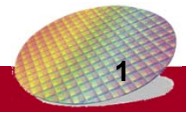
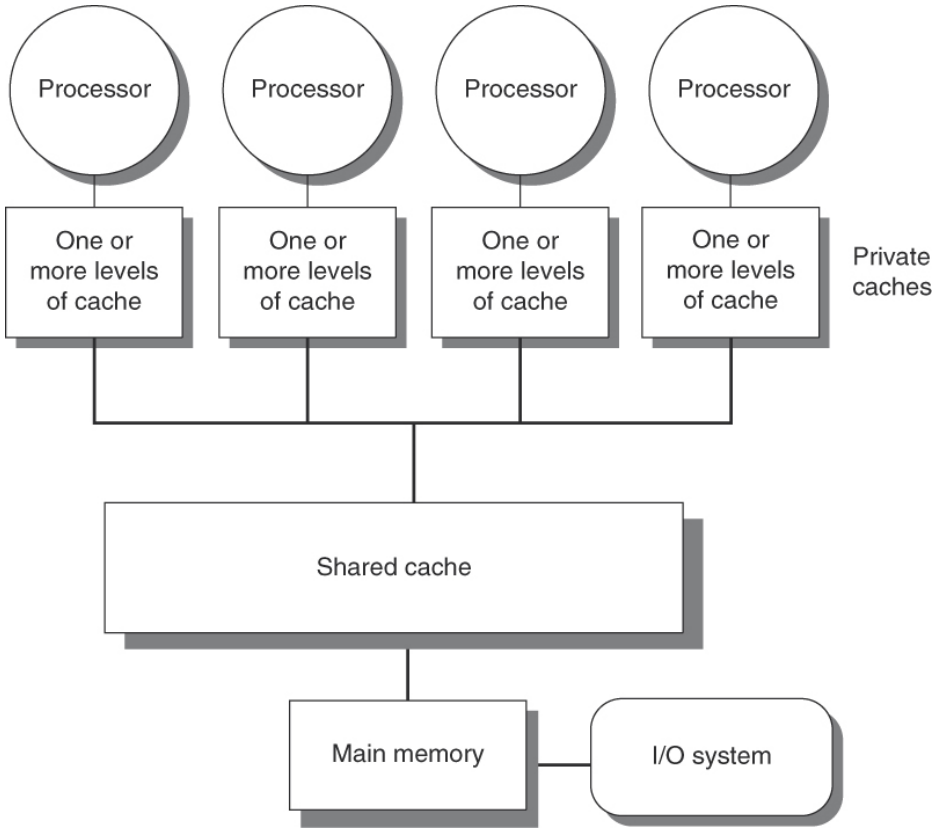
Coherence - Directory



Recall: Shared-memory multiprocessors (SMP)

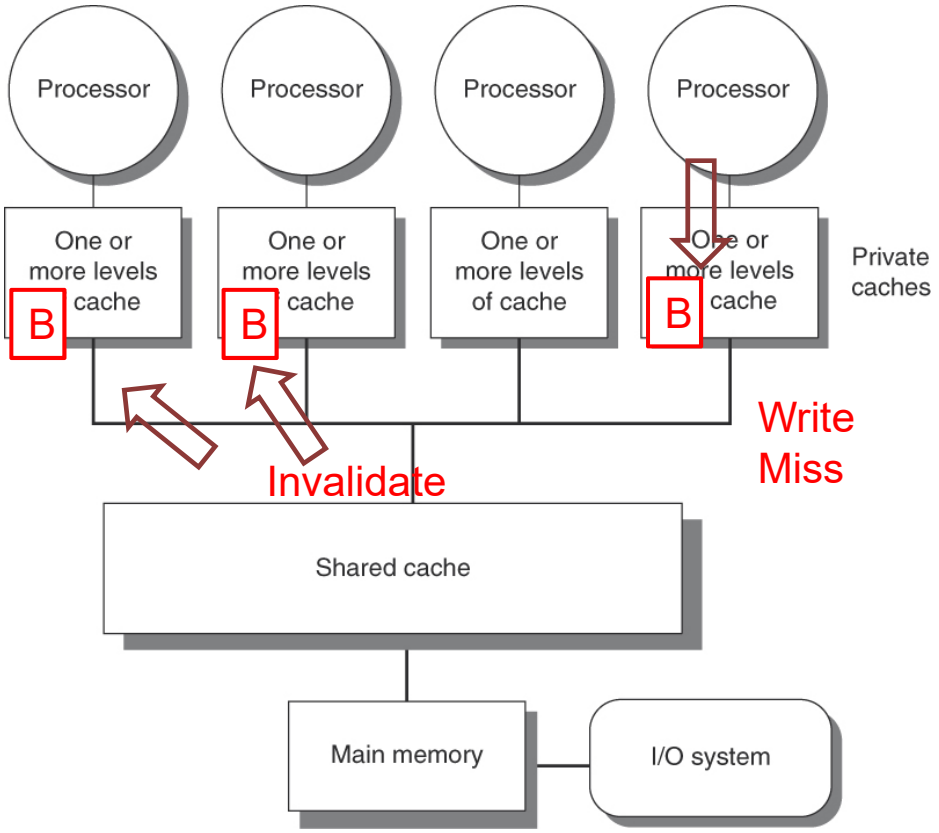
- Centralized multiprocessors
  - Share a single centralized memory all processors have equal access to
  - Small number of corers (normally < 100)
  - With large shared cache or shared memory to support memory bandwidth requirement
  - A.k.a. symmetric multiprocessors, uniform memory access multiprocessor (UMA)



Symmetric multiprocessor (UMA)

Recall: Snooping Protocol for SMP

- Snooping Protocol
  - Requires communication w/ all caches on every cache miss
    - When write miss on a shared block, need to invalidate all other shared block
  - No centralized data structure that tracks the states of the caches
    - Use broadcast
    - Pros: inexpensive
    - Cons: poor scalability
  - Not suitable for a distributed-memory multiprocessor

