

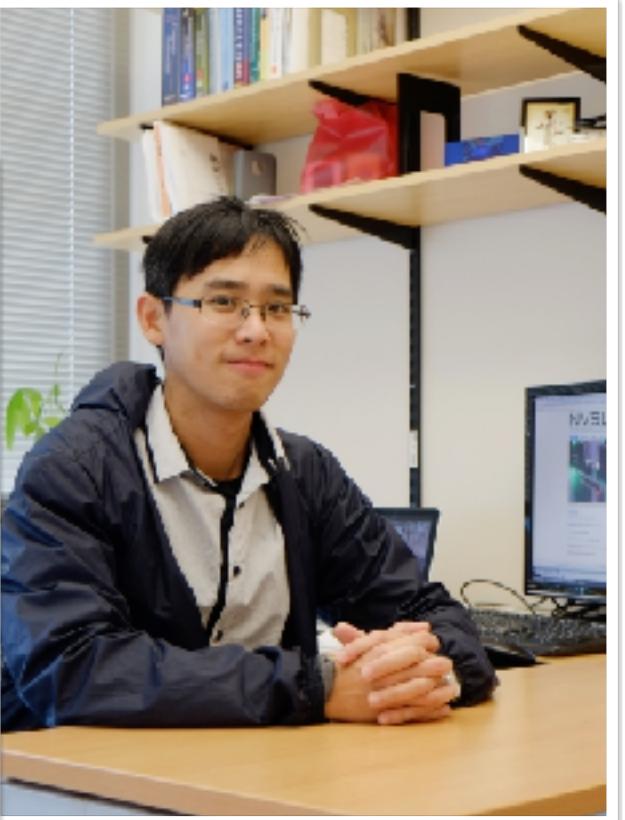
# **First Day of CS203: Advanced Computer Architecture (2024 Fall)**

Prof. Usagi a.k.a.  
Hung-Wei Tseng



# Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- E-mail: cs203 @ escal.org
- 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/RayTracing accelerators
  - Or anything else fun — we have an OpenUVR project recently
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial



ChatGPT 3.5

You  
What are the most popular topics in computer science?

ChatGPT

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

st popu

Gemini

See the latest updates to the Gemini Apps Privacy Hub

# Hello, Hung-Wei

How can I help you today?

Revise my writing and fix my grammar

Teach me the concept of game theory in simple terms

Help me plan a game night with 5 friends for under \$100

Your conversations are processed by human reviewers to improve the technologies powering Gemini Apps. Don't enter anything you wouldn't want reviewed or used.

How it works Dismiss

What are the most popular topics in computer science?

Gemini may display inaccurate info, including about people, so double-check its responses. [Your privacy](#)

Submit

# What do you care as a computer scientist?



Algorithms  
Data Structures  
**Computer Architecture**  
Programming Languages  
User Interfaces

ChatGPT 3.5

- o. **Internet of Things (IoT)**: IoT involves connecting everyday objects to the internet, enabling them to send and receive data. Topics include sensor networks, IoT devices, IoT platforms, and edge computing.
- 6. **Blockchain**: Originally developed for cryptocurrencies like Bitcoin, blockchain technology has broader applications in areas such as supply chain management, voting systems, and secure data sharing.
- 7. **Computer Vision and Image Processing**: With advancements in deep learning, computer vision has seen significant progress in areas such as object detection, image classification, and facial recognition.
- 8. **Quantum Computing**: Although still in its early stages, quantum computing has the potential to revolutionize computing by solving certain problems exponentially faster than classical computers. Topics include quantum algorithms, quantum cryptography, and quantum error correction.
- 9. **Software Engineering**: This encompasses various methodologies, tools, and practices for developing high-quality software systems. Topics include agile development, DevOps, software testing, and software architecture.
- 10. **Mobile Development**: With the widespread use of smartphones and tablets, mobile development remains an important area. Topics include iOS development, Android development, cross-platform development frameworks, and mobile UX/UI design.

These topics represent just a fraction of the diverse and dynamic field of computer science. New areas of research and innovation continually emerge as technology advances.

What are the most important topics in computer sciences?

ChatGPT can make mistakes. Consider checking important information.

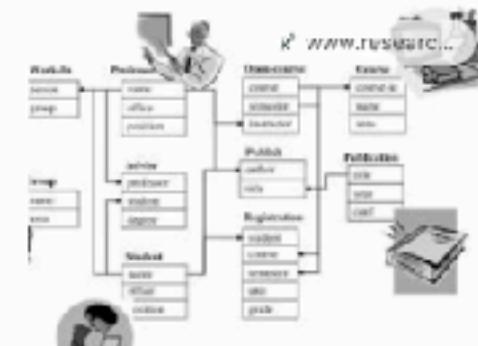
# t import

Gemini

See the latest updates to the Gemini Apps Privacy Hub

Computer Networking computer science

- **Databases and Big Data**: Databases are collections of organized data. Big data refers to the large and complex datasets that are being generated by modern businesses and organizations. Database professionals are responsible for designing, building, and maintaining databases, while big data specialists are responsible for analyzing and extracting insights from large datasets.



Databases computer science

These are just a few of the many popular topics in computer science. The field is constantly evolving, so there are always new and exciting things to learn.

Sources

1. [www.dailychina.news/showArticle?main\\_id...](http://www.dailychina.news/showArticle?main_id...)

What are the most important topics in computer sciences?

Gemini may display inaccurate info, including about people, so double-check its responses. [Your privacy & Gemini Apps](#)

## 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)



Sixth Edition

John L. Hennessy | David A. Patterson

# COMPUTER ARCHITECTURE

A Quantitative Approach



# Computer Architecture

Enables

Deep Learning

# Computer Architects — Hidden Figures

BASED ON THE UNTOLD TRUE STORY

# HIDDEN FIGURES



# **What is “Computer Architecture”?**

ChatGPT 3.5

threats, knowledge of security principles and cryptographic techniques is crucial for protecting data and systems from unauthorized access and attacks.

9. **Artificial Intelligence and Machine Learning:** AI and ML are transforming various industries and applications. Understanding concepts like neural networks, deep learning, reinforcement learning, and natural language processing is becoming increasingly important.

10. **Human-Computer Interaction (HCI):** HCI focuses on designing interactive systems that are user-friendly and efficient. Understanding user behavior, usability principles, and interaction design is essential for developing effective user interfaces.

These topics serve as foundational knowledge for computer scientists and are applicable across various domains and specialties within the field. Additionally, interdisciplinary areas such as bioinformatics, computational biology, and computational neuroscience are gaining importance, highlighting the interdisciplinary nature of computer science.

HT You  
What is computer architecture?

ChatGPT

Message ChatGPT...

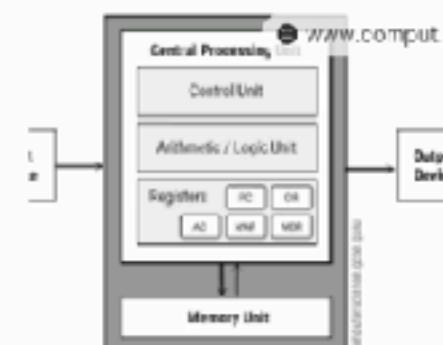
ChatGPT can make mistakes. Consider checking important information.

# puter a

Gemini

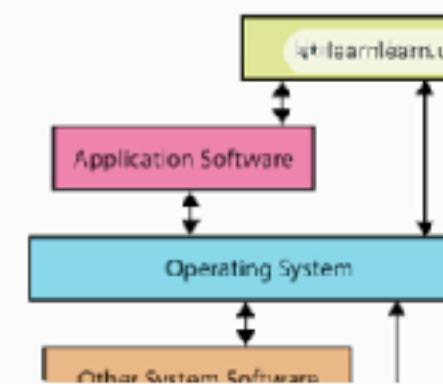
See the latest updates to the Gemini Apps Privacy Hub

well as the software that controls them. An understanding of computer architecture is essential for writing efficient programs and designing new computing systems.



Computer Architecture computer science

- Operating Systems: An operating system (OS) is the software that manages the hardware and software resources of a computer. It provides a platform for running other programs and applications. Operating systems are essential for the smooth operation of computers.



Operating Systems computer science

These are just a few of the most important topics in computer science. These foundational concepts are essential for understanding how computers work and for creating new technologies.

What is computer architecture?

Gemini may display inaccurate info, including about people, so double-check its responses. [Your privacy](#)

Submit

# What's computer architecture?



## architecture noun

ar·chi·tec·ture | \är-kuh-tek-chur \

**Definition of *architecture***

**1** : the art or science of building

*specifically* : the art or practice of designing buildings especially habitable ones

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

// *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*

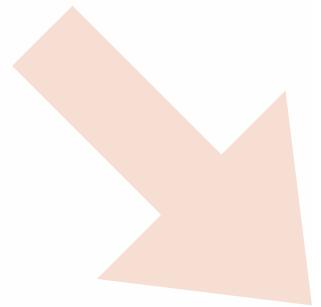
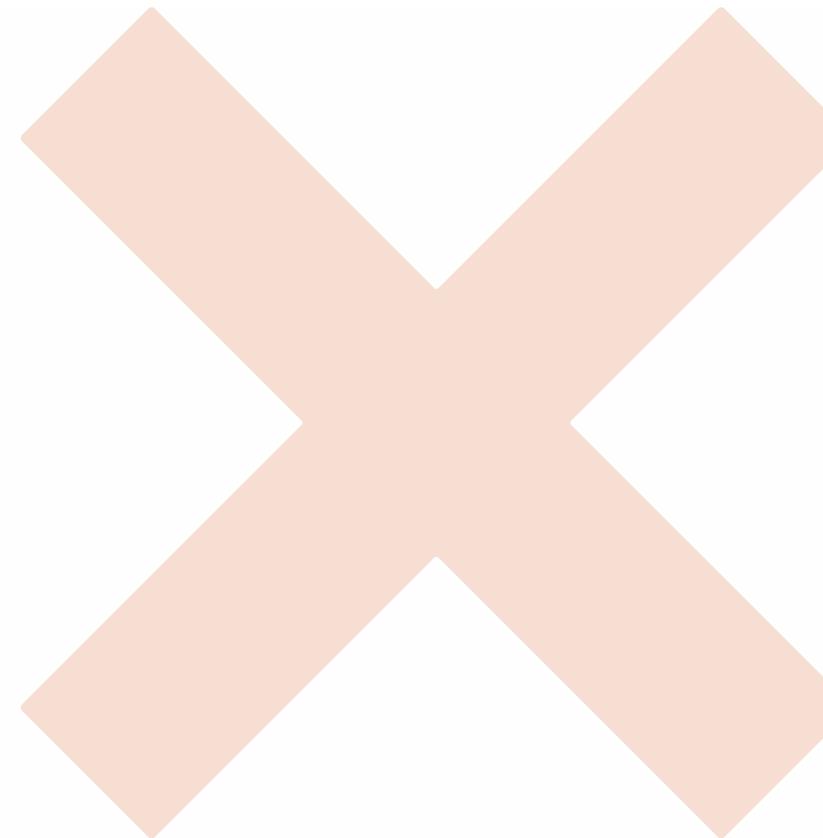
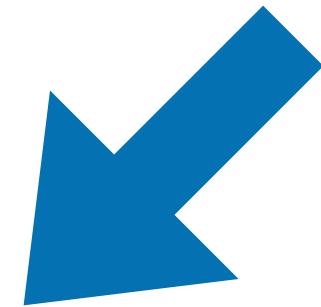
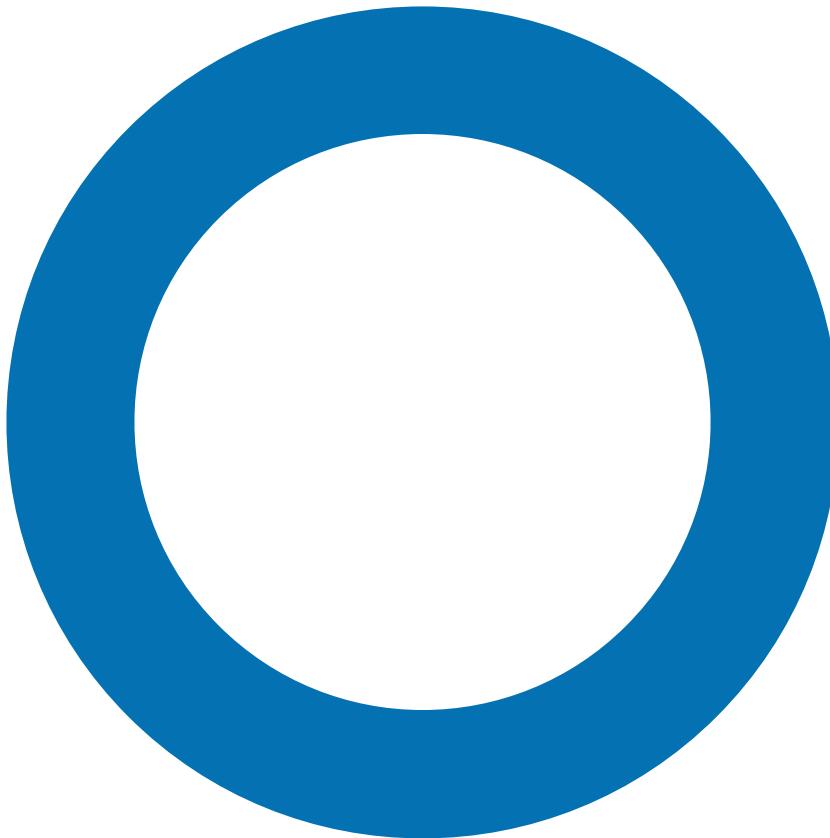
**5** : the manner in which the components of a computer or computer system are organized and integrated

// *different program architectures*

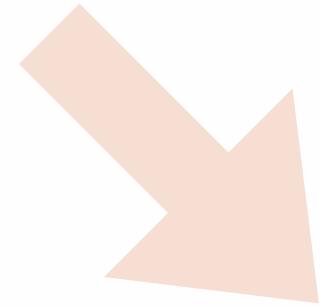
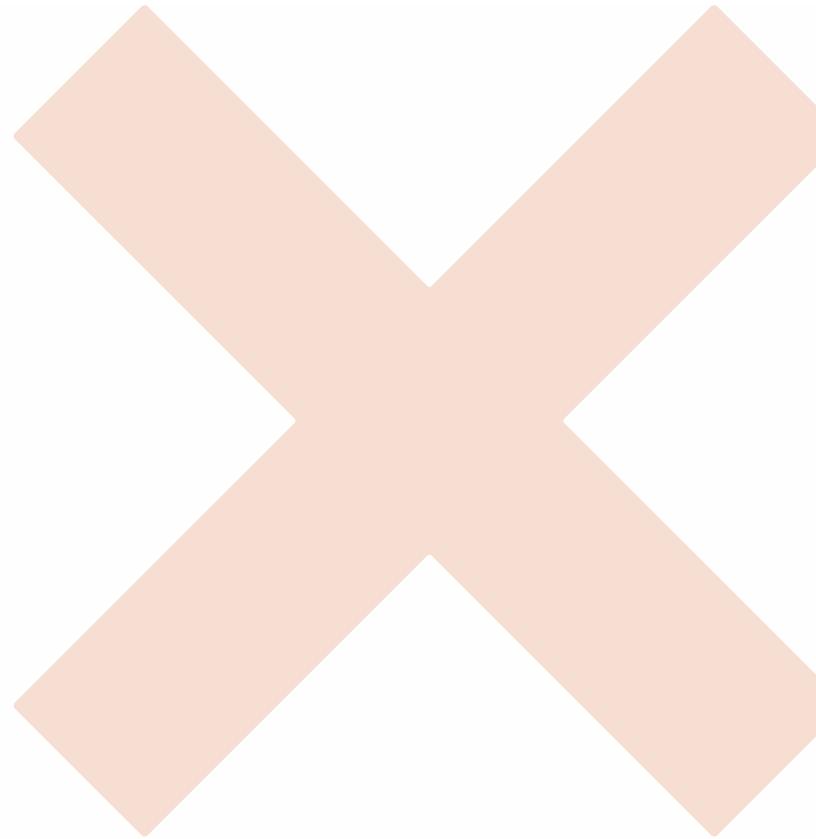
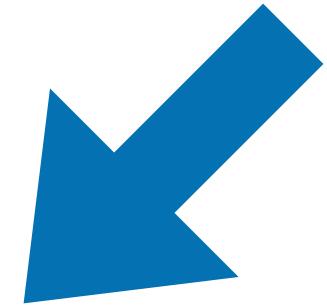
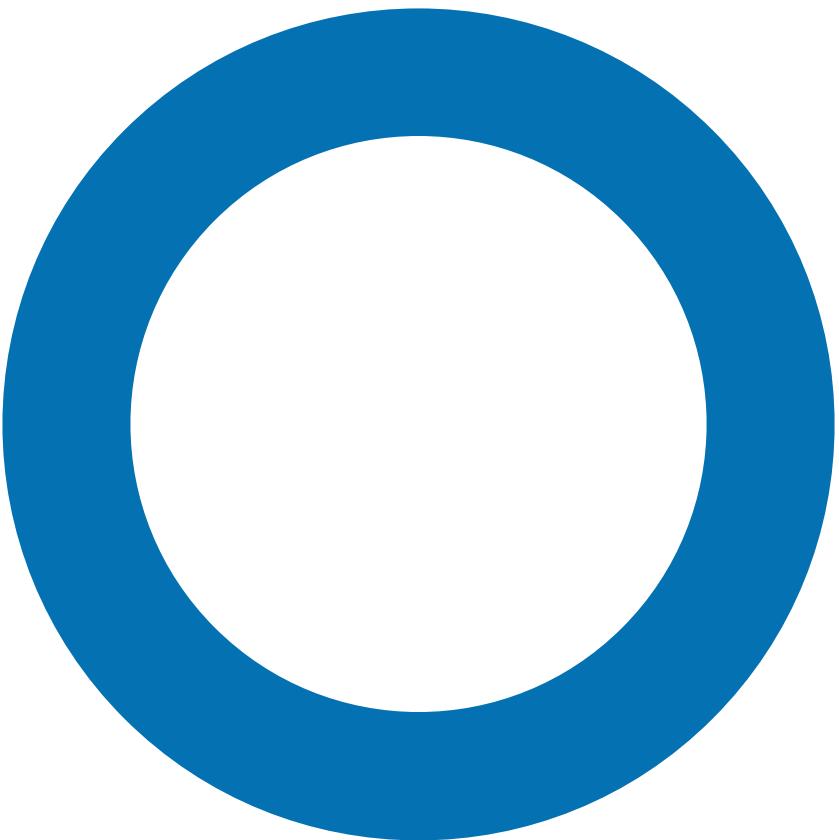
**The manner in which the components  
of a computer or computer system are  
organized and integrated**

**How much do we understand  
“Computer Architectures” for now**

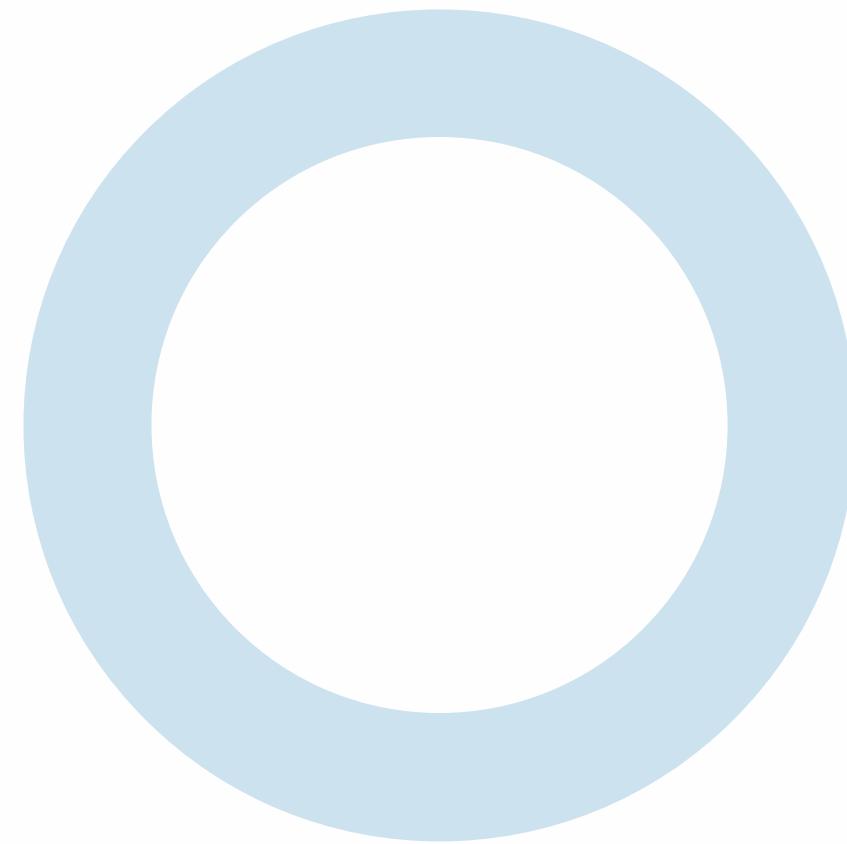
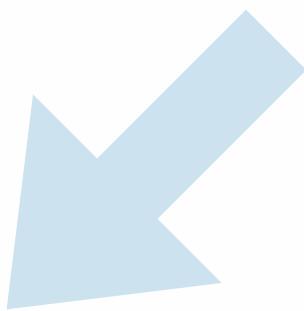
**Processors and memories are essential for most modern general-purpose computers**



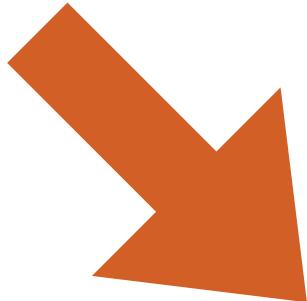
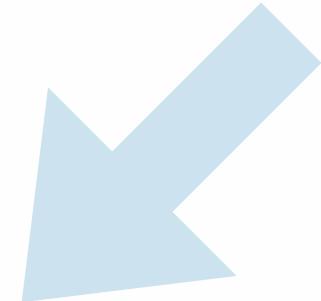
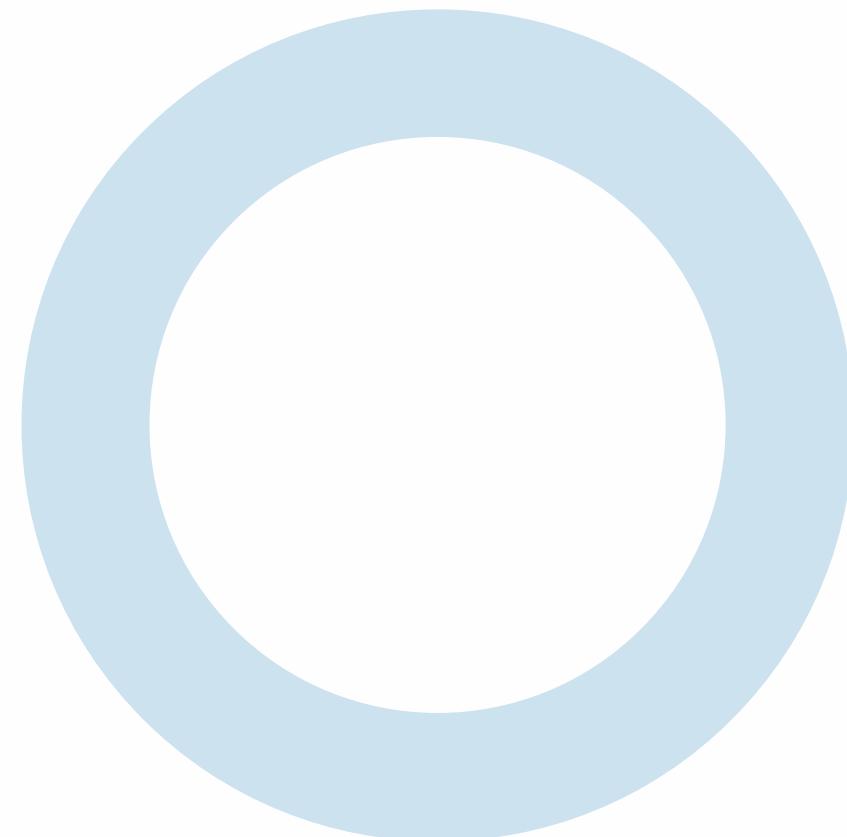
# Moore's Law is still alive



