CS 211: Intro to Computer Architecture

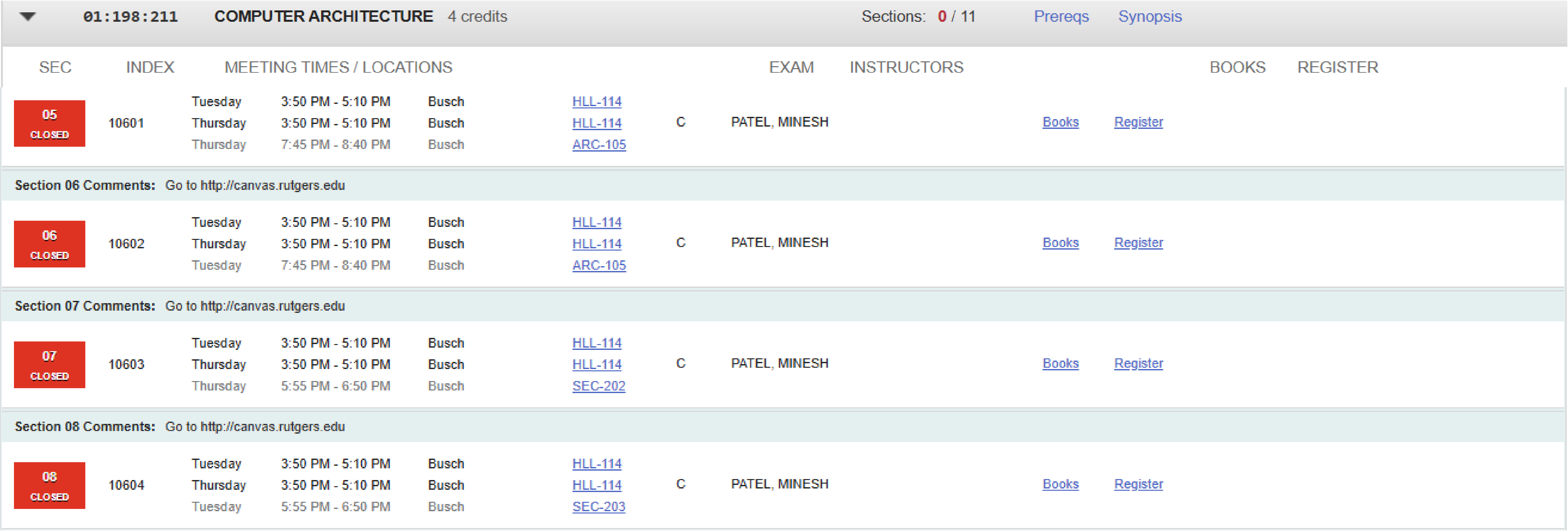
1.1: Introduction and Syllabus

Minesh Patel

Spring 2025 – Tuesday 21 January

Sanity check

- Make sure you're here for:
  - `01:198:211 COMPUTER ARCHITECTURE`
  - `Sections {05, 06, 07, 08}`



01:198:211 COMPUTER ARCHITECTURE 4 credits Sections: 0 / 11 Prereqs Synopsis

SEC	INDEX	MEETING TIMES / LOCATIONS				EXAM	INSTRUCTORS	BOOKS	REGISTER
05 CLOSED	10601	Tuesday Thursday Thursday	3:50 PM - 5:10 PM 3:50 PM - 5:10 PM 7:45 PM - 8:40 PM	Busch Busch Busch	HLL-114 HLL-114 ARC-105	C	PATEL, MINESH	Books	Register
Section 06 Comments:	Go to http://canvas.rutgers.edu								
06 CLOSED	10602	Tuesday Thursday Tuesday	3:50 PM - 5:10 PM 3:50 PM - 5:10 PM 7:45 PM - 8:40 PM	Busch Busch Busch	HLL-114 HLL-114 ARC-105	C	PATEL, MINESH	Books	Register
Section 07 Comments:	Go to http://canvas.rutgers.edu								
07 CLOSED	10603	Tuesday Thursday Thursday	3:50 PM - 5:10 PM 3:50 PM - 5:10 PM 5:55 PM - 6:50 PM	Busch Busch Busch	HLL-114 HLL-114 SEC-202	C	PATEL, MINESH	Books	Register
Section 08 Comments:	Go to http://canvas.rutgers.edu								
08 CLOSED	10604	Tuesday Thursday Tuesday	3:50 PM - 5:10 PM 3:50 PM - 5:10 PM 5:55 PM - 6:50 PM	Busch Busch Busch	HLL-114 HLL-114 SEC-203	C	PATEL, MINESH	Books	Register

CS 211 in the Core Curriculum

- Congrats on making it this far ☺

- 211 is the **only required systems course** for a CS degree
  - Programming, algorithms and data structures (`CS 111, CS 112, CS 344`)
  - Discrete Math (`CS 205, CS 206`)
  - **Systems (`CS 211`)**

- You'll get a basic idea of **how computers run code**
- I hope you will be inspired to take **more systems courses**
  - **CS 214:** Systems Programming
  - **CS 411:** Computer Architecture
  - **CS 415:** Compilers
  - **CS 416:** Operating Systems Design
  - **CS 417:** Distributed Systems
  - **CS 419:** Computer Security
  - ... and more

Agenda

- **Part 1:** CS 211 in a nutshell

- **Part 2:** Boring (but important) logistics

CS 211: Intro to Computer Architecture

???

① ②

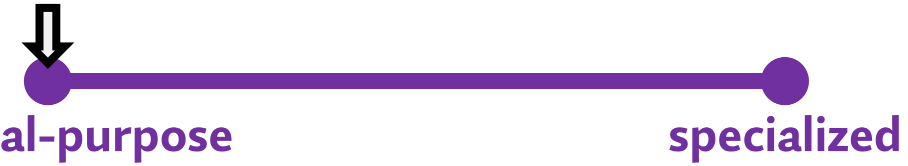
CS 211: Intro to Computer Architecture

???

Computer:

“a device that uses ***physical phenomena***
to model a ***problem being solved****”*

Examples: Phone (2022) + Laptop (2023)

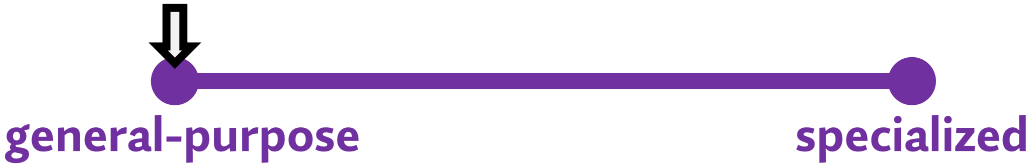


- Both use **electrons** to **perform general-purpose computations**

https://www.amazon.com/Google-Pixel-Unlocked-Smartphone-Ultrawide/dp/B09HJZPFDD

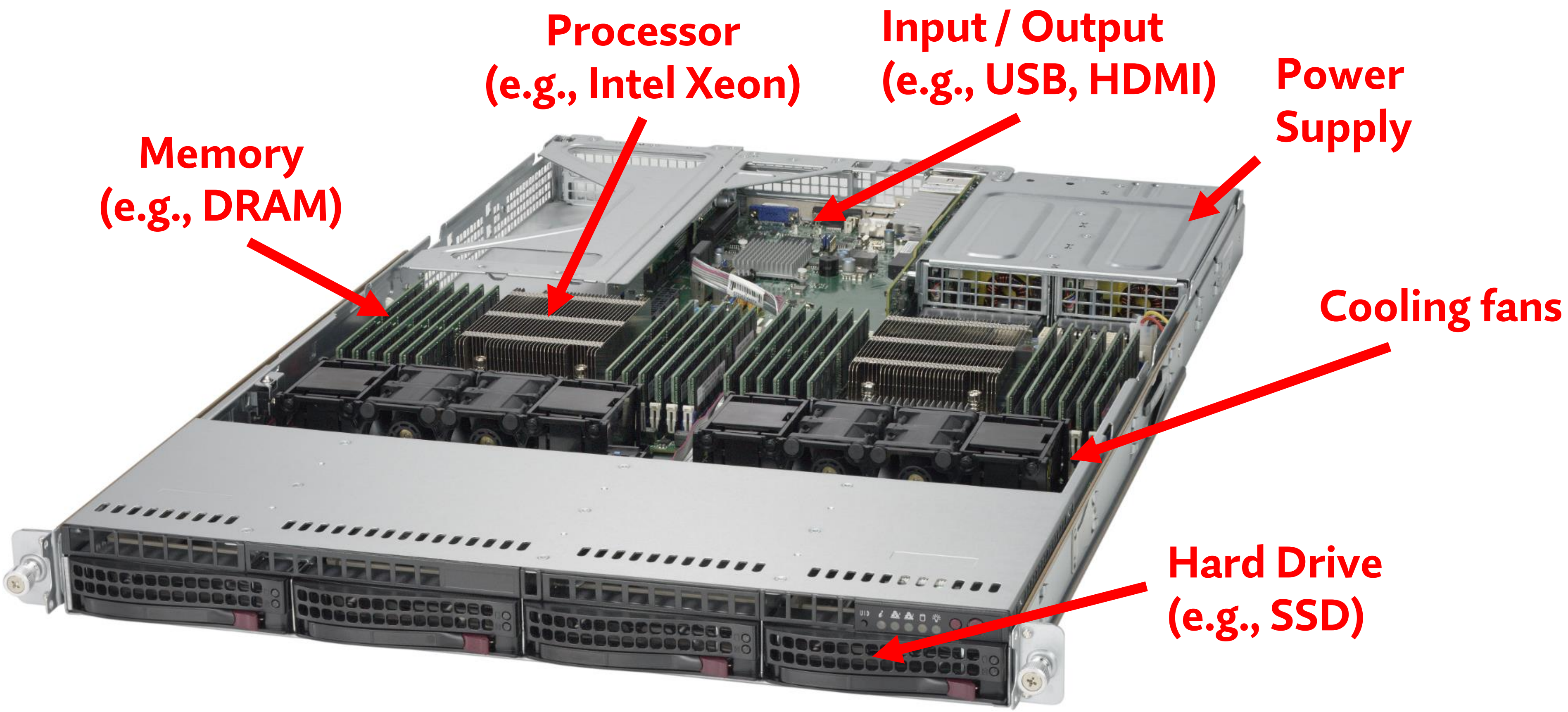
https://dlcdnwebimgs.asus.com/gain/98e77ab8-b905-4e3a-87bf-72adaed96c10/

Examples: Typical Server or PC


general-purpose

specialized

- Uses **electrons** to **perform general-purpose computations**


Processor (e.g., Intel Xeon)

Input / Output (e.g., USB, HDMI)