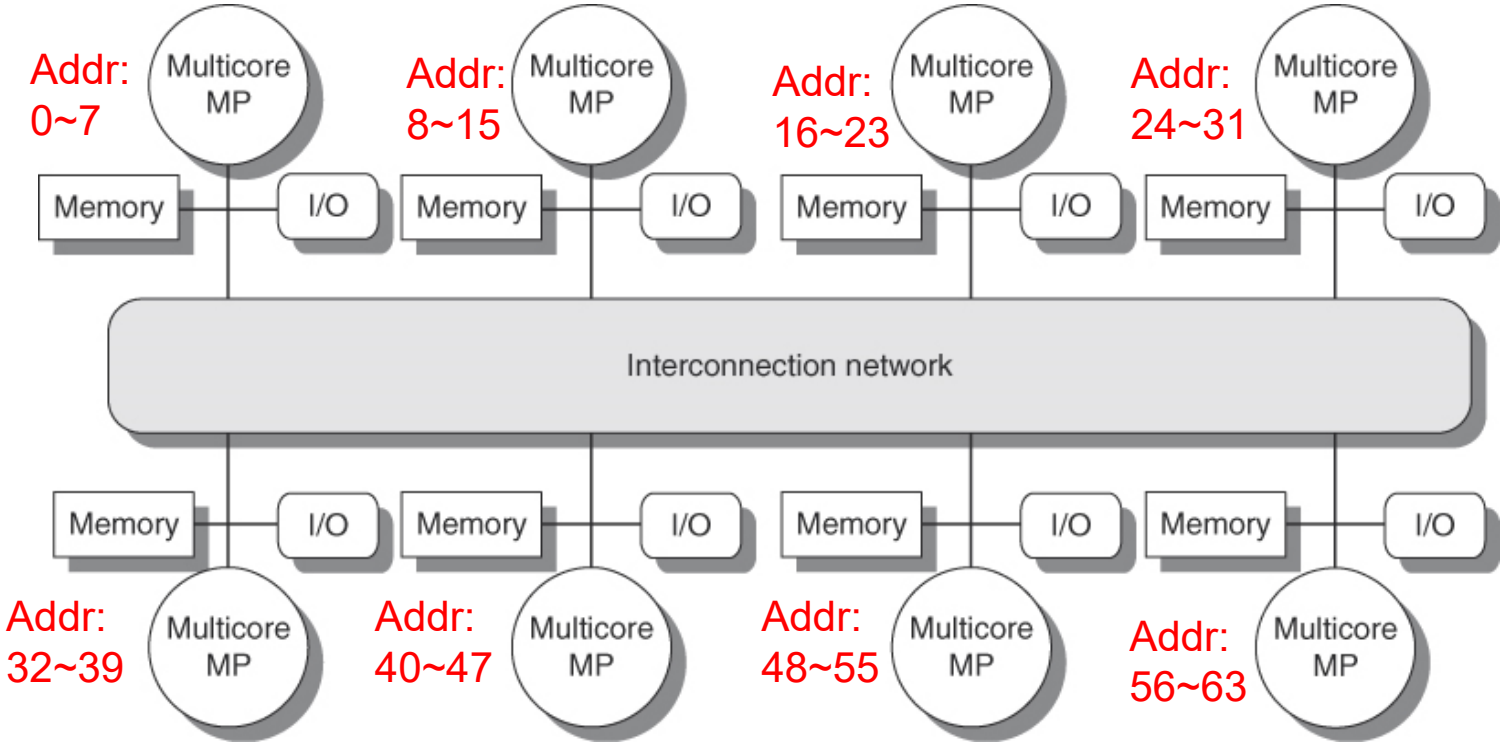


Basic Structure of Distributed-memory Multiprocessor

- Separate local memory traffic from remote memory traffic

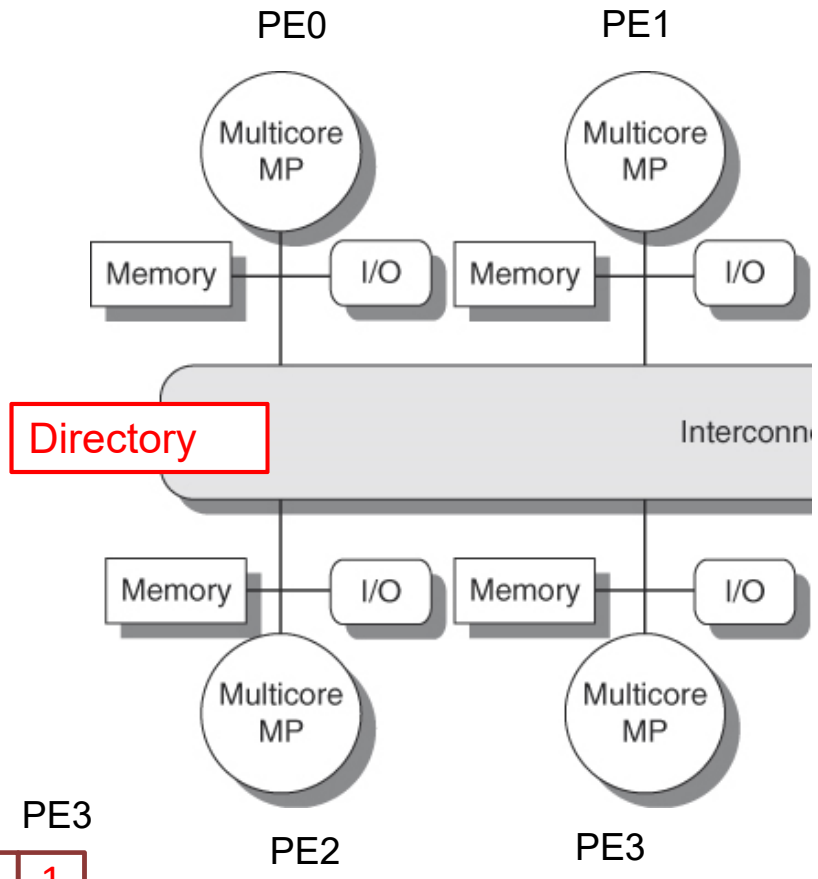
⇒ Reduce the bandwidth demands on the memory system & interconnection network

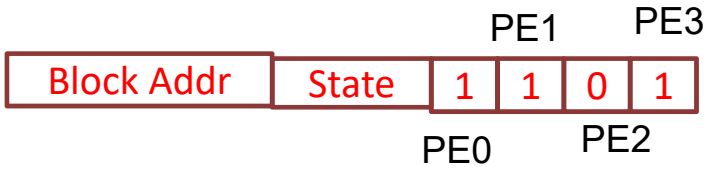
- Separate memory address space



Directory-based Protocol for Distributed-Memory Multiprocessor

- Directory: hardware to track the state of each memory block
  - Information includes which cache have copies of the block, whether if the block is clean or dirty, and so on
- An example of implementation
  - Associative an entry in the directory with each memory block
  - A bit vector in the entry indicate which private memory have copies of a block.
    - Invalidations are only sent to these memory



