CMSC 611: Advanced Computer Architecture

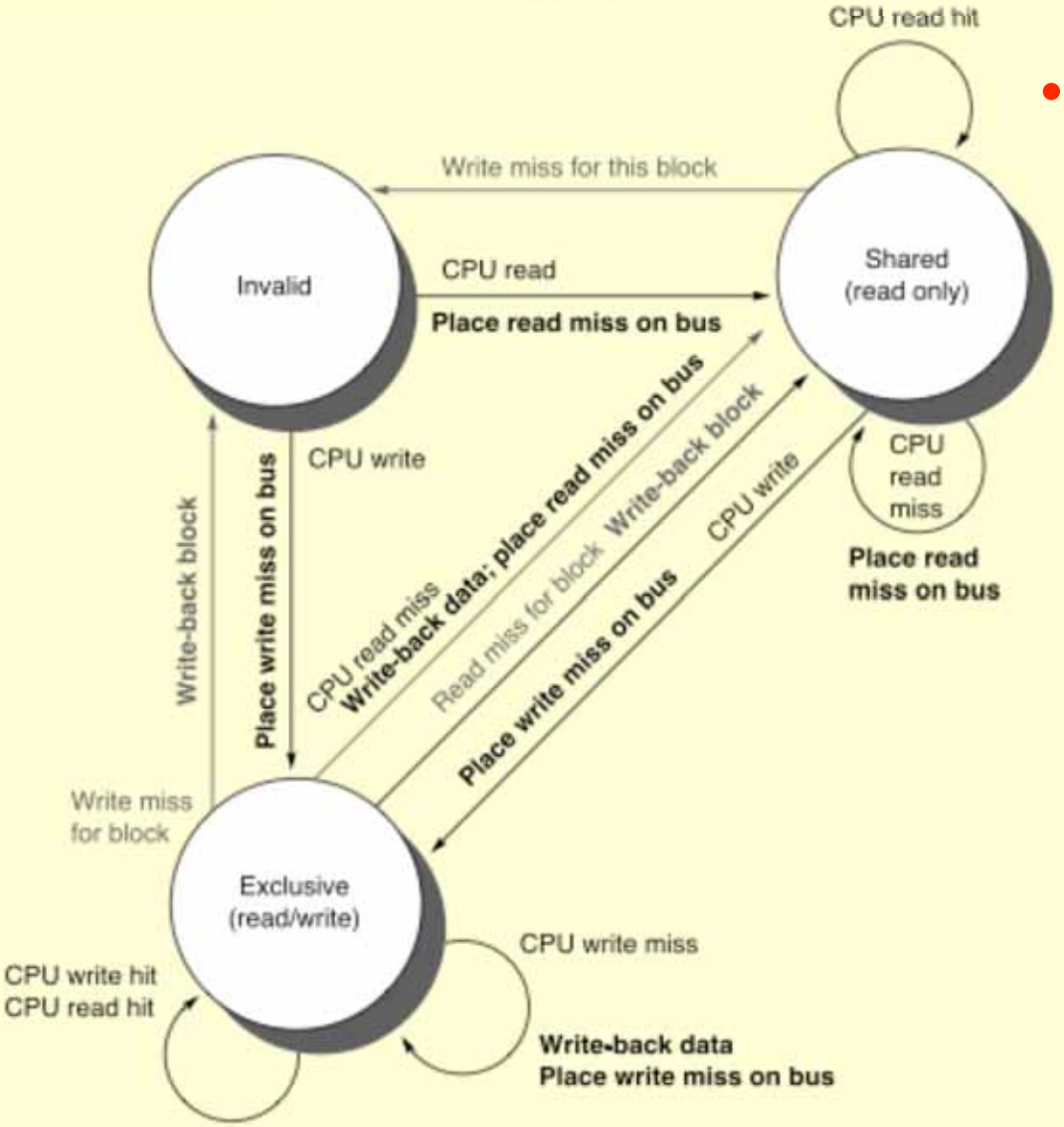
Shared Memory

Most slides adapted from David Patterson. Some from Mohomed Younis

An Example Snoopy Protocol

- Invalidation protocol, write-back cache
- Each block of memory is in one state:
  - Clean in all caches and up-to-date in memory (Shared)
  - OR Dirty in exactly one cache (Exclusive)
  - OR Not in any caches
- Each cache block is in one state (track these):
  - Shared : block can be read
  - OR Exclusive : cache has only copy, it is write-able, and dirty
  - OR Invalid : block contains no data
- Read misses: cause all caches to snoop bus
- Writes to clean line are treated as misses