Power Supply

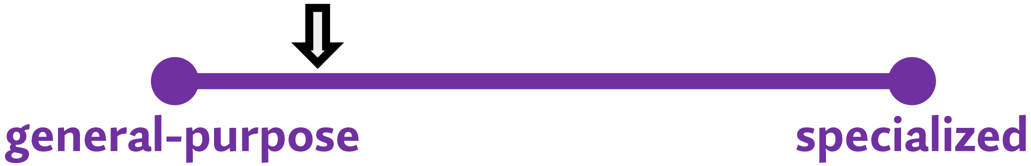
Memory (e.g., DRAM)

Cooling fans

Hard Drive (e.g., SSD)

https://www.ebay.com/itm/154513065385?epid=1252327726&hash=item23f9b245a9:g:GoIAAOSwi7VcJULN

Example: PS5 Pro (2024)



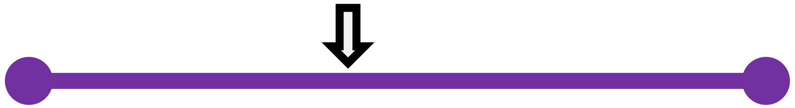
general-purpose

specialized

- Uses **electrons** to **perform graphics-focused computations**

https://static1.srcdn.com/wordpress/wp-content/uploads/2020/10/PlayStation-5-Teardown-All-Internal-Hardware-Components-Video.jpg

Example: Warehouse-Scale Computer (Google, NL)



general-purpose

specialized

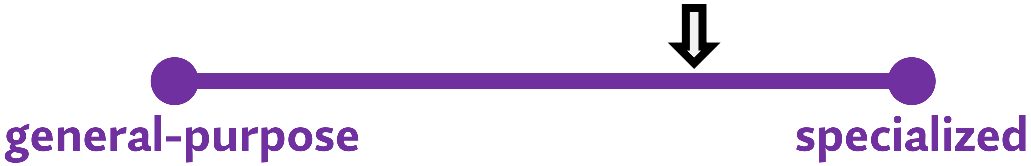
- Uses **electrons** to **perform cloud-scale computations**

Cooling Towers

Server Rooms

Raganathan+, "Twenty Five Years of Warehouse-Scale Computing," IEEE Micro, 2024.

Example: Azure's AI Accelerator (Maia 100, 2023)



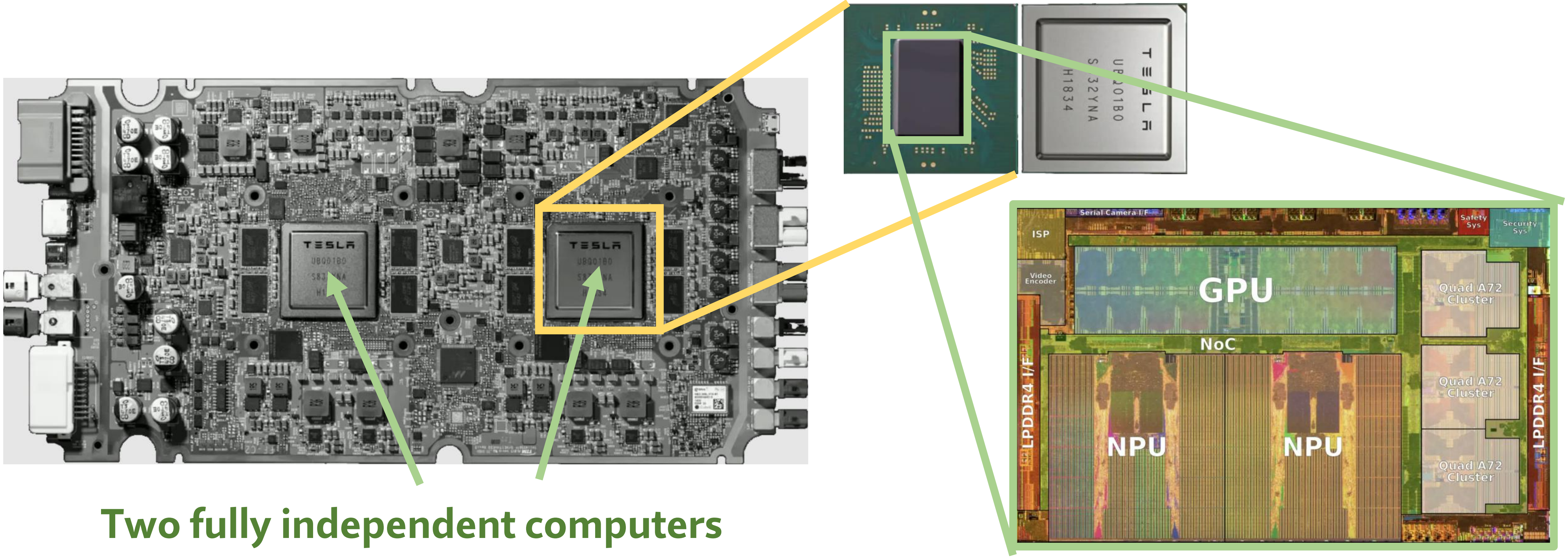
- Both use **electrons** to **perform highly-efficient AI computations**

https://www.tweaktown.com/news/100264/microsoft-lifts-the-lid-on-its-new-ai-chip-maia-100-up-to-700w-tdp-built-for-large-scale/index.html

Examples: Tesla FSD (2019)

general-purpose specialized

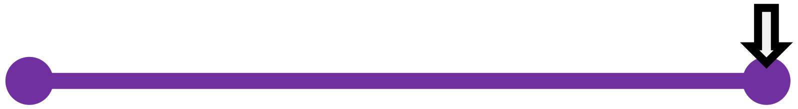
- Both use **electrons** to **perform reliable computations**



Two fully independent computers

https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip

Example: Mechanical Computers



general-purpose

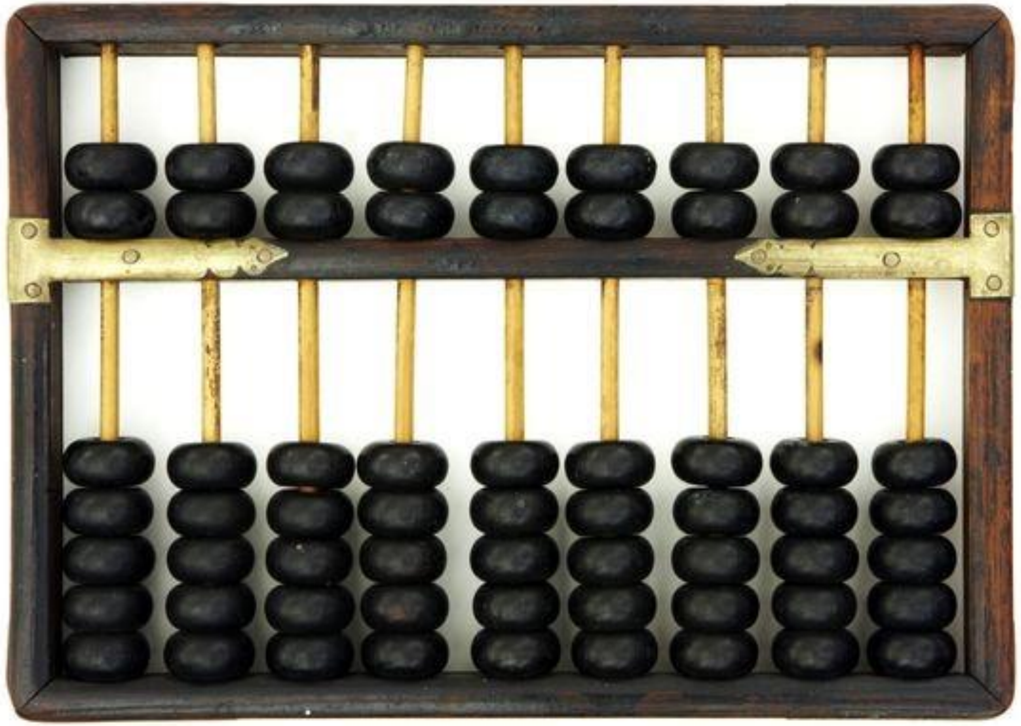
specialized

- Uses **springs/gears** to **track the time**

Pocket Watch (1920)

https://en.wikipedia.org/wiki/Pocket_watch#/media/File:Rel%C3%B3gio_de_Bolso_(watch-pocket).JPG

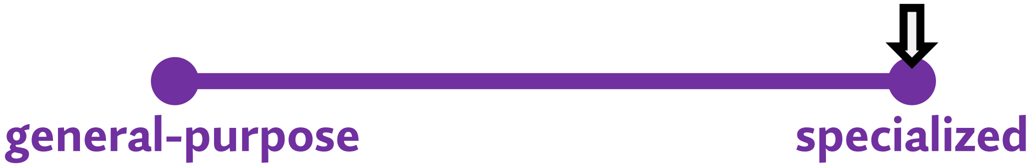
- Uses **movement** for **counting**



Abacus (~2700 BC)

https://images.computerhistory.org/revonline/images/xb93.80p-03-01.jpg?w=600

Example: Analog Computers


general-purpose

specialized

- Uses **fluid pressure** to **model diffeq's**

The Water Integrator (1936)

https://habr.com/ru/articles/228283/

- Uses **gears** to **predict eclipses**

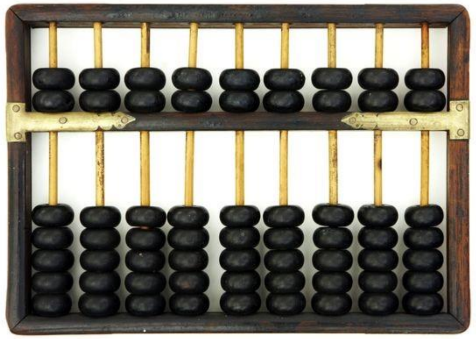
Reproduction (2007) of Antikythera (~2000 BC)

https://en.wikipedia.org/wiki/Antikythera_mechanism#/media/File:Antikythera_model_front_panel_Mogi_Vicentini_2007.JPG

Computers, More Generally

Problem Being Studied

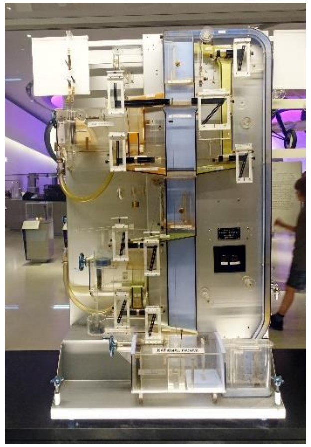
Arithmetic Operations



Abacus (~2700 BC)

Mechanics

Economic Simulation



MONIAC (1949)

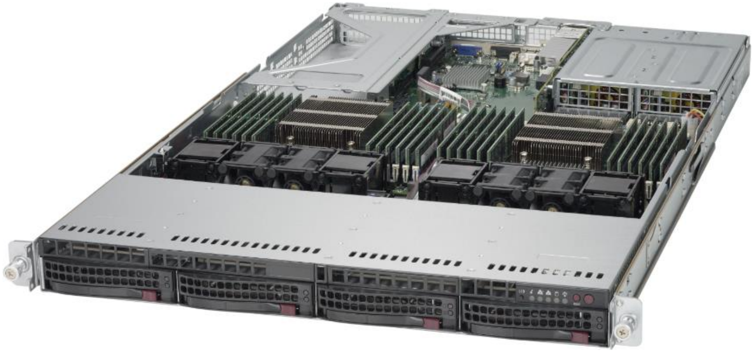
Fluidics

Differential Equations

AKAT-1 (1959)

