Introduction to Memory Hierarchy and Cache

Joannah Nanjekye

July 18, 2024

Characteristics of Memory

Location
- Internal (e.g., processor registers, cache, main memory)
- External (e.g., optical disks, magnetic disks, tapes)

Capacity
- Number of words
- Number of bytes

Unit of Transfer
- Word
- Block

Access Method
- Sequential
- Direct
- Random
- Associative

Performance
- Access time
- Cycle time
- Transfer rate

Physical Type
- Semiconductor
- Magnetic
- Optical
- Magneto-optical

Physical Characteristics
- Volatile/nonvolatile
- Erasable/nonerasable

Organization
- Memory modules

Memory Performance

- **Access time (latency):** time it takes to perform a read or write operation (random-access) or time it takes to position the read-write mechanism at the desired location (non-random-access)
- **Memory cycle time:** access time plus any additional time required before a second access can commence
- **Transfer rate:** rate at which data can be transferred into or out of a memory unit

$$T_n = T_A + \frac{n}{R}$$

where

$$T_n = \text{Average time to read or write } n \text{ bits}$$

$$T_A = \text{Average access time}$$

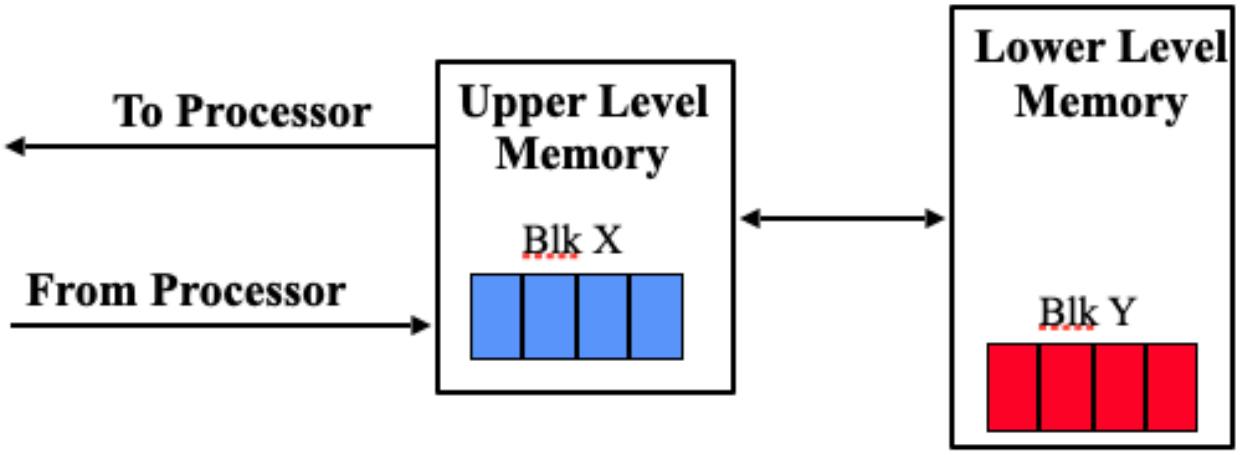
$$n = \text{Number of bits}$$

$$R = \text{Transfer rate, in bits per second (bps)}$$

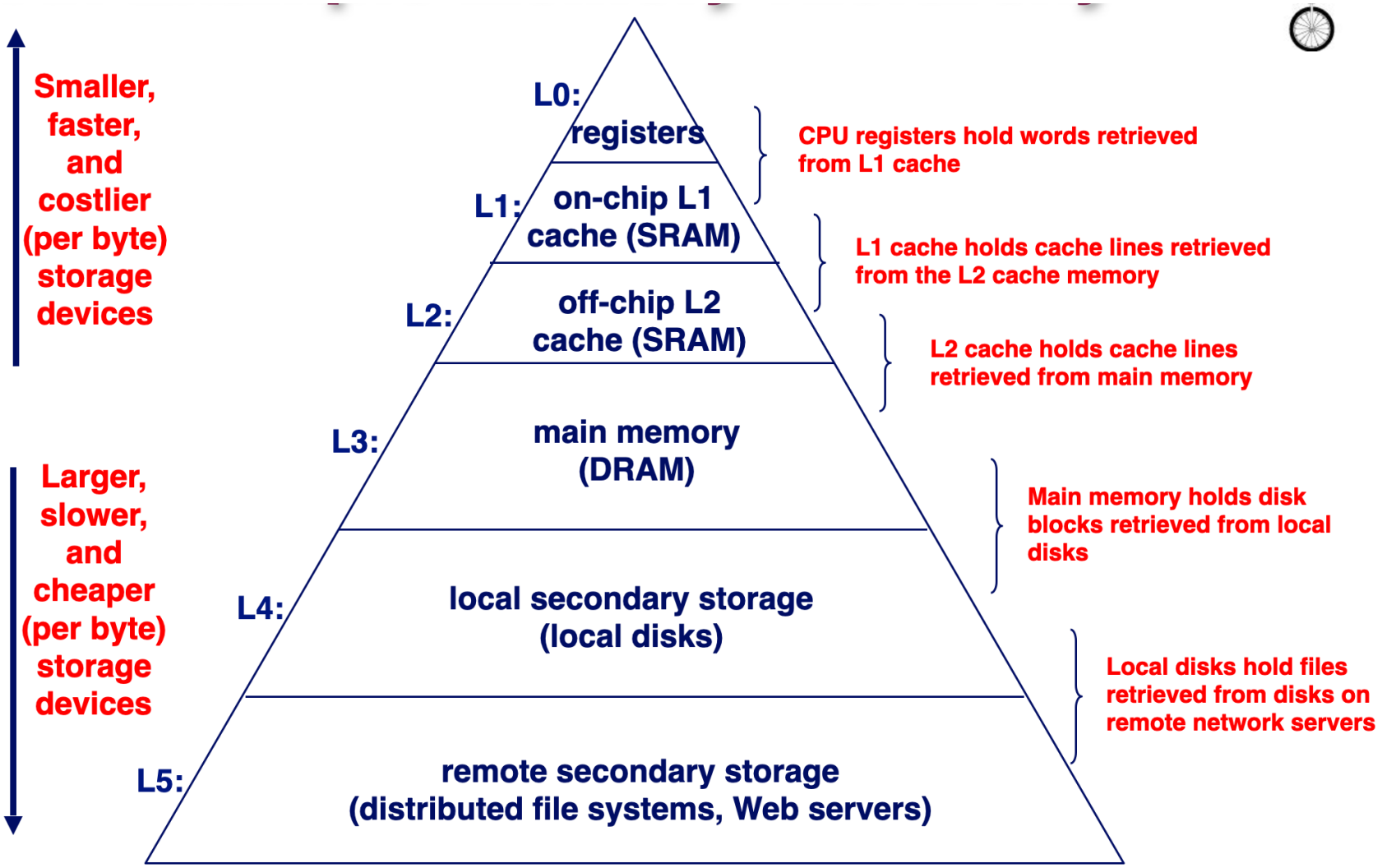
The Locality Principle

Programs tend to reuse data and instructions near those they have used or referenced recently

