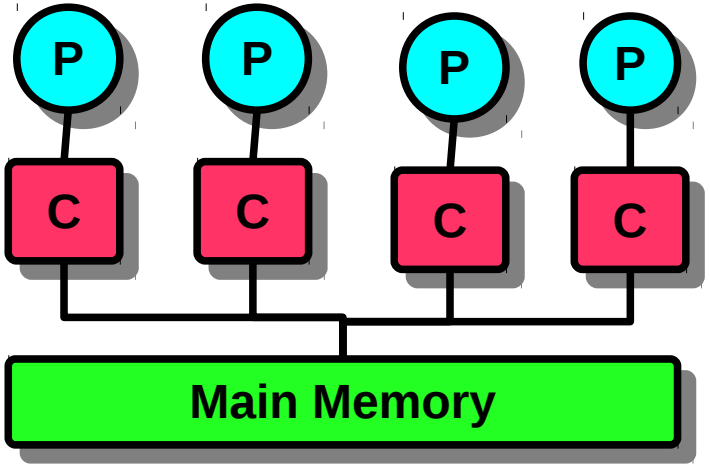
M4 – Parallelism

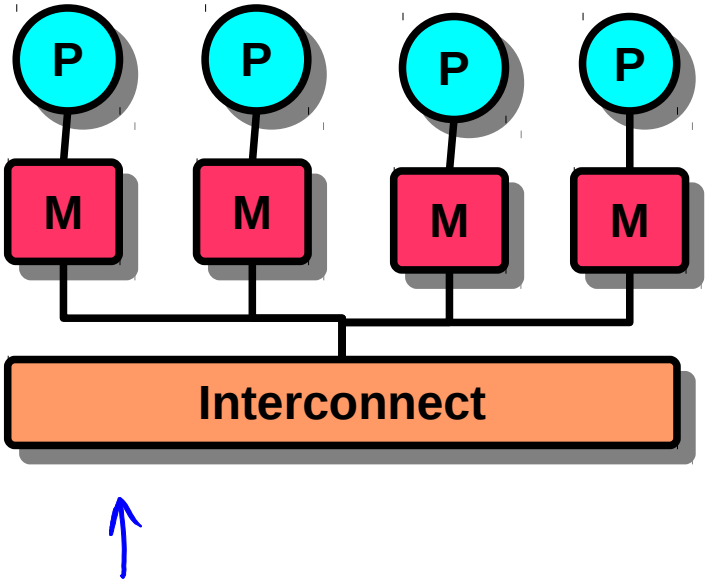
Directory based Cache Coherence Protocol

Outline

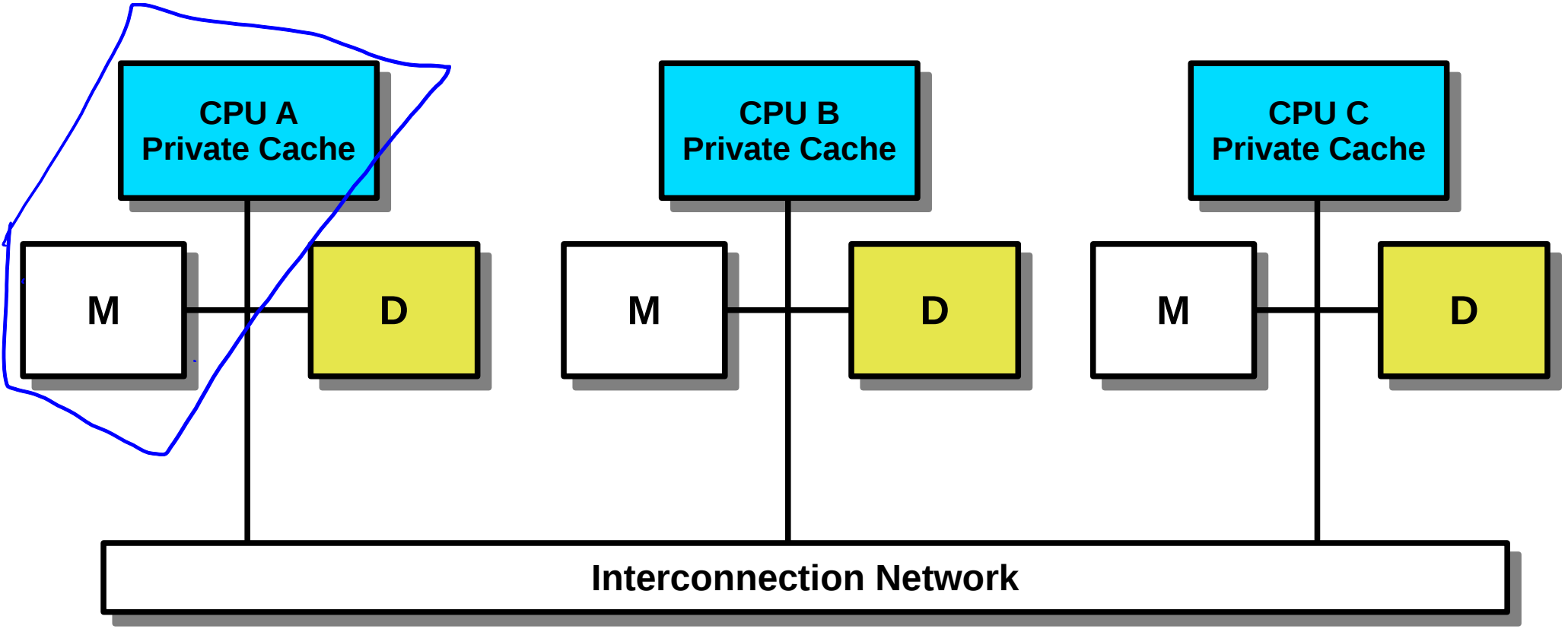
- Parallelism
- Flynn’s classification
- Vector Processing
  - Subword Parallelism
- Symmetric Multiprocessors, Distributed Memory Machines
  - Shared Memory Multiprocessing, Message Passing
- Synchronization Primitives
  - Locks, LL-SC
- Cache coherence

