CPSC 313: Computer Hardware and Operating Systems

Unit 3: The Memory Hierarchy

THE UNIVERSITY OF BRITISH COLUMBIA
Computer Science
Faculty of Science

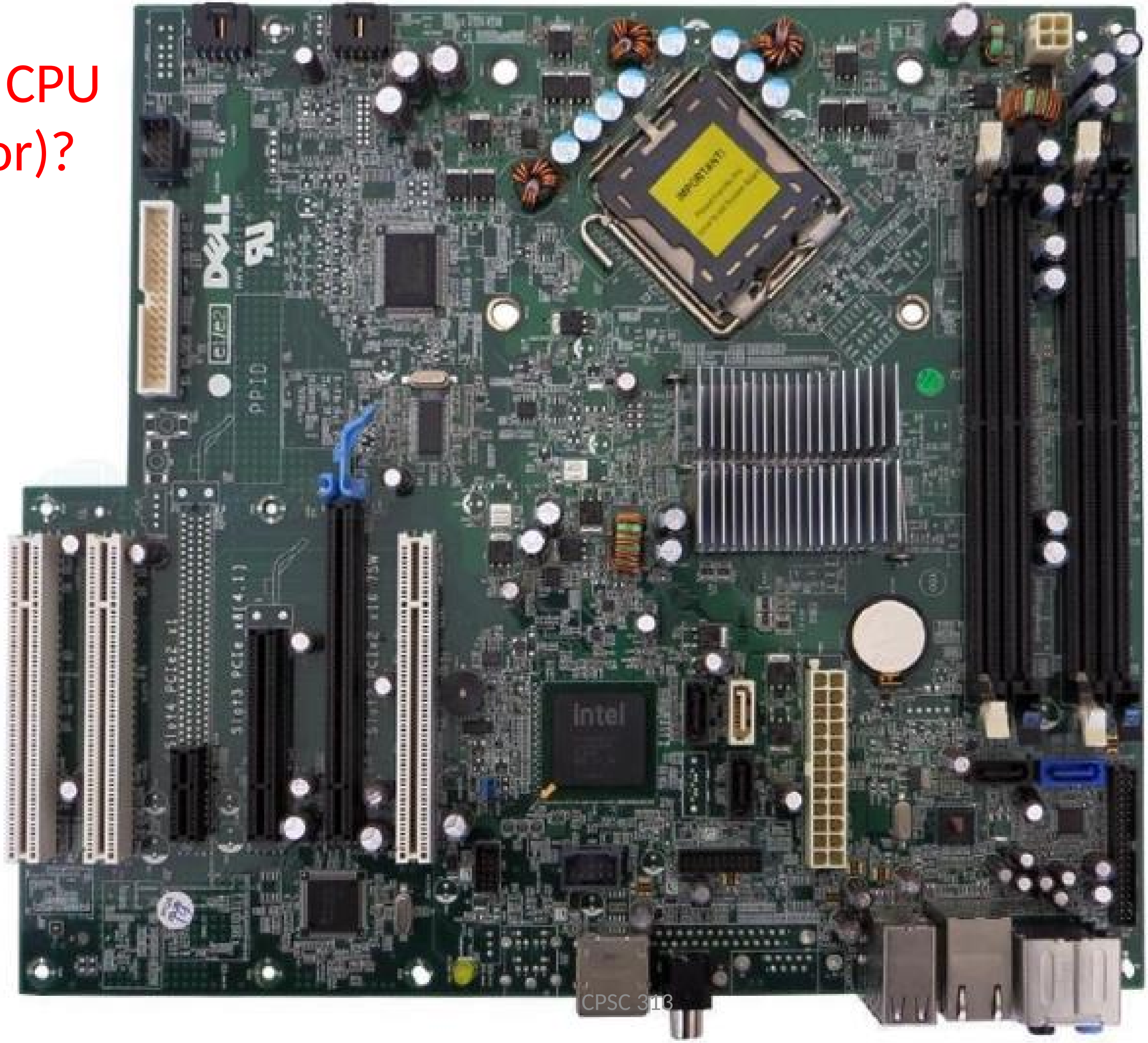
Administration

- Quiz 2: Don’t miss your time!
  - Quiz 2 information and practice quiz were released on Friday.
- Lab 5:
  - Due Sunday October 20th (in two weeks).
  - No class Friday because of Quiz 2
  - but we’ll be here to answer questions.
- Tutorial 4 this week!

