UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Memory Hierarchy
Heterogeneous Computing

Professor: Dr. Joel Fuentes - jfuentes@ubiobio.cl

Teaching Assistants:
- Daniel López - daniel.lopez1701@alumnos.ubiobio.cl
- Sebastián González - sebastian.gonzalez1801@alumnos.ubiobio.cl

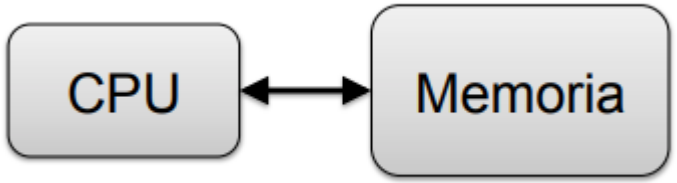
Course Website: http://www.face.ubiobio.cl/~jfuentes/classes/hc

Contents

- Basic concepts
- Memory hierarchy
- Caching

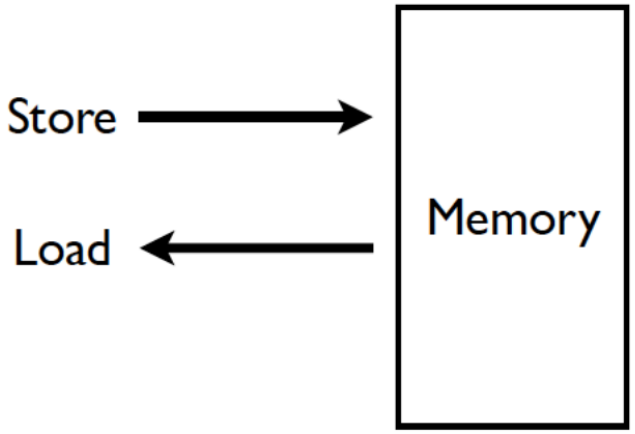
Basic concepts

- Efficiency limitations between CPU and memory are usually latency and bandwidth
- Latency: The time required for a single access
  - Memory access time >> Processor's cycle time
- Bandwidth: Number accesses per unit of time

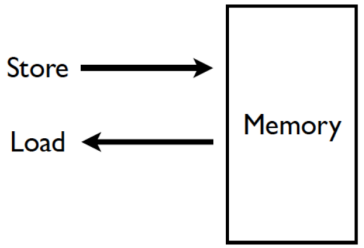


Basic concepts

- Programmer's view:

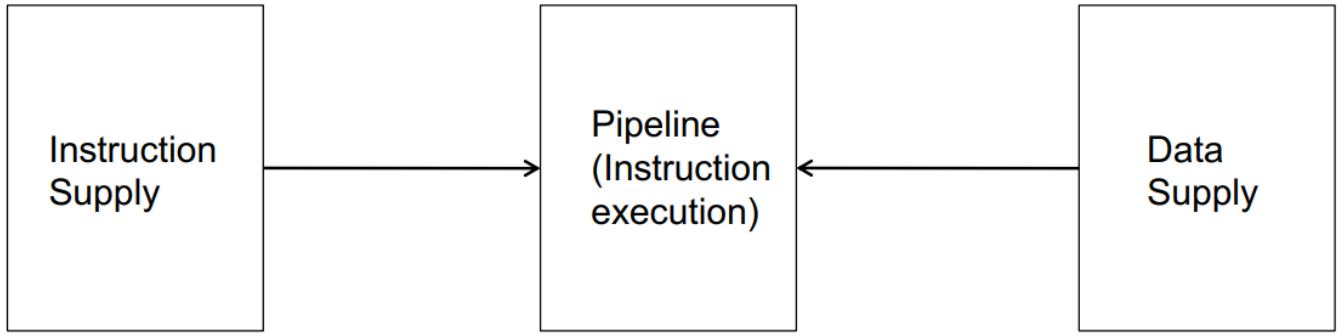


Virtual memory vs Physical memory



- A programmer sees virtual memory
  - And assumes an infinite amount of memory
- Reality: The size of the physical memory is much smaller than what the programmer assumes
- The system (software and hardware) maps virtual memory addresses to physical memory
  - This mapping is completely transparent to the programmer

