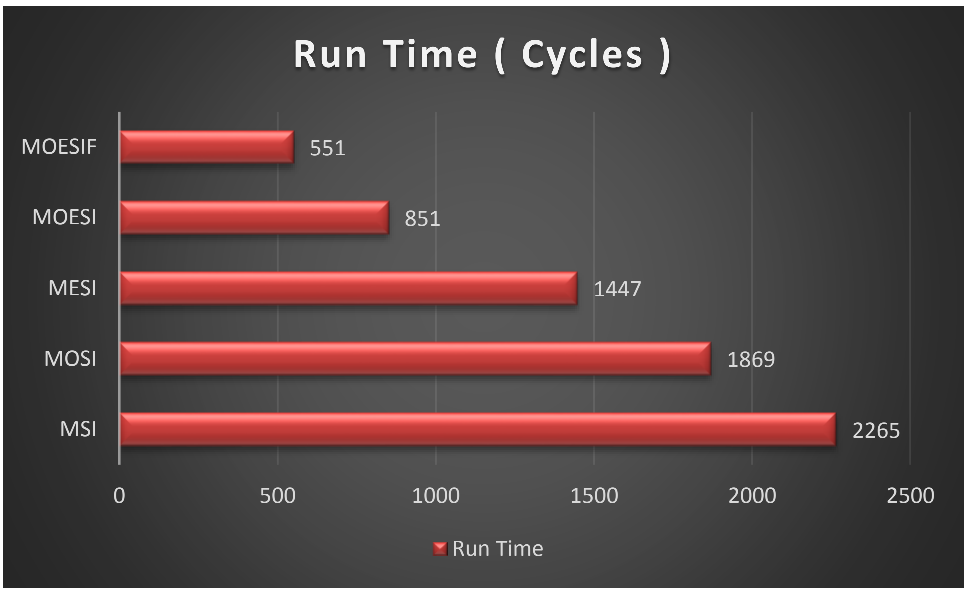
Thus, MOESIF protocol would be best for Experiment 3.

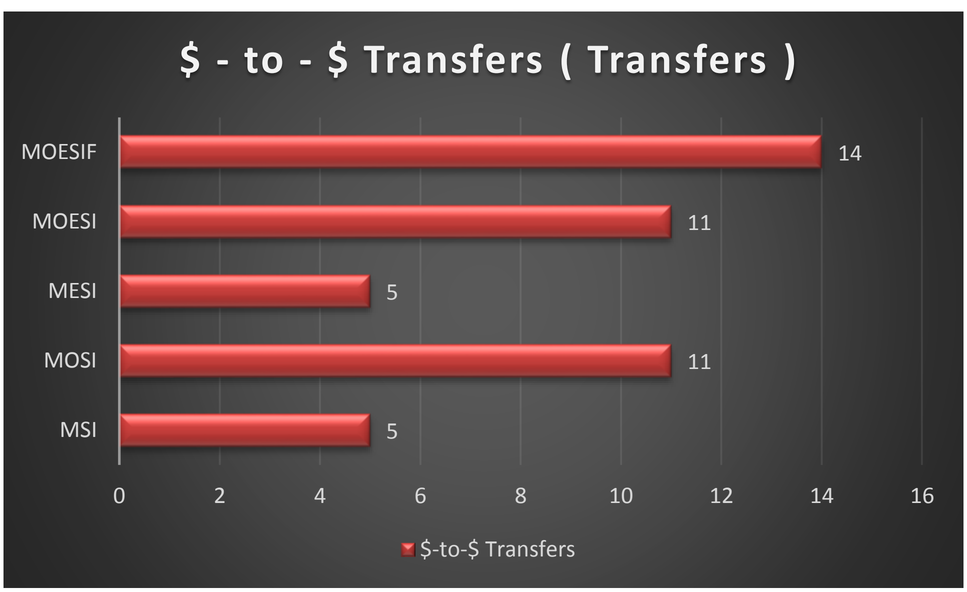
**Conclusion : MOESIF Protocol is the most suited protocol for Experiment 3.**

**4) Experiment 4**

**Experiment 4 was run for a 4 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 2265 | 1869 | 1447 | 851 | 551 |
| **Cache Misses** | 27 | 29 | 19 | 19 | 19 |
| **Cache Accesses** | 60 | 60 | 60 | 60 | 60 |
| **Silent Upgrades** | 0 | 0 | 3 | 3 | 3 |
| **$-to-$ Transfers** | 5 | 11 | 5 | 11 | 14 |



Similar to Experiment 3, it can be seen that Run time is least for MOESIF ( 551 )

MSI has a poor runtime as it does not have resources or states to perform Silent Upgrades or Cache to Cache transfers and hence gives a poor runtime value.

