

The Memory Hierarchy

15-213: Introduction to Computer Systems
11th Lecture, June 11, 2019

Instructor:

Brian Railing

Today

- **Storage technologies and trends**
- Locality of reference
- Caching in the memory hierarchy

Random-Access Memory (RAM)

■ Key features

- **RAM** is traditionally packaged as a chip.
- Basic storage unit is normally a **cell** (one bit per cell).
- Multiple RAM chips form a memory.

■ RAM comes in two varieties:

- SRAM (Static RAM)
- DRAM (Dynamic RAM)

SRAM vs DRAM Summary

	Trans. per bit	Access time	Needs refresh?	Needs EDC?	Cost	Applications
SRAM	4 or 6	1X	No	Maybe	100x	Cache memories
DRAM	1	10X	Yes	Yes	1X	Main memories, frame buffers

Enhanced DRAMs

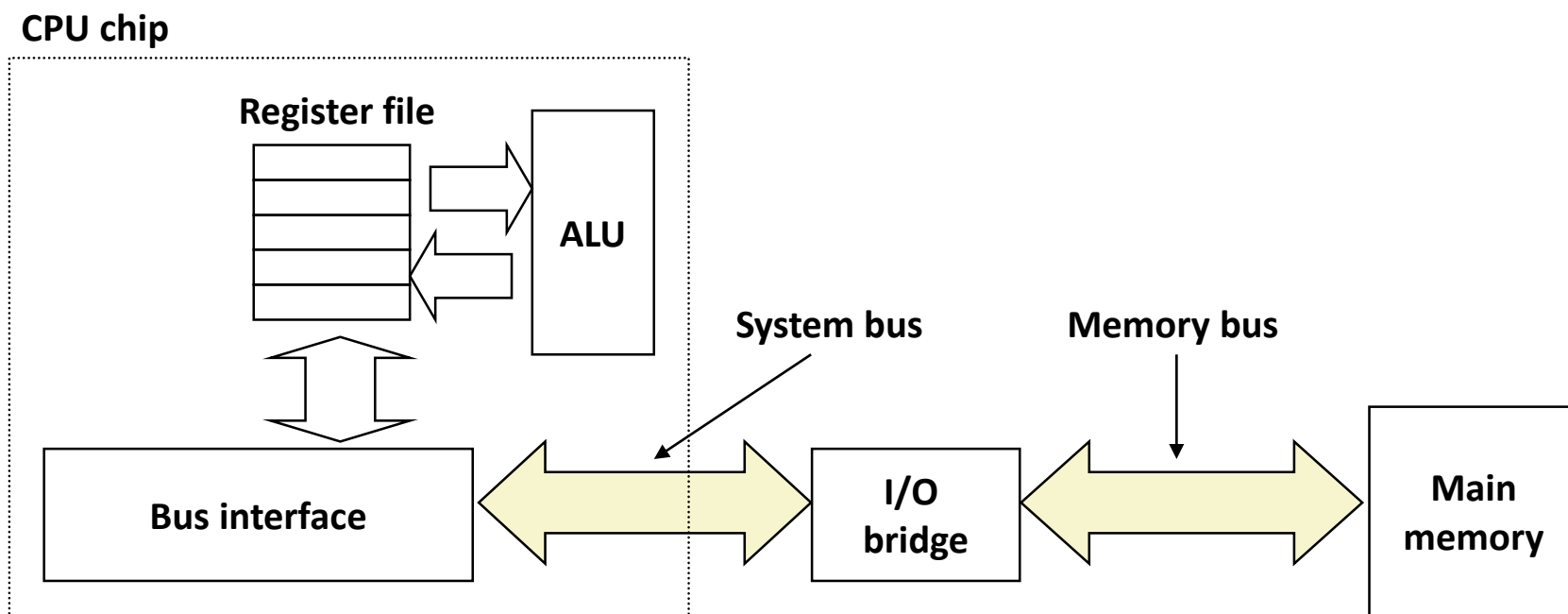
- **Basic DRAM cell has not changed since its invention in 1966.**
 - Commercialized by Intel in 1970.
- **DRAM cores with better interface logic and faster I/O :**
 - Synchronous DRAM (**SDRAM**)
 - Uses a conventional clock signal instead of asynchronous control
 - Allows reuse of the row addresses (e.g., RAS, CAS, CAS, CAS)
 - Double data-rate synchronous DRAM (**DDR SDRAM**)
 - Double edge clocking sends two bits per cycle per pin
 - Different types distinguished by size of small prefetch buffer:
 - **DDR** (2 bits), **DDR2** (4 bits), **DDR3** (8 bits)
 - By 2010, standard for most server and desktop systems
 - Intel Core i7 supports DDR3 and DDR4 SDRAM

