

CSCI 210: Computer Organization

Lecture 3: Inside Your Computer

Stephen Checkoway

Oberlin College

Slides from Cynthia Taylor

Announcements

- Problem set 0 due Friday at 11:59 p.m.
 - On GradeScope, linked from BlackBoard

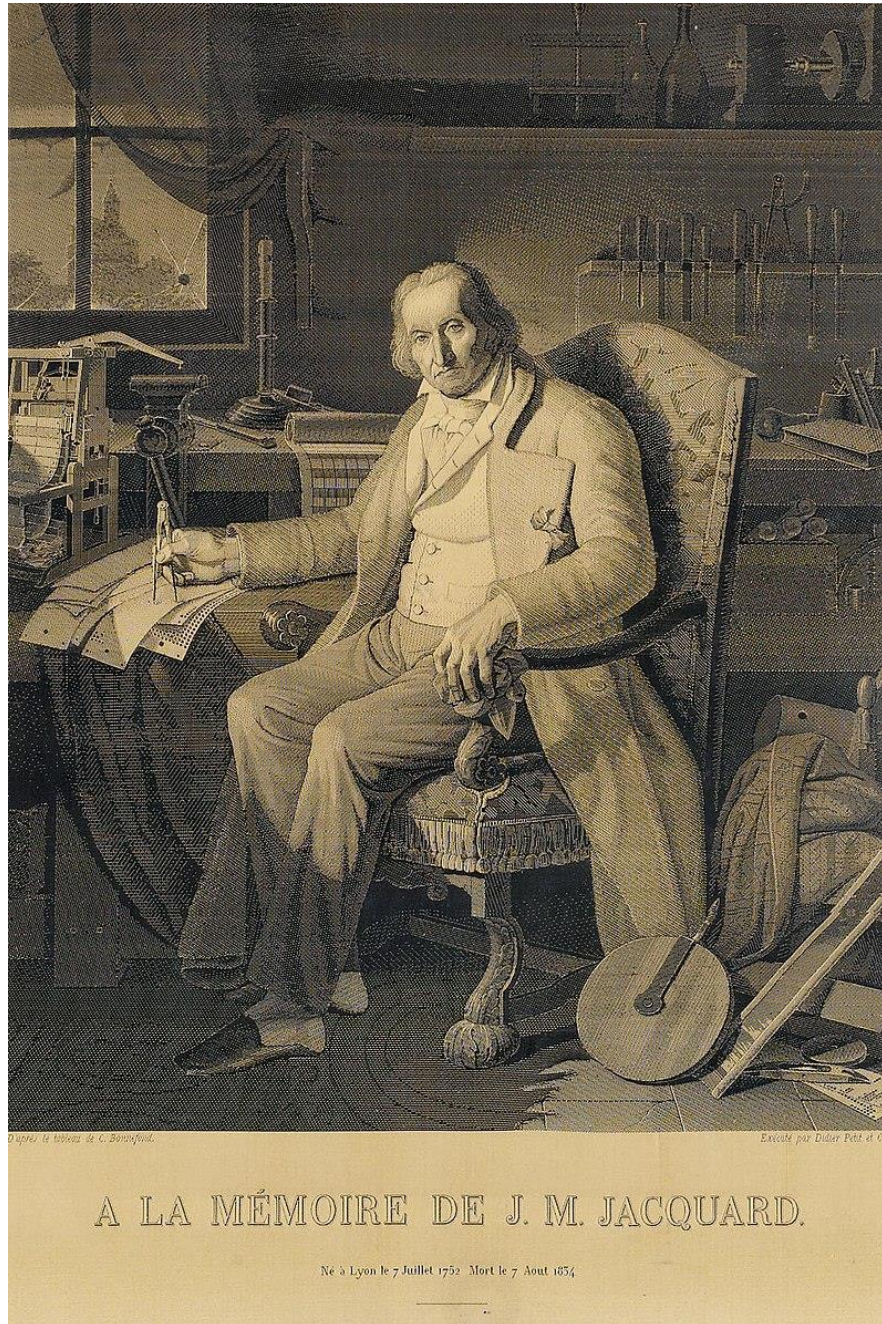
CS History: The Jacquard Loom



- Uses punch cards to store weaving patterns
- The first stored program machine
- Allows a single weaver to create intricate patterns

The Jacquard Loom

- Weavings of Jacquard are produced and sold
- Charles Babbage buys one and is inspired to use punch cards in the Analytical Engine



CS History: The Luddites

- How do we use the word “luddite”?

CS History: The Luddites

- A group of weavers angry about their skilled labor being displaced by mechanized looms
- They smashed the new mechanized looms as a strike action

Group Discussion: What are similar movements or discussions happening today?

