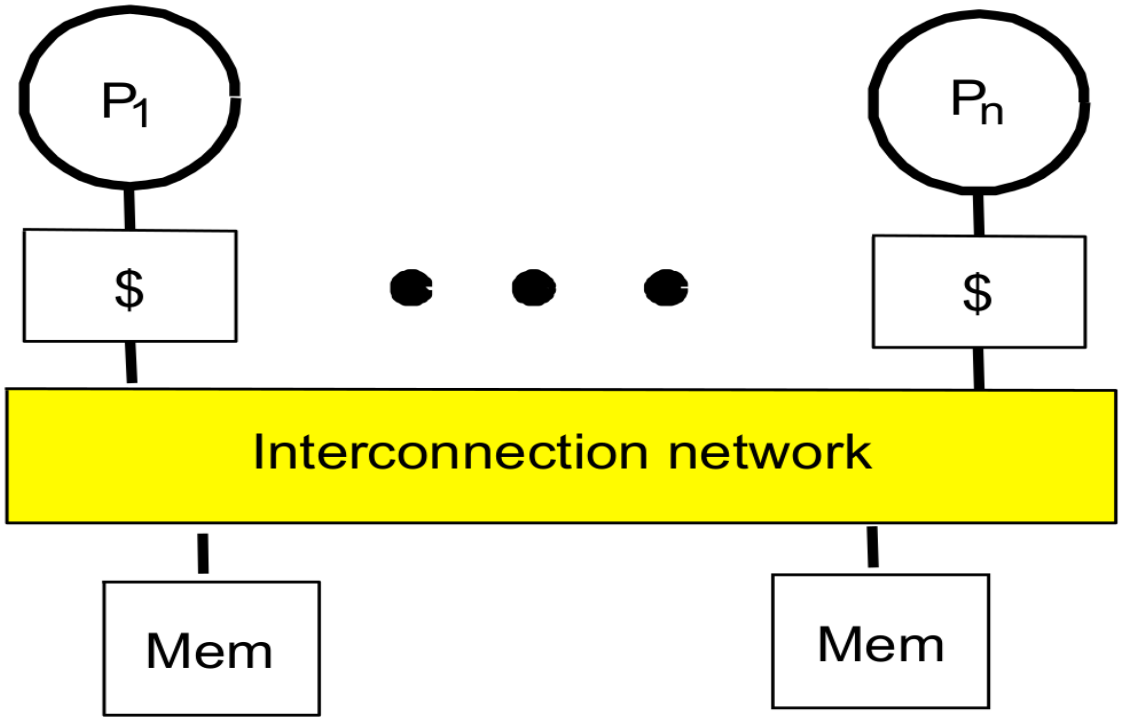
Tutorial 8

ECSE 420 – Tutorial 8

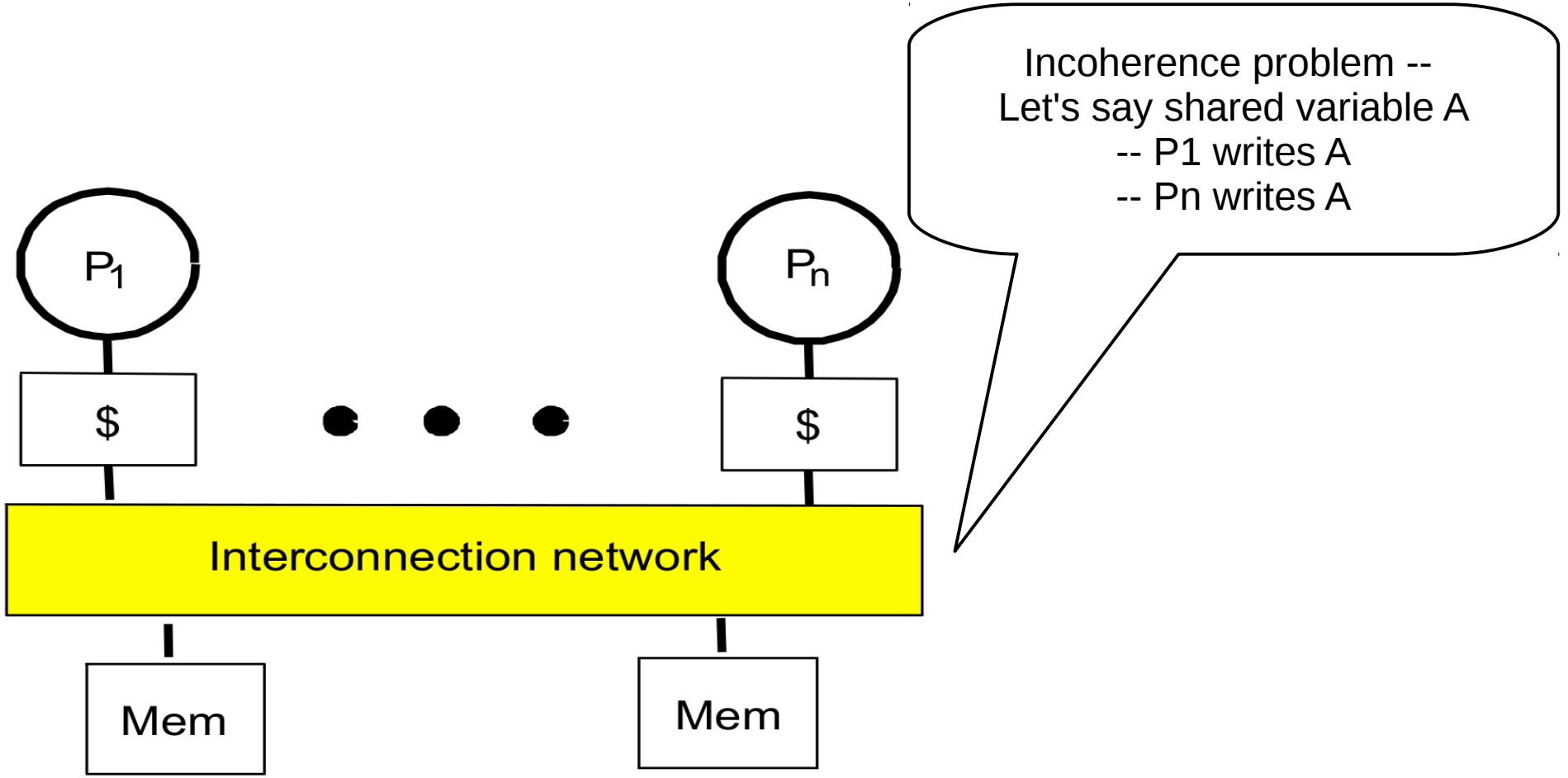
Dimitrios Stamoulis

TR 4110
November 10, 2014

Shared Memory Architectures

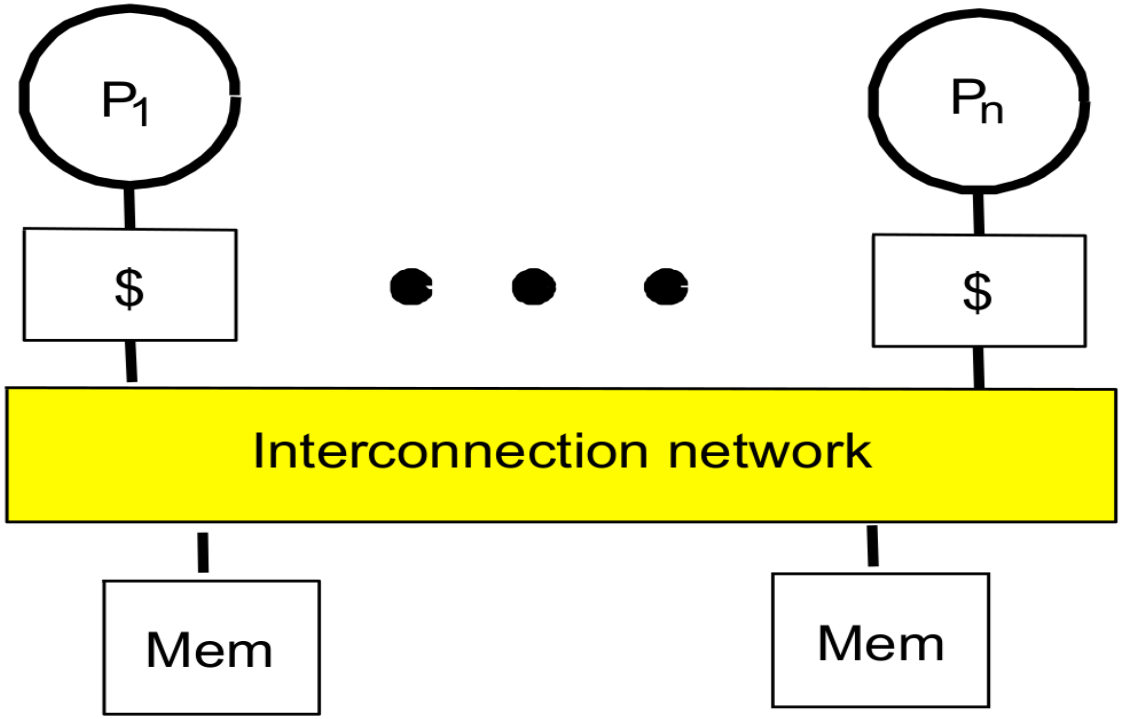


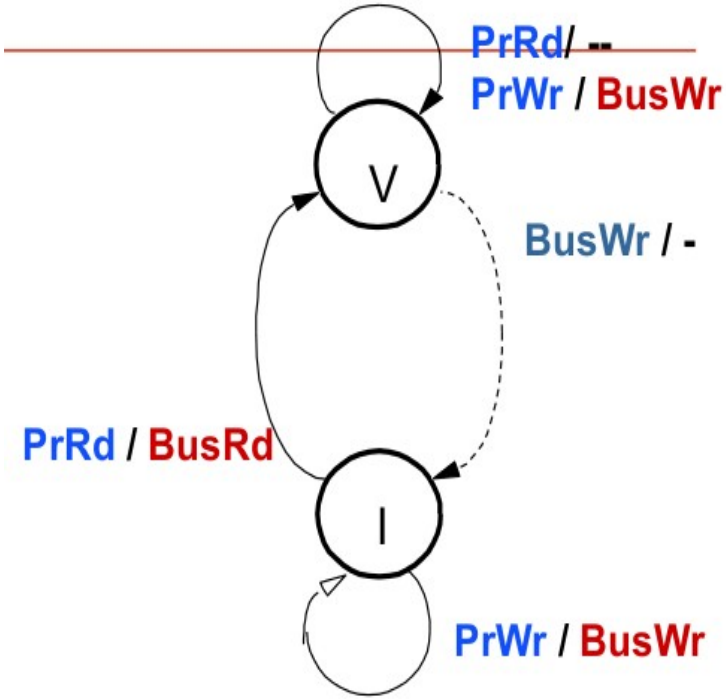
Shared Memory Architectures



Shared Memory Architectures

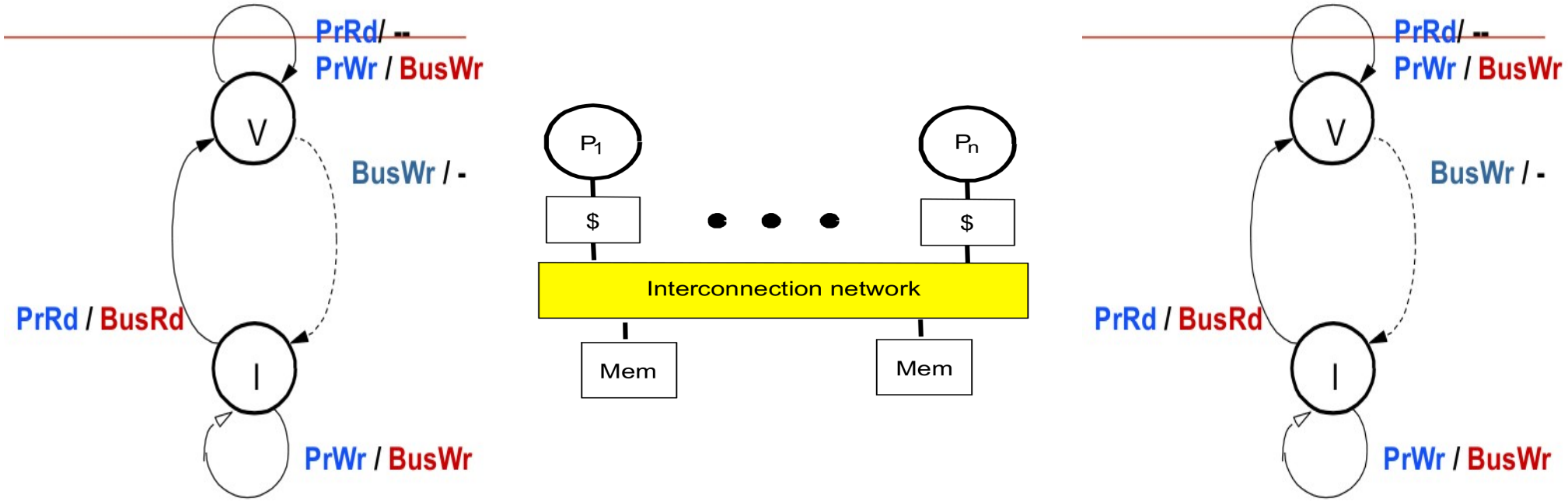
- Write-through Invalidate Protocol





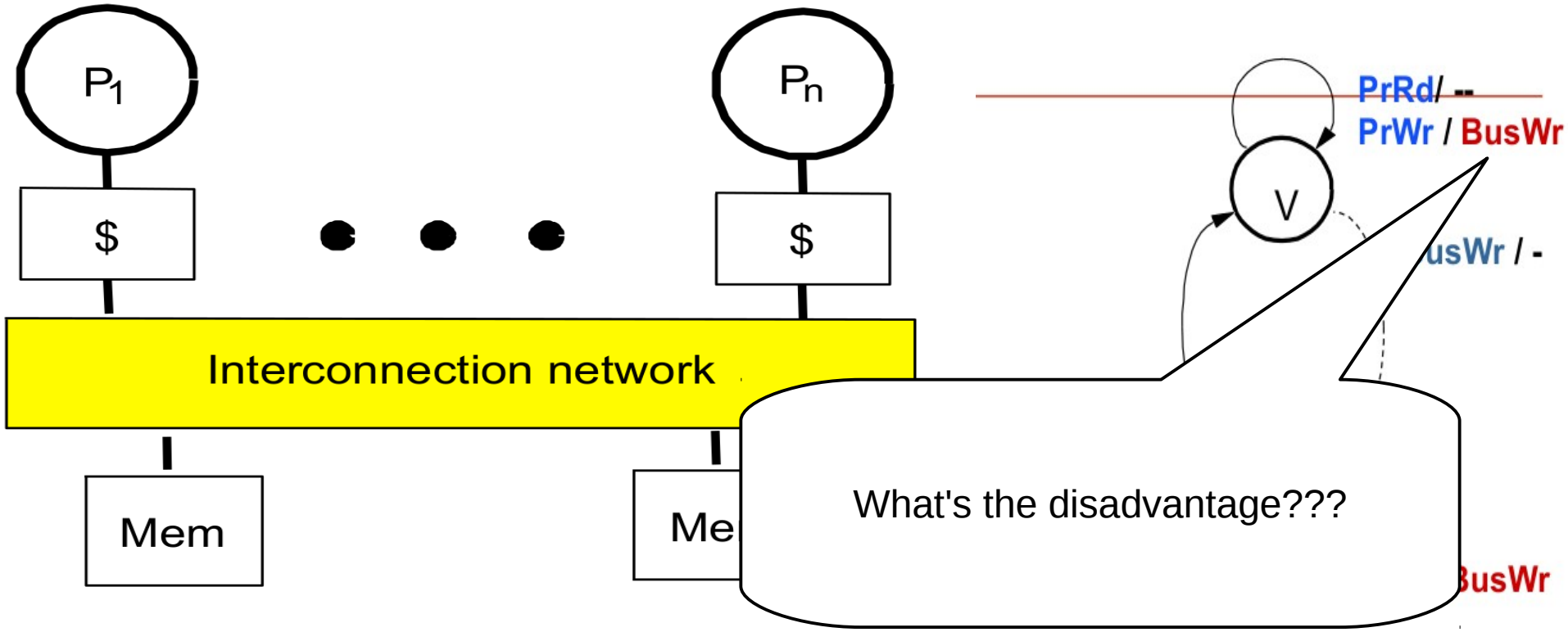
Shared Memory Architectures

- Write-through Invalidate Protocol



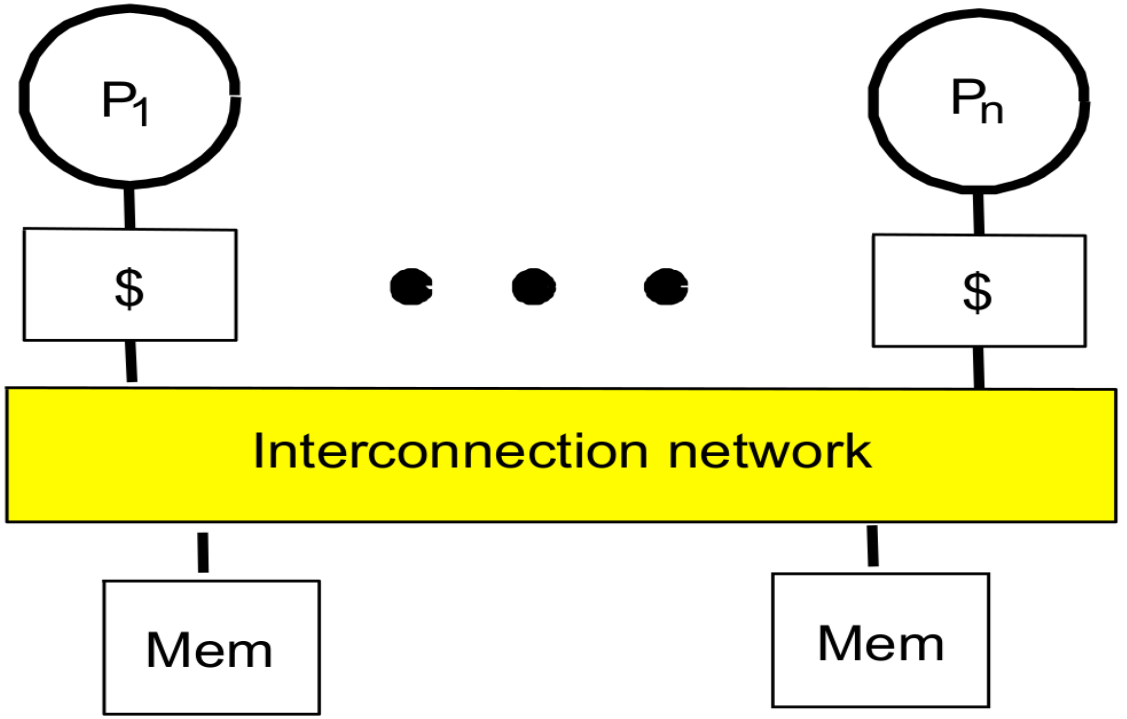
Shared Memory Architectures

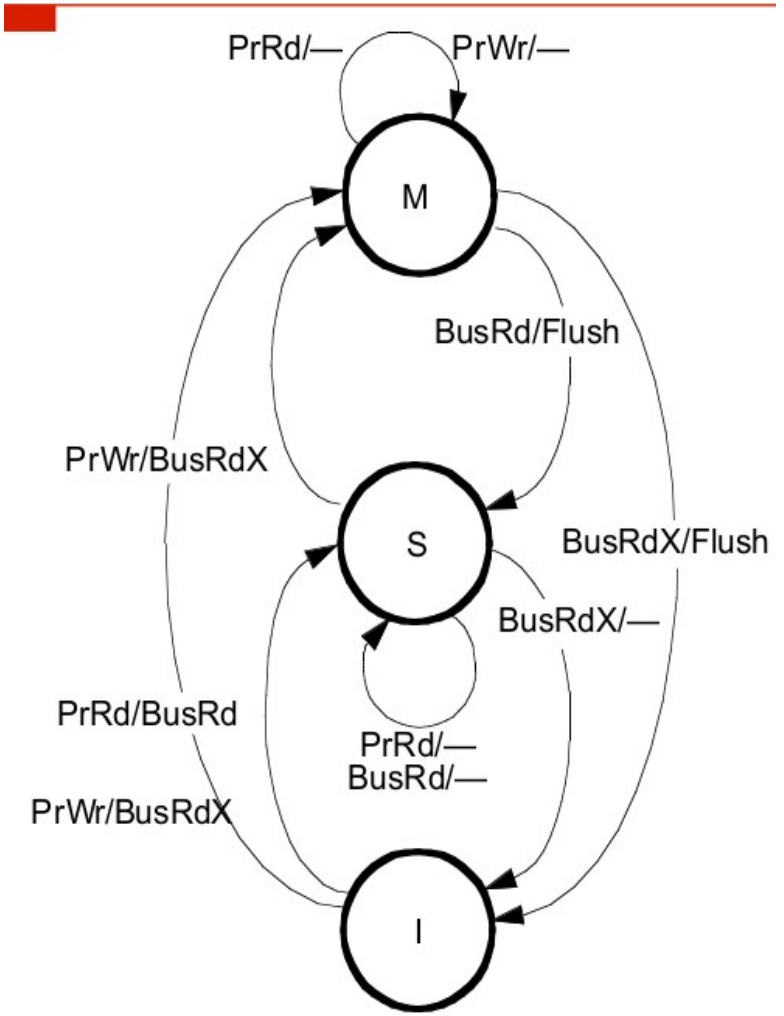
