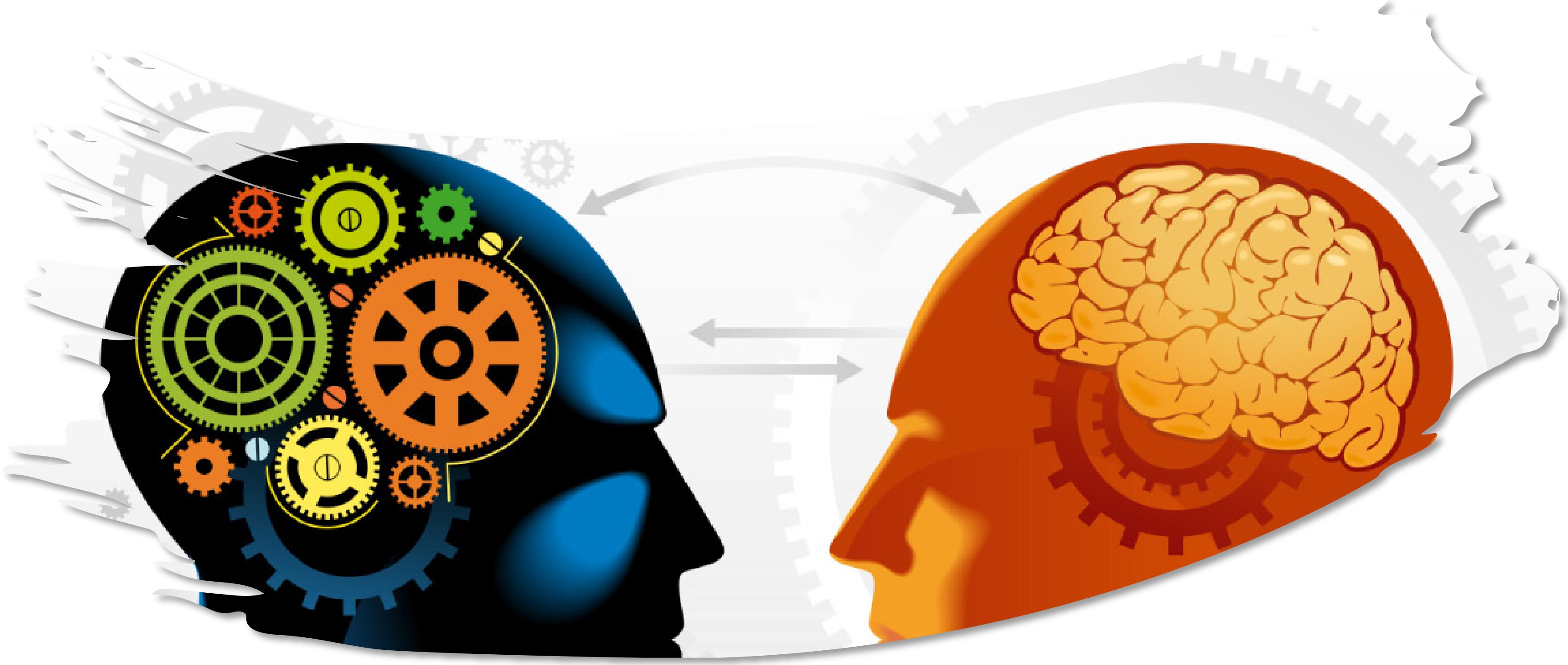


Building the Infinite Brain

COMP 690 (193)

Raghavendra Pradyumna Pothukuchi

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

raghav@cs.unc.edu

Quick Review

Homework 2 assigned, due 3/9

Why dynamic scheduling?

To be resource efficient in exploiting instruction level parallelism

What are hardware methods for dynamic scheduling?

Scoreboarding, Tomasulo's algorithm, speculation

What systems principles does dynamic scheduling involve?

Eager, speculation, and concurrency

What are some other methods to exploit ILP?

