

CS203: Overture

Hung-Wei Tseng
Kuan-Chieh Hsu

Who we are?

CS203 Fall 2022 Pre-quarter Questionnaire

Welcome to the first CS203 after campus re-opening. We all know it's going to be different – at least we have to wear our masks during the whole lecture and you're not going to see each other's mouth! Therefore, I do have a few questions that I need to collect from you to decide what's the best to do in teaching method before the first lecture.

This form is automatically collecting emails for R'Mail users. [Change settings](#)

What's your name?
You name? *

Short answer text

I am a ... *

- PhD student in CSE
- PhD student in EE
- MS student in CSE
- MS student in CEN
- MS student in EE
- Other...

What's your favorite topic in computer science?

michael

srushti sai

deepthi
gayatri

ivann
boyi anudeep

sidharth

aung

yifan

nityash

jaired

scott

sidhant

tejas

aparna gwendolyn

ruchitha

josue aaron

jamella

archana

Why're you taking CS203

jonathan

kailin

emerson

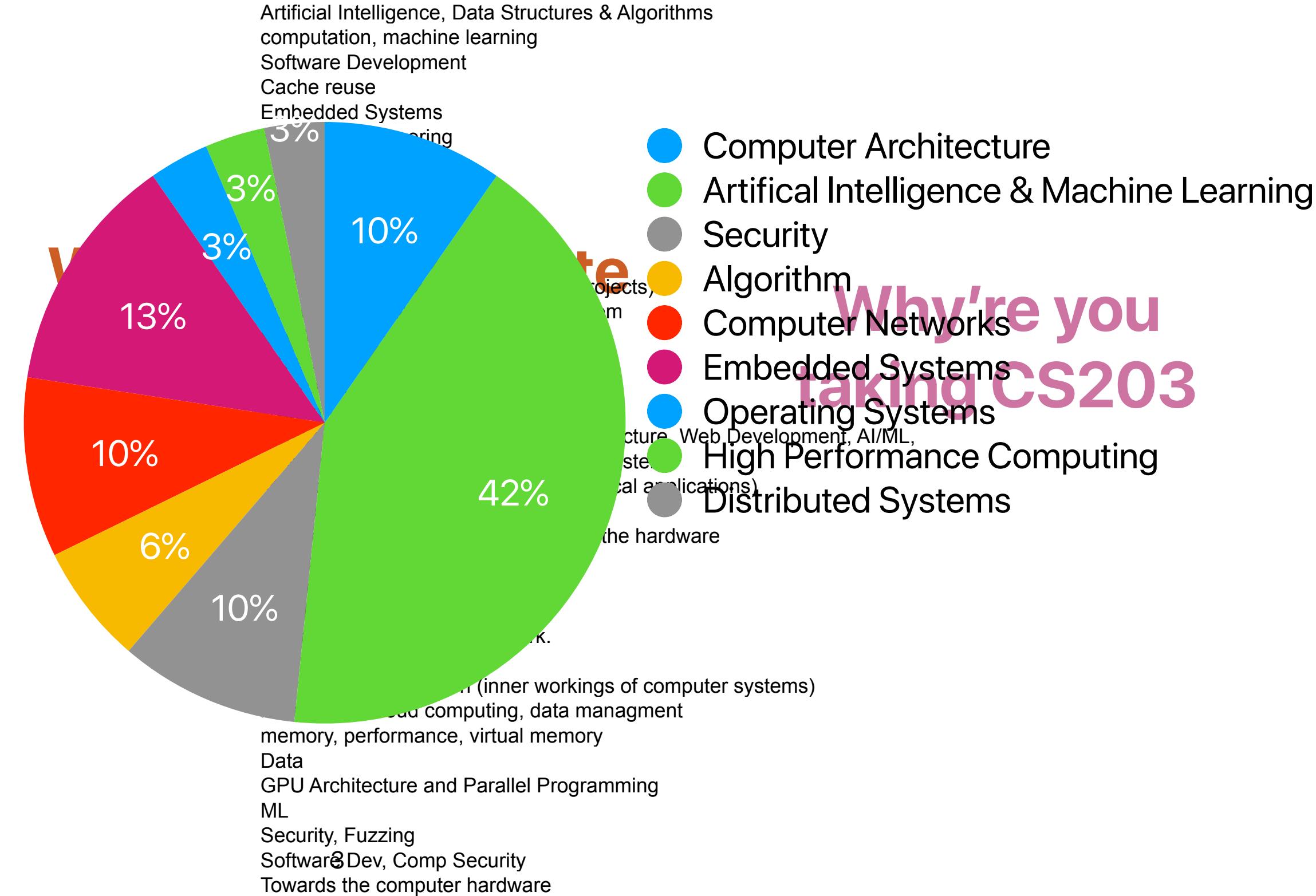
parth

Only 33 out of 80 respond...

Before we get there ...

What's your name?

Why're you taking CS203



Before we get there ...

What's your name?

A word cloud visualization of student responses to "What's your favorite topic in computer science?" and "Why're you taking CS203?". The words are colored by frequency, with larger words being more common. Two words are circled in red: "core" and "required".

The word cloud includes the following words (among many others):

- core
- required
- computer
- architecture
- science
- course
- systems
- knowledge
- hardware
- experience
- essential
- aligned
- intrigued
- depth
- tensor
- utilizing
- structure
- processing
- affects
- complete
- interested
- capabilities
- interesting
- team
- explore
- deeper
- tpu
- requirements
- attaining
- previous
- highly
- optimized
- gained
- forward
- aid
- thoroughly
- system's
- programs
- google
- concepts
- fundamental
- long-term
- help
- advanced
- quarter
- therefore
- develop
- system
- subject
- understand
- design
- developer
- fascinated
- enjoyed
- levels
- always
- looking
- lower-level
- necessary
- covered
- major
- security
- hopefully
- side need
- goal
- apply
- experience
- essential
- google's
- give
- major
- computer's
- computers
- aspects
- fully
- earned
- even
- units
- aligned
- intrigued
- believe
- better
- its my core subject
- I am interested in the lower-level design of computers.
- I am more interested in learning the Computer Architecture and deeper levels of systems.
- Computer architecture is necessary for any software developer to understand the system and the executions thoroughly. CS203 course structure
- Core course. I also hope to write an out of order stack machine for fun at some point.
- It is a required core class
- It is a requirement, but I am interested in learning the content as well.
- I want to learn more about Computer Architecture (and it's a core course ;p)
- Core requirement
- It's a core course
- I think that in order to become a good programmer and coder, it is important to know about the functions of various components and the working
- It's my core course.
- Interesting
- computer architecture is essential to the efficiency of computer systems. Understanding the underlying architectural concepts can help me
- it is a core class, and I am least familiar with architecture
- Want to have hand-on experience on C++. Beneficial for future job preferences.
- I have a background in electronics and two years of experience as a software engineer, which has given me a solid understanding of how
- I am interested in Computer Architecture and would like to know where is the focus of innovation in this field currently.

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- Visiting Researcher @ Google, 01/2023—03/2023
 - Working for TensorFlow Lite
- PhD in **Computer Science**, University of California, San Diego, 2014
- Research Interests
 - General-purpose computing on AI/ML/NN accelerators
 - Intelligent storage devices & near-data processing
 - Or anything else fun — we have an OpenUVR project recently



ChatGPT

Not true

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

Who

intra.engr.ucr.edu/~htseng/
forward, hold to see history

Wei Tseng

ABOUT ME RESEARCH PROJECTS PUBLICATIONS ADVISING &



HUNG-WEI TSENG

Associate Professor, University of California, Riverside

I am currently an associate professor in the Department of Electrical and Computer Engineering at the University of California, Riverside, and a cooperating faculty of the Department of Computer Science and Engineering. I am now leading the Extreme Storage & Computation group.

I am interested in diverse research topics that allow applications and programs to efficiently use modern heterogeneous hardware components. Together with my students, our recent work has demonstrated the potential of using emerging AI/ML algorithms (e.g., Edge TPUs) in improving the performance of non-AI/ML workloads through a framework [GitHub]. We also showed how intelligent storage devices can improve the performance, power and energy for heterogeneous computers. Our research on storage systems has been recognized by two best paper nominations from the IEEE Symposium on Microarchitecture in 2021 and 2019, IEEE Micro "Top Paper" in the "Computer Architecture Conferences" (IEEE MICRO Top Picks 2020) and Facebook Researcher Prize. In addition, we also applied our knowledge in optimizing storage systems to stacks and developed the OpenUVR project [GitHub] that enables high-quality VR experience on commodity hardware components and won the outstanding paper award in 2021.

Bard Experiment



Reset chat

Bard Activity

FAQ

Help & support

I'm Bard, your creative and helpful collaborator. I have limitations and won't always get it right, but your feedback will help me improve.

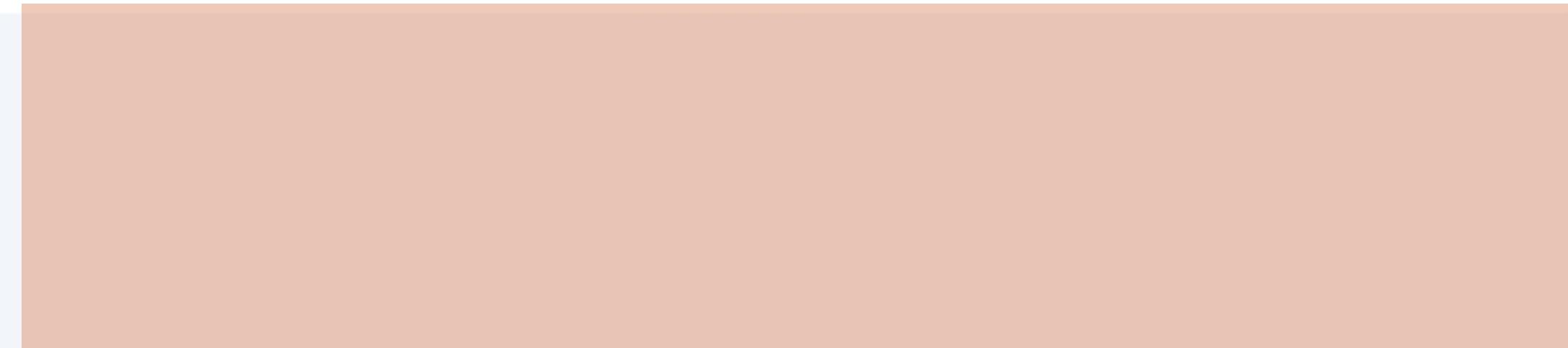
Not sure where to start? You can try:

Explain why large language models sometimes make mistakes

Help me incorporate more high-protein vegan options in my diet

I want to write a novel. How can I get started?

I wish



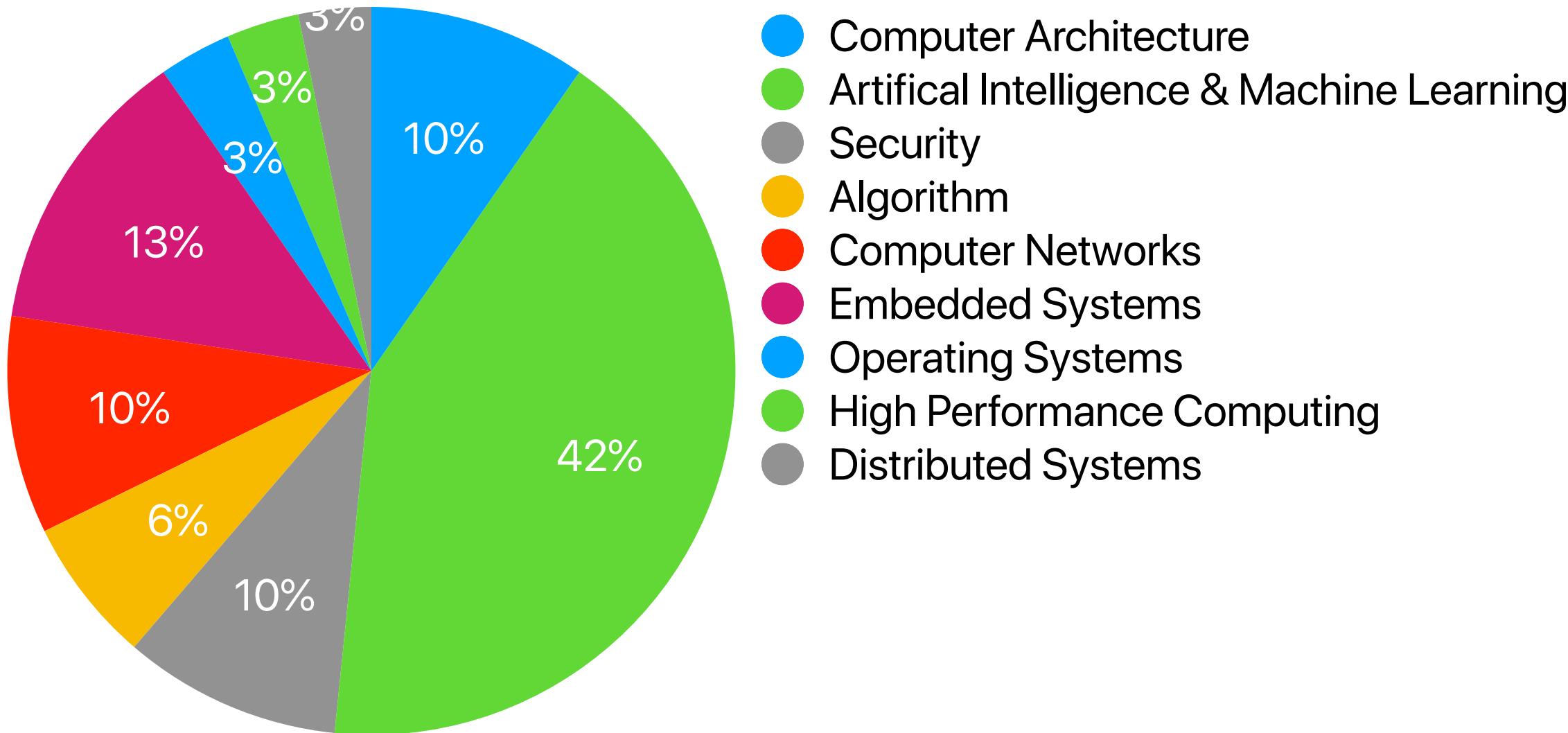
At least Bard knows how to Google

Who is



Before we get there ...

What's your favorite topic in computer science?



What do you care as a computer scientist/engineer?



Artificial Intelligence
Machine Learning
Computer Architecture
Software Engineering
Programming Languages
User Interfaces

**Why should I care about Computer
Architecture?**

The return of backpropagation

- Between 2005 and 2009 researchers (in Canada!) made several technical advances that enabled backpropagation to work better in feed-forward nets.
 - Unsupervised pre-training; random dropout of units; rectified linear units.
 - The technical details of these advances are very important to the researchers but they are not the main message.
 - The main message is that backpropagation now works amazingly well if you have two things:
 - a lot of labeled data
 - a lot of convenient compute power (e.g. GPUs)



2018 Turing Award



Hung-Wei
Tseng

David Patterson

John Hennessy

Sixth Edition

John L. Hennessy | David A. Patterson

COMPUTER ARCHITECTURE

A Quantitative Approach



Computer Architecture

Enables

Deep Learning

Introducing ChatGPT

UCR CS203 Instructor Search

chat.openai.com/chat

Incognito

+ New chat

UCR CS203 Instructor

Hung-Wei Tseng's Achievements

NV Memory Adoption Cost

Clear conversations

Upgrade to Plus NEW

Dark mode

Updates & FAQ

Log out

UCR's official website or contact the university's department of computer science directly for the most up-to-date information on course instructors.

Who is Hung-Wei Tseng?

I'm sorry, but as an AI language model, I do not have access to personal information about individuals unless they are notable public figures. Without further context or information, I am unable to determine who Hung-Wei Tseng is. Can you please provide me with more context or details about the individual in question?

Who is Hung-Wei Tseng at UCR?

Hung-Wei Tseng is an Associate Professor of Computer Science and Engineering at the University of California, Riverside (UCR). His research interests are in the area of computer networks and distributed systems, with a focus on improving network performance and security. Some of his recent research work has focused on topics such as network measurement and monitoring, network optimization, and network security. Hung-Wei Tseng has been a faculty member at UCR since 2013, and he has received several awards and recognitions for his research contributions.

Great answer!!!

Send a message...

Regenerate response

ChatGPT Mar 14 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Introducing ChatGPT | UCR CS203 Instructor Search | Bard

bard.google.com

Bard Experiment

Email: htseng@ucr.edu

Reset chat

Bard Activity

FAQ

Help & support

Who is Hung-Wei Tseng?

Hung-Wei Tseng is an associate professor in the Department of Electrical and Computer Engineering and a cooperating faculty of the Department of Computer Science and Engineering at University of California, Riverside. He is now leading the Extreme Storage & Computer Architecture Laboratory. He is interested in diverse research topics that allow applications or programmers to more efficiently use modern heterogeneous hardware components. Together with his students, they have demonstrated how intelligent storage devices can help improve performance, power and energy for data-intensive systems.

Their work has been recognized by IEEE Micro "Top Picks from the 2019 Computer Architecture Conferences" (IEEE MICRO Top Picks 2020) and Facebook Research Award, 2018. Prior to joining UCR, he served as an assistant professor for the Department of Computer Science and the Department of Electrical and Computer Engineering at NC State University. He was a PostDoc of the Non-volatile Systems Laboratory and a lecturer of the Department of Computer Science and Engineering at University of California, San Diego with Professor Steven Swanson.

Sources - Learn more

1. intra.engr.ucr.edu/~htseng/
2. www.nvsl.io/portfolio/hung-wei-tseng/

Great answer!!!

Enter a prompt here

Bard may display inaccurate or offensive information that doesn't represent Google's views.

**OK, I know computer architecture is
important. But will that give me a job?**

We're now at a "New Golden Age" of computer architecture

The image is a collage of news snippets from various media sources, primarily from the early 2010s, illustrating the trend of major tech companies developing their own custom chips:

- CNBC Article 1:** Headline: "Google is making a big-time move that should scare Nvidia". Subtitle: "Follows Google in designing a chip to make cloud computing more efficient; the chip uses designs typically seen in smartphones." Published: May 17, 2017.
- CNBC Article 2:** Headline: "Google reportedly plans to put its own chips in Chromebook laptops from 2023". Subtitle: "Google plans to use its own chips in Chromebooks and tablets that run on the company's Chrome operating system from around 2023." Published: Sep 1, 2021.
- Fortune Article:** Headline: "Tech giants are rushing to develop their own chips — here's why". Published: After Amazon, Apple And Google, Facebook To Develop Its Own ML Chip.
- WIRED Article:** Headline: "at Amazon: Its Own Chips for Cloud putting". Subtitle: "Follows Google in designing a chip to make cloud computing more efficient; the chip uses designs typically seen in smartphones."

KEY POINTS

- Google plans to use its own chips in Chromebooks and tablets that run on the company's Chrome operating system from around 2023.
- Google currently uses chips made by the likes of Intel and AMD to power Chromebooks.
- Facebook is building a machine learning chip to manage content recommendations for its users.

AI explosion stimulates the computation demand

HOME · COMPUTING · NEWS

Microsoft explains how thousands of Nvidia GPUs built ChatGPT



By Jacob Roach

March 13, 2023

fierceelectronics.com/sensors/chatgpt-runs-10k-gpus

SENSORS

Update: ChatGPT training GPUs will take thousands more

By Matt Hamblen • Feb 11, 2023 11:29am

chatbots

ChatGPT

NVIDIA

GPU

infrastructure to support ChatGPT and projects like

TECH

Meet the \$10,000 Nvidia chip powering the race for A.I.

