

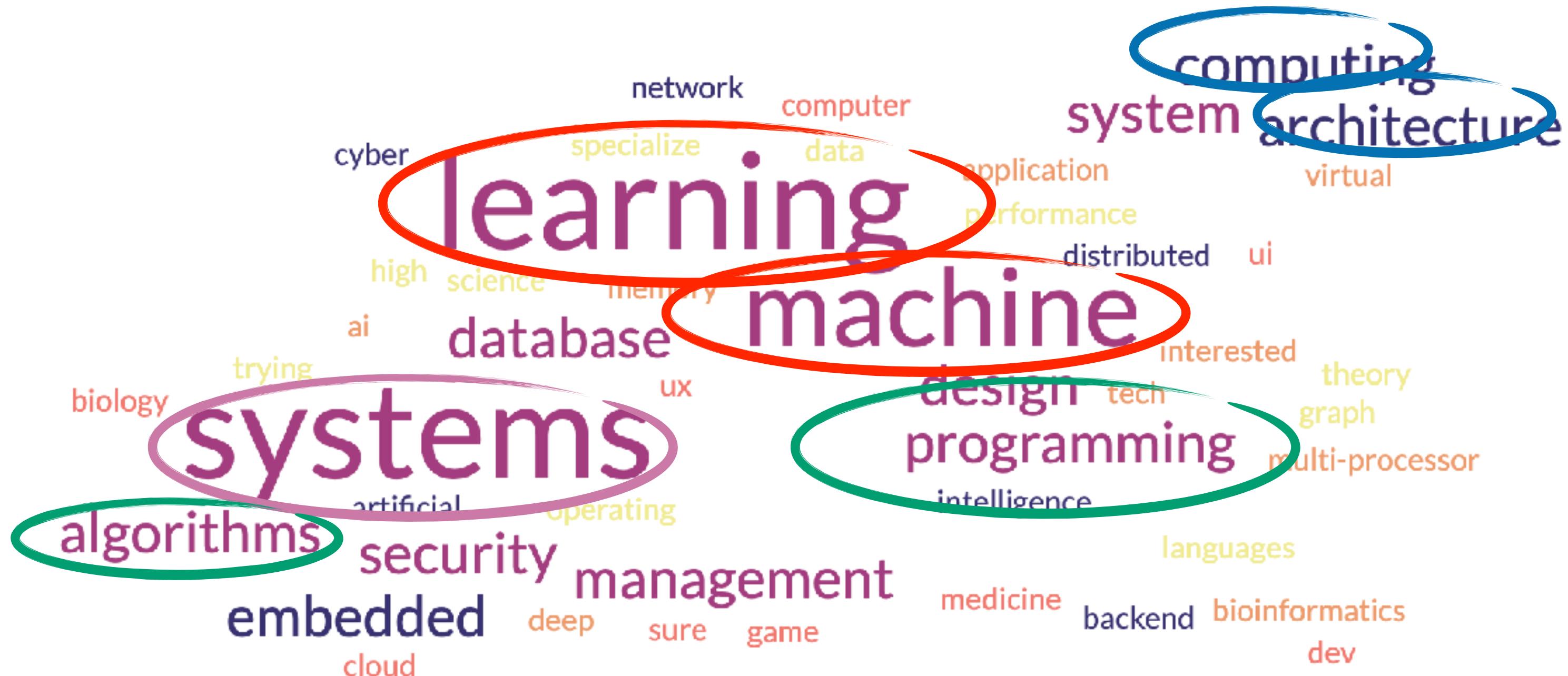
Why

CS203: Advanced Computer Architecture

Hung-Wei Tseng



What are you interested



Why are we taking CS203

ChatGPT 3.5

HT 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
Software Engineering
AI & MI
Computer Architecture
Programming Languages
User Interfaces

What are the most important topics in computer sciences?

ChatGPT 3.5

- 5. **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.

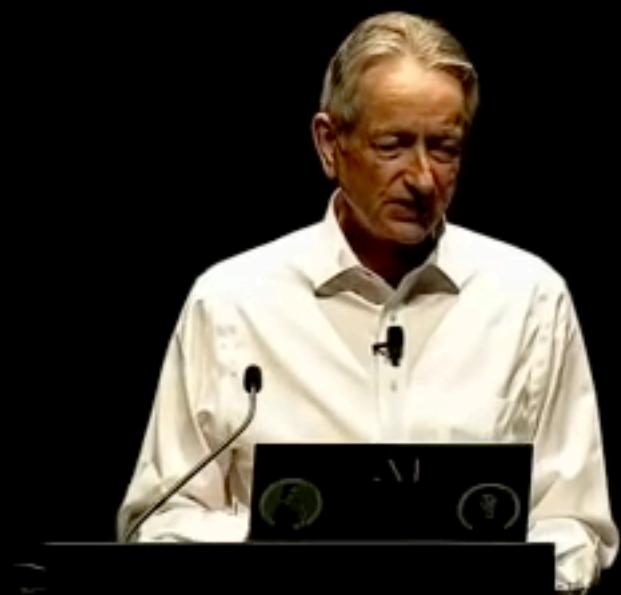
These topics represent just a fraction of the diverse and dynamic field of computer science. New areas of research and innovation continually emerge as technology advances.

The screenshot shows a Gemini search interface with the following details:

- Header:** Gemini
- Address Bar:** gemini.google.com...
- Section Header:** Computer Networking computer science
- Text Description:** 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.
- Diagram:** A flowchart illustrating the data flow from students to publications. It shows a central "UNIVERSITY" node connected to "Students", "Faculty", and "Administrators". "Students" leads to "Courses", which then lead to "Papers" and "Publications". "Faculty" leads to "Courses", which then lead to "Publications". "Administrators" leads to "Courses", which then lead to "Publications".
- Text Description:** Databases computer sciences
- Text:** 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...
- Footer:** What are the most important topics in computer sciences? I

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

ChatGPT 3.5

4. **System Bus and Interconnects:** The system bus and interconnects provide pathways for data and control signals to flow between different components of the computer system, including the CPU, memory, and I/O devices.

5. **Instruction Set Architecture (ISA):** The ISA defines the set of instructions that a CPU can execute and the format of those instructions. It serves as an interface between software (instruction-level code) and hardware (CPU implementation).

6. **Parallelism and Pipelining:** Techniques such as parallel processing and instruction pipelining are used to improve performance by executing multiple instructions simultaneously or overlapping the execution of multiple instructions.

7. **Power and Energy Management:** With the increasing emphasis on energy efficiency, computer architecture also involves techniques for reducing power consumption and managing energy usage in computer systems.

Computer architects design and optimize these components to achieve specific performance, cost, power, and reliability goals for different types of computing systems, ranging from personal computers and servers to embedded systems and supercomputers. They consider factors such as instruction throughput, memory bandwidth, latency, scalability, and trade-offs between different design choices to meet the requirements of diverse applications and workloads.

Why is computer architecture important?

architecture

Gemini

See the latest updates to the Gemini Apps Privacy Hub

breakdown of the concept:

Core Idea:

- It's a blueprint that dictates how software interacts with hardware to make a computer function.
- It focuses on the end-to-end structure, detailing how data flows and gets processed within the system.

Two Key Levels:

- Instruction Set Architecture (ISA):** This is like a common language between the software and the CPU. It defines the set of instructions the processor understands and how data is formatted for processing.
- Microarchitecture:** This dives deeper into the actual physical components and their connections within the CPU. It focuses on how data is processed within the processor itself.

Balancing Act:

- Computer architecture is concerned with achieving a good balance between various factors like:
 - Performance: How fast the system can execute instructions.
 - Efficiency: How well the system utilizes resources.
 - Cost: The affordability of the system's design and implementation.
 - Reliability: How dependable the system is in performing its tasks.

Analogy:

Imagine a computer architecture as a recipe for a delicious meal. The ISA is like the ingredient list and instructions, while the microarchitecture is like the specific cooking techniques and tools used in the kitchen. Both are essential for creating the final product (the uncooked dish or the cooked meal).

Enter a prompt here

ChatGPT can make mistakes. Consider checking important information.

Computer Architects — Hidden Figures

BASED ON THE UNTOLD TRUE STORY

HIDDEN FIGURES



Your impression about Computer Architecture

engineer
systems
course
components
helps etc
together

flow
think
interaction
including

structure
little
study

design

fundamental used work compiler
processed close

layout concepts working
things functions internal chips
general data cpu logical embedded
system required classes basic
decent processors

memory disk
organisation developed
idea made
computers developer

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...

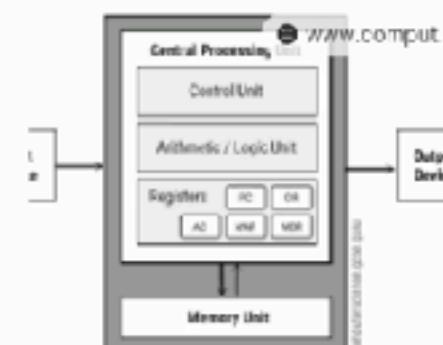
ChatGPT can make mistakes. Consider checking important information.

puter a

Gemini

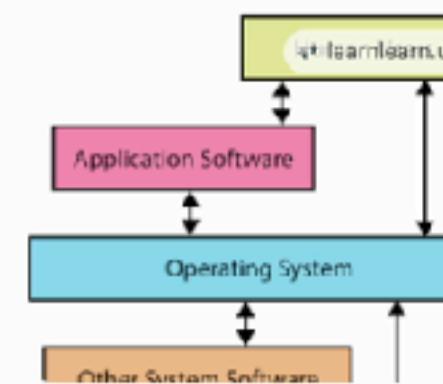
See the latest updates to the Gemini Apps Privacy Hub

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-ki-ték-chör \

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 and 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

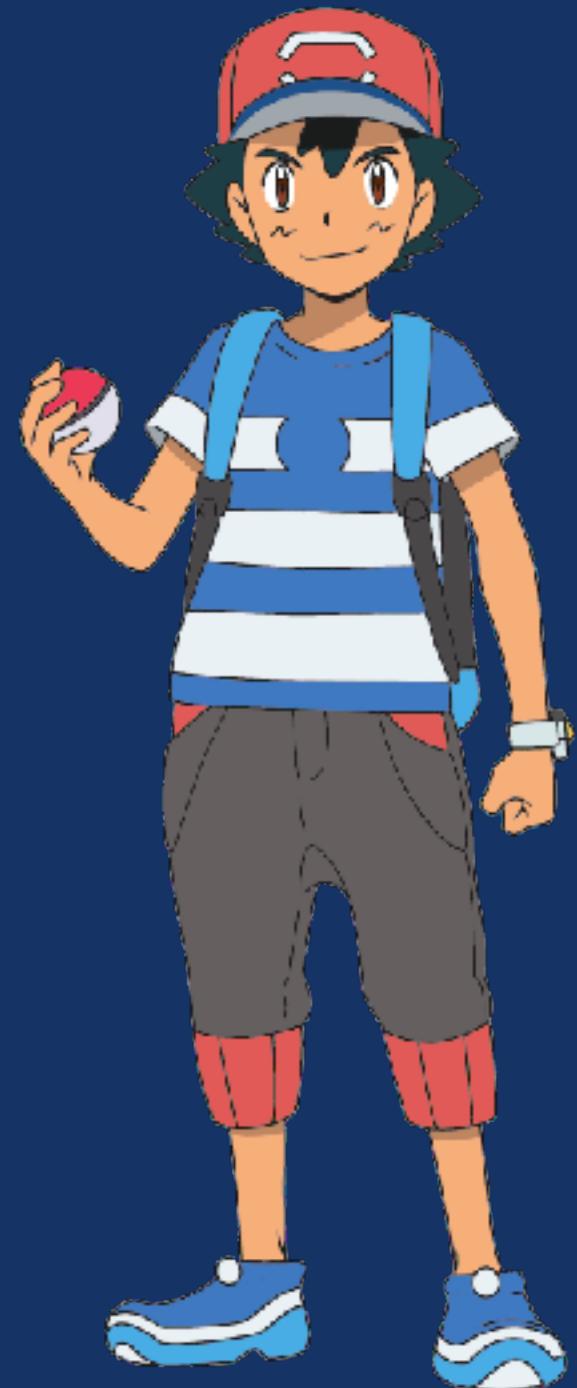
4 : a method or style of building

// 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**



?????

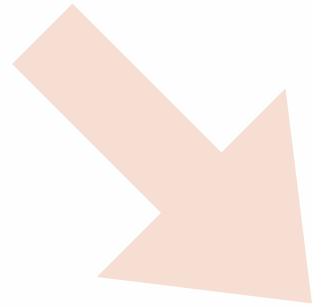
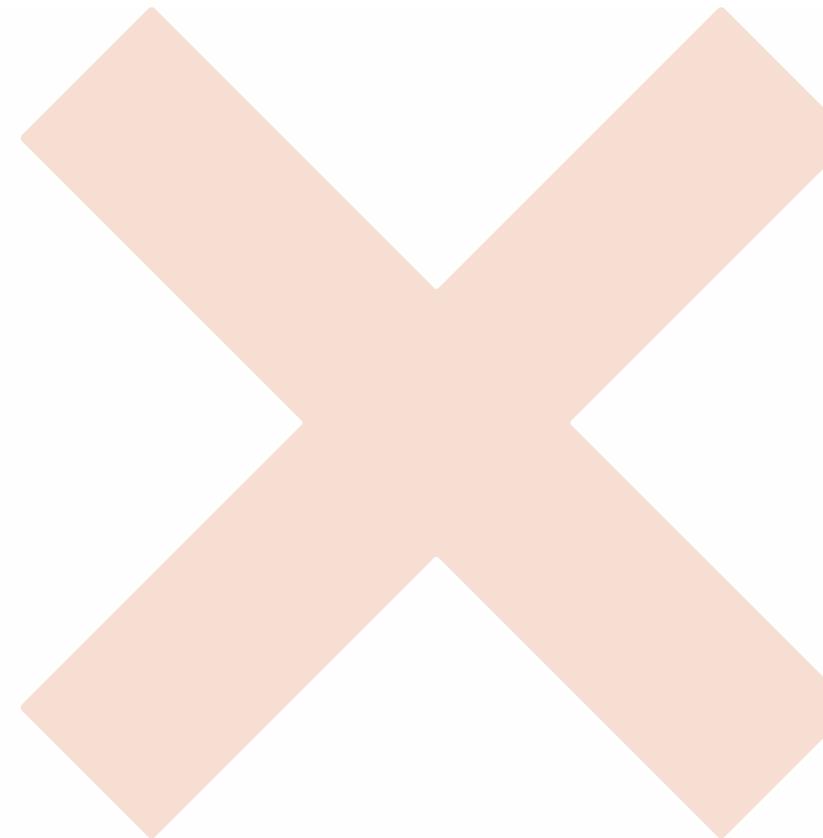
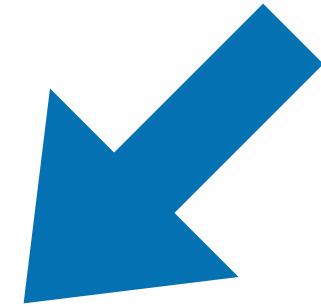
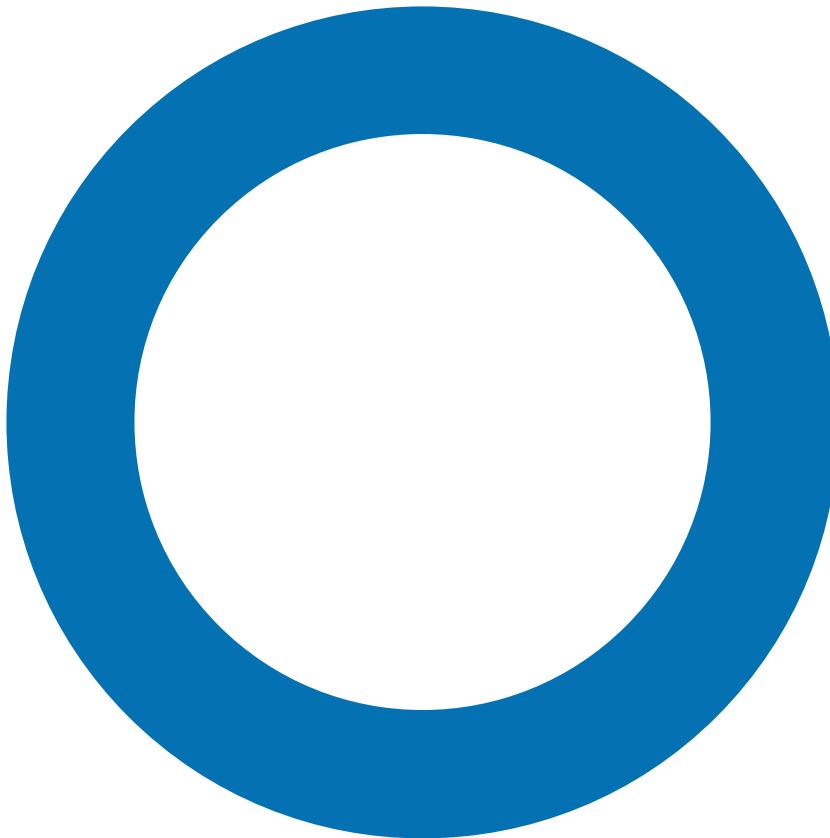


Or a Tesla

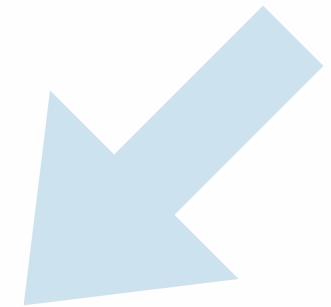
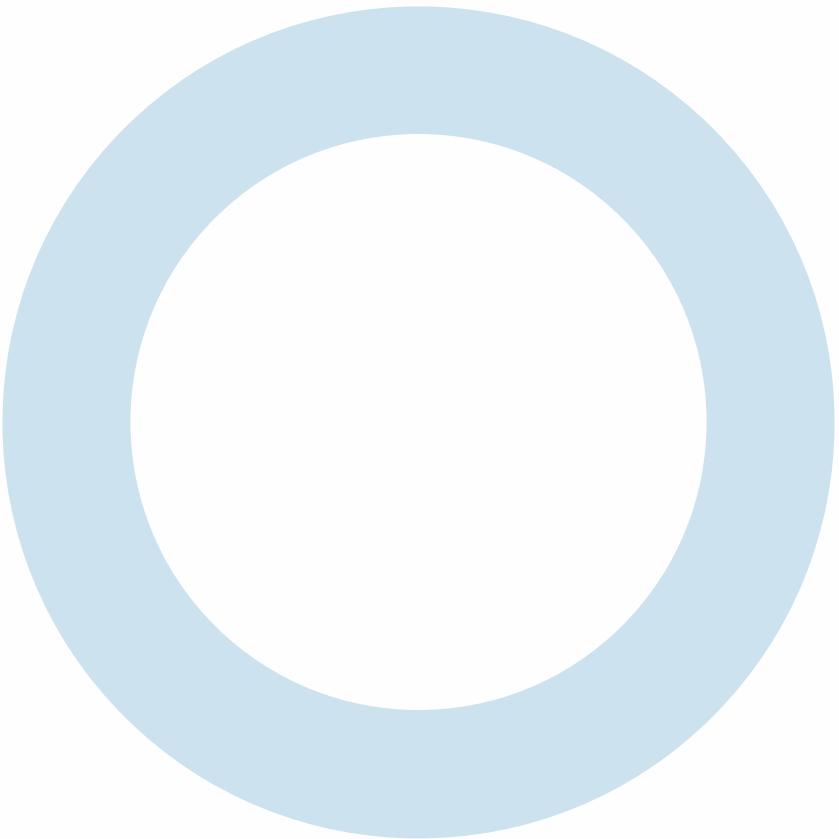


**How much do we understand
“Computer Architectures” for now**

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



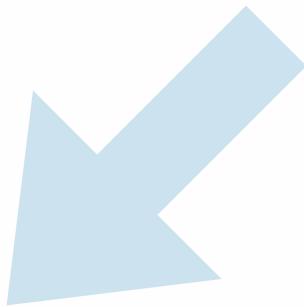
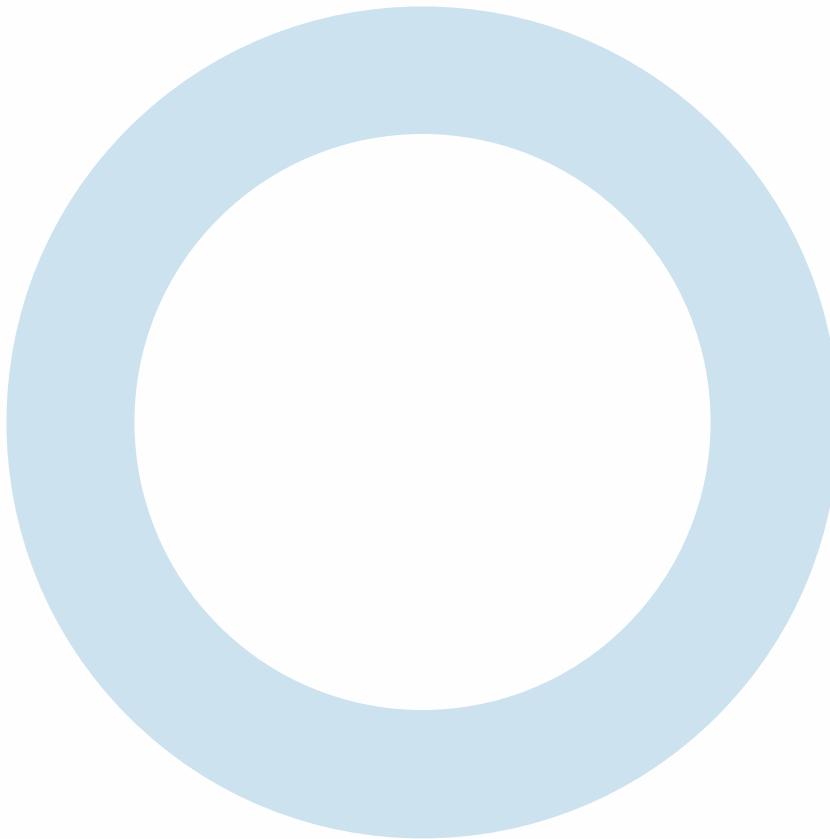
On the same hardware, programs with lower computational complexities are more efficient



