CASPER

CS773-2022-Autumn: Computer Architecture for Performance and Security

Lecture 3: Catch the Cache

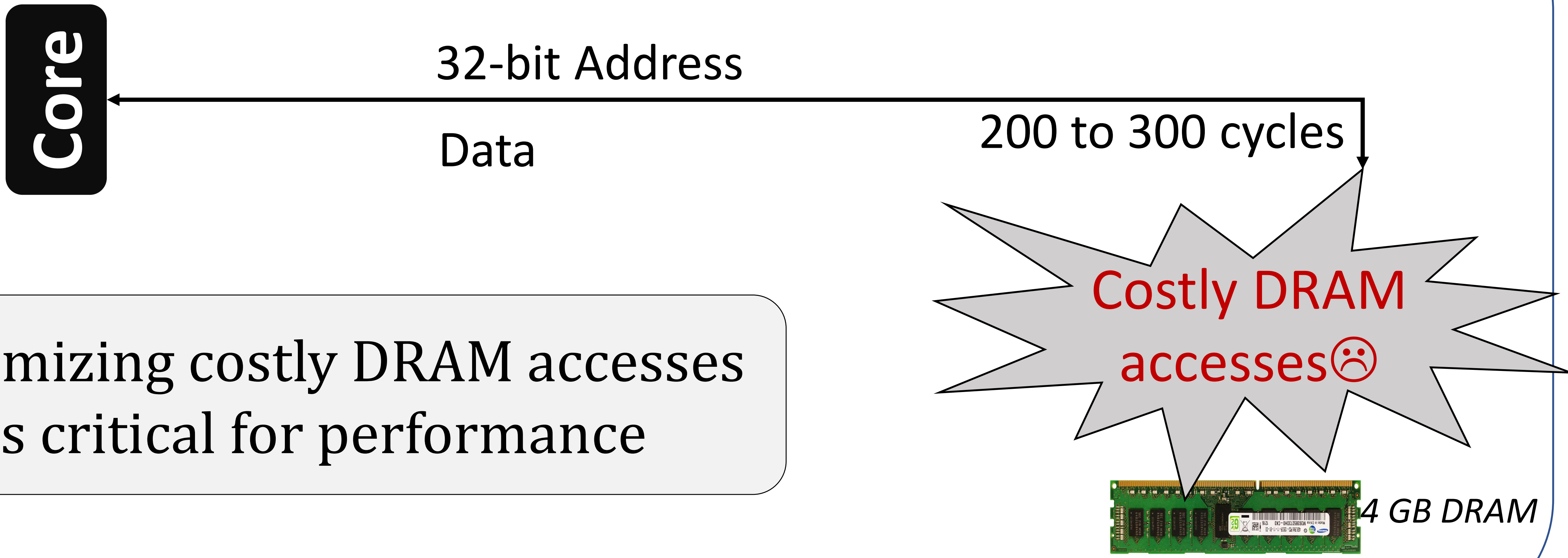
https://www.cse.iitb.ac.in/~biswa/

Phones on silence, please

Thank You

Microarchitecture 101: World with no caches

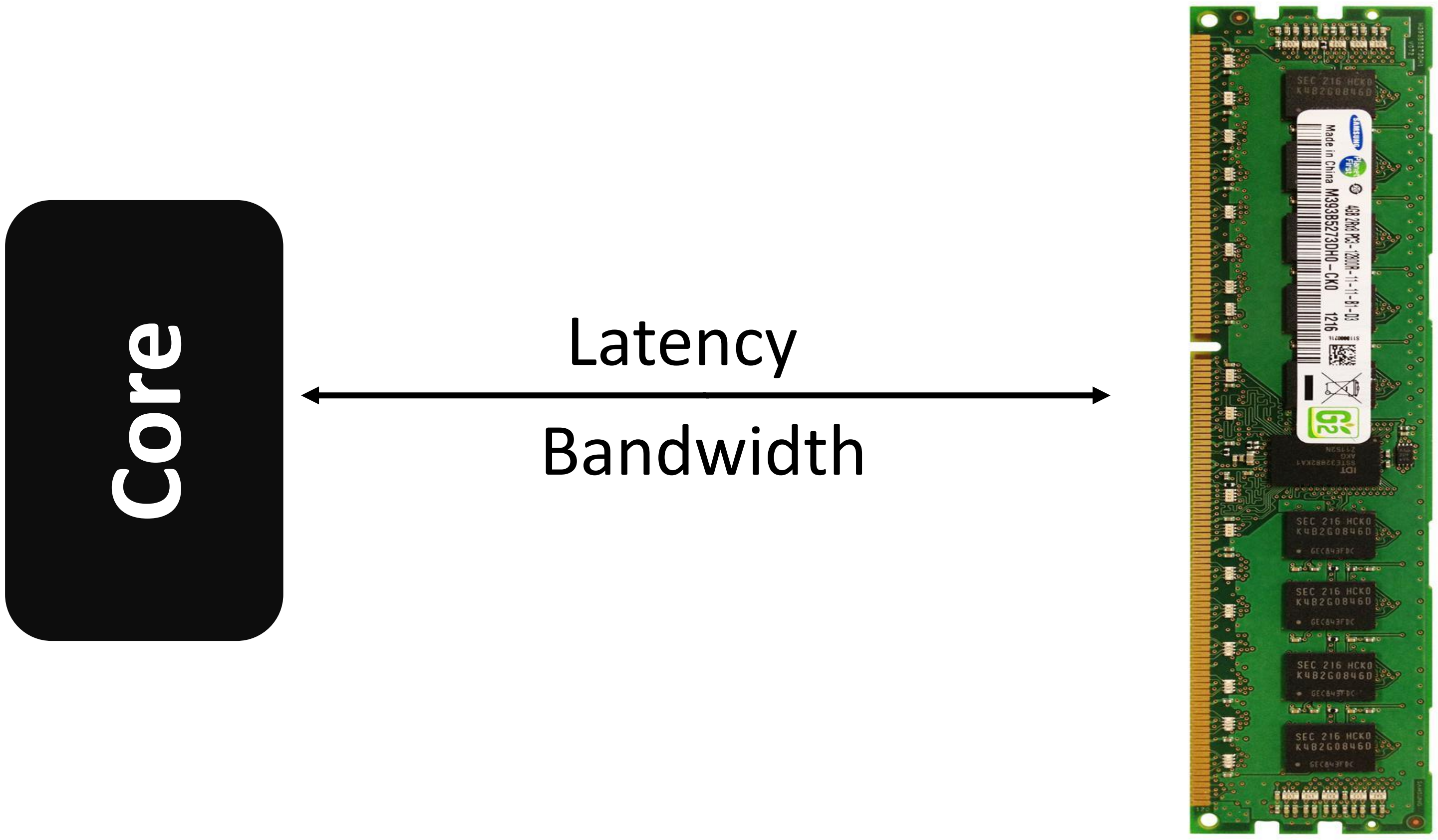
North pole ☹



Minimizing costly DRAM accesses is critical for performance

South pole ☹

Remember Latency and Bandwidth



Core

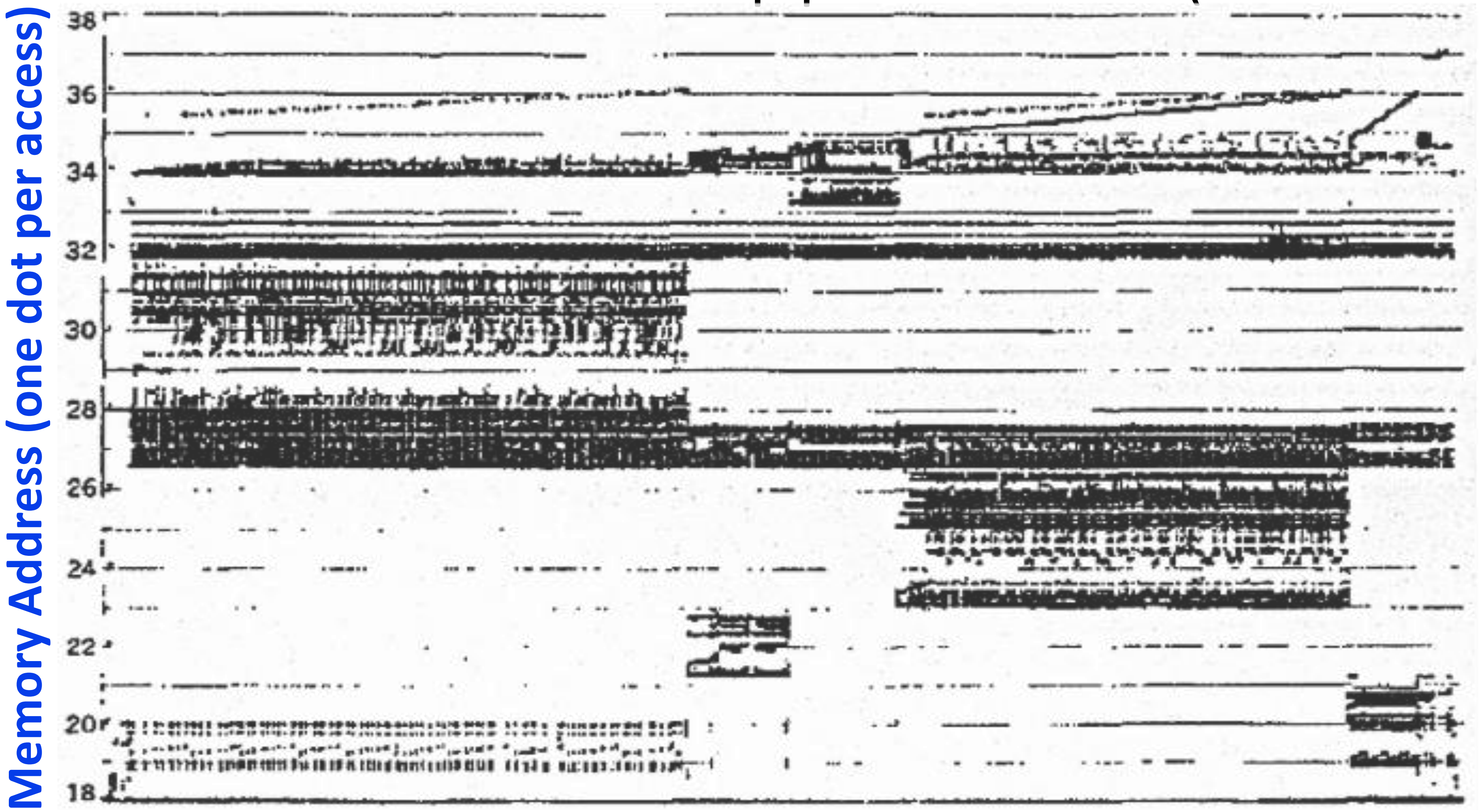
Latency

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

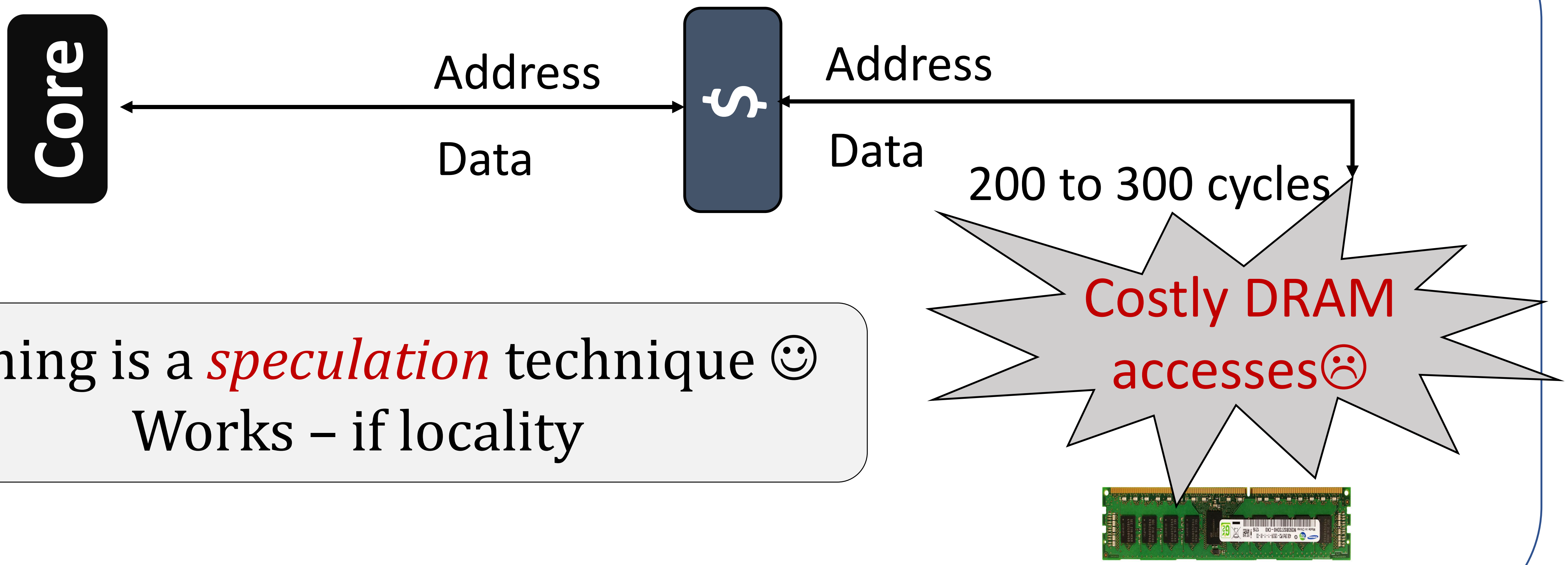
Let’s look at the Applications (benchmarks)