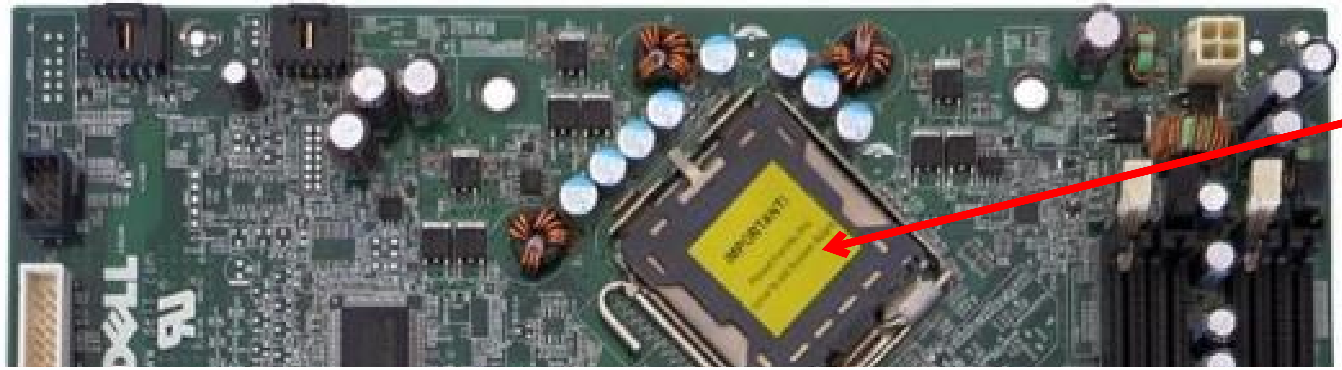
Today

- Learning Outcomes
  - Define memory hierarchy.
  - Evaluate the performance differences found at the different layers of the hardware memory hierarchy.
  - Explain the different kinds of caching that processors and hardware systems perform to mitigate the performance differences between the levels of the memory hierarchy.
- Reading
  - 6.2, 6.3

Where is the CPU (aka processor)?