Software pipelining, unrolling that target static ILP

For Today

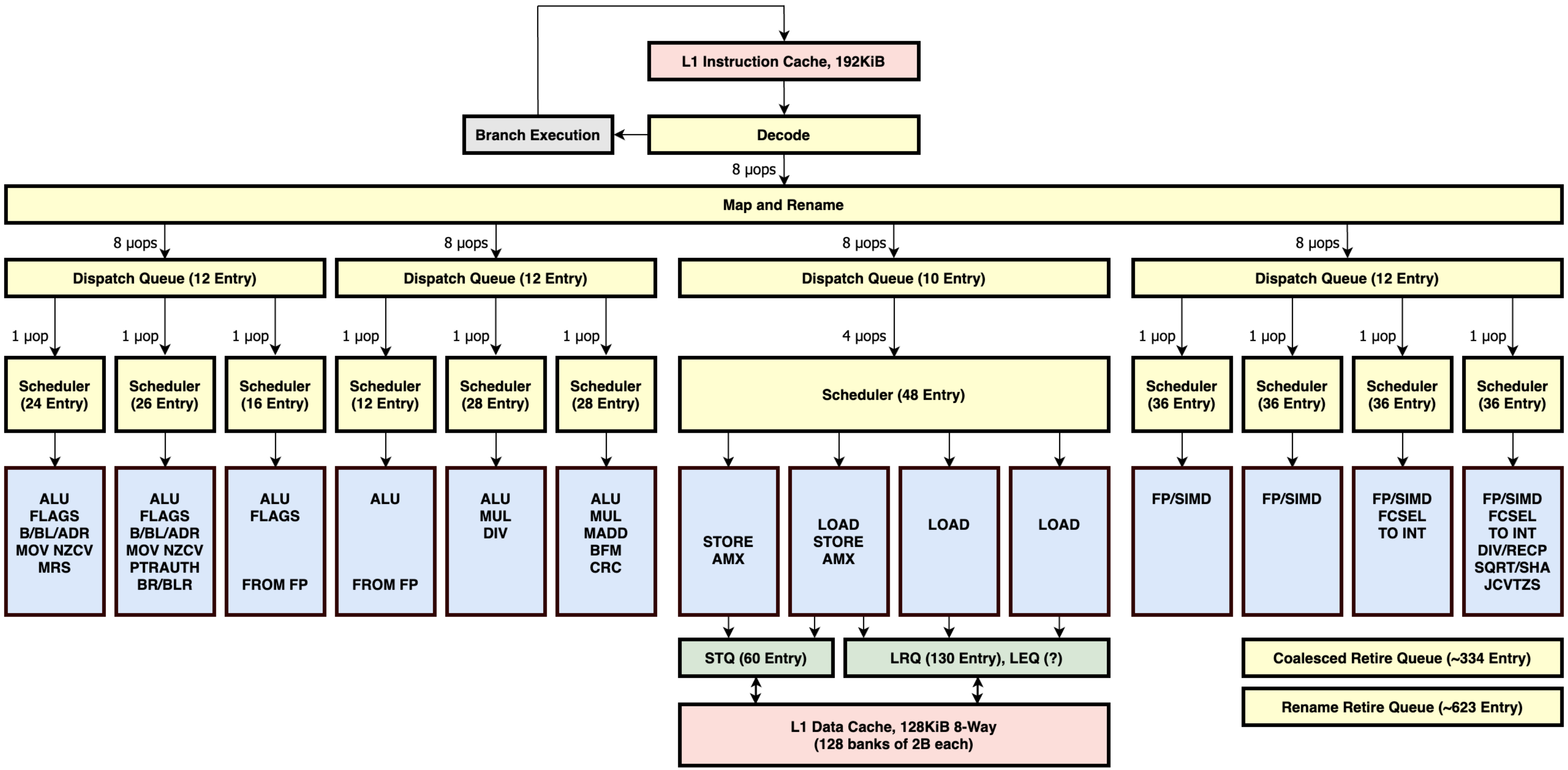
- Quick review
- **Caches**

Motivation

Memory accesses are slow

Typically, 100×

Methods to regroup work (dynamic scheduling) still cannot stop clogged up pipelines: *bottleneck*



How To Improve Performance?

“Exploit technology”

	Registers	SRAM	DRAM	NVM	Disk
Latency	1× (tens of ps)	50× (low ns)	500× (tens of ns)	500,000× (tens of µs)	500,000,00× (ms)
Area/bit	1×	0.1×	0.01×	0.001×	0.01×
Cost/bit	$$$$$	$$$$	$$$	$$	$

Pick one?