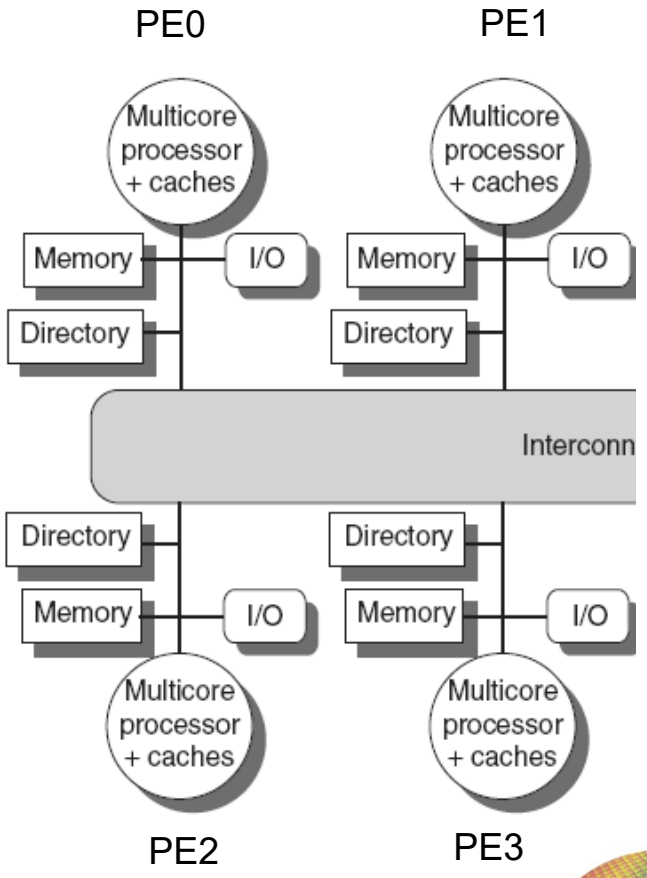


Directory Size

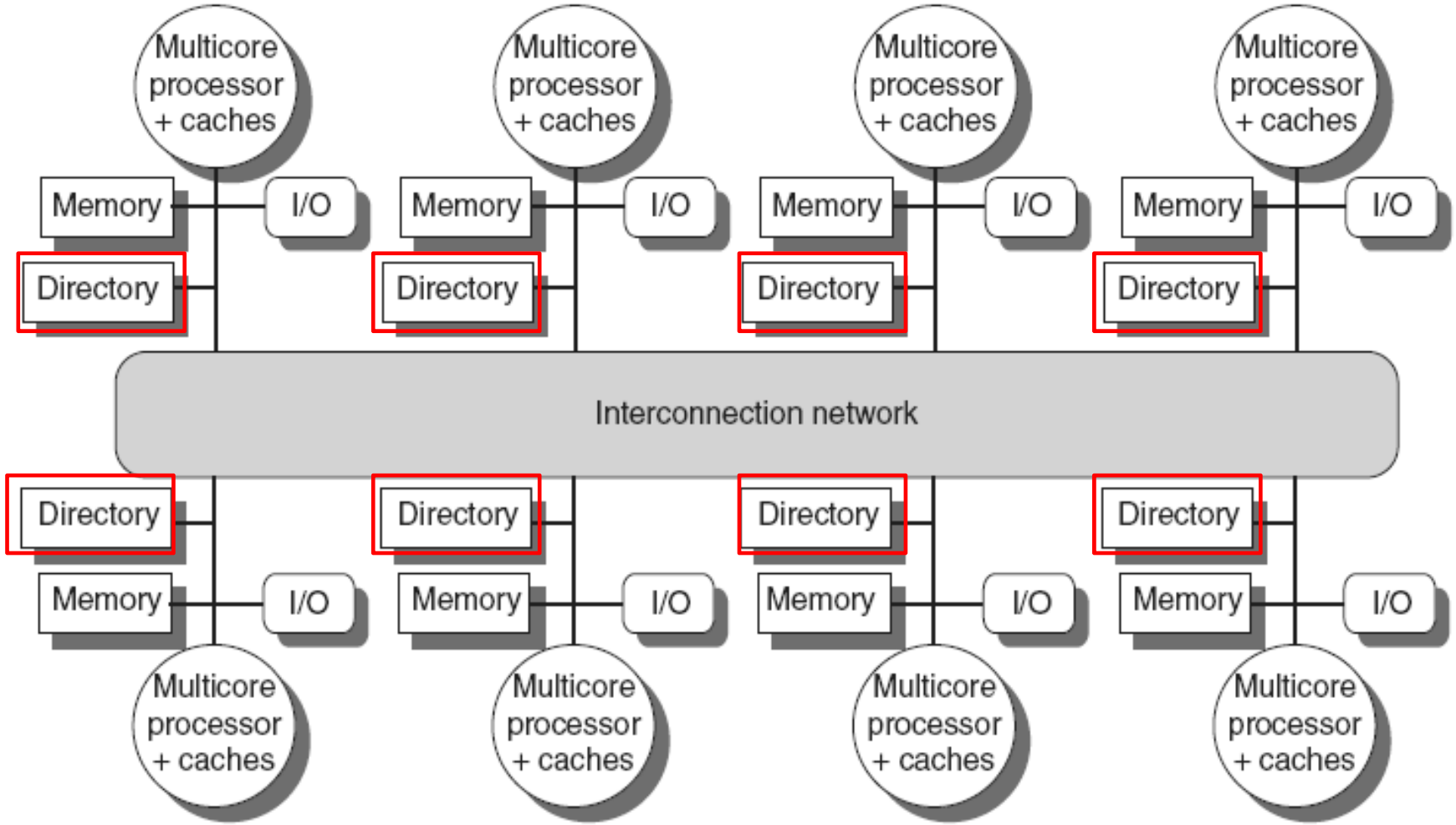
Directory size = # total memory block * info size
= # memory block of each PE * #PE * info size

Block Addr	State	1	1	0	1

- Directory can be centralized or distributed
  - Prevent the directory from becoming the bottleneck
  - Each PE has a directory to handle it physical memory => Distributed directory



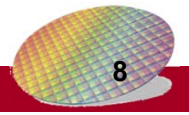
Structure of Distributed Memory Multiprocessor with distributed Directory



A. Directory-based Cache Coherence Protocols: The basics

- Two primary operations
  - Handling a read miss
  - Handling a write to a shared, clean cache block
- Write miss to a shared block= read miss + write to shared clean block)