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

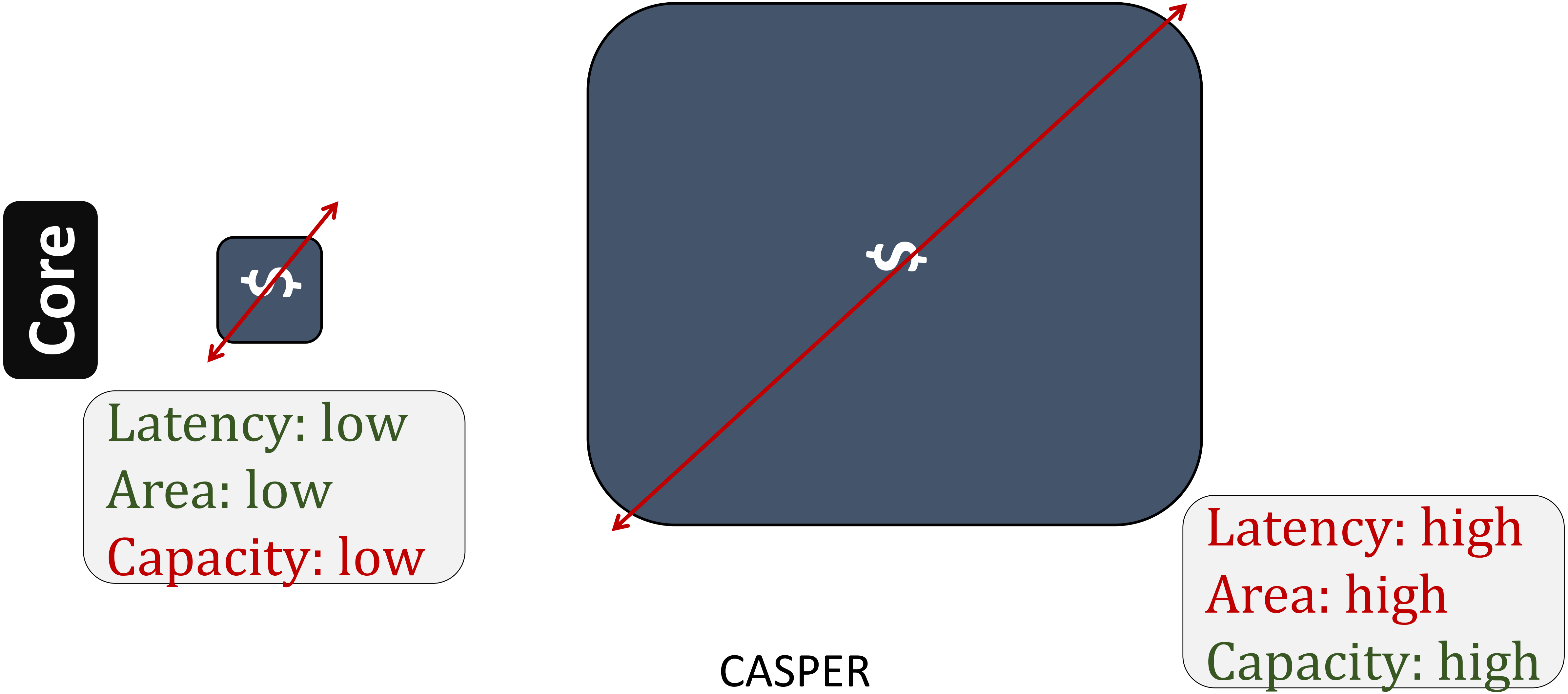
Caching: 10K Feet View

North pole ☺



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

How big/small?