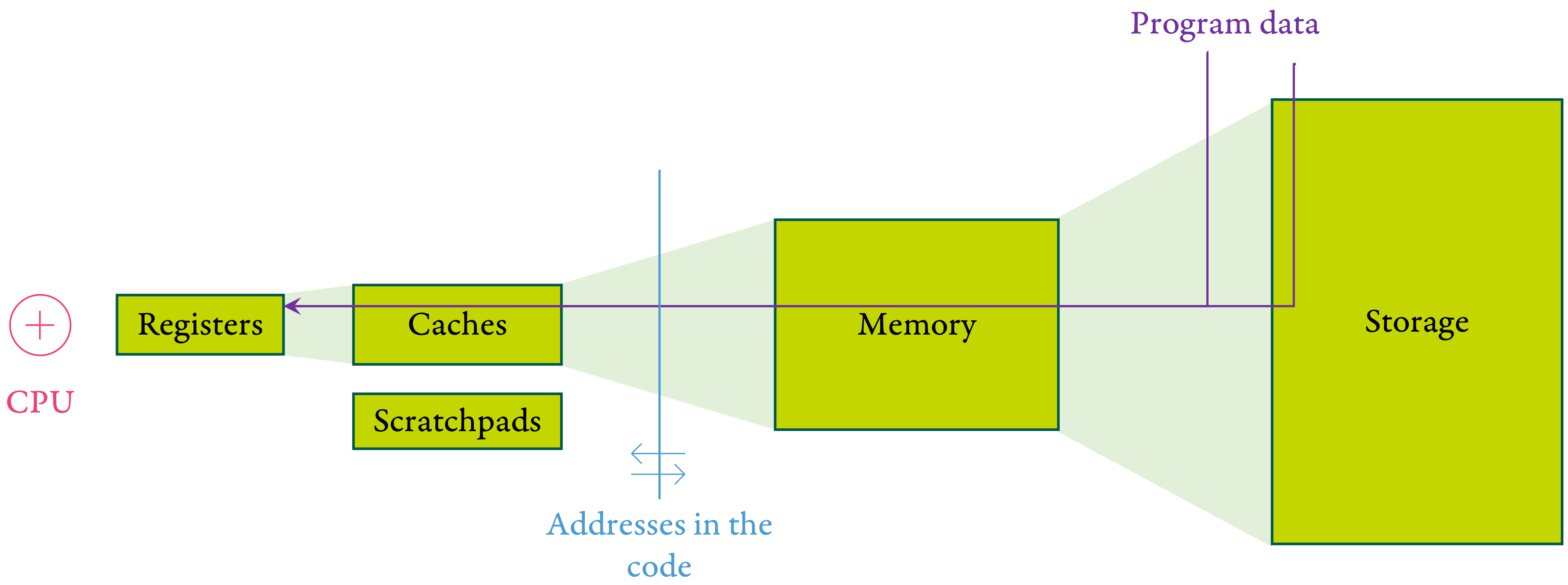
“Yummy”: *Who will buy it?*

Is it “Efficient”?

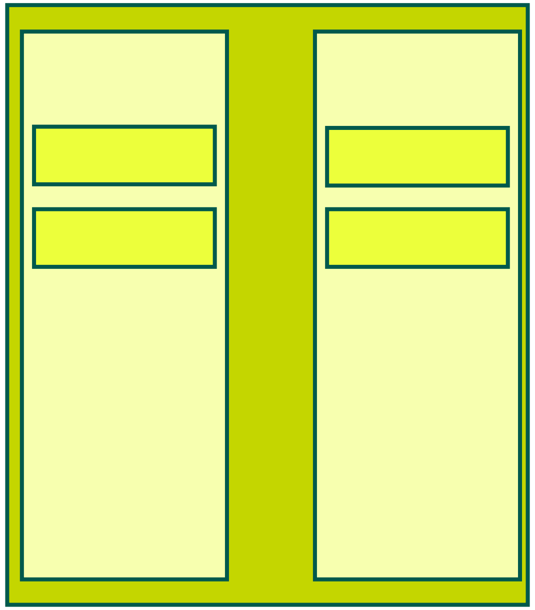
“Divide and conquer” and “Fast Path”

Optimize different structures for different goals, and make unwanted case uncommon

Memory Hierarchy



Caching

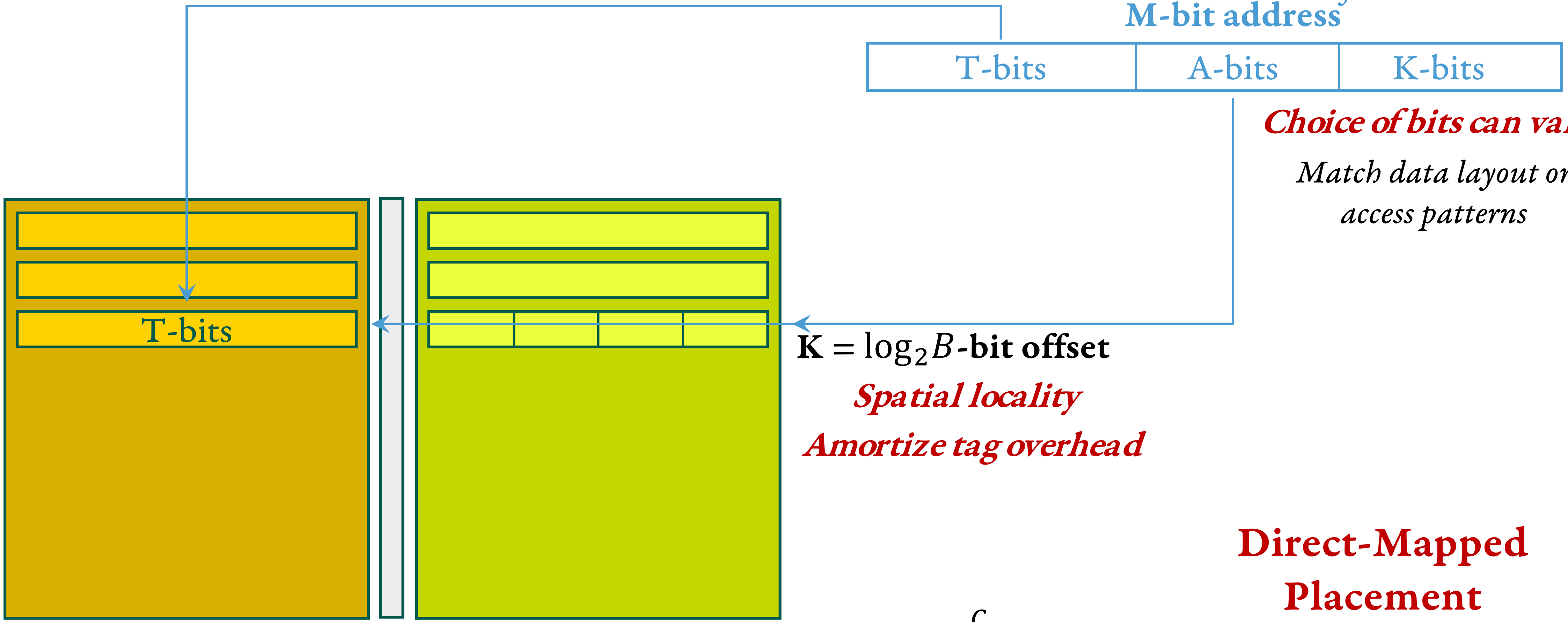


Organization and indexing
(address-to-entry mapping, placement, replacement)

