Washington University in St. Louis

CSE/ESE 569M

Parallel Architectures and Algorithms

Cache Coherence Protocol

Tian Song

Feb.19, 2013

About Today

- Coherence Protocol: illinois

- Coherence Protocol: MOESI

- Coherence Protocol: dragon

Review - MSI

***Write back invalidation* protocol: MSI**

- We have **three** cache line **states** in MSI state diagram

- I (Invalid)

- S (Shared): line valid, and dirty bit is not set (same to memory)

- M (Modified): line valid, but dirty bit is set. (exclusive)

For processors / cores, the two operations are:

(1) ***p-load***: read a value from memory system

(2) ***p-store***: write a value to memory system

Review - MSI

Write back invalidation protocol: MSI

For snooping the bus, there are three operations:

(1) ***bus-read***: read a value from memory

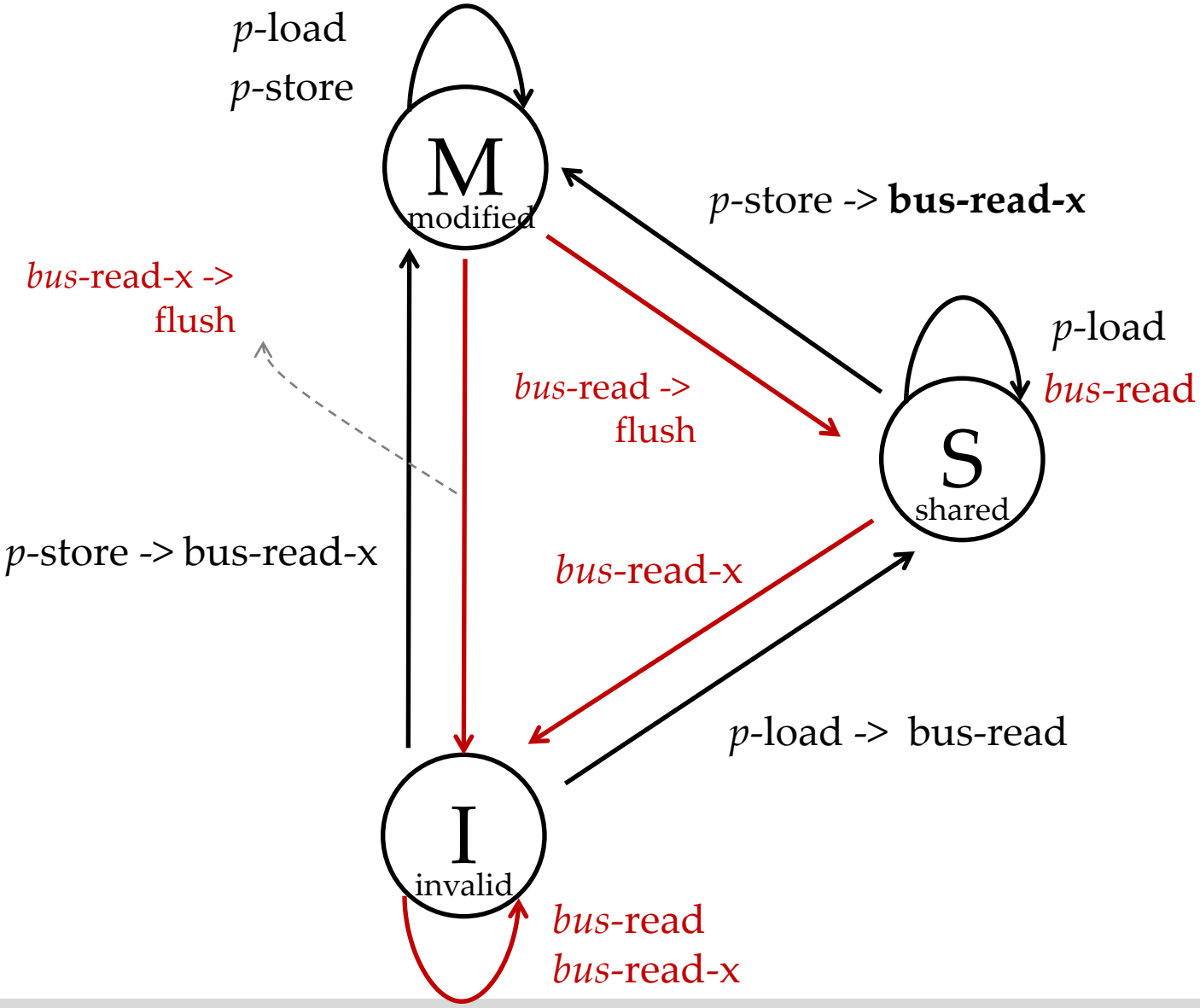
(2) ***bus-read-x***: bus exclusive read, means:

i) inform others that some memory location will be written in a cache

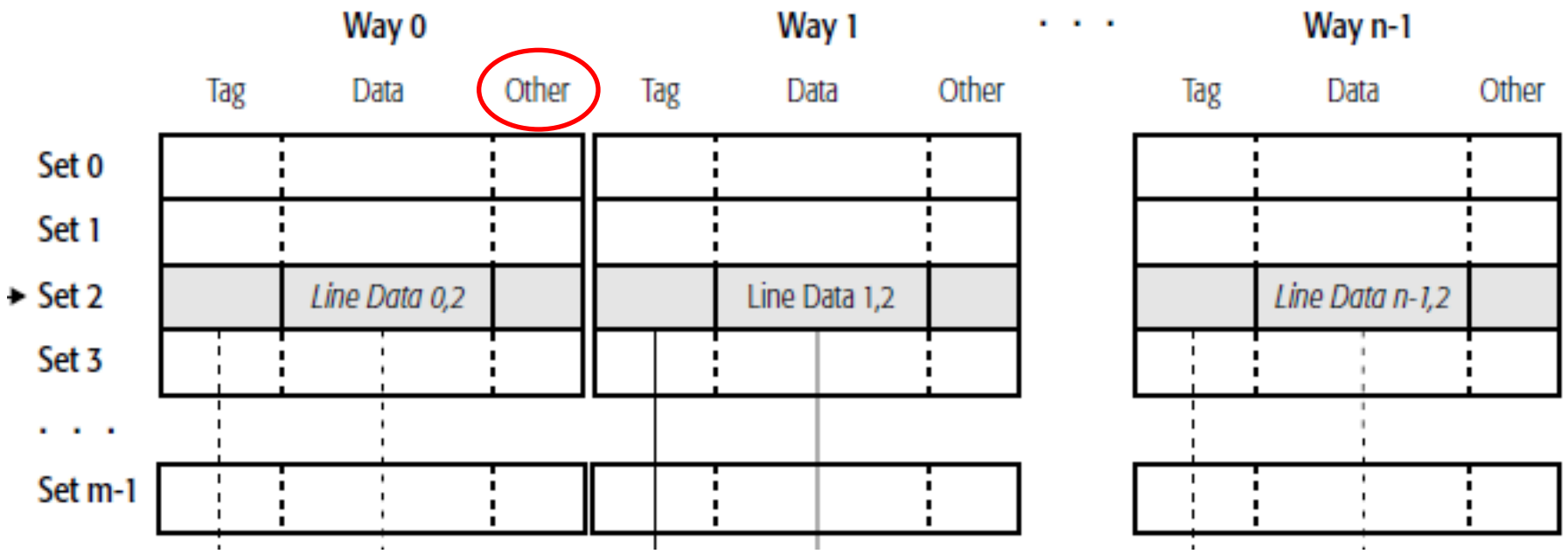
ii) write a value to the cache

(3) ***flush***: write a cache line to the memory

Write back invalidation protocol: MSI state diagram



AMD64 Cache line



MESI

Write back invalidation protocol

- MESI, a.k.a. **Illinois protocol**

- E: (Exclusive) cache line is present only in the current cache, but is clean

Ref: M. S. Papamarcos and J. H. Patel, "*A low-overhead coherence solution for multiprocessors with private cache memories*," Proc. 11th Annual Int. Symp. on Computer Architecture (*ISCA*), pp. 348-354, June 1984.

MESI

Details

- M: Modified, holds the most recent copy (the copy in memory is stale)

- E: Exclusive, exclusive hold the recent copy (copy in memory is valid).

- S: Shared, share the most recent copy.