In an ideal world…



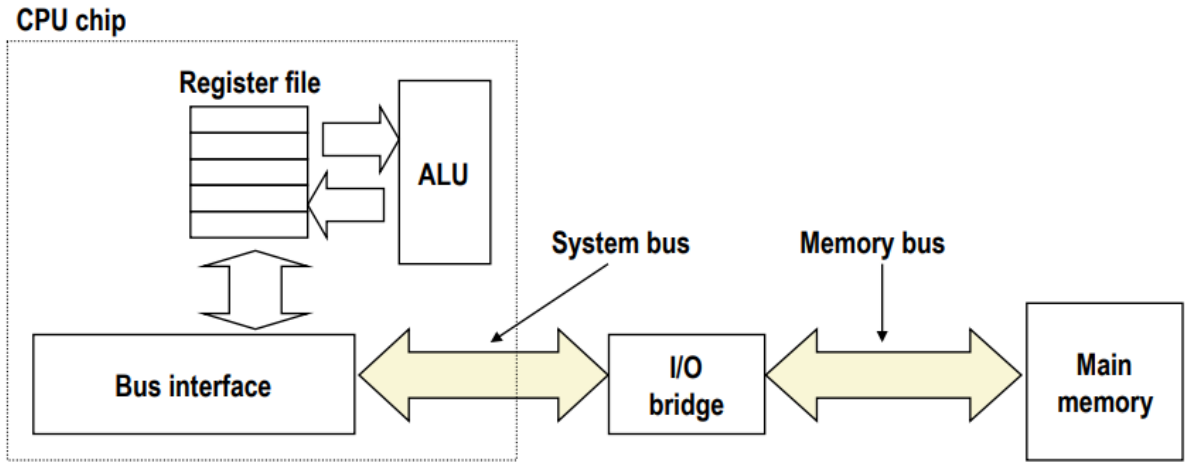
- Zero latency access
- Infinite capacity
- Zero cost
- Perfect control flow

- No pipeline stalls
- Perfect data flow (reg/memory dependencies)
- Zero-cycle interconnect (operand communication)
- Enough functional units
- Zero latency compute

- Zero latency access
- Infinite capacity
- Infinite bandwidth
- Zero cost

**Retrieved from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

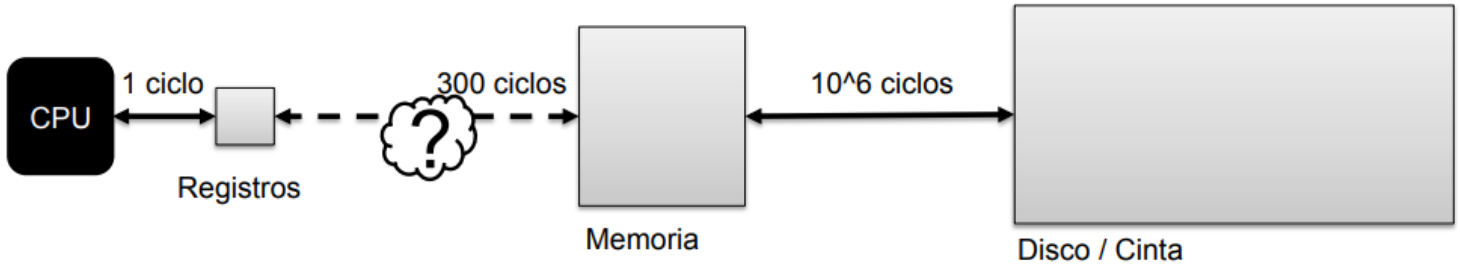
Memory hierarchy is born



- Idea: Have multiple storage tiers
- Progressively larger and slower as they get further away from the processor, but ensuring that the largest amount of data that the processor needs is at the fastest tiers.