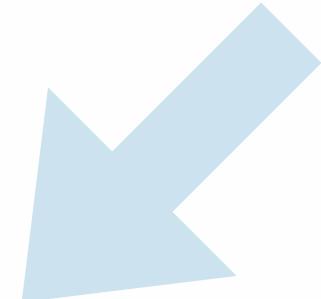
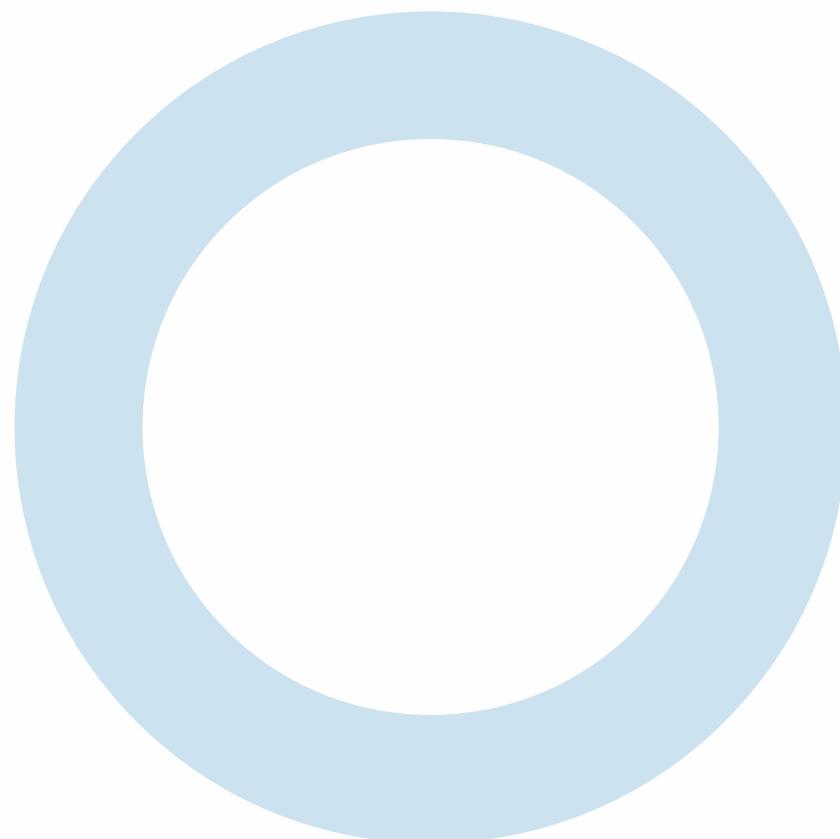
**On the same machine, programs with fewer lines  
of code will have shorter execution time**



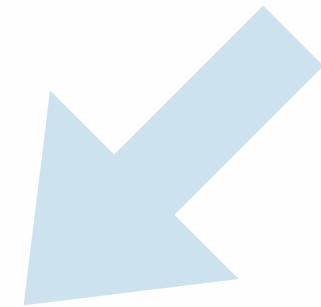
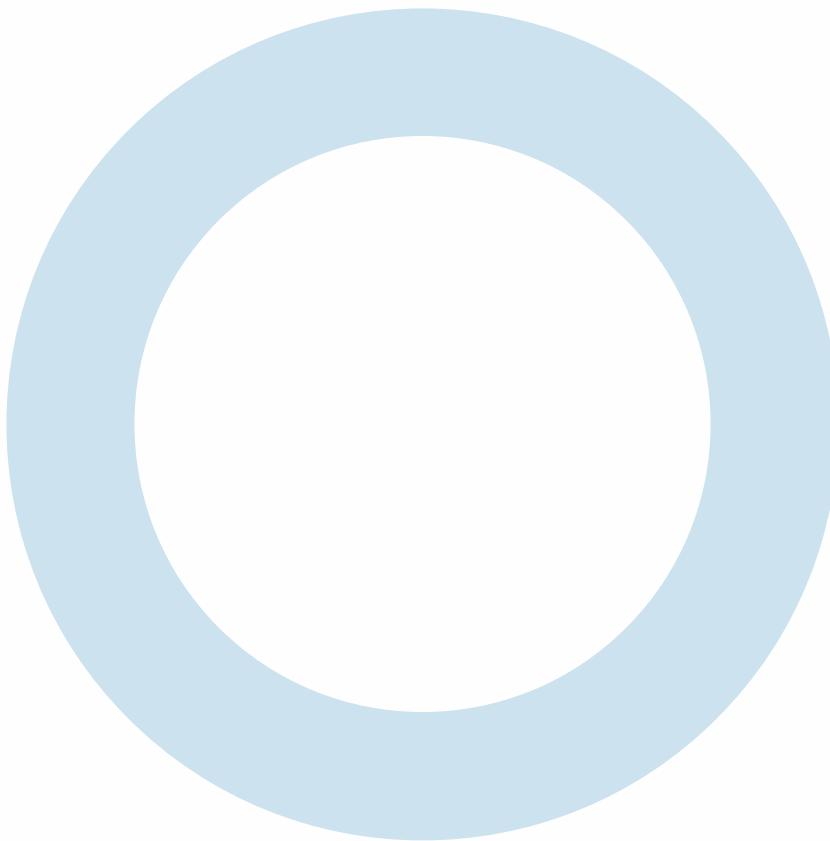
**On the same machine, an algorithm with lower computational complexities will have shorter execution time when solving the same problem with the same input**



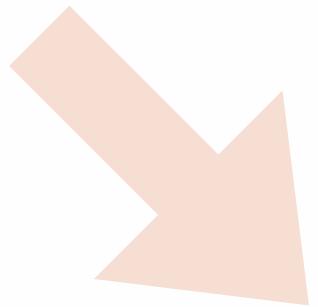
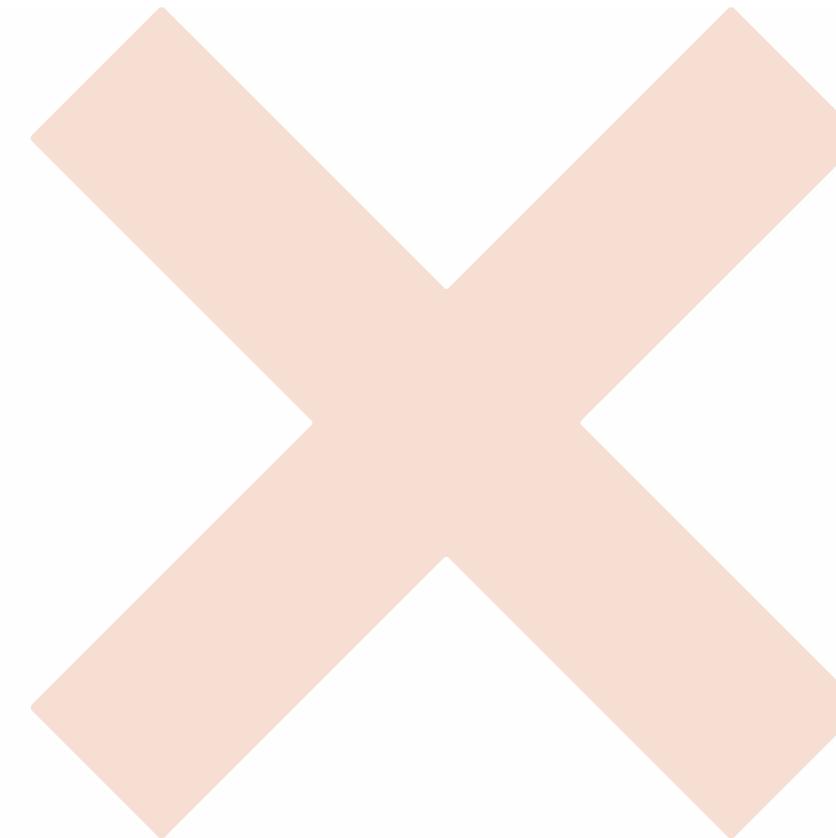
**On the same machine, algorithm implementations with the same computational complexities will similar execution time when solving the same problem with the same input**



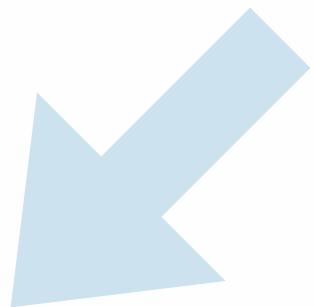
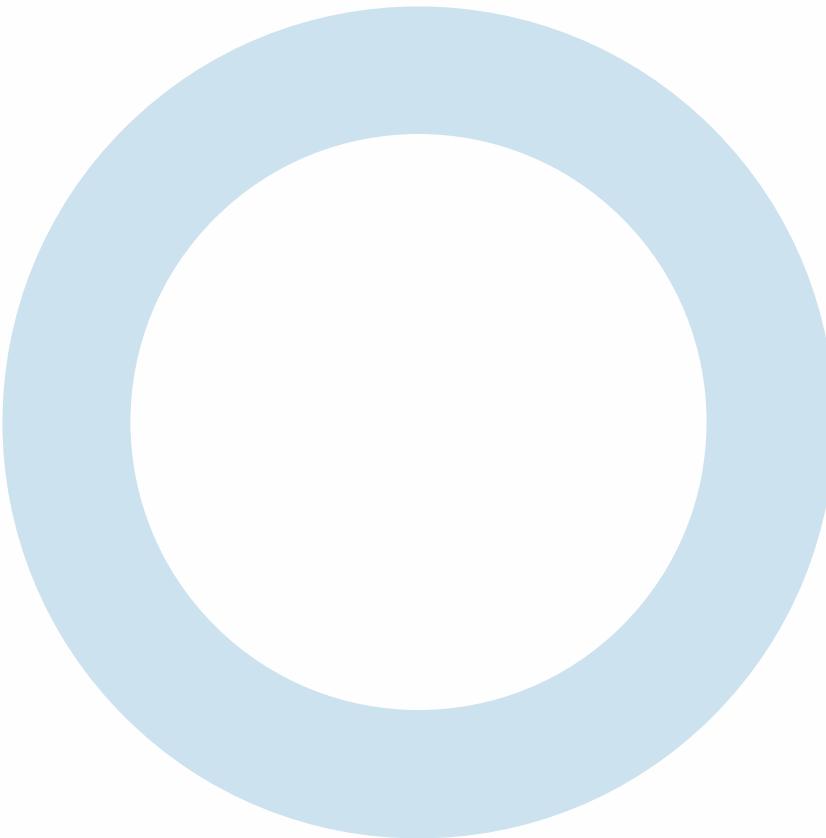
**Leveraging more “bit-wise” operations in C code will make the program significantly faster**



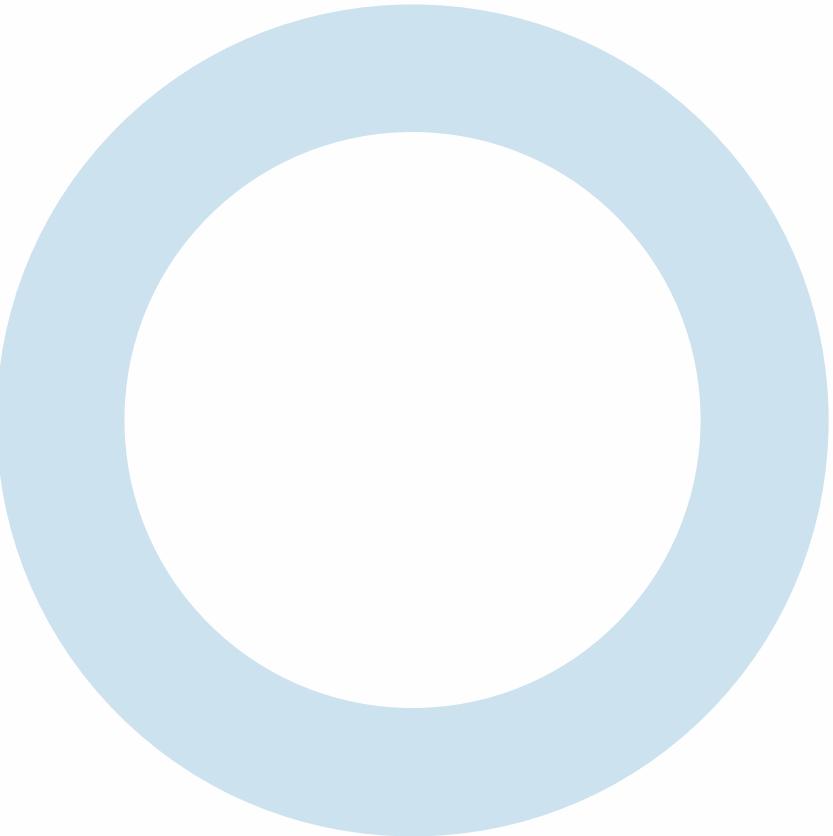
**Algorithm complexity is less important if we have rich parallel processing capabilities**



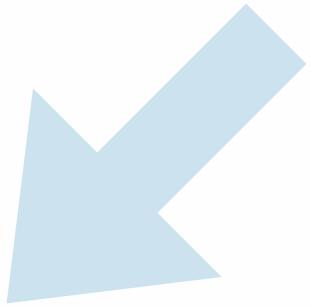
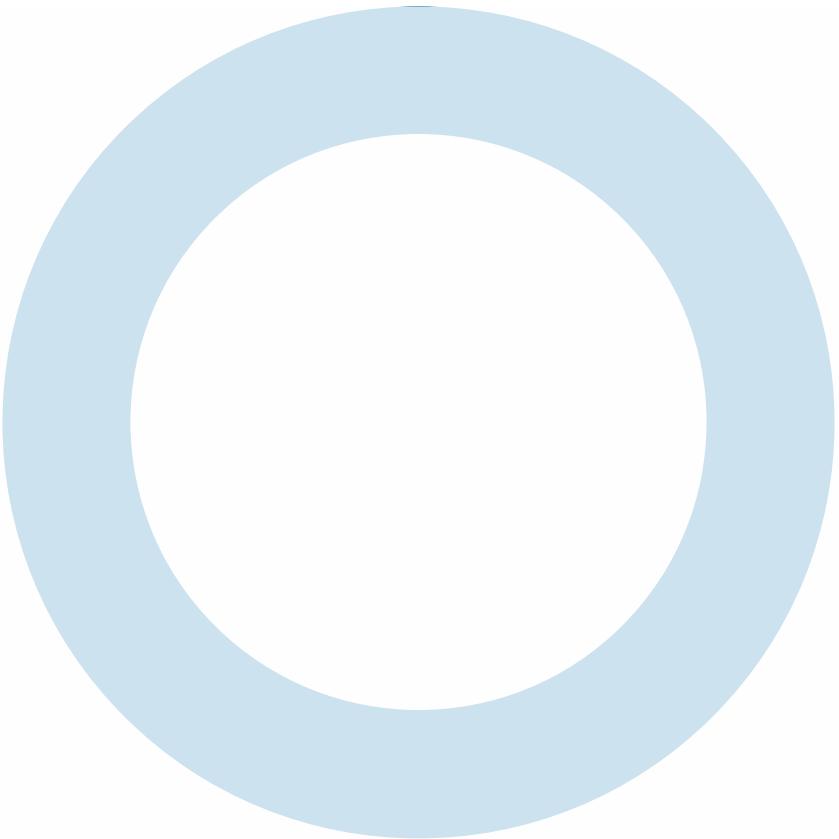
**The smaller size of a transistor,  
the smaller power consumption of it**



**GPUs surpass the limitations of Moore's Law better than CPUs**



**The performance growth of recent GPU architectures  
can match the demand of efficient AI/ML training**



**Isn't "algorithm" and "computational complexity" good enough?**

# 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*)**

**On the same machine, programs with fewer lines of code will have shorter execution time**



**On the same machine, an algorithm with lower computational complexities will have shorter execution time when solving the same problem with the same input**



# Demo (2)

A

```
for(i = 0; i < ARRAY_SIZE; i++)  
{  
    for(j = 0; j < ARRAY_SIZE; j++)  
    {  
        c[i][j] = a[i][j]+b[i][j];  
    }  
}
```

$O(n^2)$

B

```
for(j = 0; j < ARRAY_SIZE; j++)  
{  
    for(i = 0; i < ARRAY_SIZE; i++)  
    {  
        c[i][j] = a[i][j]+b[i][j];  
    }  
}
```

$O(n^2)$

Complexity

Performance?

# Demo (2)

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
    for(j = 0; j < ARRAY_SIZE; j++)
    {
        c[i][j] = a[i][j]+b[i][j];
    }
}
```

$O(n^2)$

A Lot Better!

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
    for(i = 0; i < ARRAY_SIZE; i++)
    {
        c[i][j] = a[i][j]+b[i][j];
    }
}
```

$O(n^2)$

Complexity

Performance?

Worse

**On the same machine, algorithm implementations with the same computational complexities will similar execution time when solving the same problem with the same input**



# Demo (3) — Bitwise operations?

A

```
void regswap(int* a, int* b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

B

```
void xorswap(int* a, int* b) {  
    *a ^= *b;  
    *b ^= *a;  
    *a ^= *b;  
}
```

**Leveraging more “bit-wise” operations in C code will make the program significantly faster**



# Demo (4) — quick sort v.s. bitonic sort on GPU

## Quick Sort

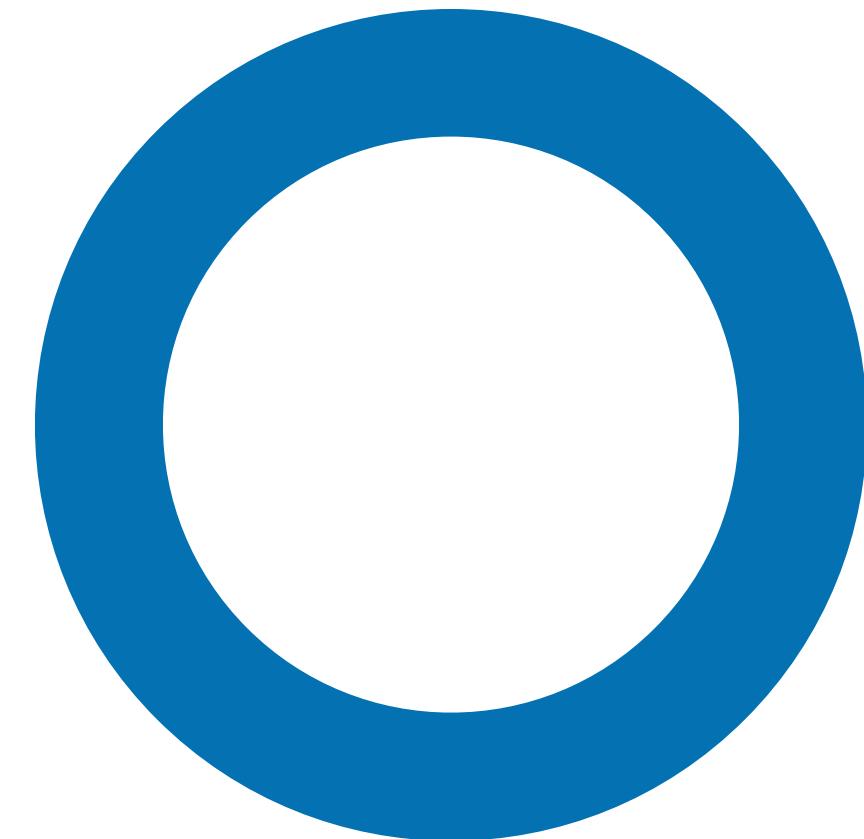
 $O(n \log_2 n)$ 

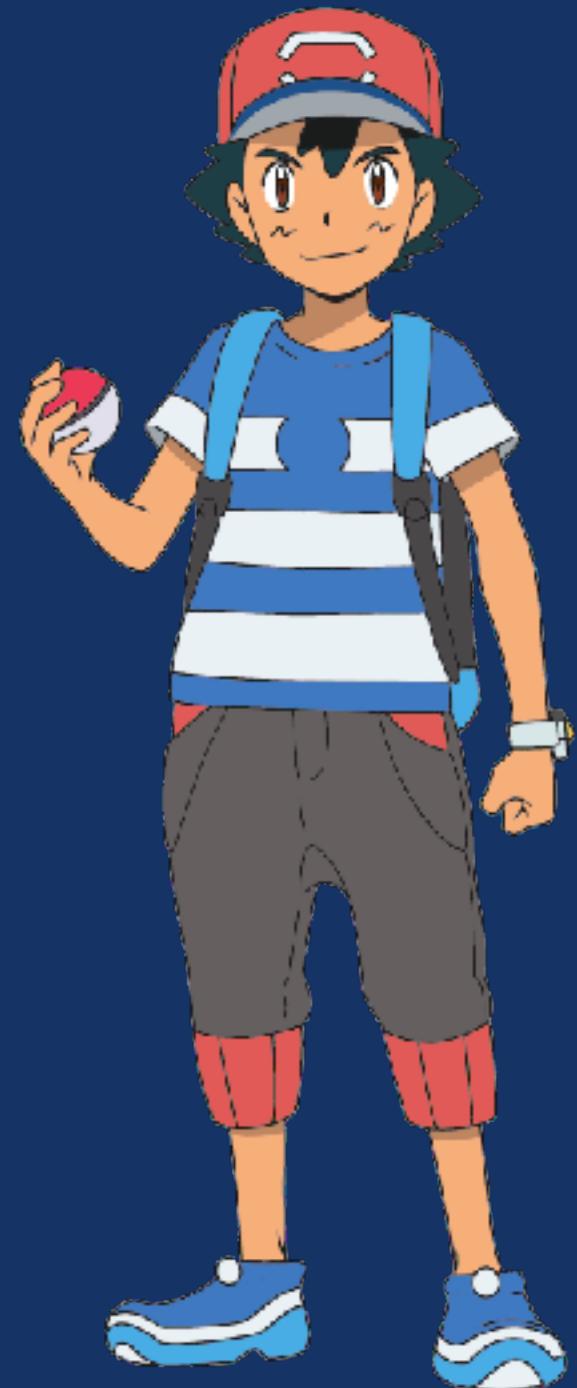
## Bitonic Sort

 $O(n \log_2^2 n)$ 

