

2010 Ear Zealand The University of Auckland | New

References

Sweazey & Smith, "A class of compatible cache consistency protocols and their support by the IEEE FUTUREBUS", *ISCA-13*, 1986.

M. Herlihy and J.E.B. Moss, "Transactional memory: architectural support for lock-free data structures," *ISCA-20*, pp. 289–300, May 1993.

Snooping Cache

Observation: Don't have to notify others on every write Only notify when changing from clean to dirty

- Main memory is passive; others must remember if stale
- Basic idea:

±2010

PRESENTATION

The University of Auckland I New

- cache data permitted, but must take responsibility for inconsistencies
- Writers must notify all possible caches to eliminate/update copies
- Readers must update/invalidate on notification (must snoop)

MESI States

- Invalid
 - Data is not present in cache
 - Space may or may not be allocated
- Shared
 - Copy is in cache; can read but not write
 - Not guaranteed to be exclusive copy
- Exclusive
 - Copy is in cache; can read or write
 - No other copies except main memory
 - On remote read, must change to Shared
 - On local write, must change to Modified
 - Can silently discard
- Modified
 - Copy is in cache; can read or write
 - Must write back on purge
 - On remote read or write, must respond by writing, change to Shared

Permitted States for MESI

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

Many variations

- Eliminate Exclusive mode (optimization)
- Assume write privilege if not shared
- Read with Intention to modify (RFO)
 - Can we guess intention?
- Ownership: Multiple copies in caches, main memory stale
- Write update (Shared-Modified)
- Reflection: updating memory on transfer of Modified data
- Snarfing: intercepting data recently invalidated

Many Published Protocols

- Synapse
 - Frank, "Tightly coupled multiprocessor system speeds memory-access times," Electronics, 1984
- Write-once
 - Goodman, "Using cache memory to reduce processor-memory traffic," ISCA-10, 1983.
- Dragon (Xerox PARC)
 - McCreight, "The Dragon computer system: An early overview," Technical Report, Xerox Corp., 1984.
- Firefly (DEC)
 - Thacker, Stewart & Satterthwaite, "Firefly: A multiprocessor workstation," *IEEE Trans.* on Computers, 37(8), Aug. 1988.
- (Unnamed)
 - Rudolph &Segall, "Dynamic decentralized cache schemes for MIMD parallel processors," ISCA-11, 1984.
- Illinois
 - Papamarcos & Patel, "A low overhead coherence solution for multiprocessors with private cache memories," ISCA-11, 1984.
- Berkeley
 - Katz, Eggers, Wood, Perkins & Sheldon, "Implementing a cache consistency protocol," ISCA-12, 1985.

The Generalized MOESI Model

Sweazey & Smith, "A class of compatible cache consistency protocols and their support by the IEEE futurebus", *ISCA-13*, 1986.

- Three attributes:
 - Ownership
 - Exclusiveness
 - Validity
- Five states:
 - Invalid ()
 - Shared (Valid)
 - Exclusive (Valid, Exclusive)
 - Shared-Modified[originally Owned] (Valid, Ownership)
 - Modified (Valid, Ownership, Exclusive)

147

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 M, others I

Cache Coherence

Read to Unshared



151

Cache Coherence

Read to Shared



Cache Coherence

Read to Dirty



Cache Coherence

Write to Shared



Cache Coherence

Write to Dirty



156

158