

First Day of
CSE142: Computer Architecture: Software
Perspective (2024 Summer Session II)

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: htseng @ ucsd.edu
- 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
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial



What do you care as a computer scientist?



Algorithms
Data Structures
Computer Architecture
Programming Languages
User Interfaces

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)



Computer Architects — Hidden Figures

BASED ON THE UNTOLD TRUE STORY

HIDDEN FIGURES



COMPUTER ORGANIZATION AND DESIGN MIPS EDITION

THE HARDWARE/SOFTWARE INTERFACE

SIXTH EDITION



MK
MORGAN KAUFHANN

DAVID A. PATTERSON & JOHN L. HENNESSY

Computer Architecture

Enables

Deep Learning

**Why should I care about
“Computer Architecture”**

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

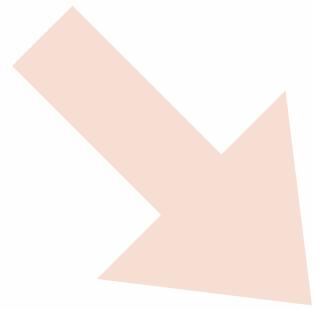
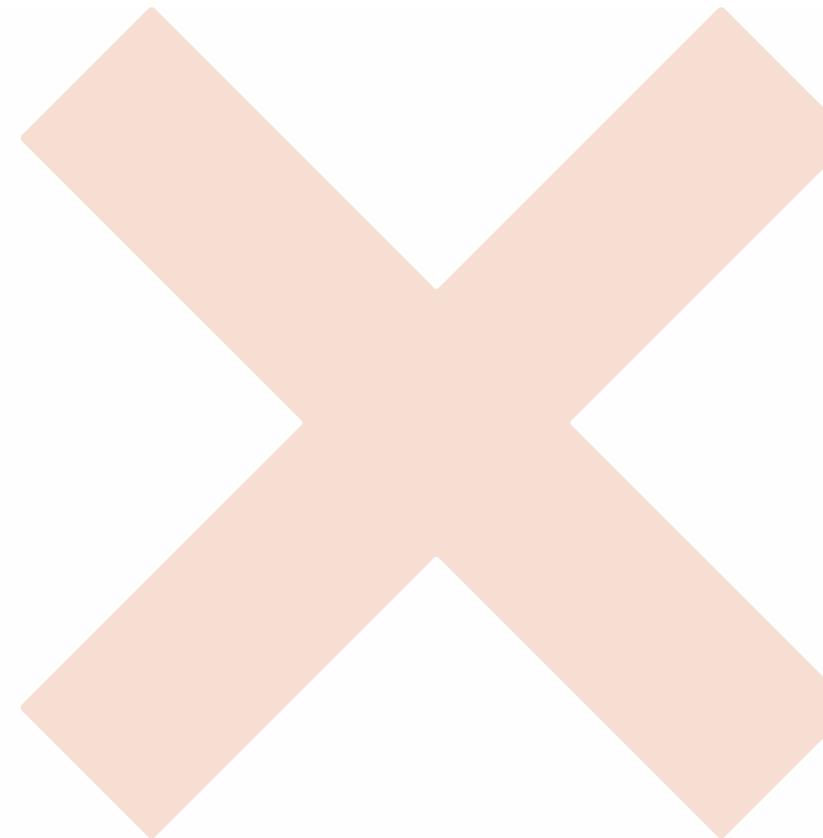
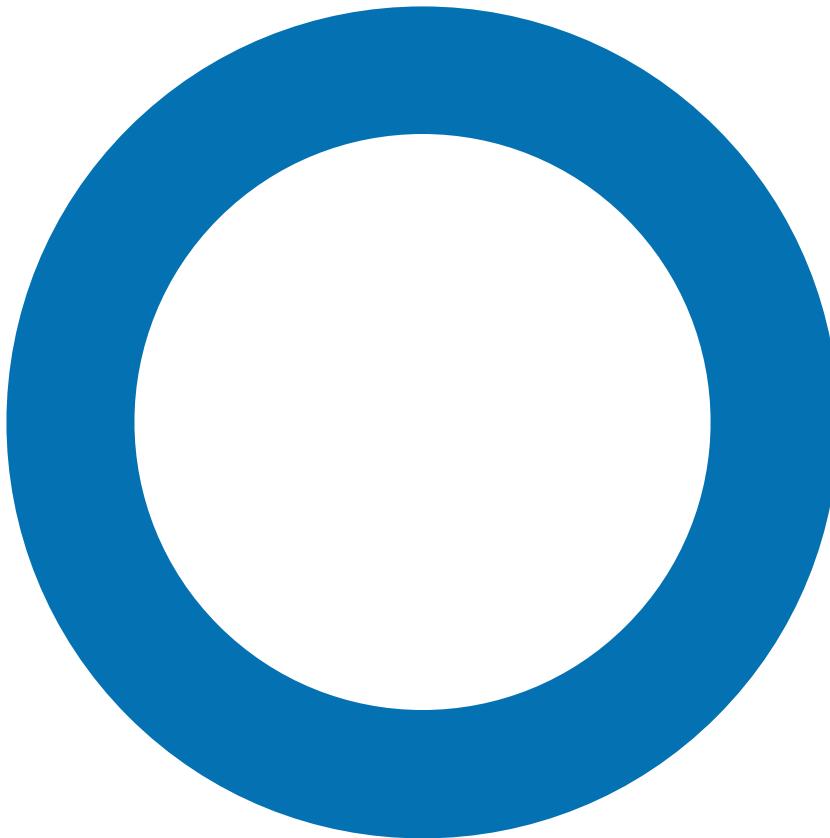
5 : the manner in which the components of a computer or computer system are organized and integrated

// different program architectures

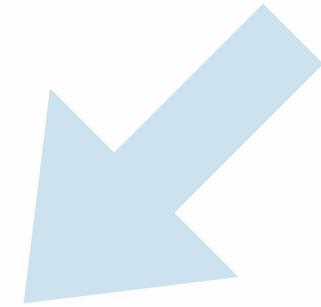
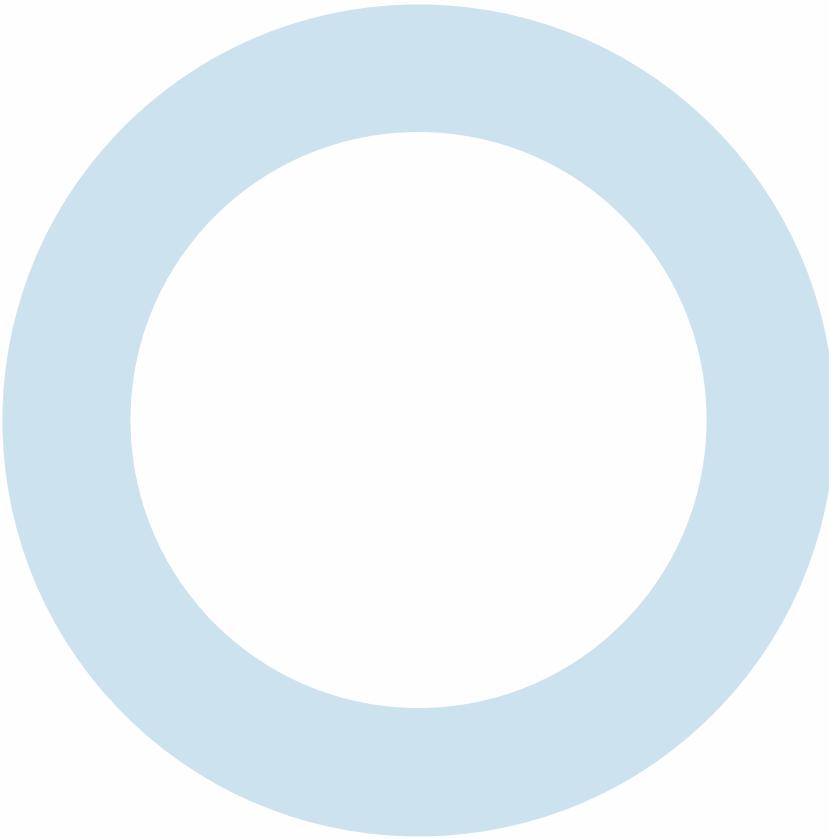
**The manner in which the components
of a computer or computer system are
organized and integrated**

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

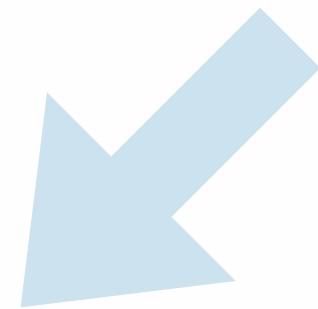
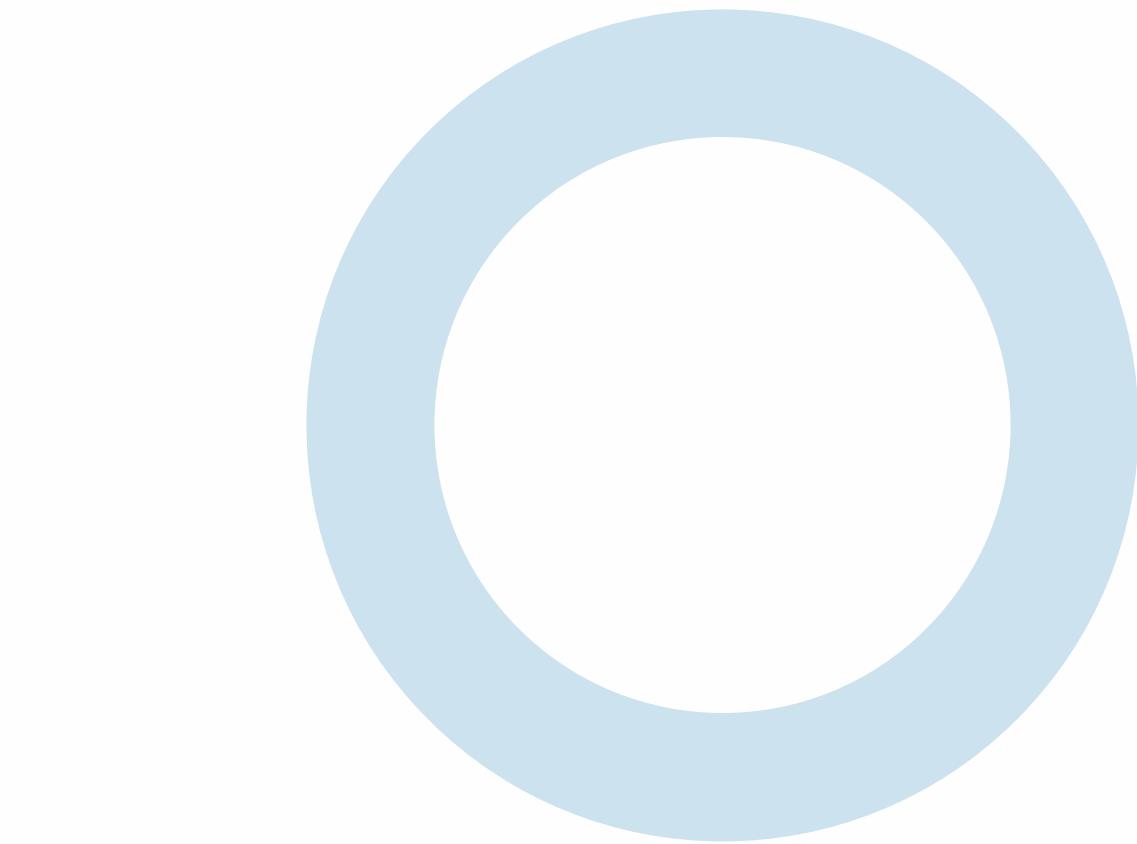
Processors and memories are essential for most modern general-purpose computers



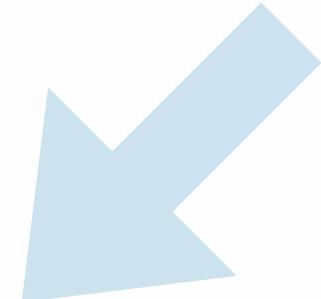
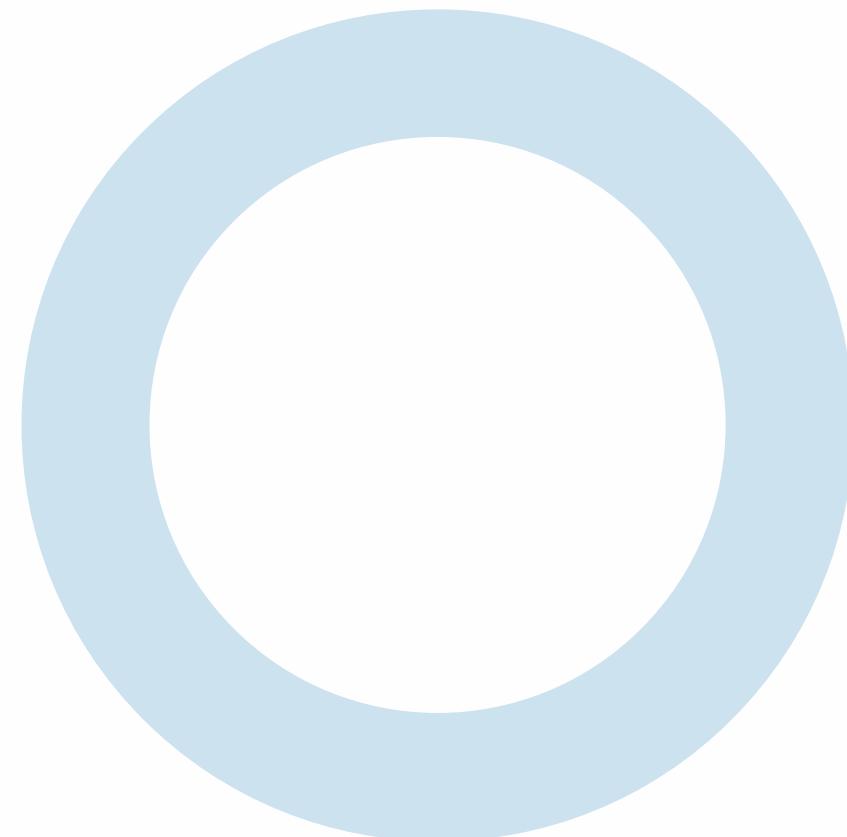
Moore's Law is currently discontinued



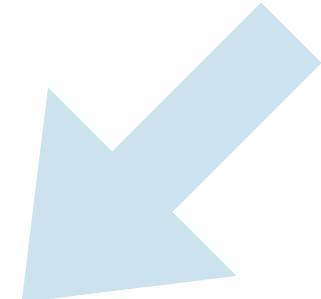
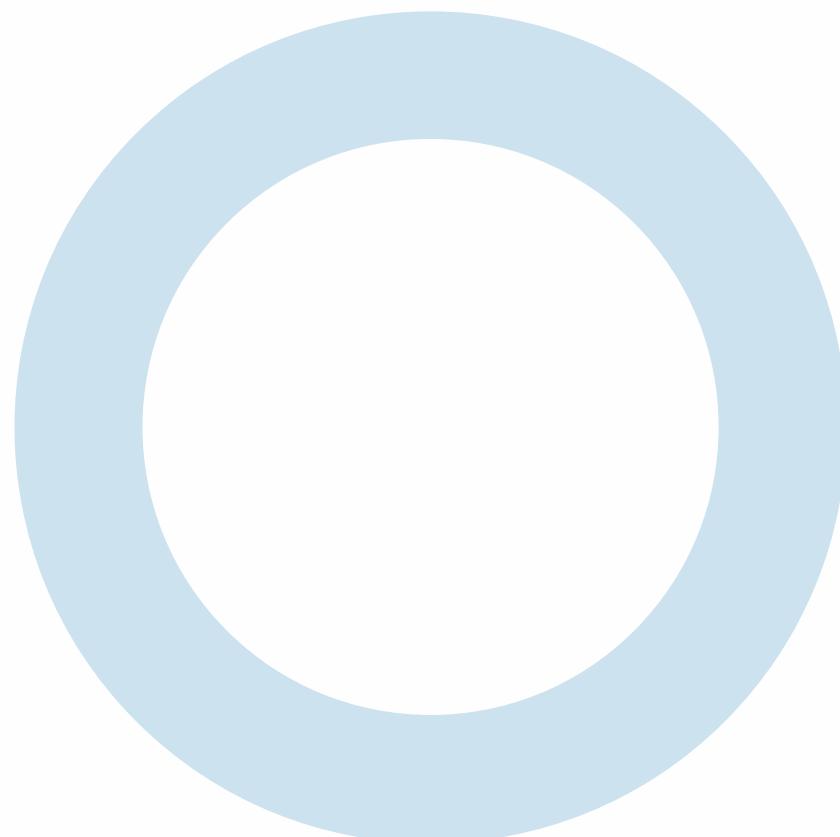
**On the same machine, programs with fewer lines
of code will have shorter execution time**



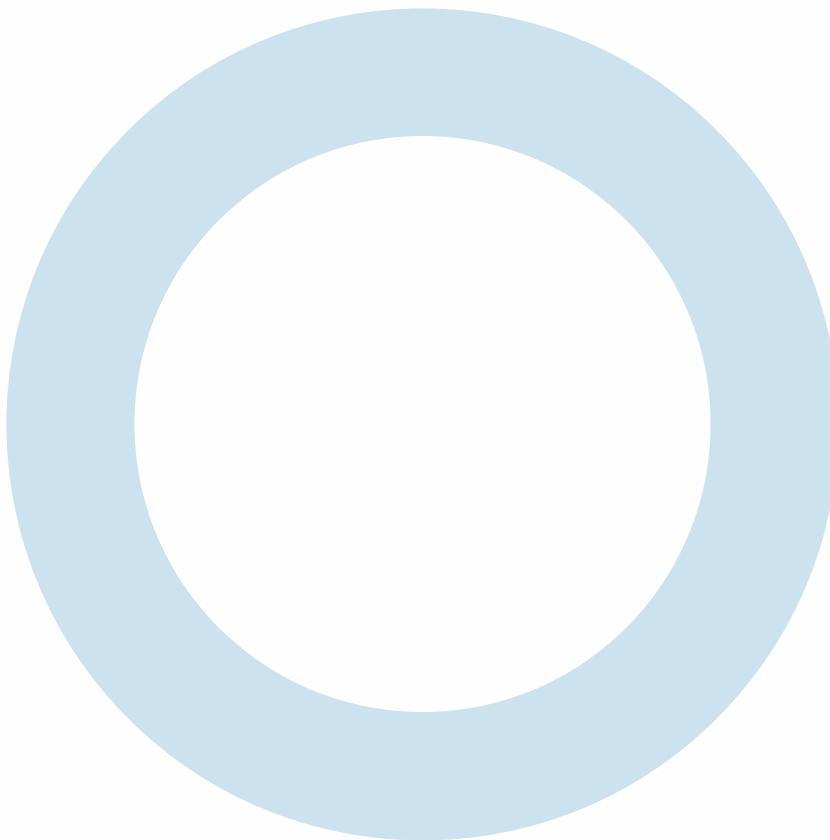
On the same machine, an algorithm with lower computational complexities will have shorter execution time when solving the same problem with the same input



