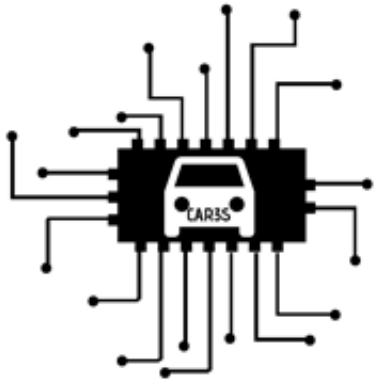


@WINTER SCHOOL

@CSE-IIT Kanpur*

21st December 2020

*Joining IIT Bombay in May 2021

CACHES@WINTER SCHOOL

@CSE-IIT Kanpur*

21st December 2020

*Joining IIT Bombay in May 2021

Protocol

- Feel free to ask questions anytime. I will pause after slide no. 39 and 83
- Lectures won't be self-contained because of time constraints
- The goal is to open-up your natural neural network (a.k.a. brain) and build up some curiosity (so I may confuse you).
- Feel free to reach out after/during the winter-school through: email/LinkedIn/Twitter.

Ping me on MS teams: @WS-Biswa or biswa-interaction (channel)

The World with No Cache

North pole ☹

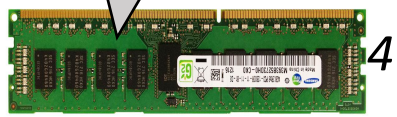
Core

32-bit Address

Data

200 to 300 cycles

Costly DRAM accesses☹

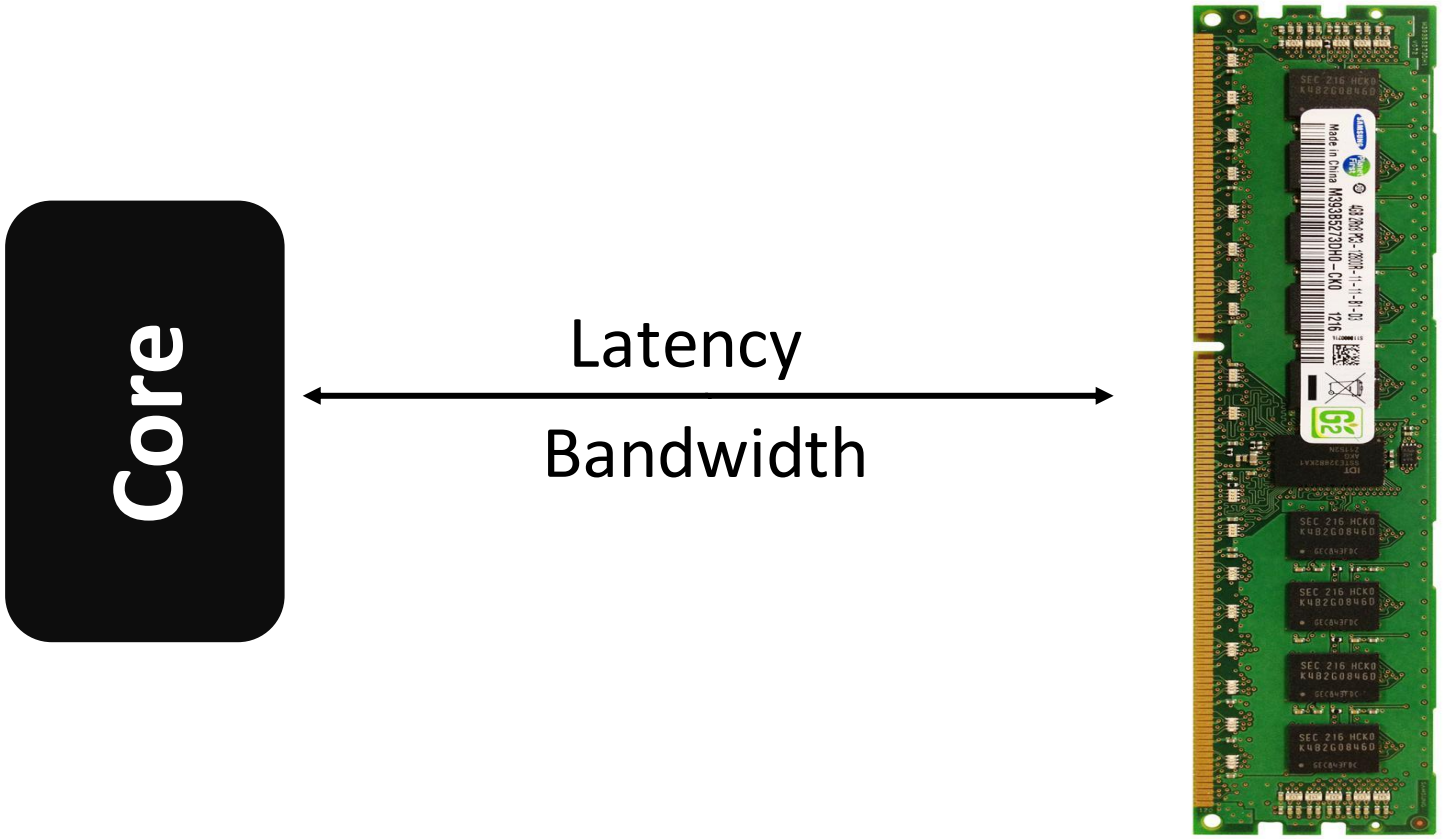


4 GB DRAM

South pole ☹

Minimizing costly DRAM accesses is critical for performance

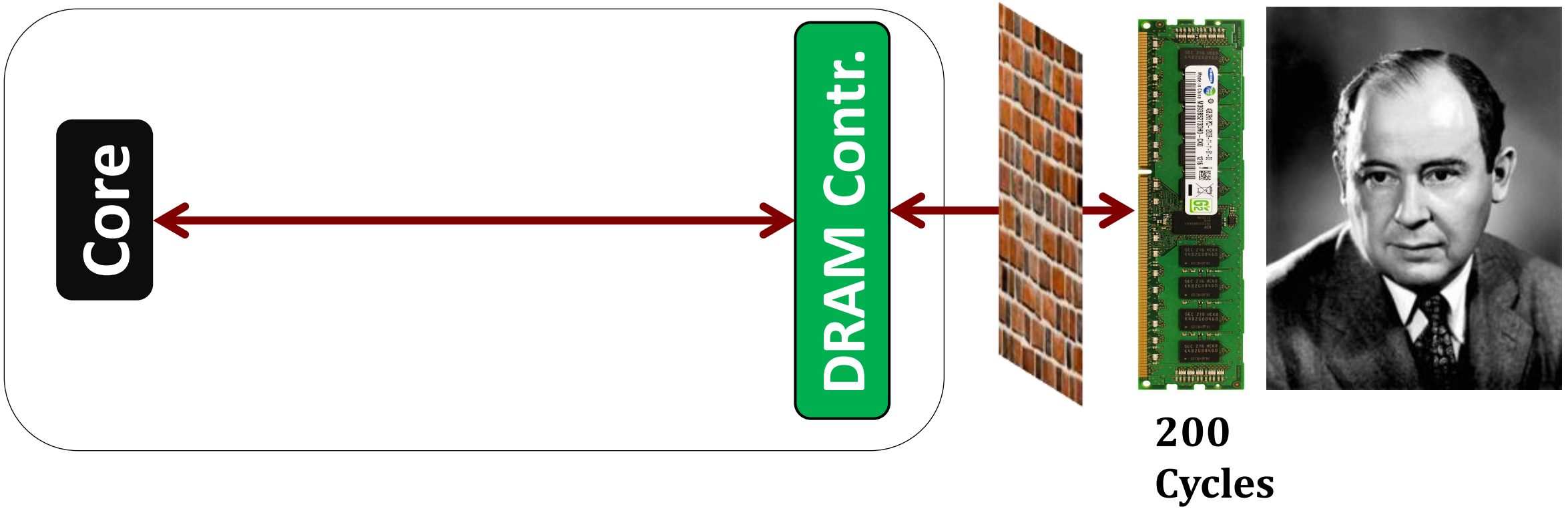
World of Latency and Bandwidth



Latency ☹

Bandwidth problems can be cured with money.
Latency problems are harder because the speed of light is fixed – you can't bribe God

Why Access Memory?



Memory stores CODE and DATA

Processor accesses for LOADs (reads) and STOREs(writes)

Memory Wall: Grandmother of all the walls ☺

Hang on!! I got the Mantra!!

Reduction in DRAM accesses ~ Improvement in execution time

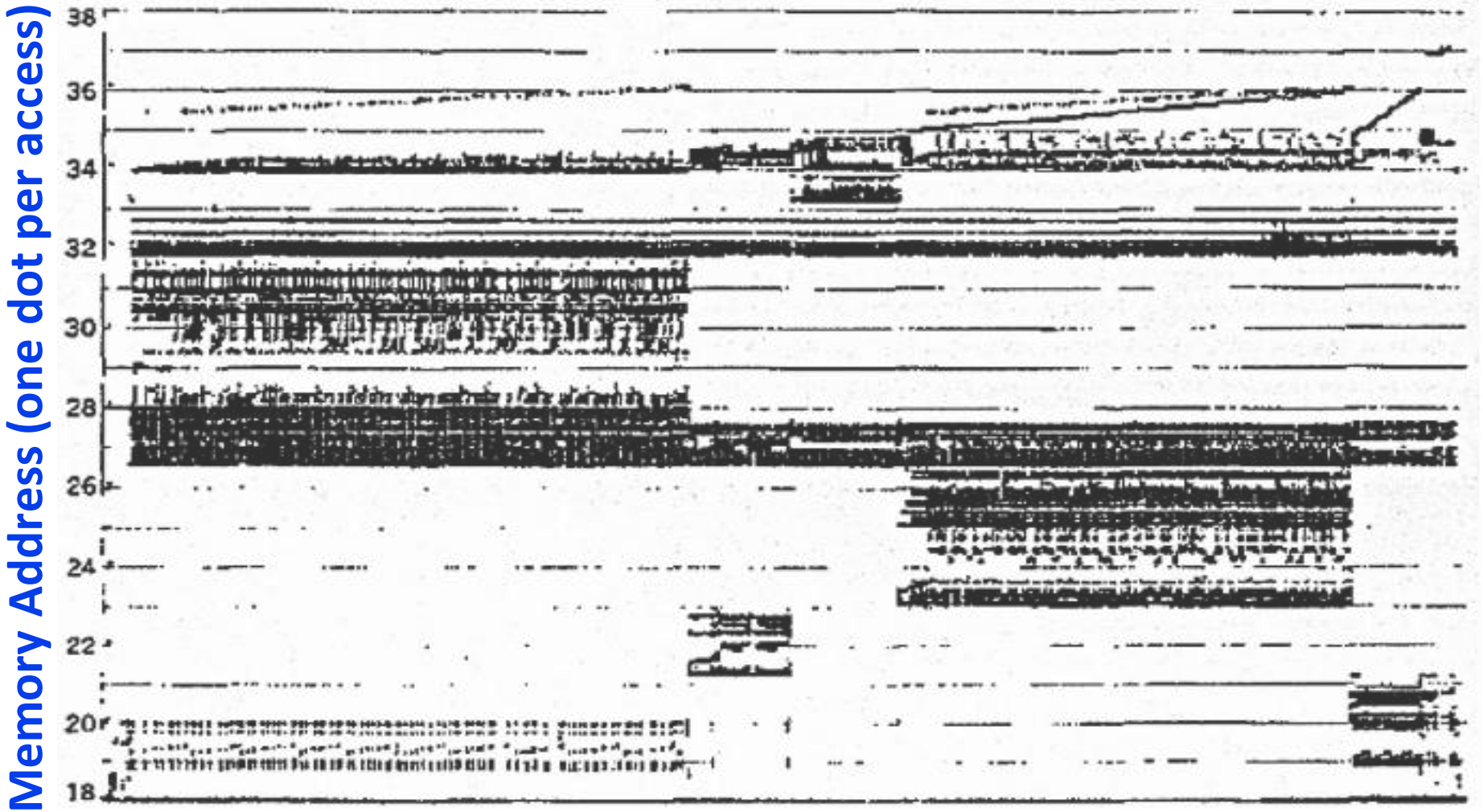
WRONG!

- First Law of Performance:

  Make the common case fast

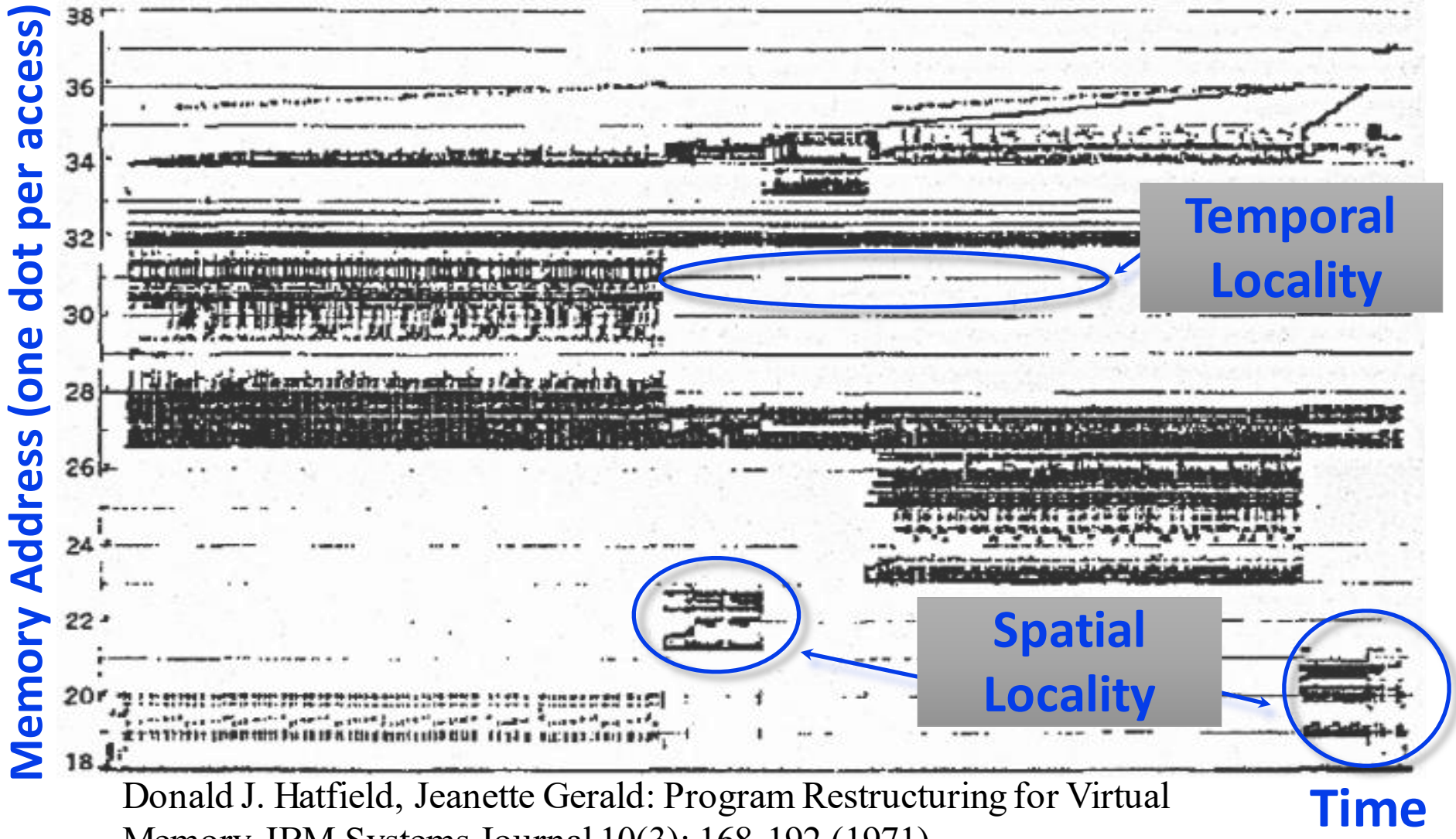
What if your program is not memory intensive

Let's Look at Applications (Programs)



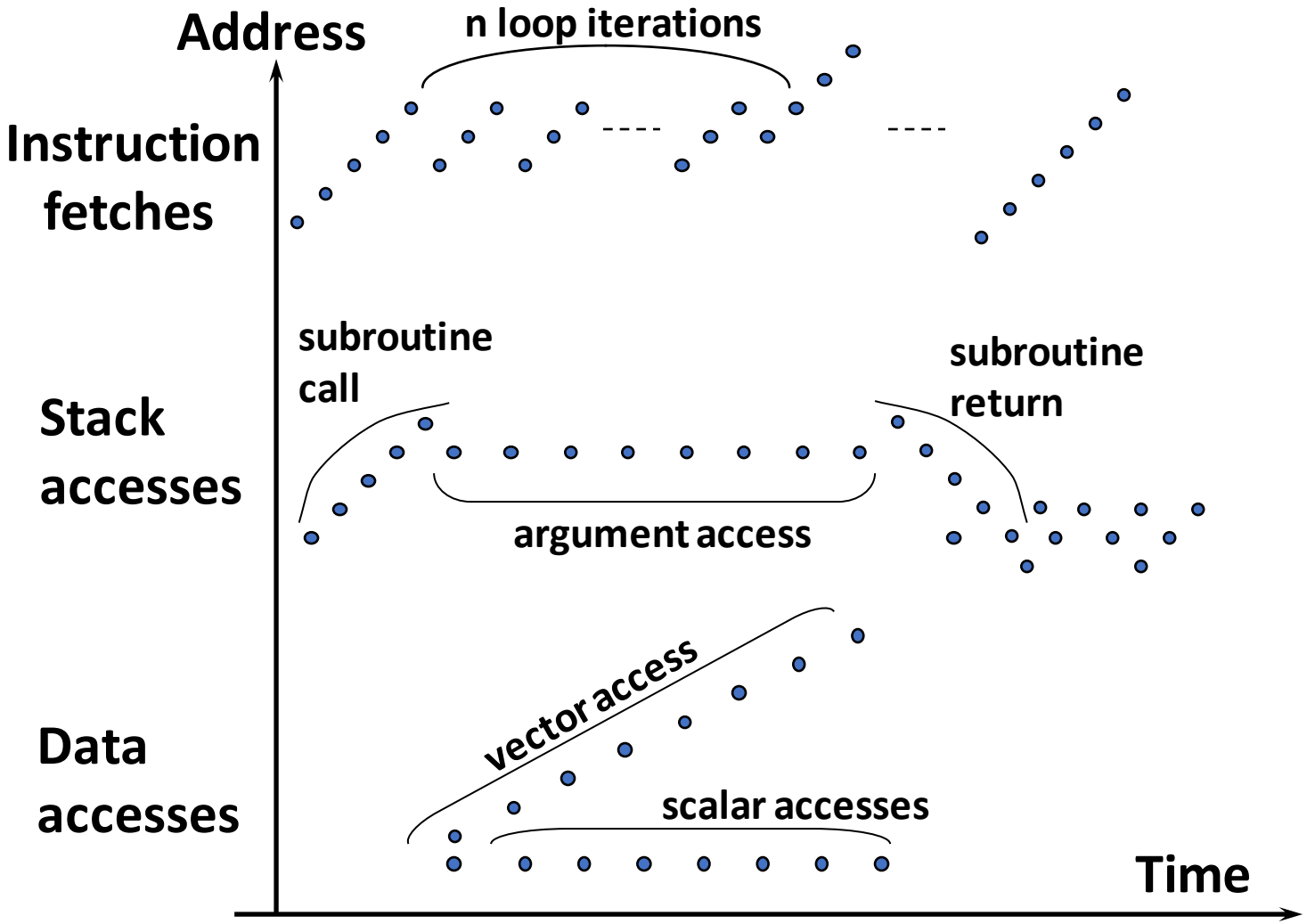
Donald J. Hatfield, Jeanette Gerald: Program Restructuring for Virtual Memory. IBM Systems Journal 10(3): 168-192 (1971)

Oh Yes Locality



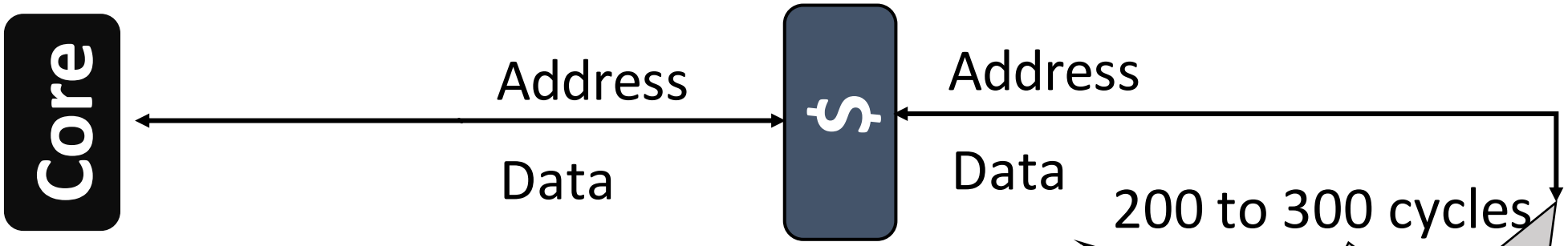
Donald J. Hatfield, Jeanette Gerald: Program Restructuring for Virtual Memory. IBM Systems Journal 10(3): 168-192 (1971)

Few Examples



Cache: 50K Feet View

North pole ☺

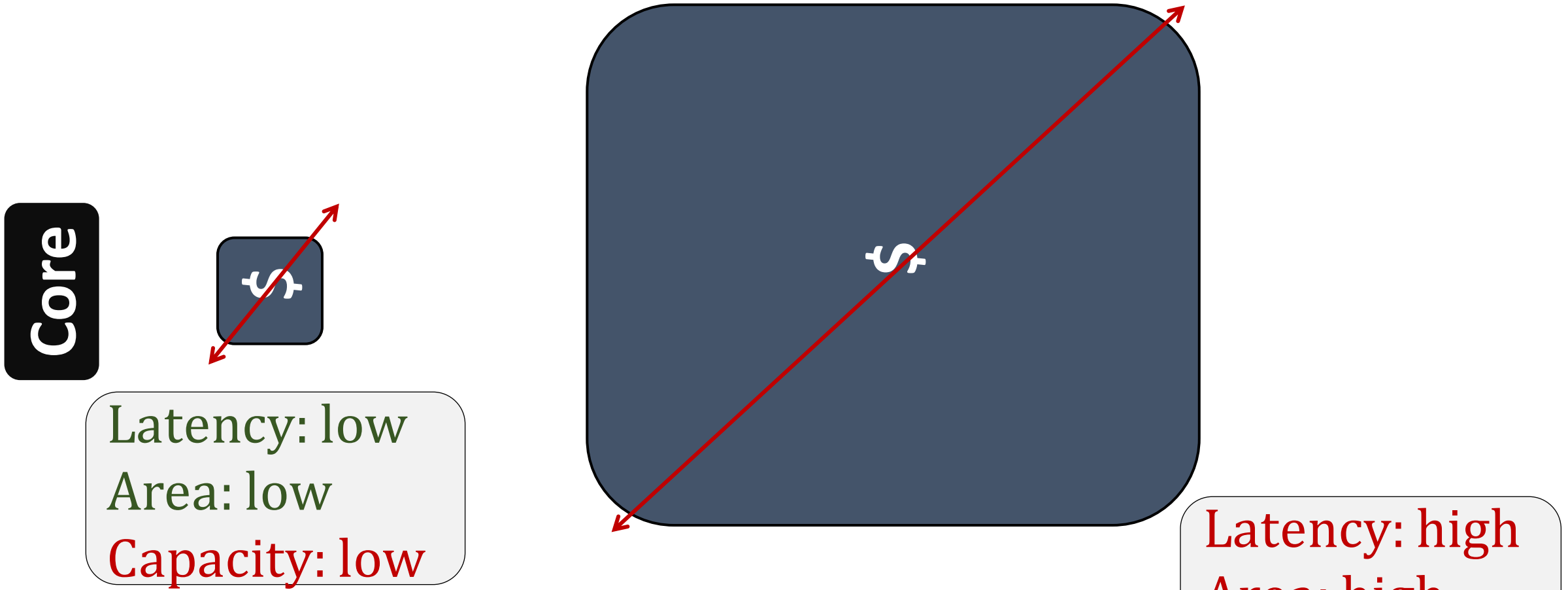


Caching is a *speculation* technique ☺
Works – if locality

Costly DRAM accesses☹

South pole ☺

Cache: How Small/Large? 1BHK/3BHK



Let's Pause

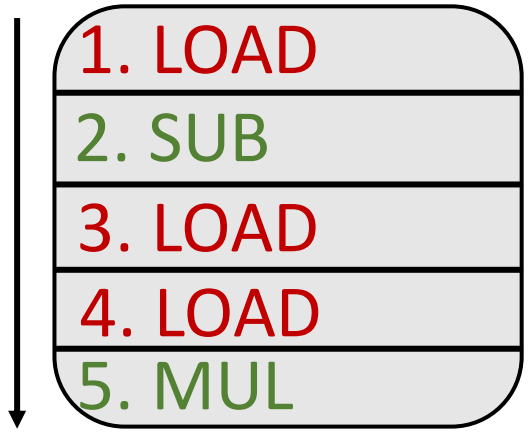
Cache: What should be the capacity **(size)** then?

Cache: implemented as registers, **SRAM array**, DRAM array?

Think about it.