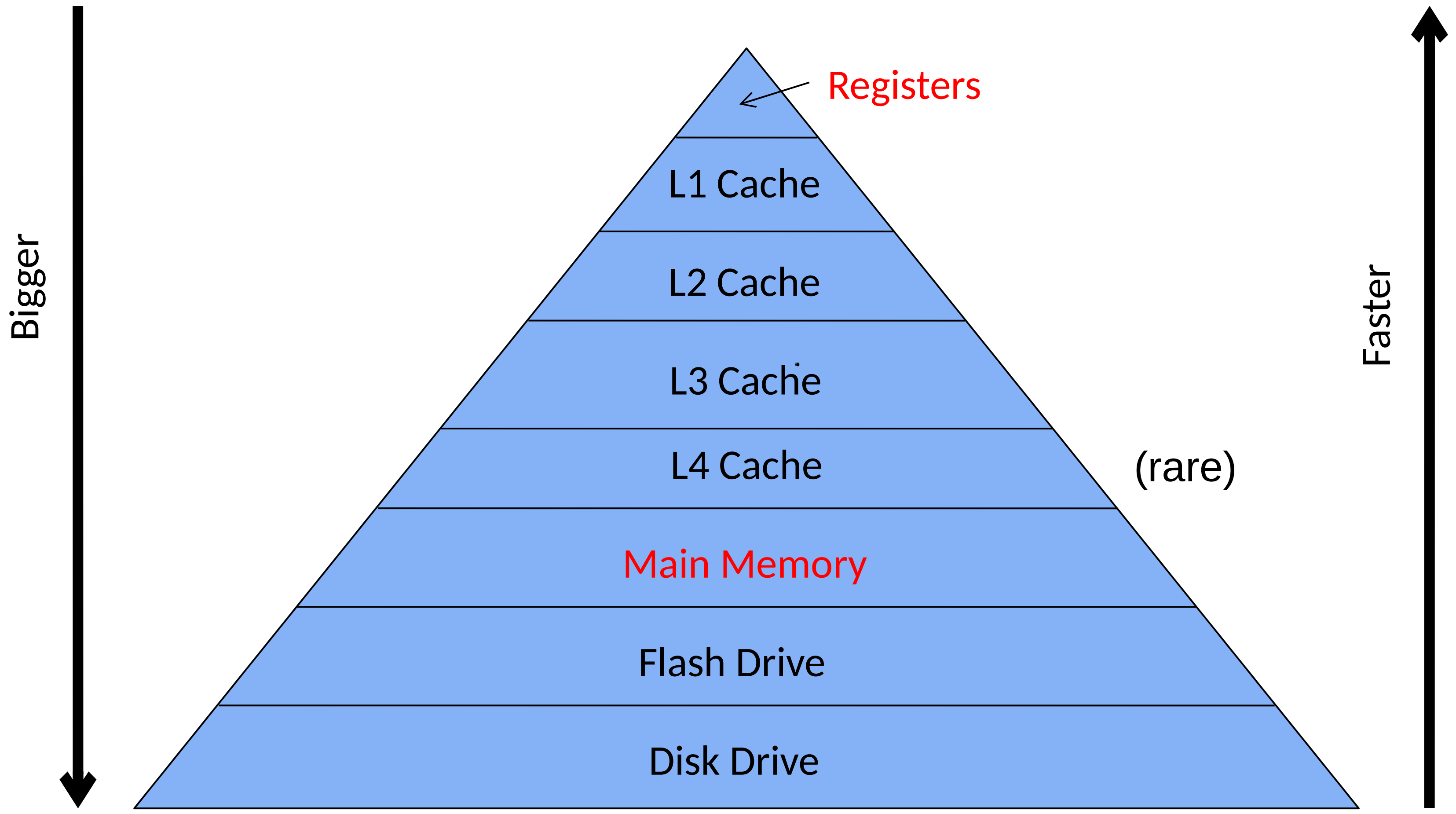


Where is the memory?