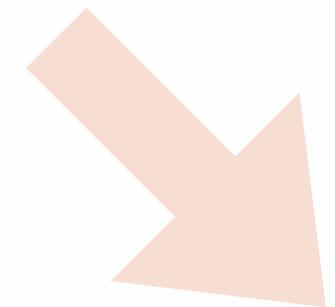
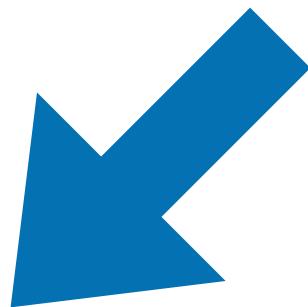
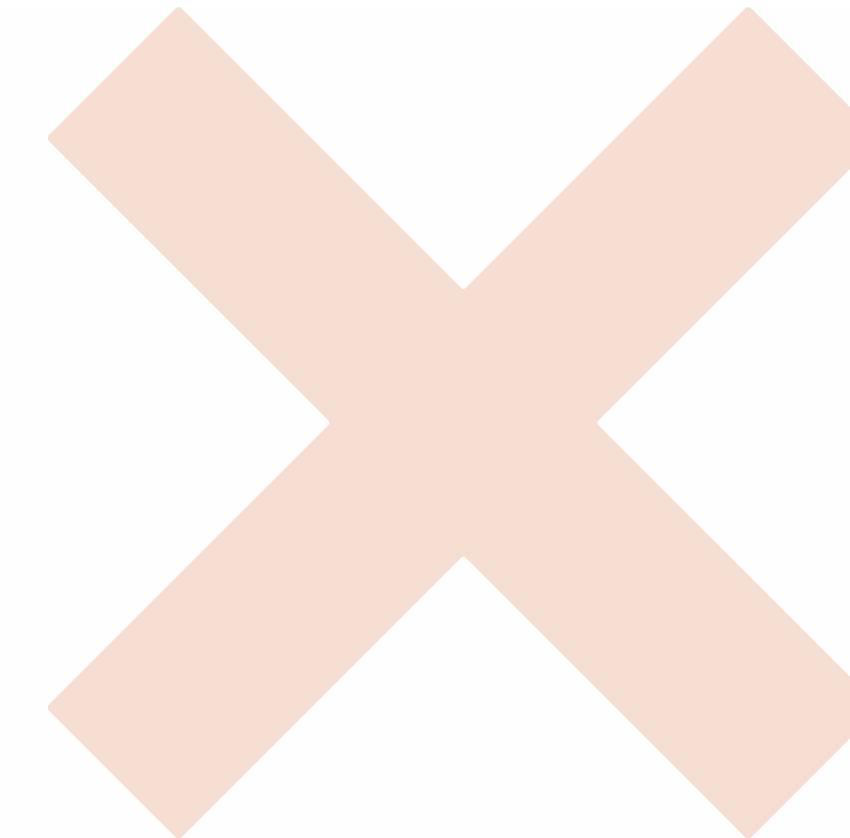
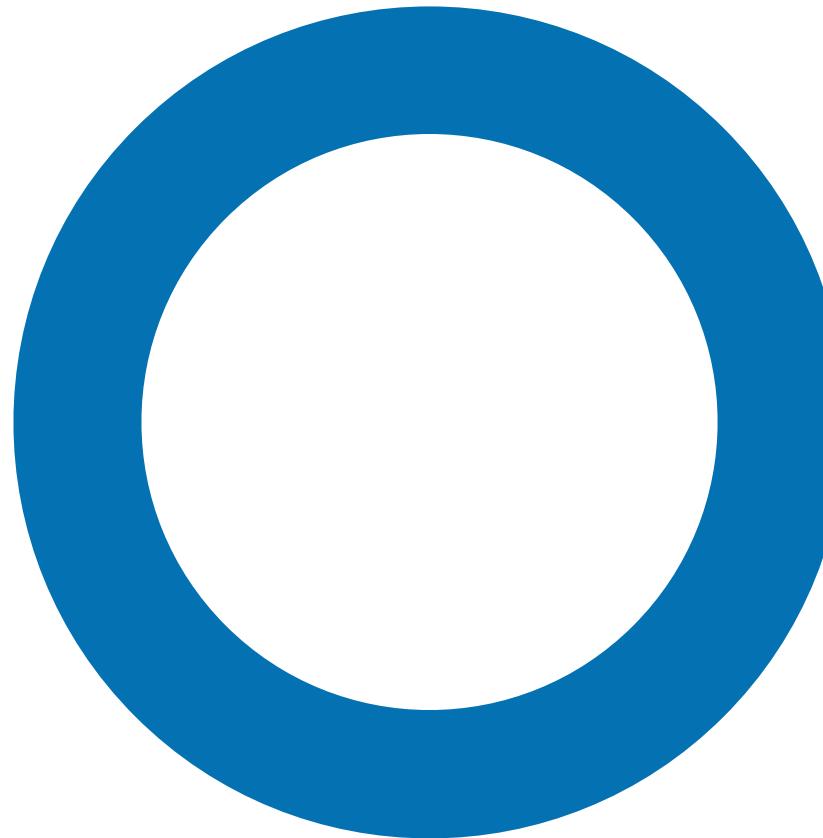
On the same machine, algorithm implementations with the same computational complexities will similar execution time when solving the same problem with the same input



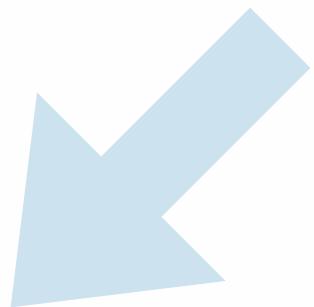
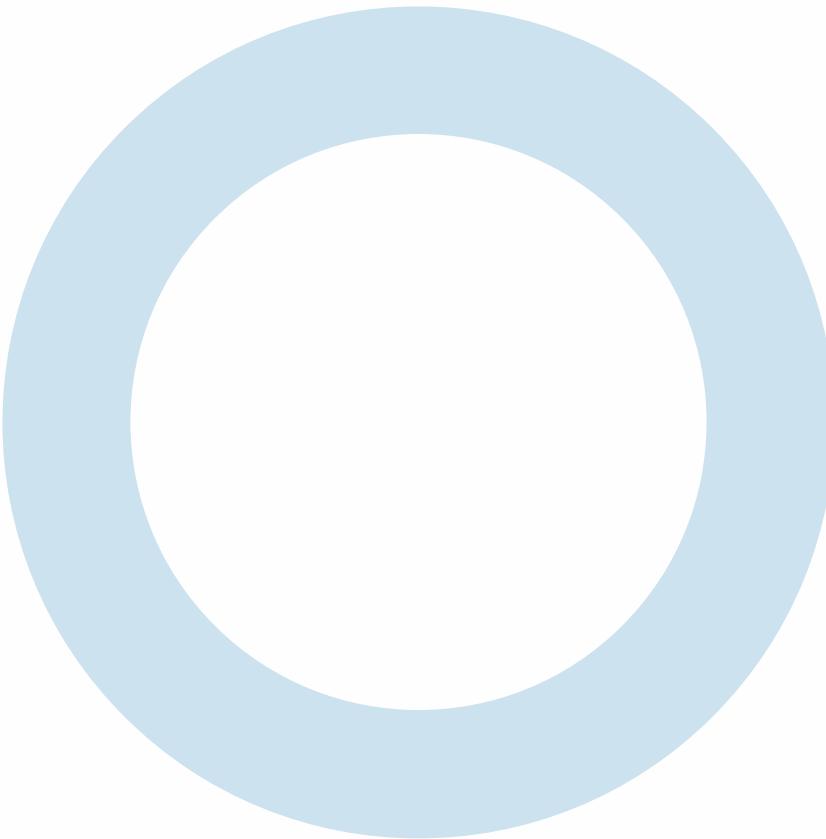
Leveraging more “bit-wise” operations in C code will make the program significantly faster



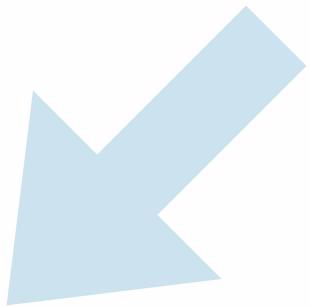
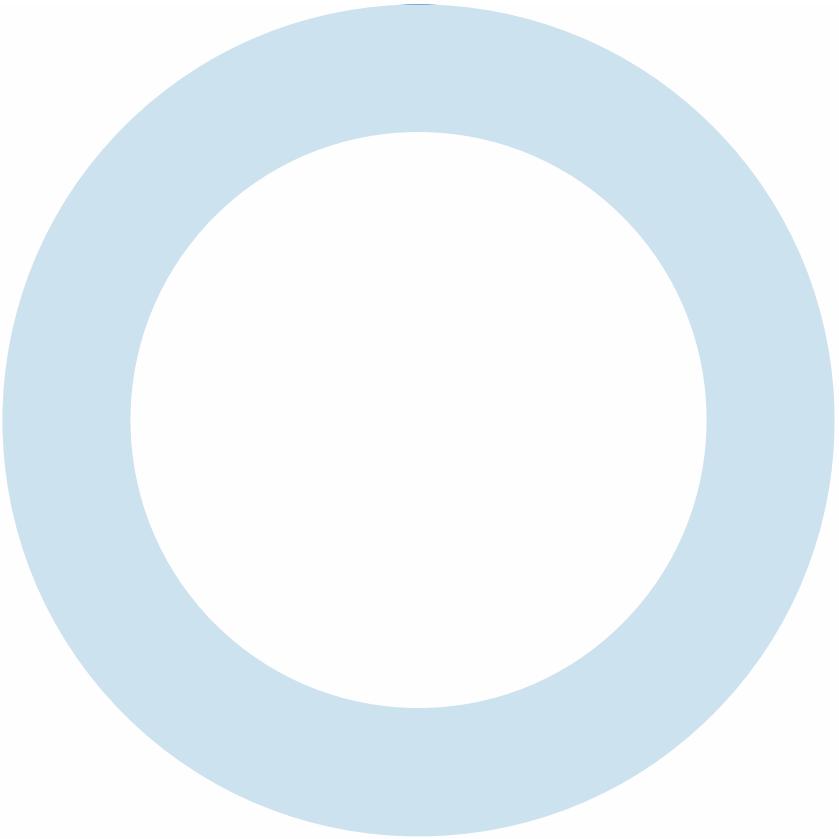
Algorithm complexity is less important if we have rich parallel processing capabilities



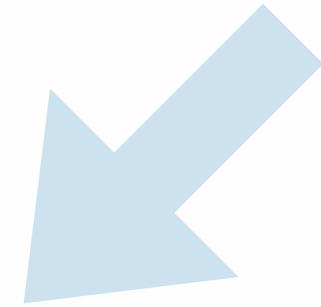
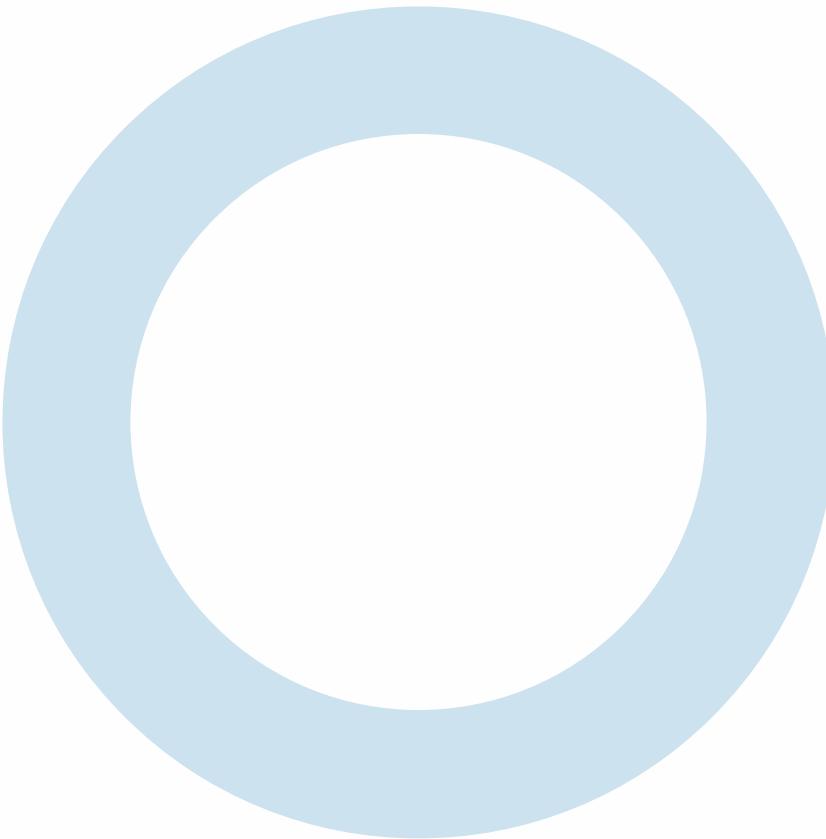
**The smaller size of a transistor,
the smaller power consumption of it**



