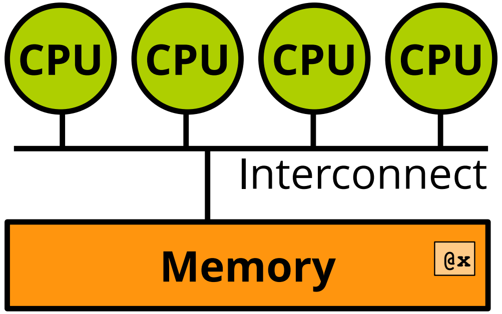
MANCHESTER
1824
The University of Manchester

COMP35112 Chip Multiprocessors

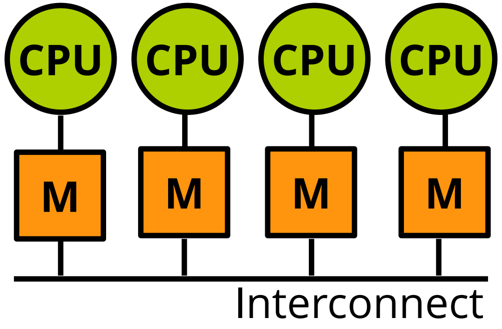
Shared Memory Multiprocessors

Pierre Olivier

Multiprocessor Structure



Shared Memory

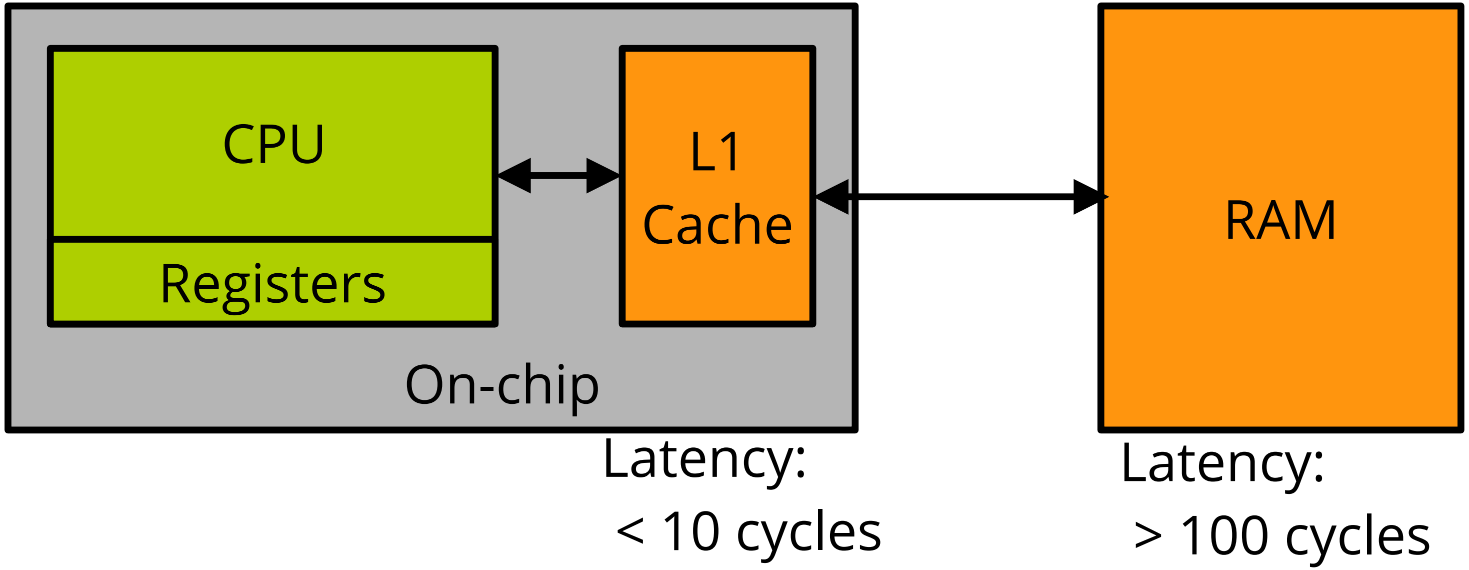


Distributed Memory

- Most general purpose multiprocessors are shared memory
  - **easier to program**, however hardware is **more complex**
- Let's study the scalability issues of cache coherence systems
  - Focusing on bus-based ones