Block (entry size, B)

Capacity (total size, C)

A Basic Cache



Choice of bits can vary

Match data layout or access patterns

K $= \log_2 B$**-bit offset**

Spatial locality

Amortize tag overhead

Tag array **Valid** **Data array**

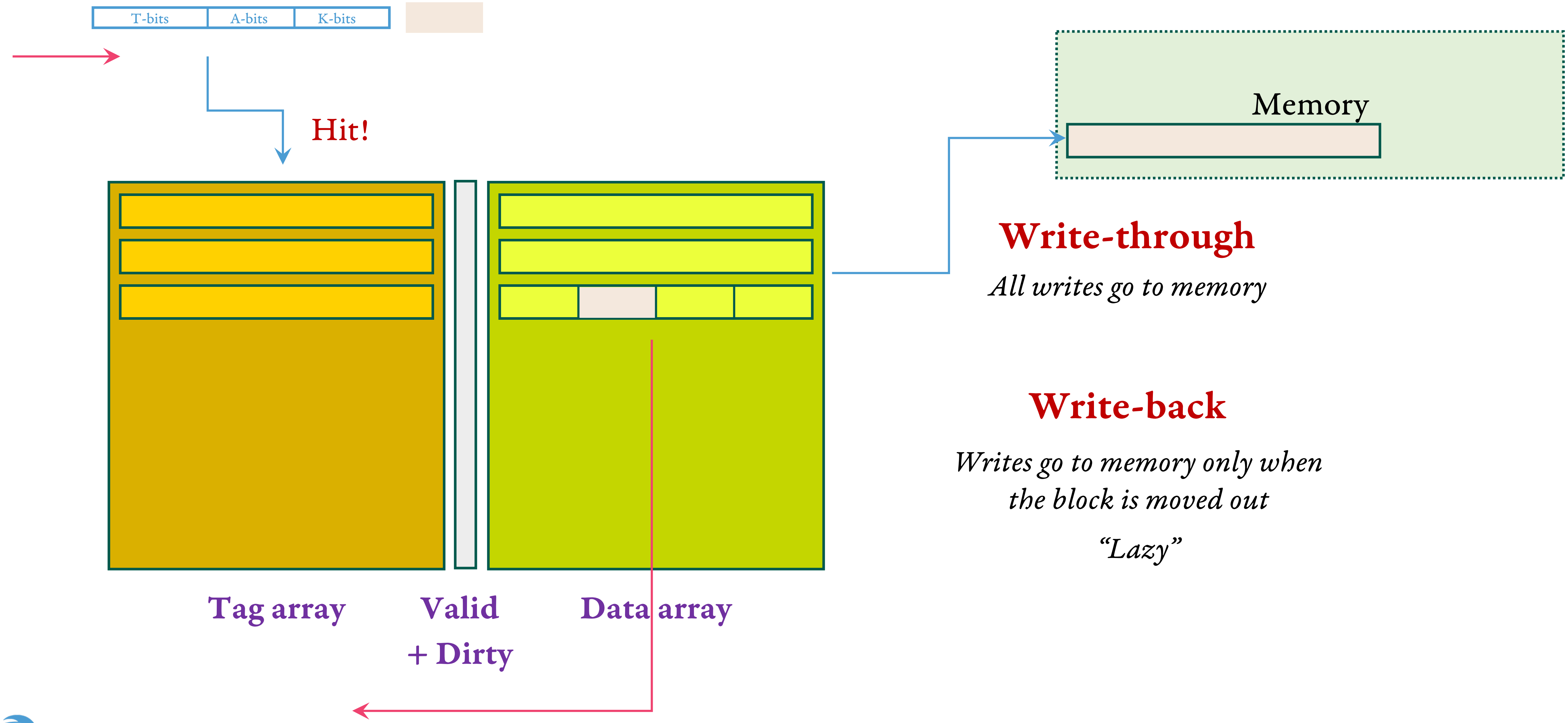
Overhead

N entries $= \frac{C}{B}$

A $= \log_2 N$**-bit address**

Direct-Mapped Placement

A Basic Cache: Reads and Writes



Write-through

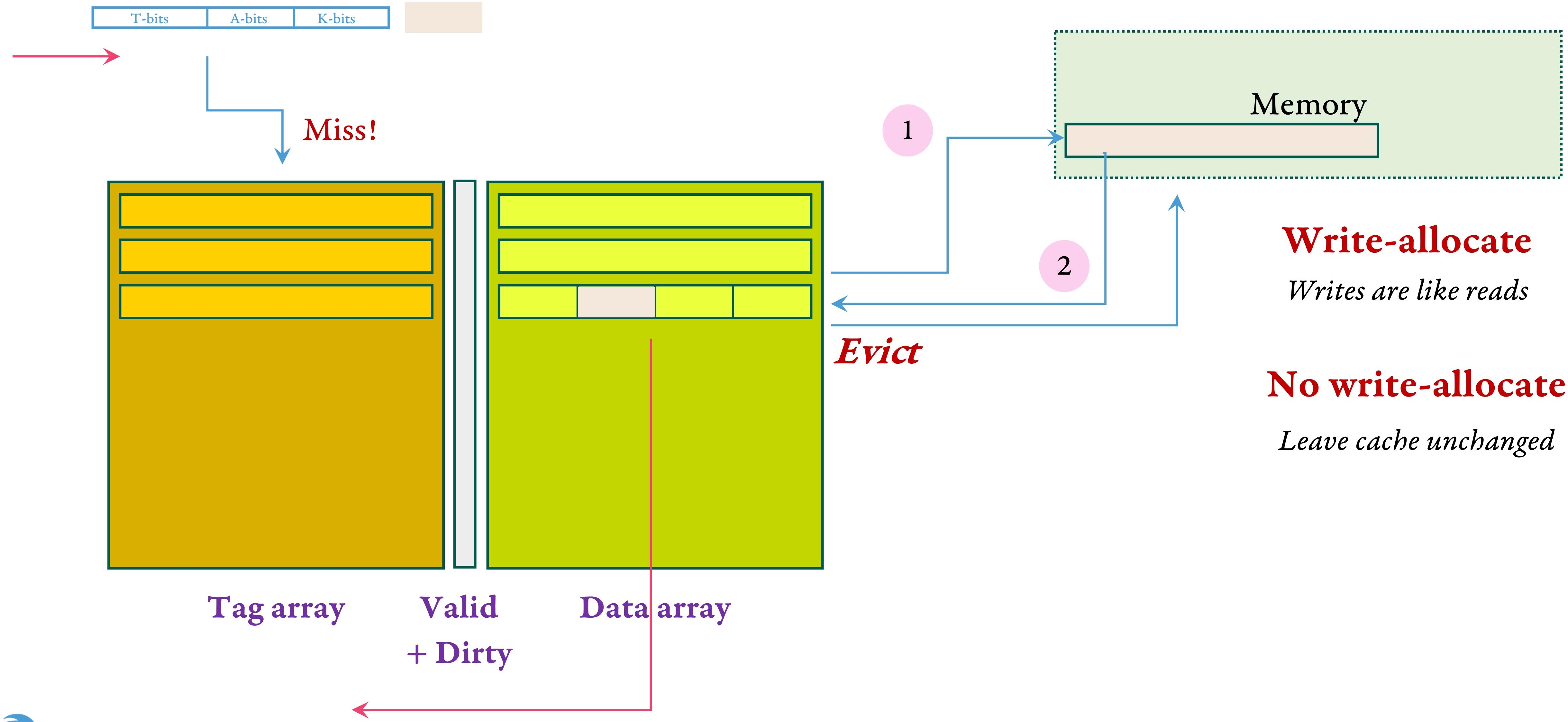
All writes go to memory

Write-back

Writes go to memory only when the block is moved out

"Lazy"

A Basic Cache: Reads and Writes



Write-allocate

Writes are like reads

No write-allocate

Leave cache unchanged

Cache Performance Metrics

Misses per kilo instructions (MPKI)

Miss ratio

$$\frac{\text{Misses}}{\text{Accesses}}$$

Average memory access time (AMAT)

$$T_{hit} + \text{Miss ratio} \times T_{miss}$$

Recall "*fast path*" from Lecture 2

Expand recursively to all levels

Average latency seen in the pipeline

Understanding Misses

Compulsory misses

First reference to a block misses in the cache

Capacity misses

Cache is too small for the *working set*

Conflict misses

Cache blocks evicted due to mapping conflicts

Improving the Likelihood of the Fast Path

Compulsory misses

First reference to a block misses in the cache

Sizing right is key—can hurt hits!