The most recent GPUs can match the demand of efficient AI/ML training



GPUs are going to replace CPUs



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

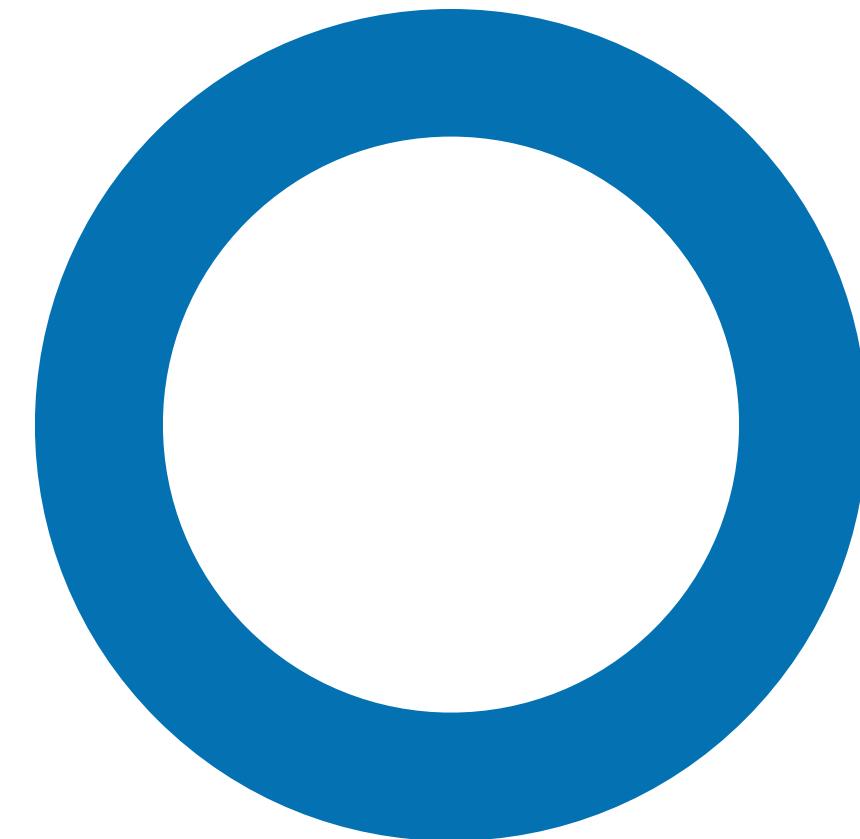
 $O(n \log_2 n)$

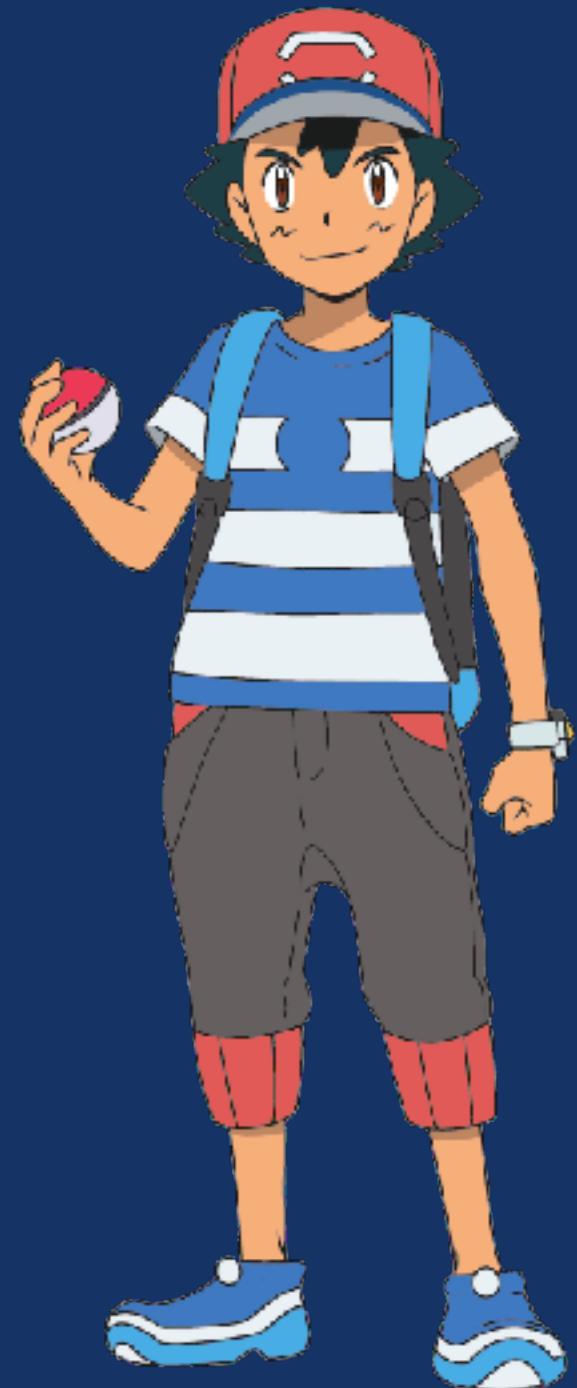
Bitonic Sort

 $O(n \log_2^2 n)$

```
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 CSE142?

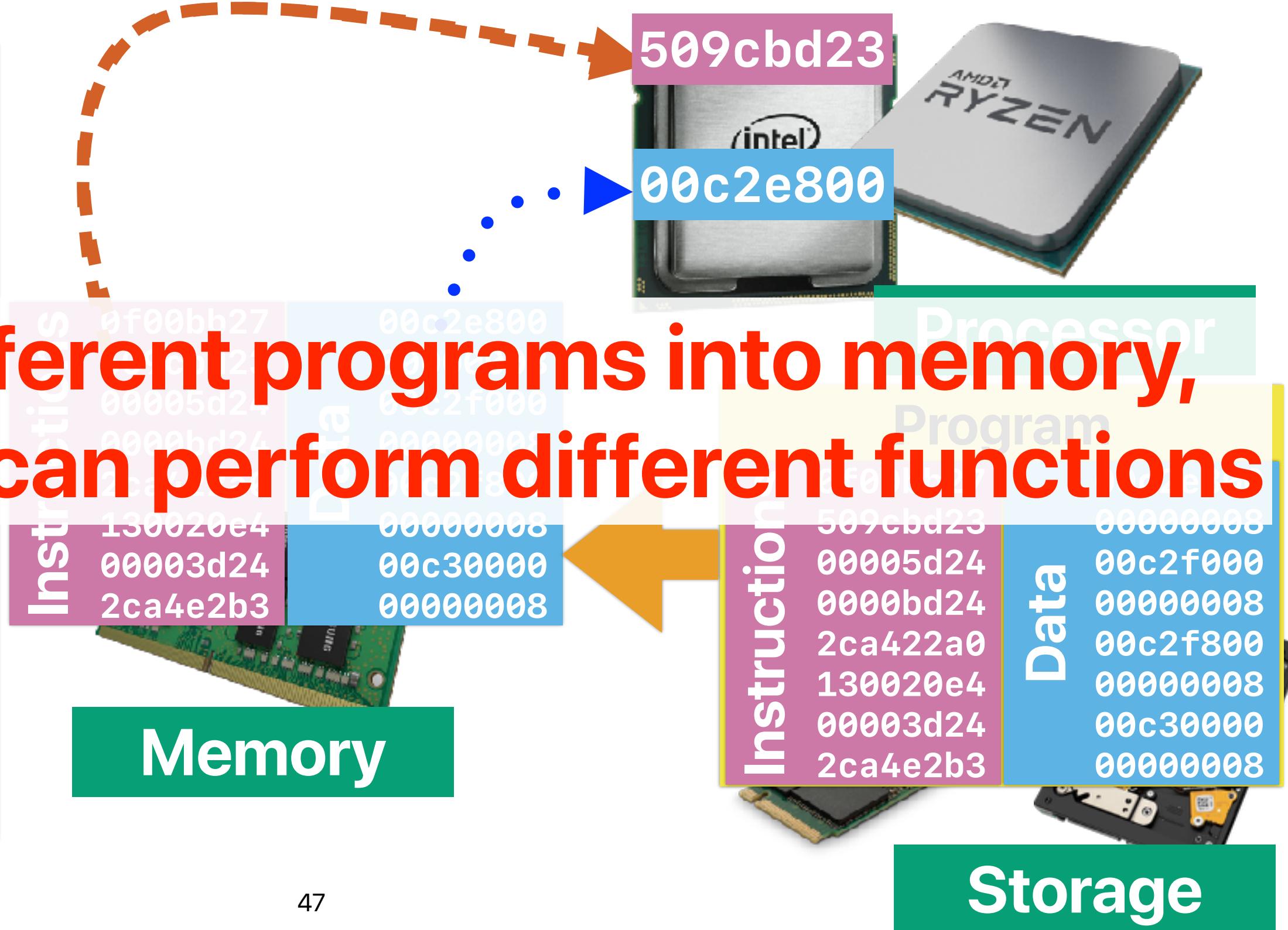
- Algorithm complexity does not work well on “real” computers

Big Picture: the Von Neumann Architecture

von Neumann Architecture



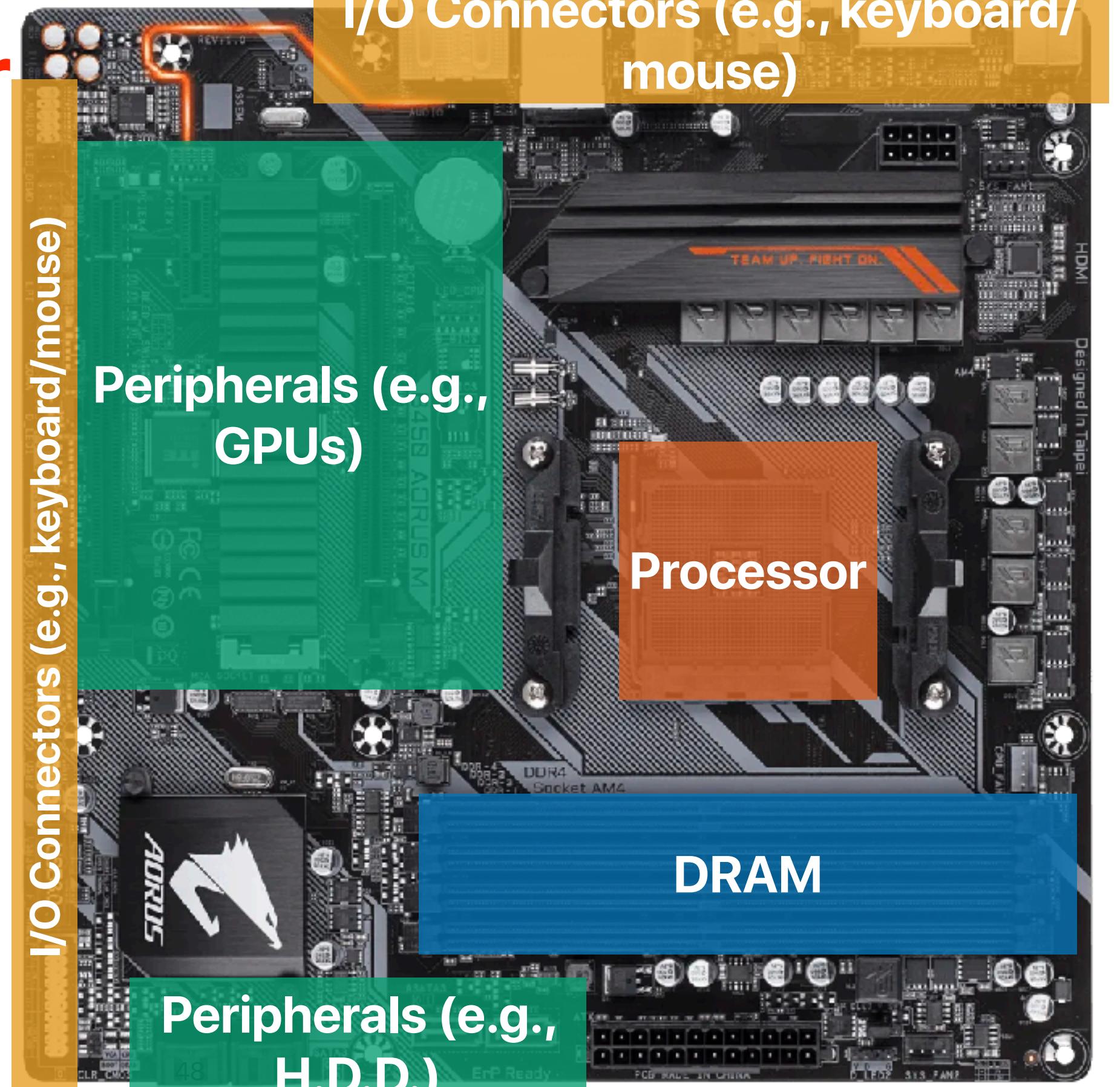
By loading different programs into memory,
your computer can perform different functions



Desktop Computer



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



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

Peripherals (e.g., GPUs)

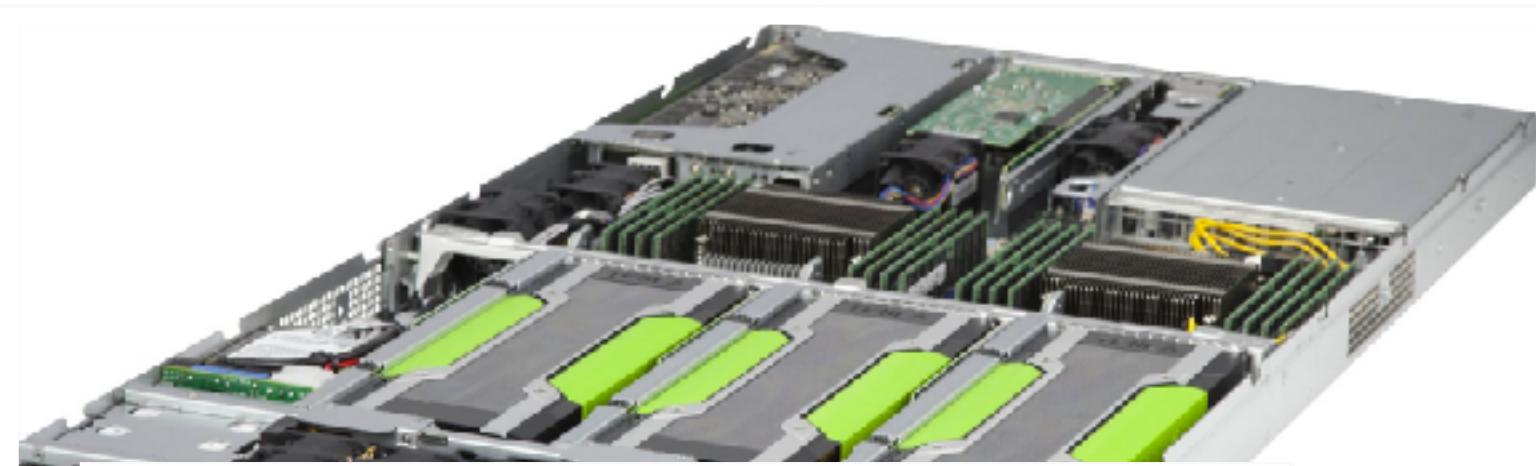
Processor

DRAM

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

Server

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



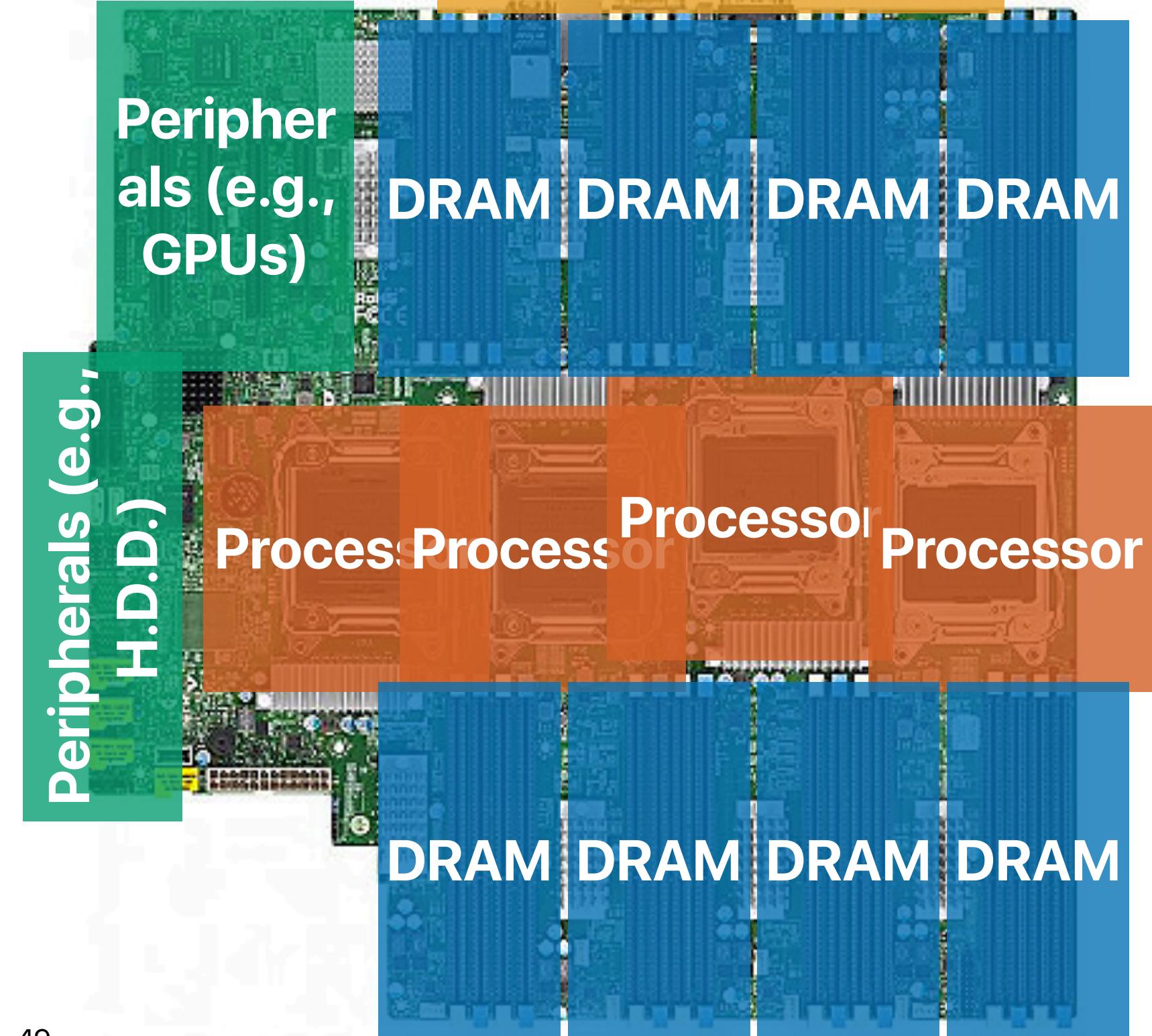
Peripherals (e.g., H.D.D.)

Peripherals (e.g., GPUs)

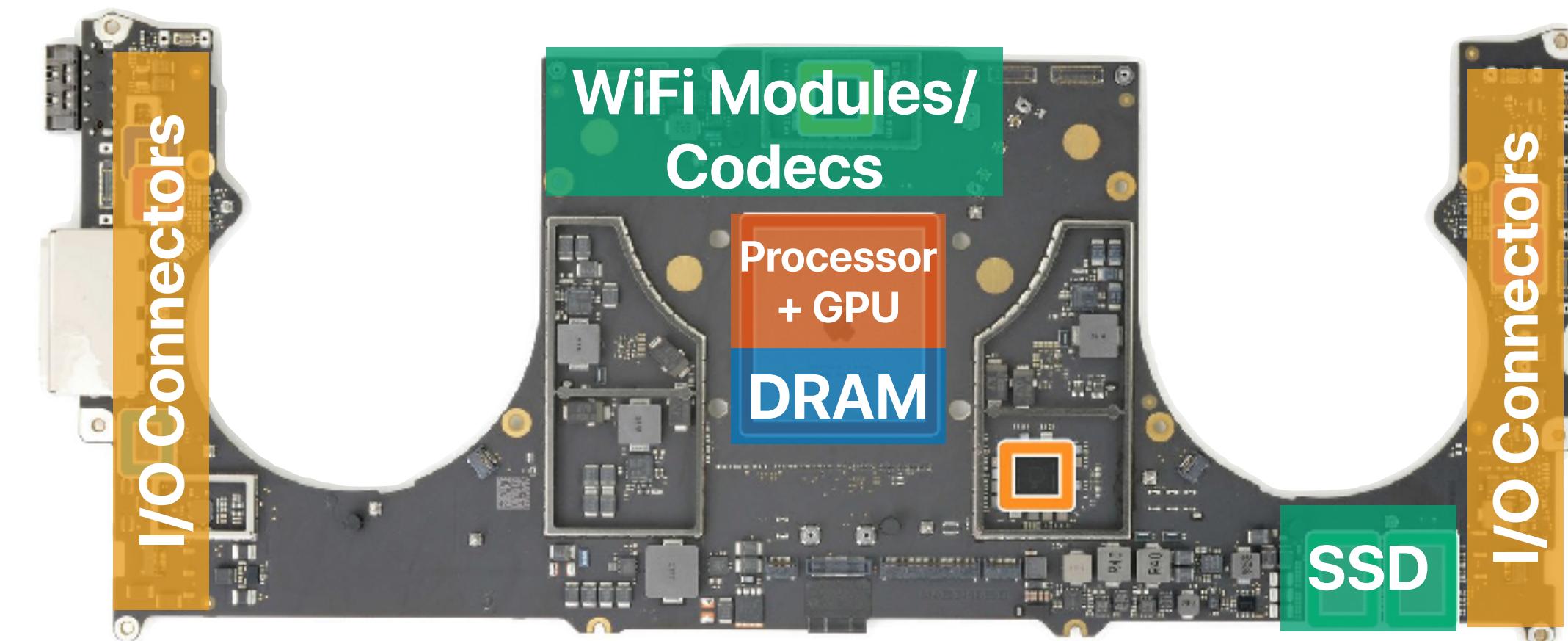
DRAM DRAM DRAM DRAM

Processor Processor Processor Processor

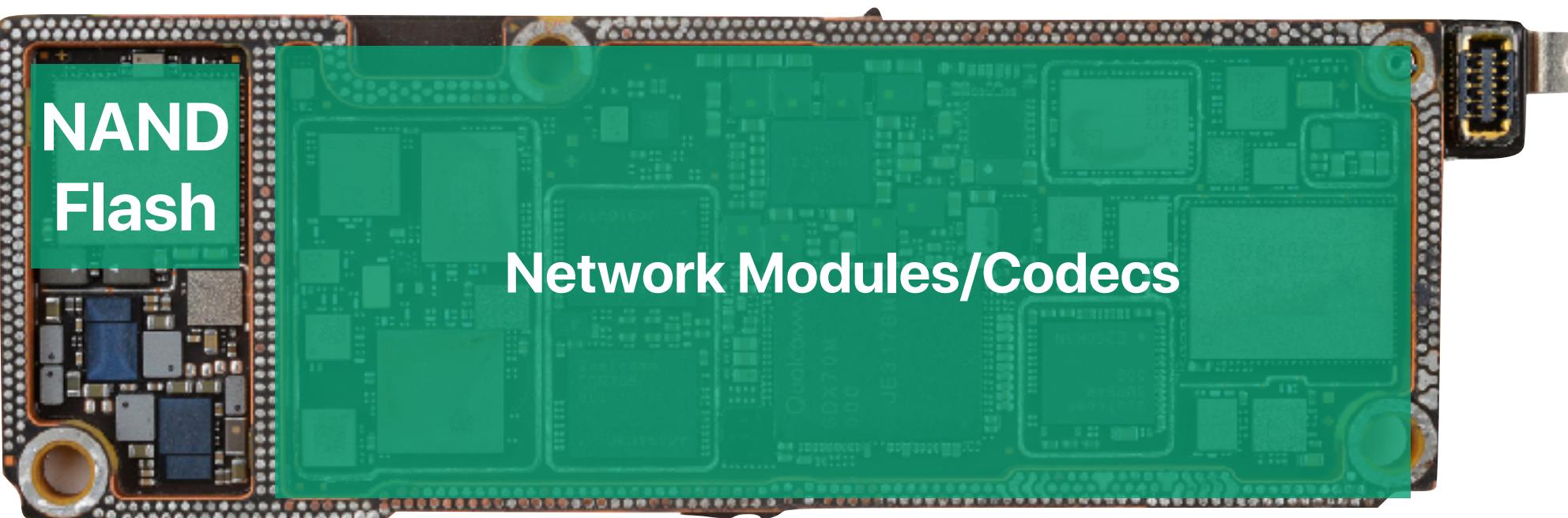
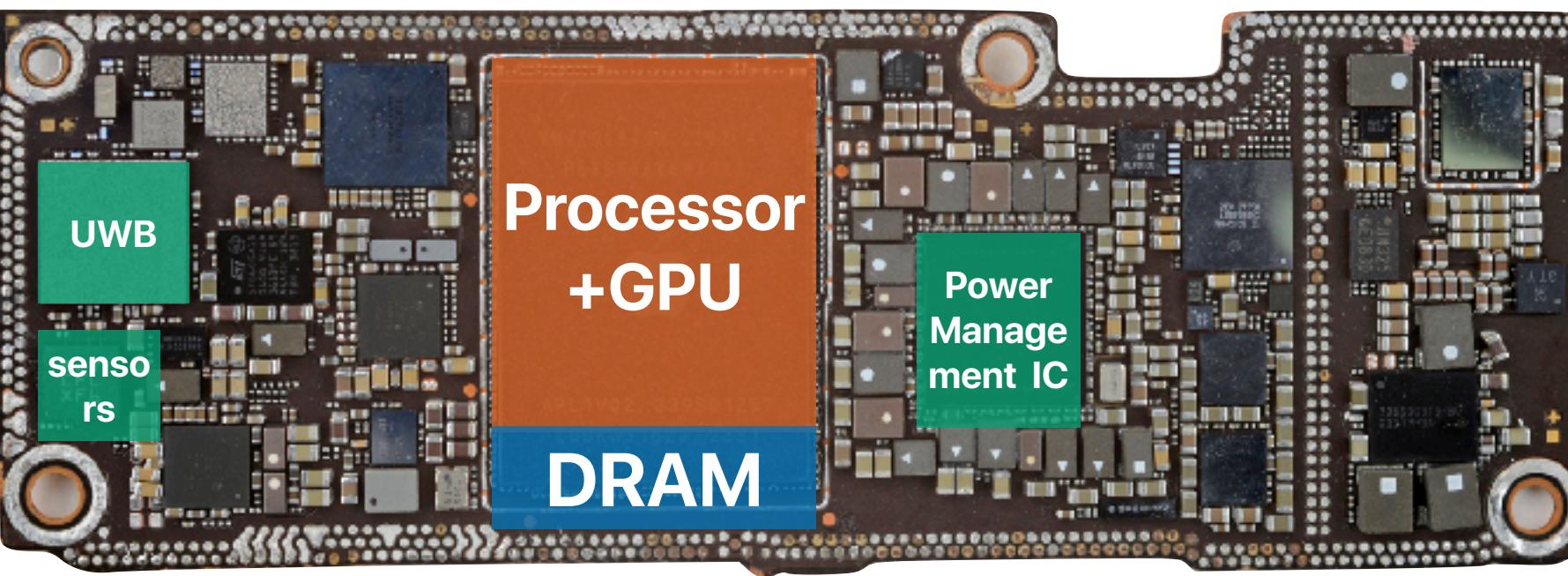
DRAM DRAM DRAM DRAM



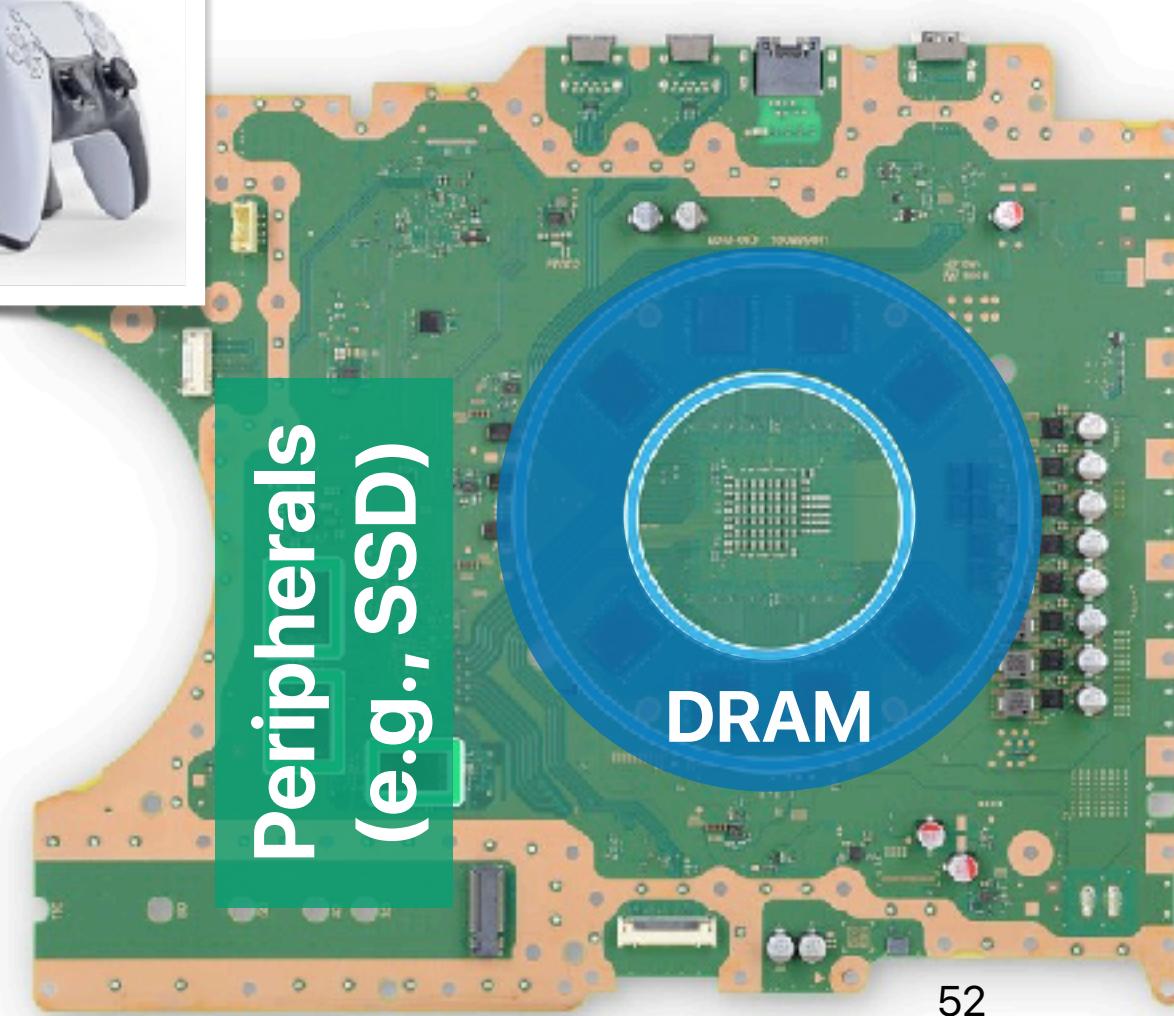
MacBook Pro M3



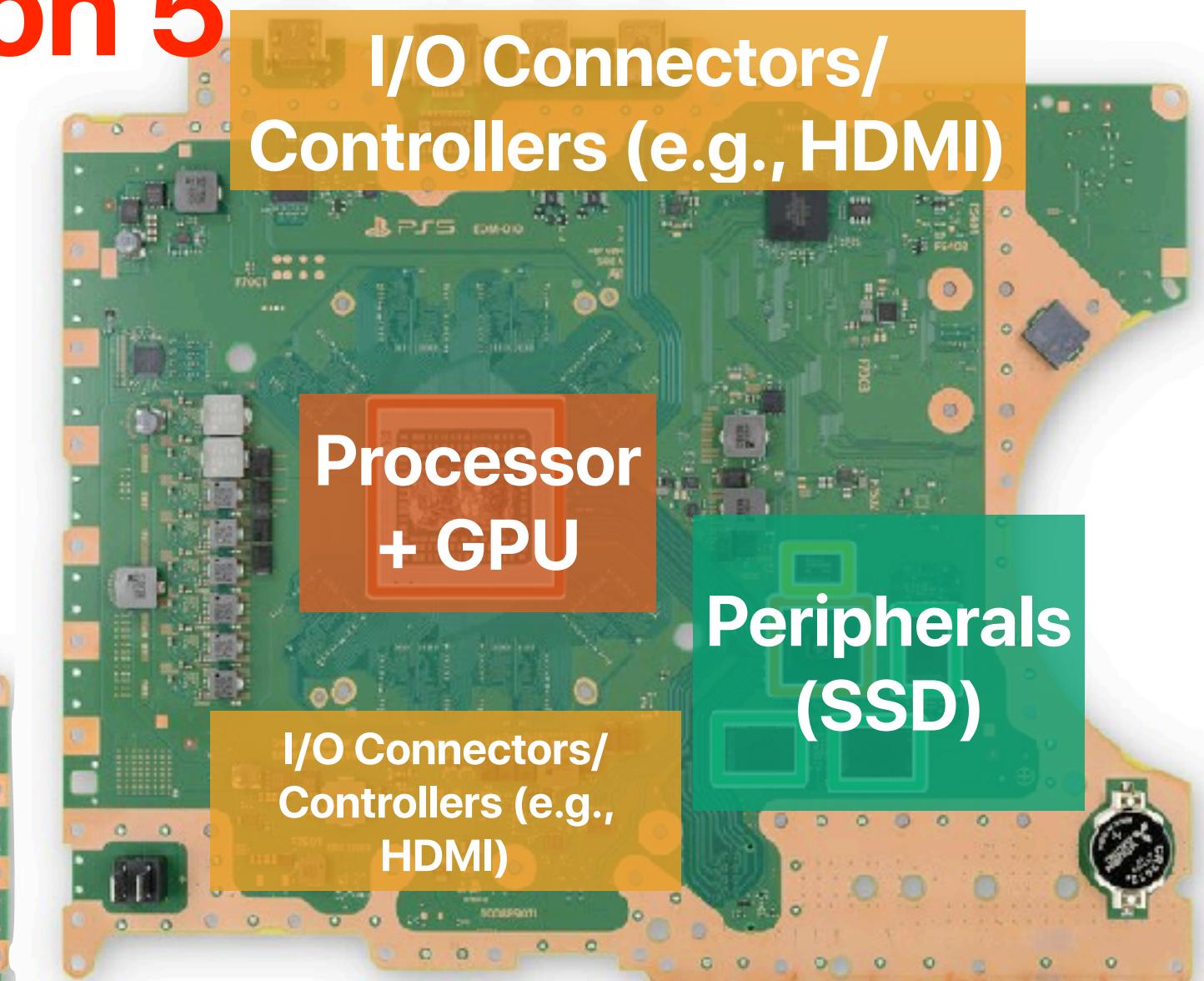
iPhone 15 Pro



Play Station 5



Peripherals
(e.g., SSD)



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

Processor
+ GPU

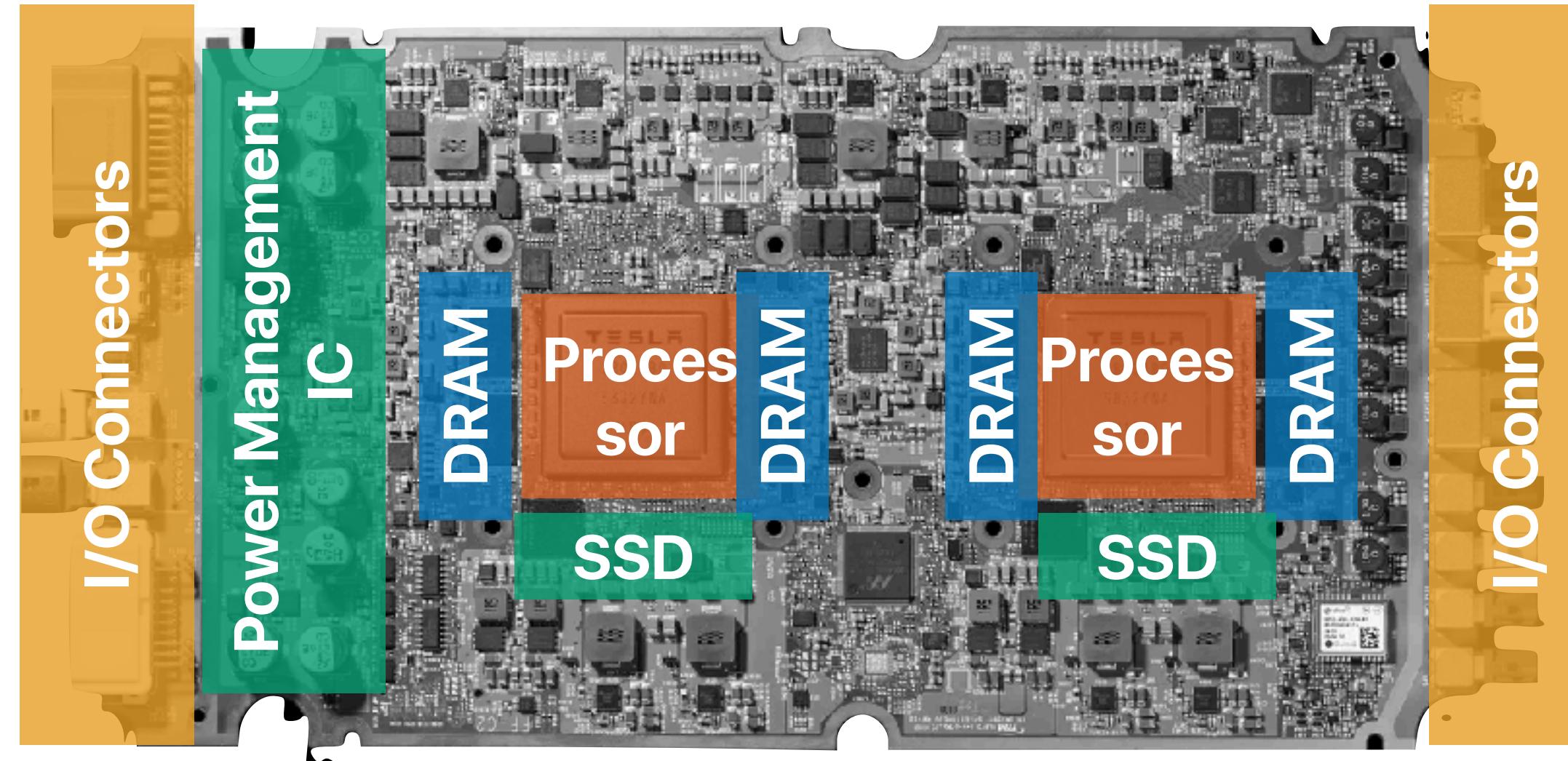
Peripherals
(SSD)

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

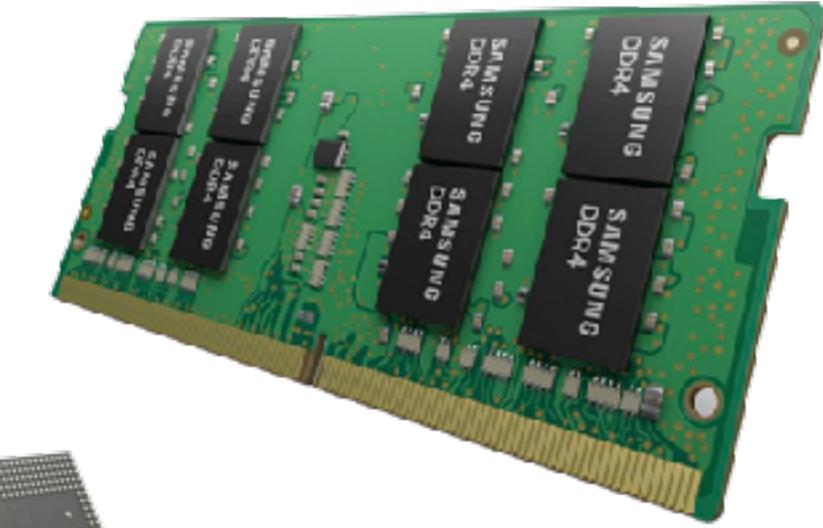
Nintendo Switch



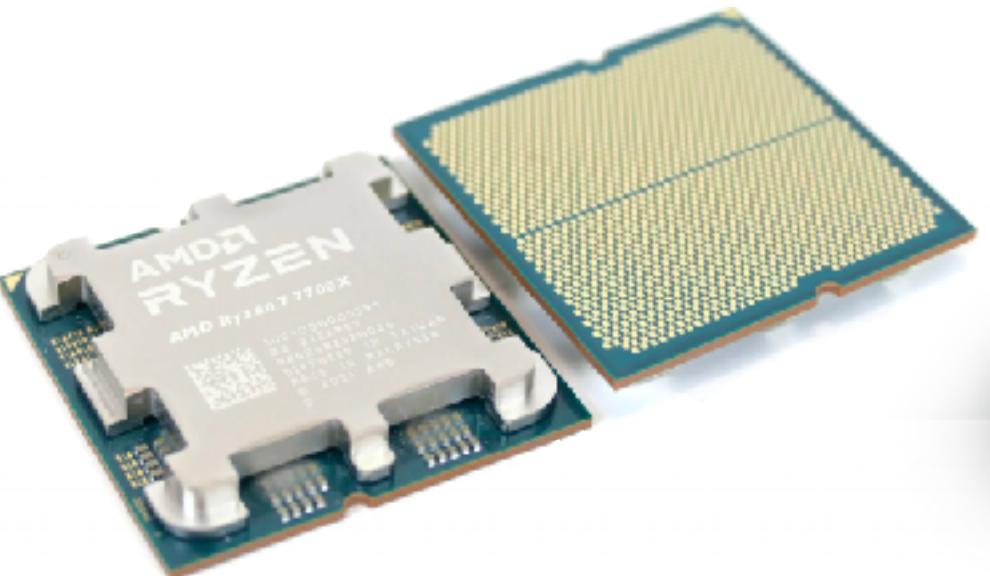
Tesla Model 3



Processors and memory modules are everywhere!



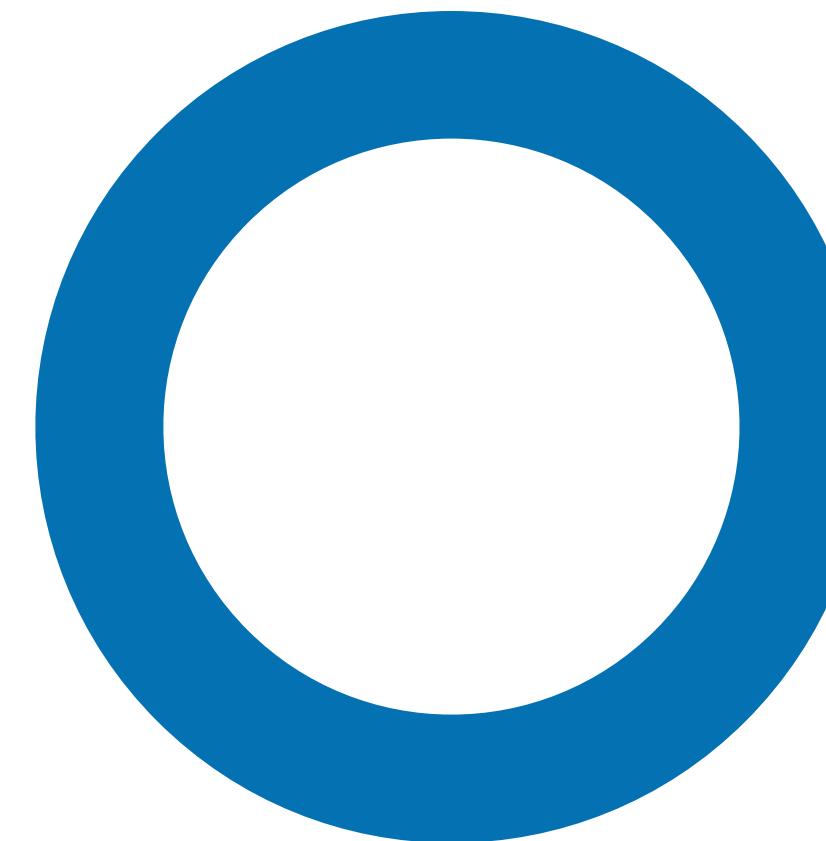
Processors



Memory



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

main:
.LFB0:
endbr64
pushq %rbp
movq %rsp, %rbp
movl \$0, -8(%rbp)
movl \$0, -4(%rbp)
movl \$0, -8(%rbp)
jmp .L2

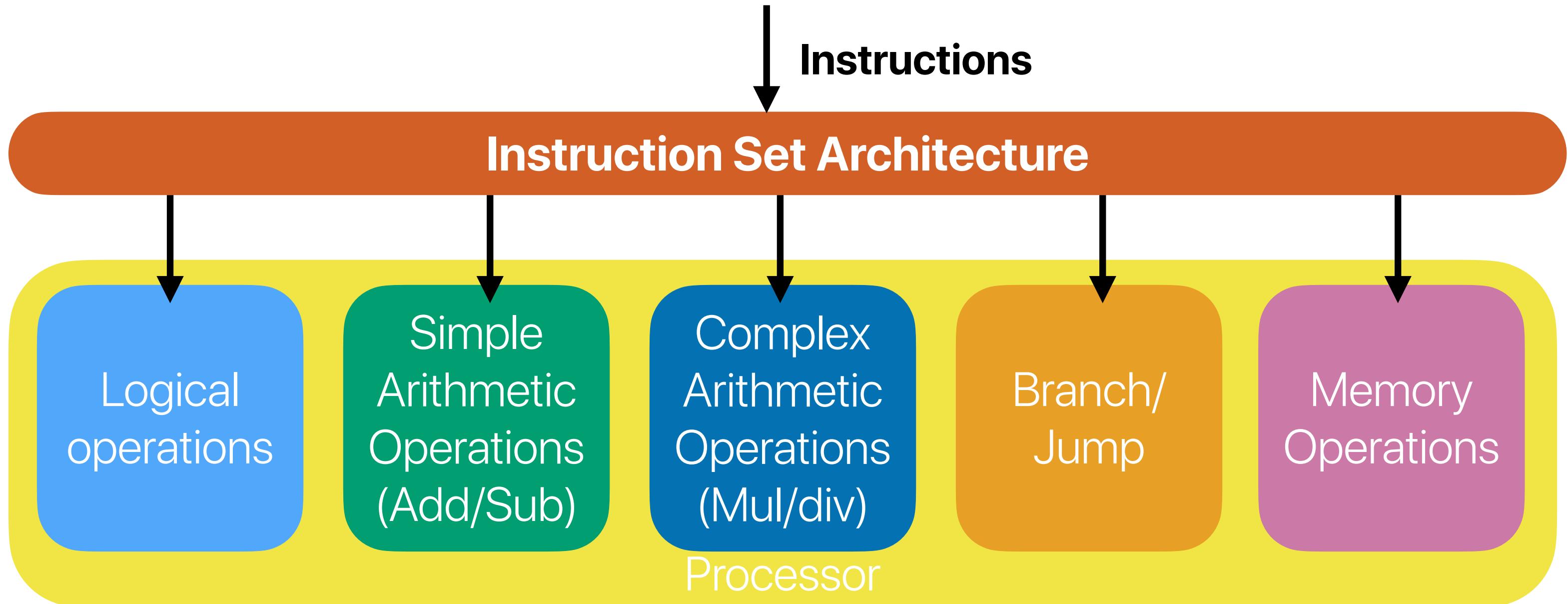
.L3:
movl -8(%rbp), %eax
cltq
leaq 0(%rax,4), %rdx
leaq A(%rip), %rax

Compiler

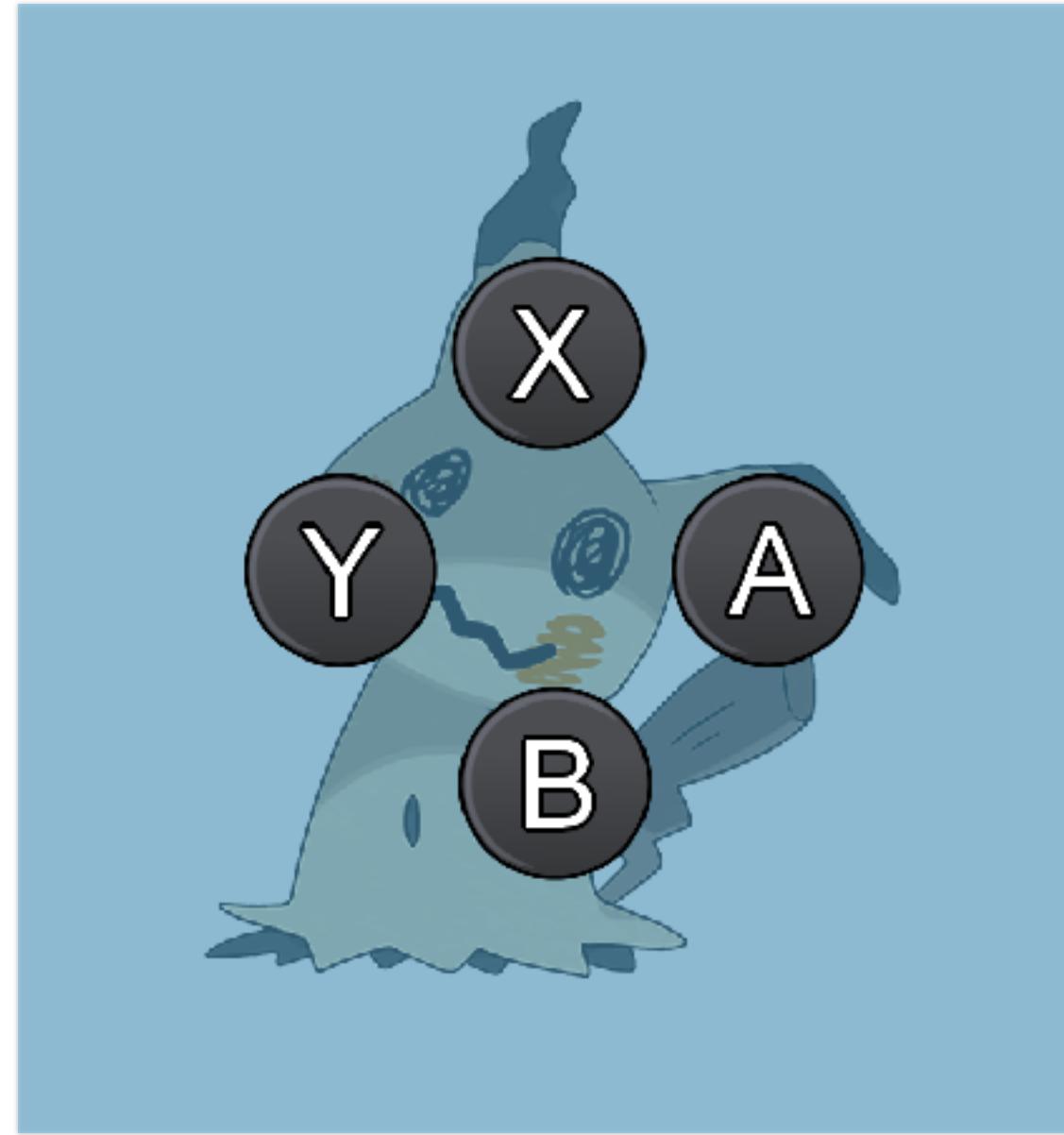
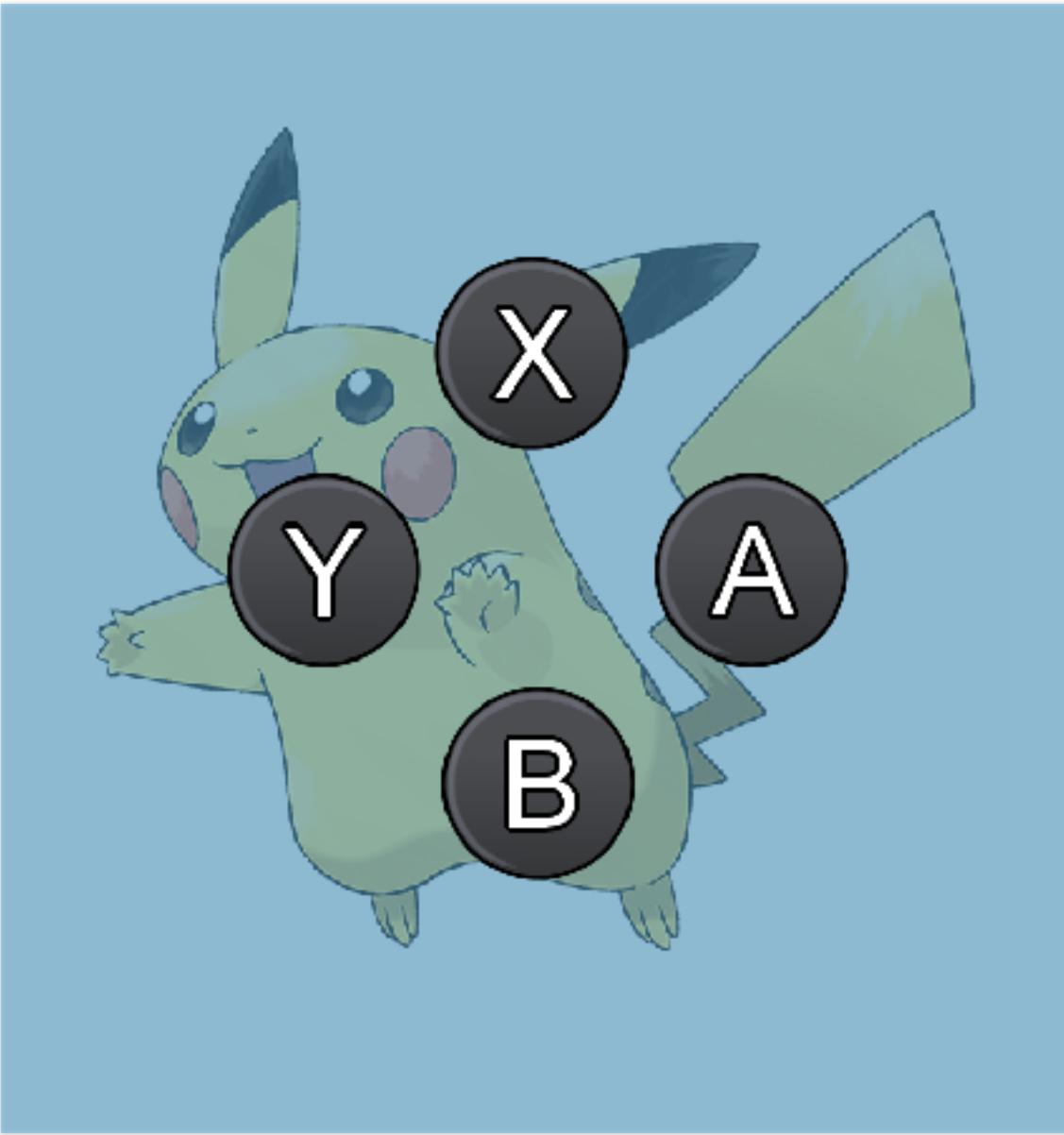
movl (%rdx,%rax),
%eax
addl %eax, -4(%rbp)
addl \$1, -8(%rbp)
.L2:
cmpl \$19, -8(%rbp)
jle .L3
movl \$0, %eax
popq %rbp
ret

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⁽¹⁾

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

Moore's Law⁽¹⁾

