COMPUTER ARCHITECTURE

MSI and MESI Protocols

Contents

- □ **MSI Protocol.**
- □ MESI Protocol.

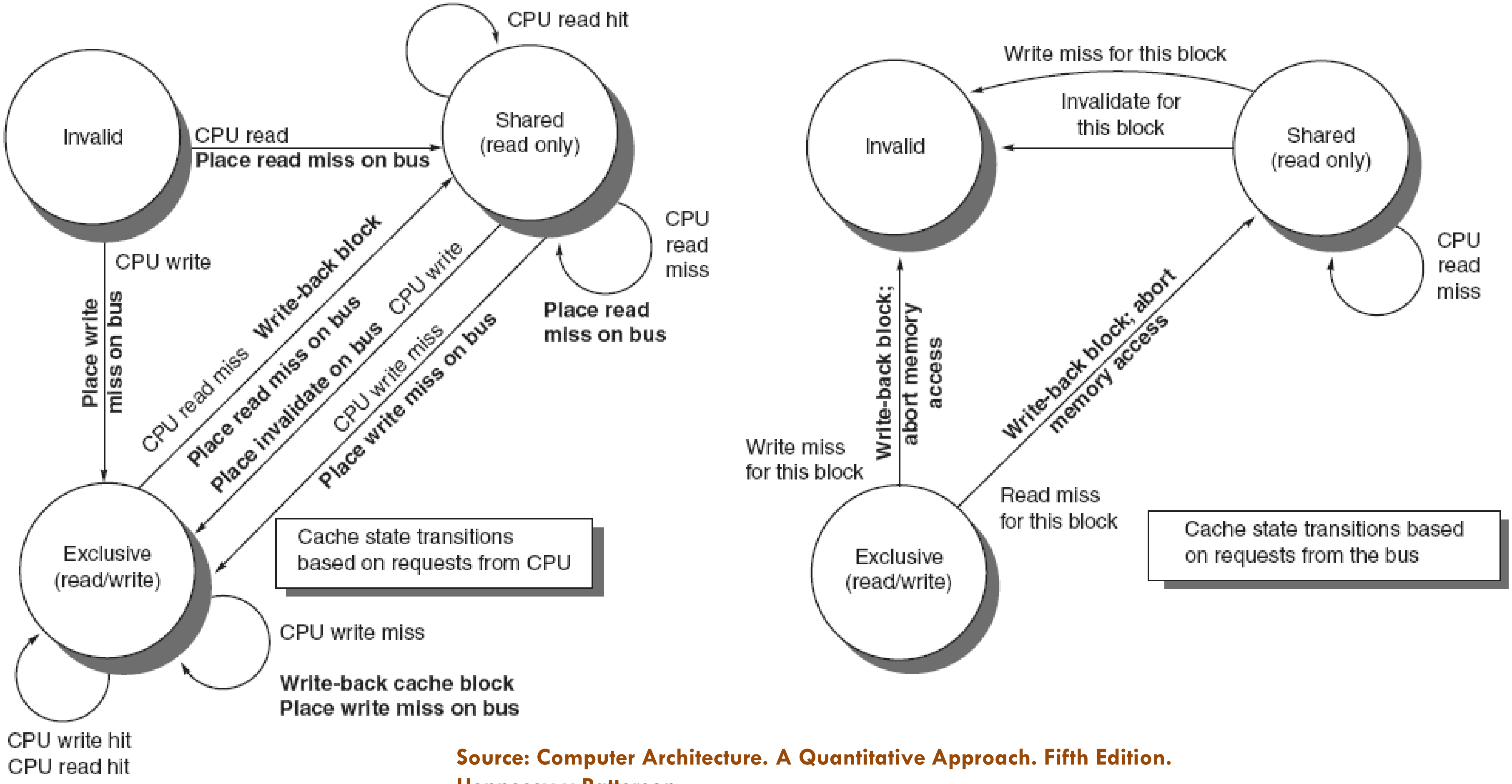
Processor generated actions protocol

Request	State	Action type	Description
Read hit	S o M	Hit	Read data in local cache.
Read miss	I	Miss	Place read miss on bus.
Read miss	S	Replacement	Address conflict miss. Place read miss on bus.
Read miss	M	Replacement	Address conflict miss. Write back block, place read miss on bus
Write hit	M	Hit	Write data in local cache.
Write hit	S	Coherence	Place invalidate on bus.
Write miss	I	Miss	Place write miss on bus.
Write miss	S	Replacement	Address conflict miss. Place write miss on bus.
Write miss	M	Replacement	Address conflict miss. Write back block, place write miss on bus.