Shared Memory vs. Distributed Memory

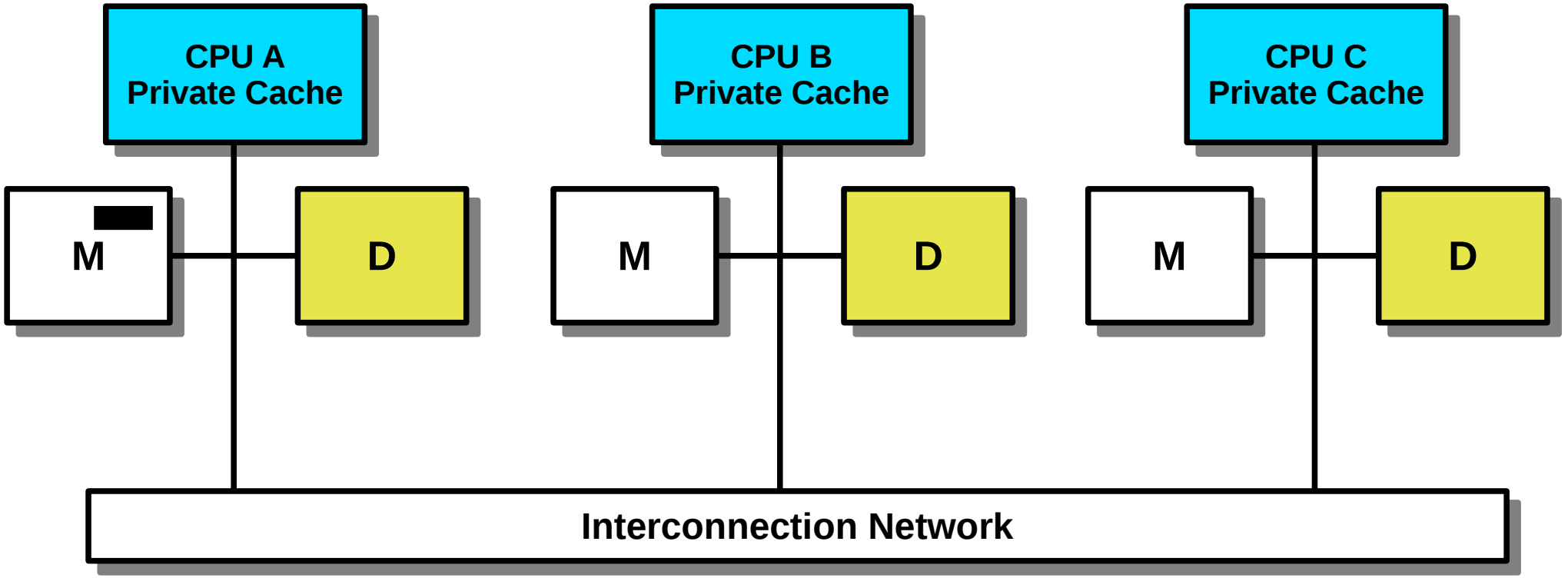




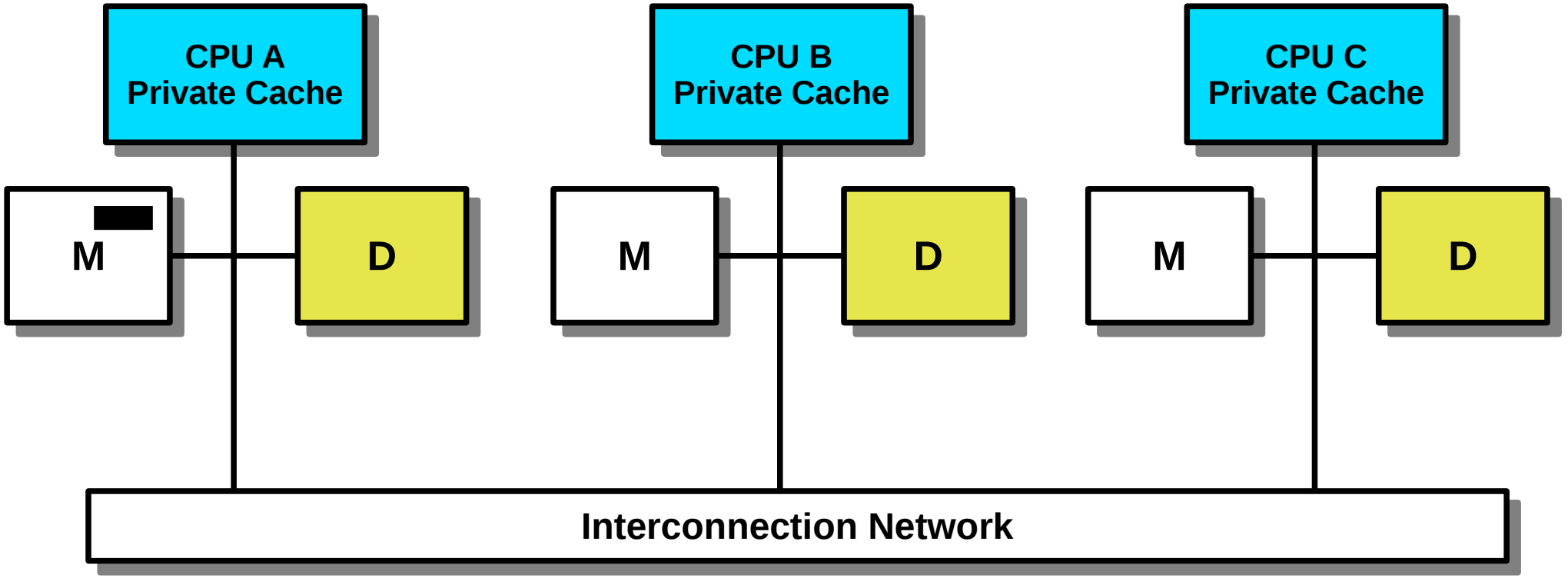
Directory Based Cache Coherence



Directory Based Cache Coherence

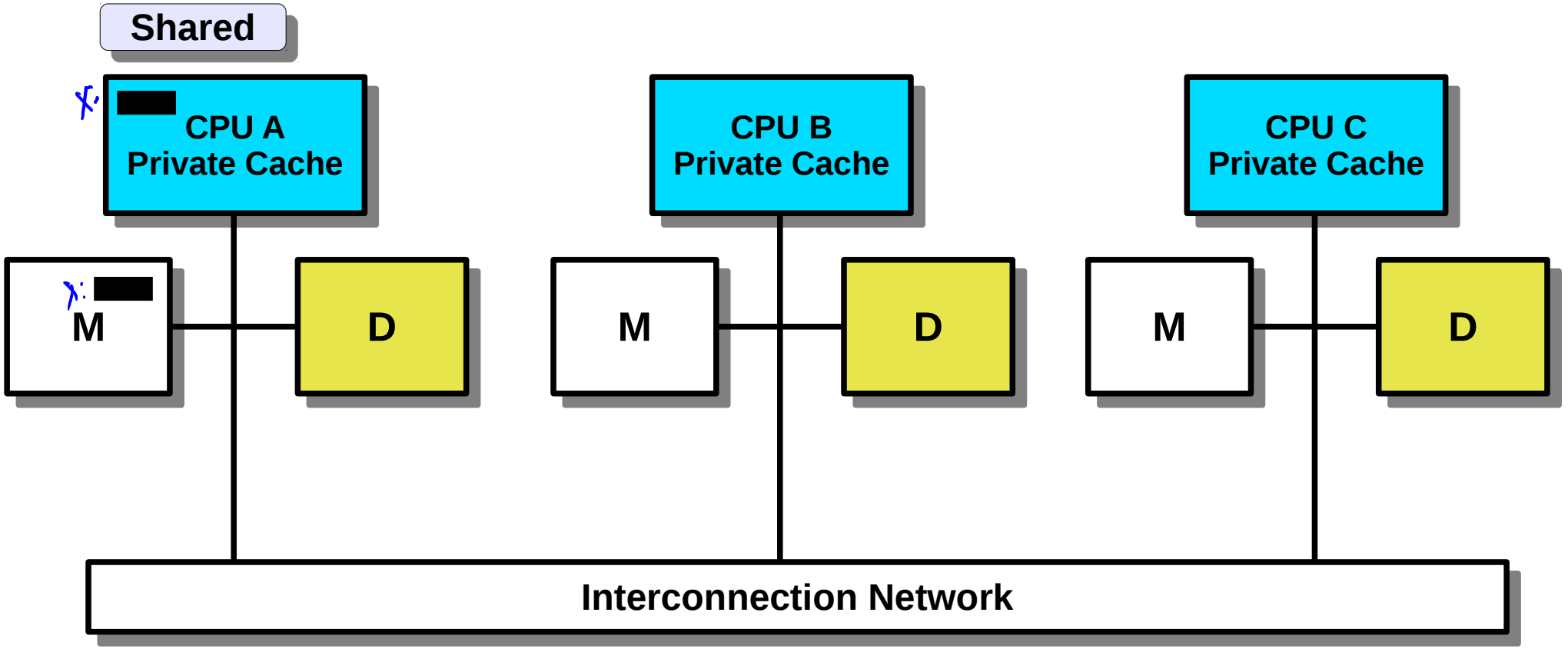


Directory Based Cache Coherence



