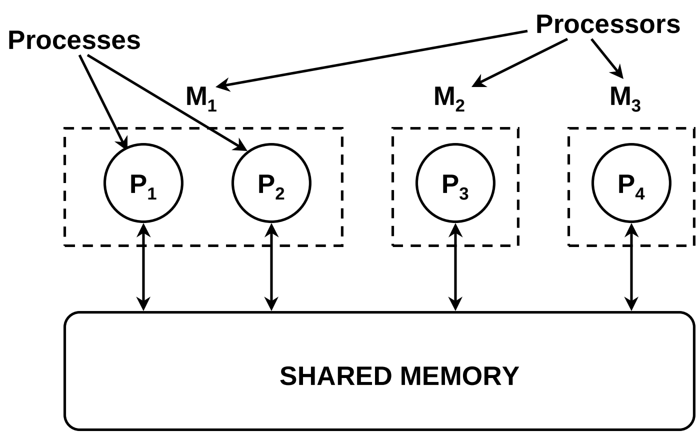
6004 Spring 1998: L47

Cache Coherence

1

"Shared Memory" Multiprocessors

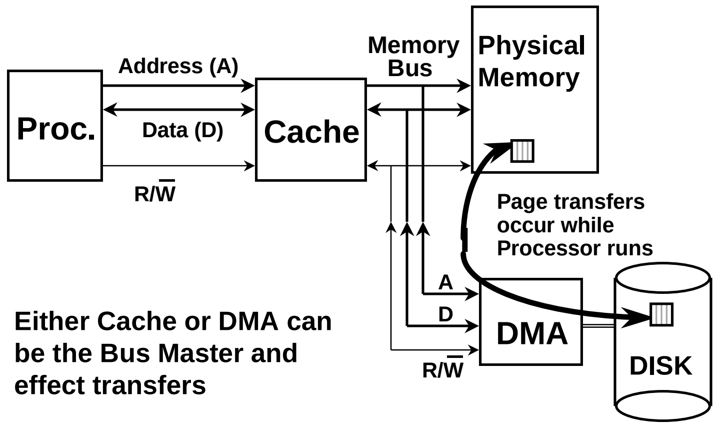


Map different processes onto different processors

2

6004 Spring 1998: L47

Warmup: Parallel I/O

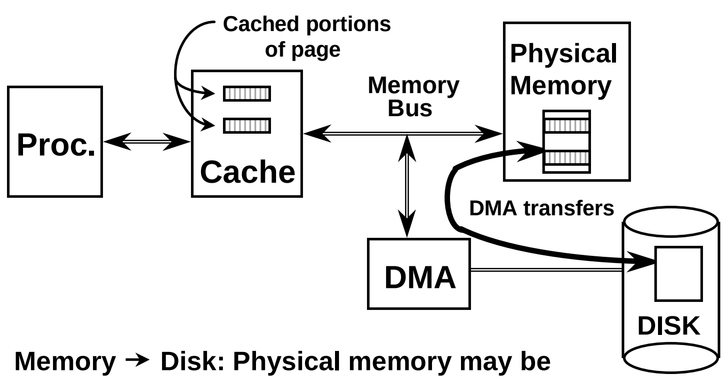


Either Cache or DMA can be the Bus Master and effect transfers

DMA stands for Direct Memory Access

3

Problems with Parallel I/O



Memory → Disk: Physical memory may be stale if Cache is ________

Disk → Memory: Cache may have data corresponding to _______

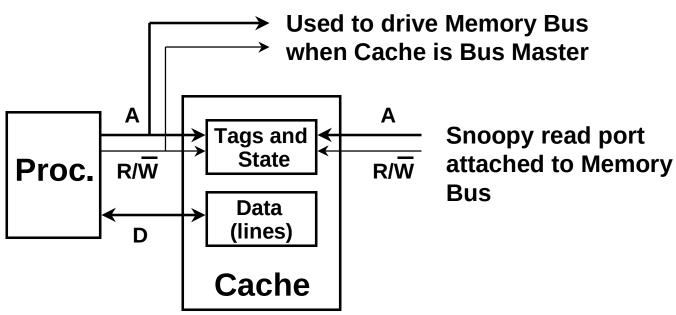
4

Solutions

- Explicitly flush cache (i.e., invalidate all cache entries) before or after Memory ↔ Disk transfers effected by DMA
- Provide new mechanism so that Processor's view of memory and DMA's view of memory are coherent