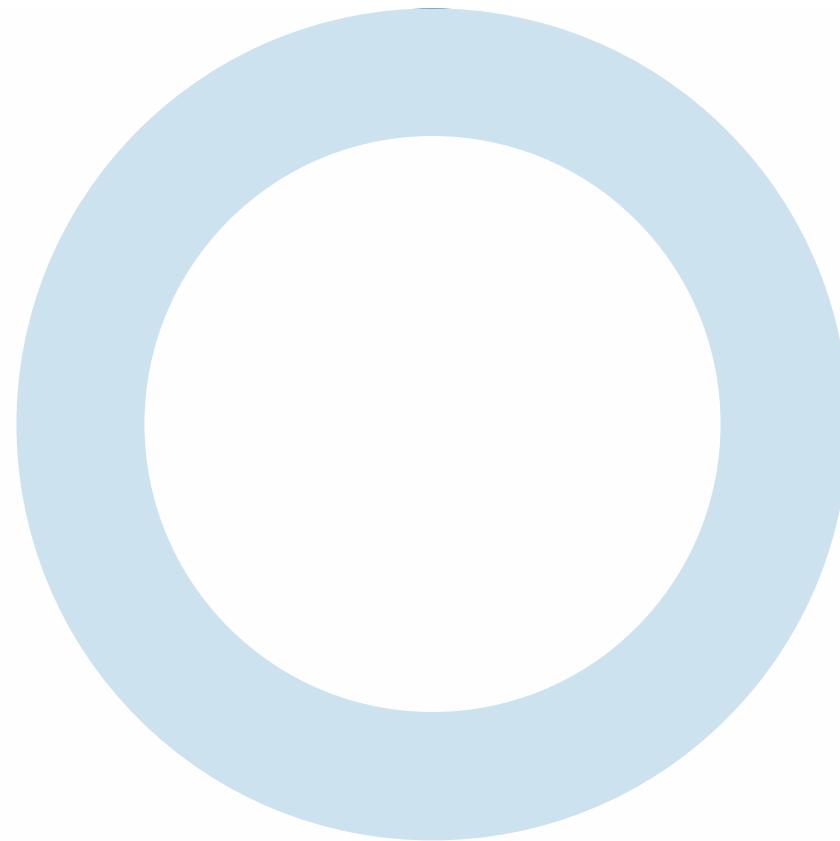
On the same hardware, program performance scales with its main algorithm complexity



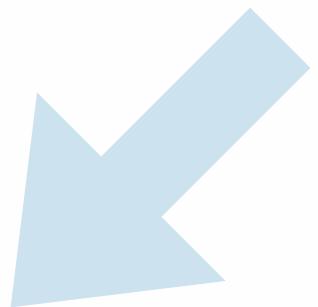
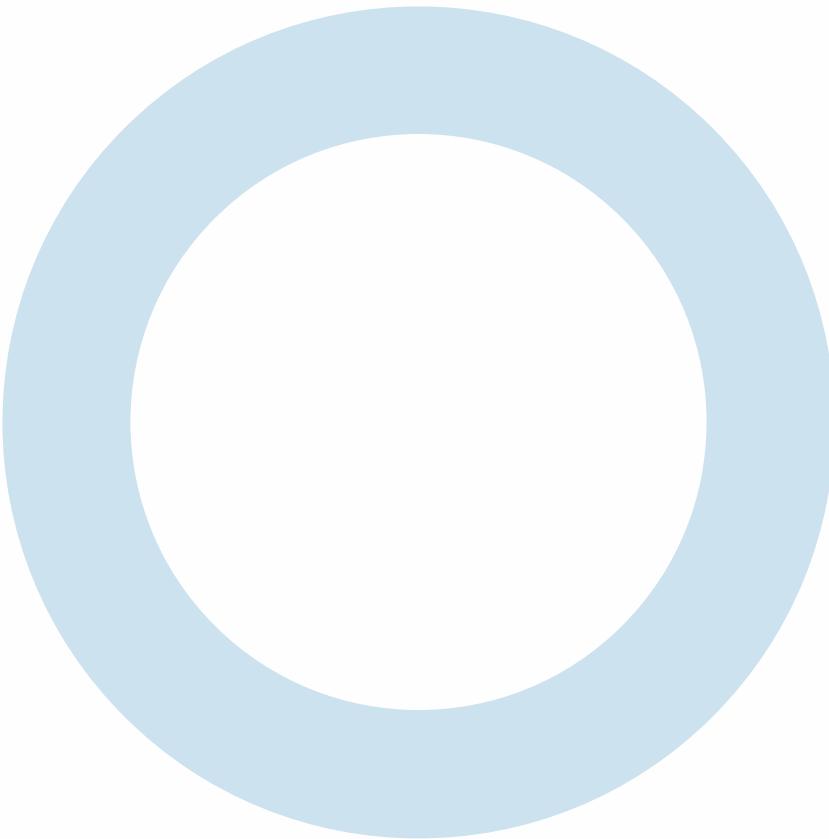
Leveraging more “bit-wise” operations in C code will make the program significantly faster



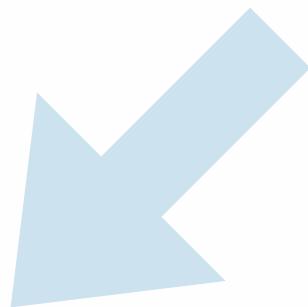
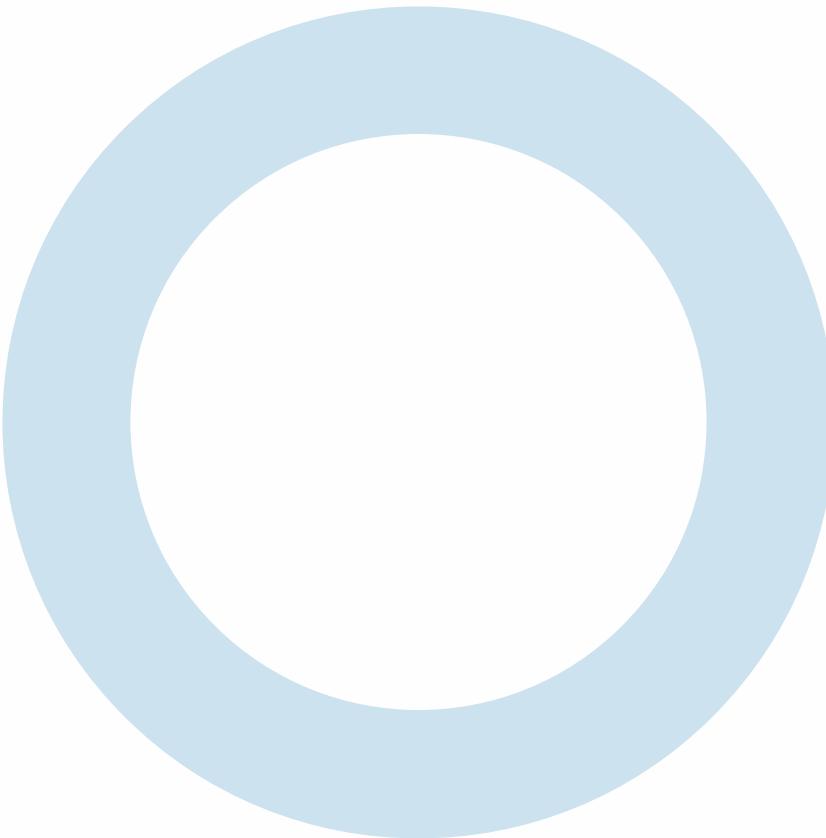
On the same CPU, each instruction typically takes the same orders of magnitude of execution time



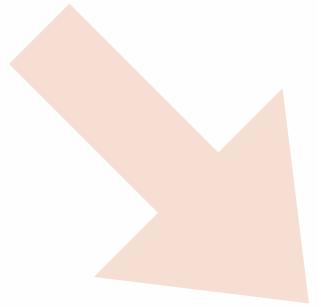
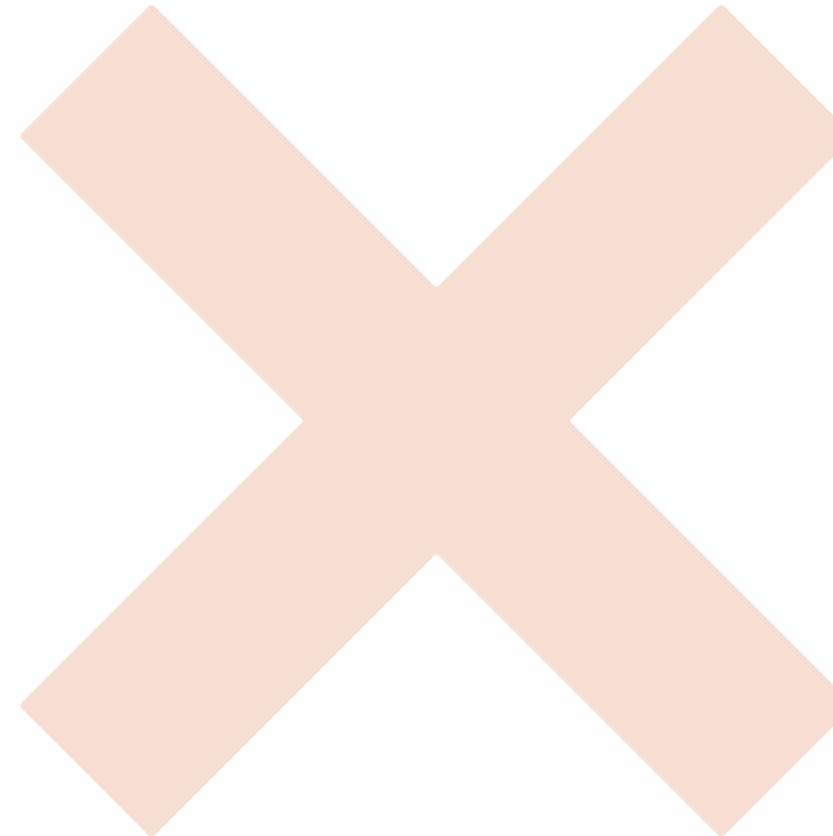
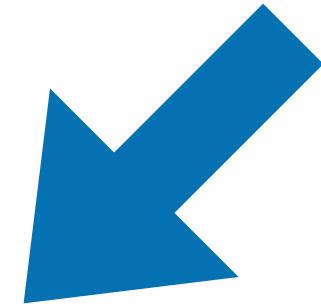
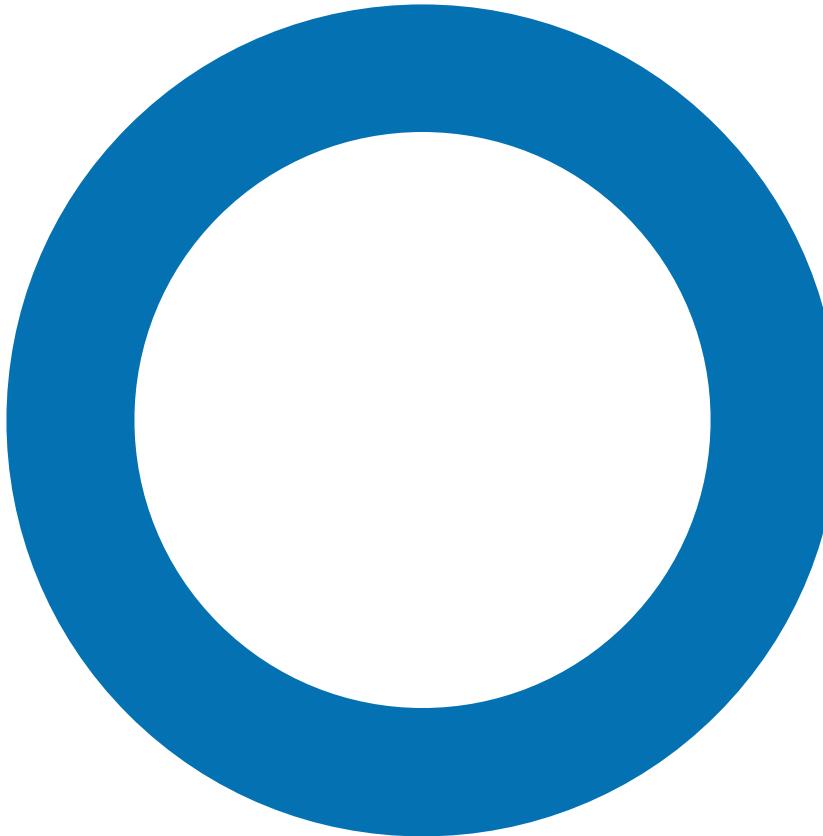
Moore's Law is current slowing/discontinuing



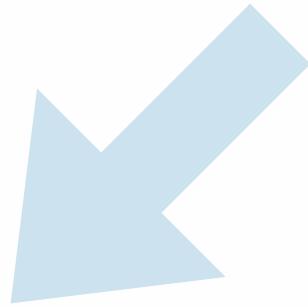
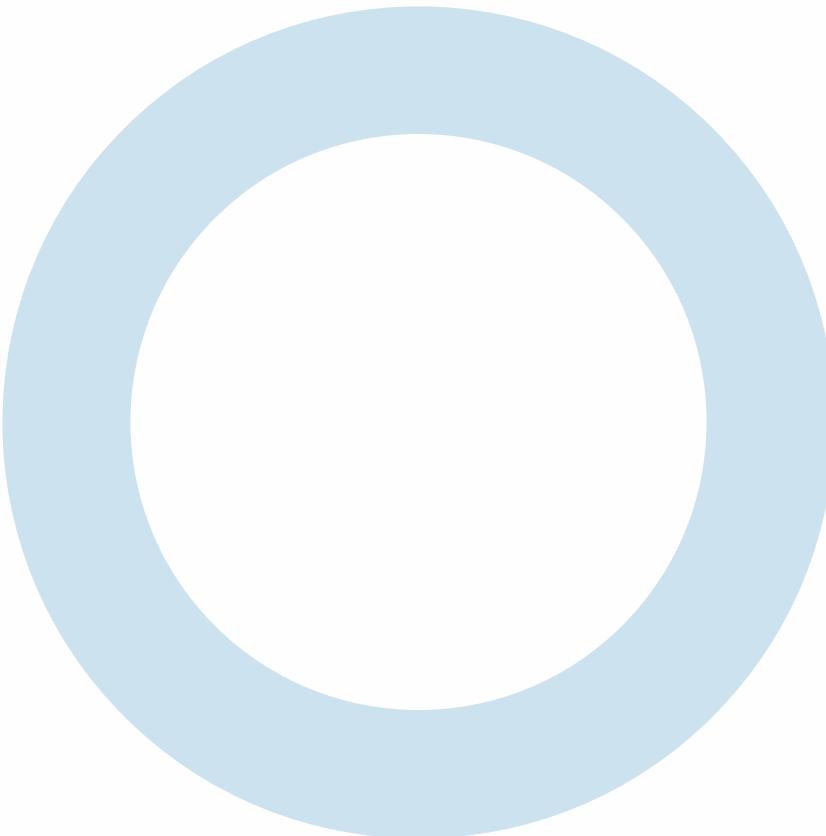
**The smaller size of a transistor,
the smaller power consumption of it**



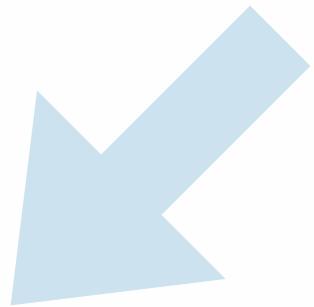
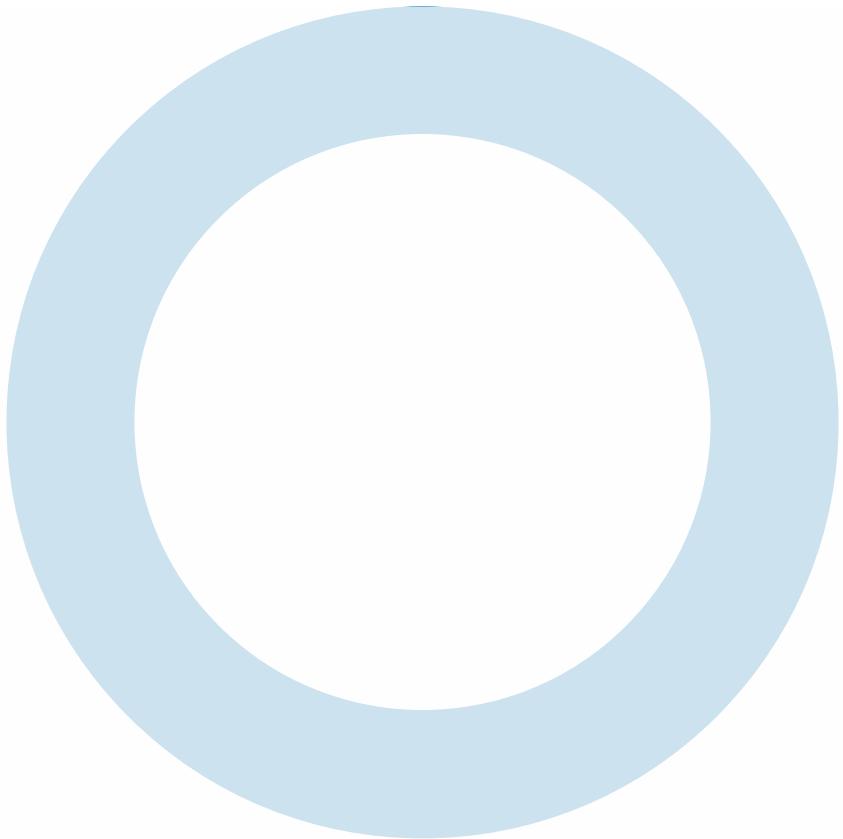
Algorithm complexity is less important if we have unlimited parallelism



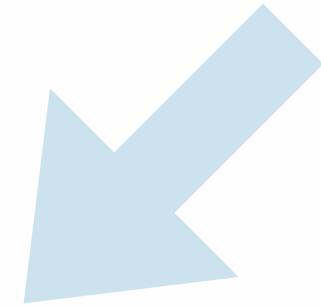
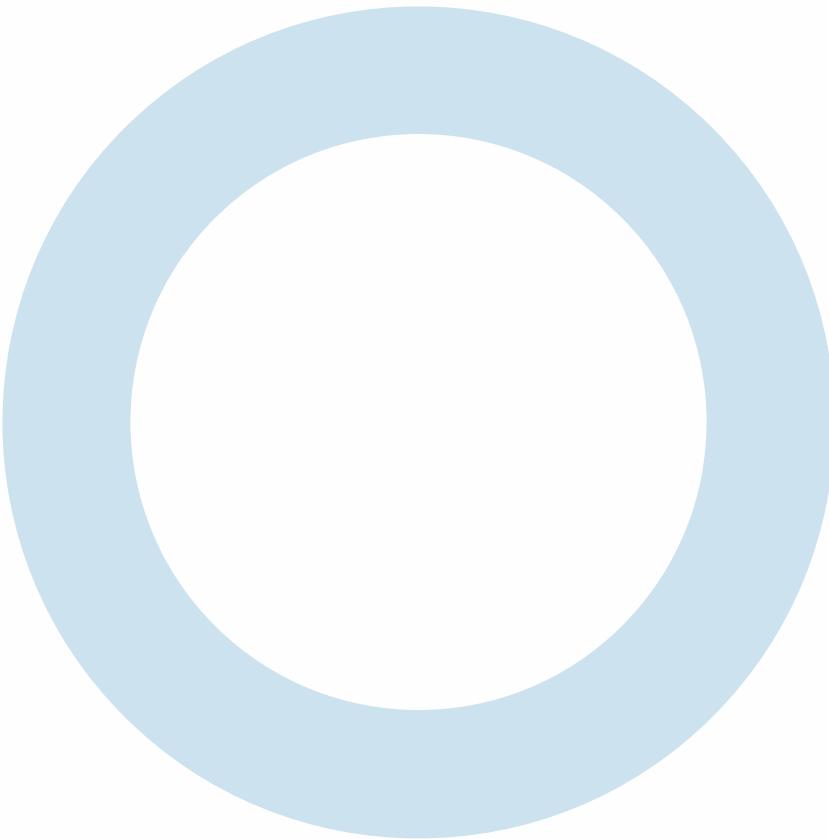
GPUs are more power efficient per transistor than CPUs