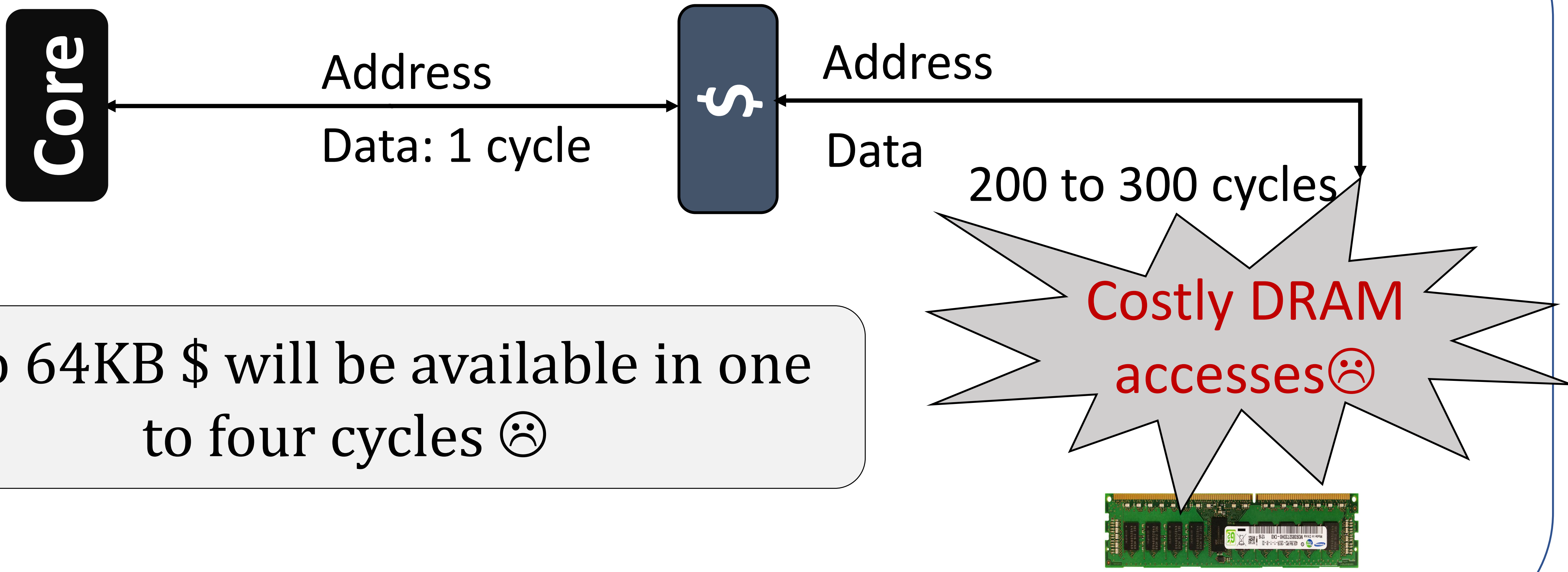
Latency: low
Area: low
Capacity: low

Latency: high
Area: high
Capacity: high

CASPER

Cache with latency

North pole ☺

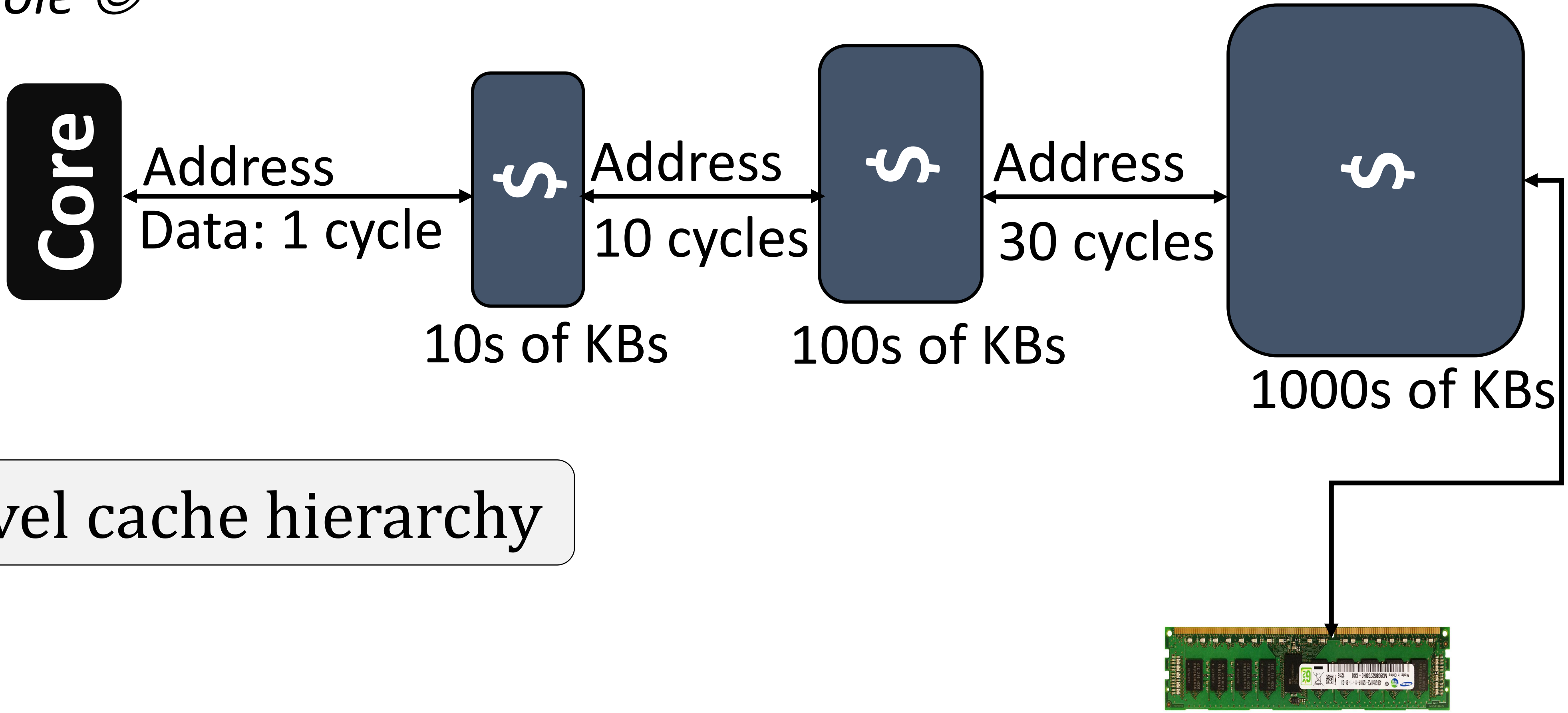


32 to 64KB $ will be available in one to four cycles ☹

South pole ☺

Cache hierarchy with latency

North pole ☺



Core

Address

Data: 1 cycle

$

10s of KBs

Address

10 cycles

$

100s of KBs