- Write-through Invalidate Protocol



Shared Memory Architectures

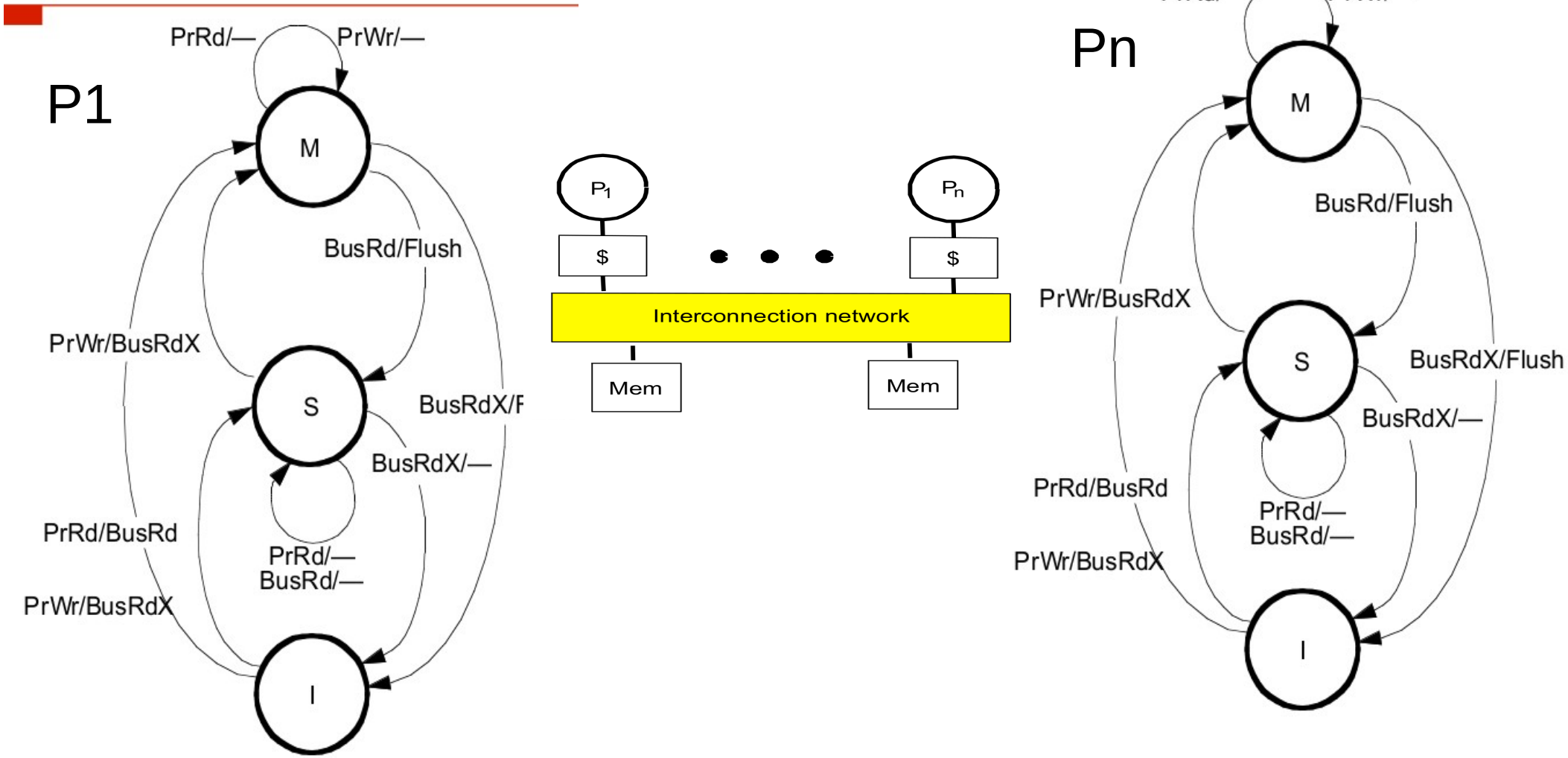
- Write-back Caches - MSI





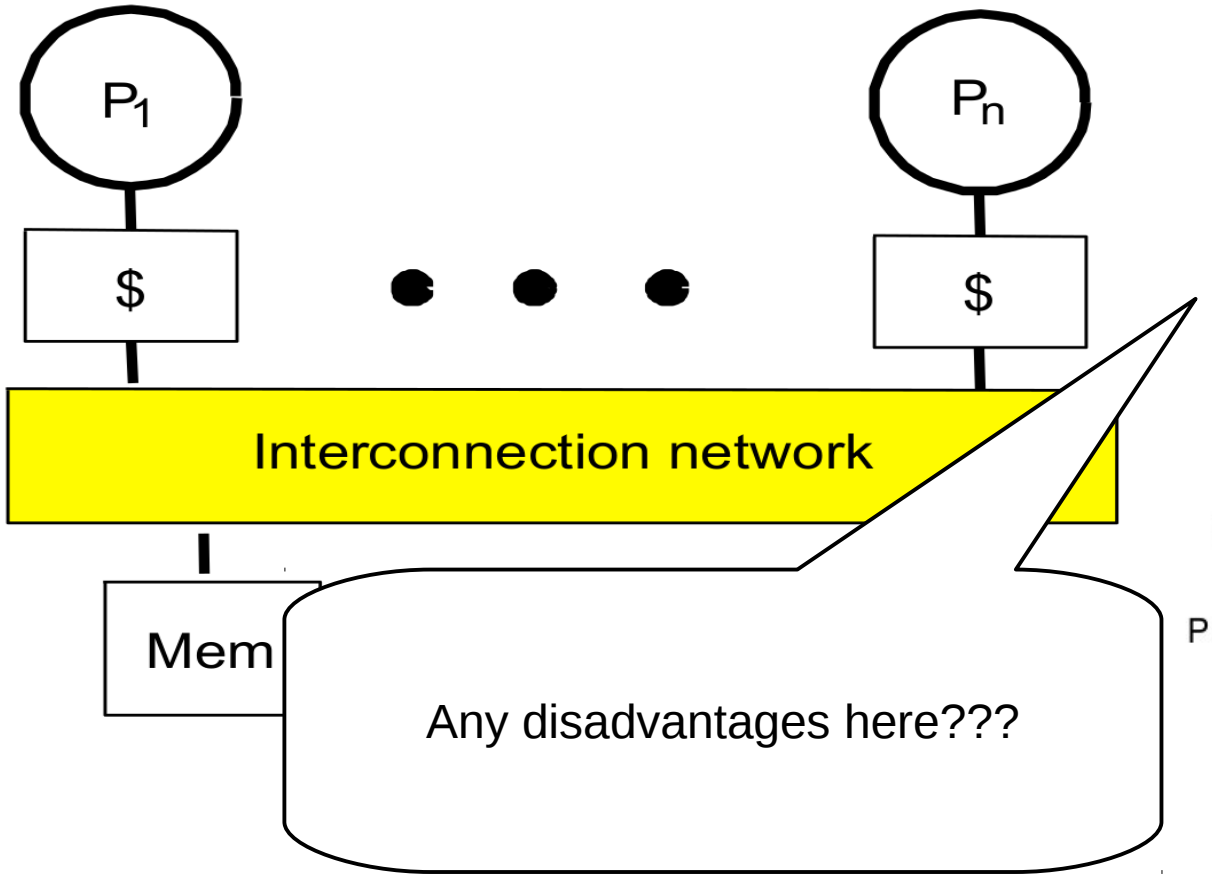
Shared Memory Architectures

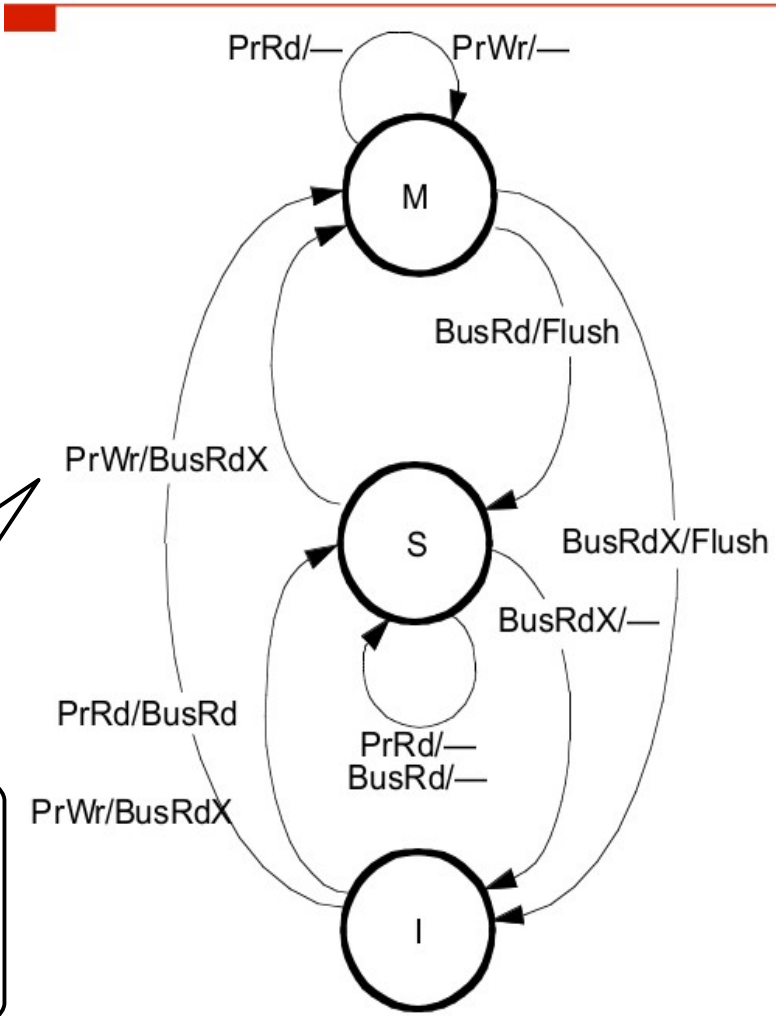
- Write-back Caches - MSI



Shared Memory Architectures

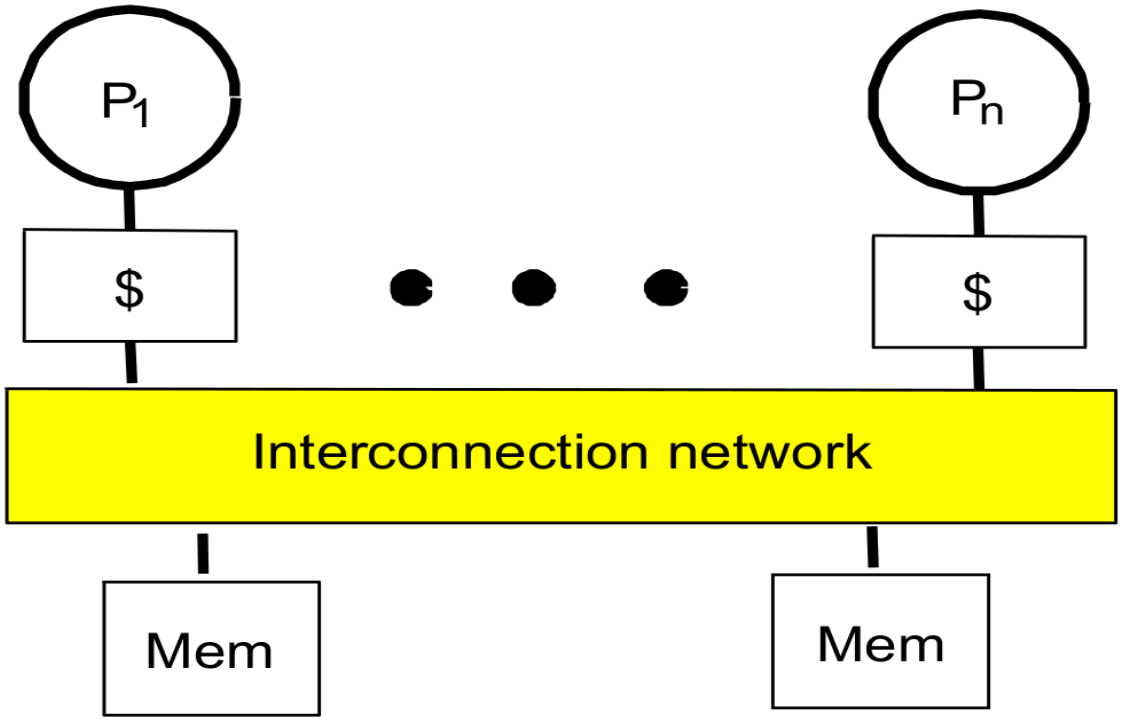
- Write-back Caches - MSI

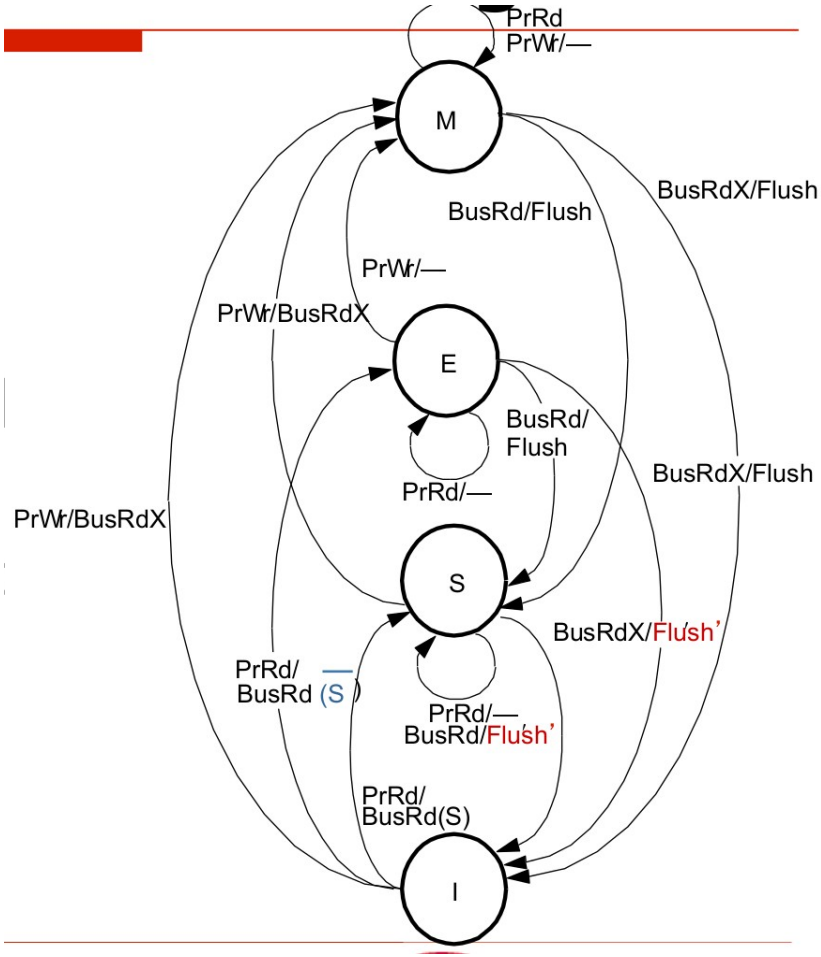




Shared Memory Architectures

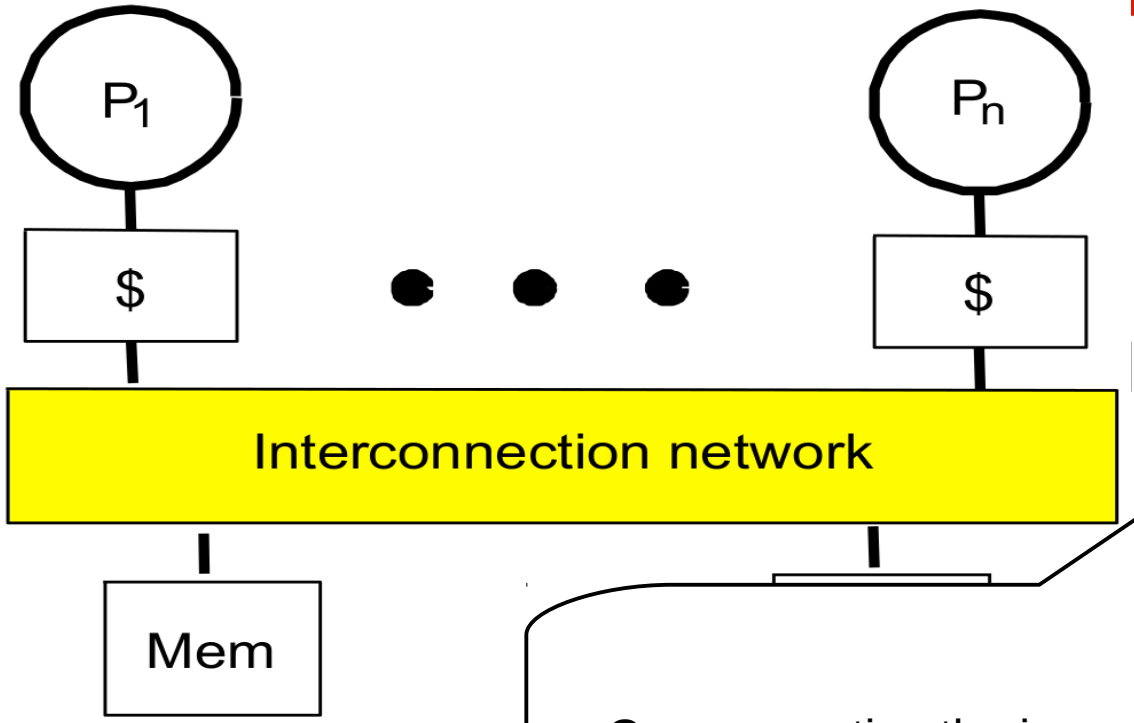
- Write-back Caches - MESI



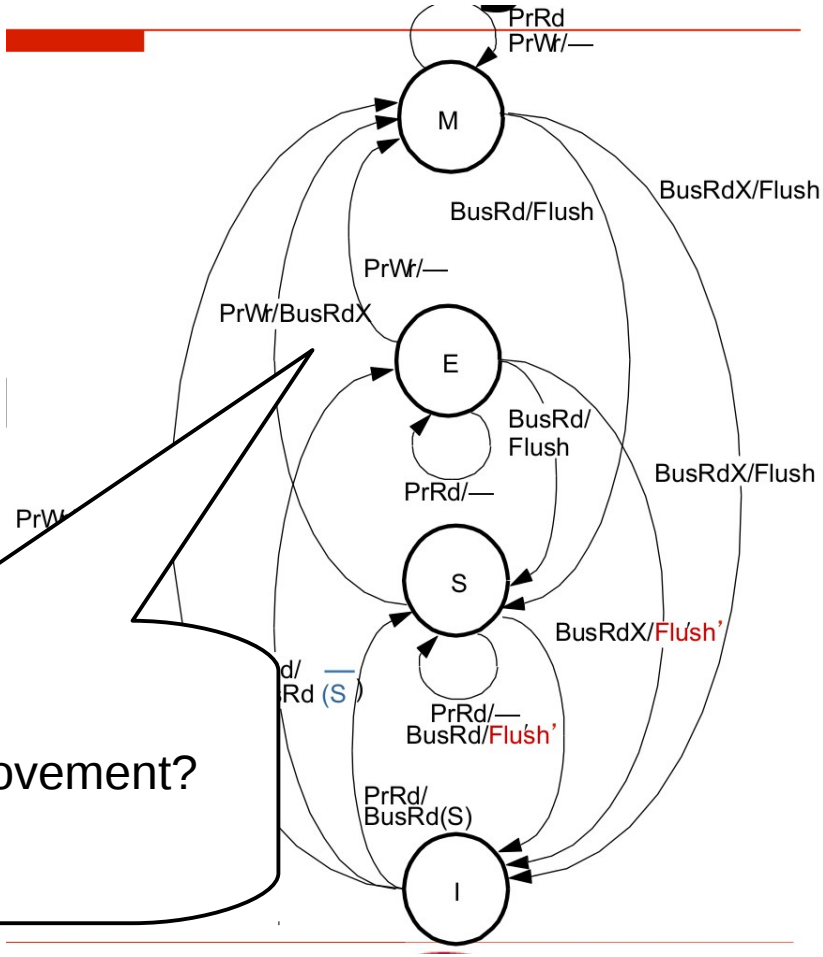


Shared Memory Architectures

- Write-back Caches - MESI