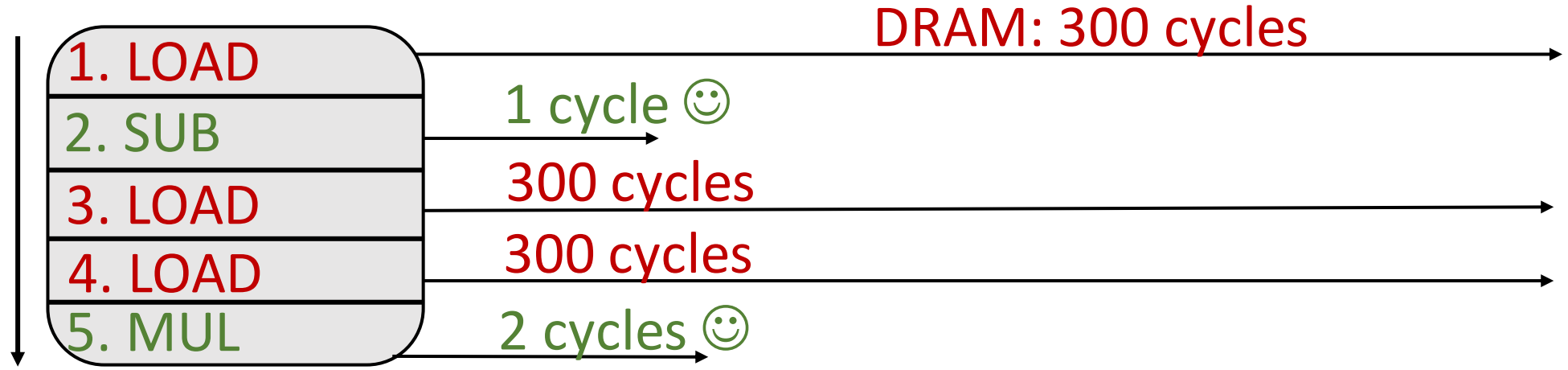
Let's ask the processor core

1K Feet View of an O3 core: A bit Deeper



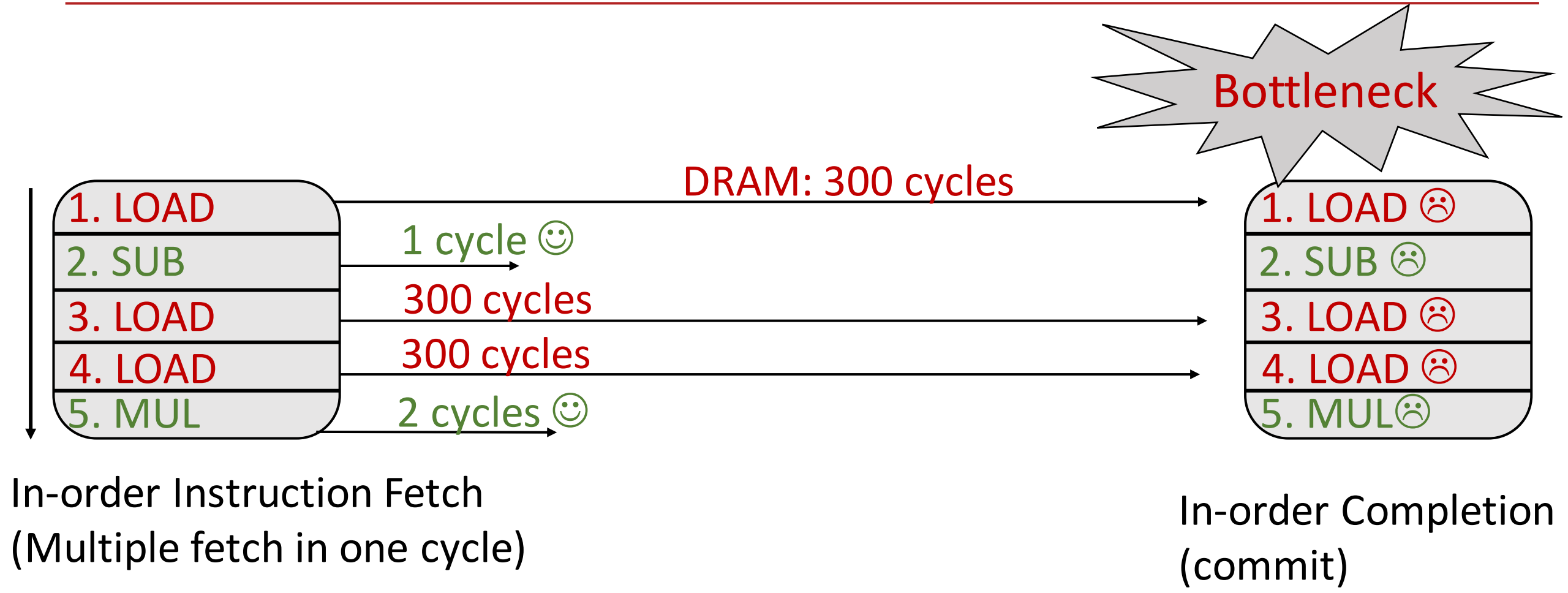
In-order Instruction Fetch
(Multiple fetch in one cycle)

1K Feet View of an O3 core: A bit Deeper



In-order Instruction Fetch
(Multiple fetch in one cycle)

Let's ask the Processor Core (the O3 one)

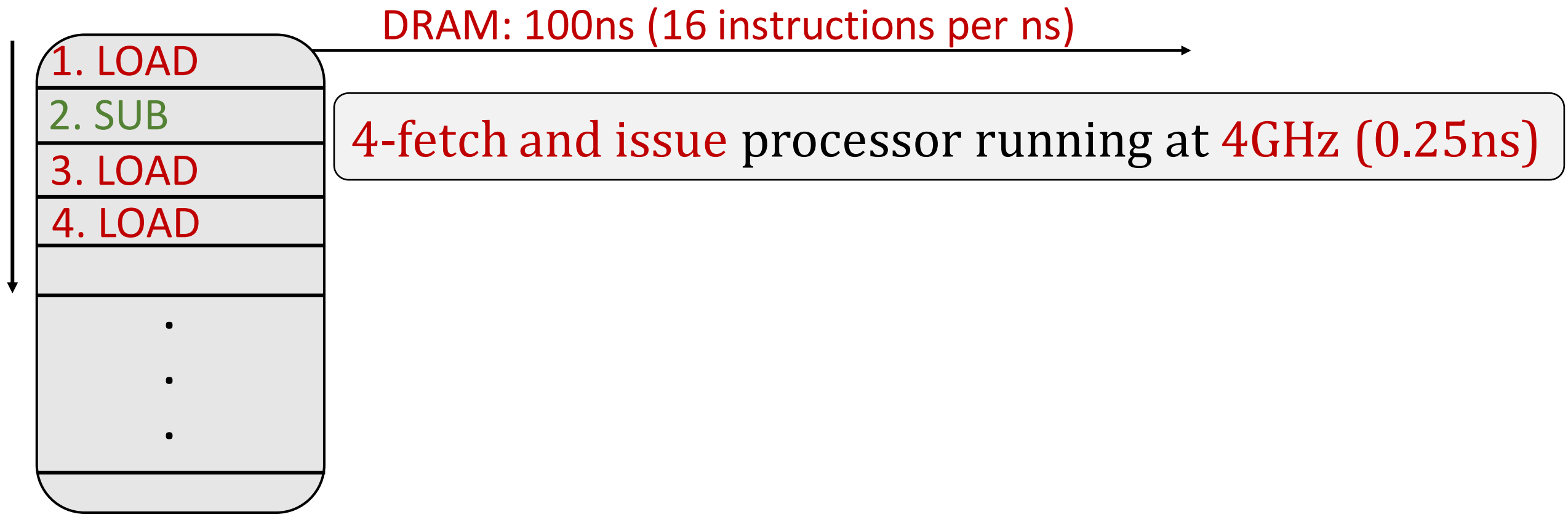


In-order Instruction Fetch
(Multiple fetch in one cycle)

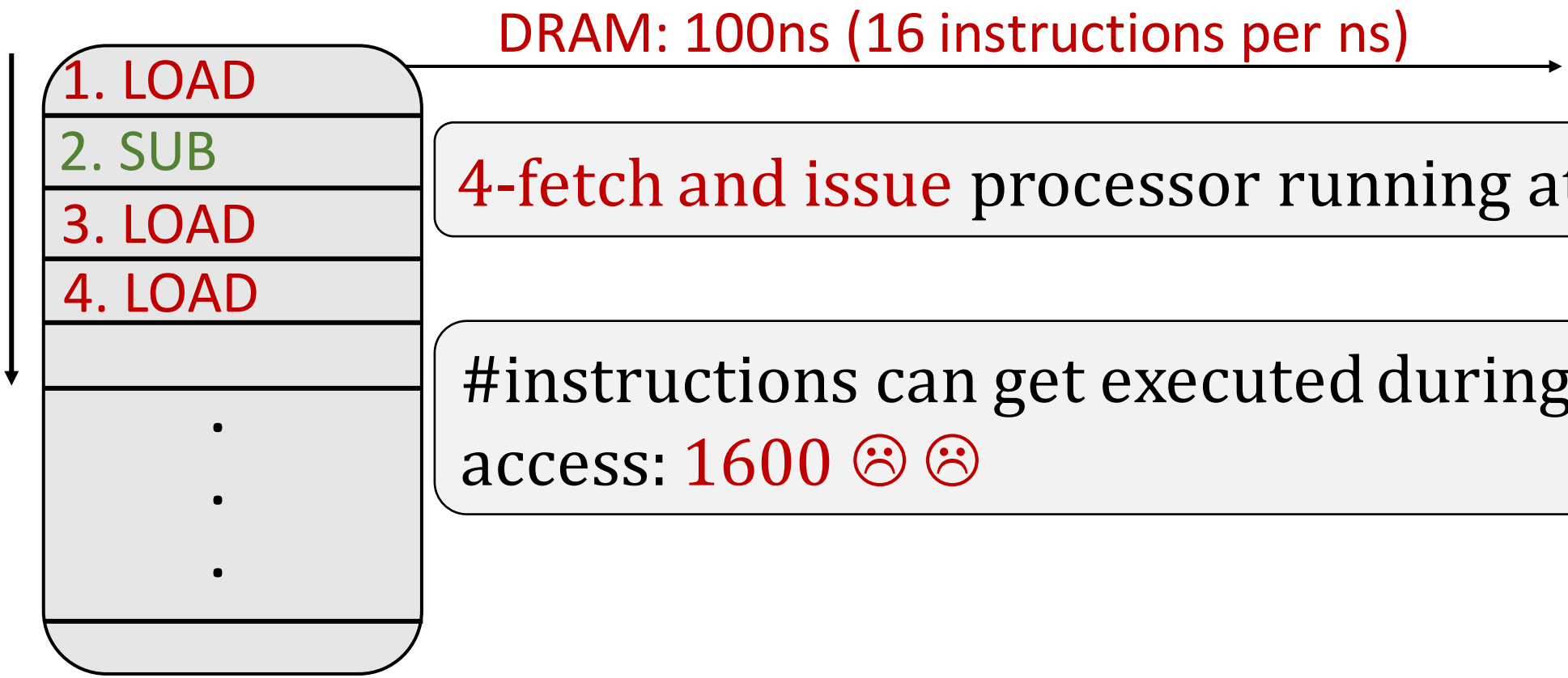
In-order Completion
(commit)

Processor core says all LOADs should take one cycle. Ehh!

Impact of One DRAM Access



Impact of One DRAM Access

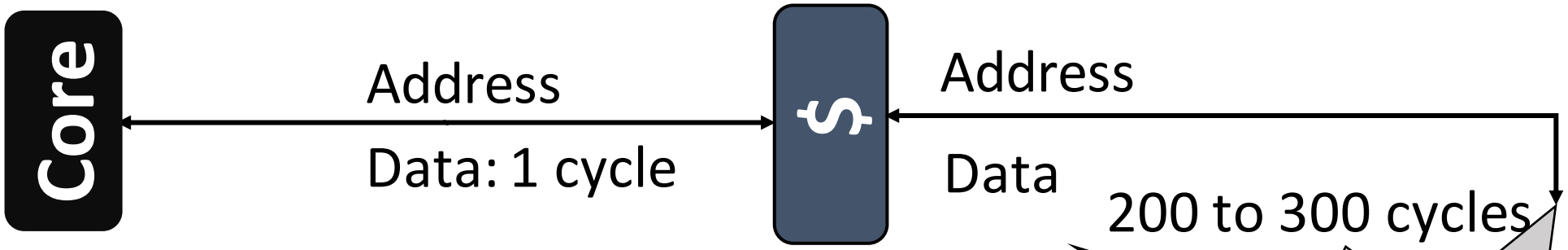


4-fetch and issue processor running at 4GHz (0.25ns)

#instructions can get executed during one DRAM access: 1600 ☹ ☹

Cache: With Latency

North pole ☺

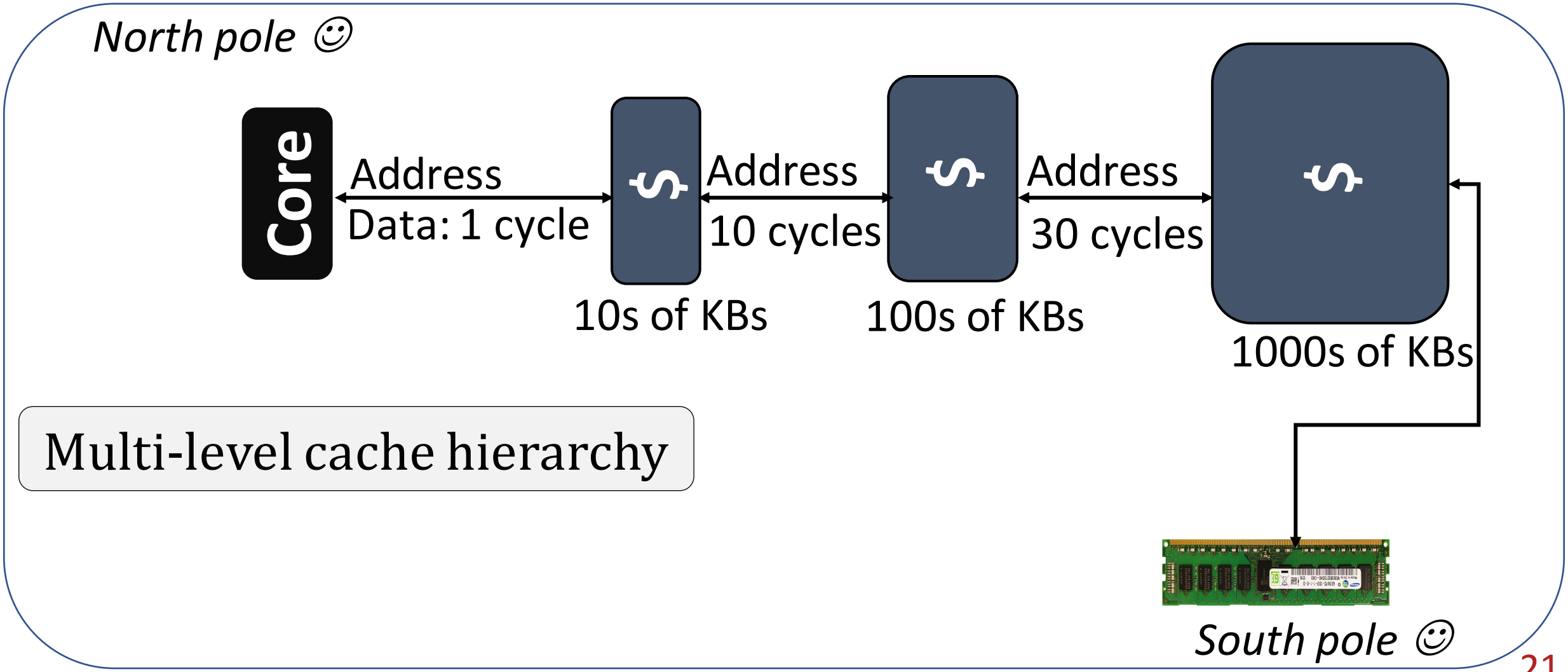


32 to 64KB $ will be available in one to four cycles ☹

Costly DRAM accesses☹

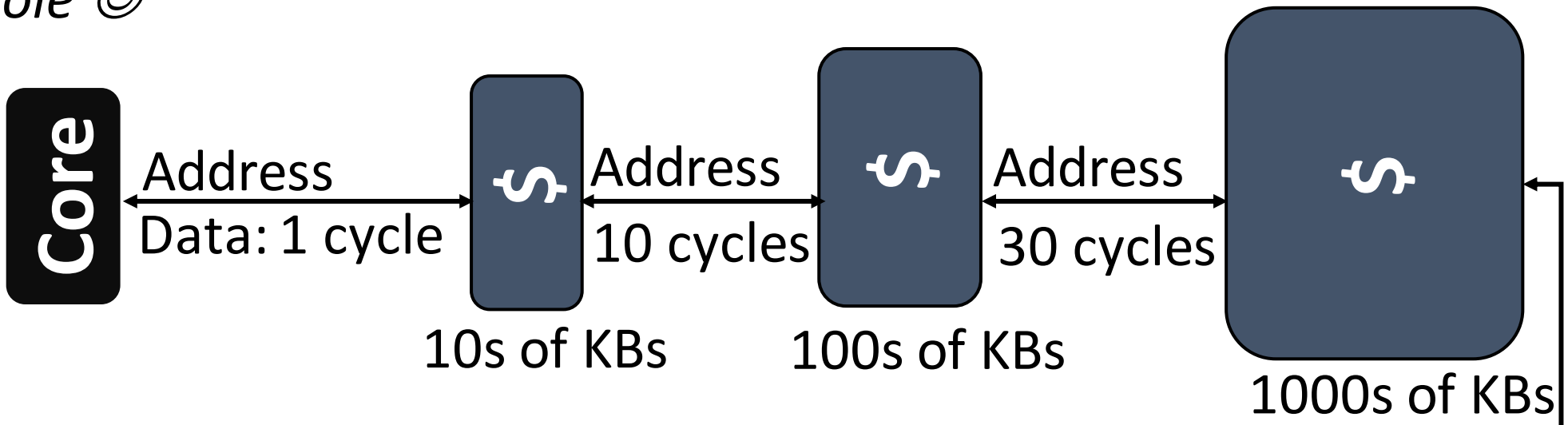
South pole ☺

Cache-Hierarchy: With Latency



Cache-Hierarchy: With Latency

North pole ☺



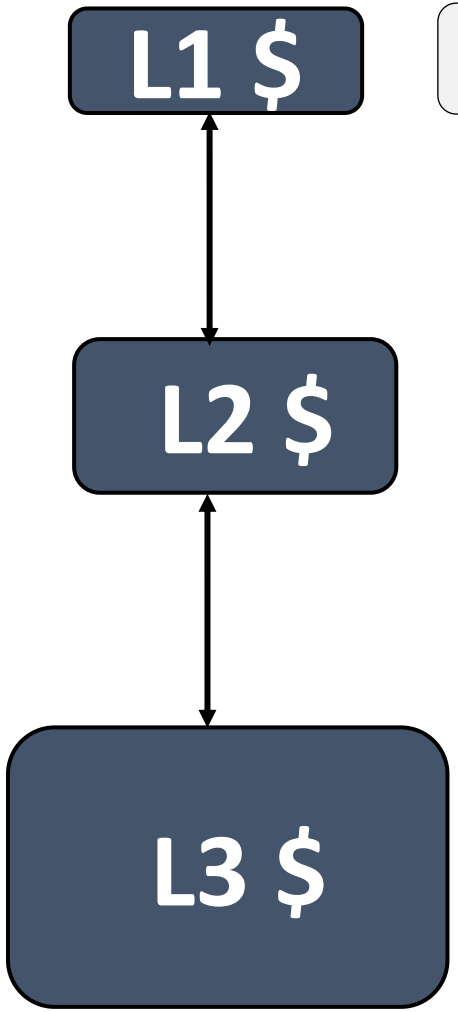
Multi-level cache hierarchy

How many levels ?

Total latency < DRAM latency

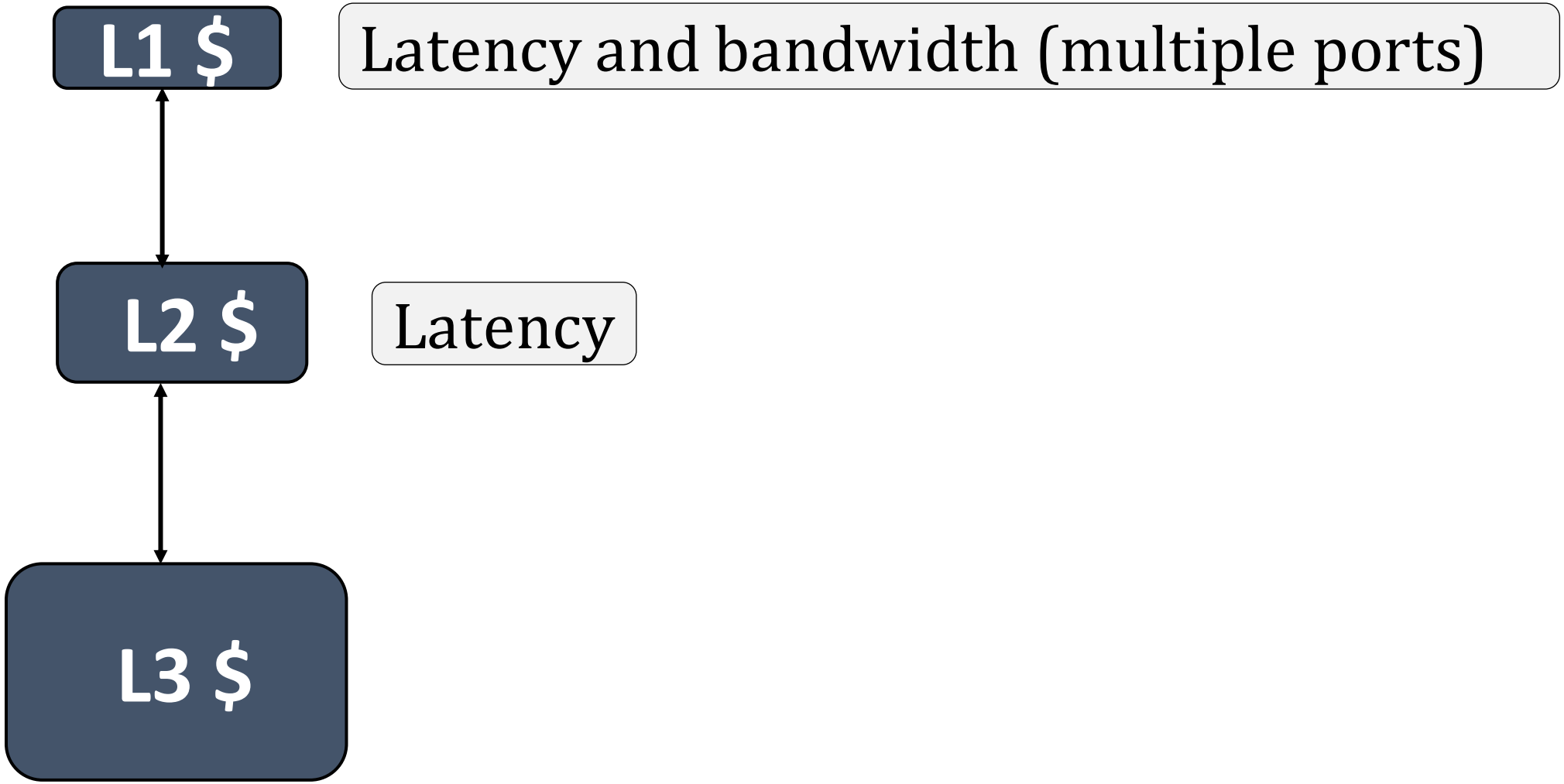
South pole ☺

Takeaway

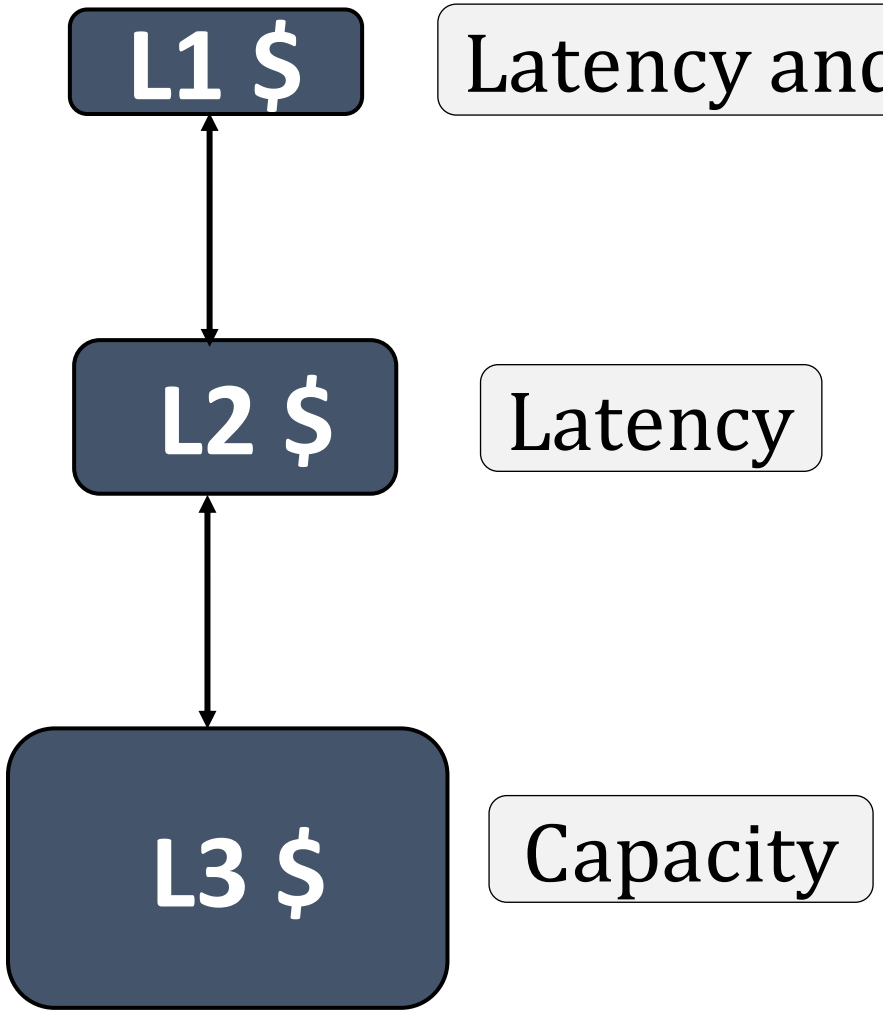


Latency and bandwidth (multiple ports)

Takeaway



Takeaway

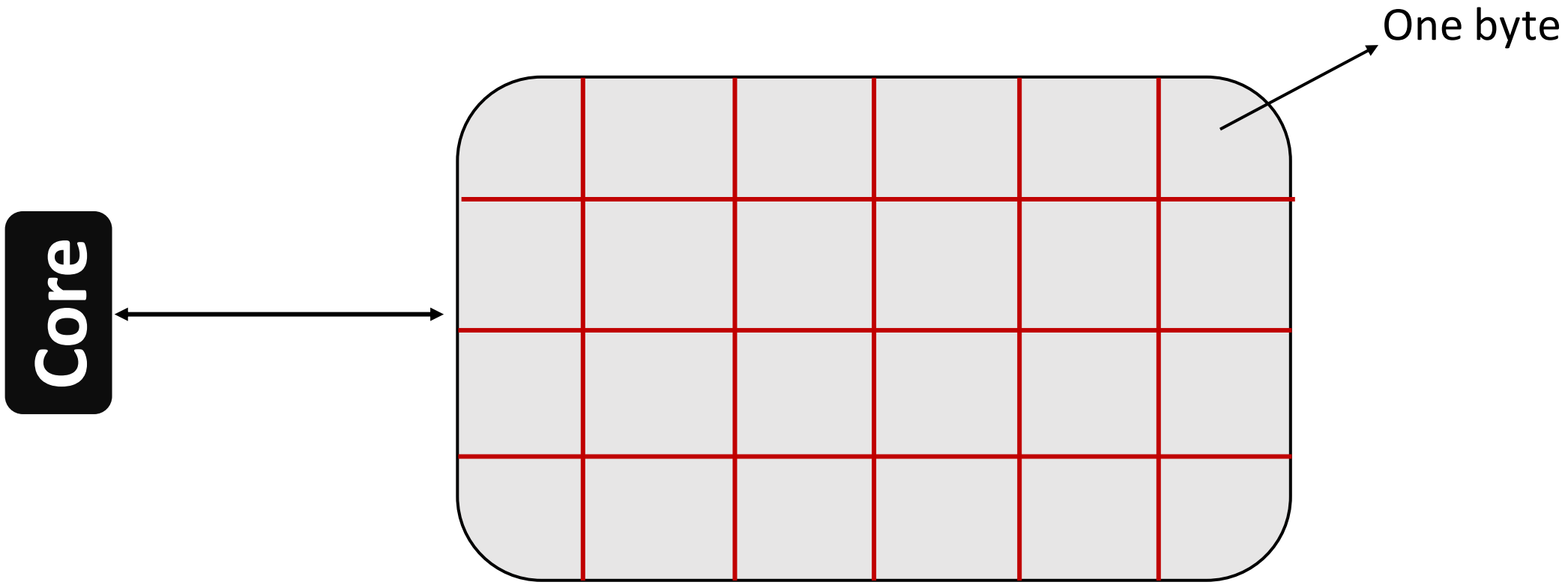


Latency and bandwidth (multiple ports)

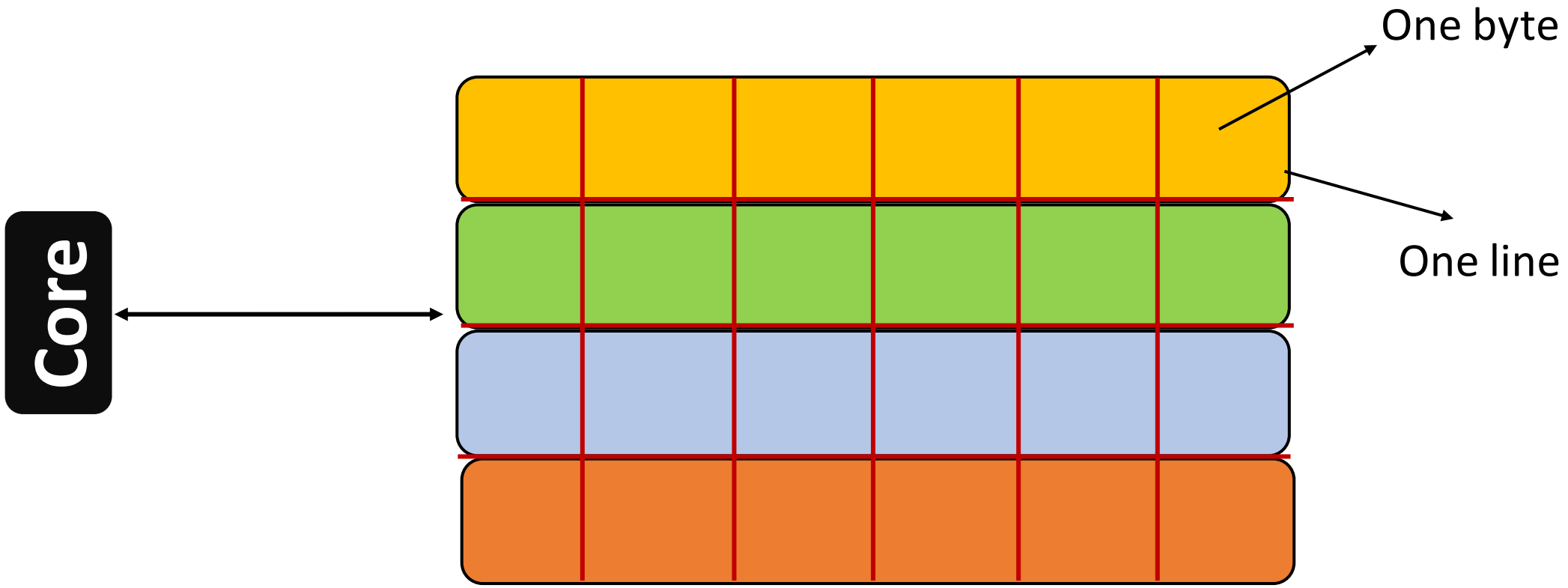
Latency

Capacity

How to Access Contents of a Cache?

