ECE1166 - Cache-Coherence Protocol Analysis

Zachary M. Mattis

March 22, 2019

I. State-Transition Diagrams

1. MSI w/ BusUpgr

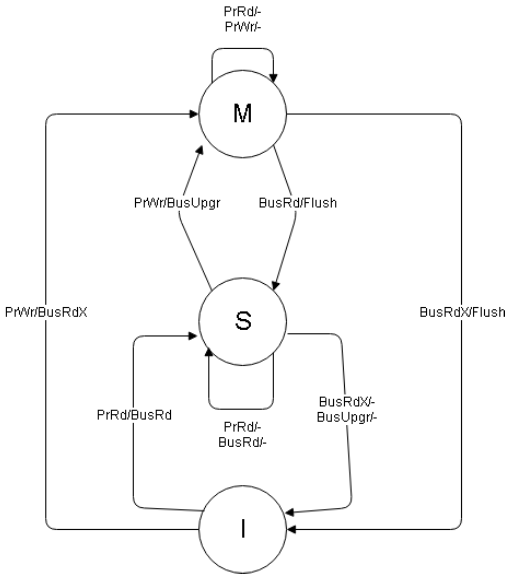

Figure 1: MSI w/ BusUpgr

2. MESI w/ no c2c sharing

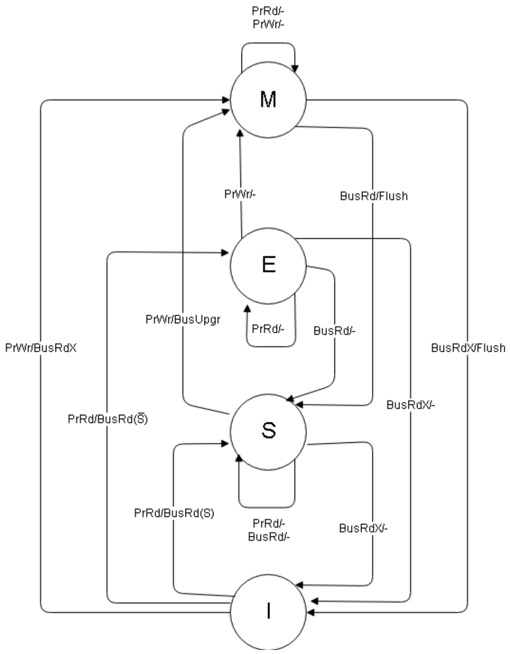

Figure 2: MESI w/ no c2c sharing

3. MESI w/ M–I downgrade on BusRd

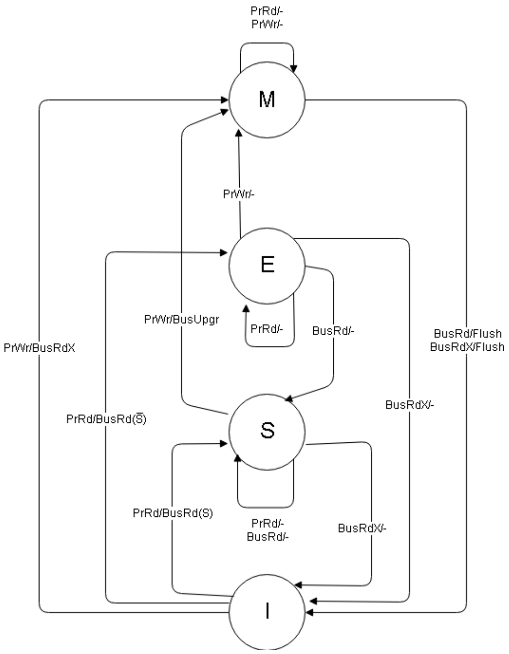

Figure 3: MESI w/ M–I downgrade

Note: In the above diagram, cache controller in M state will transition to the I state on an observed BusRd, but will not assert its S line, allowing the reading cache controller to transition to E state via BusRd($\bar{S}$). It will still issue a flush in order to synchronize the modified block with lower memory.

4. MESI w/ c2c sharing (M,E)

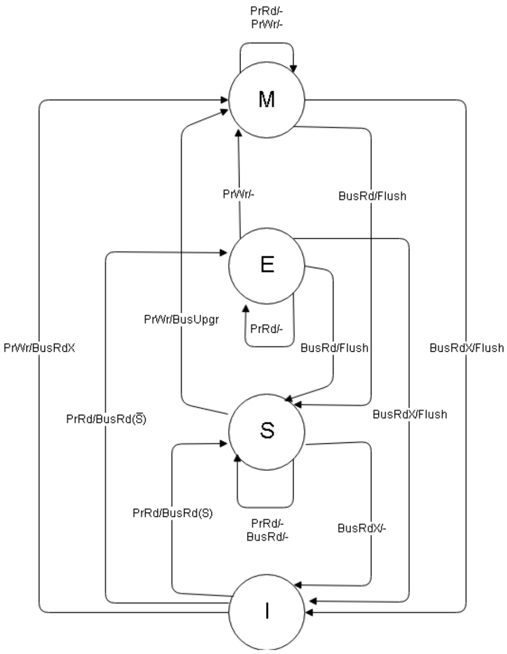

Figure 4: MESI w/ c2c sharing

Note: With cache-to-cache sharing enabled, the Flush commands issued by the M, E state cache controllers now imply direct sharing of the requested block.

