COA

Chapter 04: | Elements of Cache Design

Prof. Dr. Mostafa Elhosseini

https://youtube.com/drmelhosseini

AGENDA

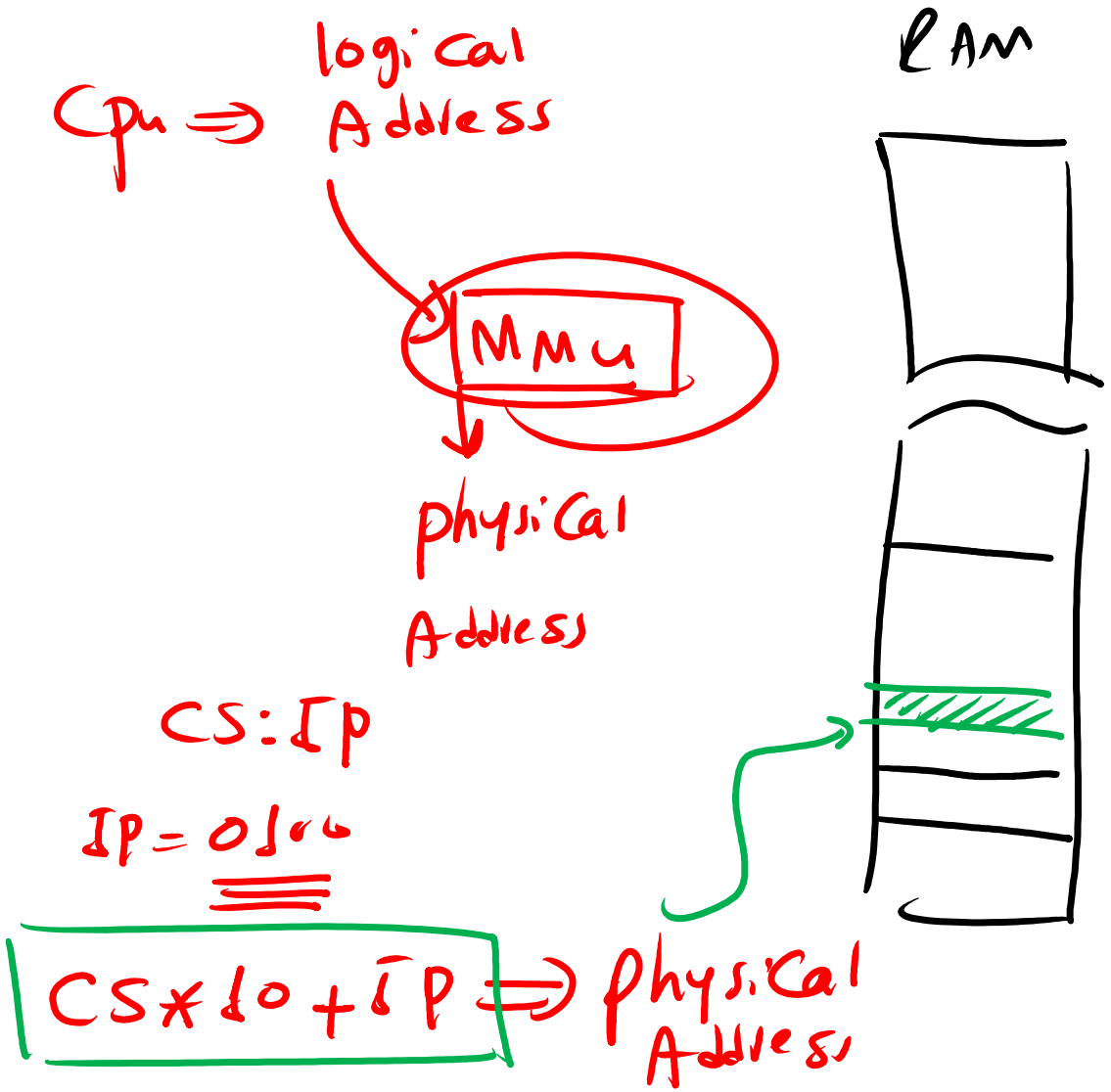
MMU

Elements of Cache Design

- Cache Addresses
  - Logical ⇒ Virtual Address
  - Physical
- Cache Size
- Mapping Function
  - Direct
  - Associative
  - Set Associative
- Replacement Algorithms
- Write Policy
- Line Size
- Number of caches