Nonvolatile Memories

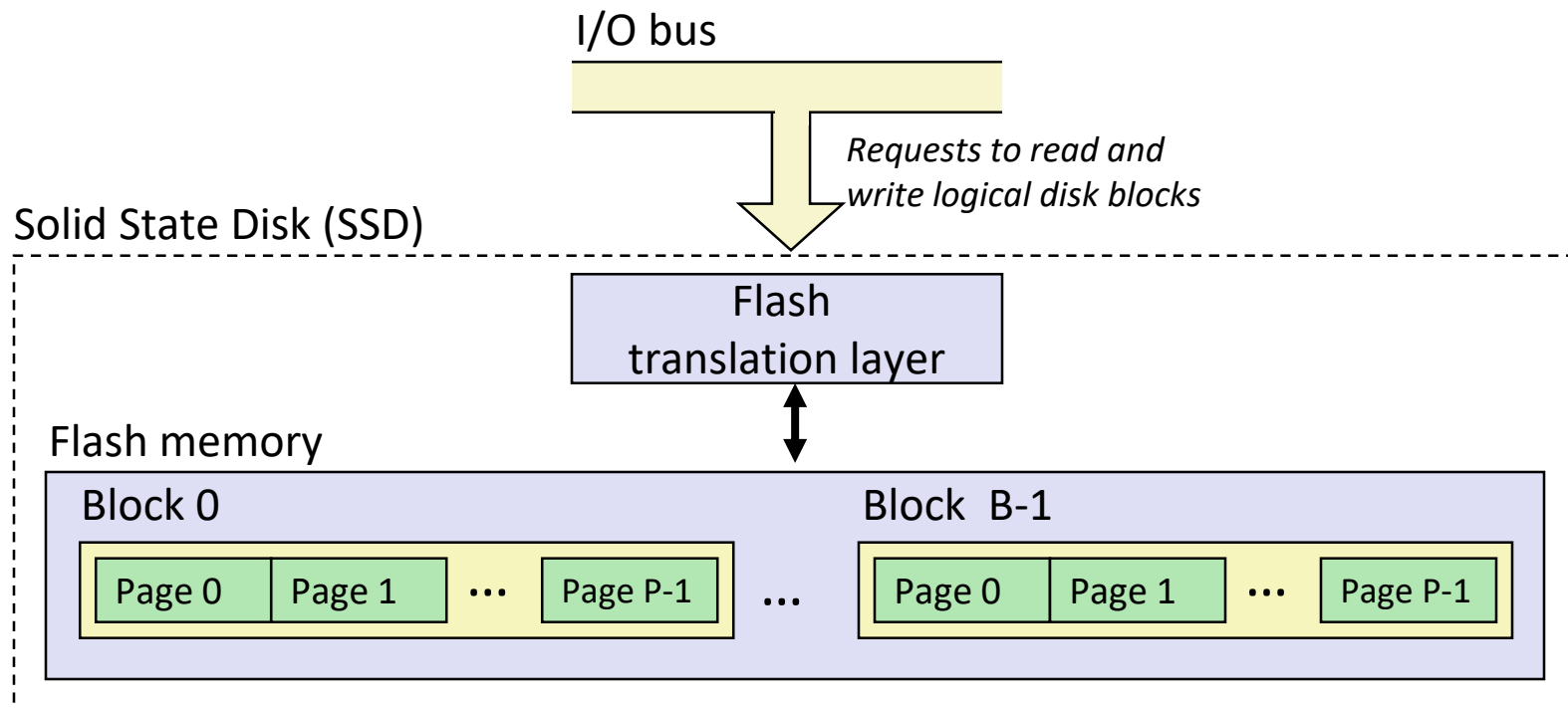
- **DRAM and SRAM are volatile memories**
 - Lose information if powered off.
- **Nonvolatile memories retain value even if powered off**
 - Read-only memory (**ROM**): programmed during production
 - Programmable ROM (**PROM**): can be programmed once
 - Erasable PROM (**EPROM**): can be bulk erased (UV, X-Ray)
 - Electrically erasable PROM (**EEPROM**): electronic erase capability
 - Flash memory: EEPROMs. with partial (block-level) erase capability
 - Wears out after about 100,000 erasings
- **Uses for Nonvolatile Memories**
 - Firmware programs stored in a ROM (BIOS, controllers for disks, network cards, graphics accelerators, security subsystems,...)
 - Solid state disks (replace rotating disks in thumb drives, smart phones, mp3 players, tablets, laptops,...)
 - Disk caches