5. MESI w/ max c2c sharing via arbitration

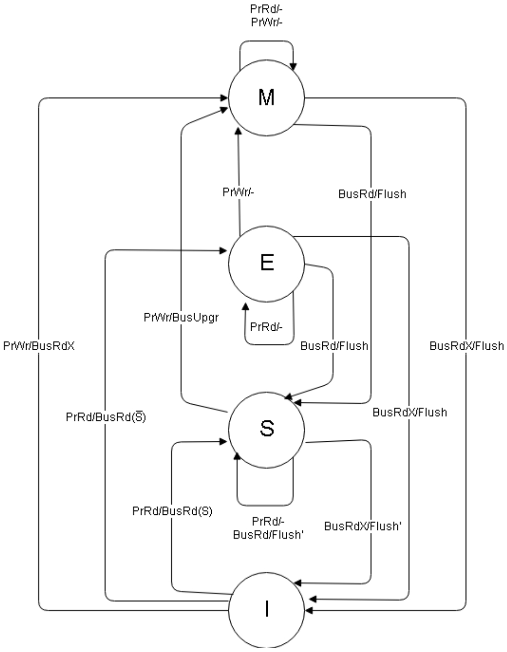

Figure 5: MESI w/ arbitration c2c sharing

Note: Flush' implies cache blocks shared via peer arbitration to the requesting cache controller.

6. MOESI

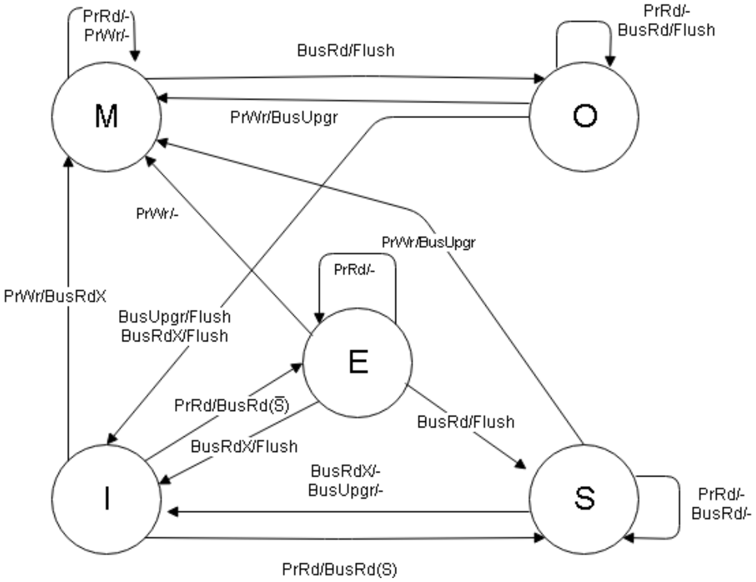

Figure 6: MOESI

7. Dragon w/ max c2c sharing, no arbitration

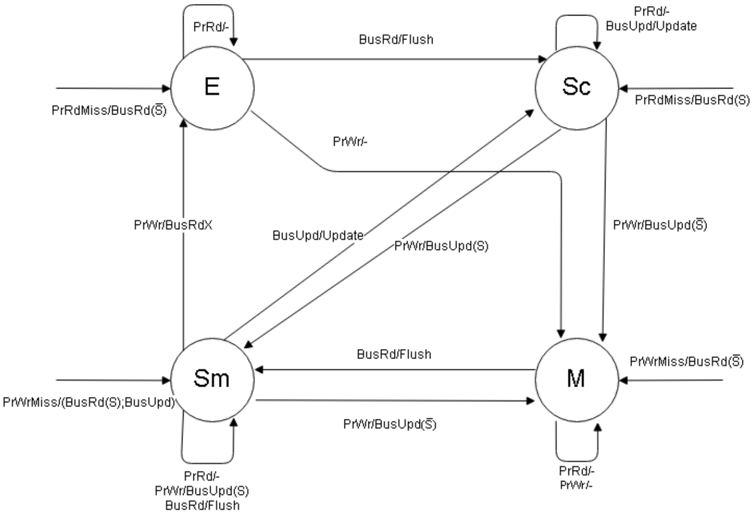

Figure 7: Dragon

II. Memory Trace Campaigns

1. MSI w/ BusUpgr

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	S	-	-	BusRd	200
2	w1	M	-	-	BusUpgr	40
3	r2	S	S	-	BusRd/Flush	200
4	w2	I	M	-	BusUpgr	40
5	r3	I	S	S	BusRd/Flush	200
6	w3	I	I	M	BusUpgr	40
7	r1	S	I	S	BusRd/Flush	200
8	w1	M	I	I	BusUpgr	40
					Total	**960**