**With 1000 NVIDIA's most high-end GPUs,
we can train a GPT-4 model within a month**

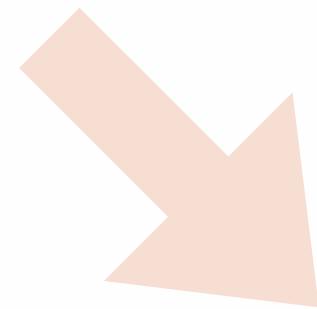
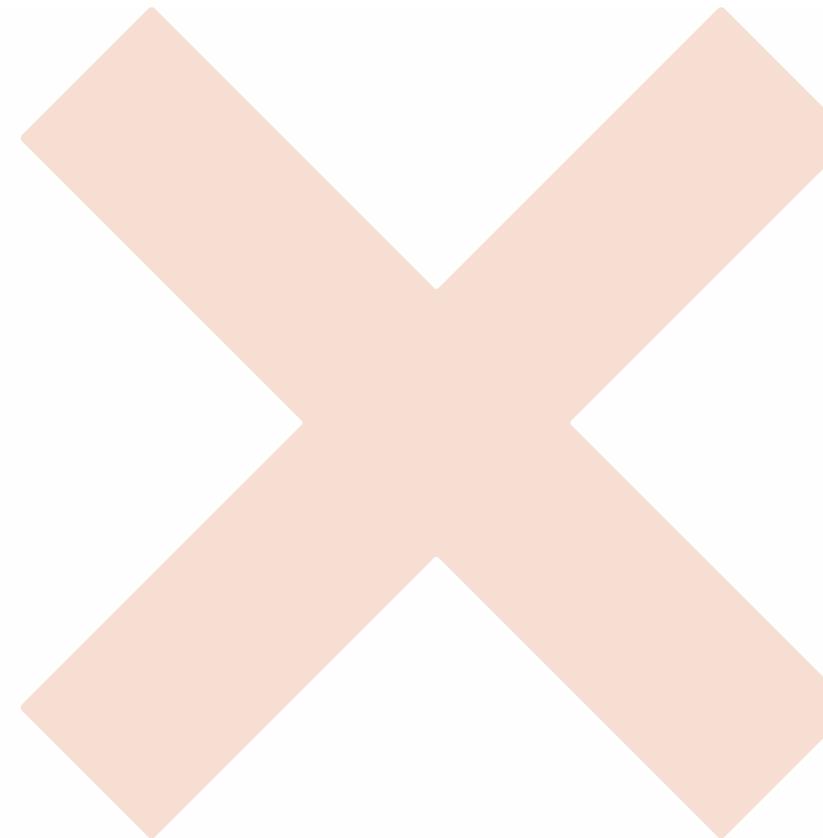
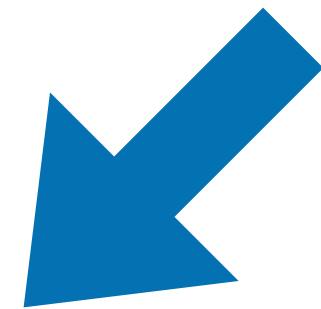
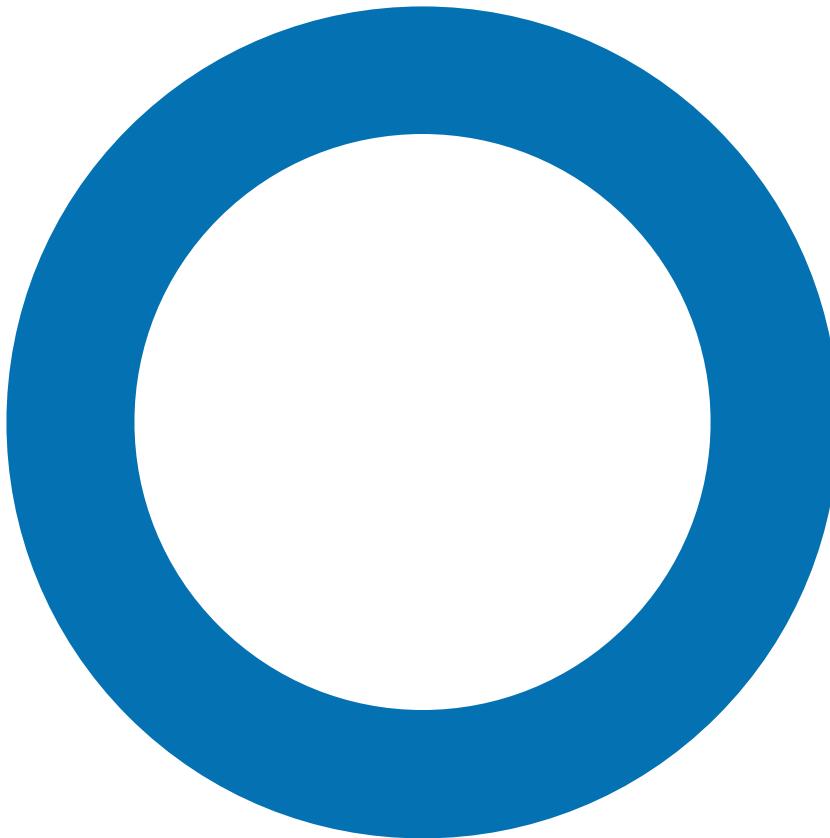


GPUs are going to replace CPUs



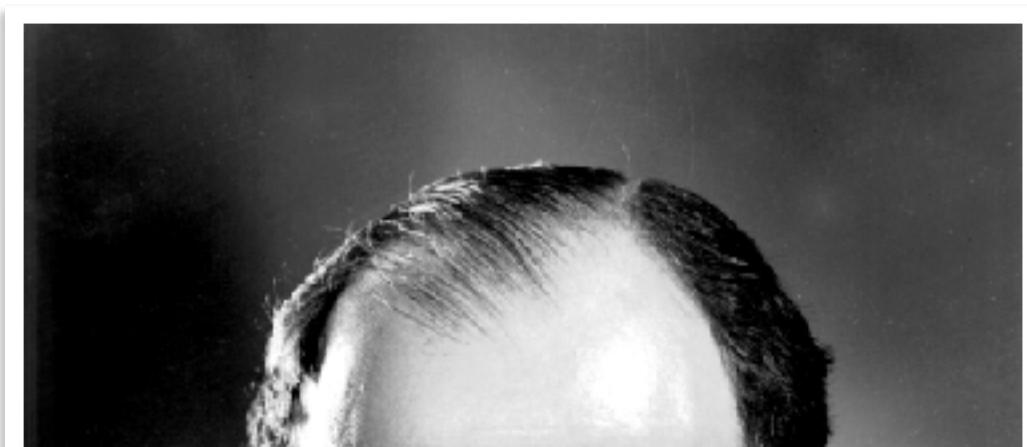
**We need to learn more about
computer architectures!**

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

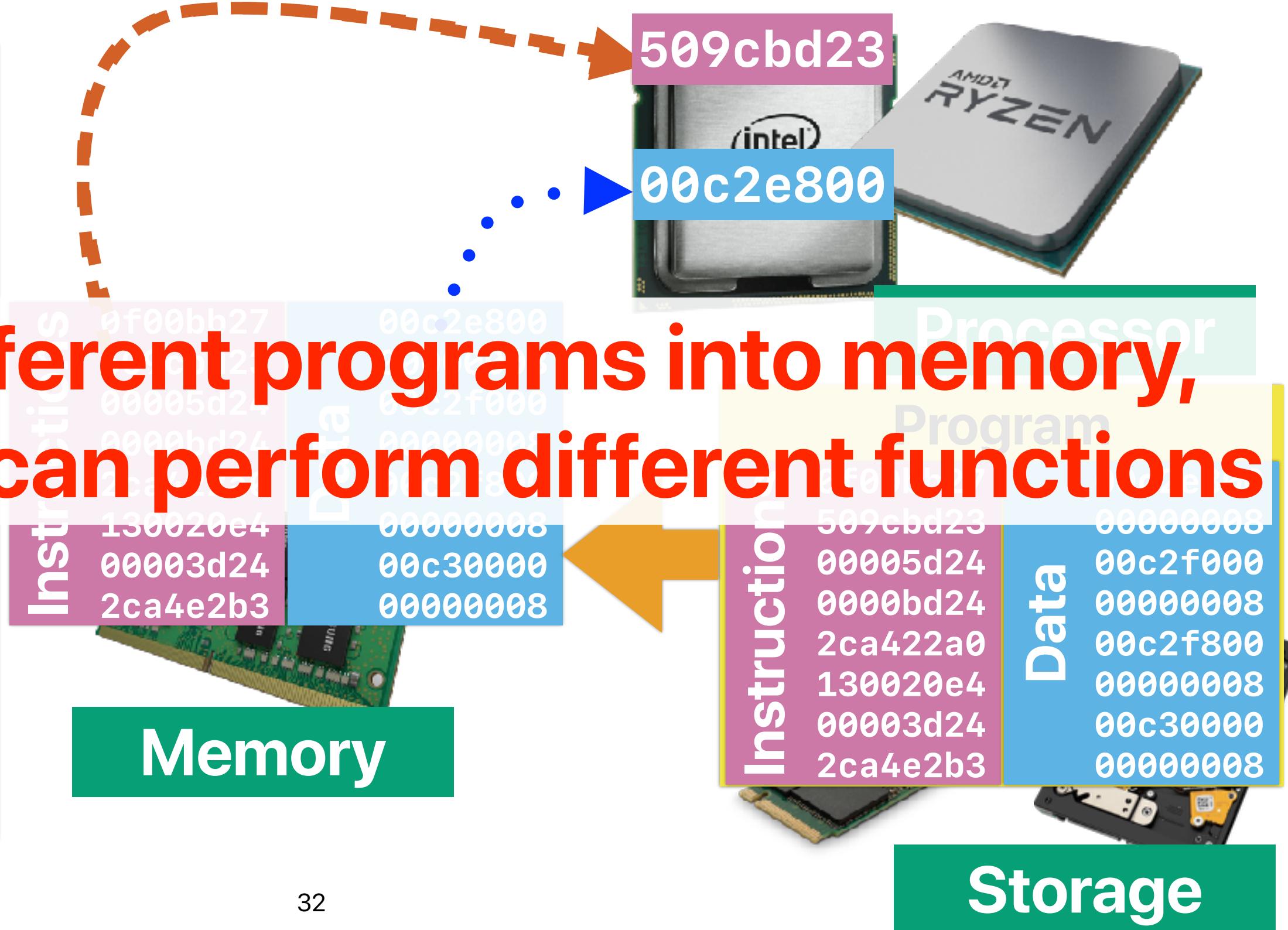


Big Picture: the Von Neumann Architecture

von Neumann Architecture



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

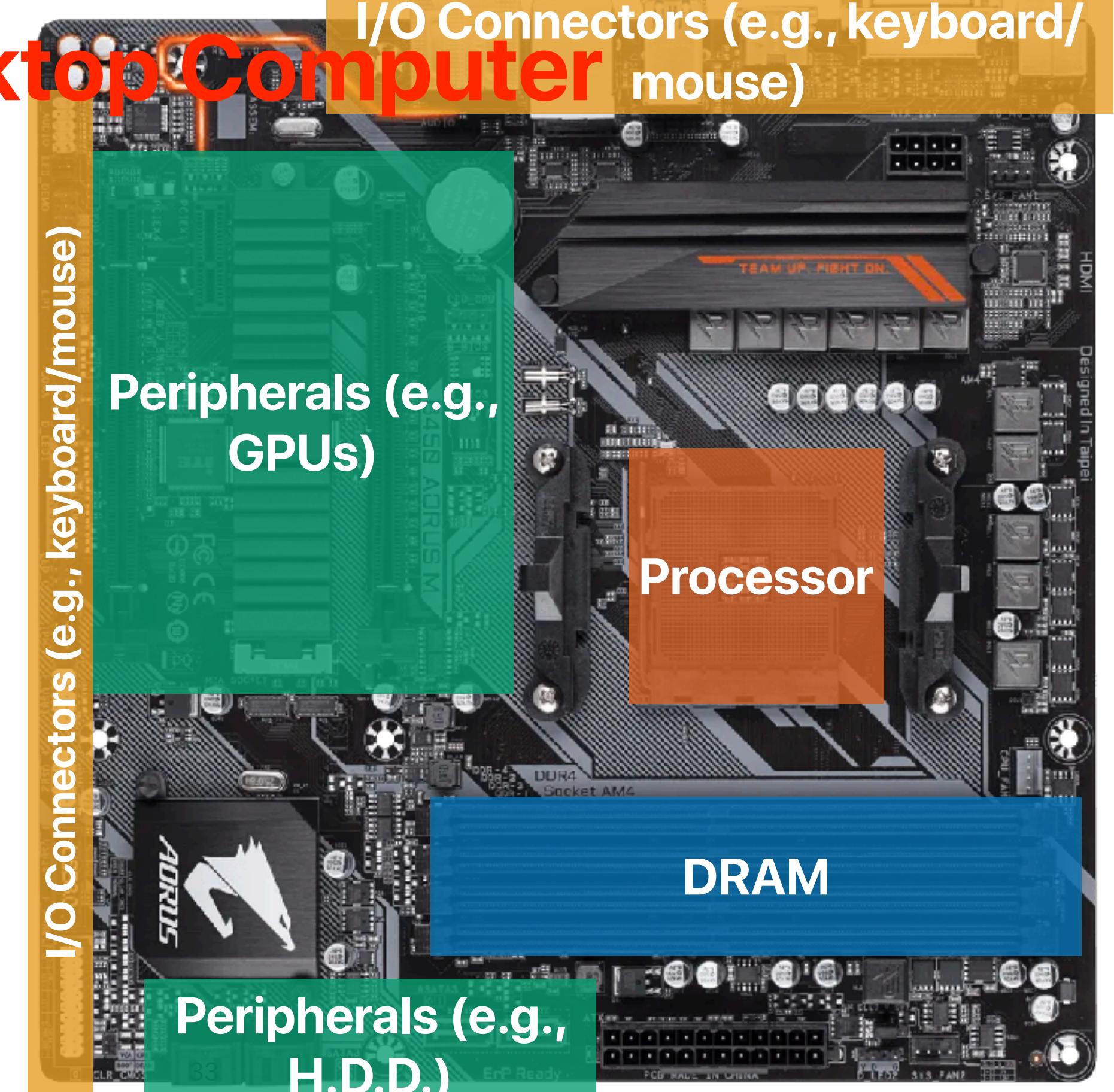


Desktop Computer

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

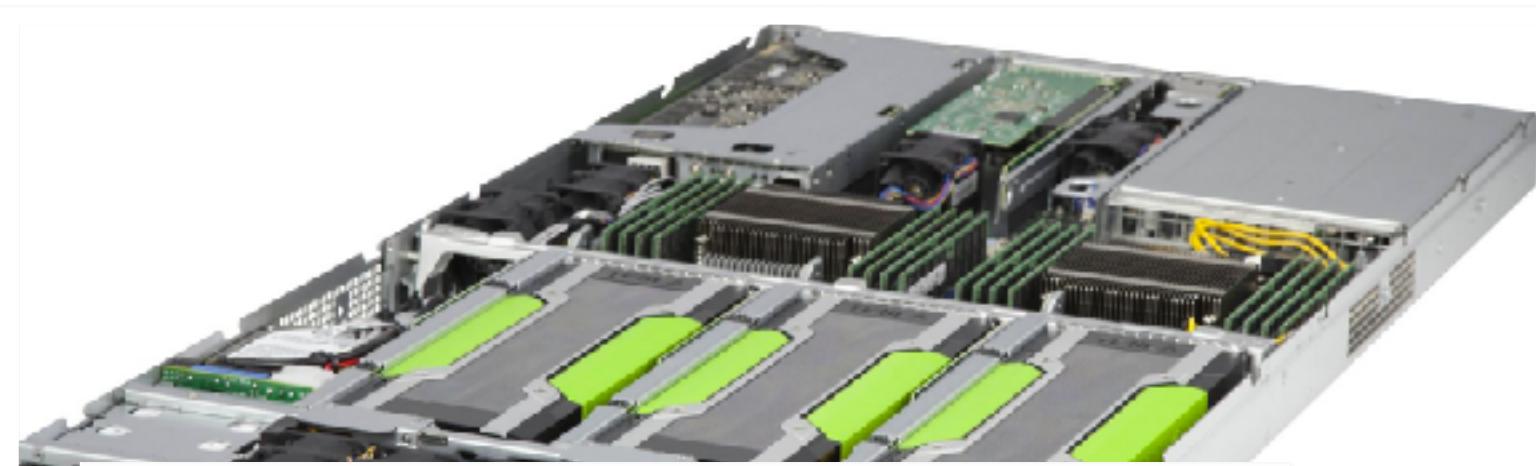


I/O Connectors (e.g., keyboard/mouse)



Server

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



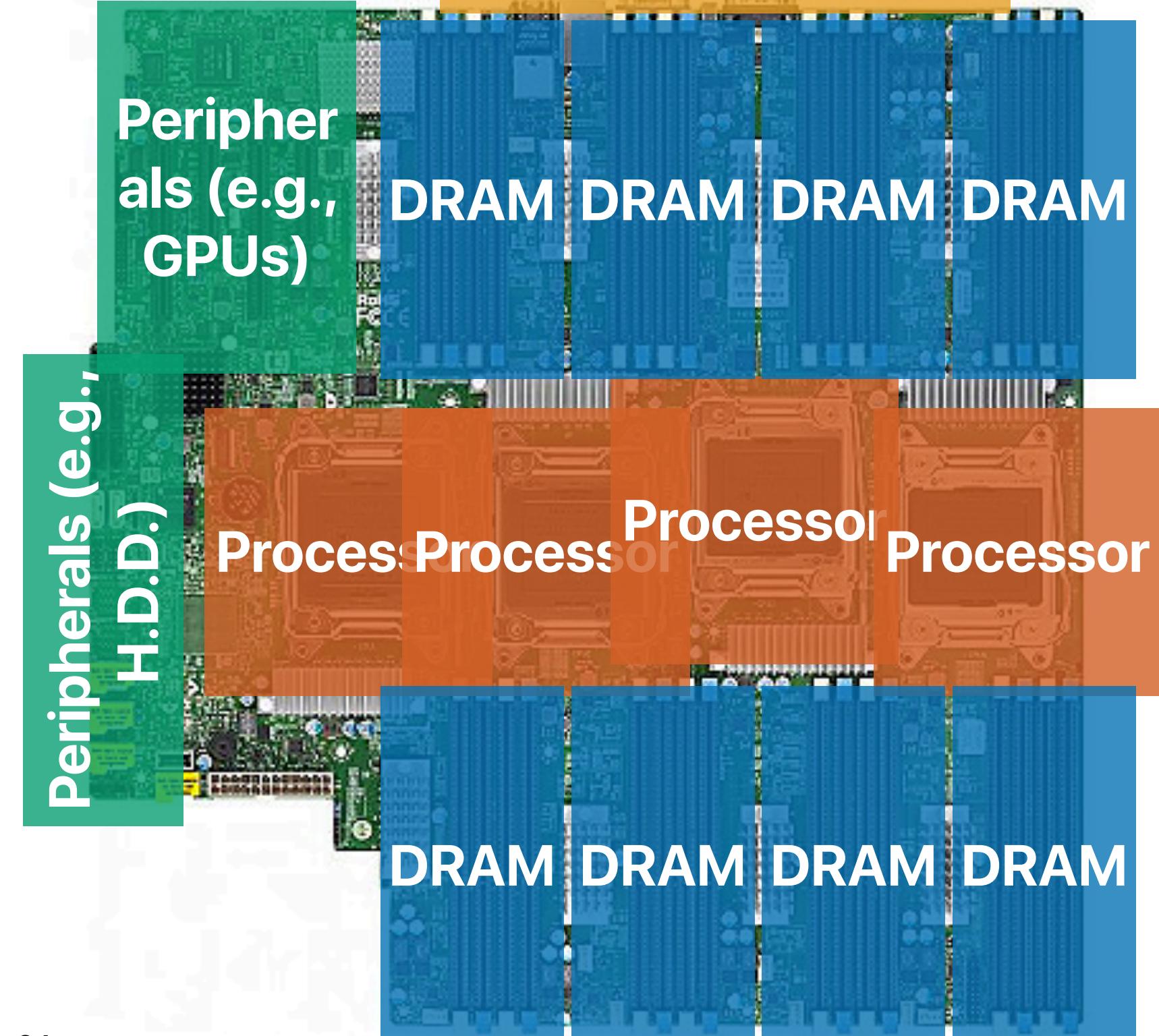
Peripherals (e.g., H.D.D.)