```
void BitonicSort() {  
    int i,j,k;  
  
    for (k=2; k<=N; k=2*k) {  
        for (j=k>>1; j>0; j=j>>1) {  
            for (i=0; i<N; i++) {  
                int ij=i^j;  
                if ((ij)>i) {  
                    if ((i&k)==0 && a[i] > a[ij])  
                        exchange(i,ij);  
                    if ((i&k)!=0 && a[i] < a[ij])  
                        exchange(i,ij);  
                }  
            }  
        }  
    }  
}
```

**Algorithm complexity is less important if we have rich parallel processing capabilities**





?????



# Thinking about the washlet



# Or a Tesla



# Take-aways: Why CS203?

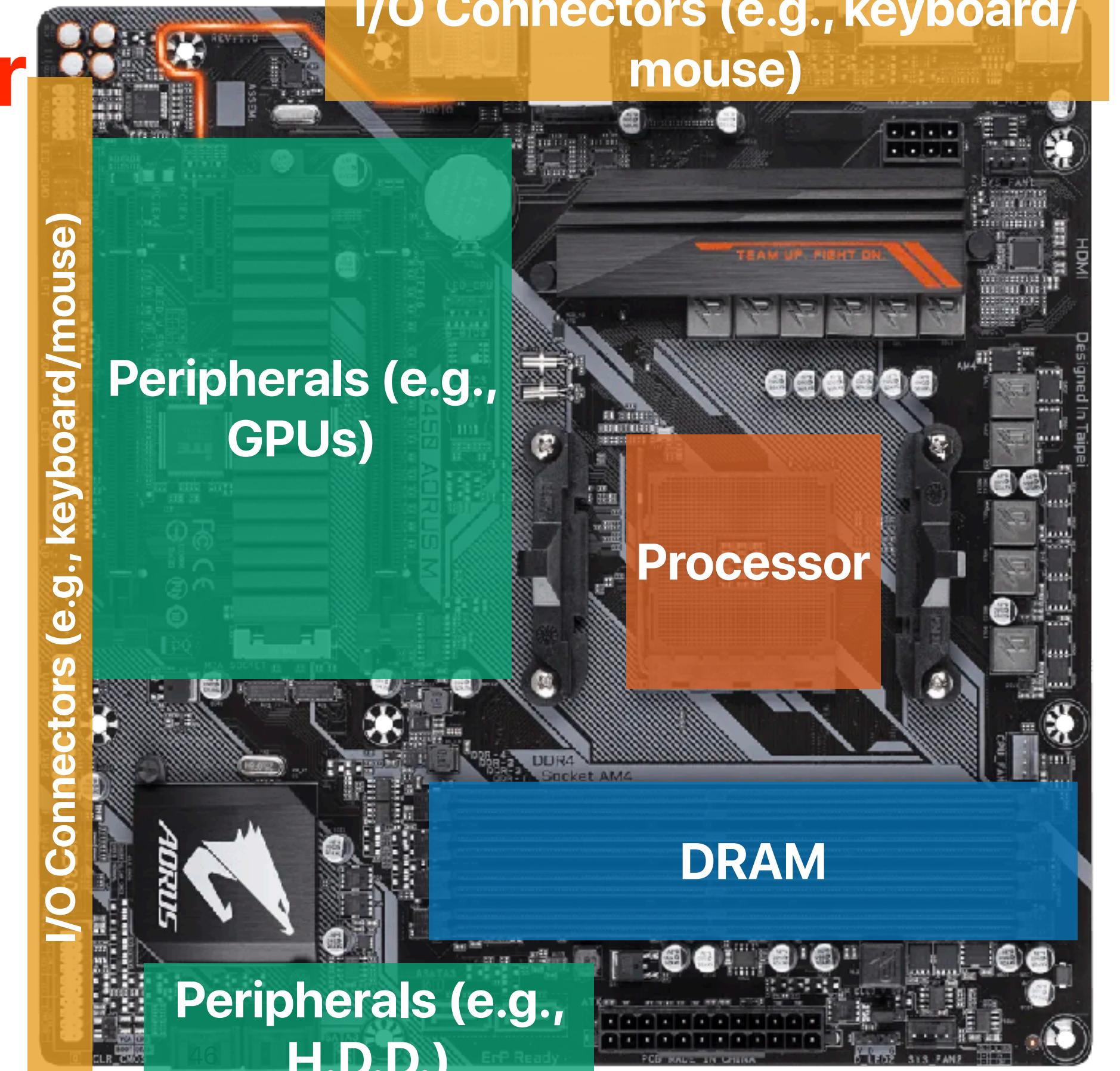
- Algorithm complexity does not work well on “real” computers

# **Big Picture of Computer Architecture: the Von Neumann Architecture**

# Desktop Computer

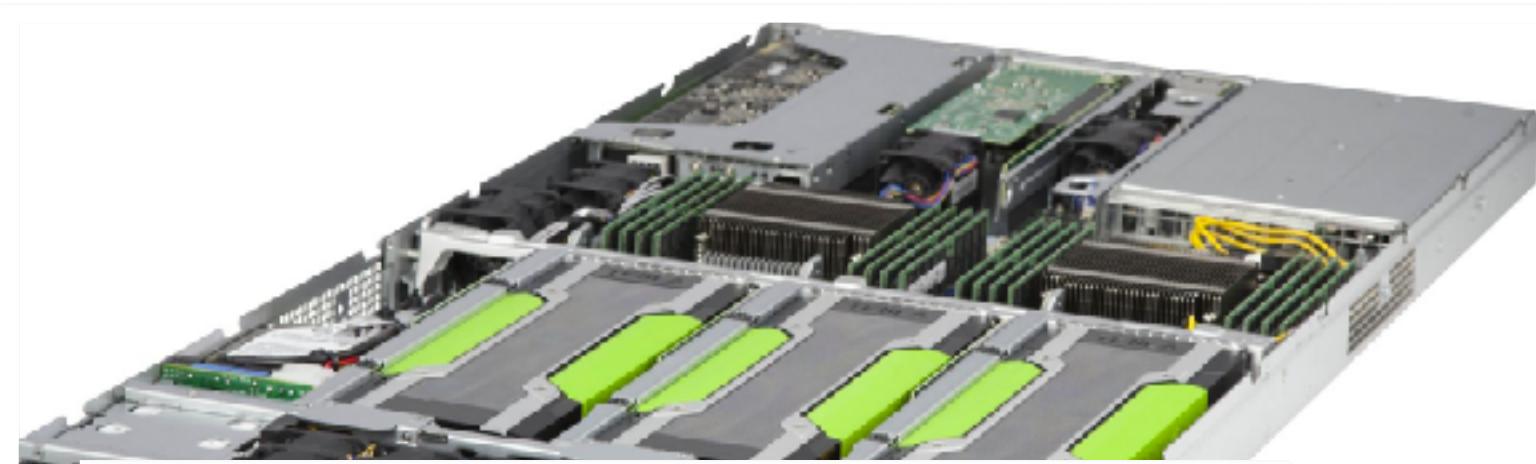


I/O Connectors (e.g., keyboard/mouse)



# Server

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



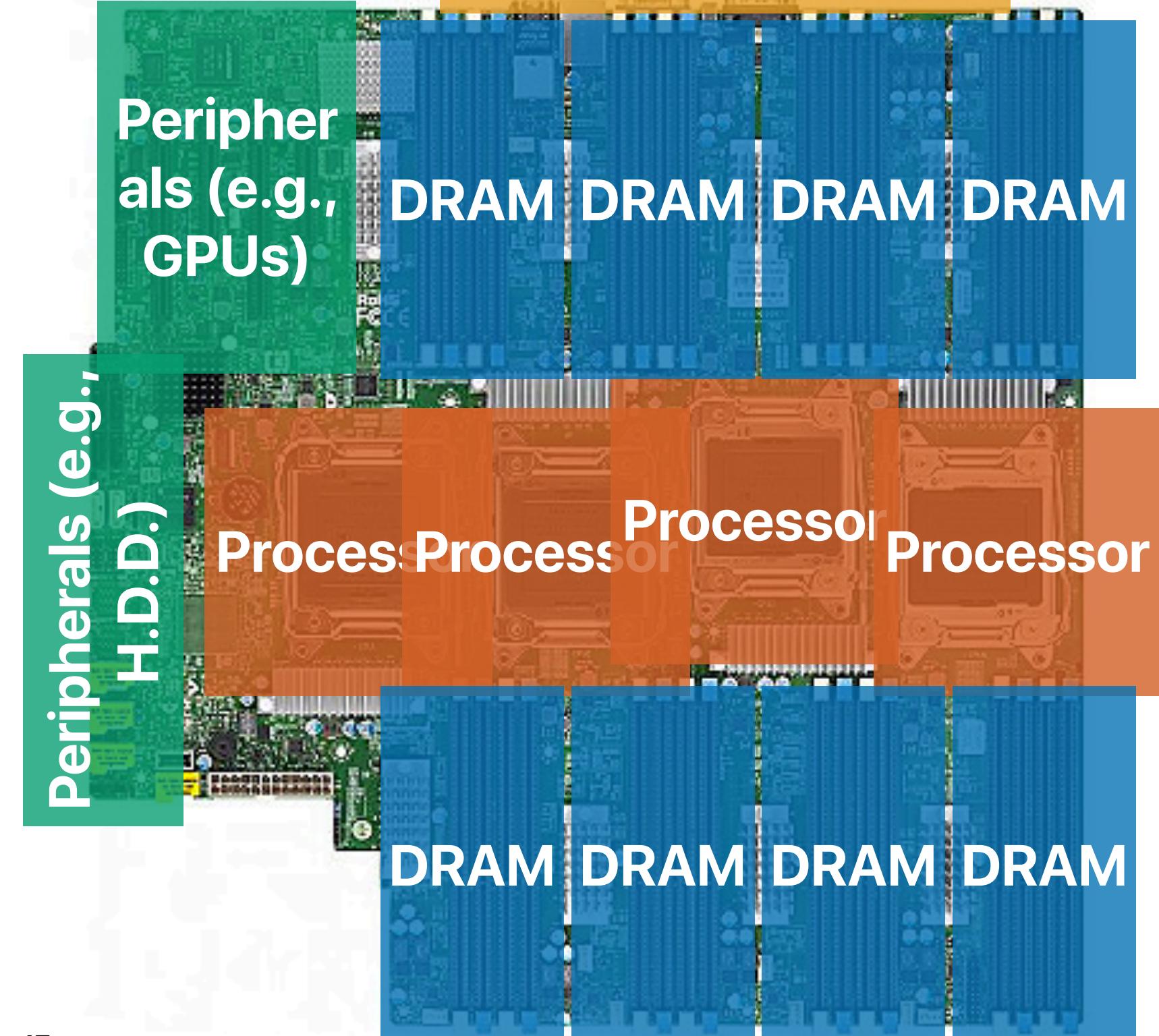
Peripherals (e.g., H.D.D.)

Peripherals (e.g., GPUs)

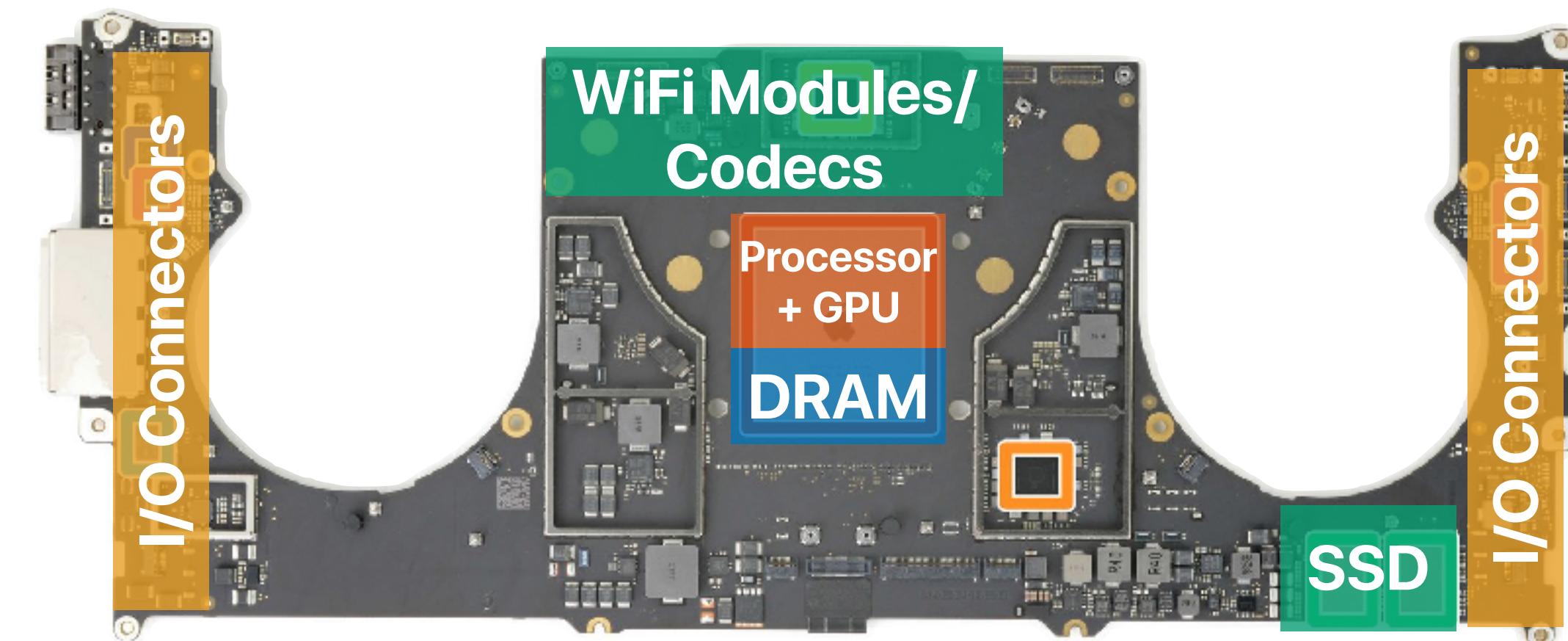
DRAM DRAM DRAM DRAM

Processor Processor Processor Processor

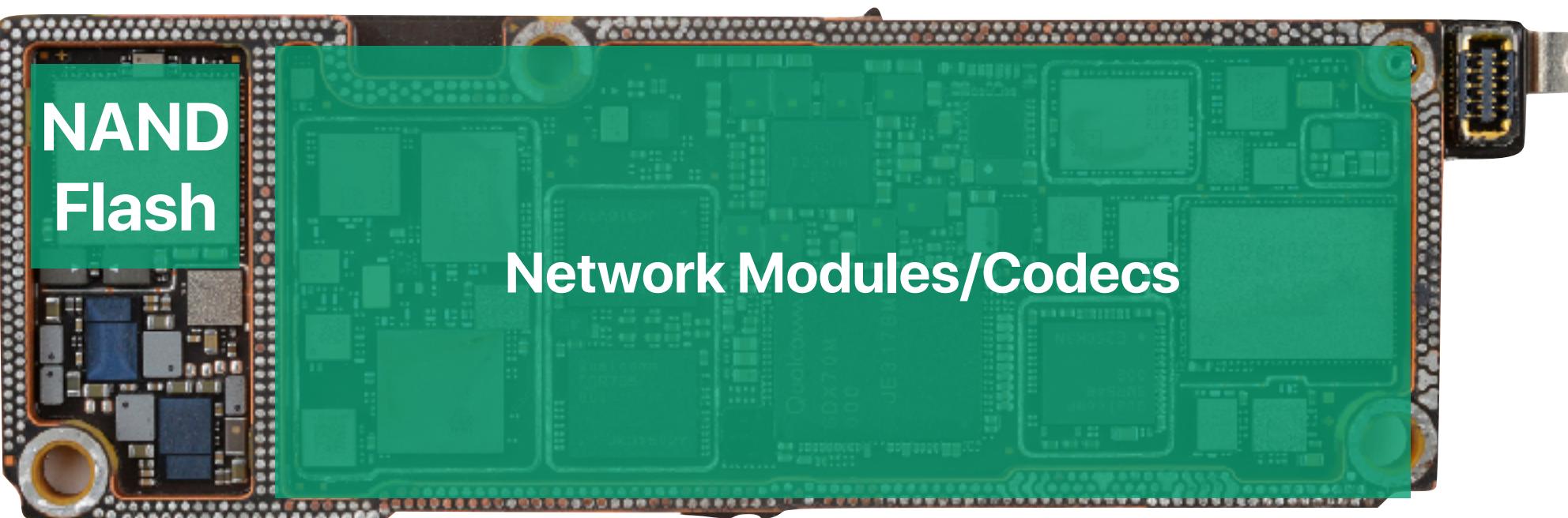
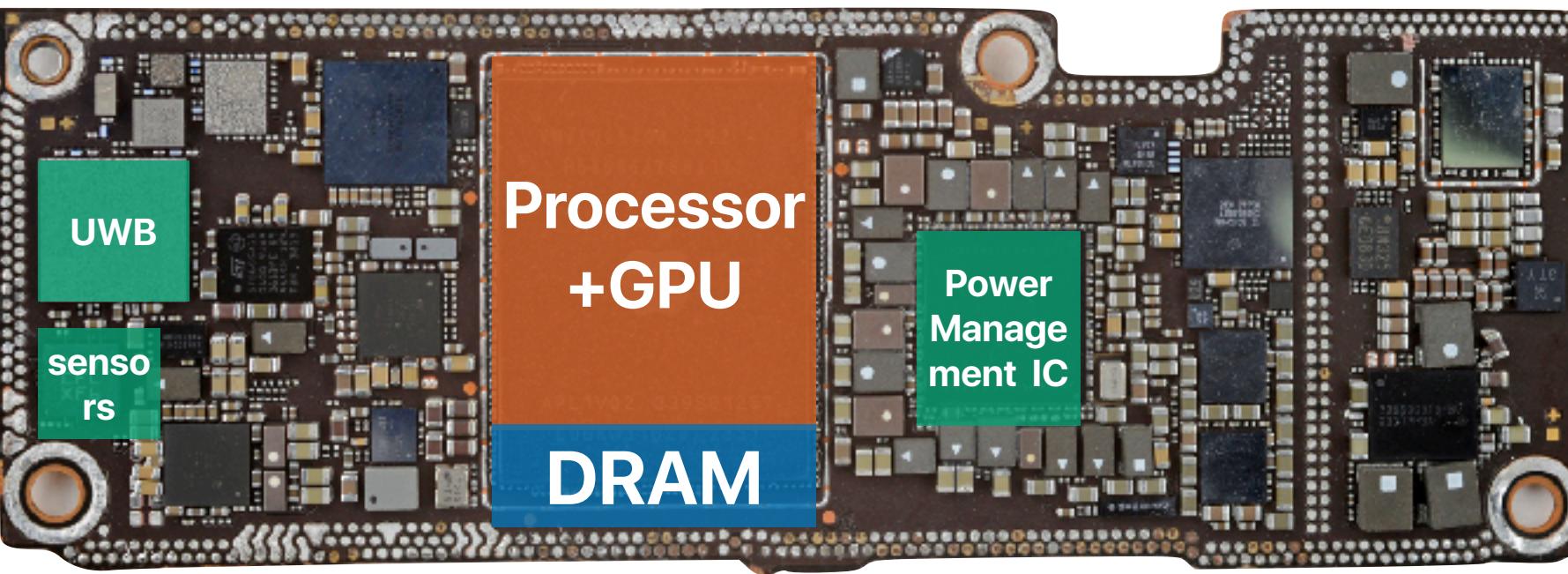
DRAM DRAM DRAM DRAM



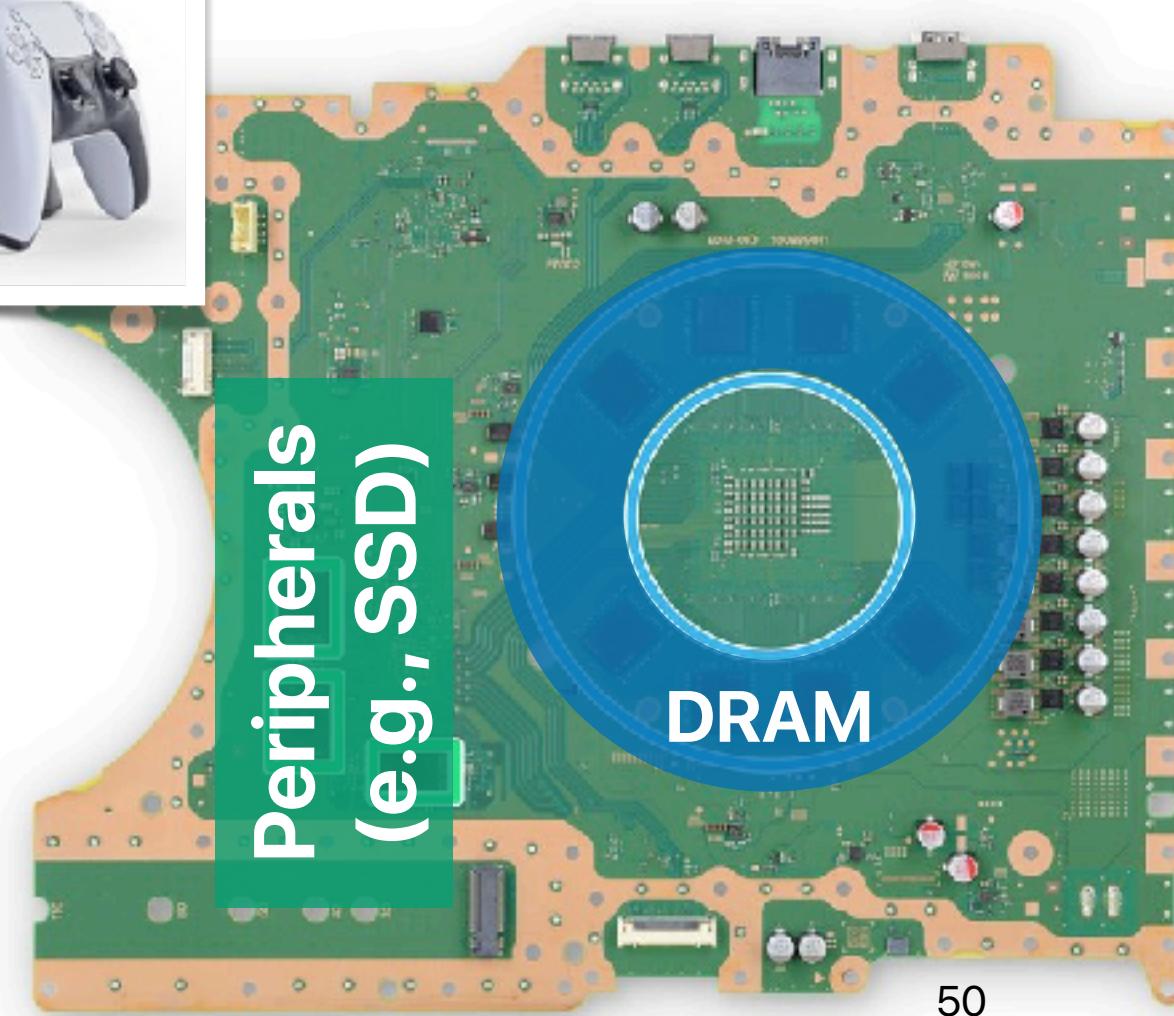
# MacBook Pro M3



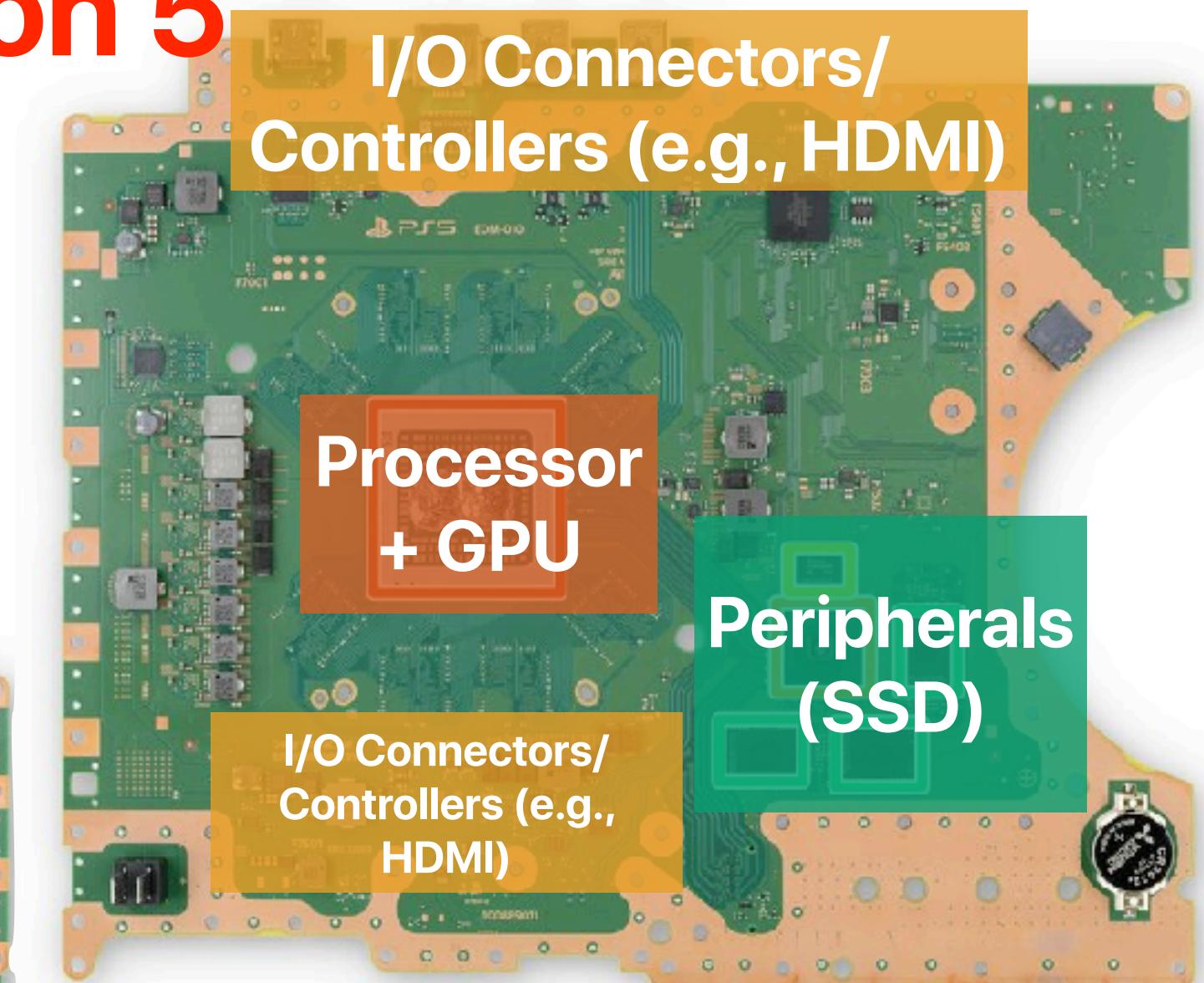
# iPhone 15 Pro



# Play Station 5



Peripherals  
(e.g., SSD)



Processor  
+ GPU

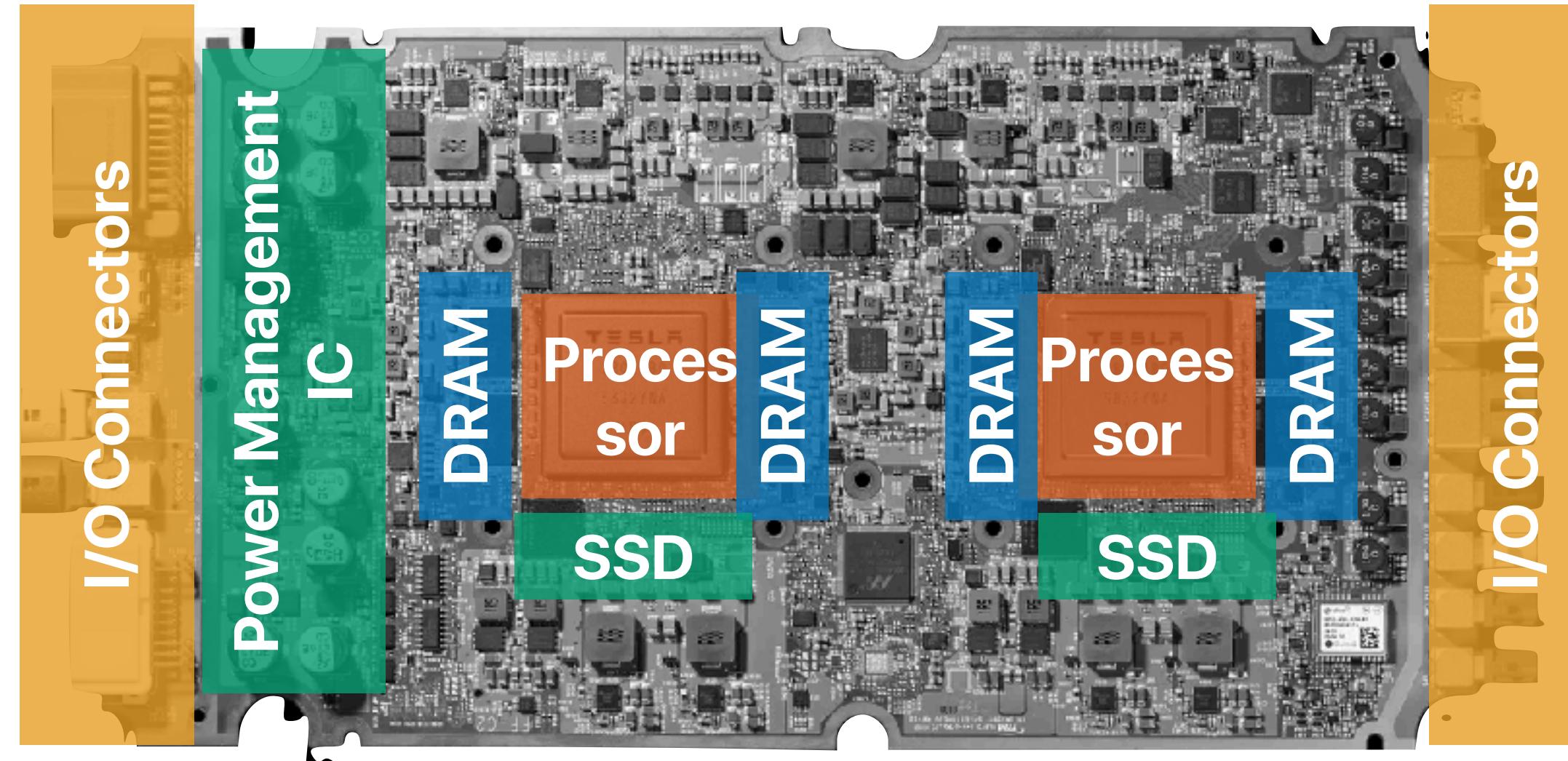
Peripherals  
(SSD)

I/O Connectors/  
Controllers (e.g.,  
HDMI)

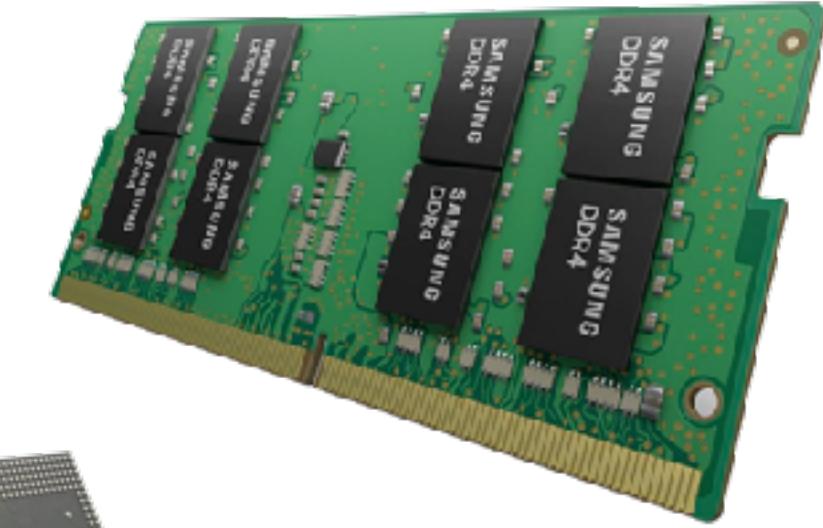
# Nintendo Switch



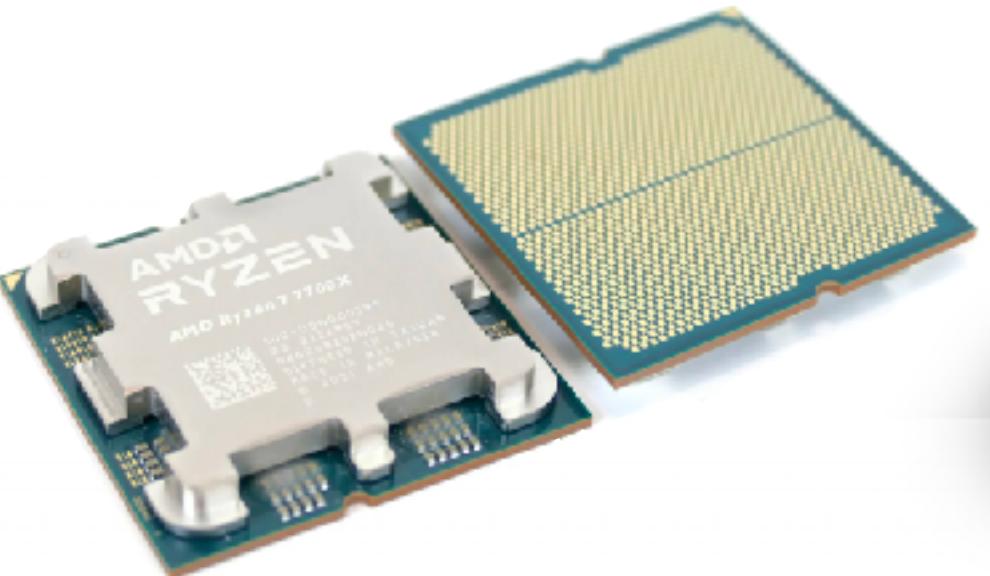
# Tesla Model 3



# Processors and memory modules are everywhere!



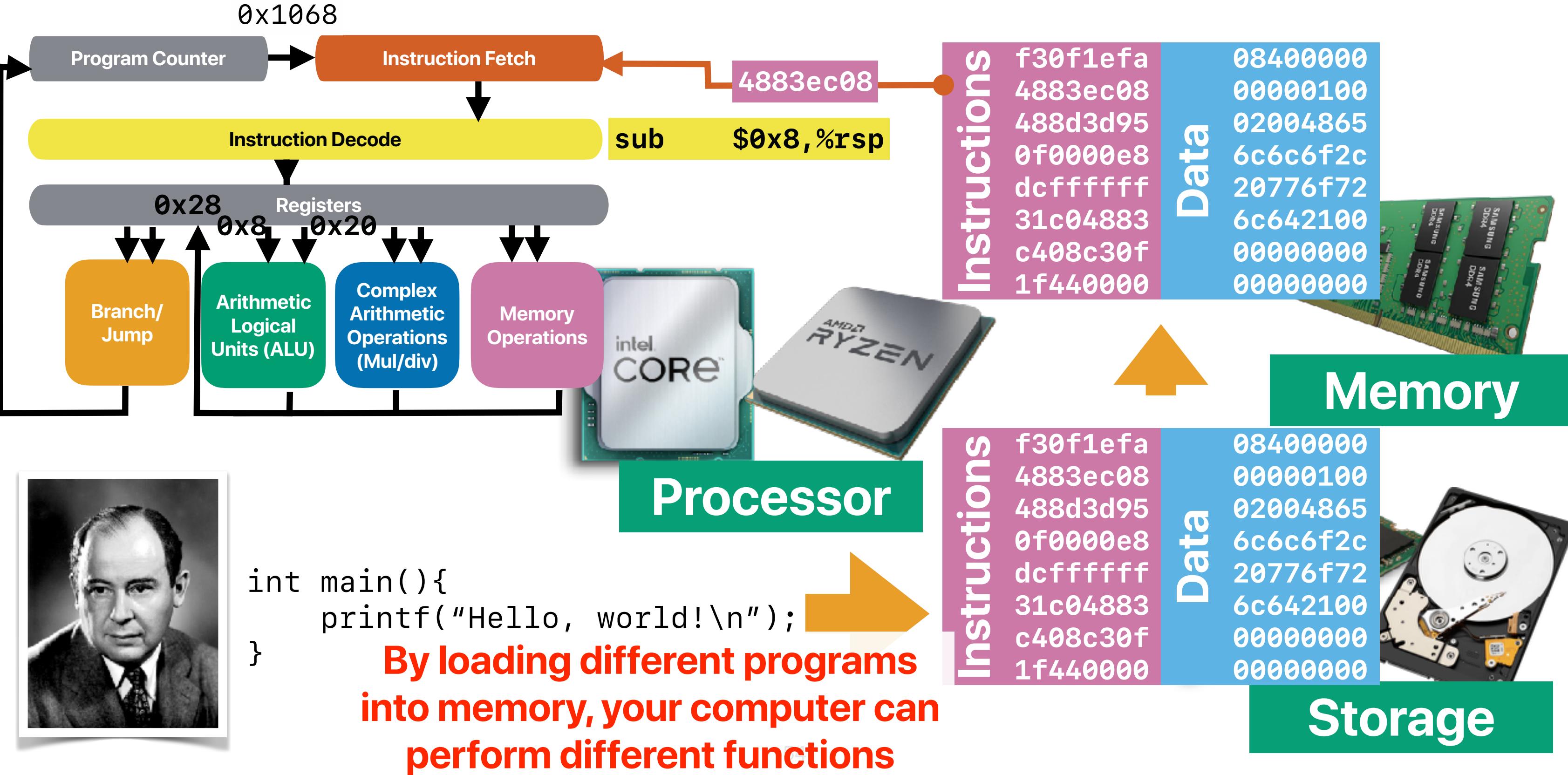
## Processors



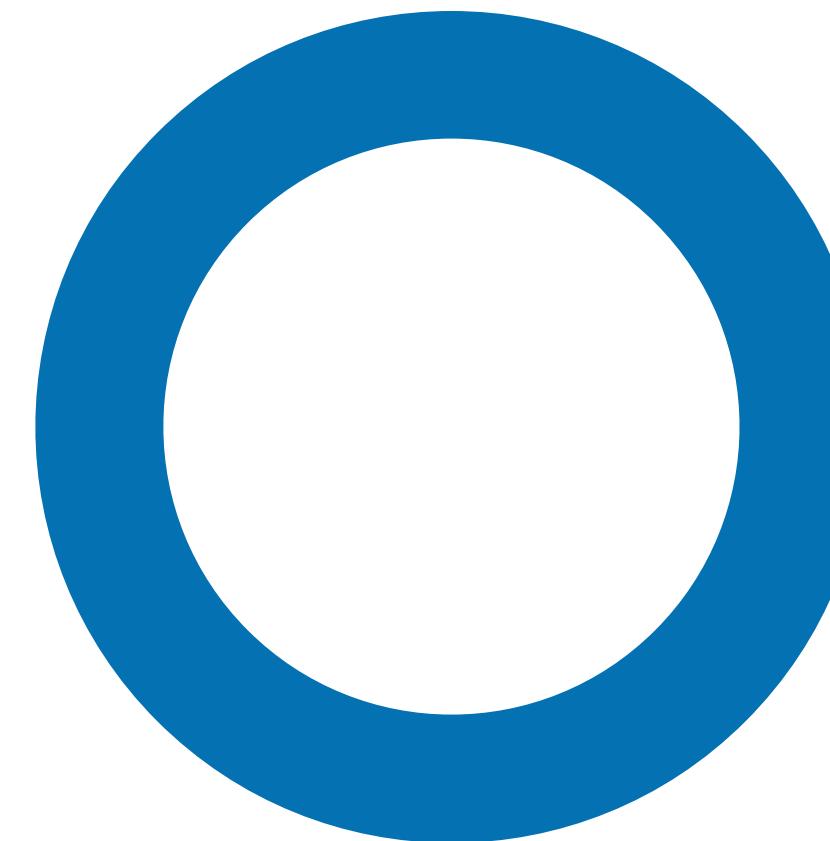
## Memory



# von Neumann architecture



**Processors and memories are essential for most modern general-purpose computers**



# Start with this simple program in C