- **Spatial locality:** items with nearby addresses tend to be referenced close together in time
- **Temporal locality:** recently referenced items are likely to be referenced in the near future



Memory Hierarchy

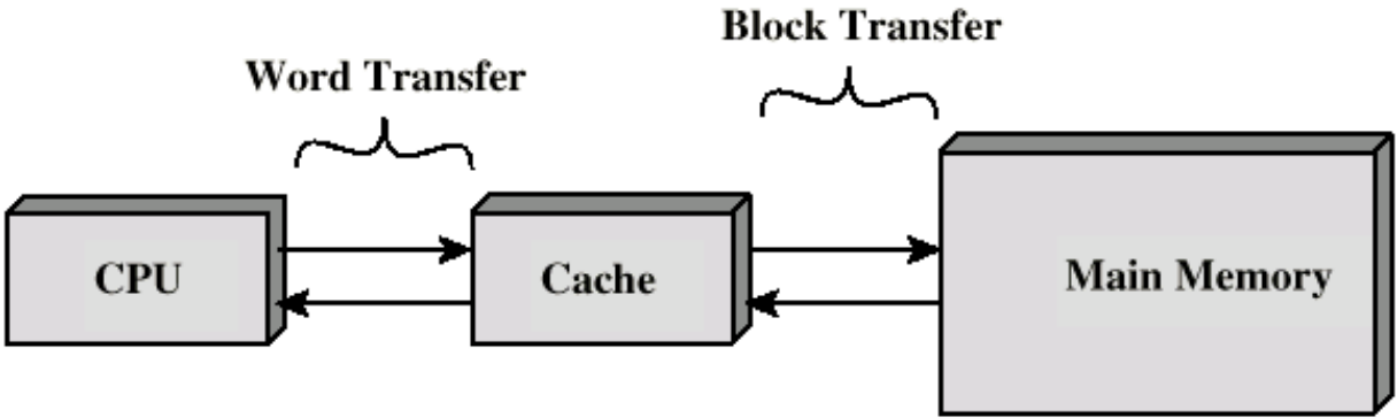


Cache

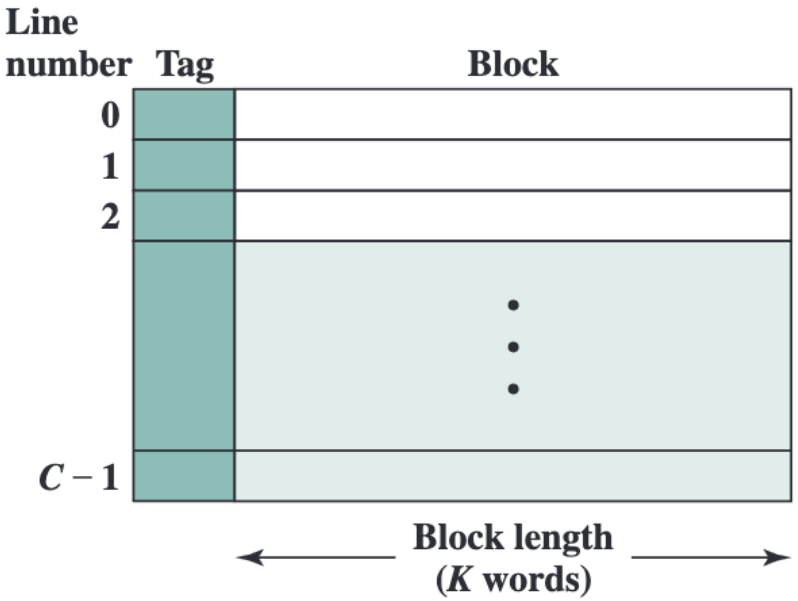
A staging area for subset of data in a larger, slower device

For each level, the faster, smaller device at level n serves as cache for larger, slower device at level n + 1

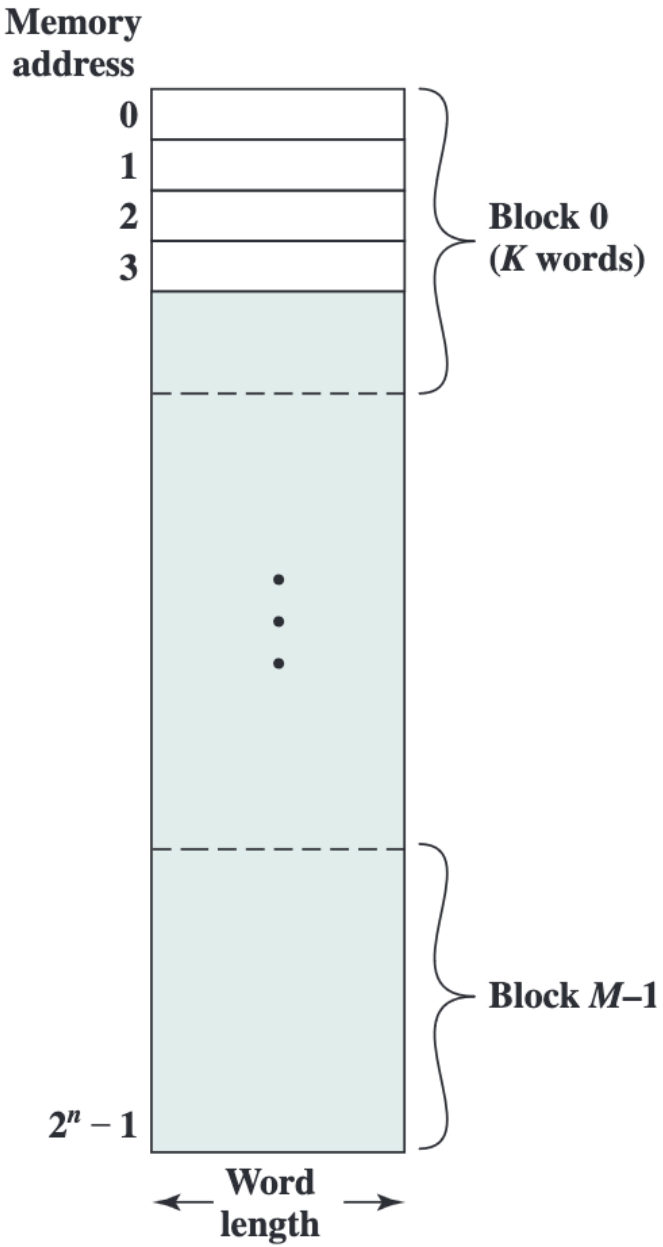
Memory hierarchies work because programs access data at level n more often than they access data at level n + 1



