2008 YEAR

PRESENTATION

The University of Auckland | New Zealand

Computer Science 703

Advance Computer Architecture

2008 Semester 1

Lecture Notes 4

12Mar08

Multiprocessing: Cache Coherence

James Goodman

Mark Hill: Multiprocessors Should Support Simple Memory-Consistency Models

- Written in 1998, but made predictions for the next ten years.
- Claim: Relaxing consistency buys some additional performance, but not enough for the aggravation

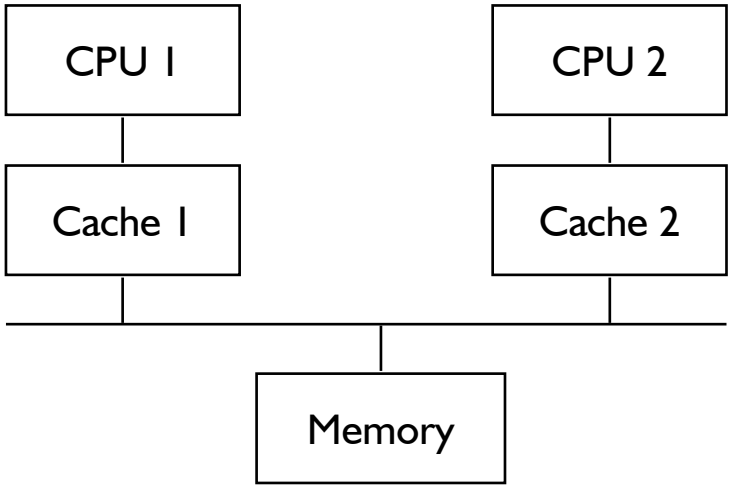
Questions in 2008

- Was Hill correct about performance and aggravation? If so, why hasn't the world changed to SC?
- In 1998, few foresaw Chip multiprocessors. Did this affect the arguments?
- Given trends toward transactional memory, does the memory model much matter?