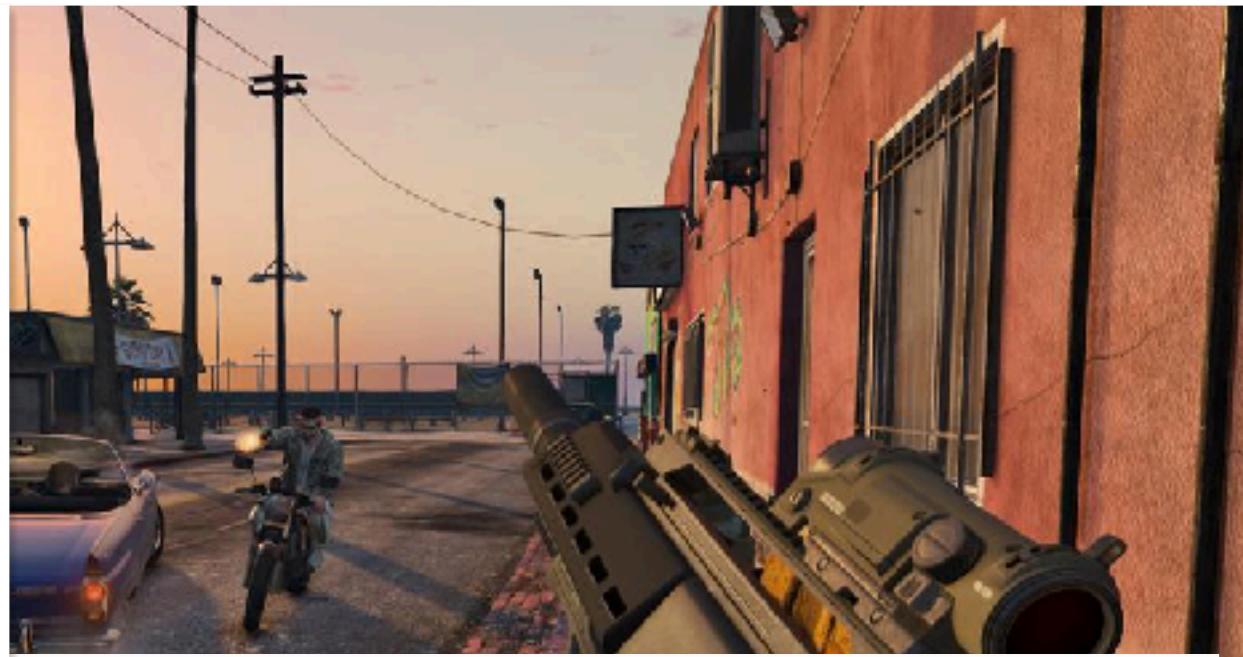
yahoo/finance

The AI explosion could help Micron and the chip industry turn the corner

60
f
t
e



Computer architecture also enables ...



**OK. Computer Architecture is
important. What is “Computer
Architecture” really about?**

Your impression about computer architecture

What's your impression about computer architecture?

Why these components?

What's computer architecture?



architecture noun

ar·chi·tec·ture | \är-ki-ték-chör \

Definition of *architecture*

1 : the art or science of building

specifically : the art or practice of designing habitable ones

2 **a** : formation or construction resulting from design and

// the architecture of the garden

b : a unifying or coherent form or structure

// a novel that lacks architecture

3 : architectural product or work

// buildings that comprise the architecture of the square

4 : a method or style of building

// Gothic architecture

5 : the manner in which the components of a computer or computer system are organized and integrated

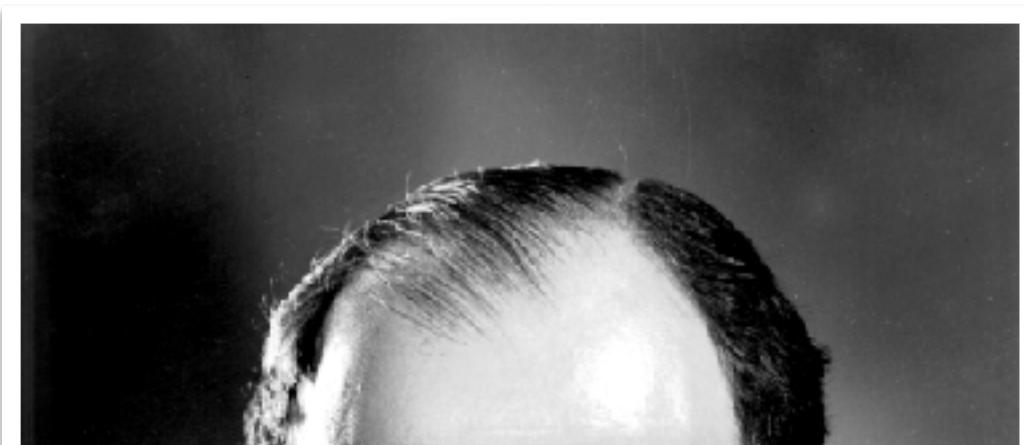
// different program architectures

**The manner in which the components
of a computer or computer system are
organized and integrated**

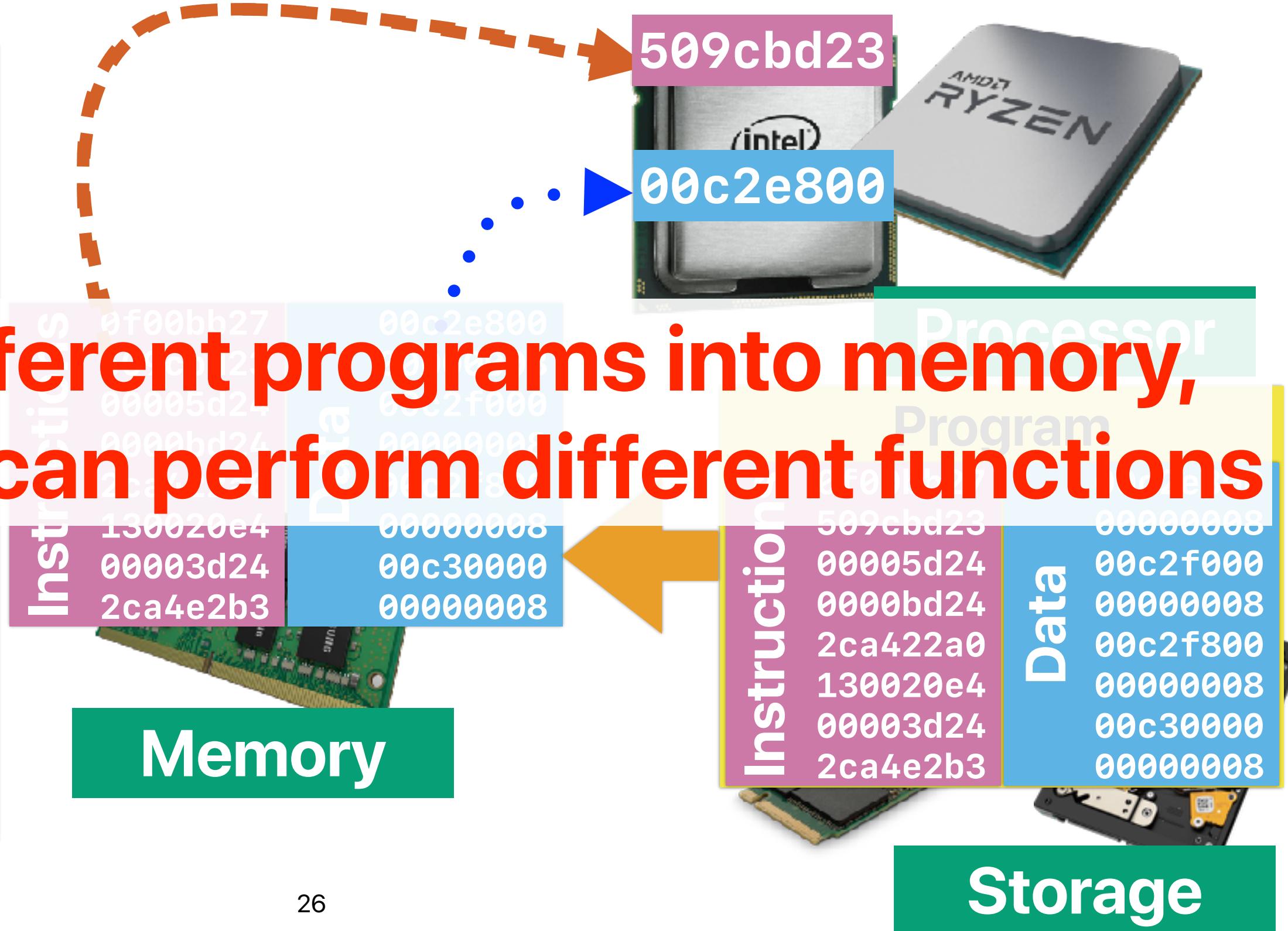
How does a computer execute a program?

The big picture of “computer architecture”

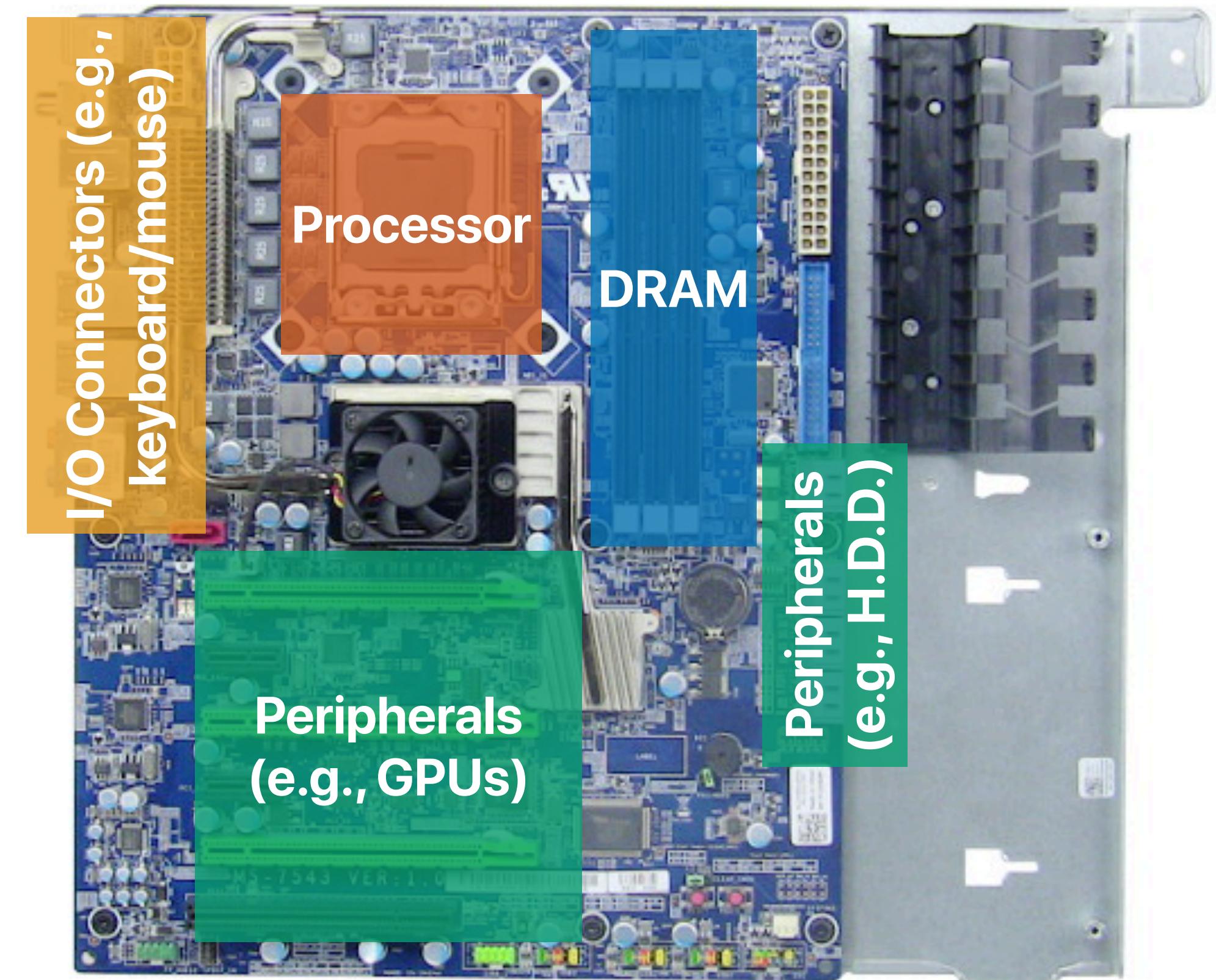
von Neumann architecture



By loading different programs into memory,
your computer can perform different functions

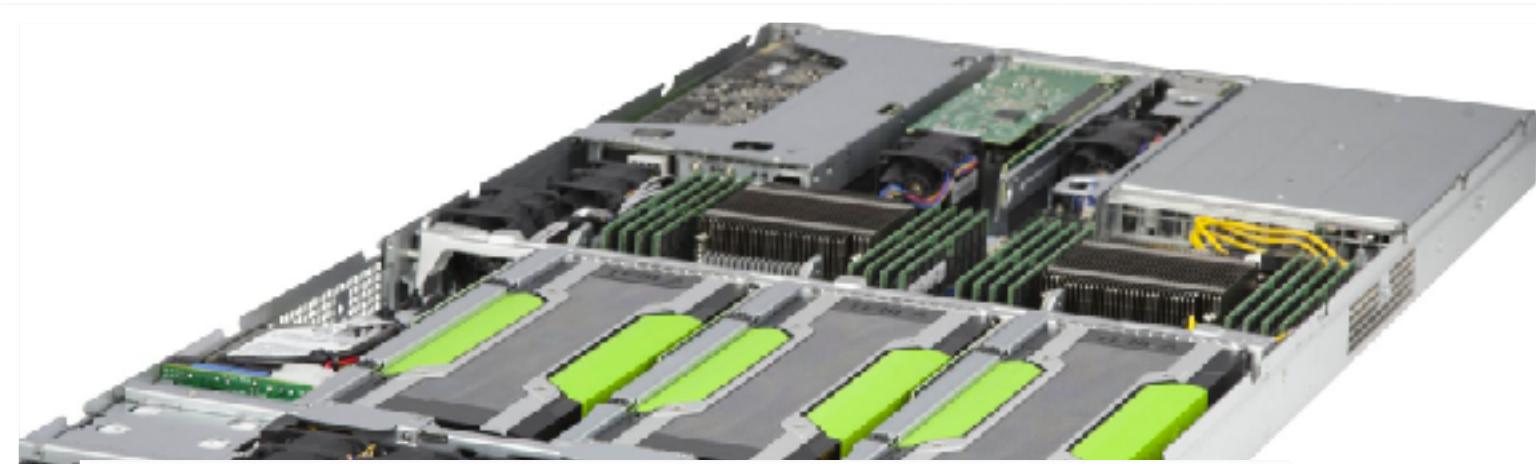


Desktop Computer



Server

I/O Connectors (e.g., keyboard/mouse)



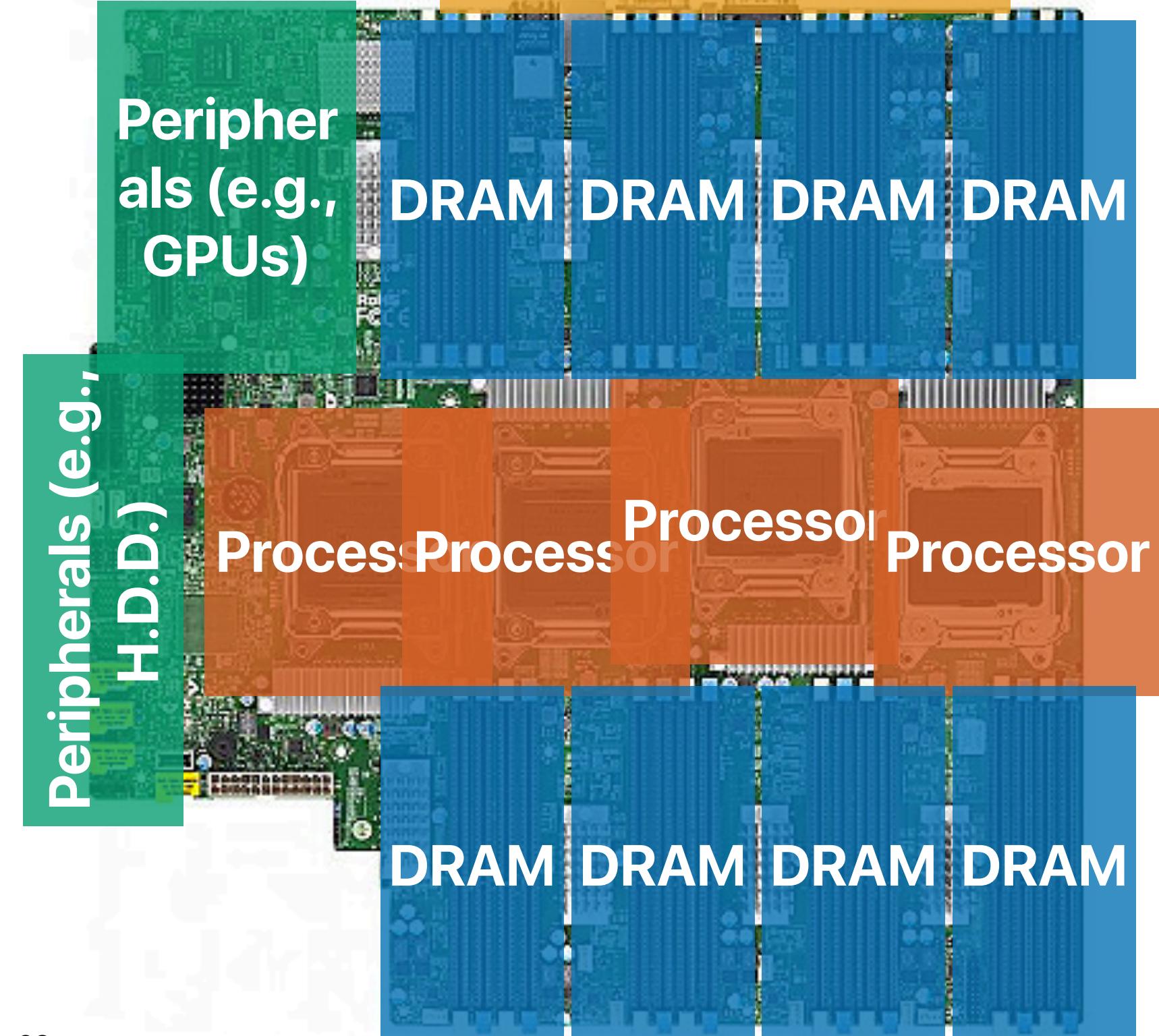
Peripherals (e.g., H.D.D.)