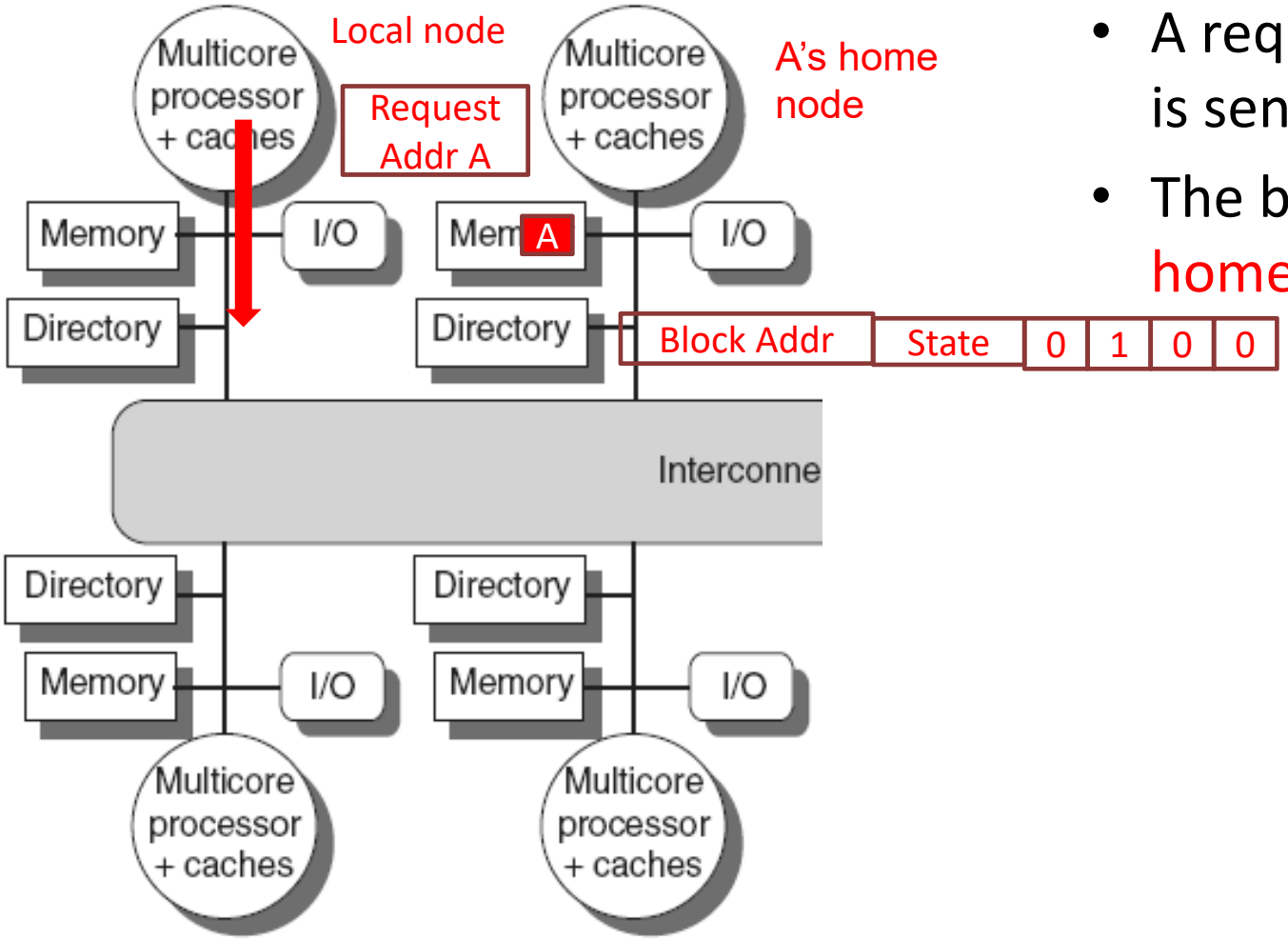
- Block state in Directory:
  - Shared: one or more PE have the block cached, and the value in memory is up-to-date (as well as in all caches)
  - Uncached: no PE has a copy of the cache block
  - Modified (Exclusive): exactly one PE has a copy of the cache block, and it has written the block. So the memory copy is out of date (The processor is called the owner of the cache block)



Type of Nodes: local, home, and remote

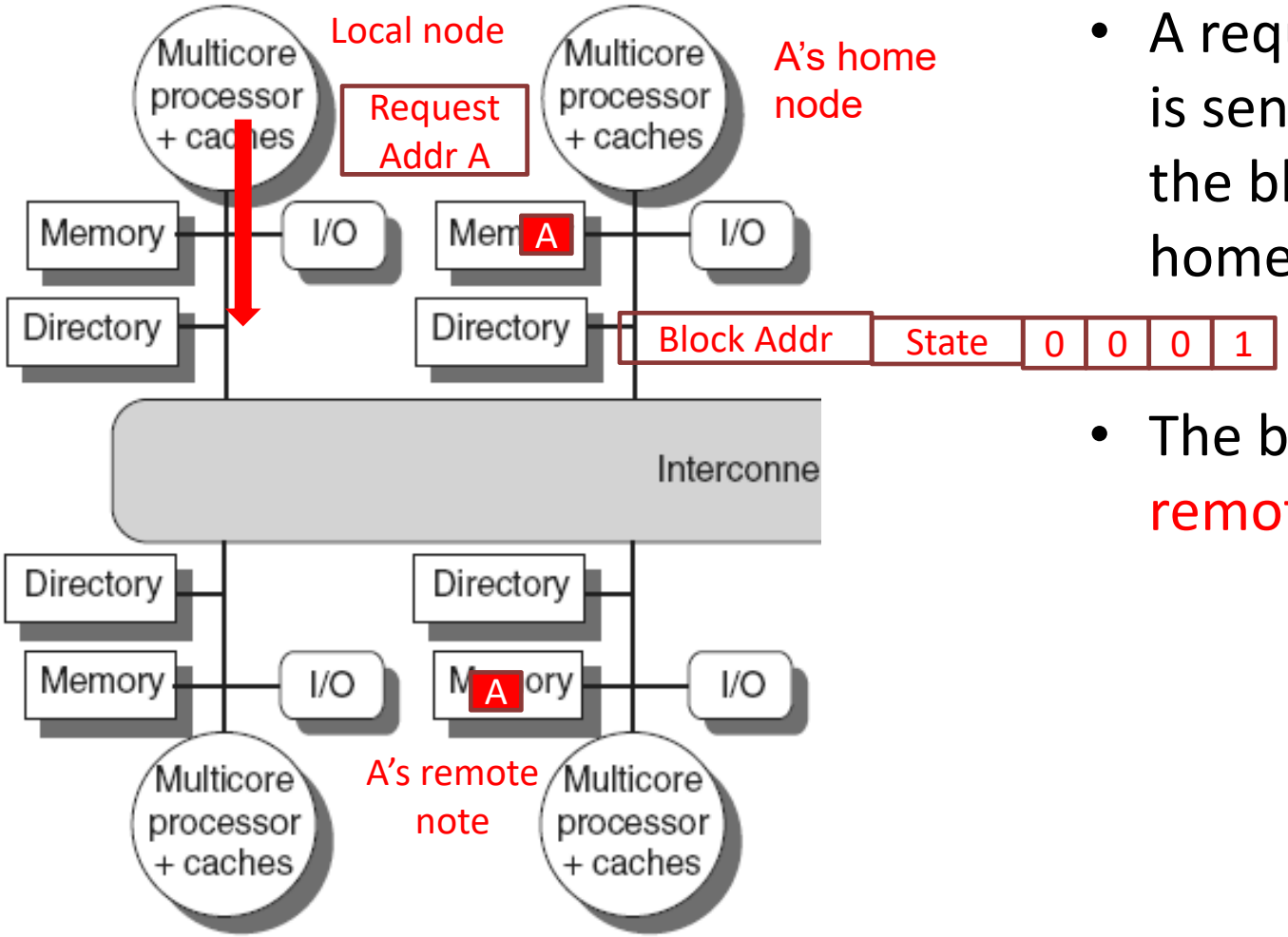
- Local node: the node where a request originates
- Home node: the node where the memory location and the directory entry of an address reside
  - The local node may also be the home node
  - The directory must be accessed when the home node is in the local node
- Remote node: the node that has a copy of a cache block, whether exclusive or shared

Type of Nodes: local, home and remote node



- A request for Address A is sent by a local node
- The block is in the home node

Type of Nodes: local, home and remote node



- A request for Address A is sent by a local node, the block is not is the home node
- The block is in the remote node