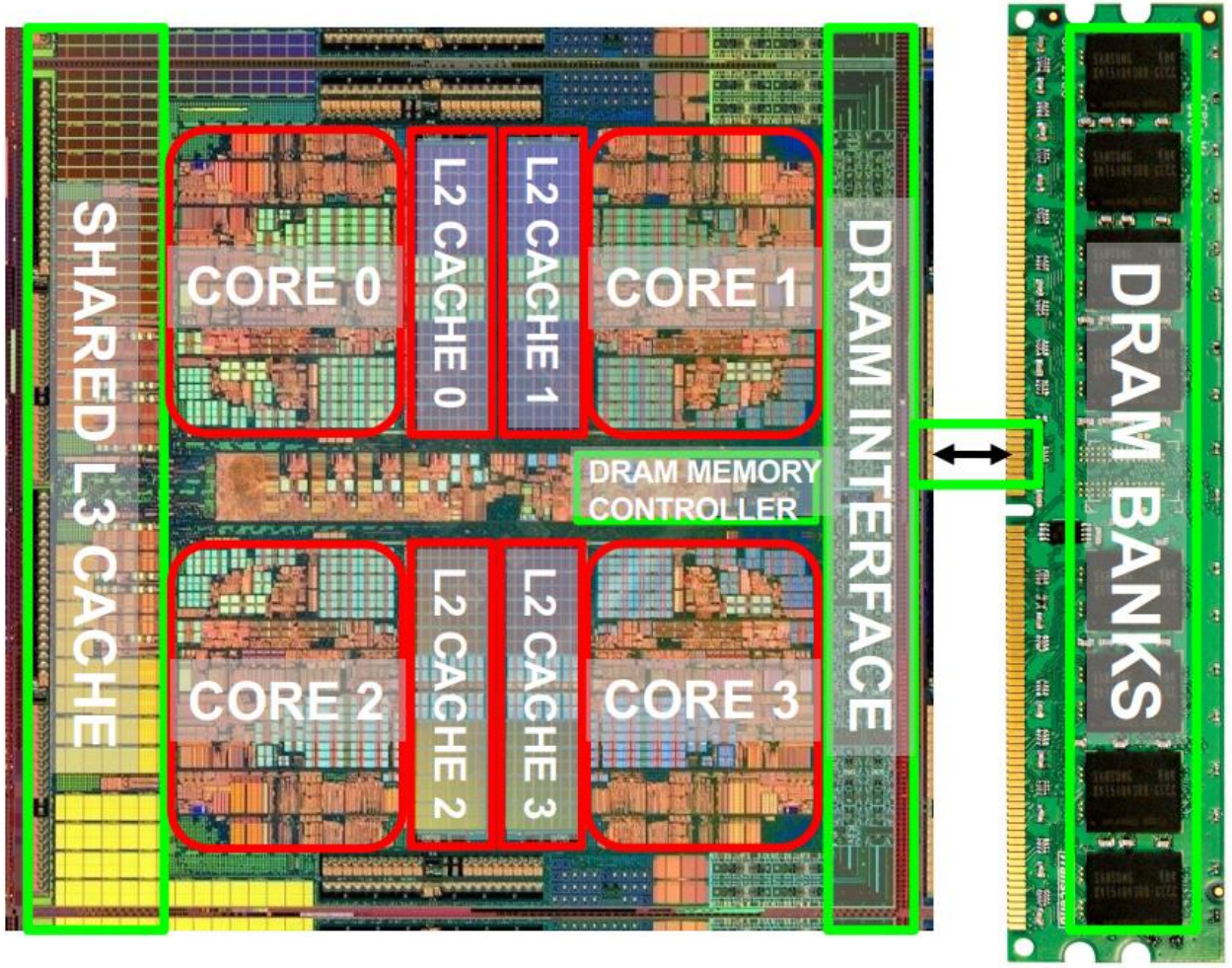
Memory hierarchy is born

- There is a long way (number of cycles) to read/write data between different memories.