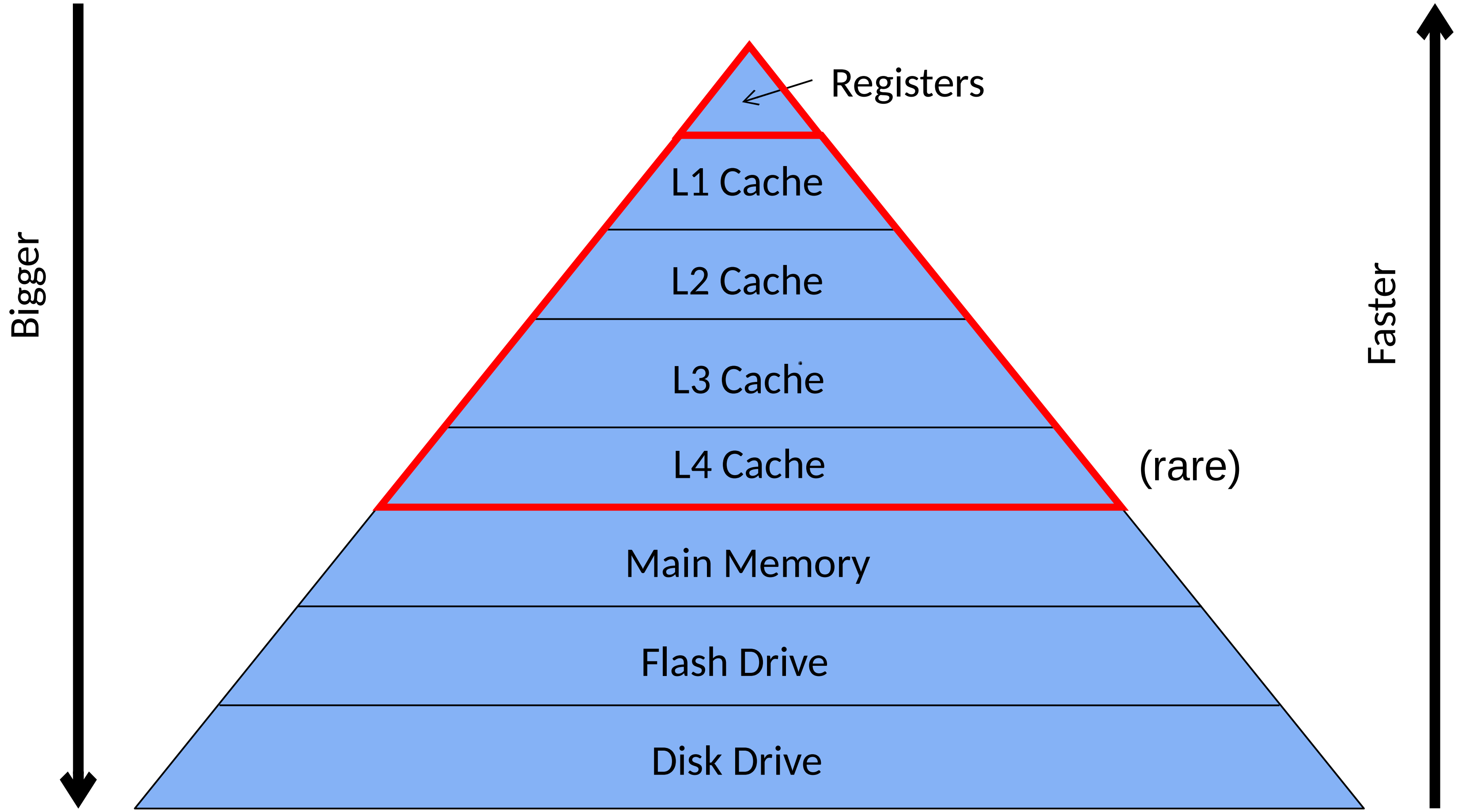


Factor of 100x in performance!

The Memory Hierarchy



The Memory Hierarchy



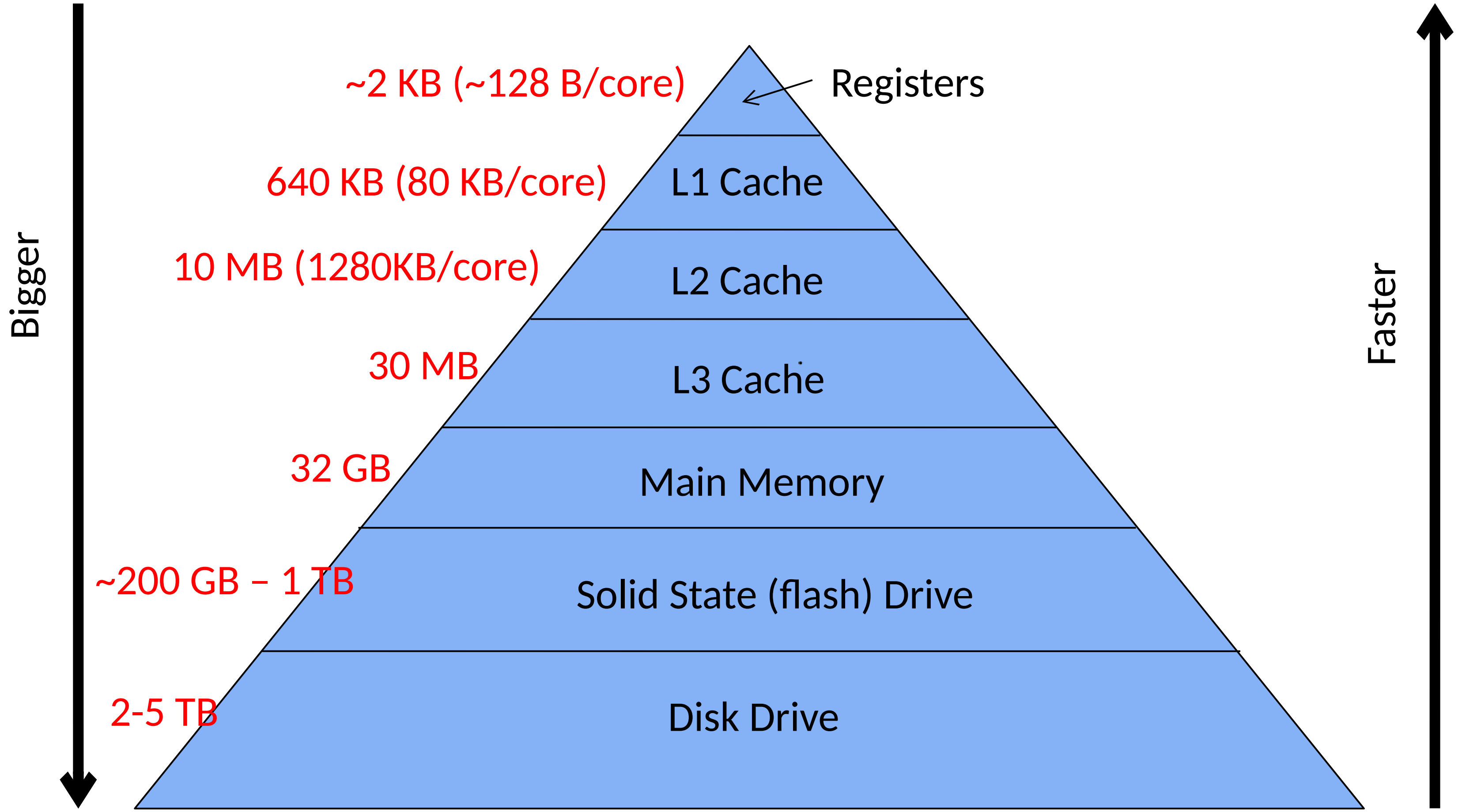
Intel Golden Cove (late 2021)