Directory

- Directory can
  - Track the state of each potentially shared memory block
  - Track the processors that have copies of the block when it is shared => Those copies will need to invalidated on a write
- Difference between directory and snooping

  - Not broadcast
    - many messages must have explicit response (message with address)
    - Assumption: all messages will be received and acted upon in the same order they are sent
      - Ensure that invalidate sent by a PE are honored immediately
  - The interconnection is not longer a bus
    - The interconnect cannot be used as a single point of arbitration

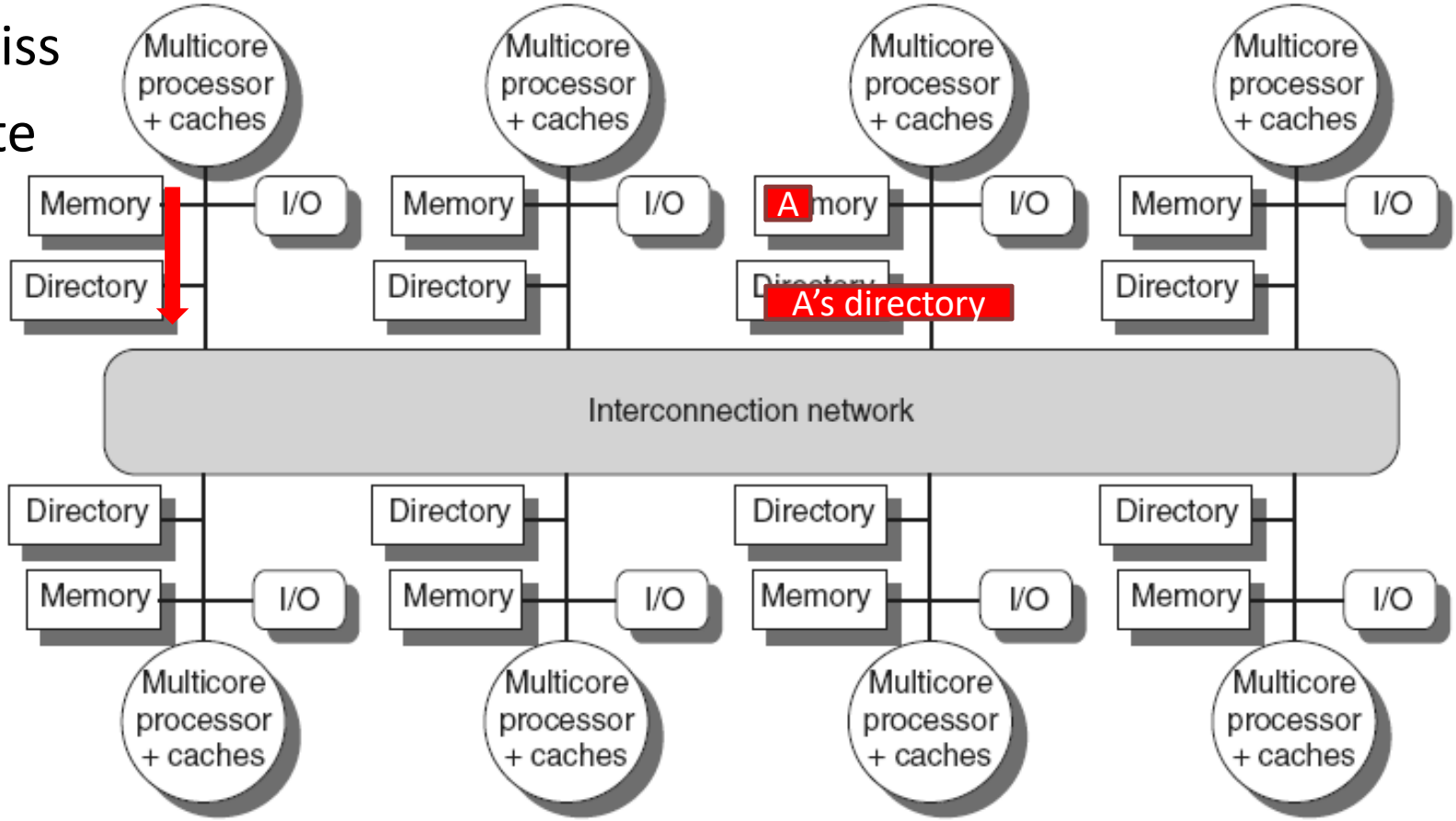
Types of message sent among nodes

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

P: requesting Processor

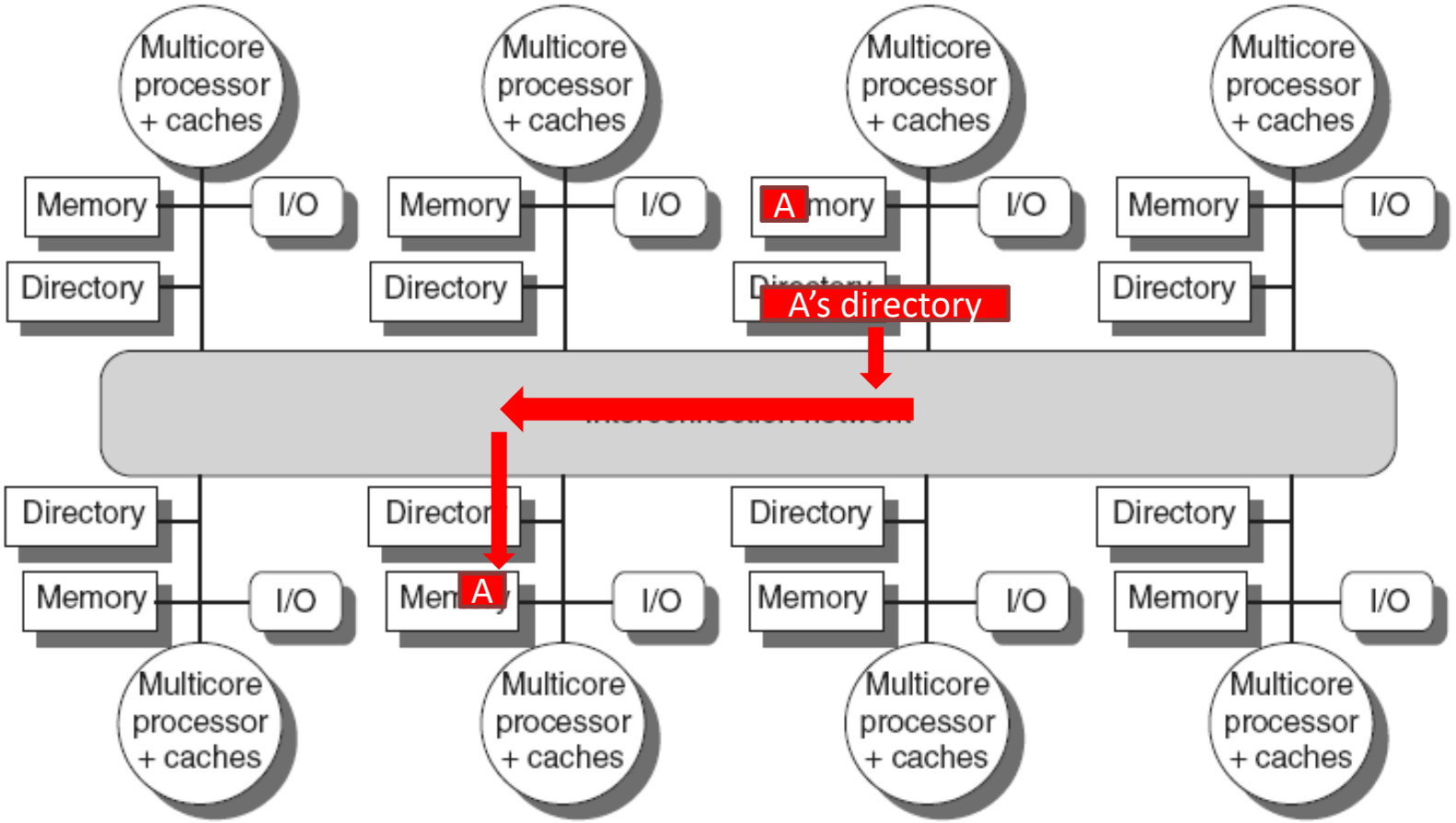
Case 1~3: From local cache to home directory