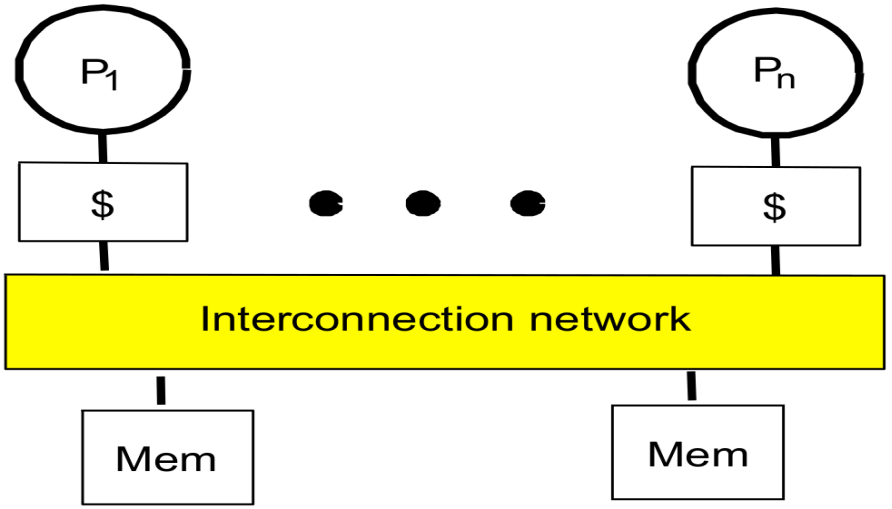


Can you notice the improvement?



Let's solve by hand an example

- 2 CPUs
- Write-back caches
- MESI protocol
- 16 bytes per line
- Caches initially empty

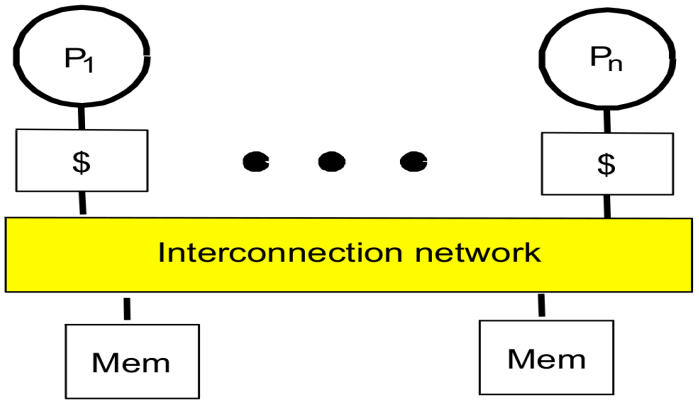


Instructions
P0 writes to 8c34
P0 reads 7c34
P1 writes to 8c30
P1 reads 7c34
P1 reads 9c38
P0 reads 8c3c
P0 writes to 8c3c

Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c

Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000



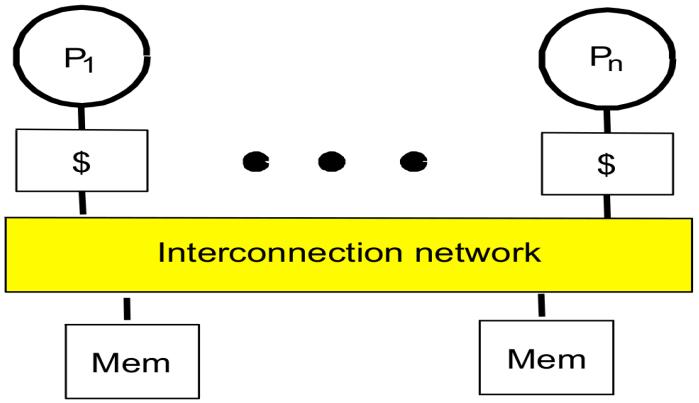
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c

Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