Address

30 cycles

$

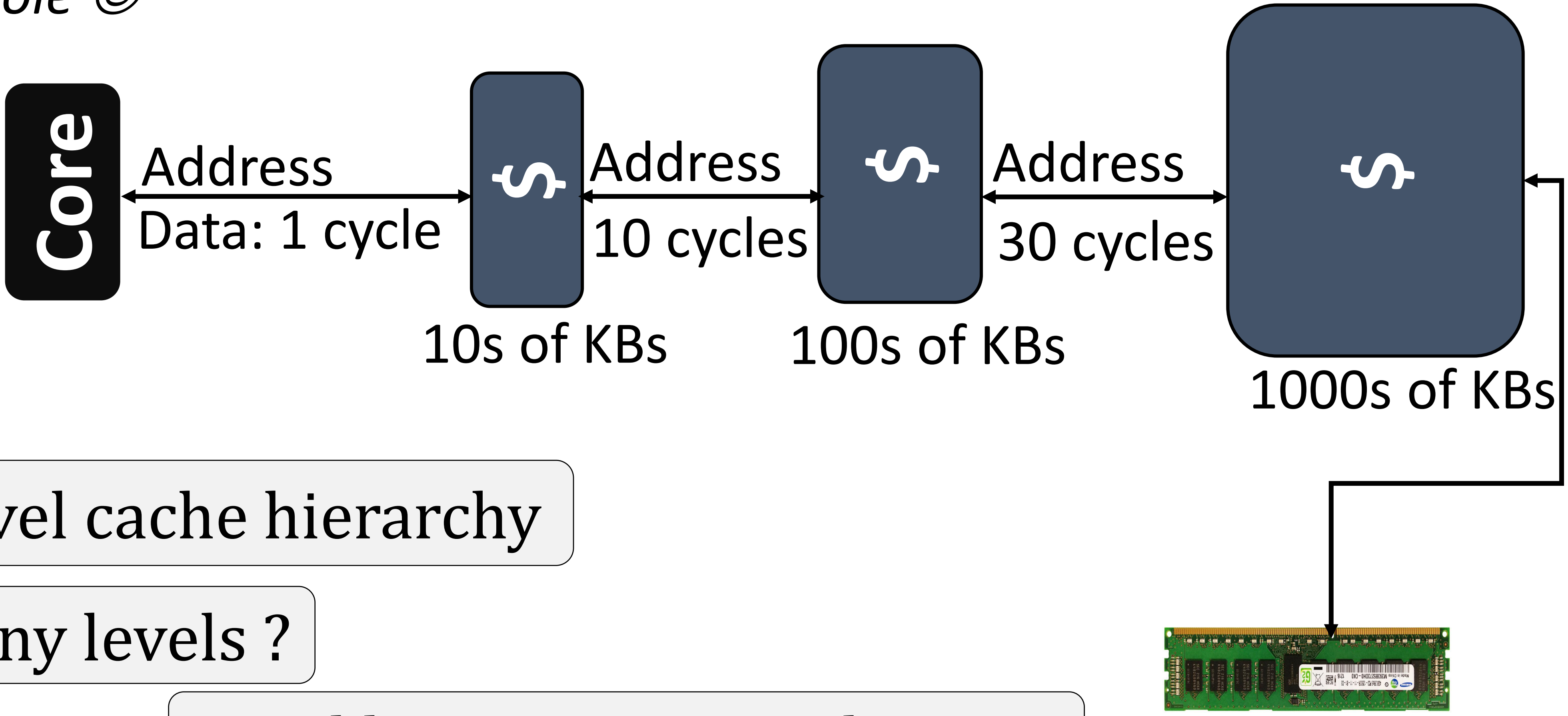
1000s of KBs

Multi-level cache hierarchy

South pole ☺

Cache hierarchy with latency

North pole ☺



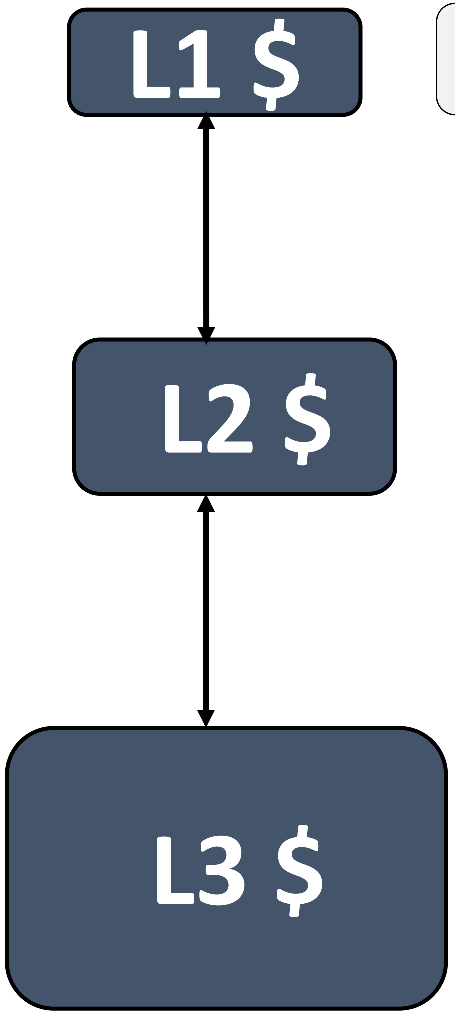
Multi-level cache hierarchy

How many levels ?

Total latency < DRAM latency

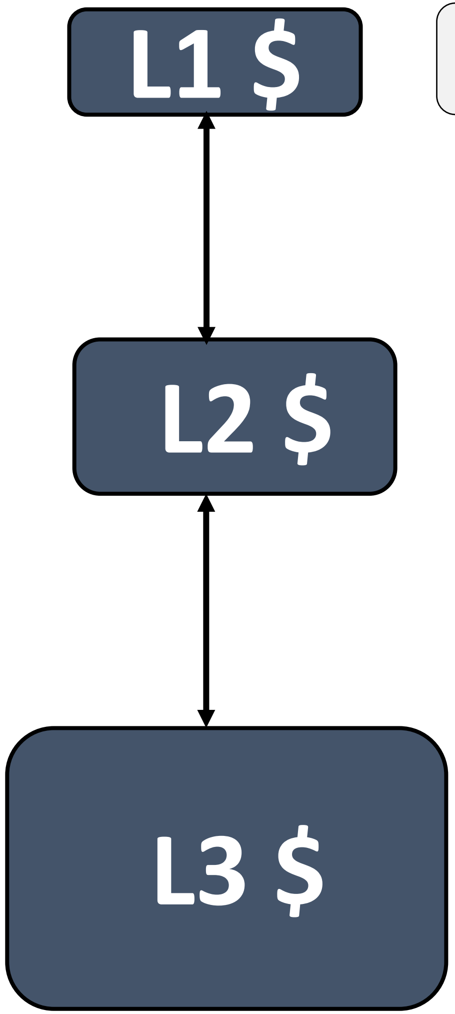
South pole ☺

Takeaway



Latency and bandwidth (multiple ports)