```
int A[] =  
{1,2,3,4,5,6,7,8,9,10,1,2,3,4  
,5,6,7,8,9,10};
```

Compiler

Contents of section .data:  
0000 01000000 02000000 03000000 04000000  
0010 05000000 06000000 07000000 08000000  
0020 09000000 0a000000 0b000000 0c000000  
0030 03000000 04000000 05000000 06000000  
0040 07000000 08000000 09000000 0a000000

control flow  
operations  
logical operations

```
int main()  
{  
    int i=0, sum=0;  
    for(i = 0; i < 20; i++)  
    {  
        sum += A[i];  
    }  
    return 0;  
}
```

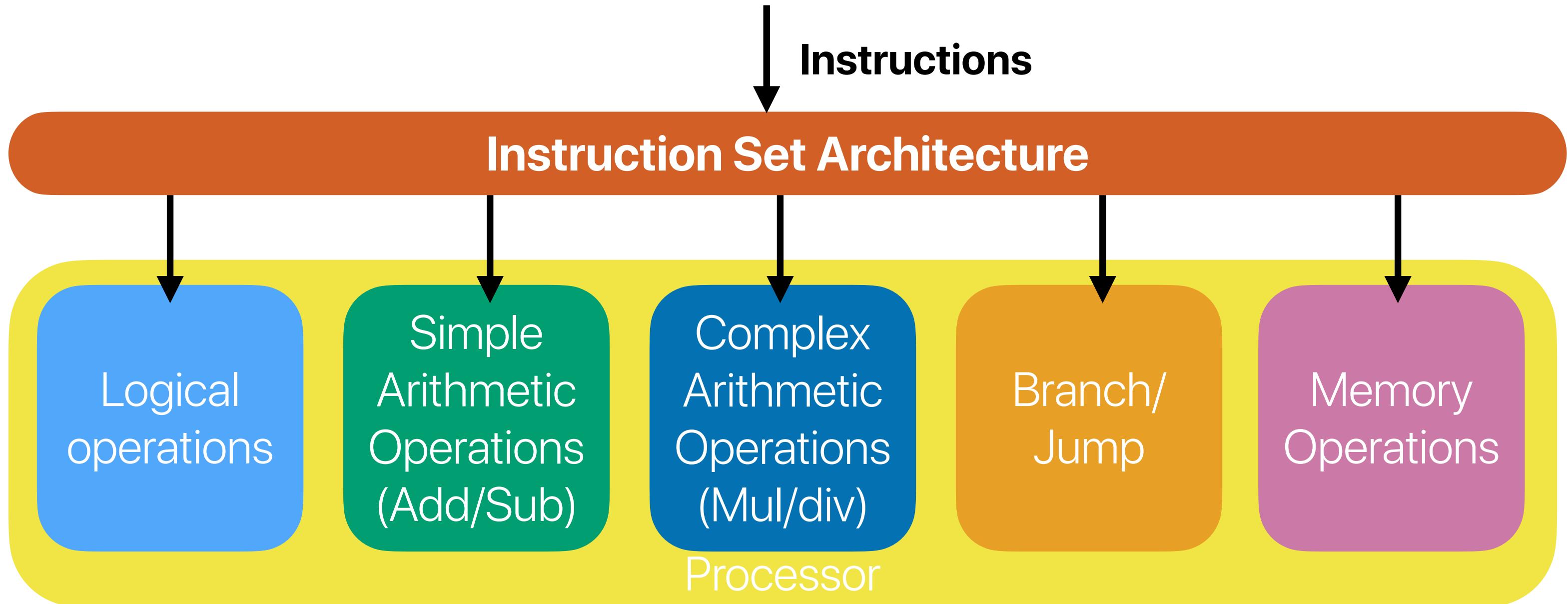
memory access  
arithmetic operations

Compiler

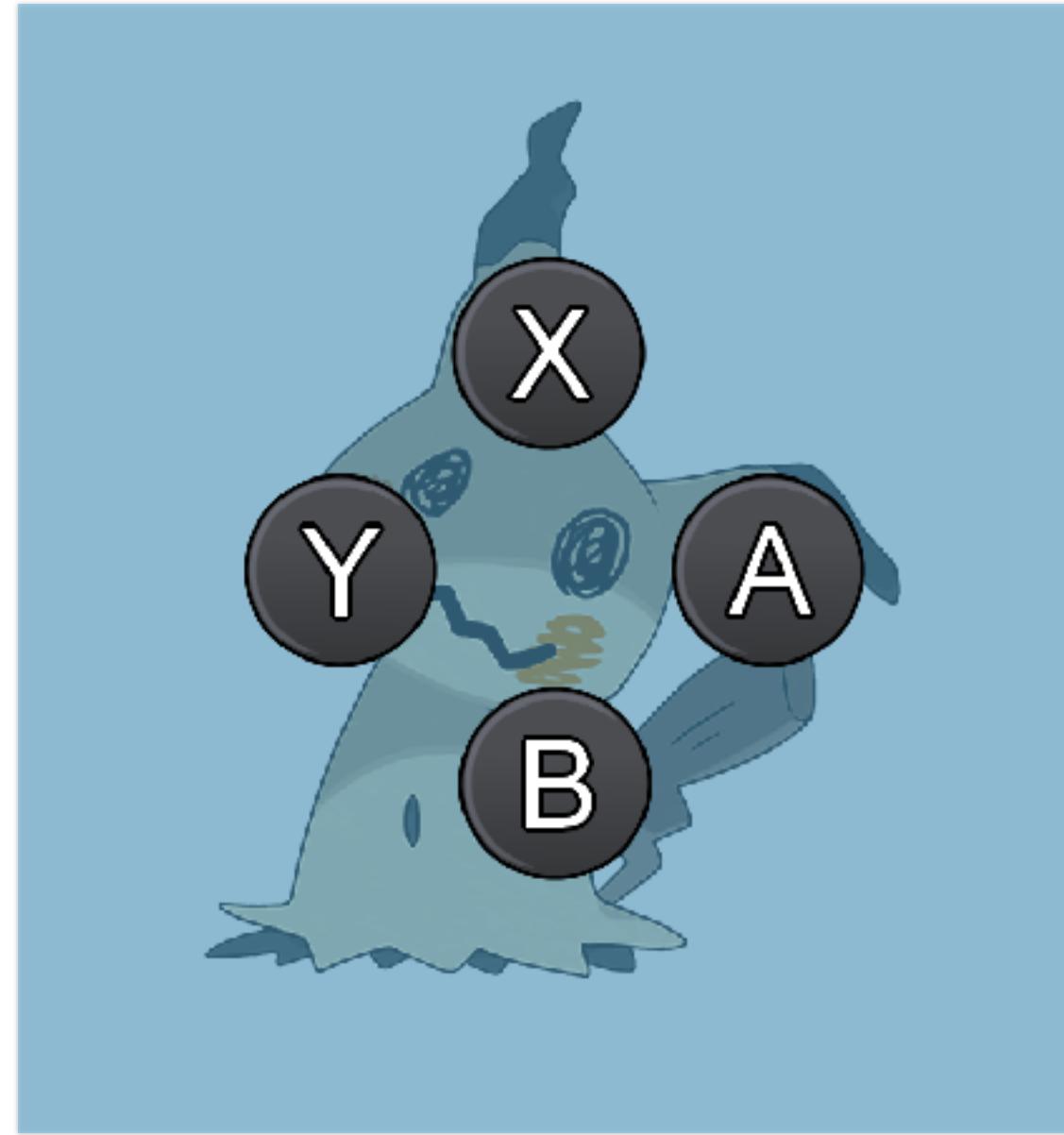
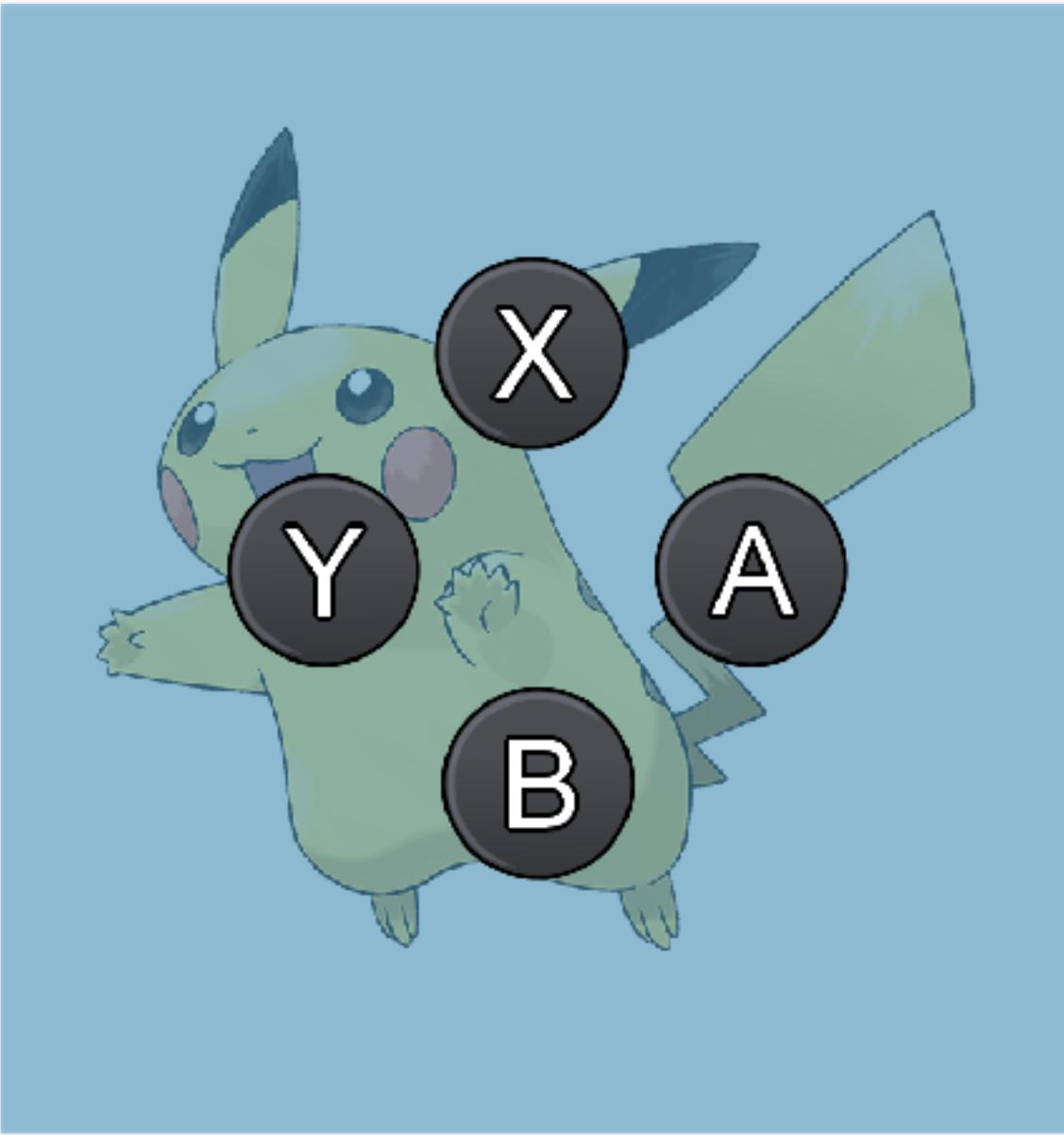
main:  
.LFB0:  
endbr64  
pushq %rbp  
movq %rsp, %rbp  
movl \$0, -8(%rbp)  
movl \$0, -4(%rbp)  
movl \$0, -8(%rbp)  
jmp .L2  
.L2:  
cmpl \$19, -8(%rbp)  
jle .L3  
movl \$0, %eax  
popq %rbp  
ret

Contents of section .text:  
0000 f30f1efa 554889e5 c745f800 000000c7  
0010 45fc0000 0000c745 f8000000 00eb1e8b  
0020 45f84898 488d1405 00000000 488d0500  
0030 0000008b 04020145 fc8345f8 01837df8  
0040 137edcb8 00000000 5dc3

# Microprocessor — a collection of functional units



# ISA — the “abstraction” of processor features



# **Challenges of von Neumann Architecture**

# Moore's Law<sup>(1)</sup>

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

# Moore's Law<sup>(1)</sup>

## 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

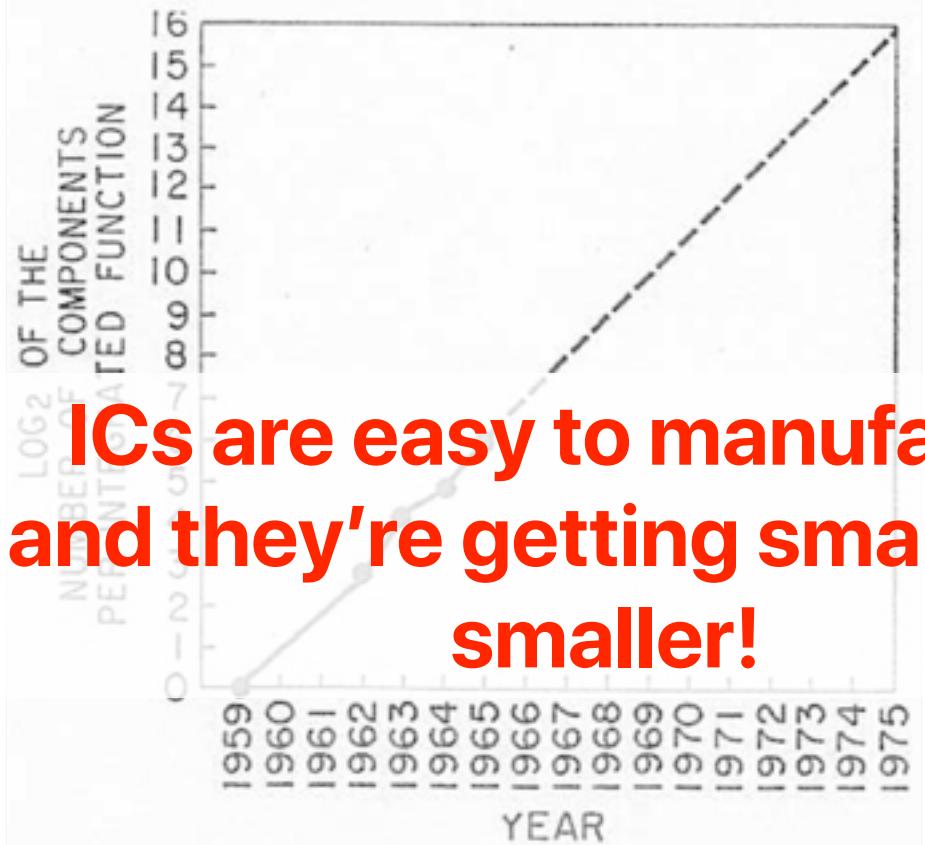
**ICs are small**

(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

units. The lack of large-value capacitors and

In almost every application, ICs have demonstrated higher reliability than discrete components—it offers reduced systems cost, and in many systems improved performance has been realized.

**ICs are more reliable**

## Reliability count

shed light on new and valuable

parallel, for the reliability that failure as the

### Heat problem

Will it be possible to remove the heat generated by tens of thousands of components in a single silicon chip?

**Moore's Law sets the pace**

### Day of reckoning

Clearly, we will be able to build such component-crammed equipment. Next, we ask under what circumstances we should do it. The total cost of making a particular system function must be minimized. To do so, we could amortize the engineering over several identical items, or evolve flexible techniques for the engineering of large functions so that no disproportionate expense need be borne by a particular array. Perhaps newly devised design automation procedures could translate from any special engineering.

**Designing ICs can be easy**

'components onto integrated circuits', Electronics 38 (8).

## Moore's Law Alive and Well

---



**RTX 4080 Why so expen\$ive?**

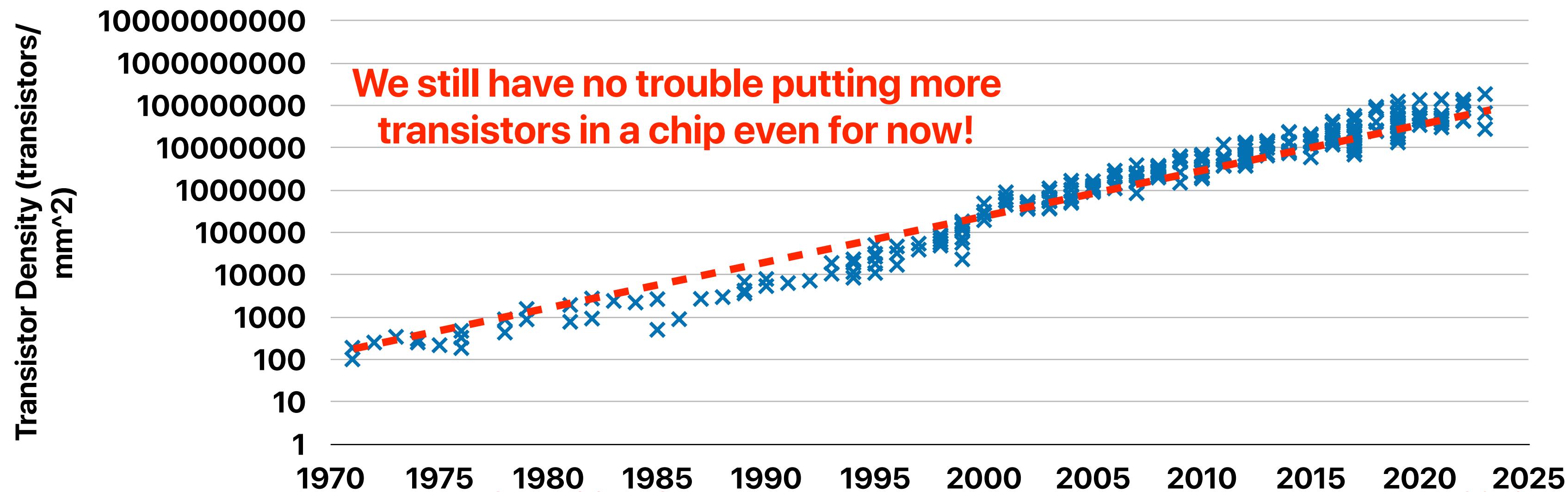


*"Moore's Law  
Is Dead"*



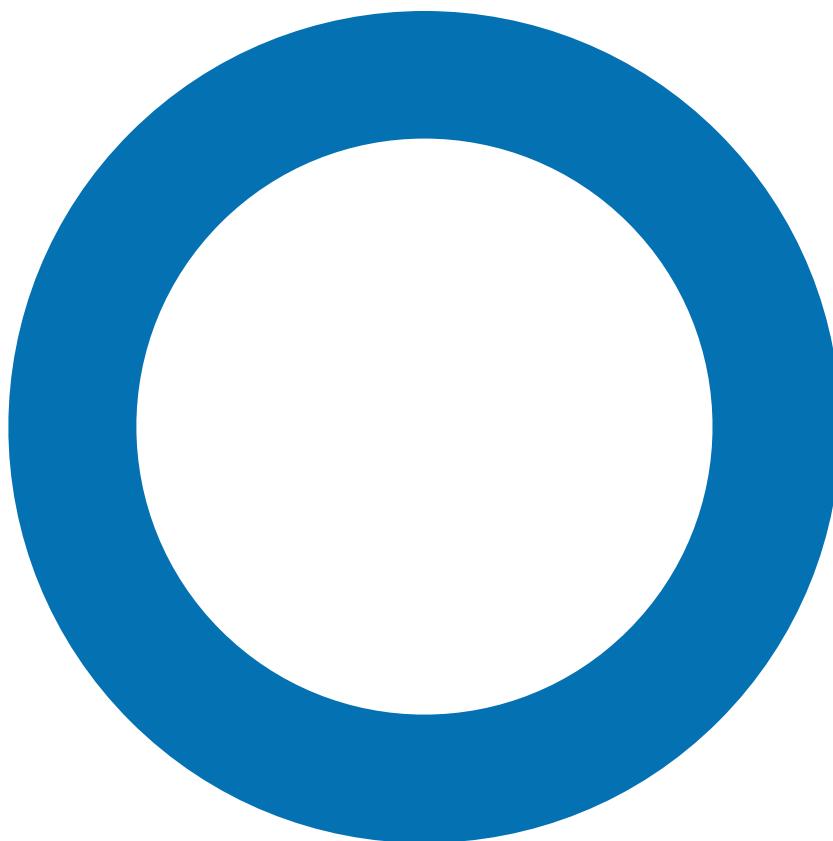
# Moore's Law<sup>(1)</sup>

- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.
- Moore's Law "was" the most important driver for historic CPU performance gains



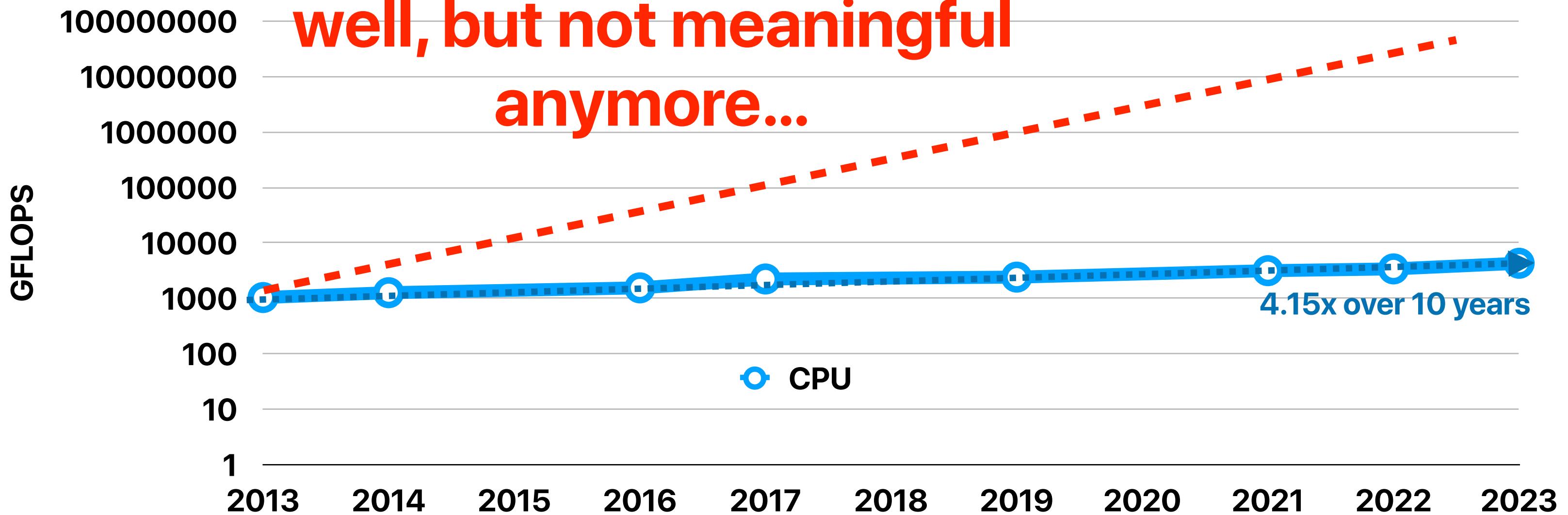
(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8).

# **Moore's Law is still alive**

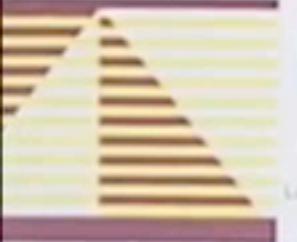


# CPU Performance v.s. Moore's Law

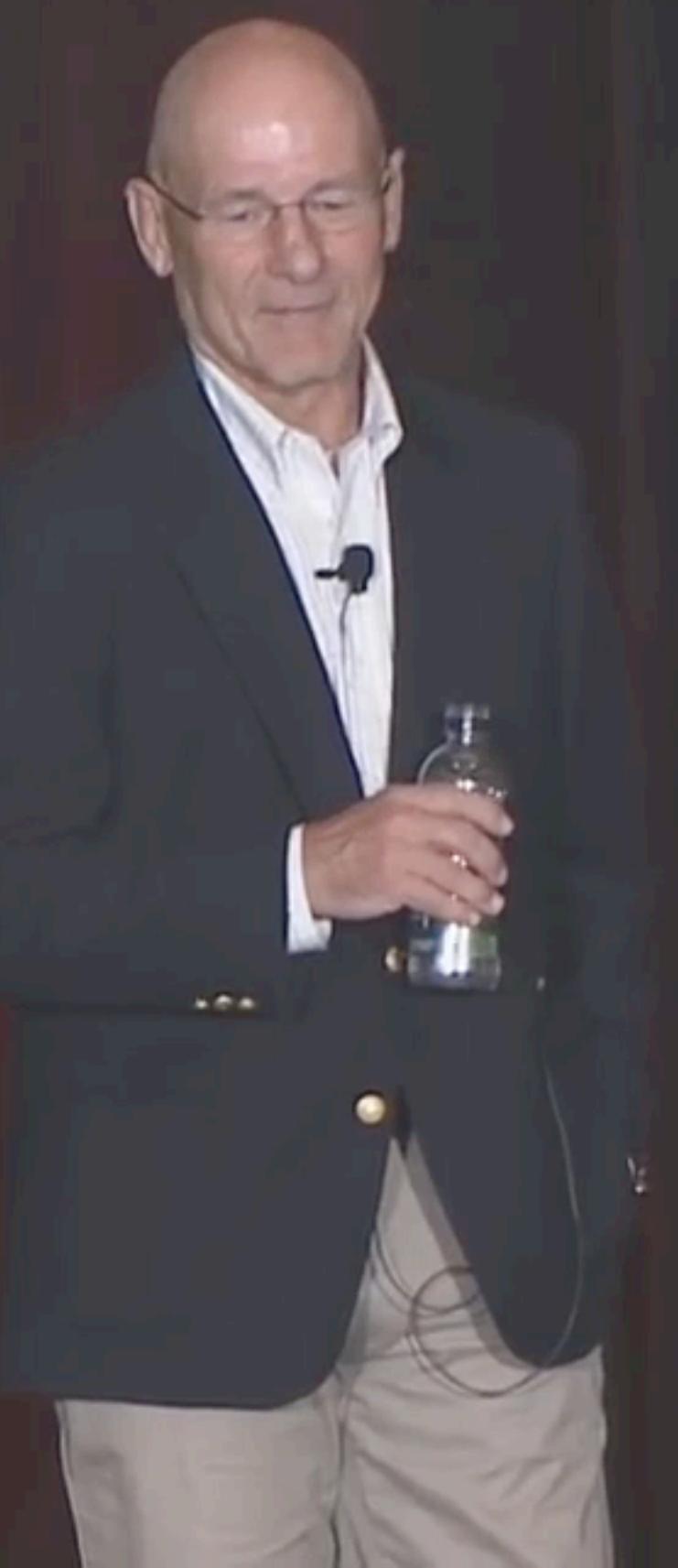