Peripherals (e.g., GPUs)

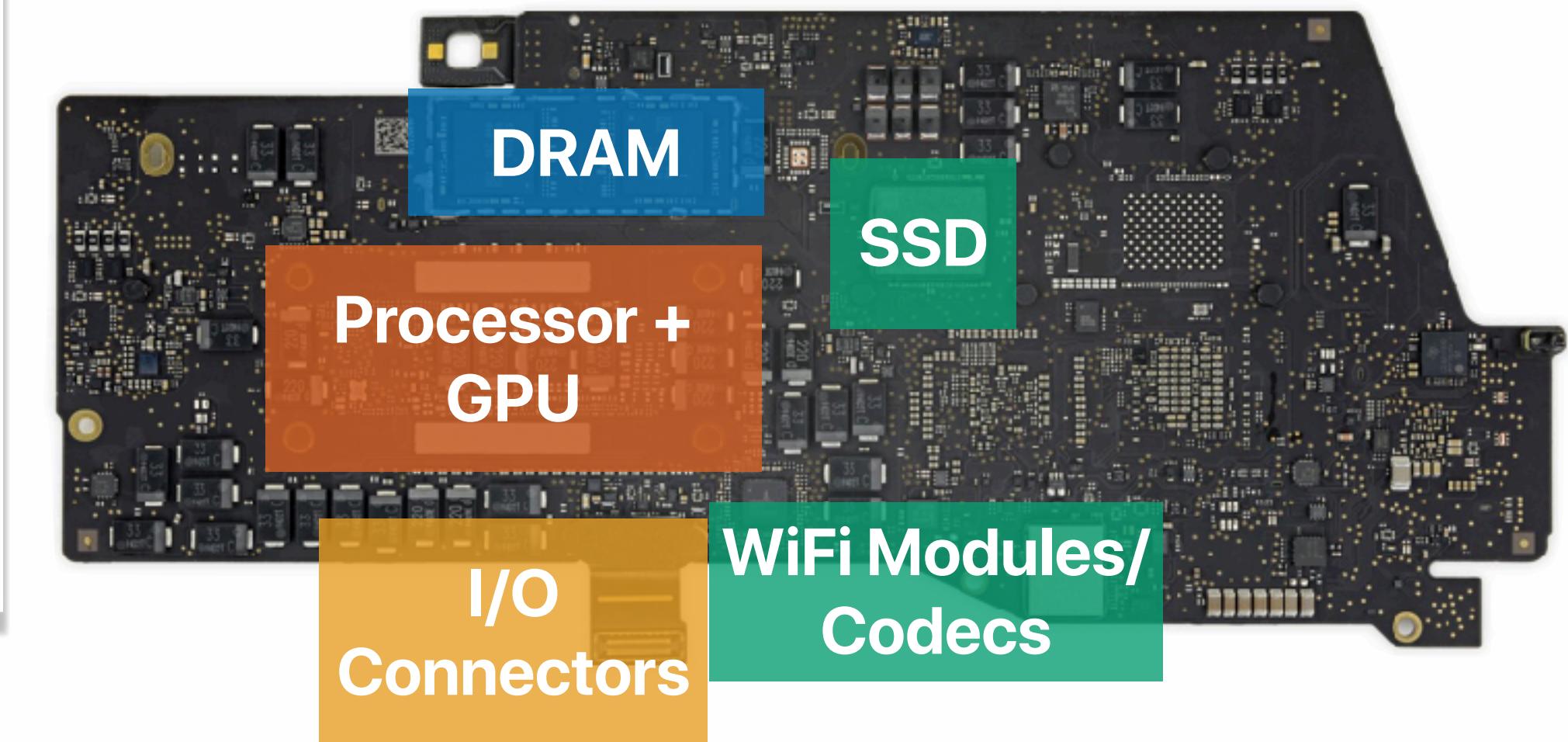
DRAM DRAM DRAM DRAM

Processor Processor Processor Processor

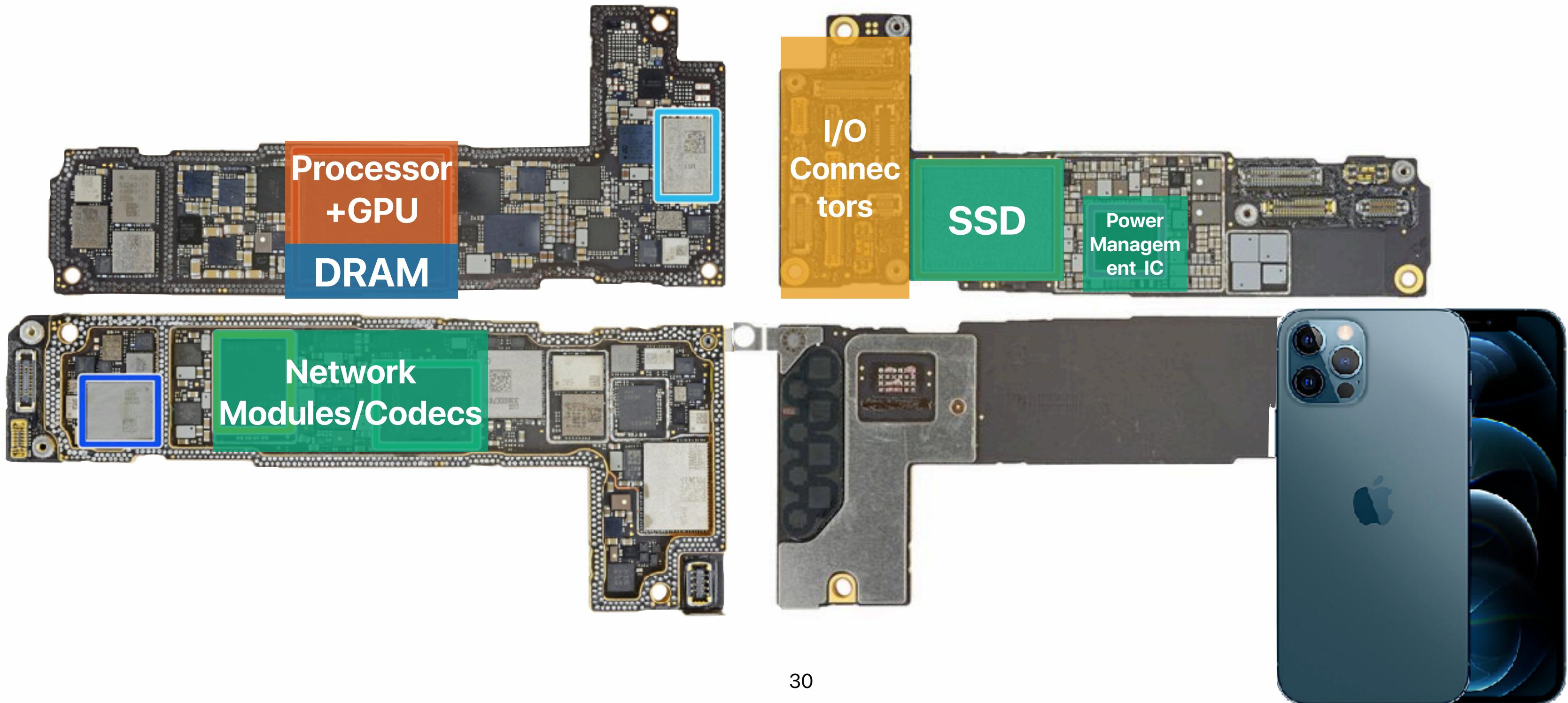
DRAM DRAM DRAM DRAM



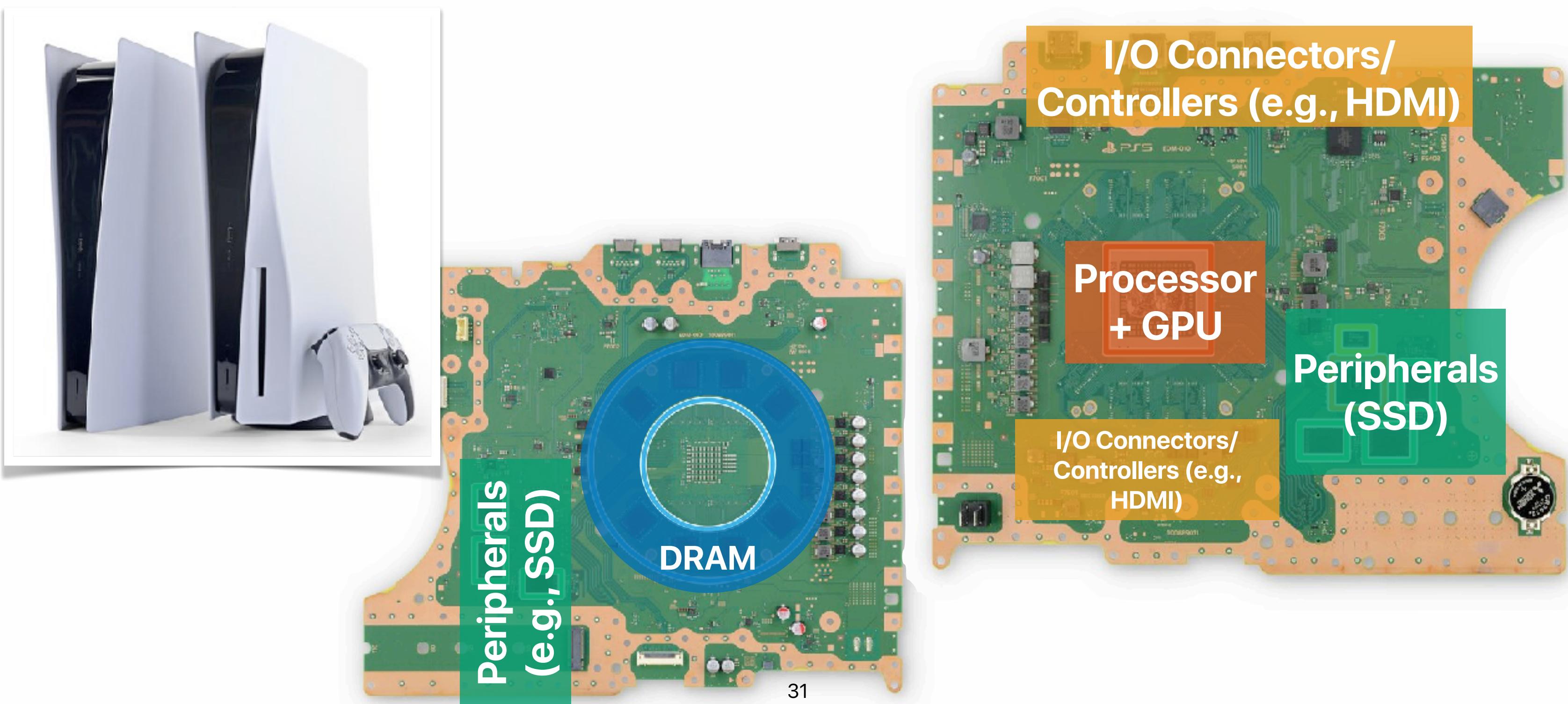
MacBook Pro 13"



iPhone 14 Pro



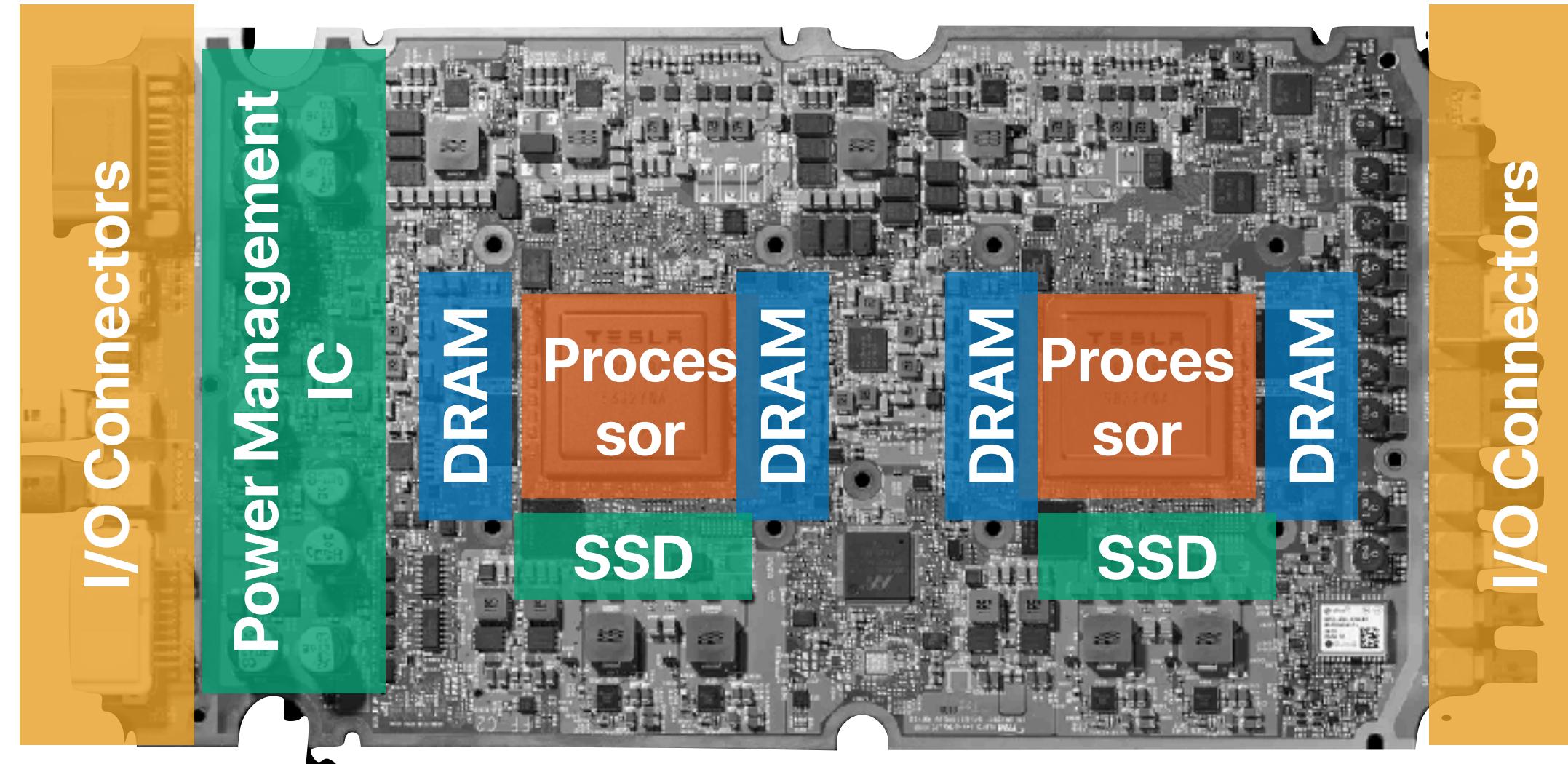
Play Station 5



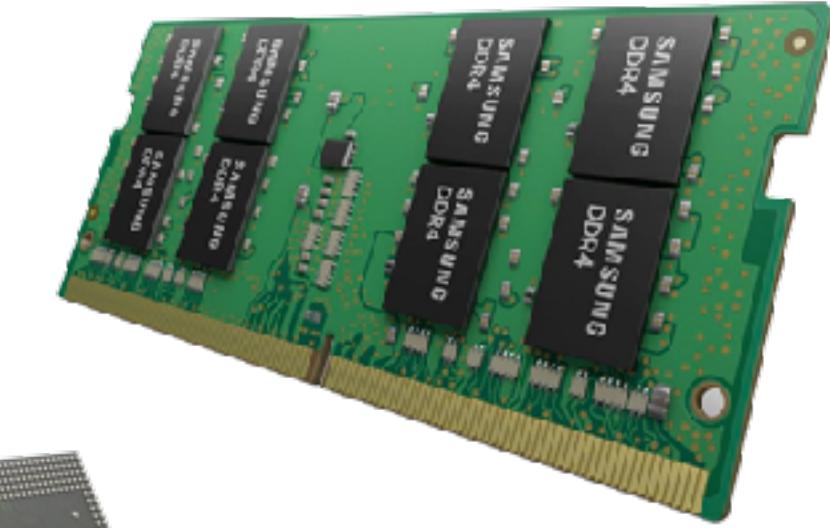
Nintendo Switch



Tesla Model 3



Processors and memory modules are everywhere!



Processors



Memory



Challenges of von Neumann Architecture

Gordon Moore

- January 3, 1929 – March 24, 2023
- Famous for “Moore’s Law”
- Intel Co-founder

Gordon E. Moore, Intel Co-Founder Behind Moore’s Law, Dies at 94

His prediction in the 1960s about rapid advances in computer chip technology charted a course for the age of high tech.



Gordon E. Moore in 1990 at the Silicon Valley headquarters of Intel, which he founded in 1968 with Robert Noyce. Alamy

Moore's Law

Present and future

By integrated electronics, I mean technologies which are referred to today as well as any additional result in electronics functions supplied as irreducible units. These technologies include the ability to miniaturize electronics equipment, increasingly complex electronic functions in space with minimum weight. Several have evolved, including microassembly of individual components, thin-film and semiconductor integrated circuits.

Two-mil squares

With the dimensional tolerances already being employed in integrated circuits, isolated high-performance transistors can be built on centers two thousandths of an inch apart. Such a two-mil square can also contain several kilohms of resistance or

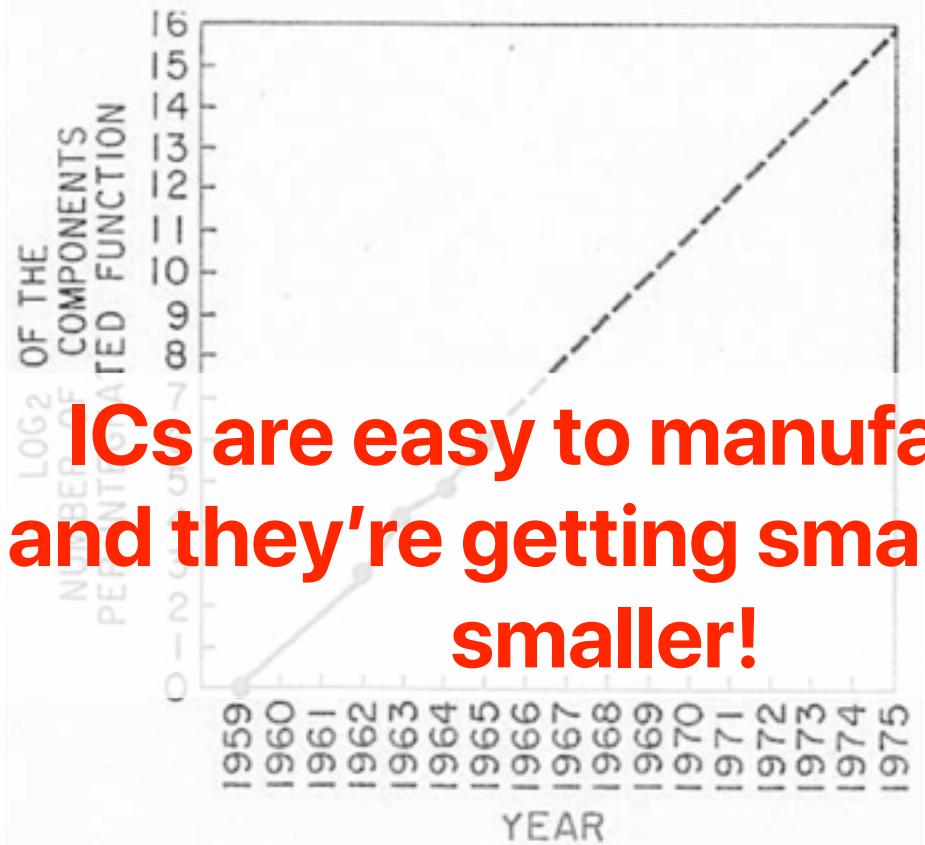
ICs are small

(1) Mo

The establishment

Increasing the yield

There is no fundamental obstacle to achieving device yields of 100%. At present, packaging costs so far exceed the cost of the semiconductor structure itself that there is no incentive to improve yields, but they can be raised as high as is economically justified. No barrier exists comparable to the thermodynamic equilibrium considerations



ICs are easy to manufacture and they're getting smaller and smaller!

Linear circuitry

Integration will not change linear systems as radically as digital systems. Still, a considerable degree of integration will be achieved with linear

circuits. The lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement

linear functions in discrete form.

Reliability count

In almost every field of electronics, ICs have demonstrated higher reliability than discrete components. The lack of large-value capacitors and inductors makes it difficult to implement linear functions in discrete form.

Integration will not change linear systems as radically as digital systems. Still, a considerable degree of integration will be achieved with linear

circuits. The lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

functions in discrete form.

However, the lack of large-value capacitors and

inductors makes it difficult to implement linear

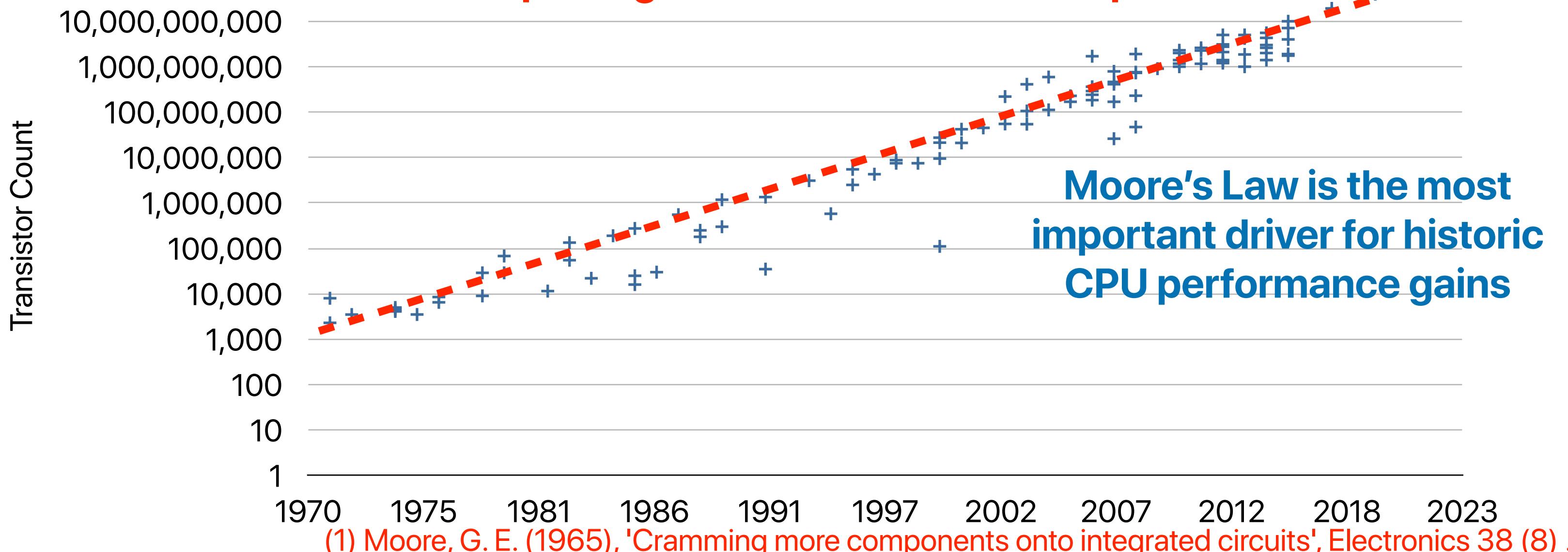
functions in discrete form.

</

Moore's Law

- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.

We still have no trouble putting more transistors in a chip even for now!