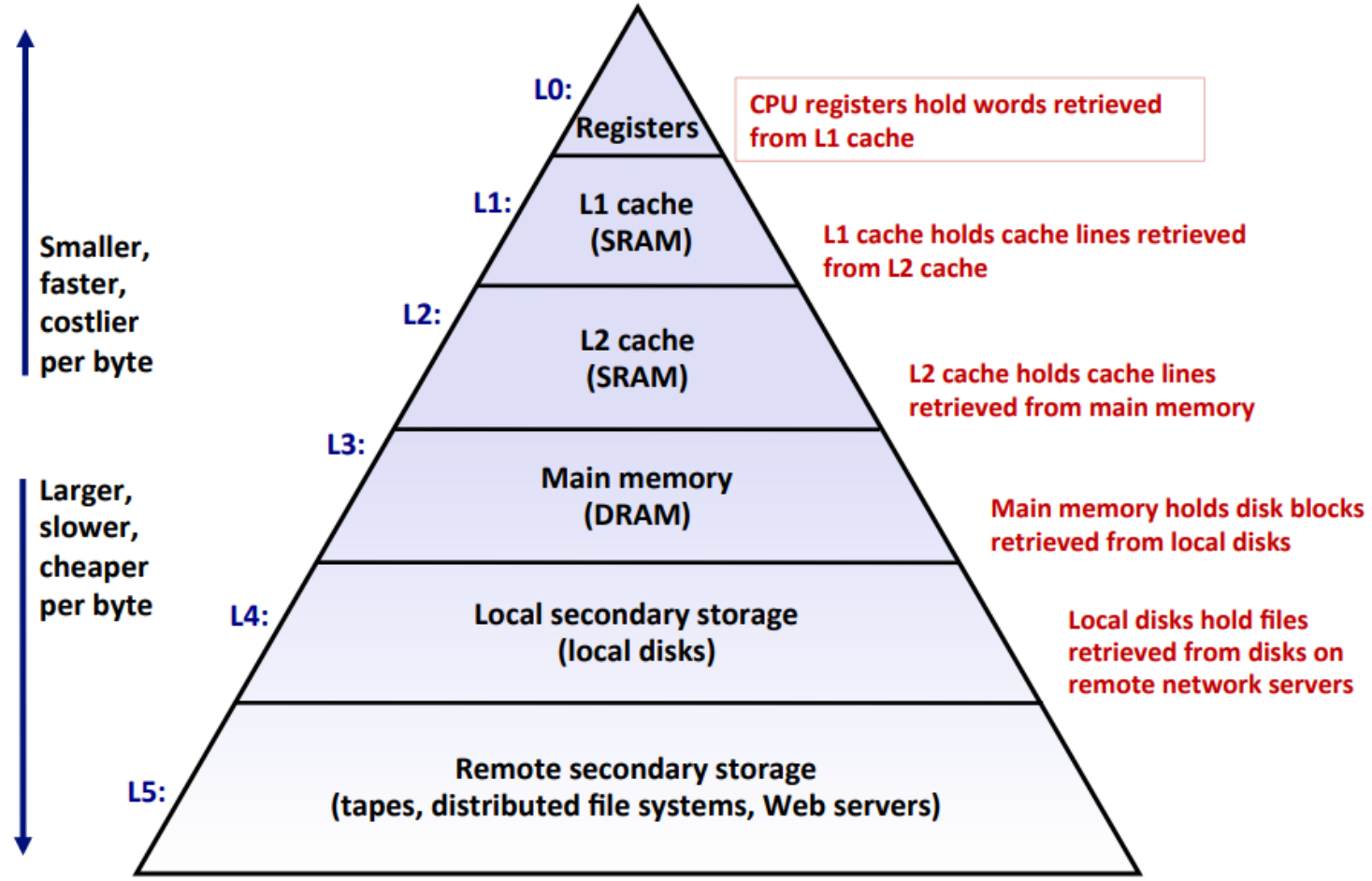
- What would improve this process?

Memory hierarchy



**Retrieved from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Memory hierarchy



** Retrieved from 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

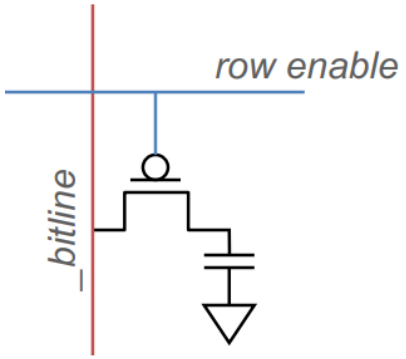
Main problems

- The larger the memory, the slower it is
  - As it grows larger, it is more complex to determine the location (physical memory address)
- Faster memories are more expensive
  - Technologies: SRAM vs DRAM vs Disk
- Higher bandwidth is more expensive
  - Need more memory banks, ports, higher frequency, faster technology, etc

