

Cache Coherence Protocols

Quick Review and Examples

Snoopy Coherence Protocols

Request	Source	State of addressed cache block	Type of cache action	Function and explanation
Read hit	Processor	Shared or modified	Normal hit	Read data in local cache.
Read miss	Processor	Invalid	Normal miss	Place read miss on bus.
Read miss	Processor	Shared	Replacement	Address conflict miss: place read miss on bus.
Read miss	Processor	Modified	Replacement	Address conflict miss: write-back block, then place read miss on bus.
Write hit	Processor	Modified	Normal hit	Write data in local cache.
Write hit	Processor	Shared	Coherence	Place invalidate on bus. These operations are often called upgrade or <i>ownership</i> misses, since they do not fetch the data but only change the state.
Write miss	Processor	Invalid	Normal miss	Place write miss on bus.
Write miss	Processor	Shared	Replacement	Address conflict miss: place write miss on bus.
Write miss	Processor	Modified	Replacement	Address conflict miss: write-back block, then place write miss on bus.
Read miss	Bus	Shared	No action	Allow shared cache or memory to service read miss.
Read miss	Bus	Modified	Coherence	Attempt to share data: place cache block on bus and change state to shared.
Invalidate	Bus	Shared	Coherence	Attempt to write shared block; invalidate the block.
Write miss	Bus	Shared	Coherence	Attempt to write shared block; invalidate the cache block.
Write miss	Bus	Modified	Coherence	Attempt to write block that is exclusive elsewhere; write-back the cache block and make its state invalid in the local cache.

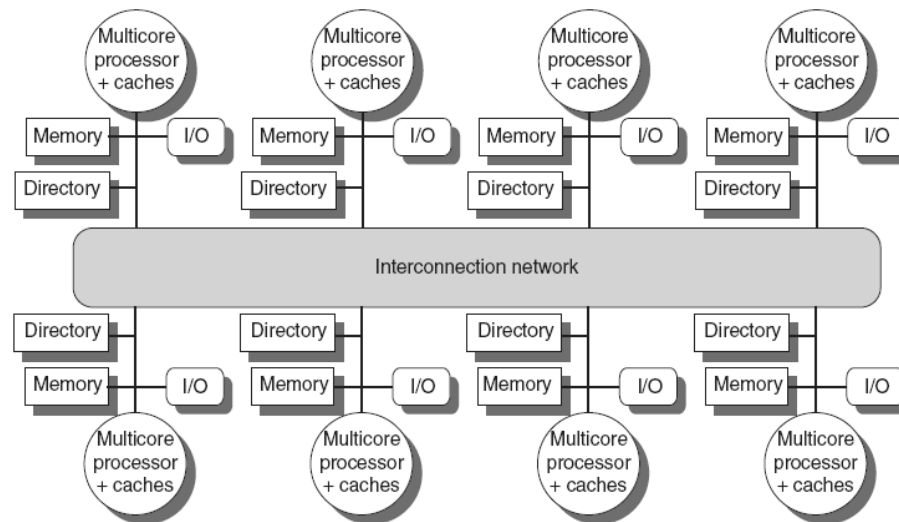
Snoopy Cache Coherence

	P1			P2			Bus			Mem		
	State	Addr	Value	State	Addr	Value	Action	Proc	Addr	Value	Addr	Value
P1 writes 10 to A1	Excl	A1	10				WM	P1	A1			
P1 reads A1	Excl	A1	10									
P2 reads A1							RM	P2	A1			
	Sh	A1	10								A1	10
				Sh	A1	10						
P2 writes 20 to A1							WM	P2	A1			
	Inv			Excl	A1	20						
P2 writes 40 to A2							WM	P2	A2		A1	20
				Excl	A2	40						

A1 and A2 maps to the same block -- when A2 is brought to the cache, A1 is written back to the memory --- WM=Write Miss RM=Read Miss

Directory Protocols

- Directory keeps track of every block
 - Which caches have each block
 - Dirty status of each block
- Implement in shared L3 cache
 - Keep bit vector of size = # cores for each block in L3
 - Not scalable beyond shared L3
- Implement in a distributed fashion:



Directory Protocols

- For each block, maintain state:
 - Shared
 - One or more nodes have the block cached, value in memory is up-to-date
 - Set of node IDs
 - Uncached
 - Modified
 - Exactly one node has a copy of the cache block, value in memory is out-of-date
 - Owner node ID
- Directory maintains block states and sends invalidation messages

