Lecture: Coherence Protocols

- Topics: snooping-based protocols

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)

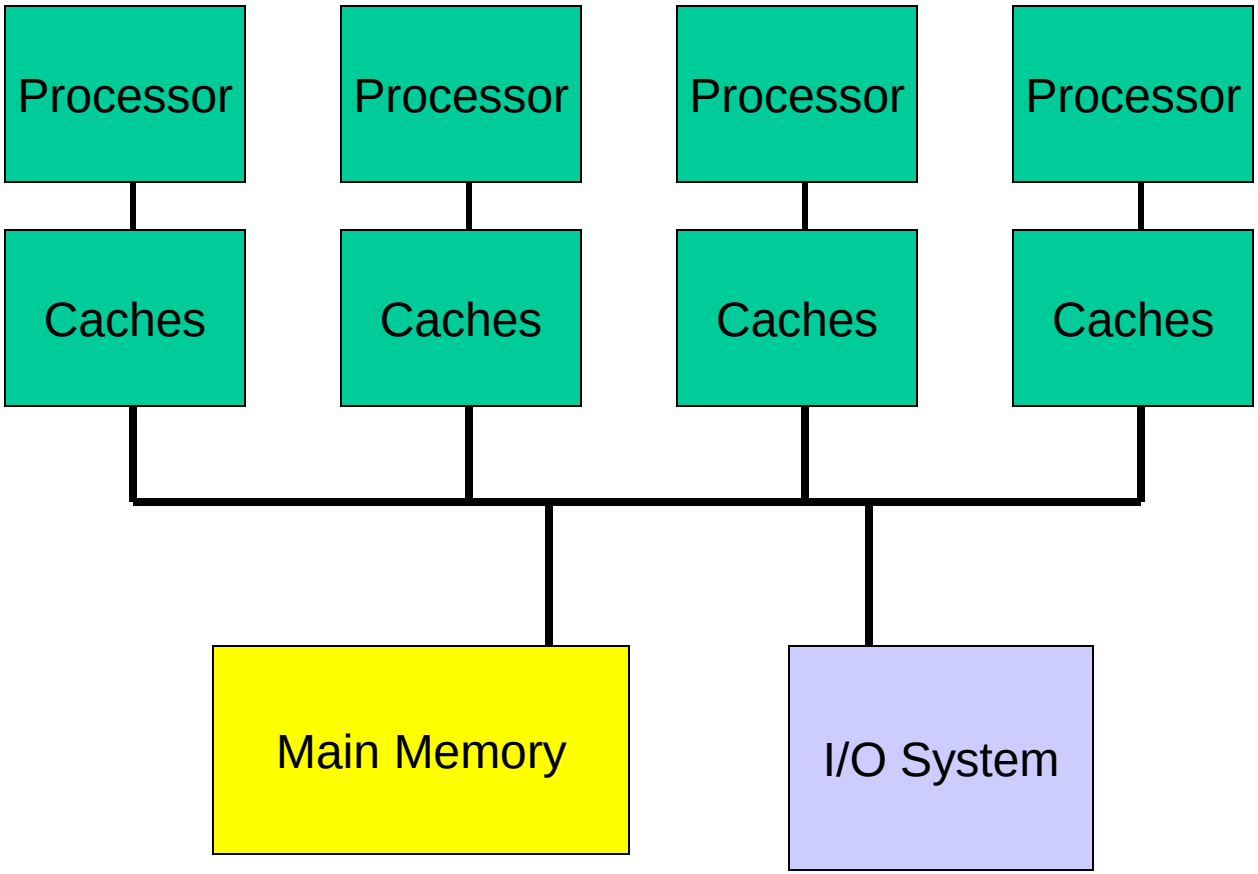
Cache Coherence Protocols

- 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