Caches

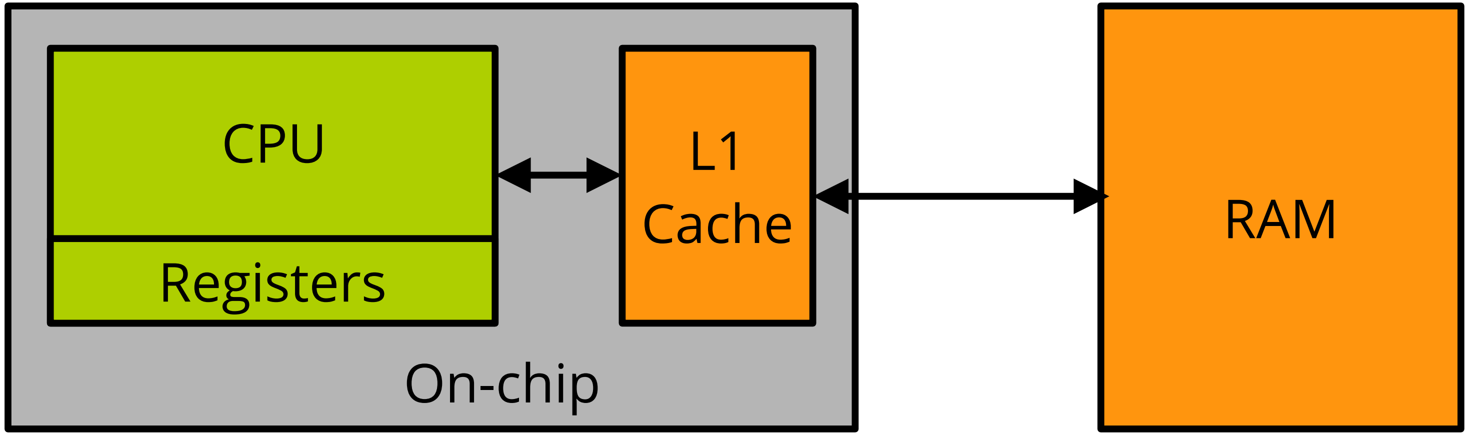
- A high performance uniprocessor:



- Cache: **fast** and **small** local memory holding recently used data and instruction

Caches

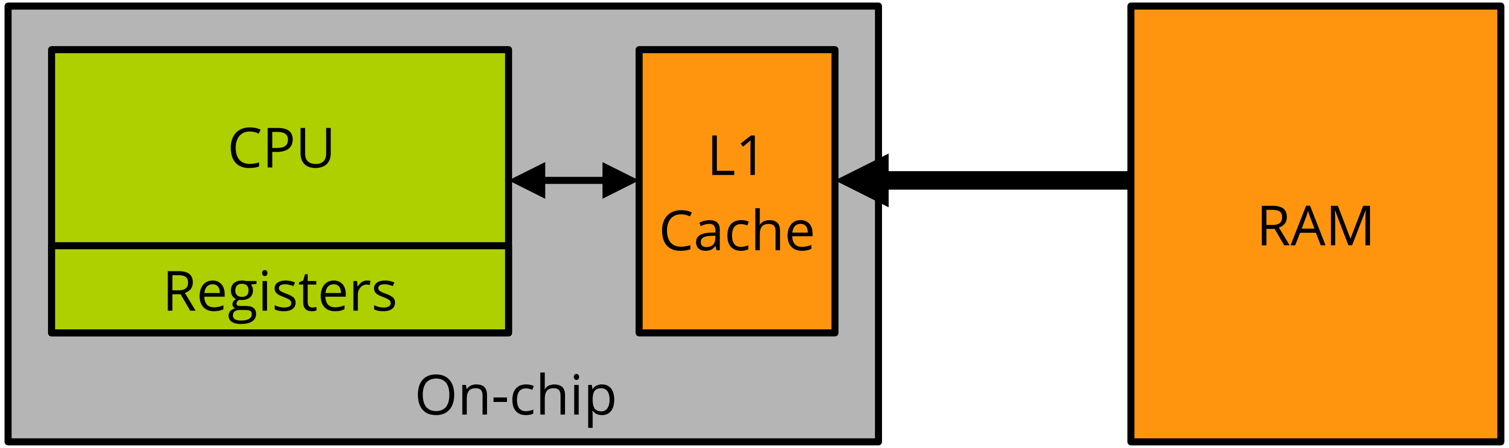
- A high performance uniprocessor:



- There may be multiple levels (L1, L2, L3)