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

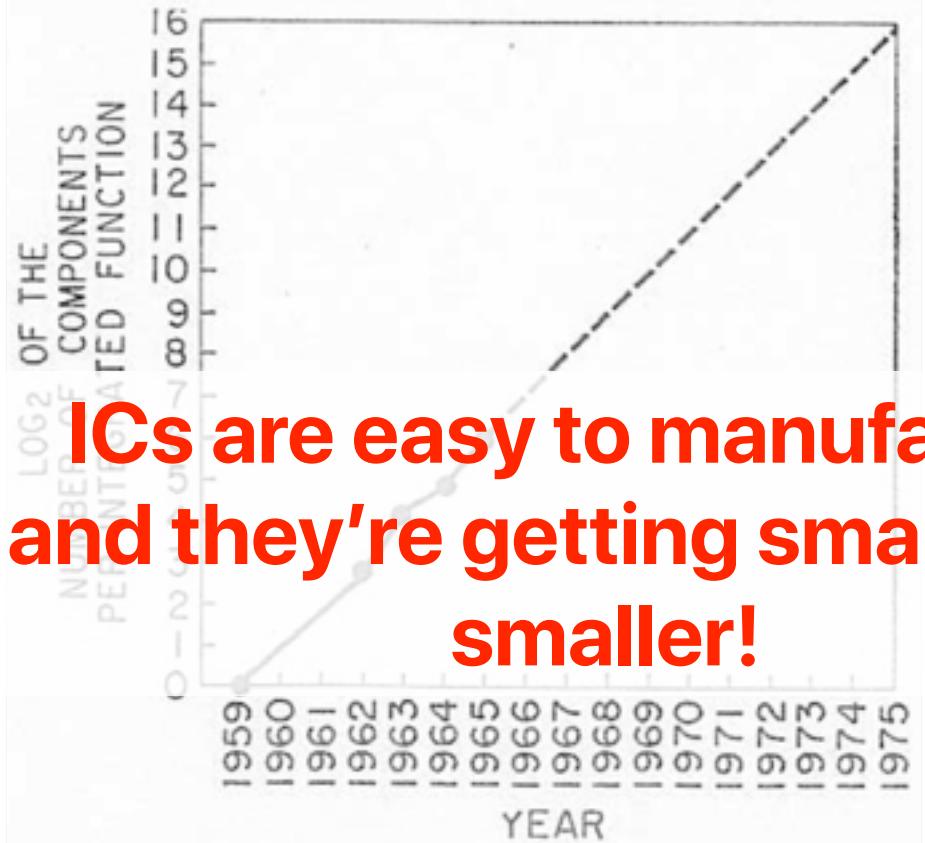
ICs are small

(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

systems. However, the lack of discrete compo-

nents—such as resistors, capacitors, and in-

ductors—makes it easier to implement linear

systems. The lack of discrete components makes

it easier to implement linear systems. The lack

of discrete components makes it easier to imple-

Reliability count

In almost every field of electronics, ICs have demonstrated higher reliability than discrete components. At the same level of production—low compared to that of discrete components—it offers reduced systems cost, and in many systems improved performance has been realized.

ICs are more reliable

Heat problem

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

**Establish
Moore's Law
important
historic**

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 de-

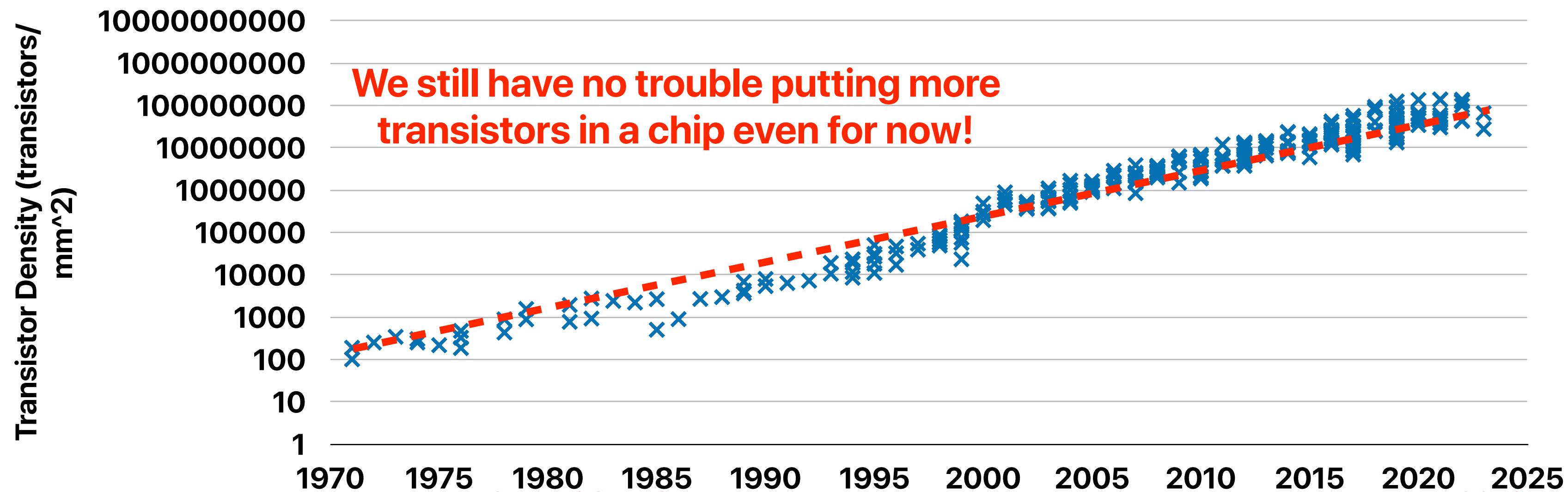
sign automation procedures could translate from any special engineering.

Designing ICs can be easy

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

Moore's Law⁽¹⁾

- 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

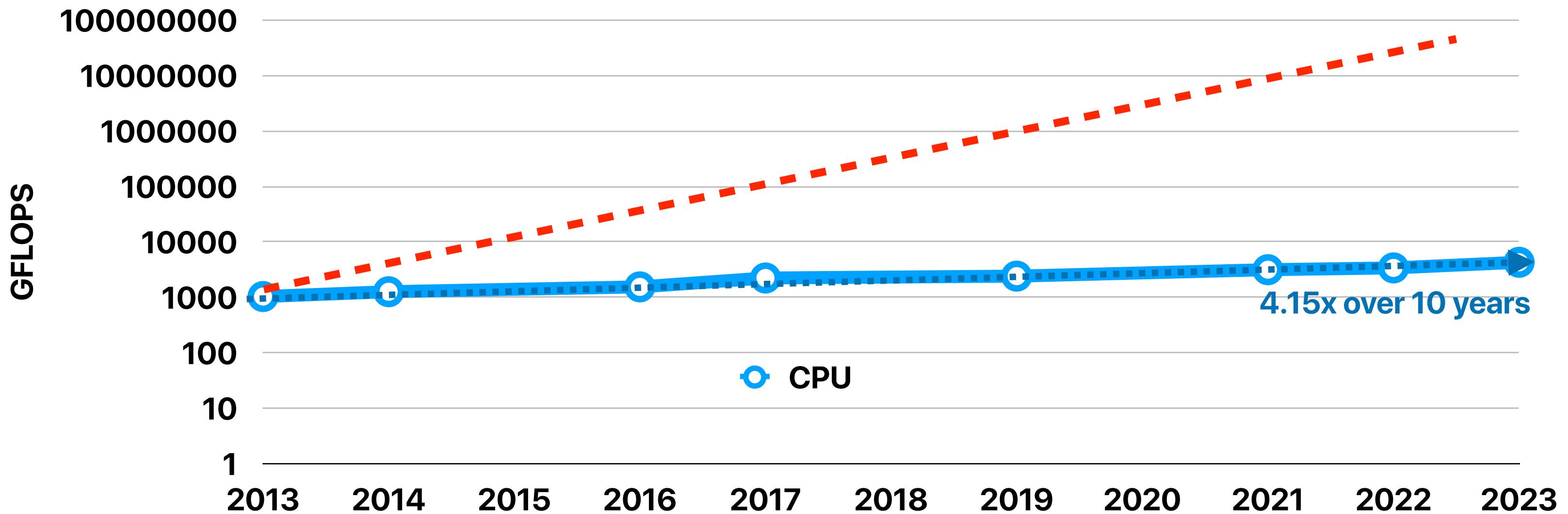


(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8).

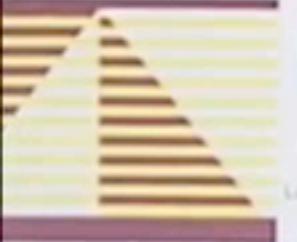
Moore's Law is currently discontinued



CPU Performance v.s. Moore's Law

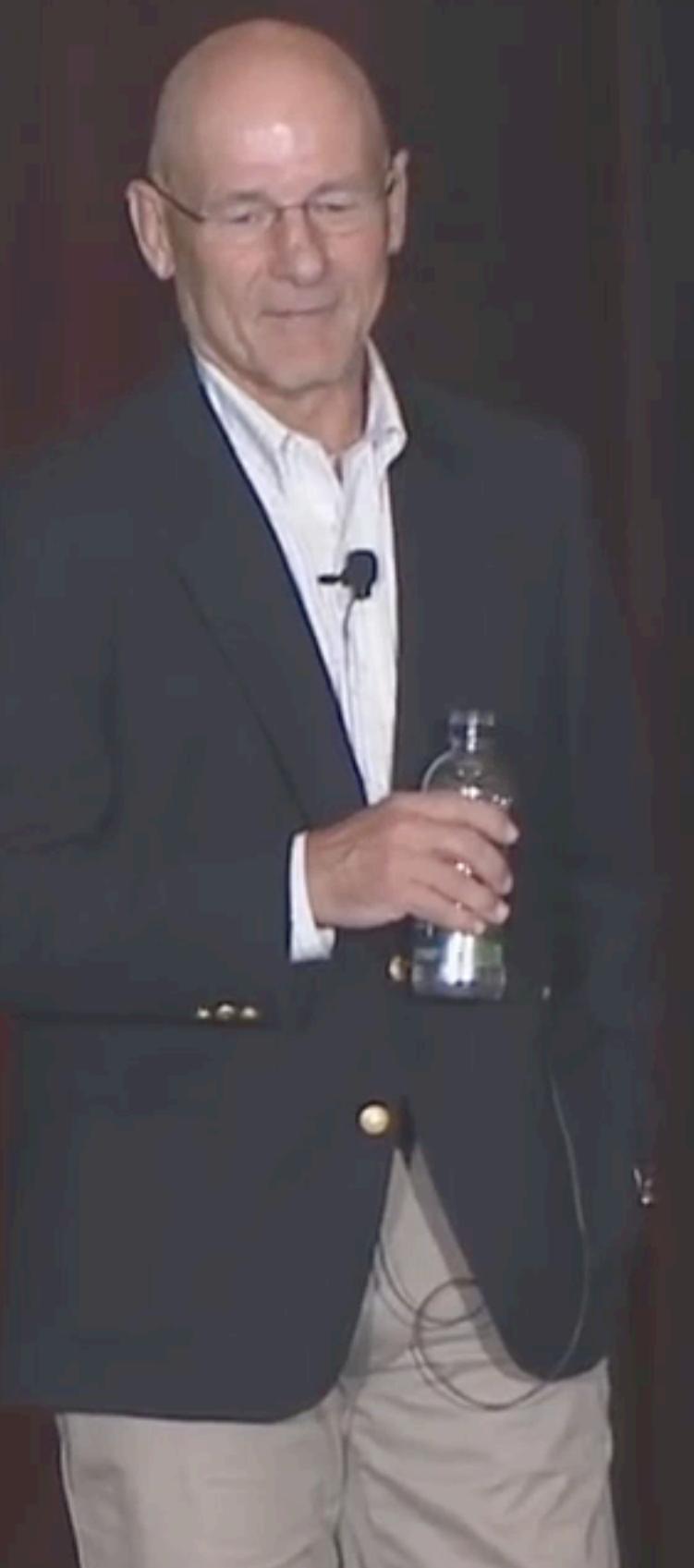
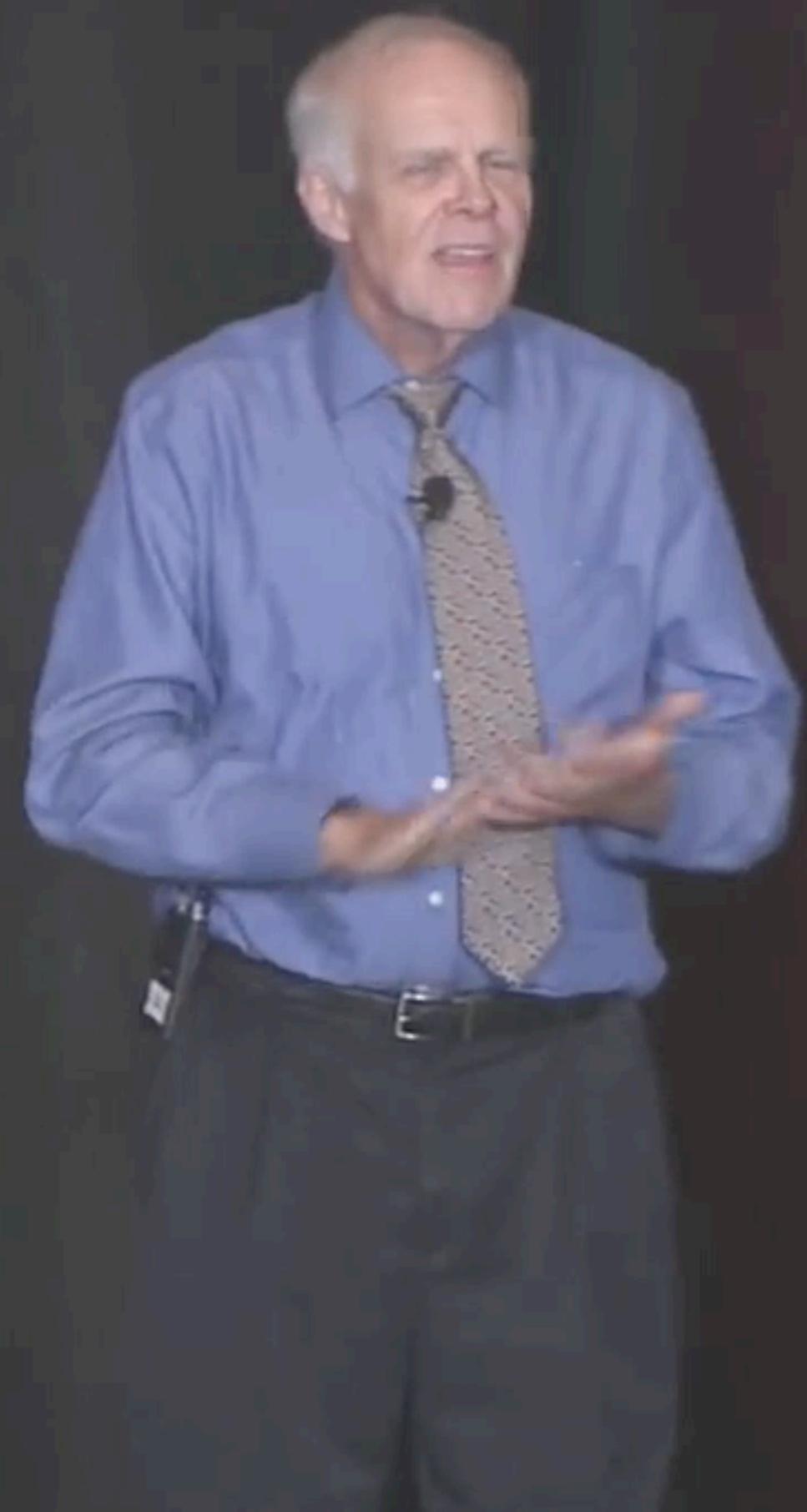


<https://ourworldindata.org/grapher/artificial-intelligence-training-computation>

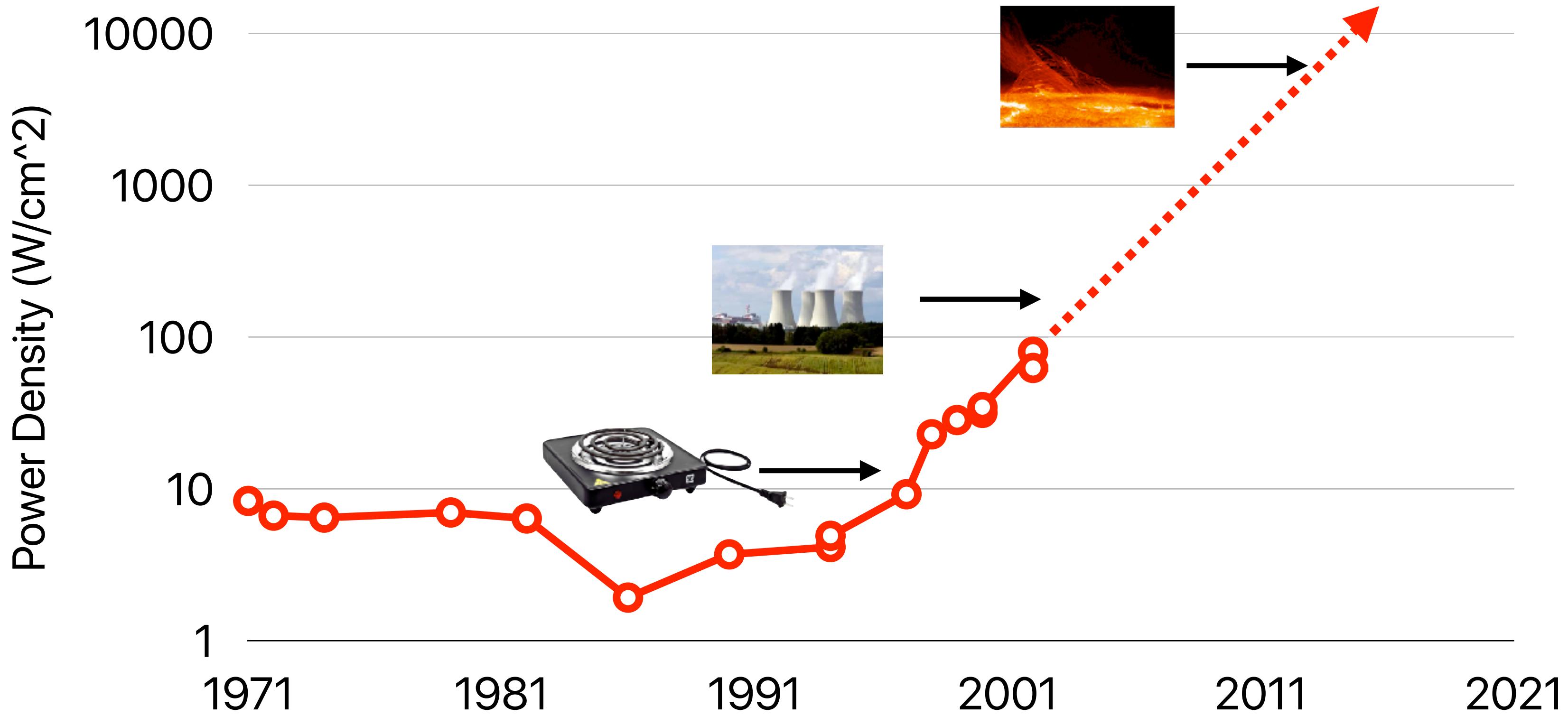


The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA

ISCA 2018
uring Lecture



Power Density of Processors



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.

5.77 W/1B transistors

TSMC 4nm

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.

5.73 W/1B transistors

Intel 7nm

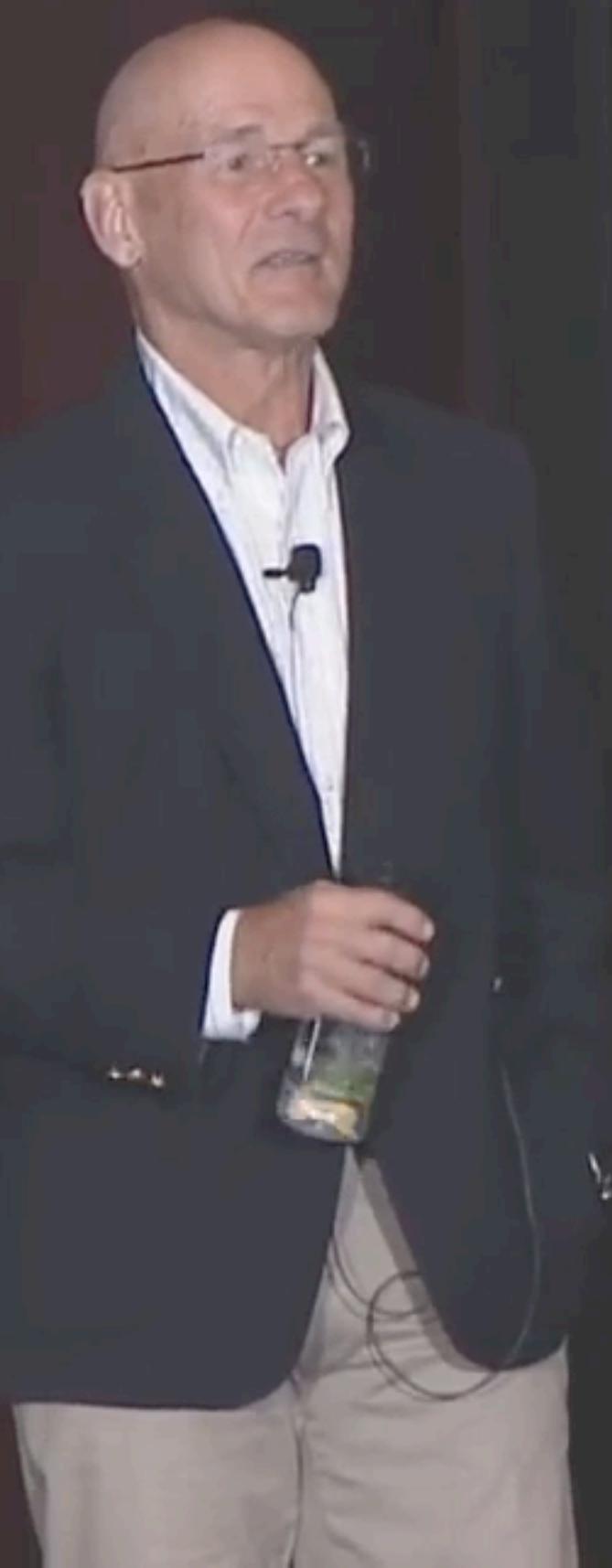
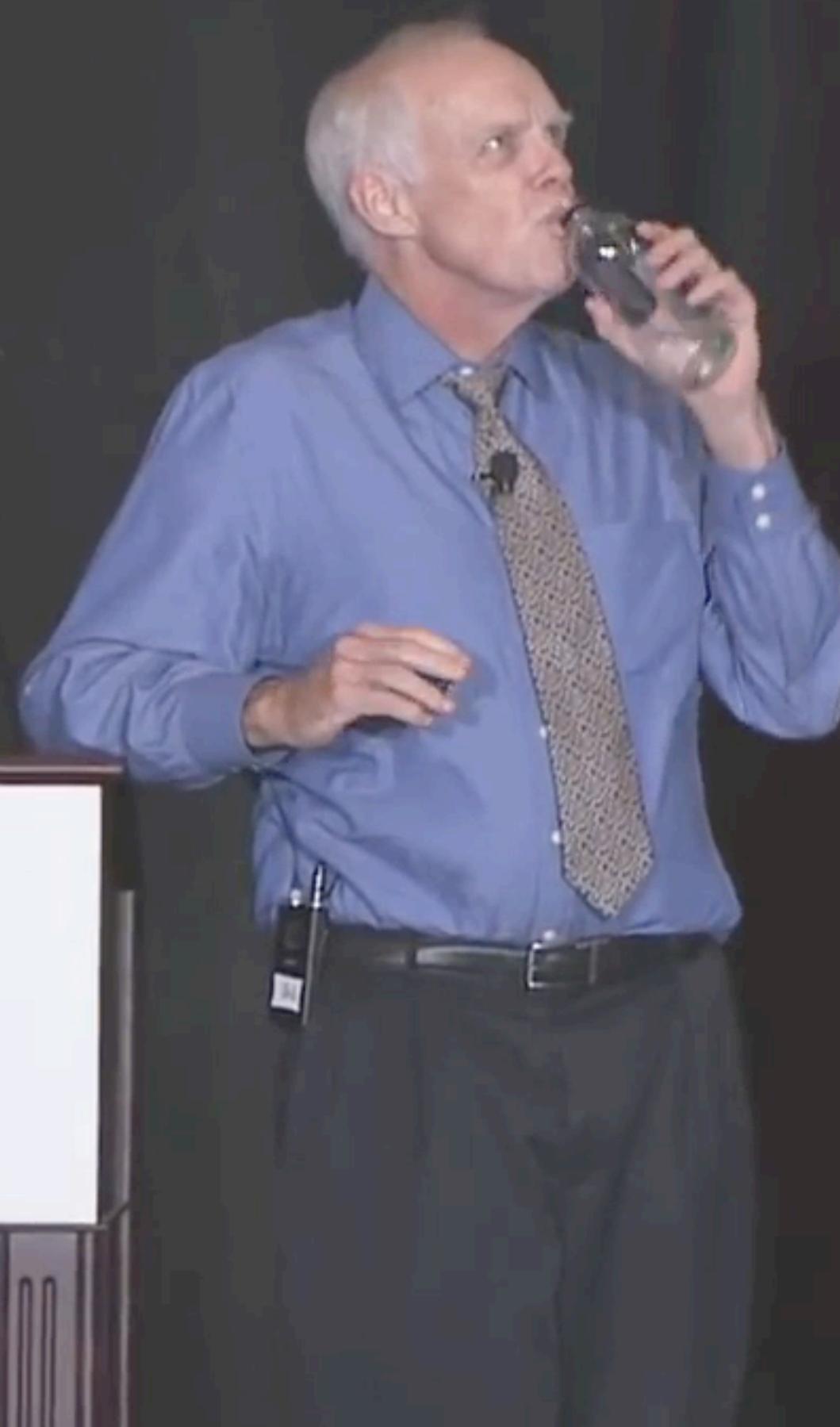
**The smaller size of a transistor,
the smaller power consumption of it**





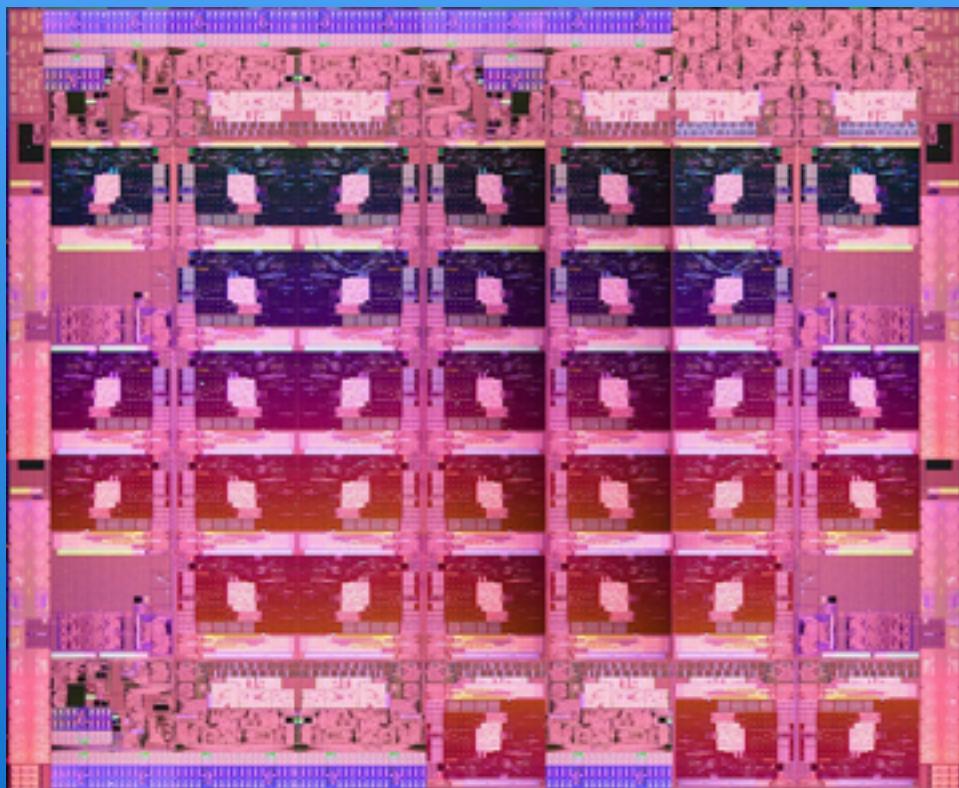
The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA

A 2018
g Lecture



Alternatives to scaling single cores

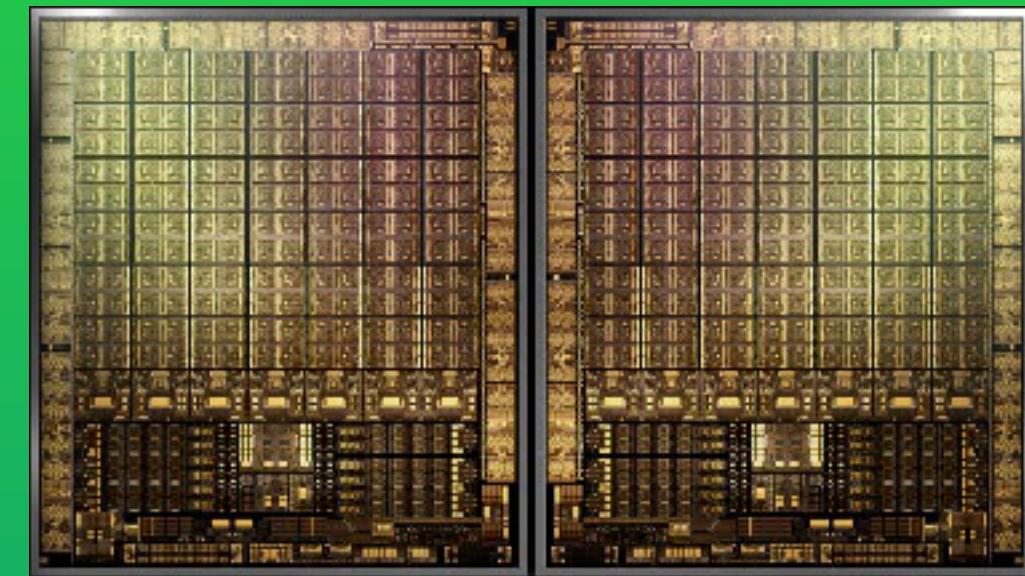
Multicore Processors



34-core Intel
Sapphire Rapids

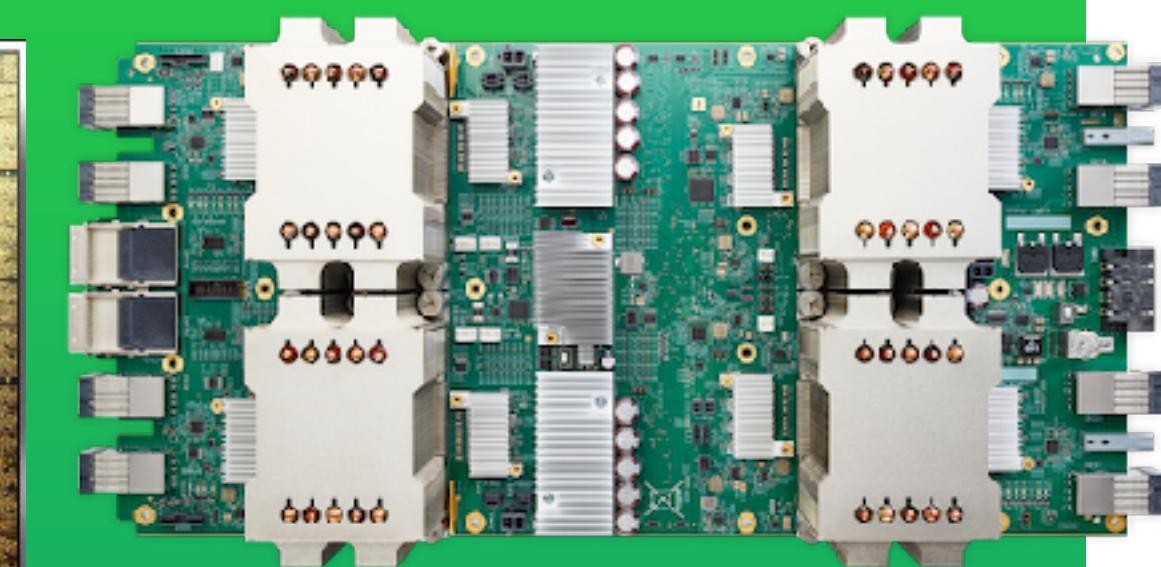
Thread-level parallelism

Hardware Accelerators



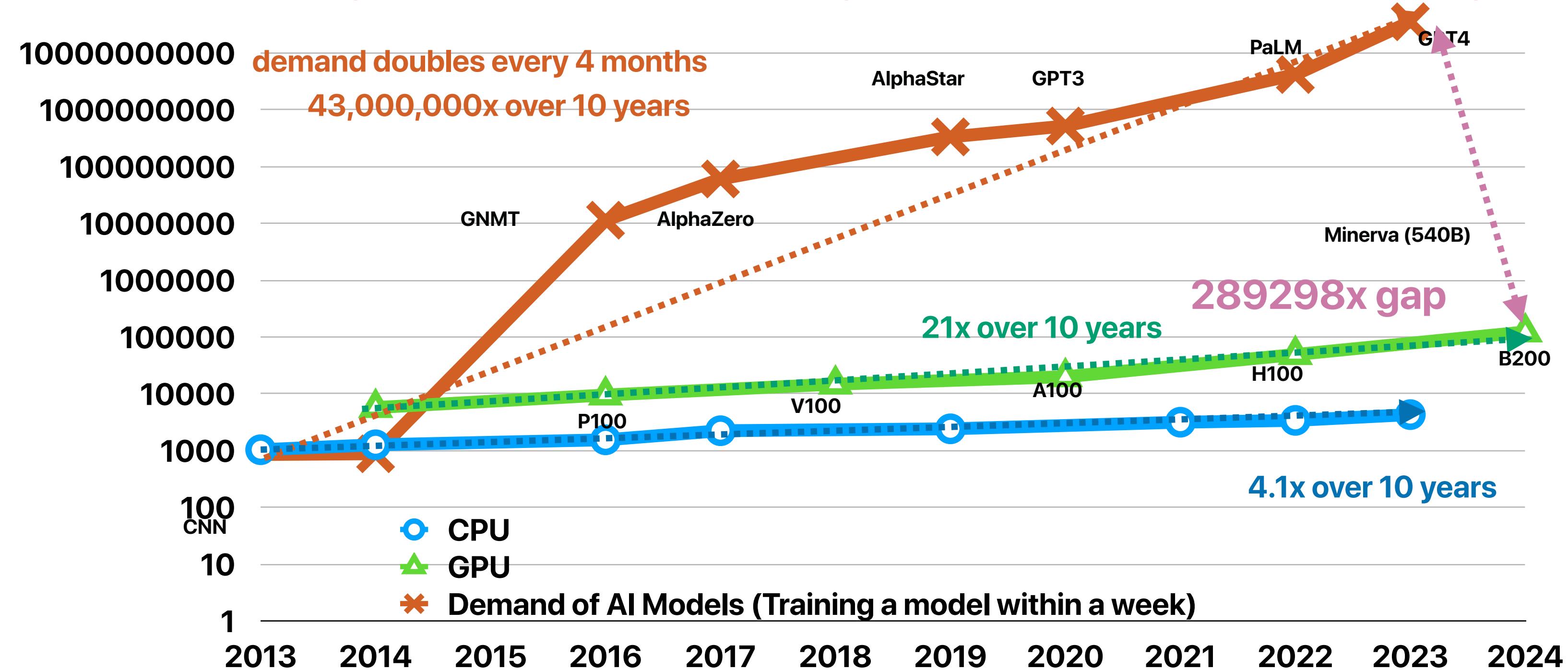
NVIDIA H100
Graphic Processing Units

Data-level parallelism