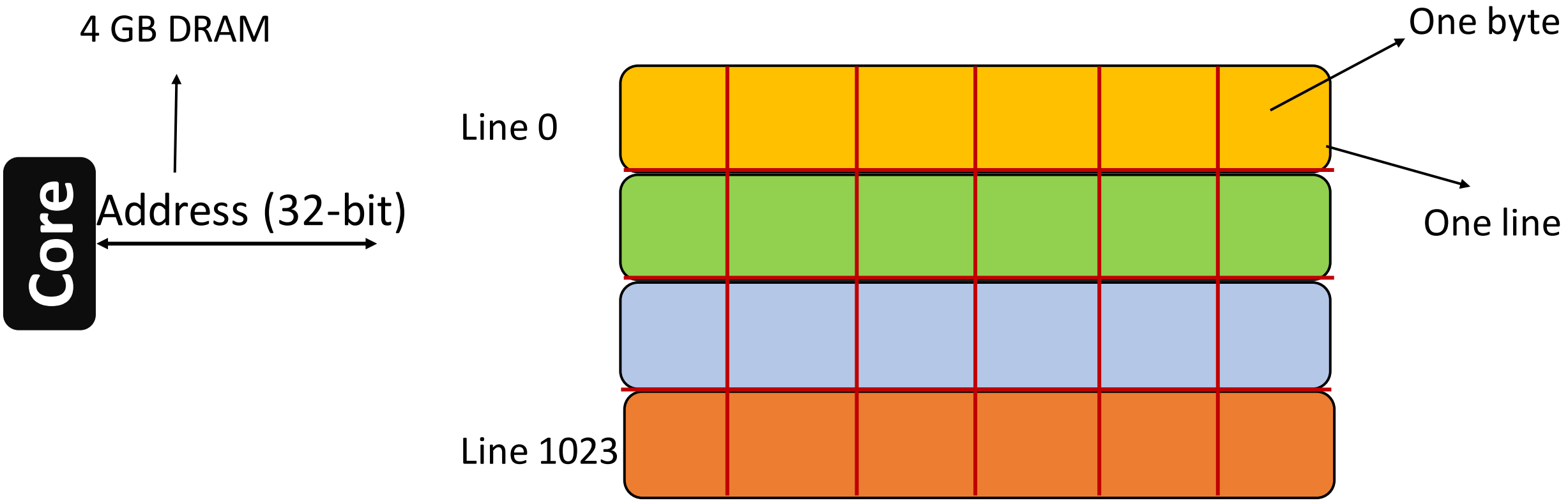


From Bytes to Blocks (Lines)

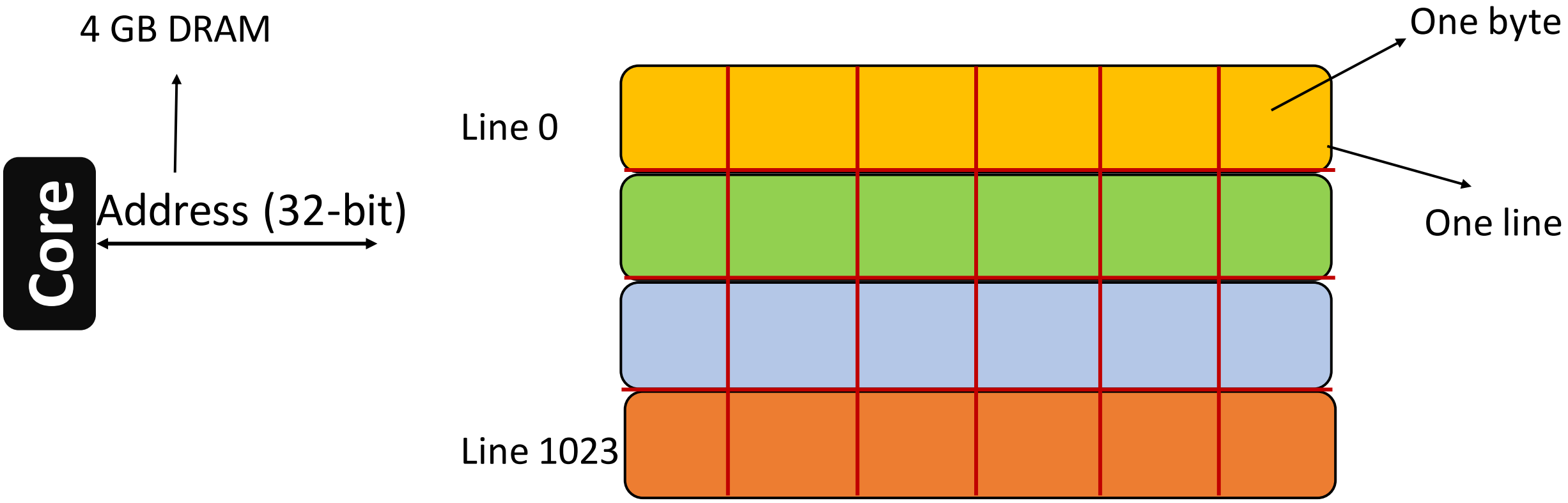


Typical line size: 64 to 128 Bytes

A Bit Deeper: Cache with 1024 lines each of 32B



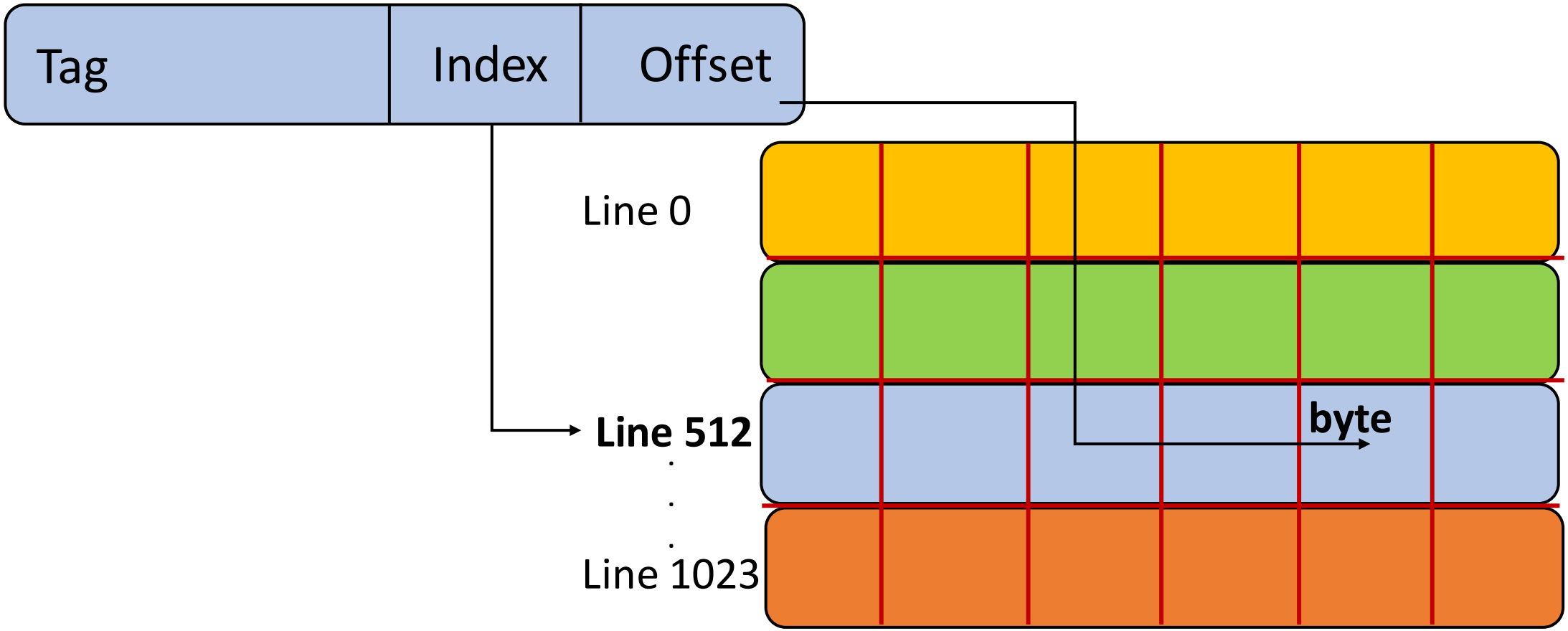
A Bit Deeper: Cache with 1024 lines each of 32B



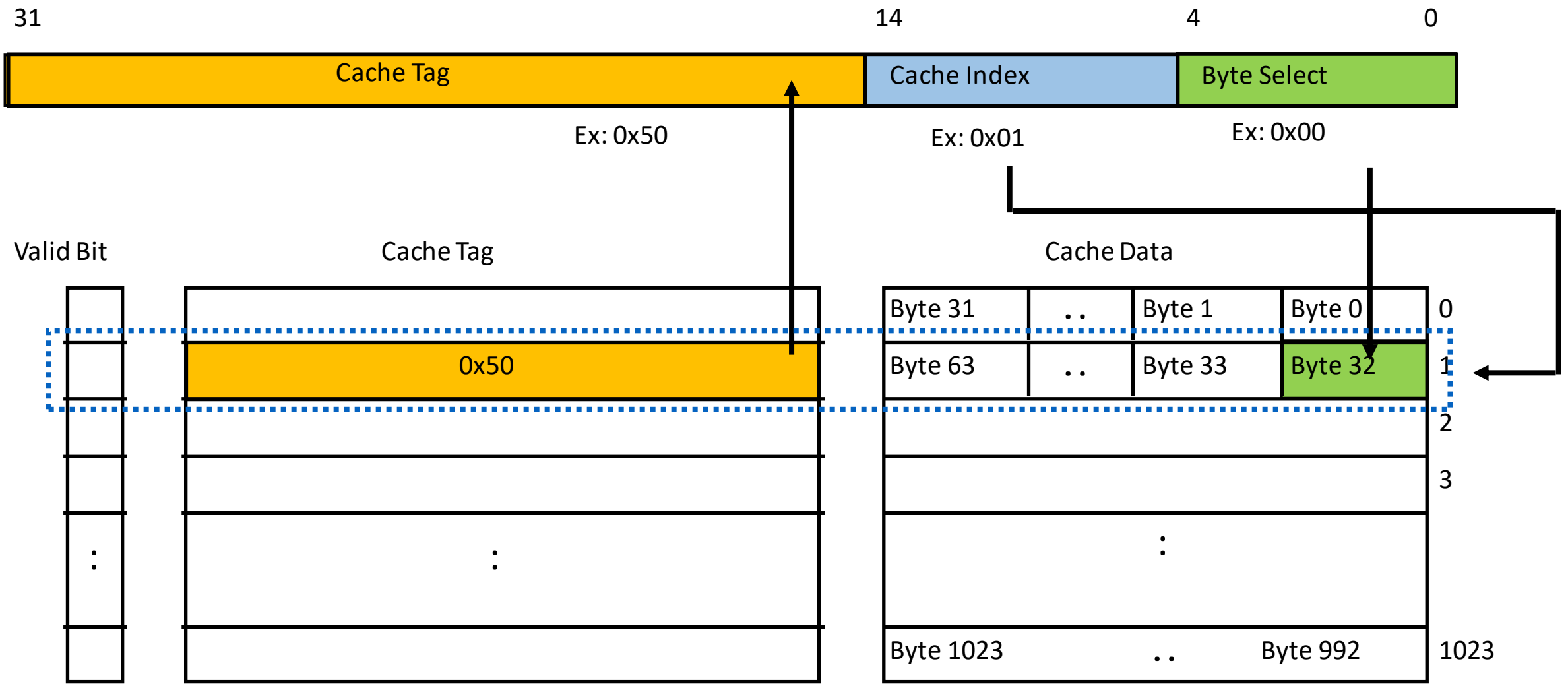
Line number (index): 10 bits

Byte offset (offset): 5 bits

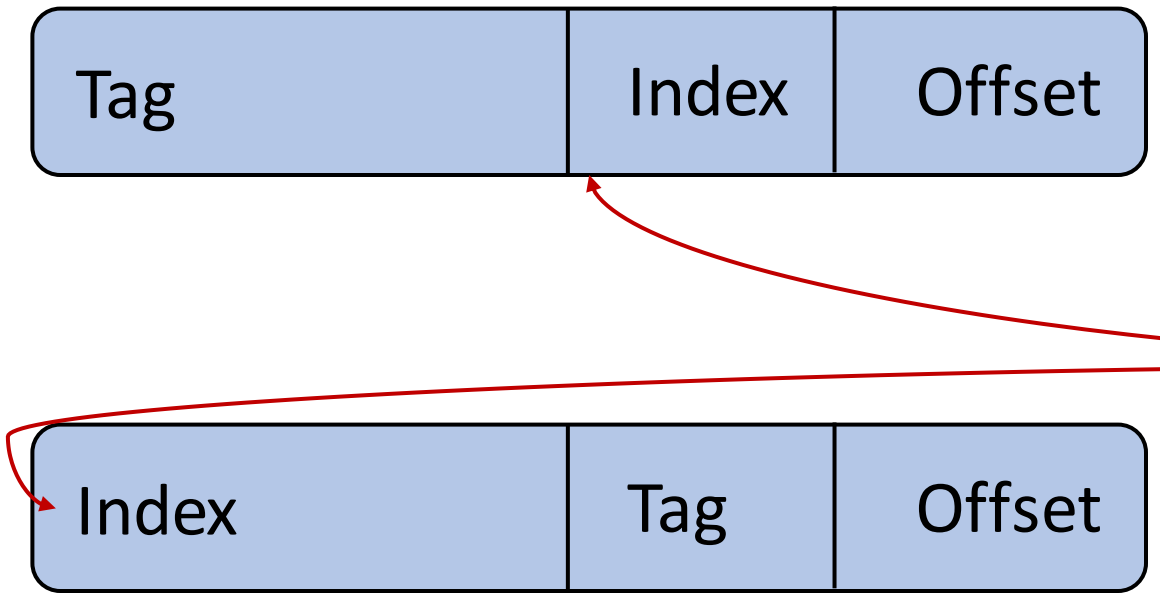
Directed Mapped Cache



Direct Mapped in Action

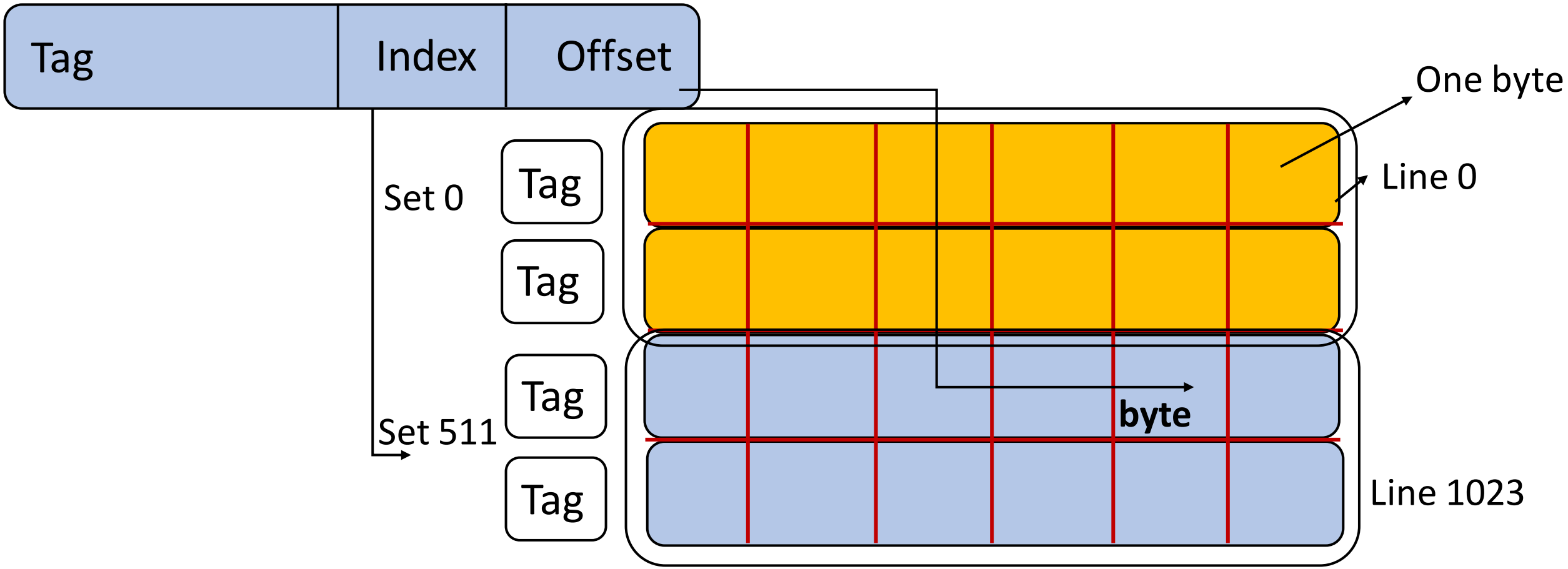


Food for Thought?

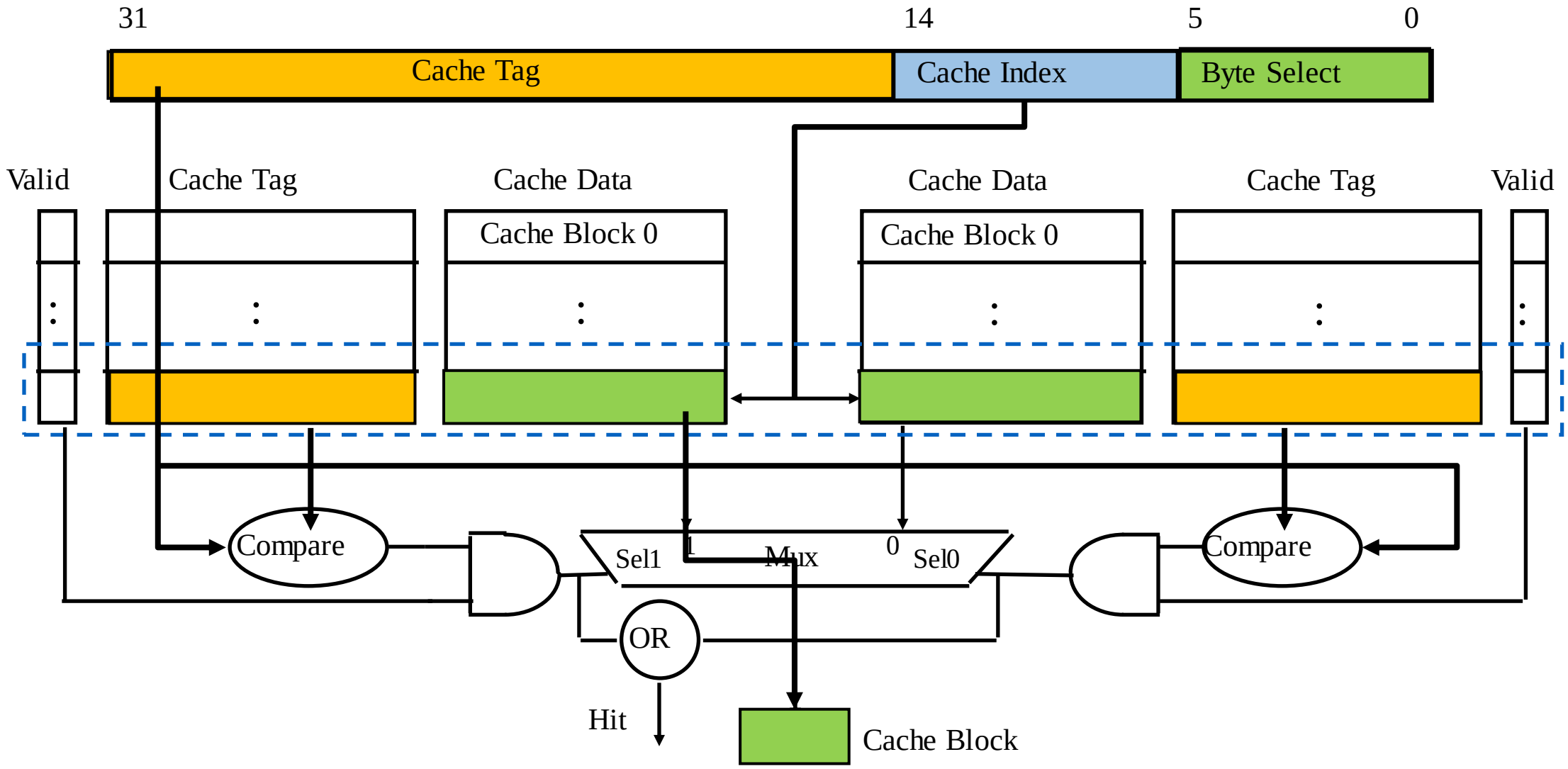



```
sum = 0;
for (i = 0; i < n; i++)
  sum += a[i];
return sum;
```

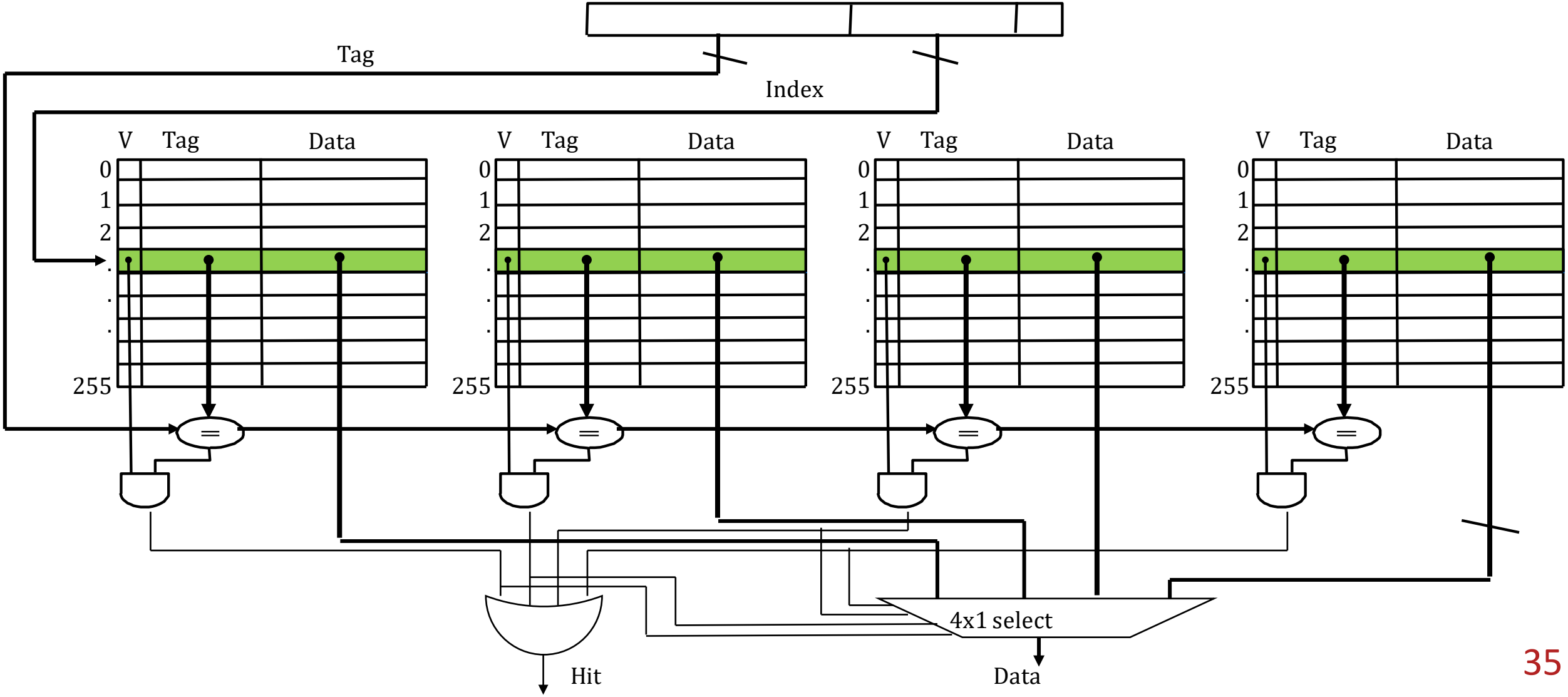
What if We Provide Multiple Ways?



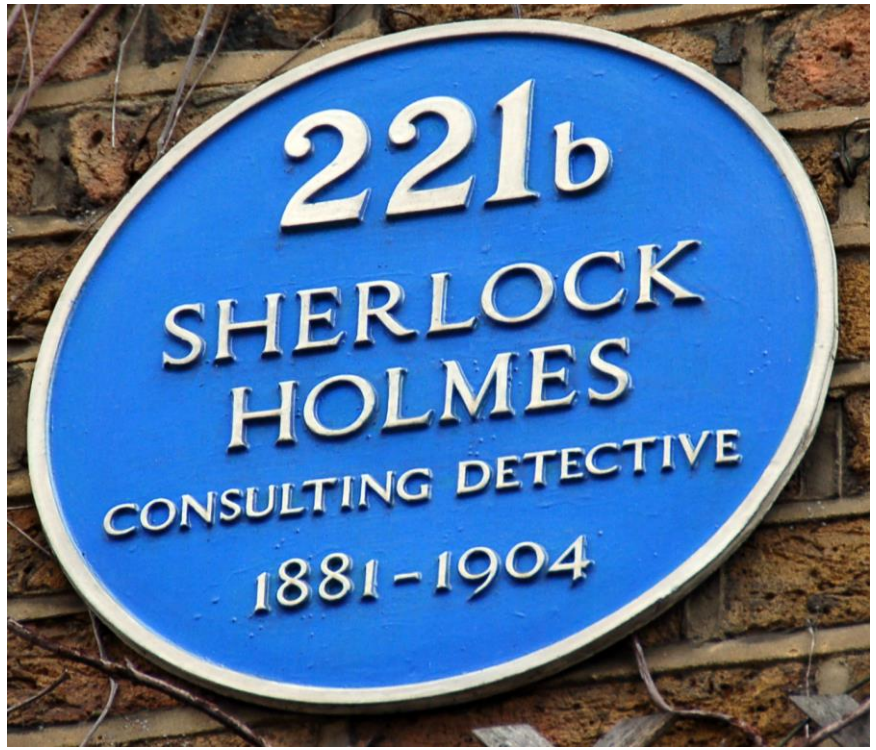
2-Way Associative Cache in Action



4-way : Just a Better Picture



A Bit Different Way

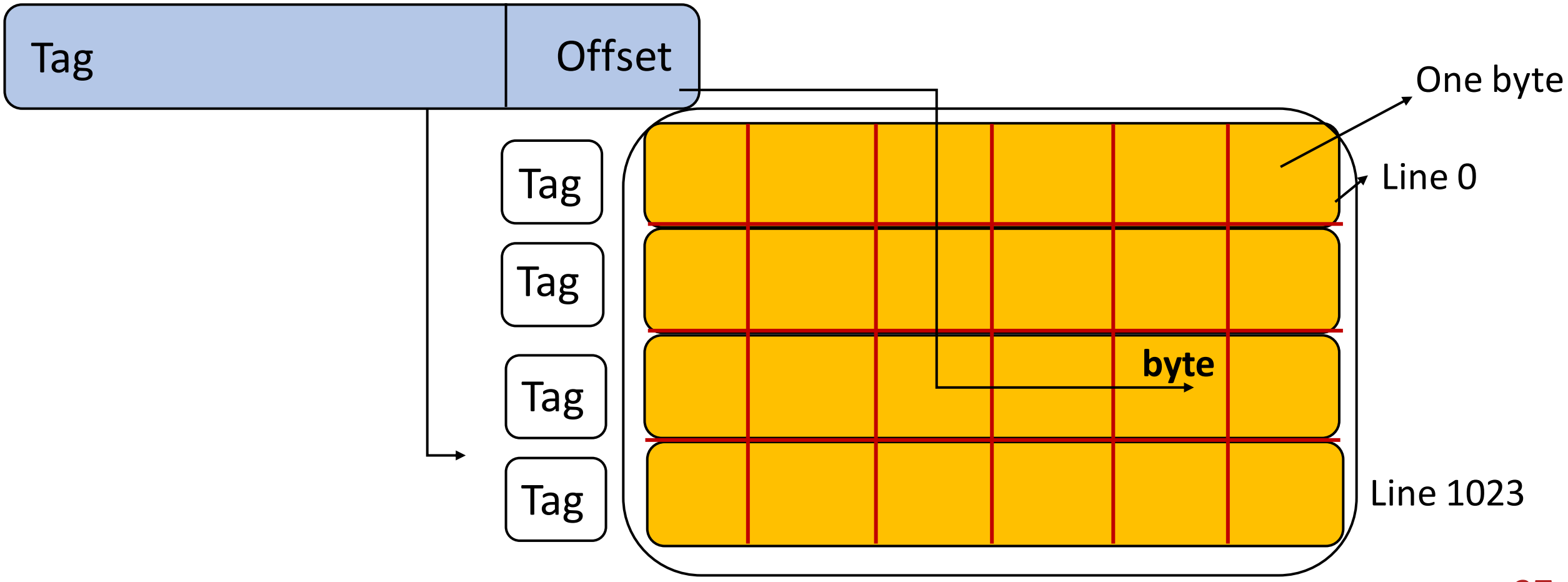


Baker Street: Cache Index ☺

221b: Tag bits ☺

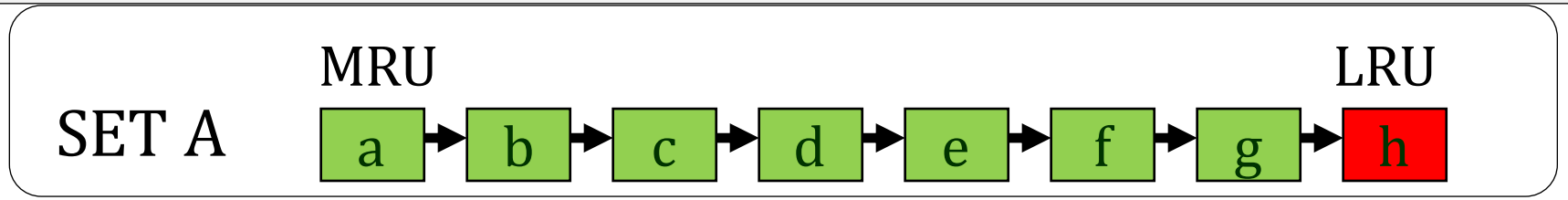
Sherlock Holmes: Byte offset ☺ ☺

The Extreme: One Cache is One Set (Fully-assoc.)