Analog Electronics

General Purpose Computation



Server Blade (2020)

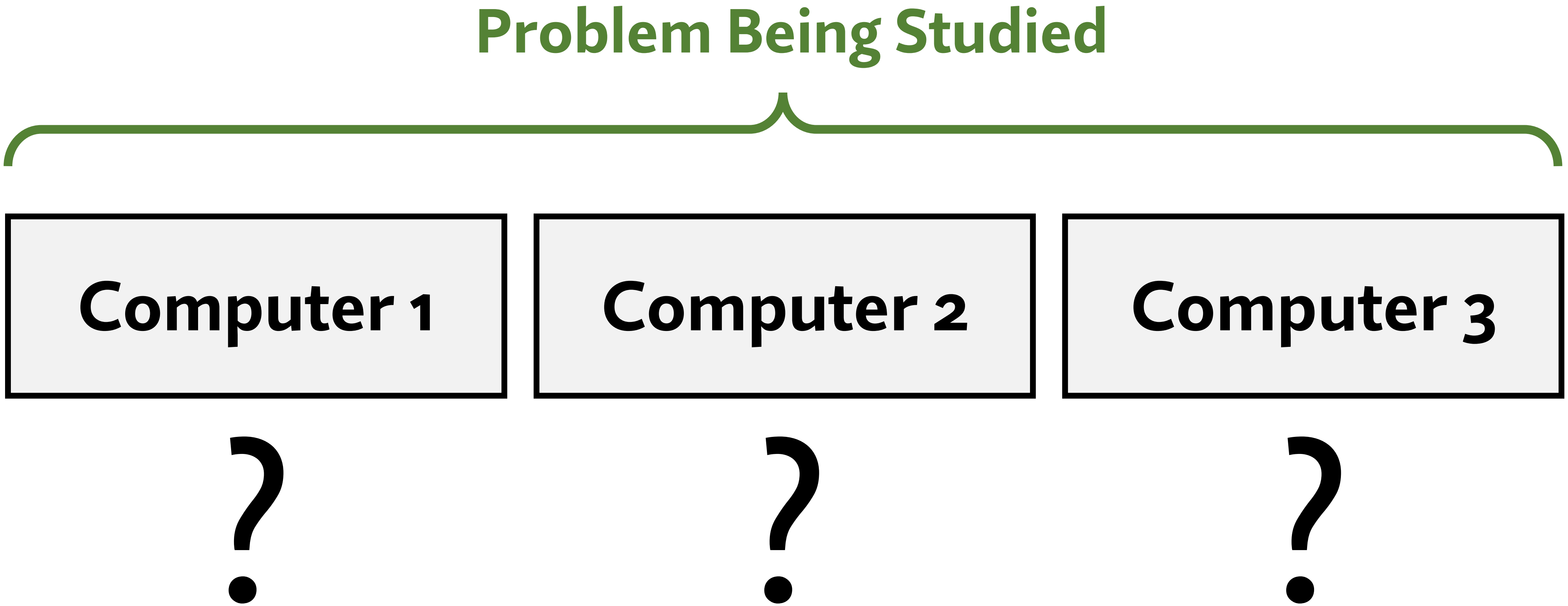
Digital Electronics

Physical Phenomena

https://images.computerhistory.org/revonline/images/xb93.80p-03-01.jpg?w=600

https://www.britannica.com/technology/analog-computer#/media/1/22416/127313

Which Computer, Then?



Problem Being Studied

Computer 1

Computer 2

Computer 3

?

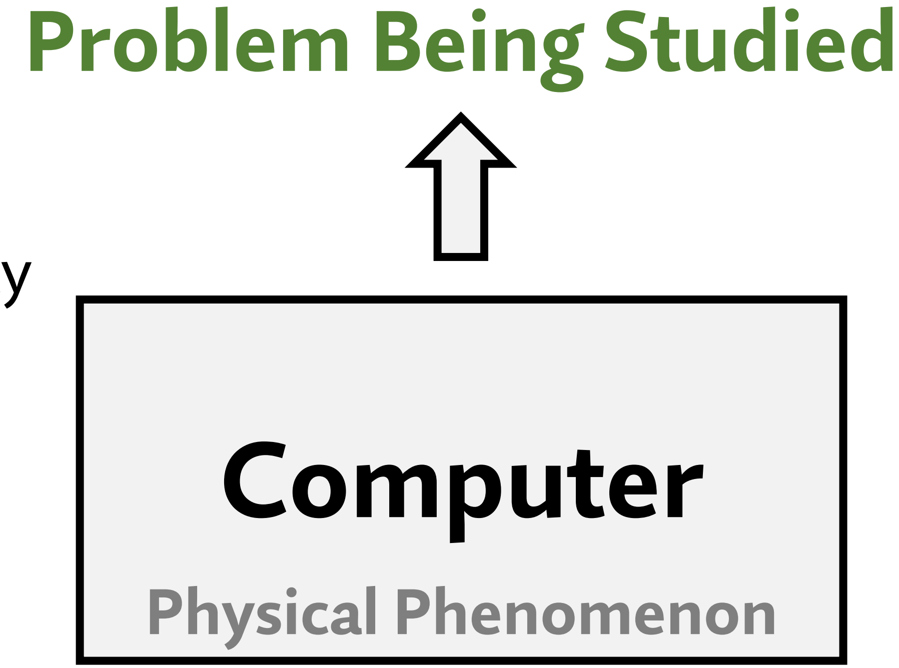
?

?

- Choose based on **problem requirements** and **computer tradeoffs**

Computers, More Generally

- Every computer provides different **tradeoffs**
  - Cost
  - Performance
  - Reliability
  - Security
  - Battery life
  - Heat
  - Inputs/outputs
  - Weight
  - Sustainability
  - ...
- A computer is **a tool** for addressing a problem



Problem Being Studied

Computer

Physical Phenomenon

CS 211 Goal #1:

Know your tools

(and their tradeoffs)

① ②

CS 211: Intro to Computer Architecture

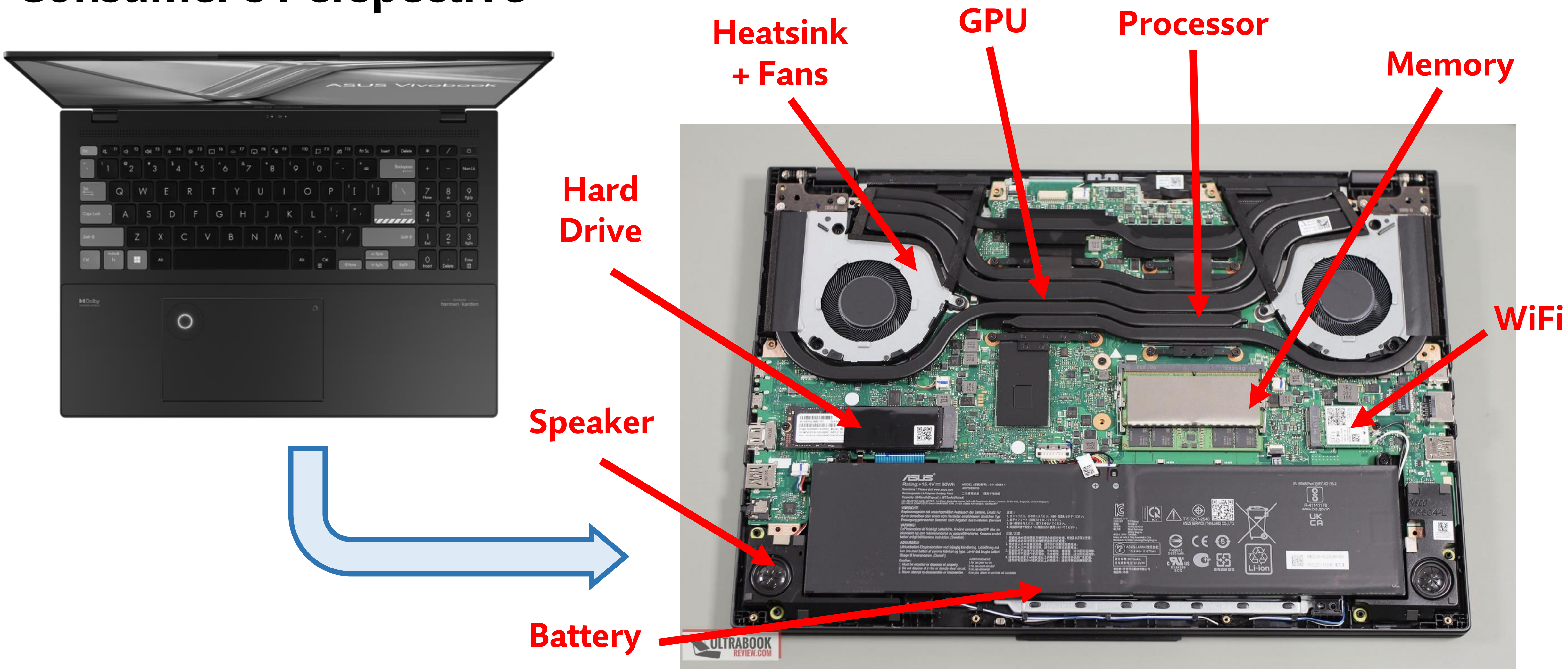
???