- Cycle time: 3.2 to 5.1 GHz => .2 to .3 ns/cycle (registers)
- L1: 1.25 ns (5 clock cycles; 80K/core = 640K total) PER CORE
- L2: 3.7 ns (15 clock cycles; 1280K/core = 10 MB total) PER CORE
- L3: 17 ns (67 clock cycles; 30 MB) SHARED
- Memory: 80 ns (i.e., Main memory, RAM)

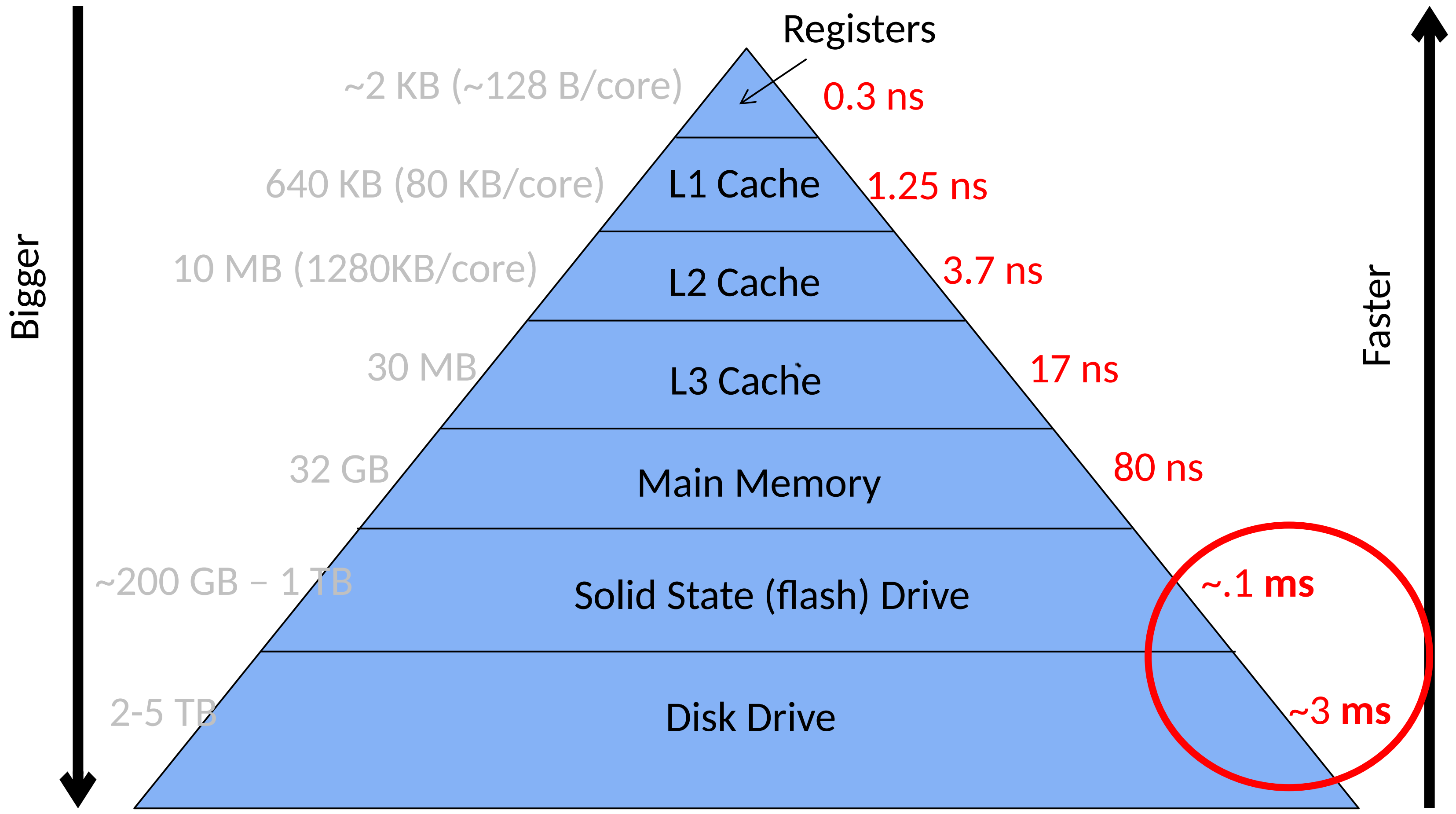
Intel Golden Cove (late 2021)