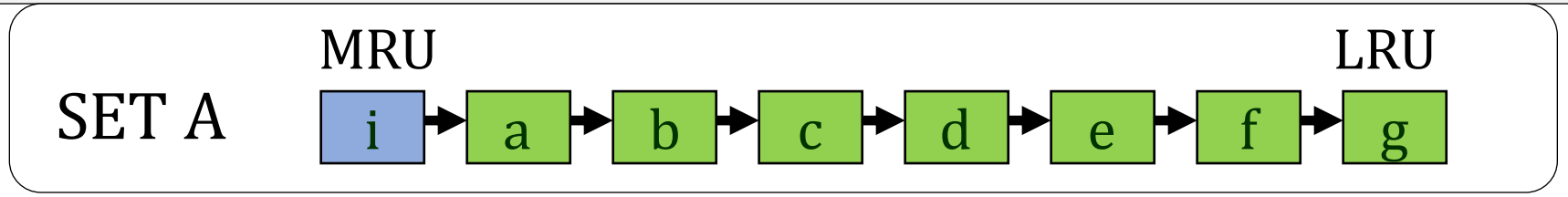


On a Miss: Replacement Policy

Cache Eviction Policy: On a miss (block *i*), which block to evict (replace)?



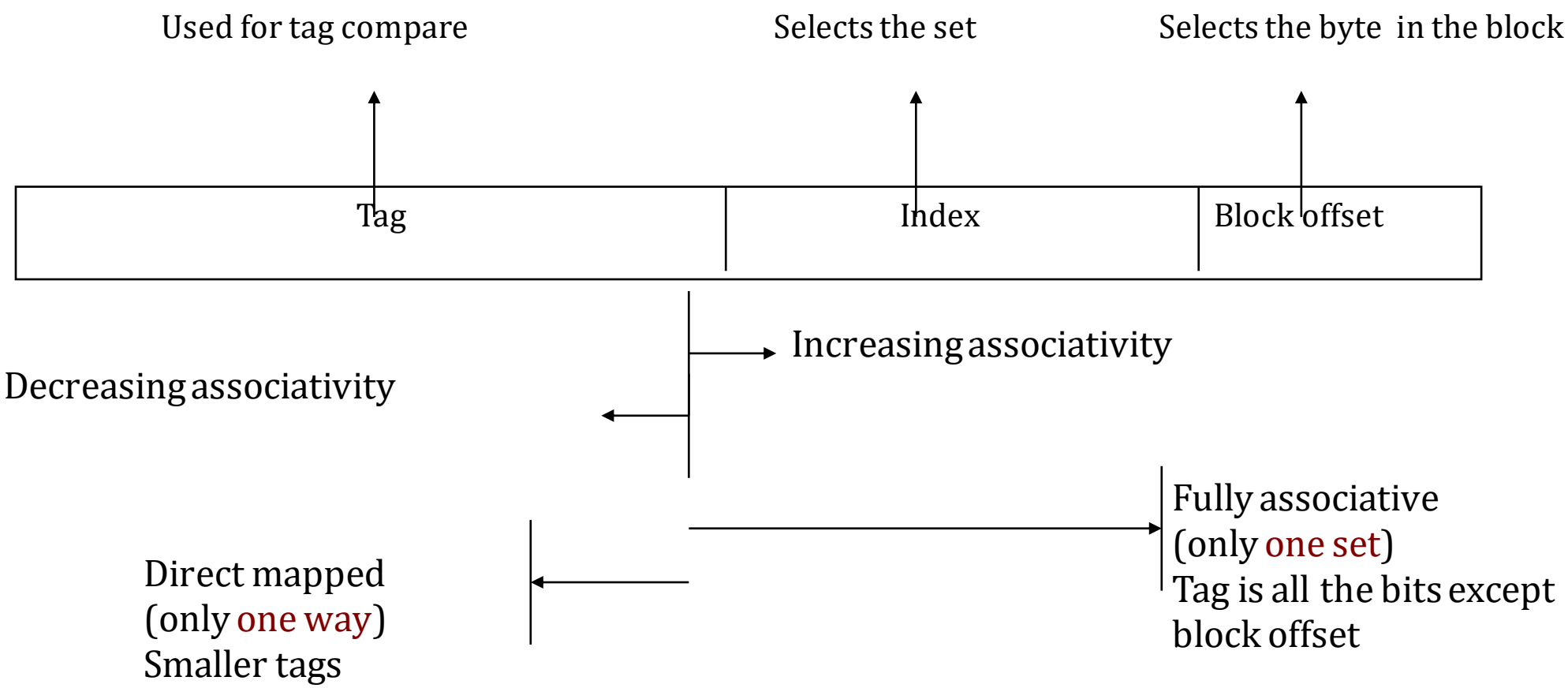
Cache Insertion Policy: New block *i* inserted into MRU.



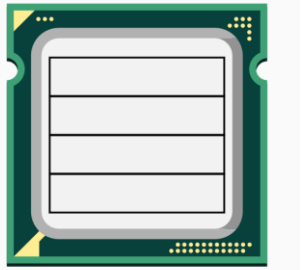
Cache Promotion Policy: On a future hit (block *i*), promote to MRU

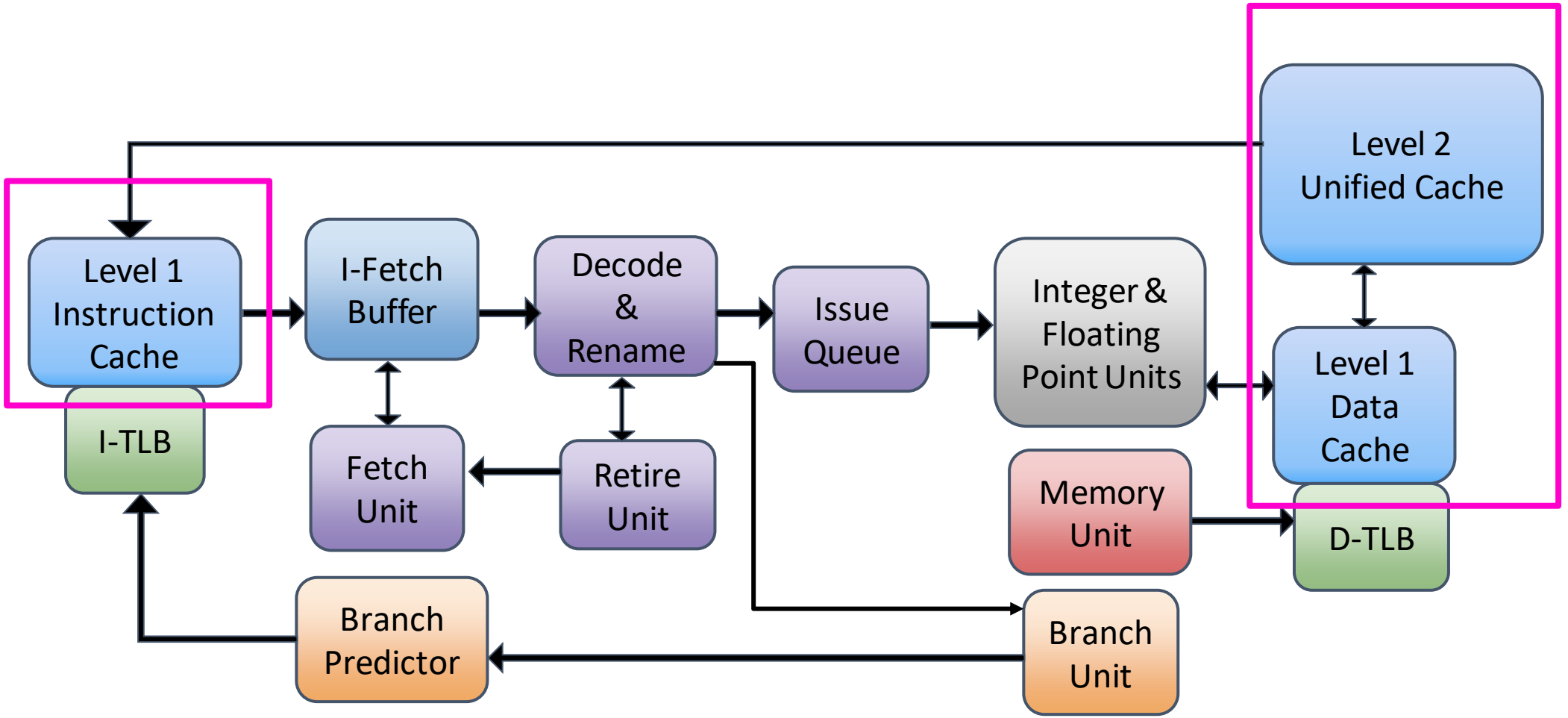
LRU causes thrashing when working set > cache size, RRIP based policies: commercial machines

A Bit More

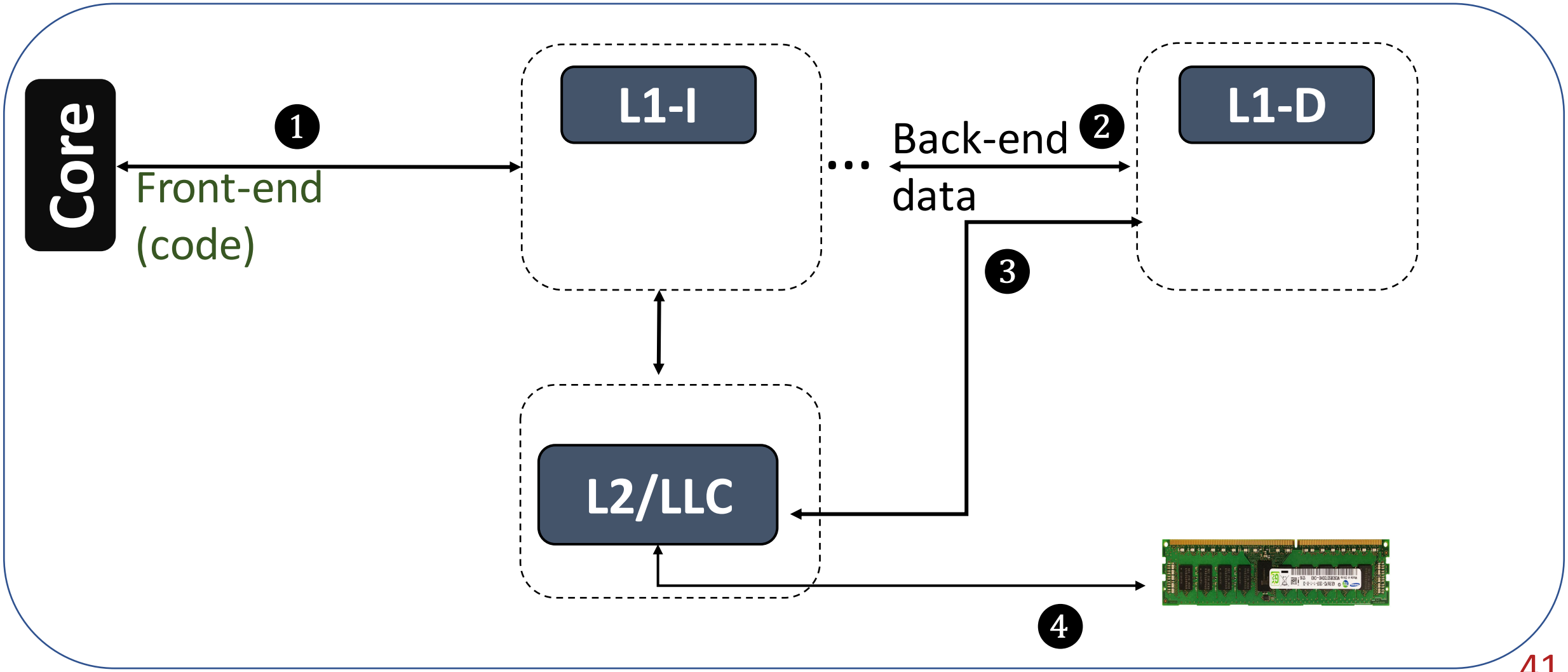

Used for tag compare
Selects the set
Selects the byte in the block
Tag
Index
Block offset
Increasing associativity
Decreasing associativity
Fully associative
(only one set)
Tag is all the bits except
block offset
Direct mapped
(only one way)
Smaller tags

Where Are the Caches?

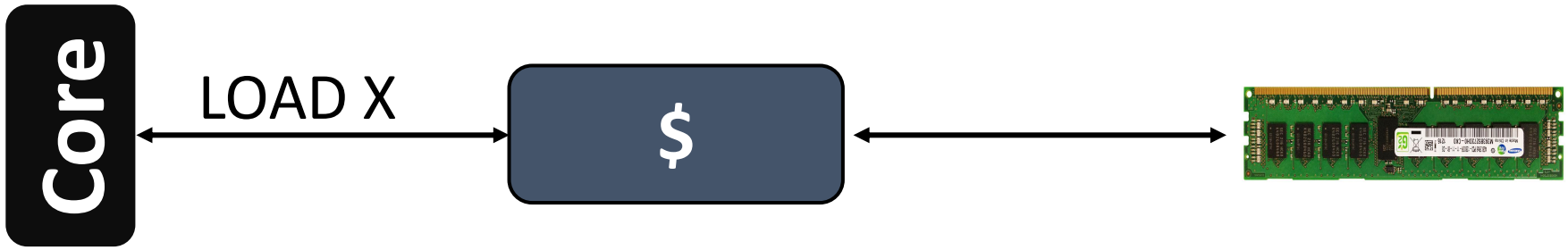




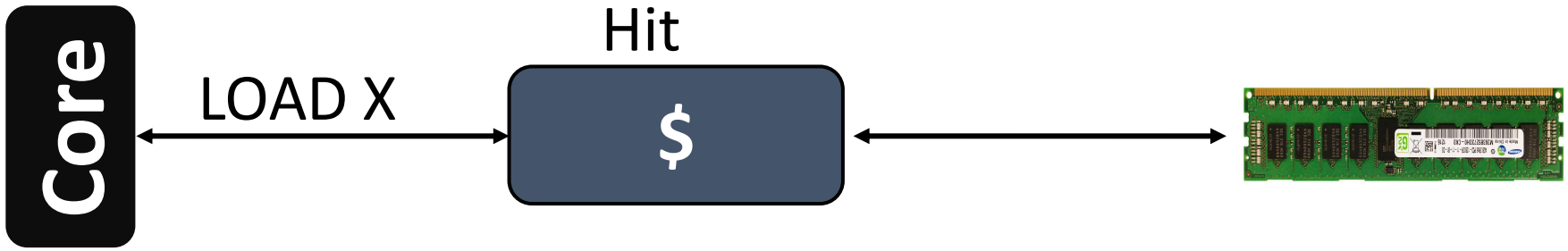
A Bit Lighter View



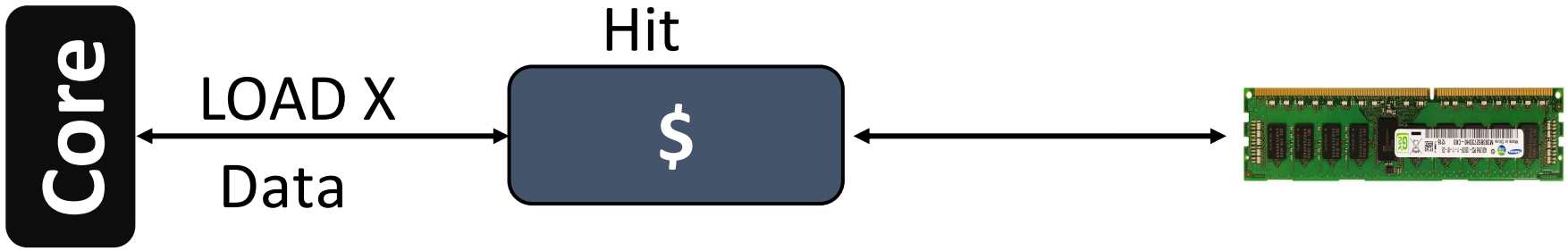
Core, Cache, And DRAM Interaction



Core, Cache, And DRAM Interaction

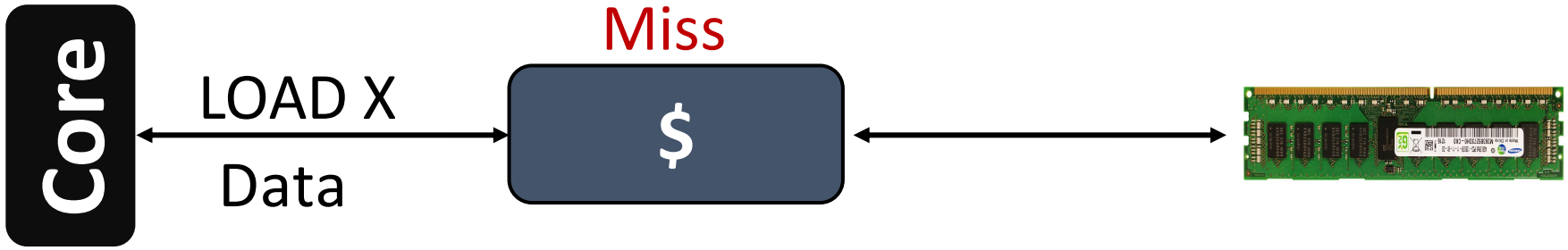


Core, Cache, And DRAM Interaction



Few cycles

Core, Cache, And DRAM Interaction

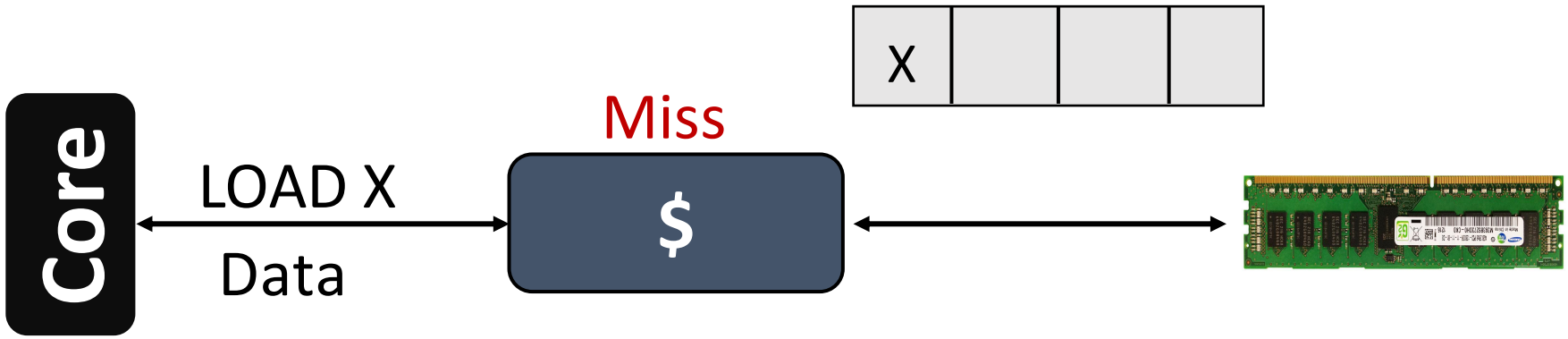


100s of cycles

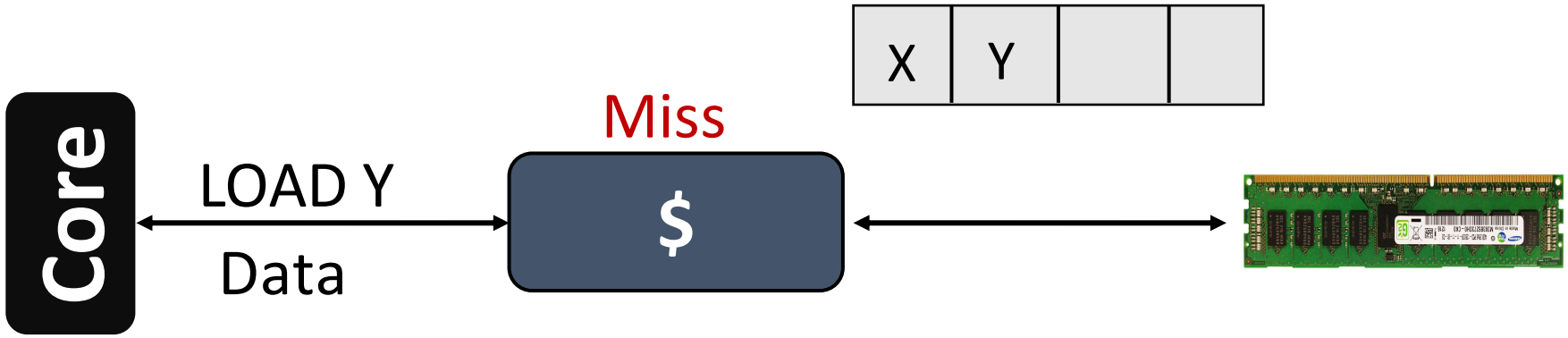
I am an out-of-order core

One cache miss and can't handle anymore misses

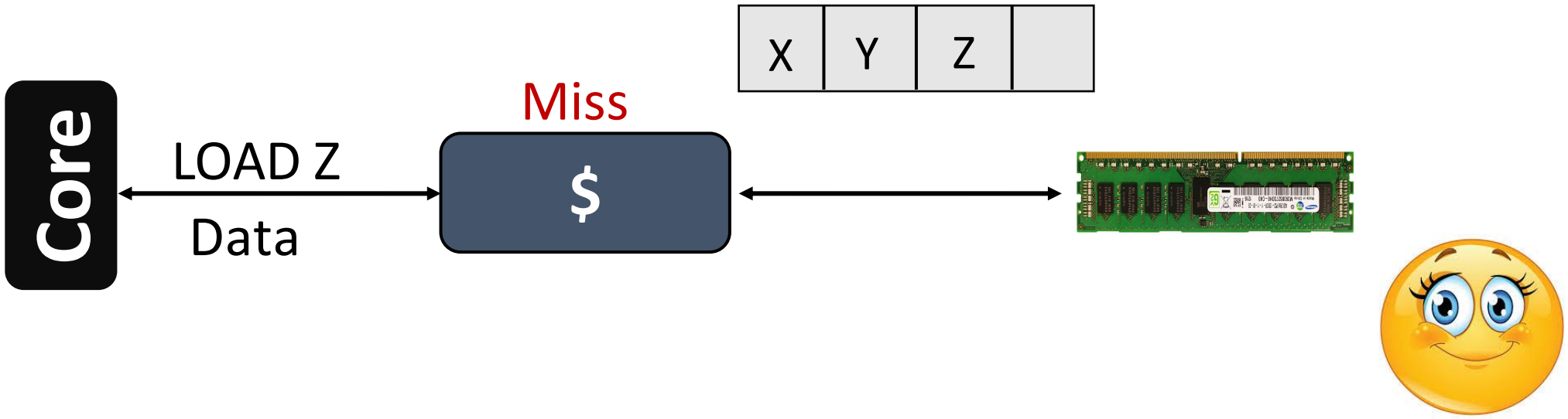
World of MSHRs (Miss Status Holding Registers)



World of MSHRs (Miss Status Holding Registers)

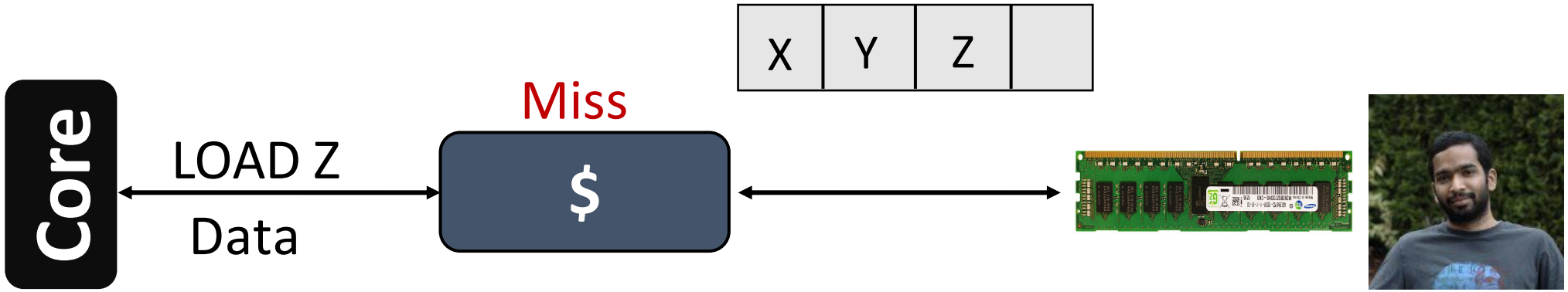


World of MSHRs (Miss Status Holding Registers)



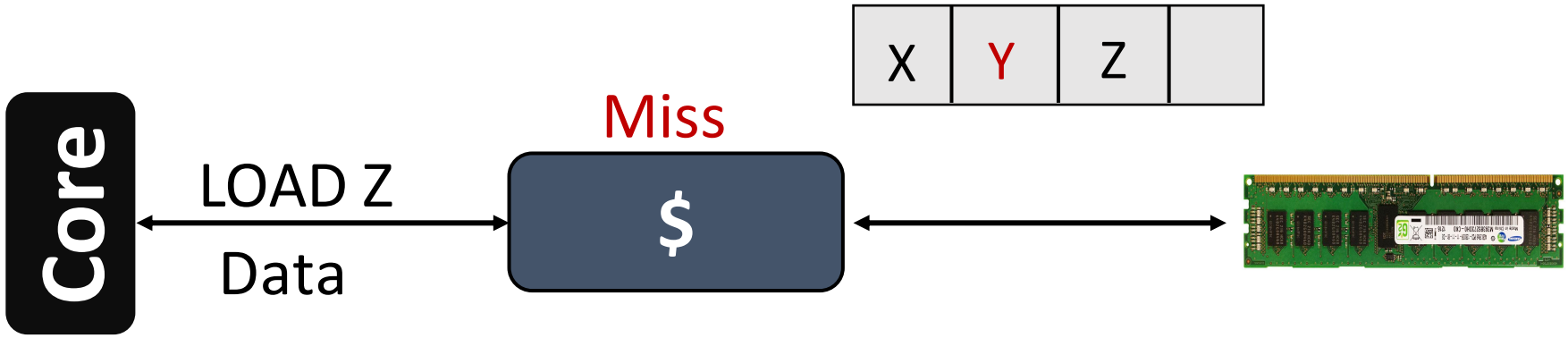
K-entry MSHR allows K outstanding misses from outer levels of memory hierarchy: provides memory-level parallelism

World of MSHRs (Miss Status Holding Registers)



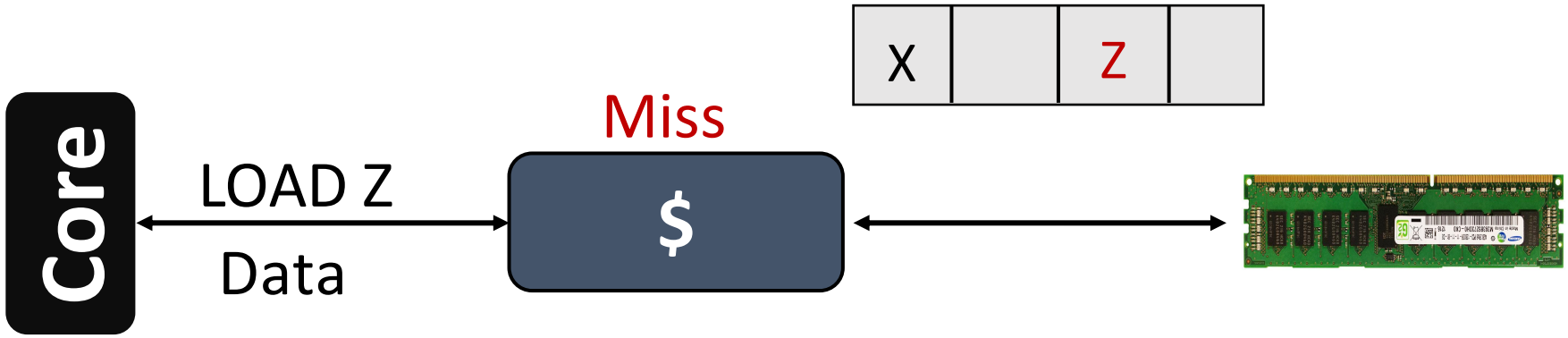
DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

World of MSHRs (Miss Status Holding Registers)



