ECE – 6100

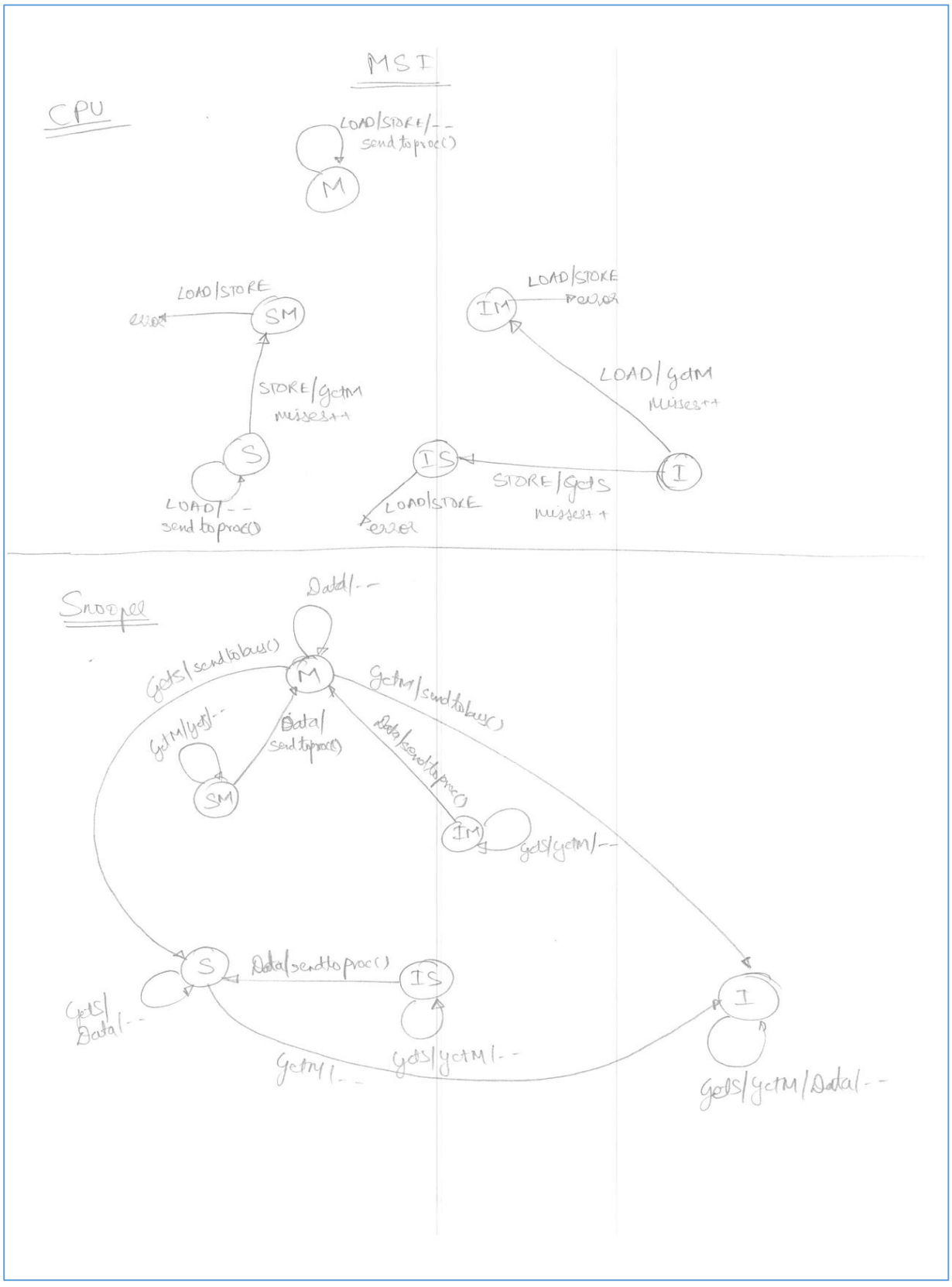
Cache Coherence

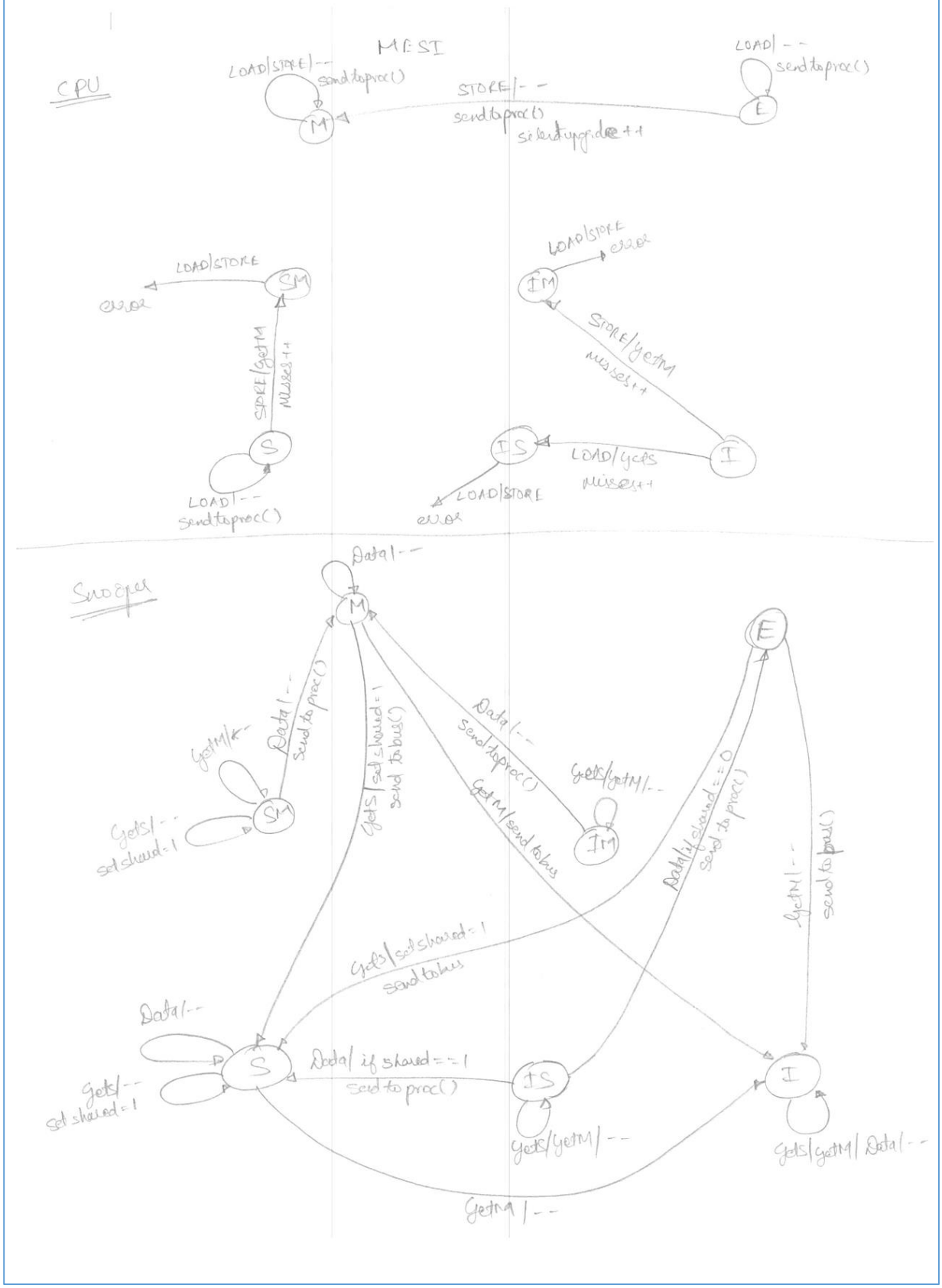
PROJECT REPORT

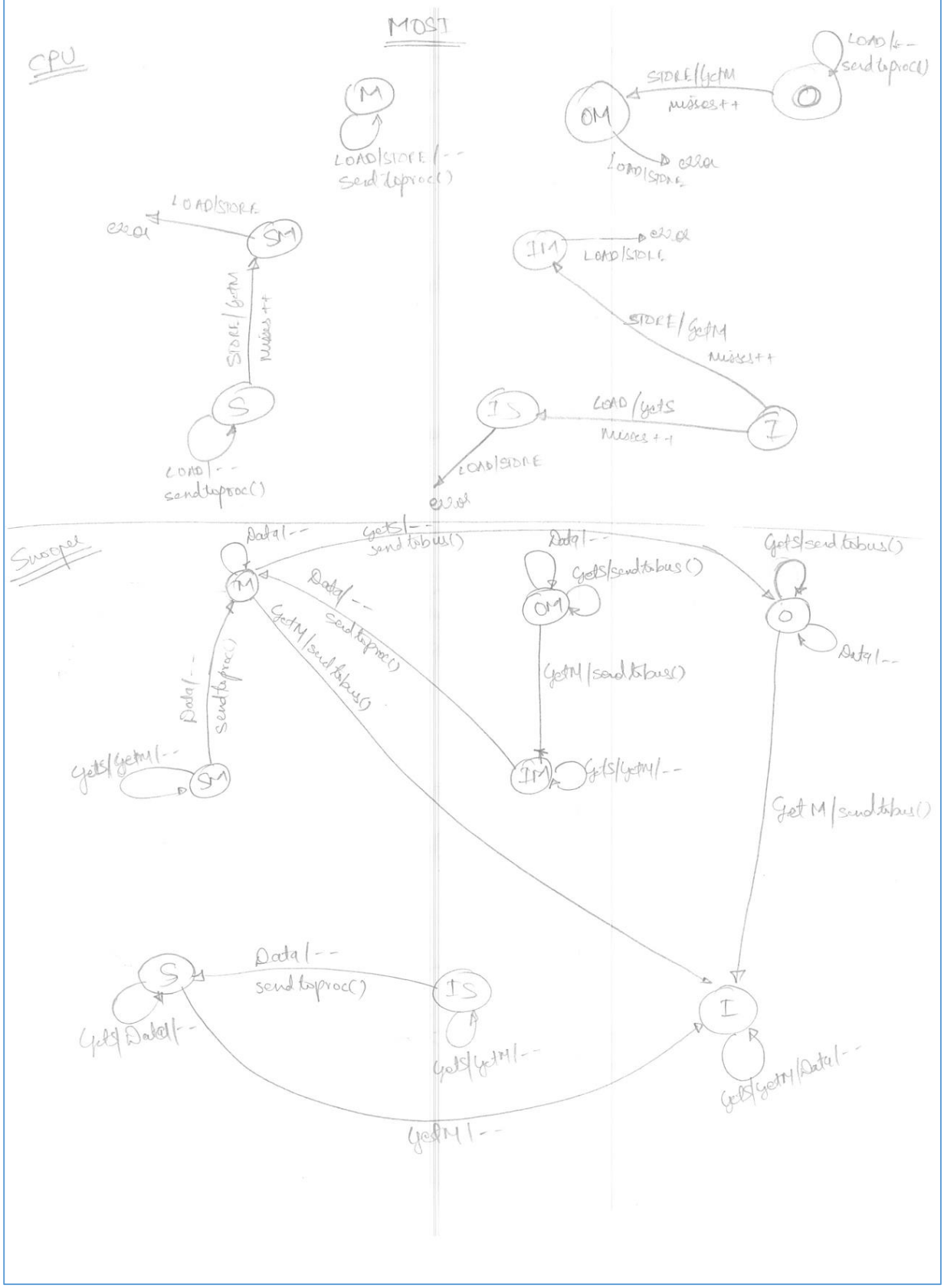
Balaji Mamidala

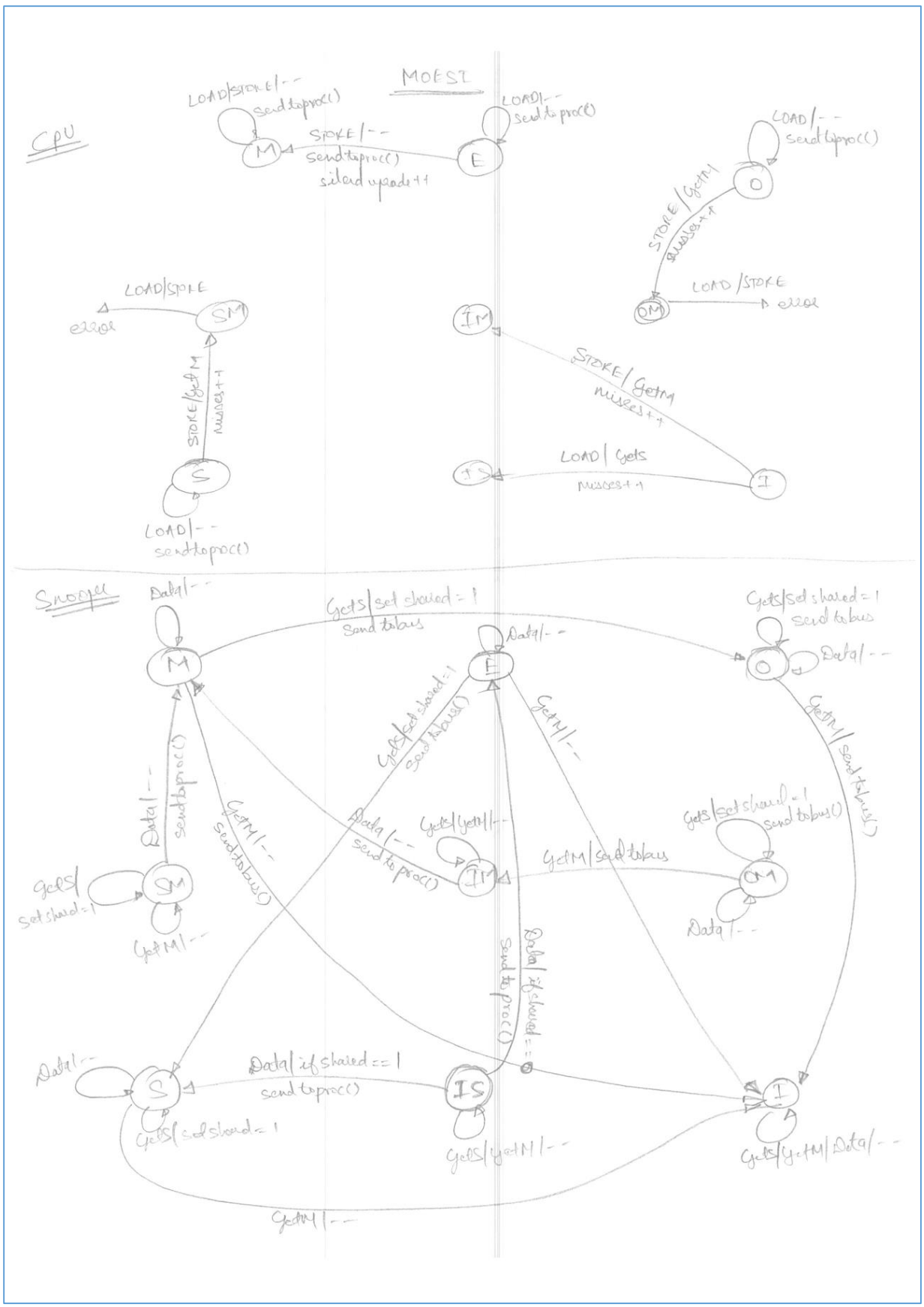
GTID - 903060531

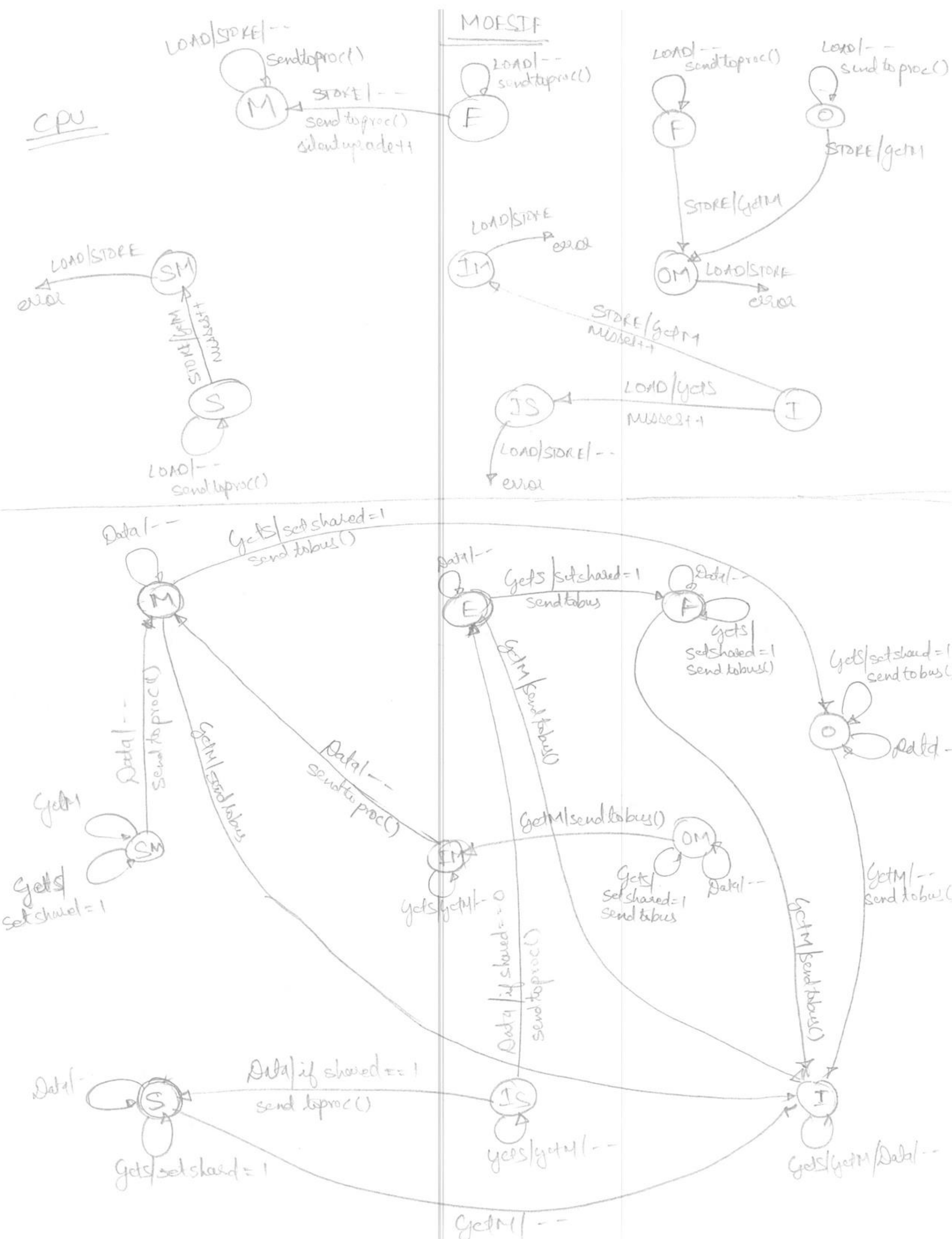
Below I have attached the state-machines used to implement the protocols:




MESI
CPU
LOAD|STORE|--
send to proc()
M
STORE|--
send to proc()
set dirty = 1
E
LOAD|--
send to proc()
LOAD|STORE
error
SM
STORE|getM
misses++
S
LOAD|--
send to proc()
LOAD|STORE
error
IM
STORE|getM
misses++
IS
LOAD|getS
misses++
I
LOAD|STORE
error
Snooper
Data|--
M
Data|--
send to proc()
getM|K-
SM
getS|--
set shared=1
getS|set shared=1
send to bus()
Data|--
send to proc()
getM|send to bus
IM