Multiprocessing Issues

- Memory Consistency
- "Missing Update Problem"
- **Cache Coherence**

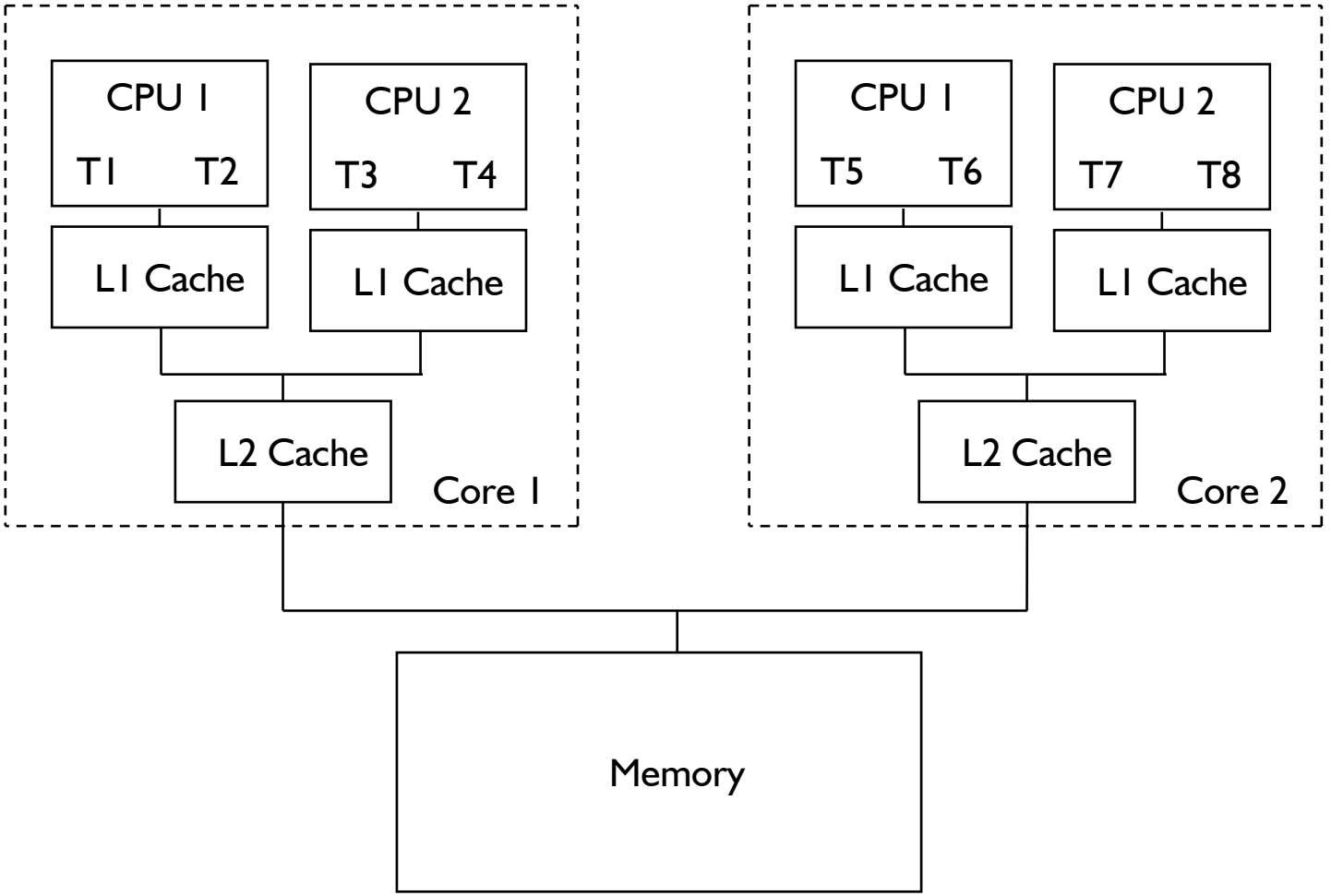
Basic MESI Protocol



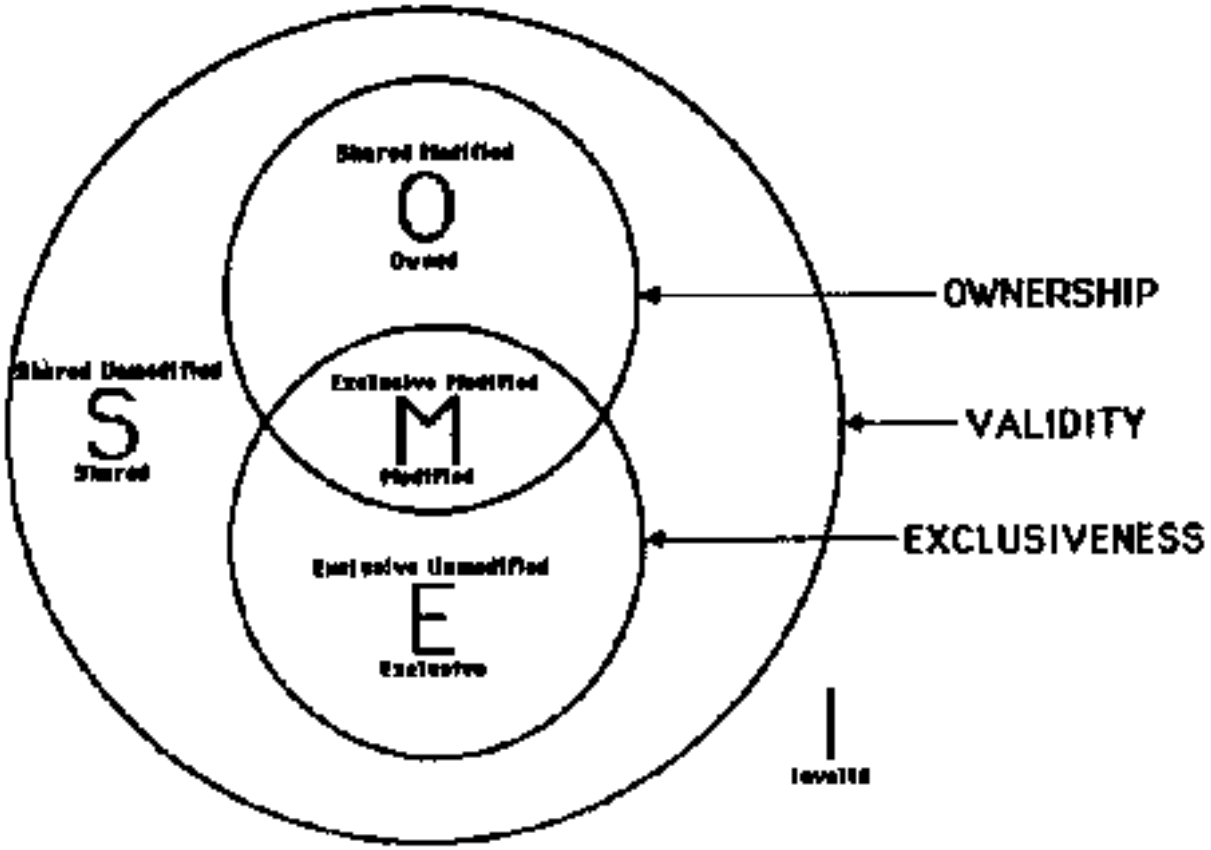
Cache blocks may be in one of four possible states