The run time for MESI (1447) is less than that for MOSI (1869). This is again similar to experiment 3. This can happen when memory blocks are not frequently shared between different processors. Also since the blocks are in E state they allow for cache to cache transfers and incur minimum penalty for the same reason. Also they allow silent upgrades and downgrades thus allowing for an even better runtime.

The runtime for MOESI (851) is better than that for MOSI and MESI as it has both the E and O states thus allowing for more Cache to Cache Transfers and better runtime.

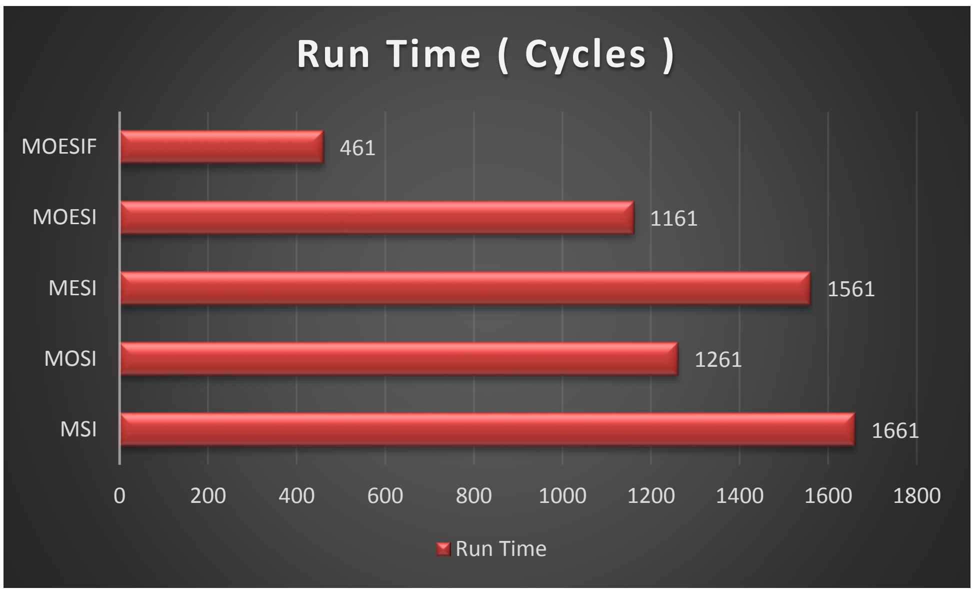
MOESIF, as in the previous case proves to be the best choice for this experiment too.

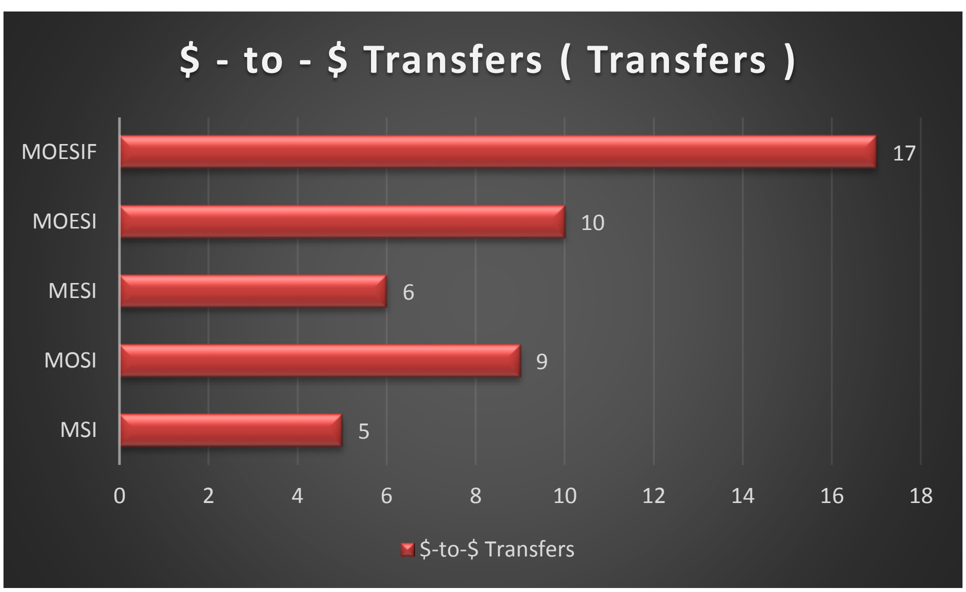
**Conclusion : MOESIF would be the most ideal protocol for Experiment 4.**