Google
Tensor Processing Units

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



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

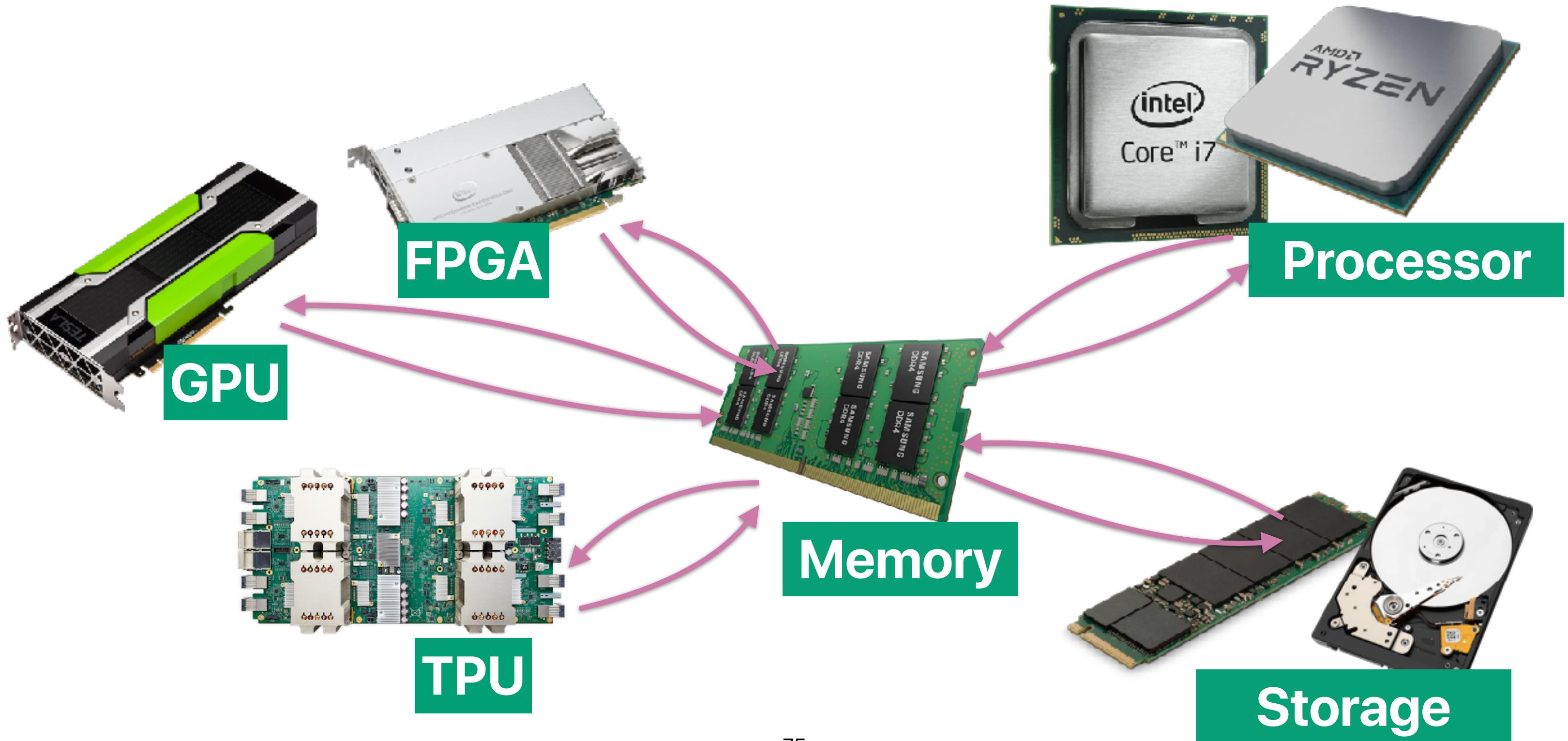
The most recent GPUs can match the demand of efficient AI/ML training



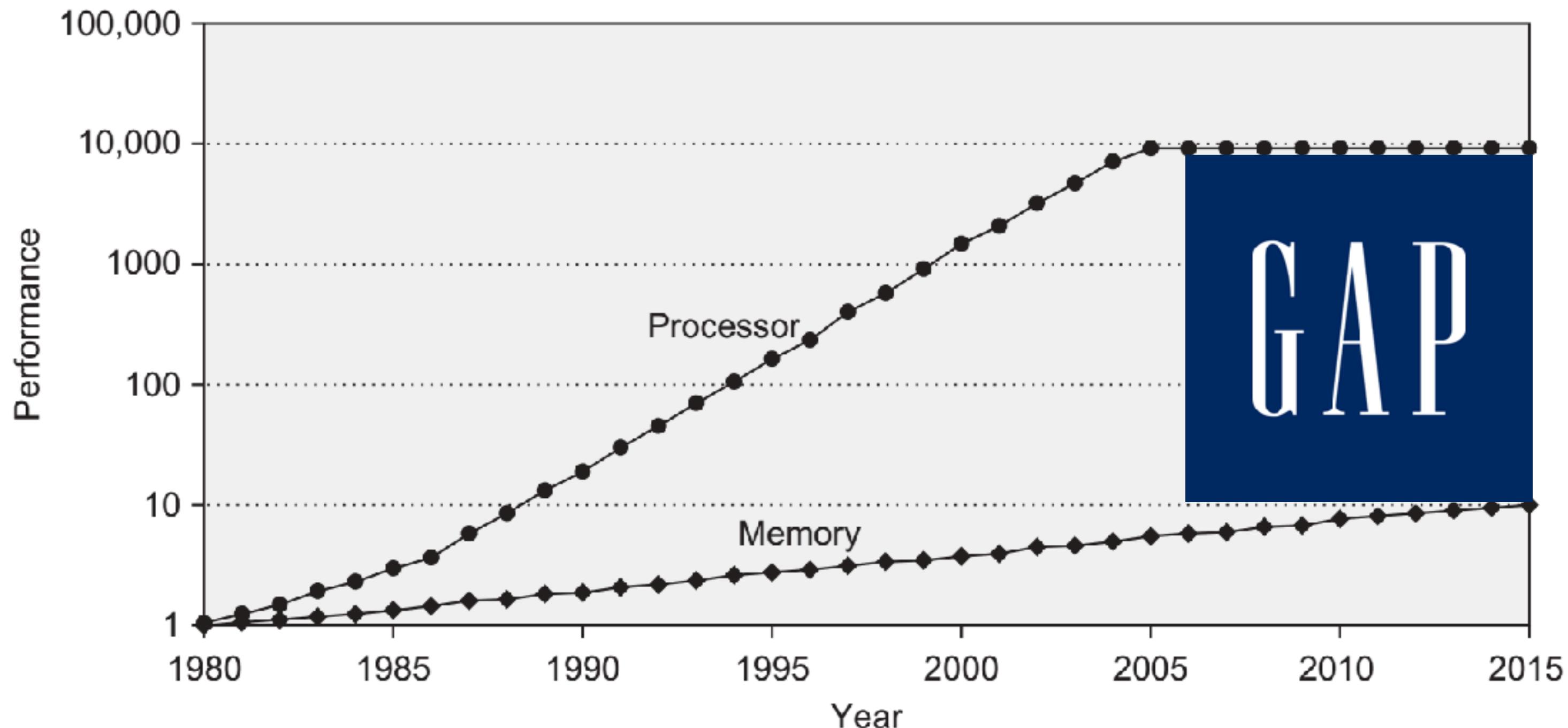
Take-aways: Why CSE142?

- 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!

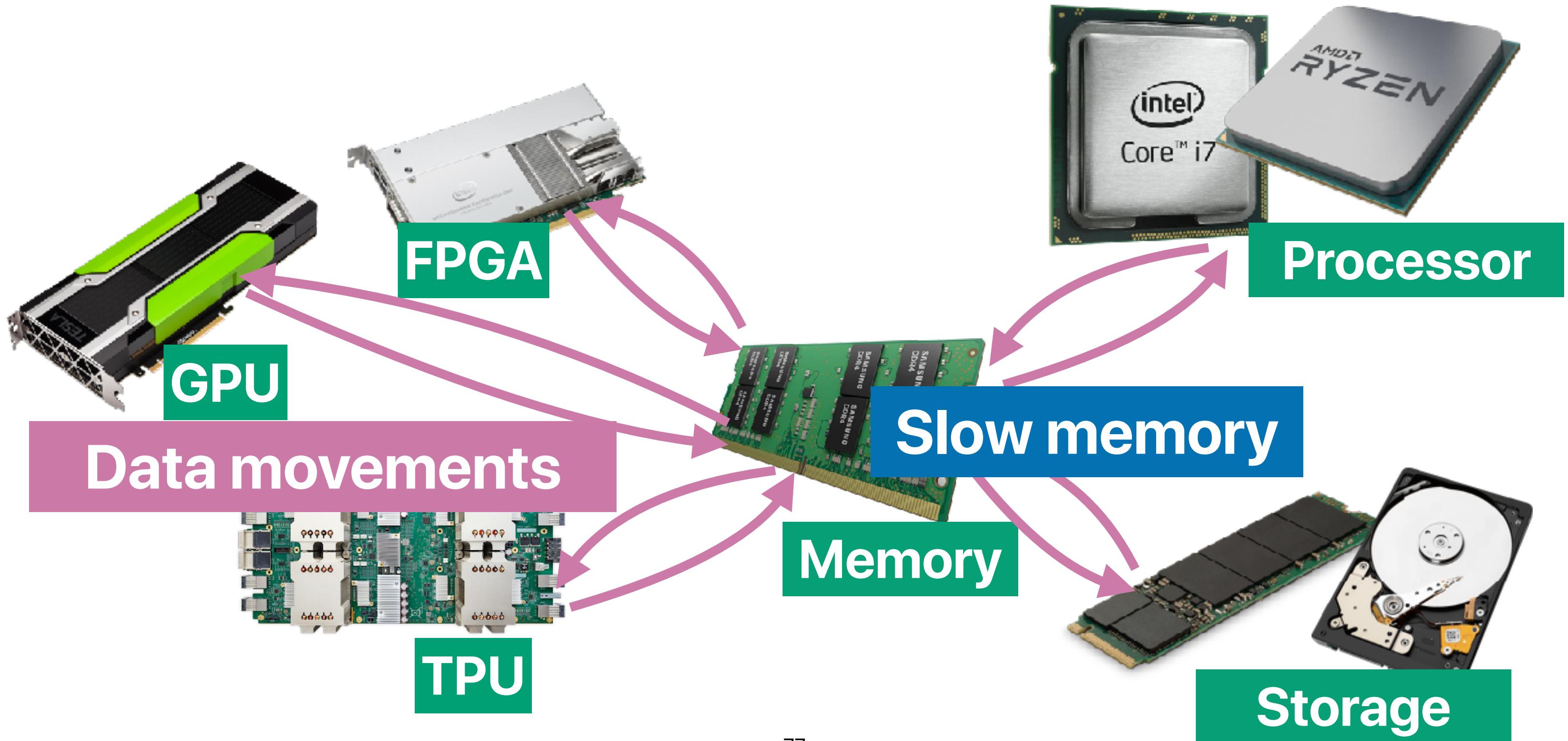
Heterogeneous Computer Architecture



Performance gap between Processor/Memory



Heterogeneous Computer Architecture



GPUs are going to replace CPUs



Take-aways: Why CSE142?

- 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

CSE142'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 optimize programs and fulfill real-world demands

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

| Date | Topic | Slides (Release) |
|------------|--|------------------|
| 08/05/2024 | Introduction: the Big Picture of Computers | |
| 08/06/2024 | Performance: What affects performance? | |
| 08/07/2024 | Performance (II) | |
| 08/08/2024 | Performance (III) | |
| 08/12/2024 | Memory Hierarchy | |
| 08/13/2024 | Memory Hierarchy (II) | |
| 08/14/2024 | Memory Hierarchy (III) | |
| 08/15/2024 | Memory Hierarchy (IV) | |
| 08/19/2024 | Midterm | |
| 08/20/2024 | Virtual Memory | |
| 08/21/2024 | Pipeline Processors | |
| 08/22/2024 | Pipelined processors (II) | |
| 08/26/2024 | Pipelined processors (III) | |
| 08/27/2024 | Pipelined processors (IV) | |
| 08/28/2024 | Parallelism (I): ILP | |
| 08/29/2024 | Parallelism (II): TLP | |
| 09/03/2024 | Parallelism (III): CMP | |
| 09/04/2024 | Parallelism (IV): SIMD and DSA | |
| 09/05/2024 | The Golden Age of Computer Architecture | |
| 09/06/2024 | Final Exam | |

Performance

Memory

Processor

Parallelism

Learning eXperience

Most lectures today ...



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 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

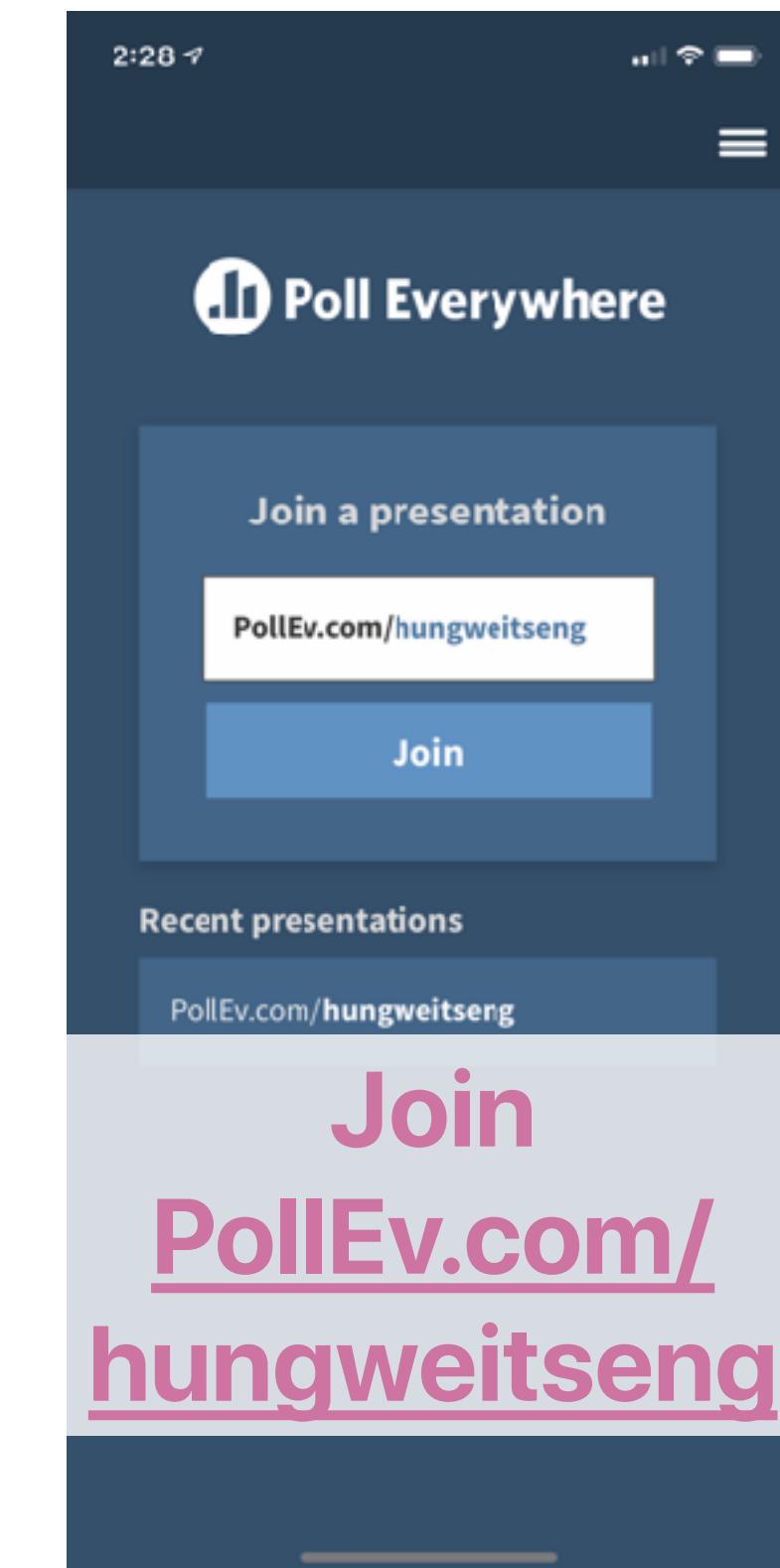
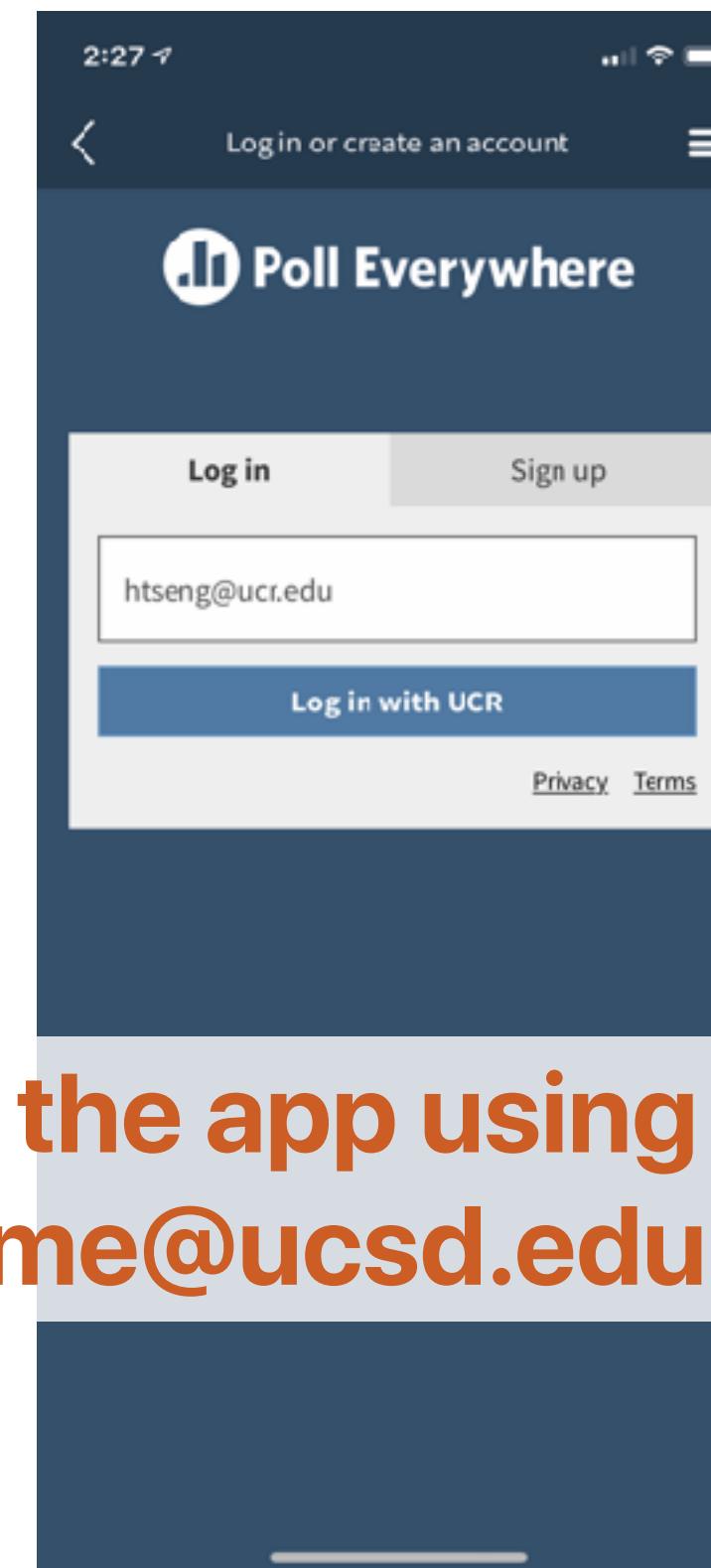
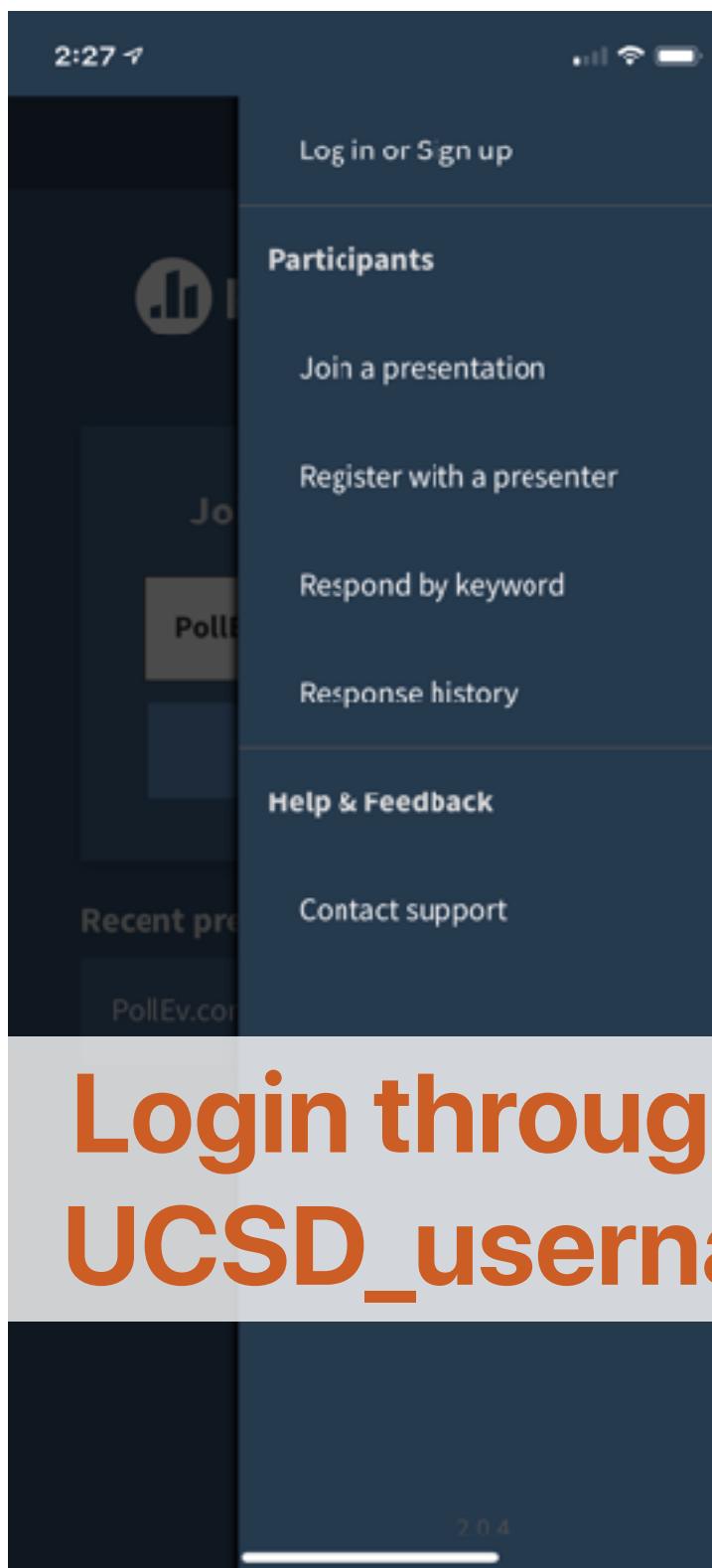
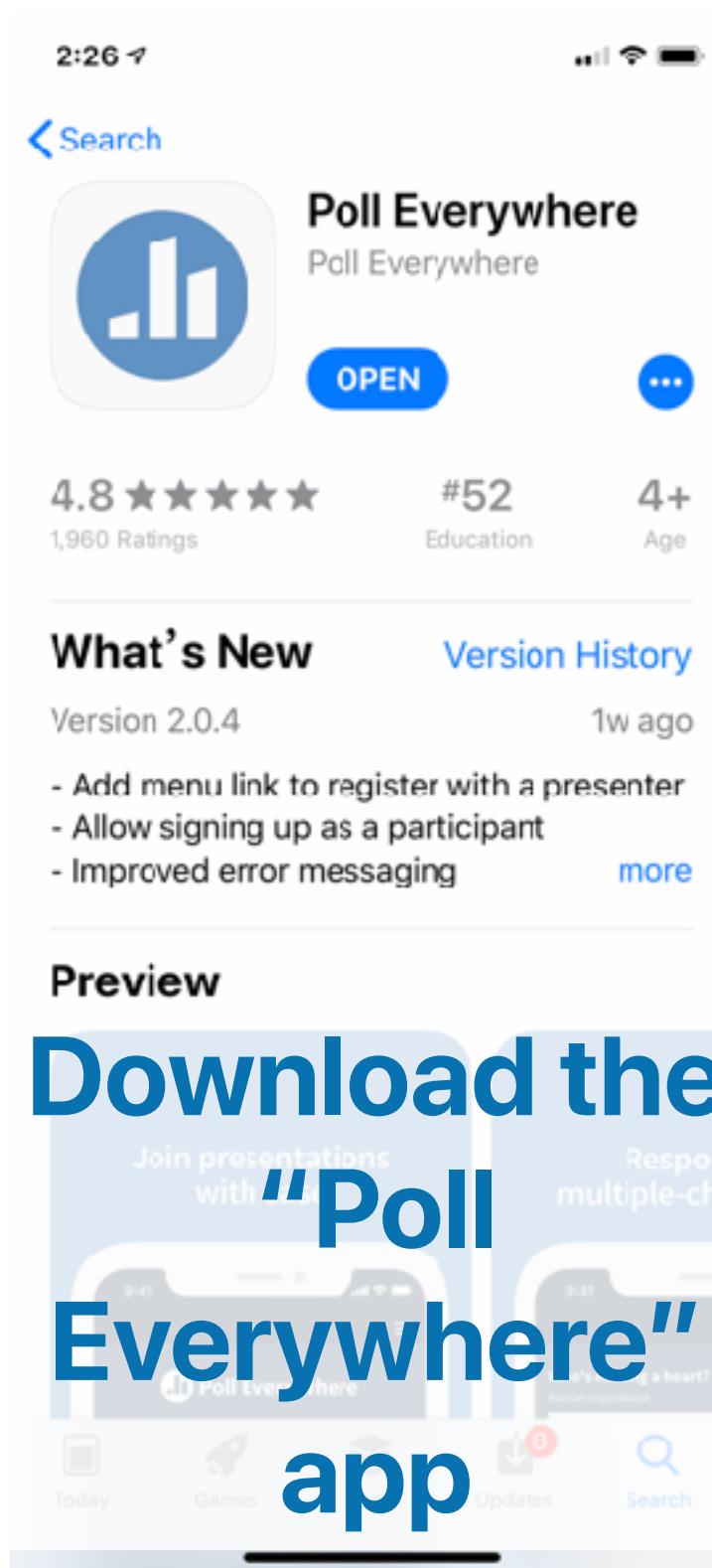
Think

Discuss

Before lectures: reading quizzes

- We need to prepare you for peer instruction activities and discussions!
- Reading assignments from
 - Randal E. Bryant and David R. O'Hallaron. Computer Systems: A Programmer's Perspective, 3/E
 - David Patterson and John Hennessy. Computer Organization and Design MIPS Edition: The Hardware/Software Interface. 6th Edition
 - Or other assigned materials