Takeaway



L1 $

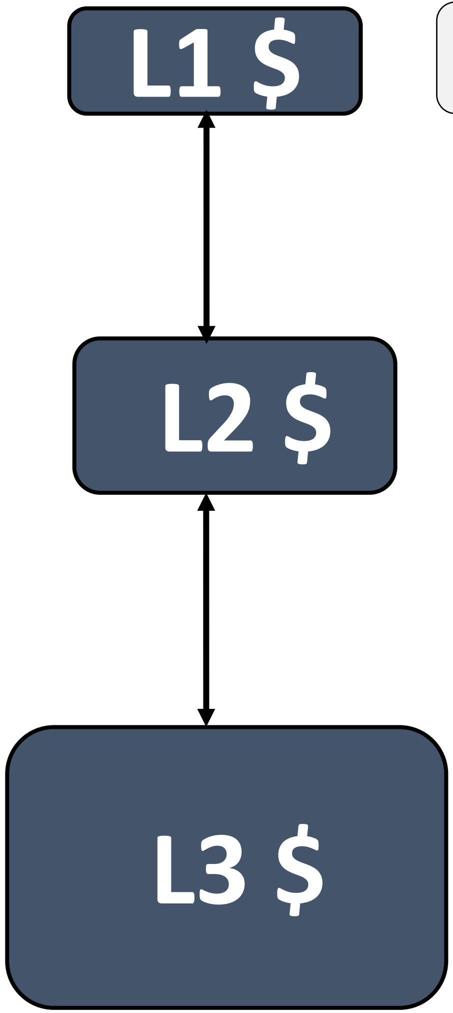
L2 $

L3 $

Latency and bandwidth (multiple ports)

Latency

Takeaway (Do not forget the word microarch.)



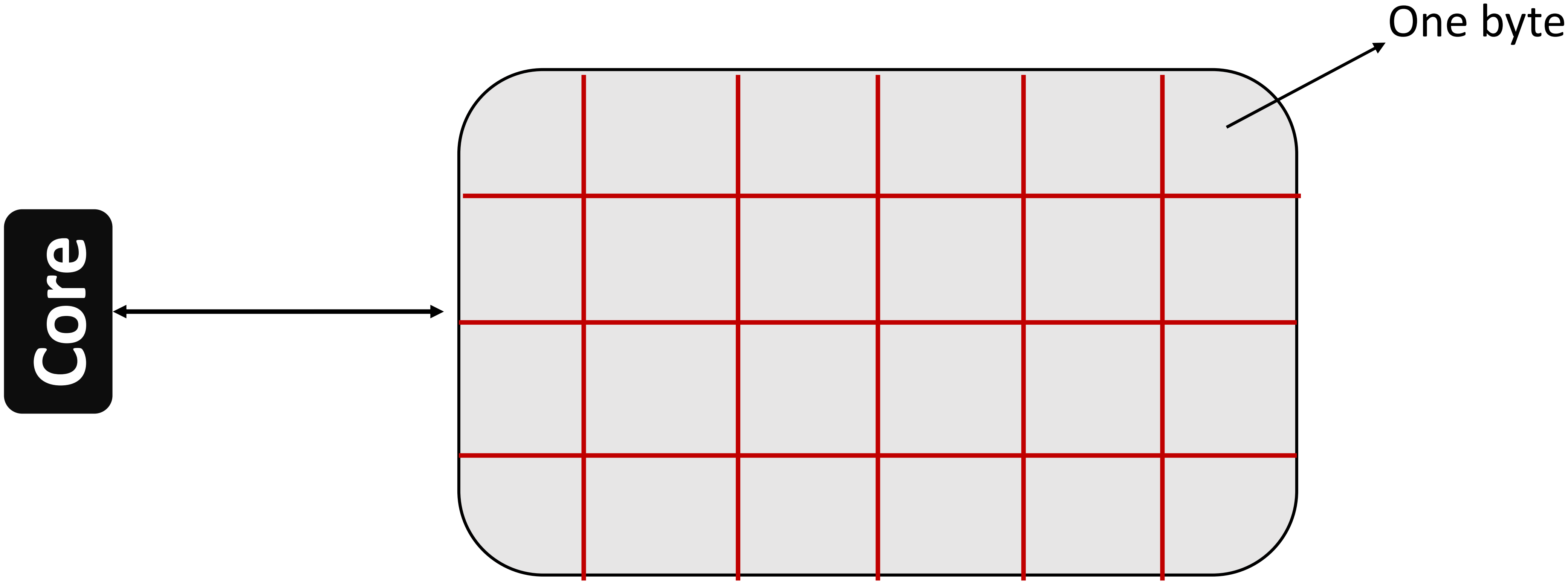
Latency and bandwidth (multiple ports)