- Cycle time: 3.2 to 5.1 GHz => .2 to .3 ns/cycle (registers)
- L1: 1.25 ns (5 clock cycles; 80K/core = 640K total) PER CORE
- L2: 3.7 ns (15 clock cycles; 1280K/core = 10 MB total) PER CORE
- L3: 17 ns (67 clock cycles; 30 MB) SHARED
- Memory: 80 ns (i.e., Main memory, RAM)

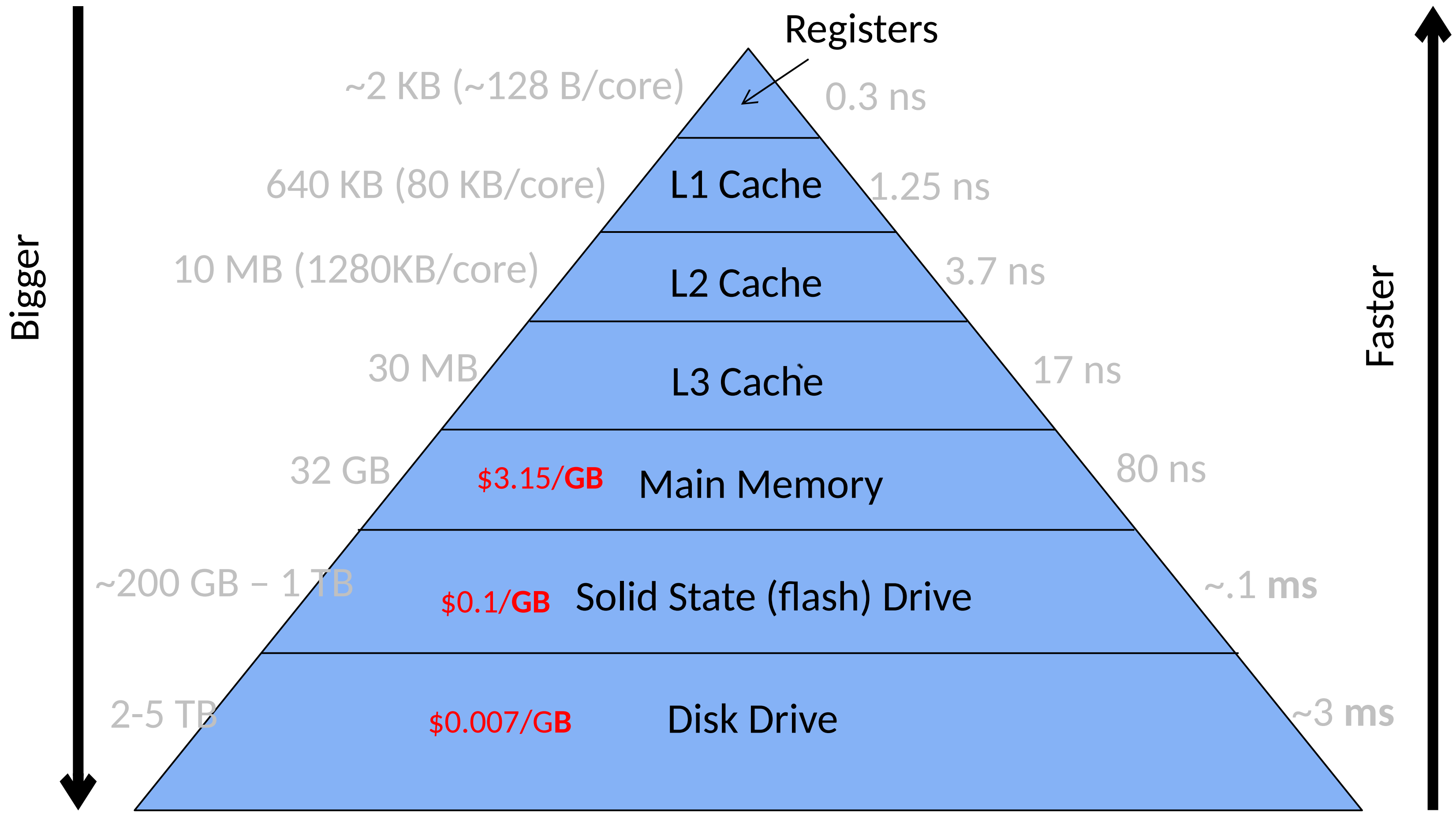