Peripherals (e.g., GPUs)

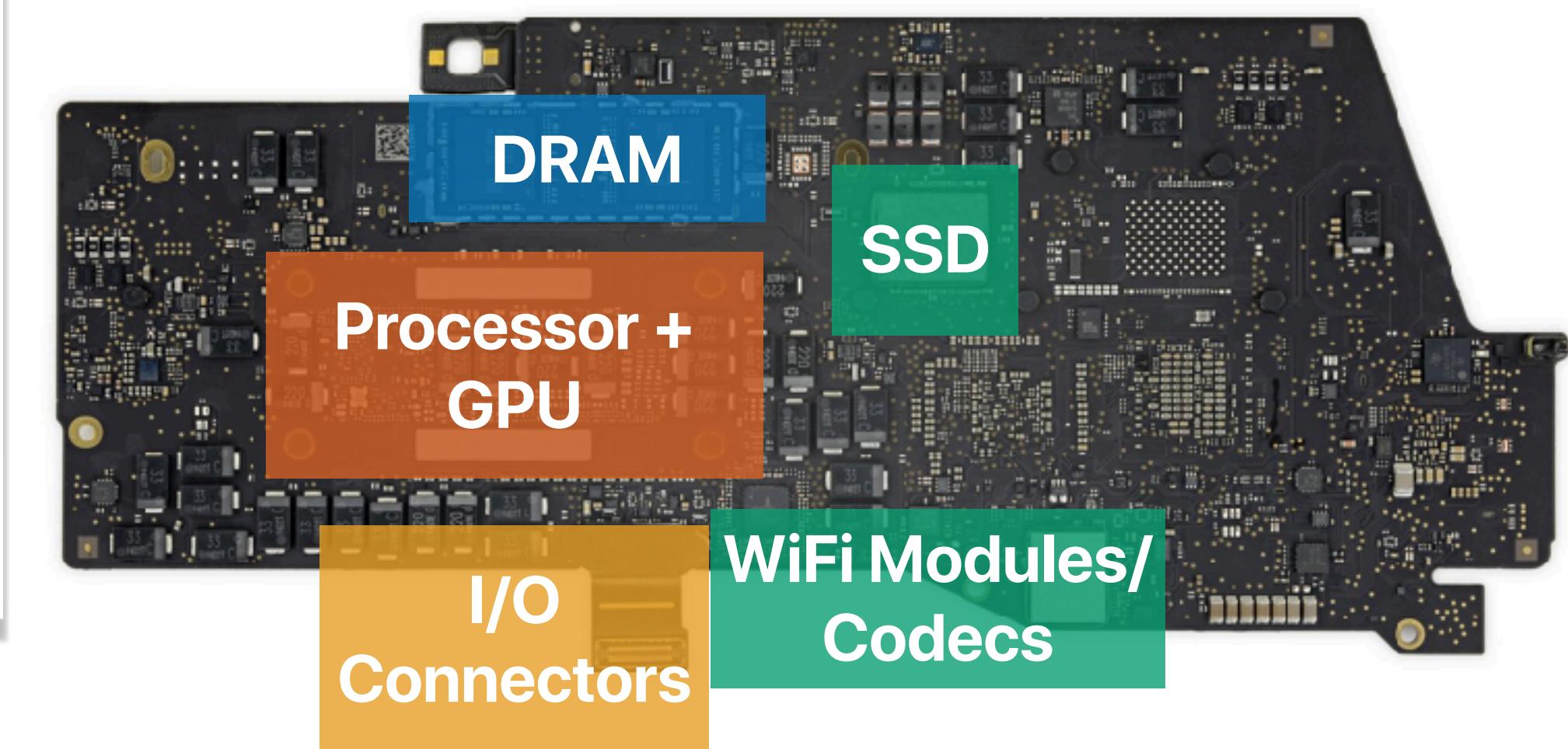
DRAM DRAM DRAM DRAM

Processor Processor Processor Processor

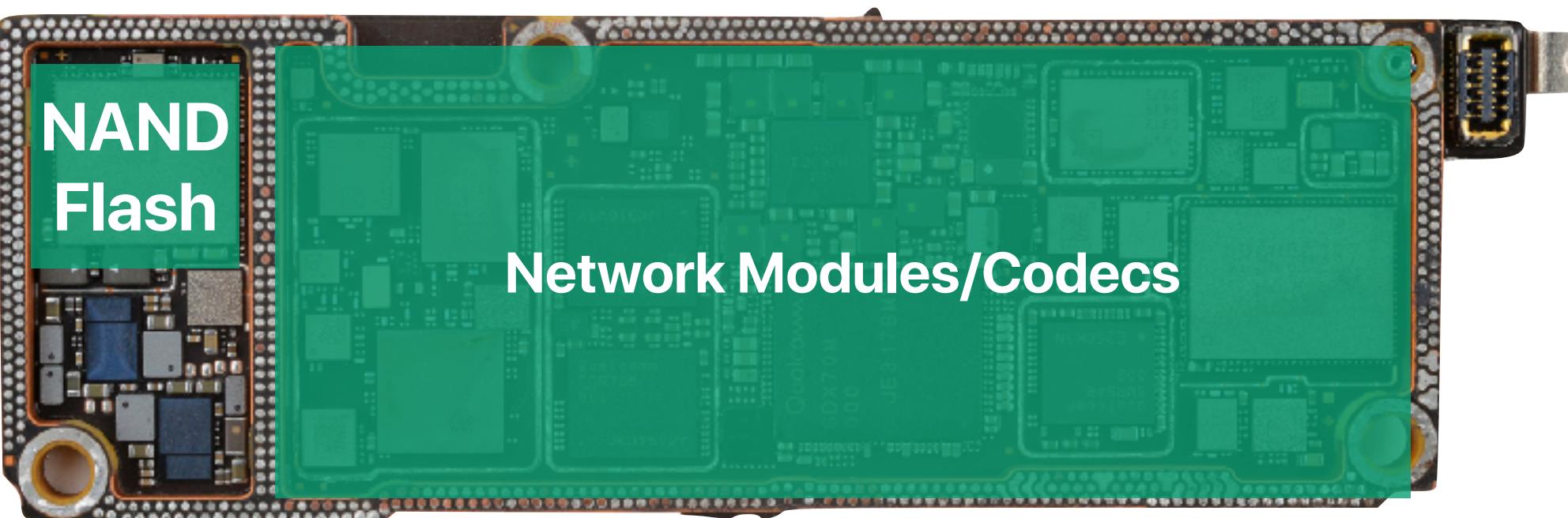
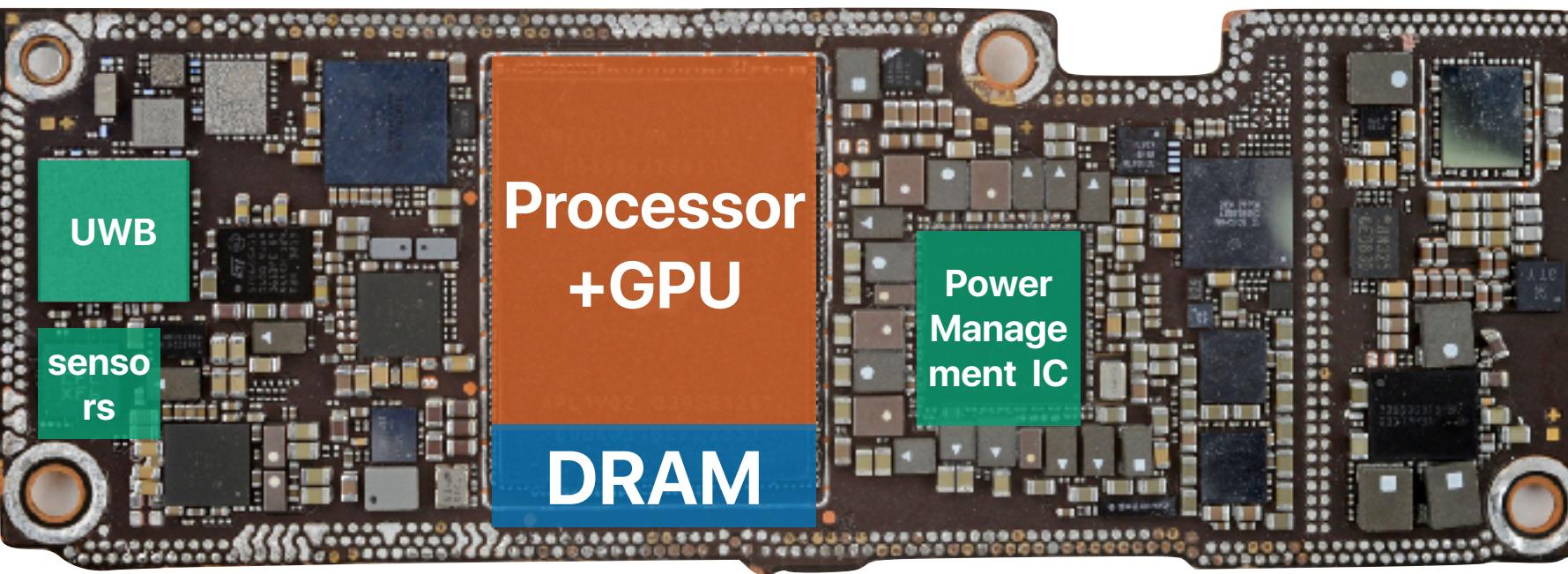
DRAM DRAM DRAM DRAM



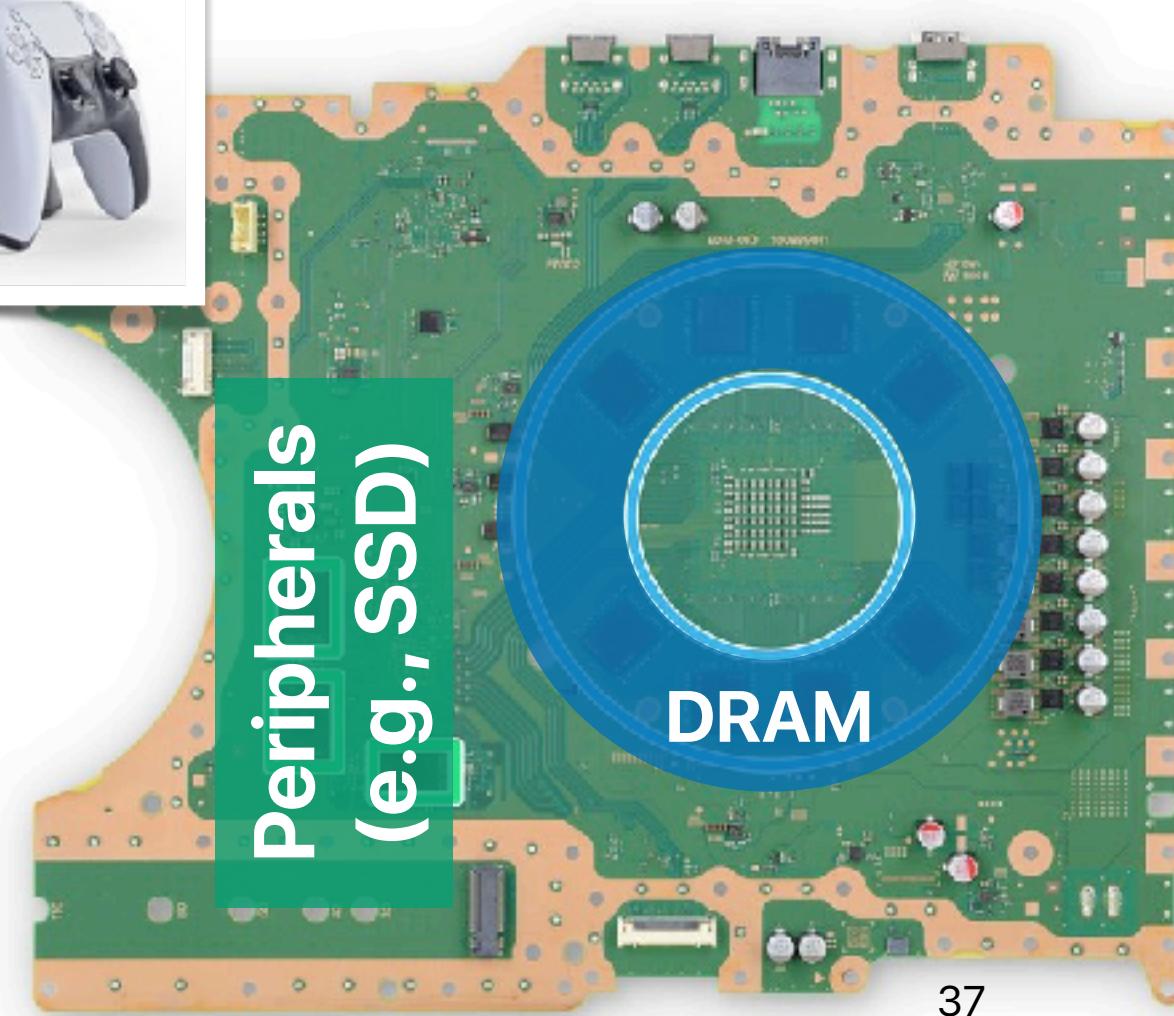
MacBook Pro 13"



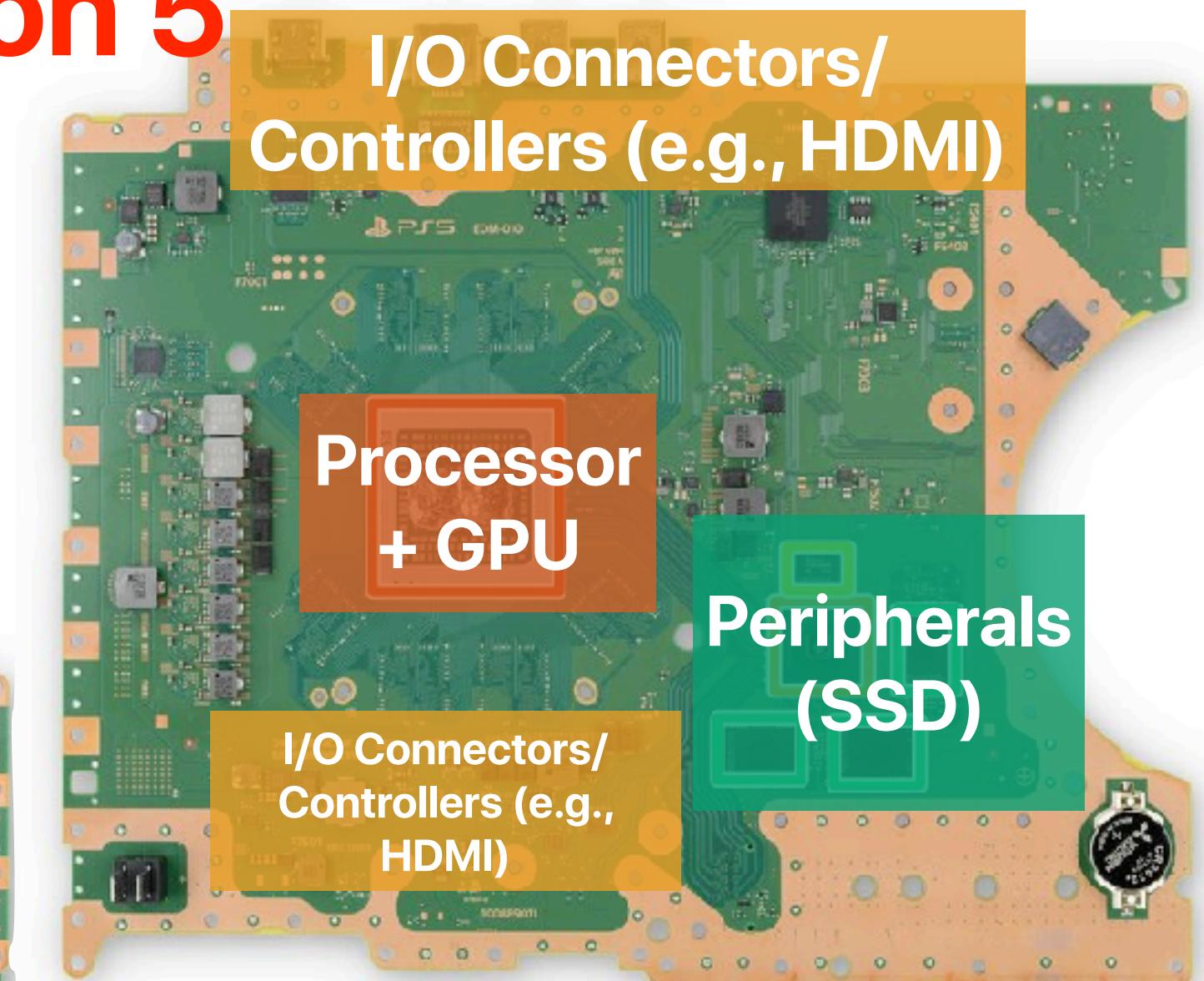
iPhone 15 Pro



Play Station 5



Peripherals
(e.g., SSD)



Processor
+ GPU

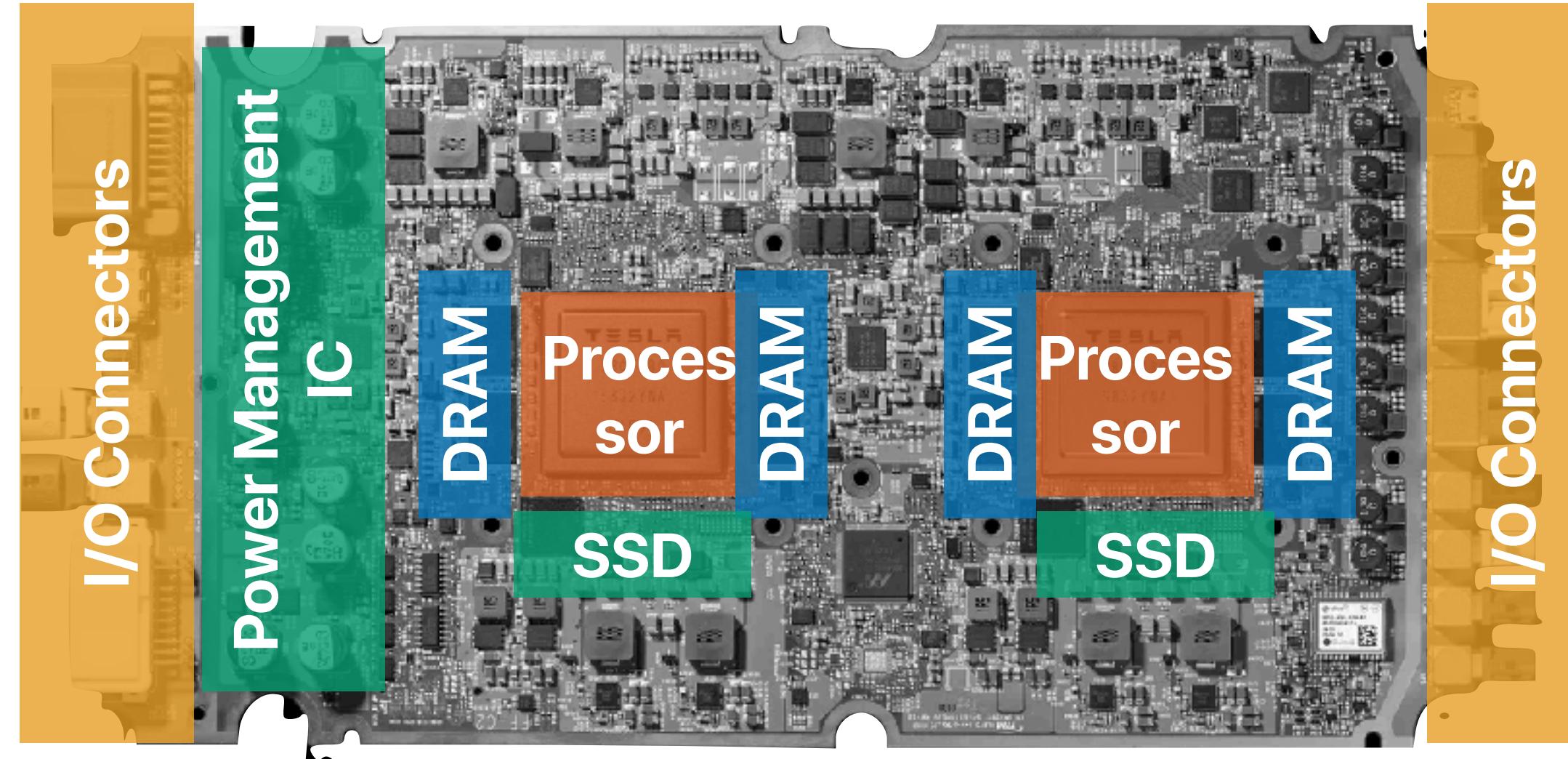
I/O Connectors/
Controllers (e.g.,
HDMI)

Peripherals
(SSD)

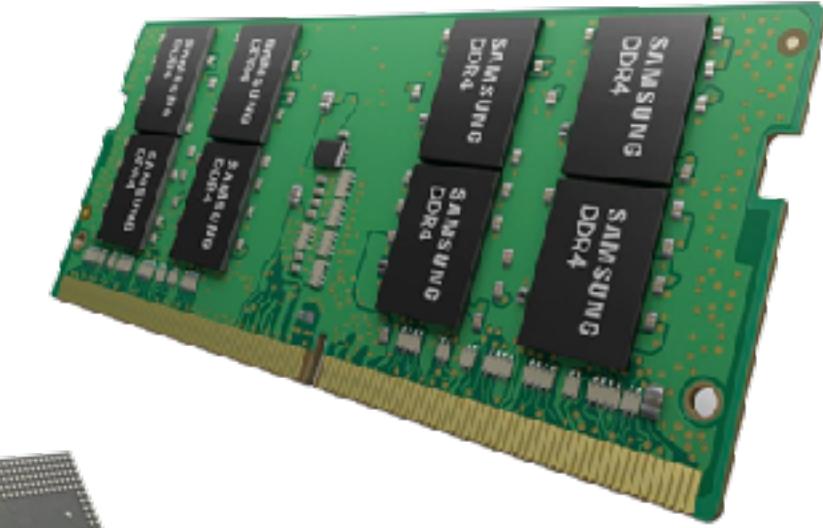
Nintendo Switch



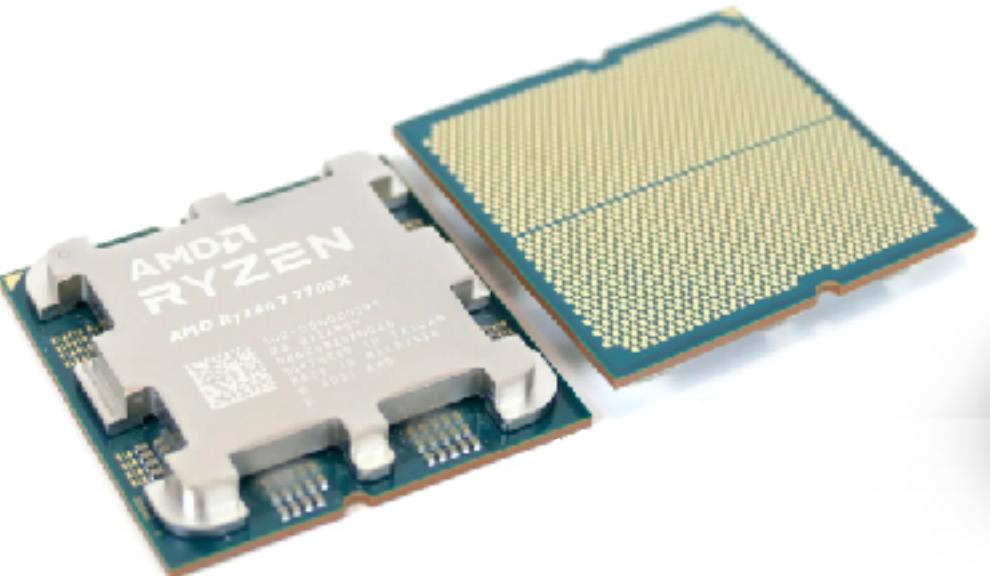
Tesla Model 3



Processors and memory modules are everywhere!