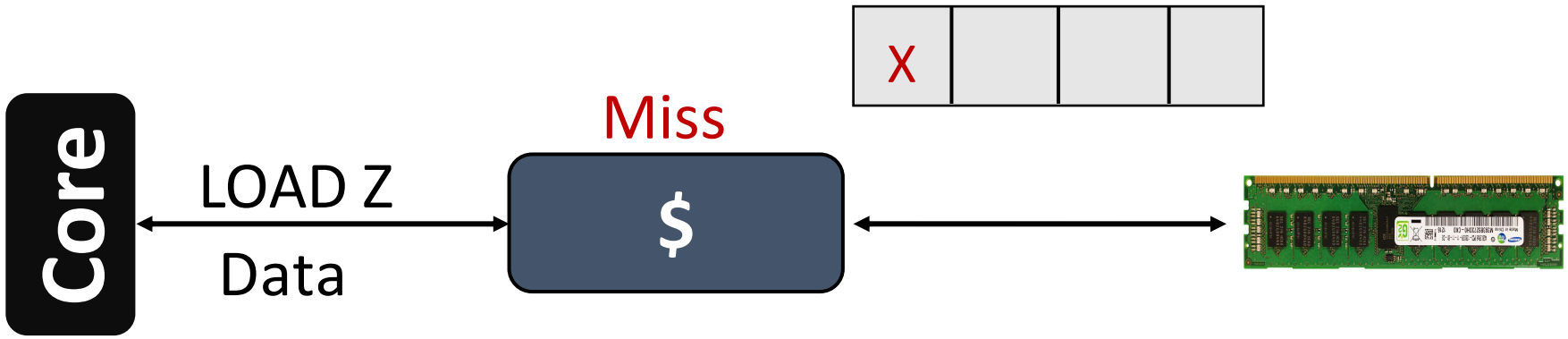
DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

World of MSHRs (Miss Status Holding Registers)



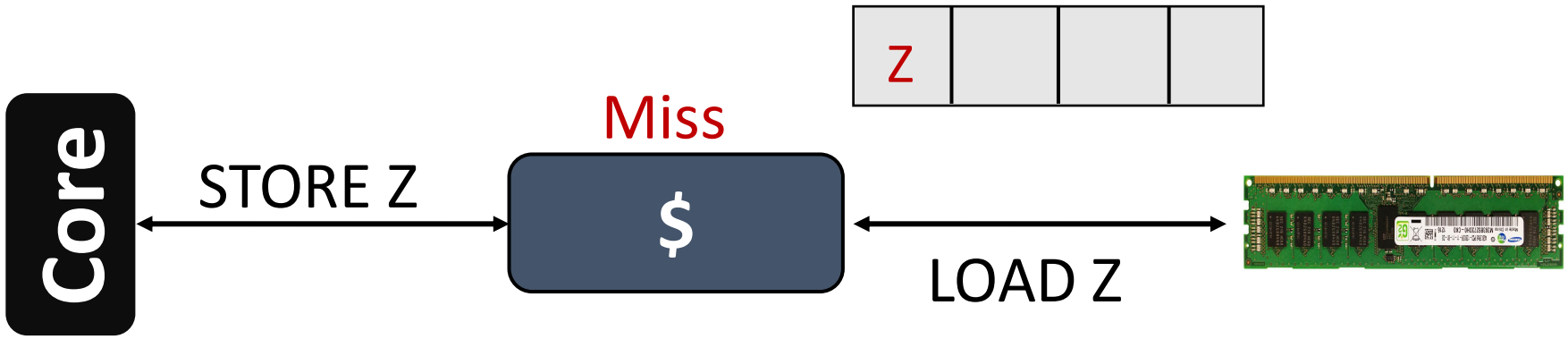
DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

World of MSHRs (Miss Status Holding Registers)

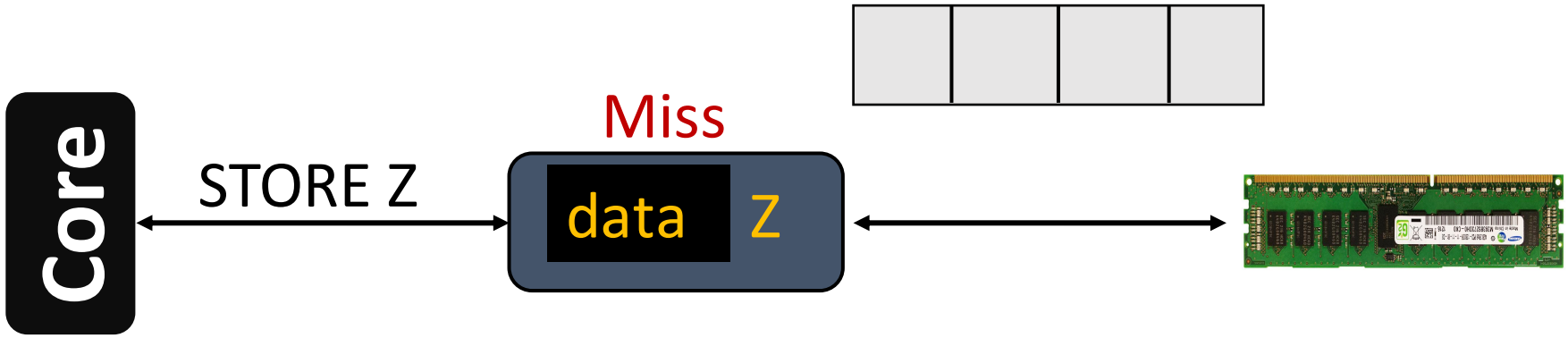


DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

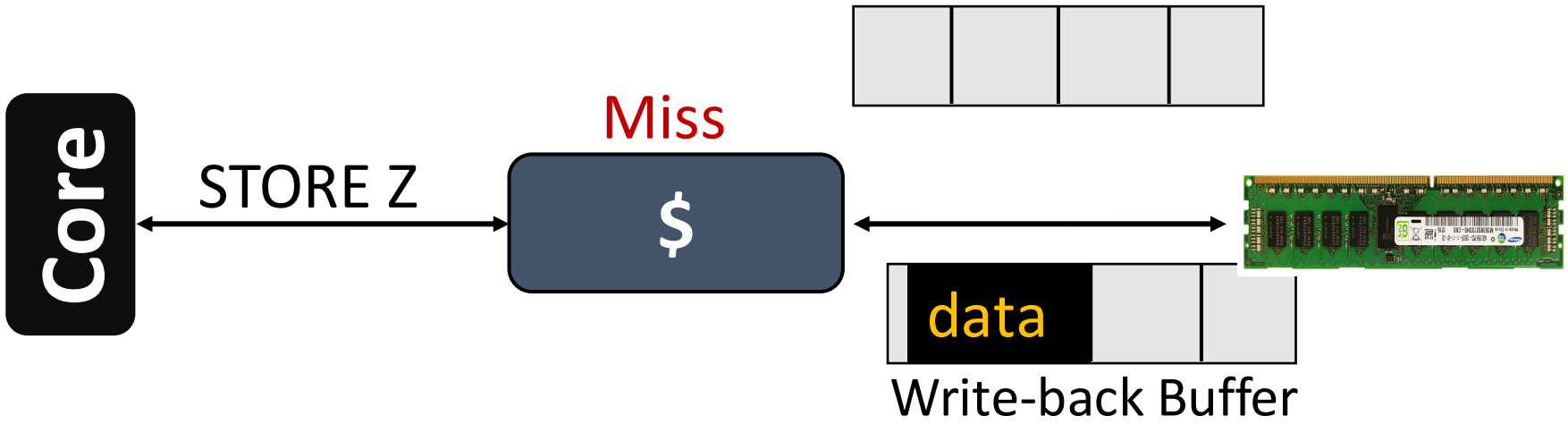
What About Writes (STOREs)?



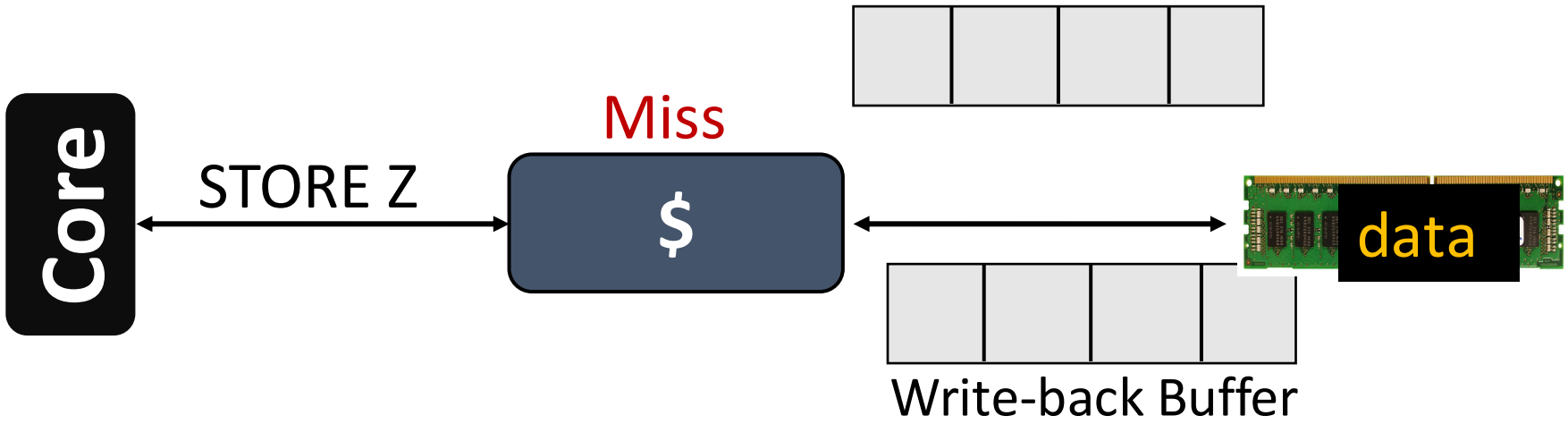
What About Writes (STOREs)?



What About Writes (STOREs)?



What About Writes (STOREs)?



Write-back cache

In general, STOREs are not critical for performance. But why?

Misses (3C)

Cold Miss: cache starts empty and this is the first reference

Conflict Miss: Many mapped to the same index bits

Capacity Miss: Cache size is not sufficient

Coherence Miss: in Multi-core systems, only [not I/O coherence]

Food For Thought

Given a cache and memory references, write a program that will provide a count of different cache misses

Hint: It is a subtle question. So do not jump.
Think when you have time to think
Some may take days
Some years
Some natural lifespan
Ping me if you are sure you have done it correctly

Knobs of Interest: Cache

Line size, associativity, cache size

Tradeoff: latency, complexity, energy/power

Tips: Think about the extremes:
Line size = one byte or cache size
Associativity = one or #lines
Cache size = Goal oriented: latency/bandwidth or capacity
Think: How these affect 3C misses ☺

Cache Performance

How good is the cache for a given application?

Hit rate

Miss rate

Misses per kilo instructions (MPKI)

But Why?

Average Access Time

On average, how much time it takes for a LOAD to complete

Average memory access time (AMAT) =

Hit time + Miss rate x Miss penalty

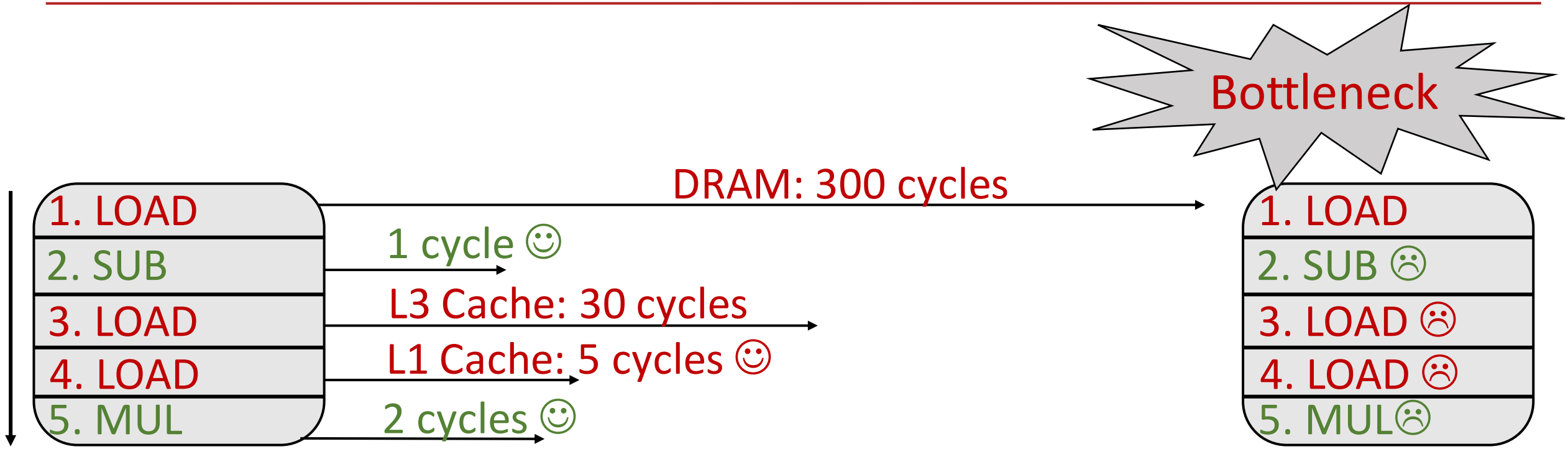
Hit time: Low

Miss rate: Low

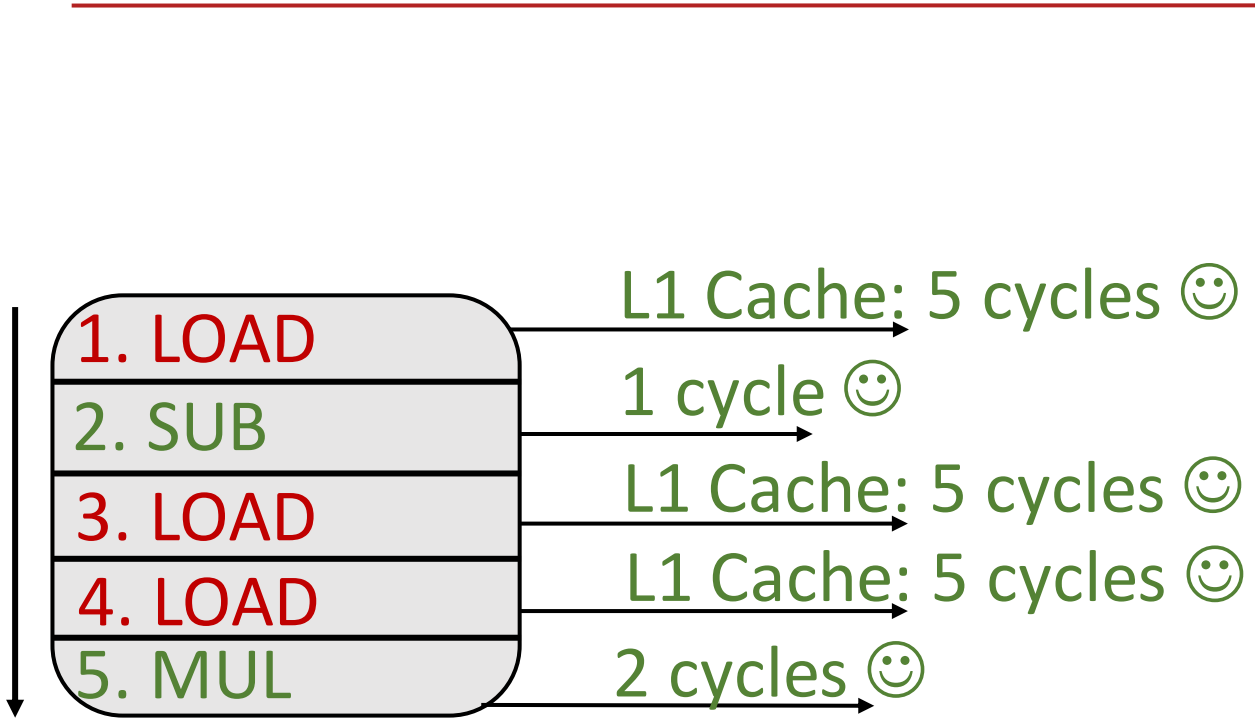
Miss penalty: Low

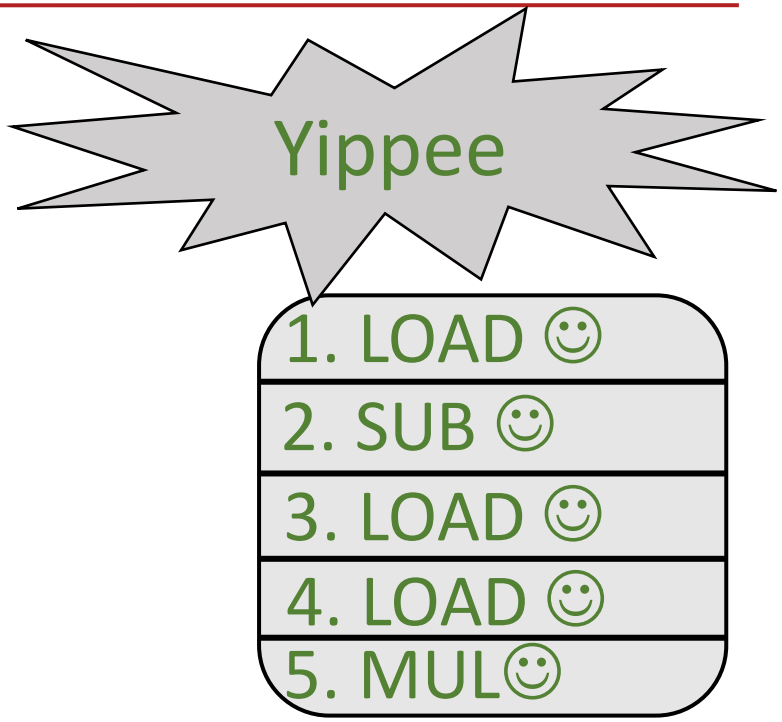
Ideally, miss rate = 0.00% and hit time should be one cycle, so all LOADs will take just one cycle ☺

The Implications of L1-D Hit rate of 100%



A Dreamy World





Refer H&P, or Online Resources

- **Reducing hit time**

1. Small and simple caches
2. Way prediction
3. Trace caches

- **Increasing cache bandwidth**

4. Pipelined caches
5. Multibanked caches
6. Nonblocking caches (MSHRs)

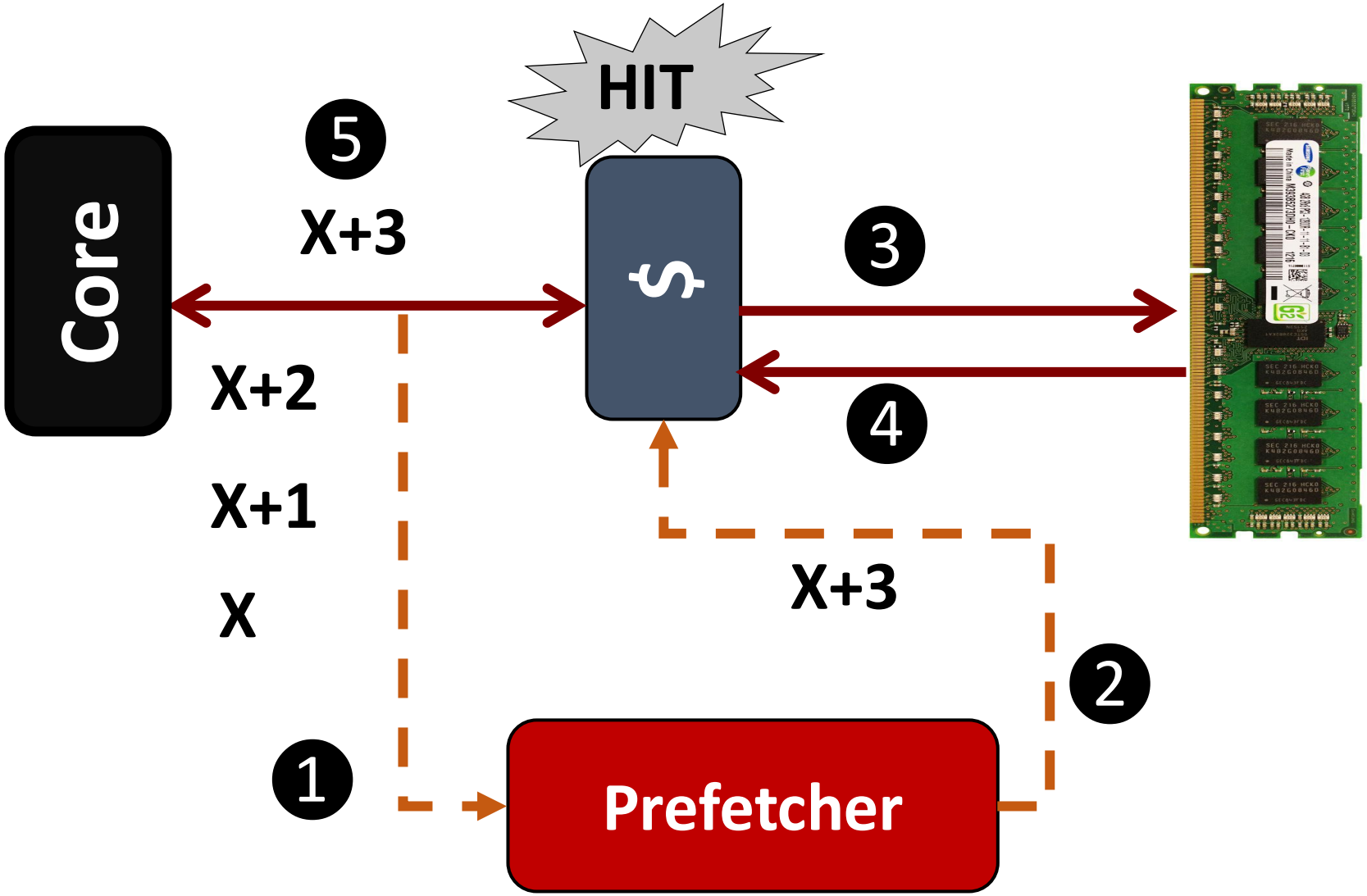
- **Reducing Miss Penalty**

7. Critical word first
8. Early Restart

- **Reducing Miss Rate**

9. Victim Cache
10. Hardware prefetching
11. Compiler prefetching
12. Compiler Optimizations

Hardware Prefetching



If Interested, Read

Bouquet of Instruction Pointers: Instruction Pointer Classifier-based Hardware Prefetching

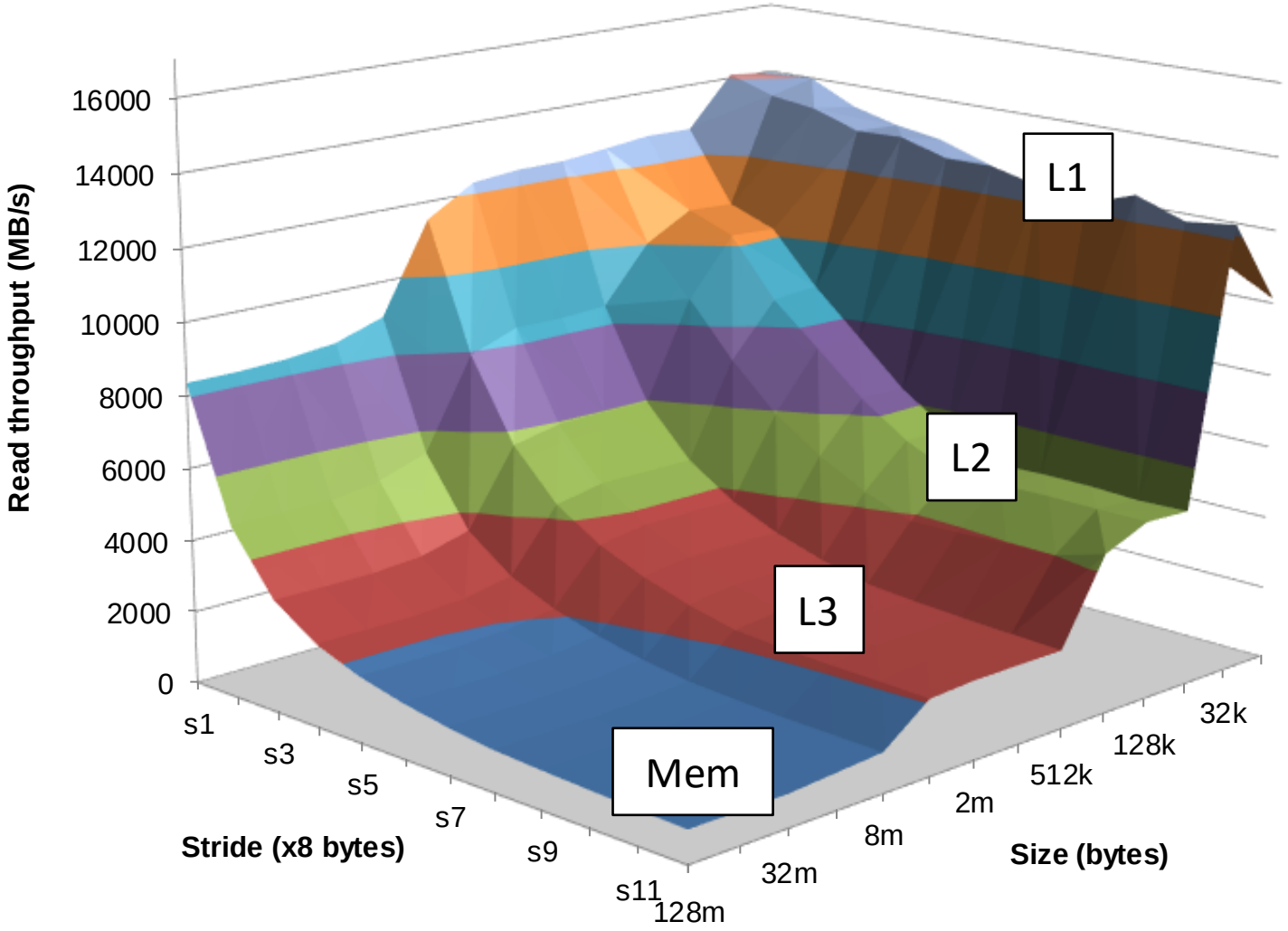
DPC3@ISCA '19

ISCA '20

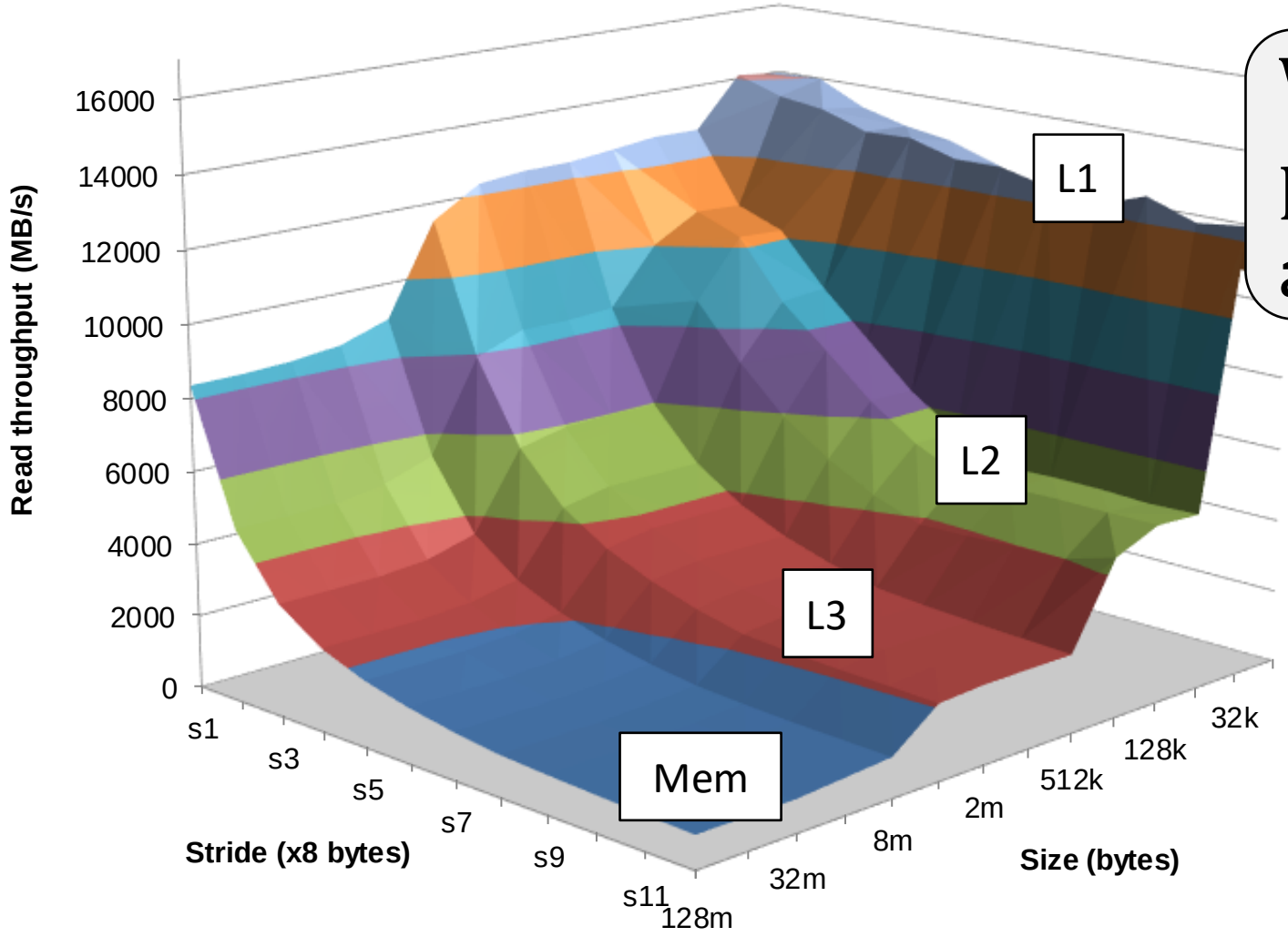
https://www.cse.iitk.ac.in/users/biswap/IPCP_ISCA20.pdf

As a Programmer: What Can you Do?