- *Modified*: copy in cache is different than copy in main memory, which is stale
- *Exclusive*: copy in cache can be modified without external permission, but is the same as main memory.
- *Shared*: copy in cache is valid for reading, but may not be modified without eliminating other potential copies
- *Invalid*: data in cache is stale

Multilevel Cache Coherence



The MOESI States



MOESI Rules

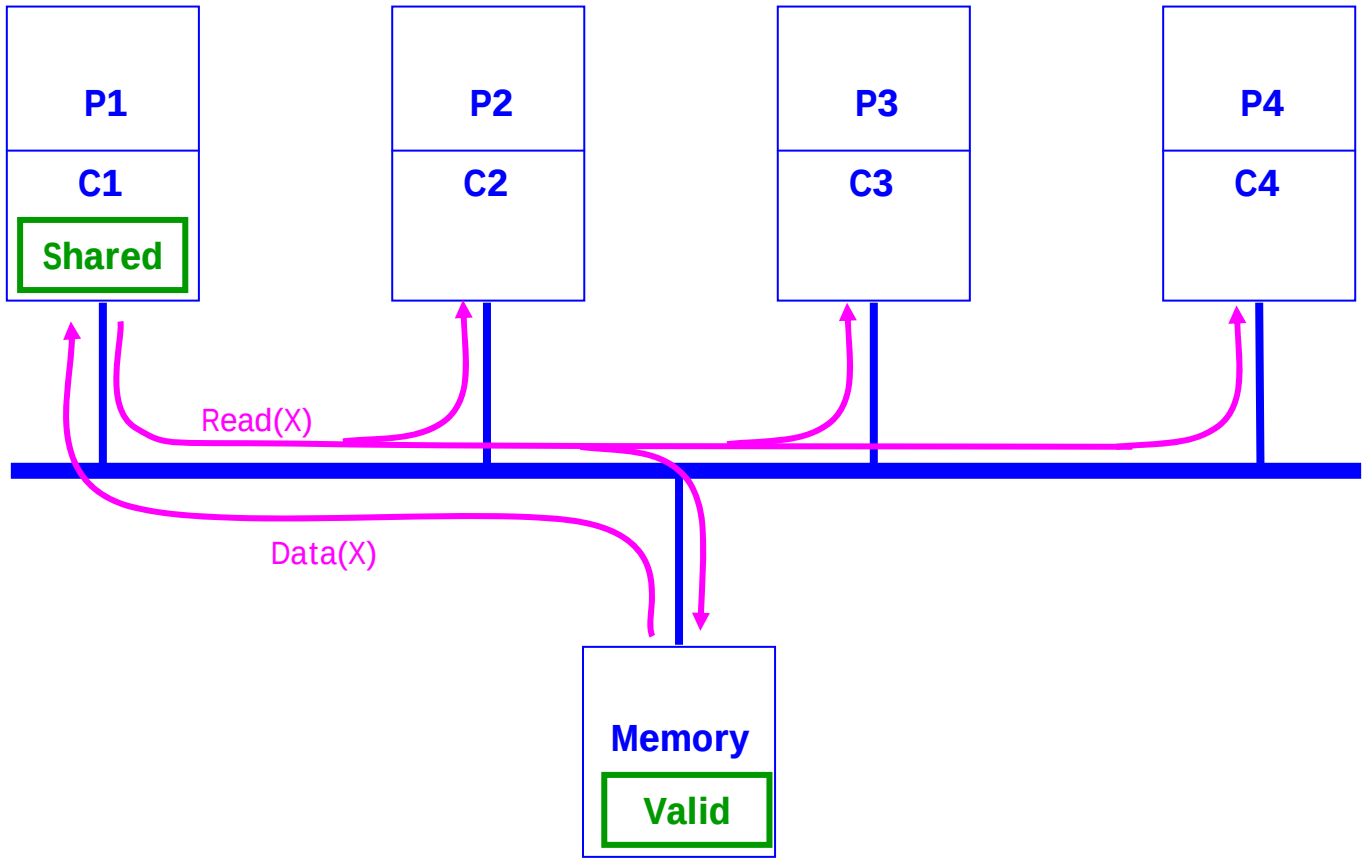
- Ownership -> memory is not valid
- Modified, Shared-Modified must respond to request, update memory
- All other changes, must notify
- Can change E->M silently
- Can change E->I silently
- Can change S->I silently