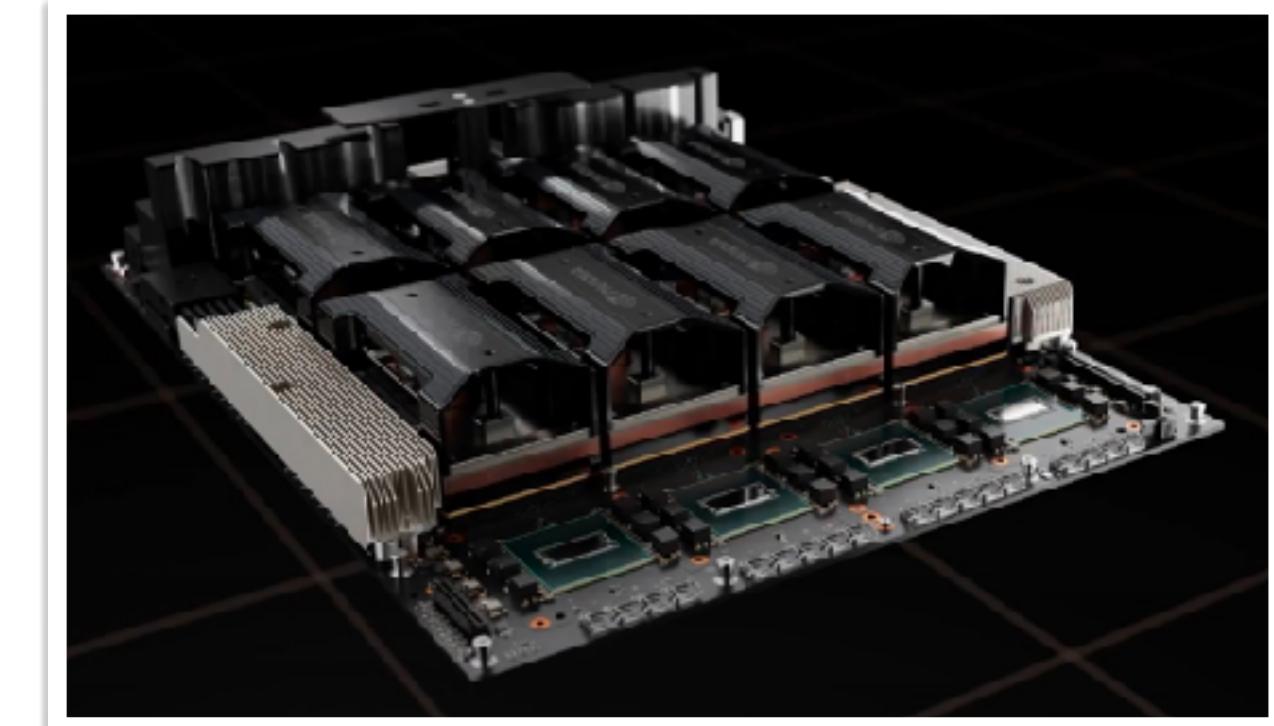


We still have no trouble cramming more transistors

NVIDIA Accelerator Specification Comparison			
	H100	A100 (80GB)	V100
FP32 CUDA Cores	16896	6912	5120
Tensor Cores	528	432	640
Boost Clock	~1.78GHz (Not Finalized)	1.41GHz	1.53GHz
Memory Clock	4.8Gbps HBM3	3.2Gbps HBM2e	1.75Gbps HBM2
Memory Bus Width	5120-bit	5120-bit	4096-bit
Memory Bandwidth	3TB/sec	2TB/sec	900GB/sec
VRAM	80GB	80GB	16GB/32GB
FP32 Vector	60 TFLOPS	19.5 TFLOPS	15.7 TFLOPS
FP64 Vector	30 TFLOPS	9.7 TFLOPS (1/2 FP32 rate)	7.8 TFLOPS (1/2 FP32 rate)
INT8 Tensor	2000 TOPS	624 TOPS	N/A
FP16 Tensor	1000 TFLOPS	312 TFLOPS	125 TFLOPS
TF32 Tensor	500 TFLOPS	156 TFLOPS	N/A
FP64 Tensor	60 TFLOPS	19.5 TFLOPS	N/A
Interconnect	NVLink 4 18 Links (900GB/sec)	NVLink 3 12 Links (600GB/sec)	NVLink 2 6 Links (300GB/sec)
GPU	GH100 (814mm ²)	GA100 (826mm ²)	GV100 (815mm ²)
Transistor Count	80B	54.2B	21.1B
TDP	700W	400W	300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFN
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta

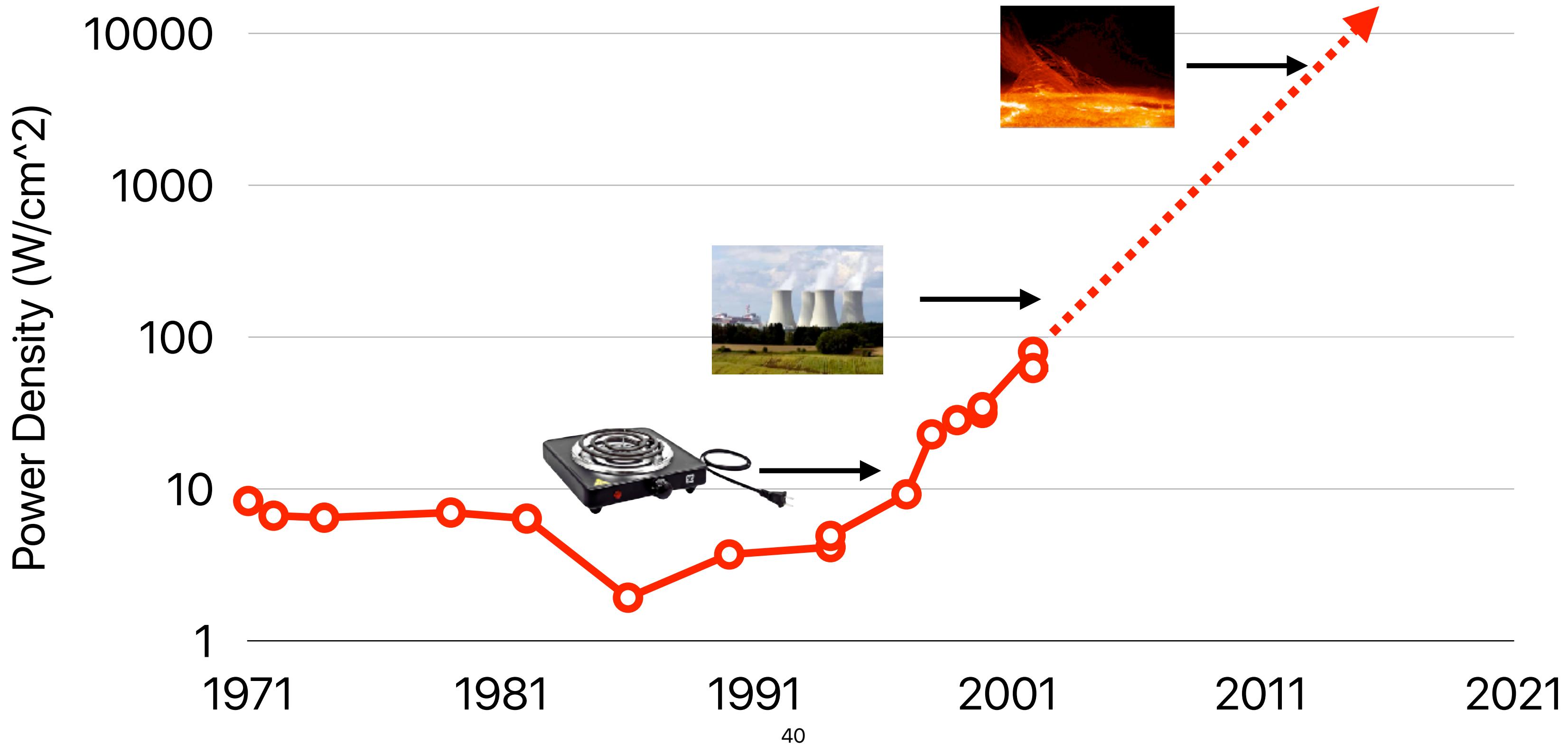


<https://www.workstationspecialist.com/product/nvidia-tesla-a100/>

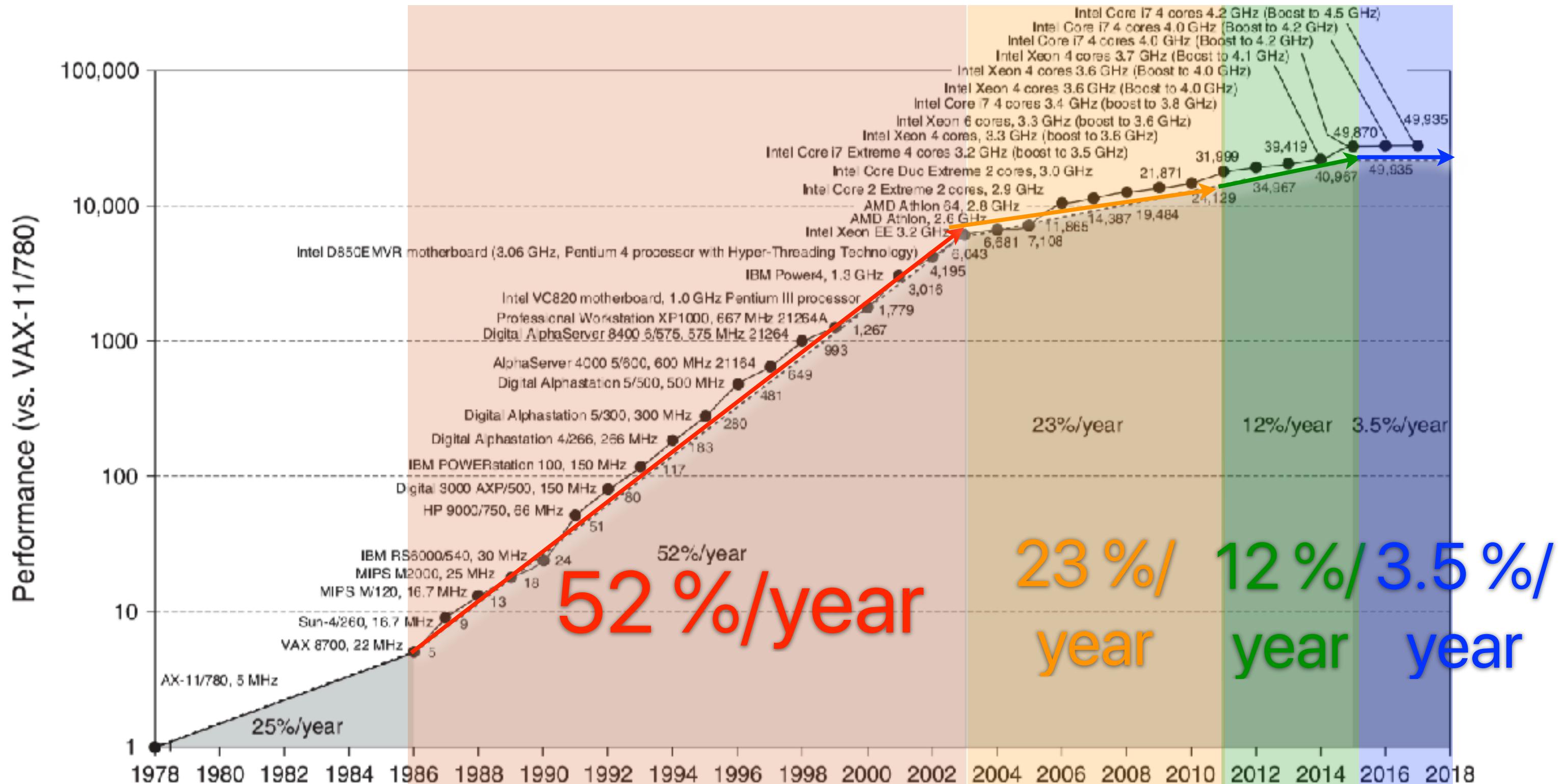


<https://www.servethehome.com/wp-content/uploads/2022/03/NVIDIA-GTC-2022-H100-in-HGX-H100.jpg>

Power Density of Processors



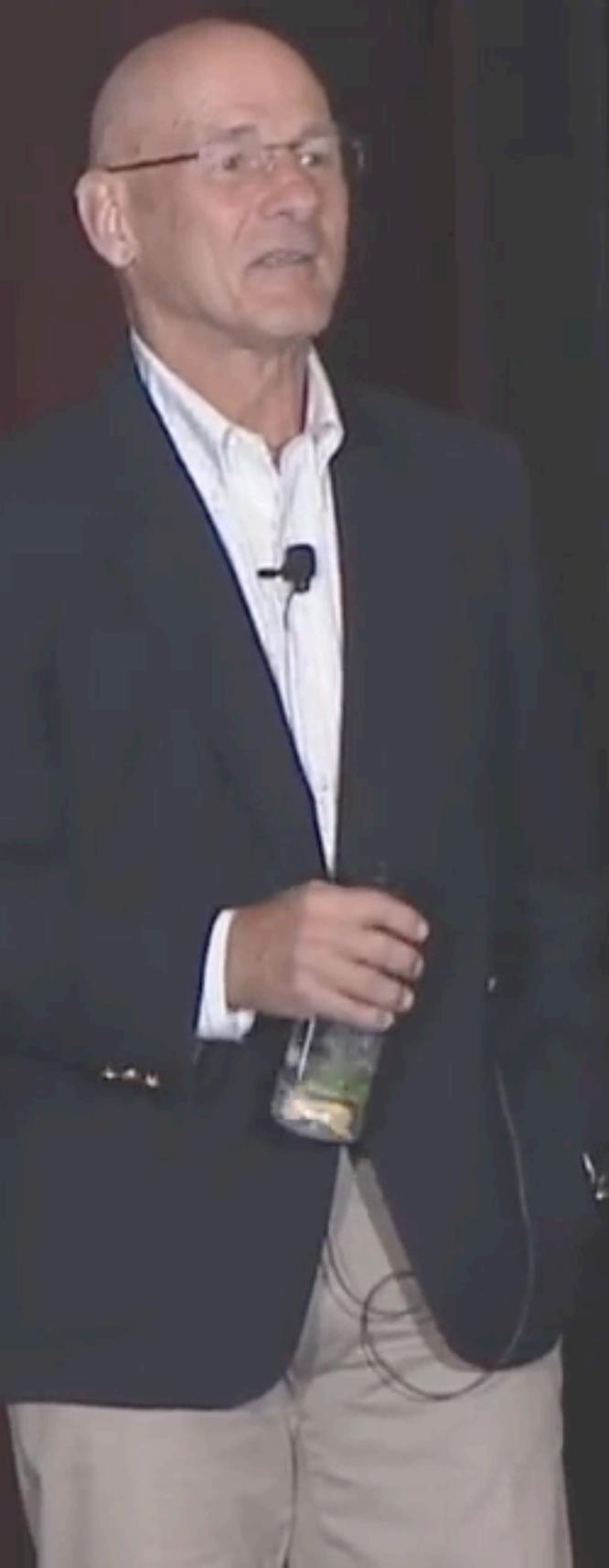
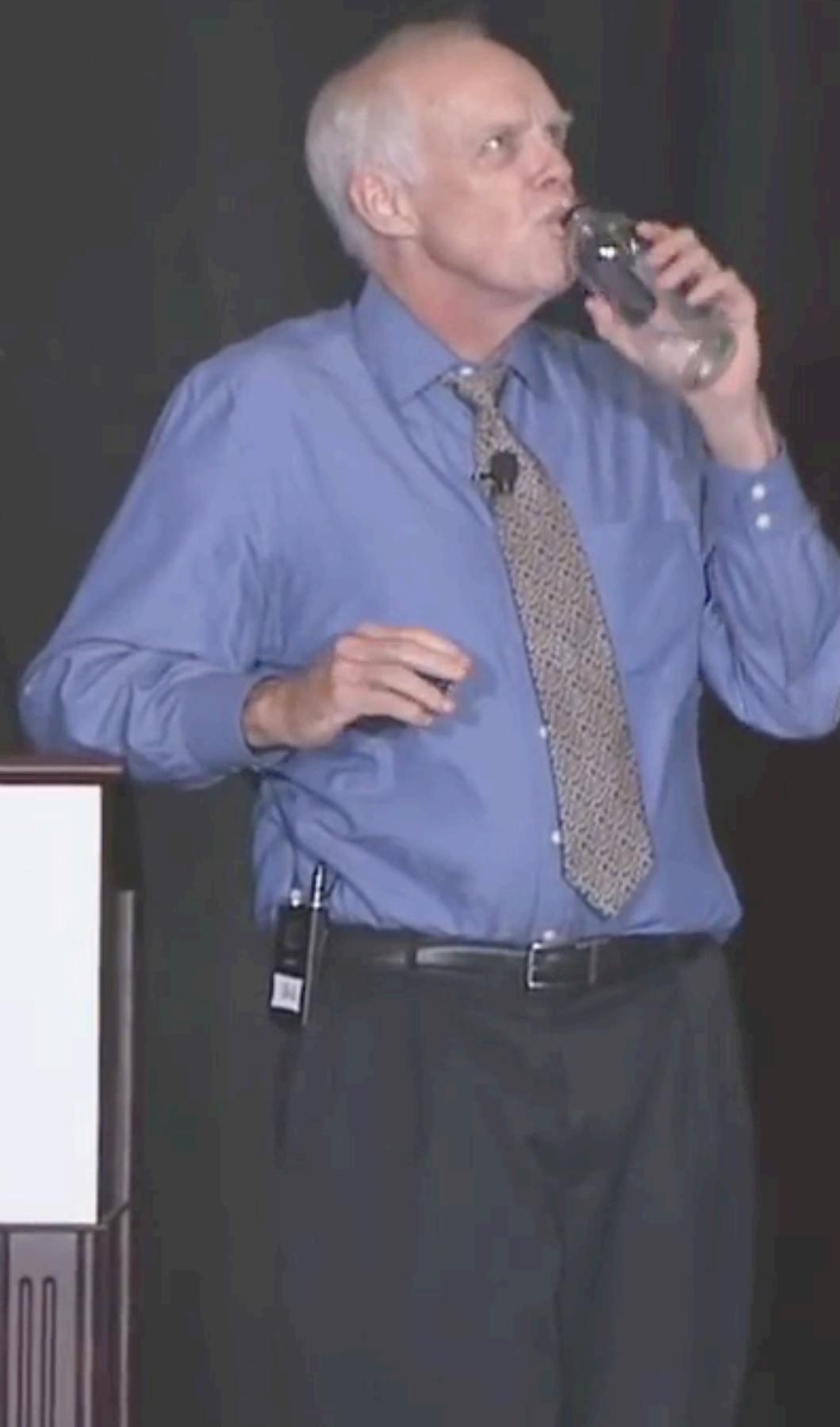
Single CPU core performance scaling slows down