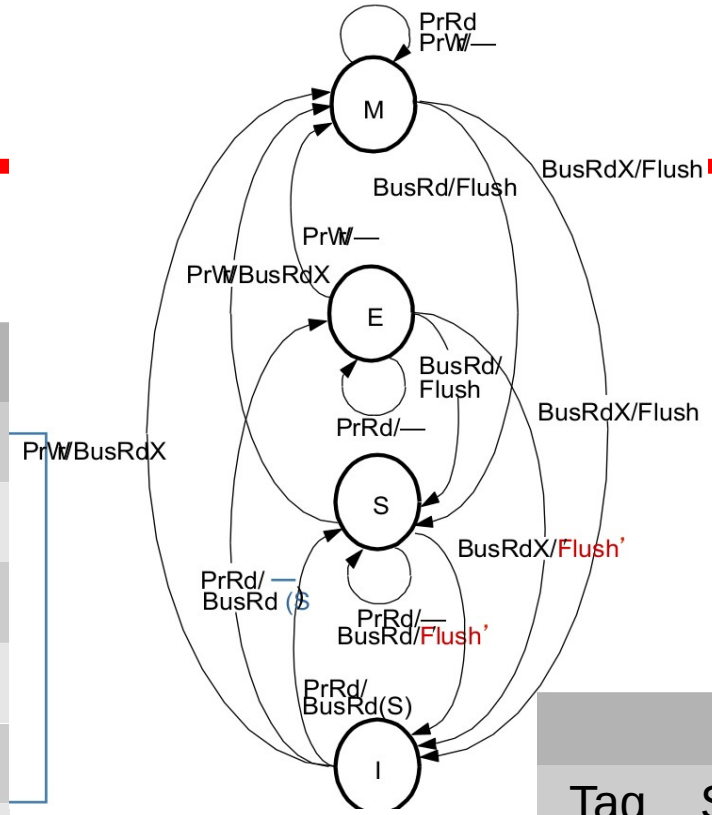
P0 cache				
Tag	Set	MESI	Data	

P1 cache				
Tag	Set	MESI	Data	



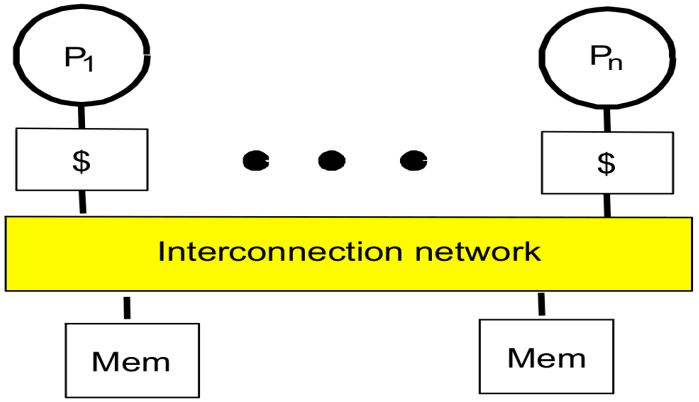
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

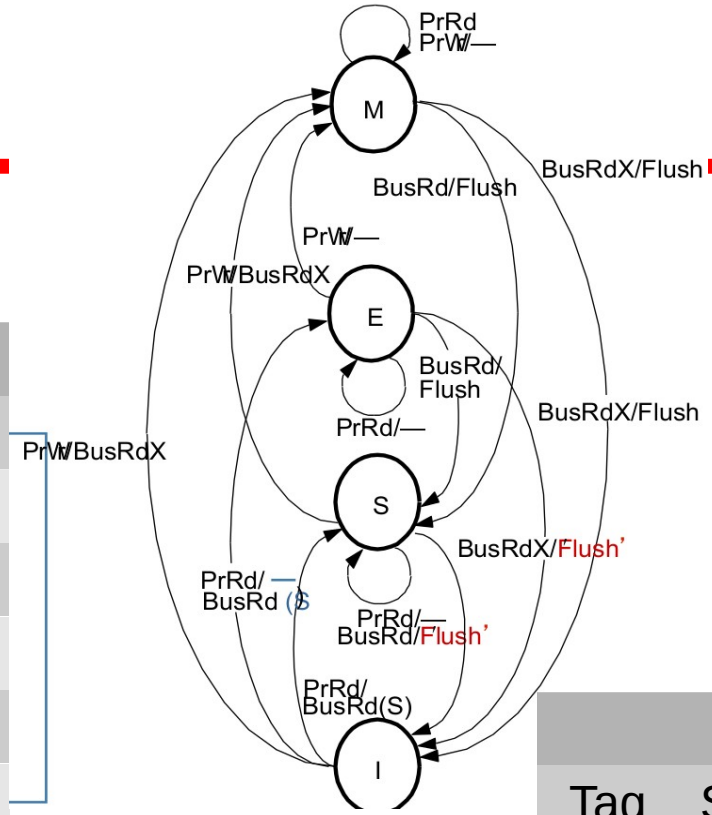
P0 cache				
Tag	Set	MESI	Data	



P1 cache				
Tag	Set	MESI	Data	

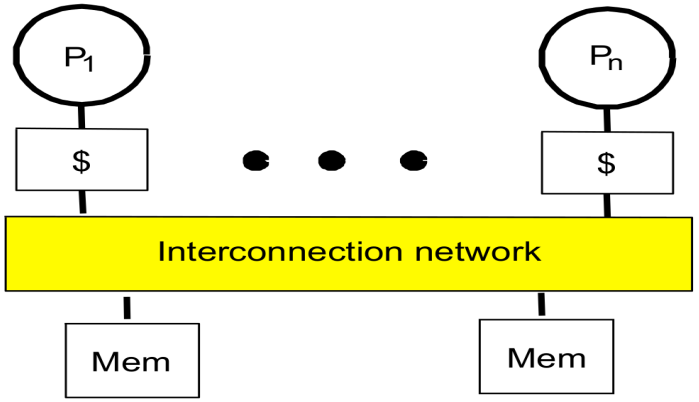
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

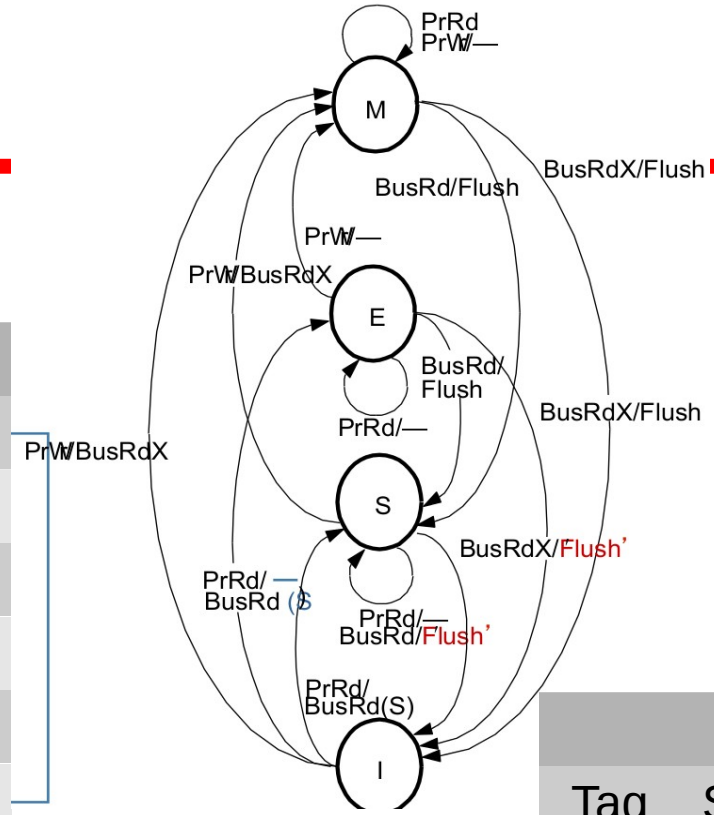
P0 cache				
Tag	Set	MESI	Data	



P1 cache				
Tag	Set	MESI	Data	

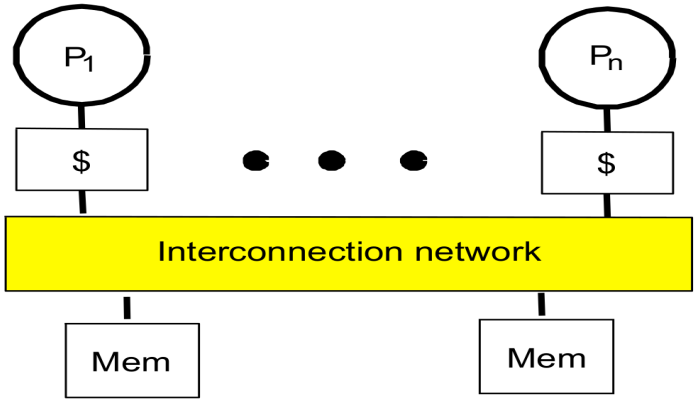
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