getS|getM|--
E
Data|if shared==0
send to proc()
getM|--
send to bus()
getS|set shared=1
send to bus
Data|--
S
getS|--
set shared=1
Data| if shared==1
send to proc()
IS
getS|getM|--
I
getS|getM|Data|--
getM|--


MOST
CPU
M
LOAD/STORE/--
send toproc()
OM
STORE/getM
misses++
LOAD/--
send toproc()
LOAD/STORE
error
IM
LOAD/STORE
STORE/GetM
misses++
I
LOAD/getS
misses++
IS
LOAD/STORE
error
SM
LOAD/STORE
error
STORE/getM
misses++
S
LOAD/--
send toproc()
Snooper
M
Data/--
GetS/--
send tobus()
Data/--
send toproc()
GetM/send tobus()
Data/--
send toproc()
GetS/getM/--
SM
OM
GetS/send tobus()
Data/--
GetM/send tobus()
IM
GetS/getM/--
O
GetS/send tobus()
Data/--
Get M/send tobus()
S
Data/--
send toproc()
GetS/Data/--
IS
GetS/getM/--
I
GetS/getM/Data/--
GetM/--

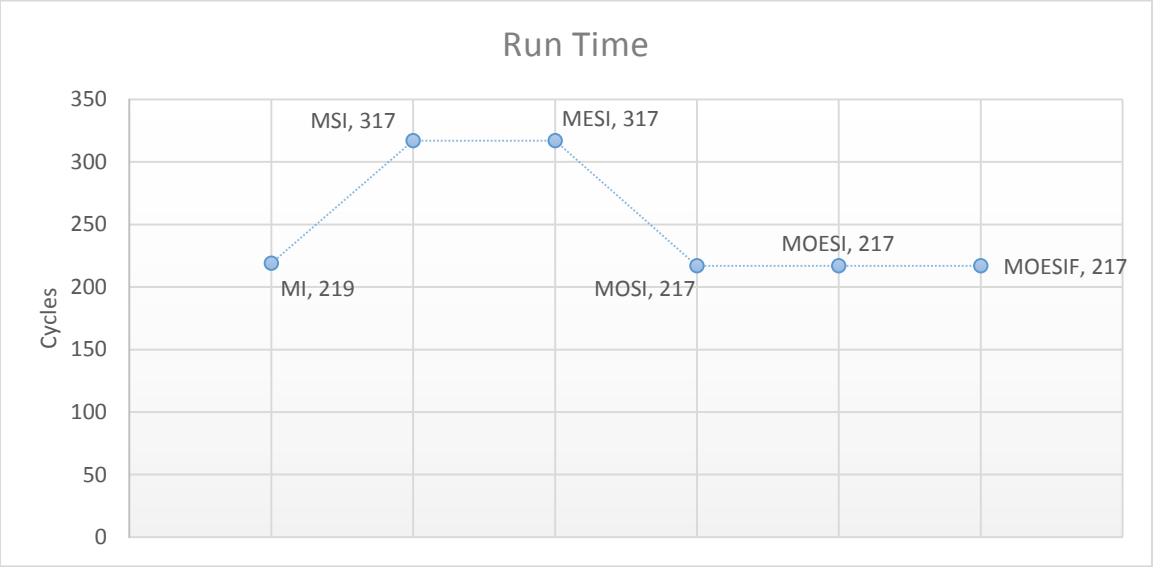

CPU
MOESI
LOAD/STORE/--
send to proc()
M
STORE/--
send to proc()
silent upgrade++
E
LOAD/--
send to proc()
O
LOAD/--
send to proc()
STORE/GetM
misses++
OM
LOAD/STORE
error
LOAD/STORE
error
SM
STORE/GetM
misses++
S
LOAD/--
send to proc()
IM
STORE/GetM
misses++
IS
LOAD/Gets
misses++
I
Snooper
Data/--
M
Gets/set shared=1
send to bus
E
Gets/set shared=1
send to bus
O
Data/--
Data/--
send to proc()
GetM/--
Gets/set shared=1
send to bus()
Gets/get M/--
Data/--
send to proc()
GetM/send to bus
SM
Gets/
set shared=1
GetM/--
GetM/--
send to bus()
Data/--
send to proc()
Gets/set shared=1
send to bus()
OM
Data/--
Data/if shared==1
send to proc()
S
Data/--
Gets/set shared=1
IS
Gets/get M/--
I
Gets/get M/Data/--
GetM/--