Snoopy-Cache Controller

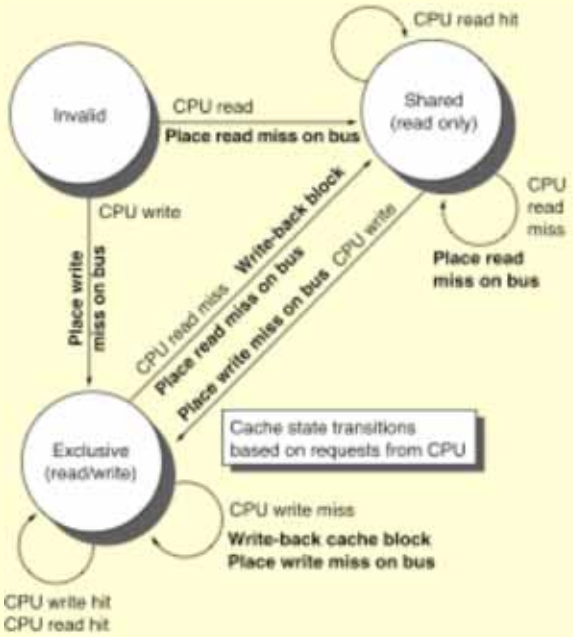


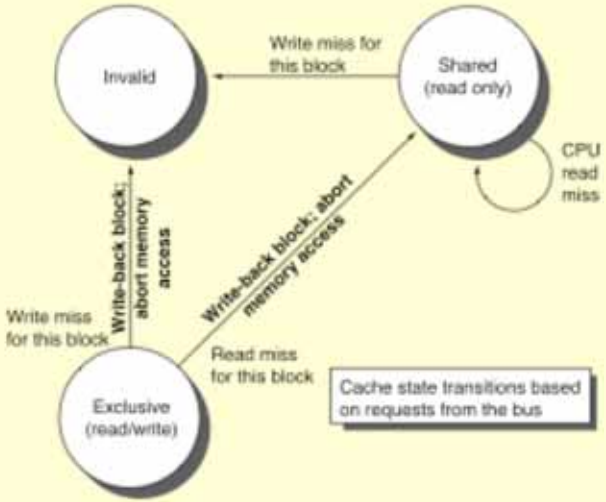
- Complications
  - Cannot update cache until bus is obtained
  - Two step process:
    - Arbitrate for bus
    - Place miss on bus and complete operation
  - Split transaction bus:
    - Bus transaction is not atomic
    - Multiple misses can interleave, allowing two caches to grab block in the Exclusive state
    - Must track and prevent multiple misses for one block

Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1												
P1: Read A1												
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assumes memory blocks A1 and A2 map to same cache block, initial cache state is invalid

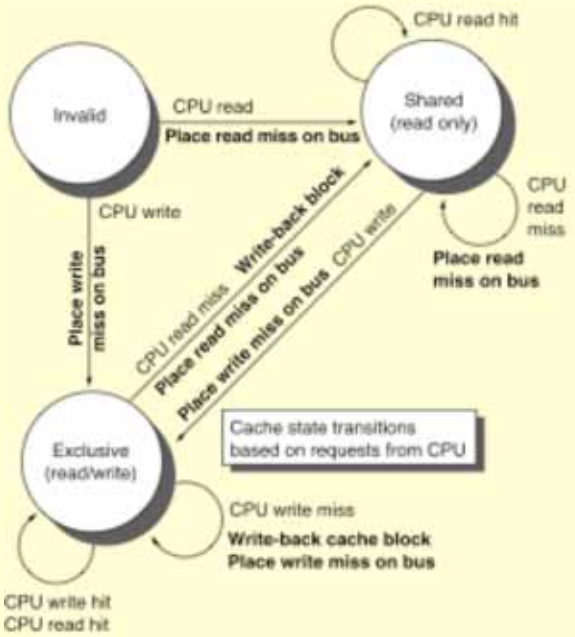


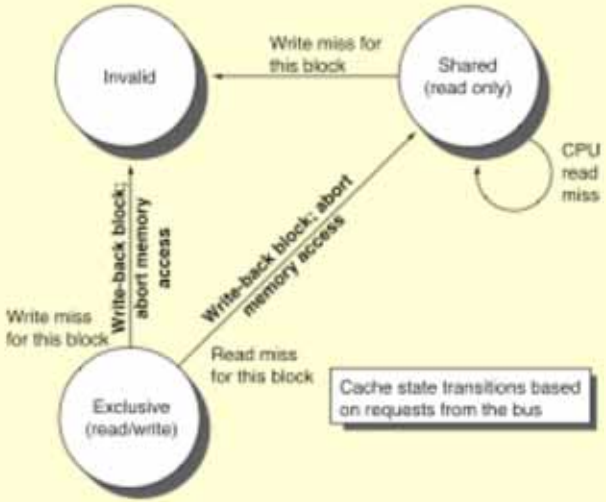


Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	Excl.	A1	10				WrMs	P1	A1			
P1: Read A1												
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assumes memory blocks A1 and A2 map to same cache block

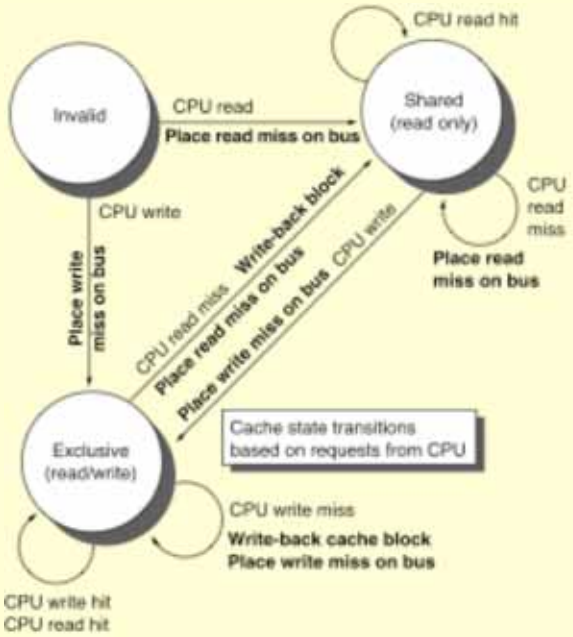


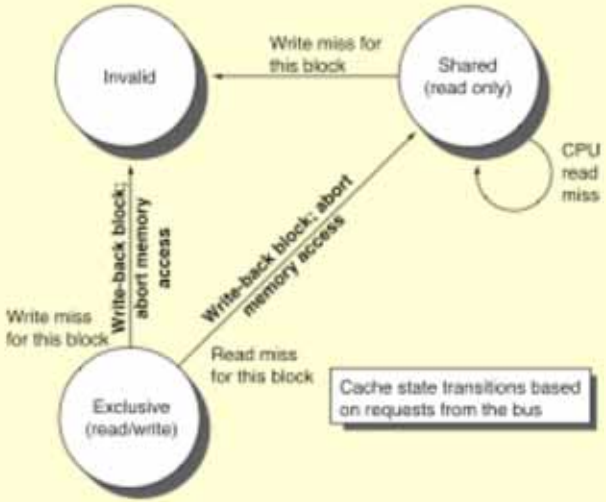


Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	Excl.	A1	10				WrMs	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1												
P2: Write 20 to A1												
P2: Write 40 to A2												