Cache Structure



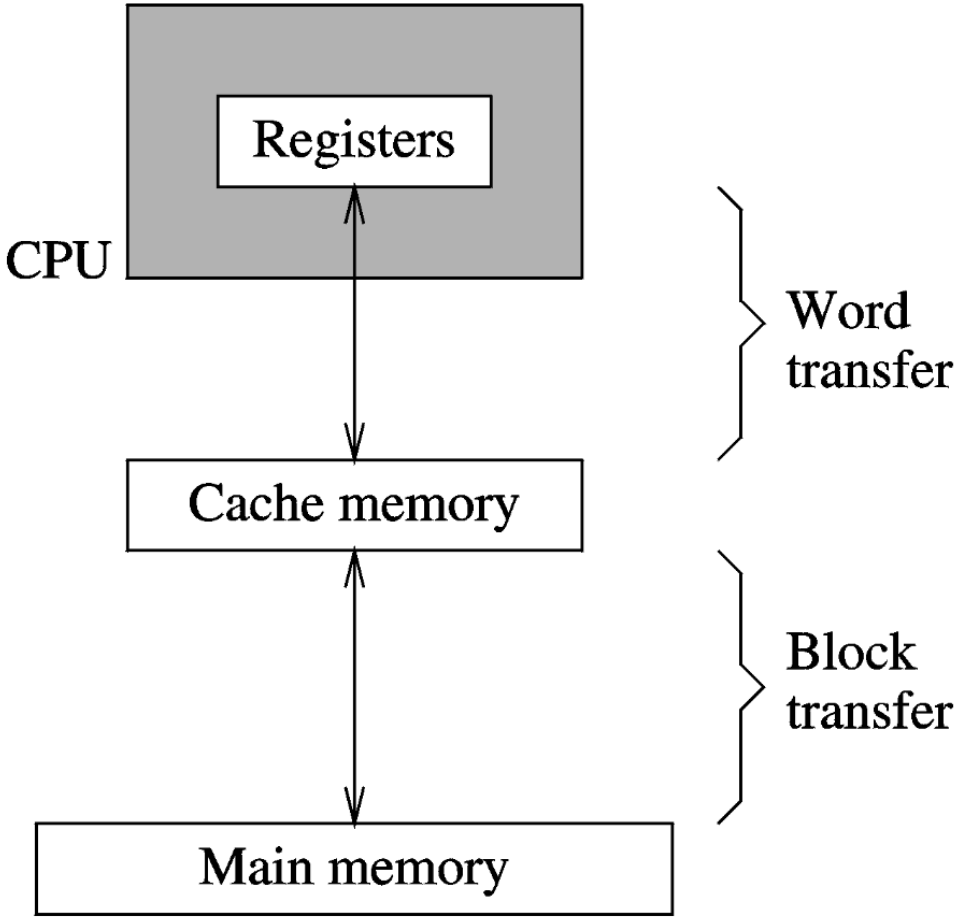
(a) Cache



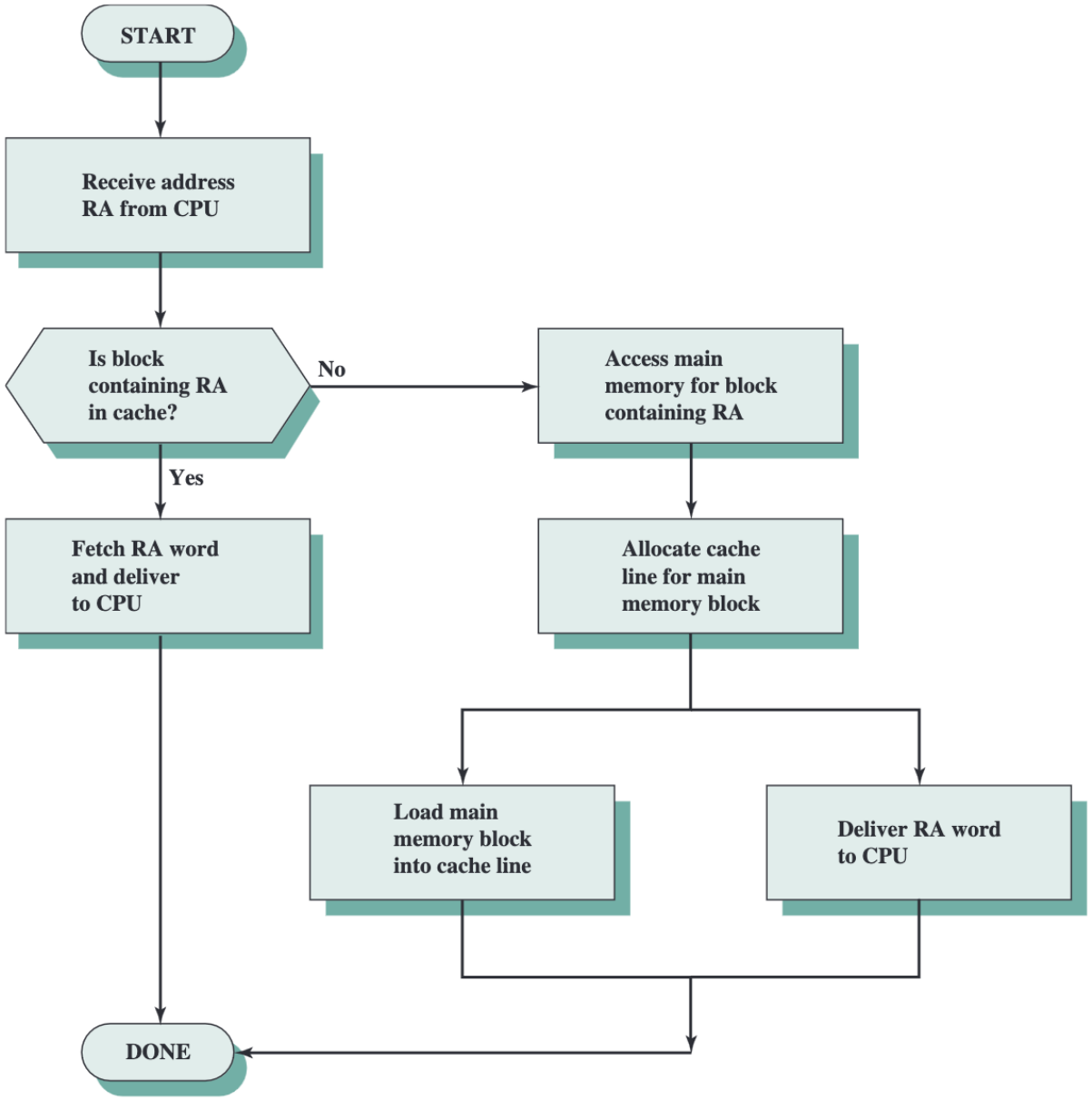
(b) Main memory

Cache Basics

- Transfer between main memory and cache
  - In units of blocks
  - Applies spatial locality
- Transfer between main memory and cache
  - In units of words
- Algorithms are required for:
  - Block placement
  - Mapping function
  - Block identification
  - Write policies