Prefetching

Eager, speculation

Increase block size

Locality

Capacity misses

Cache is too small for the *working set*

Increase cache size

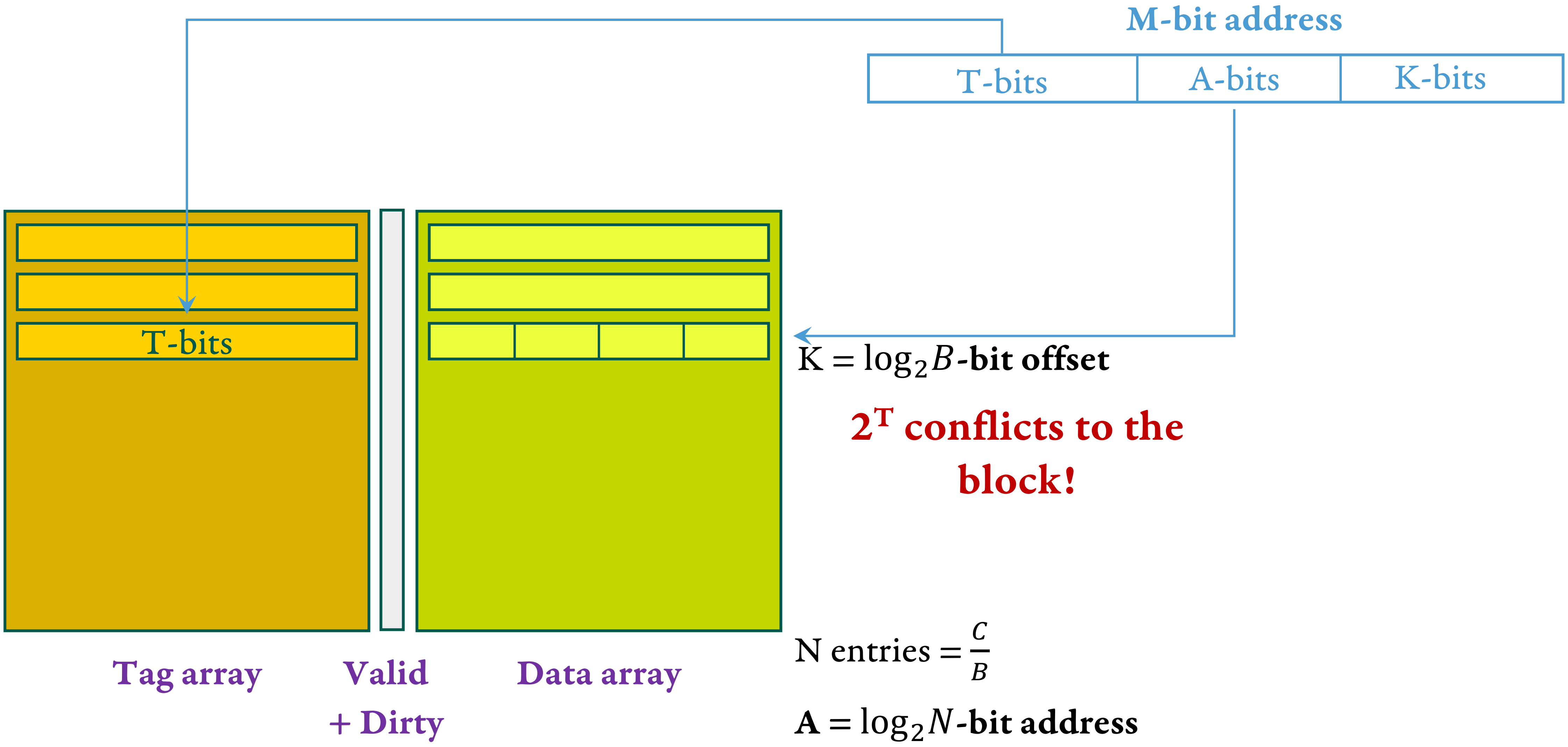
Conflict misses

Cache blocks evicted due to mapping conflicts

Increase entries

Change mapping?

Cache Mapping and Associativity



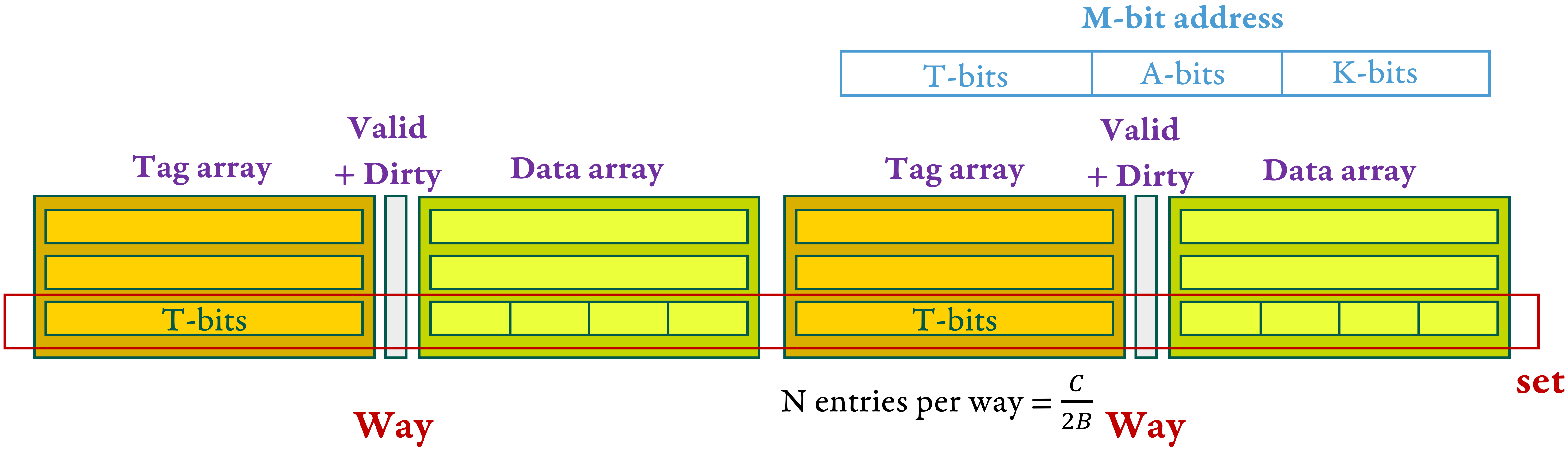
$K = \log_2 B$**-bit offset**

2^T conflicts to the block!