Table 1: Access Sequence 1, Protocol 1
MSI w/ BusUpgr

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	S	-	-	BusRd	200
2	r2	S	S	-	BusRd	200
3	w1	M	I	-	BusUpgr	40
4	r3	S	I	S	BusRd/Flush	200
5	w2	I	M	I	BusRdX	200
6	r1	S	S	I	BusRd/Flush	200
7	w3	I	I	M	BusRdX	200
8	r2	I	S	S	BusRd/Flush	200
9	r3	I	S	S	-	1
					Total	**1441**

Table 2: Access Sequence 2, Protocol 1
MSI w/ BusUpgr

2. MESI w/ no c2c sharing

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	S	S	-	BusRd(S)/Flush	200
4	w2	I	M	-	BusUpgr	40
5	r3	I	S	S	BusRd(S)/Flush	200
6	w3	I	I	M	BusUpgr	40
7	r1	S	I	S	BusRd(S)	200
8	w1	M	I	I	BusUpgr	40
					Total	**921**

Table 3: Access Sequence 1, Protocol 2
MESI w/ no c2c sharing

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	S	S	-	BusRd(S)	200
3	w1	M	I	-	BusUpgr	40
4	r3	S	I	S	BusRd(S)/Flush	200
5	w2	I	M	I	BusRdX	200
6	r1	S	S	I	BusRd(S)/Flush	200
7	w3	I	I	M	BusRdX	200
8	r2	I	S	S	BusRd(S)/Flush	200
9	r3	I	S	S	-	1
					Total	**1441**

Table 4: Access Sequence 2, Protocol 2
MESI w/ no c2c sharing

3. MESI w/ M–I downgrade on BusRd

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	I	E	-	BusRd($\bar{S}$)/Flush	200
4	w2	I	M	-	-	1
5	r3	I	I	E	BusRd($\bar{S}$)/Flush	200
6	w3	I	I	M	-	1
7	r1	E	I	I	BusRd($\bar{S}$)/Flush	200
8	w1	M	I	I	-	1
					Total	**804**

Table 5: Access Sequence 1, Protocol 3
MESI w/ M–I downgrade on BusRd

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	S	S	-	BusRd(S)	200
3	w1	M	I	-	BusUpgr	40
4	r3	I	I	E	BusRd($\bar{S}$)/Flush	200
5	w2	I	M	I	BusRdX	200
6	r1	E	I	I	BusRd($\bar{S}$)/Flush	200
7	w3	I	I	M	BusRdX	200
8	r2	I	E	I	BusRd($\bar{S}$)/Flush	200
9	r3	I	S	S	BusRd(S)	200
					Total	**1640**

Table 6: Access Sequence 2, Protocol 3
MESI w/ M–I downgrade on BusRd

4. MESI w/ c2c sharing (M,E)

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	S	S	-	BusRd(S)/Flush	80
4	w2	I	M	-	BusUpgr	40
5	r3	I	S	S	BusRd(S)/Flush	80
6	w3	I	I	M	BusUpgr	40
7	r1	S	I	S	BusRd(S)/Flush	80
8	w1	M	I	I	BusUpgr	40
					Total	**561**

Table 7: Access Sequence 1, Protocol 4
MESI w/ c2c sharing (M,E)

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	S	S	-	BusRd(S)/Flush	80
3	w1	M	I	-	BusUpgr	40
4	r3	S	I	S	BusRd(S)/Flush	80
5	w2	I	M	I	BusRdX	200
6	r1	S	S	I	BusRd(S)/Flush	80
7	w3	I	I	M	BusRdX	200
8	r2	I	S	S	BusRd(S)/Flush	80
9	r3	I	S	S	-	1
					Total	**961**

Table 8: Access Sequence 2, Protocol 4
MESI w/ c2c sharing (M,E)

5. MESI w/ max c2c sharing via arbitration

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	S	S	-	BusRd(S)/Flush	80
4	w2	I	M	-	BusUpgr	40
5	r3	I	S	S	BusRd(S)/Flush	80
6	w3	I	I	M	BusUpgr	40
7	r1	S	I	S	BusRd(S)/Flush	80
8	w1	M	I	I	BusUpgr	40
					Total	**561**