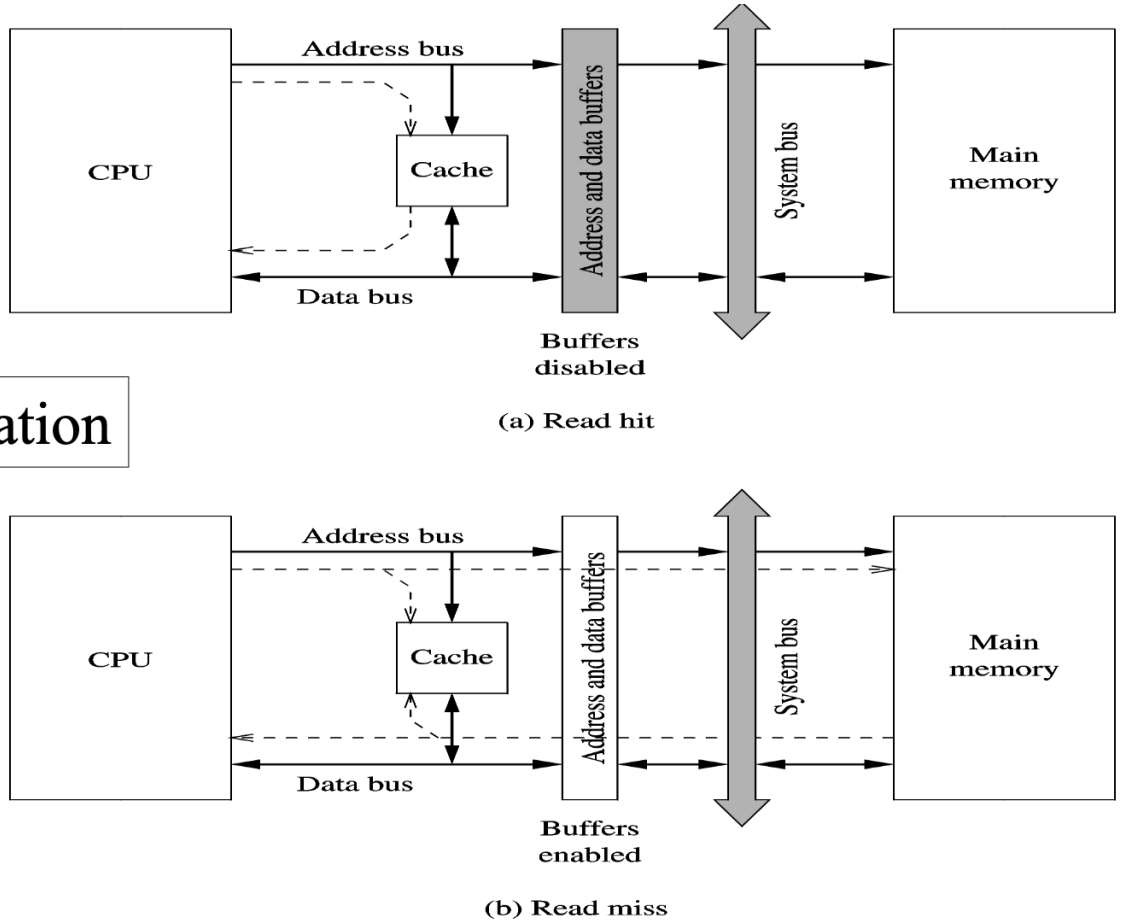


Cache Operation



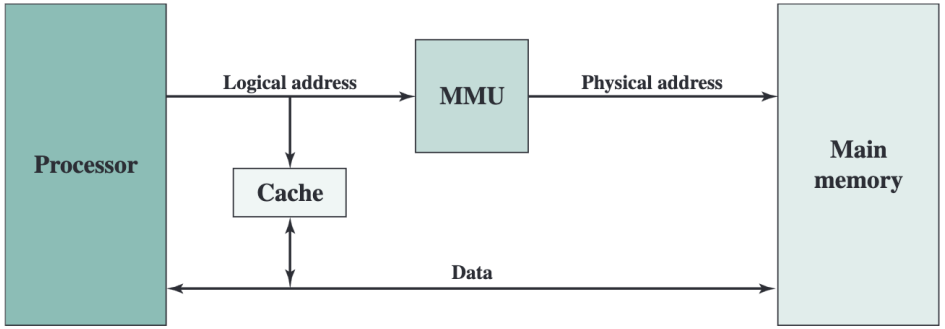
Cache Operation

Cache read operation

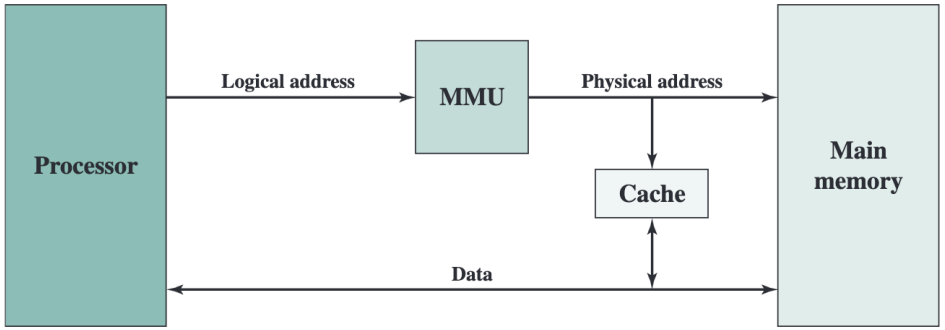


Cache Addresses

- **Logical/Virtual Cache:** stores data using virtual addresses
- **Physical Cache:** stores data using main memory physical addresses