Architecture:

“the complex or carefully designed ***structure of [a computer]****”*

Example: My Laptop (ASUS K6604)

Consumer's Perspective



Heatsink + Fans

GPU

Processor

Memory

Hard Drive

WiFi

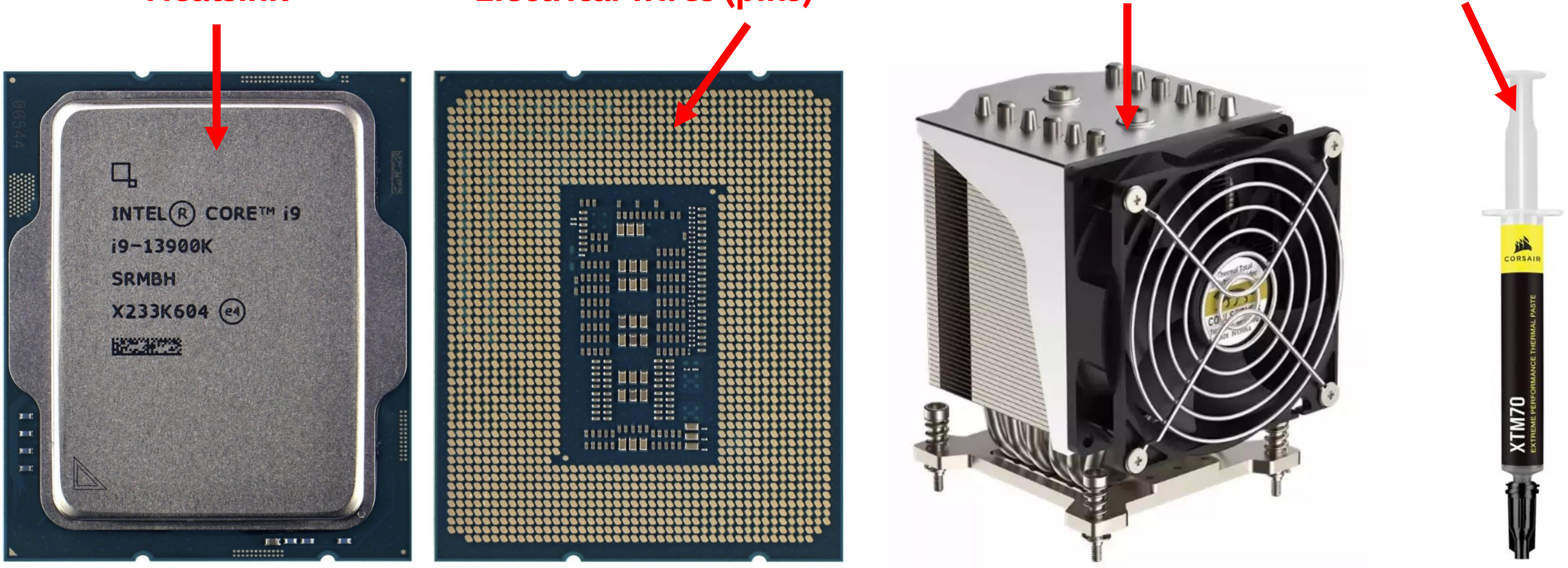
Speaker

Battery

https://www.ultrabookreview.com/64501-asus-vivobook-pro-16x-oled-review/

Example: My Laptop’s CPU (Intel i9-13xxx)

CPU Integrator’s Perspective



Heatsink

Electrical wires (pins)

Heatsink + fan

Thermal paste

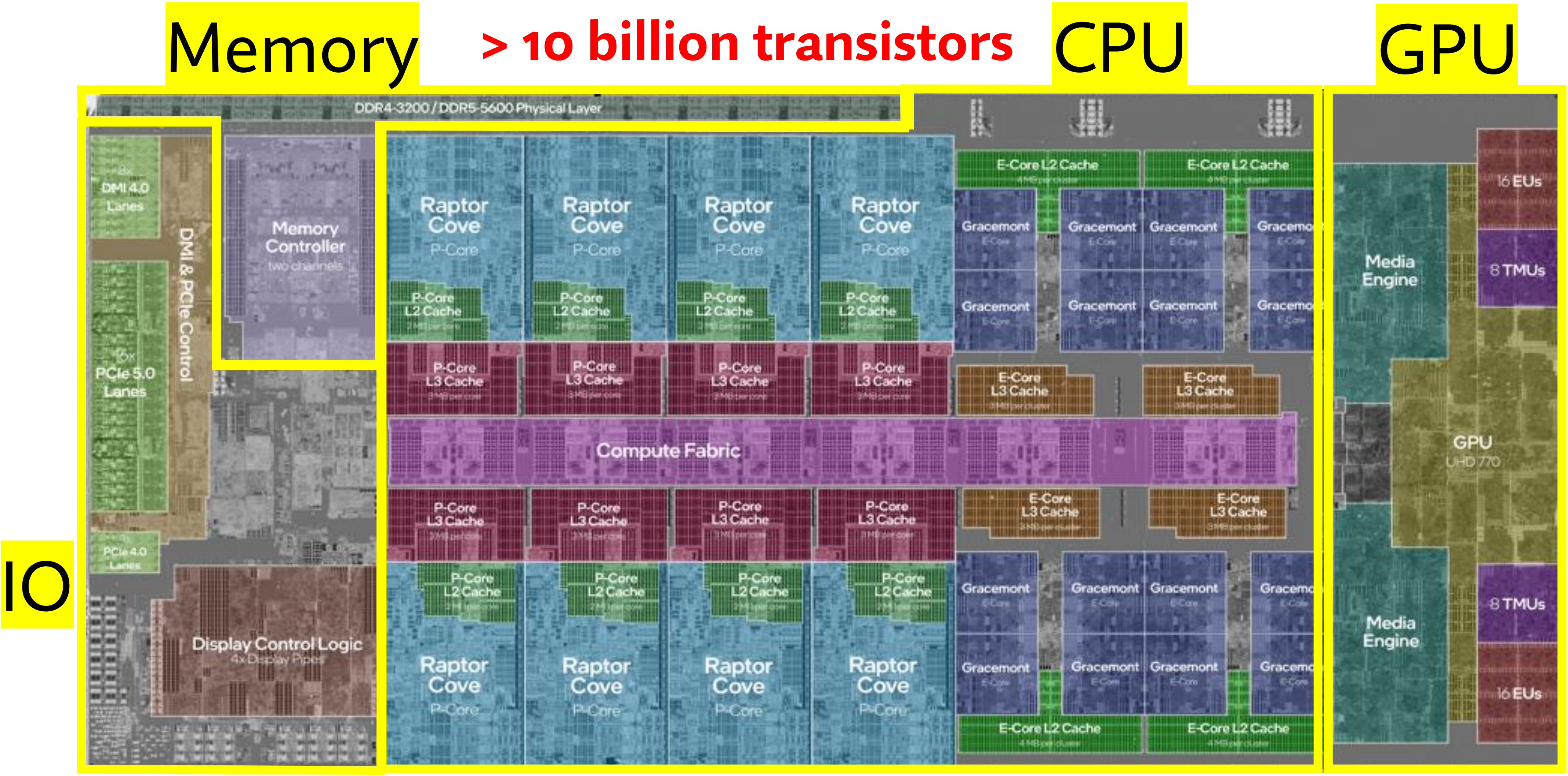
https://www.hwcooling.net/en/intel-core-i9-13900k-review-biggest-predator-in-the-lake/

https://www.ebay.com/itm/166275134036

https://pisces.bbystatic.com/image2/BestBuy_US/images/products/6518/6518226_sd.jpg;maxHeight=640;maxWidth=550;format=webp

Example: My Laptop’s CPU (Intel i9-13xxx)

CPU Manufacturer’s Perspective



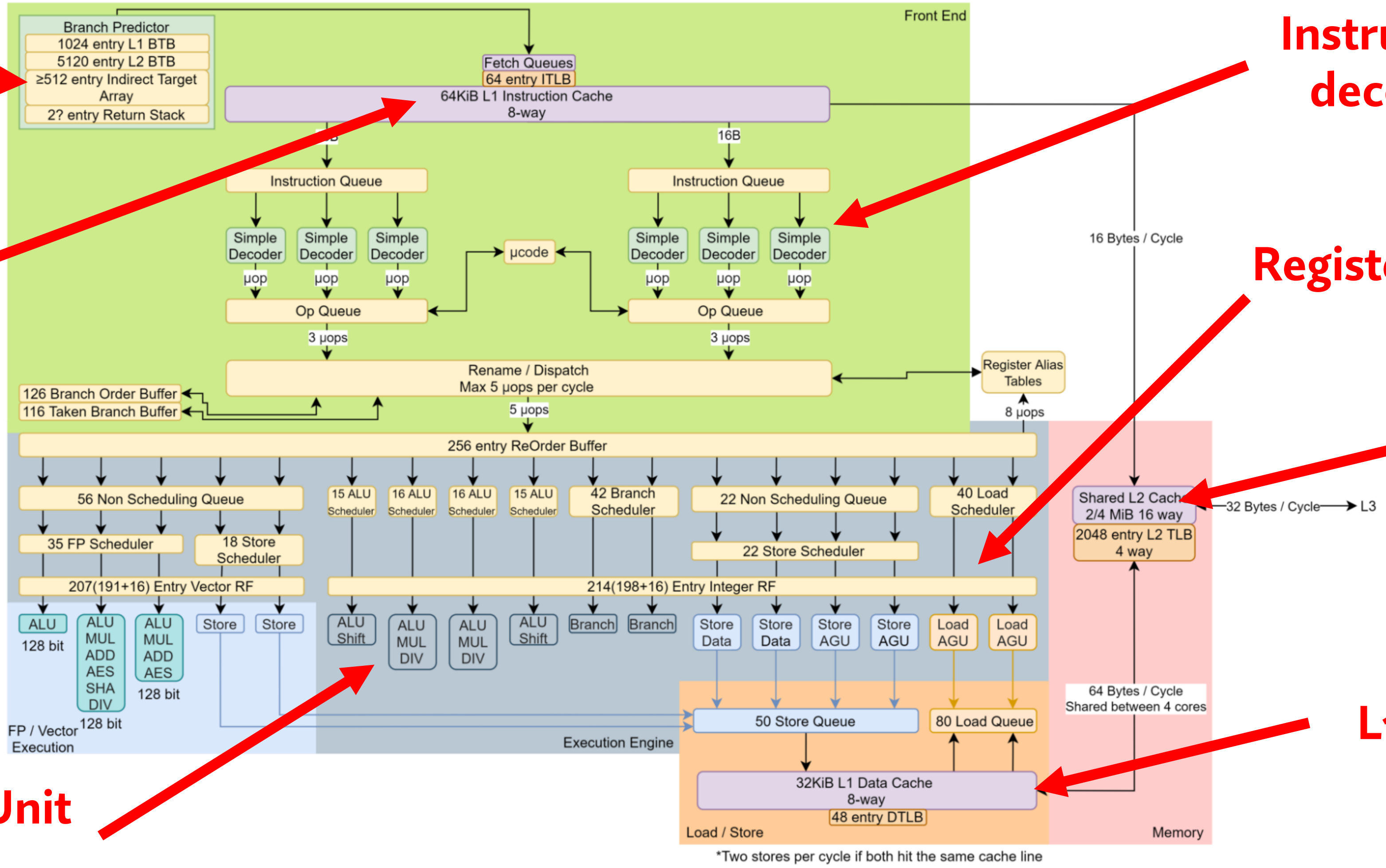
> 10 billion transistors

https://upload.wikimedia.org/wikipedia/commons/a/a4/Intel_Core_i9-13900K_Labelled_Die_Shot.jpg

Example: My Laptop’s CPU (Intel i9-13xxx)