Virtual Memory

- **Virtual Memory** is a facility that allows programs to address memory from a logical point of view, without regard to the amount of main memory **physically available**
- The **address fields of machine instructions** contain **virtual addresses**.

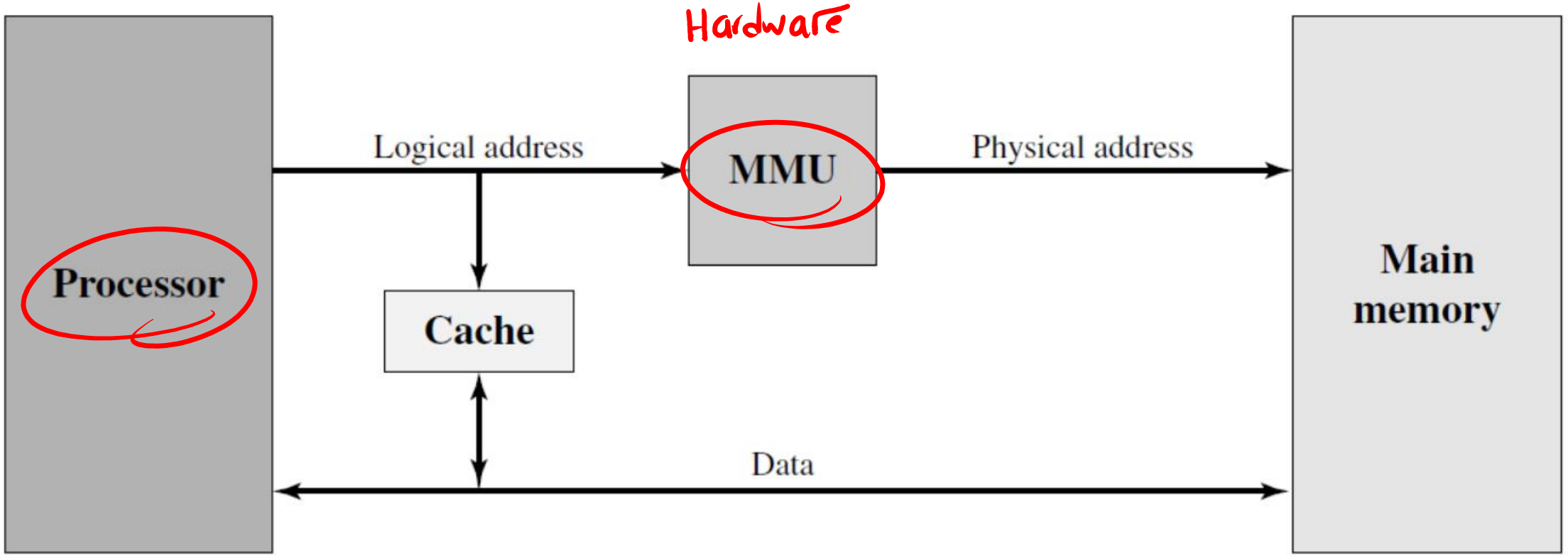


Cache Addresses

- **When virtual addresses are used**, the system designer may choose to **place the cache** between the processor and the MMU or between the MMU and main Memory
  - A **logical cache**, also known as a **virtual cache**, stores data using virtual addresses. The processor accesses the cache directly, without going through the MMU
  - A **physical cache** stores data using main memory physical addresses.