** Retrieved from 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

DRAM

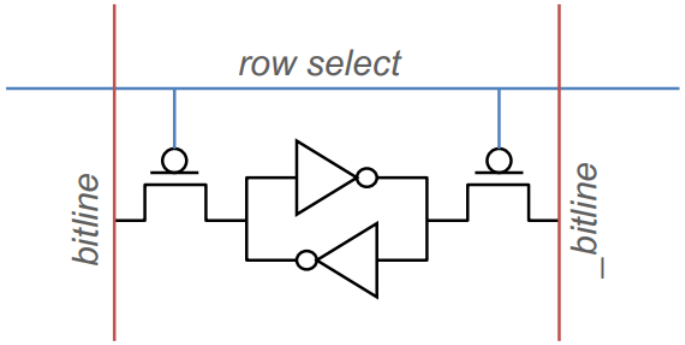
- Dynamic Random Access Memory
- The capacitor's state of charge indicates the stored value
  - Whether the capacitor is charged or discharged indicates a stored value of 1 or 0.
  - 1 capacitor
  - 1 access transistor
- Capacitor leaks through the RC path
  - DRAM cells lose charge over time
  - DRAM cells need to be refreshed



**Retreived from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

SRAM

- Static Random Access Memory
- Typically known as Cache Memory
- Two cross-coupled inverters store a single bit
  - Feedback path enables the stored value to persist in the cell
  - 4 transistors for storage
  - 2 transistors for access



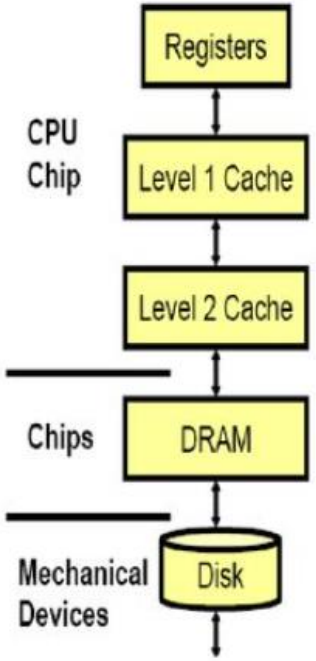
Retreived from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.

DRAM vs SRAM

- DRAM
  - Slower access (capacitor)
  - Higher density (1 Transistor 1 capacitor per cell)
  - Lower cost
  - Requires memory refreshing (energy, performance, circuitry)
  - Manufacturing requires putting capacitor and logic together
- SRAM
  - Faster access (no capacitor)
  - Lower density (6 transistors per cell)
  - Higher cost
  - Does not require memory refreshing
  - Manufacturing compatible with logic process (no capacitor)

**Retreived from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Memory hierarchy



	Access time	Capacity	Managed by
Registers	1 cycle	~500B	Software/compiler
Level 1 Cache	1-3 cycles	~64KB	hardware
Level 2 Cache	5-10 cycles	1-10MB	hardware
DRAM	~100 cycles	~10GB	Software/OS
Disk	10^6-10^7 cycles	~100GB	Software/OS