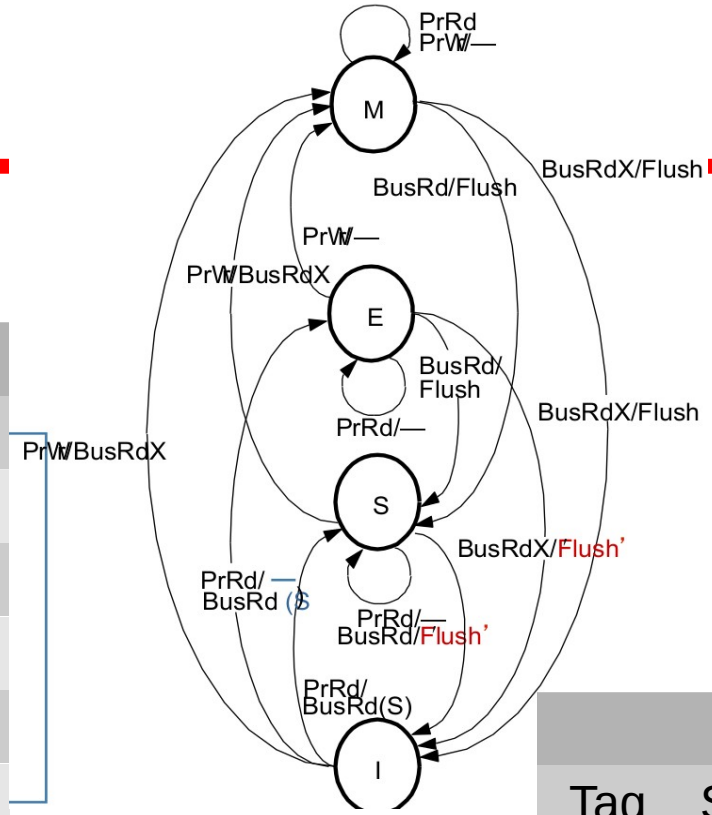
P0 cache						
Tag	Set	MESI	Data			
8	c3	M	0000	1111	0000	0000



P1 cache						
Tag	Set	MESI	Data			

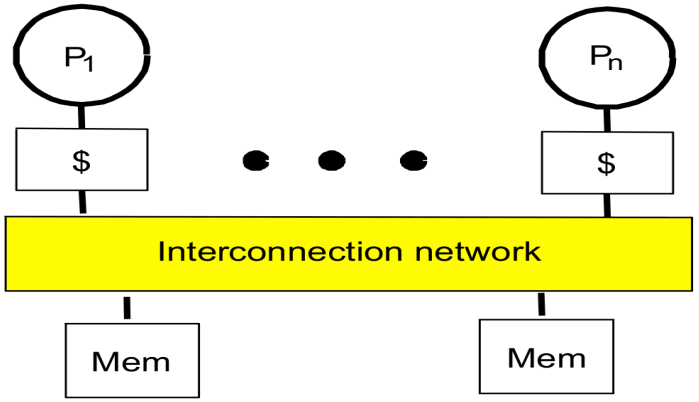
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

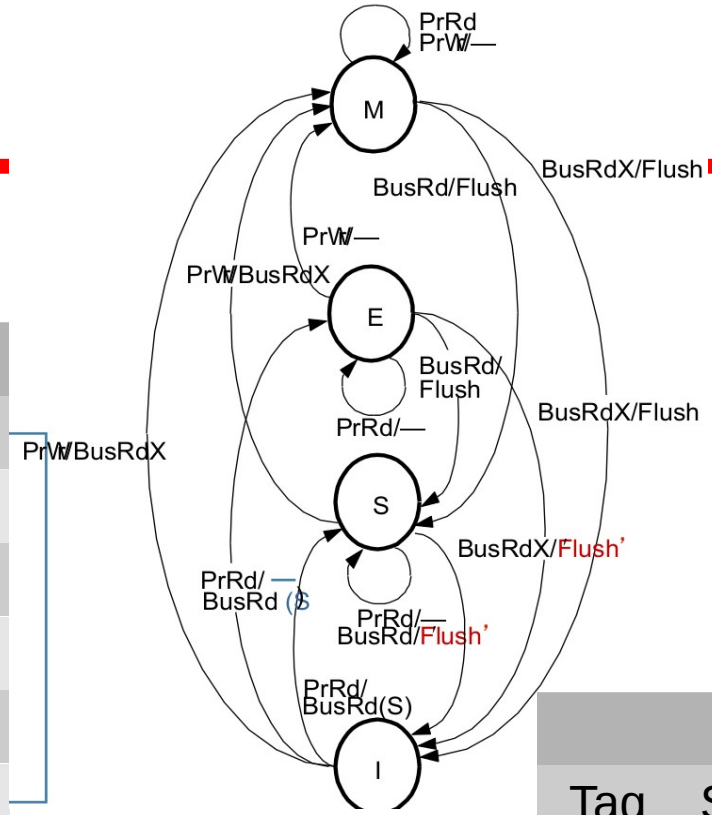
P0 cache						
Tag	Set	MESI	Data			
8	c3	M	0000	1111	0000	0000



P1 cache						
Tag	Set	MESI	Data			

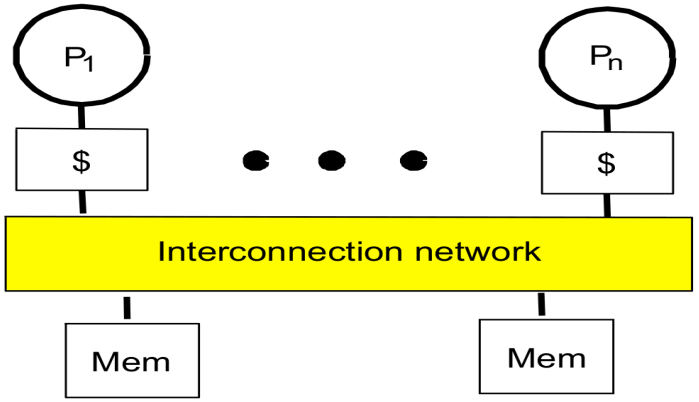
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

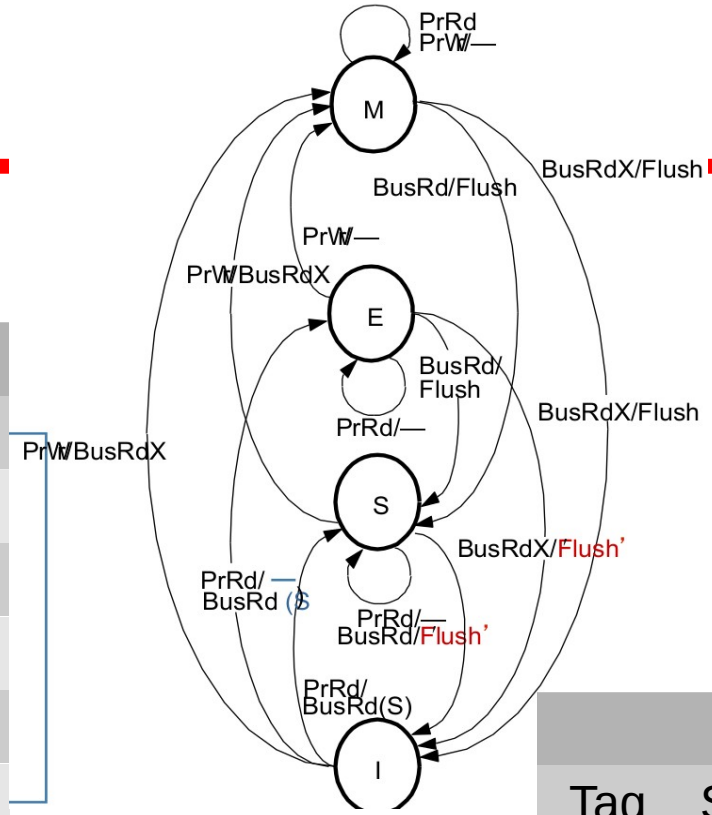
P0 cache						
Tag	Set	MESI	Data			
8	c3	M	0000	1111	0000	0000
7	c3	E	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			

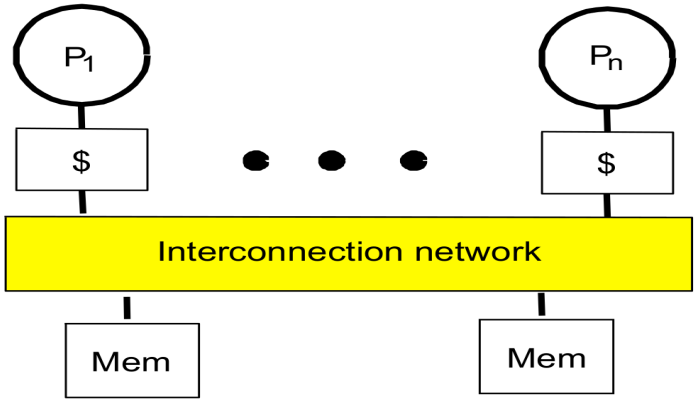
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	0000	0000	0000
9c3x	0000	0000	0000	0000

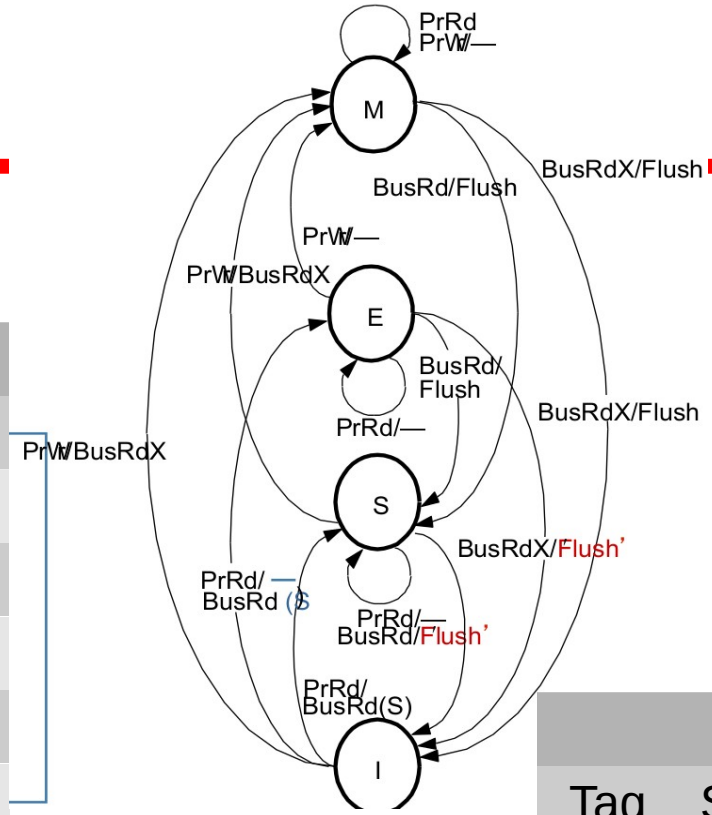
P0 cache						
Tag	Set	MESI	Data			
8	c3	M	0000	1111	0000	0000
7	c3	E	0000	0000	0000	0000



P1 cache					
Tag	Set	MESI	Data		

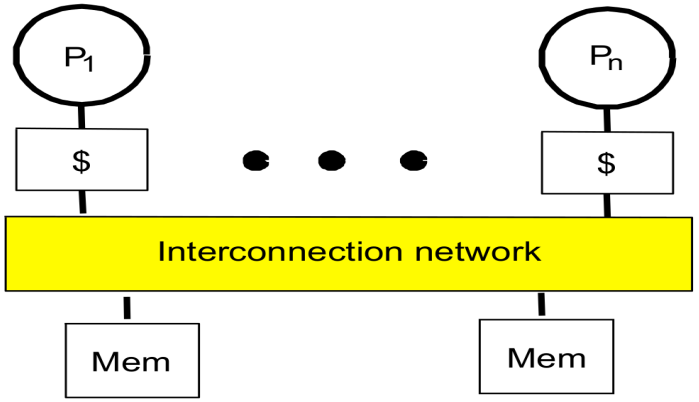
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	1111	0000	0000
9c3x	0000	0000	0000	0000