Processors

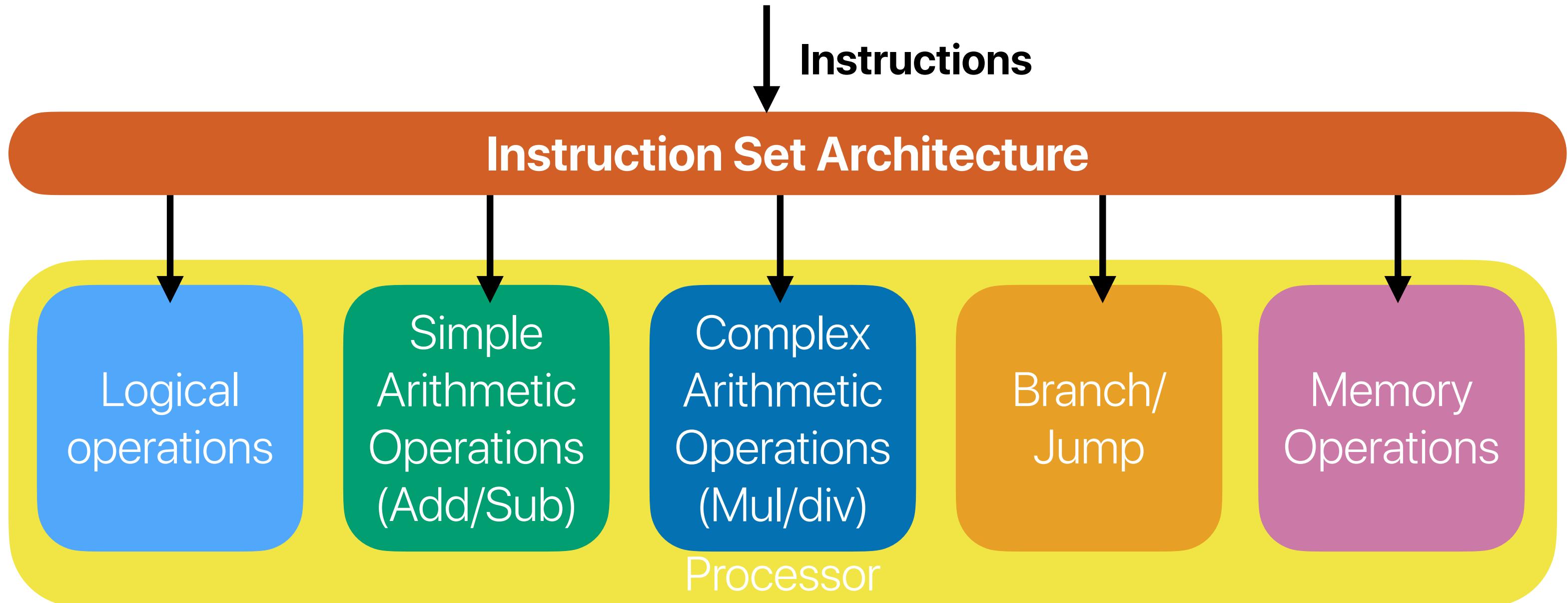


Memory

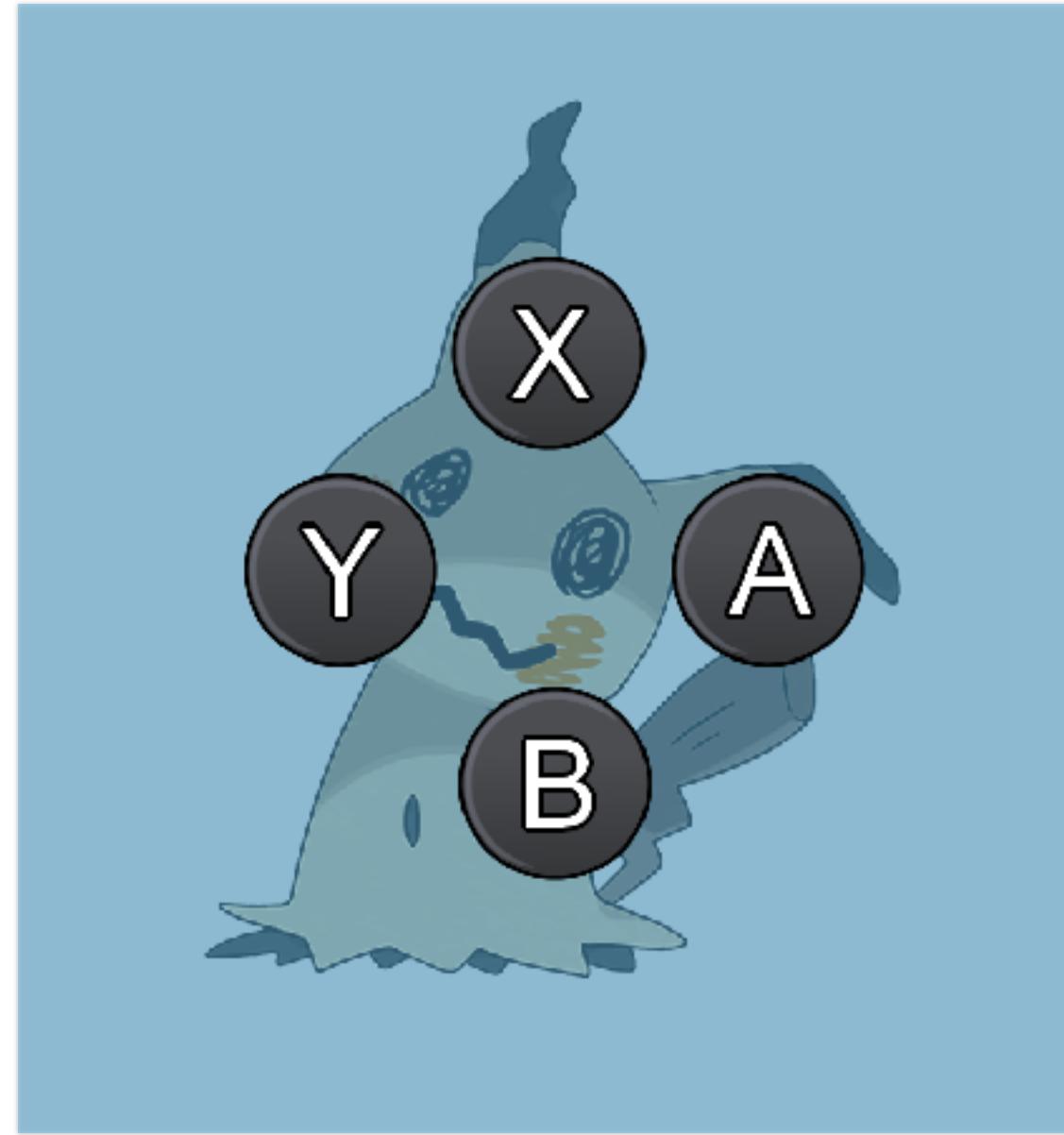
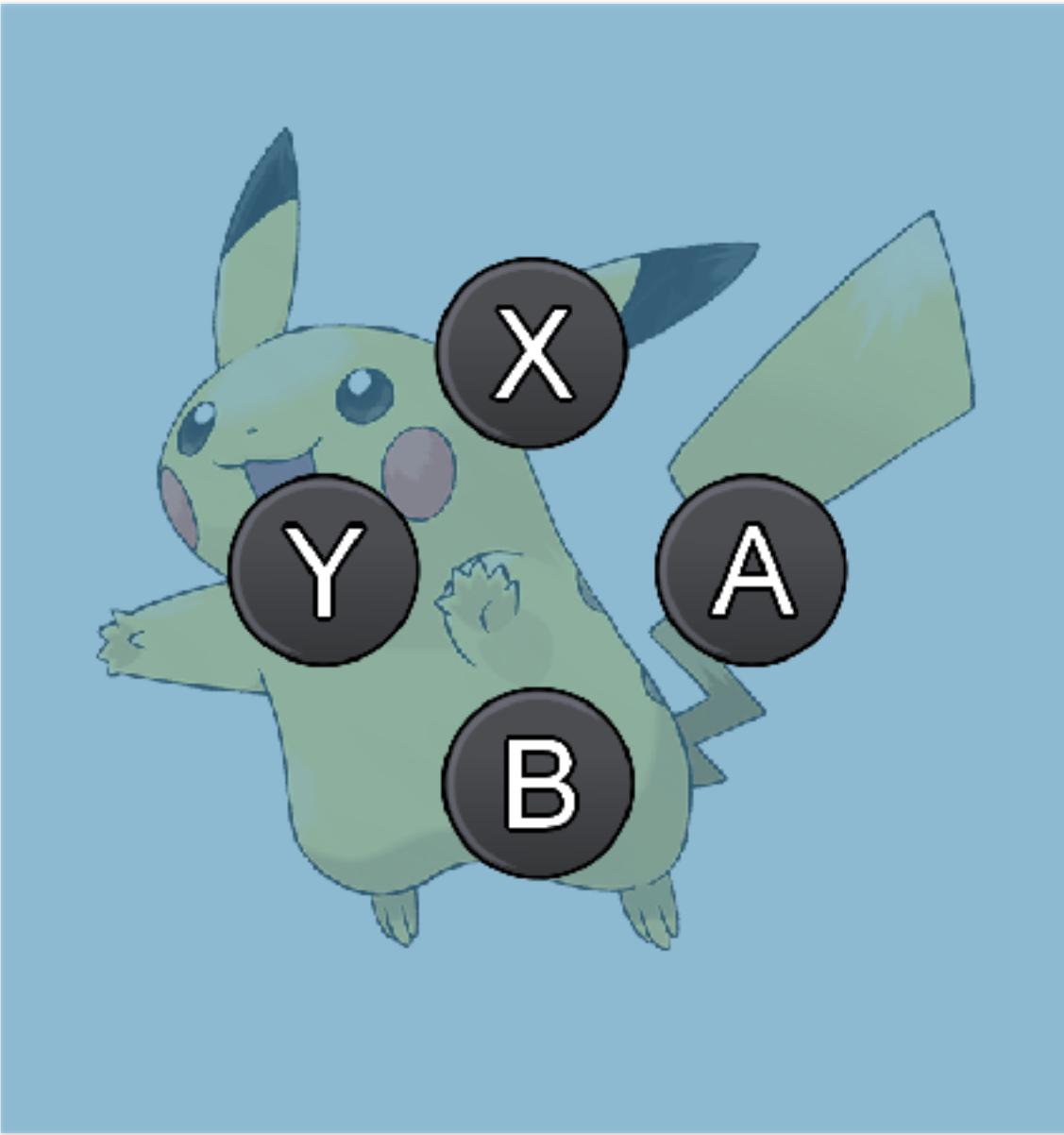
Take-aways: Why CS203?

- Processors/Memories are essential for modern computer systems

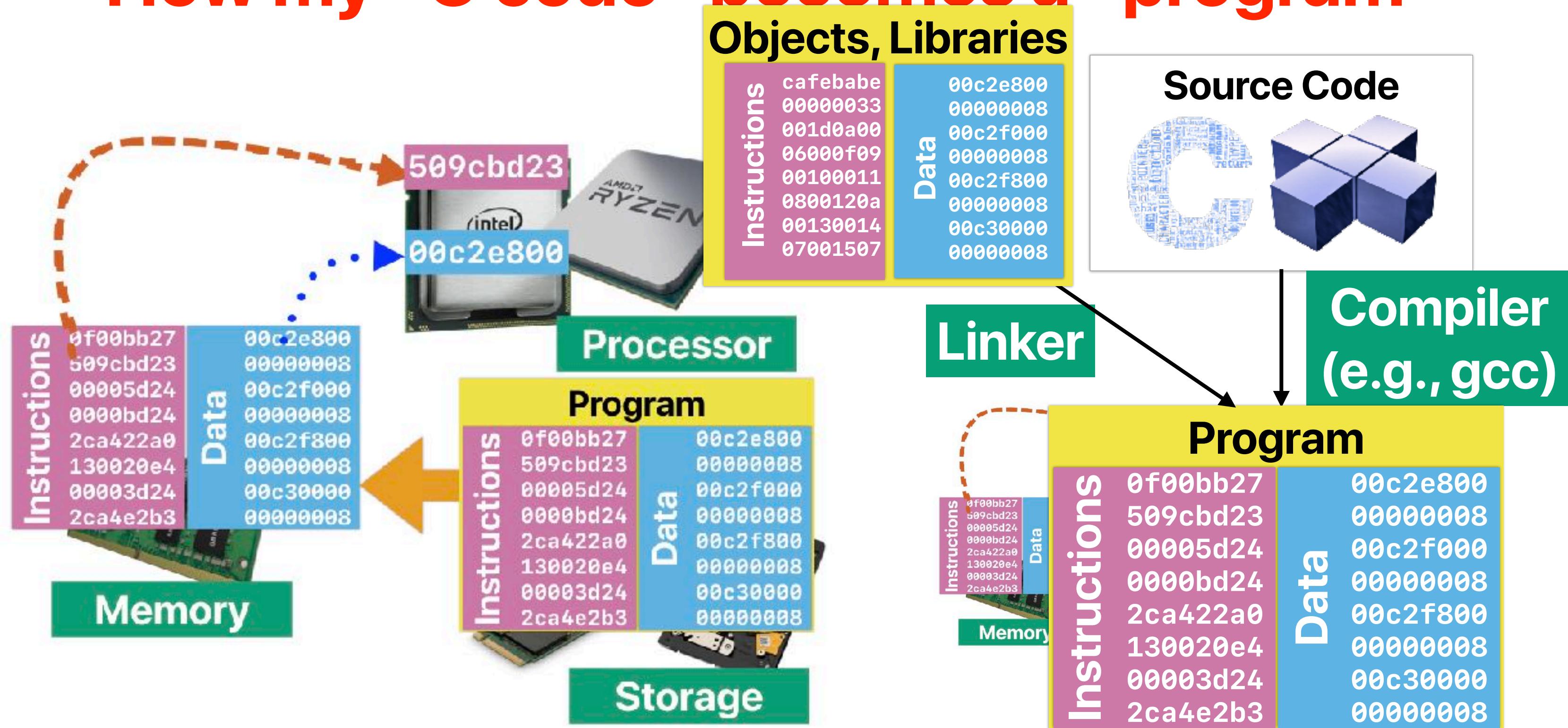
Microprocessor — a collection of functional units



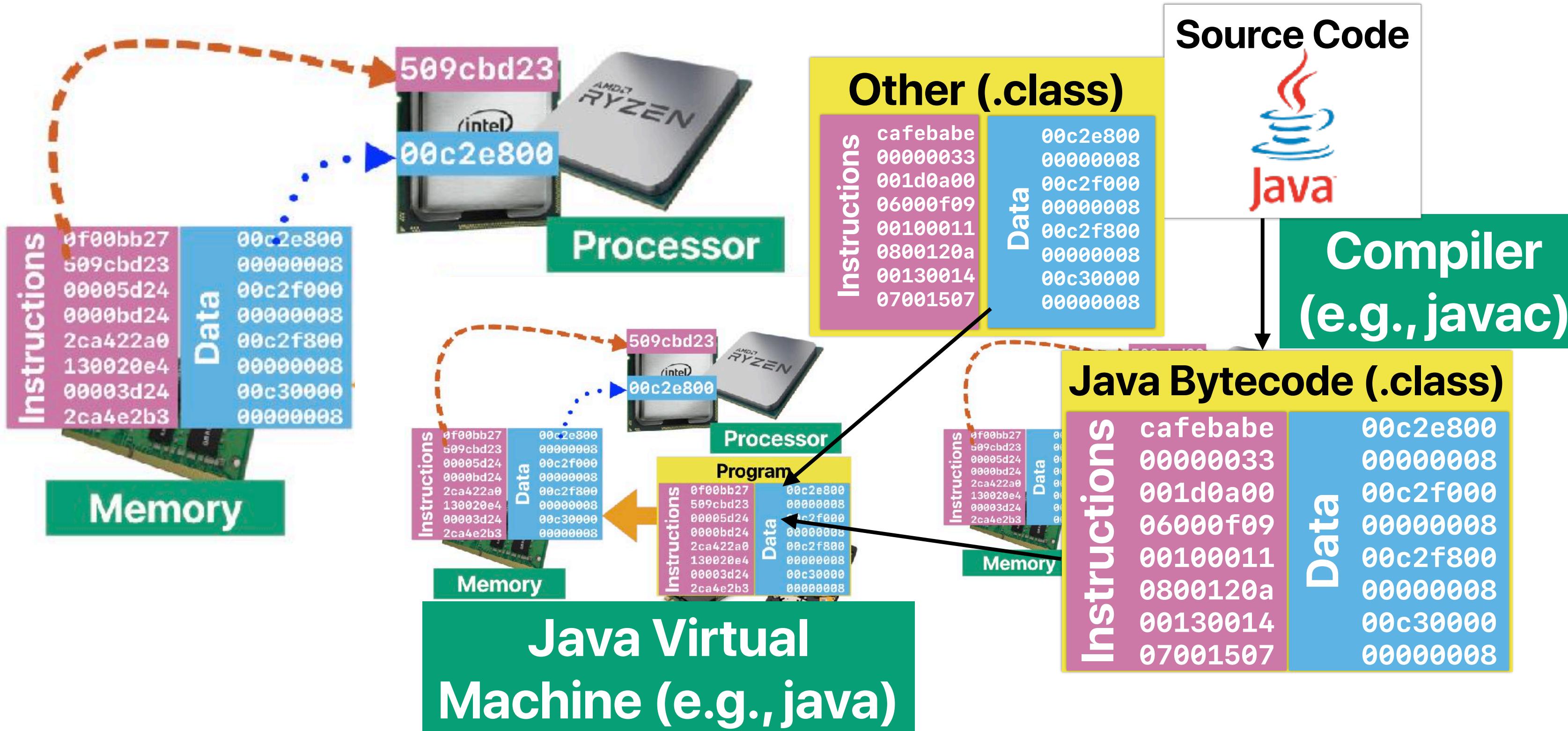
ISA — the “abstraction” of processor features