Latency

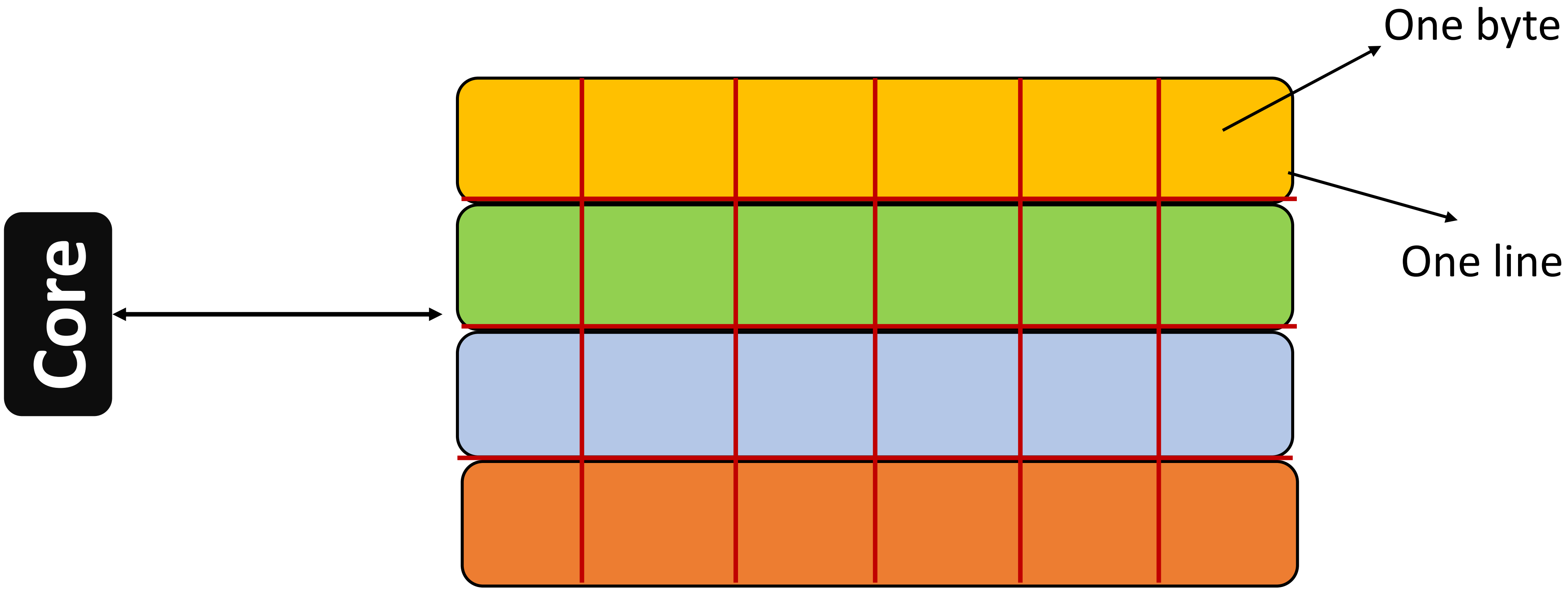
Capacity

PAUSE

Accessing a cache

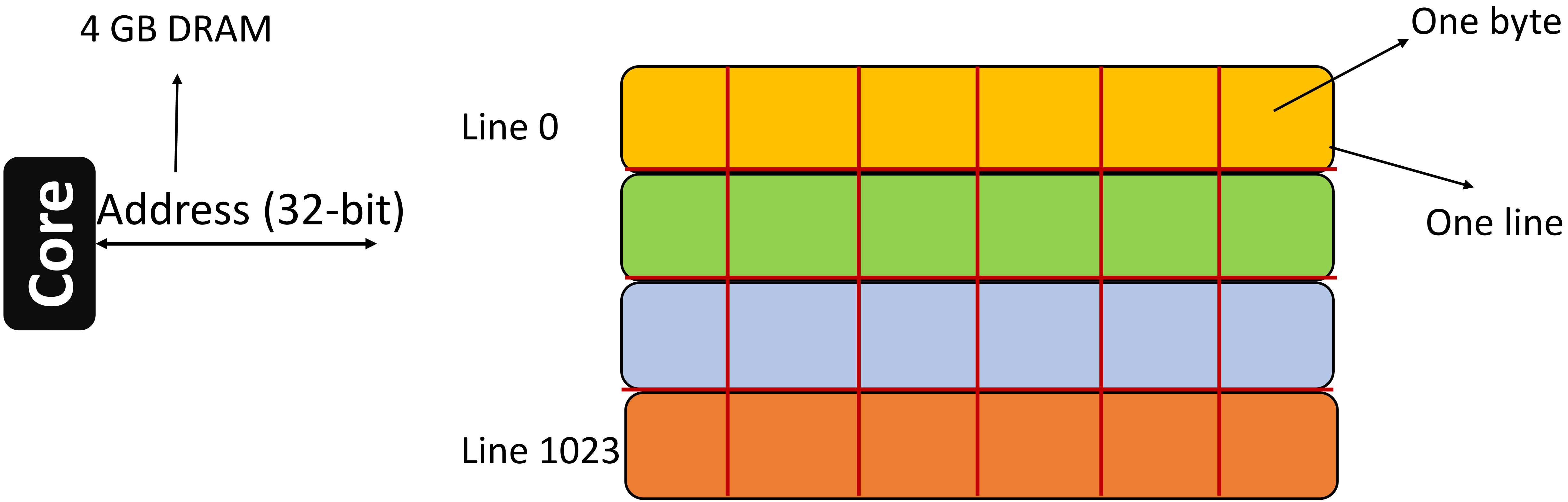


Bytes to blocks (lines)



Typical line size: 64 to 128 Bytes

A bit deeper: 1024 lines each of 32B



4 GB DRAM

Core

Address (32-bit)

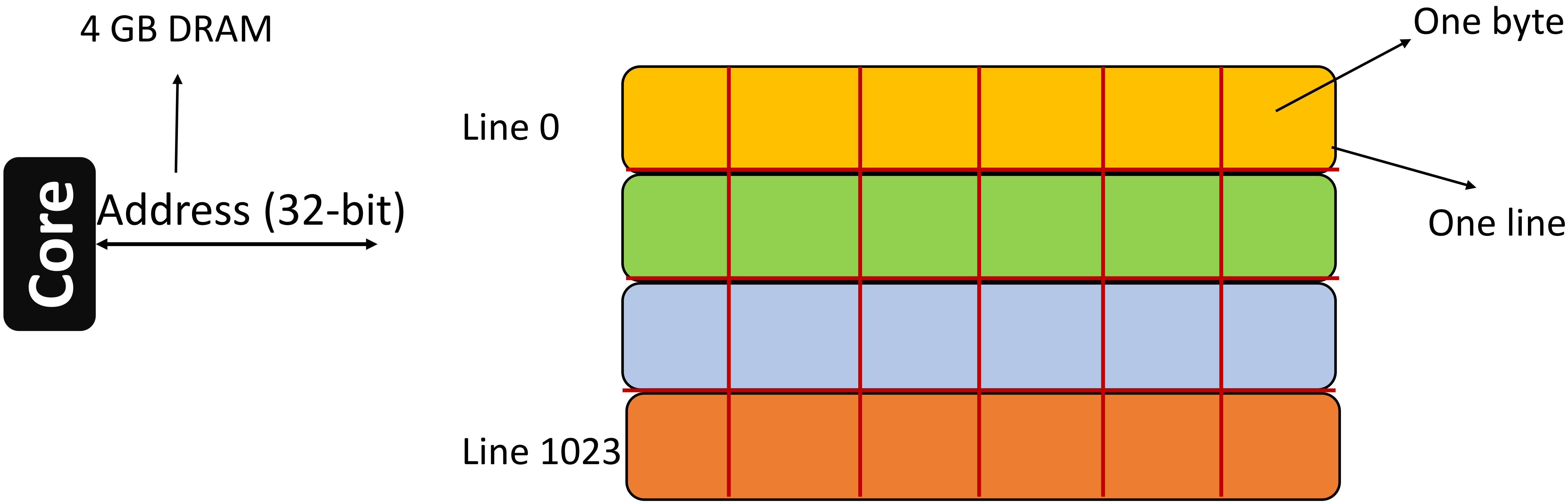
Line 0

Line 1023

One byte

One line

A bit deeper: 1024 lines each of 32B



4 GB DRAM

Core

Address (32-bit)

