

CS422

Computer Architecture

Spring 2004

Lecture 25, 30 Mar 2004

Bhaskaran Raman
Department of CSE
IIT Kanpur

<http://web.cse.iitk.ac.in/~cs422/index.html>

Topic for Today's Lecture

- Cache coherence protocols
-
- *Scribe for today?*

Cache Coherence

- In what kind of multi-processors do we need cache coherence?
- What are the kinds of data which are cached?
 - Shared (read) data – replication
 - Private data – migration

Notions of Coherence and Consistency

- Coherence:
 - Program order preservation within a processor
 - Write by P1, Read by P2, if “sufficiently separated”, should get the value of write
 - Write serialization: same order of writes seen by all processors
- Specifying when a read should get the value of a write: *memory consistency model*

Styles of Coherence Protocols

- **Directory-based:** central directory maintains the “status” of each block
- **Snooping-based:**
 - In a centralized shared memory machine
 - Each processor snoops on the common bus
 - Also maintains the “status” of a block locally (no central directory)
 - Snooping helps maintain coherence

Styles of Snooping Protocols

- **Write-invalidate:** processor makes sure that it has the only copy of a block before writing
 - Invalidates other copies by sending an invalidate command on the bus
- **Write-update or write-broadcast:** processor updates all copies of a block when it writes
 - Send the written data on the common bus

Write-invalidate vs. Write-update

Write-invalidate

Consecutive writes to a location does not cause repeated traffic on bus

Writes to consecutive locations does not cause extra traffic on bus

Write-update

Writes appear for readers with lesser latency