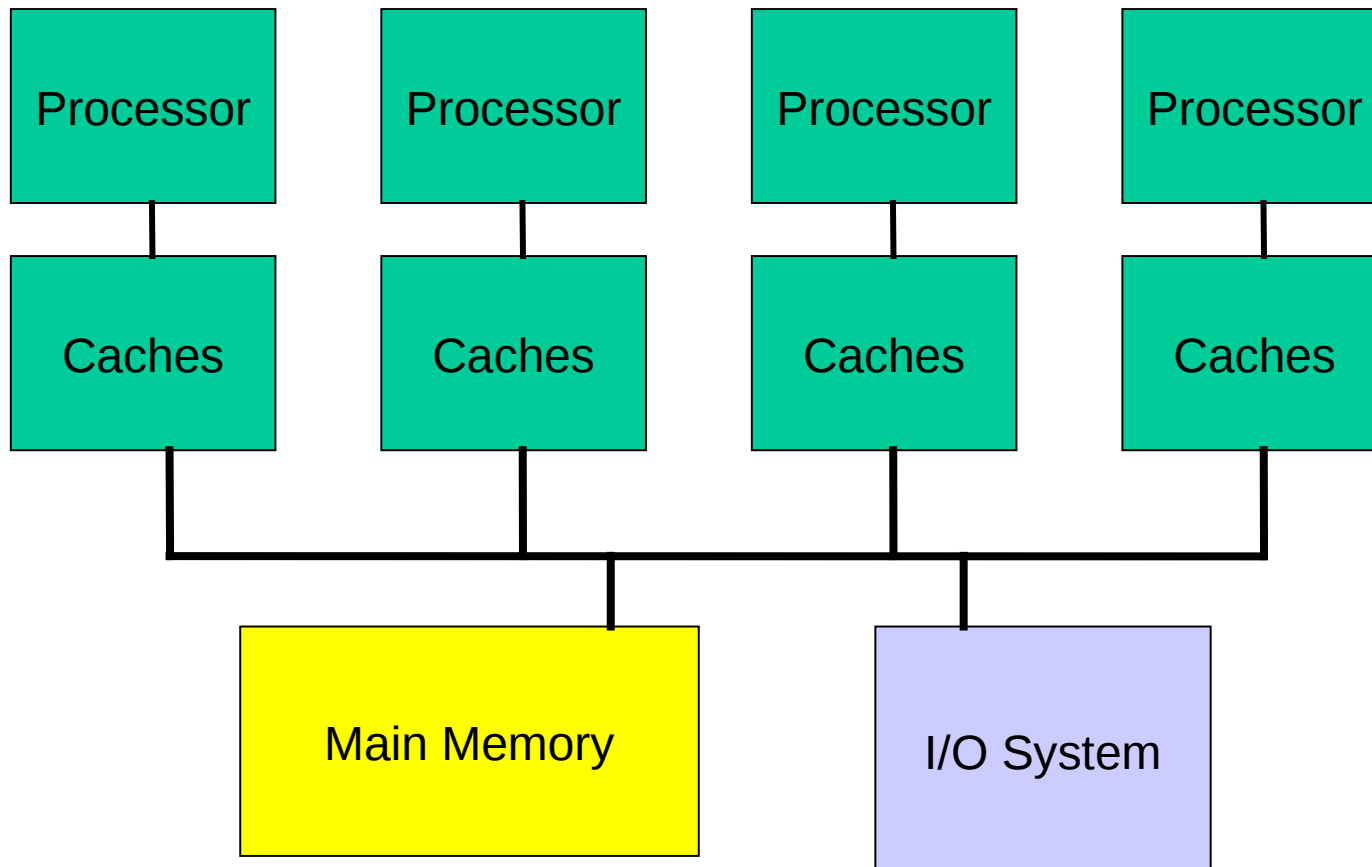# 250P: Computer Systems Architecture

Lecture 13: Cache-Coherence

Anton Burtsev
February, 2019

SMP/UMA/Centralized Memory Multiprocessor

SMPs

- Centralized main memory and many caches → many copies of the same data

- A system is cache coherent if a read returns the most recently written value for that word

| Time	Event	Value of X in Cache-A	Cache-B	Memory
0		-	-	1
1	CPU-A reads X	1	-	1
2	CPU-B reads X	1	1	1
3	CPU-A stores 0 in X	0	1	0