Memory Mountain

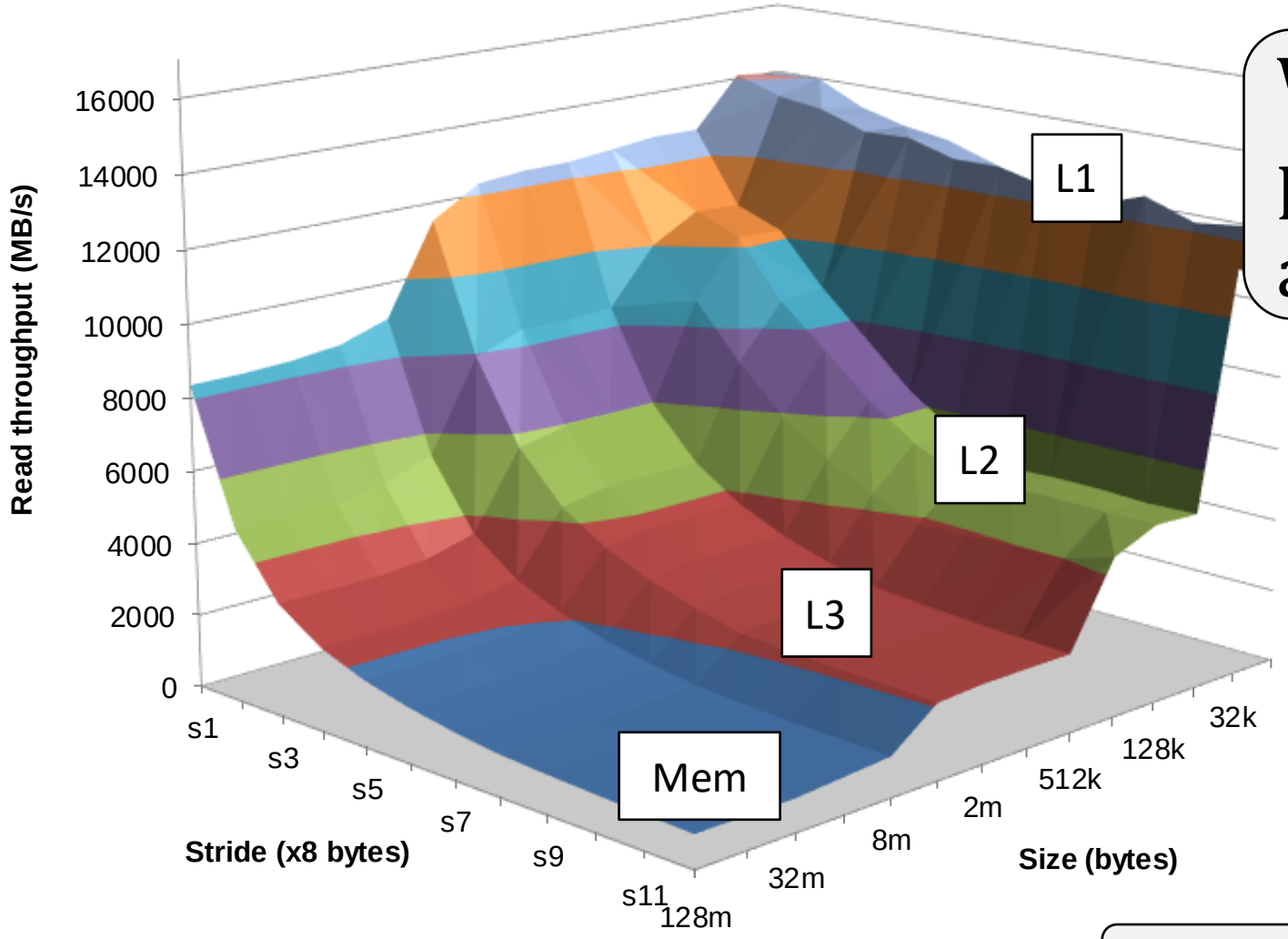


Memory Mountain



When you write your program next time, think about this mountain ☺

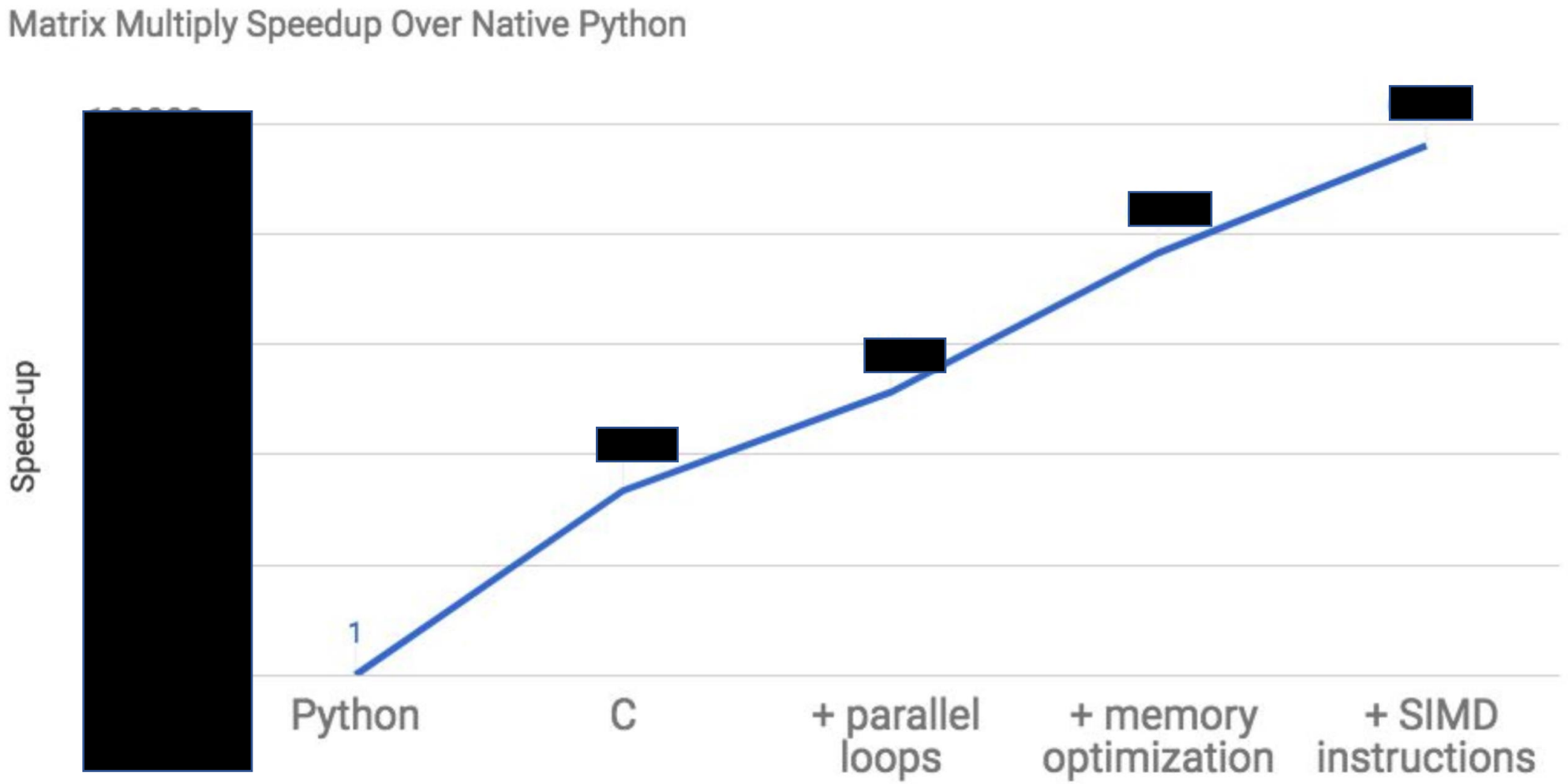
Memory Mountain



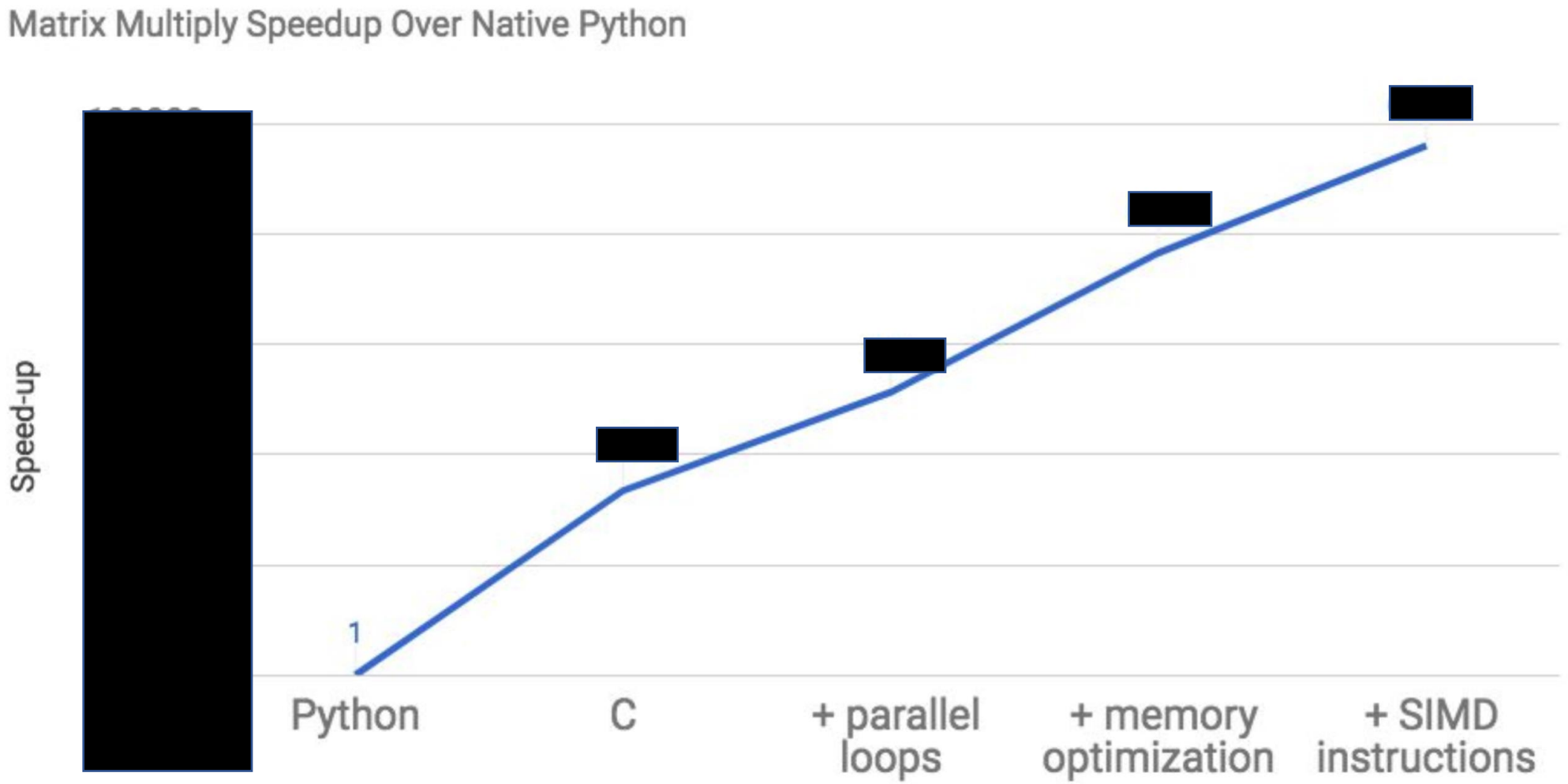
When you write your program next time, think about this mountain ☺

Later in the afternoon ☺

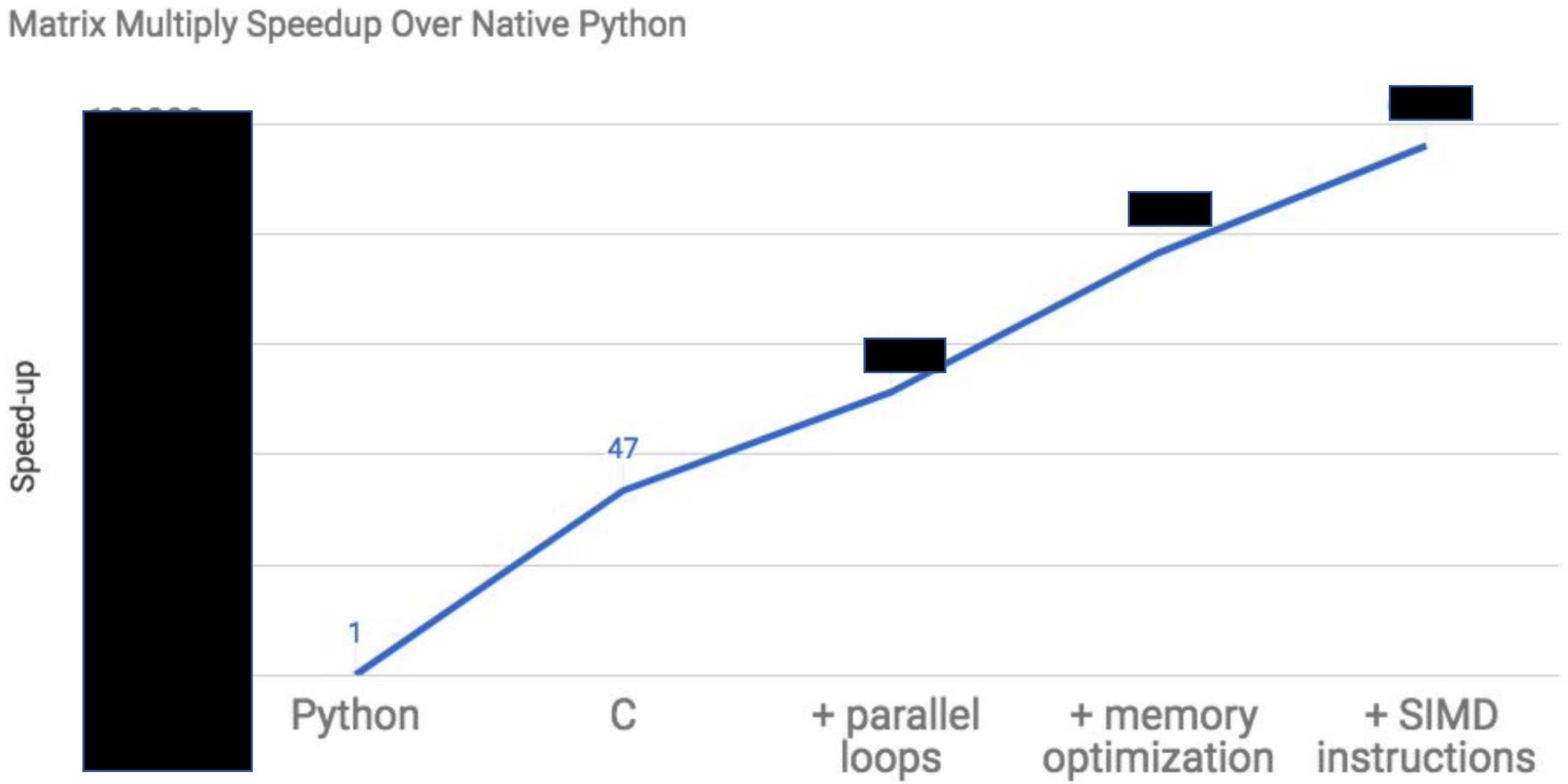
Does Programming Language Matter?



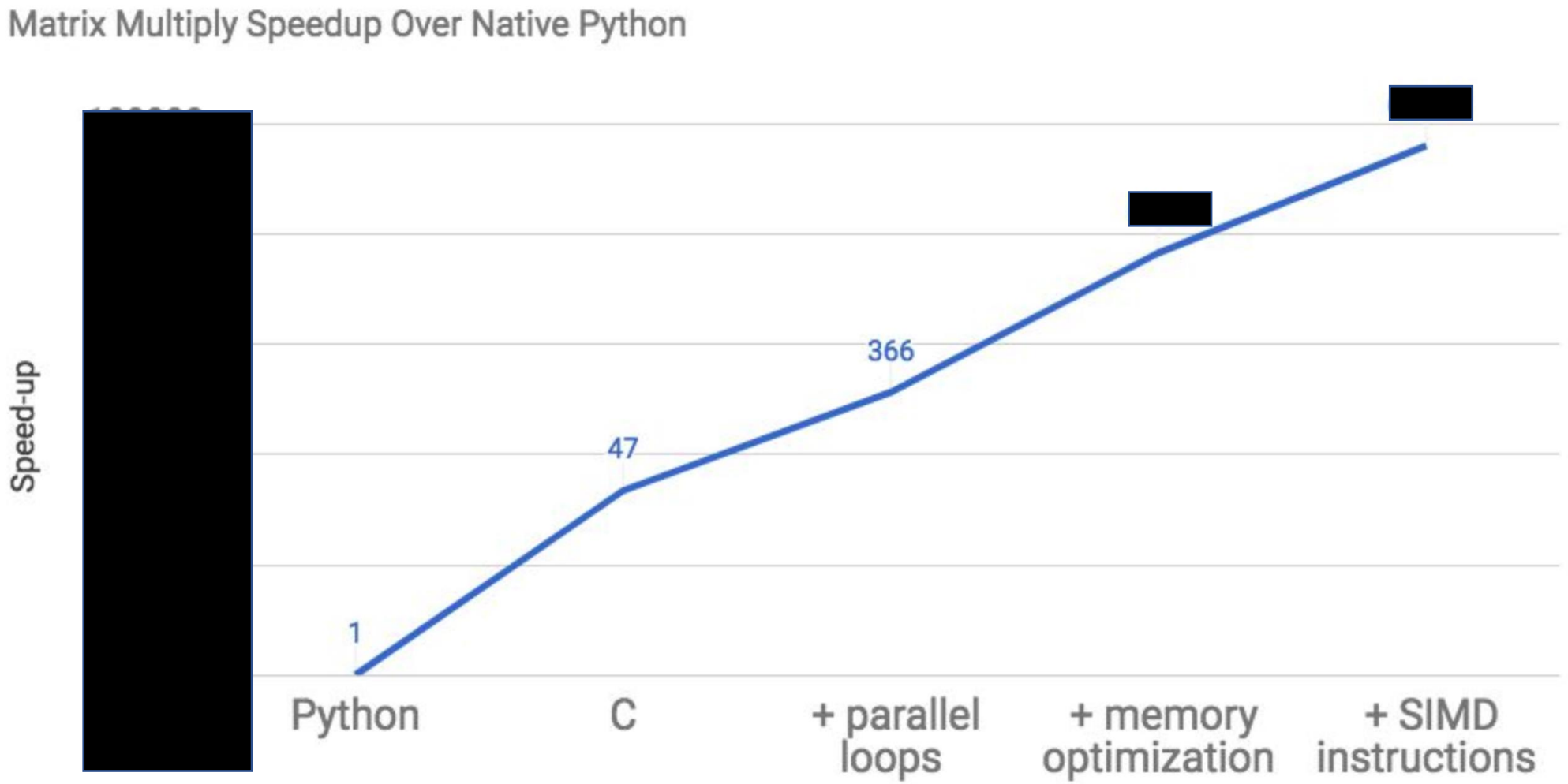
OK. I am ready


Matrix Multiply Speedup Over Native Python
Speed-up
1
Python
C
+ parallel loops
+ memory optimization
+ SIMD instructions

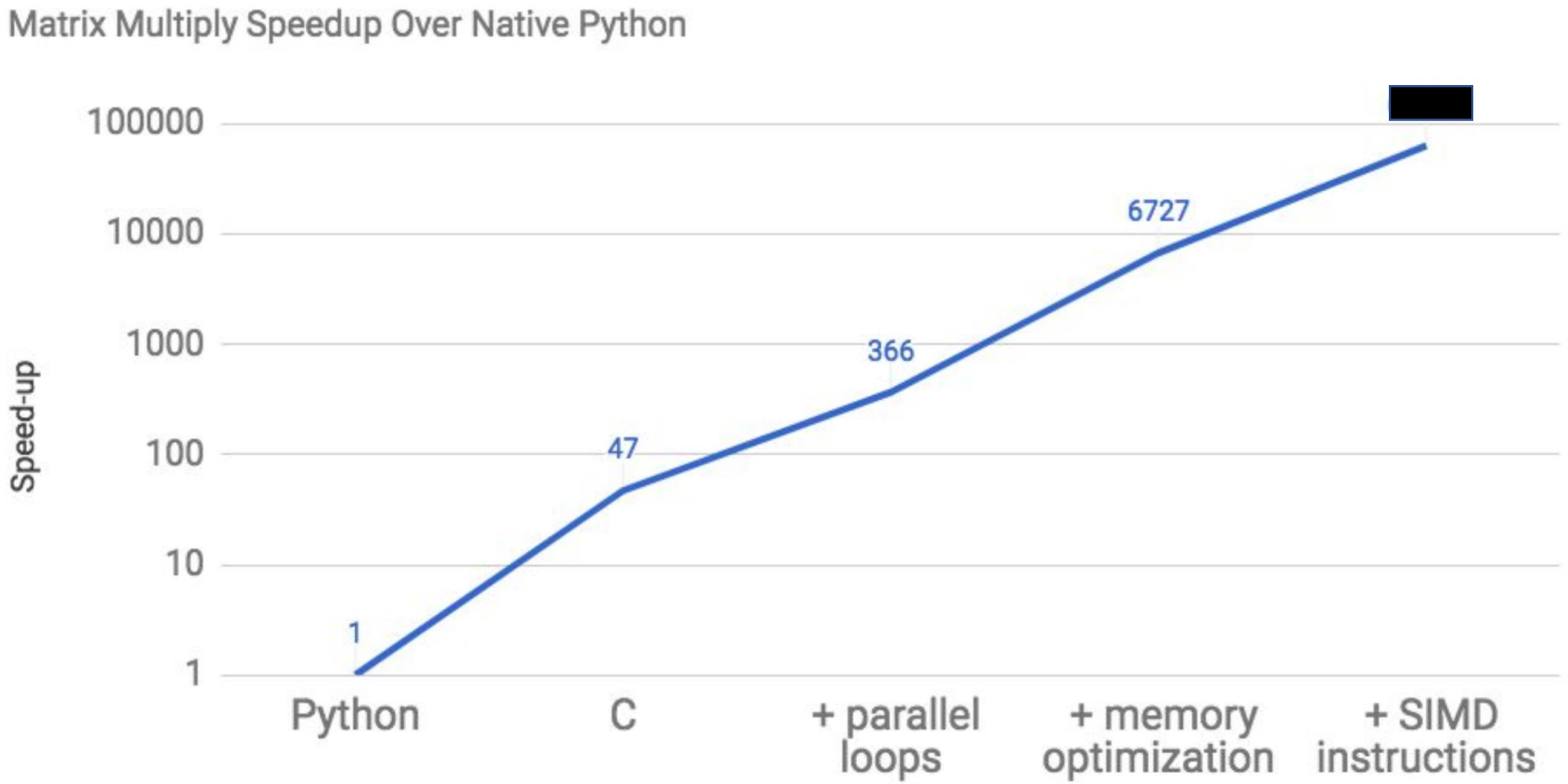
Seriously?


Matrix Multiply Speedup Over Native Python
Speed-up
47
1
Python
C
+ parallel loops
+ memory optimization
+ SIMD instructions

That’s it. You can’t do better. Period !!