Cache Addresses – Logical Cache

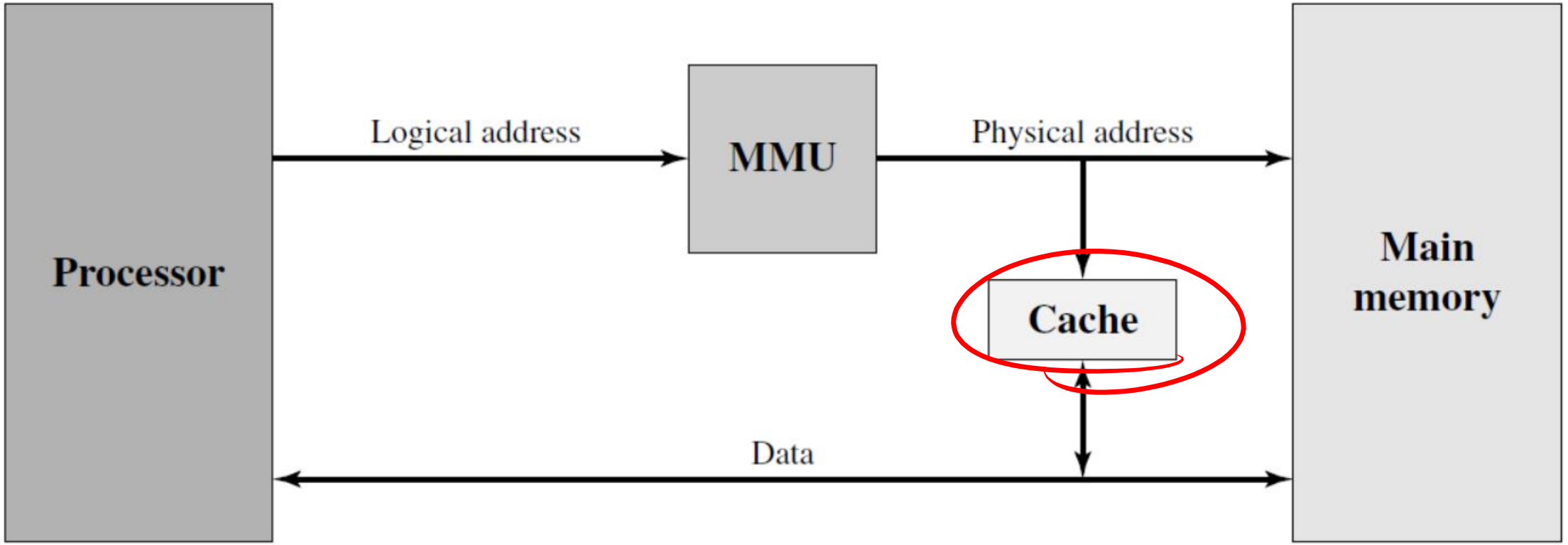
- Cache **access speed is faster** than for a physical cache, because the cache can respond before the MMU performs an address translation



Cache Addresses – Logical Cache

- Each application sees a virtual memory that starts at address 0. Thus, the same virtual address in two different applications refers to two different physical addresses.
- The cache memory must therefore be completely flushed with each application context switch, or extra bits must be added to each line of the cache to identify which virtual address space this address refers to

Cache Addresses – Physical Cache



Cache Size

Cost / bit ⇒ ↑

- **Small enough** so that the overall average **cost per bit** is **close** to that of **main memory** alone and
- **Large enough** so that the overall average *access time is close* to that of the **cache** alone

trade off

- The **larger the cache**, the larger the number of gates involved in addressing the cache. The result is that large caches tend to be slightly slower than small ones—even when built with the same integrated circuit technology and put in the same place on chip and circuit board
- The **available chip and board area** also **limits** cache size.
- Because the **performance of the cache** is very **sensitive** to the nature of the **workload**, it is **impossible** to arrive at a single "**optimum**" cache size