Traditional Bus Structure Connecting CPU and Memory

- A **bus** is a collection of parallel wires that carry address, data, and control signals.
- Buses are typically shared by multiple devices.



Solid State Disks (SSDs)



- **Pages: 512KB to 4KB, Blocks: 32 to 128 pages**
- **Data read/written in units of pages.**
- **Page can be written only after its block has been erased**
- **A block wears out after about 100,000 repeated writes.**

SSD Performance Characteristics

Sequential read tput	550 MB/s	Sequential write tput	470 MB/s
Random read tput	365 MB/s	Random write tput	303 MB/s
Avg seq read time	50 us	Avg seq write time	60 us

- **Sequential access faster than random access**
 - Common theme in the memory hierarchy
- **Random writes are somewhat slower**
 - Erasing a block takes a long time (~1 ms)
 - Modifying a block page requires all other pages to be copied to new block
 - In earlier SSDs, the read/write gap was much larger.

Source: Intel SSD 730 product specification.

SSD Tradeoffs vs Rotating Disks

■ Advantages

- No moving parts → faster, less power, more rugged

■ Disadvantages

- Have the potential to wear out
 - Mitigated by “wear leveling logic” in flash translation layer
 - E.g. Intel SSD 730 guarantees 128 petabyte (128×10^{15} bytes) of writes before they wear out
- In 2015, about 30 times more expensive per byte

■ Applications

- MP3 players, smart phones, laptops
- Beginning to appear in desktops and servers

The CPU-Memory Gap

The gap *widens* between DRAM, disk, and CPU speeds.

Locality to the Rescue!

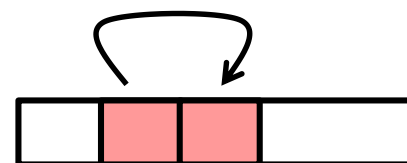
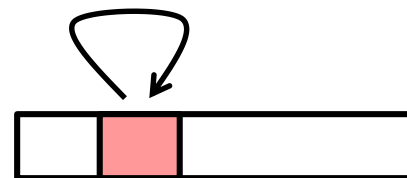
The key to bridging this CPU-Memory gap is a fundamental property of computer programs known as **locality**

Today

- Storage technologies and trends
- **Locality of reference**
- Caching in the memory hierarchy

Locality

- **Principle of Locality:** Programs tend to use data and instructions with addresses near or equal to those they have used recently
- **Temporal locality:**
 - Recently referenced items are likely to be referenced again in the near future
- **Spatial locality:**
 - Items with nearby addresses tend to be referenced close together in time



