

# COMP35112 Chip Multiprocessors

## MESI/MOESI Cache Coherence

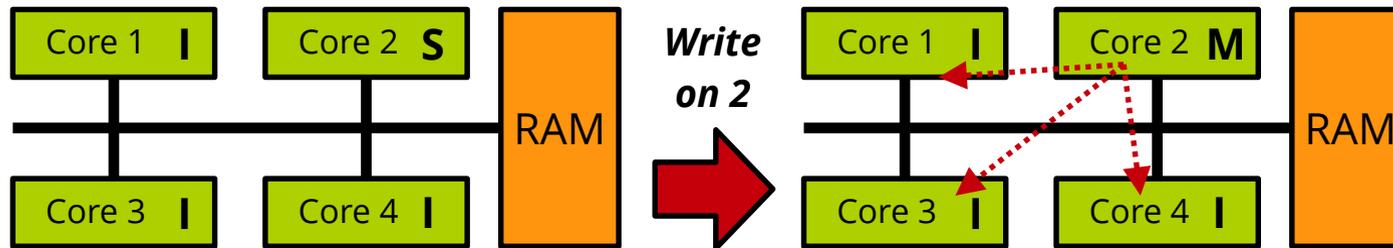
Pierre Olivier

# Unnecessary Communication

- Bus: critical shared resource, unnecessary use impacts performance

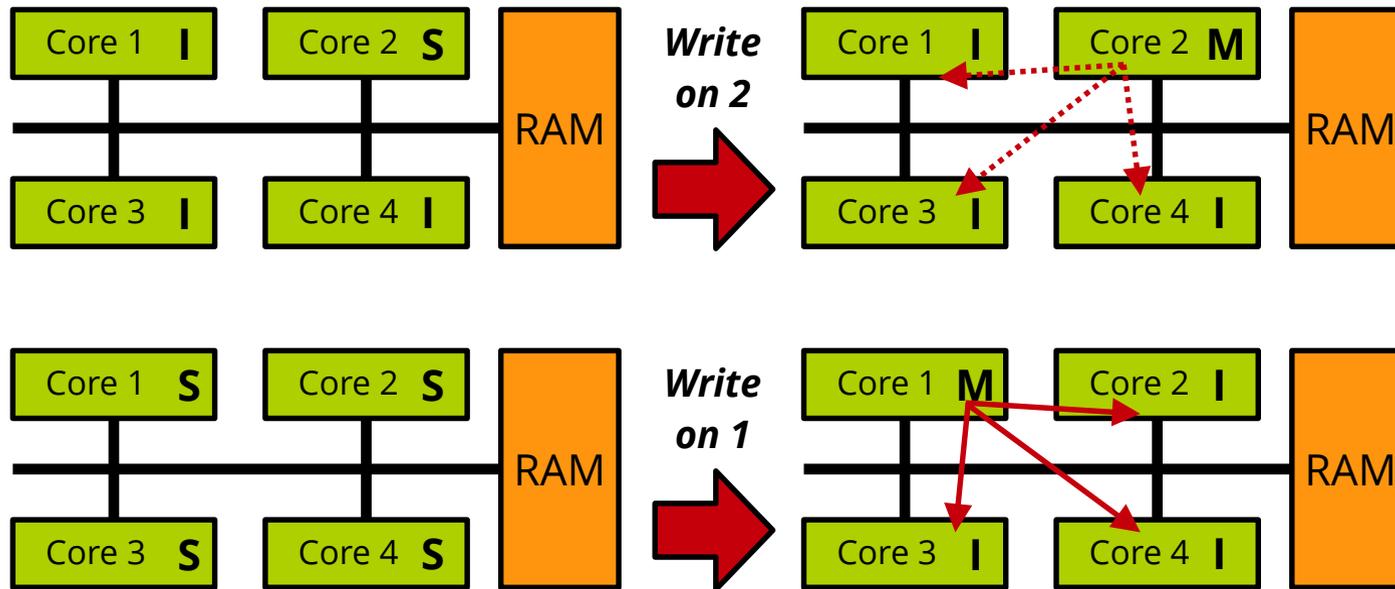
# Unnecessary Communication

- Bus: critical shared resource, unnecessary use impacts performance



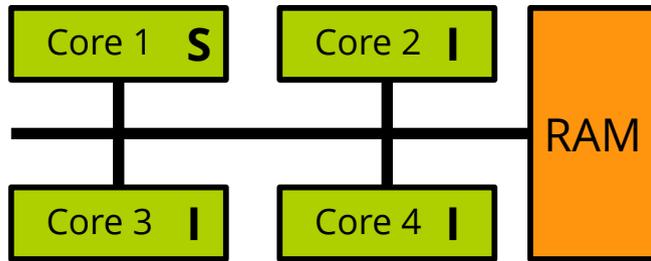
# Unnecessary Communication

- Bus: critical shared resource, unnecessary use impacts performance



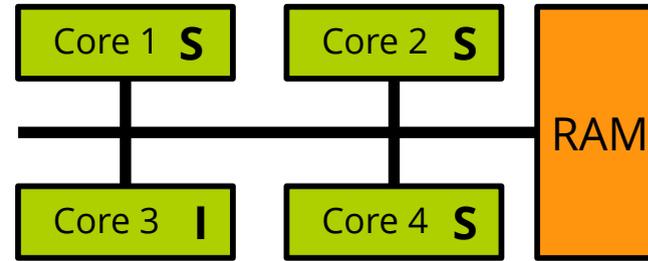
# Optimising for Non-Shared Values

- Distinguish between the two shared cases



Cache holds **the only shared copy**

**No need for invalidate broadcast**

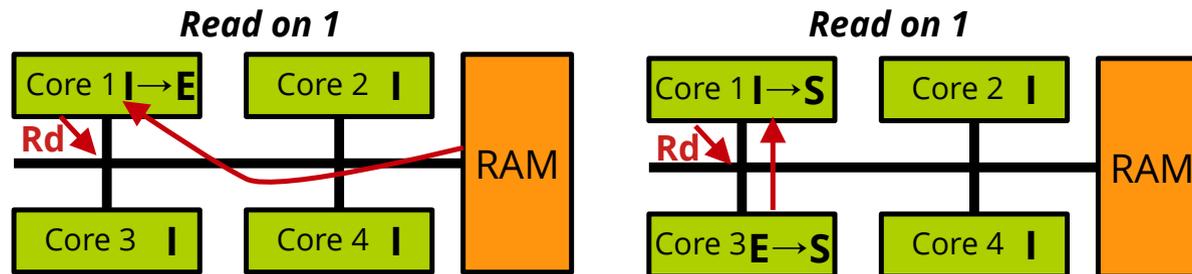


Cache holds **one of the shared copies**

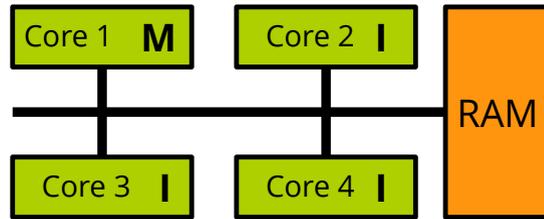
**Need to broadcast invalidate**

# MESI Protocol

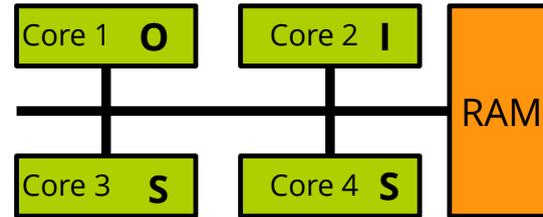