Cache Coherence

A memory system is coherent if:

- Write propagation: P1 writes to X, sufficient time elapses, P2 reads X and gets the value written by P1

- Write serialization: Two writes to the same location by two processors are seen in the same order by all processors

- The memory *consistency* model defines "time elapsed" before the effect of a processor is seen by others and the ordering with R/W to other locations (loosely speaking – more later)
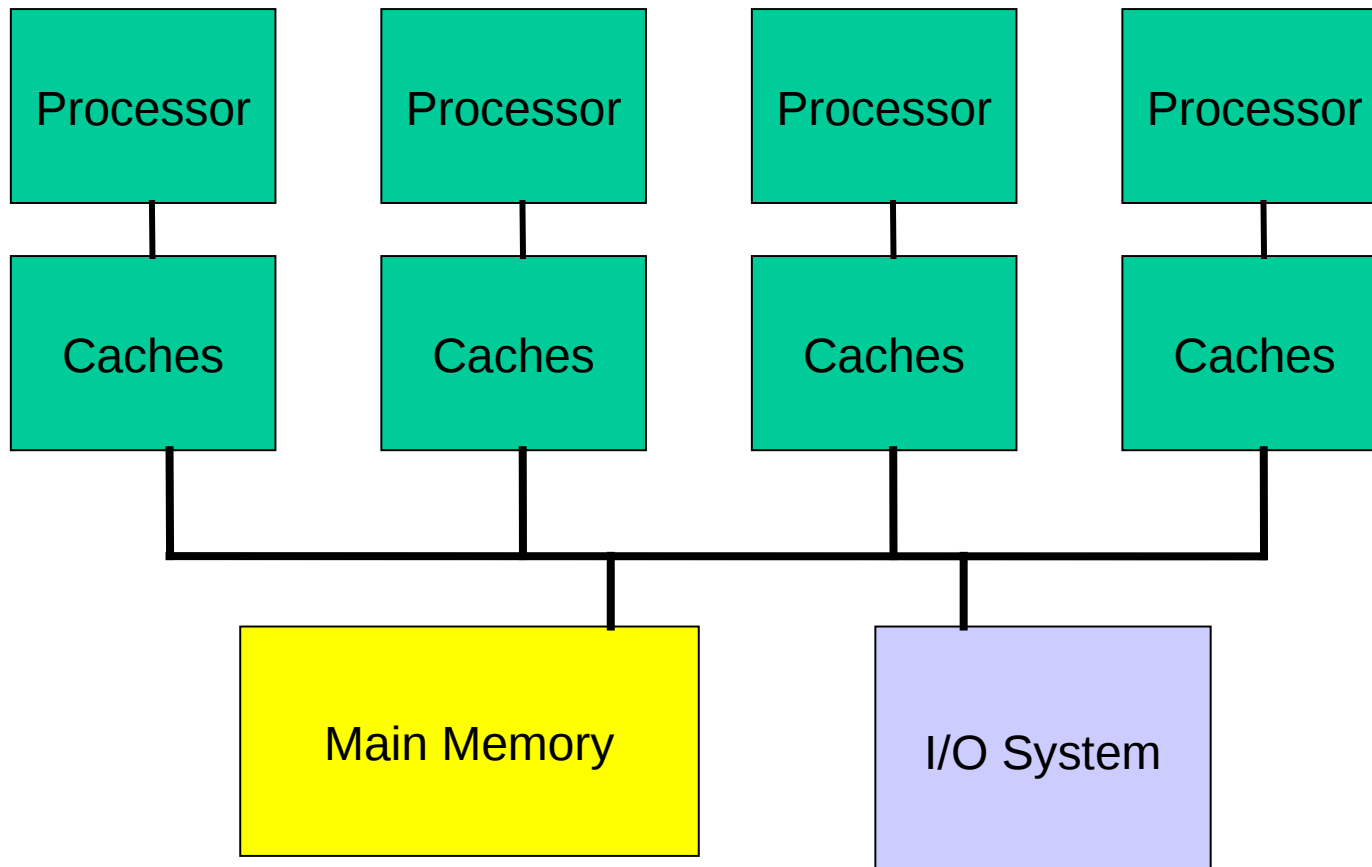
Cache Coherence Protocols