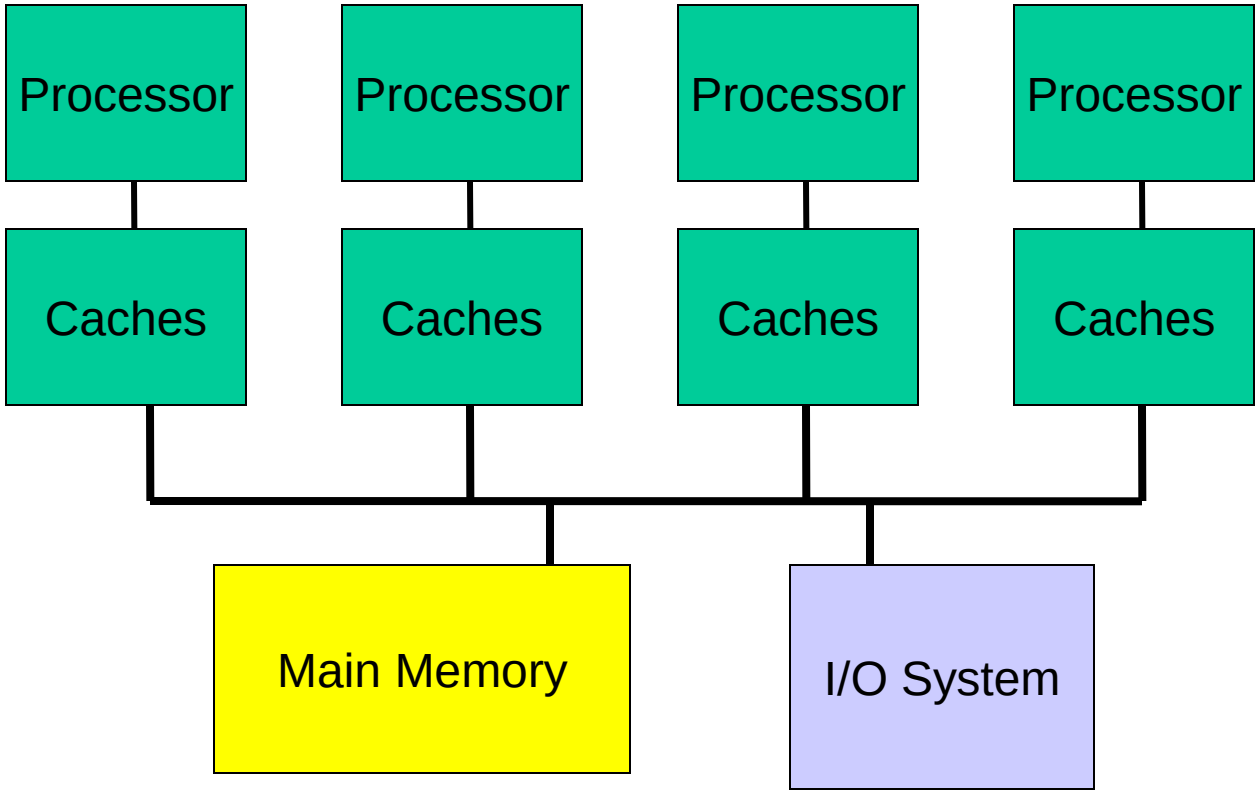
  - Write-invalidate: a processor gains exclusive access of a block before writing by invalidating all other copies
  - Write-update: when a processor writes, it updates other shared copies of that block