- Reading quizzes — 15% of your grades:
 - On Gradescope
 - Due before the lecture, usually twice a week. Check the schedule on our webpage
 - You will have two chances. We take the average
 - 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

- 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 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

One set of assignments for two classes

- The best way to prepare the examines
- The same jupyter notebook as CSE142L

- Watch this before your first assignment!

<https://youtu.be/7PTzdq3l4E4>

- Submit through gradescope
 - You may use the escalab.org/datahub service to finish your assignment
 - You have to click the GitHub classroom link to begin with (will post the link of each assignment on the course webpage once released)
 - Complete the questions labeled as “CSE142&CSE142L” and “CSE142 Only”
 - Submit your GitHub repo contains the CSE142/CSE142L assignment lab
 - An autograder will grade your assignment immediately and you can resubmit before the deadline — the earlier you start, the higher chances you can get full credits

You'll notice that there are three kinds of questions: "CSE142&CSE142L", "CSE142 Only", and "CSE142L Only".

• **CSE142&CSE142L:** If you're submitting CSE142 assignments, you need to complete all CSE142&CSE142L question. You also need to complete all CSE142 Only question to get full credits.

• **CSE142 Only:** If you're submitting CSE142 assignment/lab, you need to complete all CSE142&CSE142L and CSE142 Only question.

• **CSE142L Only:** If you're submitting CSE142L assignment/lab, you need to complete all CSE142&CSE142L and CSE142L Only question.

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/

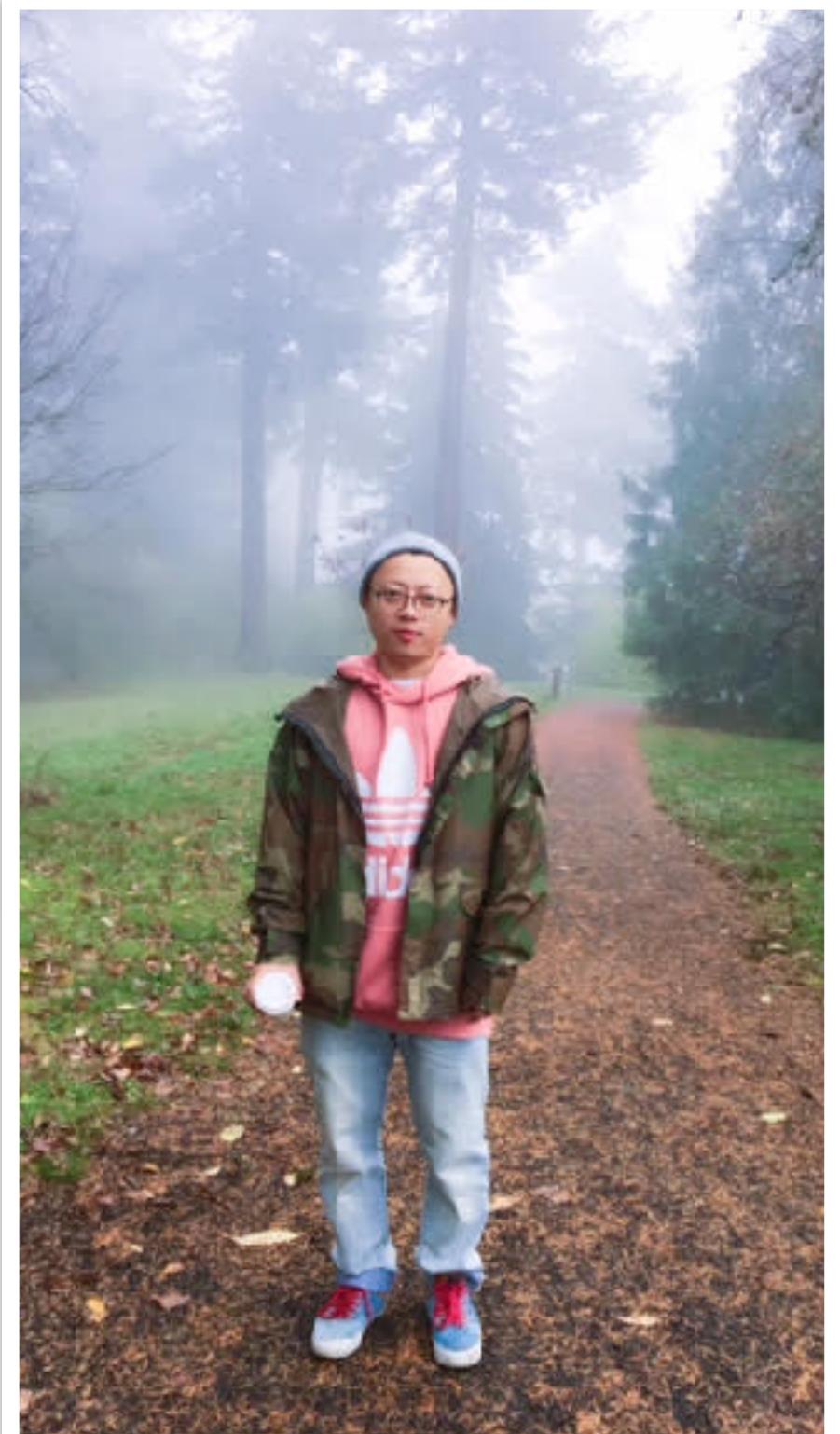
Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- E-mail: htseng @ ucsd.edu
- 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
- Fun fact: Hung-Wei was once considering a career path as a the unsuccessful trial

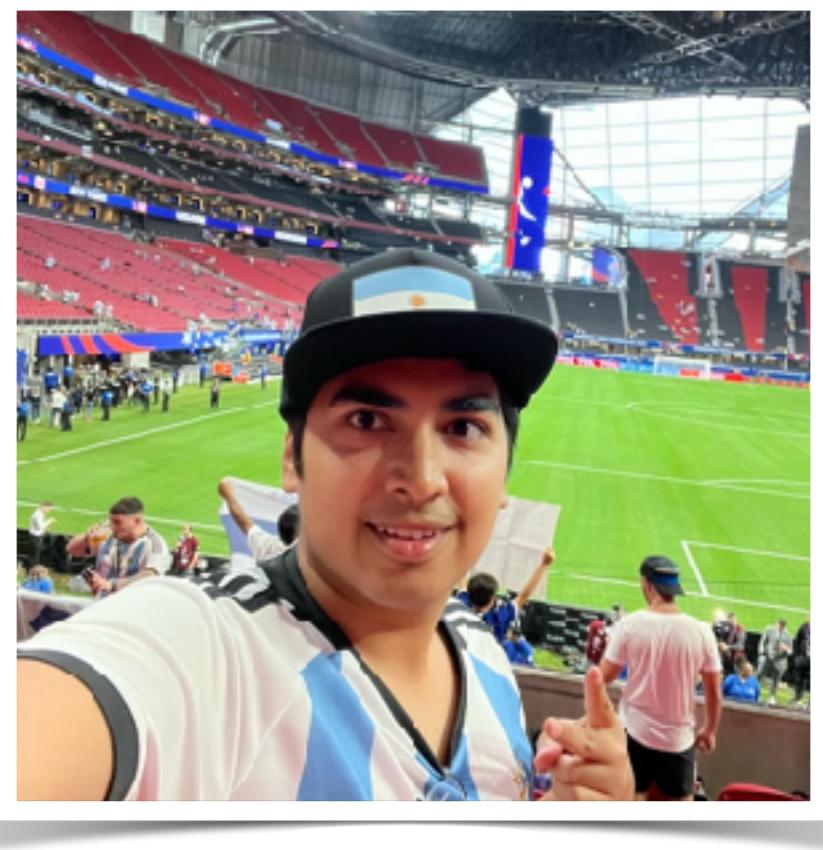


Teaching Assistants

- Fucheng Shang
- Office hours: TuF 10a-12p @ CSE B215
- E-mail: cse142_su24 @ escalab.org
- Fun fact: I had a dog who liked to poop on my bed and I gave him to my cousin.



Tutors



Peeyush Jha

**TuTh 9a-11a @ B270A
W 5p-6p Online**



Will Luo

**Tu 6p-8p @ B250A
W 5p-8p @ B250A**



Jian Chen Yan

**MTu 3:30p-5p
W 12p-2p
Online**

Subscribe to our google calendar!

| | | | | | |
|--|--|--|---|--|--|