- The unshared case is very common
- **Split the S state into:**
  - **E: exclusive**
    - Switch to E after a read causing a fetch from memory
  - **S: (truly) shared**
    - Switch to S after a read that gets value from another cache



# MOESI Protocol



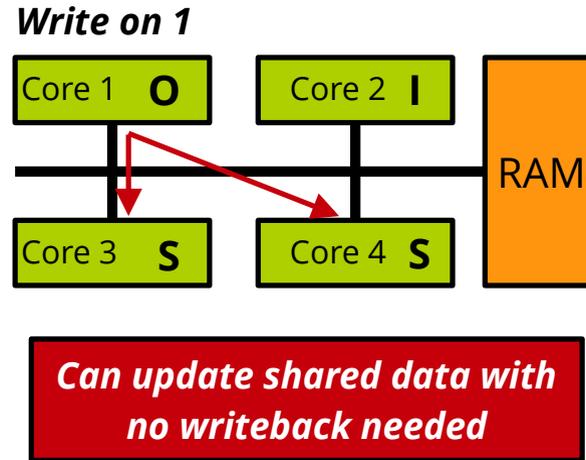
**Modified:** not in sync with memory only copy



**Owned:** not in sync with memory, other valid copies in S

- Split the M state into two **Modified** and **Owned**

# MOESI Protocol



- Owner has exclusive rights to make changes
  - Broadcast the changes to the shared copies
    - **No memory writeback needed**
  - Writeback only when data in O or M is evicted

# Summary

- Bus-based cache systems
  - can improve performance by reducing bus usage
  - 2 optimisations: MESI, MOESI
- Still, bus-based systems can't scale to large multiprocessor counts
- Next video: directory-based coherence systems