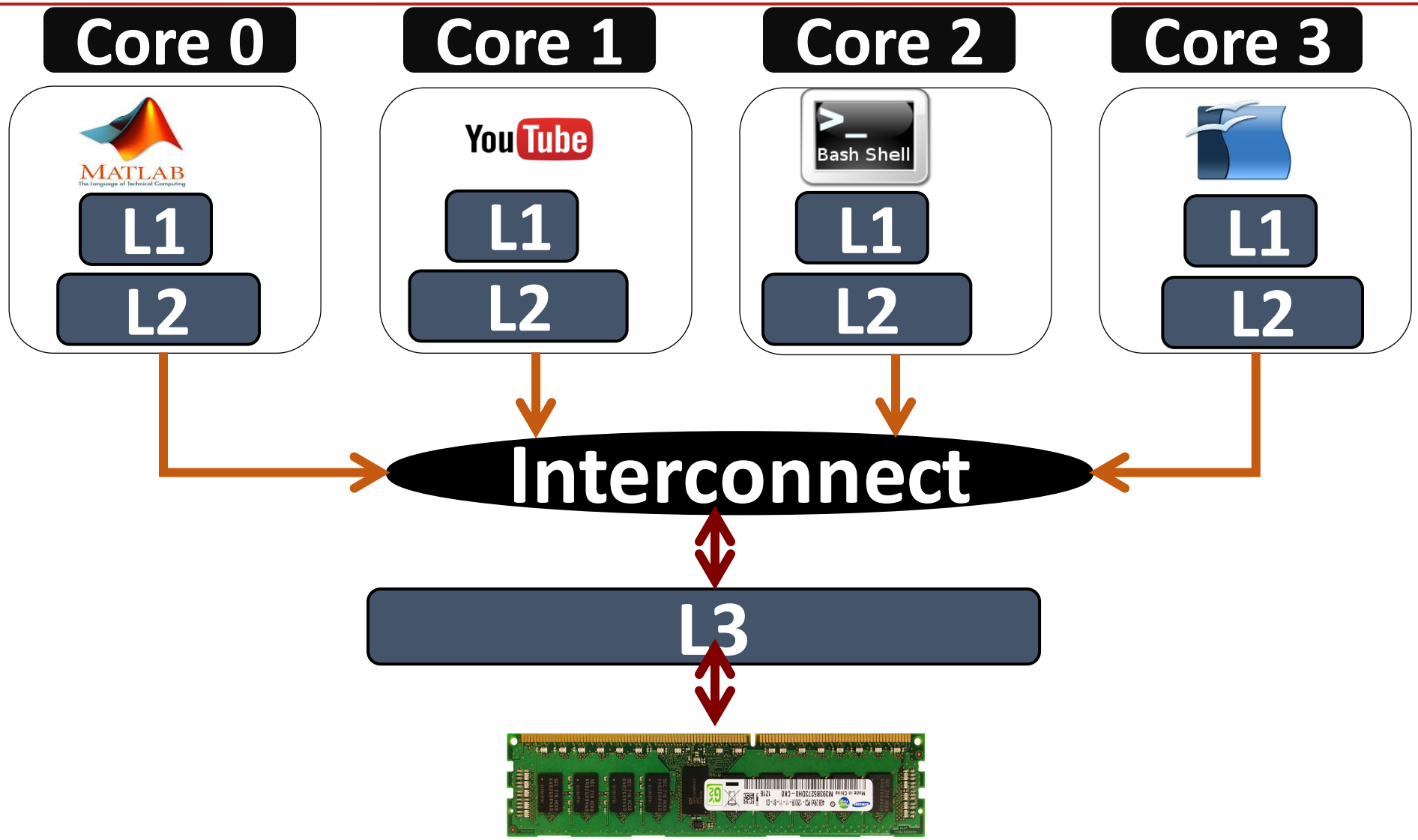


This is Insane

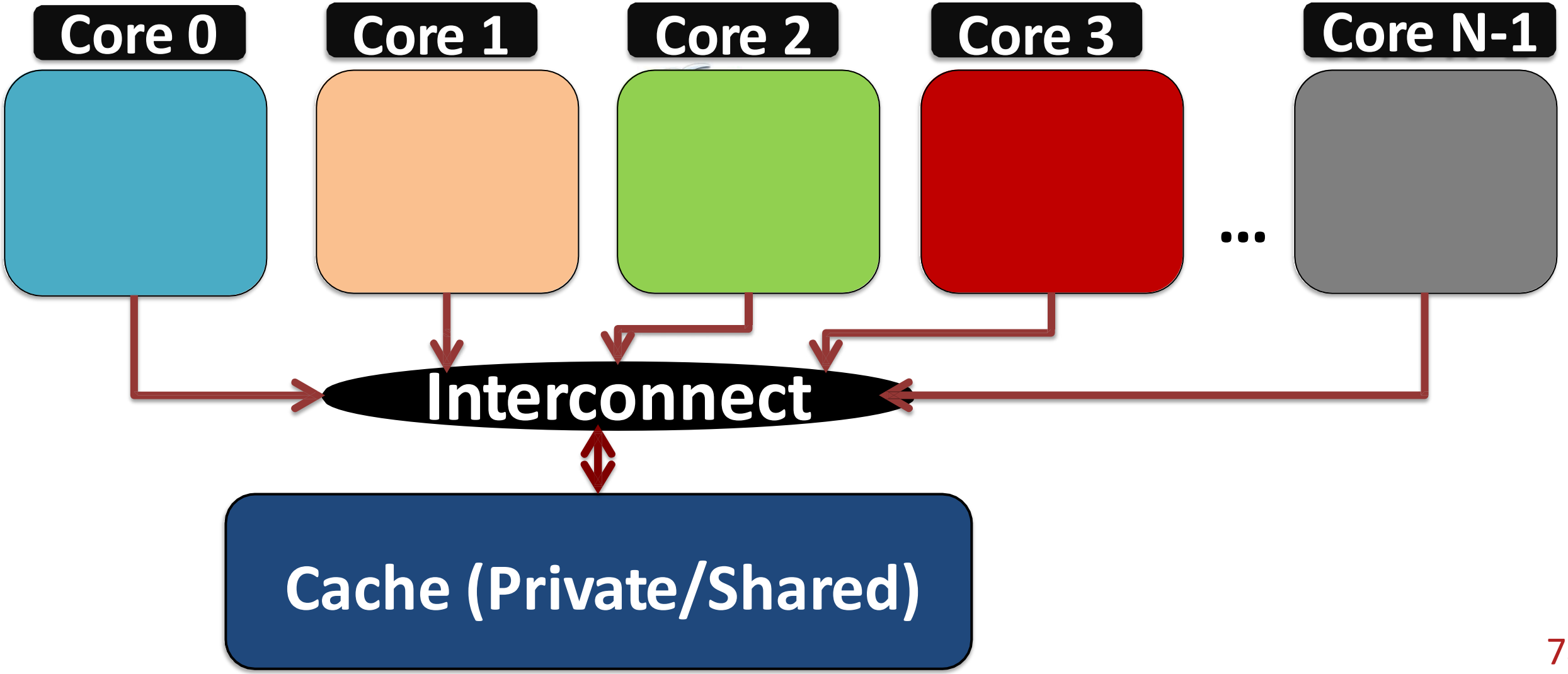



62,806X

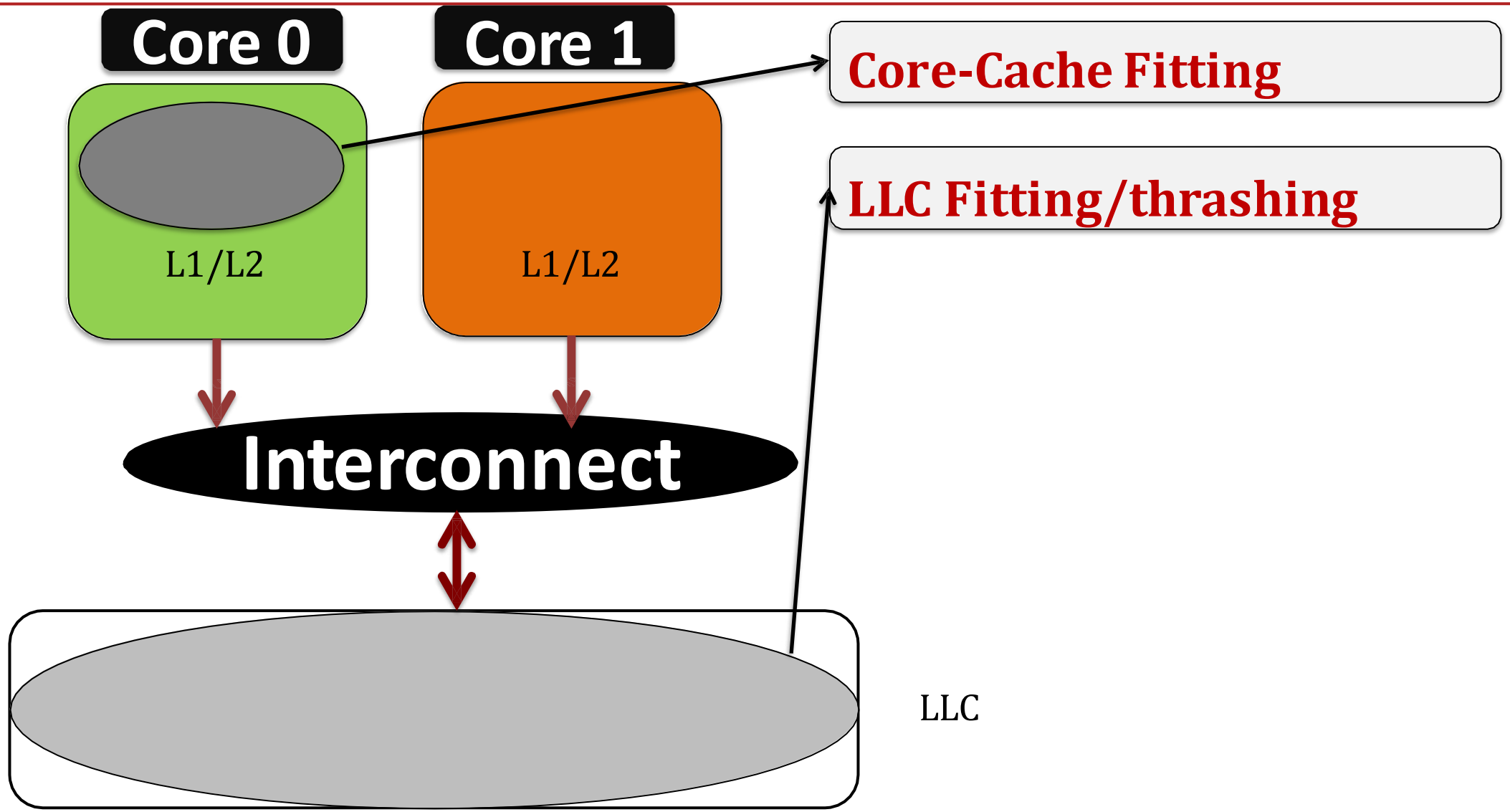
Multi-core


Core 0
Core 1
Core 2
Core 3
MATLAB
YouTube
Bash Shell
L1
L2
L1
L2
L1
L2
L1
L2
Interconnect
L3

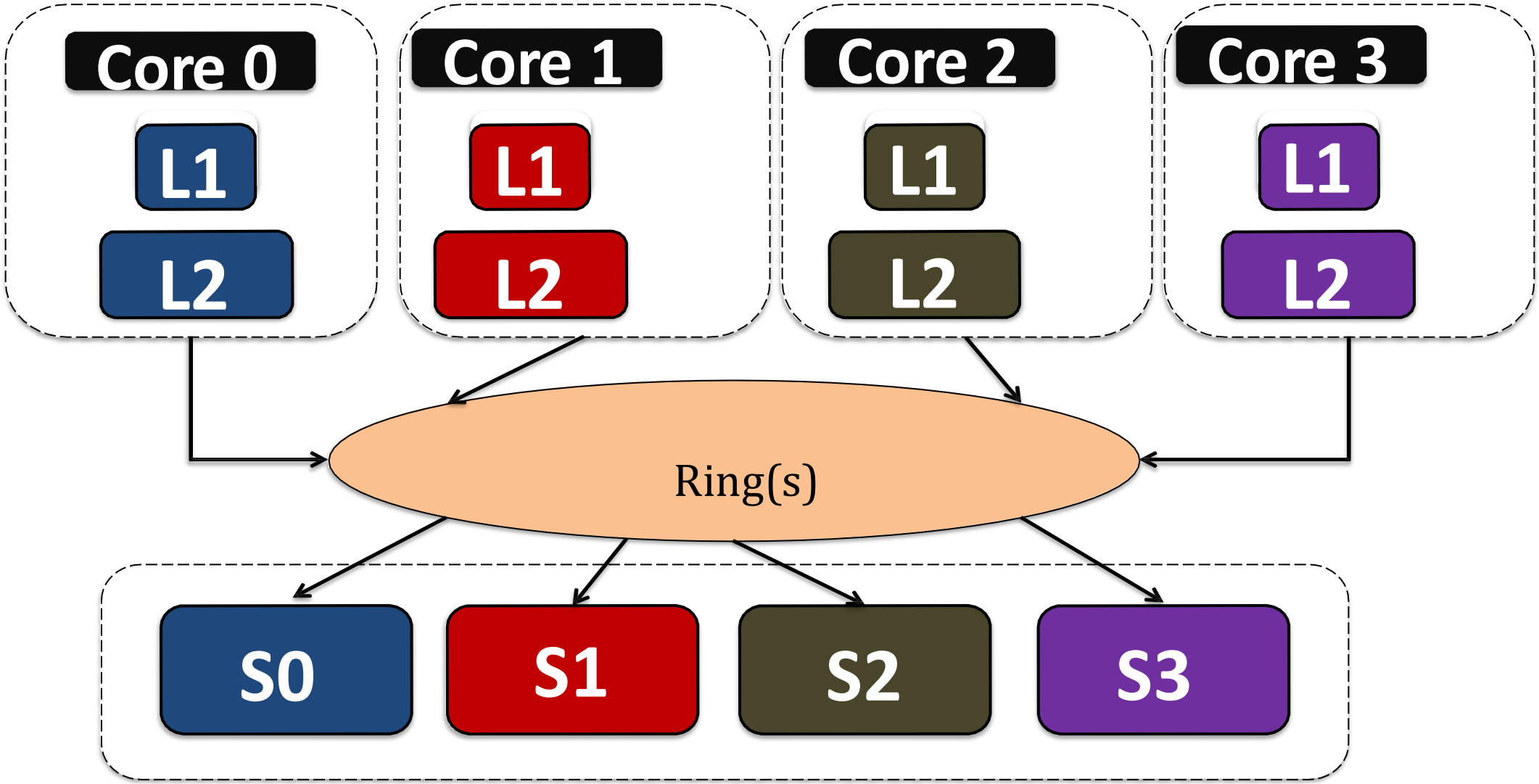
Private vs Shared?



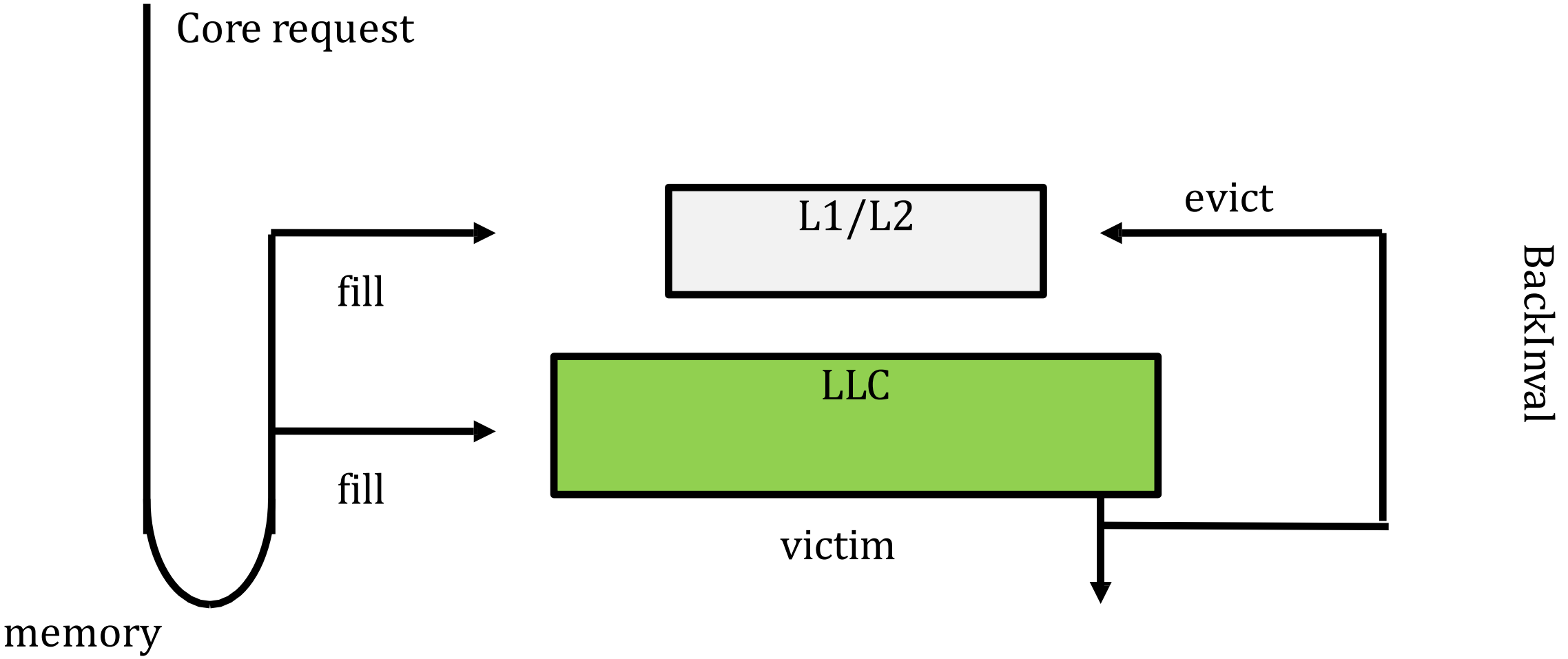
Application Behavior



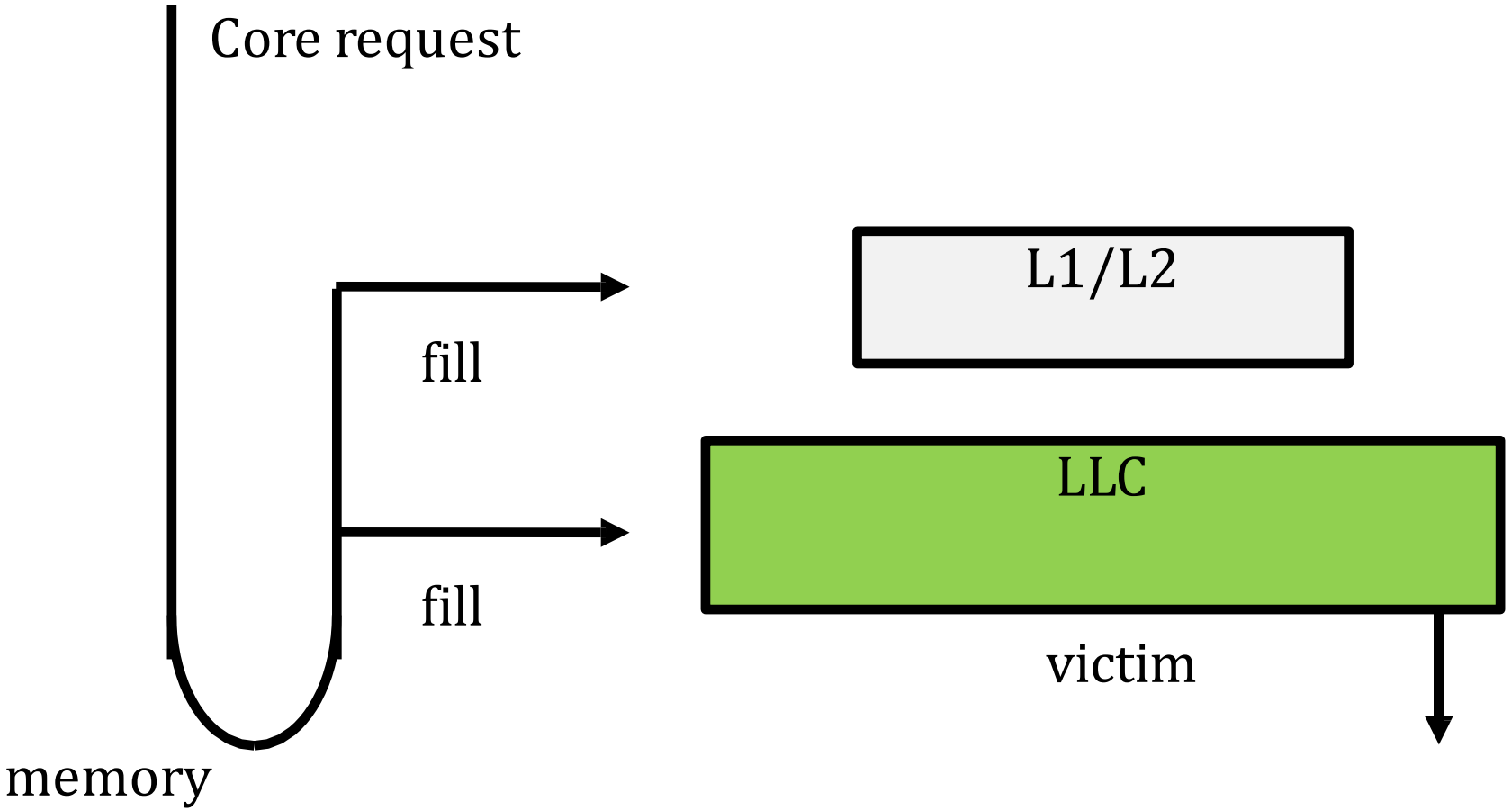
Shared Last-level Cache (LLC): Banked or Sliced



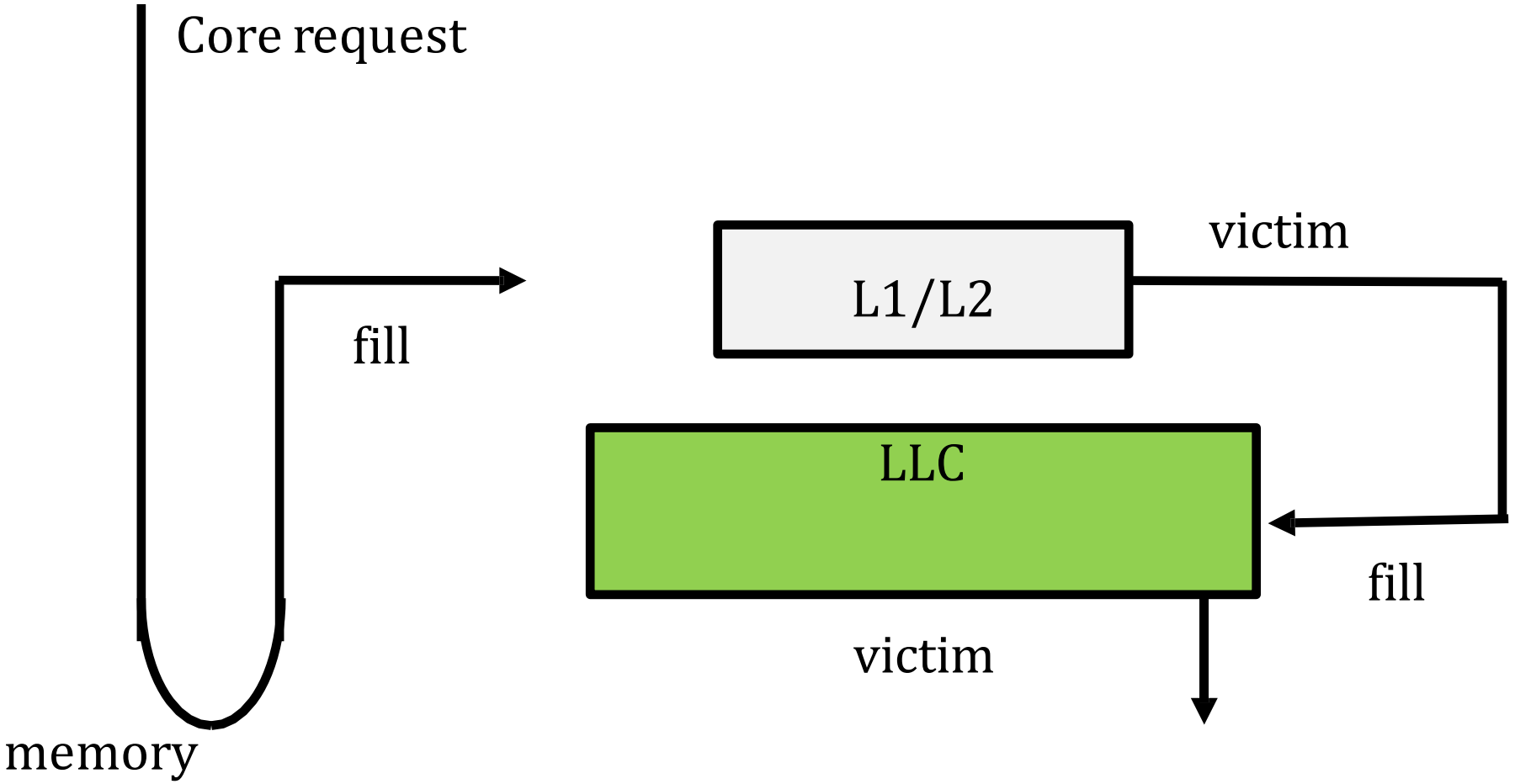
Inclusiveness



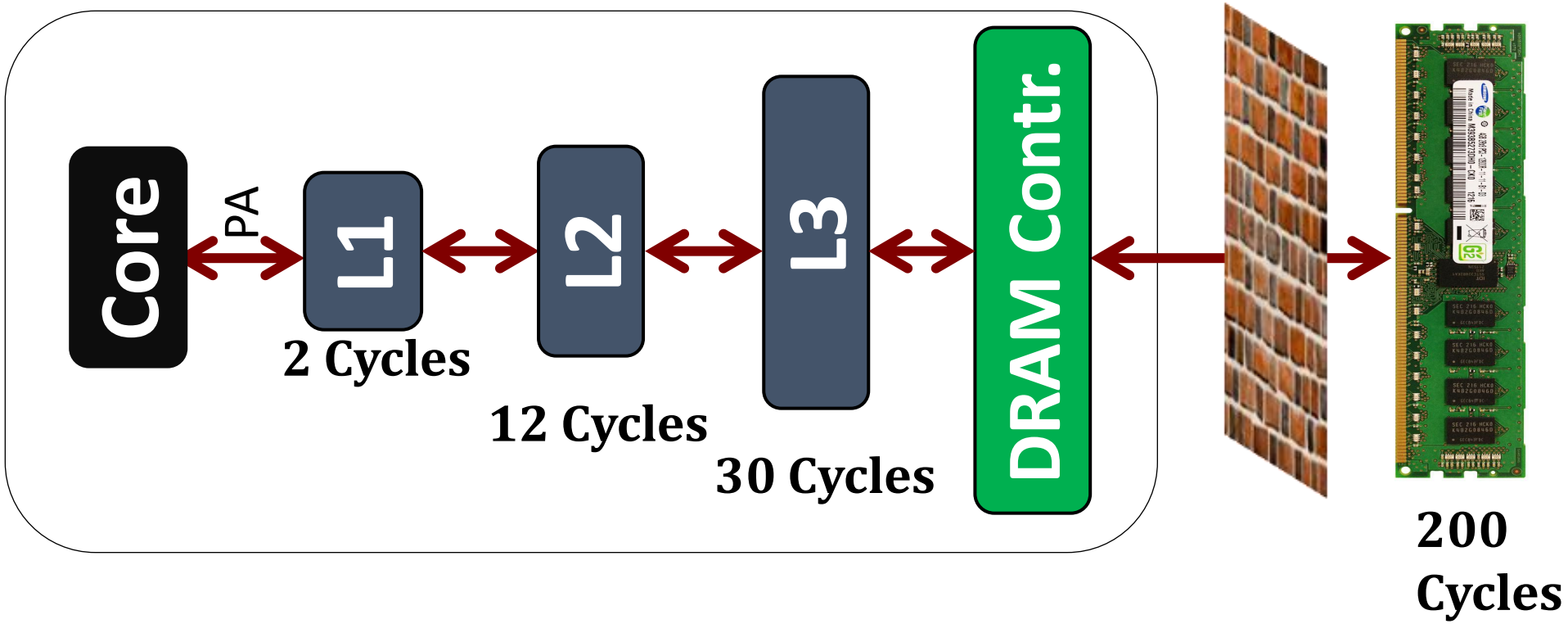
Non-inclusive (Commercial machines)



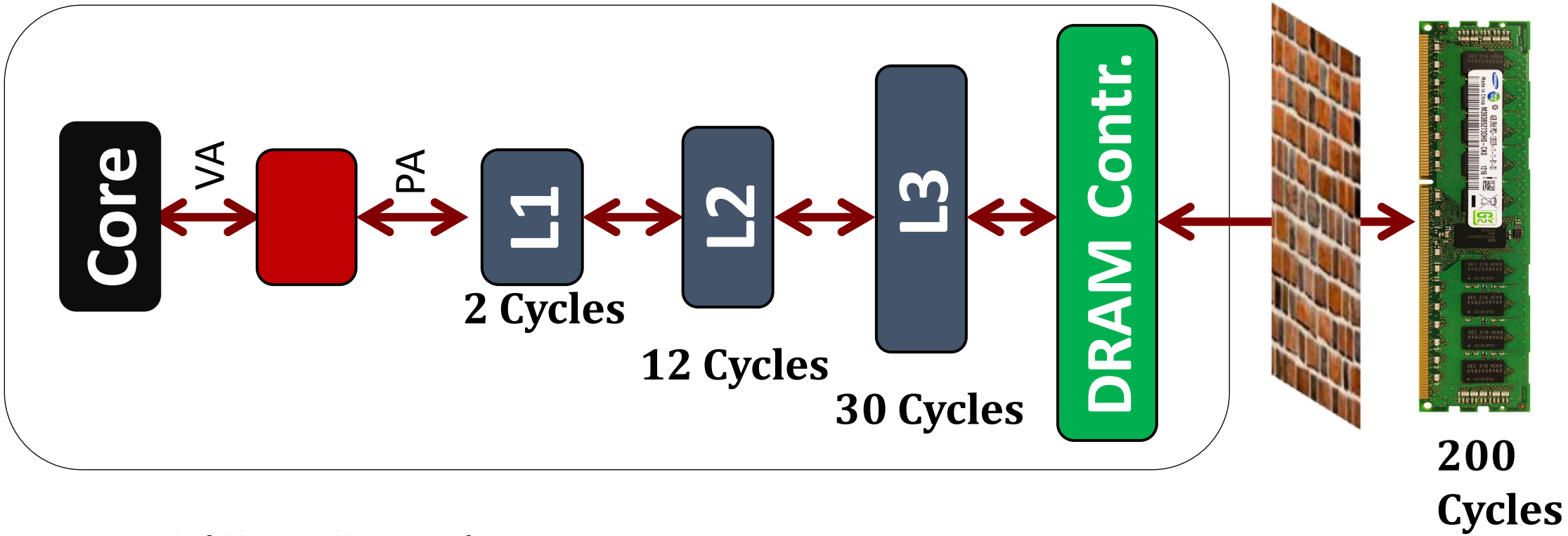
Exclusive



So far view of Memory Hierarchy



Welcome to the Virtual World



Printf (“%d”, &a);

Virtual address

Virtual Memory

App. 1

Virtual address space

Printf (“%d”, &a);

App. 2

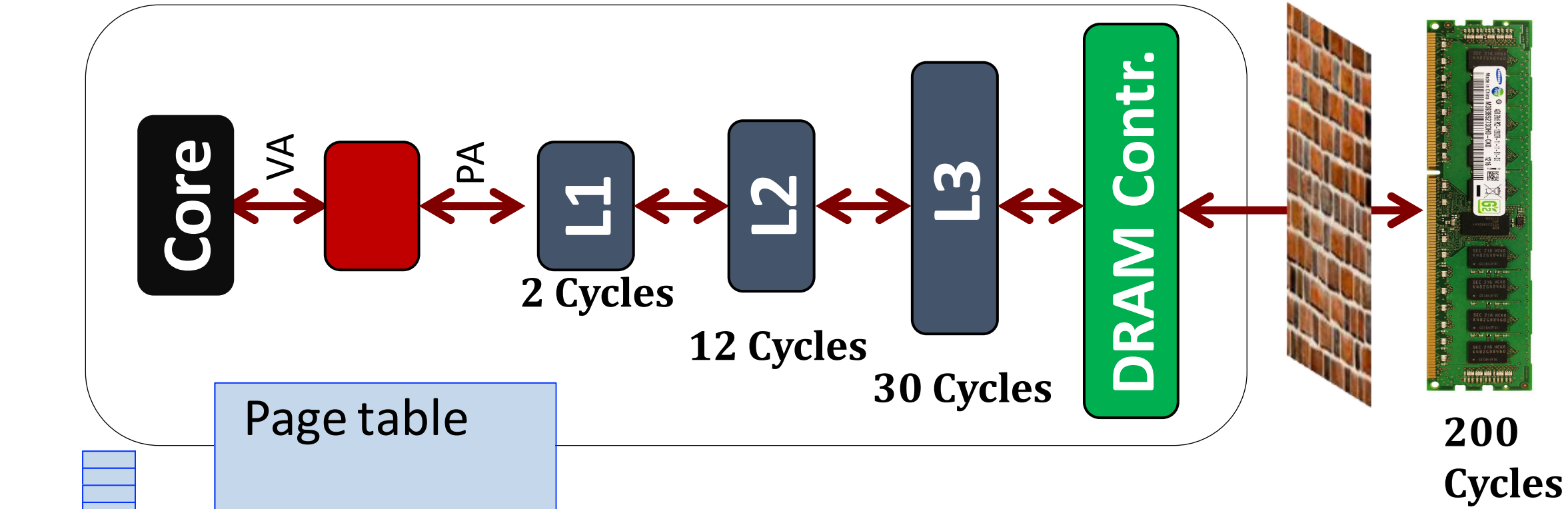
Virtual address space

Printf (“%d”, &a);