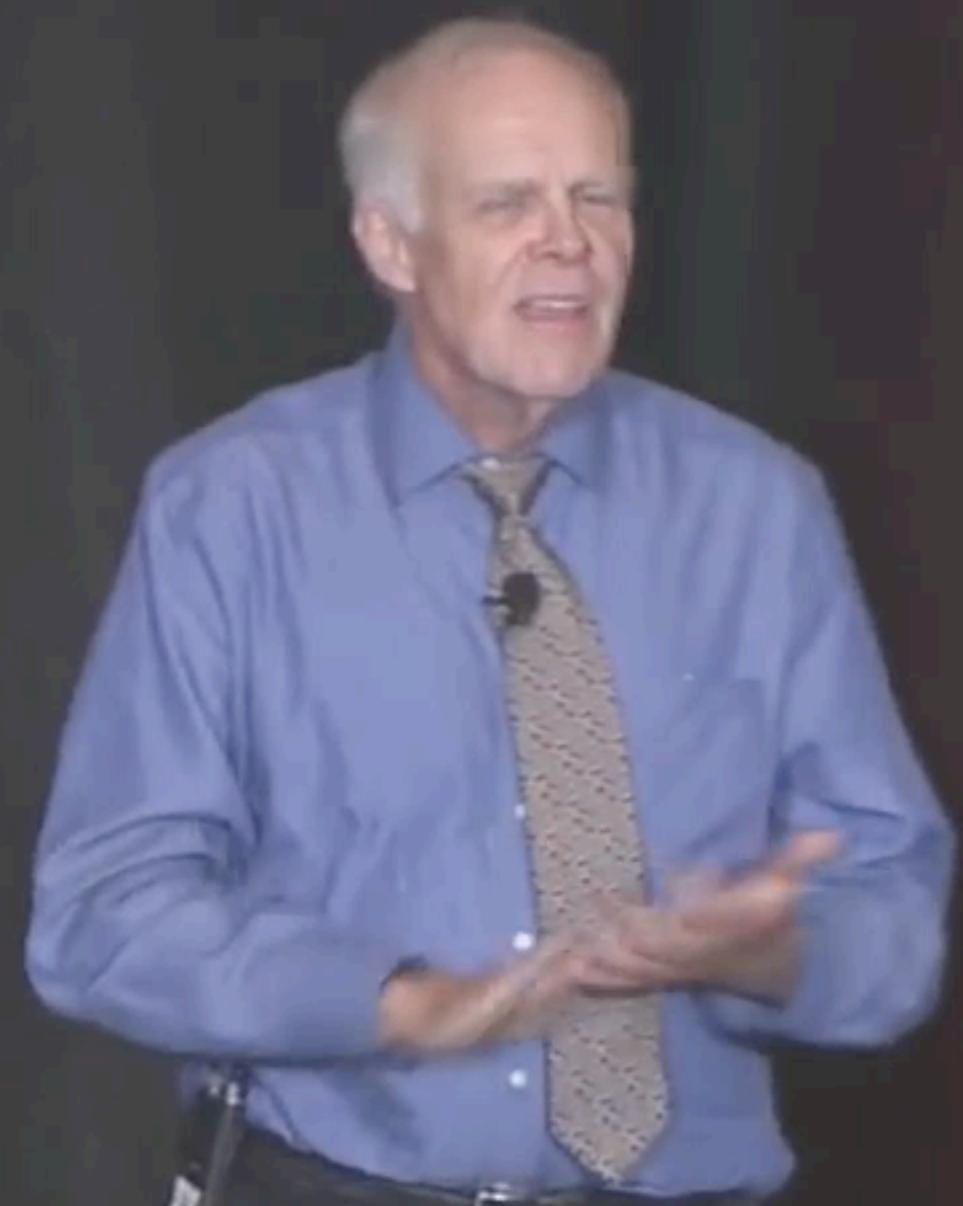
Moore's Law is alive and  
well, but not meaningful  
anymore...



<https://ourworldindata.org/grapher/artificial-intelligence-training-computation>



**ISCA 2018**  
uring Lecture



# Power consumption per transistor

## GB200 GPU Is The Full Blackwell Specs, 500W More Power Than Hopper

During the launch, there was a particularly big confusion surrounding all the Blackwell GPU and platform variants. Jensen stated that Blackwell isn't a GPU, it's an entire platform & the platform has a range of products but they are still based on GPUs. As of right now, NVIDIA has announced three official Blackwell GPU variants.

These include the flagship and full-spec B200 which is being used by the [GB200 Superchip platforms](#). This chip has the highest-rated computing capabilities and has a maximum TDP of 1200W. This is 500 Watts more than the Hopper H100 which featured a 700W TDP. The entire Superchip is equipped with two of these B200 GPUs and a Grace CPU for up to 2700W power (1200W x 2 for B200 + 300W CPU/IO).

World's Most Powerful Chip — Packed with 208 billion transistors, Blackwell-architecture GPUs are manufactured using a custom-built 4NP TSMC process with two-reticle limit GPU dies connected by 10 TB/second chip-to-chip link into a single, unified GPU.

## 61 billion transistors

The XCC has [61 billion](#) transistors. The MCC die for Emerald Rapids has up to 32 cores exposed to the outside world, and probably has 36 cores in the design, again to improve yield. Dec 14, 2023



The Next Platform

<https://www.nextplatform.com> › Compute

[Intel "Emerald Rapids" Xeon SPs: A Little More Bang, A Little ...](#)

The top-line Emerald Rapids chip, the 5th gen Xeon 8592+, will have 64 cores, an improvement from the 60 cores in Sapphire Rapids. The chip will operate at a 1.9GHz frequency that can max out at 3.9GHz in turbo mode. It has 320MB of cache, draws [350](#) watts of power, and fits into two-socket systems. It costs a whopping \$11,600.

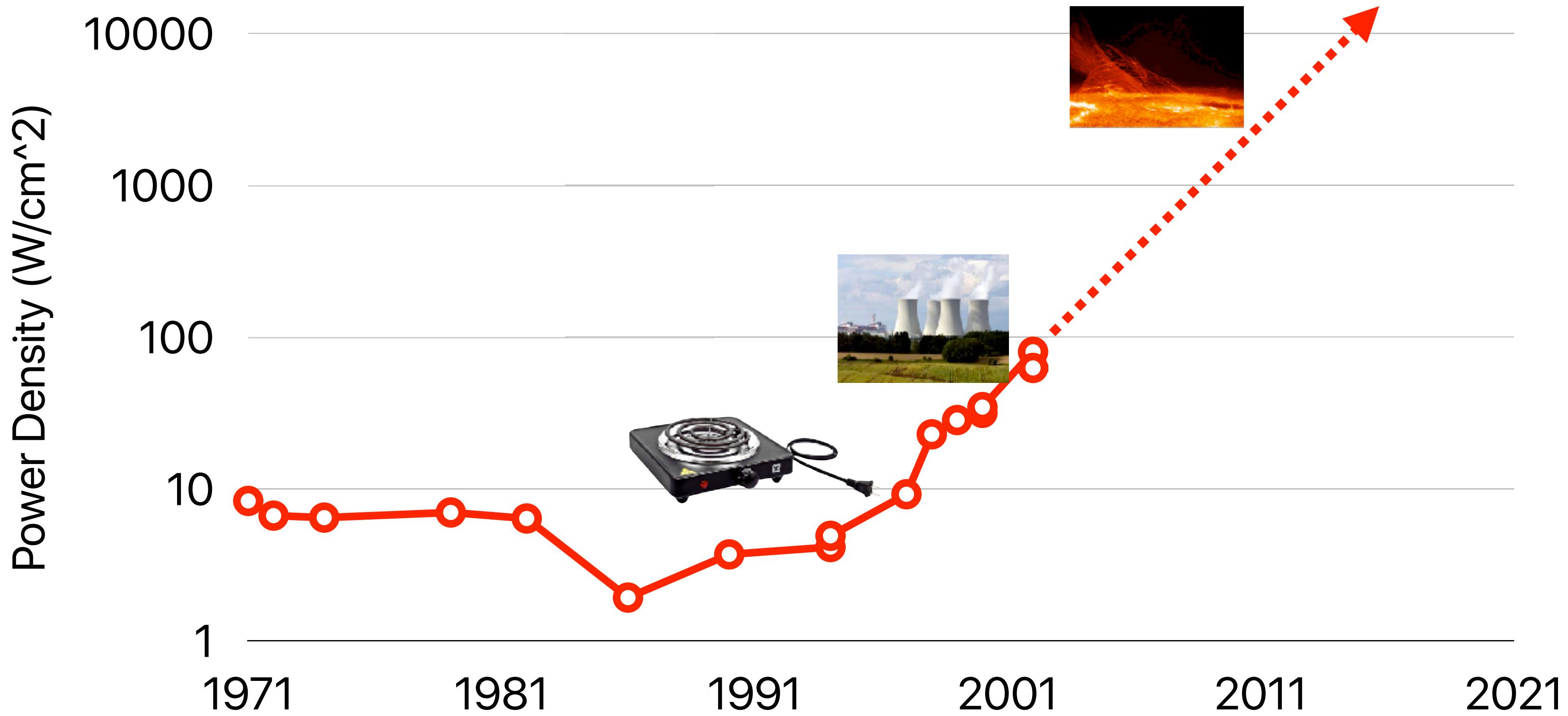
**5.77 W/1B transistors**

TSMC 4nm

**5.73 W/1B transistors**

Intel 7nm

# Power Density of Processors

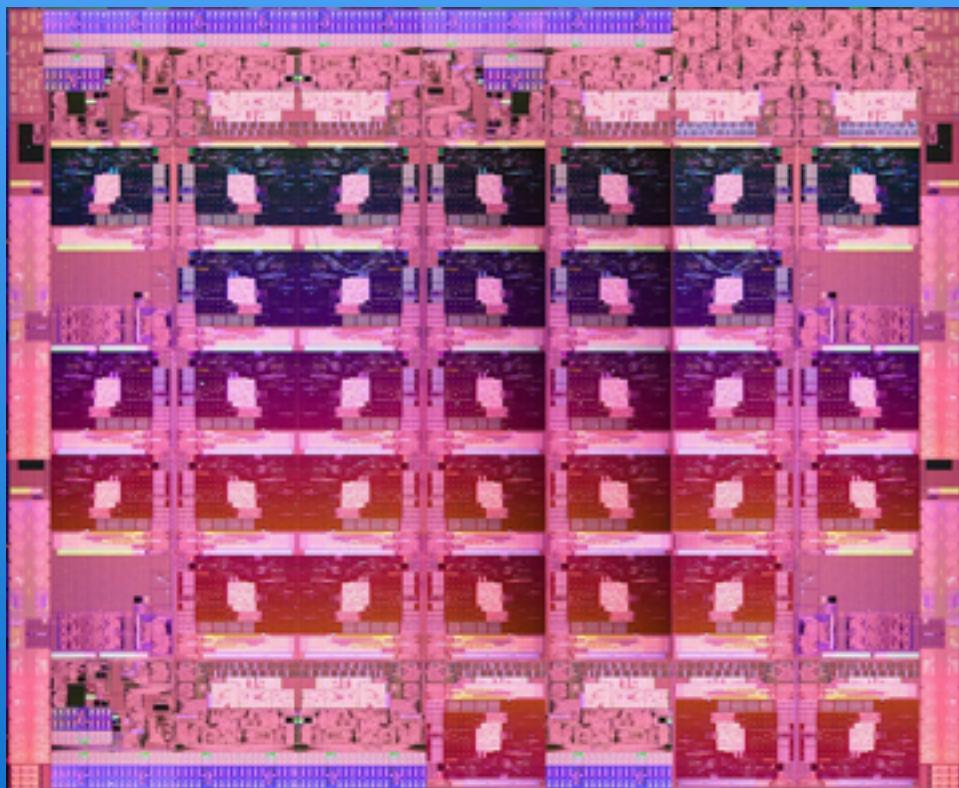


**The smaller size of a transistor,  
the smaller power consumption of it**



# Alternatives to scaling single cores

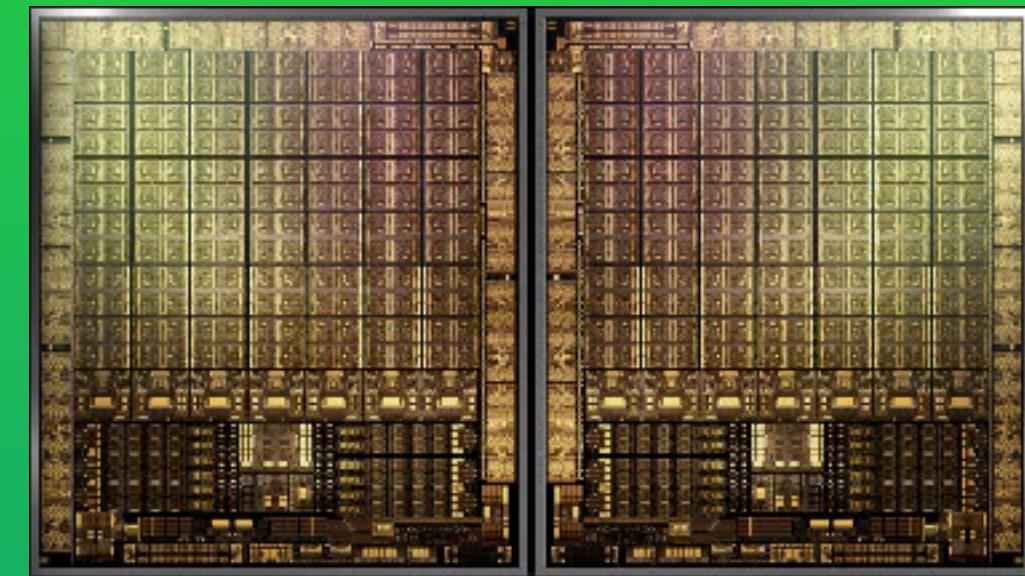
## Multicore Processors



34-core Intel  
Sapphire Rapids

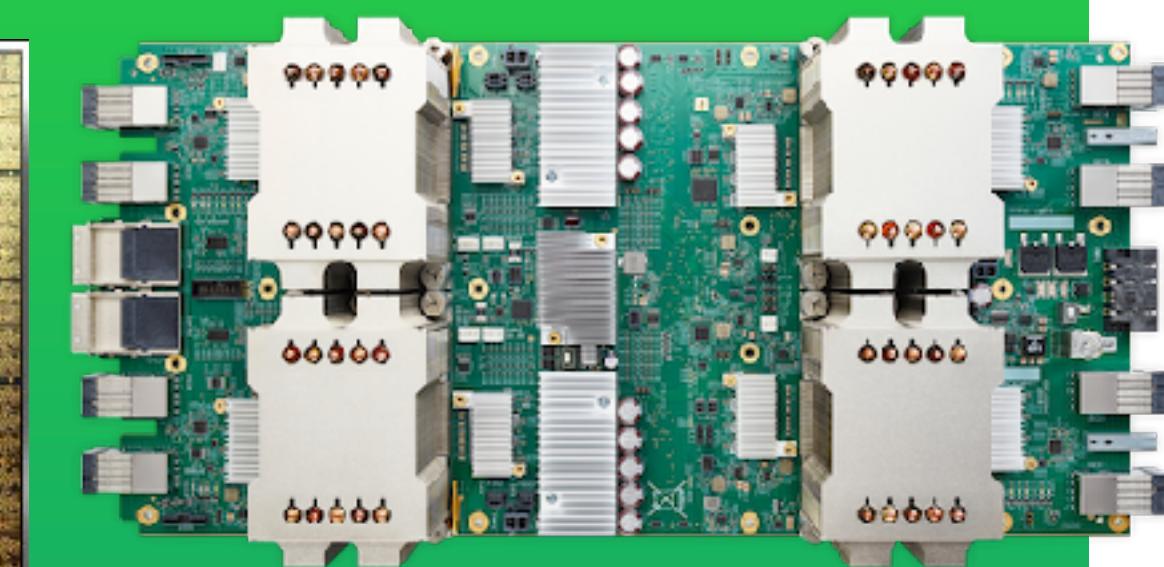
Thread-level parallelism

## Hardware Accelerators



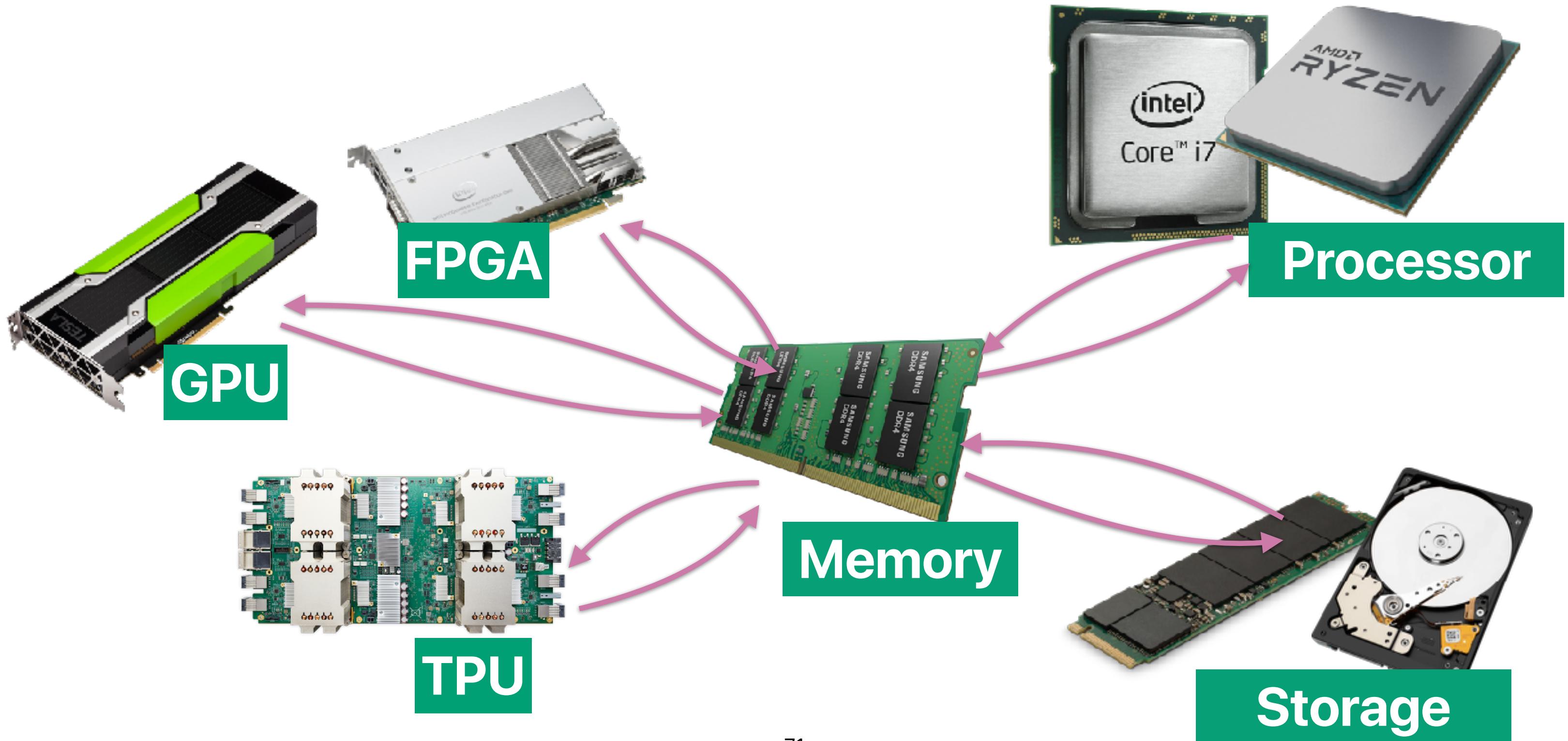
NVIDIA H100  
Graphic Processing Units

Google  
Tensor Processing Units



Data-level parallelism

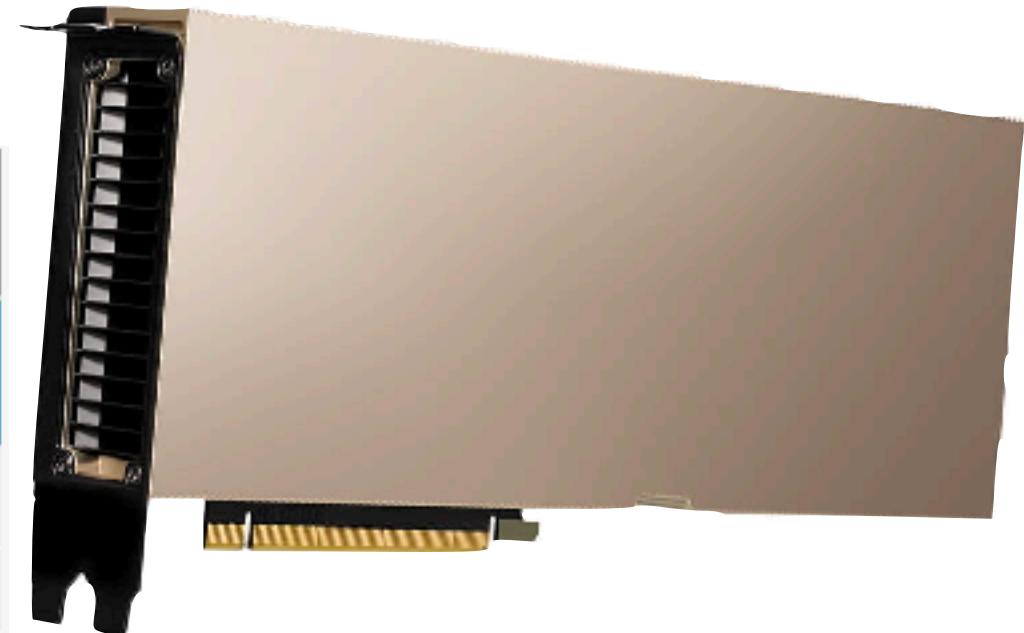
# Heterogeneous Computer Architecture



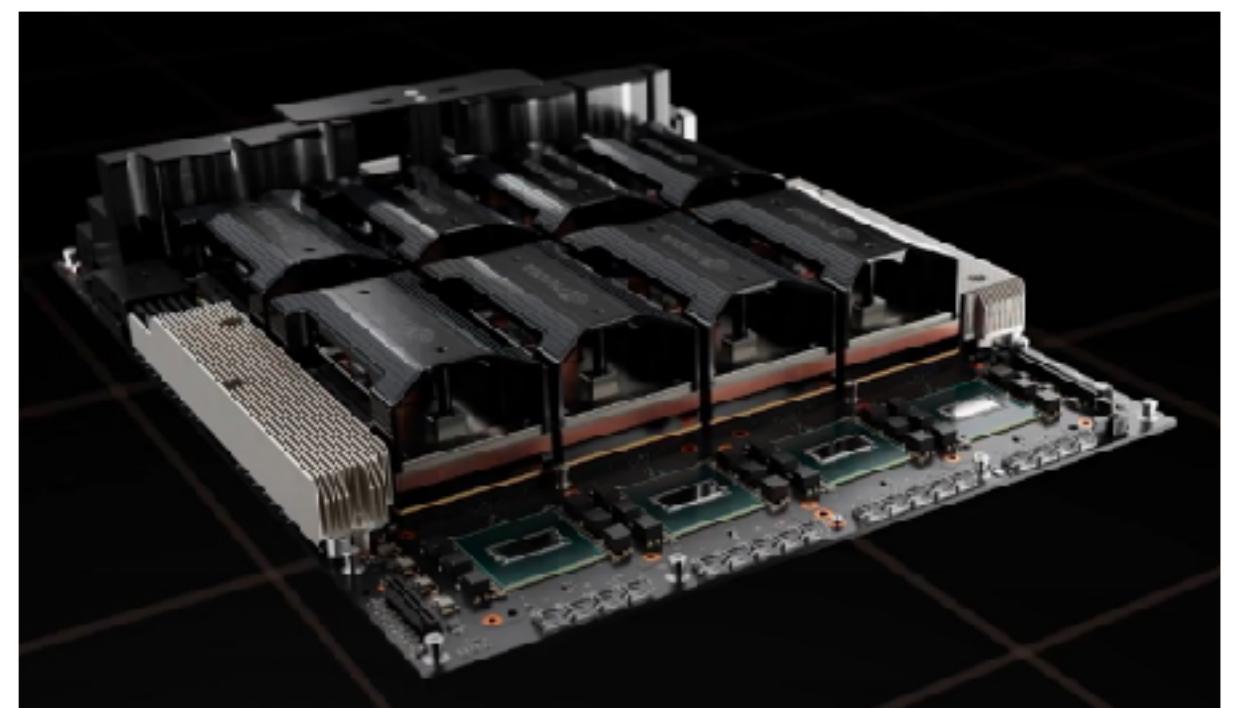
# But still limited by the power consumption

	H100	A100 (80GB)	V100
NVIDIA Accelerator Specification Comparison			
	H100	A100 (80GB)	V100
FP32 CUDA Cores	16896	6912	5120
Tensor Cores	528	432	640
GPU	GH100 (814mm <sup>2</sup> )	GA100 (826mm <sup>2</sup> )	GV100 (815mm <sup>2</sup> )
Transistor Count	80B	<b>1.46x</b> <b>1.75x</b>	54.2B
TDP	700W	400W	21.1B 300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFC
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta
Transistor Count	80B	54.2B	21.1B
TDP	700W	400W	300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFC
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta

**B Transistors B Transistors**



<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>

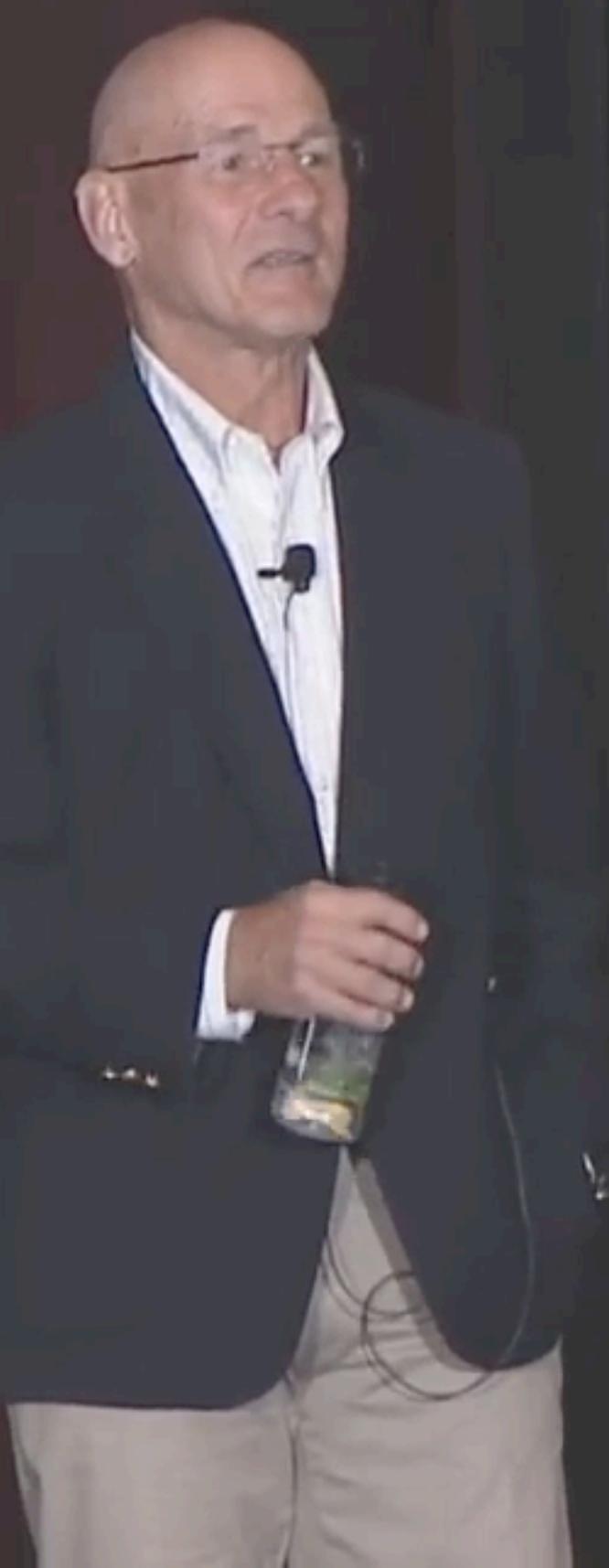
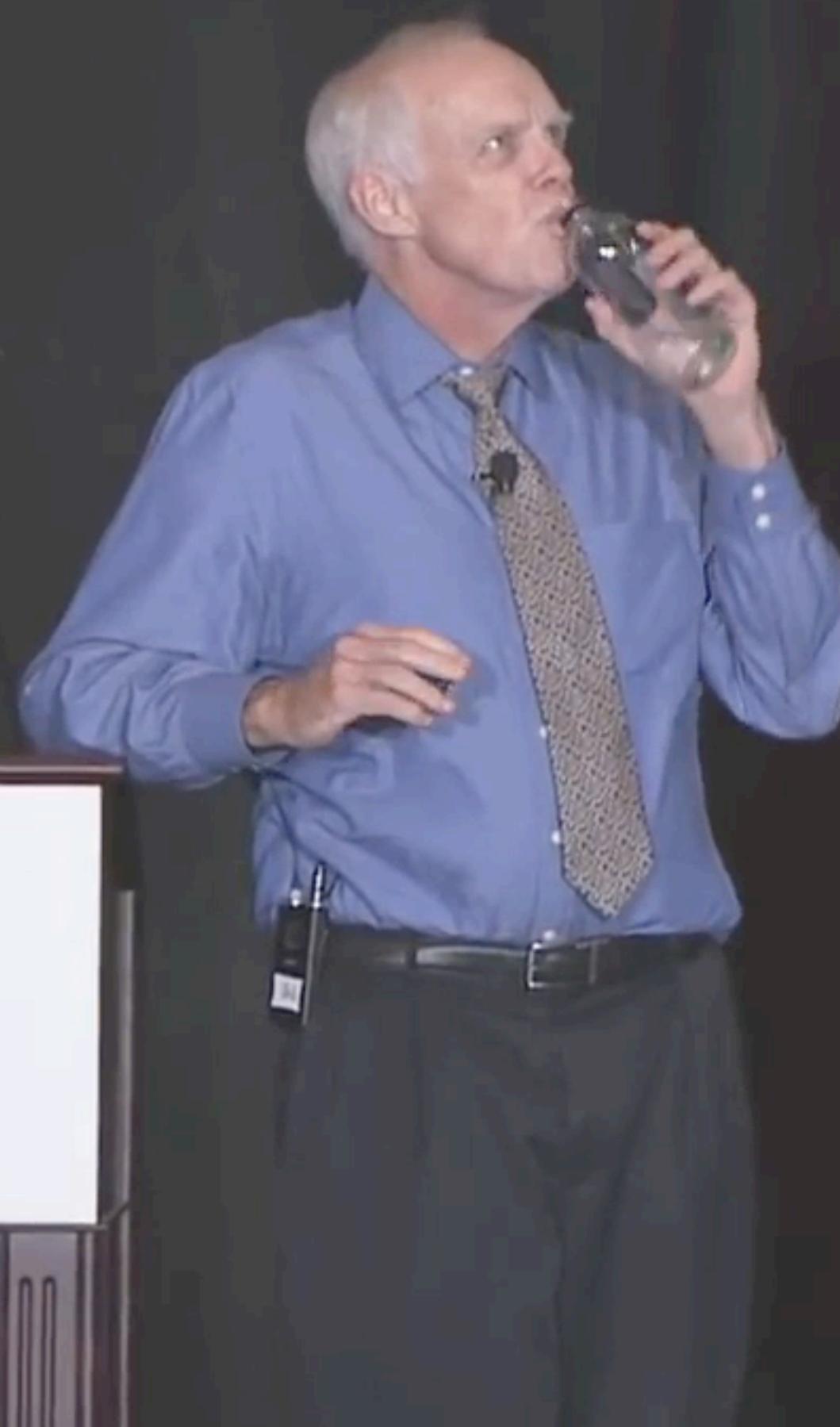
**GPUs surpass the limitations of Moore's Law better than CPUs**



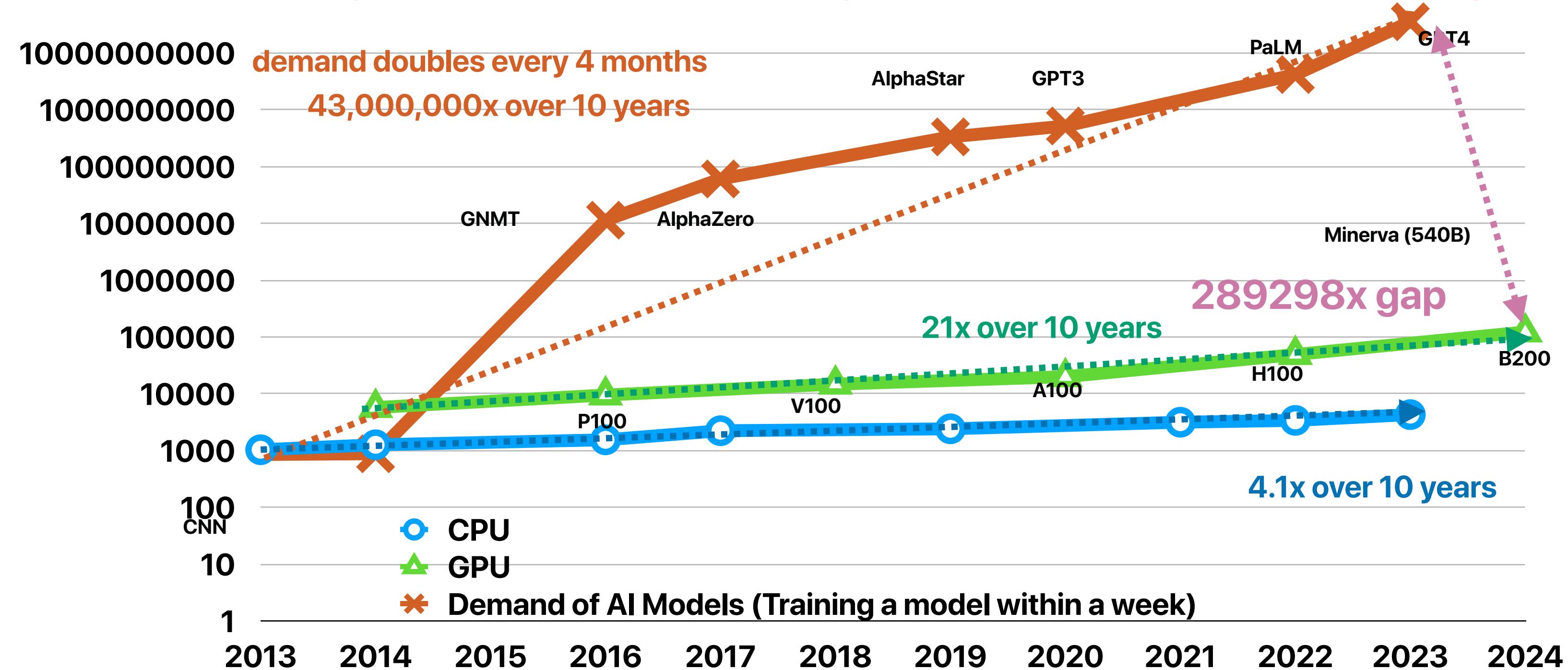


The 45th  
ACM/IEEE  
International  
Symposium  
on Computer  
Architecture  
Los Angeles, USA

A 2018  
g Lecture



# Mis-matching AI/ML demand and general-purpose processing



<https://ourworldindata.org/grapher/artificial-intelligence-training-computation>

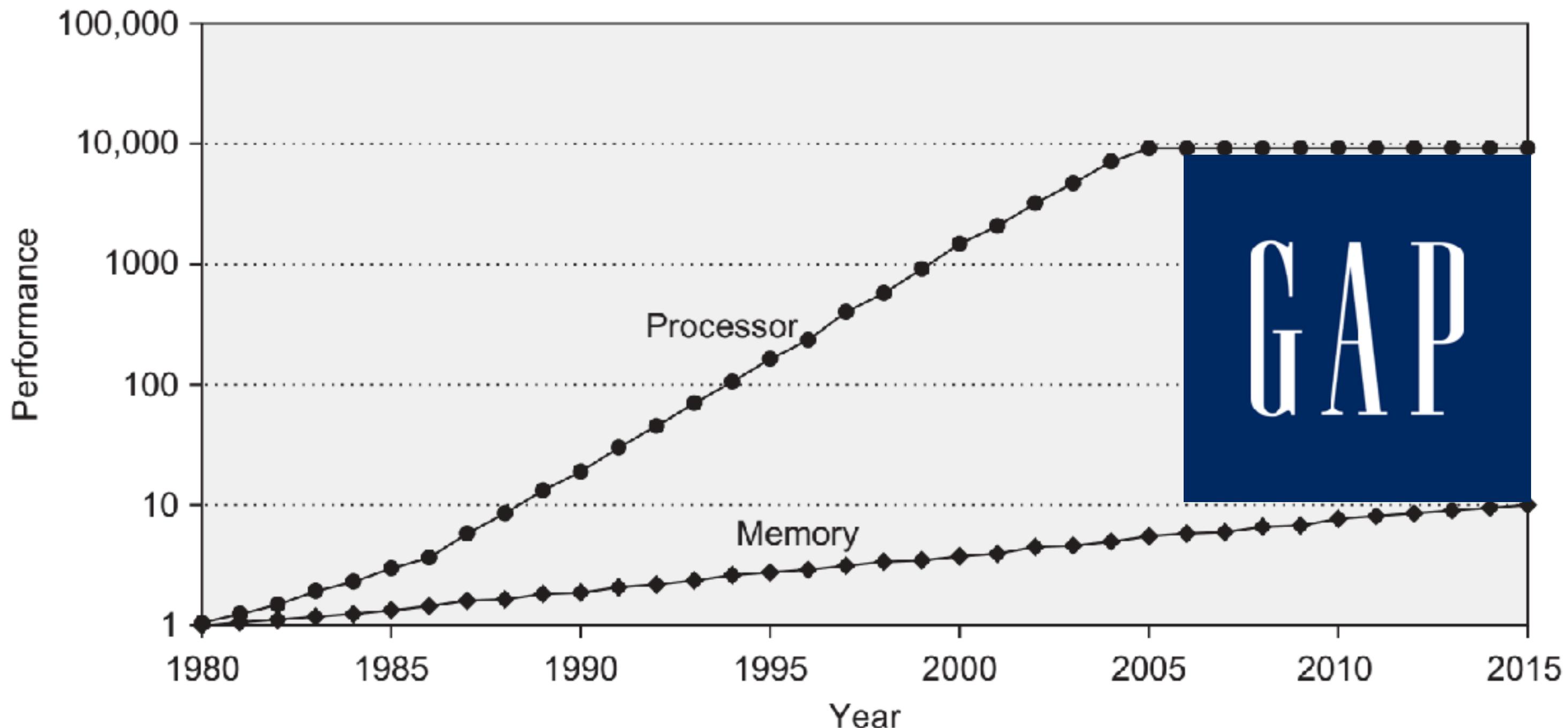
**The performance growth of recent GPU architectures  
can match the demand of efficient AI/ML training**



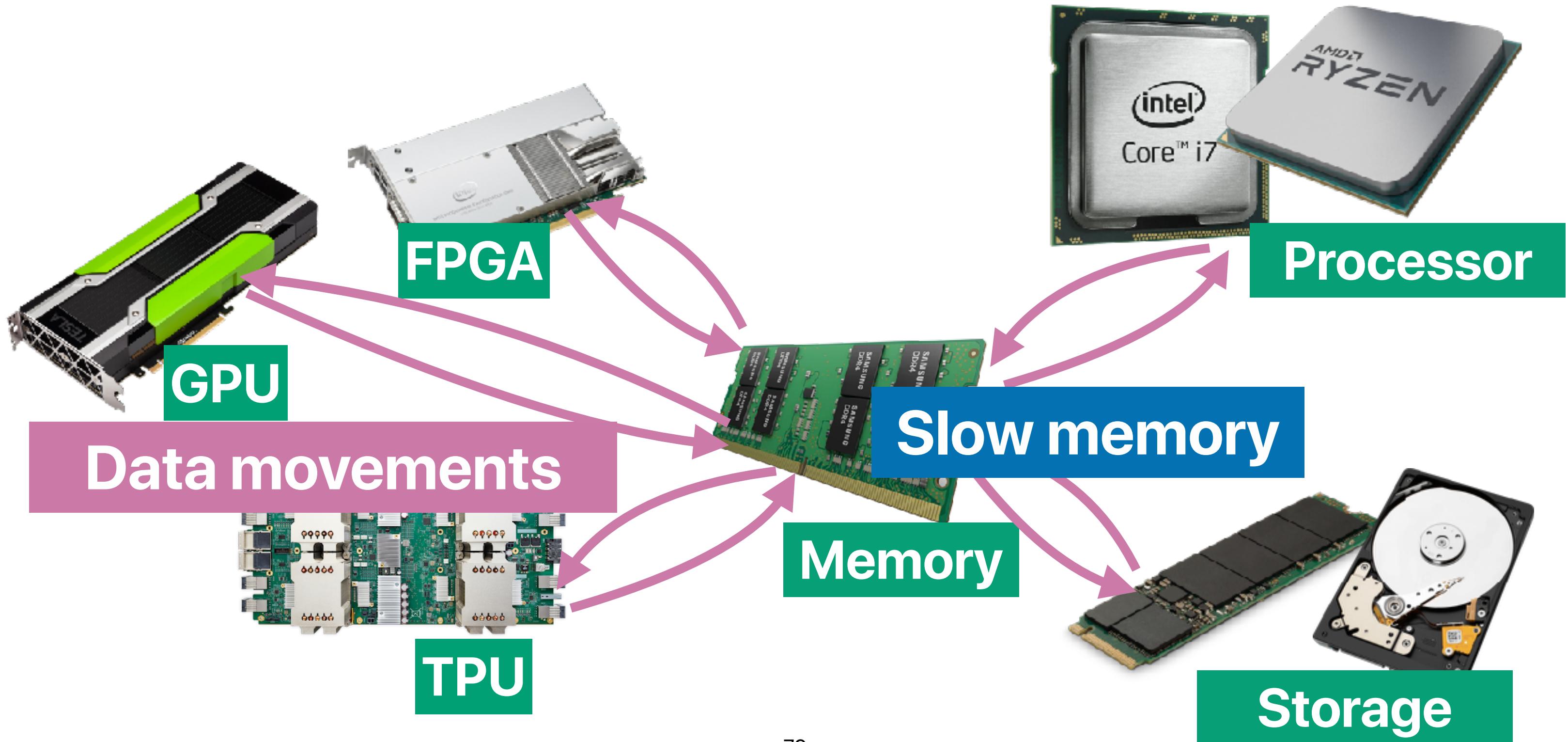
# Take-aways: Why CS203?

- Algorithm complexity does not work well on “real” computers
- Processors/memories are essential for modern computer systems but their performance improves slowly in recent decades
  - Moore’s Law continues, but Dennard Scaling discontinues
  - Cannot catch up the demand of applications — programmers need to do something!

# Performance gap between Processor/Memory



# Heterogeneous Computer Architecture



# Take-aways: Why CS203?

- Algorithm complexity does not work well on “real” computers
- Processors/memories are essential for modern computer systems but their performance improves slowly in recent decades
  - Moore’s Law continues, but Dennard Scaling discontinues
  - Cannot catch up the demand of applications — programmers need to do something!
- We have to rethink about programming as computers become more parallel, heterogeneous, application-specific

# **CS203's vision and missions**

# Vision

- Be programmers who write efficient programs that save time, power, energy, carbon footprint.

# Missions

- Using the knowledge of computer architecture and the general guidelines to identify performance problems and design solutions to tackle the challenges of real-world applications

# Heterogeneous Computer Architecture

- ## Performance
- Performance measurement
  - What affects performance
  - Amdahl's Law
  - Metrics

- ## Memory
- Memory hierarchy
  - Hardware optimizations
  - Software optimizations

- ## Processor
- Pipelining
  - OoO Execution
  - Branch predictions
  - Software optimizations

- ## Parallelism
- Parallel hardware
  - Thread-level
  - Data-level
  - Accelerators
  - Software optimizations

TPU

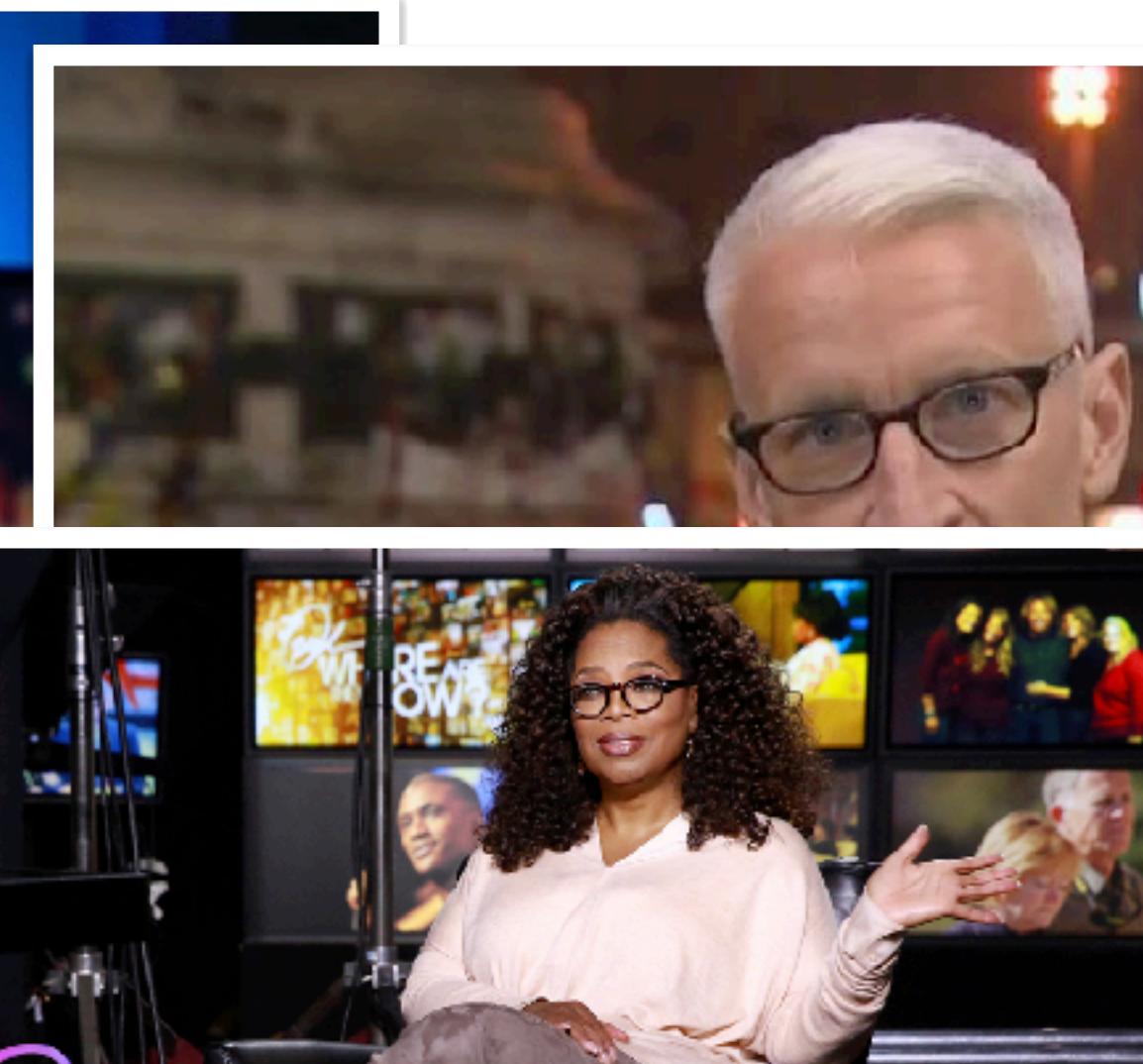
Storage

# Tentative Schedule

	Topic	Due Dates
9/26/2024	Introduction	
10/1/2024	Performance Evaluation (I)	Reading Quiz #1
10/3/2024	Performance Evaluation (II)	Programming Assignment #1 due
10/8/2024	Performance Evaluation (III)	Reading Quiz #2
10/10/2024	Memory Hierarchy (1): The Basics	Assignment 1 due
10/15/2024	Memory Hierarchy (2)	Reading Quiz #3
10/17/2024	Memory Hierarchy (3): Optimizing Cache Performance Applications	
10/22/2024	Memory Hierarchy (4): Programmer's optimizations	Reading Quiz #4
10/24/2024	Virtual Memory	Assignment 2 due
10/29/2024	Virtual Memory (II)	Reading Quiz #5
10/31/2024	Basic Processor Design & Branch Prediction	Programming Assignment #2 due