Assumes memory blocks A1 and A2 map to same cache block

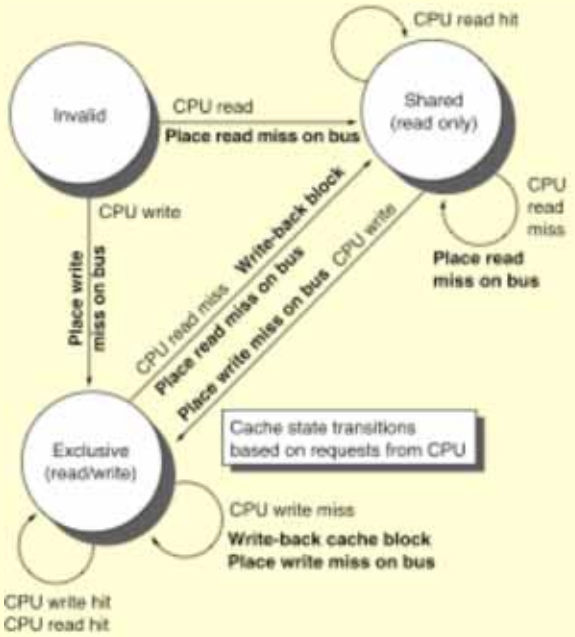


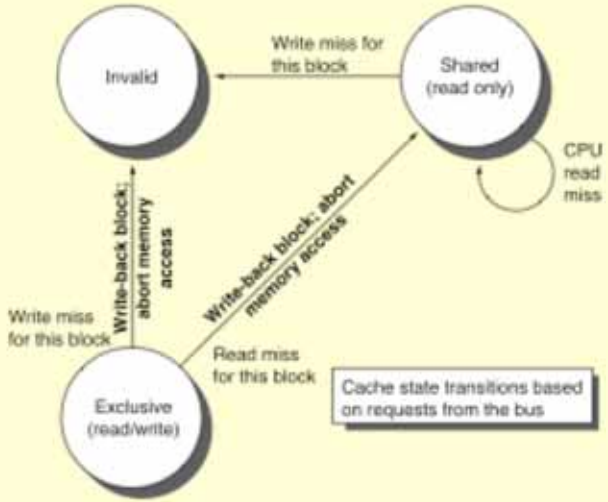


Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	Excl.	A1	10				WrMs	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				Shar.	A1		RdMs	P2	A1			
	Shar.	A1	10				WrBk	P1	A1	10		10
				Shar.	A1	10	RdDa	P2	A1	10		10
P2: Write 20 to A1												10
P2: Write 40 to A2												10
												10

Assumes memory blocks A1 and A2 map to same cache block

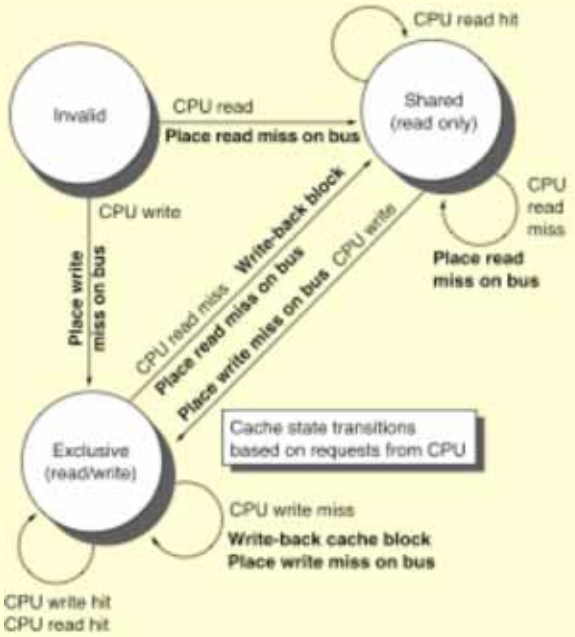


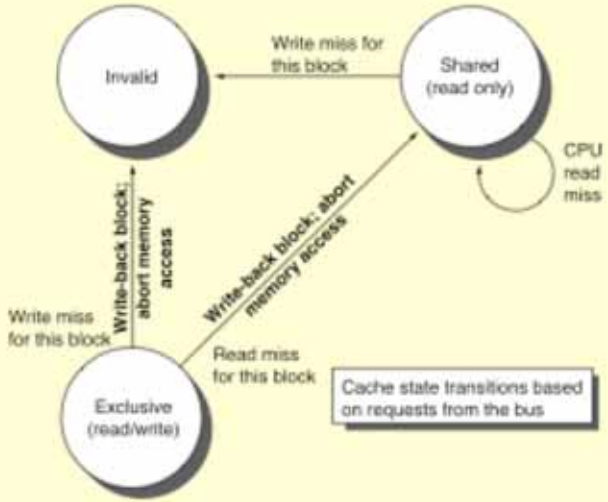


Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	Excl.	A1	10				WrMs	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				Shar.	A1		RdMs	P2	A1			
	Shar.	A1	10				WrBk	P1	A1	10		10
				Shar.	A1	10	RdDa	P2	A1	10		10
P2: Write 20 to A1	Inv.			Excl.	A1	20	WrMs	P2	A1			10
P2: Write 40 to A2												10
												10