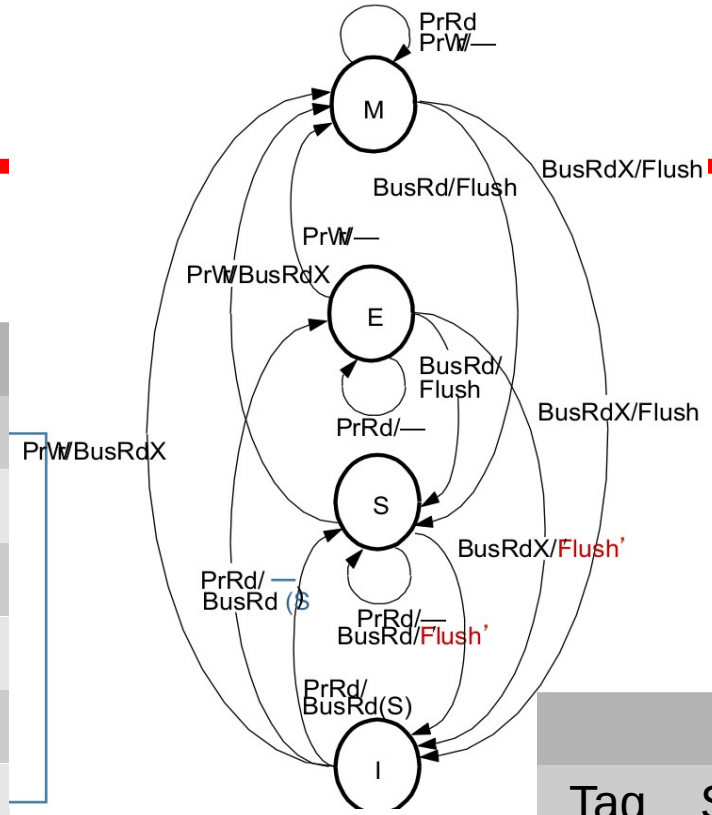
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	E	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	0000

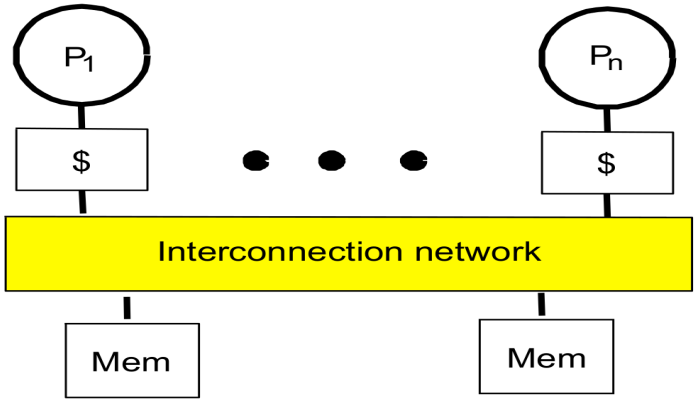
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	1111	0000	0000
9c3x	0000	0000	0000	0000

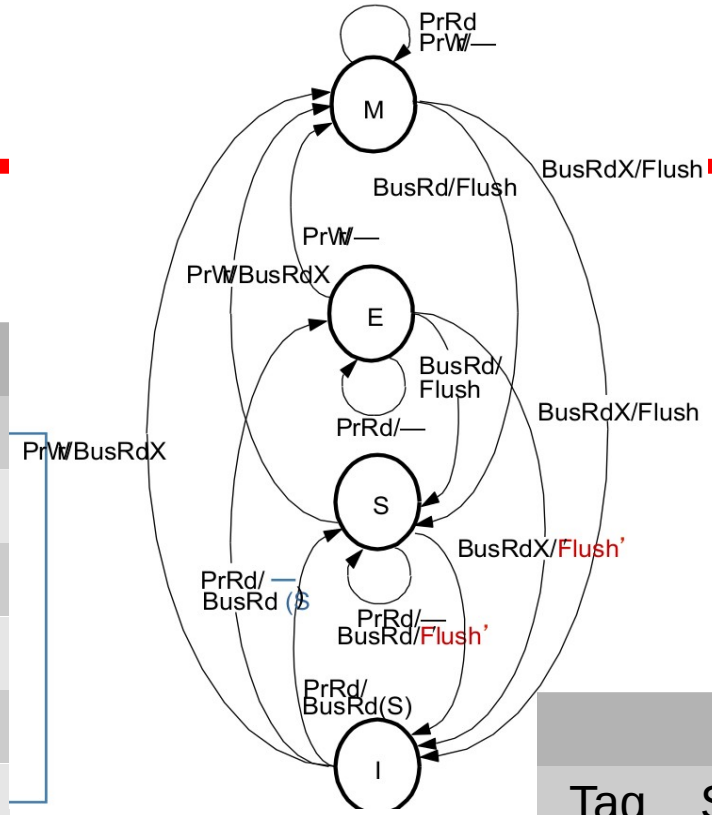
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	E	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	0000

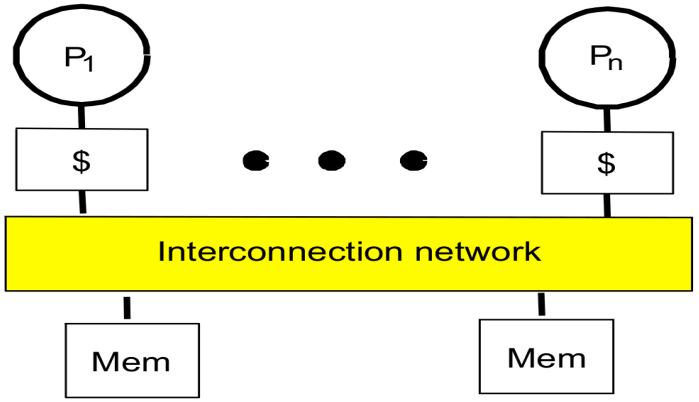
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	1111	0000	0000
9c3x	0000	0000	0000	0000

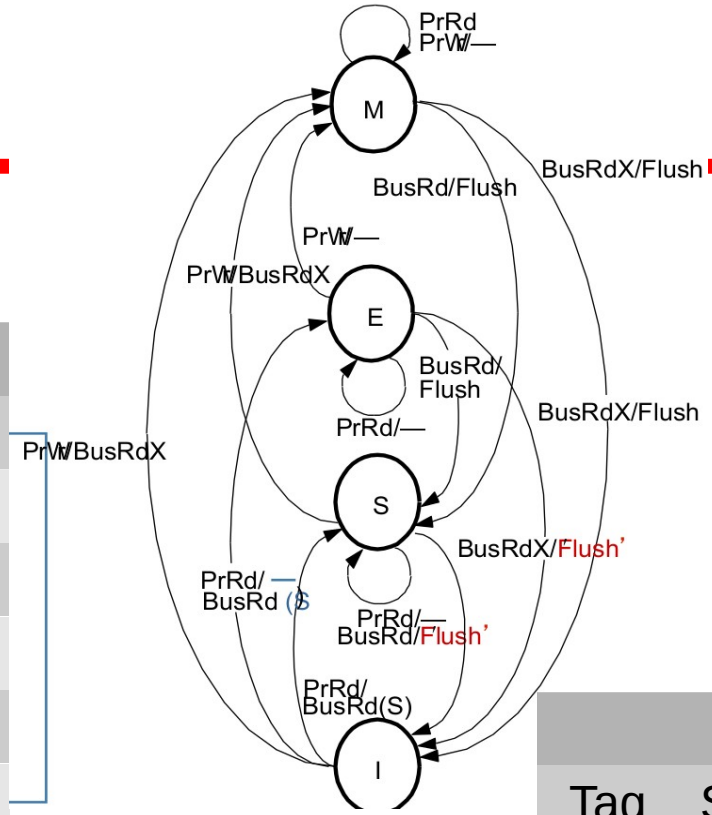
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	0000
7	c3	S	0000	0000	0000	0000

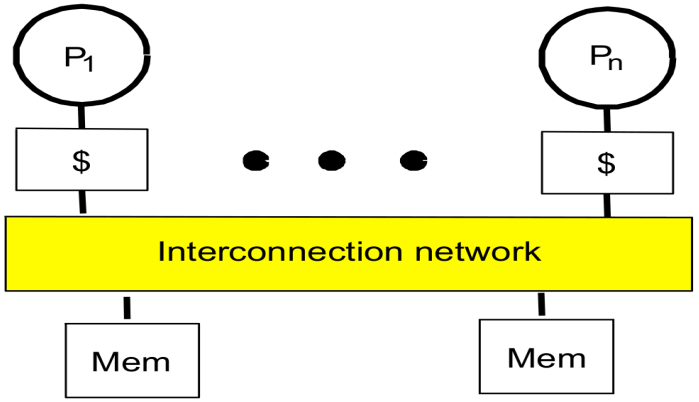
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	0000	1111	0000	0000
9c3x	0000	0000	0000	0000

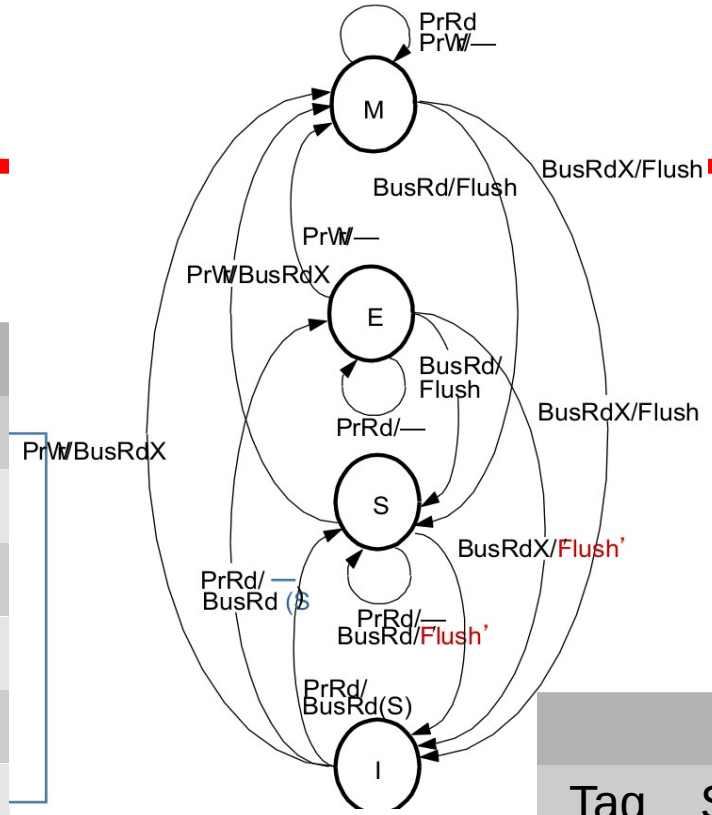
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	0000
7	c3	S	0000	0000	0000	0000

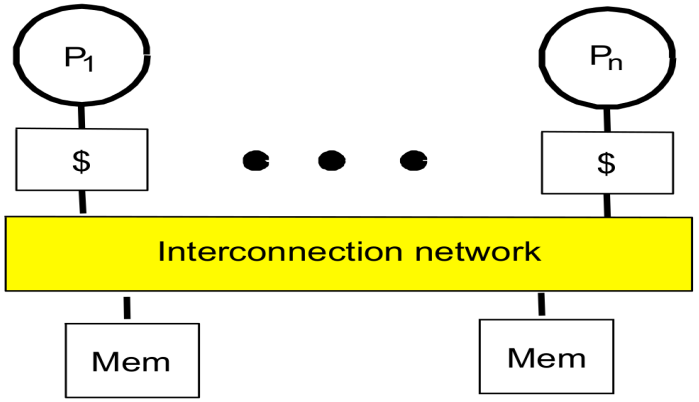
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

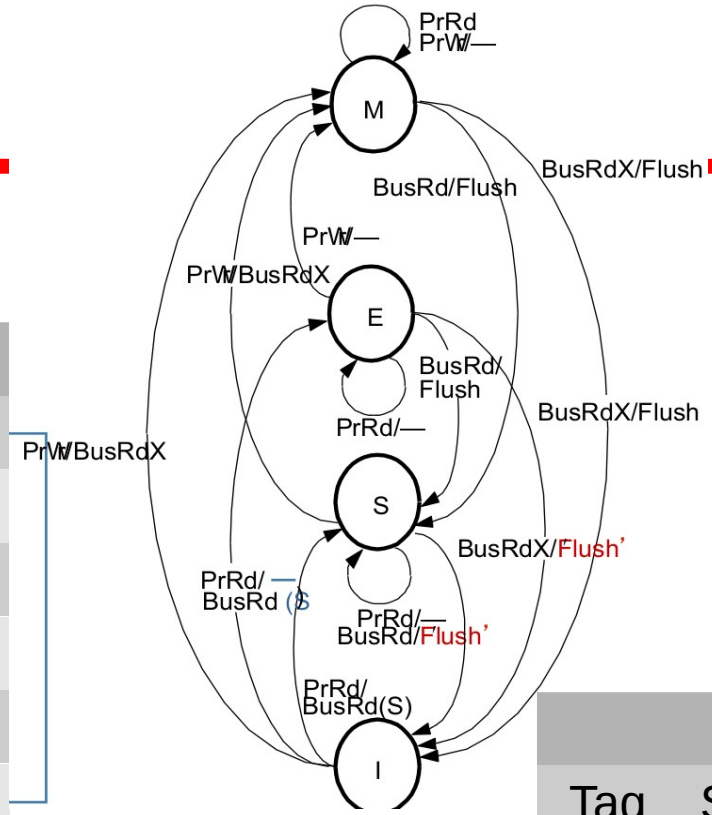
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