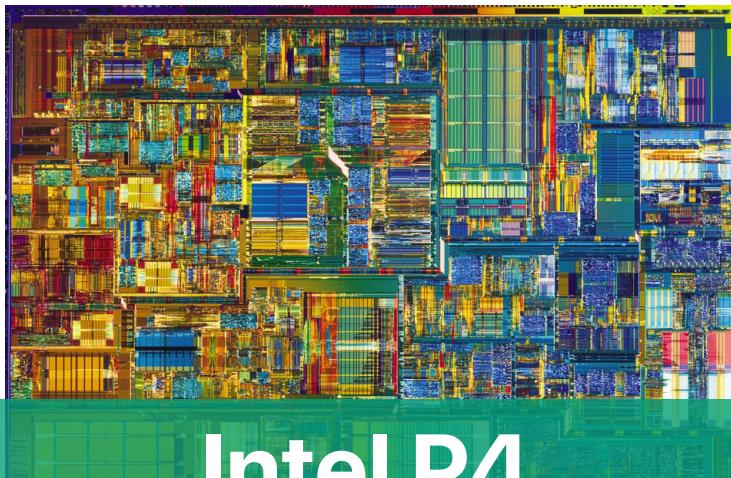


The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA

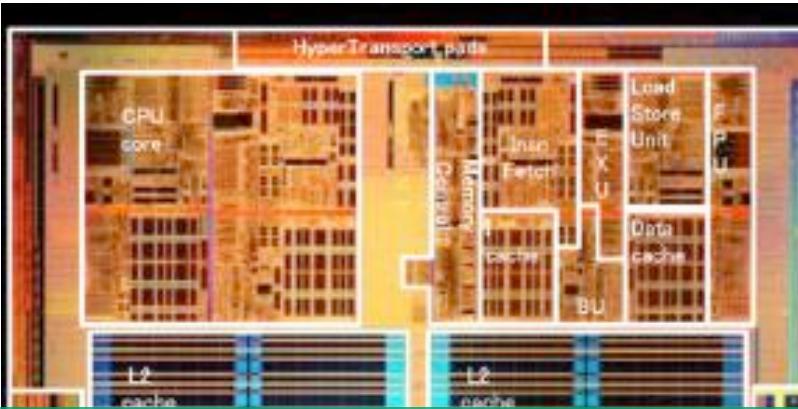
A 2018
g Lecture



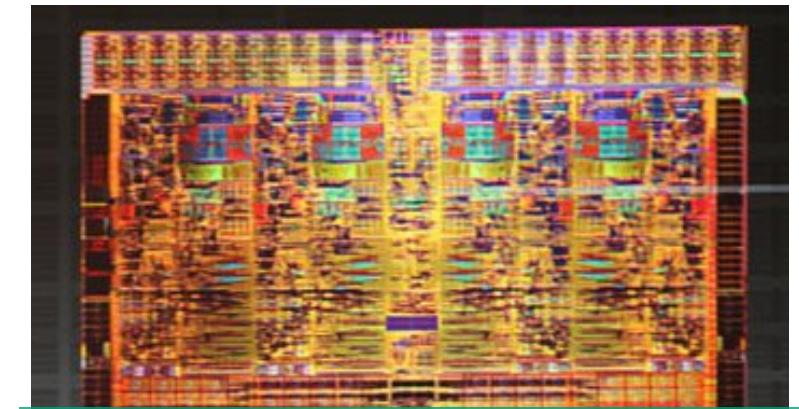
The era of chip multiprocessors



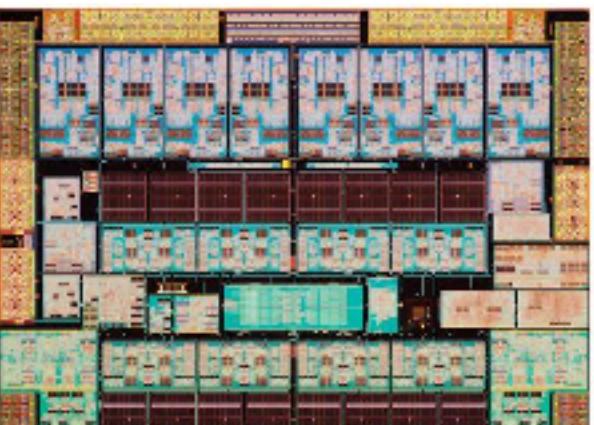
Intel P4
(2000)
1 core



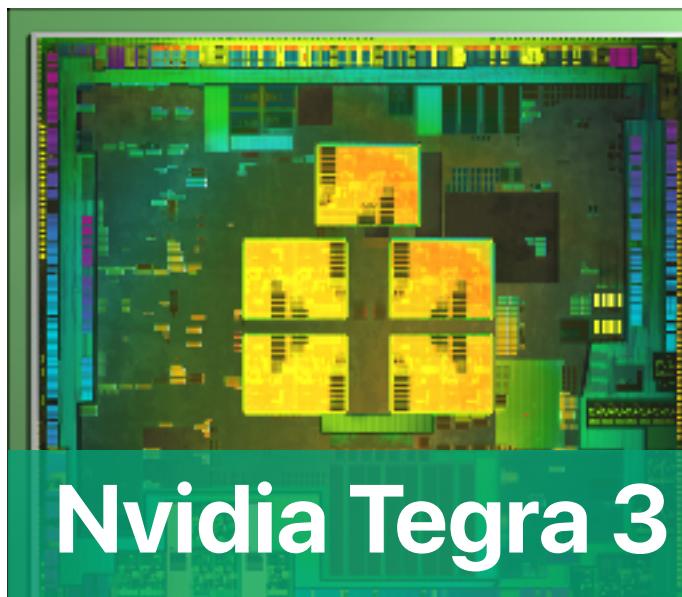
AMD Athlon 64 X2
(2005)
2 cores



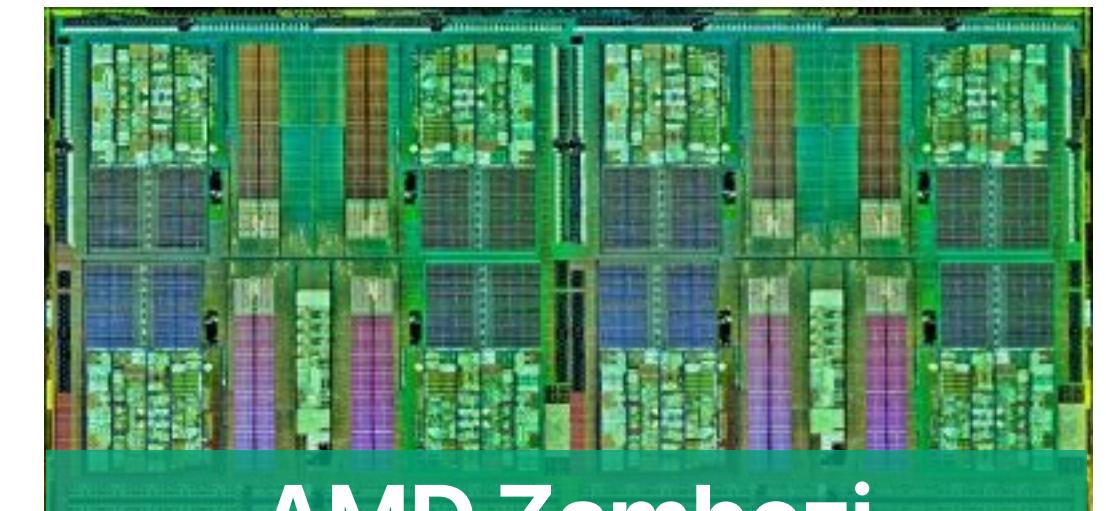
Intel Nahalem
(2010)
4 cores



SPARC T3
(2010)
16 cores

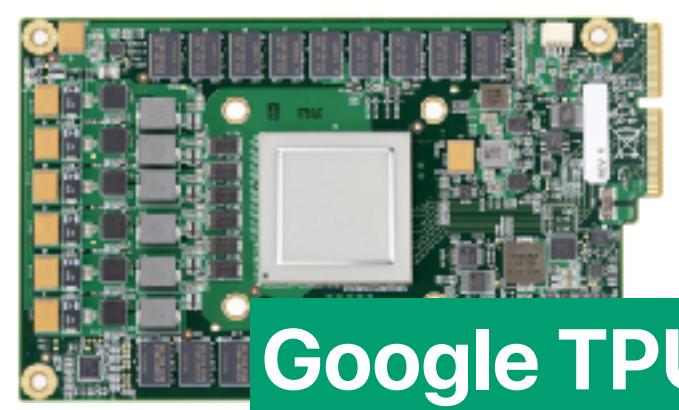


Nvidia Tegra 3
(2011)
5 cores

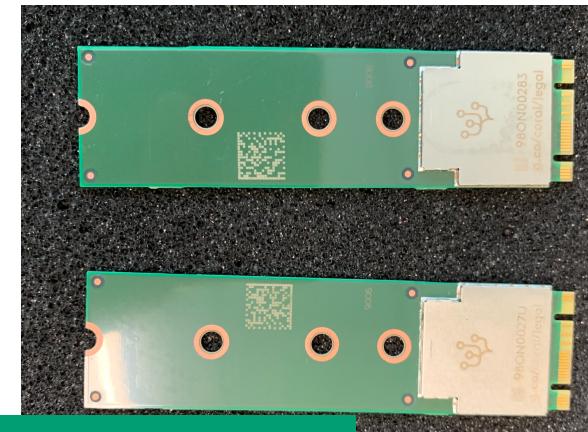


AMD Zambezi
(2011)
16 cores

The explosion of AI/ML accelerators



Google TPUv1



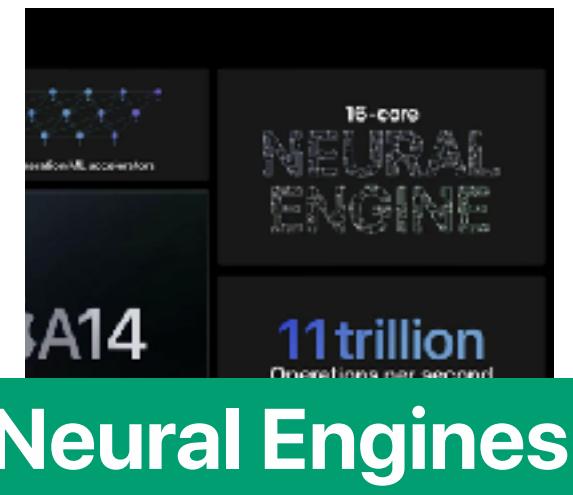
Edge TPUs



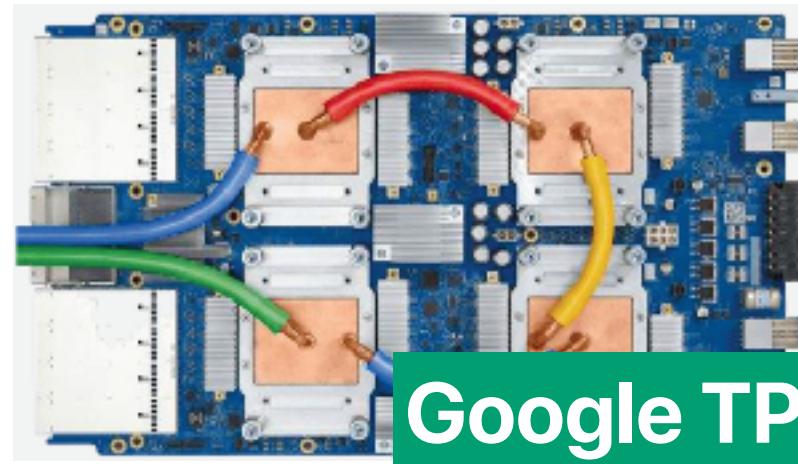
Google TPUv2



Qualcomm Hexagon



A14
Apple Neural Engines

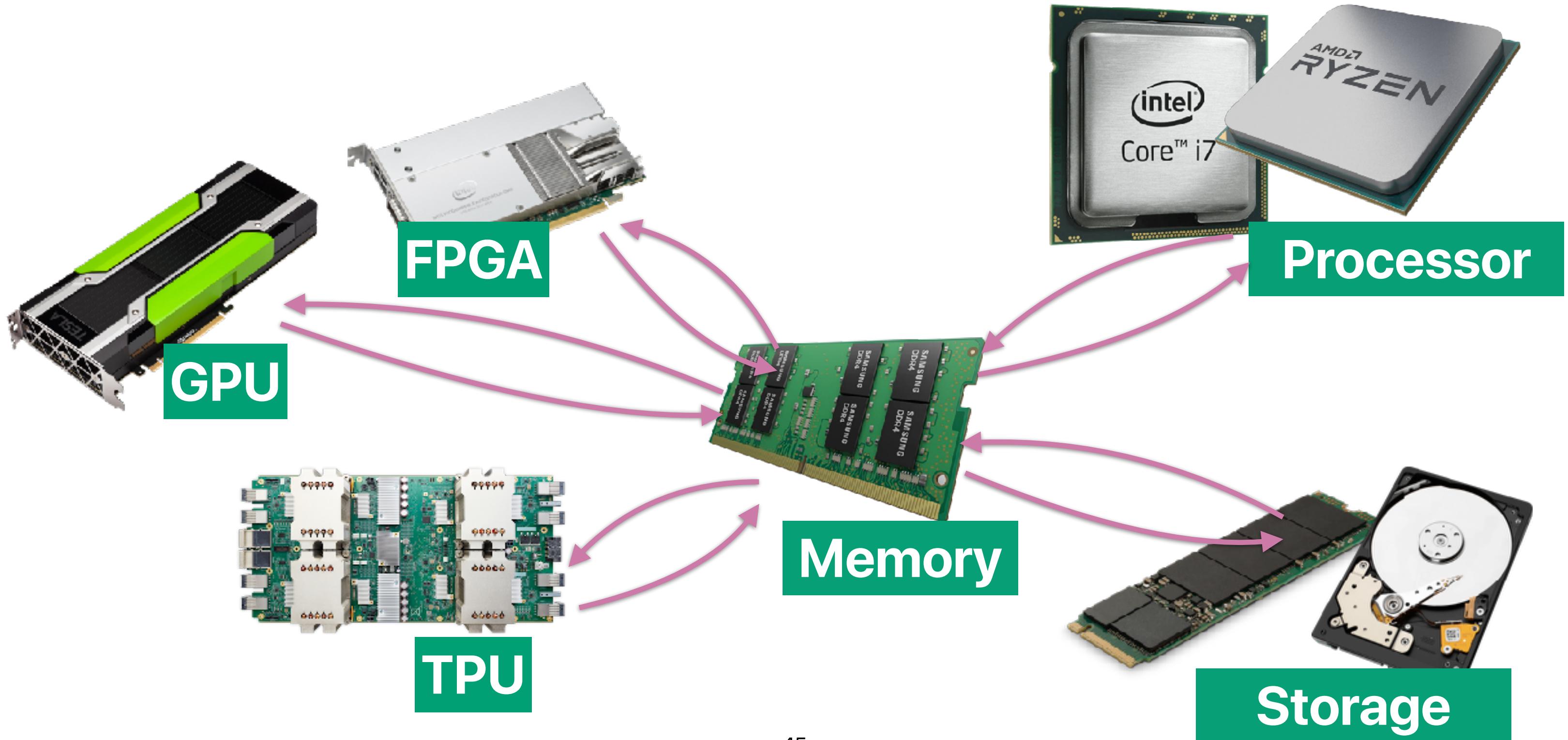


Google TPUv3

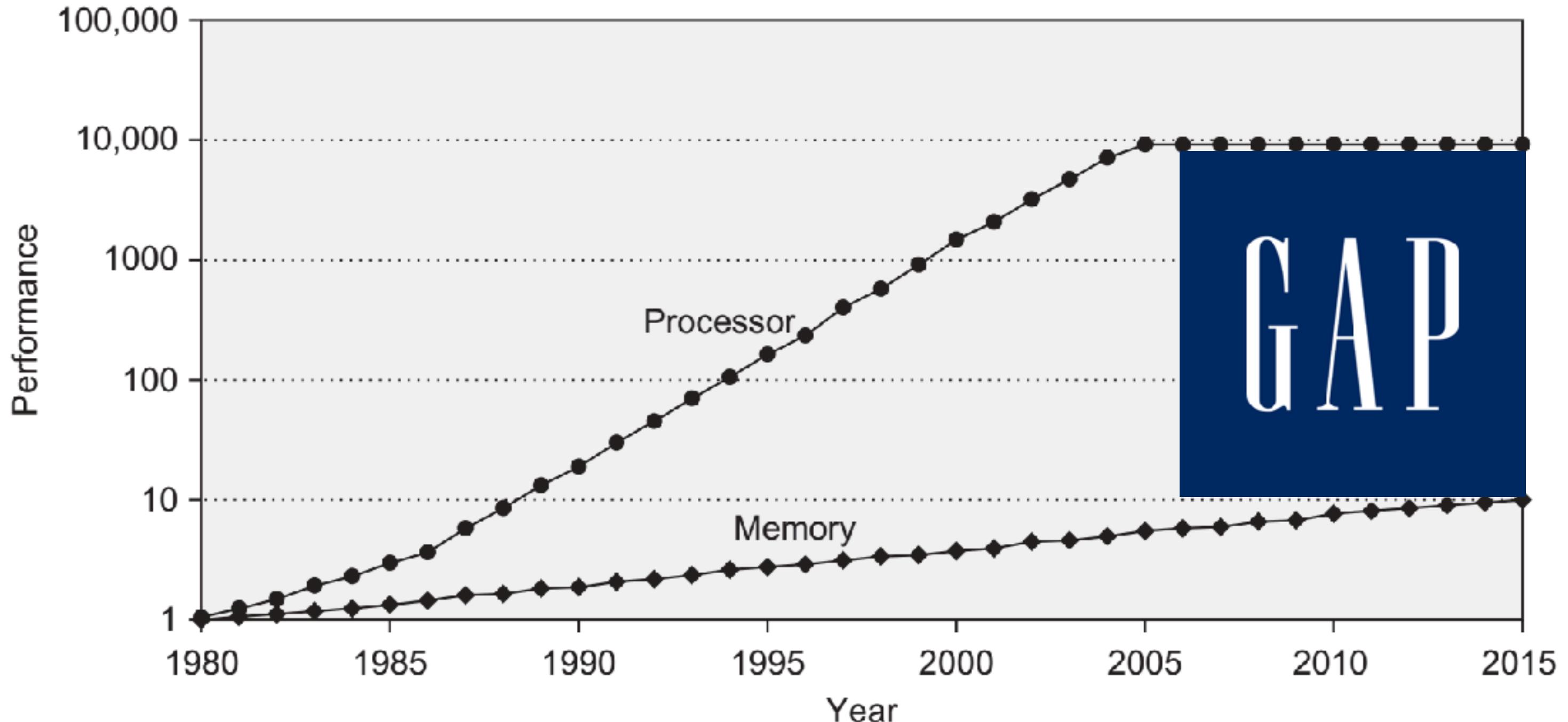


NVIDIA
Tensor Cores

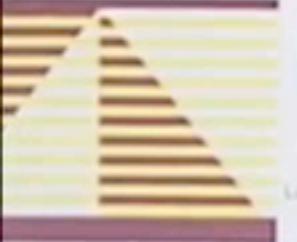
Heterogeneous computer architecture



Processor/memory performance gap

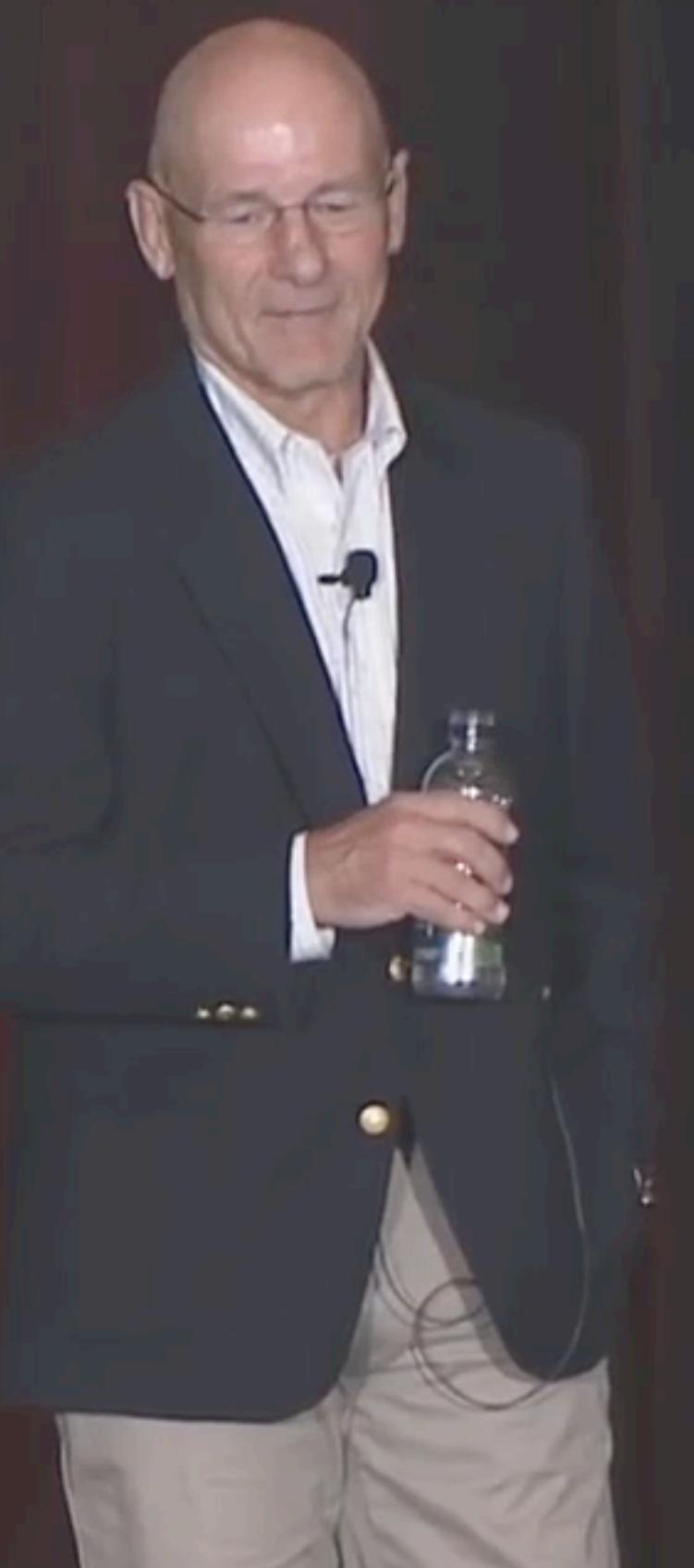
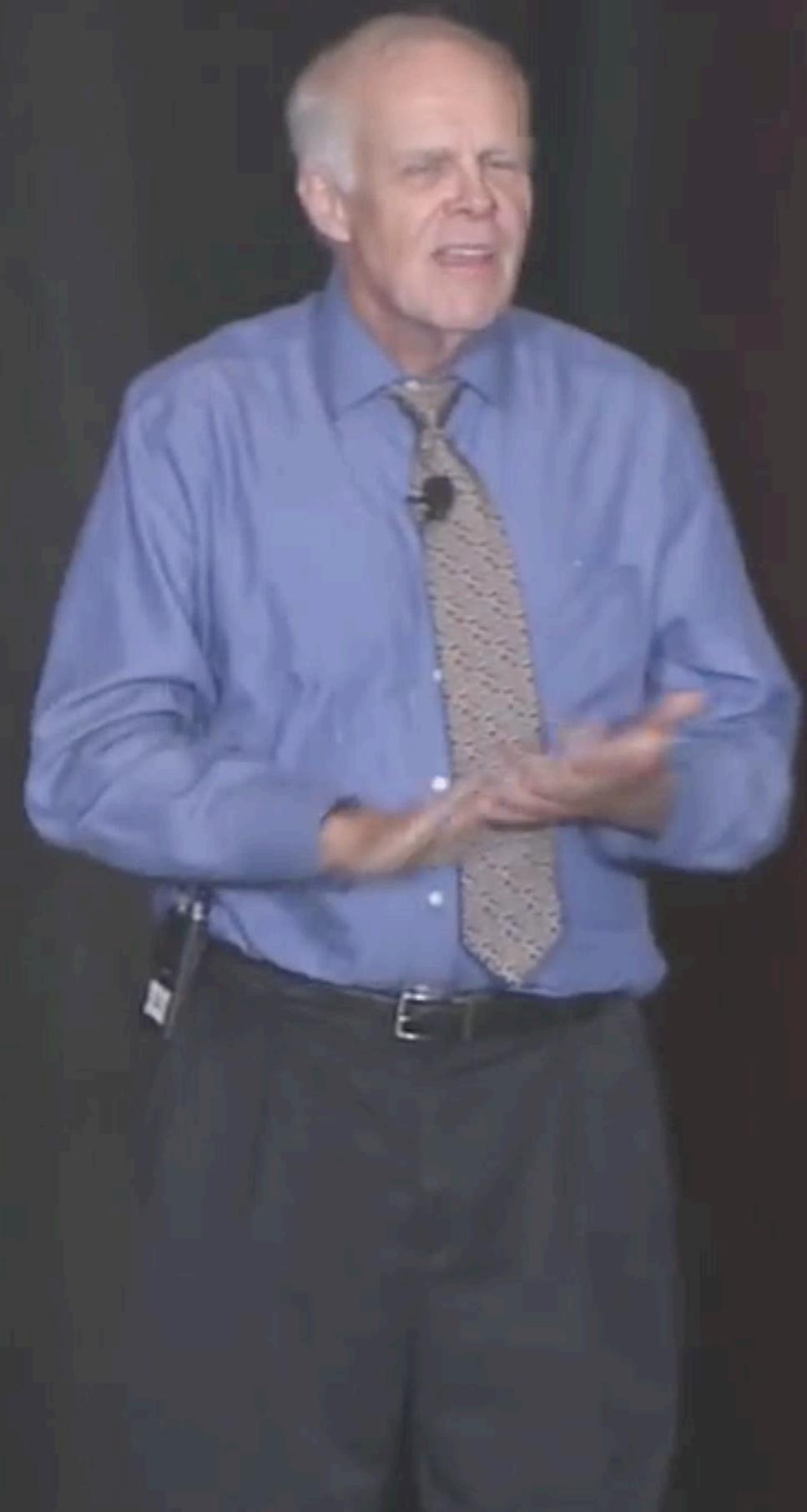


**I just want to writing programs, why
should I care about computer
architectures?**



The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA

ISCA 2018
uring Lecture



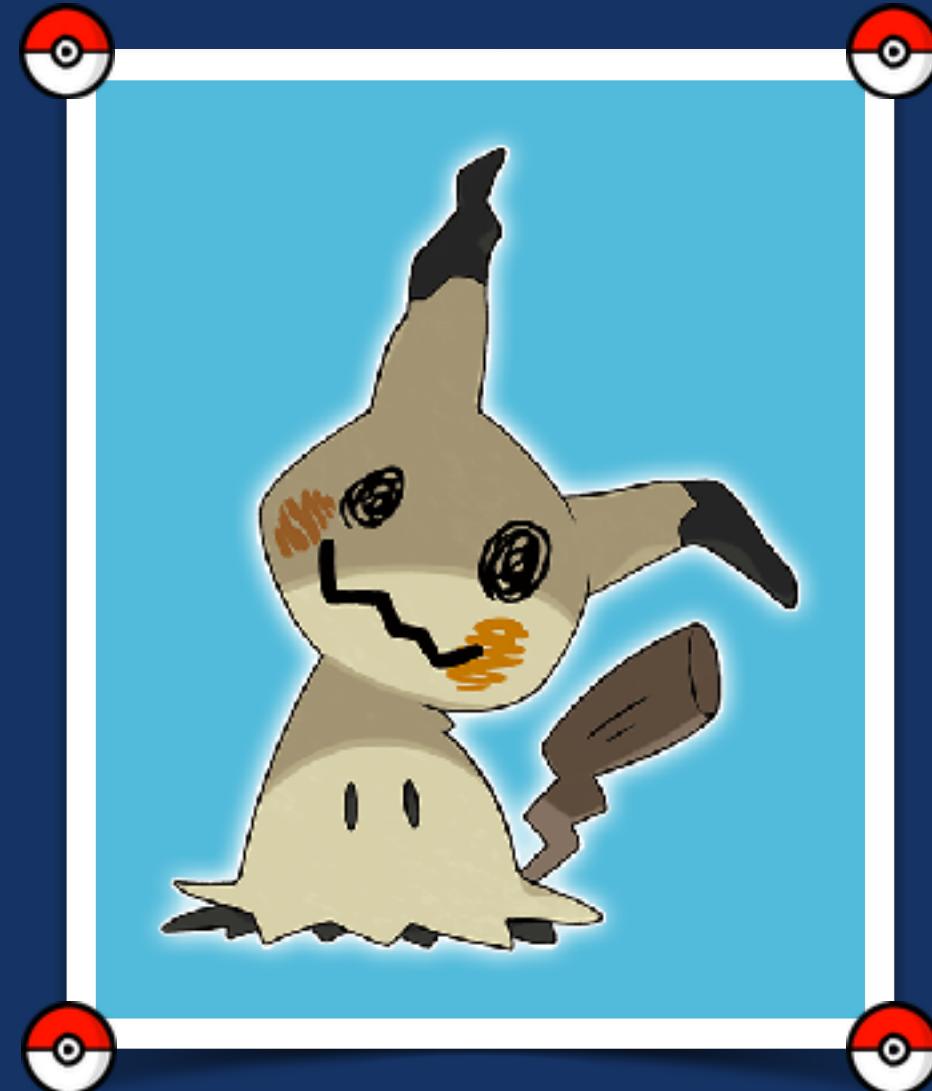
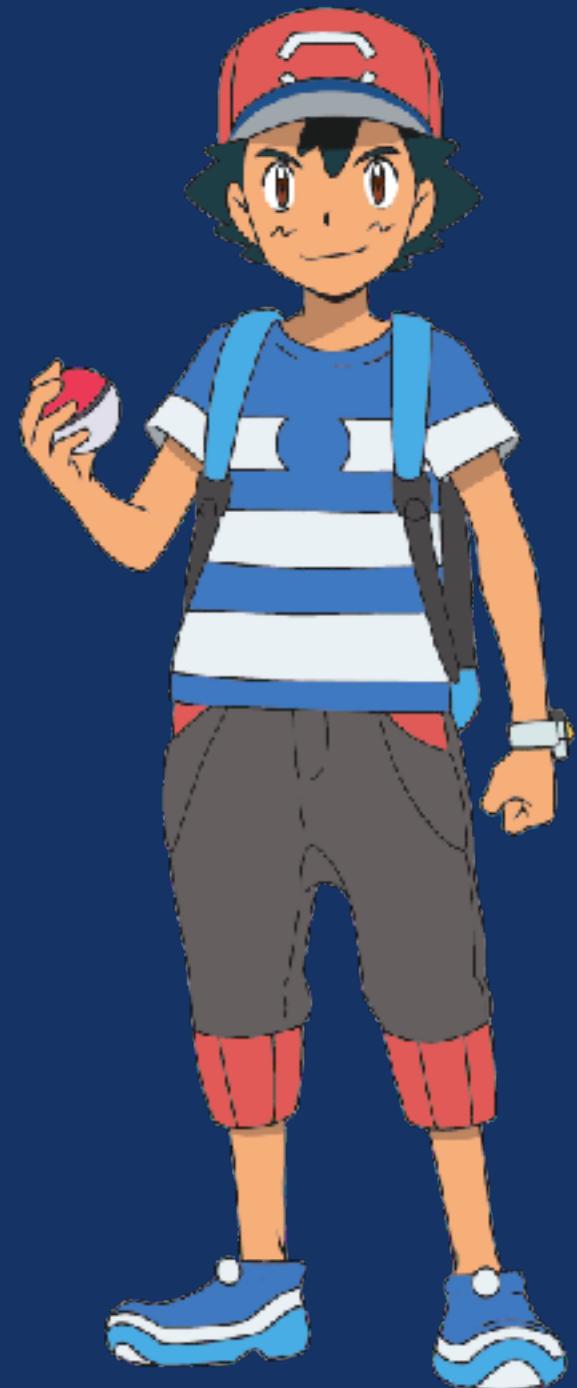
Demo

```
if(option)
    std::sort(data, data + arraySize);      O(nlog2n)
for (unsigned c = 0; c < arraySize*1000; ++c) {
    int t = std::rand();
    if (data[c%arraySize] >= t)            O(n)
        sum++;
}
if option is set to 1: O(nlog2n)
```

otherwise, O(n): *O(n*)



Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app



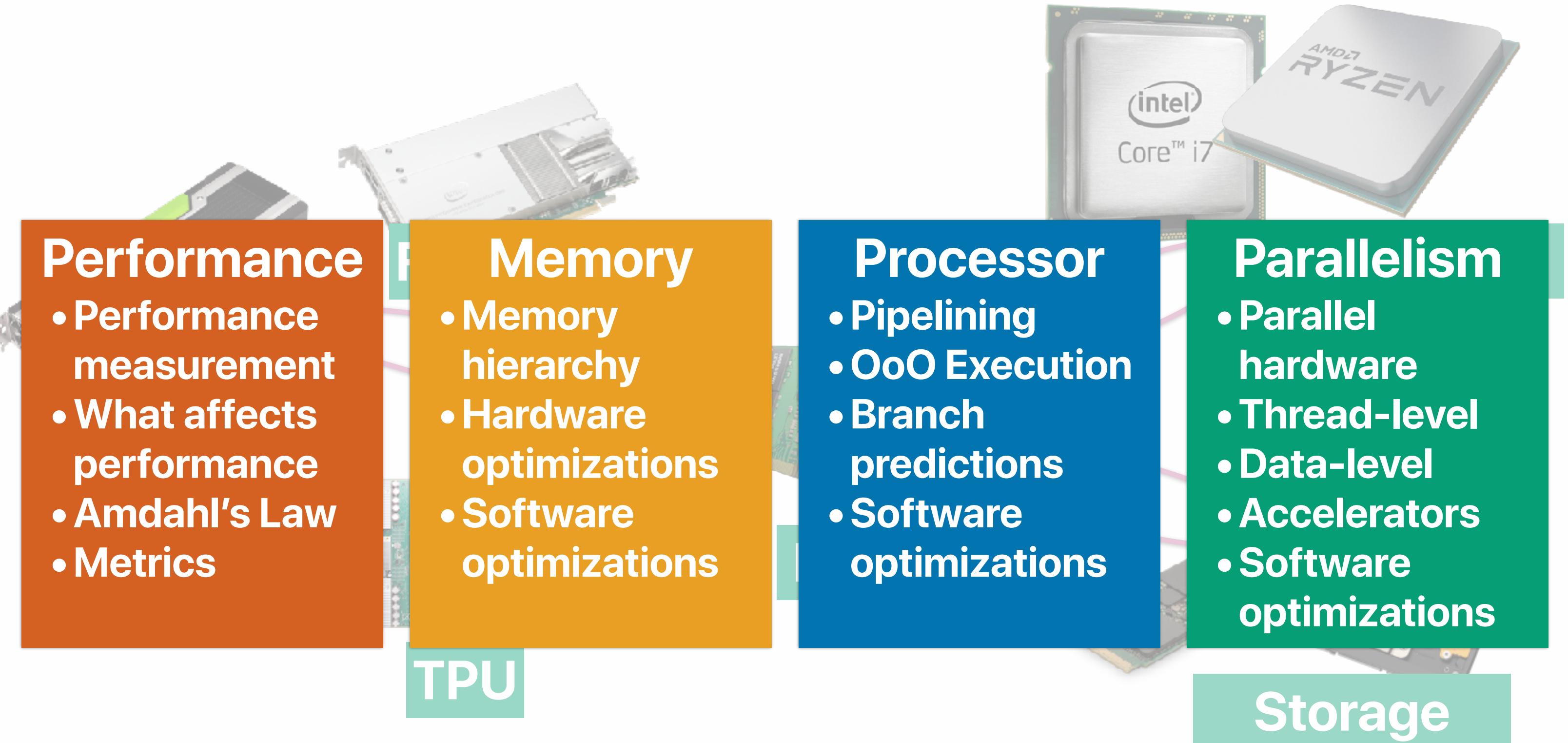
?????





**In CS203, we're teaching you to
write fast code (not code fast)**

Heterogeneous computer architecture



Tentative schedule

	Topic	Reading	Due
4/4/2023	Introduction	- G.E. Moore. Cramming More Components Onto Integrated Circuits. Electronics, pp. 114–117, April 19, 1965. - Chapter 1.1-1.6	
4/6/2023	Performance Evaluation (I)	- Chapter 1.3 & 1.8-1.9	Reading Quiz #1
4/11/2023	Performance Evaluation (II)		Reading Quiz #2
4/13/2023	Performance Evaluation (III)	- M. D. Hill and M. R. Marty. Amdahl's Law in the Multicore Era. in Computer, vol. 41, no. 7, pp. 33-38, July 2008. - V. Sze, Y.-H. Chen, T.-J. Yang and J. S. Emer. How to Evaluate Deep Neural Network Processors: TOPS/W (Alone) Considered Harmful. In IEEE Solid-State Circuits Magazine, vol. 12, no. 3, pp. 28-41, Summer 2020. - (Optional) Andrew Davison, Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers. in Proceedings of the International Conference on Parallel Processing (ICPP), 2005.	Assignment #0
4/18/2023	Memory Hierarchy (1): The		Reading Quiz #3