Snoopy Cache [*Goodman 1983*]

- Idea: Have cache watch (or snoop upon) DMA transfers, and then "do the right thing"
- Snoopy cache tags are dual-ported



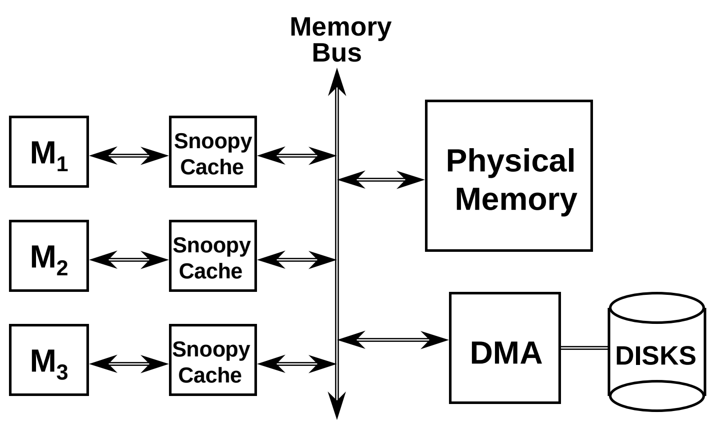
6004 Spring 1998: L47

Snoopy Cache Actions

Observed Bus Cycle	Cache State	Cache Action
Read Cycle Memory → Disk	Address not cached	______________
	Cached, unmodified	______________
	Cached, modified	______________
Write Cycle Disk → Memory	Address not cached	______________
	Cached, unmodified	______________
	Cached, modified	______________

7

Generalization to Several Processors



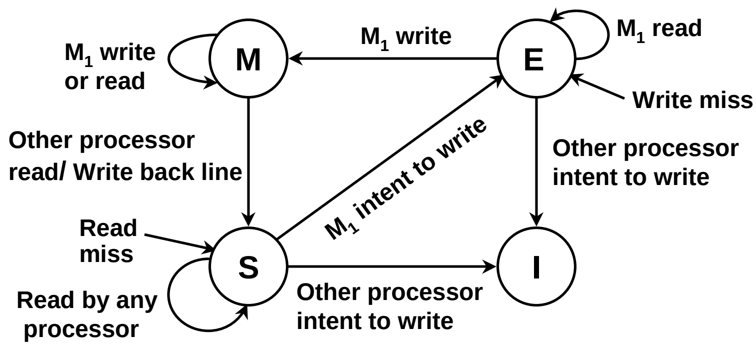
Use snoopy mechanism to keep all processors' view of memory coherent