Assumes memory blocks A1 and A2 map to same cache block

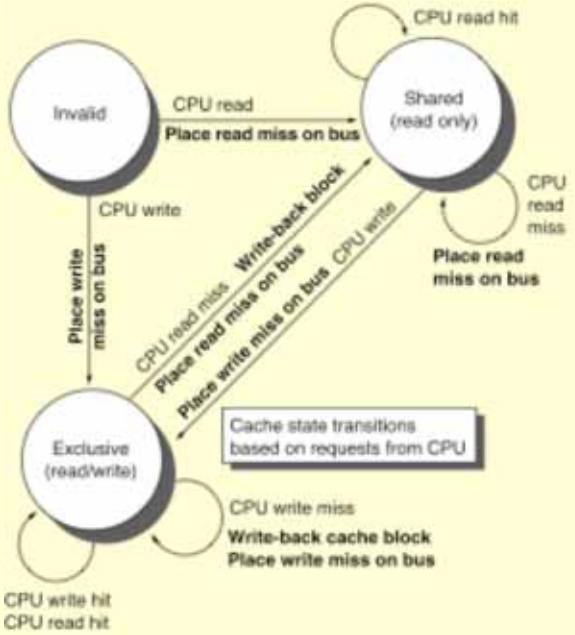


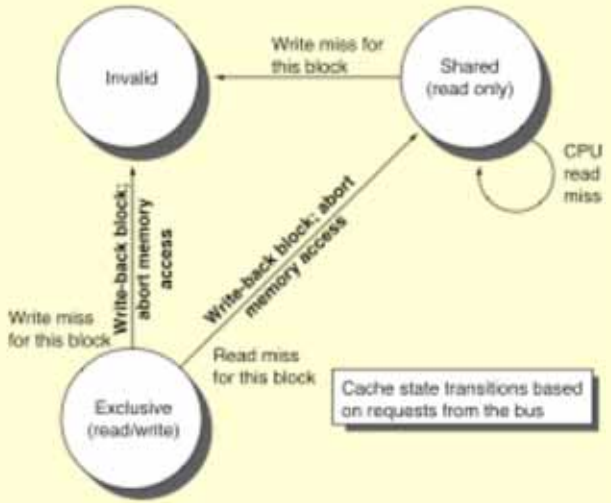


Example

	P1			P2			Bus				Memory	
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	Value
P1: Write 10 to A1	Excl.	A1	10				WrMs	P1	A1			
P1: Read A1	Excl.	A1	10									
P2: Read A1				Shar.	A1		RdMs	P2	A1			
	Shar.	A1	10				WrBk	P1	A1	10		10
				Shar.	A1	10	RdDa	P2	A1	10	*A1*	10
P2: Write 20 to A1	Inv.			Excl.	A1	20	WrMs	P2	A1			10
P2: Write 40 to A2							WrMs	P2	A2			10
				Excl.	A2	40	WrBk	P2	A1	20	*A1*	20

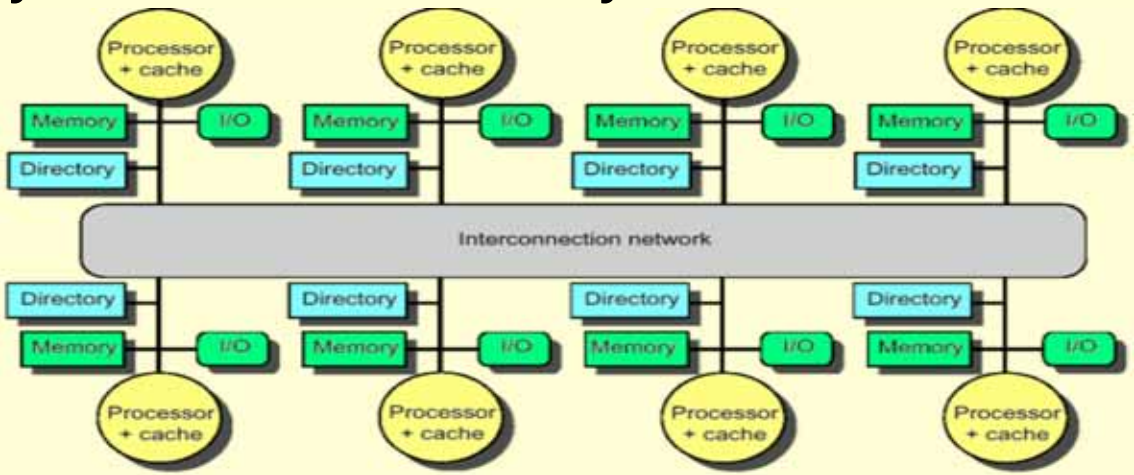
Assumes memory blocks A1 and A2 map to same cache block





Distributed Directory Multiprocessors

- Directory per cache that tracks state of every block in every cache
  - Which caches have a block, dirty vs. clean, ...
  - Info per memory block vs. per cache block?
    - \+ In memory => simpler protocol (centralized/one location)
    - – In memory => directory is f(memory size) vs. f(cache size)
- To prevent directory from being a bottleneck
  - distribute directory entries with memory
  - each tracks of which processor has their blocks



Directory Protocol

- Similar to Snoopy Protocol: Three states
  - Shared: Multiple processors have the block cached and the contents of the block in memory (as well as all caches) is up-to-date
  - Uncached No processor has a copy of the block (not valid in any cache)
  - Exclusive: Only one processor (owner) has the block cached and the contents of the block in memory is out-to-date (the block is dirty)
- In addition to cache state, must track which processors have data when in the shared state
  - usually bit vector, 1 if processor has copy

Directory Protocol

- Keep it simple(r):
  - Writes to non-exclusive data => write miss
  - Processor blocks until access completes
  - Assume messages received and acted upon in order sent
- Terms: typically 3 processors involved
  - Local node where a request originates
  - Home node where the memory location of an address resides
  - Remote node has a copy of a cache block, whether exclusive or shared
- No bus and do not want to broadcast:
  - interconnect no longer single arbitration point
  - all messages have explicit responses