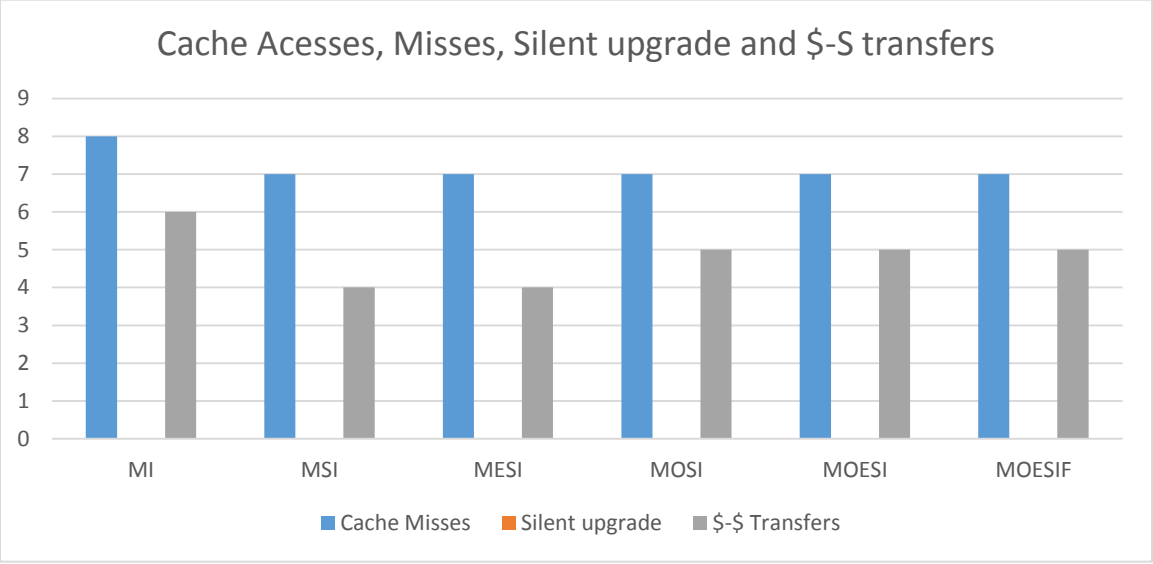


CPU
MOESIF
LOAD|STORE|--
sendtoproc()
STORE|--
send to proc()
silentupgrade++
LOAD|--
sendtoproc()
STORE|GetM
LOAD|STORE
error
misses++
STORE|GetM
LOAD|GetS
LOAD|STORE|--
Data|--
GetS|set shared=1
send tobus()
GetS|setshared=1
send tobus
GetS|set shared=1
send to bus()
GetM|send tobus()
Data|--
send to proc()
GetM|send tobus
GetM|--
send tobus()
GetS|GetM|--
Data|if shared == 1
send toproc()
Data|if shared == 0
send toproc()
GetS|GetM|--
GetS|set shared=1
GetS|GetM|Data|--
GetM|--

EXPERIMENT#1

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	219	8	12	0	6
MSI	317	7	12	0	4
MESI	317	7	12	0	4
MOSI	217	7	12	0	5
MOESI	217	7	12	0	5
MOESIF	217	7	12	0	5



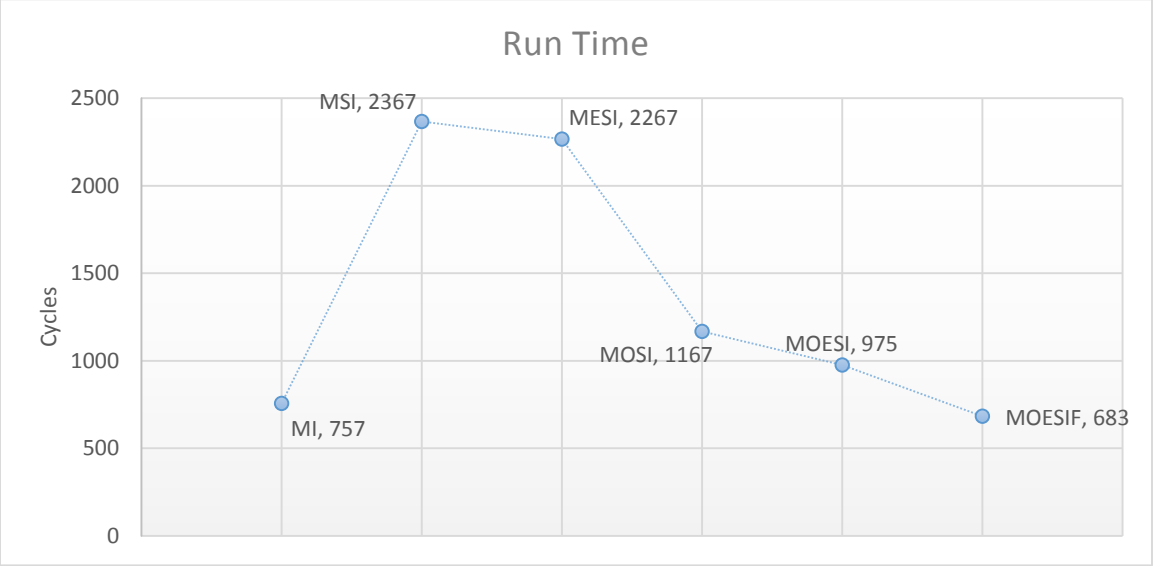


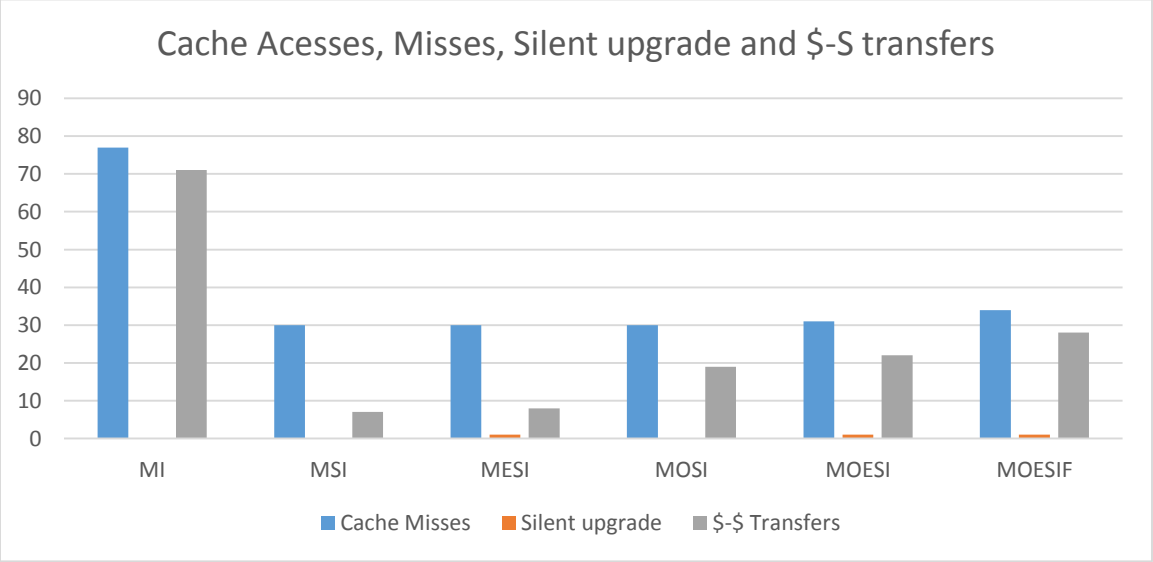
Observations:

1. MOSI protocol has the best run-time because of high $-$ transfers.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding Exclusive state to MSI doesn't change the run-time. Thus, this trace doesn't benefit from E-state which can be observed in the fact that MOESI and MOSI have the same run-time.
4. Adding owned state reduces run-time by increasing $-$ transfer by 1.
5. Adding F-state does not improve run-time.
6. As O gives significant improvement run-time this experiment might have write-shared property.