8

Cache State Transition Diagram

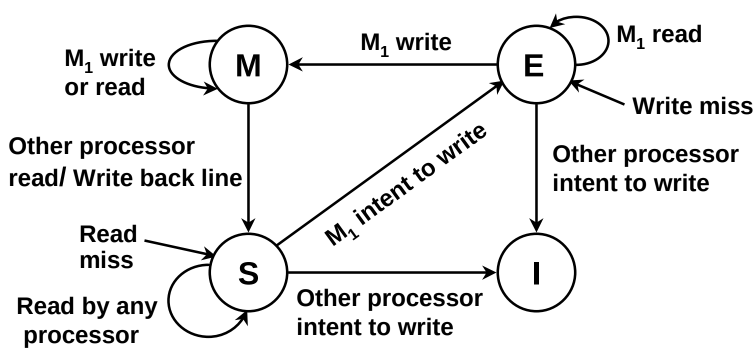
Each cache line tag

M: Modified Exclusive
E: Exclusive, unmodified
S: Shared
I: Invalid



9

Observation



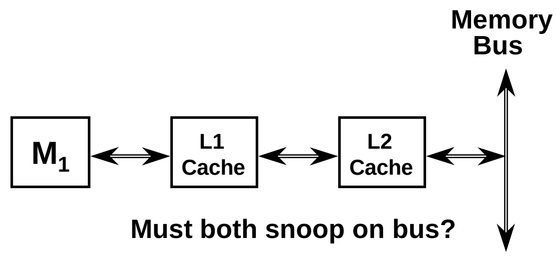
If a line is in M or E, then no other cache has a valid copy of the line!

– Memory stays coherent, multiple differing copies cannot exist

10

6004 Spring 1998: L47

Multilevel Caches



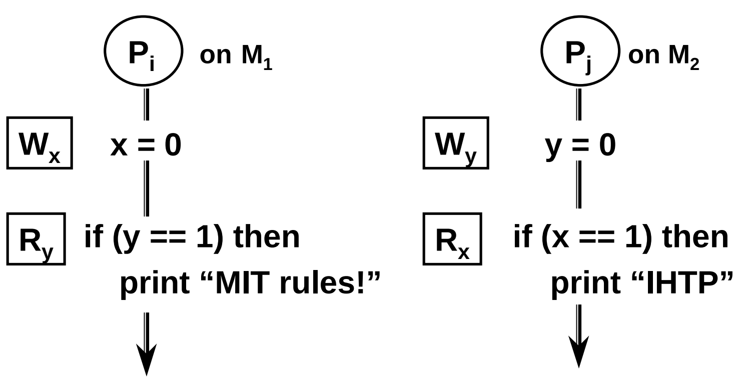
Can avoid if L2 <u>includes</u> all entries of L1

If L2 is an inclusion cache:
L2 snoops; if address matches, then check L1

11

Multiple Processes and Processors

x = y = 1



12

Interleaving

Different permutations

P_i	P_j	Print
– W_x R_y	W_y R_x	"MIT rules!"

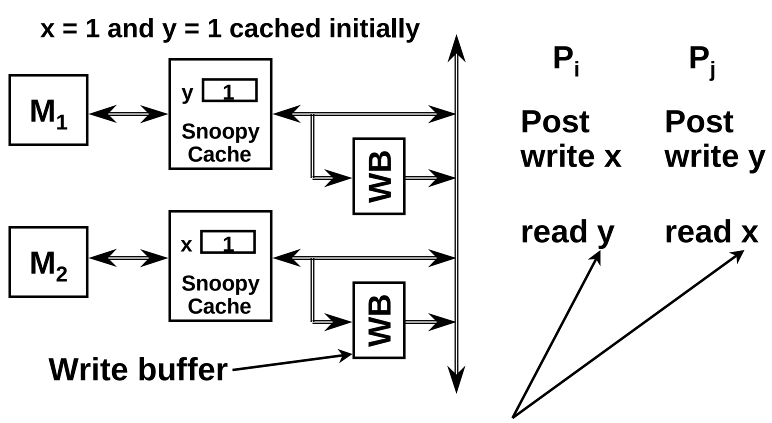
P_j	P_i	
– W_y R_x	W_x R_y	"IHTP"

– W_x W_y ⟷	R_x R_y ⟷	nothing

Any way that both messages are printed?

13

Write Buffers



Because of write buffer delay, cache entries are not invalidated. Both hit respective caches and return 1.

14

Sequential Consistency

- **In addition to coherence we must provide "sequential consistency"**
- **[Lamport 1979]: "The result of any execution is the same as if the operations of all processors were in some sequential order, and the operations of each individual processor appear in this sequence in the order specified by its program".**
  - **Any parallel execution must correspond to some interleaved execution on a single processor**
- **Implementing consistency**
  - **Limit use of write buffers**
  - **Use semaphores to lock variables so they cannot be read until writes complete**

15