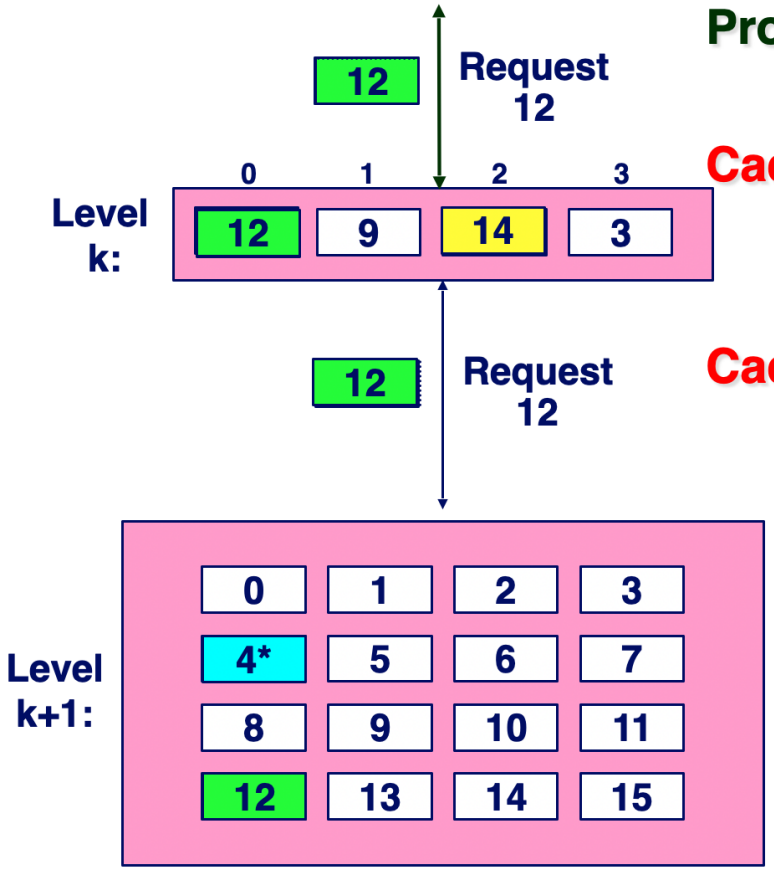
(a) Logical cache



Terminology

- **Cache Hit:** data appears in some block in the upper level or cache
- **Cache Hit Rate:** the fraction of memory access found in the lower level
- **Cache Hit Time:** time to access the lower level which consists of RAM access time + time to determine hit/miss
- **Cache Miss:** data needs to be retrieve from a block in the lower level
- **Cache Miss Rate** = 1 - (Hit Rate)
- **Cache Miss Penalty:** time to replace a block in the upper level + time to deliver the block the processor

Cache Miss and Hit



Program needs object d, which is stored in some block b

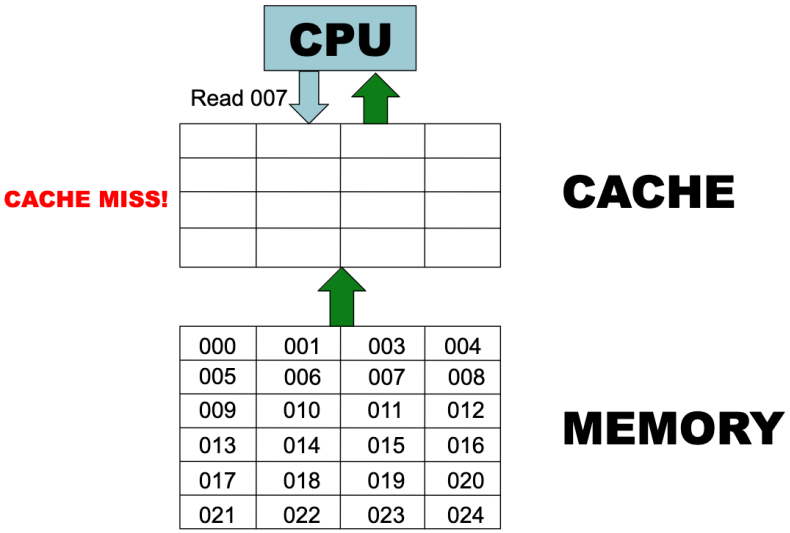
Cache hit

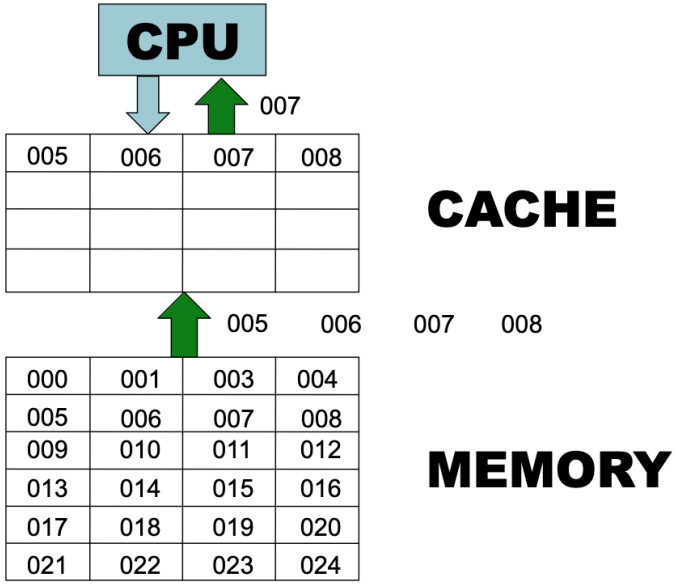
- Program finds b in the cache at level k. E.g., block 14

Cache miss

- b is not at level k, so level k cache must fetch it from level k+1. E.g., block 12
- If level k cache is full, then some current block must be replaced (evicted). Which one is the "victim"?
  - **Placement policy:** where can the new block go? E.g., b mod 4
  - **Replacement policy:** which block should be evicted? E.g., LRU

Cache Miss


CPU
Read 007
CACHE MISS!
CACHE
000 001 003 004
005 006 007 008
009 010 011 012
013 014 015 016
017 018 019 020
021 022 023 024
MEMORY


CPU
007
005 006 007 008
CACHE
005 006 007 008
000 001 003 004
005 006 007 008
009 010 011 012
013 014 015 016
017 018 019 020
021 022 023 024
MEMORY