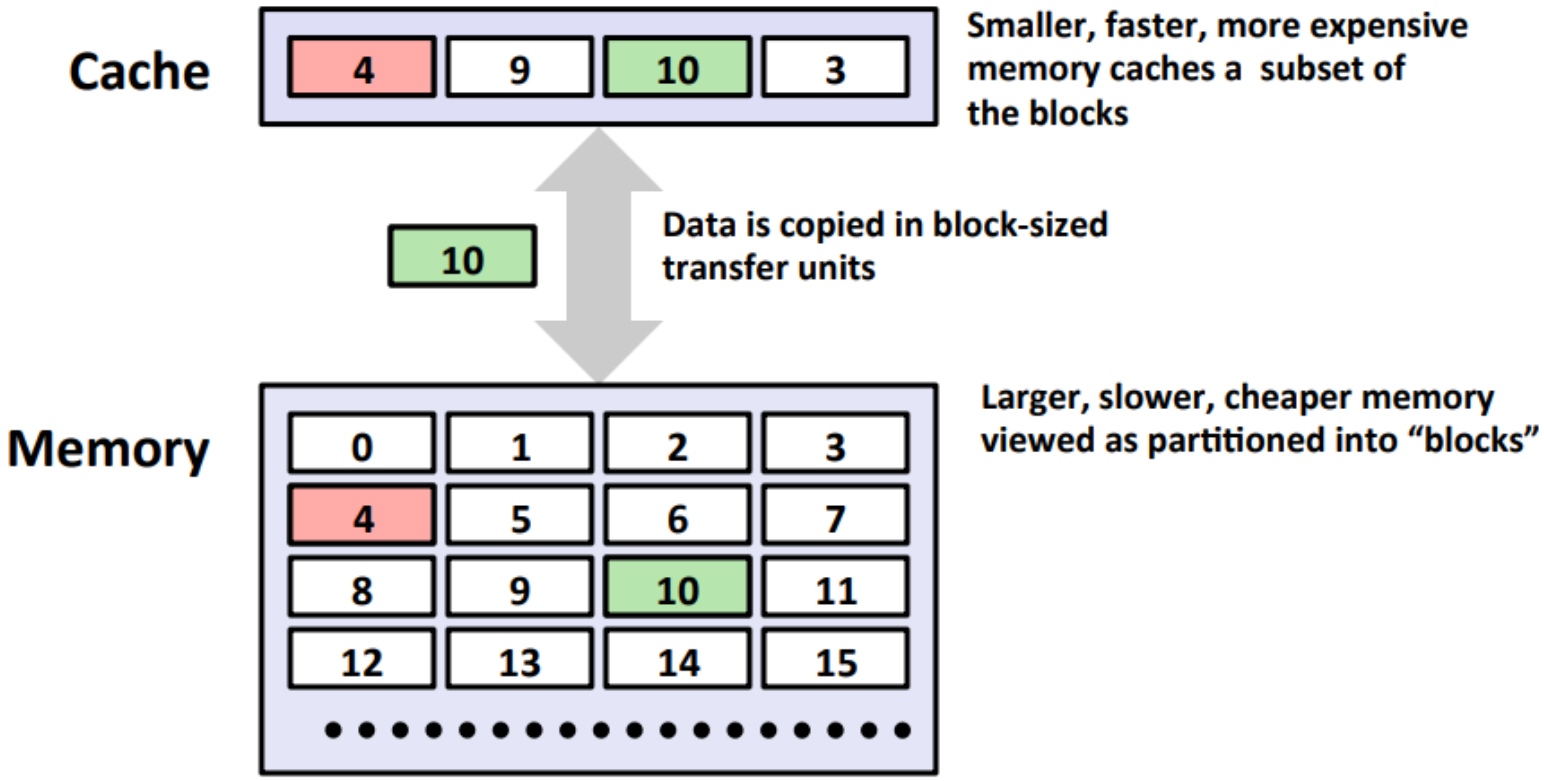
**Retreived from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Conceptos sobre Caching

- Block line: Unit of storage in the cache
  - Memory is logically divided into cache blocks that map to locations in the cache
- Hit: If the data is in cache, use cached data instead of accessing memory
- Miss: If the data is not in cache, bring block into cache
- Placement: Where and how to find a block in cache
- Replacement: What data to remove to make room in cache

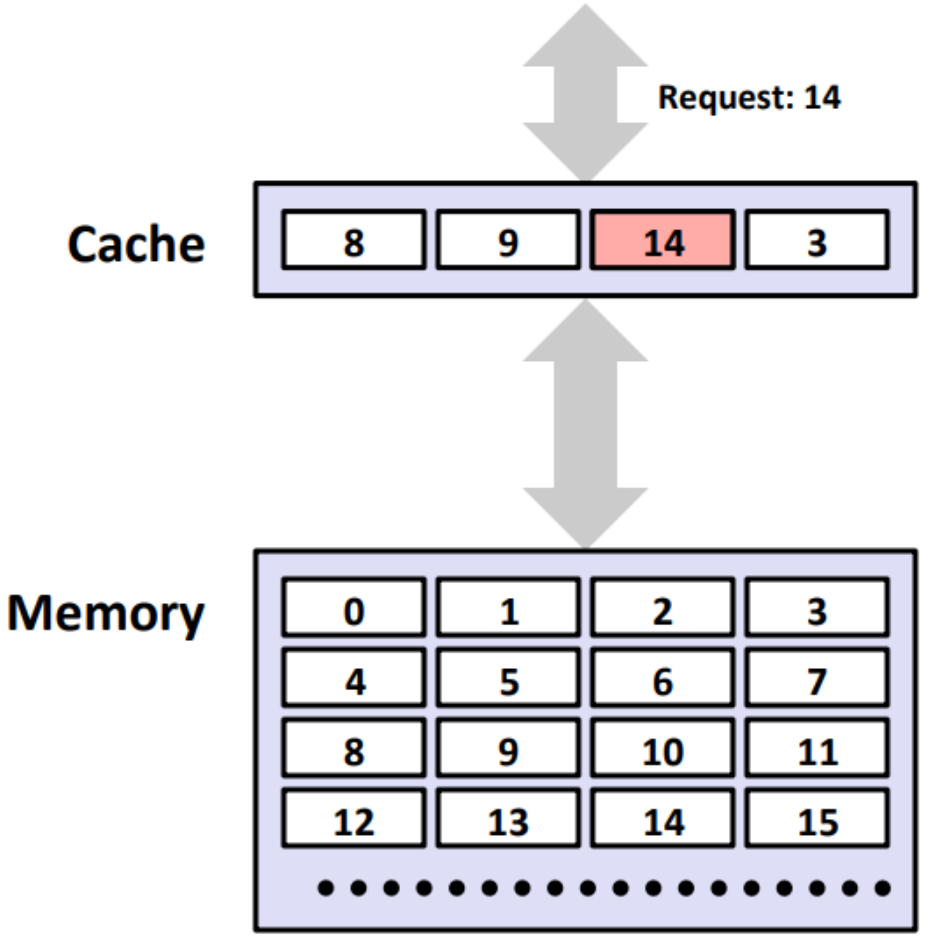
**Retreived from CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Caching