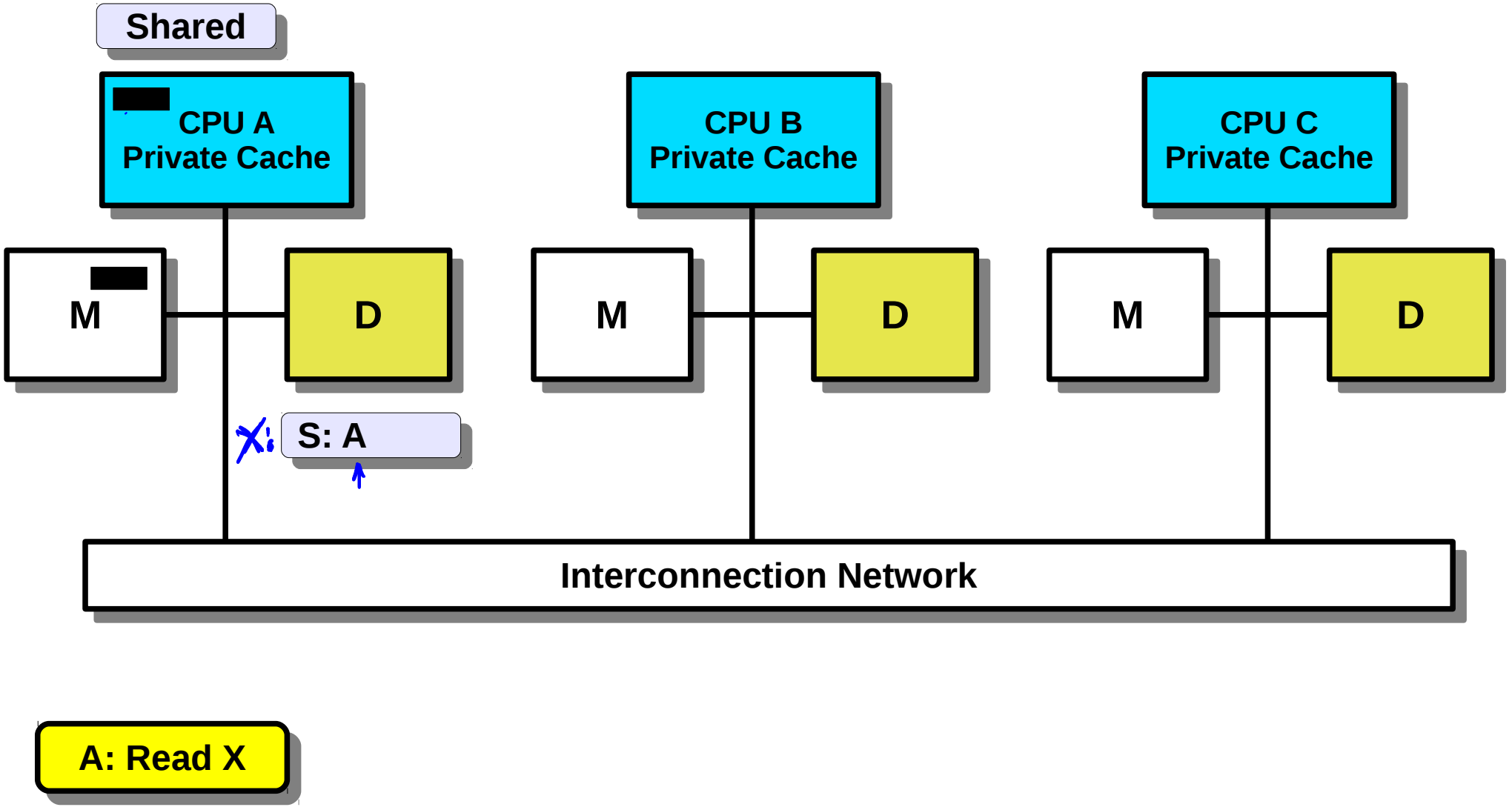
A: Read X

Directory Based Cache Coherence

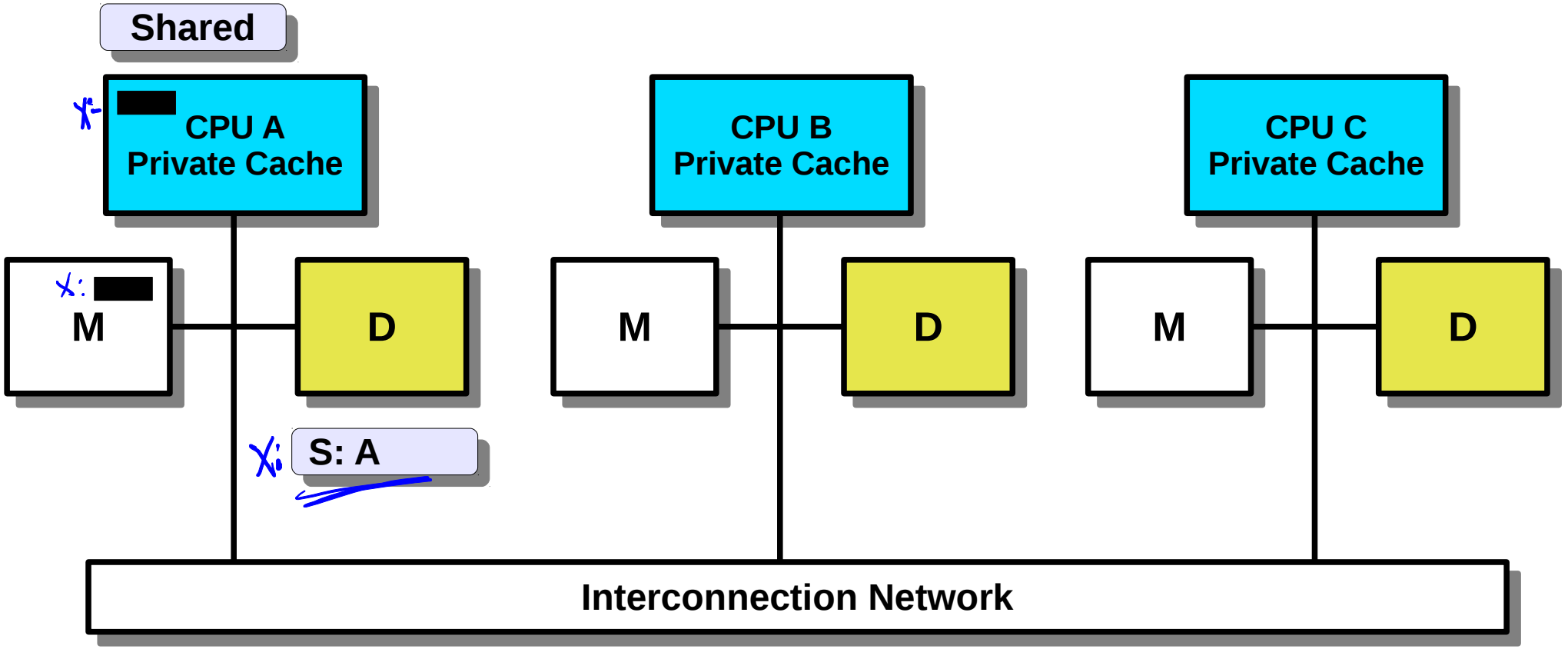


A: Read X

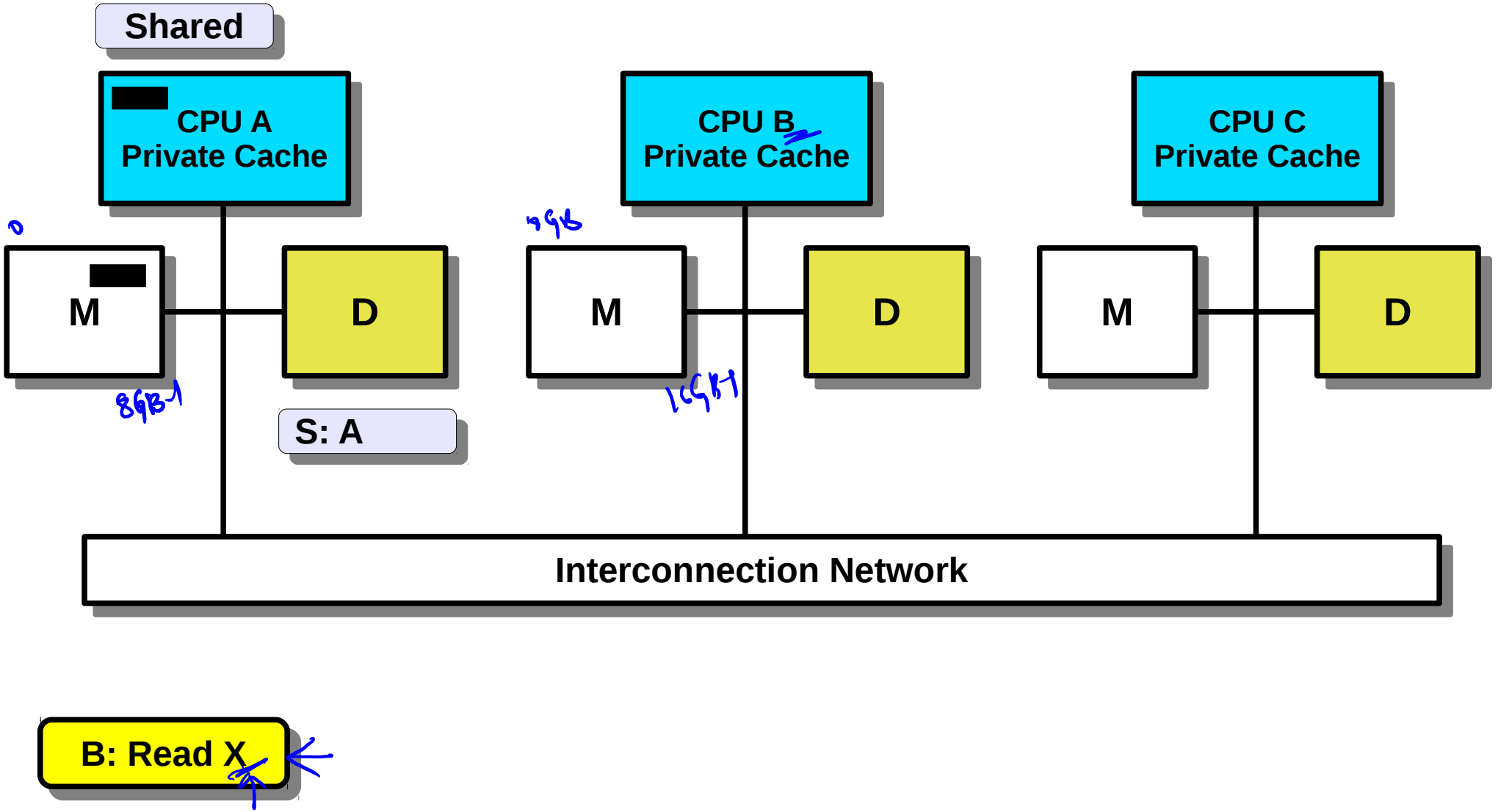
Directory Based Cache Coherence



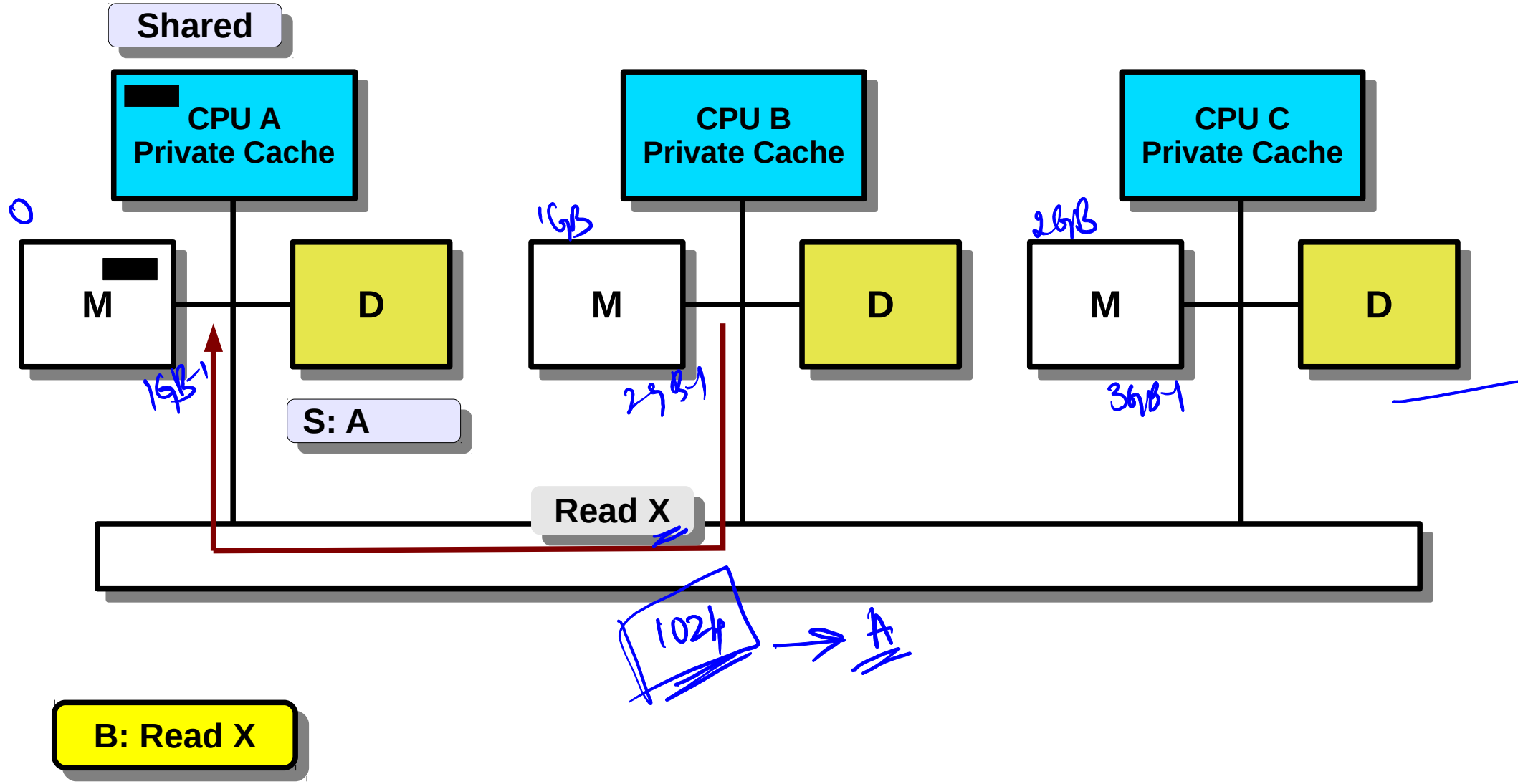