Caches

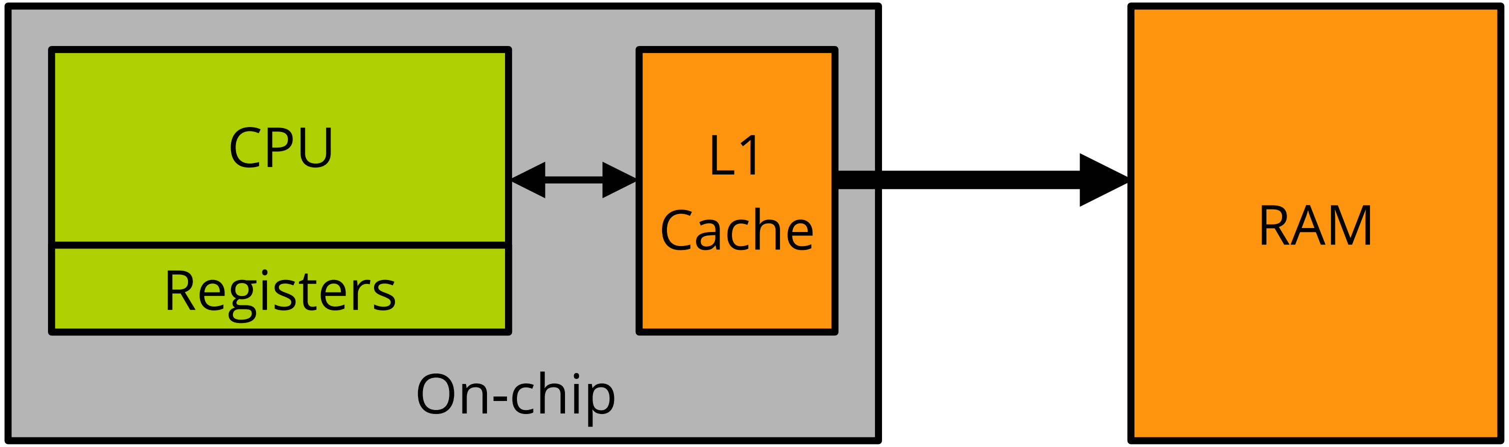
- A high performance uniprocessor:



Still need to fetch from RAM on cache misses

Caches

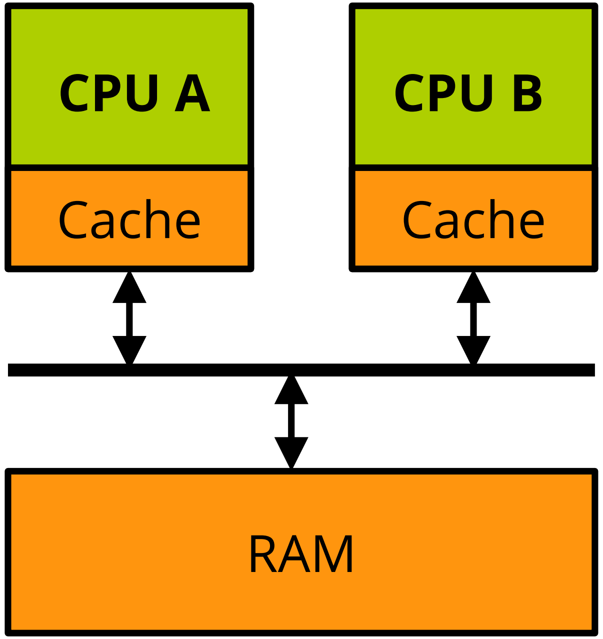
- A high performance uniprocessor:



Eventually need to write data back in RAM upon modifications

The Cache Coherency Problem

- What happens with multiprocessors?