- Case 1: Read Miss
- Case 2: Write Miss
- Case 3: Invalidate



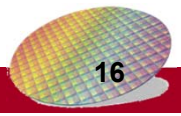
Case 4~6: From local cache to home directory

- Case 4: Invalidate
- Case 5: Fetch
- Case 6: Fetch/Invalidate



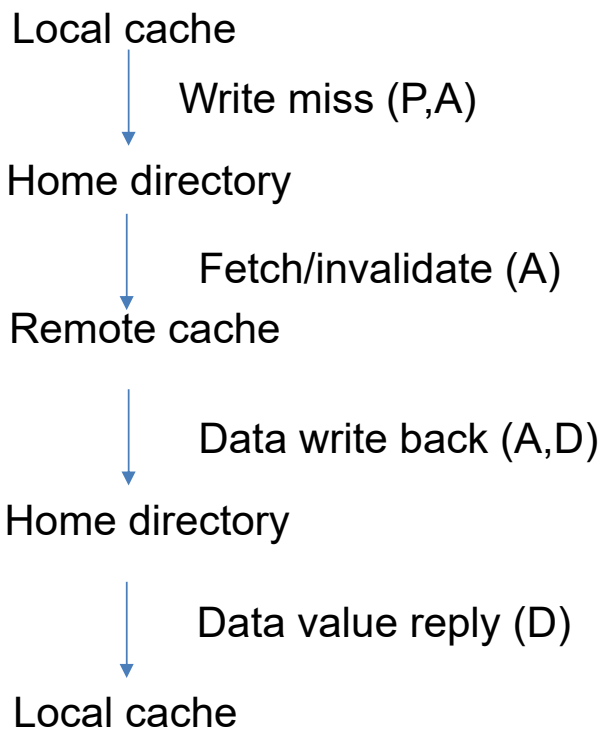
- 1-3: requests sent by the local cache to the home directory
- 4-6: messages sent to a remote cache by the home when the home needs the data to satisfy a read or write miss request
- 7: send a value from home back to the requesting node
- 8: data value write backs occur for two reasons
  - A block is replaced in a cache and must be written back to the home directory
  - In reply to fetch or fetch/invalidate messages from the home

Message type	Source	Destination	Msg content
Read miss	Local cache	Home directory	P, A
Write miss	Local cache	Home directory	P, A
Invalidate	Local cache	Home directory	A
Invalidate	Home directory	Remote cache	A
Fetch	Home directory	Remote cache	A
Fetch/invalidate	Home directory	Remote cache	A
Data value reply	Home directory	Local cache	D
Data write-back	Remote cache	Home directory	A, D



- A simple example when write miss occurs when a processor writes its cache

P: requesting PE
A: requested addr
D: data contents



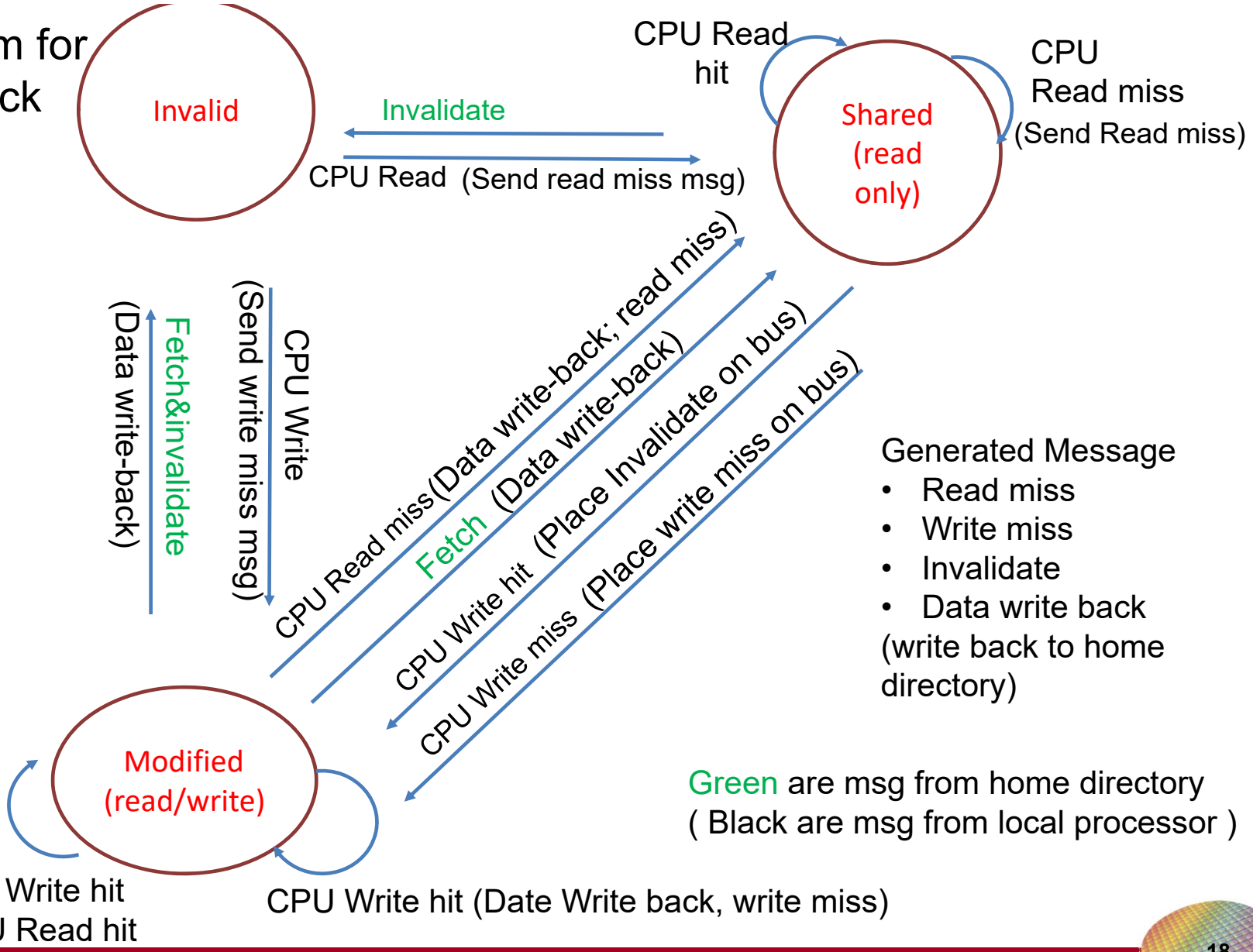
Message type	Source	Destination	Msg content
Read miss	Local cache	Home directory	P, A
Write miss	Local cache	Home directory	P, A
Invalidate	Local cache	Home directory	A
Invalidate	Home directory	Remote cache	A
Fetch	Home directory	Remote cache	A
Fetch/invalidate	Home directory	Remote cache	A
Data value reply	Home directory	Local cache	D
Data write-back	Remote cache	Home directory	A, D