CPU Designer’s Perspective

Gracemont Core Block Diagram



*Two stores per cycle if both hit the same cache line

Branch Predictor

Instruction decoder

L1 Instruction Cache

Registers

L2 Cache

L1 Data Cache

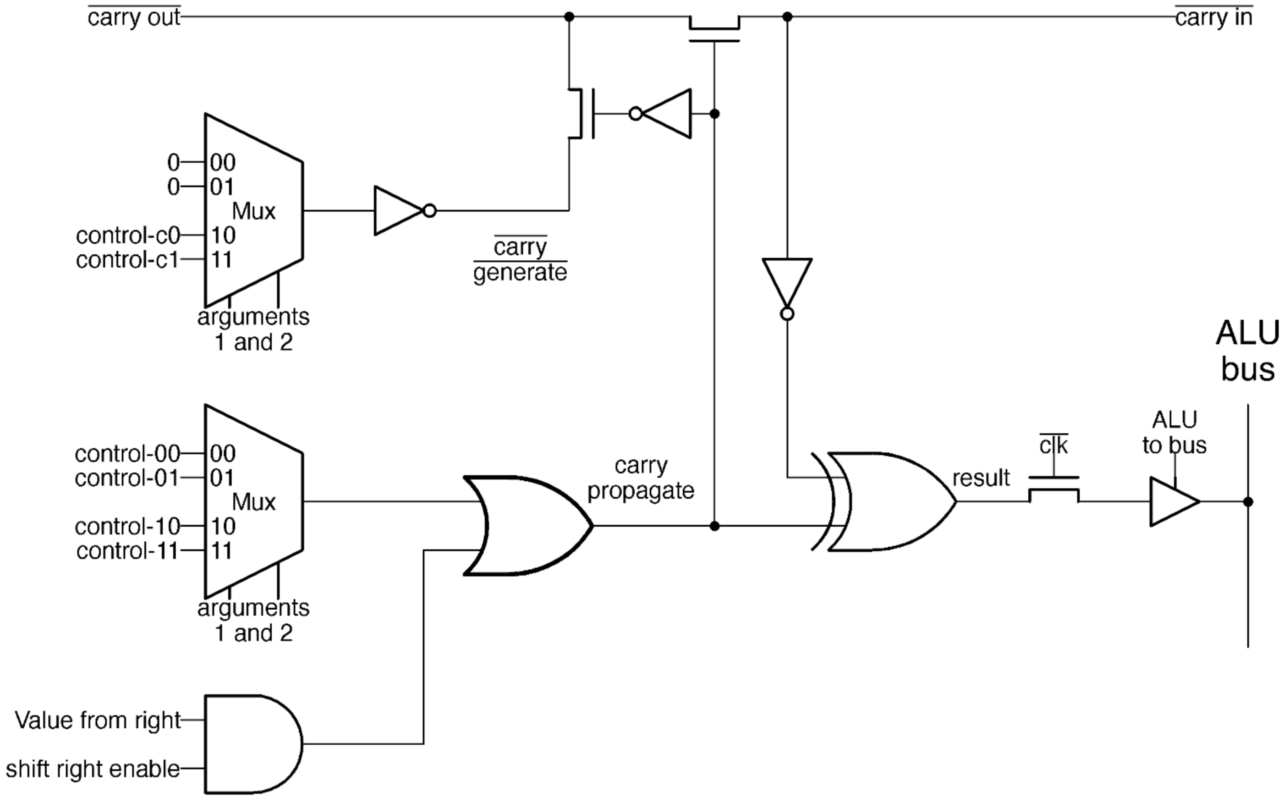
Arithmetic + Logic Unit (ALU)

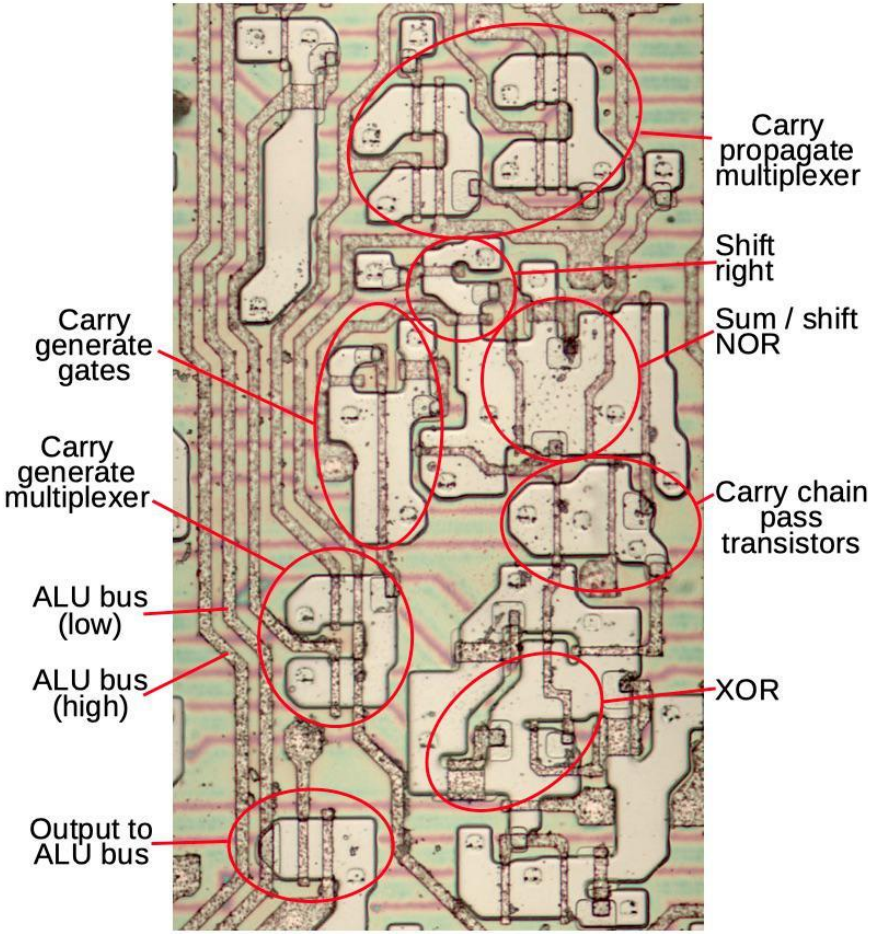
https://www.techpowerup.com/forums/attachments/gracemontrevised-png.280665/

Example: Add Operations in the Intel 8086 (1978)

Logic Designer's Perspective

Logic circuit for an adder

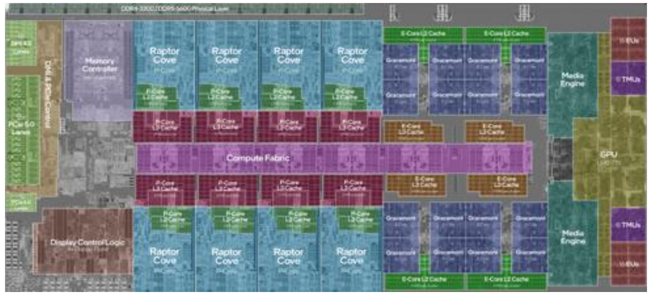




https://www.righto.com/2020/08/reverse-engineering-8086s.html

Example: My Laptop’s CPU (Intel i9-13xxx)

Everyone has a different perspective based on their **level of abstraction**

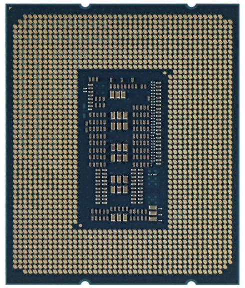


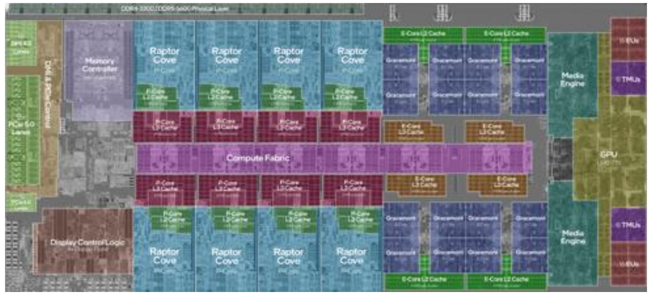
Computing Abstractions

- Computer scientists and engineers are **masters of abstraction**

Software

Hardware



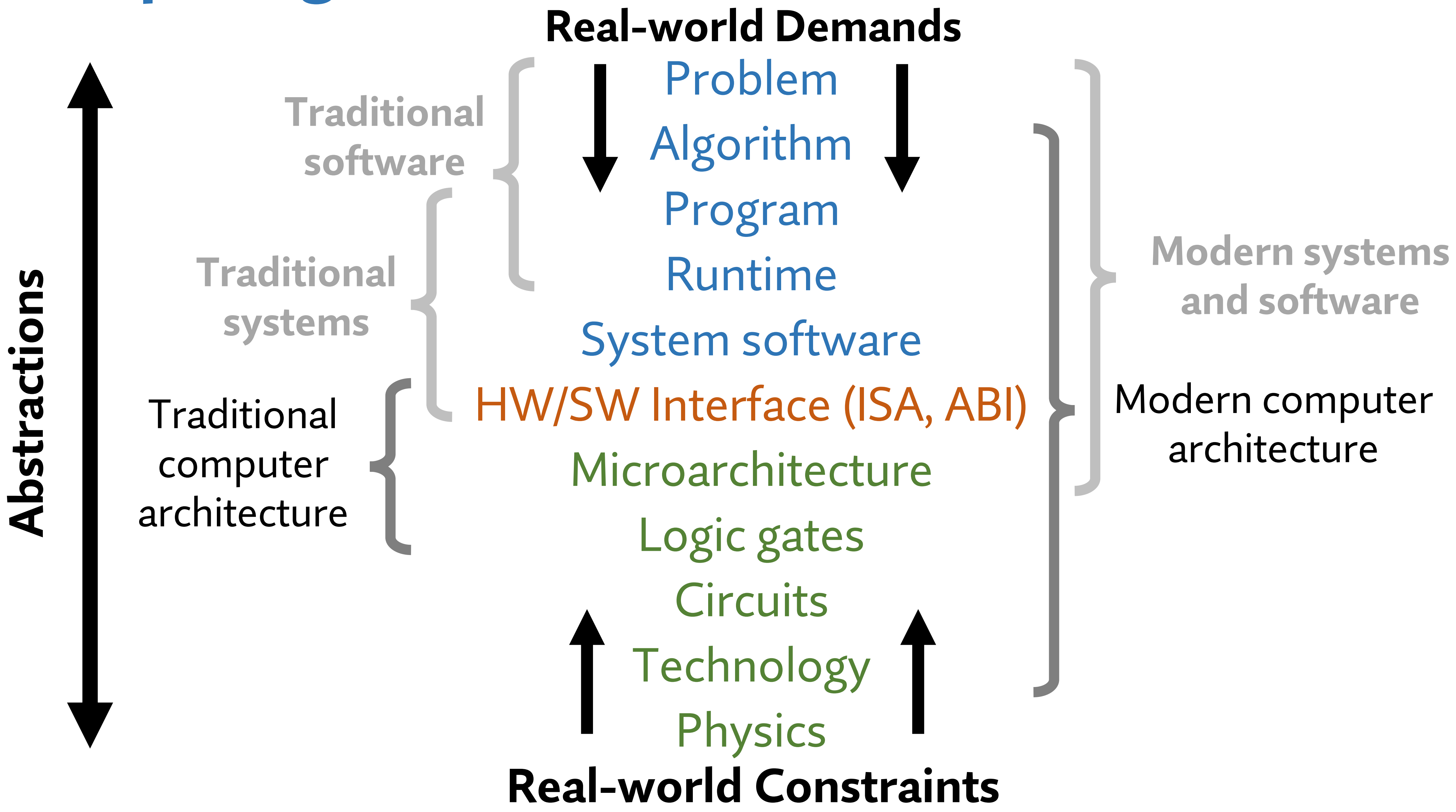


Problem
Algorithm
Program
Runtime (OS)