EXPERIMENT#2

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	757	77	104	0	71
MSI	2367	30	104	0	7
MESI	2267	30	104	1	8
MOSI	1167	30	104	0	19
MOESI	975	31	104	1	22
MOESIF	683	34	104	1	28



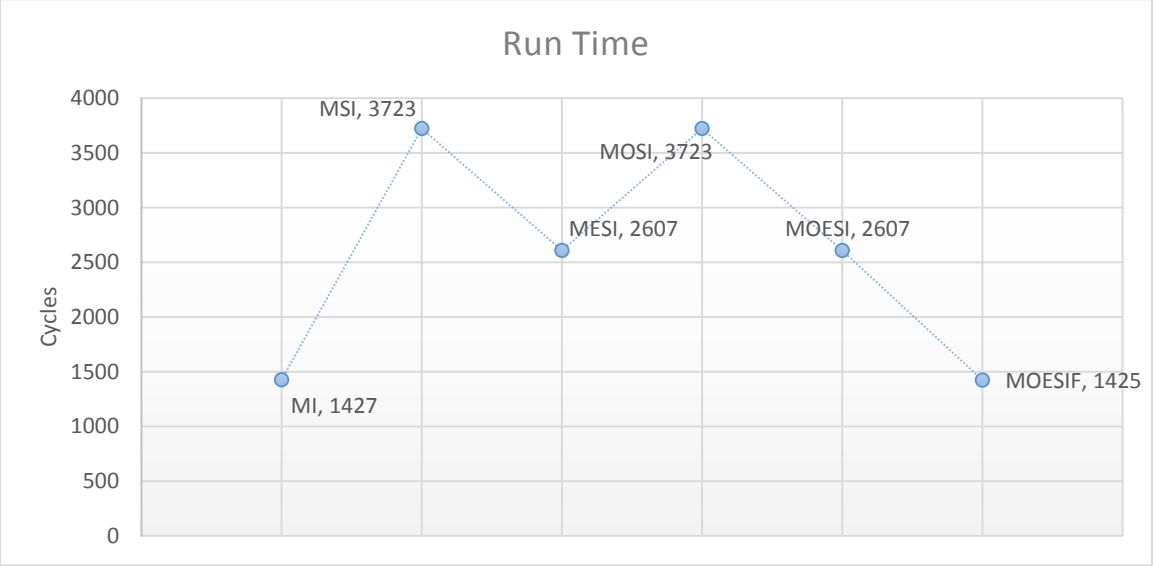


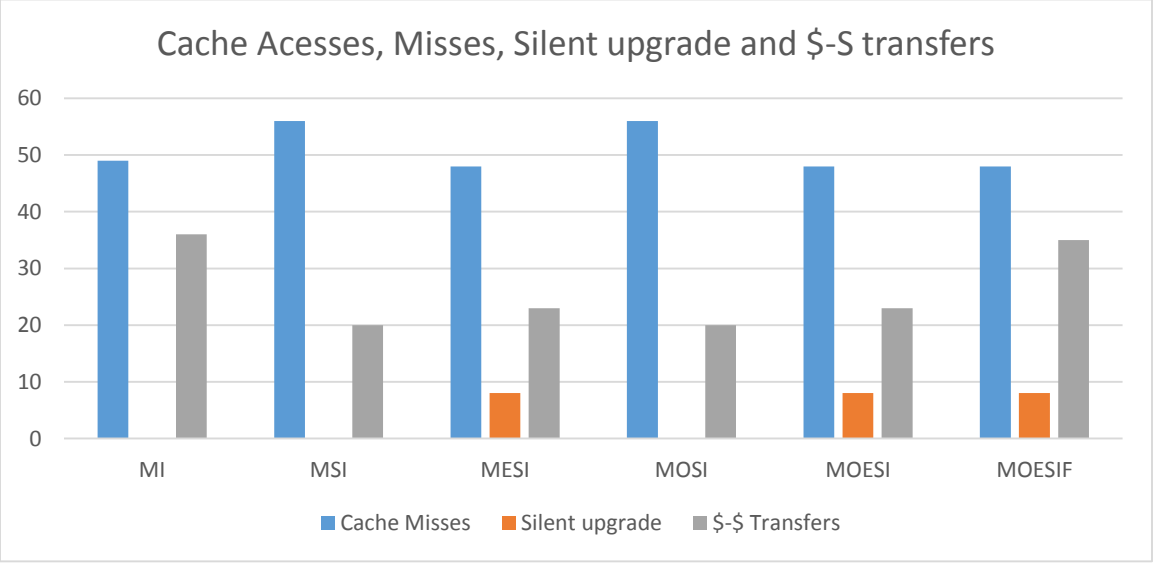
Observations:

1. MOESIF protocol has the best run-time.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding Exclusive state to MSI doesn't change the run-time drastically. Thus, this trace doesn't benefit from E-state significantly which can be observed in the fact that MOESI and MOSI have very close run-time. Also, adding E-state only gives 1 silent upgrade hence the performance gain by adding E is not worth it.
4. Adding owned state reduces run-time significantly by increasing $-$ transfer. It can be observed in the fact that MSI had 7 $-$ transfers whereas MOSI had 19 $-$ transfers.
5. Adding F-state provides improvement in run-time by increasing $-$ transfers by 6.
6. Thus, MI requires very little states but gives very good performance. But MI also produces very large $-$ transfers. MOESIF has just 28 $-$ transfers and produces best performance. However, we could have eliminated E-state and had MOSIF as E-doesn't add much performance. Note that adding more $-$ transfer increases the traffic on bus.
7. As O and E provide significant improvement this experiment probably has write-shared and read-only patterns.

EXPERIMENT#3

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	1427	49	200	0	36
MSI	3723	56	200	0	20
MESI	2607	48	200	8	23
MOSI	3723	56	200	0	20
MOESI	2607	48	200	8	23
MOESIF	1425	48	200	8	35



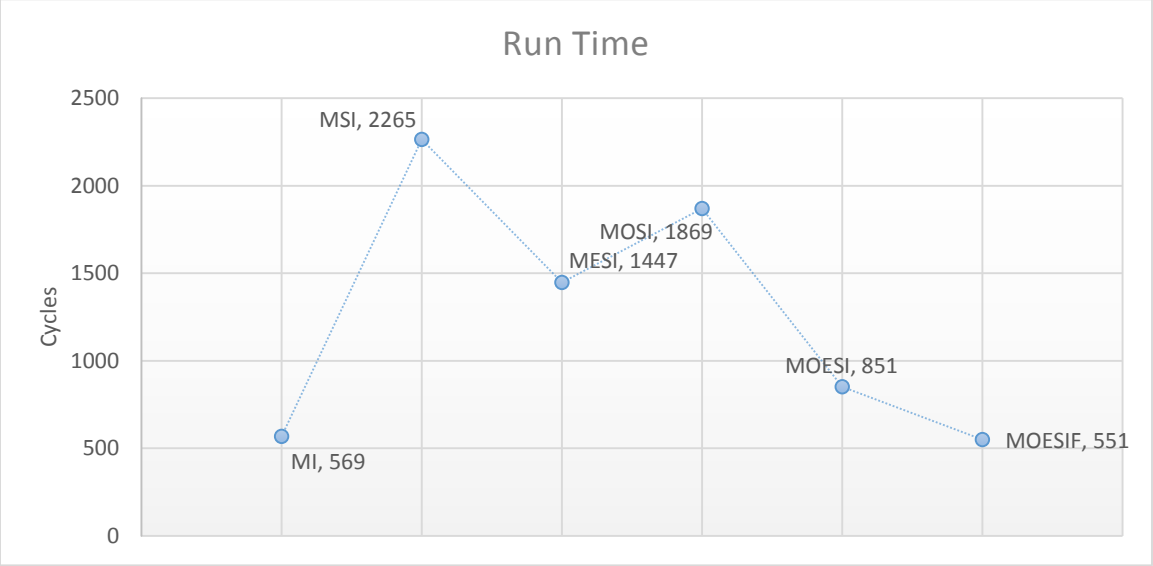


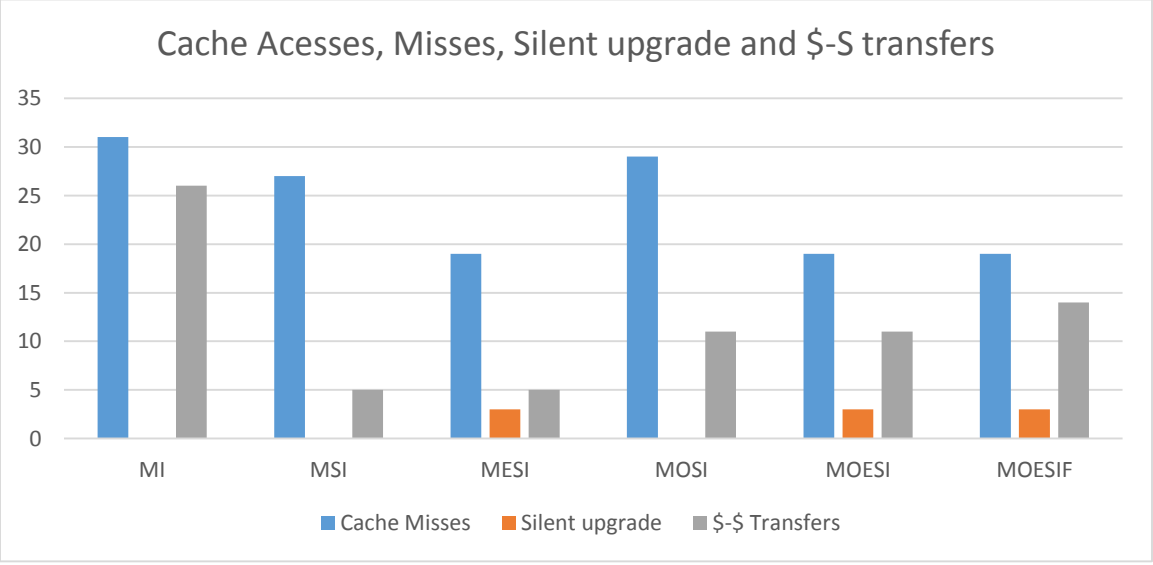
Observations:

1. MI and MOESIF protocol has the best run-time.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned state to MSI doesn't change the run-time drastically. Thus, this trace doesn't benefit from O-state significantly which can be observed in the fact that MOESI and MESI have very close run-time.
4. Adding exclusive state reduces run-time significantly by increasing silent upgrades. It can be observed in the fact that adding E-state gives 8 silent upgrade.
5. Adding F-state provides significant improvement in run-time by increasing $-$ transfers by 12.
6. Thus, MI requires very little states but gives very good performance. Both MI and MOESIF produce roughly equal $-$ transfers, hence, MI is the best suited protocol for this experiment.
7. As E and F provide significant improvement this experiment probably has private and read-only patterns.

EXPERIMENT#4

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	569	31	60	0	26
MSI	2265	27	60	0	5
MESI	1447	19	60	3	5
MOSI	1869	29	60	0	11
MOESI	851	19	60	3	11
MOESIF	551	19	60	3	14



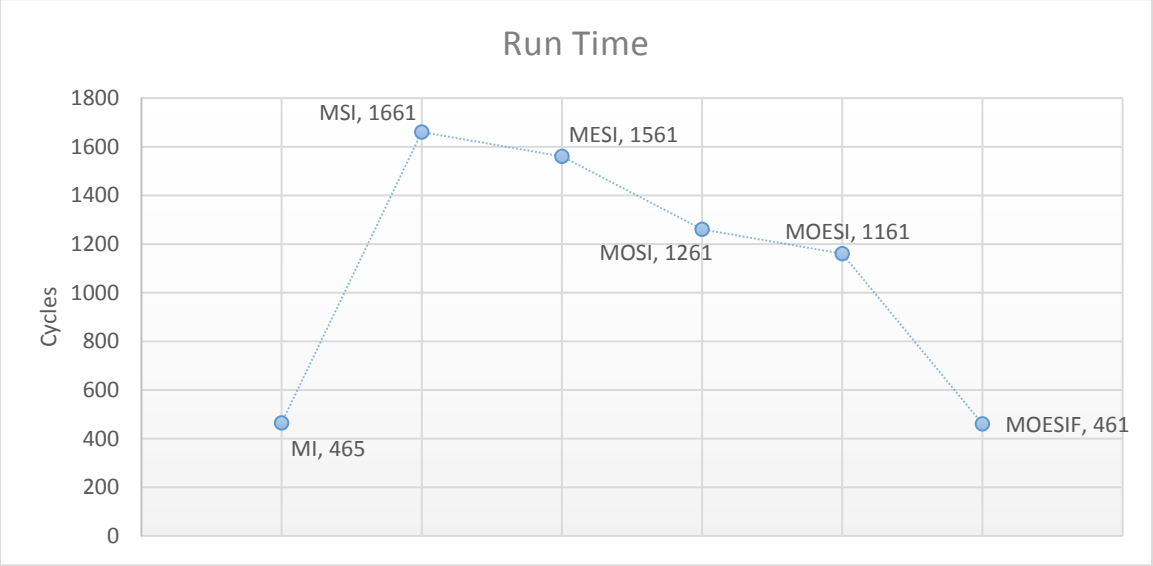


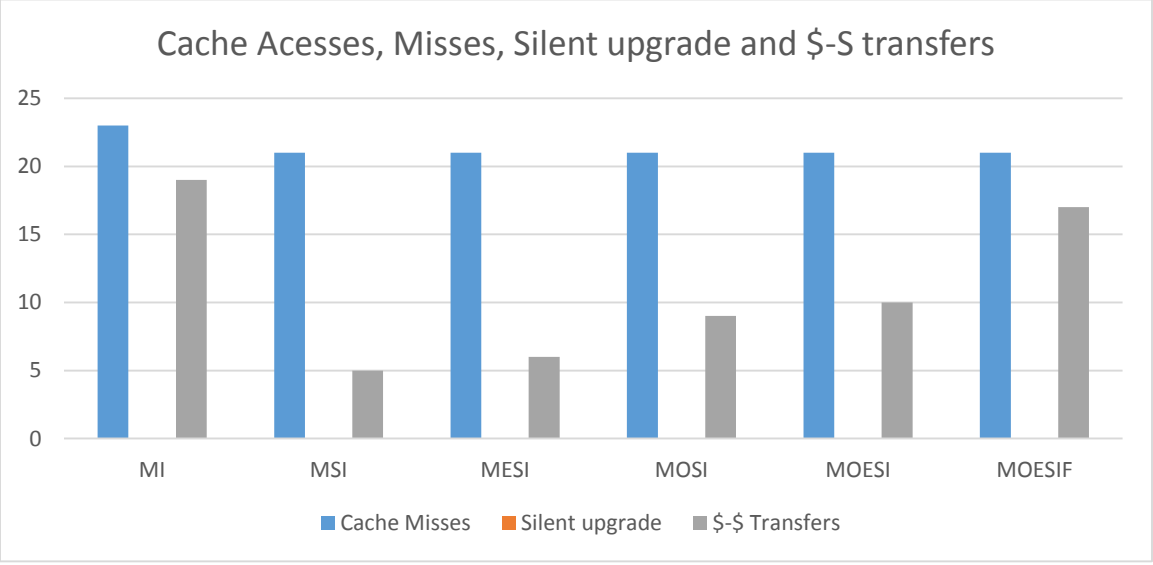
Observations:

1. MI and MOESIF protocol has the best run-time. However, MI produces 26 $-$ transfers compared to 14 $-$ transfers in MOESIF protocol. Thus, if we are concerned about bus traffic then we should choose MOESIF protocol for this experiment.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned state reduces run-time significantly by increasing $-$ transfer. It can be observed in the fact that MSI had 5 $-$ transfers whereas MOSI had 11 $-$ transfers.
4. Adding exclusive state reduces run-time significantly by increasing silent upgrades. It can be observed in the fact that adding E-state gives 3 silent upgrade.
5. Adding F-state provides significant improvement in run-time by increasing $-$ transfers by 3.
6. In this experiment O, E and F states provide significant improvement in performance making MOESIF a justifiable choice.
7. As O, E and F provide significant improvement this experiment probably has write-shared and read-only and private patterns.