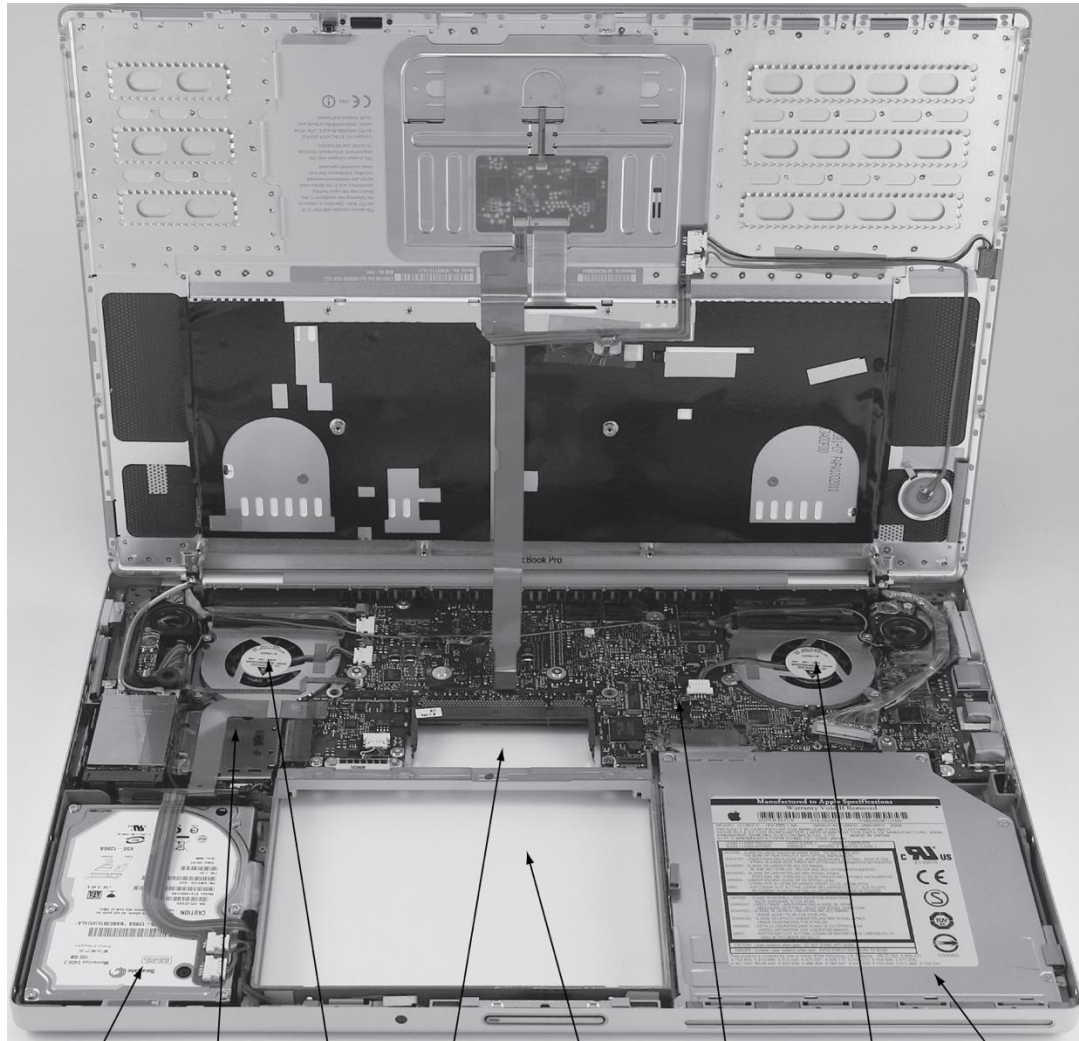
What's Inside a Computer

- CPU
 - Processes instructions
- Hard drive/Solid state drive (SSD)
 - Stores data, nonvolatile
- RAM
 - Stores data currently in use