Directory based

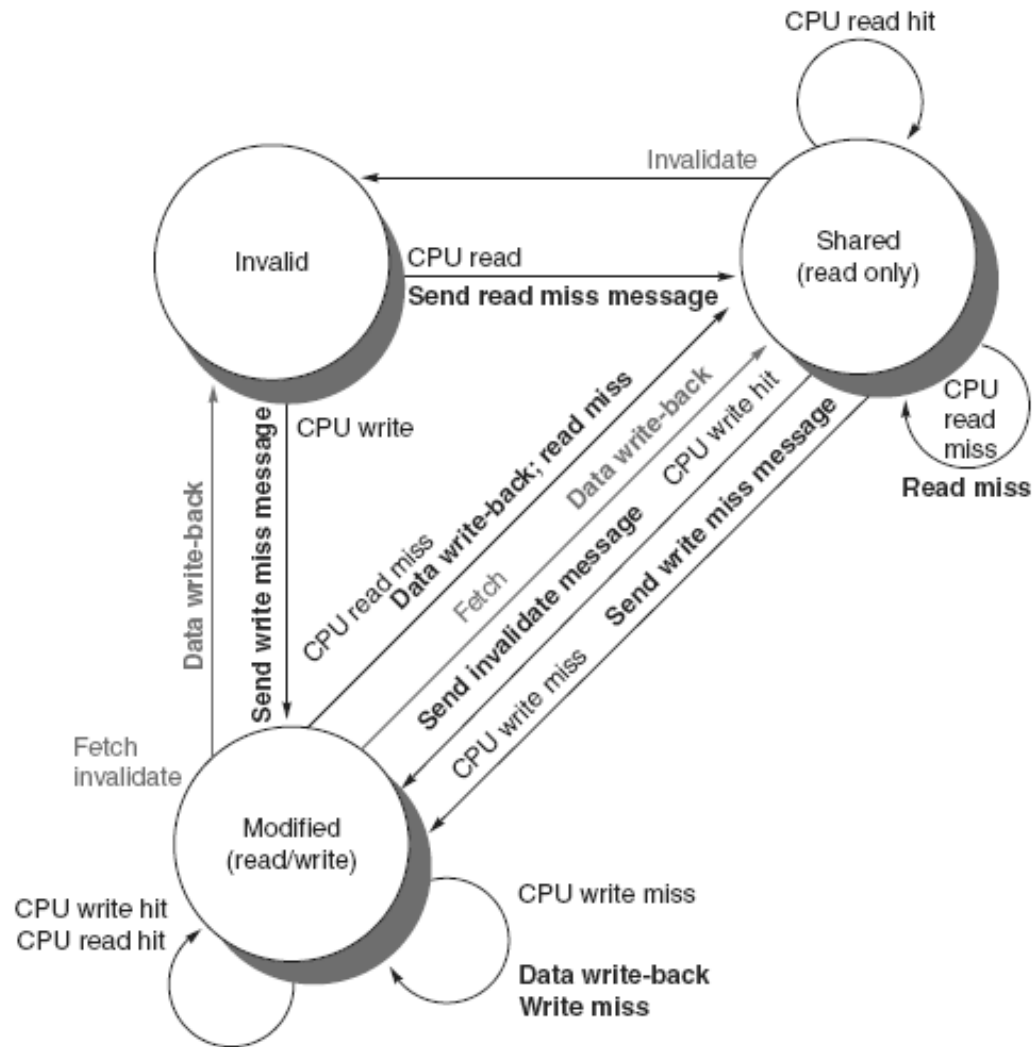
Message type	Source	Destination	Message contents	Function of this message
Read miss	Local cache	Home directory	P, A	Node P has a read miss at address A; request data and make P a read sharer.
Write miss	Local cache	Home directory	P, A	Node P has a write miss at address A; request data and make P the exclusive owner.
Invalidate	Local cache	Home directory	A	Request to send invalidates to all remote caches that are caching the block at address A.
Invalidate	Home directory	Remote cache	A	Invalidate a shared copy of data at address A.
Fetch	Home directory	Remote cache	A	Fetch the block at address A and send it to its home directory; change the state of A in the remote cache to shared.
Fetch/invalidate	Home directory	Remote cache	A	Fetch the block at address A and send it to its home directory; invalidate the block in the cache.
Data value reply	Home directory	Local cache	D	Return a data value from the home memory.
Data write-back	Remote cache	Home directory	A, D	Write-back a data value for address A.

Directory based

	P1			P2			Net				Dir			
	state	Addr	Value	State	Addr	Val	Action	Proc	Add	Value	Add	State	Proc	Value
P1 writes 10 to A1	Ex	A1	10				WM	P1	A1		A1	Ex	{P1}	
P1 reads A1	Ex	A1	10											
P2 reads A1							WM	P2	A1		A1	Sh	{P1,2}	
	Sh	A1	10	Sh	A1	10								
P2 writes 20 to A1							WM	P2	A1		A1	Ex	{P1}	
	Inv			Ex	A1	20								
P2 writes 40 to A2							WM	P2	A2		A1 A2	--- EX	--- {P2}	20
				EX	A2	40								

A1 and A2 maps to the same block -- when A2 is brought to the cache, A1 is written back to the memory

Directory Protocols



Directory Protocols

- For uncached block:
 - Read miss
 - Requesting node is sent the requested data and is made the only sharing node, block is now shared
 - Write miss
 - The requesting node is sent the requested data and becomes the sharing node, block is now exclusive
- For shared block:
 - Read miss
 - The requesting node is sent the requested data from memory, node is added to sharing set
 - Write miss
 - The requesting node is sent the value, all nodes in the sharing set are sent invalidate messages, sharing set only contains requesting node, block is now exclusive

Directory Protocols

- For exclusive block:
 - Read miss
 - The owner is sent a data fetch message, block becomes shared, owner sends data to the directory, data written back to memory, sharers set contains old owner and requestor
 - Data write back
 - Block becomes uncached, sharer set is empty
 - Write miss
 - Message is sent to old owner to invalidate and send the value to the directory, requestor becomes new owner, block remains exclusive