The Memory Hierarchy -- Size



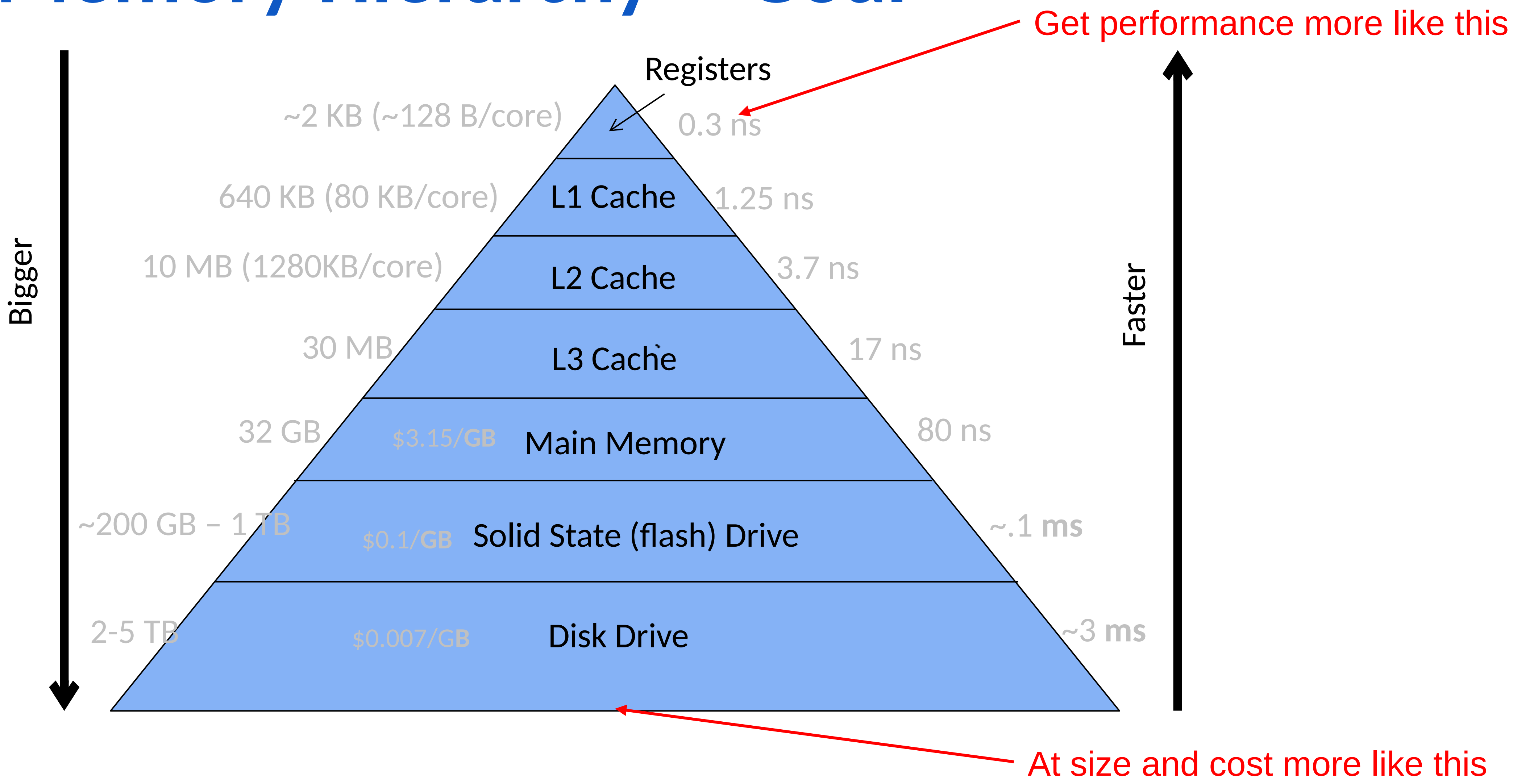
The Memory Hierarchy -- Latency



The Memory Hierarchy -- Price



The Memory Hierarchy -- Goal



Implications

How might this affect how machines are built or how we program? What are the tradeoffs/priorities?