Bus generated actions protocol

Request	State	Action type	Description
Read miss	S	-	Shared cache or memory service miss
Read miss	M	Coherence	Attempt to share data. Place cache block on bus and transition to shared.
Invalidate	S	Coherence	Attempt to write shared block. Invalidate block.
Write miss	S	Coherence	Attempt to write shared block. Invalidate block.
Write miss	M	Coherence	Attempt to write block that is exclusive elsewhere. Write-back cache block and make state invalid.

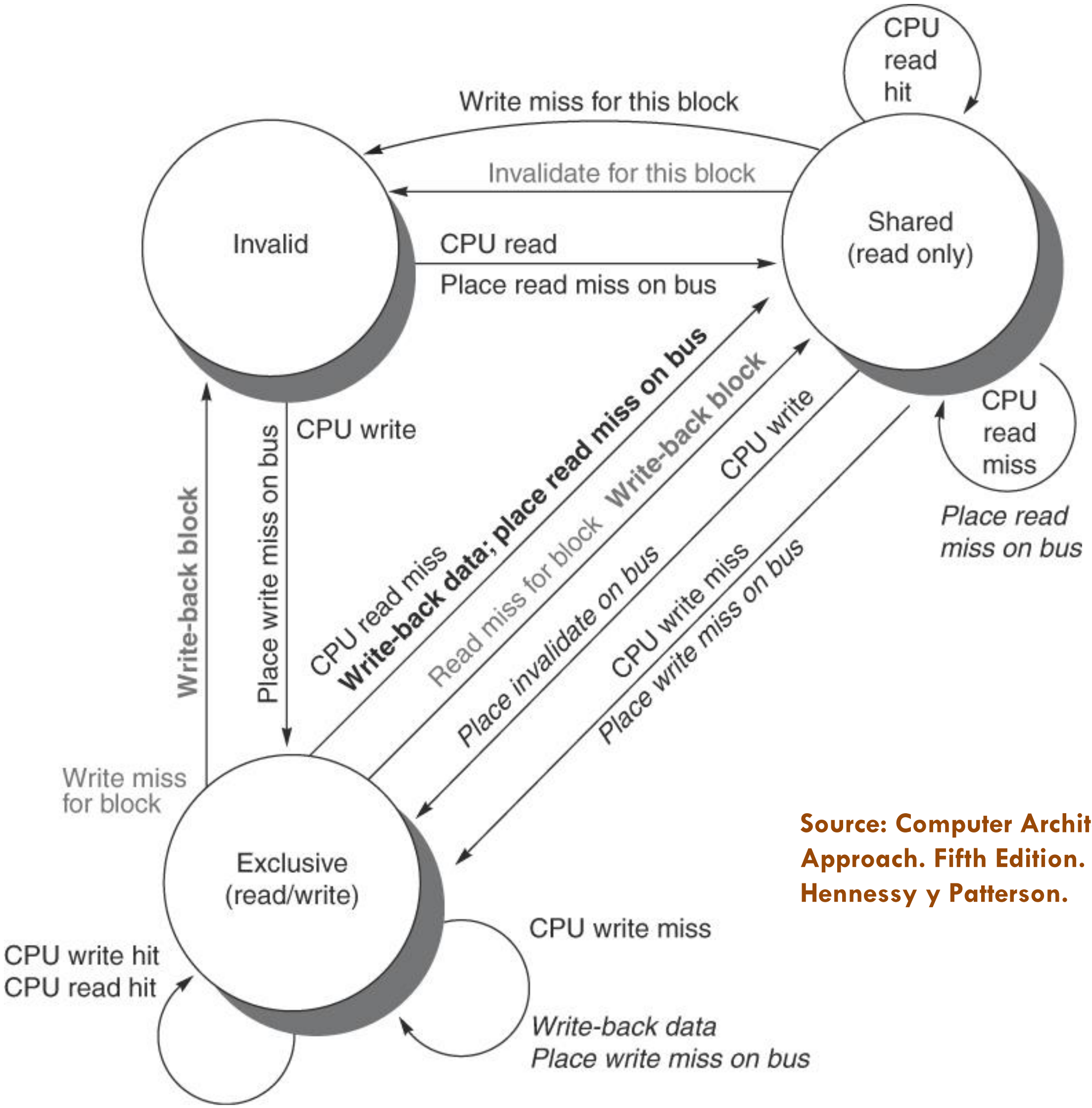
Protocol



Source: Computer Architecture. A Quantitative Approach. Fifth Edition. Hennessy y Patterson.