Causes of misses

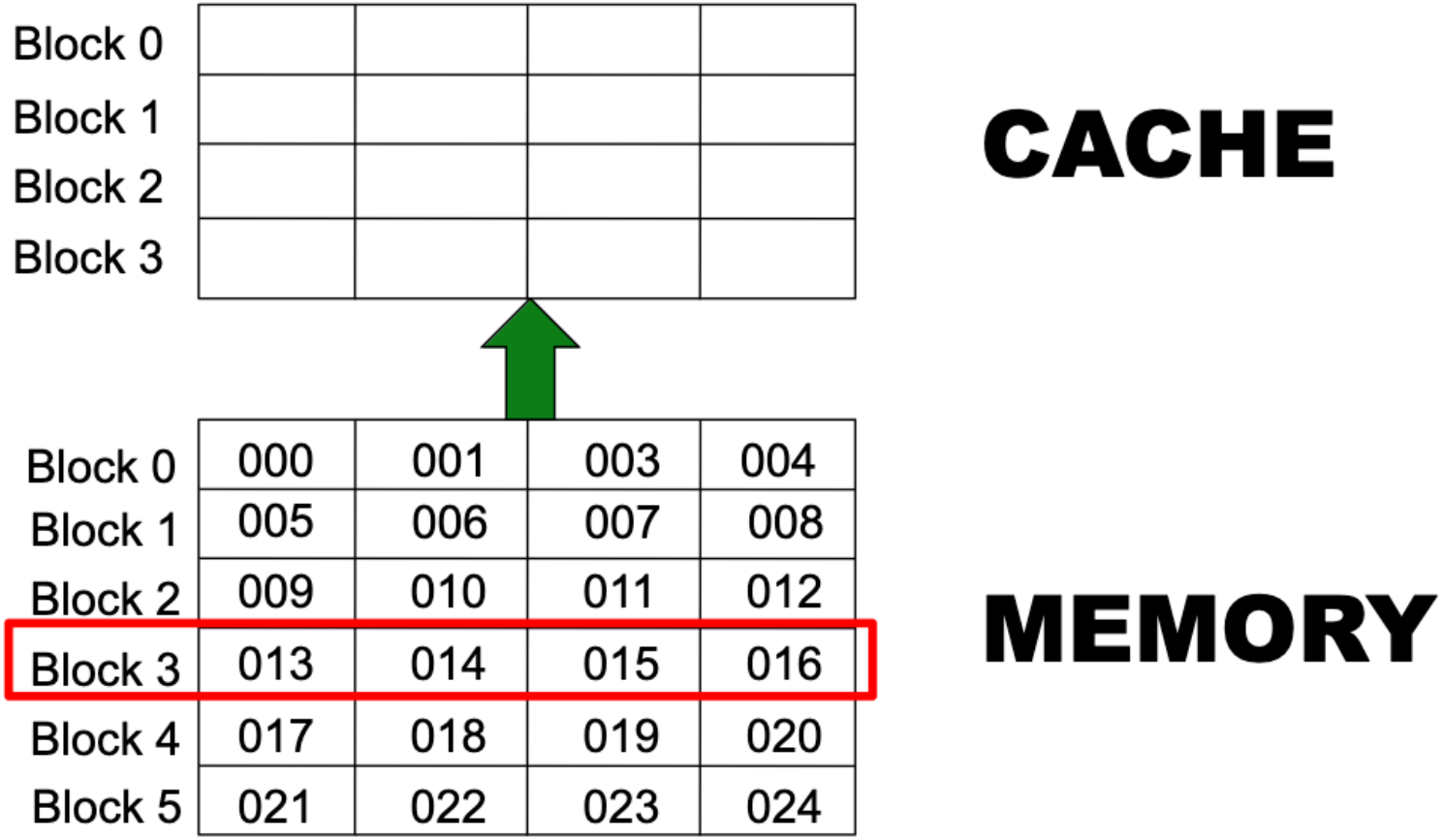
- **Compulsory:** first reference to a block. Would happen even for infinite caches
- **Capacity:** blocks discarded and later retrieved
- **Conflict:** program makes repeated references to multiple addresses from different blocks that map to the same location in the cache

Block Placement

- **Set associative:** block is mapped into a set and the block is placed anywhere in the set
- **Finding a block:** map block address to set and search set (usually in parallel) to find block
- n blocks in a set: n-way associative cache
- One block per set (n=1): direct-mapped cache
- One set per cache: fully associative cache