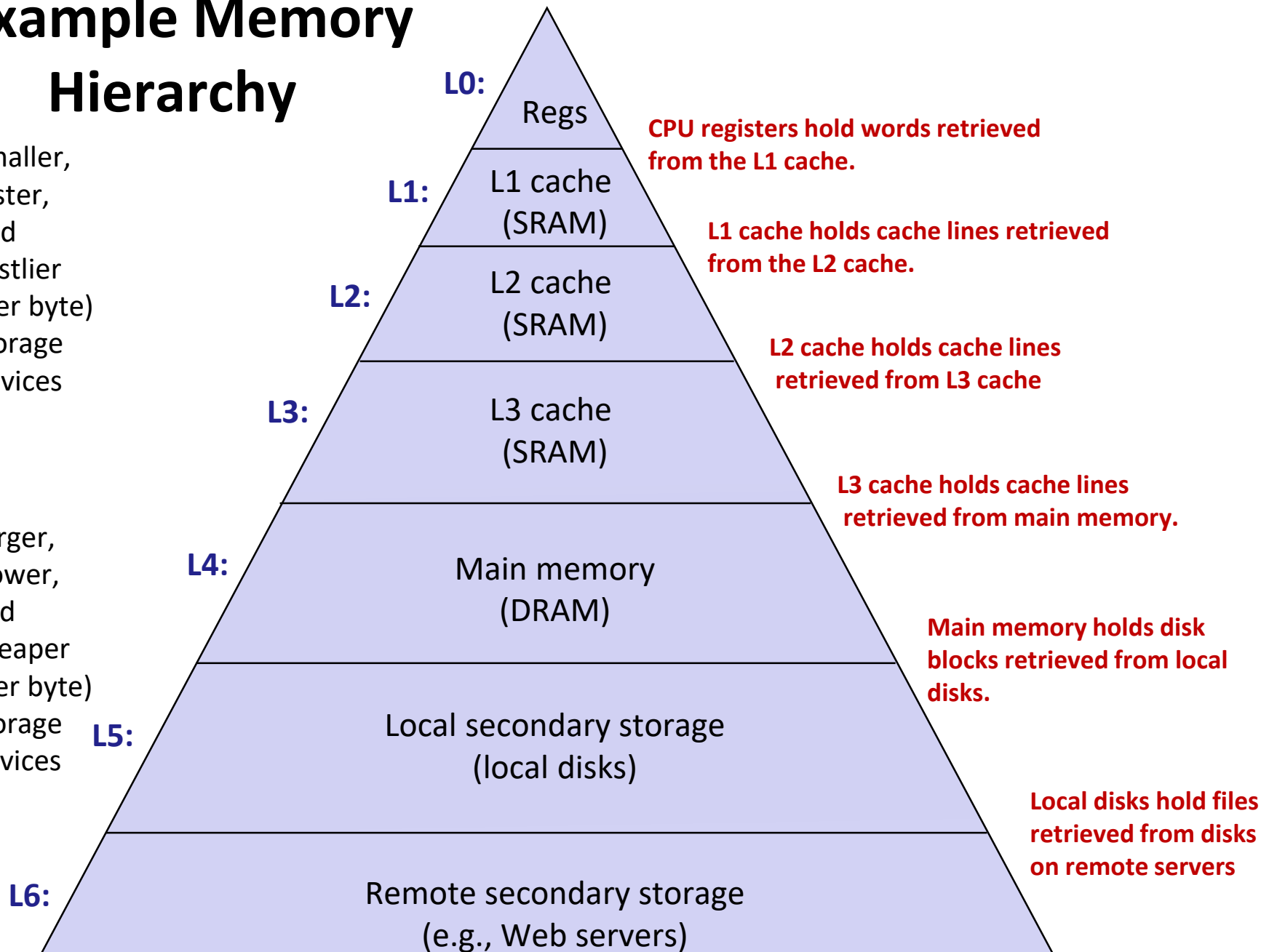
Memory Hierarchies

- **Some fundamental and enduring properties of hardware and software:**
 - Fast storage technologies cost more per byte, have less capacity, and require more power (heat!).
 - The gap between CPU and main memory speed is widening.
 - Well-written programs tend to exhibit good locality.
- **These fundamental properties complement each other beautifully.**
- **They suggest an approach for organizing memory and storage systems known as a **memory hierarchy**.**