100

100

Page Table



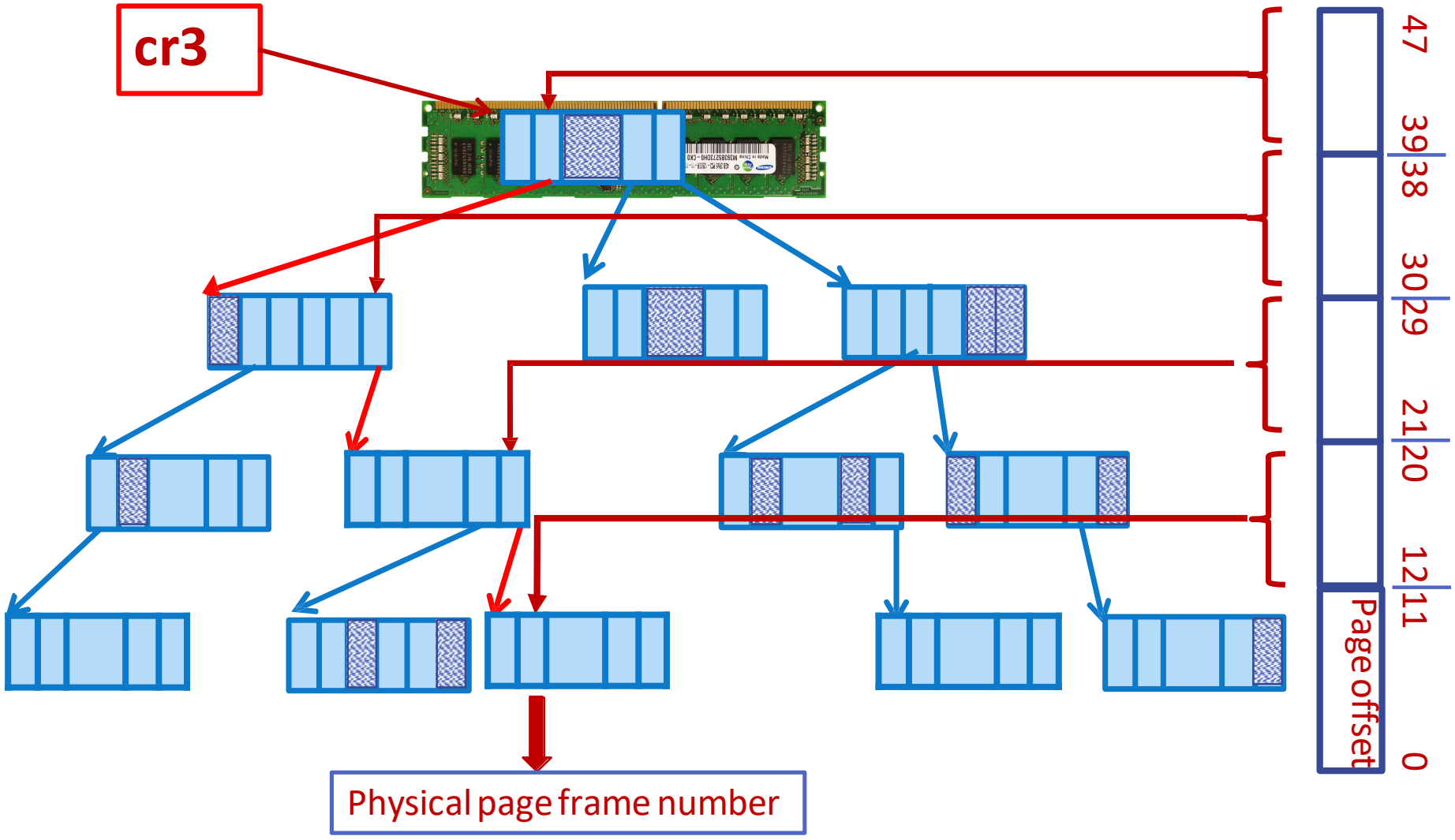
Page table

VP0 → PP8

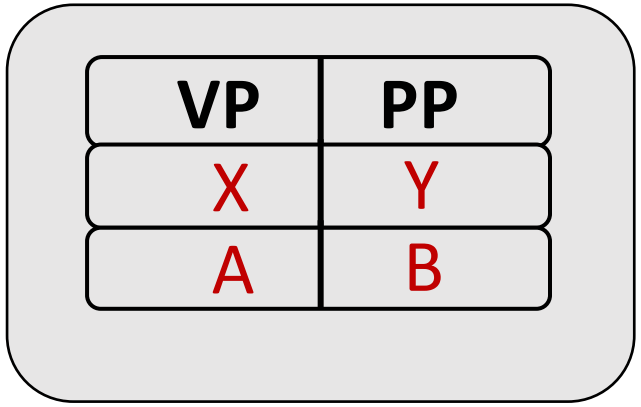
VP1 → PP3

VP2 → PP2

Multi-level One (four to five-level Walk)

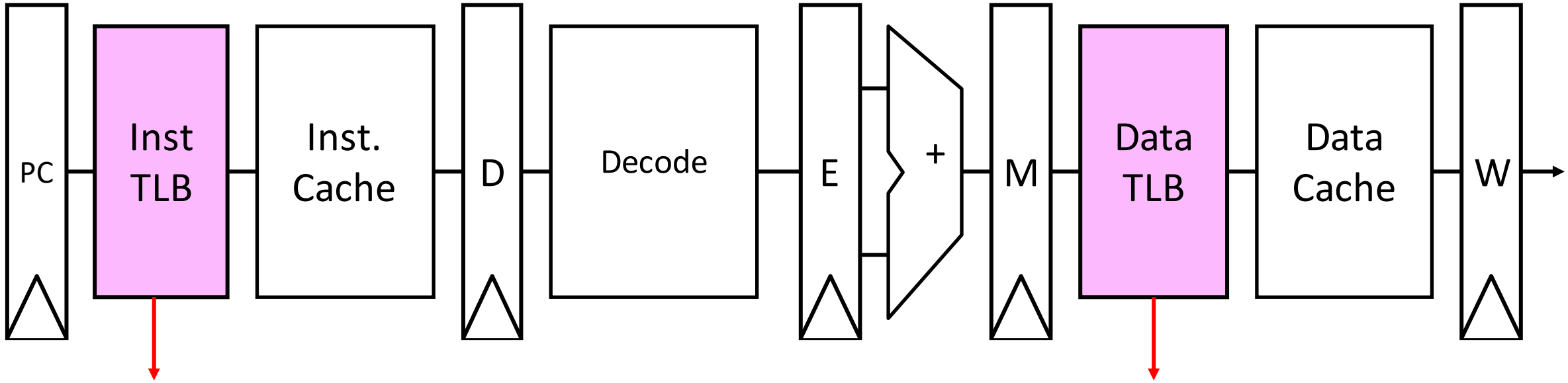


Can We Cache Translations too? Why Not



Translation Look-aside Buffers (TLBs)

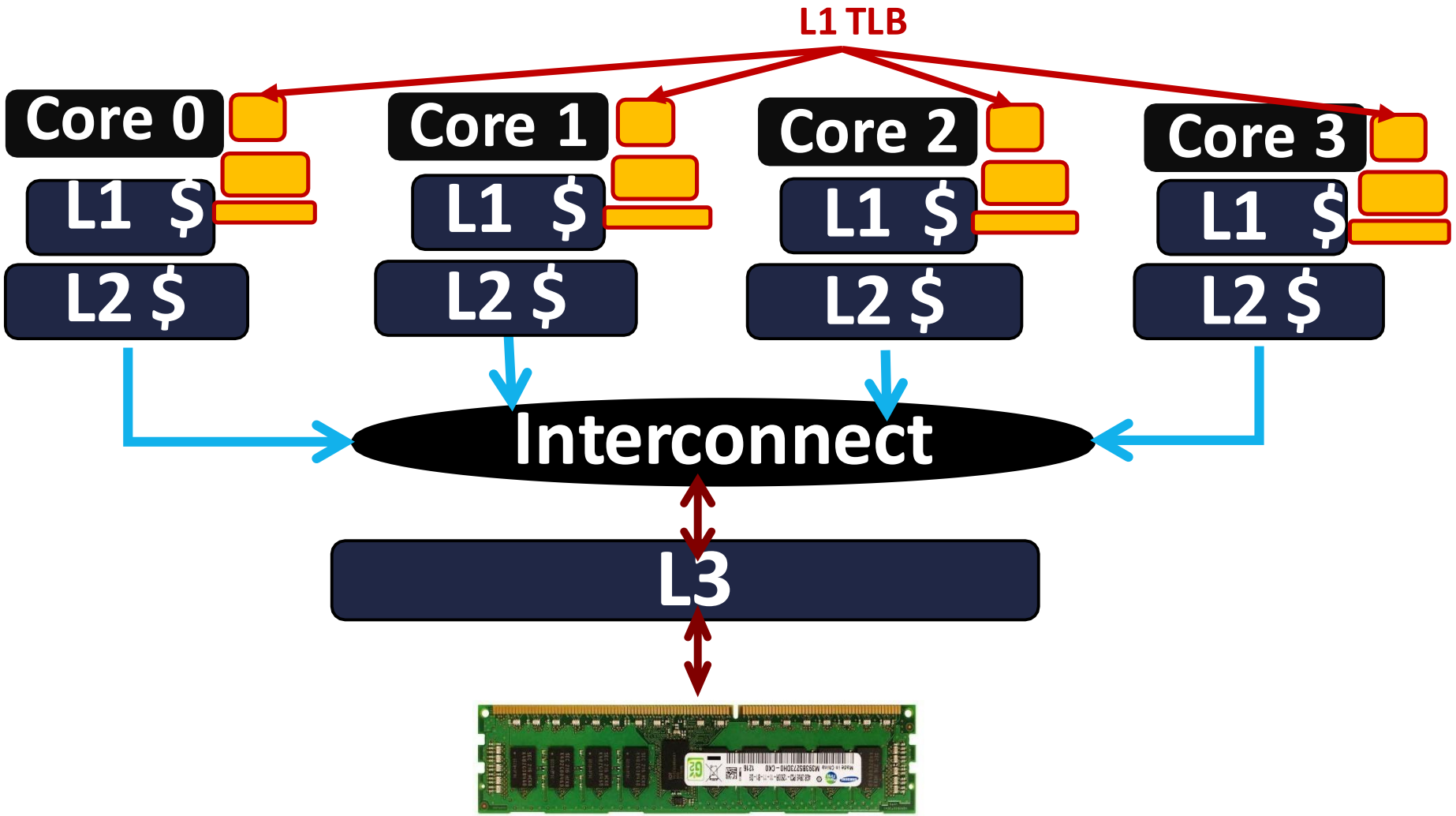
TLB in your text-book Processor Pipeline



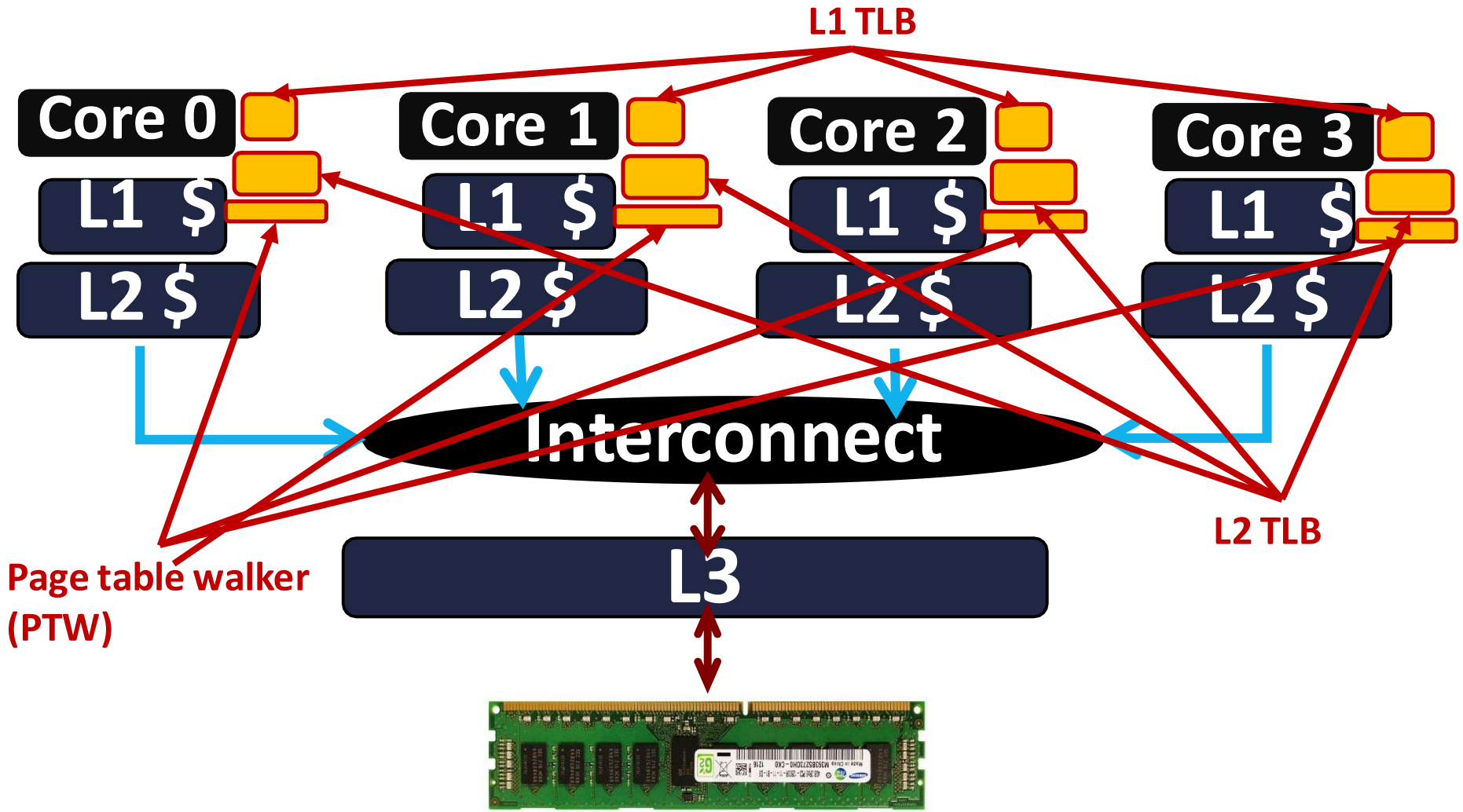
TLB miss? Page Fault?
Protection violation?

TLB miss? Page Fault?
Protection violation?

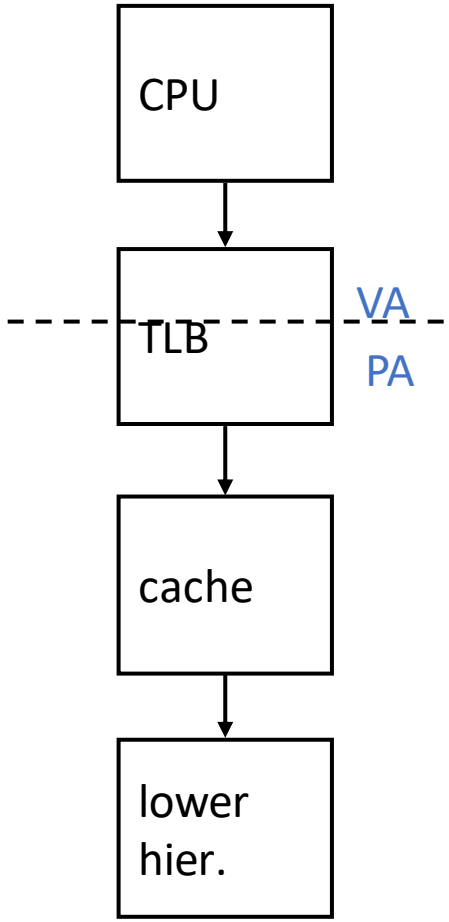
TLBs and Page-table Walkers (PTWs)



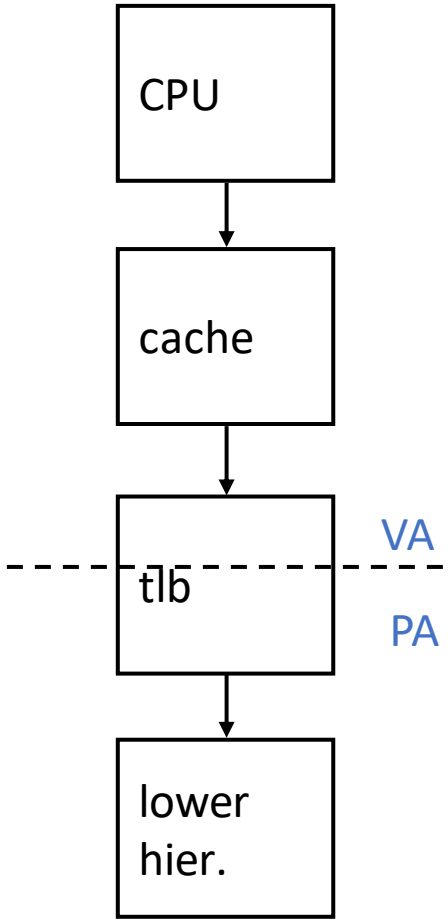
TLBs and Page-table Walkers (PTWs)



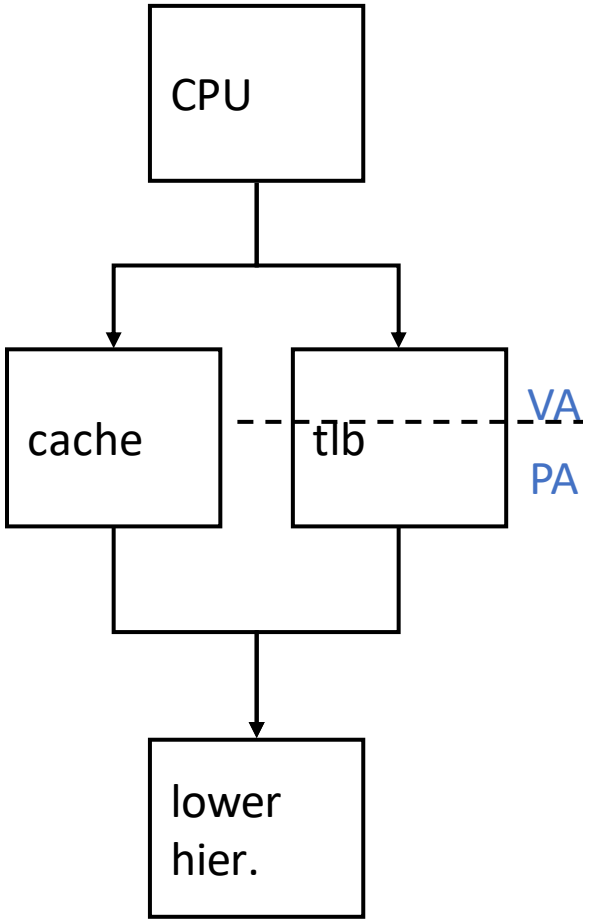
Caches: Virtual/Physical, Possibilities?



physical cache

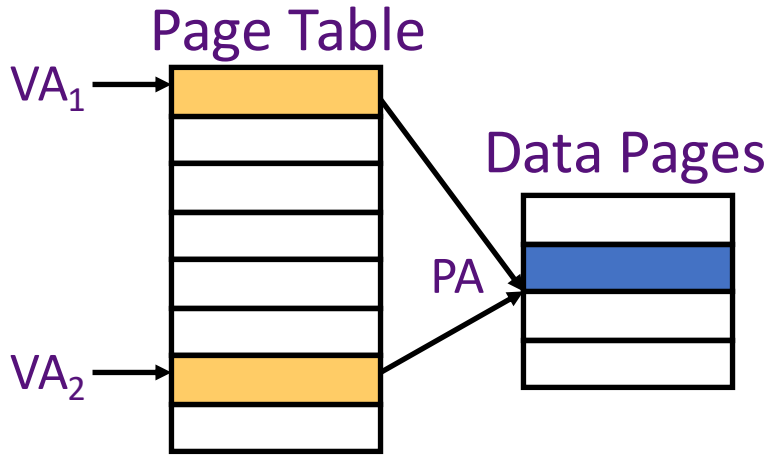


virtual (L1) cache



virtual-physical cache

Synonym Problem with Virtual Cache



Two virtual pages share one physical page

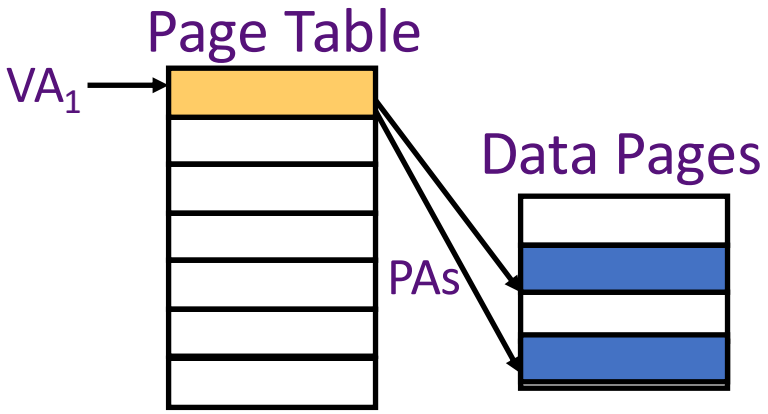
Tag	Data
VA_1	1st Copy of Data at PA
VA_2	2nd Copy of Data at PA

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!

General Solution: *Prevent aliases coexisting in cache*

Software (i.e., OS) solution for direct-mapped cache

Homonym Problem

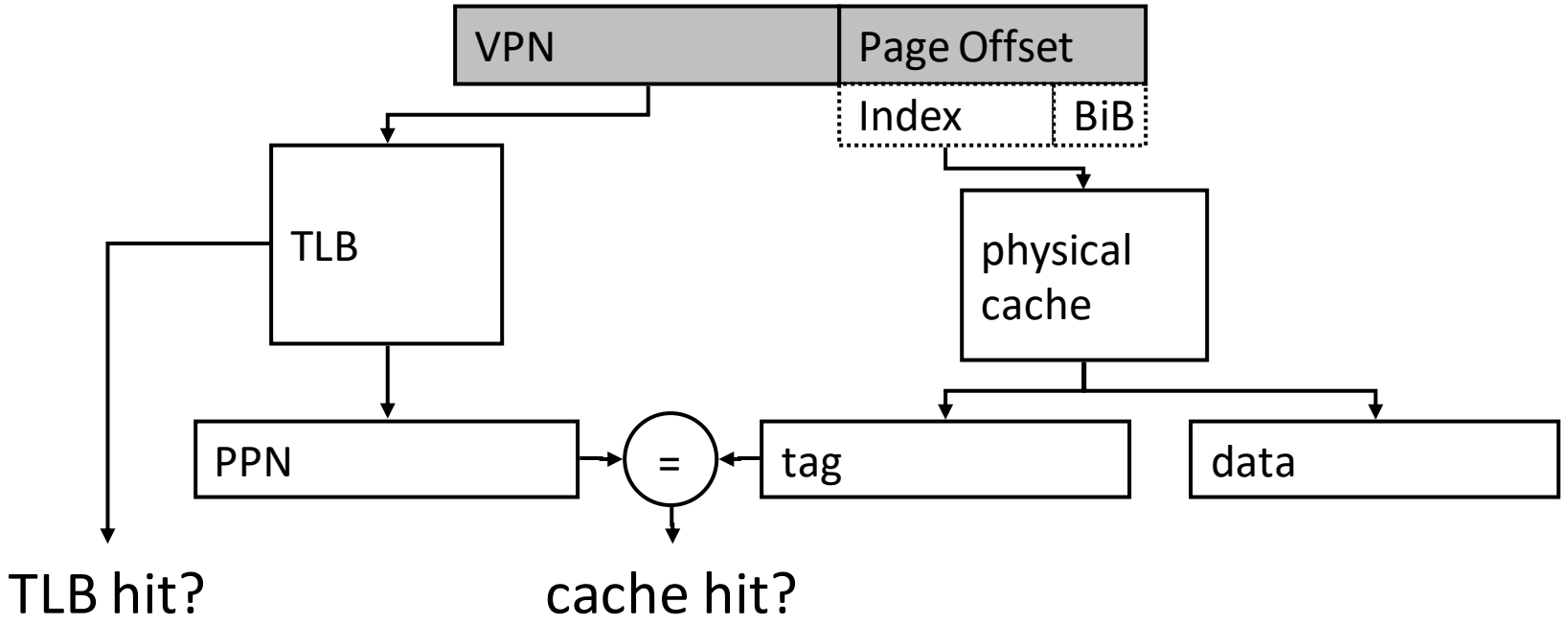


One virtual page maps to two physical pages

Tag may not uniquely identify cache data
Solution: Add ASID with tag
Or
Physical tags
Or
Flush on context switch

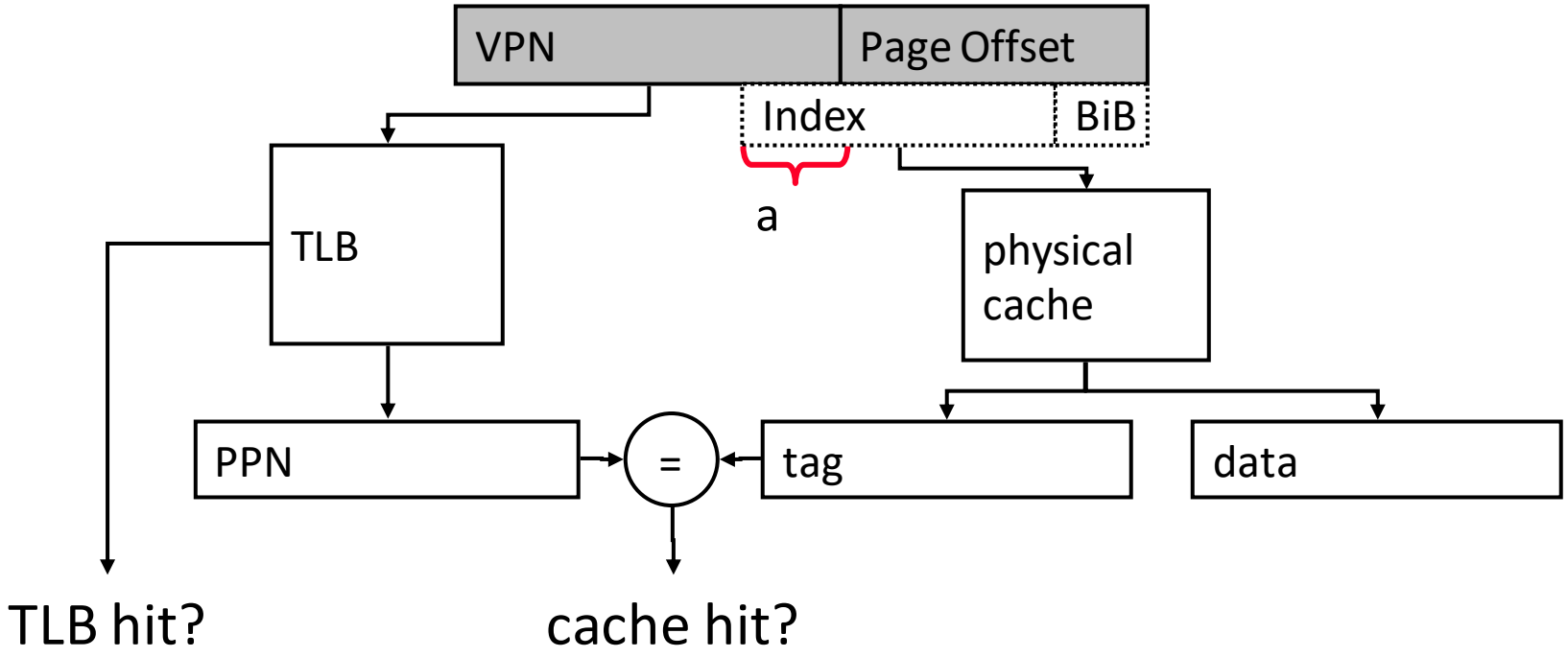
VIPT Cache (Virtually Indexed Physically Tagged)

- If C≤(page_size × associativity), the cache index bits come only from page offset (same in VA and PA)
- If both cache and TLB are on chip: index both arrays concurrently using VA bits, check cache tag (physical) against TLB output at the end



What If? Think About PIPT, VIVT, PIVT, VIPT

- If C>(page_size × associativity), the cache index bits include VPN ⇒ Synonyms can cause problems
  - The same physical address can exist in two locations
- Solutions?



Real Caches ☺

https://en.wikichip.org/wiki/intel/microarchitectures/sunny_cove

Research Questions (Latency and bandwidth)

- Achieving performance closer to ideal TLBs, L1D, L1I
- Domain specific memory hierarchy: graph processing, ML, vision
- Non-volatile caches ☺
- OS-memory-hierarchy interactions
- Microarchitecture for datacenters
- Memory hierarchy for Virtualization

- Look at papers in top forums like ISCA, MICRO, HPCA, ASPLOS, PACT, ICS

- *Microarchitects can kill their grandmothers for 0.5% performance improvement – anonymous ☺*

What You Need to have?

- You should be *passionate* about solving the problem.
- Interested in *whys and hows, and not only whats*?
- Feynman on Scientific method
- How to start doing research? Blog link:

https://medium.com/@__biswa/two-cents-on-computer-architecture-research-103-gollus-computer-architecture-exploration-605466a0cd09

- What are the typical warm-ups? Link:

https://www.cse.iitk.ac.in/users/biswap/notes.html

Thanks

Feel free to share/suggest your feedback/pointers

This was a 10K feet view on caches. Dig deep to know more

ACK

- Some of the slides are adapted and modified from Joel Emer’s course on computer architecture.