Permitted States for MOESI

- Multiple S, others I
- 1 E, others I
- 1 M, others I
- 1 SM, multiple S, others I

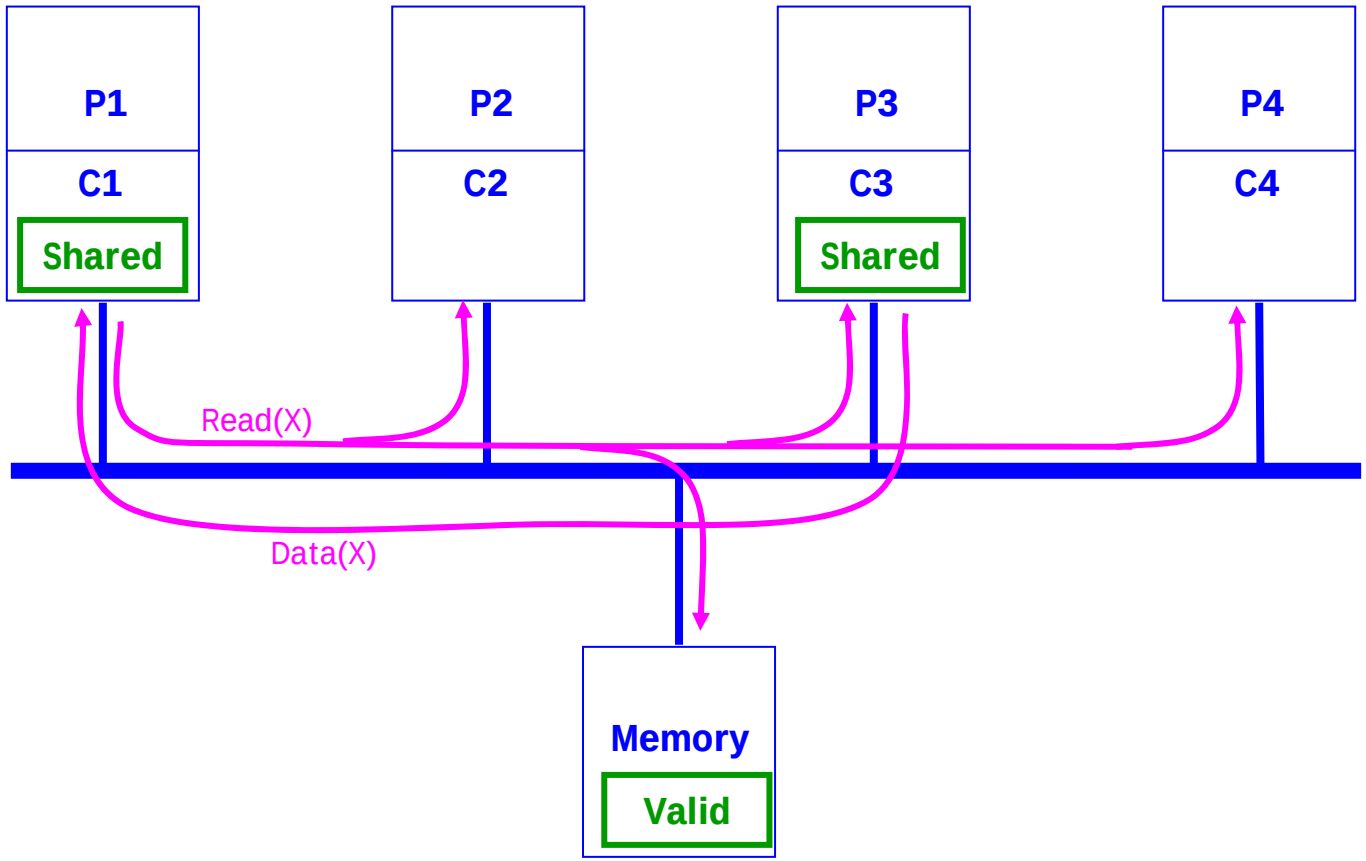
Cache Coherence

Read to Unshared



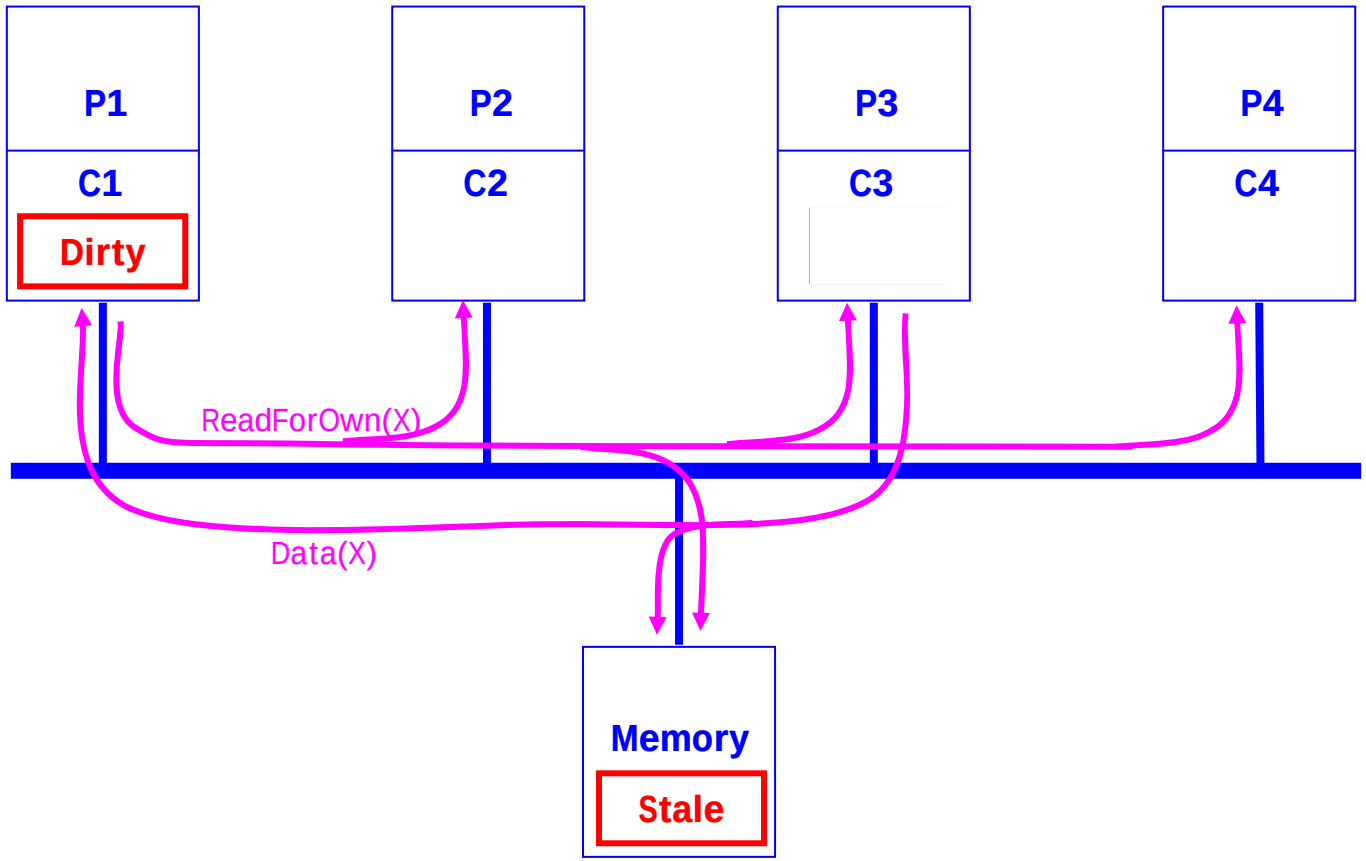
Cache Coherence

Read to Shared



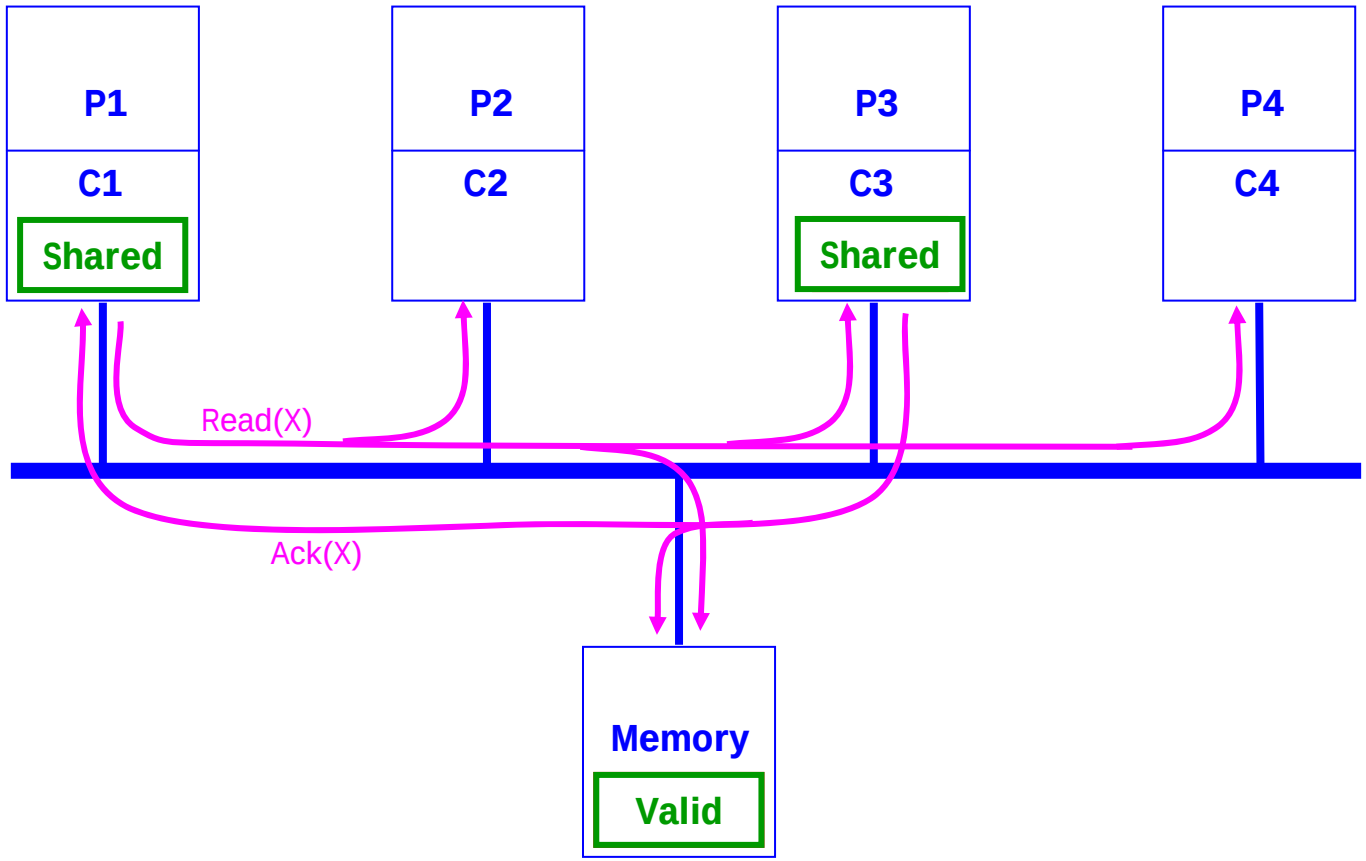
Cache Coherence

Write to Dirty



Cache Coherence

Read to Dirty

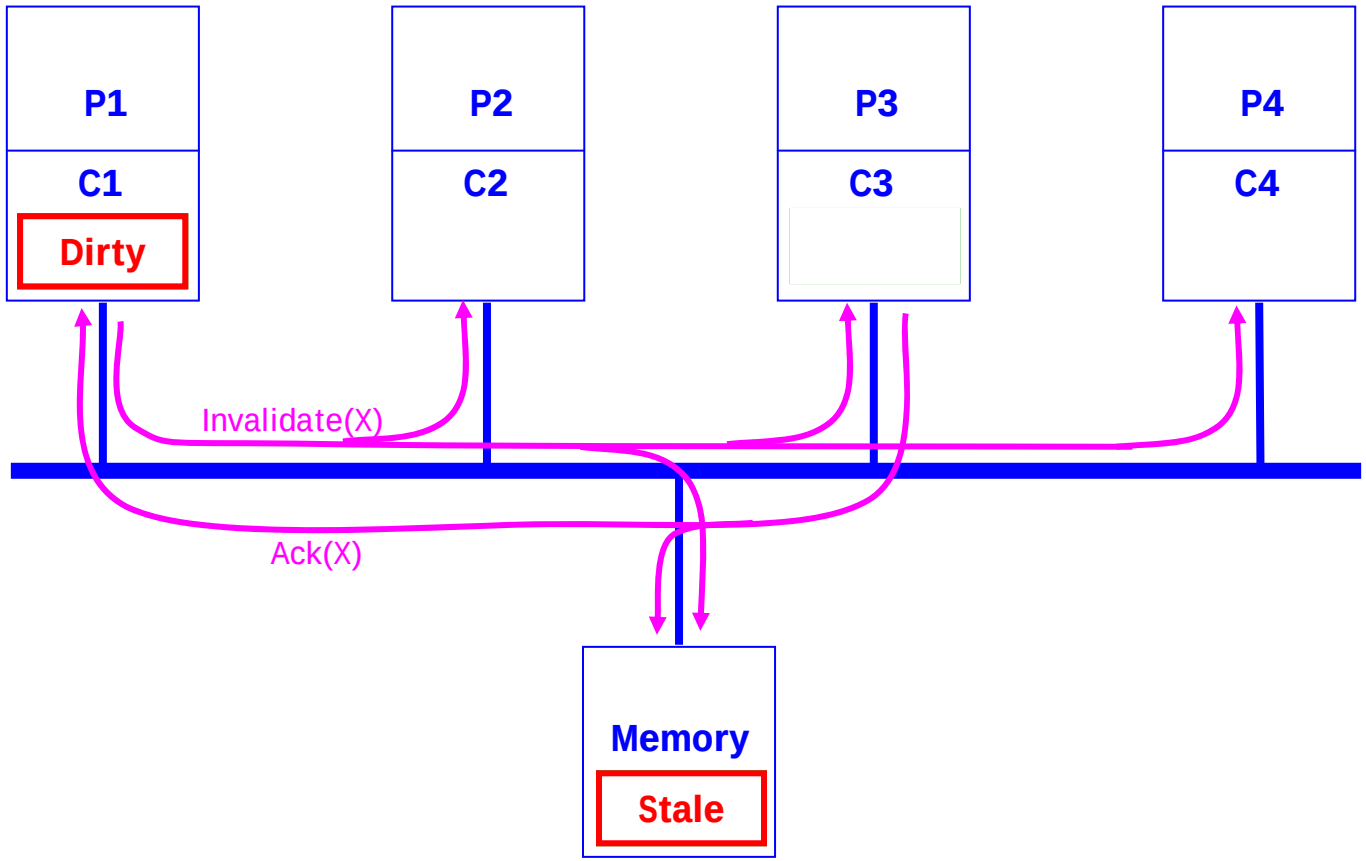


Another possibility?

- Go from M/I to S/S
- Go from M/I to I/MM

Cache Coherence

Write to Shared



Summary of Snooping Caches

- Snooping cache coherence protocols are the *dominant multiprocessor technique* used today
- Most microprocessors conform to snooping cache protocols (e.g., Intel Pentium: up to 4 processors on the bus)
- Snooping has been extended to much larger systems by a series of creative methods, but scalability is fundamentally limited by broadcast requirements

Coherency in Multiple-Bus Systems

- Scalable protocols involve maintaining a directory auxiliary information keeping track of which caches have copies of which cache lines
- Directory-based scheme can use point-to-point connections, which potentially have both higher speed and much higher bandwidth than a bus
- Because of the additional delay (typically three hops for most transactions), only very large systems benefit from directory-based schemes.
- There have been several commercial products, but so far, no directory-based scheme has been highly successful.

Terminology

- Snooping-based schemes have been extended beyond a single bus (maintaining the notion of a single "logical" bus). Broadcast-based schemes are called *Symmetric MultiProcessing* (SMP)
- Directory-based schemes continue to be widely studied, and there are many variations proposed and some products. Such schemes are often referred to as *Non-Uniform Memory Access* (NUMA) systems.