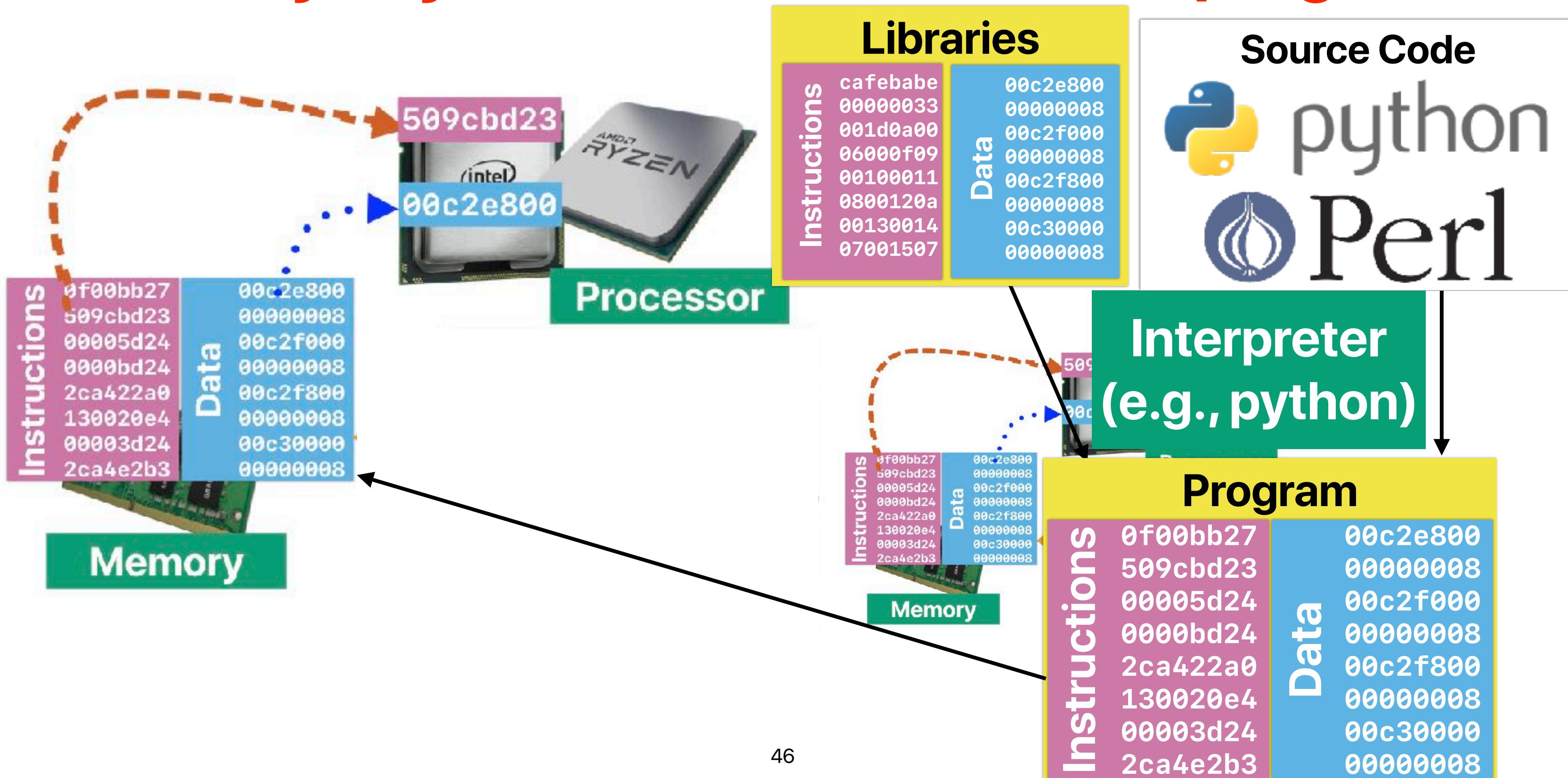
How my “C code” becomes a “program”



How my “Java code” becomes a “program”



How my “Python code” becomes a “program”



The program = Instructions + Data

You may use “objdump” to see the content of a program!

```
simple.o:      file format elf64-x86-64
```

Contents of section .text:

```
0000 f30f1efa 554889e5 c745f800 000000c7  
0010 45fc0000 0000c745 f8000000 00eb1e8b  
0020 45f84898 488d145f 00000000 488d0500  
0030 0000008b 04020145 fc8345f8 01837df8  
0040 137edcb8 00000000 5dc3
```

Instructions

```
....UH...E.....  
E.....E.....  
E.H.H.....H...  
.....E..E...}.  
.~.....].
```

Contents of section .data:

```
0000 01000000 02000000 03000000 04000000  
0010 05000000 06000000 07000000 08000000  
0020 09000000 0a000000 01000000 02000000  
0030 03000000 04000000 05000000 06000000  
0040 07000000 08000000 09000000 0a000000
```

```
.....  
.....  
.....  
.....  
.....
```

Contents of section .comment:

```
0000 00474343 3a202855 62756e74 7520392e  
0010 342e302d 31756275 6e747531 7e32302e  
0020 30342e31 2920392Data2e3000
```

.GCC: (Ubuntu 9.
4.0-1ubuntu1~20.
04.1) 9.4.0.

Contents of section .note.gnu.property:

```
0000 04000000 10000000 05000000 474e5500  
0010 020000c0 04000000 03000000 00000000
```

.....GNU.
.....

Contents of section .eh_frame:

```
0000 14000000 00000000 017a5200 01781001  
0010 1b0c0708 90010000 1c000000 1c000000  
0020 00000000 4a000000 00450e10 8602430d  
0030 0602410c 07080000
```

.....zR..x..
.....
....J....E....C.
.A.....

Most of time, I don't program at this level...

```
simple.o:      file format elf64-x86-64
```

Contents of section .text:

0000	f30f1efa	554889e5	c745f800	000000c7UH...E.....
0010	45fc0000	0000c745	f8000000	00eb1e8b	E.....E.....
0020	45f84898	488d1485	00000000	488d0500	E.H.H.....H...
0030	0000008b	04020145	fc8345f8	01837df8E..E...}.
0040	137edcb8	00000000	5dc3		.~.....].

Contents of section .data:

0000	01000000	02000000	03000000	04000000
0010	05000000	06000000	07000000	08000000
0020	09000000	0a000000	01000000	02000000
0030	03000000	04000000	05000000	06000000
0040	07000000	08000000	09000000	0a000000

Contents of section .comment:

0000	00474343	3a202855	62756e74	7520392e	.GCC: (Ubuntu 9.
0010	342e302d	31756275	6e747531	7e32302e	4.0-1ubuntu1~20.
0020	30342e31	2920392e	342e3000		04.1) 9.4.0.

Contents of section .note.gnu.property:

0000	04000000	10000000	05000000	474e5500GNU.
0010	020000c0	04000000	03000000	00000000

Contents of section .eh_frame:

0000	14000000	00000000	017a5200	01781001zR..x..
0010	1b0c0708	90010000	1c000000	1c000000
0020	00000000	4a000000	00450e10	8602430dJ....E....C.
0030	0602410c	07080000			..A.....