11/5/2024	Midterm	
11/7/2024	Advanced Branch Prediction	Reading Quiz #6
11/12/2024	OOO Scheduling	Reading Quiz #7
11/14/2024	OOO Scheduling	
11/19/2024	OOO Scheduling	Reading Quiz #8
11/21/2024	SMT & Chip Multiprocessors	Assignment 4 due
11/26/2024	Chip Multiprocessors	
12/3/2024	Dark Silicon	Reading Quiz #9
12/5/2024	TPU, FPGA	Assignment #5 and Programming Assignment #3
12/10/2024	Final Exam	

# **Learning eXperience**

# Most lectures today ...



E SAME JOB **ON CNN TONIGHT**

Paris  
4:02 AM

LIVE  
**CNN**  
7:02 PM PT

I expect the lecture to be...



# 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 **poll everywhere** 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**

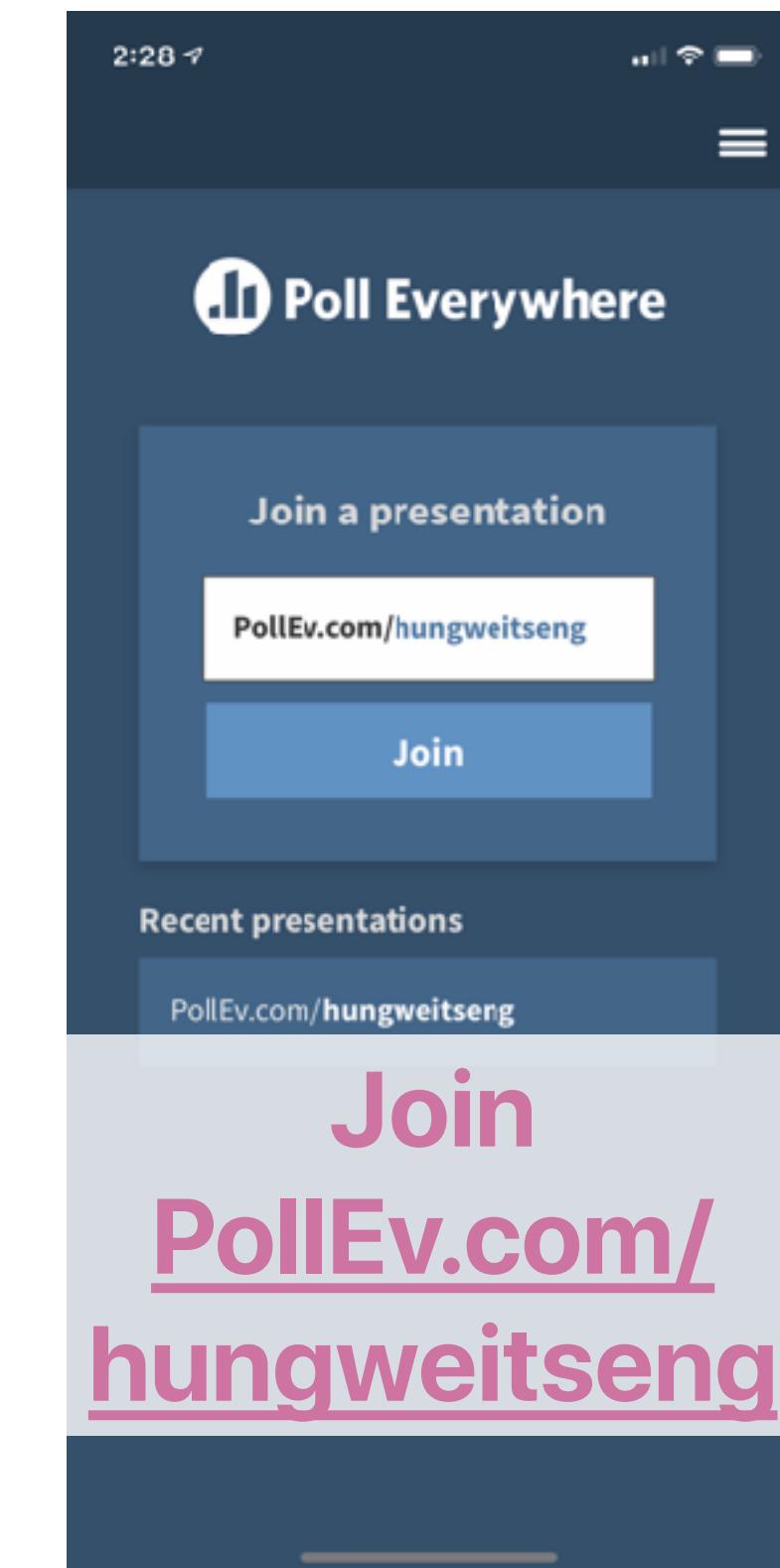
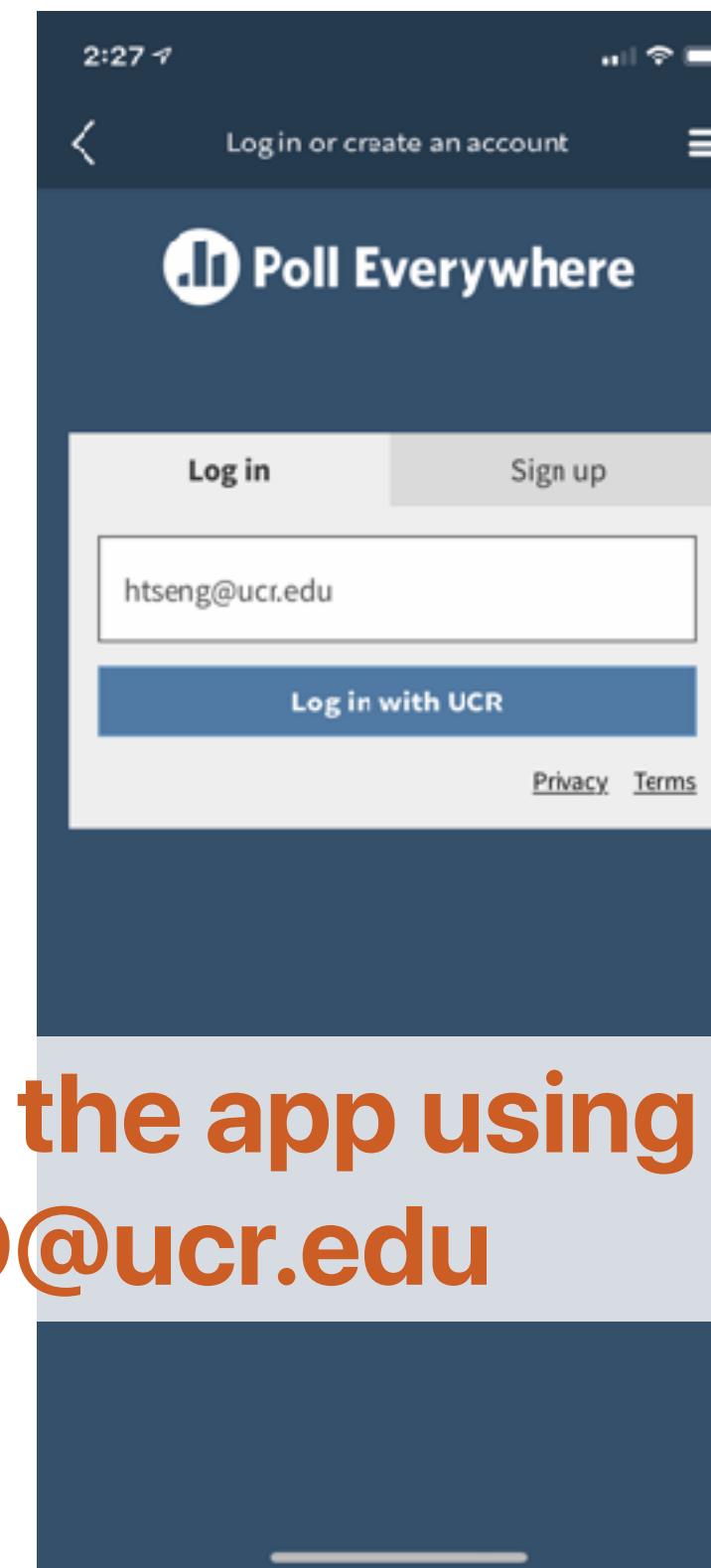
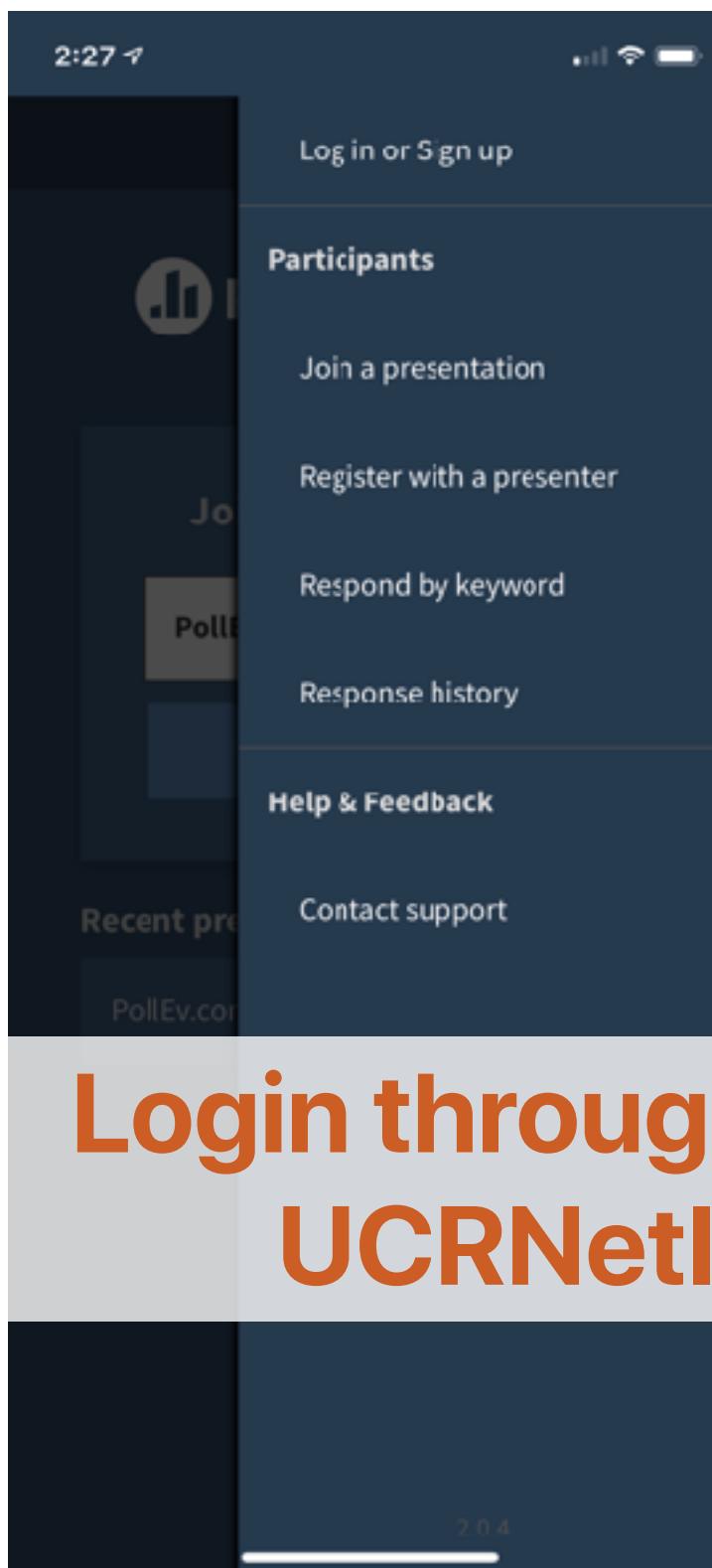
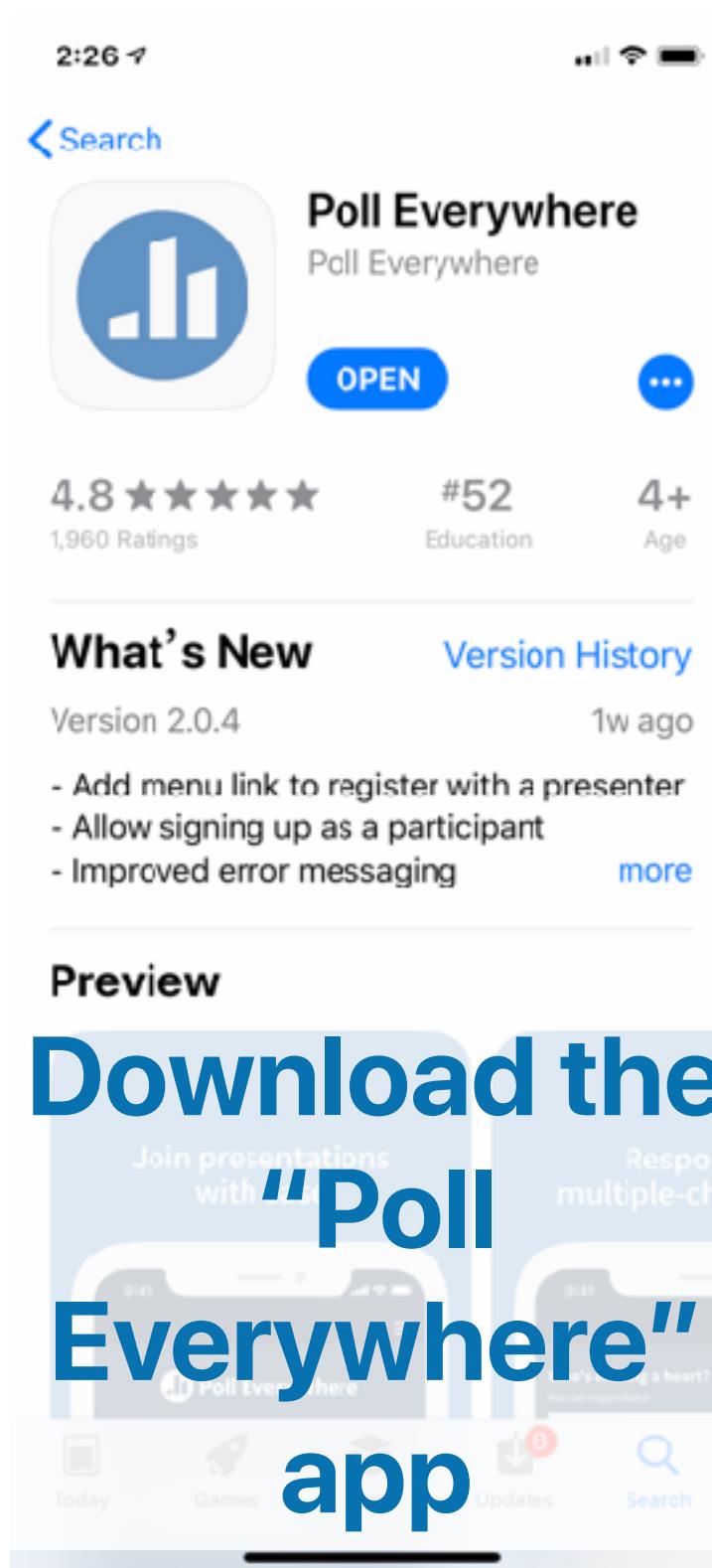
**Think**

**Discuss**

# Before lectures: reading quizzes

- We need to prepare you for peer instruction activities and discussions!
- Reading assignments from
  - Patterson & Hennessy, Computer Architecture: A Quantitative Approach, David Patterson & John Hennessy, Morgan Kaufmann, 6th Edition
  - Assigned research papers/documents
- Reading quizzes — 15% of your grades:
  - On Gradescope
  - Due before the lecture, usually once a week. Check the schedule on our webpage
  - No time limitation until the deadline
  - No make up reading quizzes — we will drop probably your lowest two at least

# About the time of the Lecture — Setup Poll Everywhere



# Peer instruction questions

- The 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 a **full-credit assignment**
    - 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

TECHNOLOGY

## Amazon's CEO Just Declared the Era of Hybrid Work Finally Over

The company is going back to how things were before the pandemic. ↗

EXPERT OPINION BY JASON ATEK, TECH COLUMNIST @JASONATEK

SEP 17, 2024

Google's ex-CEO blames working from home for the company's AI struggles: 'Google decided that work-life balance was more important than winning'

BY ORIANNA ROSA ROYLE

August 14, 2024 at 4:06 AM PDT



FORTUNE

[SIGN IN](#)[Subscribe Now](#)

ance

Leadership

Well

Recommends

Fortune 500



'Working from home was more important than winning,' Google's former CEO Eric Schmidt says.

SEAN GALLUP—GETTY IMAGES

$$ET = IC \times CPI \times CT = 5602640 \times 0.20$$

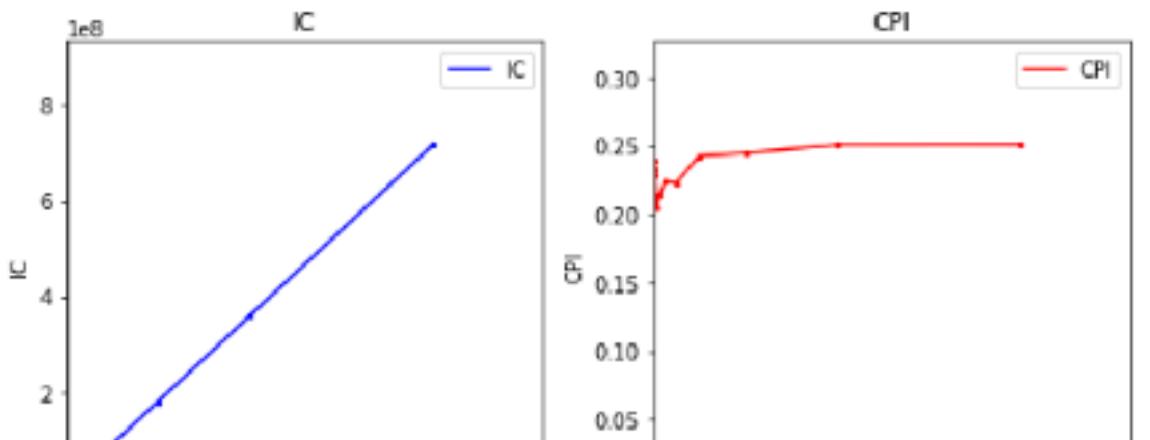
# Assignments

- Measure/Analysis/Programing using Jupyterhub/Jupyter notebooks
- Watch this before your first assignment!  
[https://youtu.be/m7OoY8y\\_lsk](https://youtu.be/m7OoY8y_lsk)
- Submit through gradescope
  - You have to click the GitHub classroom link to begin with (will post the link of each assignment on the course webpage once released)
  - You should use the escala.org/datahub service to finish your assignment
  - Submit your GitHub repo contains the CS203 assignment through Gradescope
  - An autograder will grade your assignment immediately and you can resubmit as many times as you want before the deadline—the earlier you start, the higher chances you can get full credits
  - There is no regrading after the deadline

Now, let's plot the result and verify your prediction and see how ev