- Directory-based: A single location (directory) keeps track of the sharing status of a block of memory

- Snooping: Every cache block is accompanied by the sharing status of that block – all cache controllers monitor the shared bus so they can update the sharing status of the block, if necessary

  ➢ Write-invalidate: a processor gains exclusive access of a block before writing by invalidating all other copies
  ➢ Write-update: when a processor writes, it updates other shared copies of that block

SMPs or Centralized Shared-Memory
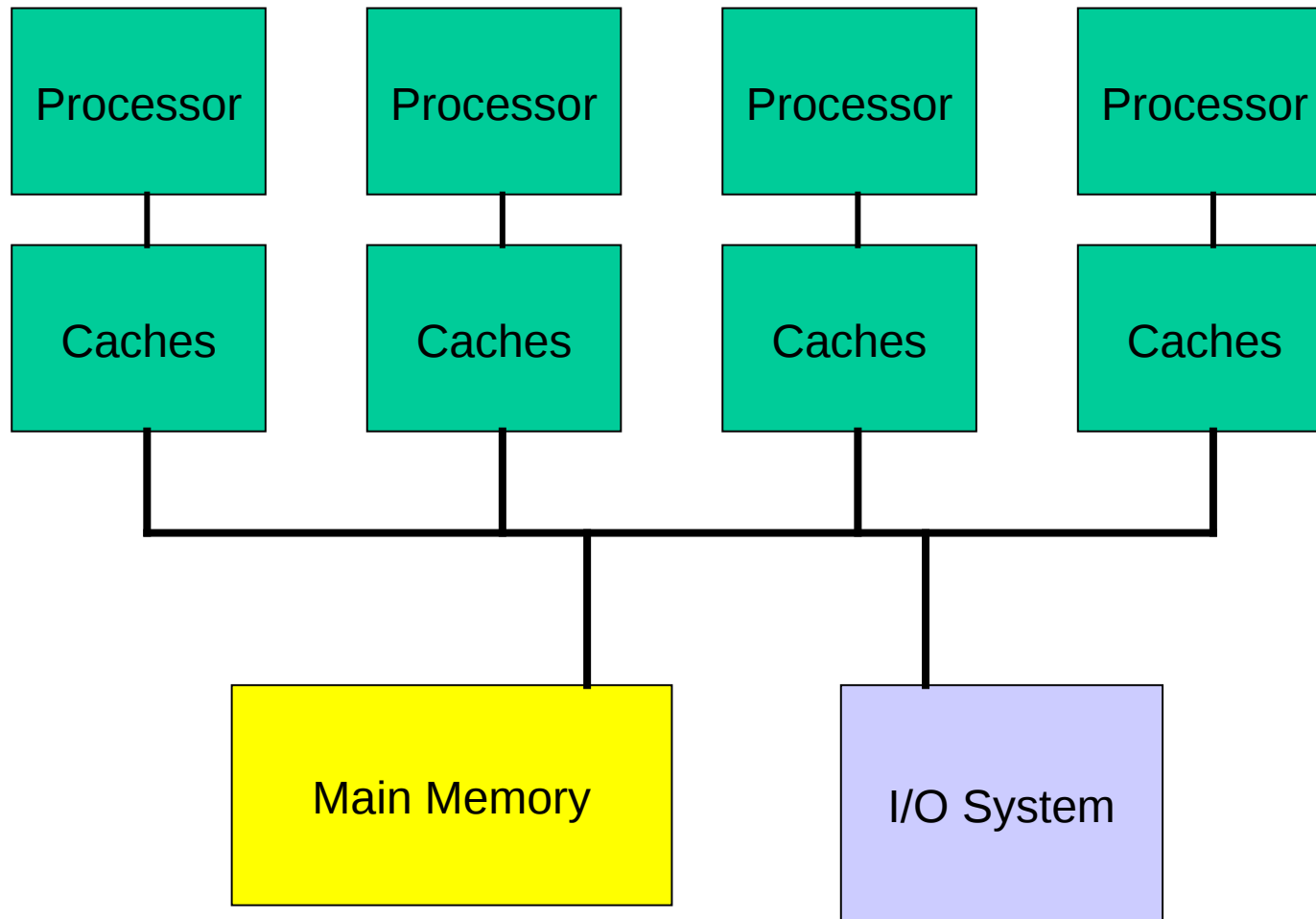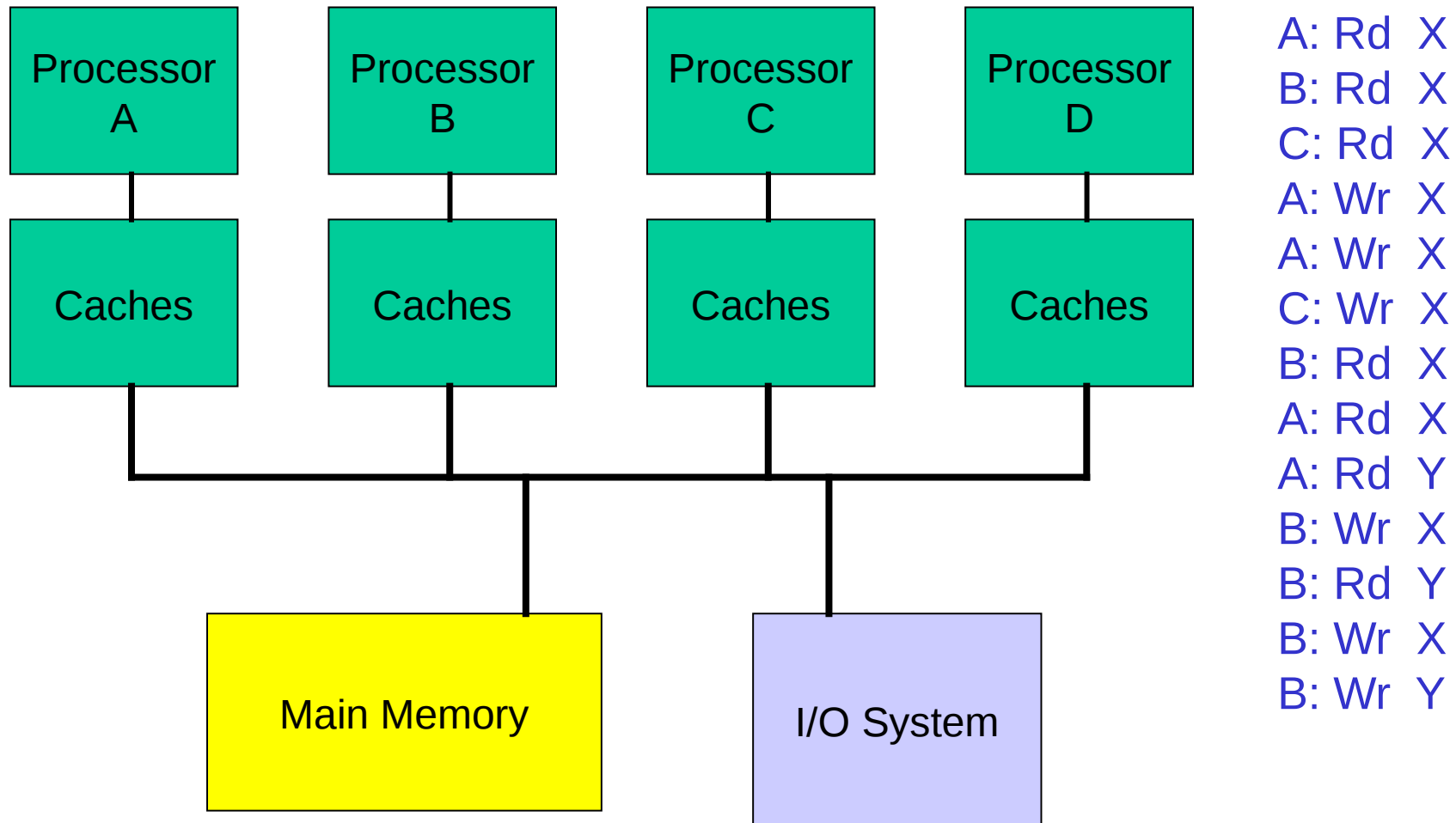
Design Issues