Example Memory Hierarchy

Smaller,
faster,
and
costlier
(per byte)
storage
devices

Larger,
slower,
and
cheaper
(per byte)
storage
devices



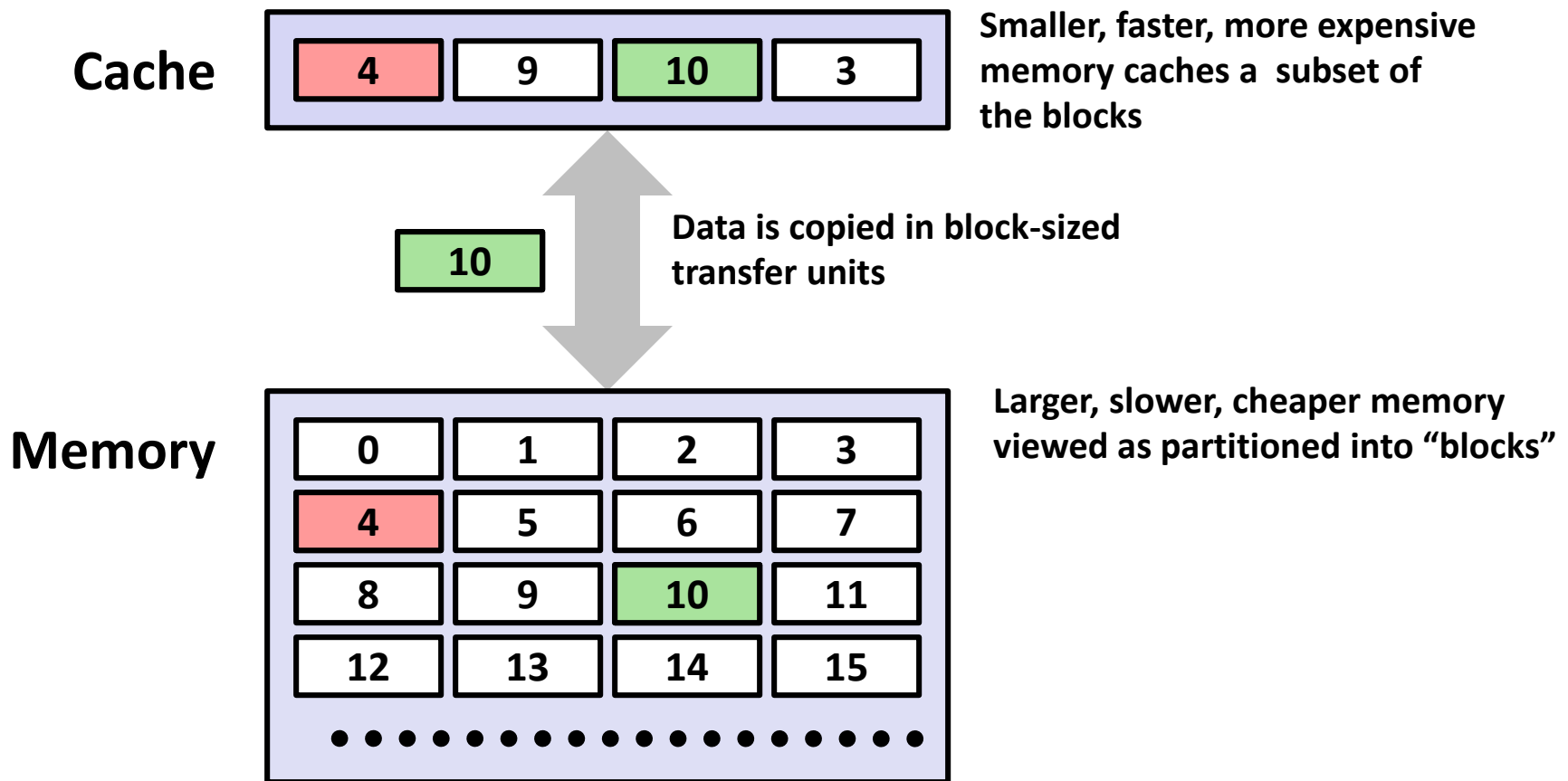
Today

- Storage technologies and trends
- Locality of reference
- **Caching in the memory hierarchy**