State transition Diagram for an individual cache block

State of a cache block
- Invalid
- Shared
- Modified (exclusive)

Req/Msg received from local processor
- Read hit
- Read Miss
- Write hit
- Write Miss (exclusive)

Req/Msg received from home directory
- Fetch
- Invalidate
- Fetch&invalidate



Generated Message
- Read miss
- Write miss
- Invalidate
- Data write back

(write back to home directory)

Green are msg from home directory
(Black are msg from local processor)

State transition Diagram for an individual cache block

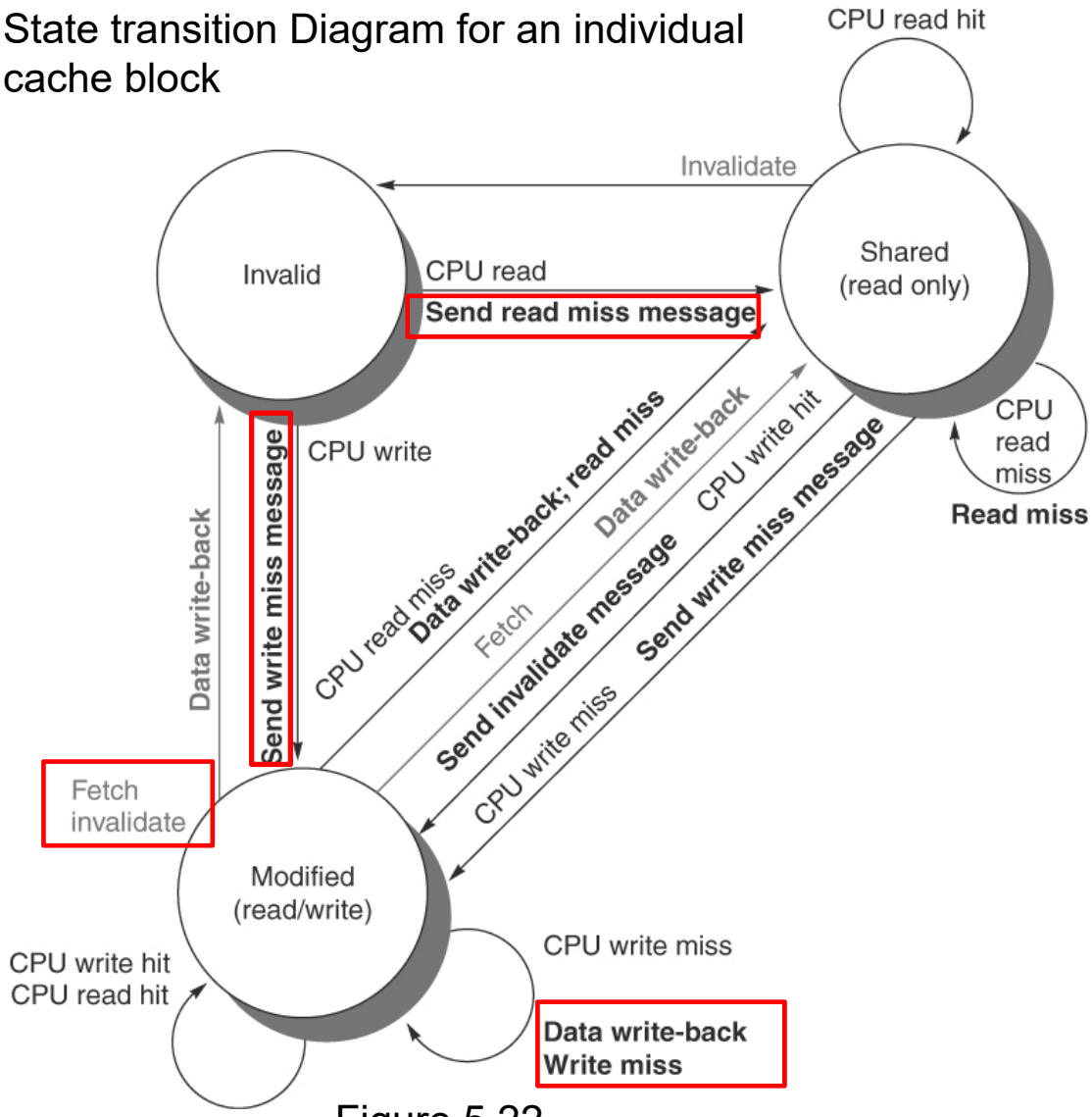


Figure 5.22

Cache coherence State diagram (snooping)