Microarchitecture
Logic Circuits
Physics

ISA (Architecture)

Computing Abstractions: Traditional vs. Modern



Real-world Demands

Problem

Algorithm

Program

Runtime

System software

HW/SW Interface (ISA, ABI)

Microarchitecture

Logic gates

Circuits

Technology

Physics

Real-world Constraints

Abstractions

Traditional software

Traditional systems

Traditional computer architecture

Modern systems and software

Modern computer architecture

Abstraction is Good, Until It's Not

- **All abstractions have limits**
  - Especially with performance, security, bugs, failures

Code 1

```
void copyij(int src[2048][2048], int dst[2048][2048])
{
     int i,j;
     for(i = 0; i < 2048; i++)
          for (j = 0; j < 2048; j++)
               dst[i][j] = src[i][j];
}
```

Runtime: 5.2 ms

Code 2

```
void copyji(int src[2048][2048], int dst[2048][2048])
{
     int i,j;
     for (j = 0; j < 2048; j++)
          for (i = 0; i < 2048; i++)
               dst[i][j] = src[i][j];
}
```

Runtime: 162 ms

>30x slowdown

Reason: nonuniform memory access times in the hardware

Computer Systems: A Programmer's Perspective (2e)
(Evaluations on a 2.7 GHz Intel Core i7)

Abstraction is Good, Until It's Not

- **All abstractions have limits**
  - Especially with performance, security, bugs, failures
- CS 211: **understand the abstractions** so you can:
  - Become more effective programmers
  - Jump to later systems courses

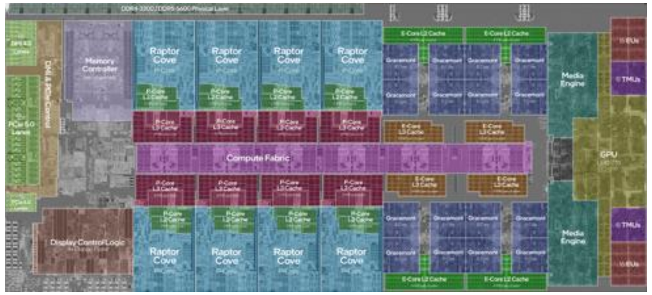
CS 211 Goal #2:

Understand computing abstractions
(and know when to break them)

Focus of CS 211

- Our focus will be on the **hardware-software interface (ISA)**

Software ⇐ ⇒ Hardware



Problem
Algorithm
Program
Runtime (OS)

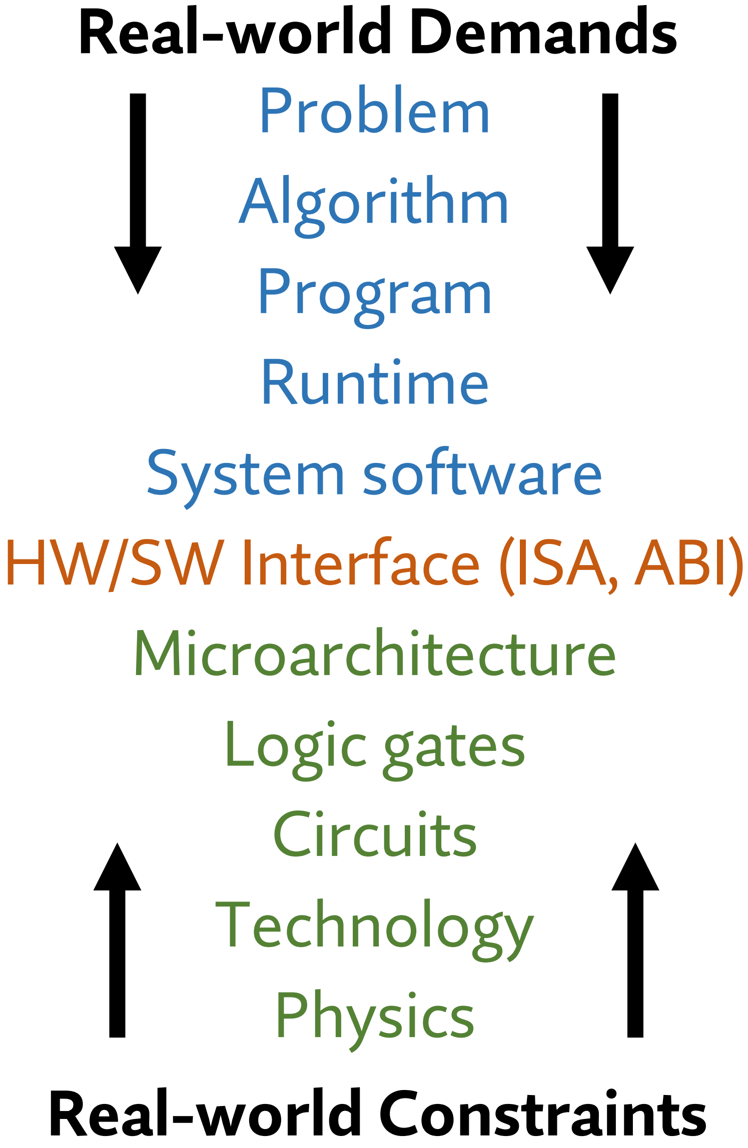
Microarchitecture
Logic Circuits
Physics

ISA (Architecture)

Breaking Computing Abstractions

- We will learn **new tools** to explore the HW-SW interface
  - C programming
  - Assembly language
  - Machine code
  - Digital logic

- These tools will give us a look **under the hood**
- CS 211 is **all about the fundamentals**
  - Maybe you won’t touch these tools again
  - Maybe you will find a job using them
  - Maybe you will take them to the next level, or go even deeper ☺


Real-world Demands

Problem

Algorithm

Program

Runtime

System software

HW/SW Interface (ISA, ABI)

Microarchitecture

Logic gates

Circuits

Technology

Physics

Real-world Constraints

Goals of CS 211

Goal #1:

Know your tools
(and their tradeoffs)

Goal #2:

Understand computing abstractions
(and know when to break them)

Goal #3:

Gain a holistic view of the system

CS 211: Intro to Computer Architecture

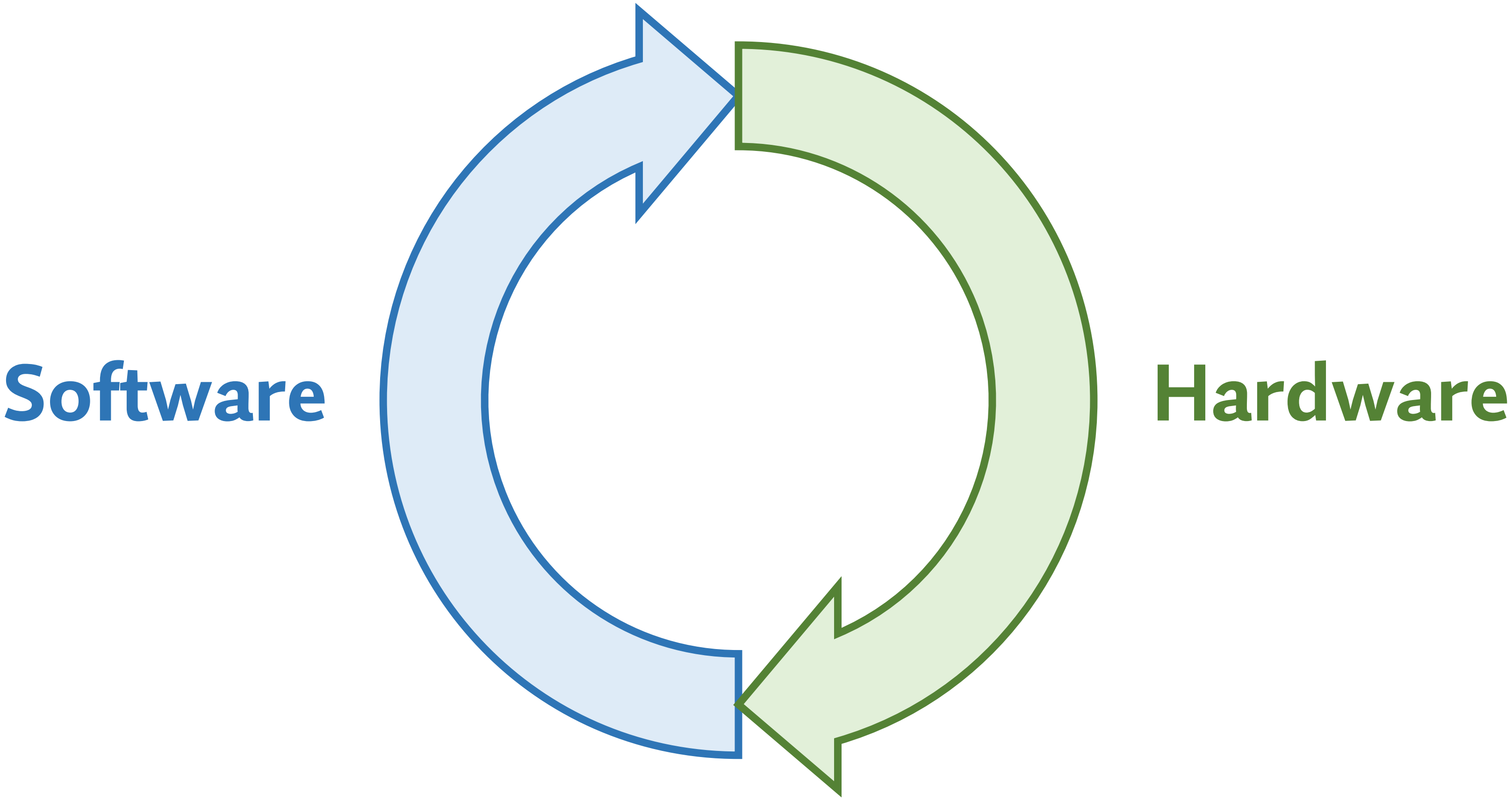
Computer Architecture:

*“**the complex structure of***
*a device that uses **physical phenomena***
*to model a **problem being solved**”*

↓

*The **science** and **art** of designing*
computing systems** to **solve a problem

Why Study Computer Architecture?