Directory Based Cache Coherence



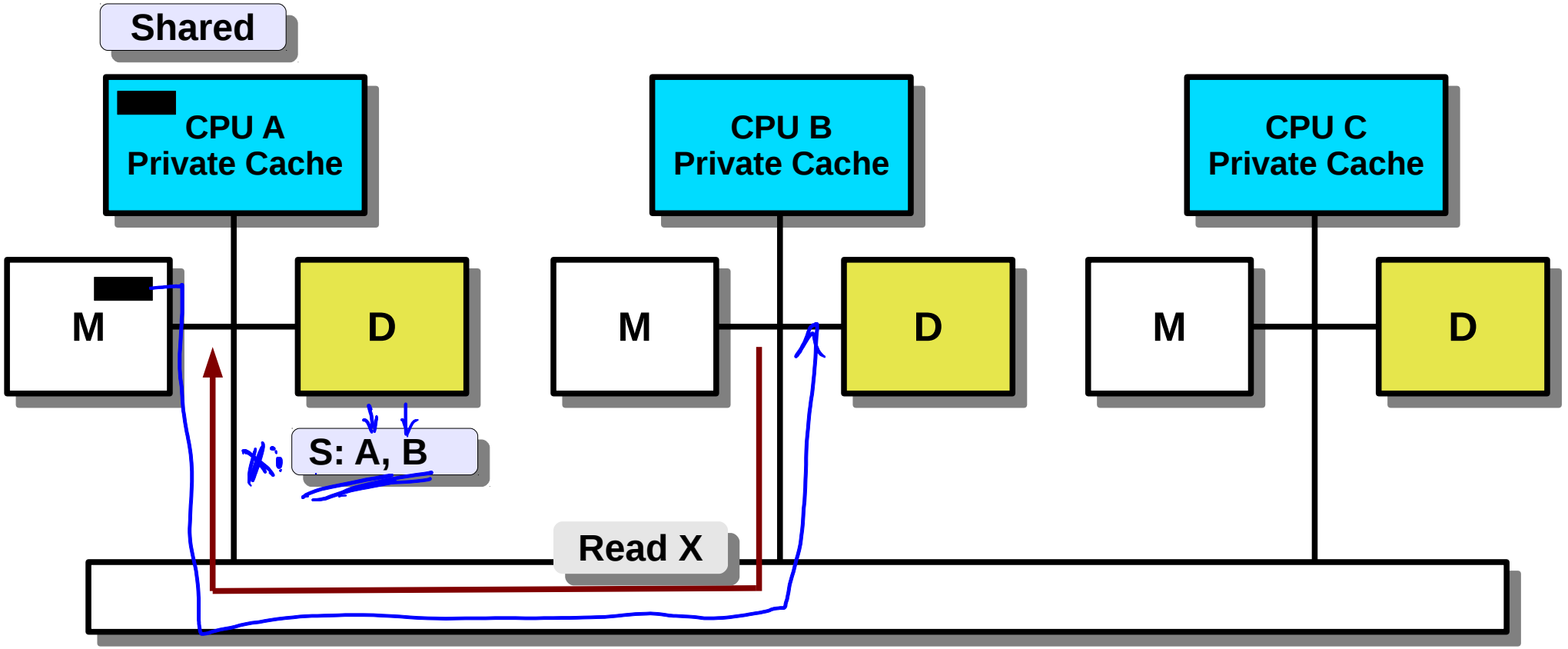
Directory Based Cache Coherence



Directory Based Cache Coherence

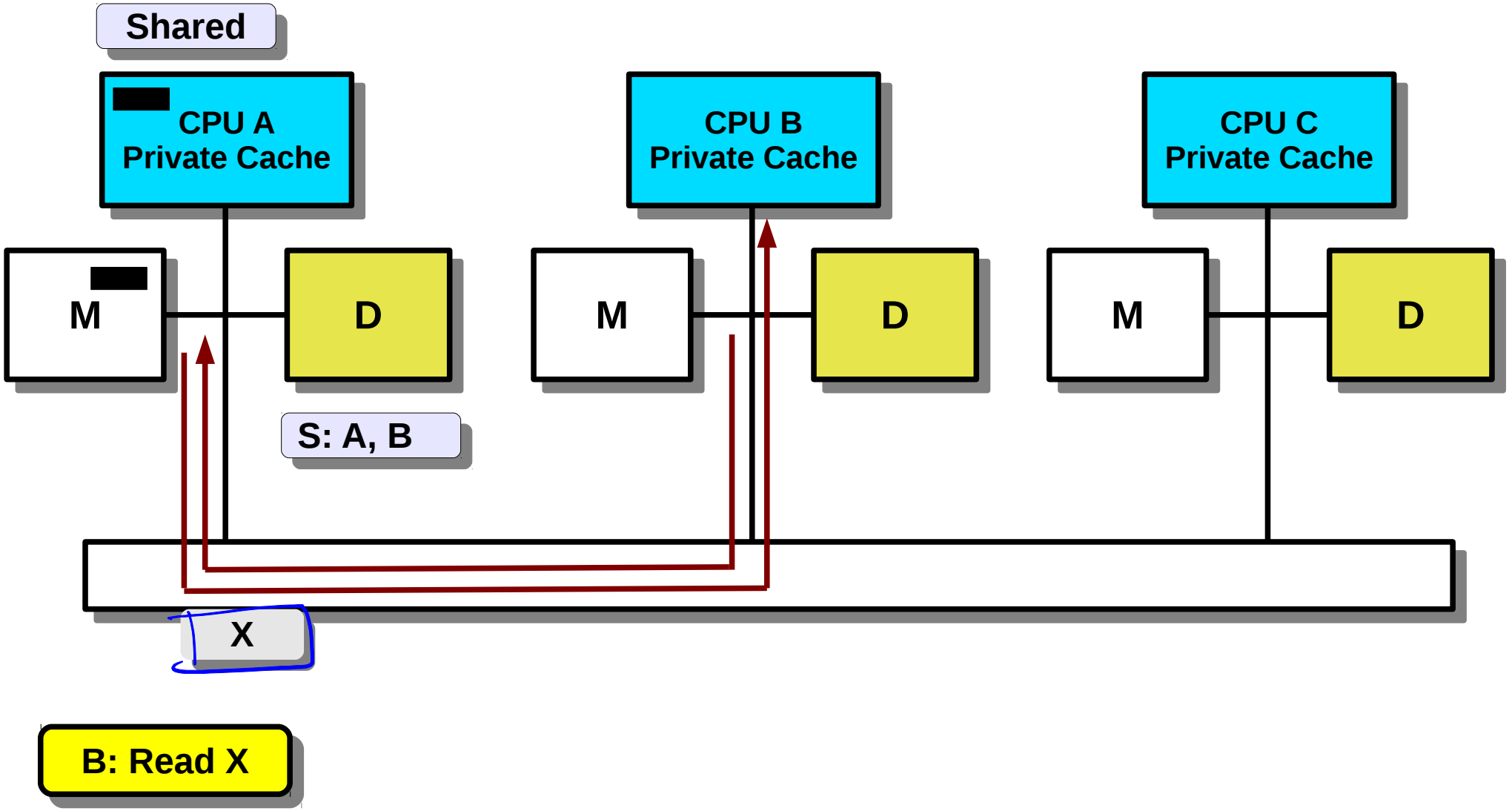


Directory Based Cache Coherence

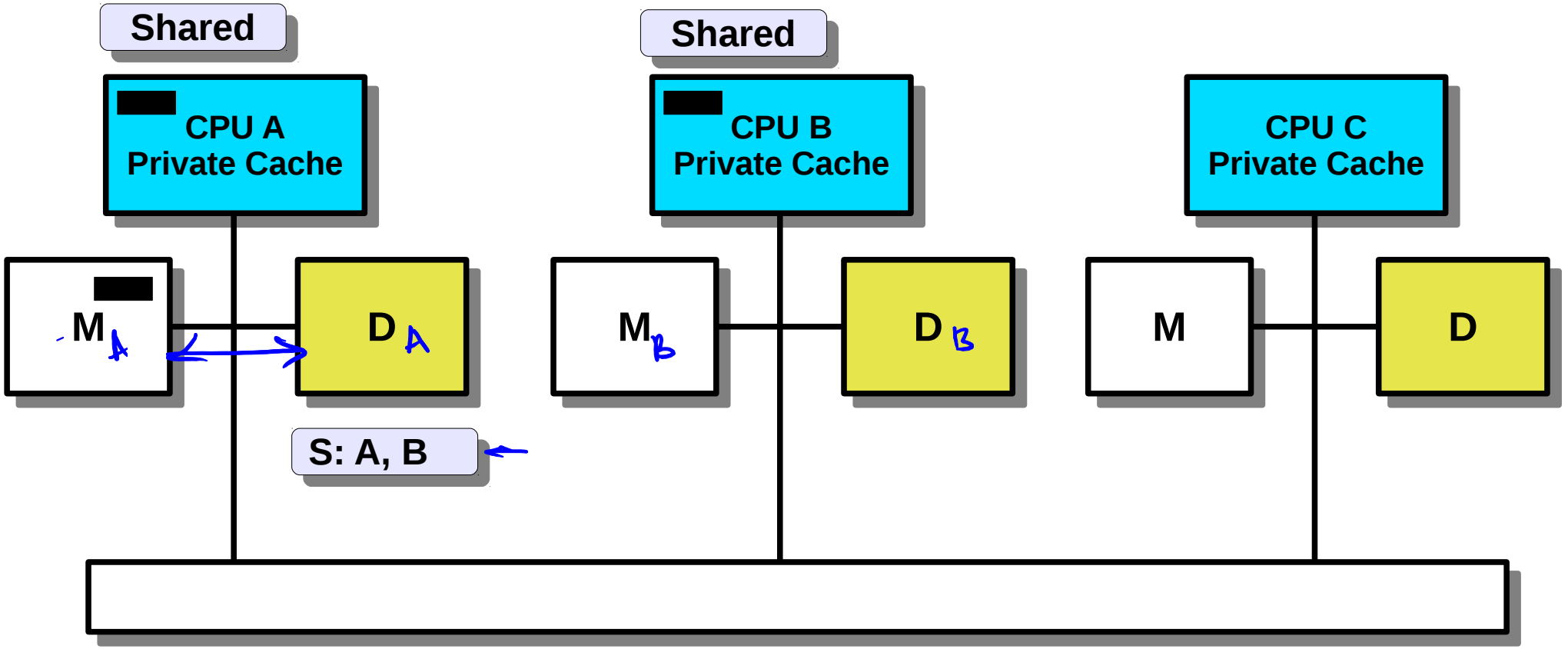


B: Read X