Example Directory Protocol

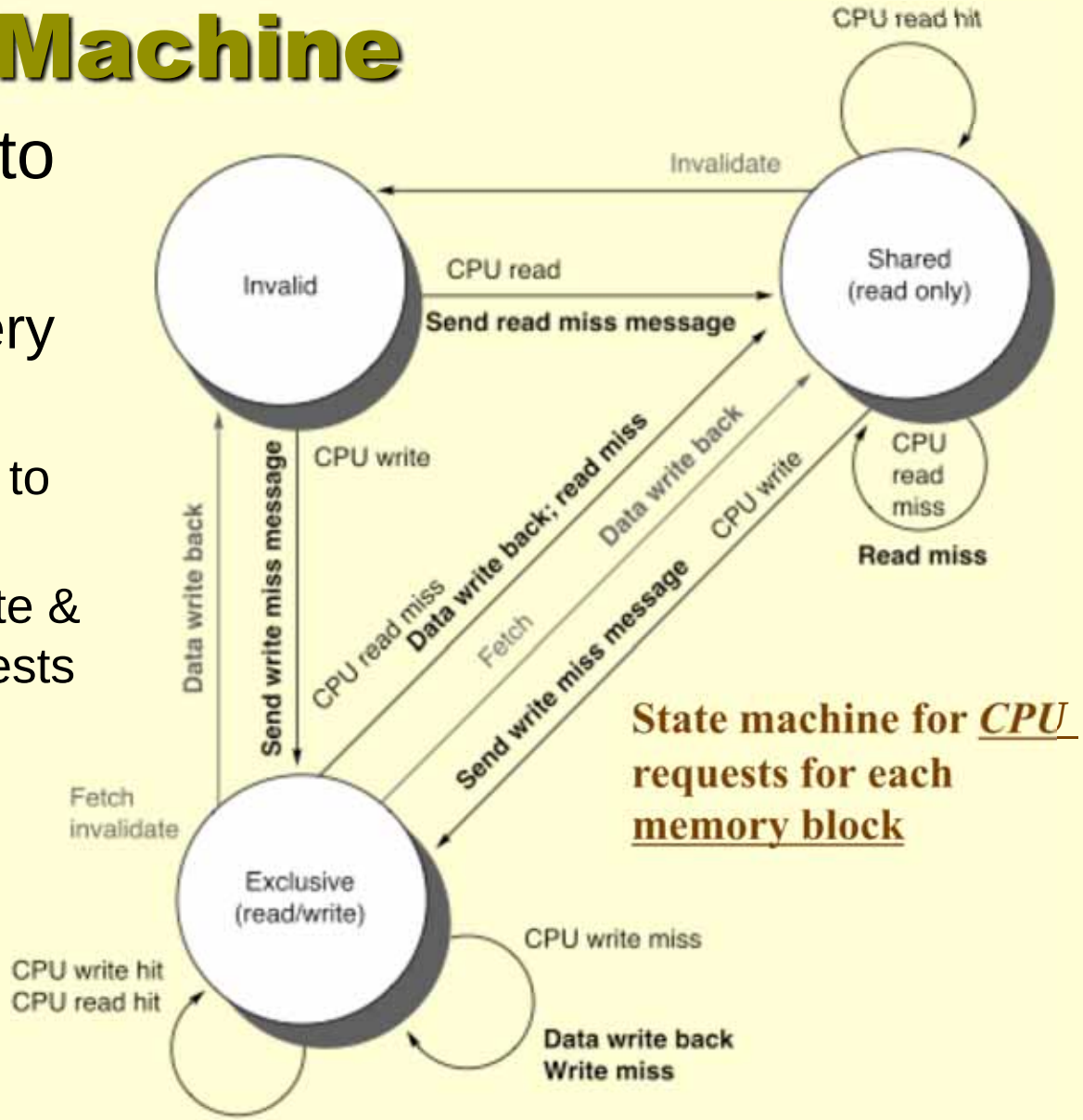
- Message sent to directory causes two actions:
  - Update the directory
  - More messages to satisfy request
- We assume operations atomic, but they are not; reality is much harder; must avoid deadlock when run out of buffers in network

Directory Protocol Messages

Type	SRC	DEST	MSG
Read miss	local cache	home directory	P,A
P has read miss at A; request data and make P a read sharer			
Write miss	local cache	home directory	P,A
P has write miss at A; request data and make P exclusive owner			
Invalidate	home directory	remote cache	A
Invalidate shared data at A			
Fetch	home directory	remote cache	A
Fetch block A home; change A remote state to shared			
Fetch/invalidate	home directory	remote cache	A
Fetch block A home; invalidate remote copy			
Data value reply	home directory	local cache	D
Return data value from home memory			
Data write back	remote cache	home directory	A,D
Write back data value for A			