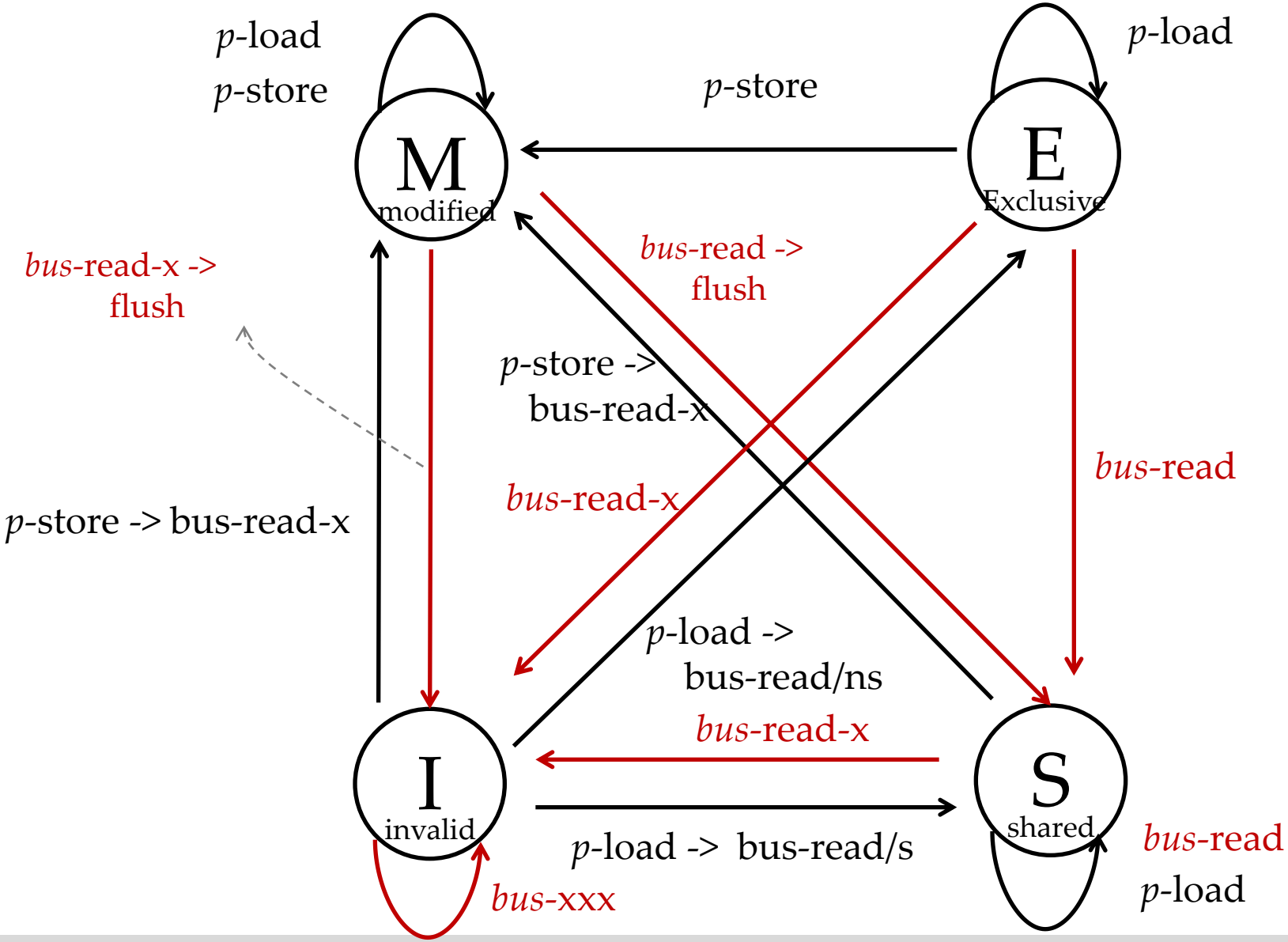
- I: Invalid, a cache line does not hold a valid copy of data.

To distinguish "exclusive", we need a respond operation of all caches

- for each bus-read, other caches should **respond** if it has the location.

- bus-read will have two forms: ***bus-read/s***, ***bus-read/ns***

Write back invalidation protocol: MESI state diagram



MESI

Principles

- Every state with a ***p-load*** remains the same stats except invalid state.