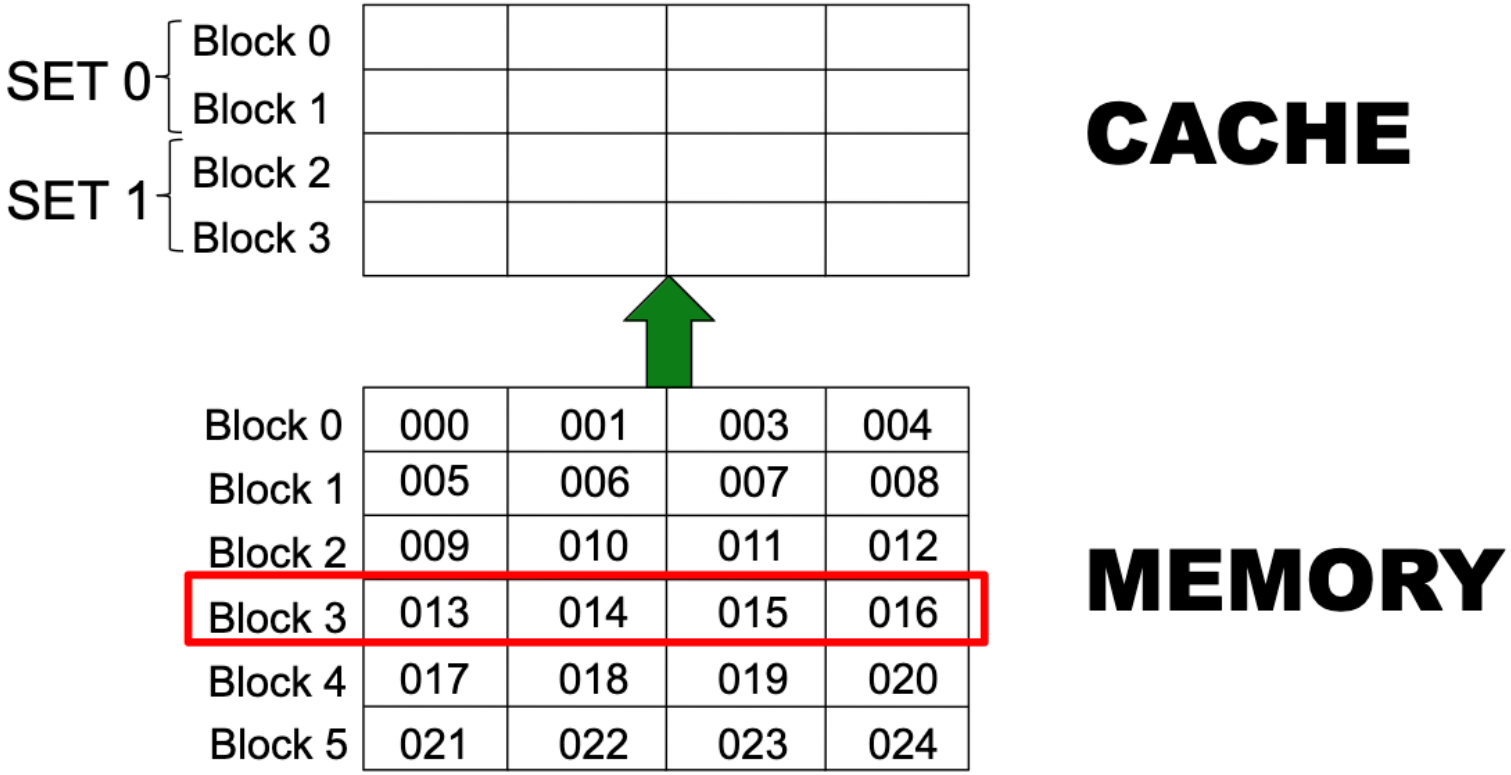
Fully Associative Cache

Memory block 3 can go anywhere in the cache



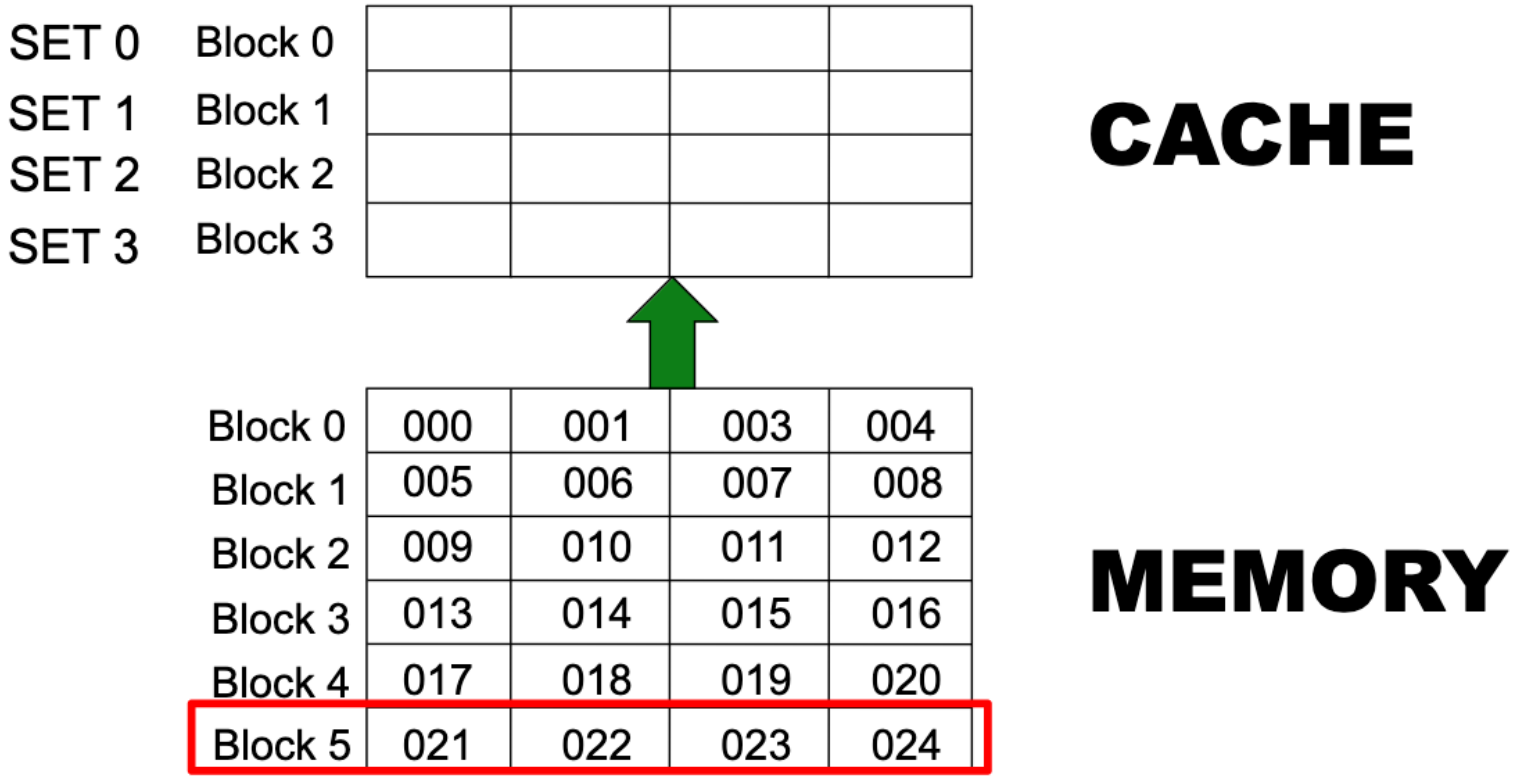
2-Way Set Associative Cache

Memory block 3 can only go into cache set (3 mod 2) = 1 in the cache



Direct-Mapped Cache

Memory block 5 can only go into cache block (5 mod 4) = 1 in the cache



Block Replacement

- Direct-mapped cache:
  - replace the block in the location where the incoming block has to go
- Fully Associative or Set Associative:
  - **Random:** spreads allocation uniformly
  - **Least Recently Used (LRU):** the block replaced is the one that has been unused for the longest time
  - **First In First Out (FIFO):** selects the oldest rather than the LRU block