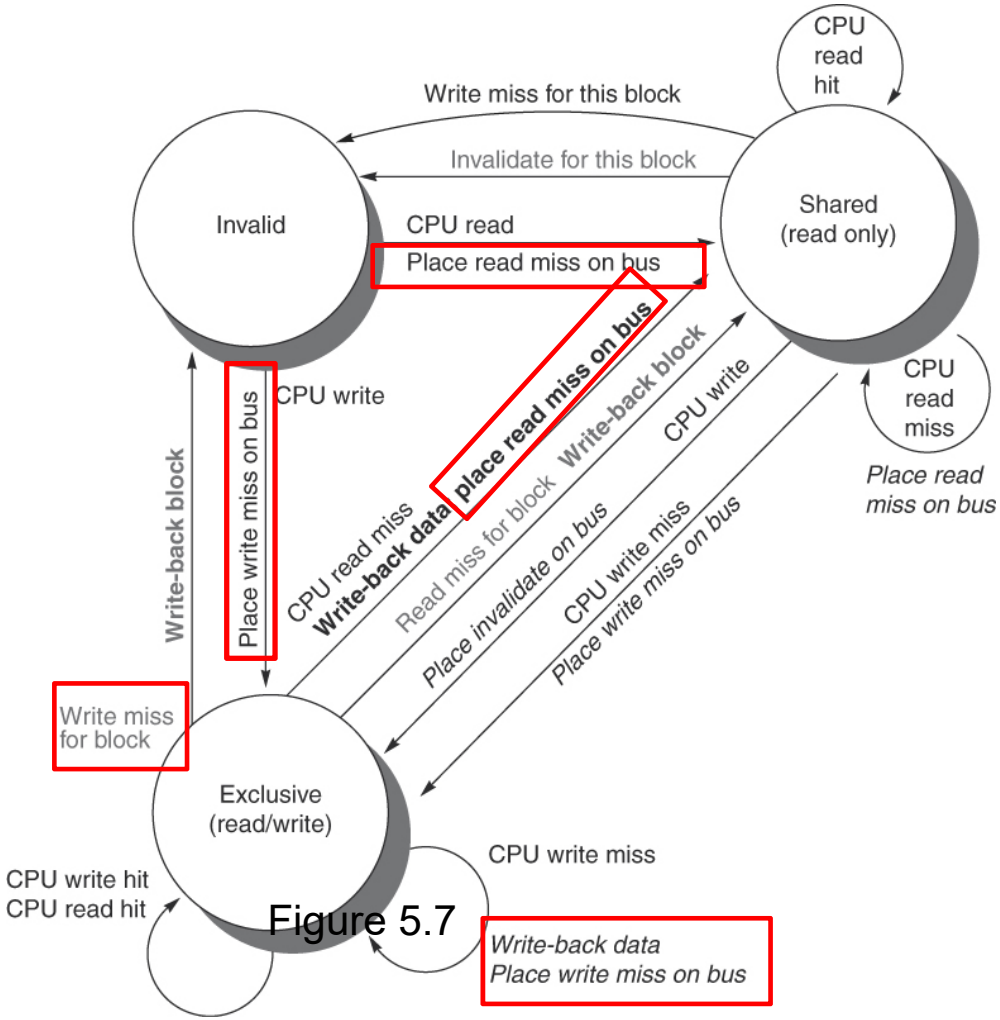


Figure 5.7

State Transition Diagram for individual cache block

- Difference between directory & snooping
  - Explicit invalid and write back requests in directory scheme
  - Broadcast write miss on bus in the snooping scheme
  - Date fetch&invalidate operations that are selectively sent by the directory controller
- Same in both
  - Attempt to write a shared cache block is treated as a miss (the same for snooping)
  - Any cache block must be in the exclusive state when it is written, and any shared block must be up-to-date in memory (the same as snooping)

State Transition Diagram for Entry in Directory

State of a memory block (in the directory)

- Uncached: block is the memory (has the latest data)
- Shared: memory value is up-to-date
- Modified (exclusive): the block is in a node identified by Sharers

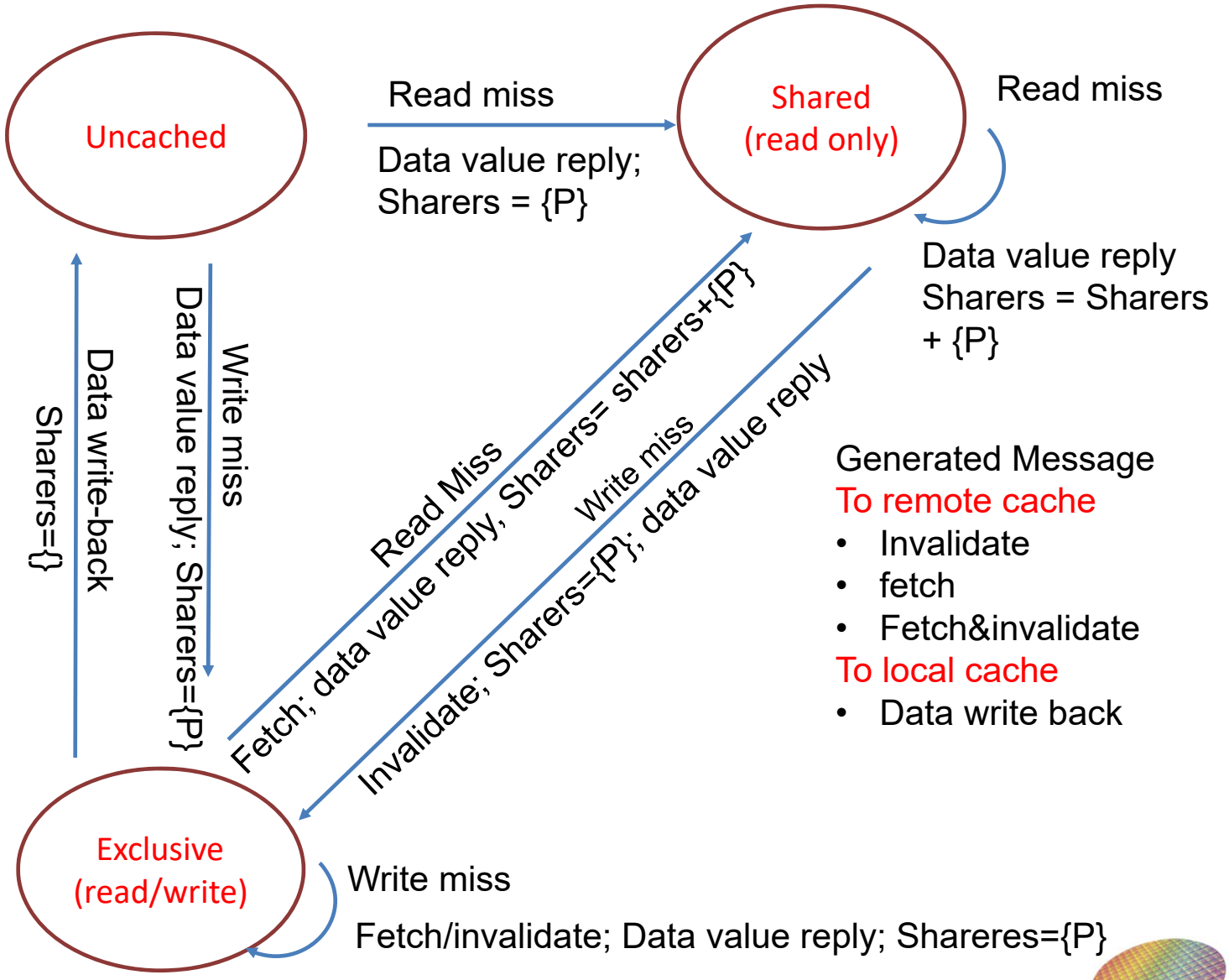
Message received from local cache

- read miss
- write miss

Msg received from remote cache

- Data write back

Sharers: PEs that have a copy of the block



Generated Message

To remote cache

- Invalidate
- fetch
- Fetch&invalidate

To local cache

- Data write back

Remarks

- A message sent to a directory causes 2 different types of actions:
  - Update of directory state
  - Sending additional messages to satisfy the request
- The directory state indicates the state of all the cached copies of a memory block, rather than for a single cache block
- The directory must track the processors (sharers) that have a copy of a block
- Assumption: actions are atomic
  - e.g. requesting a value and sending it to another not (not realistic)

Backup Slides