- A ***p-store*** operation will lead to modified state.

- Snooping a ***bus-read-x*** will lead to invalid state.

- Snooping a ***bus-read*** will lead to shared state.

- Leaving modified state will trigger a ***flush*** operation.

Coherence Protocol

What is coherence protocol?

- A coherence protocol is a protocol which maintains the ***consistency*** between all the caches in a shared memory system.

- **Invalidation based protocol**

  MSI, MESI (illinois), MESIF, MOESI, ….

- **Update based protocol**

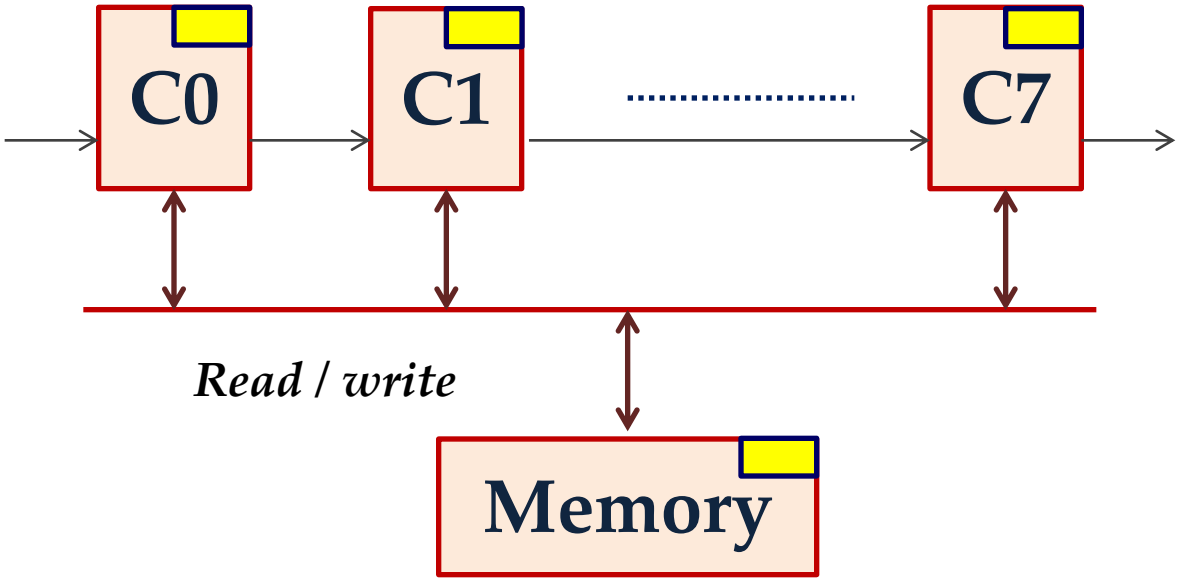
  Dragon protocol, …

MOESI

An important assumption:

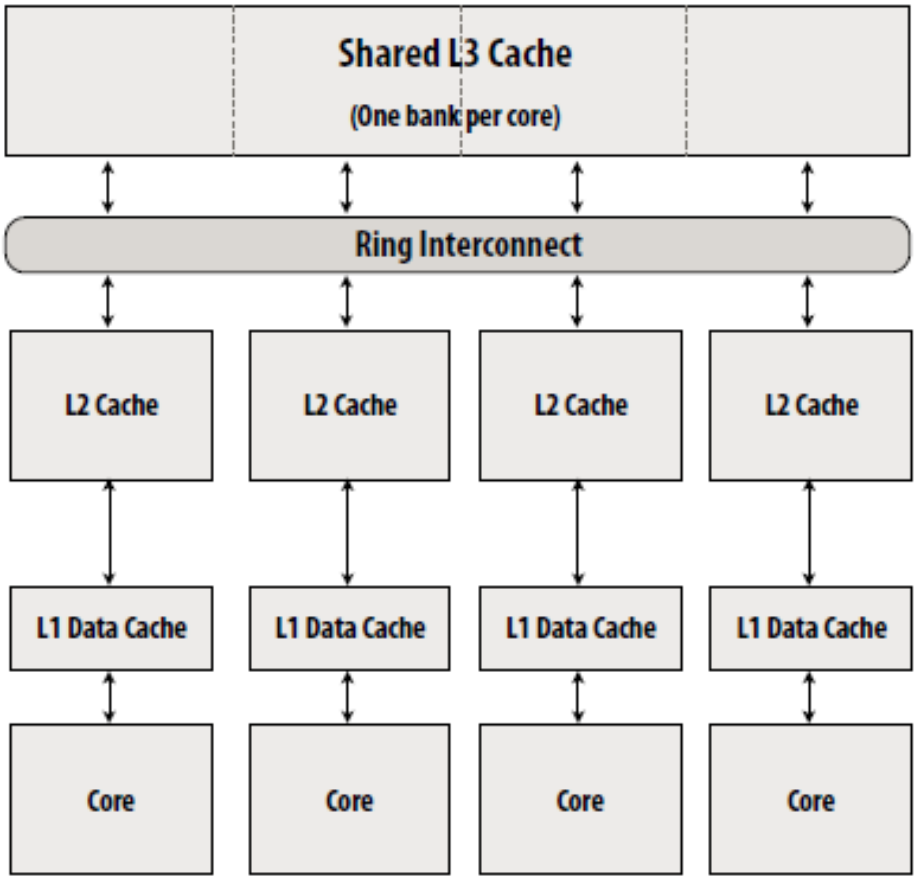
- Cache to cache communications are possible.

- On-chip communications vs. bus communications.



MOESI

Cache-to-cache Comm in Intel i7.



MOESI

AMD64's coherence protocol

- MESI + a state "O"

- M: Modified, holds the most recent copy (the copy in memory is stale)

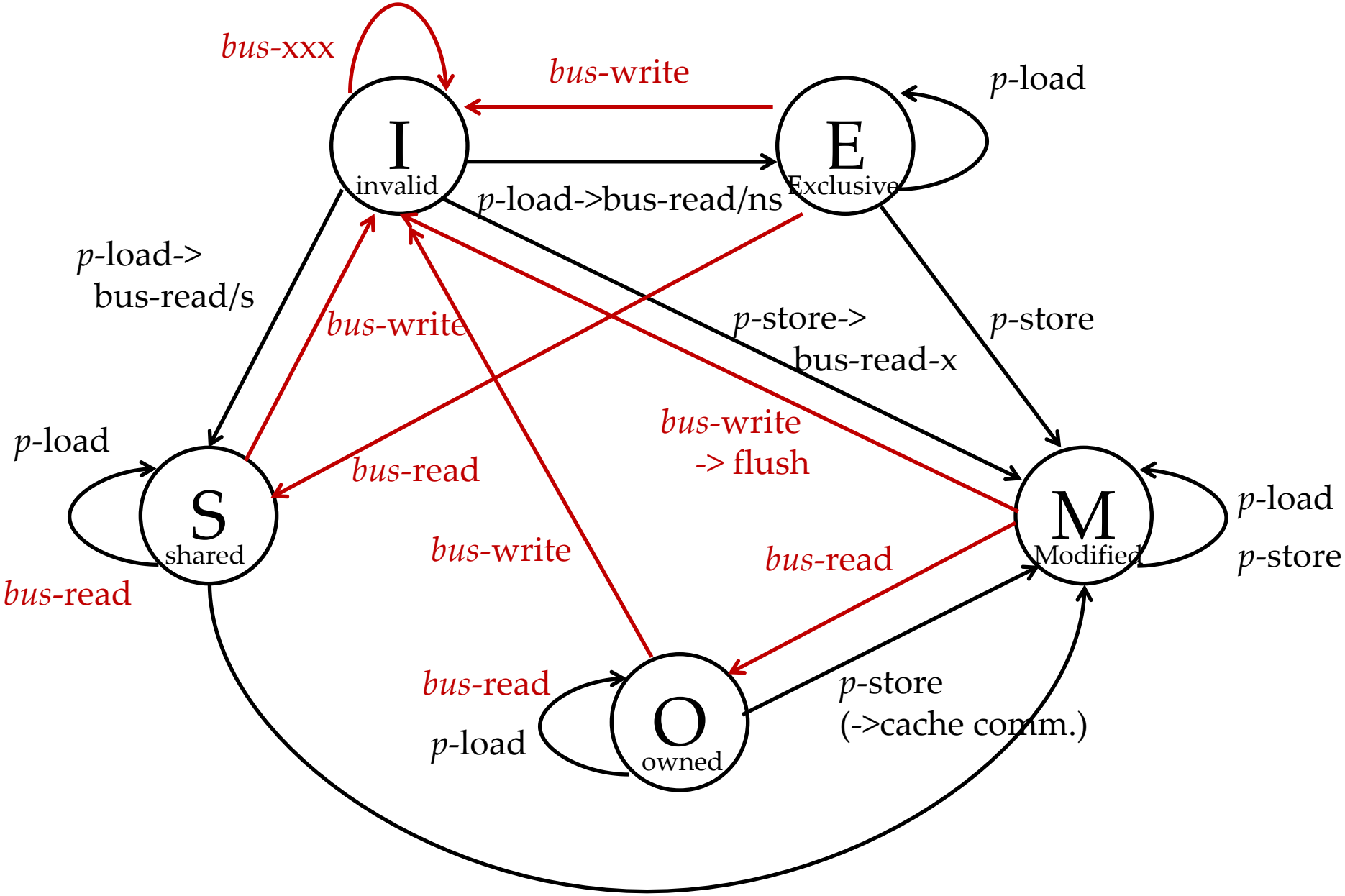
- O: Owned, ownership while allowing dirty sharing of data.