Line 0

Line 1023

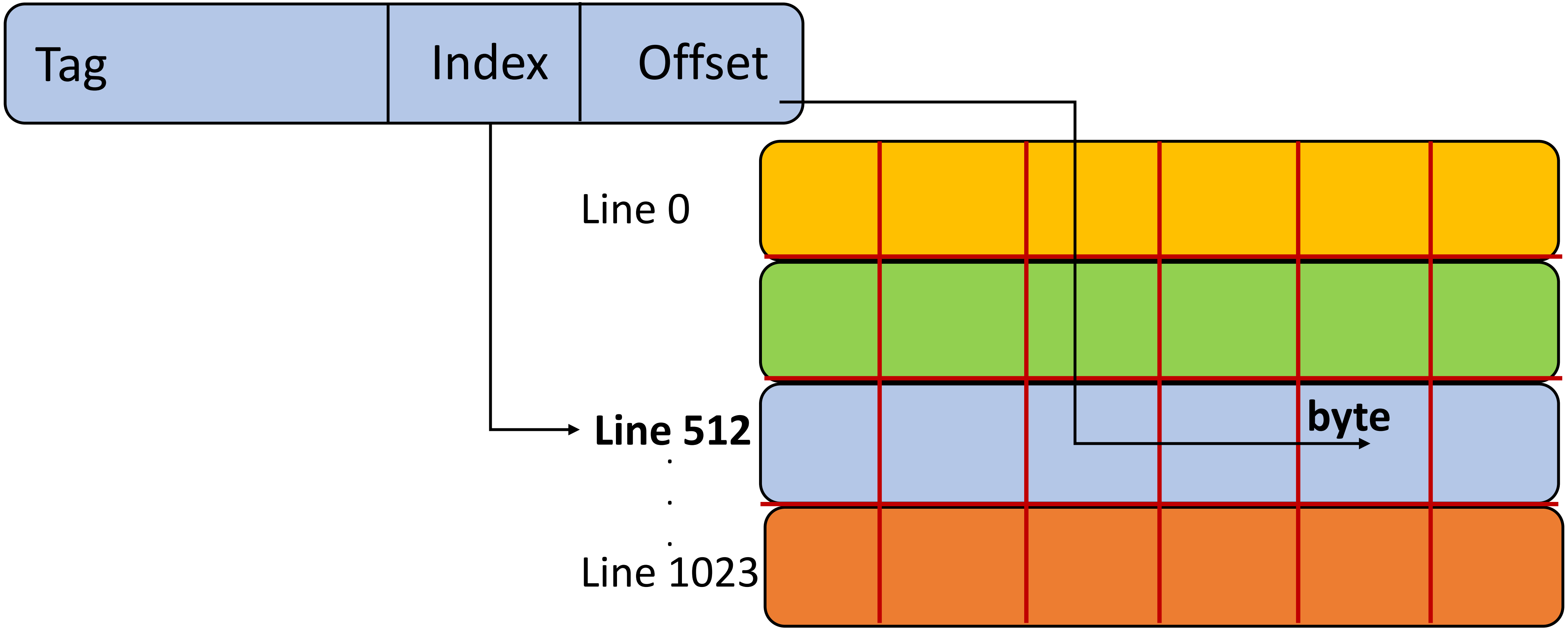
One byte

One line

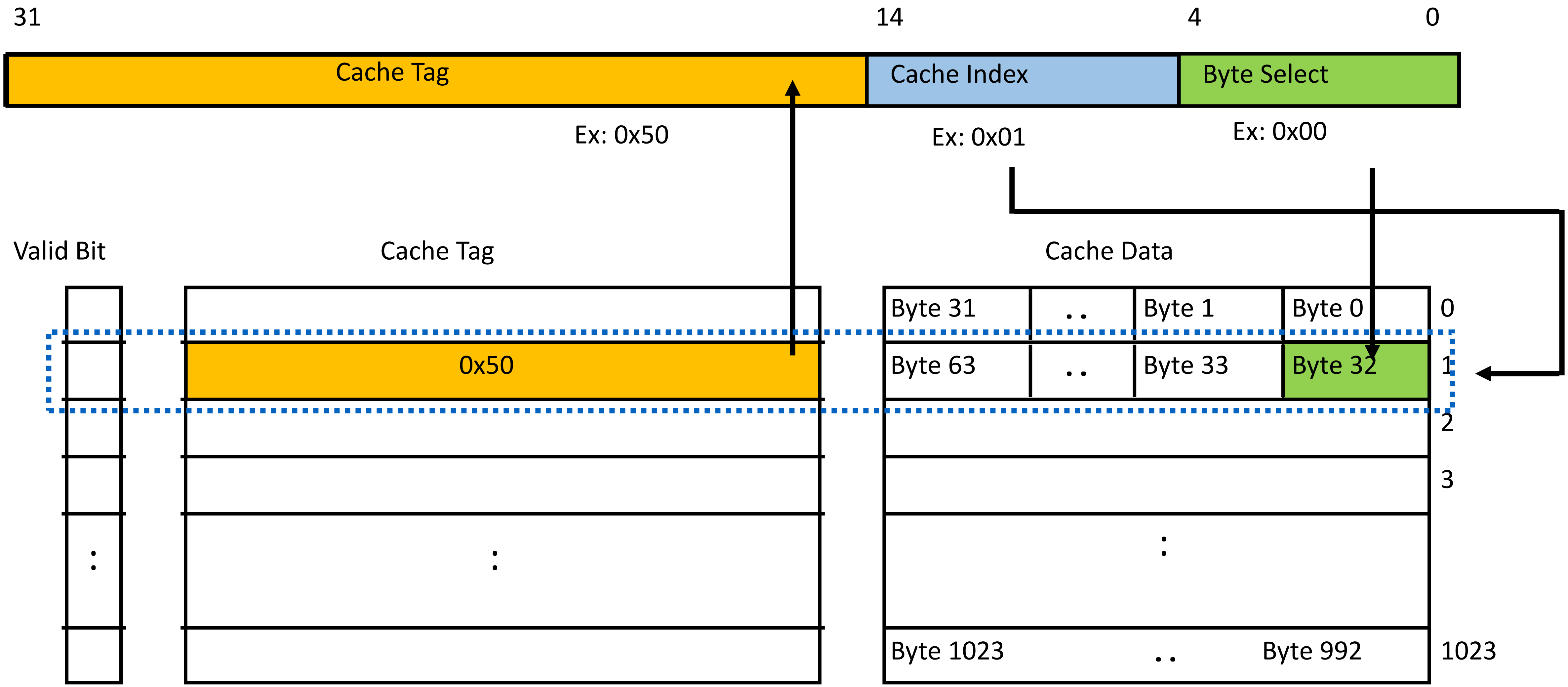
Line number (index): 10 bits

Byte offset (offset): 5 bits

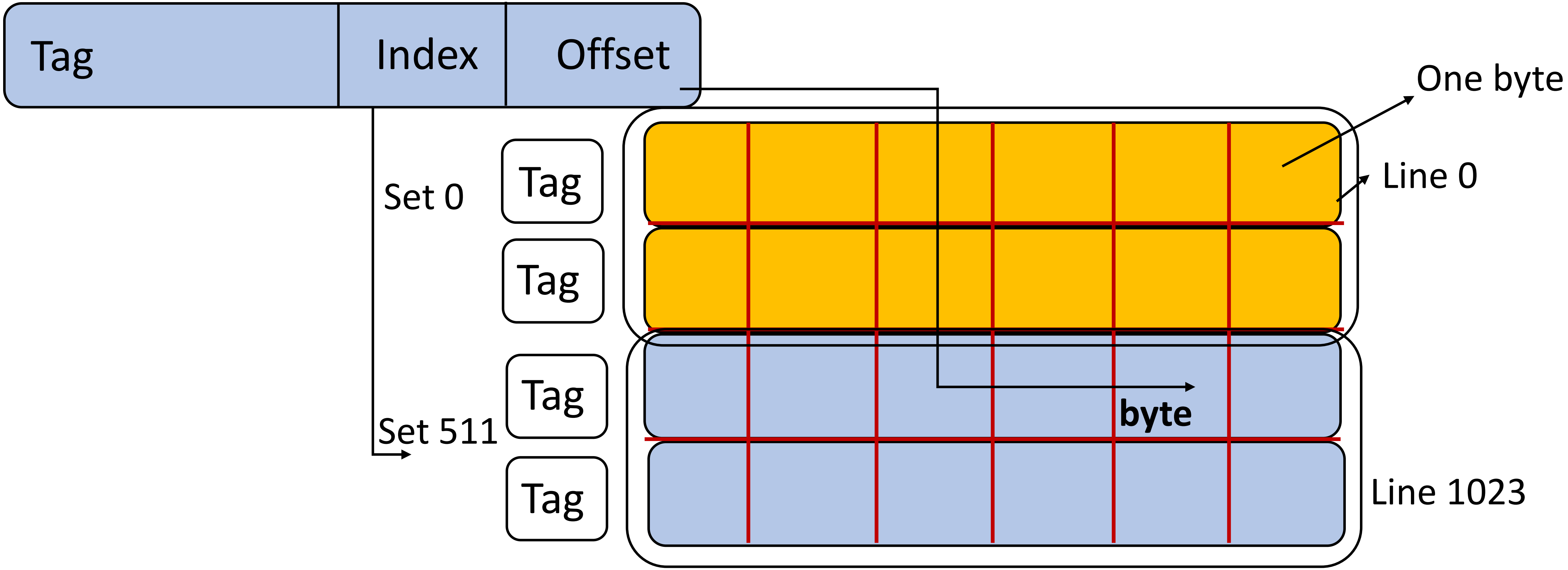
Direct Mapped Cache



Direct Mapped in Action

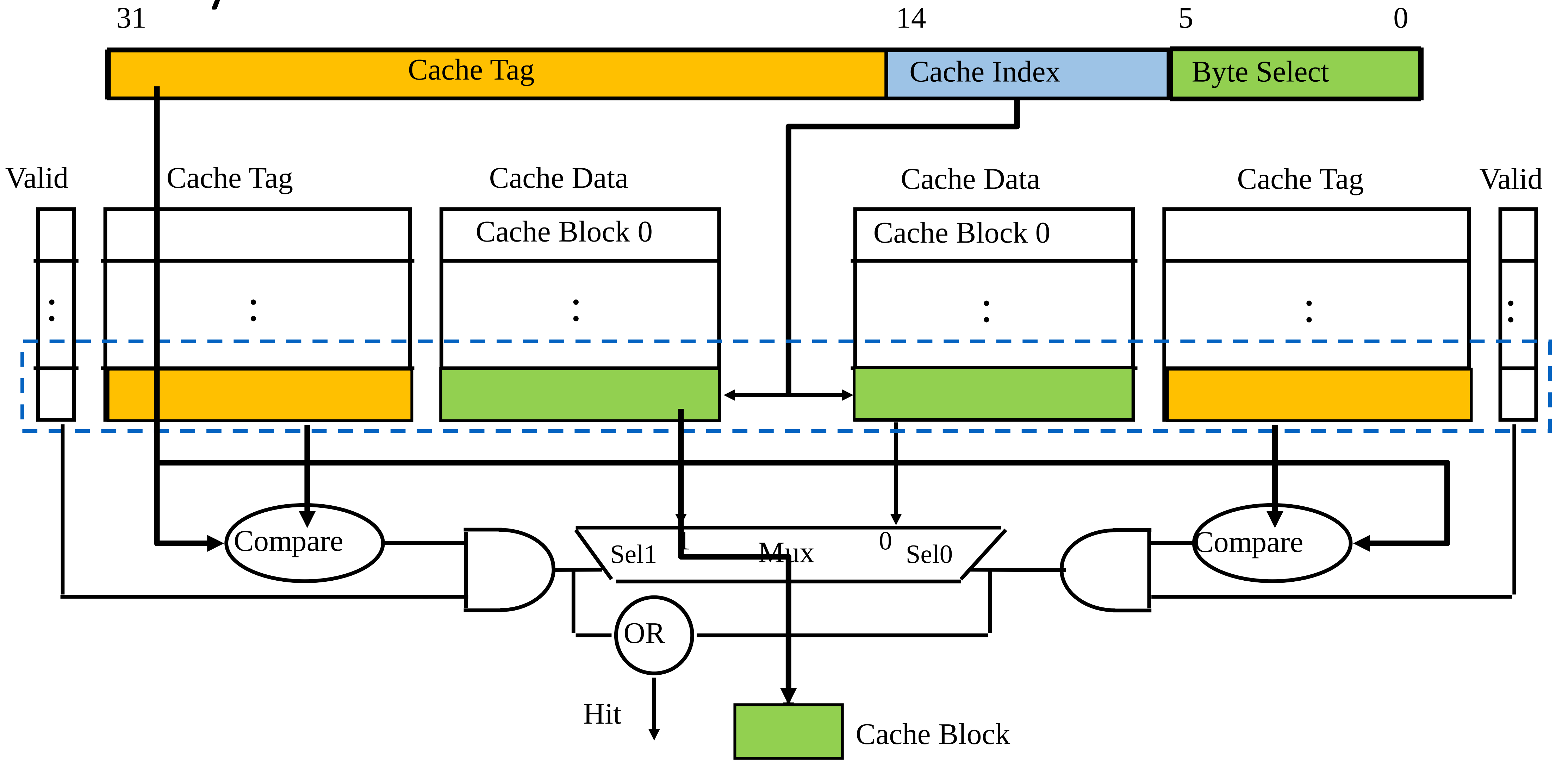


What if we have multiple ways?



CASPER

2-way associative in action



Knobs of interest

Line size, associativity, cache size

Tradeoff: latency, complexity, energy/power

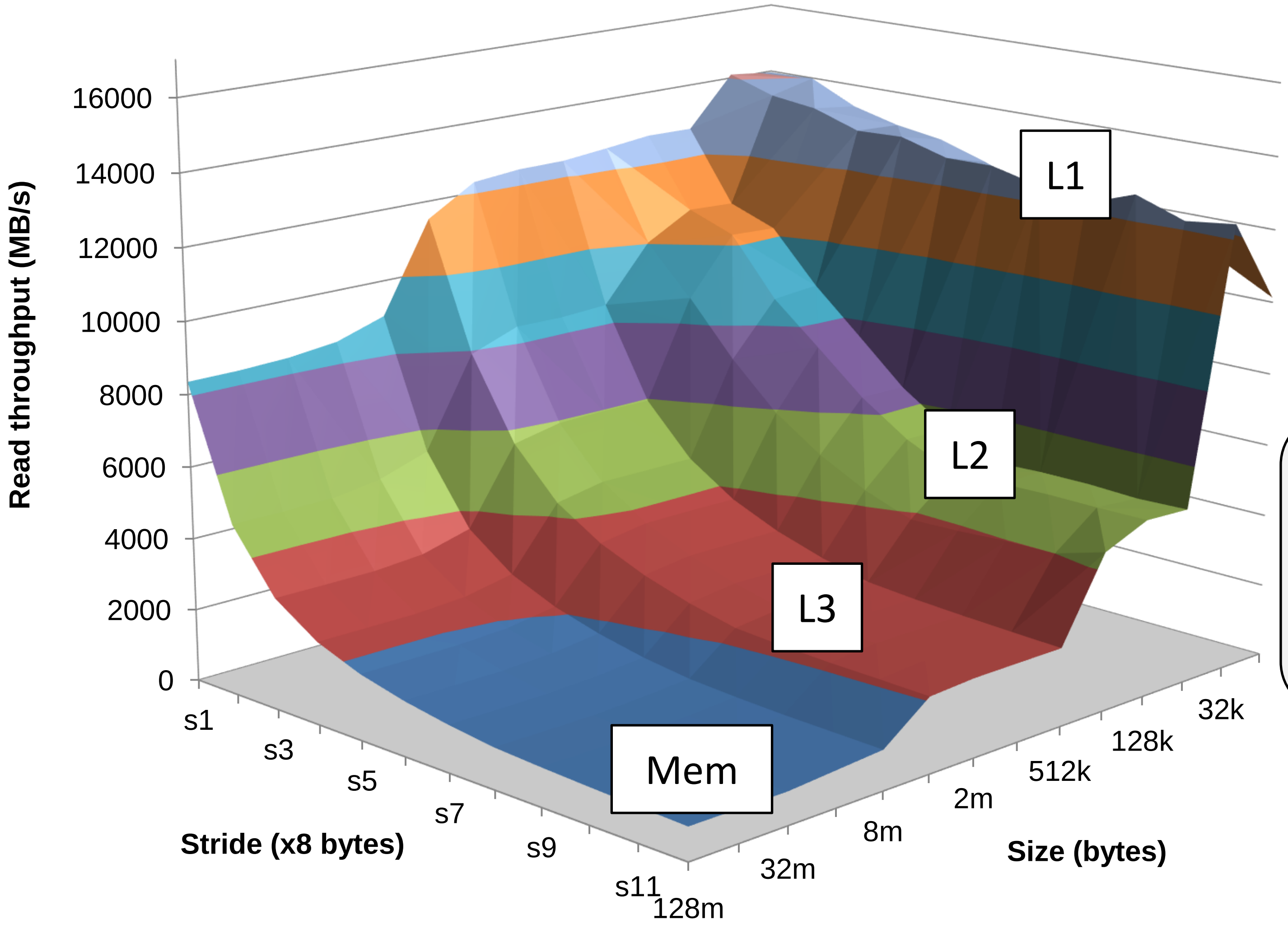
Tips: Think about the extremes:

Line size = one byte or cache size

Associativity = one or #lines

Cache size = Goal oriented: latency/bandwidth or capacity

Memory Mountain



Think about it, before you write your program: Assignment-I is live

Into the Real World

```
sudo dmidecode -t cache
cat /proc/cpuinfo
getconf -a | grep CACHE
lscpu
```

Wiki chip: https://en.wikichip.org/wiki/WikiChip

Perf tool: https://perf.wiki.kernel.org/index.php/Main_Page

```
sudo perf stat -e cache-misses ….
```

PAUSE