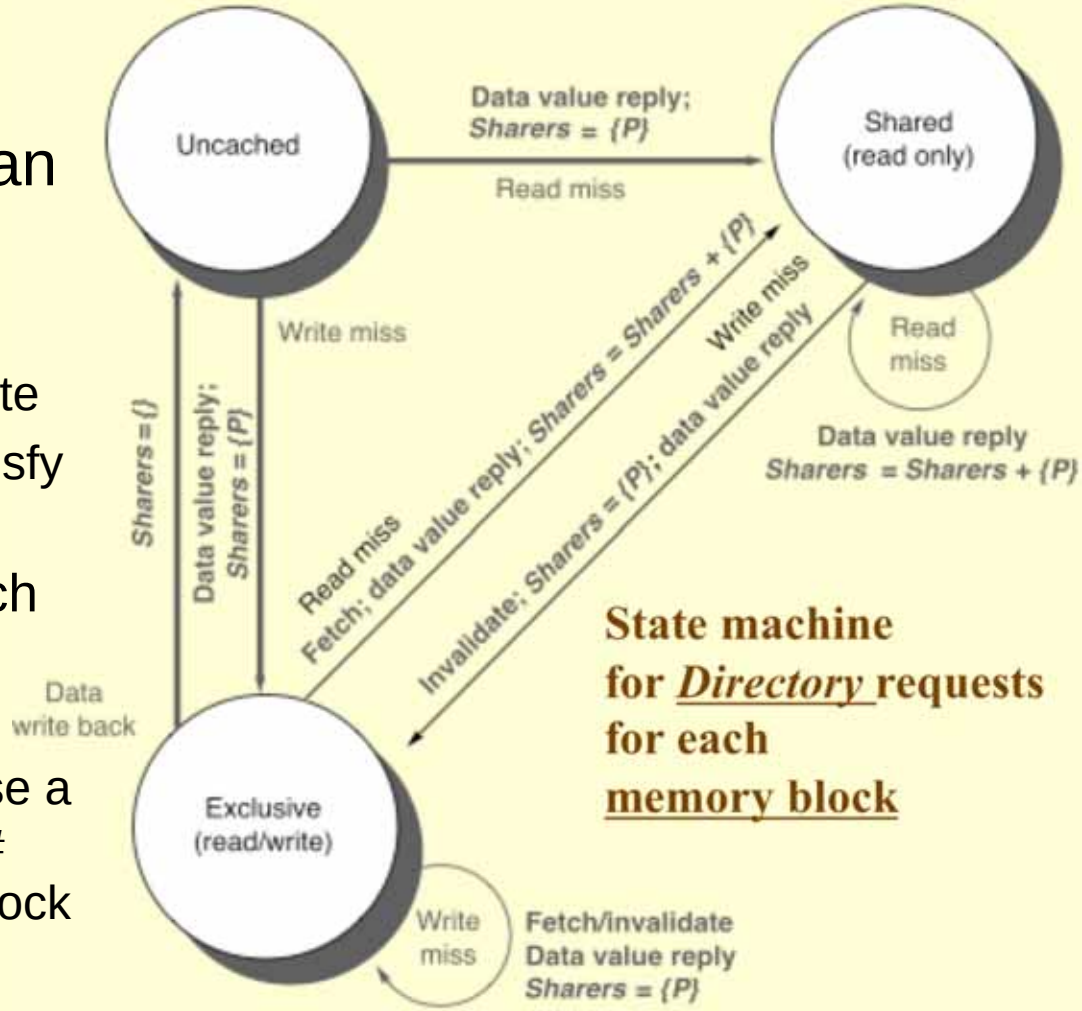
Cache Controller State Machine

- States identical to snoopy case
  - Transactions very similar.
    - Miss messages to home directory
    - Explicit invalidate & data fetch requests



Directory Controller State Machine

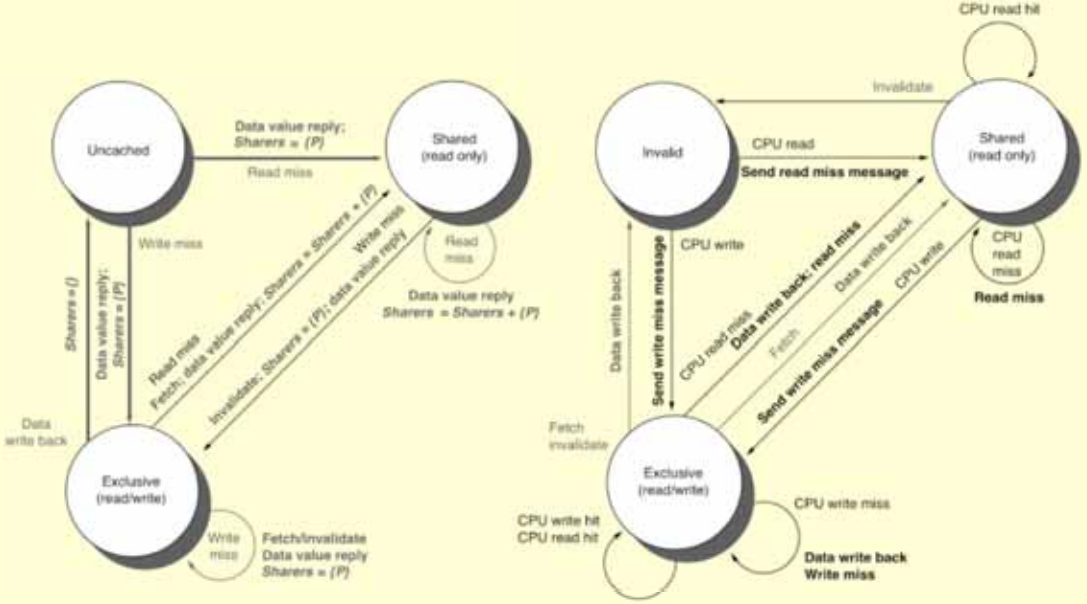
- Same states and structure as the transition diagram for an individual cache
  - Actions:
    - update of directory state
    - send messages to satisfy requests
  - Tracks all copies of each memory block
    - Sharers set implementation can use a bit vector of a size of # processors for each block



Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Actior	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1														
P1: Read A1														
P2: Read A1														
P2: Write 20 to A1														
P2: Write 40 to A2														