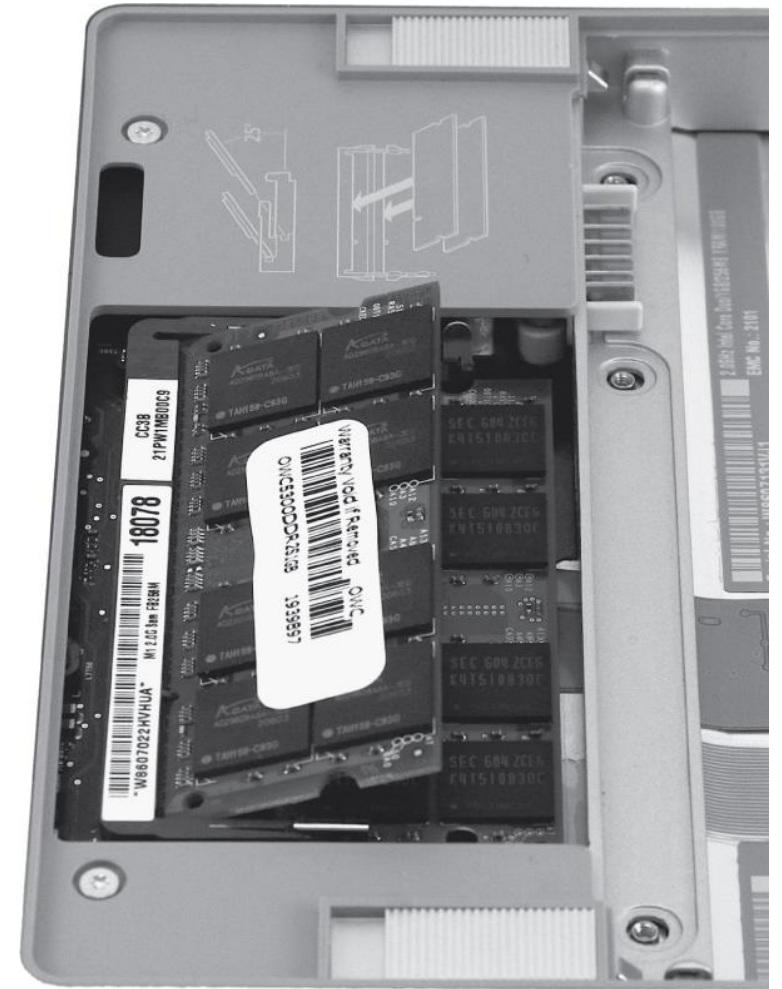
What's Inside a Computer

- Motherboard
 - Connects everything
- Graphics card, Networking Card
 - I/O devices
- Monitor, Keyboard
 - Peripherals