4/20/2023	Memory Hierarchy (2)	- Appendix B.1-B.3	Assignment #1
4/25/2023	Memory Hierarchy (3):	- Appendix B.1-B.3, 2.3	Reading Quiz #4
4/27/2023	Memory Hierarchy (4):	- Chapter 2.3 - Norman P. Jouppi. 1990. Improving direct-mapped cache performance by the addition of a small fully-associative cache and	
5/2/2023	Virtual Memory		Reading Quiz #5
5/4/2023	Virtual Memory (2)	- Chapter B.4 & B.5, 2.4	Assignment #2
5/9/2023	Midterm		
5/11/2023	Basic Processor Design &	- Chapter 3.3 - Appendix C.1 – C.4	Reading Quiz #6
5/16/2023	Advanced Branch Prediction	- M. Evers, S. J. Patel, R. S. Chappell and Y. N. Patt, "An analysis of correlation and predictability: what makes two-level branch predictors work," Proceedings. 25th Annual International Symposium on Computer Architecture (Cat. No.98CB36235), Barcelona,	Reading Quiz #7
5/18/2023	OOO Scheduling	- Chapter 3.4	
5/23/2023	OOO Scheduling	- K. C. Yeager, "The MIPS R10000 superscalar microprocessor," in IEEE Micro, vol. 16, no. 2, pp. 28-41, April 1996. - R. E. Kessler, "The Alpha 21264 microprocessor," in IEEE Micro, vol. 19, no. 2, pp. 24-36, March-April 1999.	Reading Quiz #8
5/25/2023	OOO Scheduling		Assignment #3
5/30/2023	SMT & Chip Multiprocessors	- Chapter 3.11 - Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor, Dean M. Tullsen, David O'Hallaron, and Eric M. Baskin, in IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 1, pp. 1-12, Jan 2001.	Reading Quiz #9
6/1/2023	Chip Multiprocessors	- The case for a single-chip multiprocessor, Kunle Olukotun, Basem A. Nayfeh, Lance Hammond, Ken Wilson, and Kunyung Chang, SIGPLAN Not. 31(9), pp. 1-10, 1996.	
6/6/2023	Dark Silicon	- Chapter 1.7	Reading Quiz #10
6/8/2023	TPU, FPGA	- H. Esmaeilzadeh, E. Blehm, P. St. Amant, K. Sankaralingam and D. Burger. Dark Silicon and the End of Multicore Scaling. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 1-12, Jan 2013. - Adrian W. Pouliasis, Zhi-Cai Chang, Andrew Putnam, Ivan Ingstrup, Jeremy Fosso, Michael Gascoin, Stephan Heil, Matt Humphrey, Puneet Kaur, Joo-Young Kim, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Pajerowski, Luis Verde, Sita Ramamurti, and	Assignment #4
6/15/2023	Subject to change	Final Exam	

You need to complete the Reading and Download Slides/papers Check due dates here

7pm-10pm

CS203: reinventing your Learning eXperience

Active learning: in-person lecture experience

We're back fully in-person

fortune.com/2023/03/07/google-sundar-pichai-staff-office-ghost-towns-microsoft-cloud/    

INE RANKINGS  MAGAZINE  NEWSLETTERS PODCASTS MORE  SEARCH SIGN IN

TECH · GOOGLE

Google boss Sundar Pichai says staff are bemoaning office ghost towns—‘It’s just not a nice experience’

BY CHRISTIAAN HETZNER March 7, 2023 at 4:56 AM PST

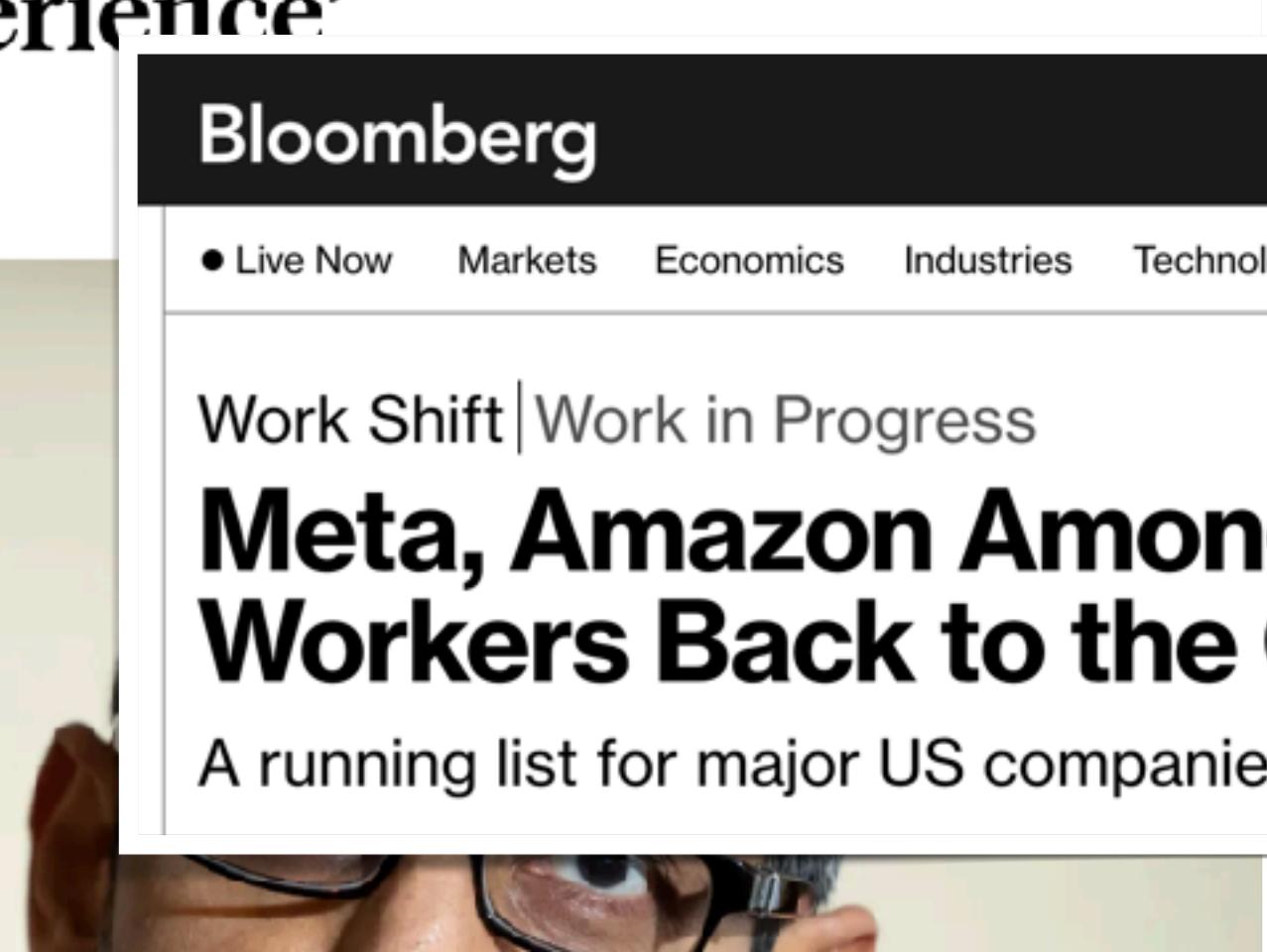
Bloomberg 

• Live Now Markets Economics Industries Technology Politics Wealth Pursuits Opinion Businessweek