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;  
}
```

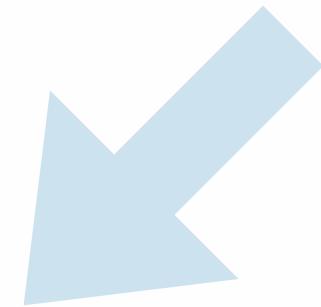
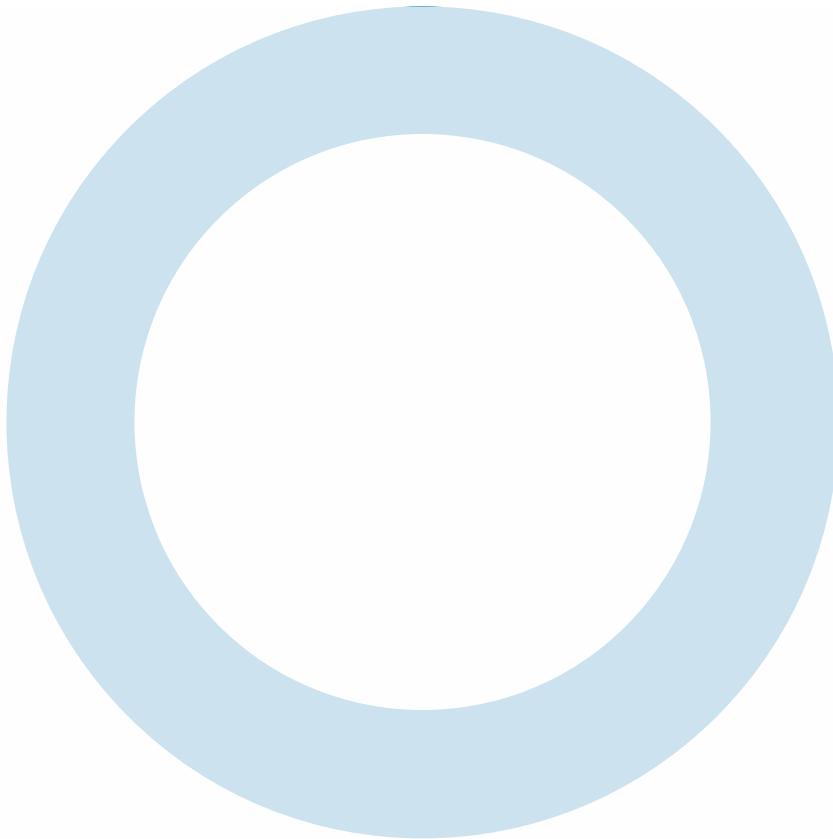
memory
access
arithmetic
operations

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

On the same hardware, programs with lower computational complexities are more efficient



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 hardware, program performance scales with its main algorithm complexity



Matrix Multiplication

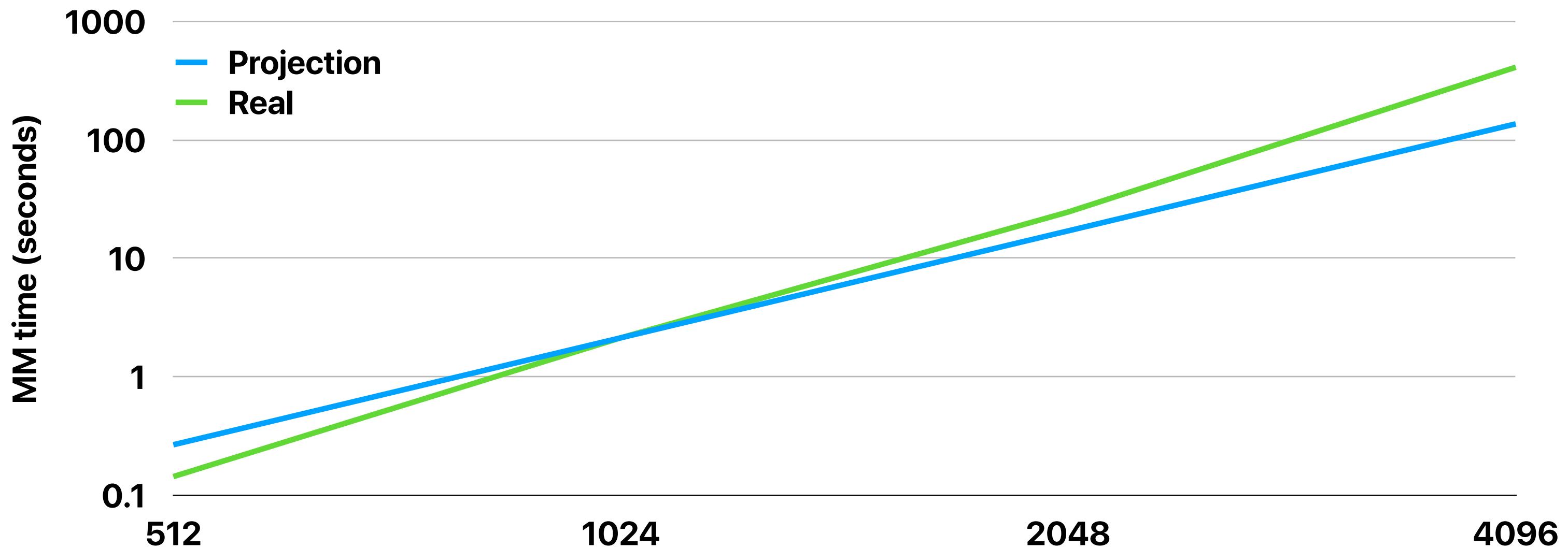
```
for(i = 0; i < ARRAY_SIZE; i++) {  
    for(j = 0; j < ARRAY_SIZE; j++) {  
        for(k = 0; k < ARRAY_SIZE; k++) {  
            c[i][j] += a[i][k]*b[k][j];  
        }  
    }  
}
```

Algorithm class tells you it's $O(n^3)$

If $n=1024$, it takes about 2 sec

How long is it take when $n=2048$?

Your projection v.s. the reality

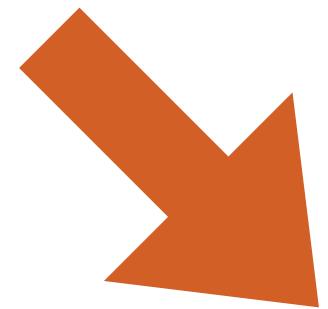
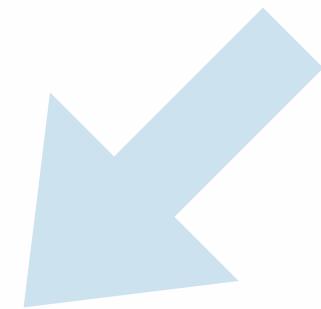
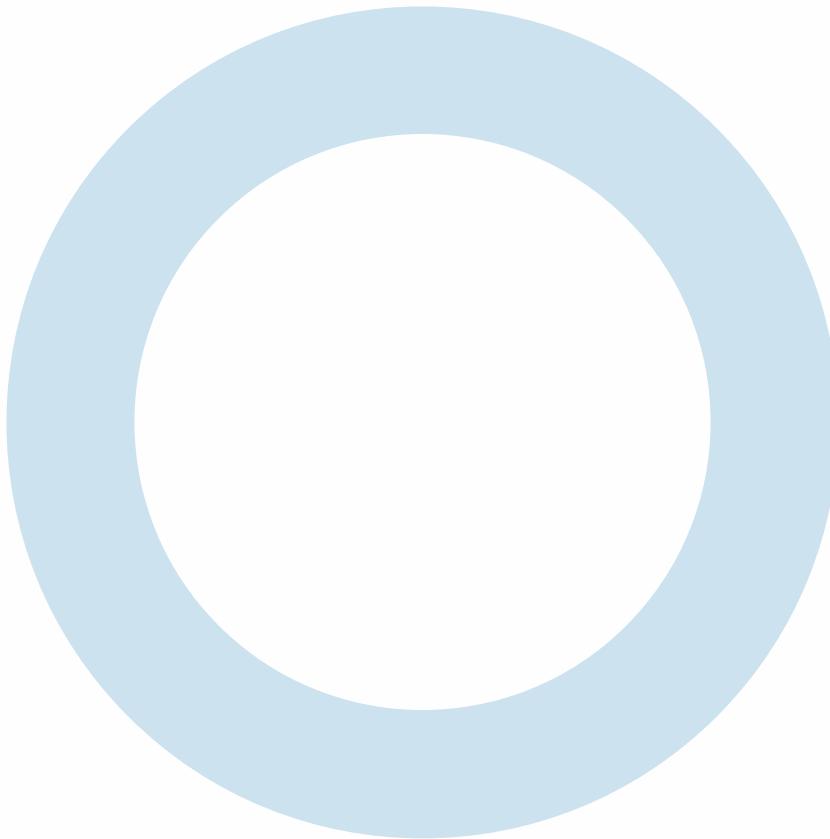


<https://ourworldindata.org/grapher/artificial-intelligence-training-computation>

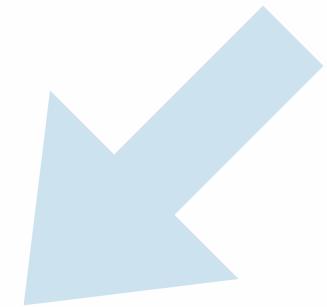
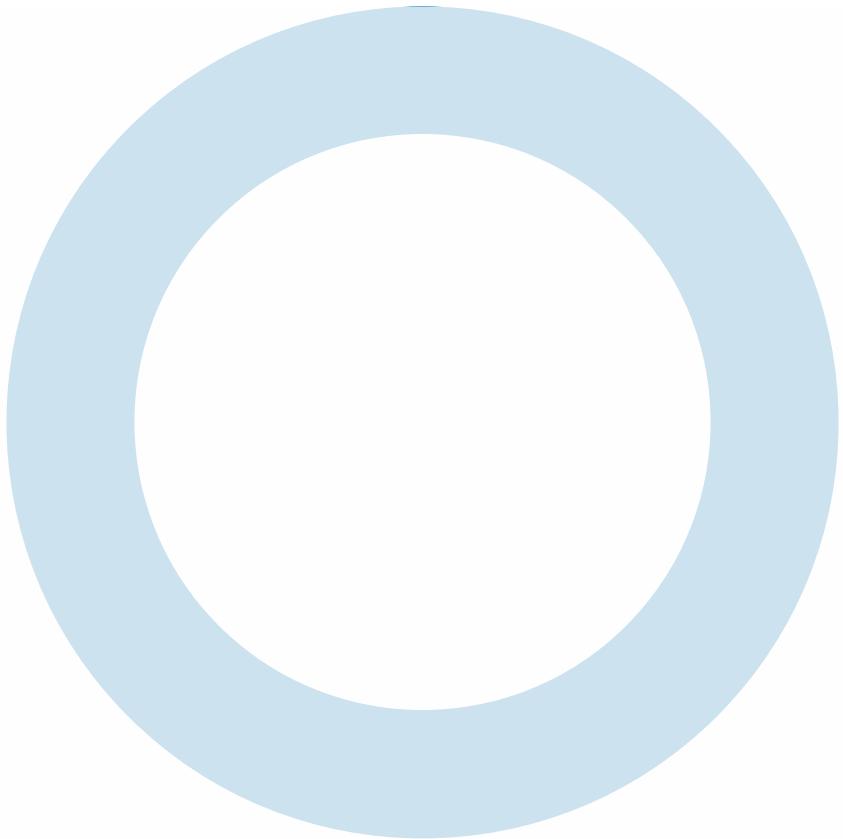
Take-aways: Why CS203?

- Processors/Memories are essential for modern computer systems
- Algorithm complexity does not work well on “real” computers

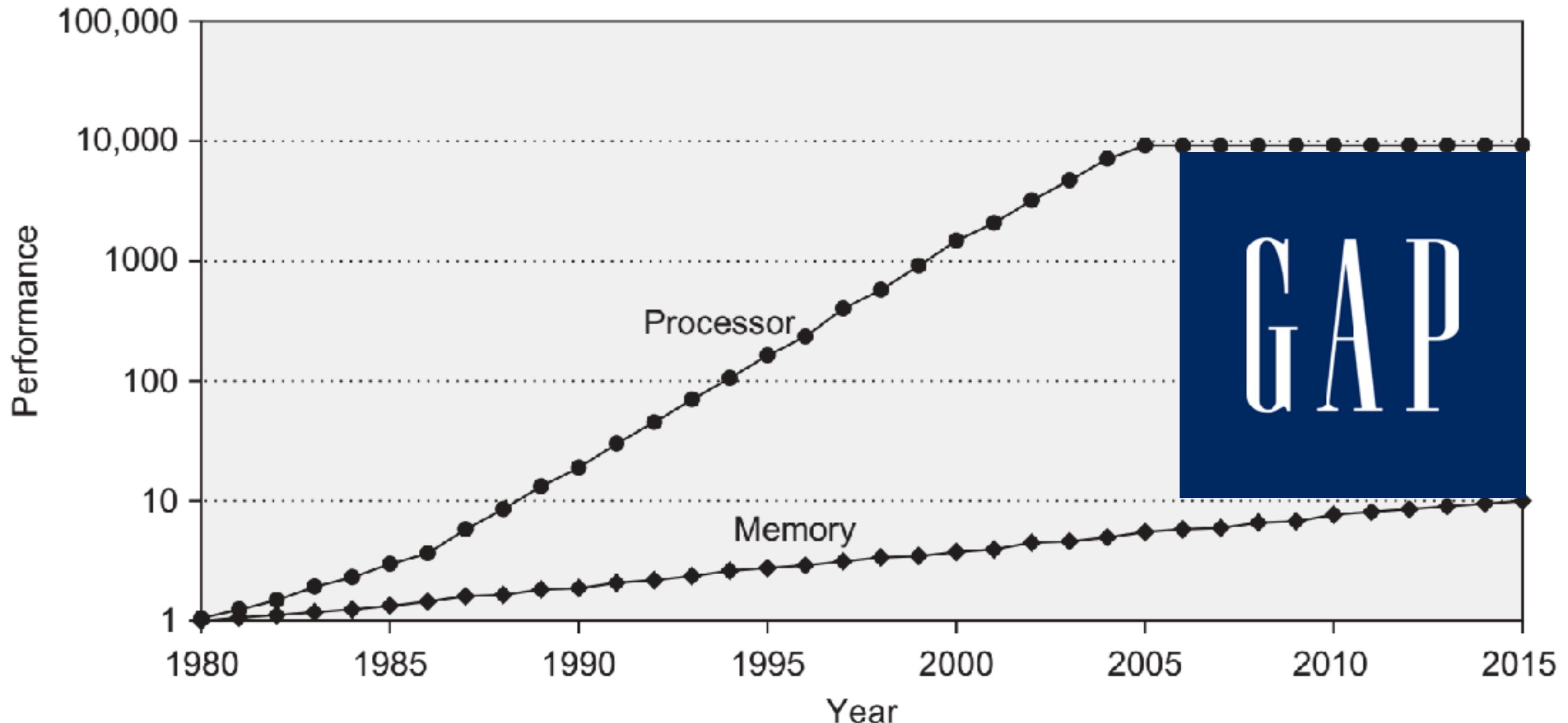
Leveraging more “bit-wise” operations in C code will make the program significantly faster



On the same CPU, each instruction typically takes the same orders of magnitude of execution time



Performance gap between Processor/Memory



Take-aways: Why CS203?

- Processors/Memories are essential for modern computer systems
- Algorithm complexity does not work well on “real” computers
- Memory operations are more expensive and unpredictable than others

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 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

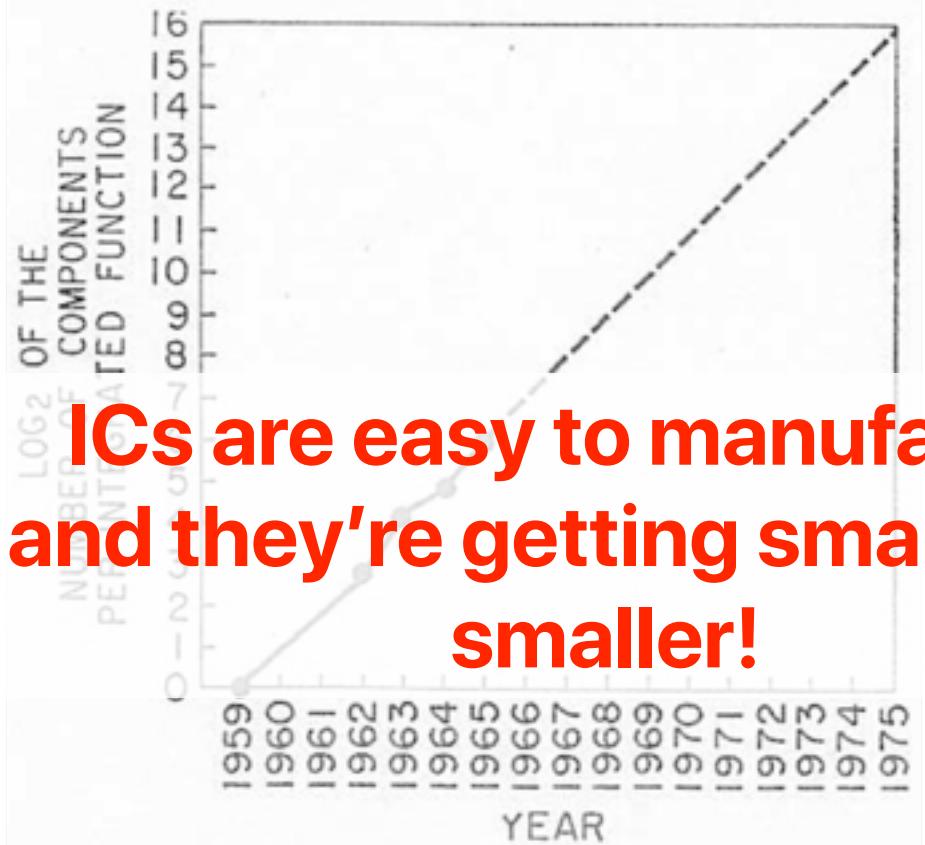
ICs are small

(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

inductors makes it difficult to implement

linear circuitry in the integrated form.

In almost every case, however, the demonstrated 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 widely applicable

Establish

Reliability count

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.

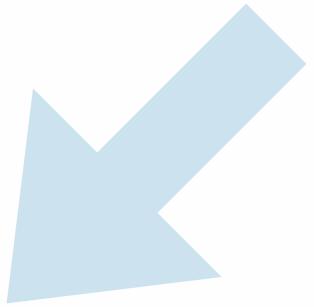
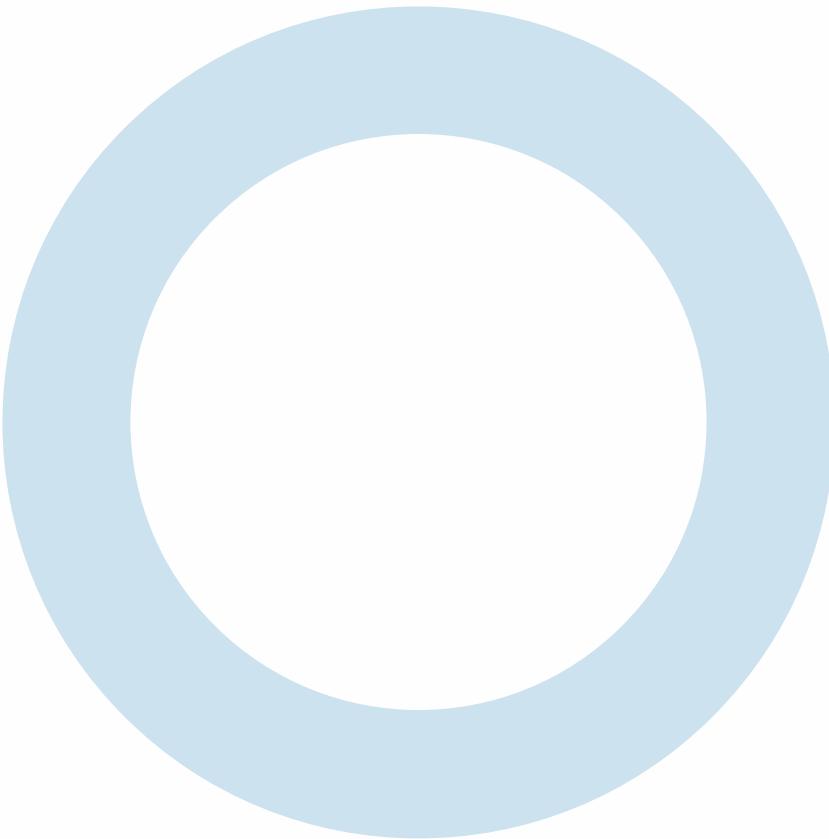
Designing ICs can be easy

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

Moore's Law⁽¹⁾

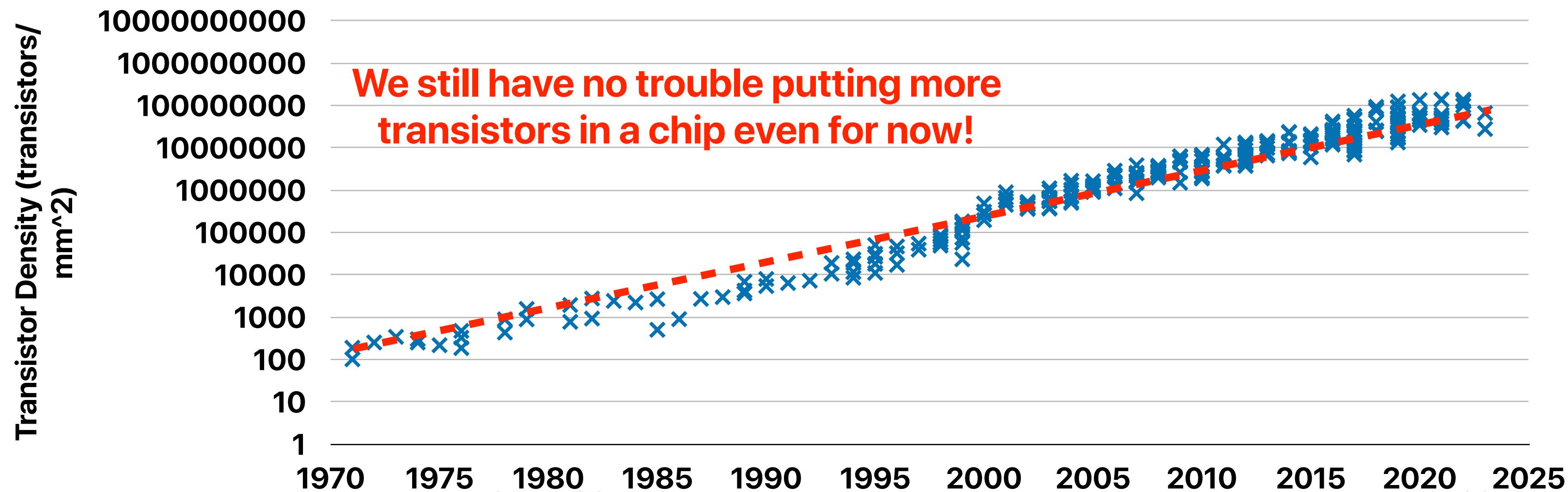
- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.

Moore's Law is current slowing/discontinuing



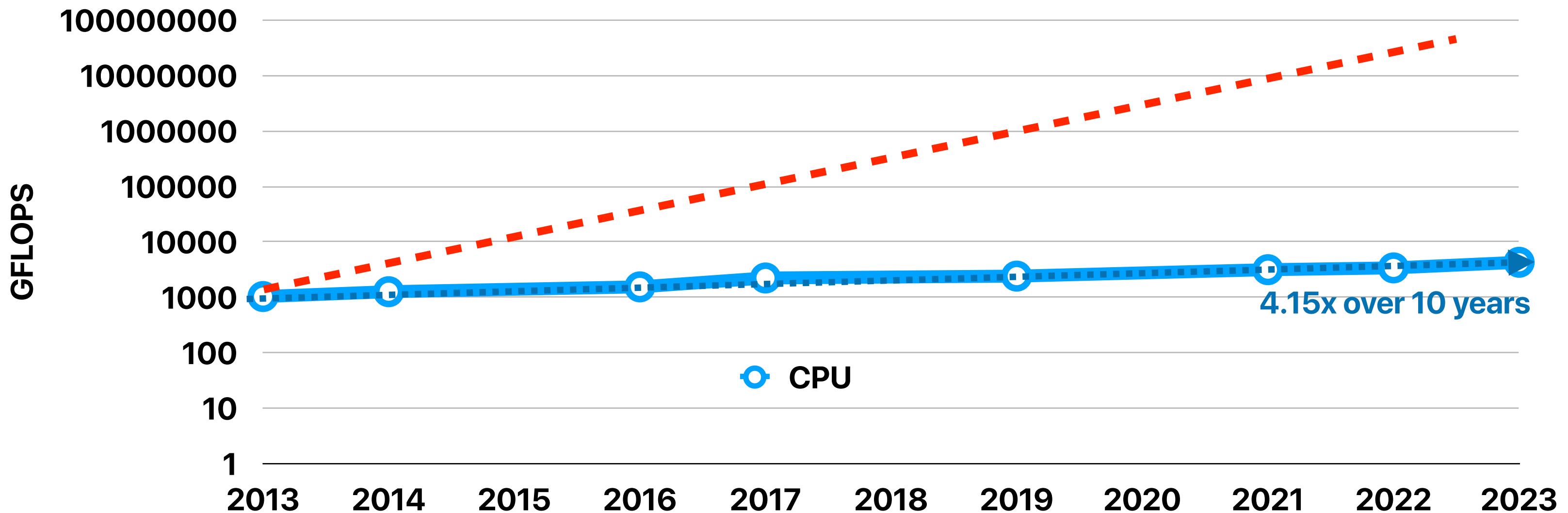
Moore's Law⁽¹⁾

- 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



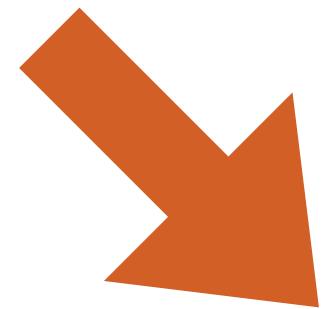
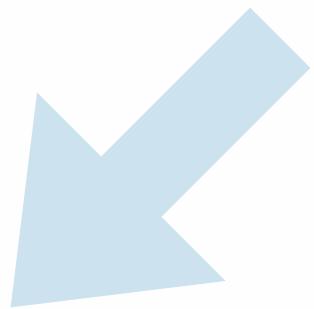
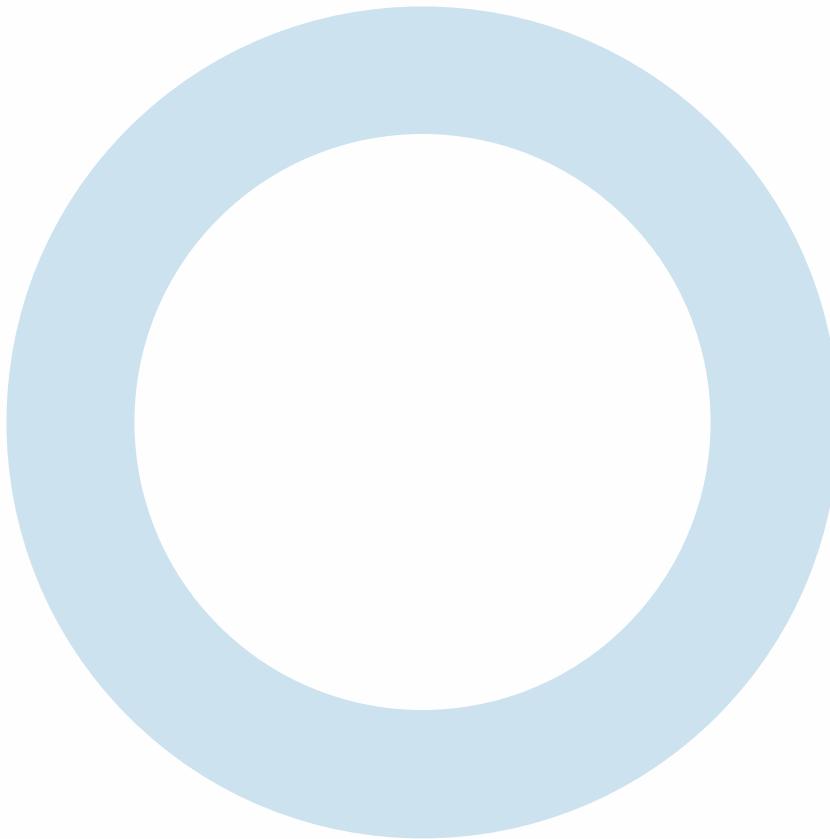
(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8).

CPU Performance v.s. Moore's Law



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

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



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

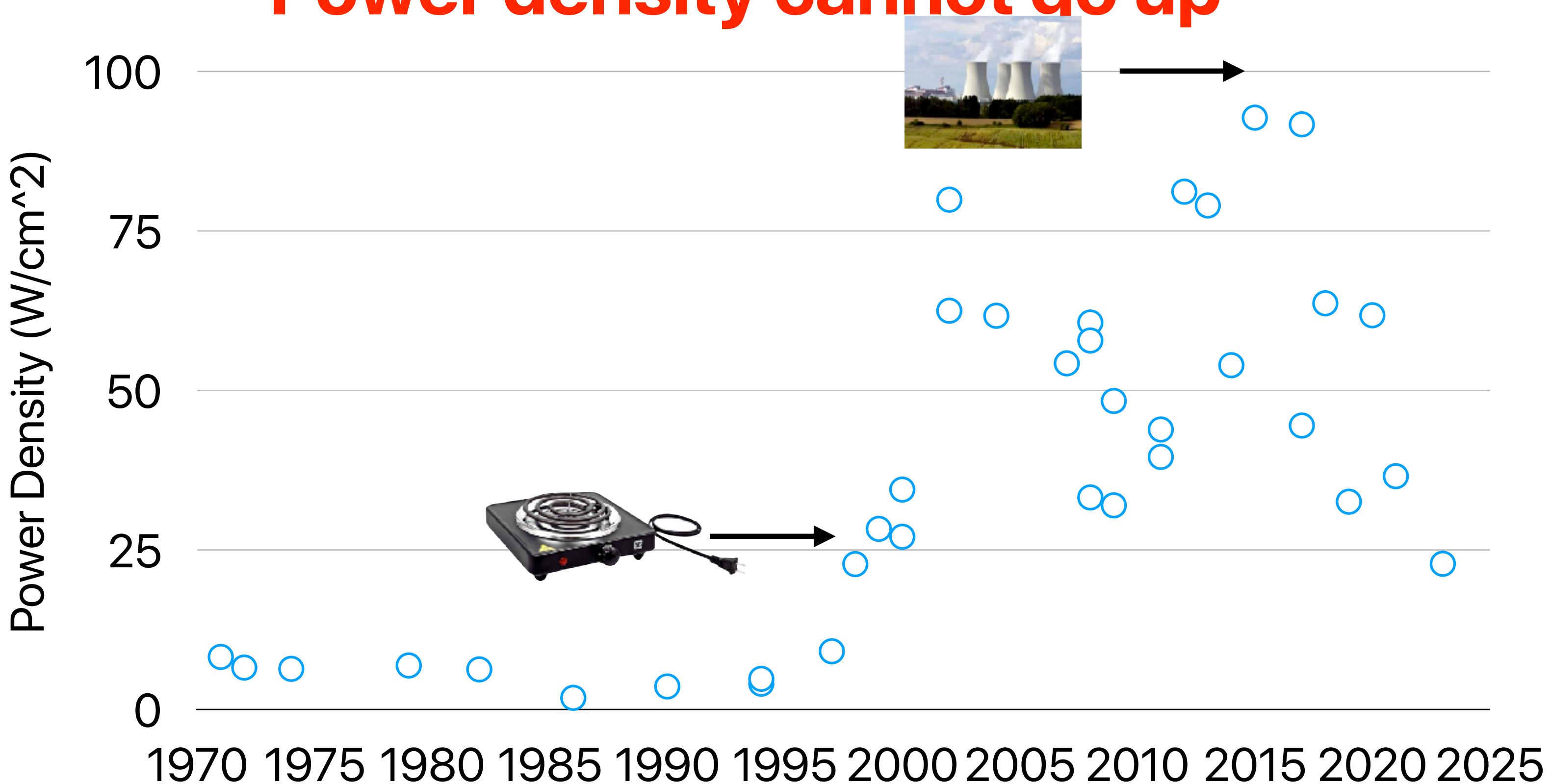
[Intel "Emerald Rapids" Xeon SPs: A Little More Bang, A Little ...](#)

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

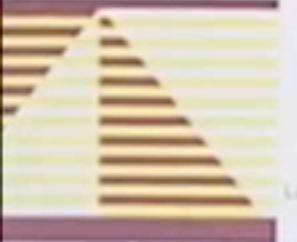
Intel 7nm

Power density cannot go up



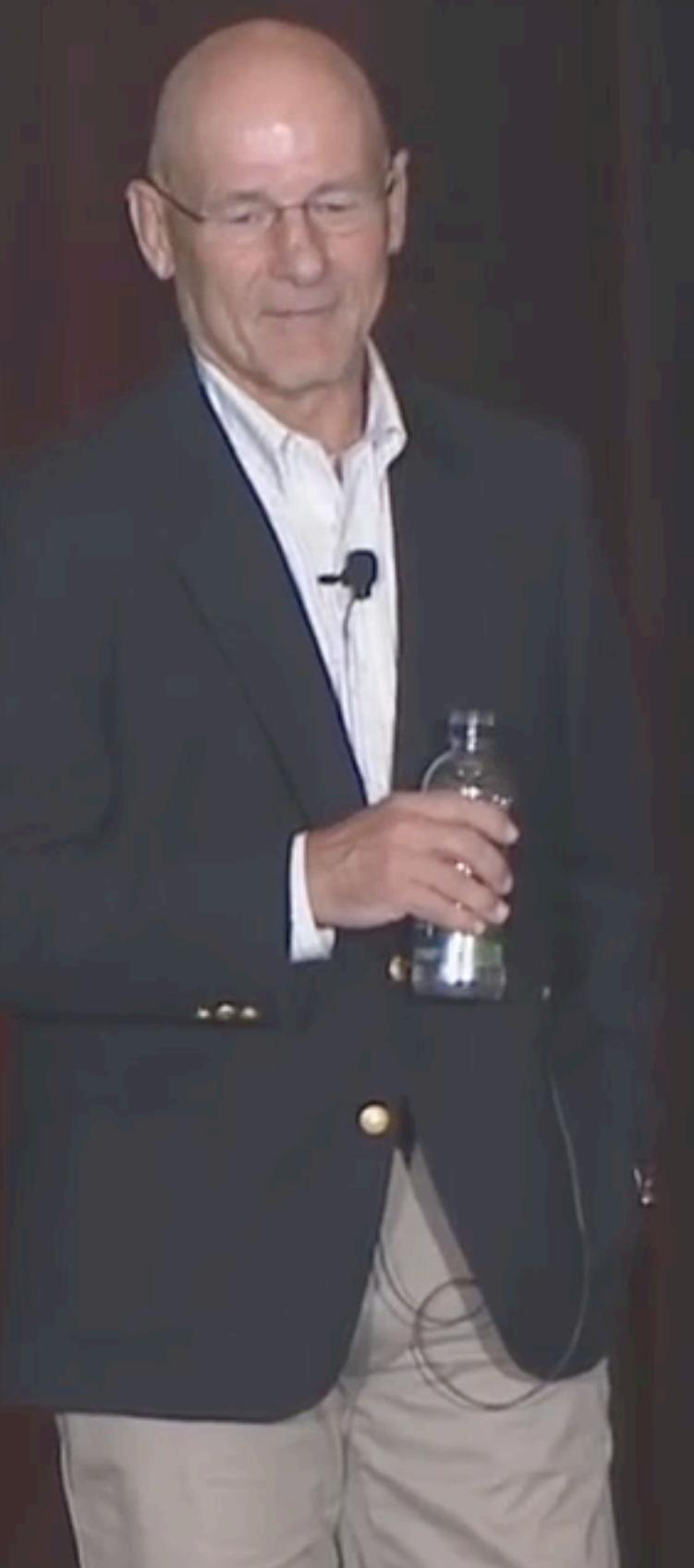
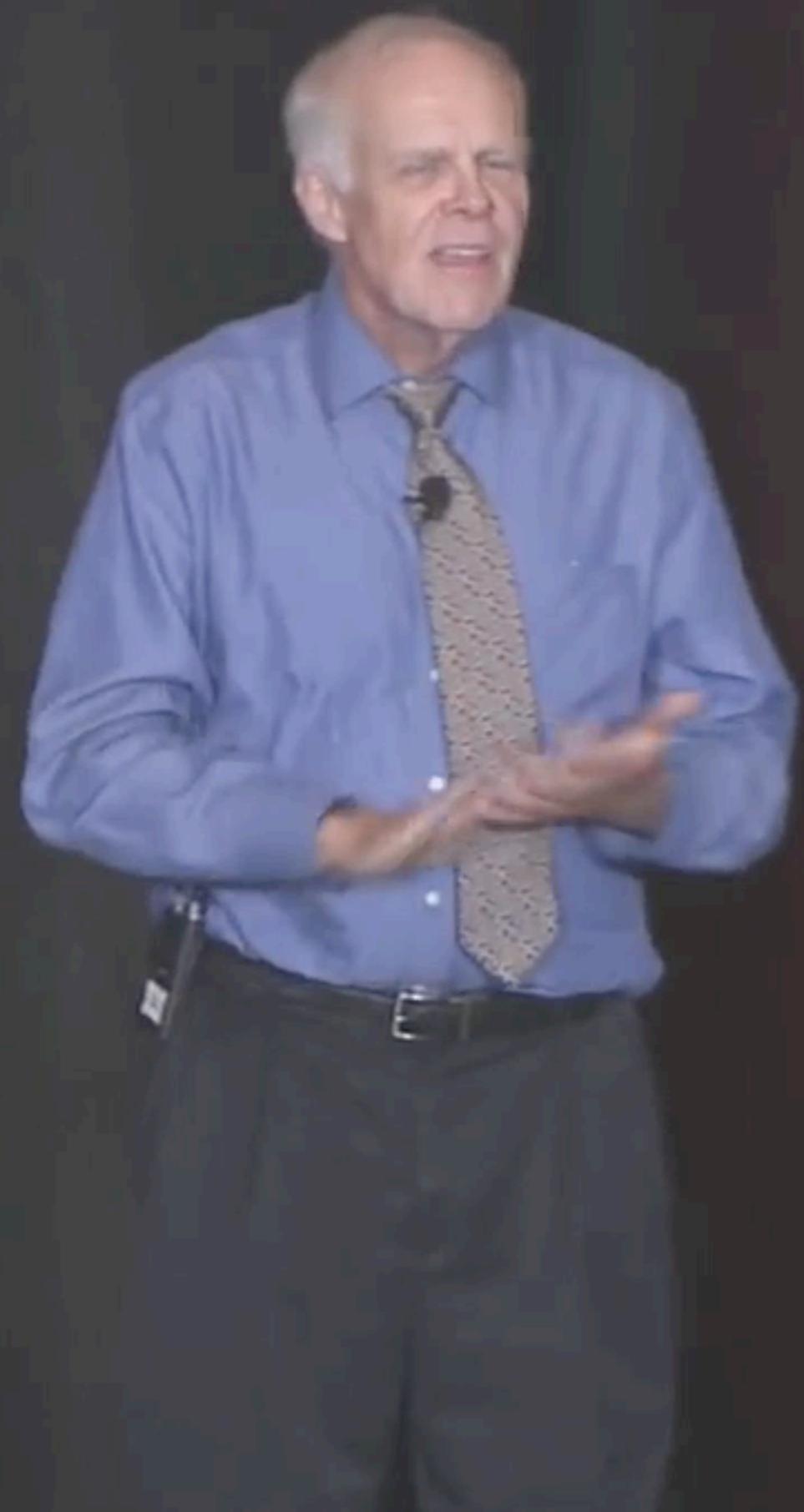
Take-aways: Why CS203?

- Processors/Memories are essential for modern computer systems
- Algorithm complexity does not work well on “real” computers
- Memory operations are more expensive and unpredictable than others
- Moore’s Law continues, but Dennard Scaling discontinues



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

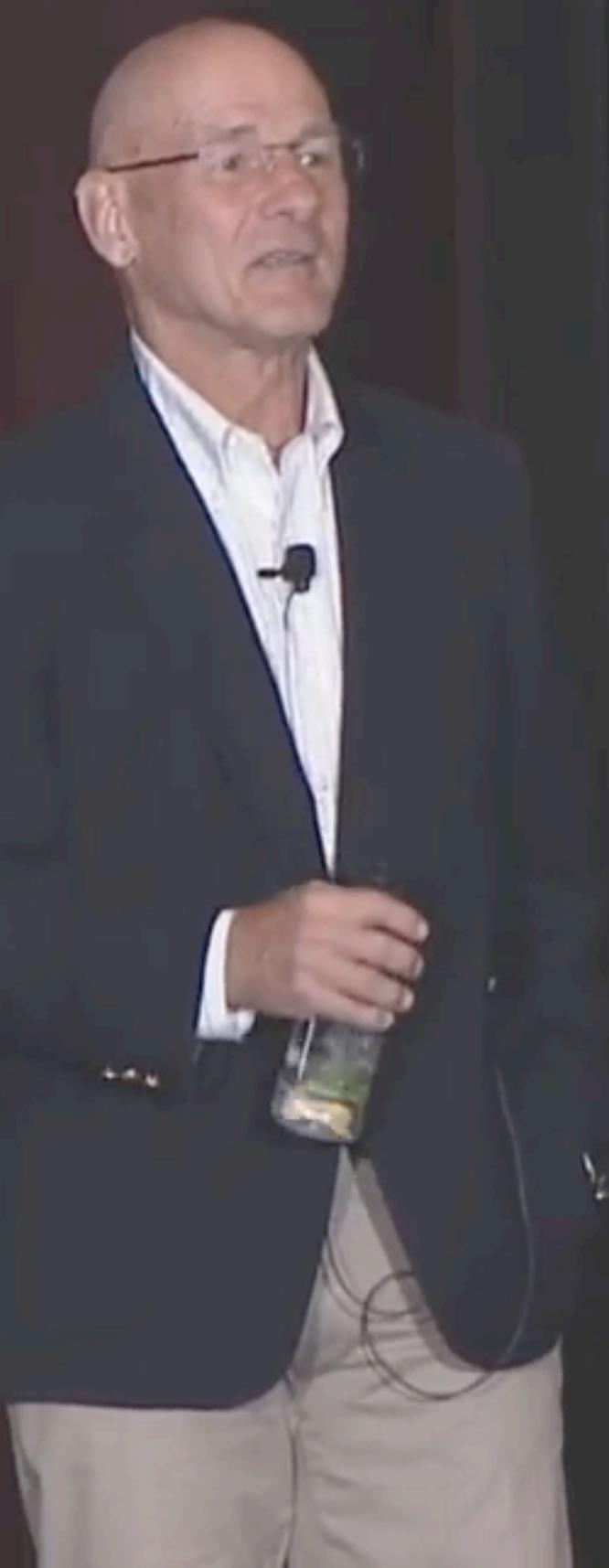
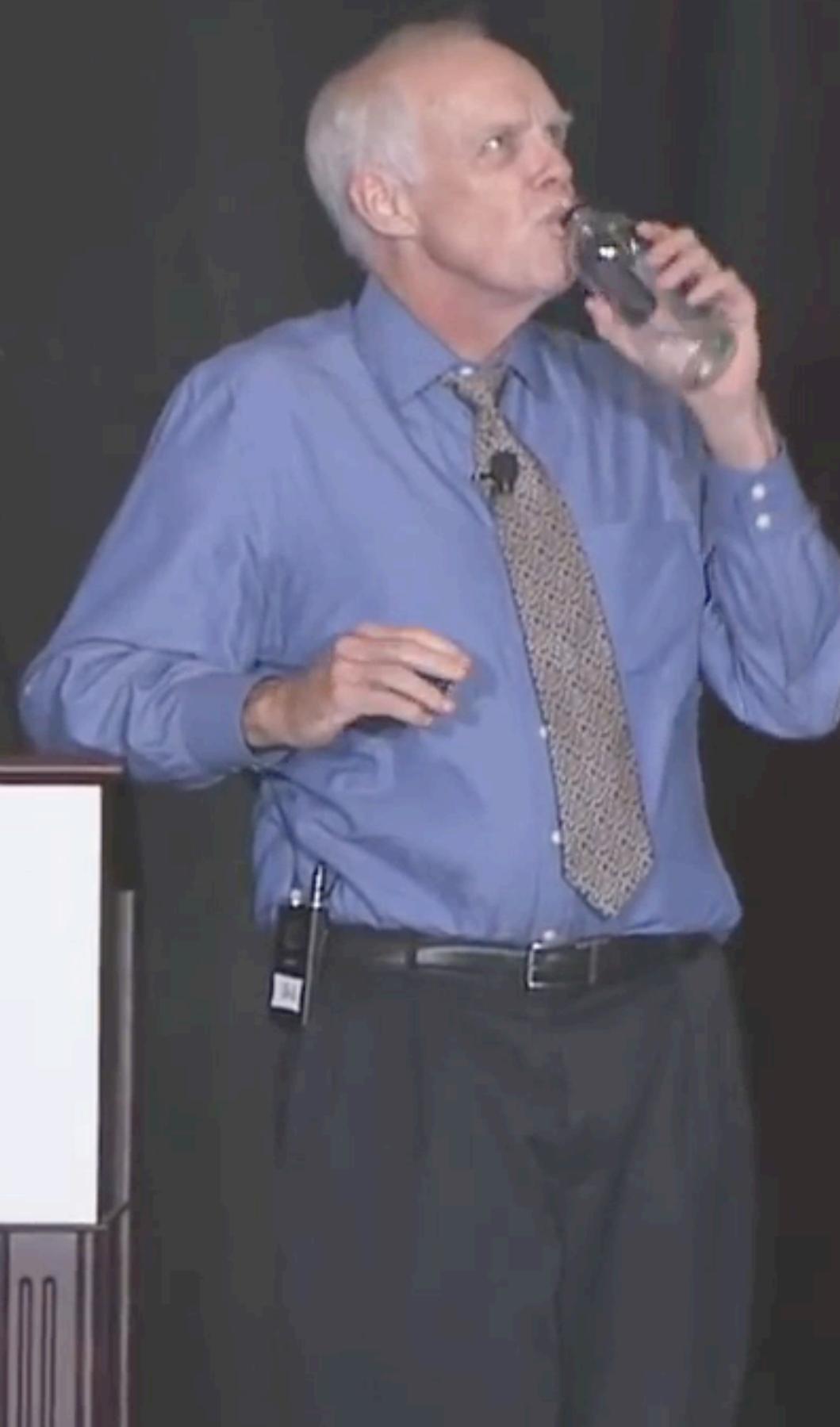
ISCA 2018
uring Lecture



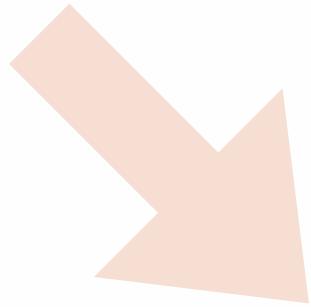
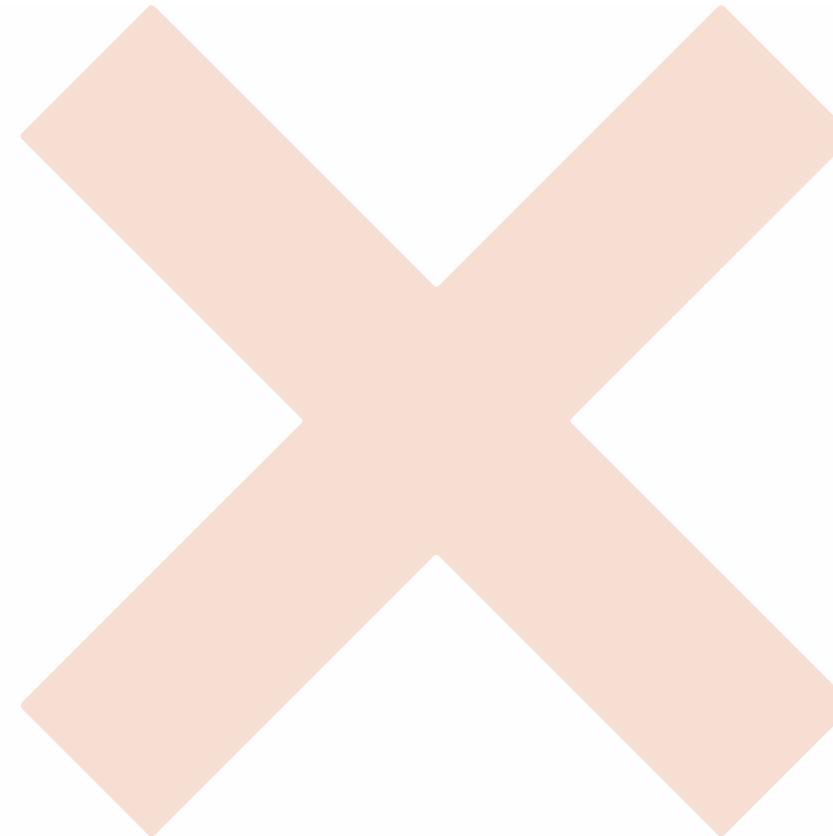
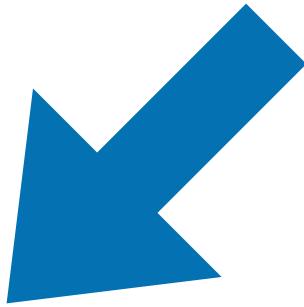
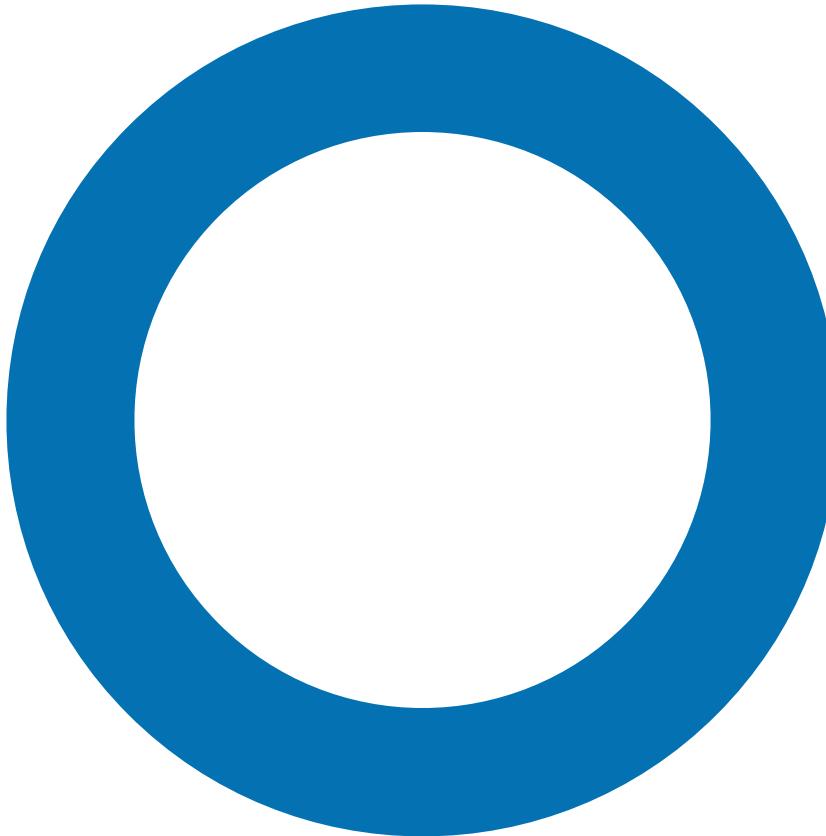


The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA

A 2018
g Lecture



Algorithm complexity is less important if we have unlimited parallelism



Demo (2) — quick sort v.s. bitonic sort on GPU

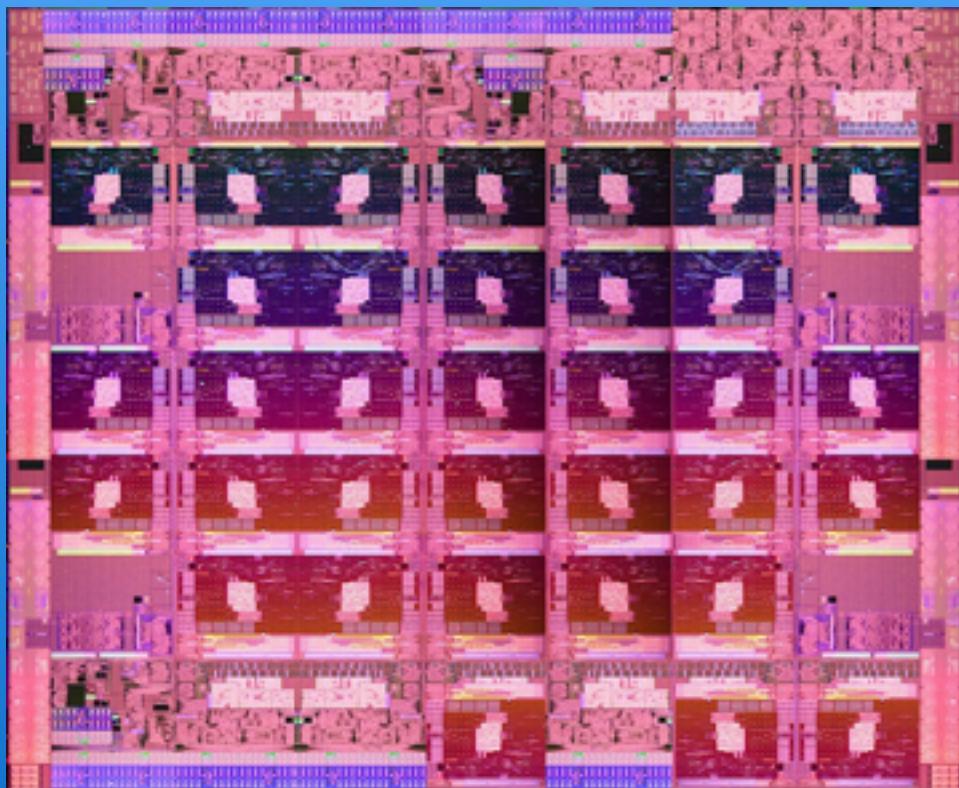
Quick Sort
 $O(n \log_2 n)$

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);  
                }  
            }  
        }  
    }  
}
```