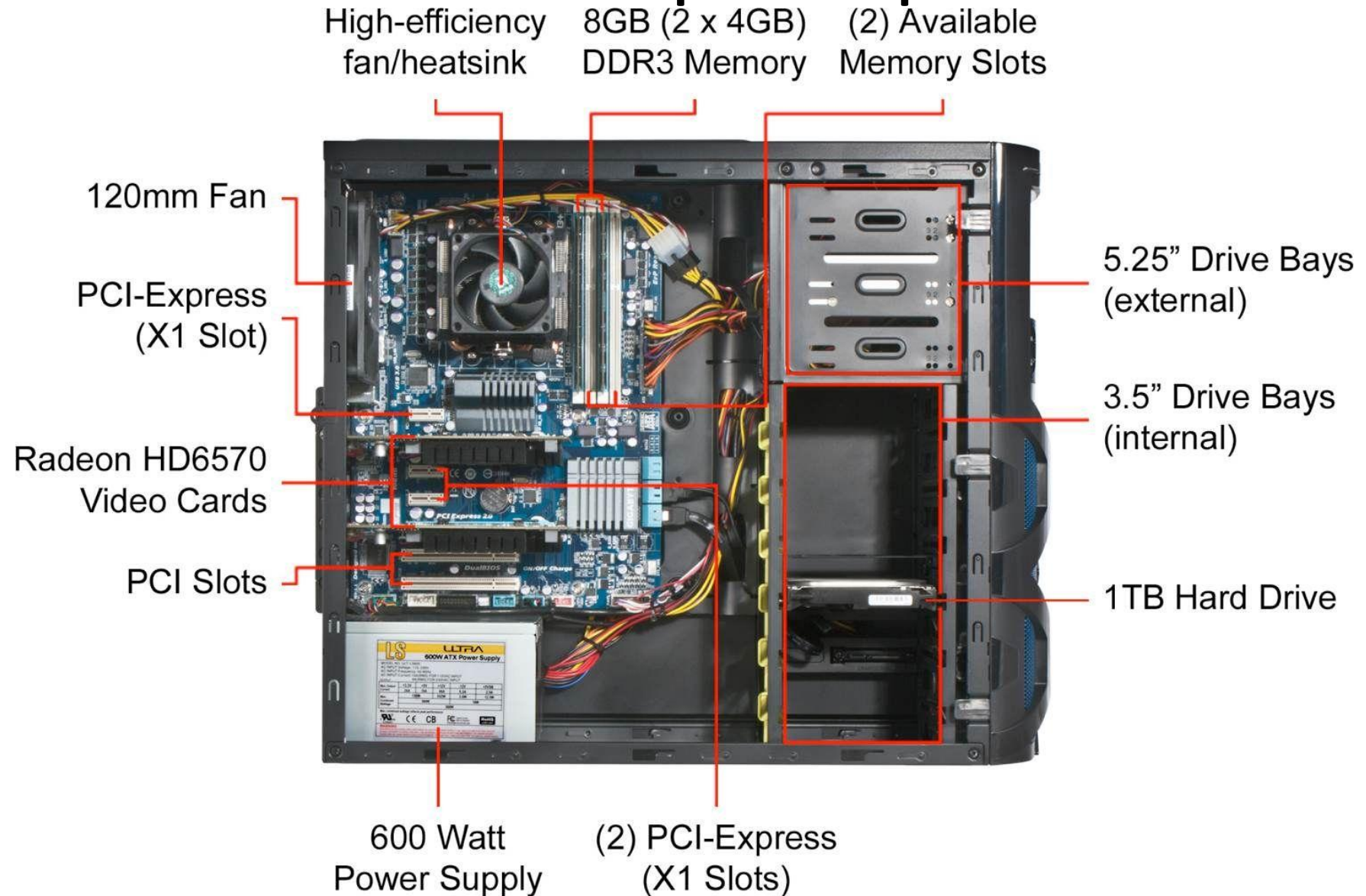
Opening the Box



Hard drive Processor Fan with cover Spot for memory DIMMs Spot for battery Motherboard Fan with cover DVD drive

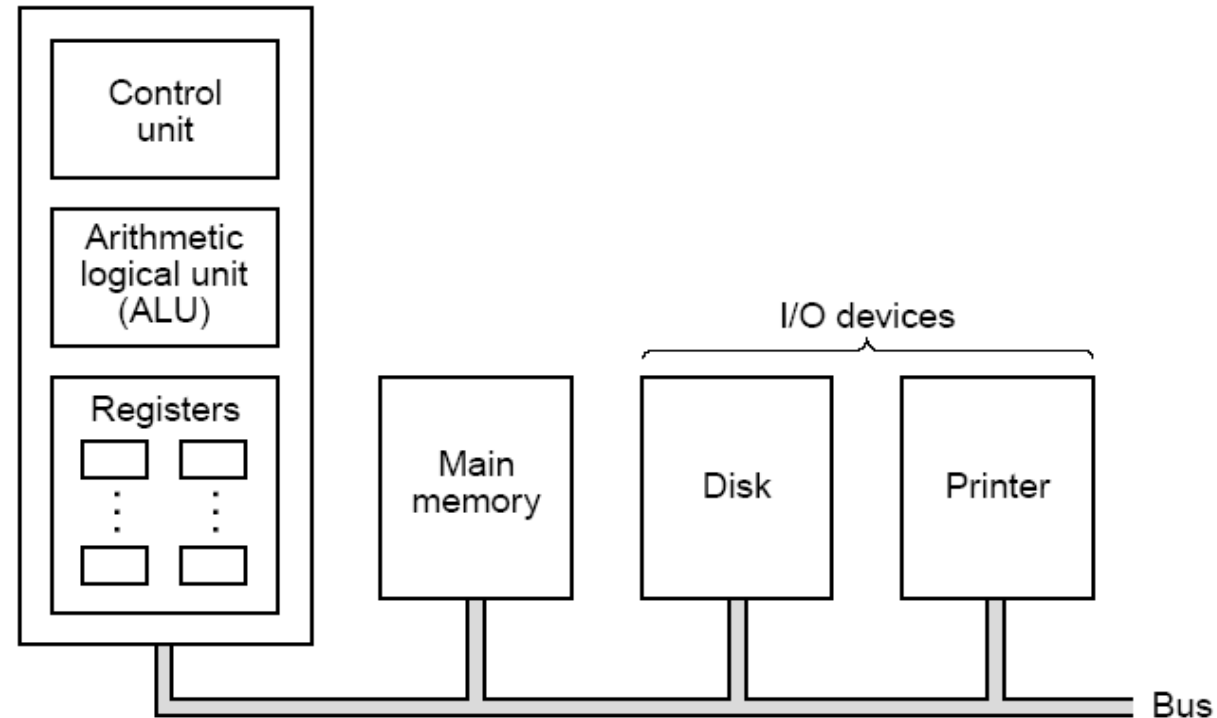


Inside a desktop computer



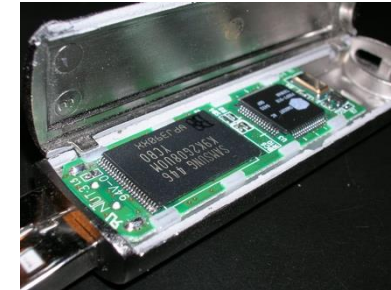
Very simplified diagram

Central processing unit (CPU)



A Safe Place for Data

- Volatile main memory
 - Loses instructions and data when power off
- Non-volatile secondary memory
 - Magnetic disk
 - Flash memory
 - Optical disk (CDROM, DVD)



Main Memory (RAM)

Index	Data
0	0011 1000
1	0000 0011
2	0111 0001
...	...
4294967295	0001 1000

4 GB of RAM

Basic structure: A 1-dimensional array of cells, each with a unique address. A cell is normally one byte (8 bits).