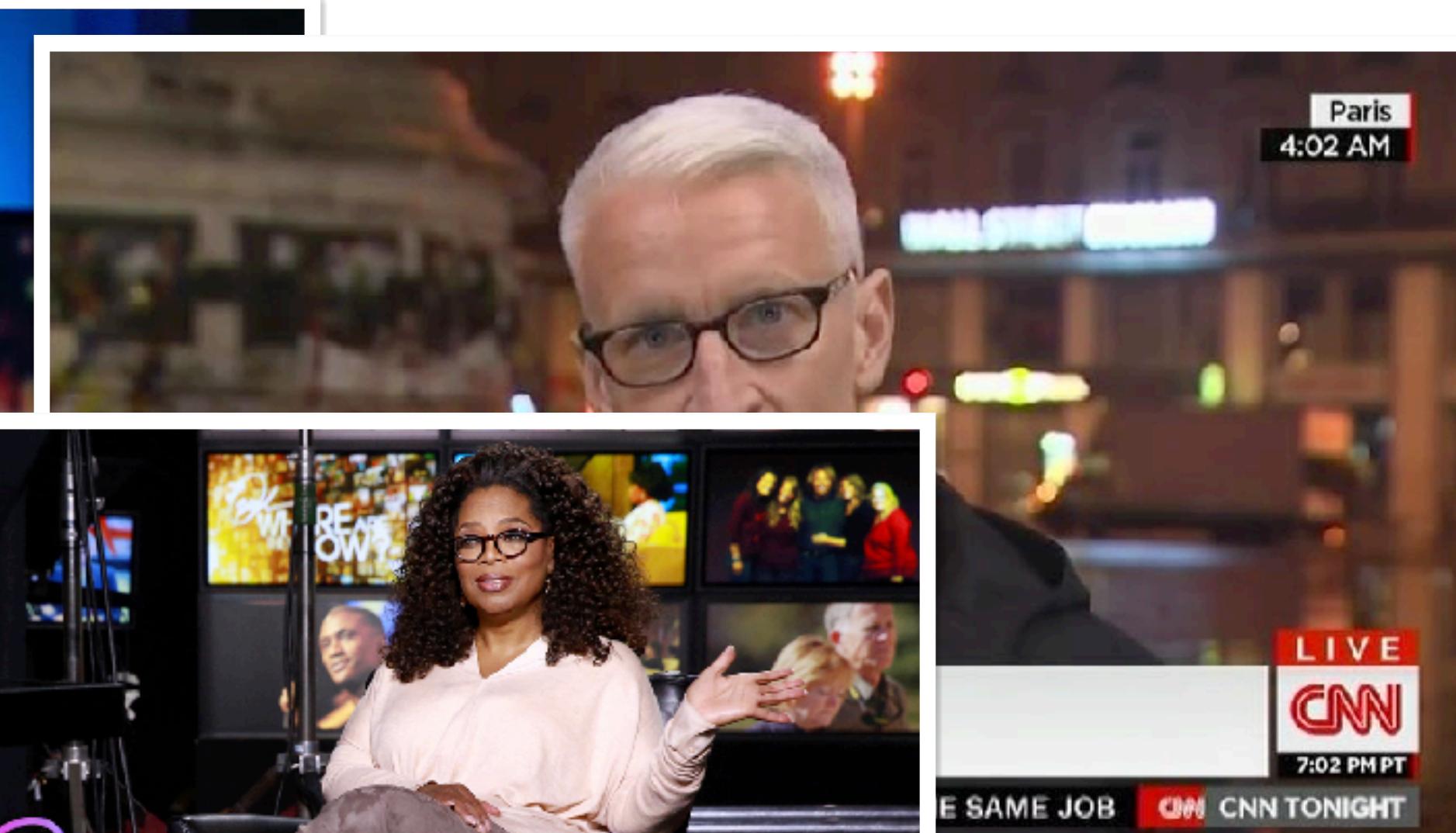
Work Shift | Work in Progress

Meta, Amazon Among Firms Calling Workers Back to the Office in 2023

A running list for major US companies' return-to-office mandates in 2023



Conventional lectures



Me

I expect the lecture to be...

You



Peer instruction

- Before the lecture — You need to complete the required **reading**
- During the lecture — I'll bring in activities to ENGAGE you in exploring your understanding of the material
 - Popup questions
 - Individual **thinking** — use your clicker to express your opinion
 - Group discussion — **discuss** with your surroundings and use your clicker to express your group's opinion
 - Whole-classroom **discussion** — we would like to hear from you

Read

Think

Discuss

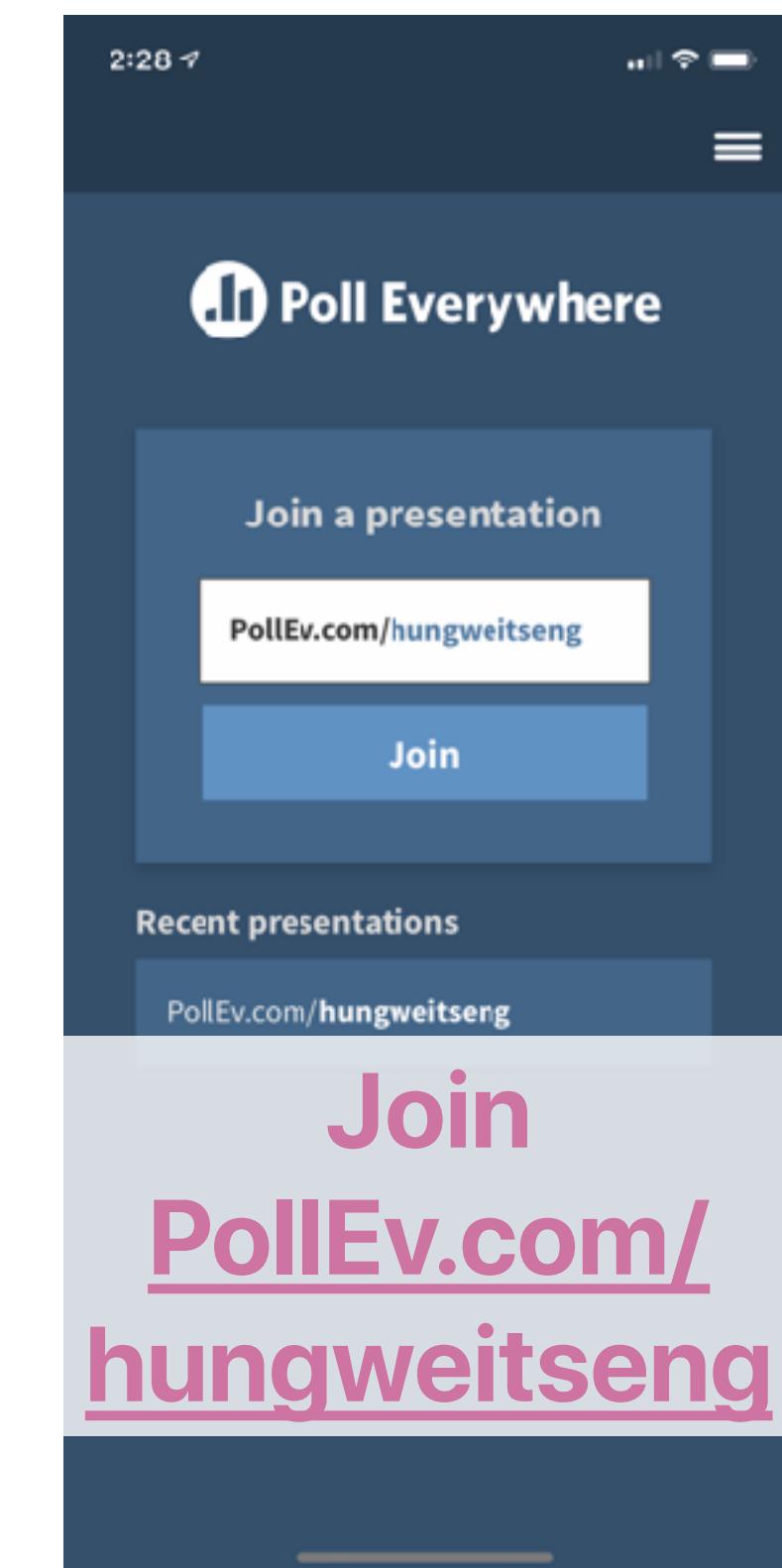
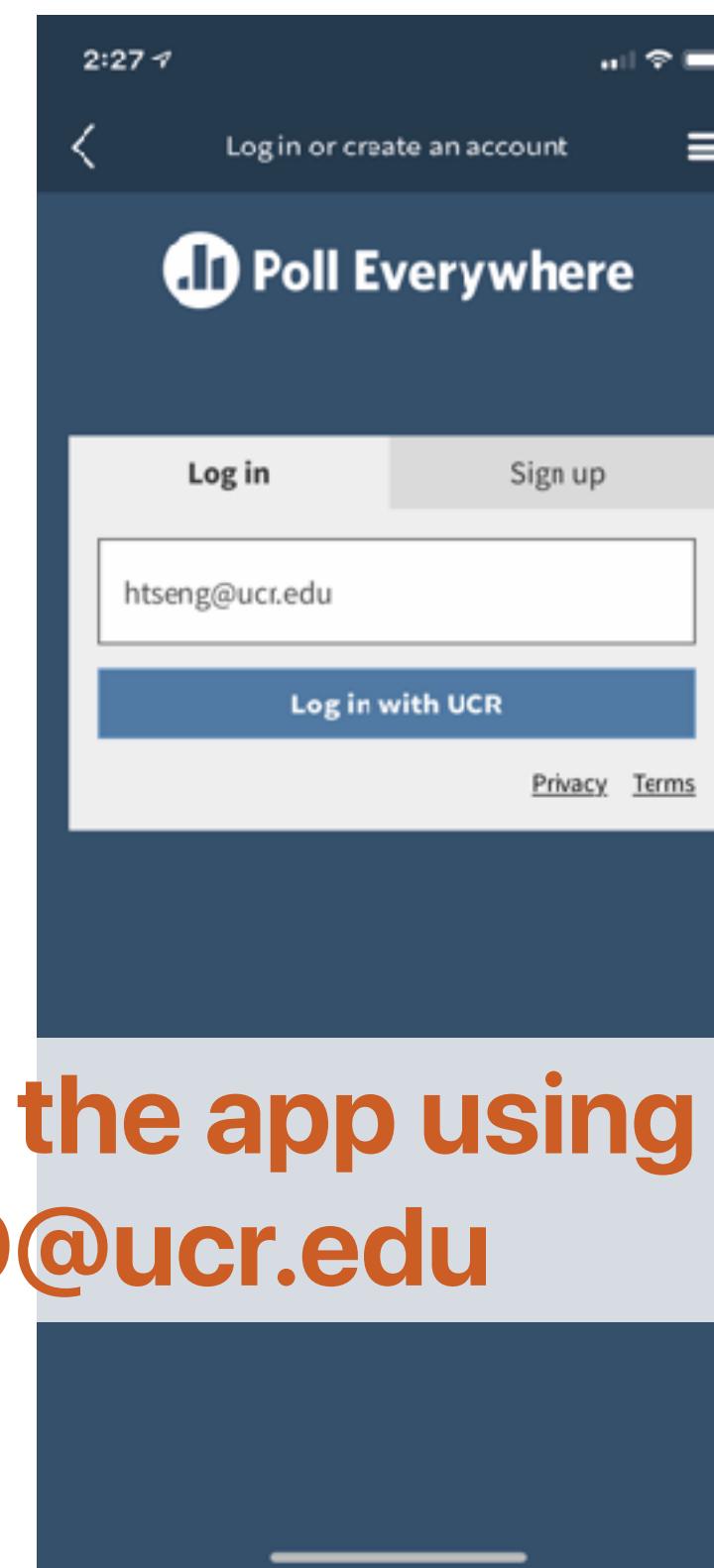
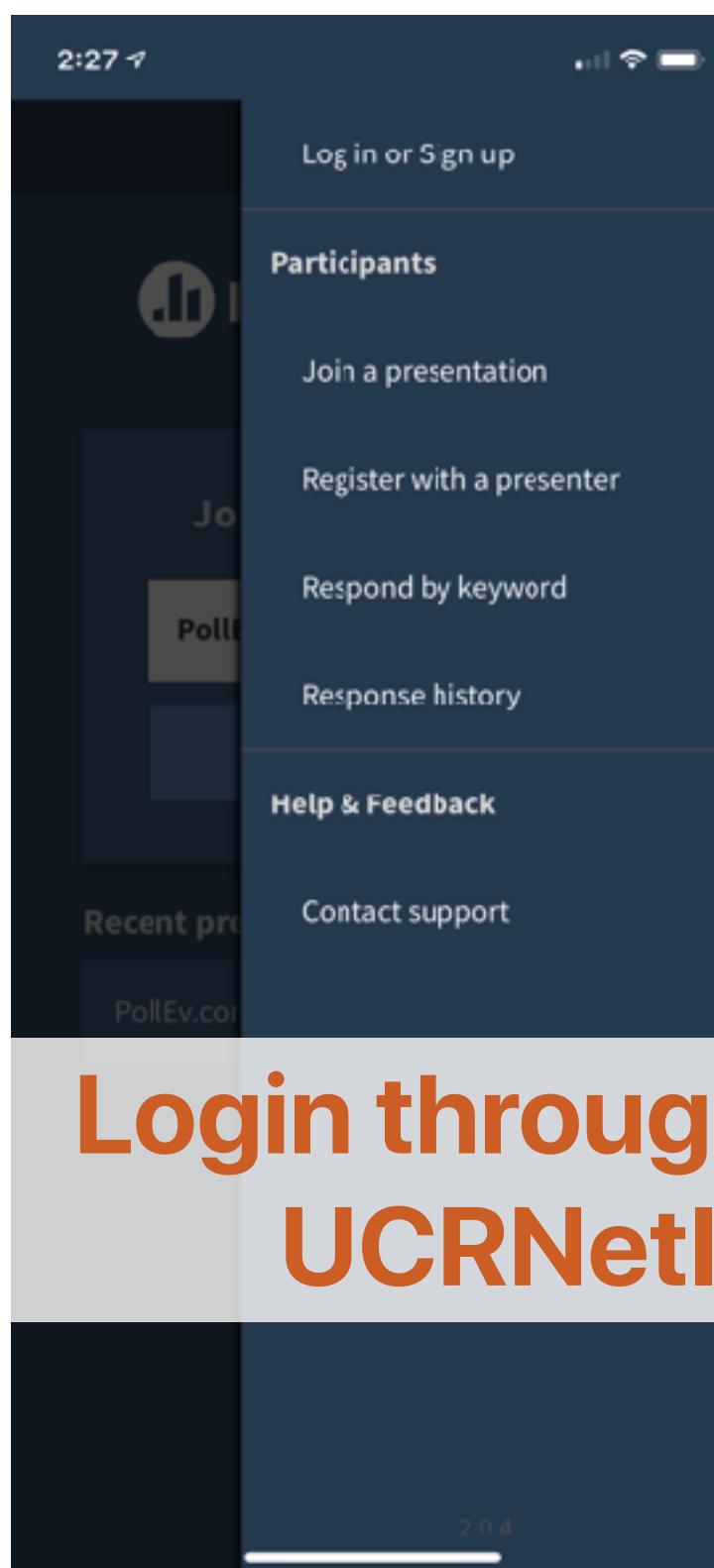
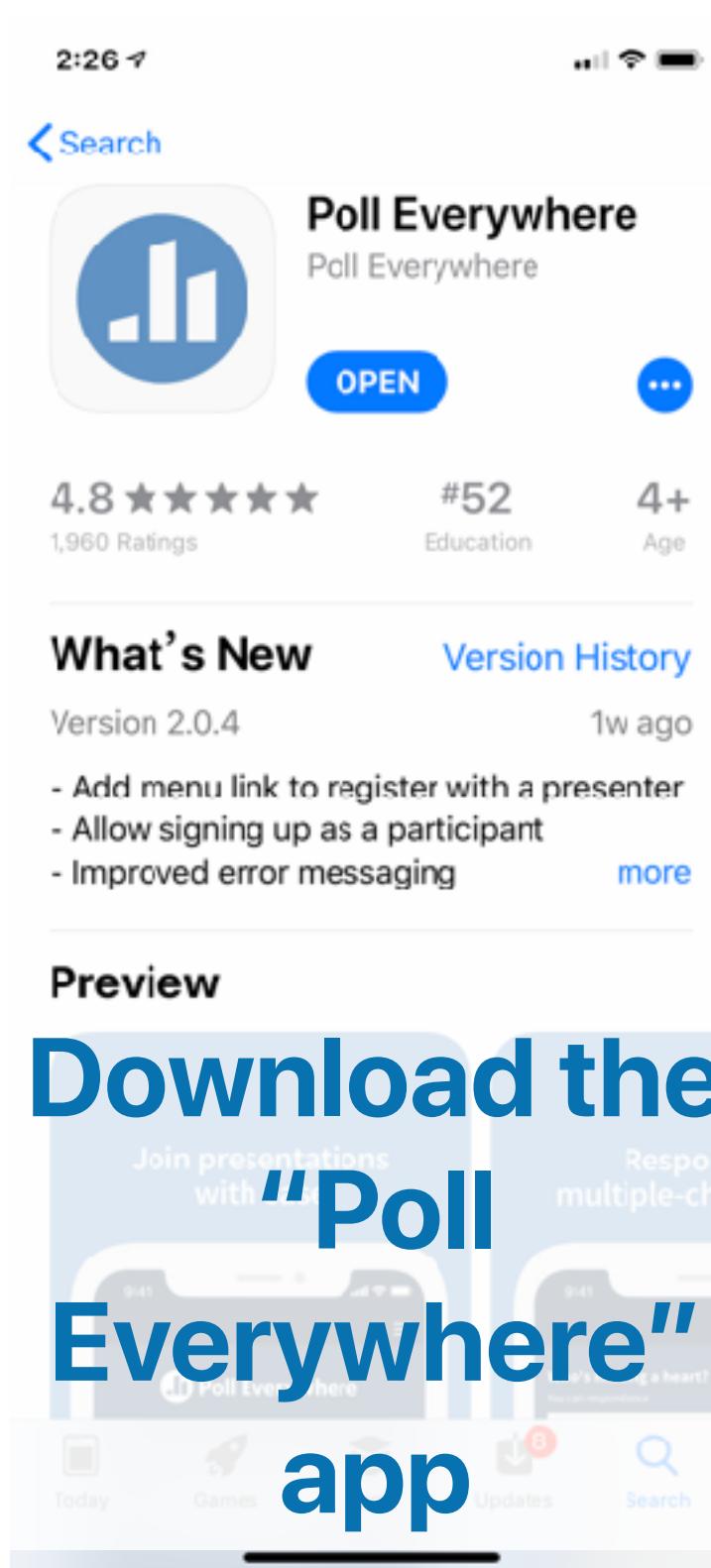
Peer instruction

- I'll bring in activities to ENGAGE you in exploring your understanding of the material
 - Let you practice
 - Bring out misconceptions
 - Let us LEARN from each other about difficult parts.
- You will be GET CREDIT for your efforts to learn in class
 - By answering questions with **Poll Everywhere**
 - Answer **50%** of the **clicker questions** in class, get **5%** of your final grade
 - Typically more than 50% of questions are individual thinking questions as individual thinking comes first
 - If you don't feel comfortable to talk with others, you can still get full credits if you made choices on all individual thinking questions

Why reading quizzes?