*Retreived from 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.

Caching: Hit

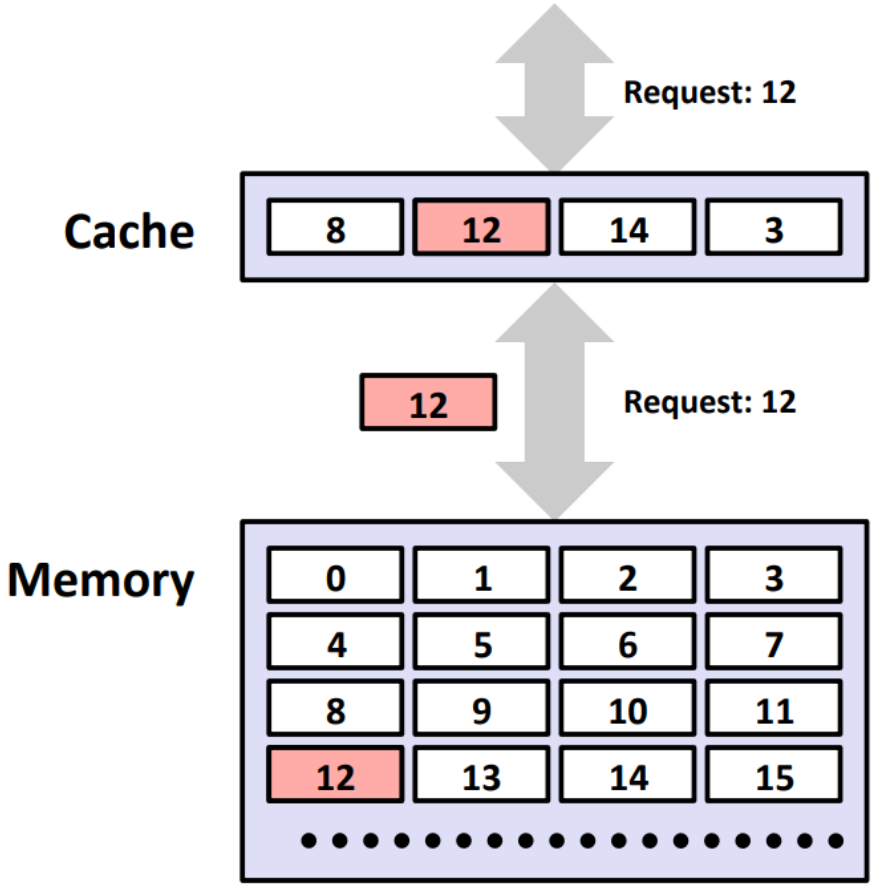


Data in block b is needed

Block b is in cache:
Hit!

**Retreived from 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

Caching: Miss



Data in block b is needed

Block b is not in cache:
Miss!

Block b is fetched from memory

Block b is stored in cache

- Placement policy: determines where b goes
- Replacement policy: determines which block gets evicted (victim)

**Retreived from 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*