**5) Experiment 5 :**

**Experiment 5 was run for an 8 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 1661 | 1261 | 1561 | 1161 | 461 |
| **Cache Misses** | 21 | 21 | 21 | 21 | 21 |
| **Cache Accesses** | 37 | 37 | 37 | 37 | 37 |
| **Silent Upgrades** | 0 | 0 | 0 | 0 | 0 |
| **$-to-$ Transfers** | 5 | 9 | 6 | 10 | 17 |



As seen from the graph for Runtime, the runtime is best again for MOESIF Protocol (461).

The result of runtime similar to that obtained in Experiment 2.

MSI has a poor runtime as it does not have the extra states to perform Silent Upgrades or Cache to Cache transfers except when in the M state and it gets a read.

The run time for MESI (1561) is greater than that for MOSI (1261). Here the state O is used more than the E state.

The runtime for MOESI (1161) is better than that for MOSI and MESI as it has the benefit of both the E and O State and hence can have more Cache to Cache Transfers.

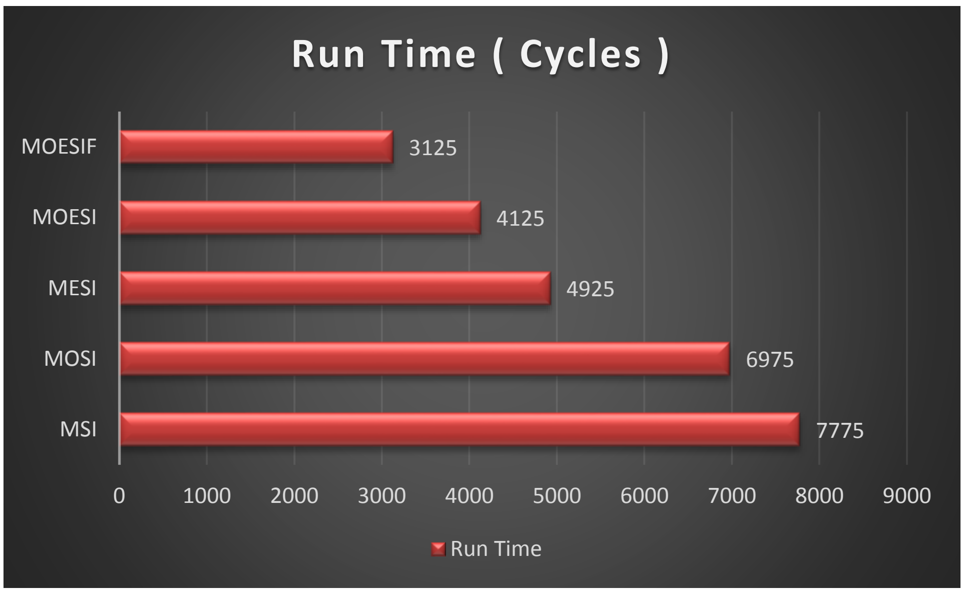
MOESIF, obviously, would be the best choice as it has the benefits of O and E State as well as an additional Forwarder state F thereby reducing Runtime.