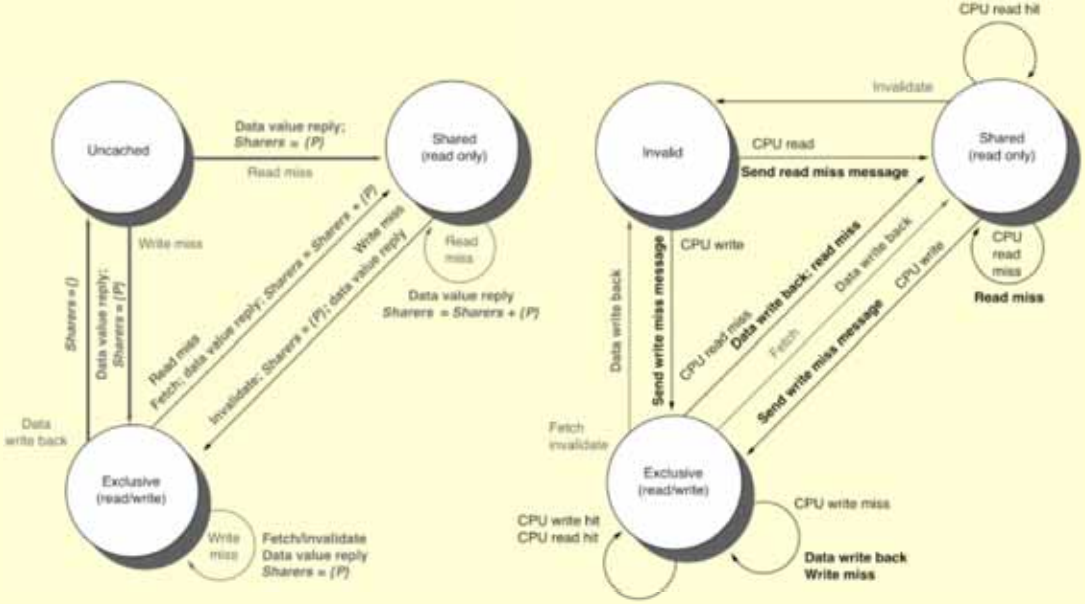
Assumes memory blocks A1 and A2 map to same cache block



Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1							*WrMs*	P1	A1		*A1*	*Ex*	*{P1}*	
	Excl.	*A1*	*10*				*DaRp*	P1	A1	0				
P1: Read A1														
P2: Read A1														
P2: Write 20 to A1														
P2: Write 40 to A2														

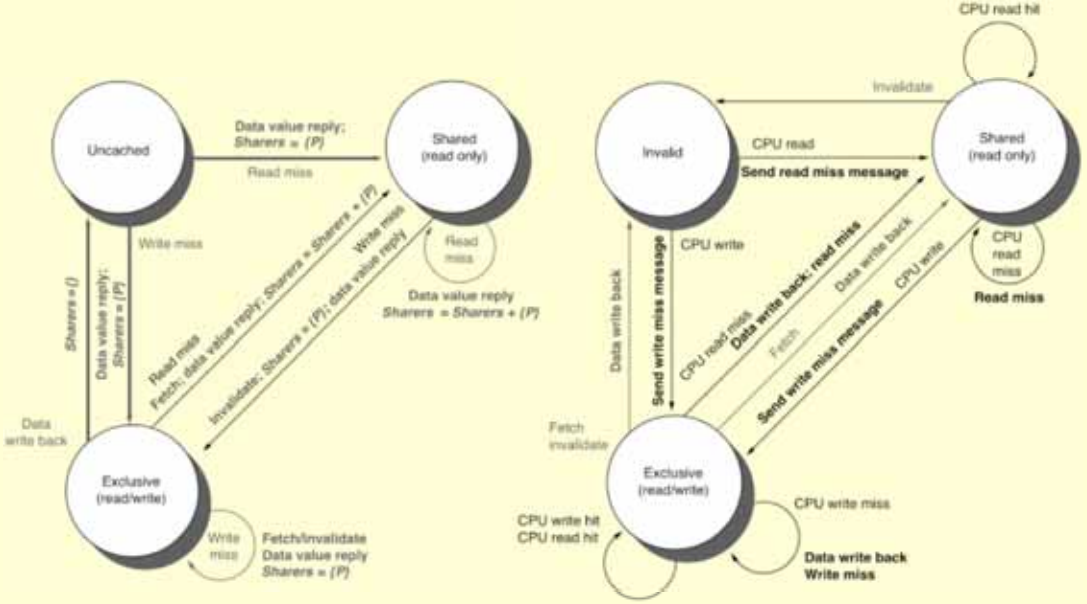
Assumes memory blocks A1 and A2 map to same cache block



Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1							*WrMs*	P1	A1		*A1*	*Ex*	*{P1}*	
	Excl.	*A1*	*10*				*DaRp*	P1	A1	0				
P1: Read A1	Excl.	A1	10											
P2: Read A1														
P2: Write 20 to A1														
P2: Write 40 to A2														