Basic Memory Operations

- read (load) the contents of the cell at a given location
- write (store) a given value to the cell at a given location
- Bytes may be grouped into 2-, 4-, or 8-byte units. A **word** is the name given to the basic unit of storage for binary integers, instructions, register sizes

Fun* word naming facts

- Different architectures use different numbers of bytes for a word
- The number of bytes in a word is typically the same as the number of bytes in a register and the number of bytes needed to refer to an arbitrary location in memory
- As architectures get updated, registers get larger but a word remains the same size
 - Intel x86 started as a 16-bit architecture so a WORD is 2 byte, a DWORD (double word) is 4 bytes, and a QWORD (quad word) is 8 bytes. This remains true today even as the registers have expanded to 8 bytes
 - RISC-V has 32-bit and 64-bit variants (called RV32 and RV64). The convention used is a word is 4 bytes and a doubleword is 8 bytes
 - Since we're using RV64 in this class, our registers will be doublewords in size

How much slower is it to get a byte from main memory (RAM) instead of the registers?

- A. 2x slower
- B. 10x slower
- C. 100x slower
- D. 1000x slower
- E. None of the above

Cache Memory

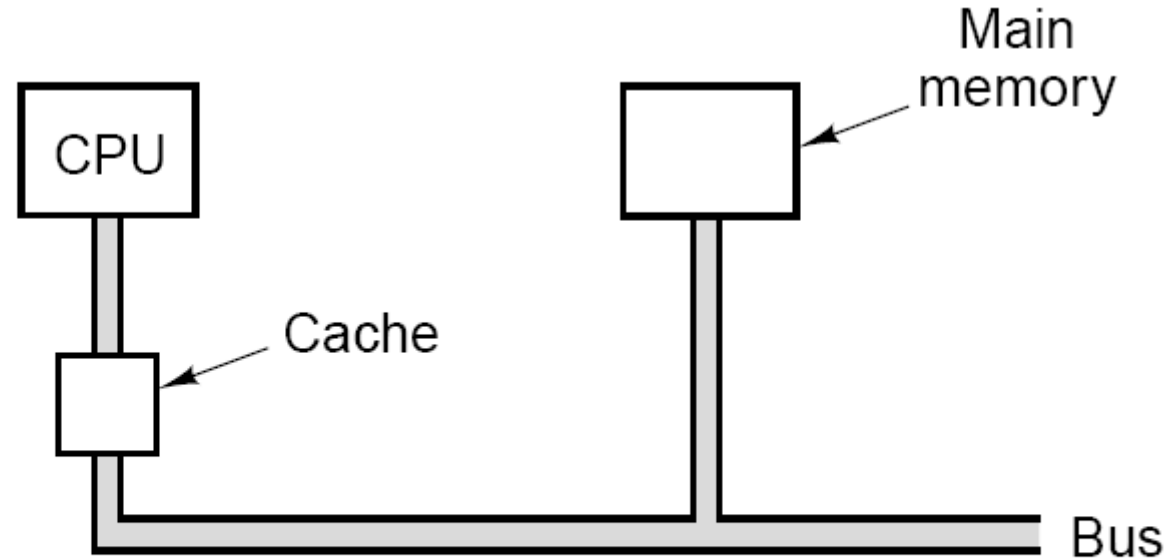
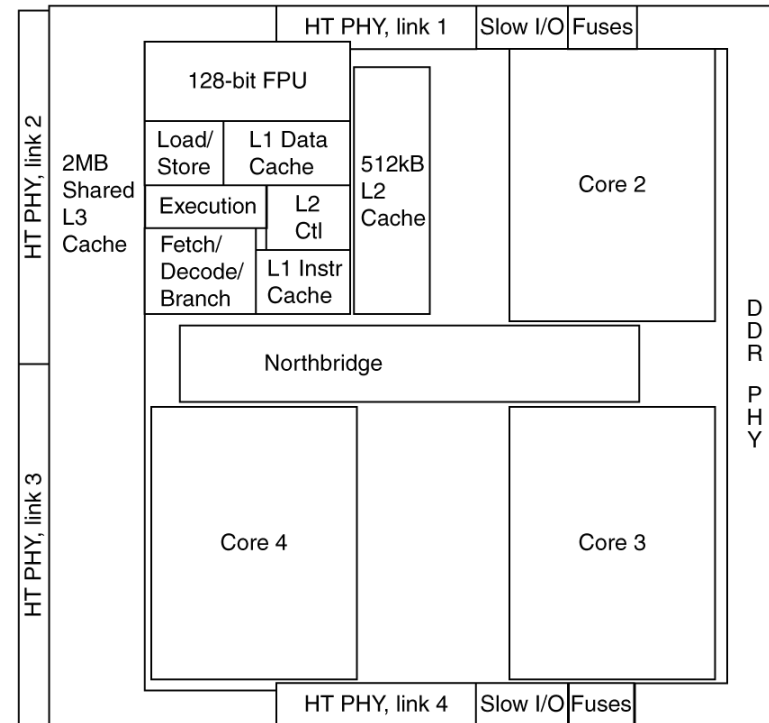
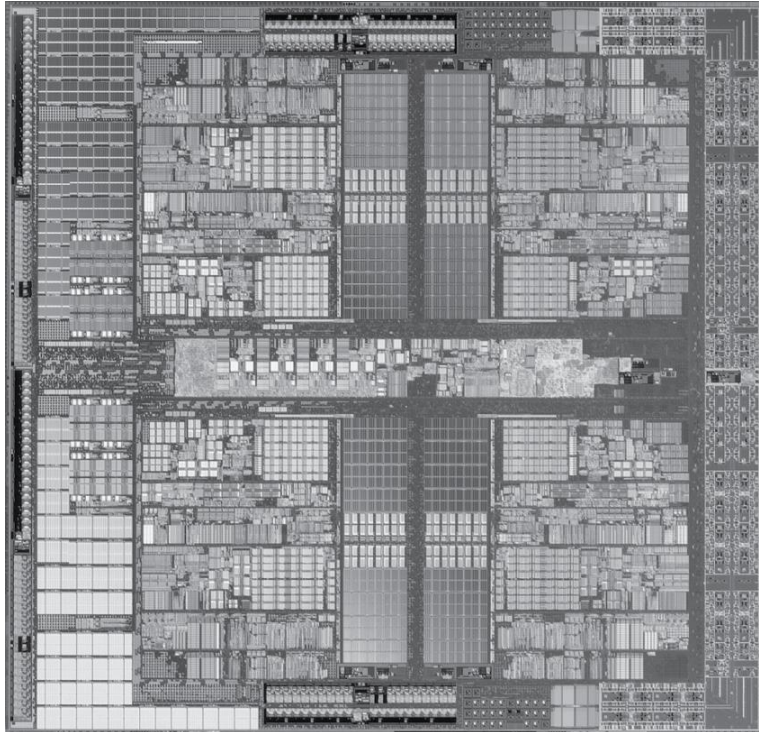


Figure 2-16. The cache is logically between the CPU and main memory. Physically, there are several possible places it could be located.