Table 9: Access Sequence 1, Protocol 5
MESI w/ max c2c sharing via arbitration

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	S	S	-	BusRd(S)/Flush	80
3	w1	M	I	-	BusUpgr	40
4	r3	S	I	S	BusRd(S)/Flush	80
5	w2	I	M	I	BusRdX/Flush'	100
6	r1	S	S	I	BusRd(S)/Flush	80
7	w3	I	I	M	BusRdX/Flush'	100
8	r2	I	S	S	BusRd(S)/Flush	80
9	r3	I	S	S	-	1
					Total	**761**

Table 10: Access Sequence 2, Protocol 5
MESI w/ max c2c sharing via arbitration

6. MOESI

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	O	S	-	BusRd(S)/Flush	80
4	w2	I	M	-	BusUpgr	40
5	r3	I	O	S	BusRd(S)/Flush	80
6	w3	I	I	M	BusUpgr	40
7	r1	S	I	O	BusRd(S)/Flush	80
8	w1	M	I	I	BusUpgr	40
					Total	**561**

Table 11: Access Sequence 1, Protocol 6
MOESI

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	S	S	-	BusRd(S)/Flush	80
3	w1	M	I	-	BusUpgr	40
4	r3	O	I	S	BusRd(S)/Flush	80
5	w2	I	M	I	BusRdX/Flush	80
6	r1	S	O	I	BusRd(S)/Flush	80
7	w3	I	I	M	BusRdX/Flush	80
8	r2	I	S	O	BusRd(S)/Flush	80
9	r3	I	S	O	-	1
					Total	**721**

Table 12: Access Sequence 2, Protocol 6
MOESI

7. Dragon w/ max c2c sharing, no arbitration

#	**Action**	P_1	P_2	P_3	**Bus**	**Cycles**
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	w1	M	-	-	-	1
3	r2	Sm	Sc	-	BusRd(S)/Flush	80
4	w2	Sc	Sm	-	BusUpd(S)	40
5	r3	Sc	Sm	Sc	BusRd(S)/Flush	80
6	w3	Sc	Sc	Sm	BusUpd(S)	40
7	r1	Sc	Sc	Sm	-	1
8	w1	Sm	Sc	Sc	BusUpd(S)	40
					Total	**482**

Table 13: Access Sequence 1, Protocol 7
Dragon

#	Action	P_1	P_2	P_3	Bus	Cycles
1	r1	E	-	-	BusRd($\bar{S}$)	200
2	r2	Sc	Sc	-	BusRd(S)/Flush	80
3	w1	Sm	Sc	-	BusUpd(S)	40
4	r3	Sm	Sc	Sc	BusRd(S)/Flush	80
5	w2	Sc	Sm	Sc	BusUpd(S)	40
6	r1	Sc	Sm	Sc	-	1
7	w3	Sc	Sc	Sm	BusUpd(S)	40
8	r2	Sc	Sc	Sm	-	1
9	r3	Sc	Sc	Sm	-	1
					Total	**483**

Table 14: Access Sequence 2, Protocol 7
Dragon

III. Cache Coherence Analysis

Protocol	Sequence 1	Sequence 2
MSI w/ BusUpgr	960	1441
MESI w/ no c2c sharing	921	1441
MESI w/ M–I downgrade	804	1640
MESI w/ c2c sharing	561	961
MESI w/ arbitration c2c sharing	561	761
MOESI	561	721
Dragon	482	483

Table 15: Cache Coherence Protocol Cycle Time

Based on the memory access patterns for the above cache coherence protocols, Dragon appeared to be the most successfully. This is largely due to the small time that it takes to update the word of block in other cache controllers, which is only 40 cycles. This benefit can be seen as the invalidated blocks of other (invalidation) protocols must then get their blocks from other caches controllers (80 cycles) or from lower memory (200 cycles).

The benefits of cache-to-cache sharing can be observed on the MESI protocols, both with and without arbitration. These protocols illustrated much greater improvements in total cycle time compared to their non-c2c sharing counterparts.

One of the more interesting protocols was the M–I downgrade for MESI on BusRd. This protocol showed performance improvements with data migratory patterns where the data read was then immediately modified. This benefit is observed as the subsequent write to the data block does not have to issue a BusUpgr command to the other cache controller, as M blocks are automatically demoted to I state. This has great drawbacks, as illustrated by the second memory access pattern, providing the worst overall access time. This is largely due to the pattern not taking advantage of the exclusive state to write to data, especially when their is no cache-to-cache sharing enabled.

It is not surprising that as the complexity of the cache-coherence protocol increases, the performance of it also increases, leading to lower total cycle times.