Assumes memory blocks A1 and A2 map to same cache block

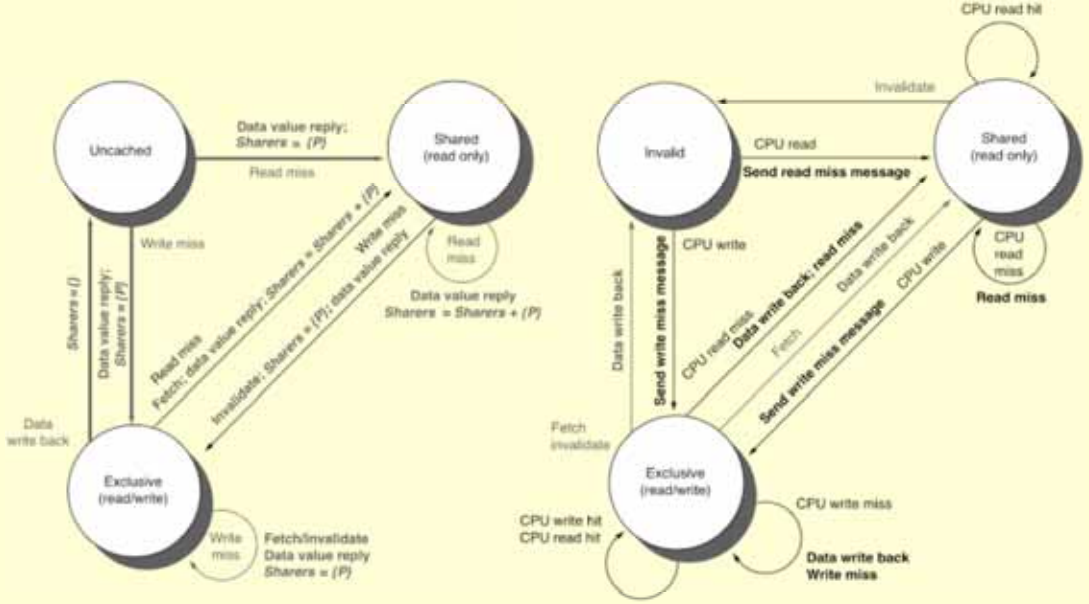


Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1							WrMs	P1	A1		A1	Ex	{P1}	
	Excl.	A1	10				DaRp	P1	A1	0				
P1: Read A1	Excl.	A1	10											
P2: Read A1				Shar.	A1		RdMs	P2	A1					
	Shar.	A1	10				Ftch	P1	A1	10			A1	10
				Shar.	A1	10	DaRp	P2	A1	10	A1	Shar.	{P1,P2}	10
P2: Write 20 to A1														
P2: Write 40 to A2														

Write Back

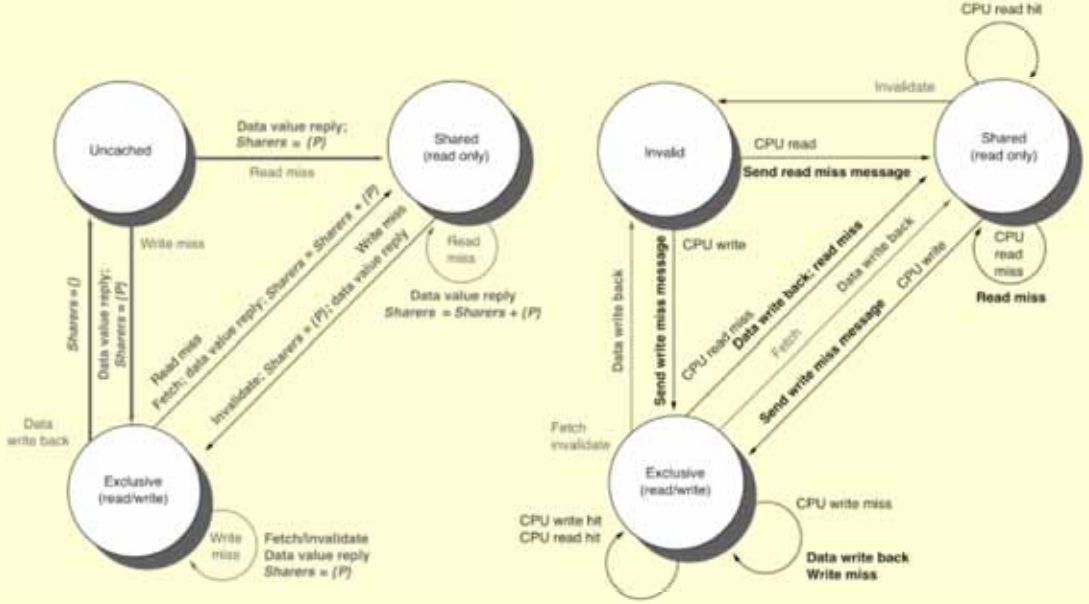
Assumes memory blocks A1 and A2 map to same cache block



Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1														
	Excl.	*A1*	*10*				*DaRp*	P1	A1	0				
P1: Read A1	Excl.	A1	10											
P2: Read A1				*Shar.*	*A1*		*RdMs*	P2	A1					
	Shar.	A1	10				*Fetch*	P1	A1	10			*A1*	*10*
				Shar.	A1	*10*	*DaRp*	P2	A1	10	A1	*Shar.*	*{P1,P2}*	10
P2: Write 20 to A1				Excl.	A1	*20*	*WrMs*	P2	A1					10
	Inv.						*Inval.*	P1	A1		A1	*Excl.*	*{P2}*	10
P2: Write 40 to A2														

Assumes memory blocks A1 and A2 map to same cache block



Example

	P1			P2			Bus				Directory			Memor
step	State	Addr	Value	State	Addr	Value	Action	Proc.	Addr	Value	Addr	State	{Procs}	Value
P1: Write 10 to A1							WrMs	P1	A1		A1	Ex	{P1}	
	Excl.	A1	10				DaRp	P1	A1	0				
P1: Read A1	Excl.	A1	10											
P2: Read A1				Shar.	A1		RdMs	P2	A1					
	Shar.	A1	10				Ftch	P1	A1	10			A1	10
				Shar.	A1	10	DaRp	P2	A1	10	A1	Shar.	{P1,P2}	10
P2: Write 20 to A1				Excl.	A1	20	WrMs	P2	A1					10
	Inv.						Inval.	P1	A1		A1	Excl.	{P2}	10
P2: Write 40 to A2							WrMs	P2	A2		A2	Excl.	{P2}	0
							WrBk	P2	A1	20	A1	Unca.	{}	20
				Excl.	A2	40	DaRp	P2	A2	0	A2	Excl.	{P2}	0

Assumes memory blocks A1 and A2 map to same cache block