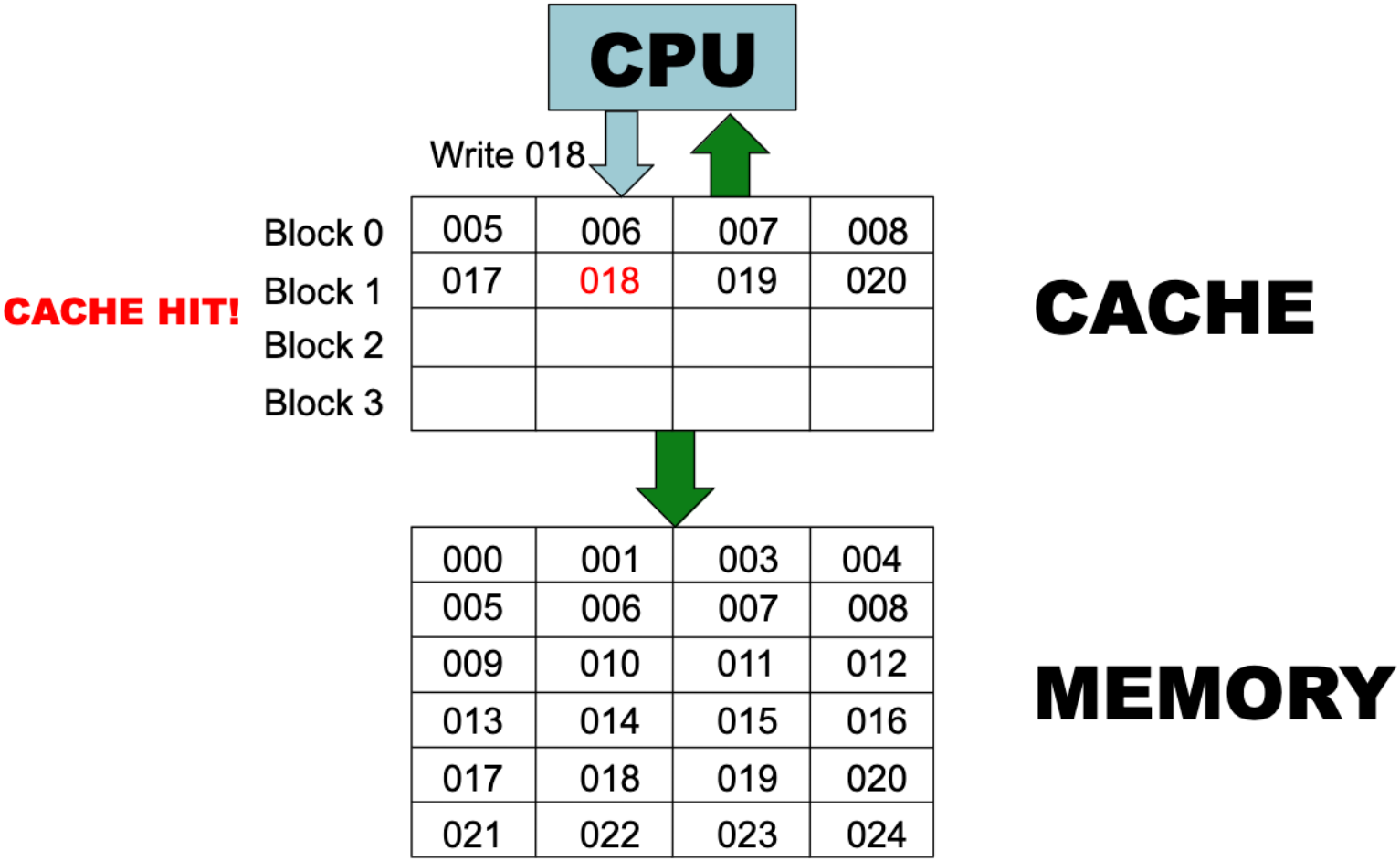
Writing to the Cache

- There are two approaches:
  - **Write-through:** Immediately update lower levels of hierarchy
  - **Write-back:** Only update lower levels of hierarchy when an updated block is replaced
- In all cases use a write buffer to make writes asynchronous

Writing-through

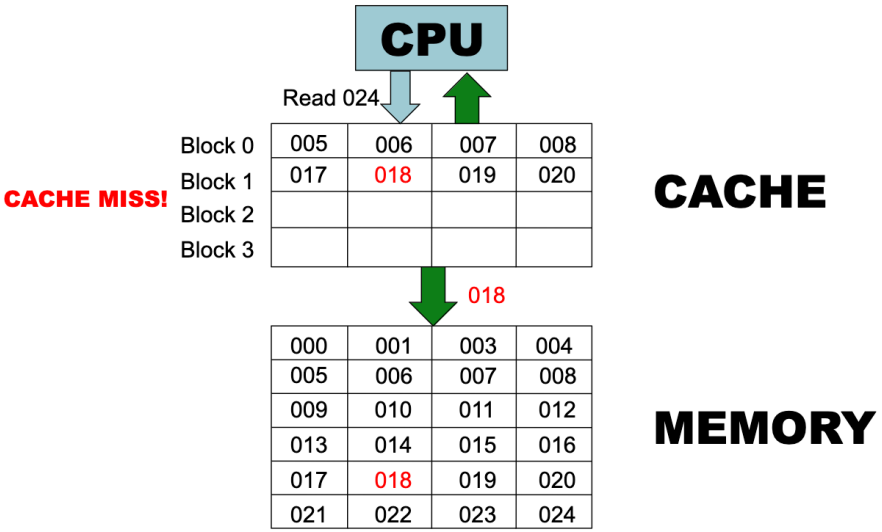
Forces all writes to update both the cache and the main memory

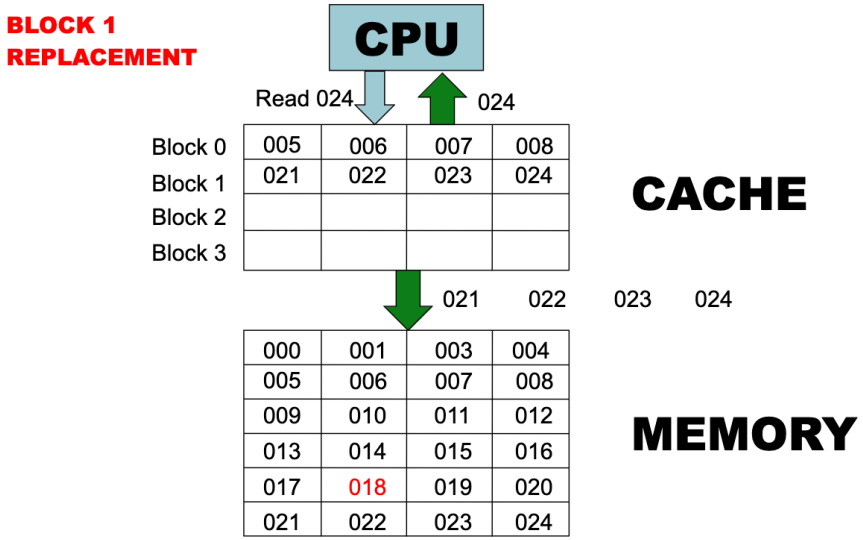
It uses up bandwidth between the cache and the memory



Writing-back

The memory is not updated until the cache block needs to be replaced





Performance Impact of Misses

$$\frac{\text{Misses}}{\text{Instruction}} = \frac{\text{Miss rate} \times \text{Memory accesses}}{\text{Instruction count}} = \text{Miss rate} \times \frac{\text{Memory accesses}}{\text{Instruction}}$$

$$\text{Average memory access time} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

$$\begin{aligned}\text{MemoryStallCycles} &= \text{NumberOfMisses} \times \text{MissPenalty} = \\ &= \text{IC} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{MissPenalty} \\ &= \text{IC} \times \frac{\text{MemoryAccesses}}{\text{Instruction}} \times \text{MissRate} \times \text{MissPenalty}\end{aligned}$$

Resources

- https://courses.cs.washington.edu/courses/cse378/09wi/lectures/lec18.pdf
- https://www.cs.utexas.edu/~mckinley/352/lectures/16.pdf
- https://www.cs.hmc.edu/slides/class11_memory.ppt