- Invalidate
- Find data
- Writeback / writethrough

- Cache block states
- Contention for tags
- Enforcing write serialization

SMP Example

Processor A — Caches
Processor B — Caches
Processor C — Caches
Processor D — Caches

Main Memory

I/O System

A: Rd X
B: Rd X
C: Rd X
A: Wr X
A: Wr X
C: Wr X
B: Rd X
A: Rd X
A: Rd Y
B: Wr X
B: Rd Y
B: Wr X
B: Wr Y

SMP Example

	A	B	C
A: Rd X			
B: Rd X			
C: Rd X			
A: Wr X			
A: Wr X			
C: Wr X			
B: Rd X			
A: Rd X			
A: Rd Y			
B: Wr X			
B: Rd Y			
B: Wr X			
B: Wr Y			

Example Protocol

| Request	Source	Block state	Action
Read hit	Proc	Shared/excl	Read data in cache
Read miss	Proc	Invalid	Place read miss on bus
Read miss	Proc	Shared	Conflict miss: place read miss on bus
Read miss	Proc	Exclusive	Conflict miss: write back block, place read miss on bus
Write hit	Proc	Exclusive	Write data in cache
Write hit	Proc	Shared	Place write miss on bus
Write miss	Proc	Invalid	Place write miss on bus
Write miss	Proc	Shared	Conflict miss: place write miss on bus
Write miss	Proc	Exclusive	Conflict miss: write back, place write miss on bus
Read miss	Bus	Shared	No action; allow memory to respond
Read miss	Bus	Exclusive	Place block on bus; change to shared
Write miss	Bus	Shared	Invalidate block
Write miss	Bus	Exclusive	Write back block; change to invalid [10]

Cache Coherence Protocols

- Directory-based: A single location (directory) keeps track of the sharing status of a block of memory

- Snooping: Every cache block is accompanied by the sharing status of that block – all cache controllers monitor the shared bus so they can update the sharing status of the block, if necessary

➢ Write-invalidate: a processor gains exclusive access of a block before writing by invalidating all other copies
➢ Write-update: when a processor writes, it updates other shared copies of that block

Directory-Based Cache Coherence

- The physical memory is distributed among all processors

- The directory is also distributed along with the corresponding memory

- The physical address is enough to determine the location of memory

- The (many) processing nodes are connected with a scalable interconnect (not a bus) – hence, messages are no longer broadcast, but routed from sender to receiver – since the processing nodes can no longer snoop, the directory keeps track of sharing state