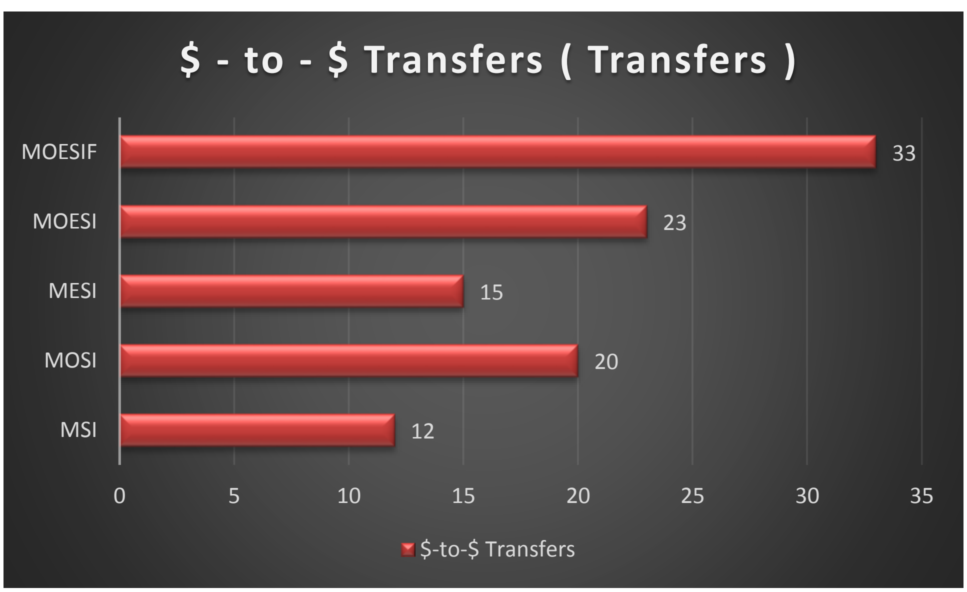
**Conclusion : MOESIF is the best protocol for Experiment 5.**

**6) Experiment 6**

**Experiment 6 was run for a 16 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 7775 | 6975 | 4925 | 4125 | 3125 |
| **Cache Misses** | 87 | 87 | 62 | 62 | 62 |
| **Cache Accesses** | 747 | 747 | 747 | 747 | 747 |
| **Silent Upgrades** | 0 | 0 | 25 | 25 | 25 |
| **$-to-$ Transfers** | 12 | 20 | 15 | 23 | 33 |



The Runtime for Experiment 6 is least for MOESIF (3125).

MSI has a poor runtime as it does not have resources or states to perform Silent Upgrades or Cache to Cache transfers and hence gives a poor runtime value.

For MESI, RunTime is is less than that for MOSI . This can happen when memory blocks are not frequently shared between different processors. Also since the blocks are in E state they allow for cache to cache transfers and incur minimum penalty for the same reason. Also they allow silent upgrades and downgrades thus allowing for an even better runtime.

In MOESI, the runtime further reduces because of Cache to Cache Transfers due to O state and E State.

In MOESIF, the runtime is least due to the extra Forwarder State F. Thus, it has the benefit of Cache to Cache transfers due to E, O and F State.

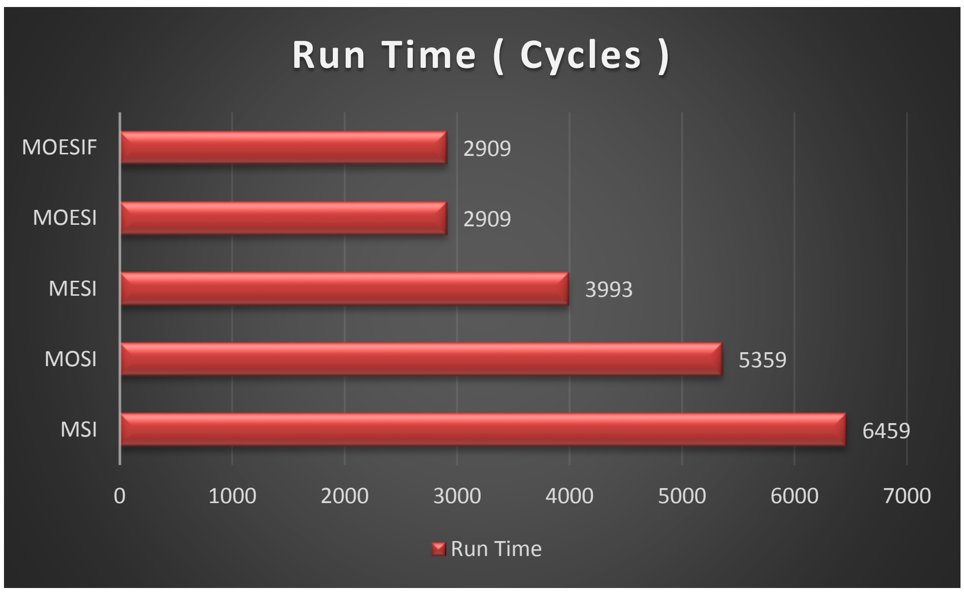
Thus, MOESIF protocol would be best for Experiment 6.