- E: Exclusive, exclusive hold the most recent copy.

- S: Shared, share the most recent copy.

- I: Invalid, a cache line does not hold a valid copy of data.

Write back invalidation protocol: MOESI state diagram



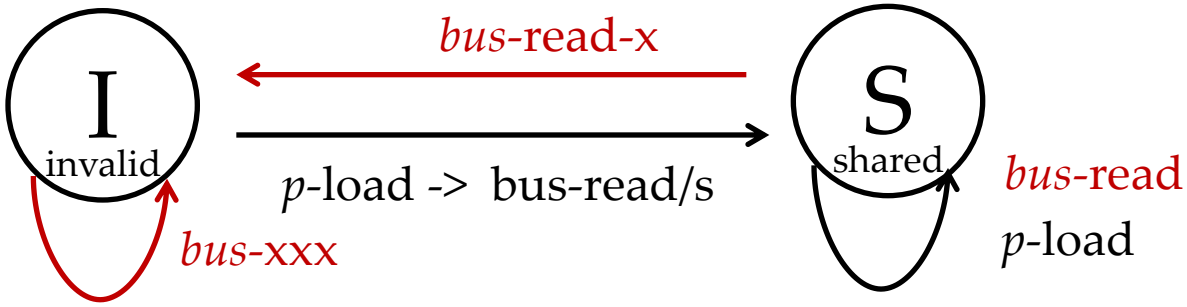
p-store->bus-read-x

MESIF

Intel's coherence protocol

- MESI + a state "F"

- F: Forward, indicate a responder for any request of the given cache line.



Reading assignment

MESIF is useful to reduce response by using cache-to-cache comm.

Dragon Protocol

Update-based coherence protocol

- developed by Xerox PARC (……)

If in cache, cannot be valid.

- Idea: update cache lines and make them valid all the time

- Dragon protocol also uses write-back cache.

  - E: Exclusive, only one cache has line, memory is up-to-date.

  - Sc: Shared-clean, multiple cache lines, memory is up-to-date.

  - Sm: Shared-modified, multiple cache lines, memory is not up-to-date.

  - M: Modified, only one cache has line, memory is not up-to-date.

Dragon Protocol

Operations

- There is no invalid state, so all cache line initialized as unused.