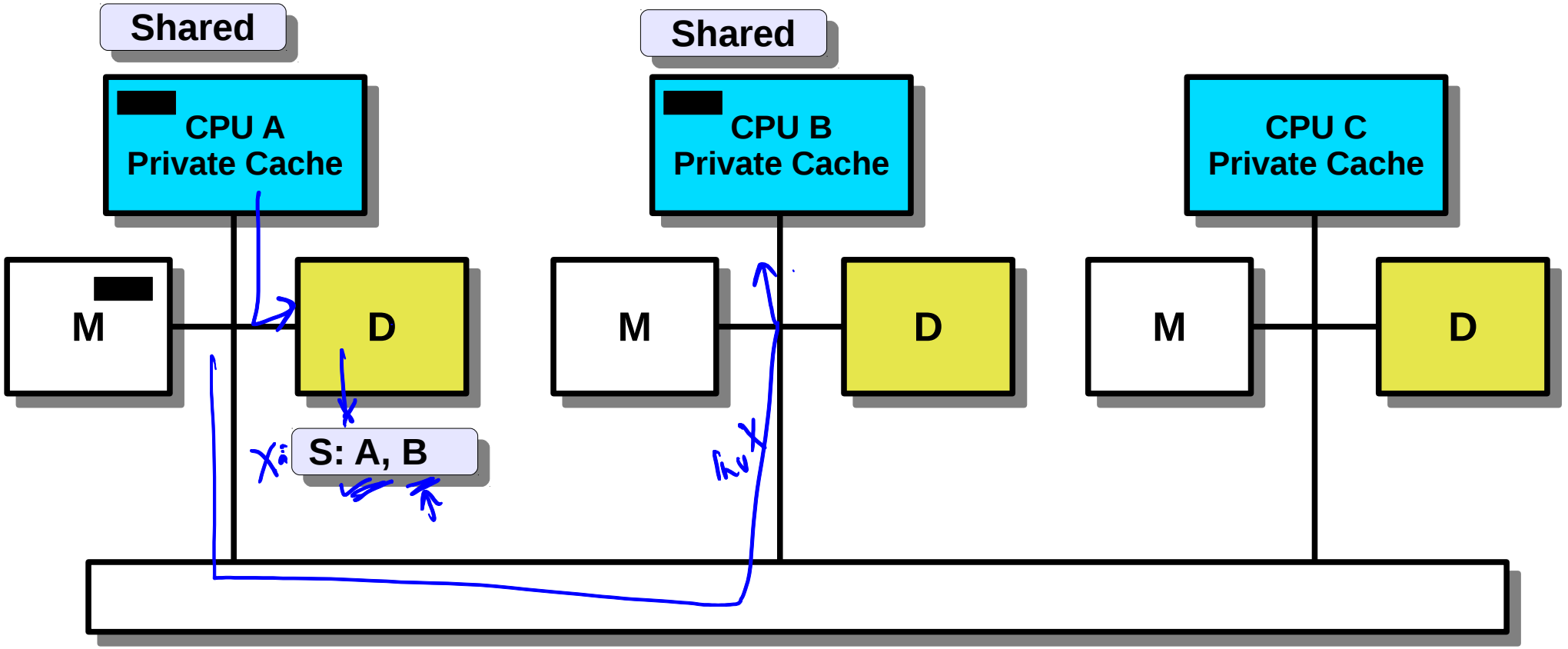
Directory Based Cache Coherence



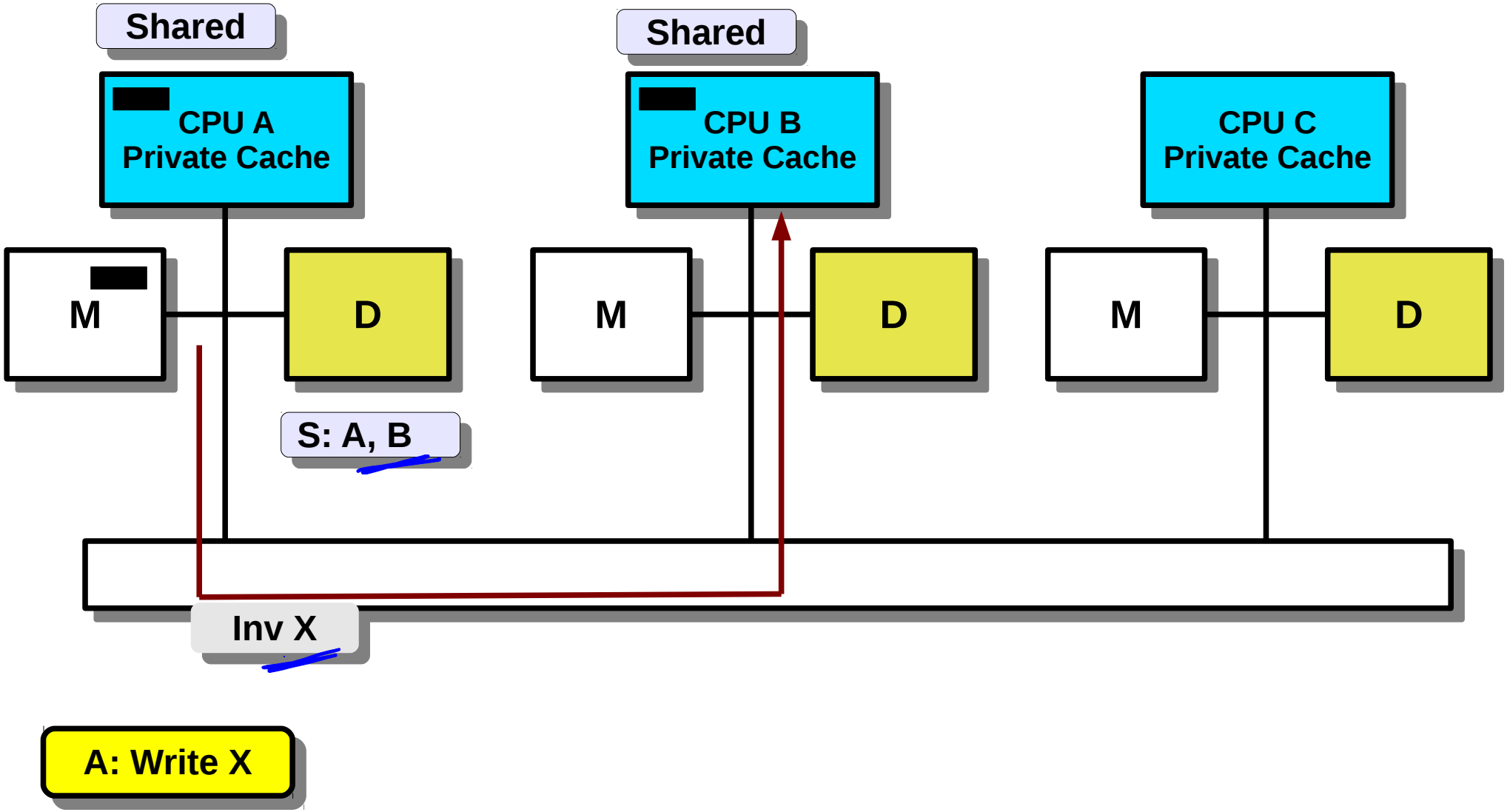
Directory Based Cache Coherence



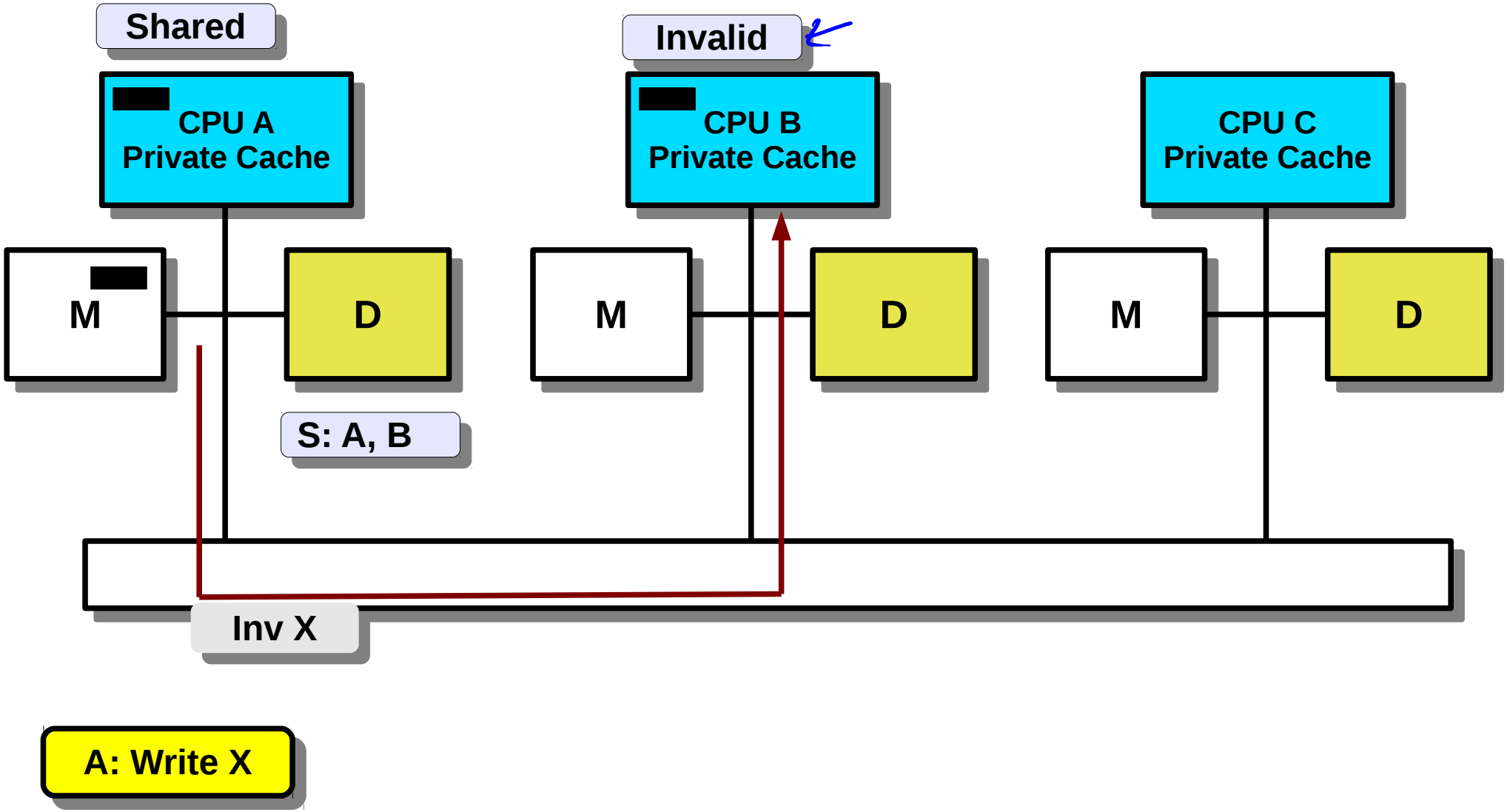
Directory Based Cache Coherence



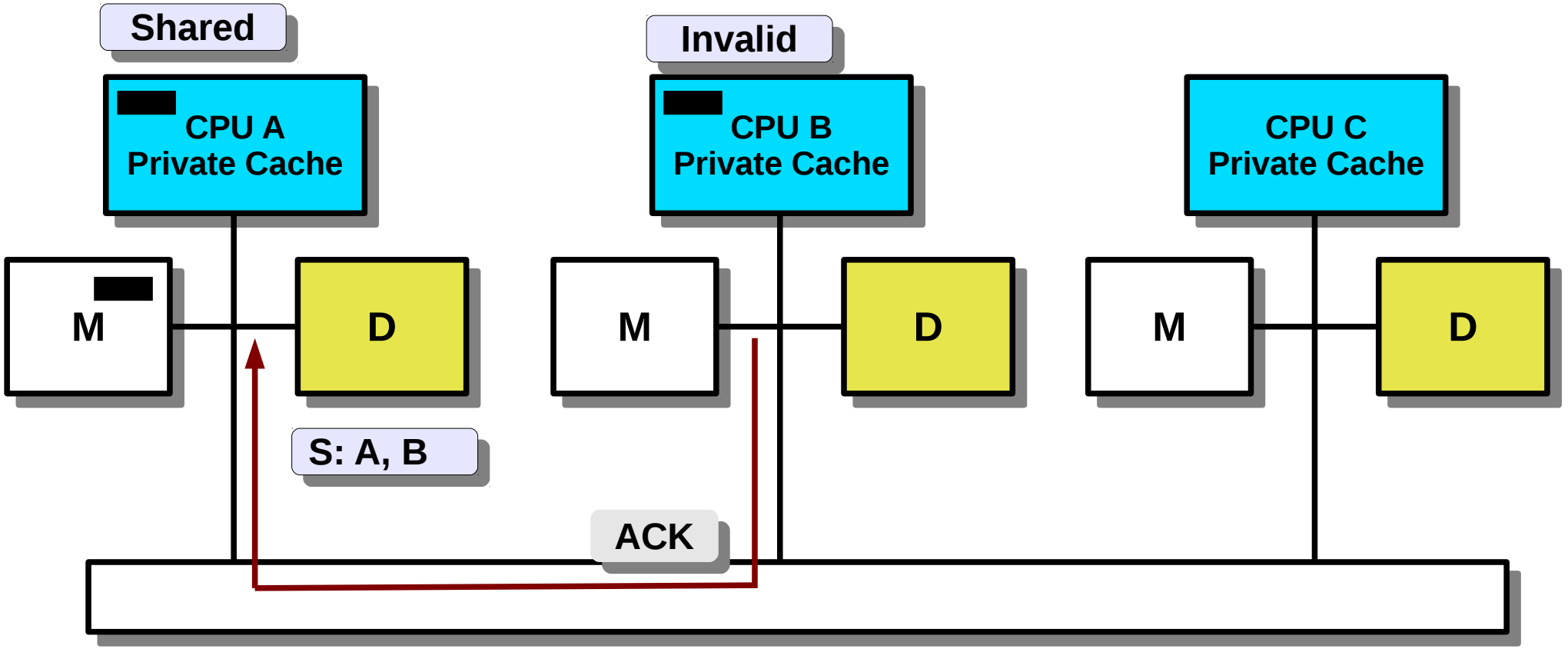