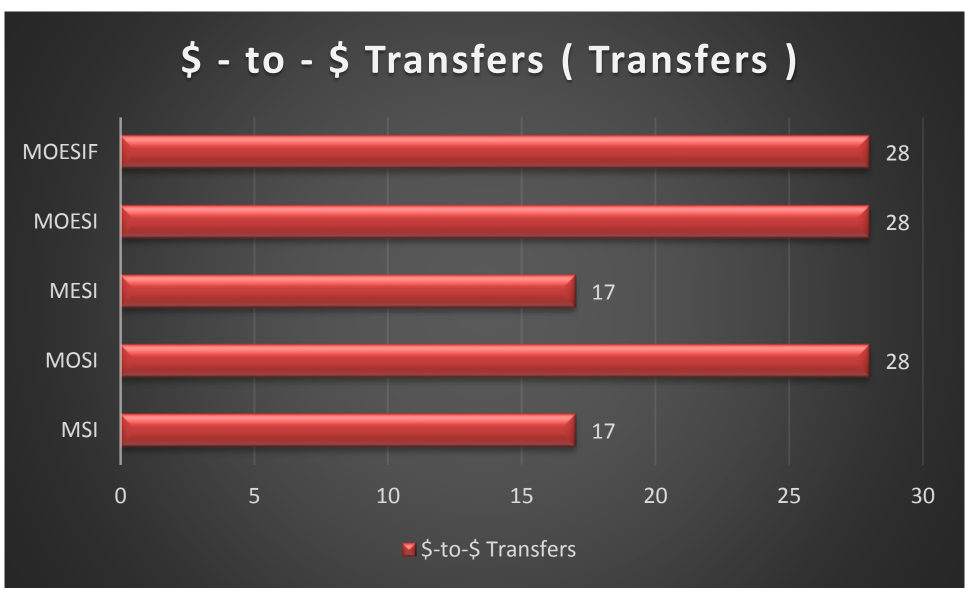
**Conclusion : MOESIF is the best protocol for Experiment 6.**

**Experiment 7 :**

**Experiment 7 was run for a 16 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 6459 | 5359 | 3993 | 2909 | 2909 |
| **Cache Misses** | 79 | 79 | 55 | 55 | 55 |
| **Cache Accesses** | 952 | 952 | 952 | 952 | 952 |
| **Silent Upgrades** | 0 | 0 | 24 | 24 | 24 |
| **$-to-$ Transfers** | 17 | 28 | 17 | 28 | 28 |



MSI again has a poor runtime.

MESI has a runtime of (3993) while MOSI has a runtime of (5359).

In this case, MESI performs better than MOSI which could be a result of greater number of blocks in the E state which leads to greater silent upgrades and downgrades.

MOESI has a better runtime (2909) than MOSI and MESI as it has the benefit of both E and O state and hence has Cache to Cache transfers and Silent Upgrades and downgrades.

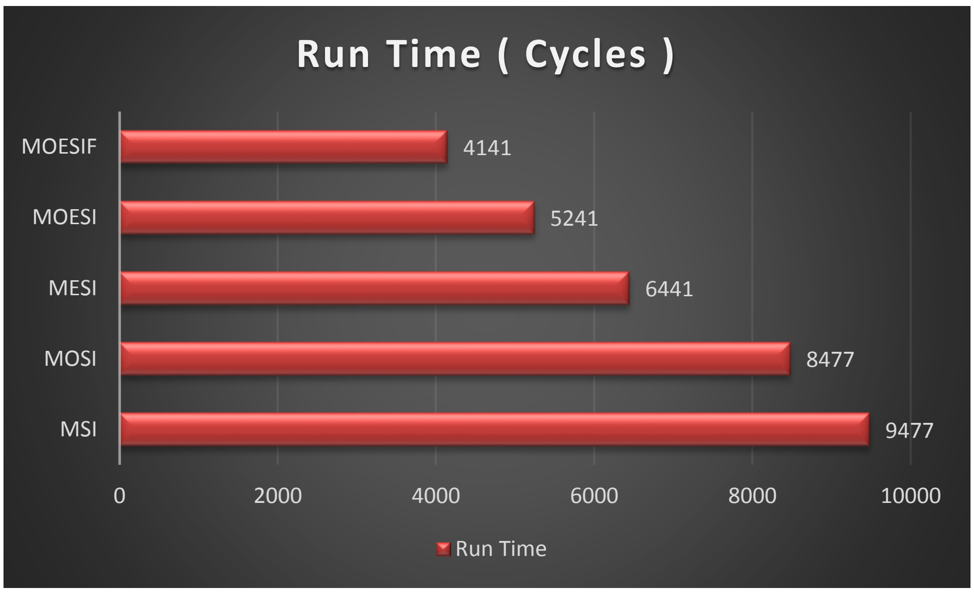
MOESIF in this case has the same runtime (2909) as MOESI. And hence it is seen that the F state is not useful at all. Thus avoiding the F state would result in cost reduction and hence it can be said that MOESI is the ideal protocol for experiment 7.