CPU A

Cache

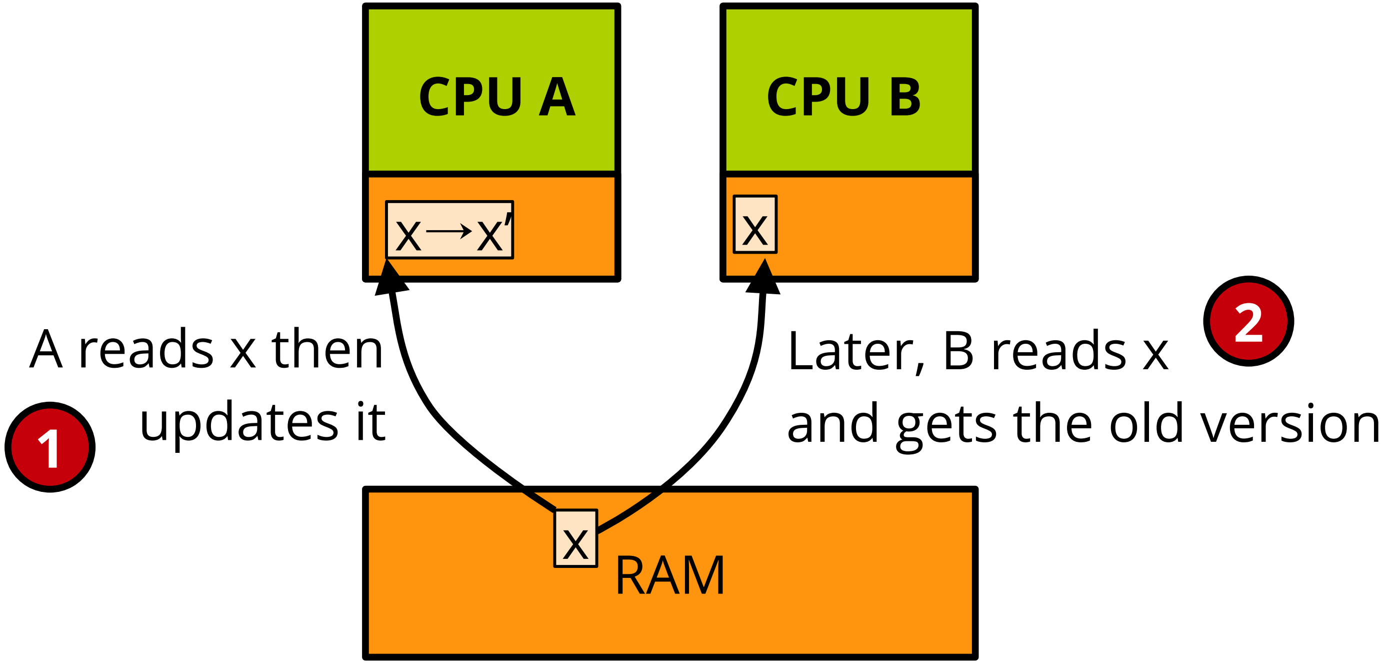
CPU B

Cache

RAM

The Cache Coherency Problem

- What happens with multiprocessors?



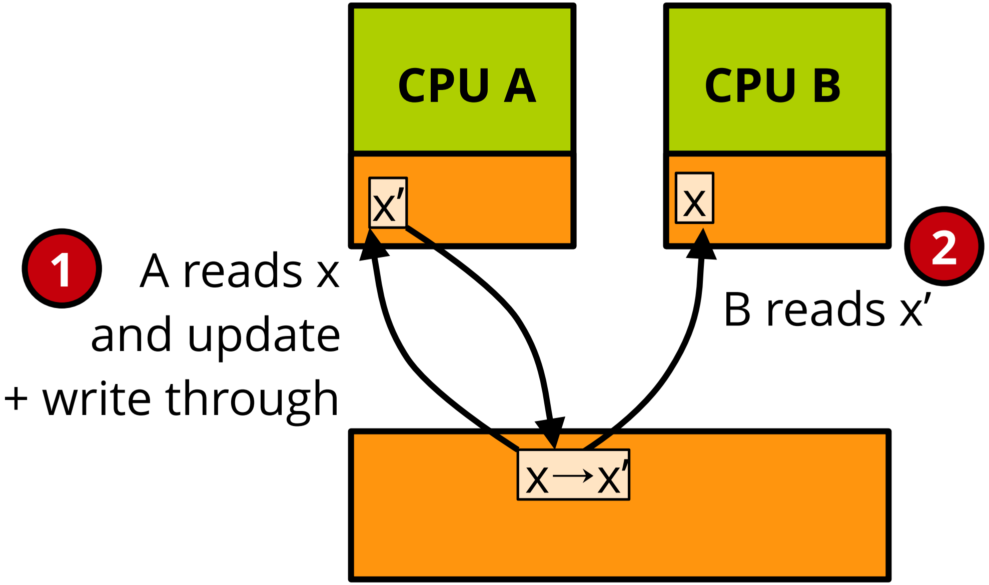
A reads x then updates it

Later, B reads x and gets the old version

We (probably) expect to get the latest value!

The Cache Coherency Problem

- Apparently obvious solution: 'write through' policy?
    - Every write is updated in memory
- Involves a lot of memory accesses, **negating cache benefits**



The Cache Coherency Problem

- **Cache-to-cache communication**?
- How to avoid separate cache copies, i.e. **how to maintain cache coherency?**
- It gets complex, we need to develop a model
    - Topic of the next lecture