| CSE 142L - Alexander's Office Hours (Remote) 11am – 2pm | CSE 142L Daniel's OH 9am – 2pm B270 | CSE 142 - Peeyush's OH 9 – 11am B270 | CSE 142L - Alexander's Office Hours (Remote) 9am – 12pm | CSE 142 - Peeyush's OH 9 – 11am B270A | CSE142 - Fucheng's OH 10am – 12pm B215 |
| CSE142 (Lecture) 2 – 3:30pm | CSE142 (Lecture) 2 – 3:30pm | CSE142 (Lecture) 2 – 3:30pm | CSE142 - JC's OH 12 – 2pm | CSE 142L - Honghao OH 11am – 1pm https://ucsd.zoom.us/j/91799877409 | CSE142L - Honghao OH 12 – 5pm https://ucsd.zoom.us/j/99877409 |
| CSE 142 - JC's OH (Virtual) 3:30 – 5pm | CSE 142 - JC's OH (Virtual) 3:30 – 5pm | CSE142/L Office Hours (Hung-Wei 3:30pm, Computer Science and En | CSE142/L Office Hours (Hung-Wei 3:30pm, Computer Science and En | CSE142 (Lecture) 2 – 3:30pm | CSE142L - Yinong's OH 12 – 5pm CSE basement B250A |
| CSE142L (Lecture) 5 – 6pm | CSE 142 - Will's OH 5 – 8pm | CSE 142 - Will's OH 5 – 8pm | CSE 142 OH - Peeyush's OH 5pm, https://ucsd.zoom.us/j/91799877409 | CSE142L (Lecture) 5 – 6pm | CSE142L - Honghao OH 7 – 9pm https://ucsd.zoom.us/j/91799877409 |
| CSE 142 - Will's OH B250A 6 – 8pm | | | | | |

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

Discussion sessions

- Fridays 1p-3p Pepper Canyon Hall 121
- Covers hints for your assignments
- Highly recommended

CSE142: Computer Architecture: S Summer Session II)

The website

- Calendar
- Schedule
- Slides
 - Preview — for the ease of note taking
 - Release — the actual slides

Online: Zoom or <https://www.youtube.com/profusagi>

Lecture: MTuWTh 2:00p – 3:20p @ WLH 2205 or Zoom

Schedule and Slides Assignments

Hung-Wei Tseng

今天 2023年8月

| 週日 | 週一 | 週二 | 週三 | 週四 | 週五 | 週六 |
|---|---|---|---|-------|-------|----|
| 30 | 31 | 8月 1日 | 2 | 3 | 4 | |
| | | | | | | |
| 6 | 7 | 8 | 9 | 10 | 11 | |
| 上午9:30 CSE 142 Ch 上午10點 Andrei office 上午10點 CSE142 Lec 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's | 上午10點 Andrei office 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午5點 CSE142 Disc 下午8:30 CSE 142L Eij | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | +2 更多 | | |
| 13 | 14 | 15 | 16 | 17 | 18 | |
| 上午9:30 CSE 142 Ch 上午10點 Andrei office 上午10點 CSE142 Lec 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | +2 更多 | | | |
| 20 | 21 | 22 | 23 | 24 | 25 | |
| 上午9:30 CSE 142 Ch 上午10點 Andrei office 上午10點 CSE142 Lec 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | +2 更多 | | | |
| 27 | 28 | 29 | 30 | 31 | 9月 1日 | |
| 上午9:30 CSE 142 Ch 上午10點 Andrei office 上午10點 CSE142 Lec 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | 上午10點 Andrei office 上午11:30 CSE 142L H 下午2點 CSE142 Lect 下午3:30 Hung-Wei's 下午4點 CSE142 Disc 下午8:30 CSE 142L Eij | +2 更多 | | | |