```
[9]: df=render_csv("array_size.csv", columns=columns, average=True)
plotPE(df=df, lines=True, what=[('size', "IC"), ("size", "CPI")])
display_mono(df)
```

	size	ET	IC	CPI	M
0	10000	0.000042	703477.840000	0.238585	3995.2118
1	20000	0.000083	1404491.760000	0.233374	3970.8904
2	40000	0.000160	2803566.080000	0.228001	3983.0703
3	80000	0.000287	5604217.320000	0.204585	3988.8586
4	160000	0.000600	11204847.920000	0.213550	3988.0081
5	320000	0.001257	22406745.080000	0.223729	3989.3769
6	640000	0.002506	44812777.640000	0.222840	3985.2323
7	1280000	0.005459	89617166.120000	0.242636	3983.1877
8	2560000	0.011036	179228929.520000	0.244669	3973.7334
9	5120000	0.022575	358452827.360000	0.250542	3978.1896
10	10240000	0.045183	716890078.920000	0.250808	3979.4289



**3 hrs.**

**Attending the  
Lecture**

**4 hrs.**

**Reading the  
textbook/  
papers**

**5 hrs.**

**Assignments**

# **Logistics**

# Grading Breakdown

	In-person session
Reading Quizzes	15% Drop lowest 2
Assignments	25% Drop lowest 1
Participation	Count as one assignment. Get full credit if you show up on 50% of PI questions If you don't show up, please do well on every assignment!
Midterm	25% In-person only test — closed book
Final	35% In-person only test — closed book

- Review the course website for policies
- Check your grades [https://www.escalab.org/my\\_grades/](https://www.escalab.org/my_grades/)

# Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- E-mail: cs203 @ escal.org
- 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/RayTracing accelerators
  - Or anything else fun — we have an OpenUVR project recently
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial



# Teaching Assistant

- Jinyoung Choi
- Office hours: Tu 4p-5p & Th 4p-6p @ WCH 110
- E-mail: cs203 @ escalab.org
- Fun fact: Choi, Jinyoung (a.k.a, CJ) chose his english nickname based on a GTA character.



# Subscribe to our google calendar!



**Please be aware that TA/tutors are unionized and under contract. They are not supposed to work outside normal hours**

Lecture: TuTh 11a – 12:20p

Where: [Watkins Hall | Room 1000](#)

# The website

- <https://www.escalab.org/classes/cs203-2024fa/>
- Calendar
- Schedule
- Slides
  - Preview — for the ease of note taking
  - Release — the actual slides

[Schedule and Slides](#)[Assignments](#)[Logistics](#)

## Instructor

Hung-Wei Tseng

email: cs203 @ escalab.org

Office Hours: W 2p-4p @ WCH 406

## Teaching Assistant

email: cs203 @ escalab.org

Office Hours:

## Other important links

- Assignment & Reading quizzes submission through [Gradescope](#): <https://www.gradescope.com/courses/1285990262>
- Discussion Forum on Discord: <https://discord.com/channels/1285989312076189787/1285990262>
- Lecture recordings: [Youtube Channel](#)
- Calendar: [Google Calendar](#)

## CS203 2024 Spring

今天 &lt; &gt; 2024年9月

列印 週 月 待辦

週日	週一	週二	週三	週四	週五	週六
9月 1日	2	3	4	5	6	7
8	9	10	11	12	13	14

# Summary of course resources

- Lectures:
  - In-person @ Watkins Hall | Room 1000
  - Repository on **Youtube**: <https://www.youtube.com/profusagi>
- Schedule, slides, grades on **course webpage**:  
<https://www.escalab.org/classes/cs203-2024fa/>
- Discussion on **discord**
- Reading quizzes and assignments through **gradescope**:  
<https://www.gradescope.com/courses/813375>
- Working environment on <https://escalab.org/databub>
- Office Hours & Locations
  - CJ — Tu 4p-5p, Th 4p-6p WCH 110
  - Hung-Wei Tseng — W 2p-4p WCH 406



# 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.

# Frequently asked questions (FAQs)

- Do I need to attend every lecture?
  - Attending the lecture is never “required”, but strongly encouraged
  - In-person students have better learning outcome even though given the same difficulty of midterm/final examines based on the experience last year
  - We encourage you to participate in in-class learning activities (peer instructions) through giving you a “full-credit assignment” if you can answer 50% of the PI questions.
  - If you have difficulties coming to the 50% of class, you can still earn all credits if you ace all assignments.
  - Please **do not** email the staff regarding individual instances of no show
- Do we podcast the class online?
  - We record all lectures and post them on the same day as the lecture on YouTube  
<https://www.youtube.com/profusagi>
- Are examines in-person?
  - Yes — for fairness and logistics regarding proctoring

# FAQs (cont.)

- When will I know my assignment grade?
  - A few minutes after your submission — everything is automatically graded
- How many times can I submit my assignment?
  - As many times as you want before the deadline
  - Start early and get feedback early — more chances to ace the assignments
- Do we have an eLearn/canvas page?
  - No — eLearn/canvas is broken in many aspects and the grade book is never accurate
  - Please refer to the class webpage for the most comprehensive links

# FAQs (cont.)

- I cannot login [escalab.org/datahub](http://escalab.org/datahub)
  - Please email [cs203@escalab.org](mailto:cs203@escalab.org) as soon as possible (M-F 9a-5p)
- I cannot login Gradescope
  - You need to officially enrolled to have access to the Gradescope of the class — I cannot help you unfortunately
- Are textbooks required?
  - Yes
- Do we have Zoom office hours
  - No
  - Please use discord to ask questions!
- What if I need help during the weekend?
  - Unfortunately, we don't guarantee we can answer you questions either through piazza or e-mail during the weekend — you should plan your schedule carefully

# 2024 Fall



# Announcements

- Check our website
- Login to datahub, gradescope, discord
  - Let us know if you have any issue — [cs203@escalab.org](mailto:cs203@escalab.org)
- Complete the course agreement by **tomorrow**
  - Please make sure your gradescope account is associated with your [UCRNetID@ucr.edu](mailto:UCRNetID@ucr.edu) — you cannot receive any credit if there is a mismatch
- Reading quiz on Gradescope due **next Tuesday before the lecture**
  - Unlimited time before Tuesday's lecture — one trial only
- Programming assignment #1 & Assignment #1 are released — working on datahub!
  - Watch this video before you start! [https://youtu.be/m7OoY8y\\_lsk](https://youtu.be/m7OoY8y_lsk)



# Computer Science & Engineering

203

つづく