Directory Based Cache Coherence



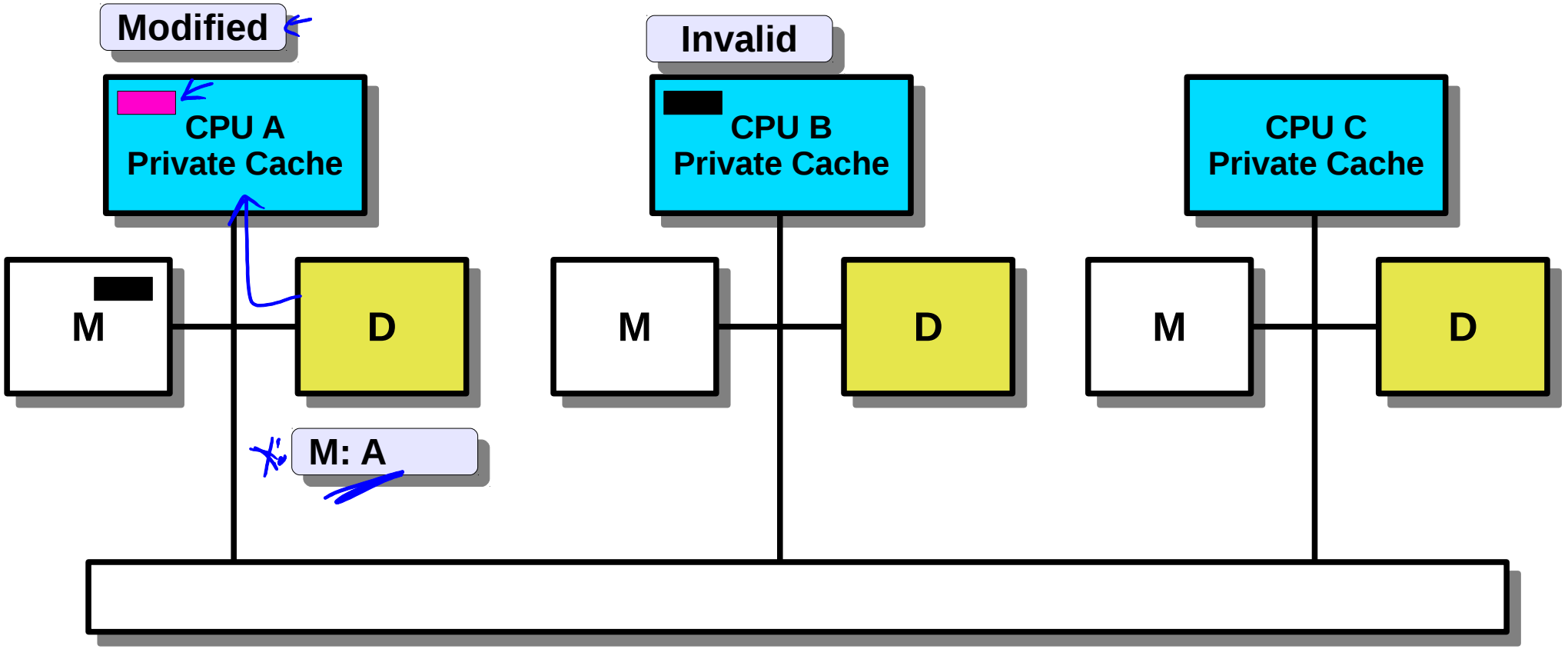
Directory Based Cache Coherence



Directory Based Cache Coherence

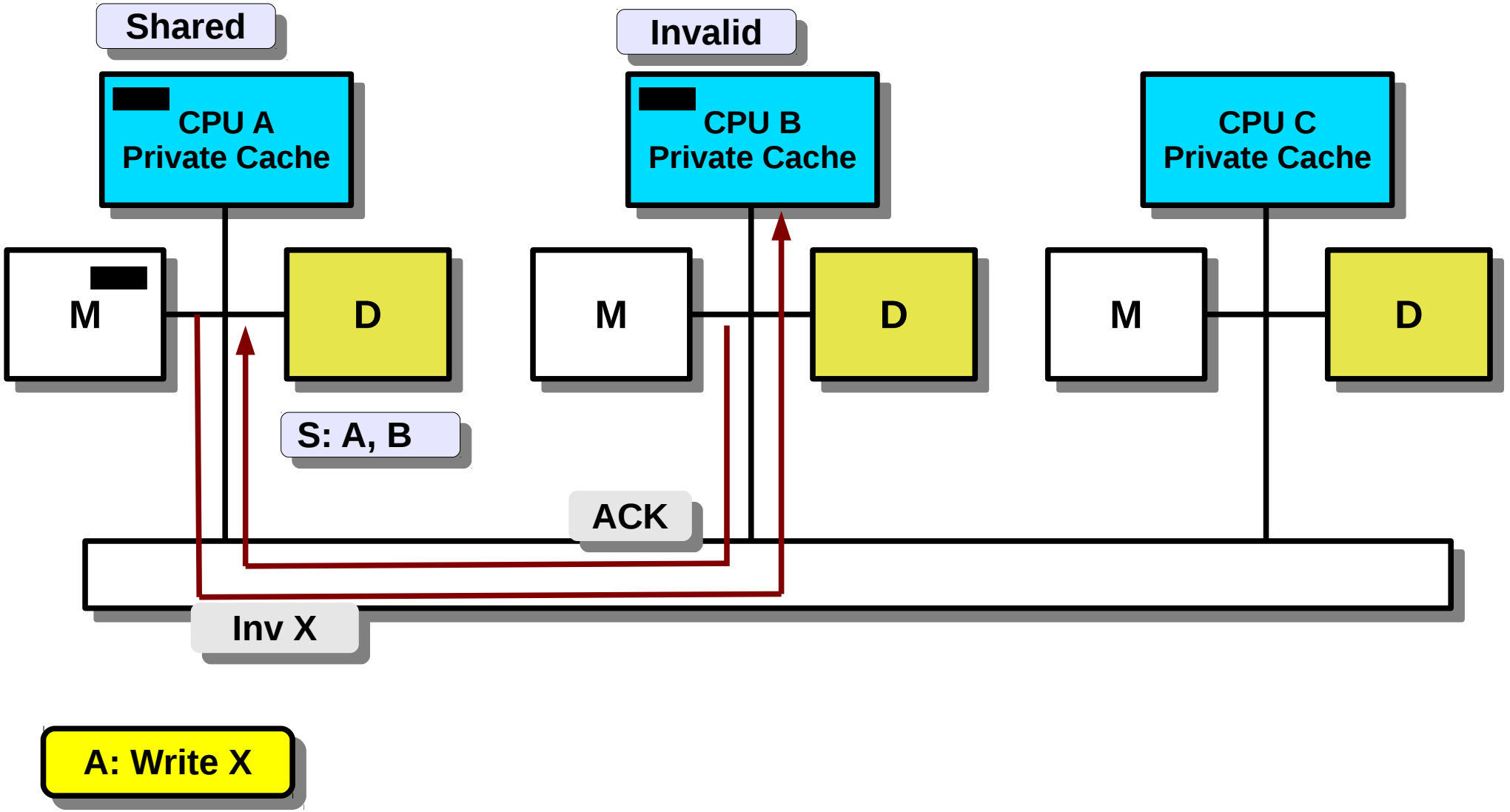


Directory Based Cache Coherence

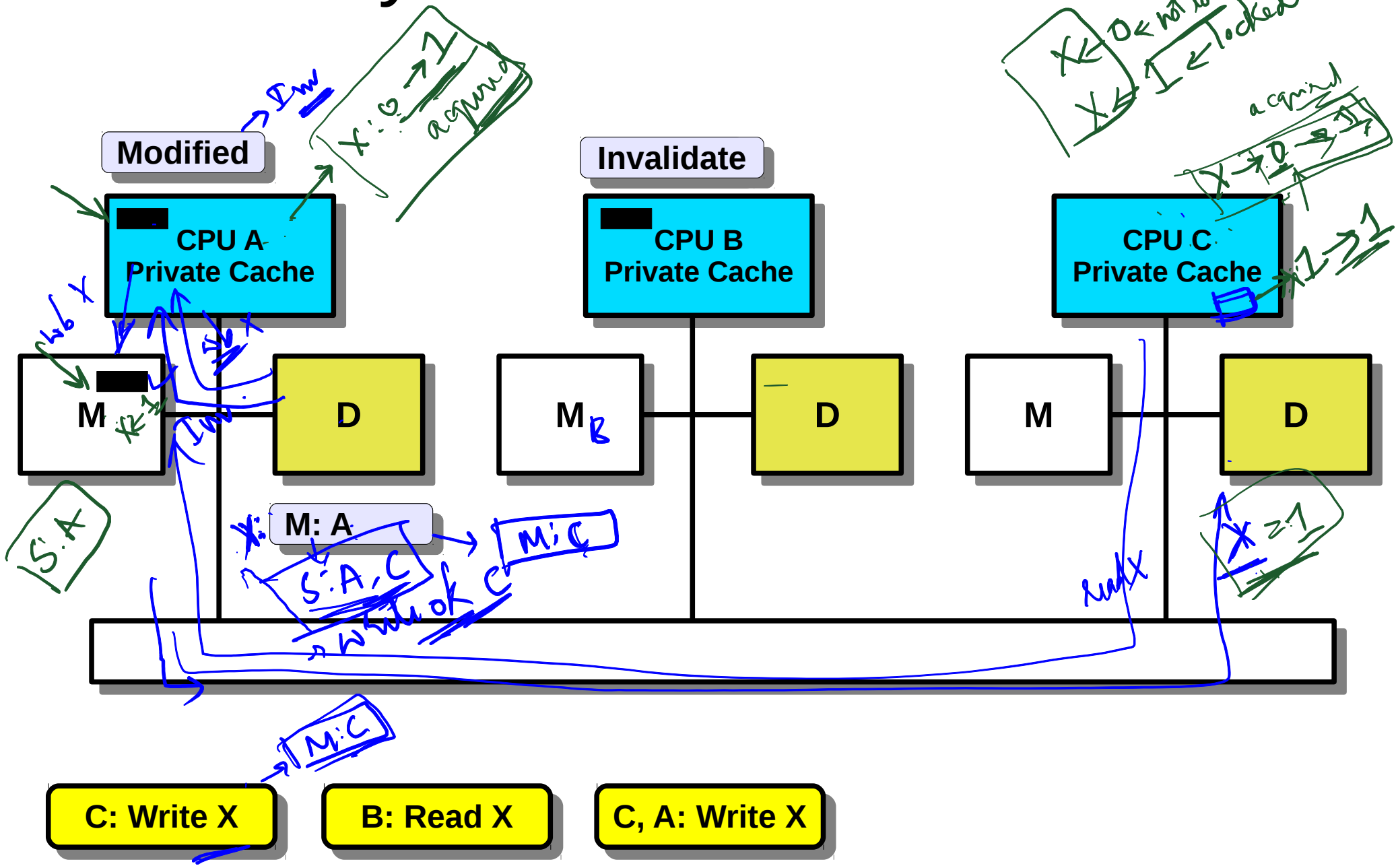


A: Write X

Directory Based Cache Coherence



Directory Based Cache Coherence



X | M : A

→ C read X

1. cache miss X
2. Read X : C → A
3. D_A : X : | M : A |
   ↳ write back X → A
4. A : wb X
5. D_A : X : | S : A |
6. D_A : sends X → C
   ∴ | S : A, C |