EXPERIMENT#5

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	465	23	37	0	19
MSI	1661	21	37	0	5
MESI	1561	21	37	0	6
MOSI	1261	21	37	0	9
MOESI	1161	21	37	0	10
MOESIF	461	21	37	0	17



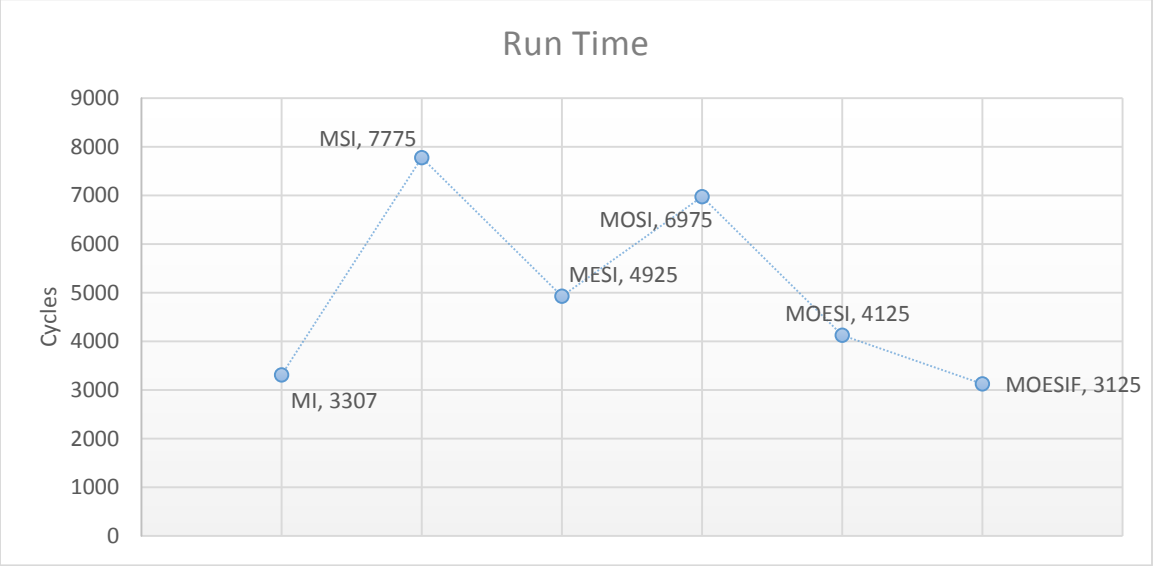


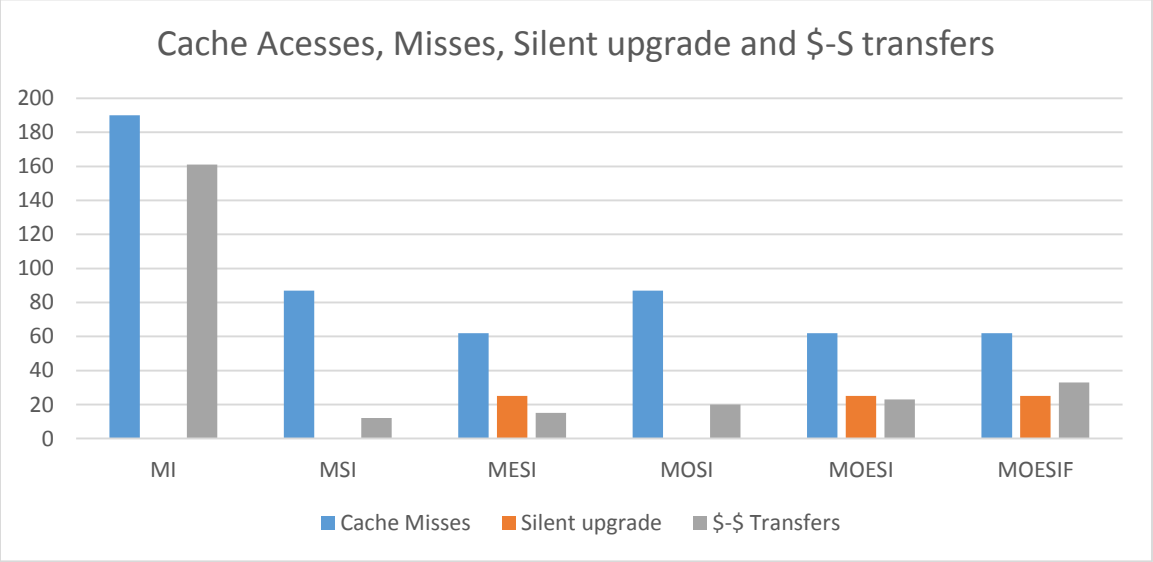
Observations:

1. MI and MOESIF protocol has the best run-time. Both MI and MOESIF produce roughly equal $-$ transfers, hence, MI is the best suited protocol for this experiment.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned and exclusive states improves performance by increasing $-$ transfers.
4. However, adding F-state provides significant improvement in run-time by increasing $-$ transfers by 7.
5. Thus MI is the best choice for this experiment.
6. As there is heavy dependent on F this experiment has read-only sharing pattern.

EXPERIMENT#6

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	3307	190	747	0	161
MSI	7775	87	747	0	12
MESI	4925	62	747	25	15
MOSI	6975	87	747	0	20
MOESI	4125	62	747	25	23
MOESIF	3125	62	747	25	33



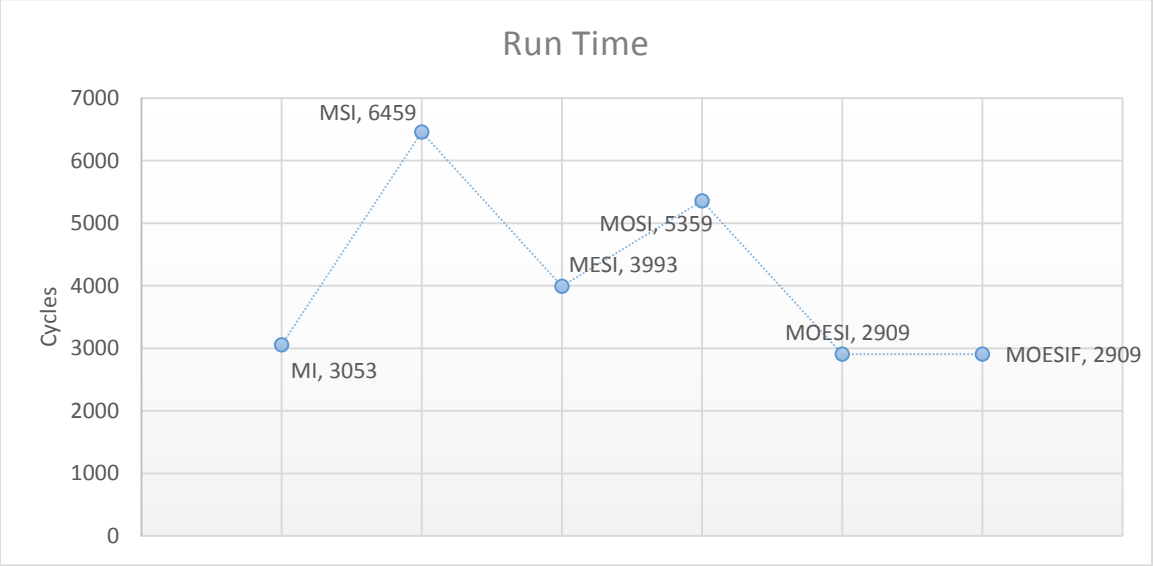


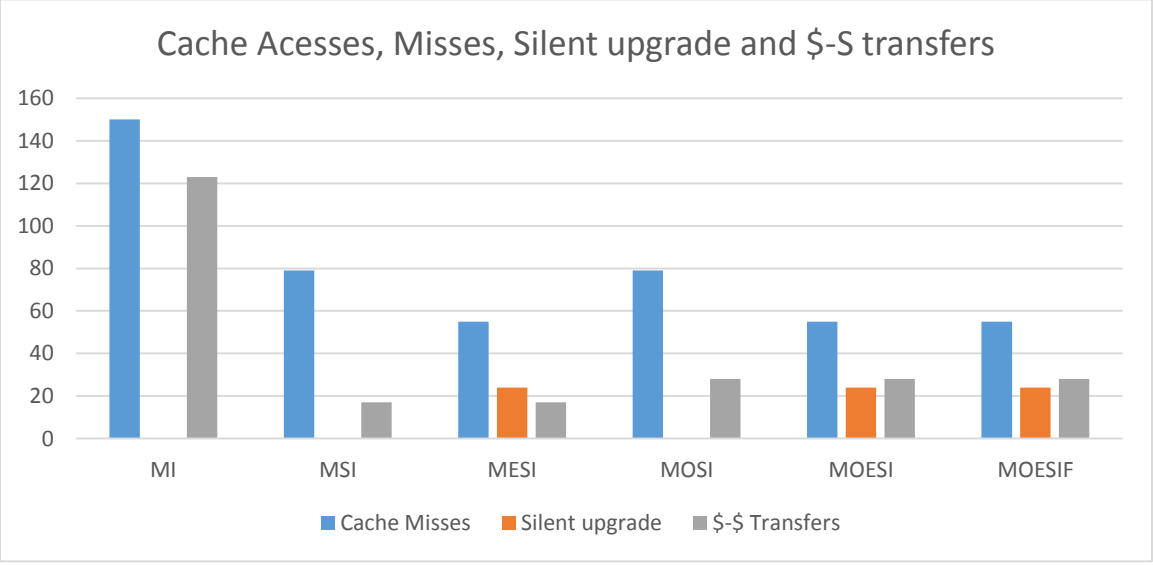
Observations:

1. MI and MOESIF protocol has the best run-time. However, MI produces 161 $-$ transfers compared to 33 $-$ transfers in MOESIF protocol. Thus, we should choose MOESIF protocol for this experiment as MI will cause a huge increase in bus traffic.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned state reduces run-time by increasing $-$ transfer. It can be observed in the fact that MSI had 12 $-$ transfers whereas MOSI had 20 $-$ transfers. However, adding E-state results in better performance compared to O-state for this experiment.
4. Adding exclusive state reduces run-time significantly by increasing silent upgrades. It can be observed in the fact that adding E-state gives 25 silent upgrade.
5. Adding F-state provides significant improvement in run-time by increasing $-$ transfers by 10.
6. In this experiment O, E and F states provide significant improvement in performance making MOESIF a justifiable choice.
7. As E and F provides significant improvement this experiment probably has private and read-only pattern.