Software

Hardware

CS 211: Intro to Computer Architecture

- Unfortunately, CS 211 won’t teach you how to build a computer
  - Need a few more classes for that ☺

- Instead, we’ll scratch the surface to prepare you for more

Agenda

- **Part 1:** CS 211 in a nutshell

- **Part 2:** Boring (but important) logistics

Instructional Staff

Course Instructor

Minesh Patel

Teaching Assistants

Ramesh Balaji

Nate Blum

Neha Jeyaram

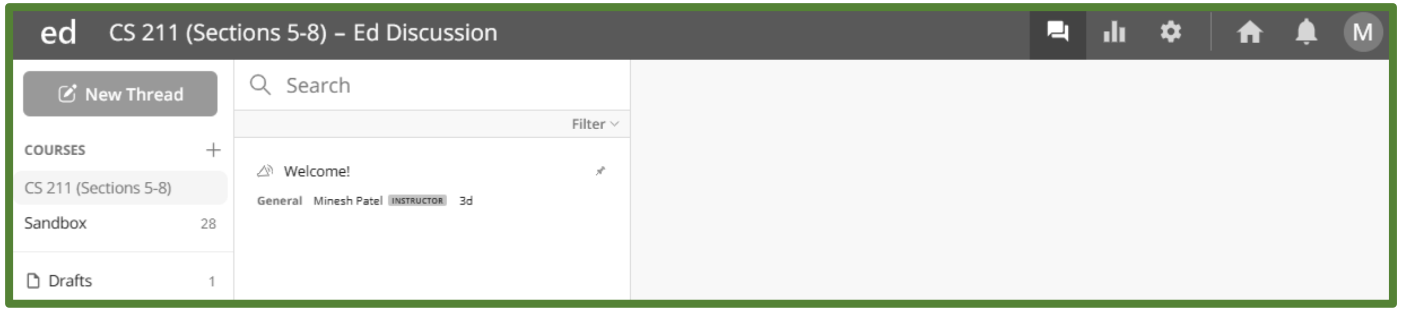
Jerlin Yuen

+1
(maybe?)

???

Contacting the Instructional Staff

1 Post on Ed (preferred)



CS 211 (Sections 5-8) – Ed Discussion

Welcome!
General Minesh Patel INSTRUCTOR 3d

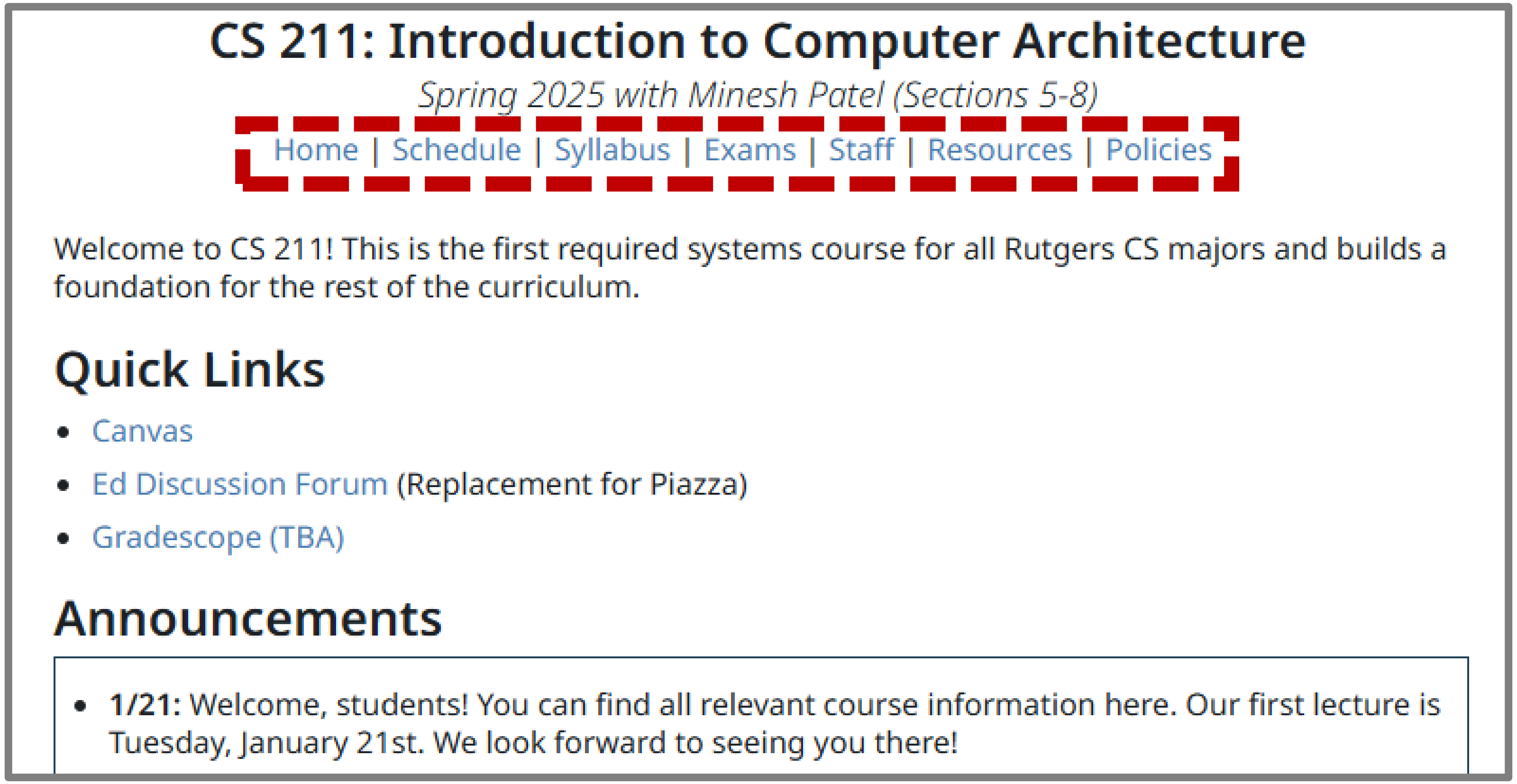
We will enroll you in Ed manually very soon

2 E-mail the mailing list
`cs211_s25_5678@email.rutgers.edu`

Please do not send general questions to individual staff members (we will likely redirect you to Ed)

Course Website

- Link is on Canvas (https://cs.rutgers.edu/~mp2099/courses/cs211-s25/index.html)
  - Syllabus, schedule, materials, etc. will be posted here



CS 211: Introduction to Computer Architecture

Spring 2025 with Minesh Patel (Sections 5-8)

Home | Schedule | Syllabus | Exams | Staff | Resources | Policies

Welcome to CS 211! This is the first required systems course for all Rutgers CS majors and builds a foundation for the rest of the curriculum.

Quick Links

- Canvas
- Ed Discussion Forum (Replacement for Piazza)
- Gradescope (TBA)

Announcements

- **1/21:** Welcome, students! You can find all relevant course information here. Our first lecture is Tuesday, January 21st. We look forward to seeing you there!

- Anybody having trouble with this? **Let us know.**

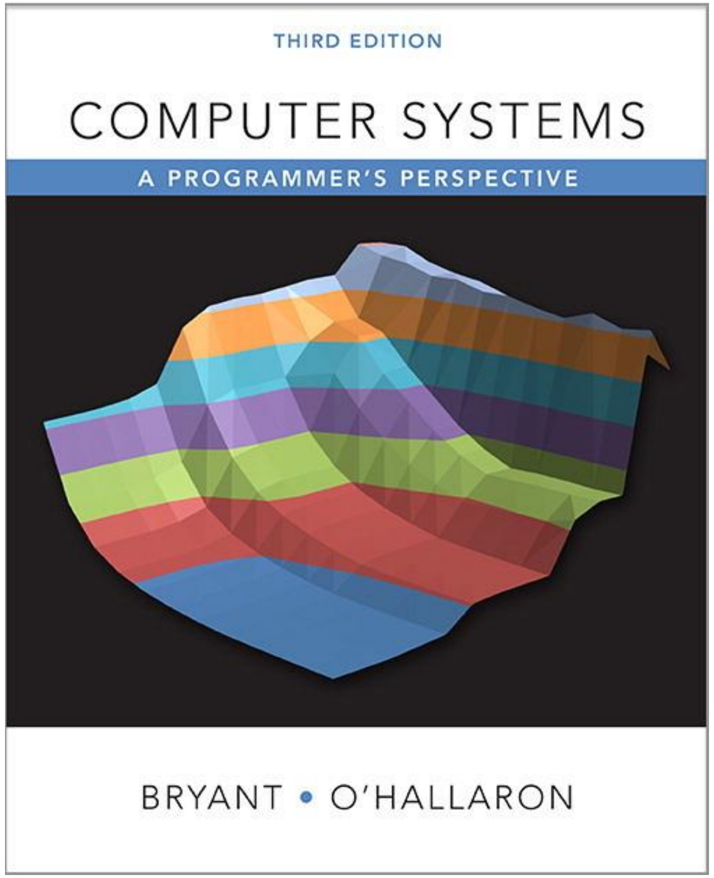
Logistics

- **Lectures:** T/Th 3:50 PM - 5:10 PM (Busch HLL-114)
- **TA Recitation**:
  - **#5**: Thursday 7:45 PM - 8:40 PM (Busch ARC-105)
  - **#6**: Tuesday 7:45 PM - 8:40 PM (Busch ARC-105)
  - **#7**: Thursday 5:55 PM - 6:50 PM (Busch SEC-202)
  - **#8**: Tuesday 5:55 PM - 6:50 PM (Busch SEC-203)
- **Office Hours:**
  - In-person TBA

- Attendance will NOT be taken: lecture and recitation are for your benefit
  - Lectures slides will be posted on Canvas and/or the website (best effort)
  - You may attend whichever and however many recitations you like

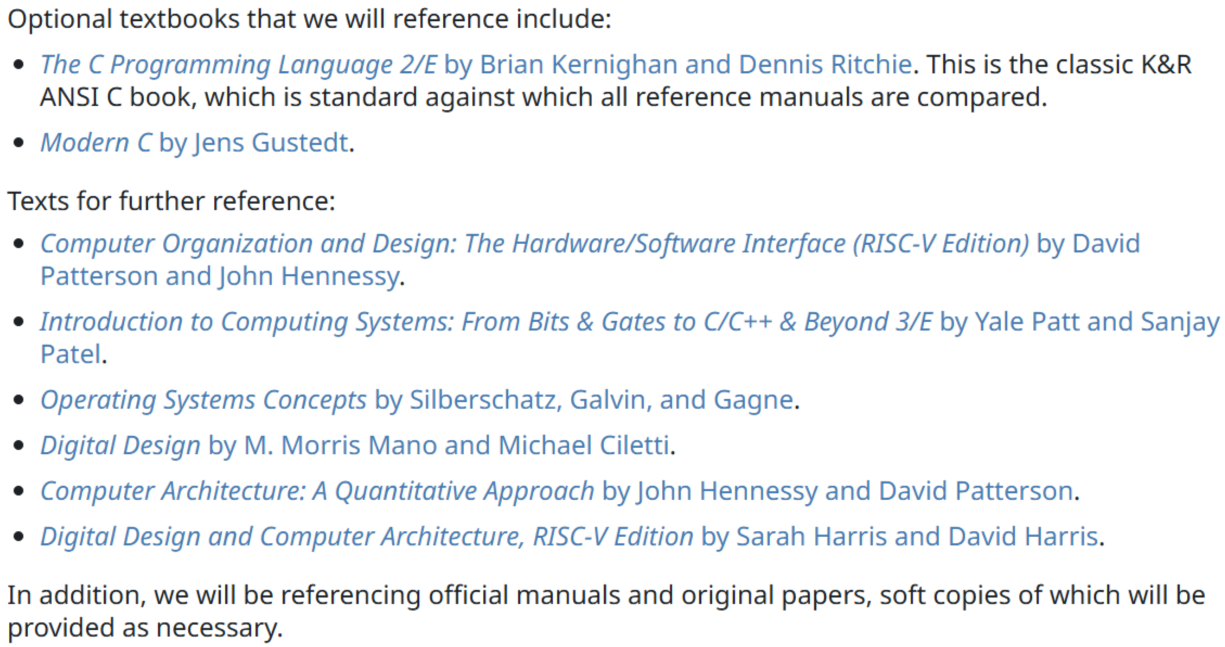
Textbook

- Technically, there is no textbook for the course
- We will roughly follow concepts in CS:APP 3e



https://csapp.cs.cmu.edu/3e/home.html

Other good reference textbooks



Optional textbooks that we will reference include:

- *The C Programming Language 2/E* by Brian Kernighan and Dennis Ritchie. This is the classic K&R ANSI C book, which is standard against which all reference manuals are compared.
- *Modern C* by Jens Gustedt.