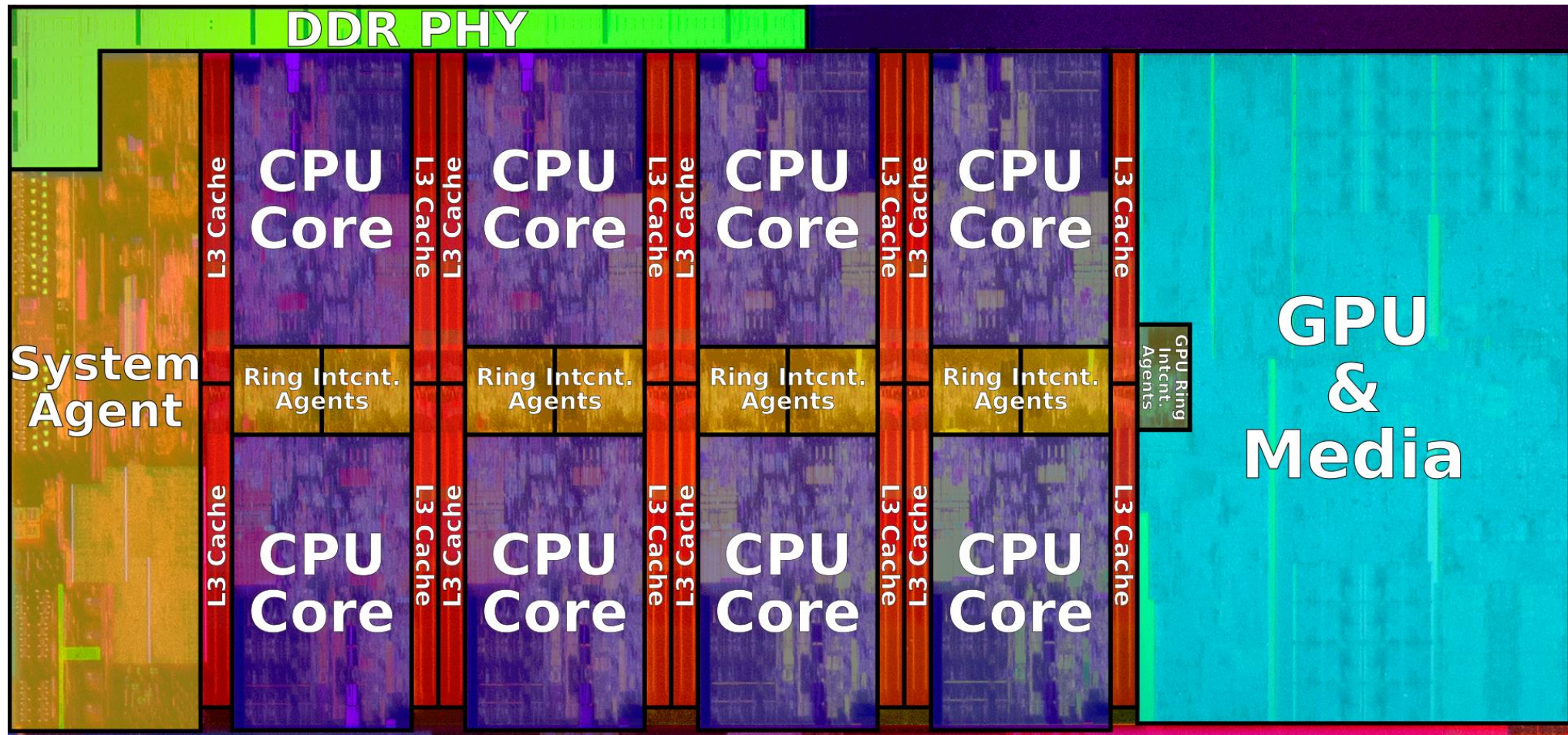
- Problem: Memory access is slower than CPU operations. Cache memory is used to speed up memory operations
- A cache is a small, fast memory positioned on the CPU, or between the CPU and the main memory
- Transparent to programmers