Distributed Memory Multiprocessors
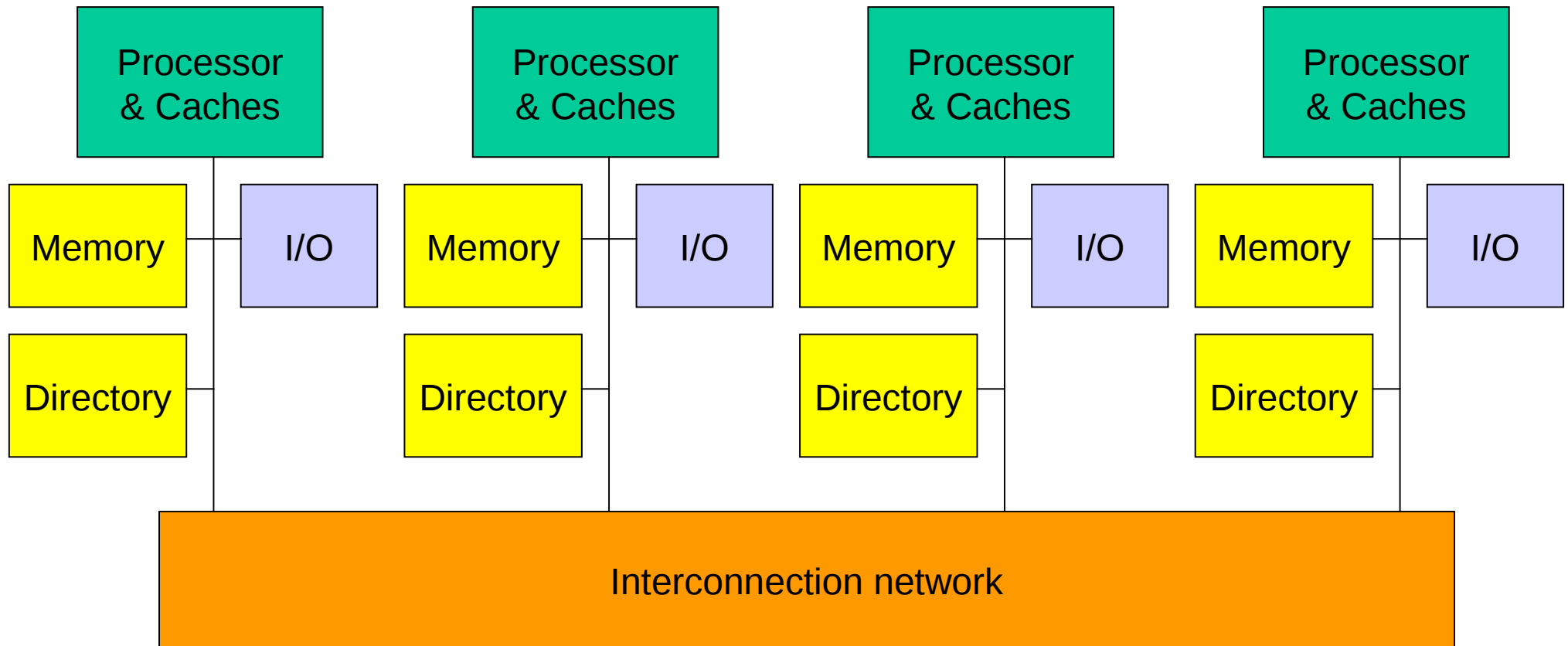
Directory Example

	A	B	C	Dir	Comments
A: Rd X					
B: Rd X					
C: Rd X					
A: Wr X					
A: Wr X					
C: Wr X					
B: Rd X					
A: Rd X					
A: Rd Y					
B: Wr X					
B: Rd Y					
B: Wr X					
B: Wr Y					

Cache Block States

- What are the different states a block of memory can have within the directory?

- Note that we need information for each cache so that invalidate messages can be sent

- The block state is also stored in the cache for efficiency

- The directory now serves as the arbitrator: if multiple write attempts happen simultaneously, the directory determines the ordering

Directory Actions

- **If block is in uncached state:**
  - ➤ Read miss: send data, make block shared
  - ➤ Write miss: send data, make block exclusive
- **If block is in shared state:**
  - ➤ Read miss: send data, add node to sharers list
  - ➤ Write miss: send data, invalidate sharers, make excl
- **If block is in exclusive state:**
  - ➤ Read miss: ask owner for data, write to memory, send data, make shared, add node to sharers list
  - ➤ Data write back: write to memory, make uncached
  - ➤ Write miss: ask owner for data, write to memory, send data, update identity of new owner, remain exclusive

Thank you!