1. Individual Traces

Experiment1

MESI	MOSI	MOESIF
Run Time: 317 cycles Cache Misses: 7 misses Cache Accesses: 12 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 4 transfers	Run Time: 217 cycles Cache Misses: 7 misses Cache Accesses: 12 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 5 transfers	Run Time: 217 cycles Cache Misses: 7 misses Cache Accesses: 12 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 5 transfers

In this experiment, MOSI and MOESIF seem to perform equivalently, while MESI lags behind. This would make MOSI the clear winner, due to its simpler implementation. It seems like there are a lot of times where a GETM is followed by a GETS, but no times when a GETS is followed by a GETM, since there are no silent upgrades from E to M.

Experiment2

MESI	MOSI	MOESIF
Run Time: 2267 cycles Cache Misses: 30 misses Cache Accesses: 104 accesses Silent Upgrades: 1 upgrades $-to-$ Transfers: 8 transfers	Run Time: 1167 cycles Cache Misses: 30 misses Cache Accesses: 104 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 19 transfers	Run Time: 683 cycles Cache Misses: 34 misses Cache Accesses: 104 accesses Silent Upgrades: 1 upgrades $-to-$ Transfers: 28 transfers

In this experiment, MOESIF performs better than MOSI which performs better than MESI. All of these performance differences are significant. MOESIF misses more, but also has much more cache to cache transfers. There is only one silent upgrade, which would indicate why MESI performs poorly. The cache misses are likely just protocol misses where an F or O state calls a GETM.

Experiment3

MESI	MOSI	MOESIF
Run Time: 2607 cycles Cache Misses: 48 misses Cache Accesses: 200 accesses Silent Upgrades: 8 upgrades $-to-$ Transfers: 23 transfers	Run Time: 3723 cycles Cache Misses: 56 misses Cache Accesses: 200 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 20 transfers	Run Time: 1425 cycles Cache Misses: 48 misses Cache Accesses: 200 accesses Silent Upgrades: 8 upgrades $-to-$ Transfers: 35 transfers

In this experiment, MOESIF performs best. MOSI performs very poorly and MESI performs moderately well. There are a significant amount of silent upgrades, which would indicate MESI's improvement over MOSI. There are a lot less misses in MOESIF and MESI than in MOSI. This is due to the added E state, since there are 8 silent upgrades and 8 more cache misses. MOESIF has more transfers than MOSI too, so the F state must be important.

Experiment4

MESI	MOSI	MOESIF
Run Time: 1447 cycles Cache Misses: 19 misses Cache Accesses: 60 accesses Silent Upgrades: 3 upgrades $-to-$ Transfers: 5 transfers	Run Time: 1869 cycles Cache Misses: 29 misses Cache Accesses: 60 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 11 transfers	Run Time: 551 cycles Cache Misses: 19 misses Cache Accesses: 60 accesses Silent Upgrades: 3 upgrades $-to-$ Transfers: 14 transfers

In this trace, MOESIF again performs the best. MESI performs better than MOSI also. MESI and MOESIF have equal numbers of cache misses, but MOESIF has a lot more transfers. This seems to indicate heavy reliance on the F state sending data to others entering the S state. This means a lot of caches tend to share data.

Experiment5

MESI	MOSI	MOESIF
Run Time: 1561 cycles Cache Misses: 21 misses Cache Accesses: 37 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 6 transfers	Run Time: 1261 cycles Cache Misses: 21 misses Cache Accesses: 37 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 9 transfers	Run Time: 461 cycles Cache Misses: 21 misses Cache Accesses: 37 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 17 transfers

MOESIF performs best once again, while MOSI performs slightly better than MESI. MOSI performs better because of the three extra cache-to-cache transfers from the O state. MESI performs worst because the E state is not used at all, so it is just MSI.

Experiment6

MESI	MOSI	MOESIF
Run Time: 4925 cycles Cache Misses: 62 misses Cache Accesses: 747 accesses Silent Upgrades: 25 upgrades $-to-$ Transfers: 15 transfers	Run Time: 6975 cycles Cache Misses: 87 misses Cache Accesses: 747 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 20 transfers	Run Time: 3125 cycles Cache Misses: 62 misses Cache Accesses: 747 accesses Silent Upgrades: 25 upgrades $-to-$ Transfers: 33 transfers

MOESIF performs best again, while MOSI performs very poorly and MESI performs in the middle. MOSI seems to perform poorly because of its large number of cache misses. This is due to the large number of silent upgrades from the E state. Clearly the E and F states are very important in this case, indicating lots of writes after reads and shared data.

Experiment7

MESI	MOSI	MOESIF
Run Time: 3993 cycles Cache Misses: 55 misses	Run Time: 5359 cycles Cache Misses: 79 misses	Run Time: 2909 cycles Cache Misses: 55 misses

Cache Accesses: 952 accesses Silent Upgrades: 24 upgrades $-to-$ Transfers: 17 transfers	Cache Accesses: 952 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 28 transfers	Cache Accesses: 952 accesses Silent Upgrades: 24 upgrades $-to-$ Transfers: 28 transfers

MOESIF performs best again, with MOSI losing out and MESI performing moderately well. There are a significant number of silent upgrades again from the E state, causing MESI to perform relatively well. Also, there are a lot of cache-to-cache transfers from mainly the O state this time, indicating a significant amount of shared data after it is written to.

Experiment8

MESI	MOSI	MOESIF
Run Time: 6441 cycles Cache Misses: 92 misses Cache Accesses: 800 accesses Silent Upgrades: 19 upgrades $-to-$ Transfers: 30 transfers	Run Time: 8477 cycles Cache Misses: 110 misses Cache Accesses: 800 accesses Silent Upgrades: 0 upgrades $-to-$ Transfers: 28 transfers	Run Time: 4141 cycles Cache Misses: 92 misses Cache Accesses: 800 accesses Silent Upgrades: 19 upgrades $-to-$ Transfers: 53 transfers

MOESIF performs best again, with MOSI once again performing poorly and MESI performing moderately well. There are a significant number of silent upgrades, preventing MOSI from performing well, and there are more cache-to-cache transfers in the MOESIF than in MOSI indicating high usage of the F state to transfer data.

2. Choosing Protocol

In the 8 experiments, MOESIF performed best in all but one of the traces where it tied MOSI. Given all the programs are equally weighted, MOESIF would obviously be the best protocol to implement. It seems that the caches being able to send data to one another, as well as to silently upgrade from the E to M state when necessary are very useful for all of the provided traces. The only case in which it would not be worth it to use MOESIF over MOSI would be experiment 1, but in the last 3 experiments MOSI performed very poorly. MOESIF is clearly the best option, so I would choose to implement it.

3. Limitations of Simulator

The only part of the simulator we were able to implement were the protocol objects of each block. In addition, we only had one line that could be used to signal. Because of this, the bus had to be atomic. This isn't necessarily the best option though, since waiting for a data message from memory can take time. It might be more realistic for the bus to not be atomic in reality, but this might make the system more complex. This would require the implementation of Abort, which was not used in this project. It would also require more states and logic in general. However, it should in general improve the latency in real time, since requests wouldn't have to wait for each other to finish.