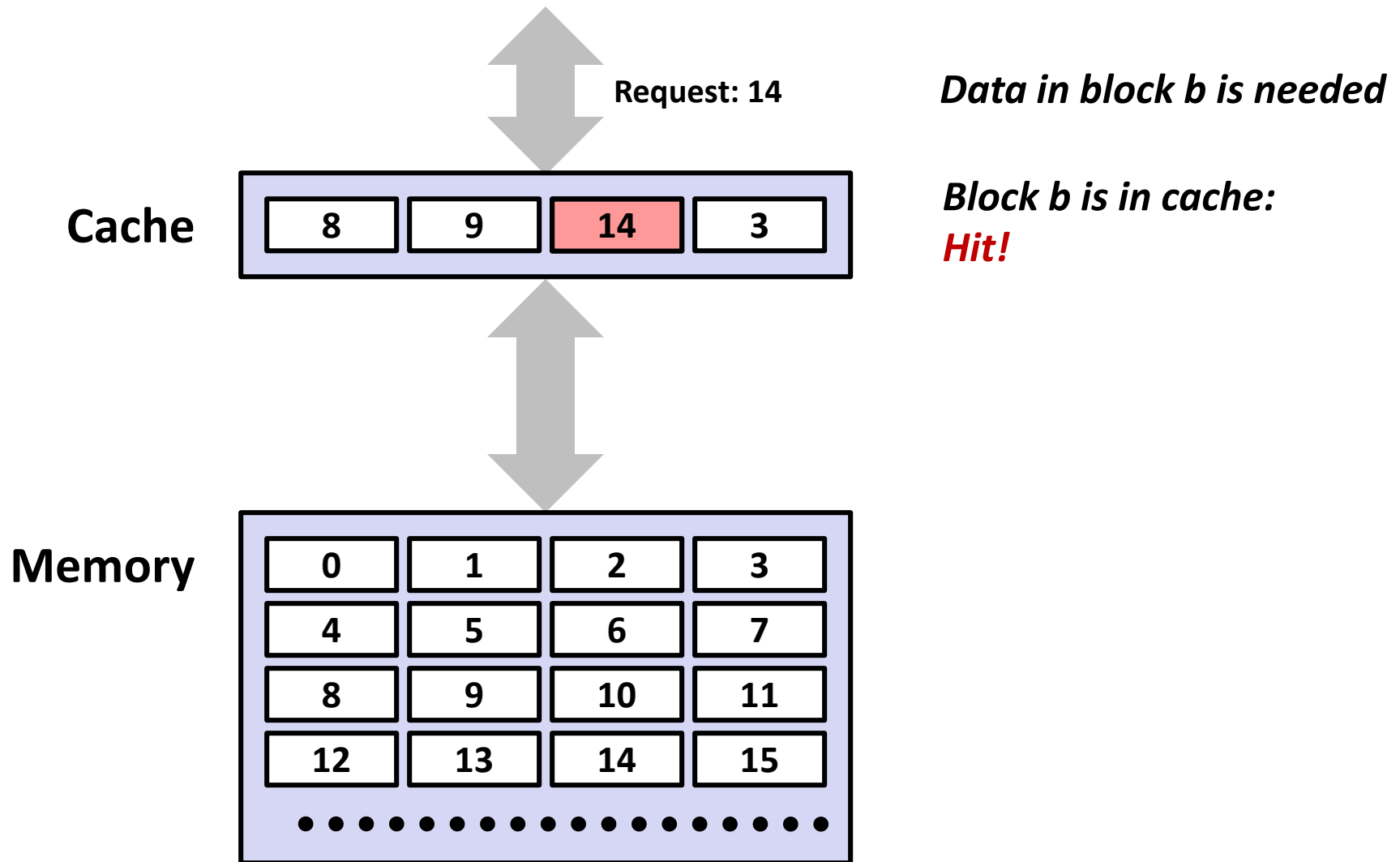
Caches

- **Cache:** A smaller, faster storage device that acts as a staging area for a subset of the data in a larger, slower device.
- **Fundamental idea of a memory hierarchy:**
 - For each k , the faster, smaller device at level k serves as a cache for the larger, slower device at level $k+1$.
- **Why do memory hierarchies work?**
 - Because of locality, programs tend to access the data at level k more often than they access the data at level $k+1$.
 - Thus, the storage at level $k+1$ can be slower, and thus larger and cheaper per bit.
- **Big Idea:** The memory hierarchy creates a large pool of storage that costs as much as the cheap storage near the bottom, but that serves data to programs at the rate of the fast storage near the top.