Universidad Carlos III de Madrid
www.uc3m.es

Protocol

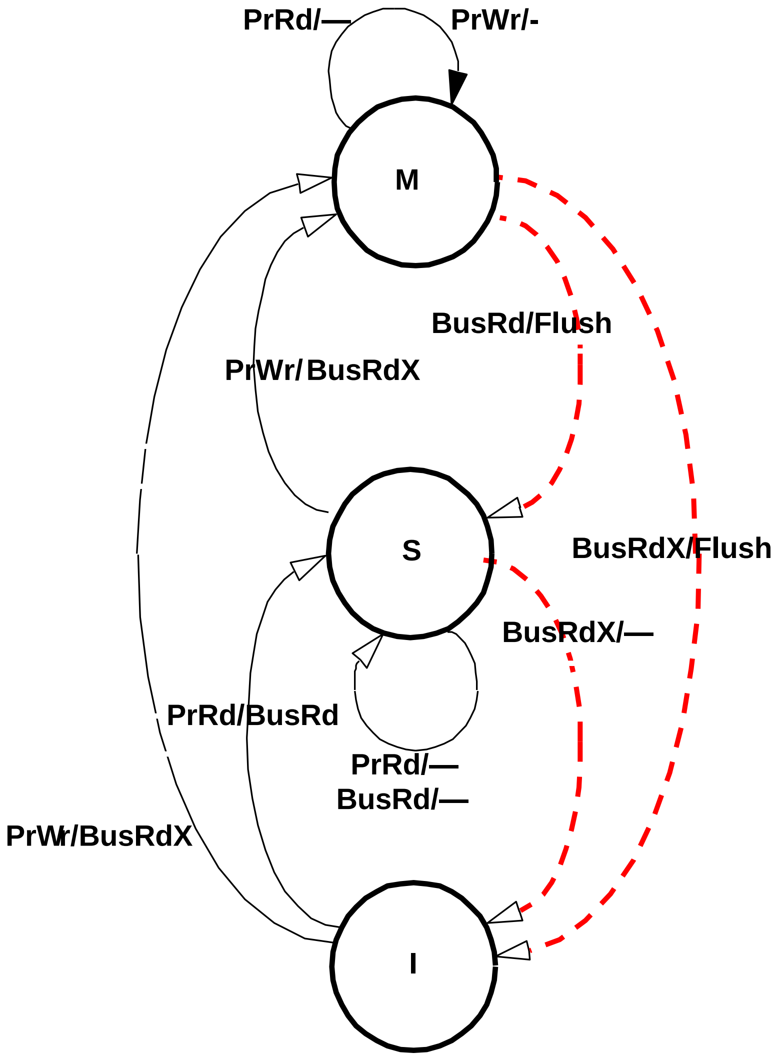


Source: Computer Architecture. A Quantitative Approach. Fifth Edition.
Hennessy y Patterson.

MSI Summary

- **States:**
  - Invalid (I): Invalid block.
  - Shared (S): One or several copies of block.
  - Dirty or Modified (M): Only one copy.
- **Processor Events:**
  - PrRd: Processor Read.
  - PrWr: Processor Write.
- **Processor Transactions:**
  - BusRd (read): Request a copy of block without modifying.
  - BusRdX (read exclusive): Request a copy to modify.
  - BusWB (write back): Update memory.
- **Actions:**
  - Update state, perform transaction through bus and dump value to bus.