- We need to prepare you for peer instruction activities and discussions!
- Reading assignments from
 - Computer Architecture: A Quantitative Approach (6th Edition)
by John Hennessy and David Patterson
 - Papers
- Reading quizzes:
 - On eLearn
 - Due before the lecture, usually once a week. Check the schedule on our webpage — the first one due this Thursday
 - You will have two chances. We take the average
 - No time limitation until the deadline
 - No make up reading quizzes — we will drop two lowest at least

About the time of the Lecture — Setup Poll Everywhere



Grading breakdown

- Login/discussion in eLearn and piazza.
- Read the text before class!
 - Computer Architecture: A Quantitative Approach (6th Edition) by John Hennessy and David Patterson — previous editions are not supported
 - I'm not going to cover everything in class, but you are responsible for all the assigned text.
 - Papers
- Reading quizzes in eLearn (15%) — will drop the lowest two
- Class participation (5%)
- Assignment throughout the course. (25%) — will drop the lowest one
- Midterm (20%)
- Cumulative final (35%)

Jupyter notebook based assignment — learning & reviewing by practicing

For every assignment

- Go to course home page:
<https://www.escalab.org/classes/cs203-2023sp/>
- Click invitation link for the upcoming assignment
- Log into <https://www.escalab.org/datahub>
- Select this option and click “Launch Environment”
- Open a terminal
- Clone your starter repo.
- Open up Assignment.ipynb
- Follow the instructions

Course Infrastructure: Github and github classroom

- We will use github classroom to distribute starter code for the labs
- You'll use git/github to manage revisions etc.
- Github classroom is easy to use
- Git can be complex, but the basics are enough for this class.

Course Infrastructure: Bare metal Servers in The Cloud

- We will do a lot of measurement in this class
 - Program performance
 - Program energy/power consumption
 - Detailed hardware behavior
- All of this requires “bare metal” servers
 - Bare metal – no virtualization
 - Full access to underlying processors (esp. performance counters)
- The Jupyter Notebook (and the autograder) run your code on some bare metal servers “in the cloud” (actually a UCR computer somewhere)

Course Infrastructure: Jupyter Notebook

- A large part of each lab is done in a Jupyter Notebook
 - Jupyter Notebook is a web-based, interactive computing environment
 - It's good for collecting and visualizing data
 - Google's Colab is based on Jupyter Notebook
- If you haven't used Jupyter Notebook before...
 - That's fine. It's not that hard.
 - We'll be accessing Jupyter Notebook via CS203's dedicated jupyterhub server <https://www.escalab.org/datahub>

ASSIGNMENT.ipynb

Launcher

Assignment.ipynb

Python 3 (ipykernel)

<> M

1. Assignment 1: The Performance Equation

2. FAQ and Updates

3. Browser Compatibility

4. About Assignments In This Class

4.1. How To Succeed On the Assignments

4.2. Getting Help

4.3. Posting Answerable Questions on Piazza

4.4. Keeping Your Lab Up-to-Date

4.5. How To Use This Document

4.5.1. Running Code

4.5.2. Telling What The Notebook is Doing

4.5.3. What to Do Jupyter Notebook It Gets Stuck

4.5.4. Common Errors and Non-Errors

4.5.5. Useful Tips

3200MHz/1550Mhz ratio

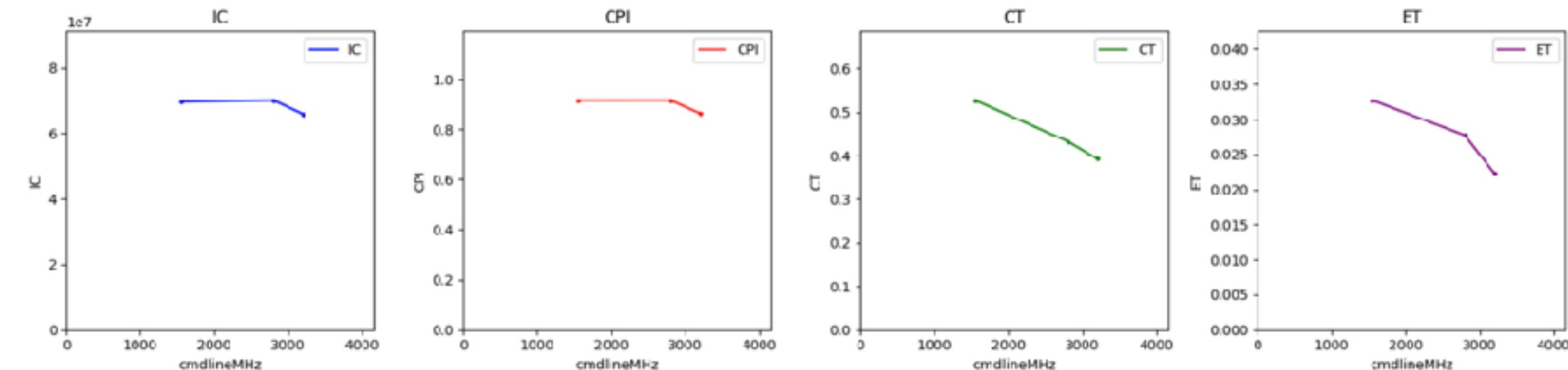
[71]: !lcs203 job run './microbench.exe -o cycle_time.csv -M 1550 2800 3200 -f baseline_int -r 50'

srun -N1 ./microbench.exe -o cycle_time.csv -M 1550 2800 3200 -f baseline_int -r 50

Execution complete

[72]: plotPE("cycle_time.csv", lines=True, what=[("cmdlineMHz", "IC"), ("cmdlineMHz", "CPI"), ("cmdlineMHz", "CT"), ("cmdlineMHz", "ET")])
render_csv("cycle_time.csv", columns=columns, average_by="cmdlineMHz")

	ET	IC	CPI	MHz	CT
cmdlineMHz					
1550	0.032720	69737541.58	0.915523	2128.051698	0.526995
2800	0.027738	70163970.86	0.915626	2328.057450	0.432629
3200	0.022307	65951524.30	0.863289	2546.401622	0.392715



In each assignment

- You should expect 20-30 questions per assignment (except for the 0th one).
- Correctness
 - Demonstrate mastery
 - Give the right answer — earn points
- Completeness
 - “forcing function” to get you to engage with the material
 - Give an answer — earn points
 - We will grade ~5 of these at random per assignment.
- Optional
 - Optional material for interested students.
 - Give the right answer — earn a sense of personal accomplishment

Course Infrastructure: cs203

- ‘cs203’ is a command line tool you’ll use to run your code.
 - It can do several things.
 - The most common is something like
- cs203 job run “echo hello world”
 - Submit your command to a job scheduling system to the dedicated cluster
 - Each task has limited amount of time to execute
 - It looks mostly like “echo hello world” ran locally.

Submitting your assignment

- All assignments will mostly be submitted via gradescope
- Jupyter Notebook PDFs
 - PDF submission
 - Submitted via upload
- Programming assignments
 - Autograded
 - Submitted via github
- Post-lab survey
 - Embedded in the assignment as a google form.

Quick walk through of your 0th assignment

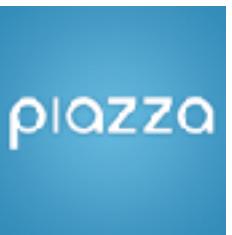
Warnings

- You only have about two weeks for each assignment
- Some exercises/demonstrations in a Jupyter Notebook
 - Interactive data collection and analysis.
 - Very little coding. Lots of thinking.
 - Worth a lot of points.
- A programming assignment
 - Write/modify some code to apply what you've learned.
 - Worth a lot of points.
- Post-assignment survey
 - Provide some feedback on the assignment.
 - Worth a few points

Logistics

Course resource

- Lectures:
 - TuTh 9:30a—10:50a @ INTN 1002
- Living streaming, video recording
<https://www.youtube.com/profusagi>
- Schedule, slides on **course webpage**:
<https://www.escalab.org/classes/cs203-2023sp/>
- Discussion on **piazza**:
https://piazza.com/ucr/spring2023/cs_203_001_23s/home
- Reading quizzes, check grades on **eLearn**:
<https://elearn.ucr.edu/courses/92451>
- Submit assignments on Gradescope:
<https://www.gradescope.com/courses/527055>



The website

CS203: Advanced Computer Architecture (Fall 2023)

- Schedule
- Slides
 - Preview — for the ease of note taking
 - Release — the actual slides
- Calendar for office hours & their locations
 - https://calendar.google.com/calendar/embed?src=ucr.edu_b8u6dvkretn6kq6igunlc6bldg%40group.calendar.google.com&ctz=America%2FLos_Angeles

Lecture: **TuTh 9:30a - 10:50a**

Where: **CHASS Interdisciplinary-North | Room 1002**

[Schedule and Slides](#)

[Assignments](#)

[Logistics](#)

Instructor

[Hung-Wei Tseng](#)

email: htseng @ ucr.edu

Office Hours: M 2p-3p W 3:30a-4:30p @ WCH 406

Teaching Assistant

Kuan-Chieh Hsu

e-mail: khsu037 @ ucr.edu

Office Hours: TuTh 3:30p-4:30p and W 10:30a-11:30a.

Other important links

- Reading Quizzes and Grading on [eLearn](#)

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- Visiting Researcher @ Google, 01/2023—03/2023
 - Working for TensorFlow Lite
- PhD in **Computer Science**, University of California, San Diego, 2014
- Research Interests
 - General-purpose computing on AI/ML/NN accelerators
 - Intelligent storage devices & near-data processing
 - Or anything else fun — we have an OpenUVR project recently



Teaching Assistant — Kuan-Chieh Hsu

- Office hours: TuTh 3:30p-4:30p &
W 10:30a-11:30a
- E-mail: khsu037 @ ucr.edu



Grading

- You can see your grades on eLearn.
- Errors in grading
 - If you feel there has been an error in how an assignment or test was graded, you have **one week** from when the assignment is returned to bring it to our attention.
 - You **MUST** submit (via email to the instructor AND the appropriate TAs) a written description of the problem. Neither I nor the TAs will discuss regrades without receiving an email from you about it first.
- For arithmetic errors (adding up points etc.)
 - you do not need to submit anything in writing, but the **one-week** limit still applies.

The screenshot shows the eLearn navigation bar on the left side of the screen. The bar includes links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. To the right of the navigation bar, the main content area displays the path 'CS_203_001 > Grades > Test Student'. Below this, a section titled 'Grades for Test Stu...' is visible, along with a dropdown menu for 'Arrange By' set to 'Due Date', and a list of assignments including 'Assignment 1', 'Assignment 2', 'Assignment 3', and 'Assignment 4'. The 'Grades' link in the navigation bar is circled with a red oval.

Academic Honesty

- Don't cheat.
 - Cheating on a test will get you an F in the class and no option to drop, and a visit with your college dean.
 - Cheating on homework means you don't have to turn them in any more, but you don't get points either. You will also take at least 25% penalty on the exam grades.
- Copying solutions of the internet or a solutions manual is cheating
 - They are incorrect sometimes
- Review the UCR student handbook
- When in doubt, ask.

Hey, I need help...

- Is question 3 on the assignment asking for execution time or speedup
 - piazza
- I'm lost on the assignment – I don't know what speedup or execution time are...
 - Office Hours (maybe discussion)
- I need to turn in the assignment late
 - No late assignment (you get to drop one)
 - Turn in what you have

Hey, I need help... (part 2)

- I'm going to miss class
 - Sorry to miss you! Please watch my youtube channel!
<https://www.youtube.com/profusagi>
 - In-depth class concept question (e.g., what's the difference between pass-by-value vs. pass-by-reference)
 - Class or Discussion (possibly piazza)
 - I can't get into eLearn and escalab.org/datahub
 - E-mail campus IT if you can't log in eLearn. Private piazza post if waitlisted and can't see course.
 - Contact Hung-Wei or Jinyoung if it's datahub related

Hey, I need help... (part 3)

- “I’m sick....”
 - We already allow you to miss 50% of classes, drop one assignment & two quizzes.
 - Issues impacting Midterm and Final require exceptional circumstances, e-mail professor
- Disability
 - E-mail paperwork from campus disability services to the Prof. by the end of week 2.

Hey, I need help ... (part 4)

- The autograder output contains an uncaught exception
- The autograder doesn't return a score.
- The autograder times out repeatedly.
- The autograder seems to be otherwise misbehaving.
 - Post on Piazza under "Autograder bugs".

Why...

- Do I really need the textbook
 - Again, we need to prepare you for lectures
 - Textbook helps to make sure we're all on the same page when we talk about something
- Why do we use so many different resources...
 - Gradescope does not allow multiple trials of reading quizzes...
 - Canvas' discussion feature isn't great and not accessible after you graduated
 - I hope some of you can start up a company to provide a one-fit-all solution...

Term of Service

- CS203 is an **Advanced Computer Architecture** class for graduate students. It's not our responsibility to recap everything that should be covered by an undergraduate computer architecture class from a regular computer science undergraduate program.
- This class requires **intensive readings** in research papers and the assigned textbook.
- This class requires you to **speak and discuss** your opinion with your classmates as well as the instructor.
- This class requires **programming projects** that uses the **C programming language**. It is **your responsibility to learn how to program in C**. It is also your responsibility to design the architecture, implementation details and tests for your coding projects.
- The instructor and course staffs reserve the right to refuse to answer inappropriate questions (e.g. directly telling if an answer is right or not).
- **It is your responsibility to track the latest schedule, information, grades and materials from our course website, e-mails from the course staffs and the piazza forum.**
- Any cheating will be treated seriously. You will get an F and we will report to the academic integrity office



By clicking this box, you are agreeing to the Terms and Conditions of CS 203, Fall 2022.

UC San Diego

NC STATE UNIVERSITY

UC RIVERSIDE



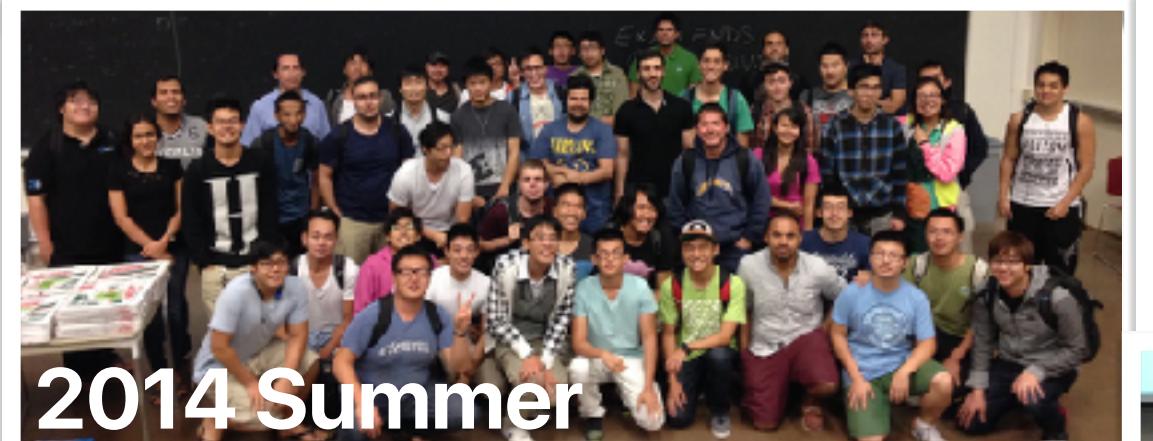
2012 Summer



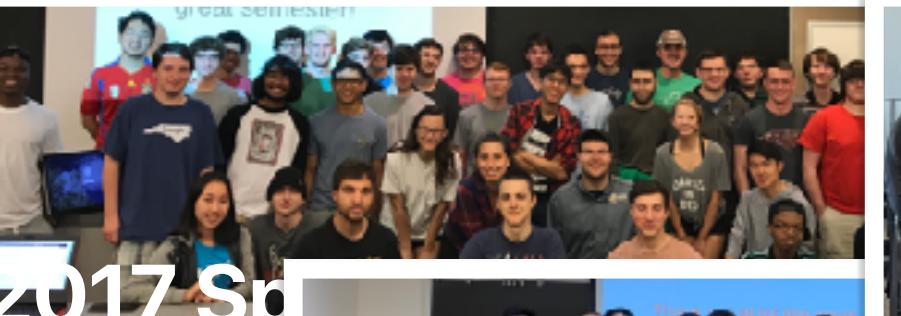
2016 Fall



2022 Winter



2014 Summer



2017 Sp



2022 Fall



2021 Fall



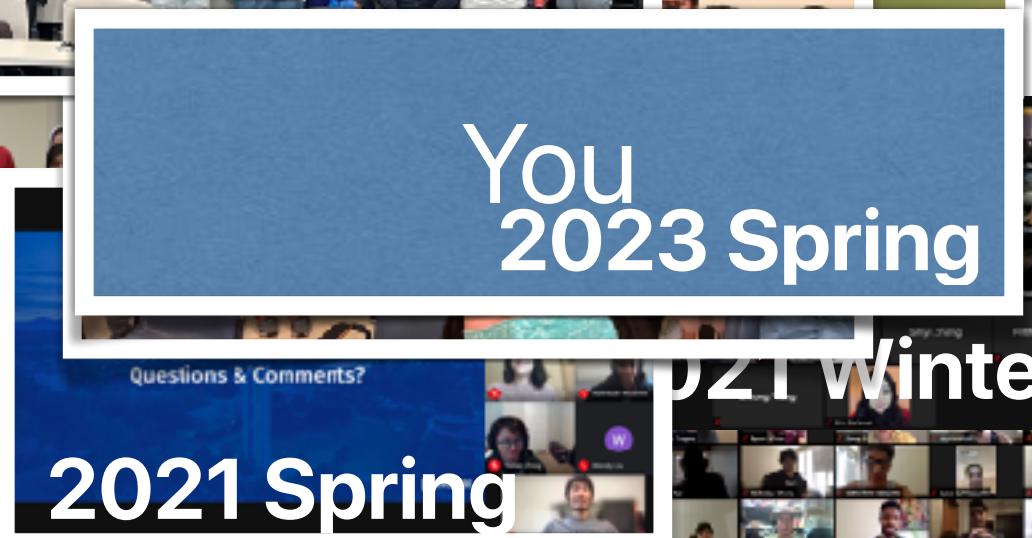
2016 Spring



2017 Fal



2019 Fall



2021 Spring



2021 Winter



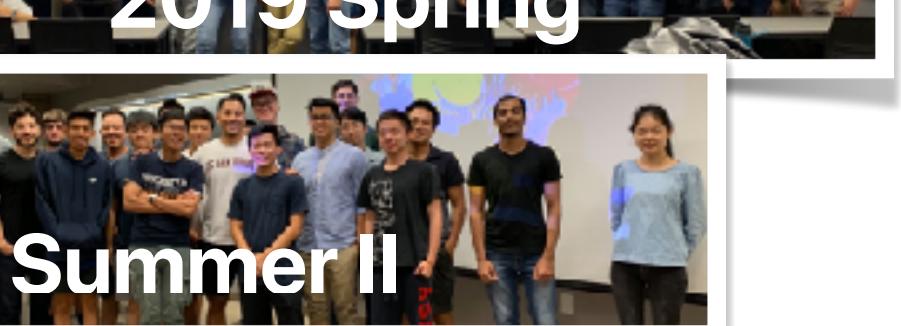
2016 Summer



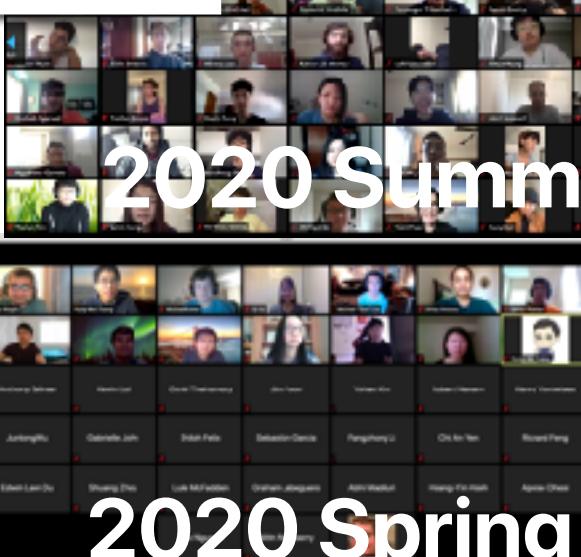
2018 Fall



2019 Summer I



2019 Summer II



2020 Spring



2020 Summer

Computer Science & Engineering

203

つづく