$\text{N entries} = \frac{C}{B}$

A $= \log_2 N$**-bit address**

Cache Mapping and Associativity



N entries per way $= \frac{C}{2B}$

A $= \log_2 N$**-bit set address**

At a high level, this cache appears to have the same number of address conflicts, but overall conflict misses are fewer

Number of sets is limited by "page size" (later)

Generalize to m-way associative caches

Fully associative cache: 1 entry per way, i.e., the entire cache is 1 set

Cache optimizations

Self study
What are the broader principles?

Split and unified caches

Instruction and Data

Nonblocking or Lockup-free caches

Caches don't stall on a miss (use Miss Status Handling Registers)

Way prediction

Banking

Request critical word first (from Memory) and enable early restart (for processor)

Write merging

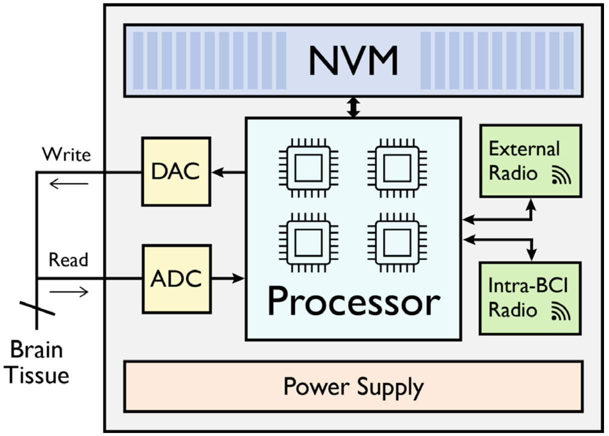
Software optimizations too!

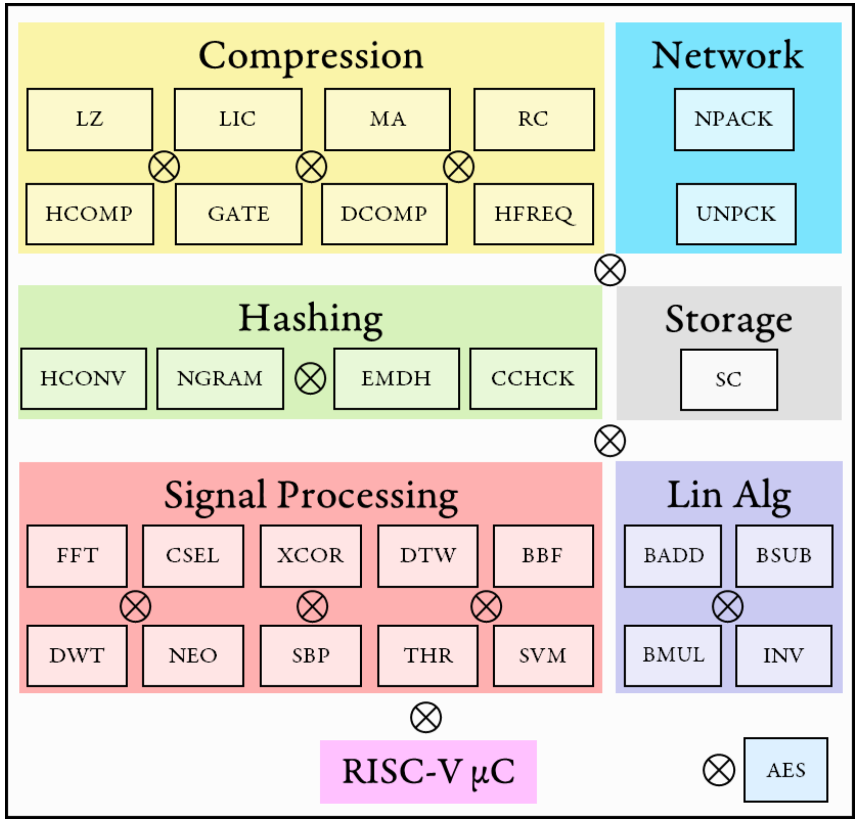
Example from a BCI Processor (SCALO)