→ C write X

D_A : X | M : B

wb : X → B

S_n : wb X

7. D_A : | S : A, C |

D_A : write inv → A

8. A : X → Invalid
9. D_A : | M : C |
10. write op. completes

Directory Based Cache Coherence

- Broadcast based snooping protocols do not scale well to large multiprocessors
- Distributed Memory Machines
  - Physical memory is distributed among all processors
- Directory tracks sharing status of a block of memory
  - Each node has a directory
- Physical address determines data location
- Coherence messages between sent over the ICN
  - Point-to-point messages (no broadcast)

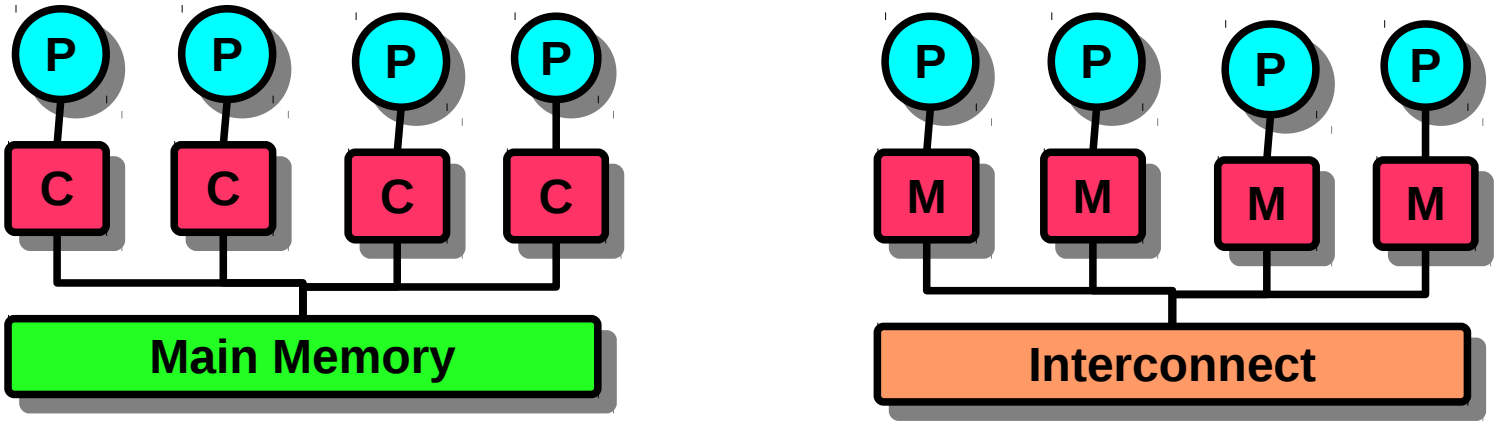
Slides Contents

- Rajeev Balasubramonian, CS6810, University of Utah.
-

Extra

Shared Memory vs. Message Passing

- **Shared Memory Machine**: processors share the same physical address space
  - Implicit Communication, Hardware controlled cache coherence
- **Message Passing Machine**
  - Explicit communication – programmed
  - No cache coherence (simpler hardware)
  - Message passing libraries: MPI



Cache Coherence

- Consistency
  - When should a written value be available to read
  - **Memory Consistency Models**
- Coherence
  - Which value to return on a read
- A memory system is coherent if:
  - Write Propagation
    - A write is visible after a sufficient time lapse
  - Write Serialization
    - All writes to a location are seen by every processor in the same order

Multiprocessor Cache Coherence

- A **read** by a processor P to a location X that **follows a write** by P to X, with no writes of X by another processor occurring between the write and the read by P, always **returns the value written** by P.
- A **read** by a processor to location X that **follows a write** by another processor to X returns the written value if the **read and write** are **sufficiently separated in time** and no other writes to X occur between the two accesses.
- **Writes** to the same location are **serialized**; that is, two writes to the same location by any two processors are seen in the same order by all processors.

Write Invalidate Coherence Protocol

Processor activity	Bus activity	Contents of CPU A's cache	Contents of CPU B's cache	Contents of memory location X
				0

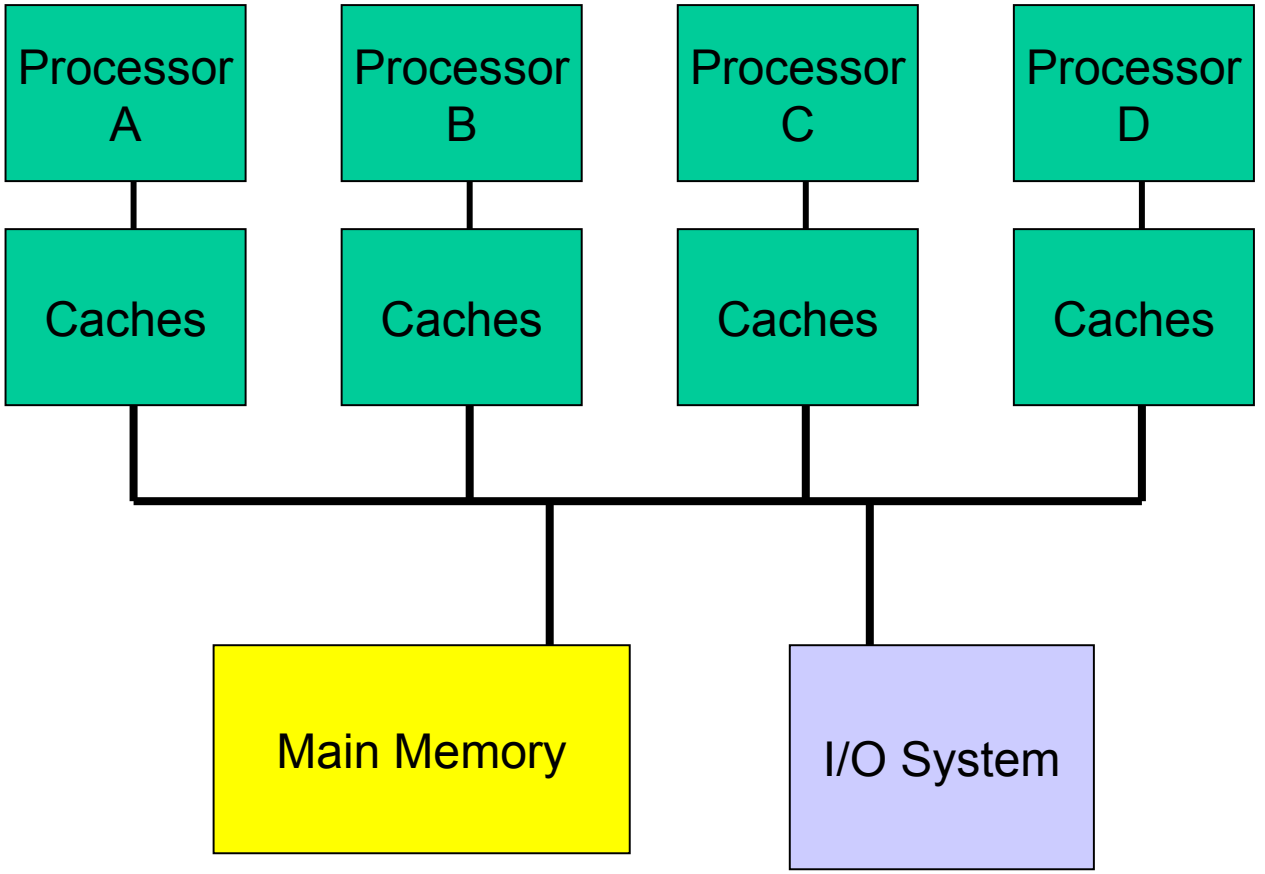
Writeback / Writethrough
Enforcing write serialization
- Bus Arbitration

Tag Contention, Duplication

SMP Cache Coherence

- MSI Protocol
- MESI Protocol
  - Exclusive state: No invalidate messages on writes.
  - Intel i7 uses MESIF
- MOESI Protocol
  - Owned state: Only valid copy in the system. Main memory copy is stale.
  - Owner supplies data on a miss.

SMP Example



A: Rd X
B: Rd X
C: Rd X
A: Wr X
A: Wr X
C: Wr X
B: Rd X
A: Rd X
A: Rd Y
B: Wr X
B: Rd Y
B: Wr X
B: Wr Y