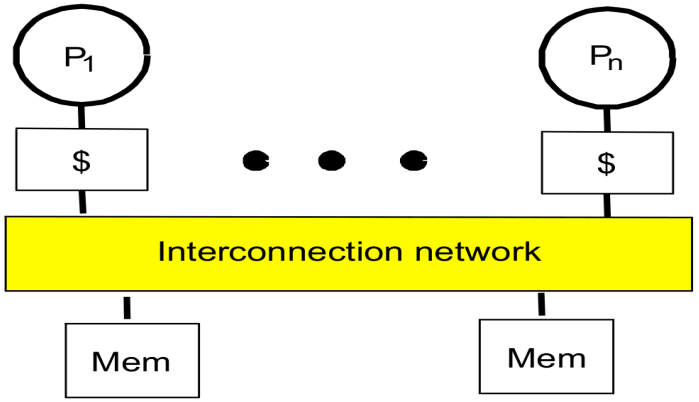
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

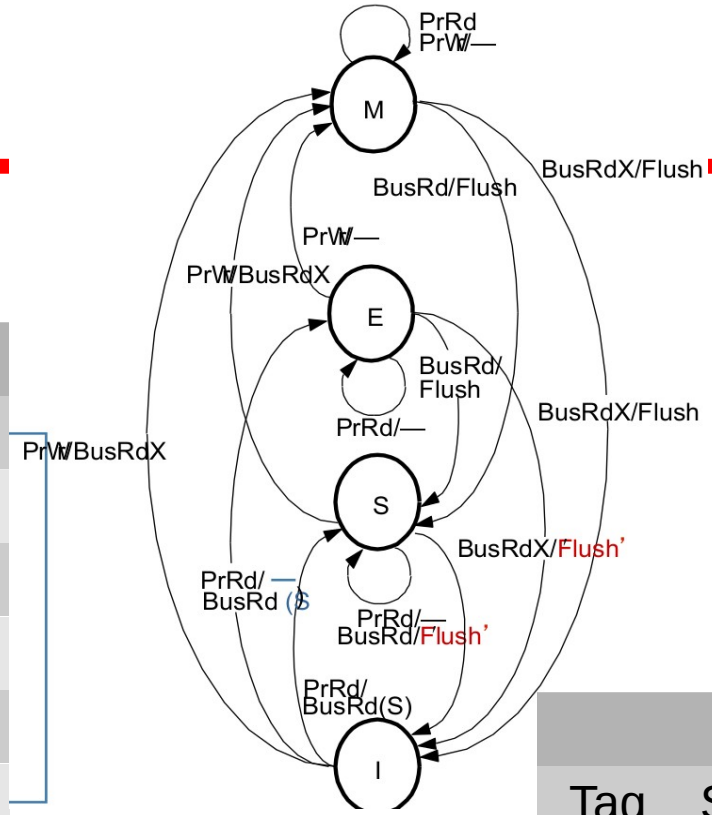
P0 cache						
Tag	Set	MESI	Data			
8	c3	I	--	--	--	--
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

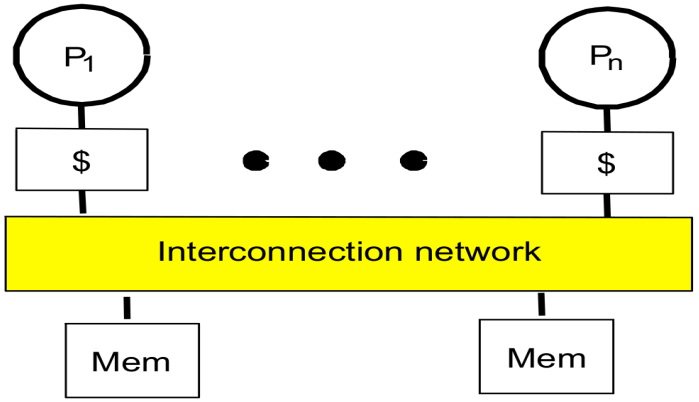
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

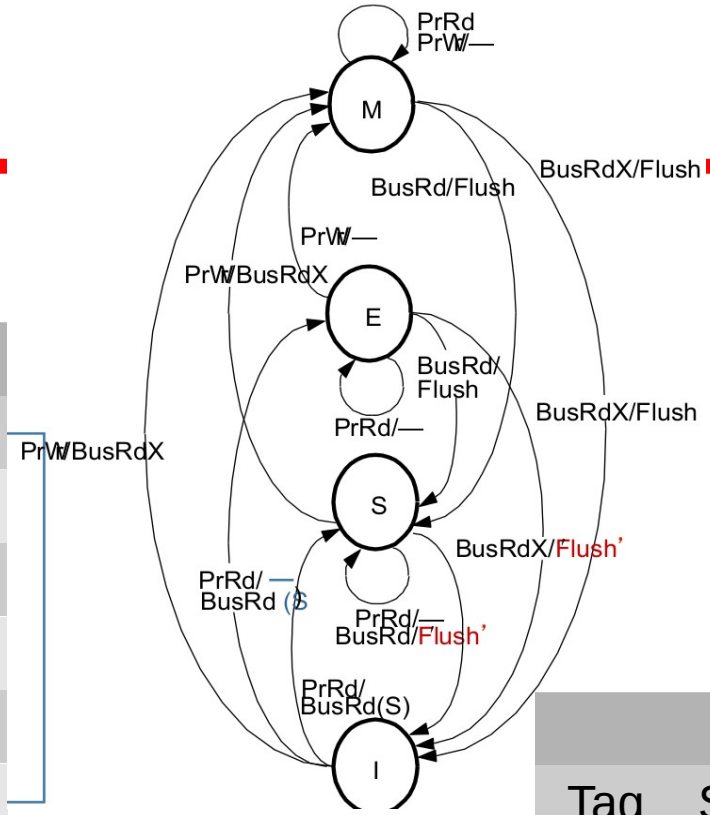
P0 cache						
Tag	Set	MESI	Data			
8	c3	E	2222	1111	0000	0000
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

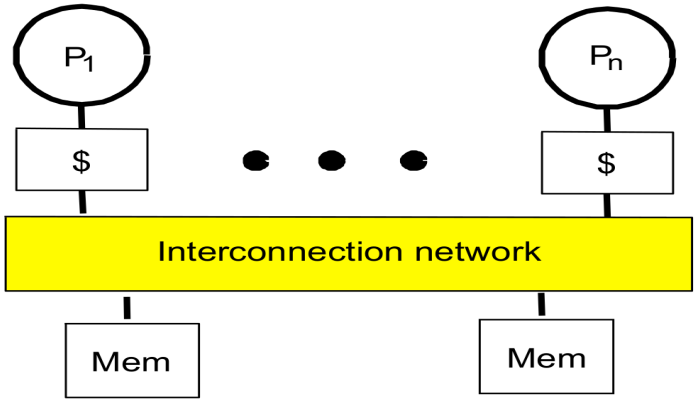
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

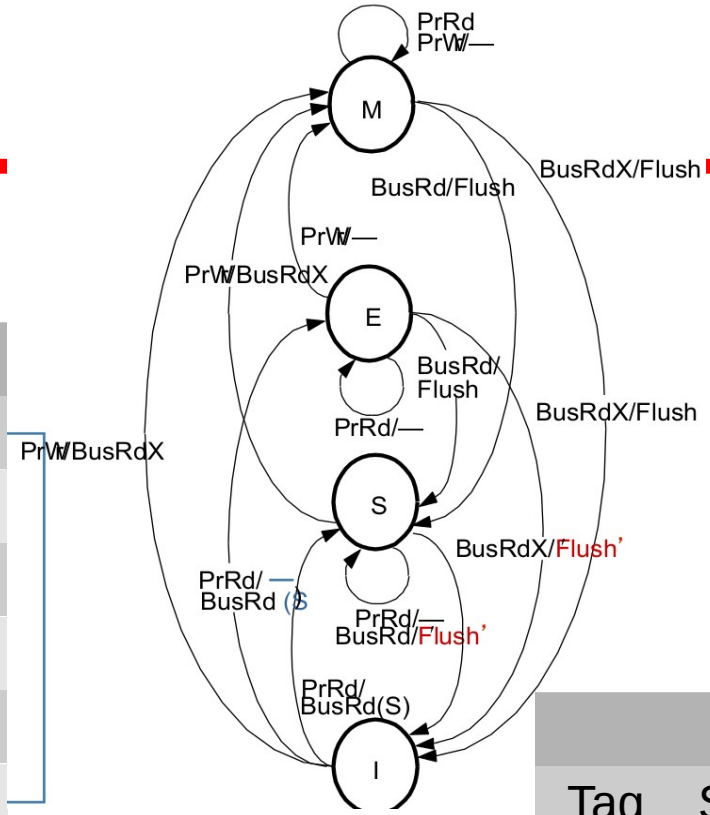
P0 cache						
Tag	Set	MESI	Data			
8	c3	E	2222	1111	0000	0000
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

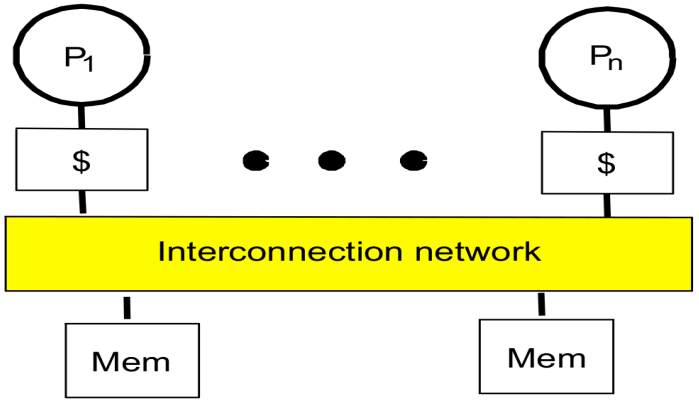
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



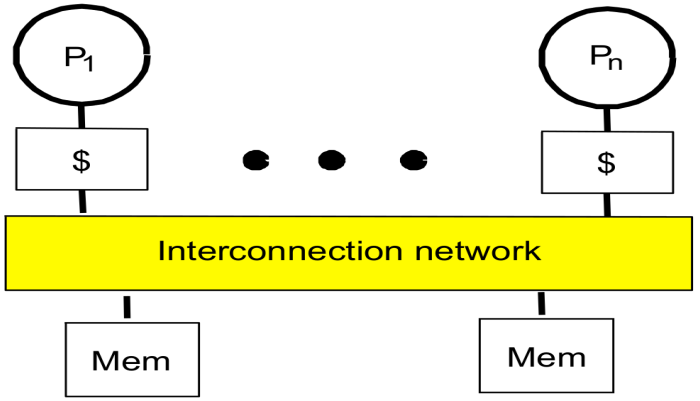
Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

P0 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	3333
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

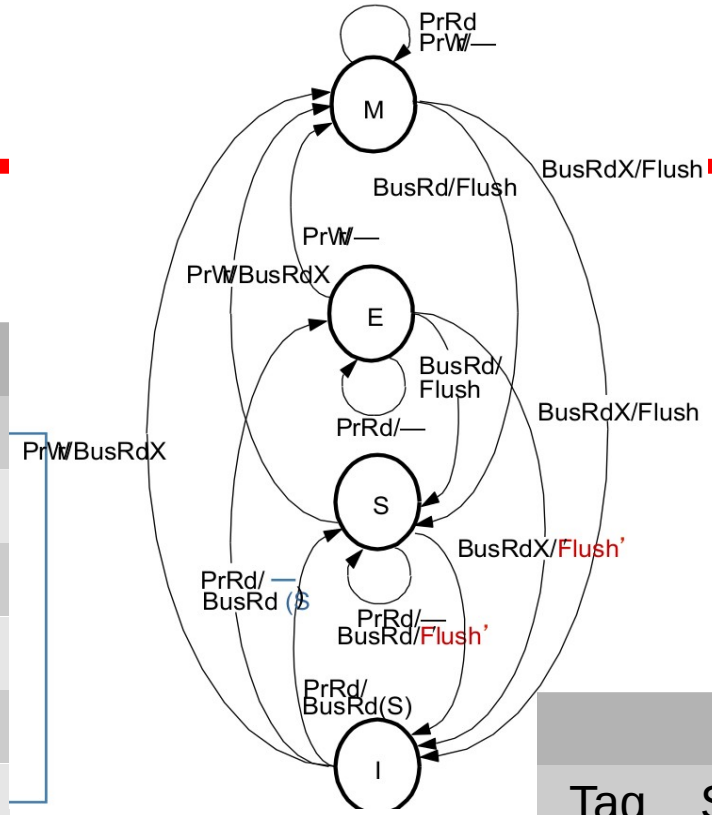
Coherence



• **Coherence** : Writes to a **single** location become visible to all in the same order