Summary state machine



- Replacements and write backs not shown.
- No bus transactions due to:
  - Rd/Wr in M state.
  - Rd in S state.
- Two bus transactions for:
  - Rd/Wr in I state.
- Wr in S state causes two bus transactions:
  - Block transition to modified.
  - Transition can be avoided as data is already there:
    - BusUpgr instead of BusRdx

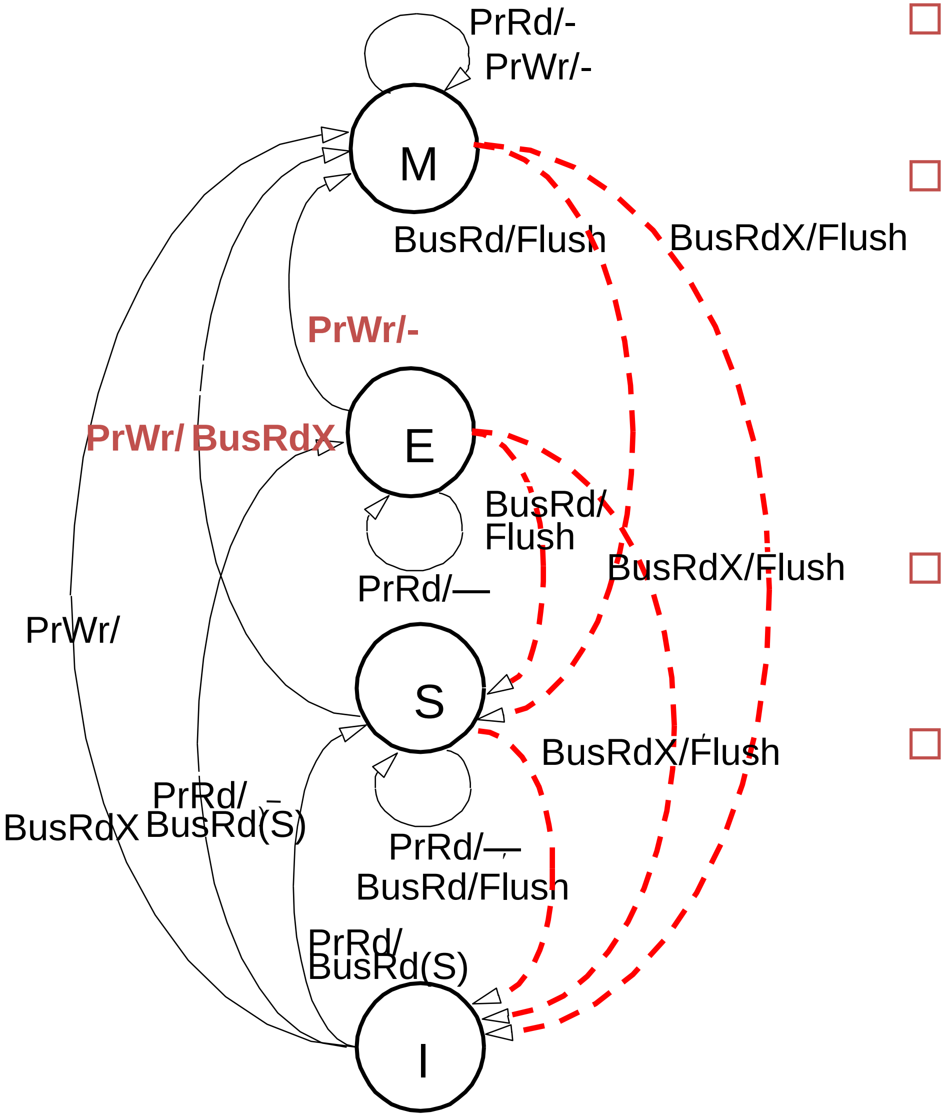
Problems with MSI

- Read and modify data are two bus transactions even if not shared.
  - Even in sequential programs!
  - BusRd (I->S) followed by BusRdX or BusUpgr (S->M)
- Solution:
  - Add new state: Exclusive (E).
- States:
  - M: Modified (dirty).
  - E: Exclusive (single non-modified copy).
  - S: Shared.
  - I: Invalid

Contents

- □ MSI Protocol.
- □ **MESI Protocol.**

MESI



- □ Hit: No transaction in bus in E state.
- □ Transitions from I:
  - To E with PrRd if no other processor has copy.
  - To S, otherwise.
- □ S, additional signal on bus.
- □ BusRd(S): If processor keeps block, set S signal to 1.