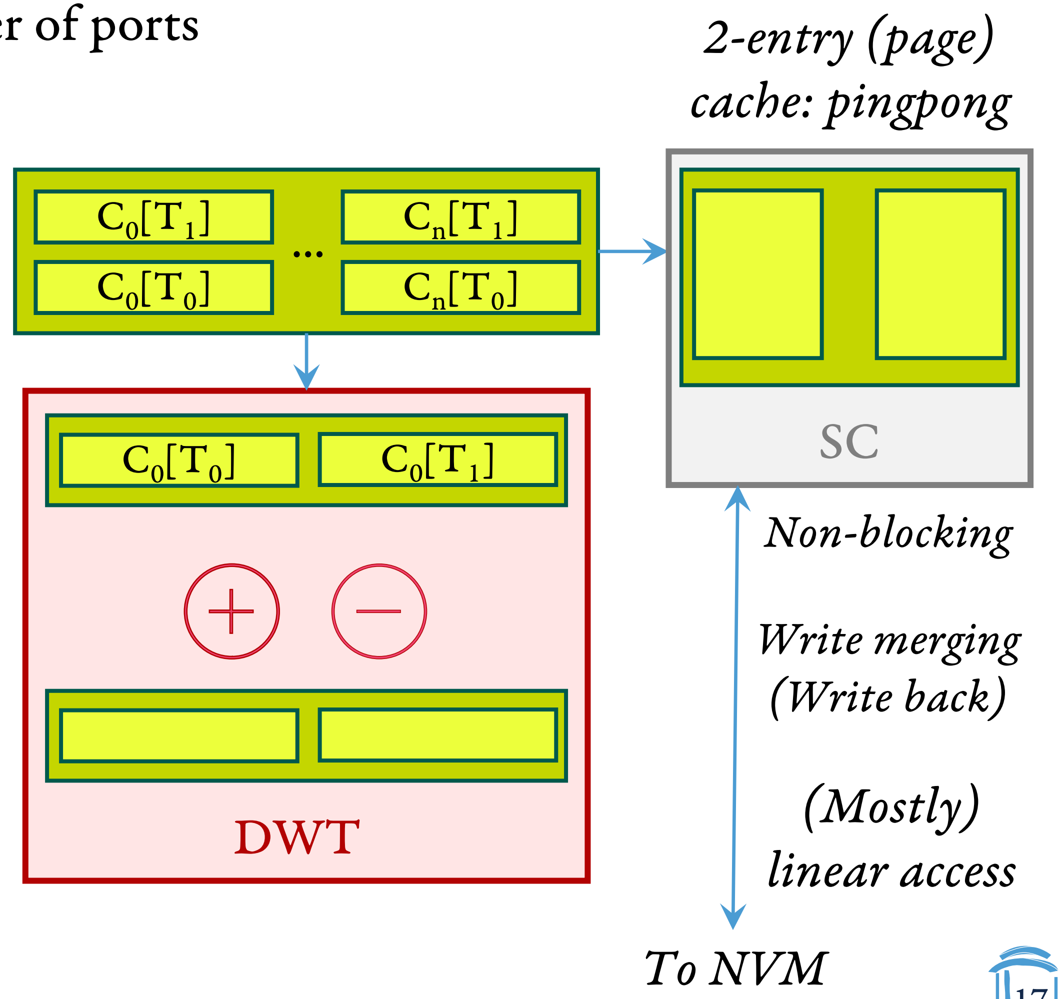
Scratchpads to support streaming dataflow, and storage to hold data

No registers, caches or memory, but there is a cache for storage

Custom port widths and number of ports







Takeaways

Review B3: Ch. 2 and App B.

What is caching?

A technique to minimize the impact of long memory latencies

What are the basic cache parameters?

Associativity (placement), block size, capacity, write through/back, write-allocate or no

What are the types of cache misses?

Compulsory, capacity, conflict

What are some ways in which cache performance can be improved?

Non-blocking, banking, software or hardware prefetching etc.

How to choose the right memory hierarchy design?

Tailor it to the access patterns and layout: general purpose to domain specific

Understand the relevant "systems" problems and identify solutions

Image Credits (Educational, Fair Use)

- Title image: VLADGRIN, https://www.istockphoto.com/vector/human_-machine-gm147409511-16840728 (Educational fair use)
- Infinite brain: Science wonder stories, May 1930, Illustrator: Frank R Paul, Editor: Hugo Gernsback
- Brain color, ICs, cloud server, black rat: No attribution required (Hiclipart)
- Hand with spoon: public domain freepng
- Signals: https://www.nature.com/articles/nrn3724
- Thought clouds: F. Willett et al./*Nature* 2021/Erika Woodrum, https://med.stanford.edu/neurosurgery/news/2022/bci-award. https://www.the-scientist.com/news-opinion/brain-computer-interface-user-types-90-characters-per-minute-with-mind-68762
- Picture of scientists: https://www.cs.auckland.ac.nz/~brian/rutherford8.html (original: Pierre de Latil), Bush (Carnegie Science), Others (Wikipedia, National Academies, IEEE, and university profile images)
- Flowchart: Pause08 – flaticon.com; Digital brain: Smashicons – flaticon.com; Quantum processor icons created by Paul J. - Flaticon
- Server rack: upklyak – freepik.com
- Arm, Lotus: Adobe stock
- Quantum processor: Rigetti computing
- Images of implanted users: Top: Case Western Reserve University (https://thedaily.case.edu/man-quadriplegia-employs-injury-bridging-technologies-move-just-thinking/), Bottom: Jan Scheuermann (University of Pittsburgh/UPMC; https://www.upmc.com/media/news/bci-press-release-chocolate)
- Images of wearable BCIs: Cognixion, NextMind
- Types of BCIs: "Brain–computer interfaces for communication and rehabilitation,
- Illustrative BCI: Neuralink
- Electrodes: "Electrochemical and electrophysiological considerations for clinical high channel count neural interfaces", Vatsyayan et al.
- Form factors: Neuropace, Medtronic, Bloomberg, "Fully Implanted Brain–Computer Interface in a Locked-In Patient with ALS" by Vansteensel et al., Blackrock Neurotech
- Jose Delgado's video: Online, various sources (CNN, Youtube)
- Video of Kennedy and Ramsey: Online, various sources (Youtube, Neural signals)
- Code snippet inspiration: ECE 252 slides at Duke (Dan Sorin et al.)
- Apple processor pipeline: https://dougallj.github.io/applecpu/firestorm.html