For processors/ cores:

- p-load / p-write / p-load-miss / p-write-miss

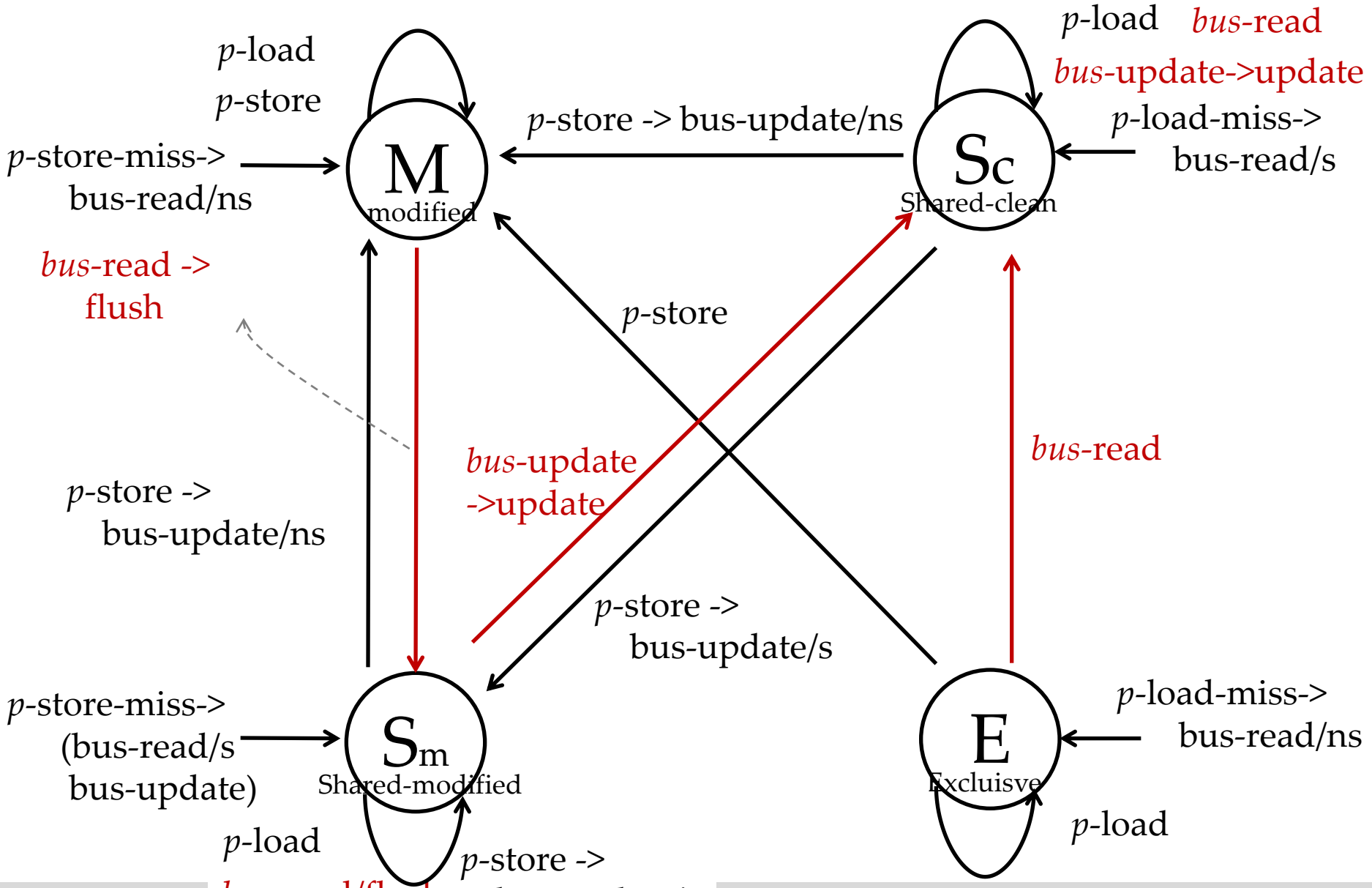
For bus:

- **bus-read(s/ns)** / **bus-update** (similar to bus-read-x)

- **flush** (write value back to memory)

- update (write values to other caches)

Write back update protocol: Dragon Protocol diagram



For many core systems

Coherence protocol

- Many-core systems include 100-1000 or more cores with shared memory.

- Scalability is the most important issue.

- Snooping based protocols may not be that efficient.

- Directory based protocols are good candidates.

Notices: Video Course on Thu.

Two topics:

- Cache Coherence

- Memory Consistency (You should know the basic ideas.)

No meeting

- Another 8-10 projects will be updated on next Tue.

Summary

Coherence Protocol

(1) illinois, MOESI, MESIF

(2) dragon