EXPERIMENT#7

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	3053	150	952	0	123
MSI	6459	79	952	0	17
MESI	3993	55	952	24	17
MOSI	5359	79	952	0	28
MOESI	2909	55	952	24	28
MOESIF	2909	55	952	24	28



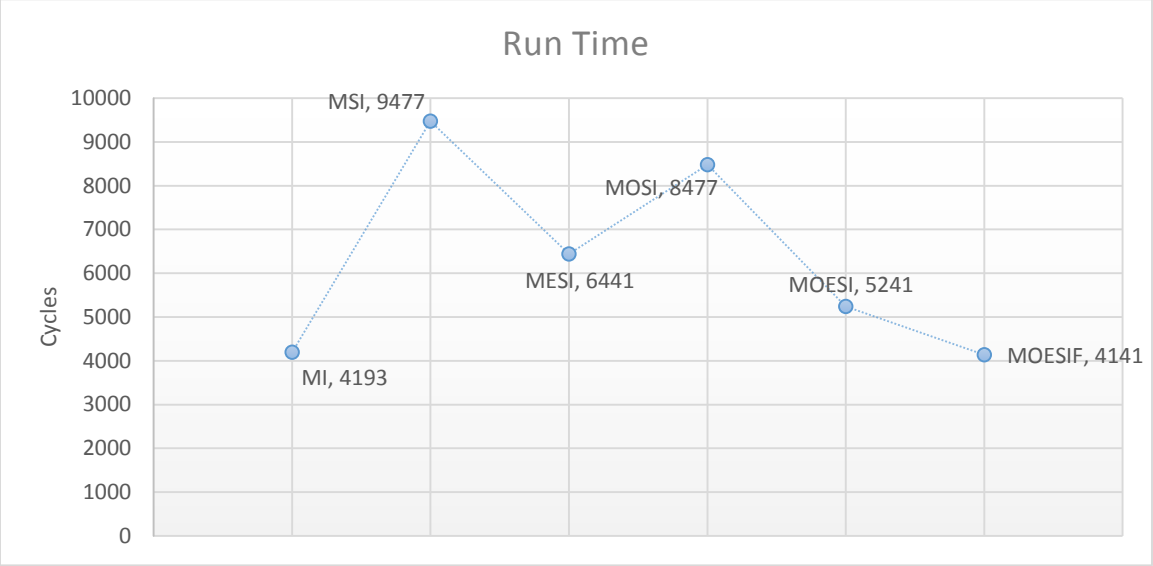


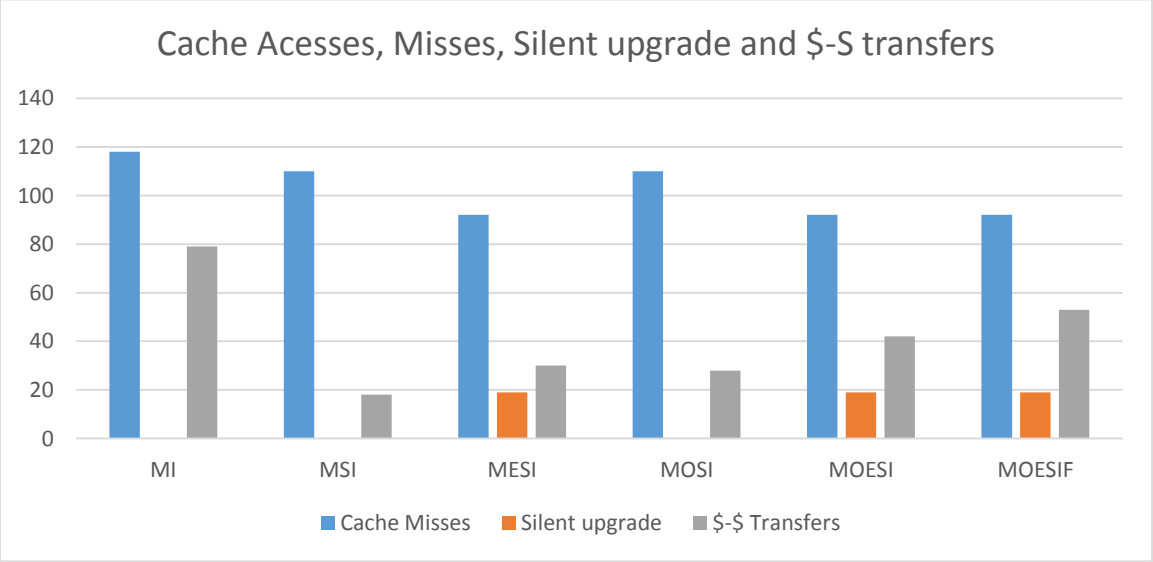
Observations:

1. MI, MOESI and MOESIF protocols have the best run-time. However, MI produces 123 $-$ transfers compared to 28 $-$ transfers in MOESI and MOESIF protocols. Thus, we should choose MOESI protocol for this experiment as MI will cause a huge increase in bus traffic.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned state reduces run-time by increasing $-$ transfer. It can be observed in the fact that MSI had 17 $-$ transfers whereas MOSI had 28 $-$ transfers. However, adding E-state results in better performance compared to O-state for this experiment.
4. Adding exclusive state reduces run-time significantly by increasing silent upgrades. It can be observed in the fact that adding E-state gives 24 silent upgrade.
5. Adding F-state provides no improvement in run-time.
6. In this experiment O and E states provide significant improvement in performance making MOESI a justifiable choice.
7. As O and E provides significant improvement this experiment probably has private and write-shared pattern.

EXPERIMENT#8

Protocol	Run Time	Cache Misses	Cache Accesses	Silent upgrade	$-$ Transfers
MI	4193	118	800	0	79
MSI	9477	110	800	0	18
MESI	6441	92	800	19	30
MOSI	8477	110	800	0	28
MOESI	5241	92	800	19	42
MOESIF	4141	92	800	19	53





Observations:

1. MI, MOESI and MOESIF protocols have the best run-time. However, MI produces 79 $-$ transfers compared to 53 $-$ transfers in MOESIF protocols. Thus, we should choose MOESIF protocol for this experiment as MI will cause a huge increase in bus traffic.
2. Performance of MSI is worse compared to MI as S-state does not allow $-$ transfers.
3. Adding owned state reduces run-time by increasing $-$ transfer. It can be observed in the fact that MSI had 18 $-$ transfers whereas MOSI had 28 $-$ transfers. However, adding E-state results in better performance compared to O-state for this experiment.
4. Adding exclusive state reduces run-time significantly by increasing silent upgrades. It can be observed in the fact that adding E-state gives 19 silent upgrade.
5. Adding F-state provides significant improvement in run-time by increasing $-$ transfers by 11.
6. In this experiment E and F states provide significant improvement in performance making MOESIF a justifiable choice.
7. As E and F provides significant improvement this experiment probably has private and read-only pattern.

Inferences:

Below table summarizes experimentation results:

Experiment#	Best suited protocol	Possible Sharing Pattern
1	MOSI	write-shared
2	MOESIF	write-shared and read-only
3	MI	private and read-only
4	MOESIF	write-shared, read-only and private patterns
5	MI	read-only
6	MOESIF	private and read-only
7	MOESI	private and write-shared
8	MOESIF	private and read-only