活動顯示的時區：太平洋時間 - 洛杉磯

Google 日曆

Summary of course resources

- Lectures:
 - In-person @ **PCYNH 121**
 - Repository on **Youtube**: <https://www.youtube.com/profusagi>
- Schedule, slides, grades on **course webpage**:
<https://www.escalab.org/classes/cse142-2024su/>
- Discussion on **piazza**:
https://piazza.com/ucsd/summer2024/cse142_s224_a00/home
- Reading quizzes and assignments through **gradescope**:
<https://www.gradescope.com/courses/813375>
- Working environment on <https://escalab.org/datahub>
- Office Hours & Locations
https://calendar.google.com/calendar/embed?src=c_373ea7ba1adb25dcb44c3a3d1cb62af934f7601955381cdc89116d91596ba4af%40group.calendar.google.com

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 UCSD 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 difficulty coming to the 50% of class, you can still earn all credits if you ace all assignments.
- 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 I need to take CSE142/L together?
 - Strongly encouraged
 - We have a combined CSE142/L assignments & lab reports this time. You just need to do a few more questions for each assignment and write two programming assignments this summer (the first PA is simply a toy one) and you can get CSE142L done.
- Do we have a canvas page?
 - No — 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
 - Please email htseng@ucsd.edu 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
 - Yes, but only to our best efforts — I cannot force tutors/TA to do Zoom office hours
 - Please use piazza 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

UC San Diego

NC STATE UNIVERSITY

UC RIVERSIDE



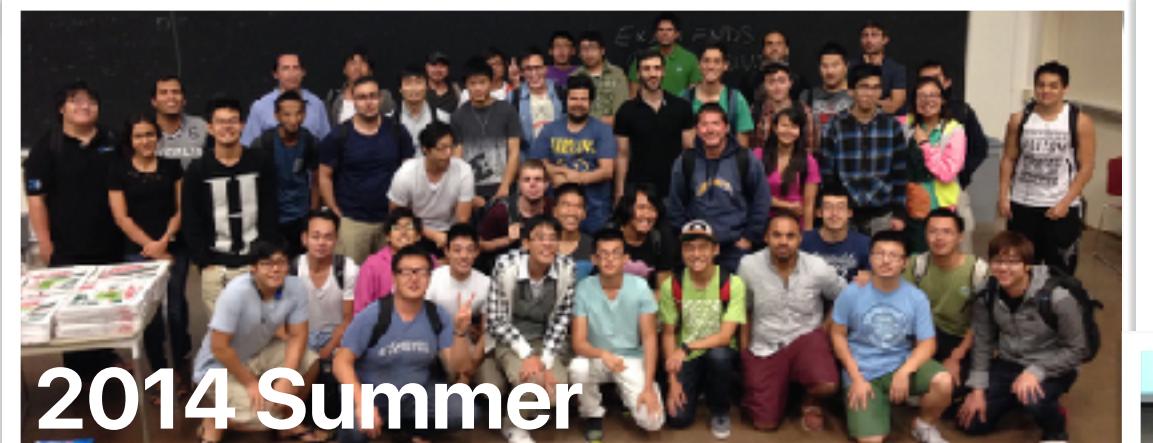
2012 Summer



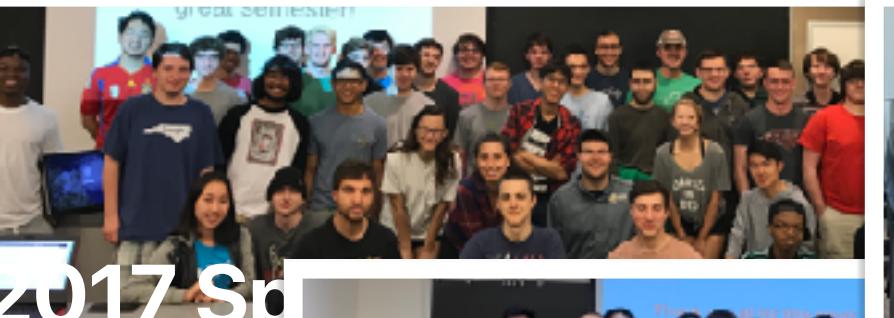
2016 Fall



2022 Winter



2014 Summer



2017 Sp



2022 Fall



2021 Fall



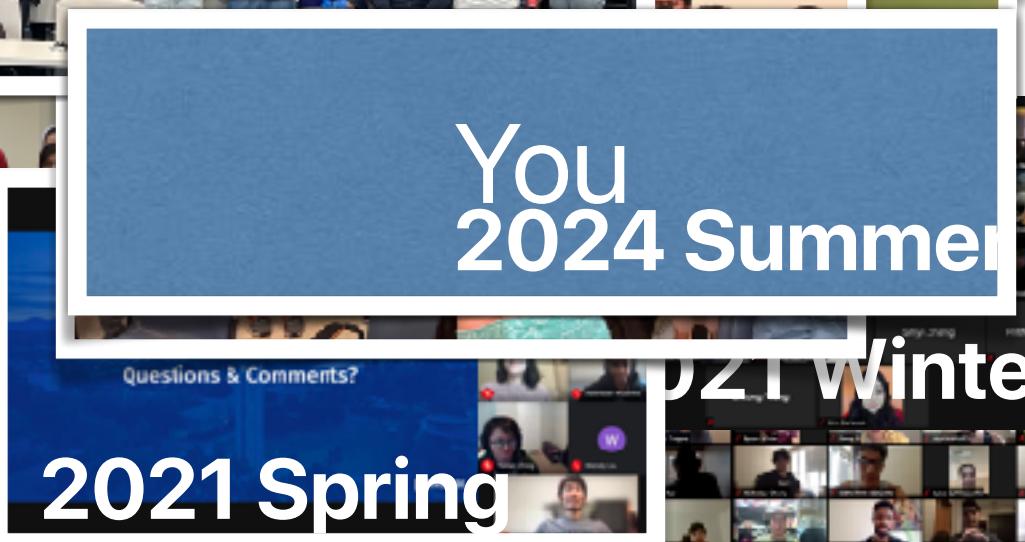
2016 Spring



2017 Fal



2019 Fall



2021 Spring



2021 Winter



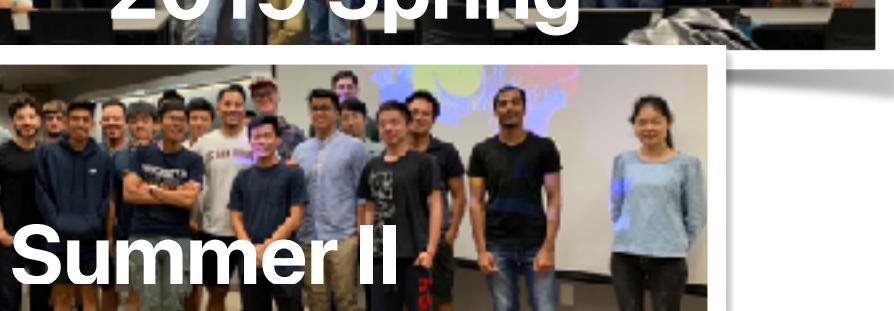
2016 Summer



2018 Fall



2019 Summer I



2019 Summer II



2020 Spring



2020 Summer

Announcements

- Login to piazza, datahub, gradescope
 - Let us know if you have any issue
cse142_su24@escalab.org
- Reading quiz on Gradescope due **tomorrow before the lecture**
- Assignment #1 is released — working on datahub!
 - Watch this video before you start! <https://youtu.be/7PTzdq3I4E4>
- Check our website
 - Check the lab website if you have sometime as well!

Computer Science & Engineering

142

つづく