Texts for further reference:

- *Computer Organization and Design: The Hardware/Software Interface (RISC-V Edition)* by David Patterson and John Hennessy.
- *Introduction to Computing Systems: From Bits & Gates to C/C++ & Beyond 3/E* by Yale Patt and Sanjay Patel.
- *Operating Systems Concepts* by Silberschatz, Galvin, and Gagne.
- *Digital Design* by M. Morris Mano and Michael Ciletti.
- *Computer Architecture: A Quantitative Approach* by John Hennessy and David Patterson.
- *Digital Design and Computer Architecture, RISC-V Edition* by Sarah Harris and David Harris.

In addition, we will be referencing official manuals and original papers, soft copies of which will be provided as necessary.

Course Load

53% 6 programming assignments (PAs)

- [3%] PA1 (Introduction to Linux)
- [10%] PA 2-6

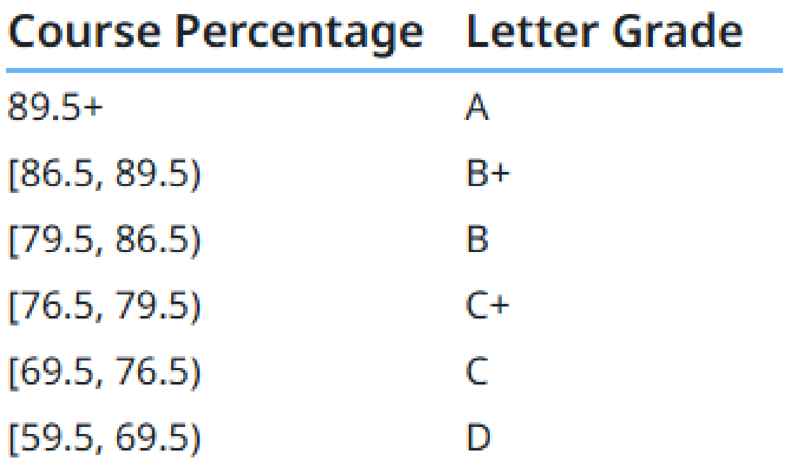
12% 11 “written” assignments (WAs)

- 3 lowest grades are dropped

15% Midterm in class (March 13)

20% Final during the exam period (May 14)

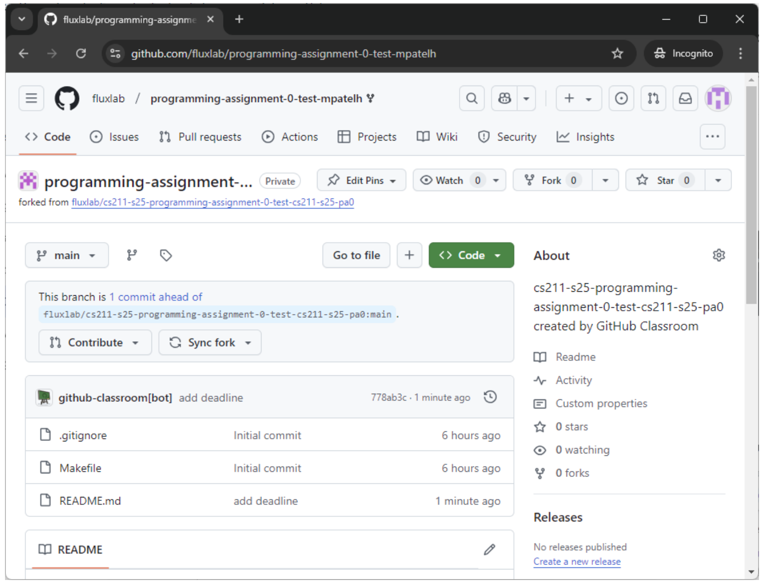
Letter Grade Assignments



Course Percentage	Letter Grade
89.5+	A
[86.5, 89.5)	B+
[79.5, 86.5)	B
[76.5, 79.5)	C+
[69.5, 76.5)	C
[59.5, 69.5)	D

Programming Assignment 1

- PA1 will orient you on the PA workflow
  - **Obtaining files:** GitHub Classroom
  - **Submitting files:** GradeScope (via GitHub)
  - **Development:** Rutgers CS Instructional Lab (ilab)
- I will demo the process next lecture



forked from fluxlab/cs211-s25-programming-assignment-0-test-cs211-s25-pa0

This branch is 1 commit ahead of `fluxlab/cs211-s25-programming-assignment-0-test-cs211-s25-pa0:main` .

github-classroom[bot] add deadline 778ab3c · 1 minute ago

.gitignore	Initial commit	6 hours ago
Makefile	Initial commit	6 hours ago
README.md	add deadline	1 minute ago

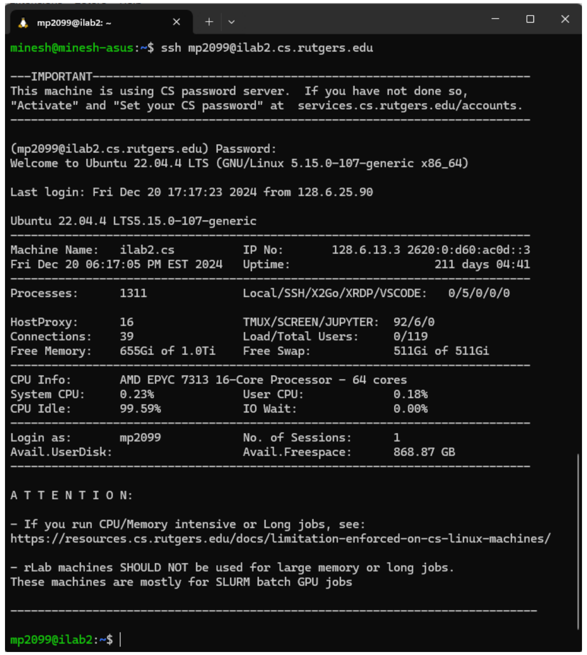
About

cs211-s25-programming-assignment-0-test-cs211-s25-pa0 created by GitHub Classroom

Releases

No releases published
Create a new release

PAs via GitHub



```
minesh@minesh-asus:~$ ssh mp2099@ilab2.cs.rutgers.edu

---IMPORTANT------------------------------------------------------------------
This machine is using CS password server.  If you have not done so,
"Activate" and "Set your CS password" at  services.cs.rutgers.edu/accounts.
------------------------------------------------------------------------------


(mp2099@ilab2.cs.rutgers.edu) Password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-107-generic x86_64)

Last login: Fri Dec 20 17:17:23 2024 from 128.6.25.90

Ubuntu 22.04.4 LTS5.15.0-107-generic
------------------------------------------------------------------------------
Machine Name:   ilab2.cs            IP No:       128.6.13.3 2620:0:d60:ac0d::3
Fri Dec 20 06:17:05 PM EST 2024   Uptime:                      211 days 04:41
------------------------------------------------------------------------------
Processes:      1311                Local/SSH/X2Go/XRDP/VSCODE:   0/5/0/0/0

HostProxy:      16                  TMUX/SCREEN/JUPYTER:  92/6/0
Connections:    39                  Load/Total Users:     0/119
Free Memory:    655Gi of 1.0Ti      Free Swap:            511Gi of 511Gi
------------------------------------------------------------------------------
CPU Info:       AMD EPYC 7313 16-Core Processor - 64 cores
System CPU:     0.23%               User CPU:             0.18%
CPU Idle:       99.59%              IO Wait:              0.00%
------------------------------------------------------------------------------
Login as:       mp2099              No. of Sessions:      1
Avail.UserDisk:                     Avail.Freespace:      868.87 GB
------------------------------------------------------------------------------


A T T E N T I O N:

- If you run CPU/Memory intensive or Long jobs, see:
https://resources.cs.rutgers.edu/docs/limitation-enforced-on-cs-linux-machines/

- rLab machines SHOULD NOT be used for large memory or long jobs.
These machines are mostly for SLURM batch GPU jobs

-------------------------------------------------------------------------------

mp2099@ilab2:~$
```

Logging in to ilab

Programming Assignments 2-6

- 50% of your total grade (10% each)

2. Introduction to C development + how computers represent numbers
3. Explicitly managing memory (when the language doesn't do it for you)
4. How computers represent programs as machine code
5. Emulating a simple RISC-V computer in C
6. Extending the emulator with more realistic memory

Written Assignments

- Questions related to weekly lecture topics
  - Helps you stay on track with lecture material
  - Helps me gauge how well students are following the material
- Administered through either **Canvas** or **GradeScope**
  - Multiple choice, short answer, and other autograded questions
- Representative of material that will be on the midterm and final

Exams

- In-person, handwritten, and manually graded

- Best way to prepare is to master the PAs and WAs

Collaboration Policy

- Please do work together to learn!
  - Discuss assignments with your classmates on Ed
  - Research and study concepts to prepare for exams

- **The bottom line:** submit your own work
  - Sharing or viewing solutions is academic misconduct
    - E.g., solutions from past semesters
    - E.g., posting your own solutions online
    - E.g., talking about your particular solution
    - ...
  - Please be careful! If in doubt, ask us and we will work with you

Welcome Survey

- We will send out a Google Form to survey where students currently stand

- We will have at least two more surveys:
  - Half-way survey
  - End of course survey

CS 211: Intro to Computer Architecture

1.1: Introduction and Syllabus

Minesh Patel

Spring 2025 – Tuesday 21 January