SMPs or Centralized Shared-Memory

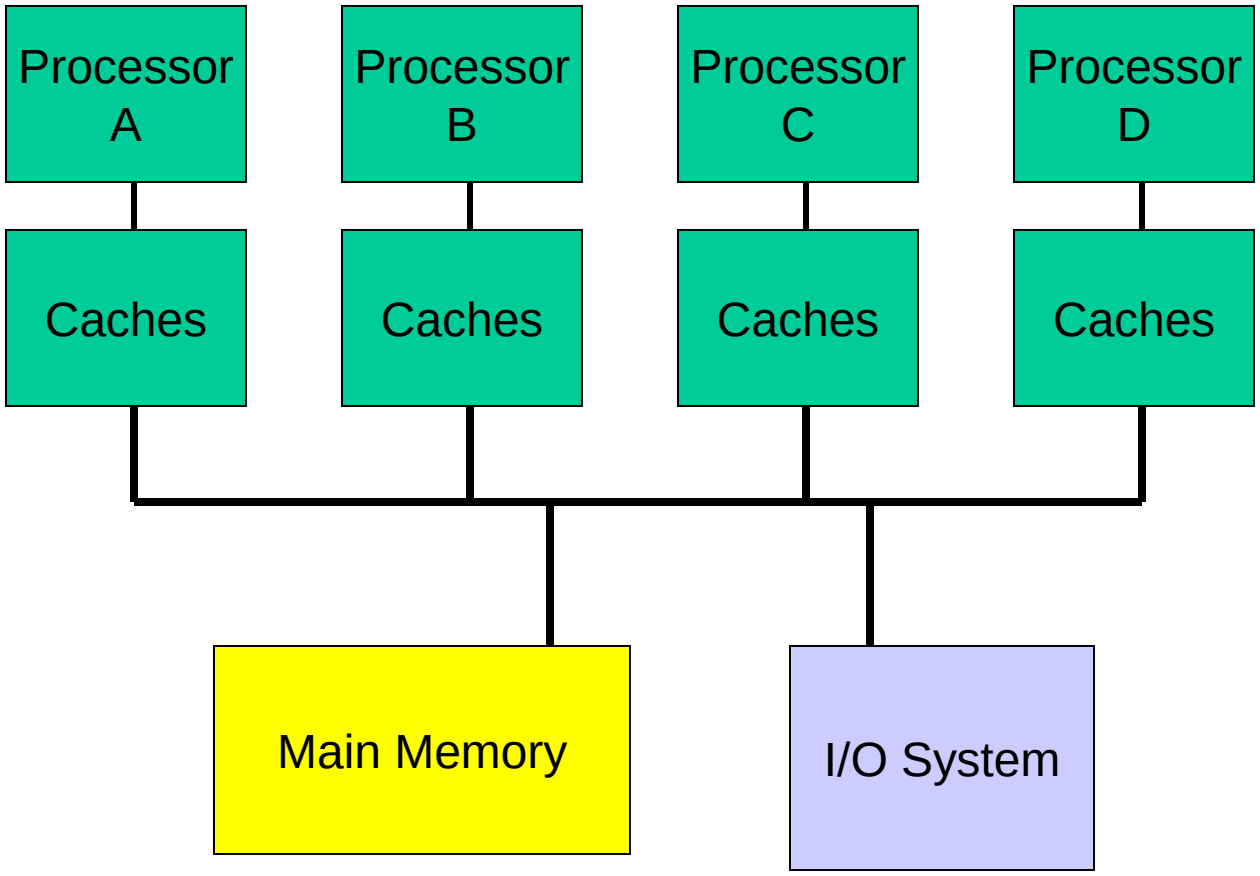


Design Issues

- Invalidate
- Find data
- Writeback / writethrough
- Cache block states
- Contention for tags
- Enforcing write serialization



SMP Example



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			

SMP Example

	A	B	C
A: Rd X	S		
B: Rd X	S	S	
C: Rd X	S	S	S
A: Wr X	M	I	I
A: Wr X	M	I	I
C: Wr X	I	I	M
B: Rd X	I	S	S
A: Rd X	S	S	S
A: Rd Y	S (Y)	S (X)	S (X)
B: Wr X	S (Y)	M (X)	I
B: Rd Y	S (Y)	S (Y)	I
B: Wr X	S (Y)	M (X)	I
B: Wr Y	I	M (Y)	I

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

Title

- Bullet