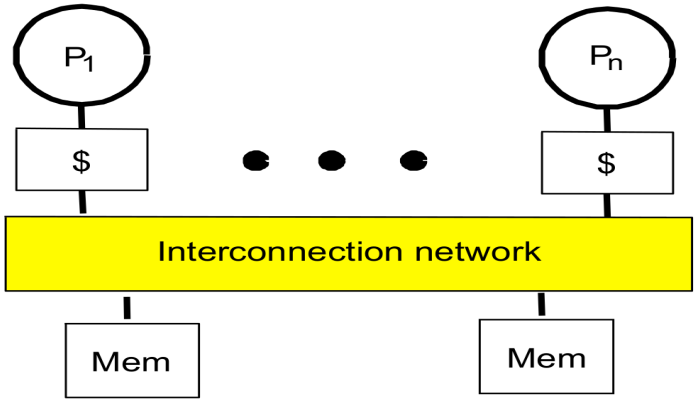
Example

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r 8c3c
P0 w 8c3c



Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

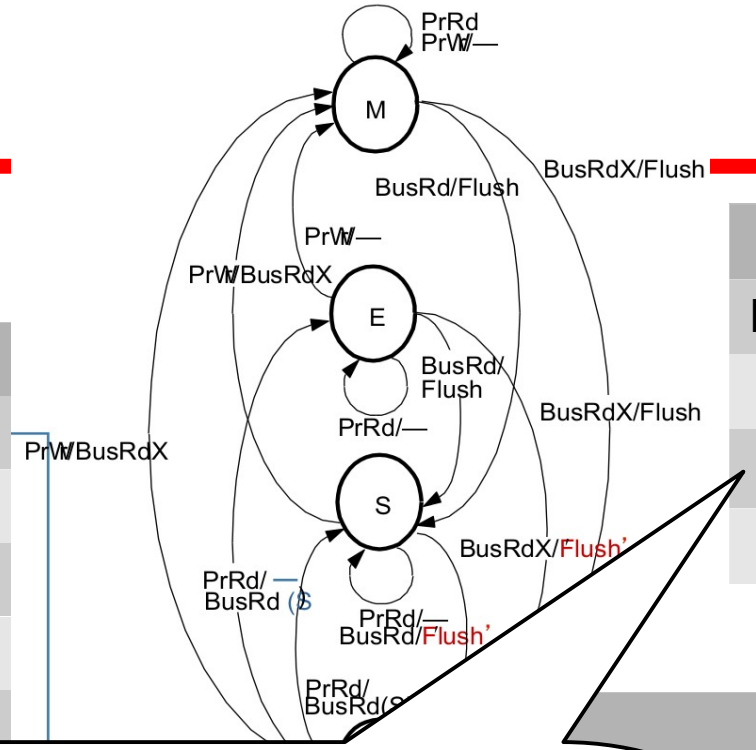
P0 cache						
Tag	Set	MESI	Data			
8	c3	M	2222	1111	0000	3333
7	c3	S	0000	0000	0000	0000



P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

Example

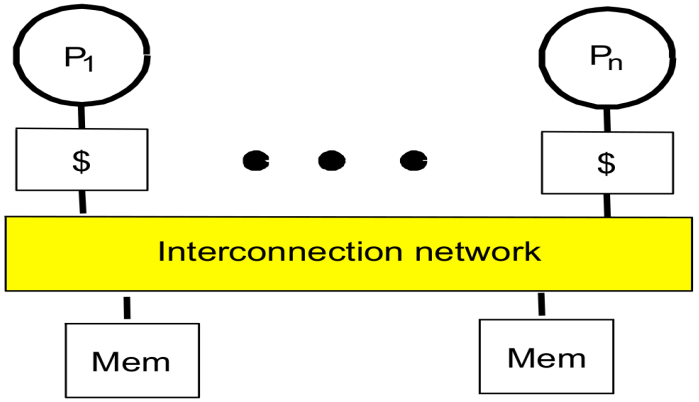
Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 r 9c38
P0 r
P0 w



A coherence protocol preserves orders among accesses to **same** location by different processes

Memory				
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000

P0 cache					
Tag	Set	MESI	Data		
	c3	M	2222	1111	0000	3333
	c3	S	0000	0000	0000	0000



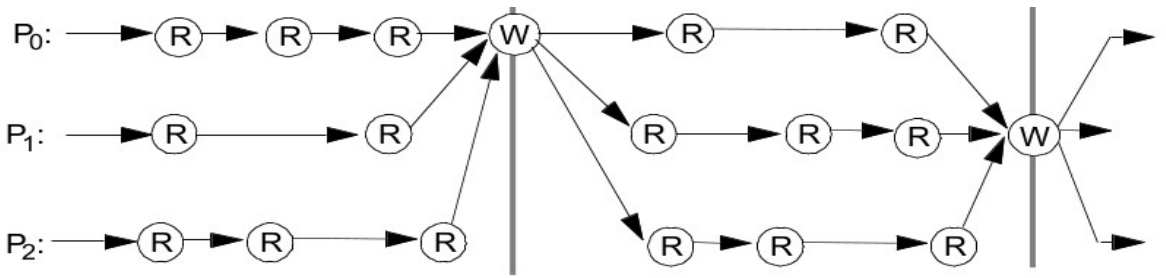
P1 cache						
Tag	Set	MESI	Data			
9	c3	E	0000	0000	0000	0000
7	c3	S	0000	0000	0000	0000

Example

Imagine an OoO execution case!!

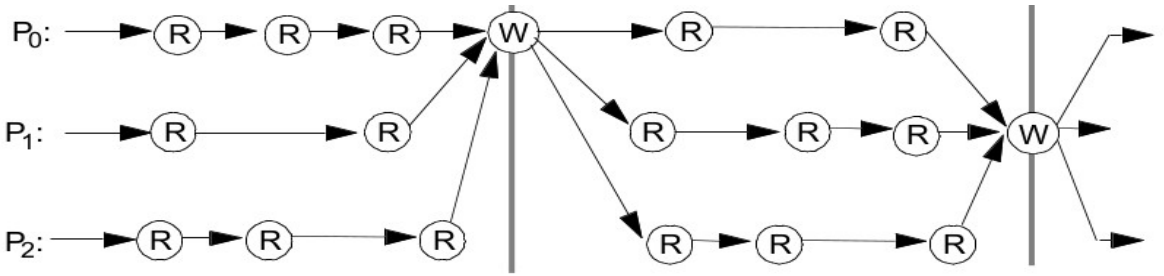
Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 w 9c38
P0 r 8c3c
P0 w 8c3c

	Memory			
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000



Mem: 8c3x

Mem: 9c3x

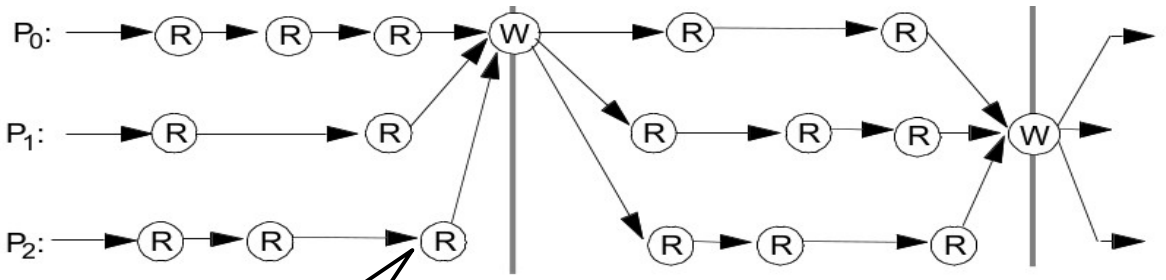


Example

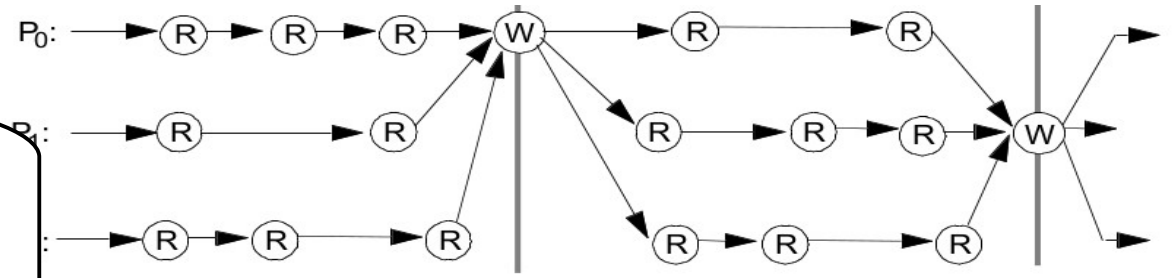
Imagine an OoO execution case!!

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 w 9c38
P0 r 8c3c
P0 w 8c3c

	Memory			
Location	0	4	8	c
7c3x	0000	0000	0000	0000
8c3x	2222	1111	0000	0000
9c3x	0000	0000	0000	0000



Mem: 8c3x



We will need to respect a "global" order...

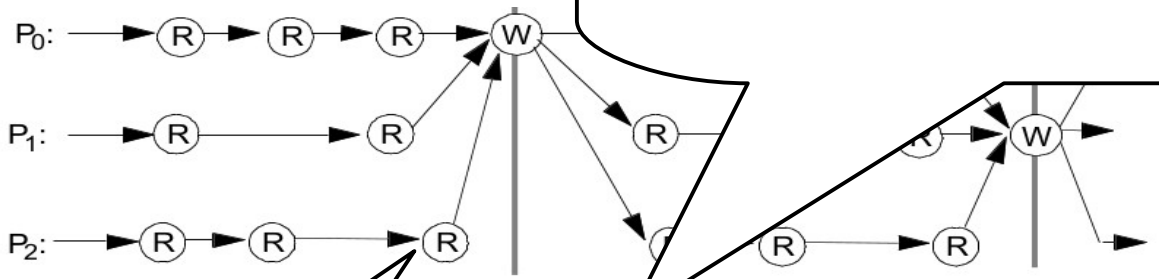
Example

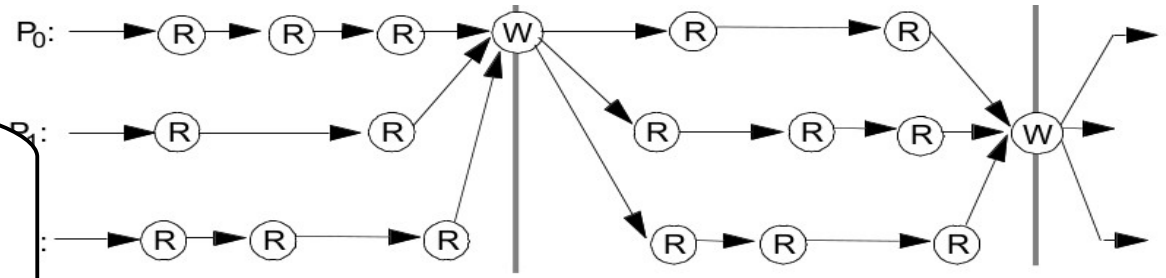
Imagine an OoO execution case!!

Instructions
P0 w 8c34
P0 r 7c34
P1 w 8c30
P1 r 7c34
P1 w 9c38
P0 r 8c3c
P0 w 8c3c

Memory				
Location	0	4	8	c

We will need an order in which memory operations (from any process) can appear to execute with **respect to one another**





We will need to respect a "global" order...

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
(1a) A = 1;	(2a) print B;
(1b) B = 2;	(2b) print A;

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0):
-
-
-

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a
-
-
-

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b
-
-
-

Memory Consistency

- Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
-
-
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0):
-
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a
-
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b
-
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
-
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2):
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
-

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

  To get A=0:

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):
  To get A=0: 2b → 1a

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

  To get A=0: 2b → 1a, but from P1 order 2a → 2b → 1a → 1b

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

  To get A=0: 2b → 1a, but from P1 order 2a → 2b → 1a → 1b

  To get B=2:

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :

- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

  To get A=0: 2b → 1a, but from P1 order 2a → 2b → 1a → 1b

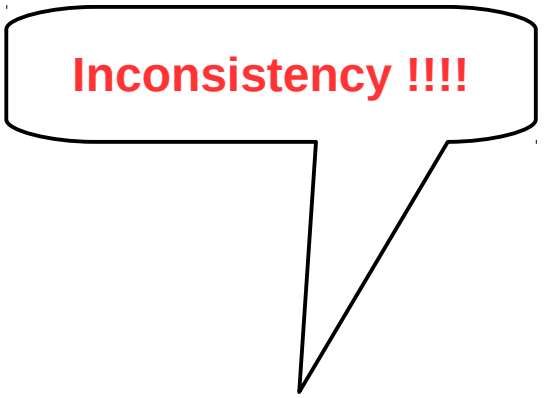
  To get B=2: 1b → 2a

Memory Consistency

• Let's check a simple example where 2 memory locations are accessed.

P_1	P_2
/*Assume initial values of A and B are 0*/	
`(1a) A = 1;`	`(2a) print B;`
`(1b) B = 2;`	`(2b) print A;`

We can have 4 outcomes for (A,B) :



- (0,0): 2a → 2b → 1a → 1b
- (1,0): 1a → 2a → 2b → 1b
- (1,2): 1a → 1b → 2a → 2b
- (0,2):

  To get A=0: 2b → 1a, but from P1 order 2a → 2b → 1a → 1b

  To get B=2: 1b → 2a, but from P2 order 1a → 1b → 2a → 2b

Lab 2 – Any questions??