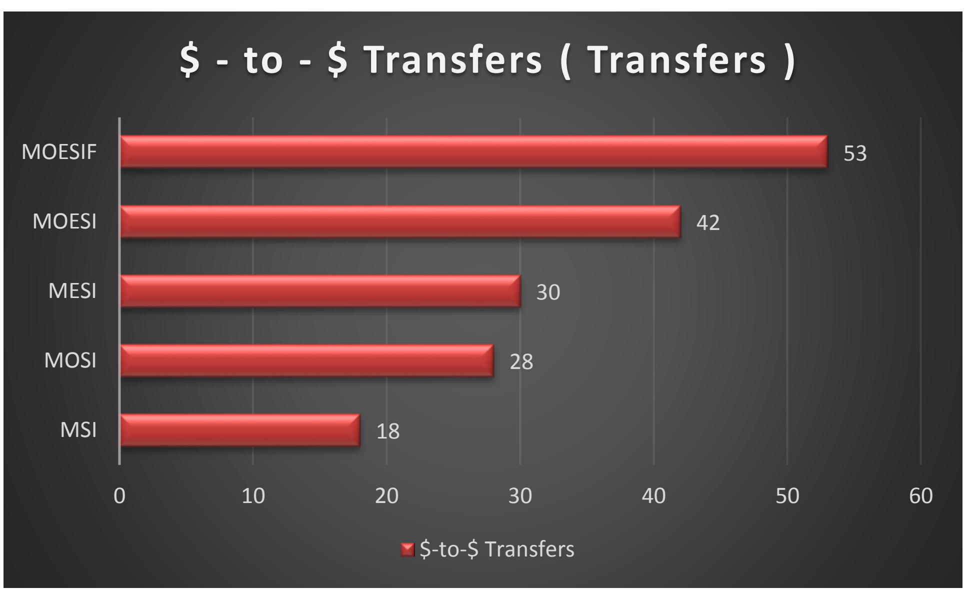
**Conclusion : For experiment 7,  MOESI would be the best protocol to use as it would cheaper to implement and give an optimum result.**

**Experiment 8 :**

**Experiment 8 was run for a 16 core processor and the results are as follows**

| Parameter | MSI | MOSI | MESI | MOESI | MOESIF |
|---|---|---|---|---|---|
| **Run Time** | 9477 | 8477 | 6441 | 5241 | 4141 |
| **Cache Misses** | 110 | 110 | 92 | 92 | 92 |
| **Cache Accesses** | 800 | 800 | 800 | 800 | 800 |
| **Silent Upgrades** | 0 | 0 | 19 | 19 | 19 |
| **$-to-$ Transfers** | 18 | 28 | 30 | 42 | 53 |



This is similar to experiment 7.

MSI obviously has a poor runtime as it does not have the required states to perform or Silent Upgrades or Cache to Cache transfers.

MESI has a runtime of (6441) while MOSI has a runtime of (8477).

In this case, MESI performs better than MOSI. This could be due to the fact that there could be a lot of reads when the block is in Estate which leads to greater number of Cache to Cache transfers and silent upgrades and downgrades.

MOESI has a better runtime (5241) than MOSI and MESI benefiting from both E and O states and hence allowing Cache to Cache transfers and Silent Upgrades.

MOESIF has the least runtime (4141) since it has the added benefit of O, E and M State. Also, it has the Forwarding state which further reduces the runtime.

**Conclusion : MOESIF is the best protocol to use for Experiment 8.**