Inside the Processor

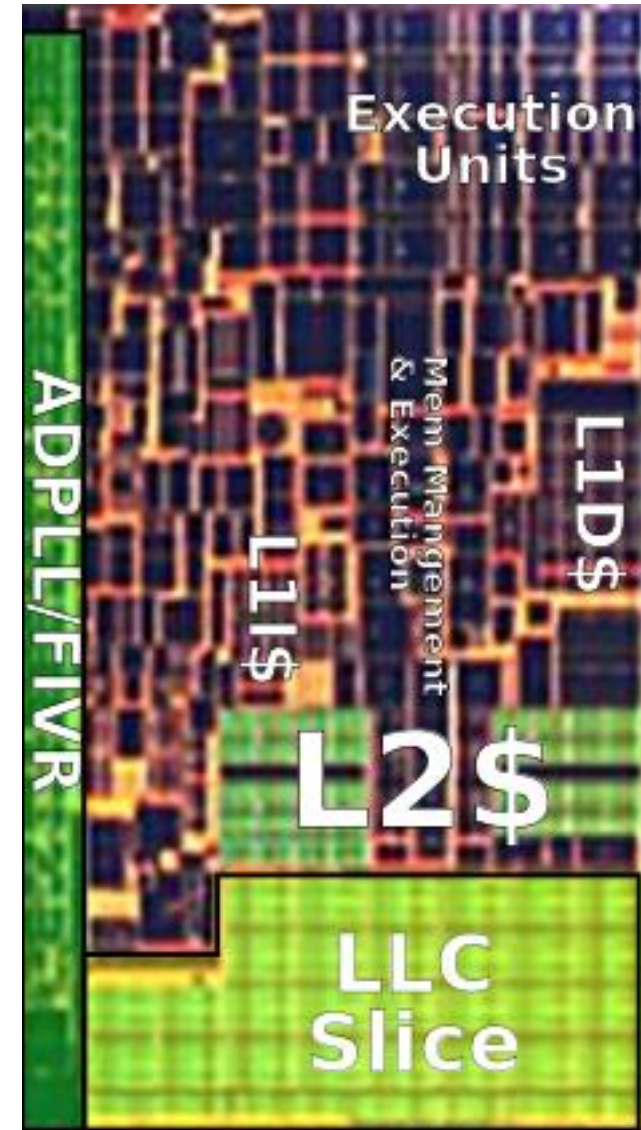
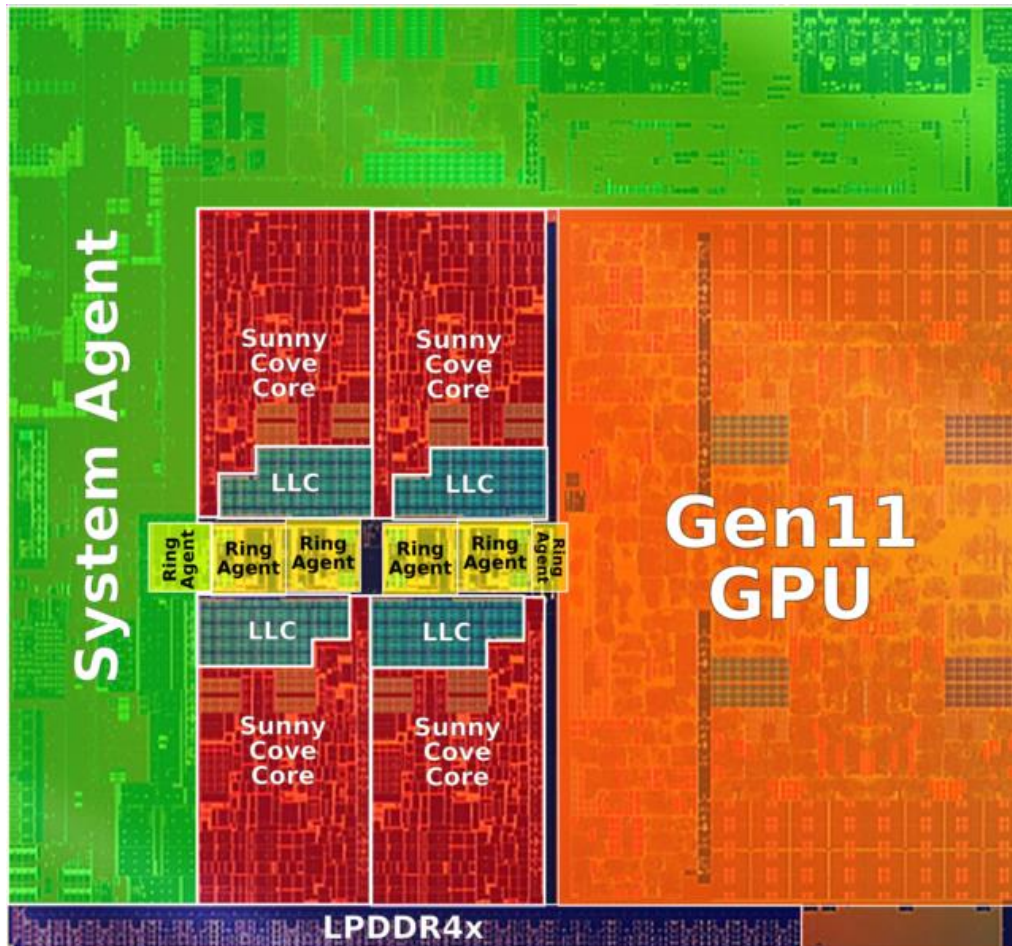
- AMD Barcelona: 4 processor cores



Inside the Intel Coffee Lake 8-core



Intel Ice Lake processor die



Central Processing Unit

- The CPU contains
 - Registers — (double) words of memory inside the CPU
 - ALU (Arithmetic and Logic Unit) — performs computations
 - Control Unit — issues control signals
- Its job is to execute (i.e., run) machine language programs, one instruction at a time.

How Programs Run

- A program is a sequence of machine language instructions, stored in consecutive memory locations.
- To execute programs, the CPU uses two special registers:
 - PC (program counter) — contains the memory address of the current or next instruction to be executed
 - IR (instruction register) — contains the current instruction being executed

How Programs Run

- Instructions are executed in a sequence of operations called the instruction cycle:
 - fetch ($IR \leftarrow \text{Memory}[PC]; PC \leftarrow PC + 1$)
 - decode
 - execute
- The instruction cycle is repeated indefinitely, as long as the machine is on.

Incrementing the PC gets us the next instruction because

- A. Instructions are stored in a linked list, and we are moving to the next node of the list.
- B. Instructions are simply an array of numbers in memory, we are indexing into the array.
- C. Instructions are stored in a special instruction array, and we are indexing into that array.

Questions about the CPU?

Reading

- Next lecture: Assembly Programming
 - Sections 2.1 and 2.2
- Problem Set 0 due this Friday!