- We saw many factors of 10 or 100 in:
  - Size
  - Performance
  - Price
- "When you see a factor of 100, it's going to affect how you program."
  – E. Kohler
- As the ratios between different parts of the system change, so do our priorities.
  - 1956:
    - $/MB(mem) : $/MB(disk)=> $411M : $9200 => 44,673 X
  - 2024:
    - $/MB(mem): $/MB(disk) => $0.00315 : $0.000007 => 450 X

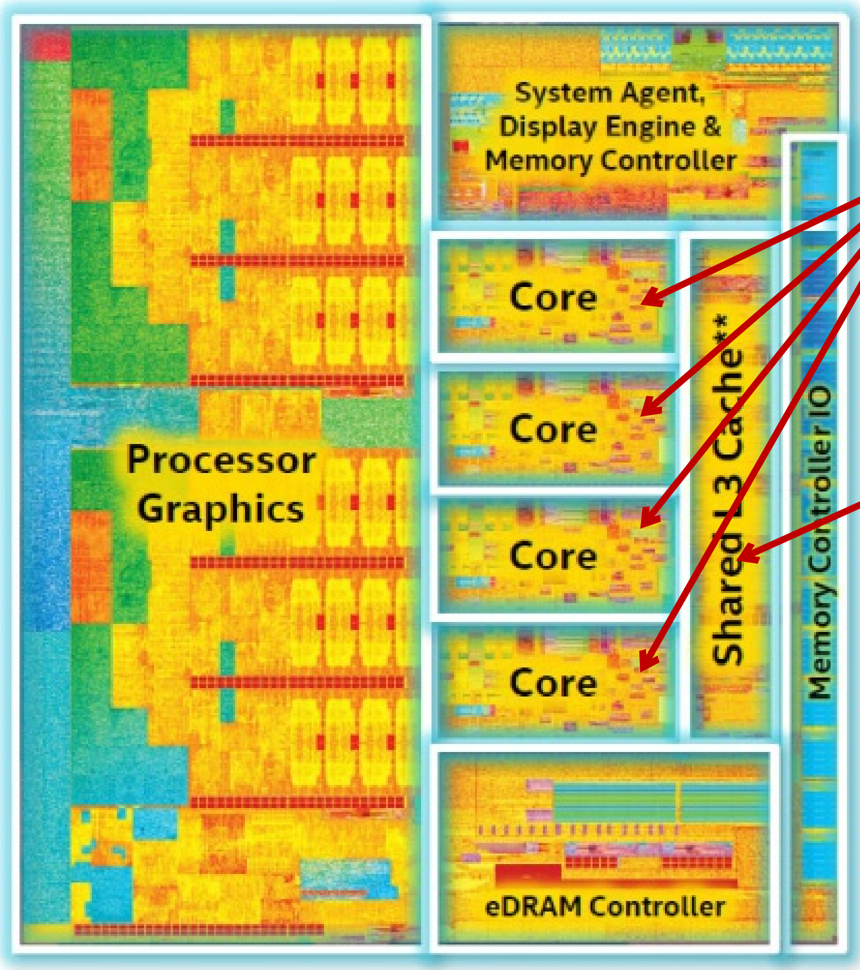
Price of memory over time

- Cost of memory in 1956 versus 2024:
  - 1956 $ / MB : 2024 $ /MB => $411M/.00315 => 130 trillion times cheaper
- Cost of disk in 1956 versus 2015:
  - 1956 $ / MB : 2024 $ /MB => $9200/.000007 => 13 trillion times cheaper

Caching

- Definition:
  - Colloquially: store away in hiding or for future use
  - Applied to computation:
    - Placing data somewhere it can be accessed more quickly
- Examples:
  - In assembly language: we move data from memory to registers.
  - In the hardware: we move data from main memory into memory banks that live on the processor (more on this in a moment).
  - In software: we read things into our program’s local buffers and manipulate them there.

Processor Caches



L1 and L2 are per core

L3 is shared across cores

Intel Core i7-5775C

Inclass Exercise: Caching is *Everywhere*
(This is ***not*** an example of the HW memory caches we'll focus on this month!)

- You will want to use the student machines, not a workspace!
- Learning objectives:
  - Use both **file system system calls** (open/close/read/write), and **standard IO calls** (fopen/fclose/fread/fwrite)
  - Explain why the following things have enormous impacts on performance:
    1. Buffer size (e.g., how many bytes you send to write or fwrite)
    2. Use of write versus fwrite.

Use UNIX (Linux) man pages

- Accessible either via the CLI `man` command, or
- Googling "man read"

Wrapping Up

- Caching is ubiquitous throughout our computing systems:
  - In the processor
  - In the operating system
  - In databases
  - In middleware
  - In applications
- Writing efficient and correct software requires a deep understanding of caching and its implications.