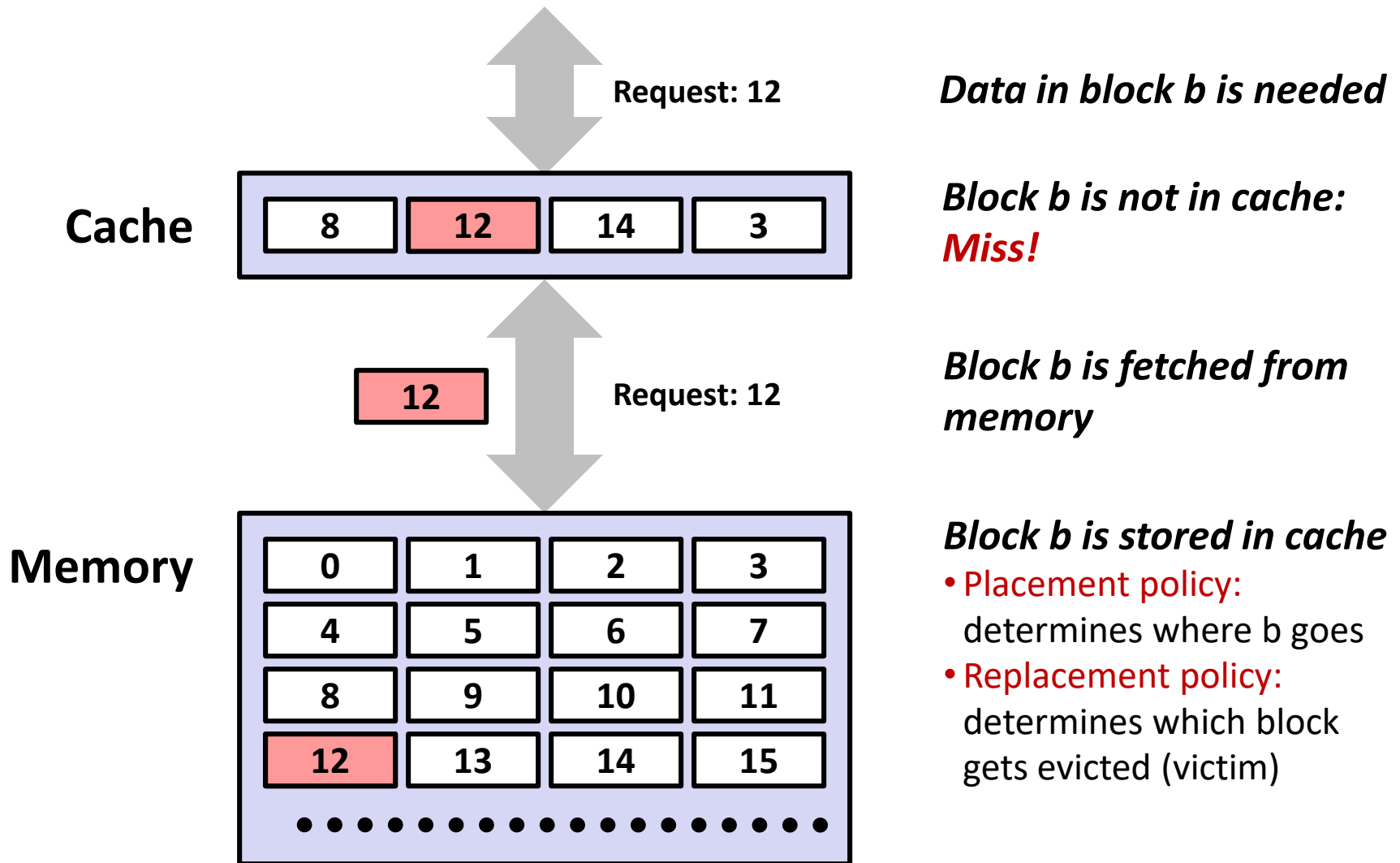
General Cache Concepts



General Cache Concepts: Hit



General Cache Concepts: Miss



General Caching Concepts:

Types of Cache Misses

■ Cold (compulsory) miss

- Cold misses occur because the cache is empty.

■ Conflict miss

- Most caches limit blocks at level $k+1$ to a small subset (sometimes a singleton) of the block positions at level k .
 - E.g. Block i at level $k+1$ must be placed in block $(i \bmod 4)$ at level k .
- Conflict misses occur when the level k cache is large enough, but multiple data objects all map to the same level k block.
 - E.g. Referencing blocks 0, 8, 0, 8, 0, 8, ... would miss every time.

■ Capacity miss

- Occurs when the set of active cache blocks (**working set**) is larger than the cache.

Examples of Caching in the Mem. Hierarchy

Cache Type	What is Cached?	Where is it Cached?	Latency (cycles)	Managed By
Registers	4-8 bytes words	CPU core	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware MMU
L1 cache	64-byte blocks	On-Chip L1	4	Hardware
L2 cache	64-byte blocks	On-Chip L2	10	Hardware
Virtual Memory	4-KB pages	Main memory	100	Hardware + OS
Buffer cache	Parts of files	Main memory	100	OS
Disk cache	Disk sectors	Disk controller	100,000	Disk firmware
Network buffer cache	Parts of files	Local disk	10,000,000	NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

Summary

- **The speed gap between CPU, memory and mass storage continues to widen.**
- **Well-written programs exhibit a property called *locality*.**
- **Memory hierarchies based on *caching* close the gap by exploiting locality.**