Alternatives to scaling single cores

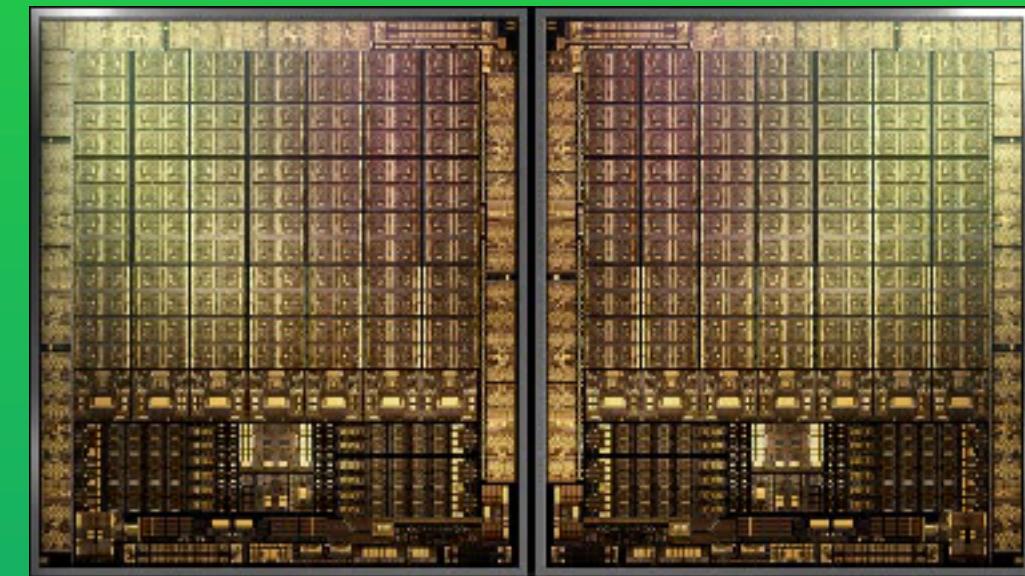
Multicore Processors



34-core Intel
Sapphire Rapids

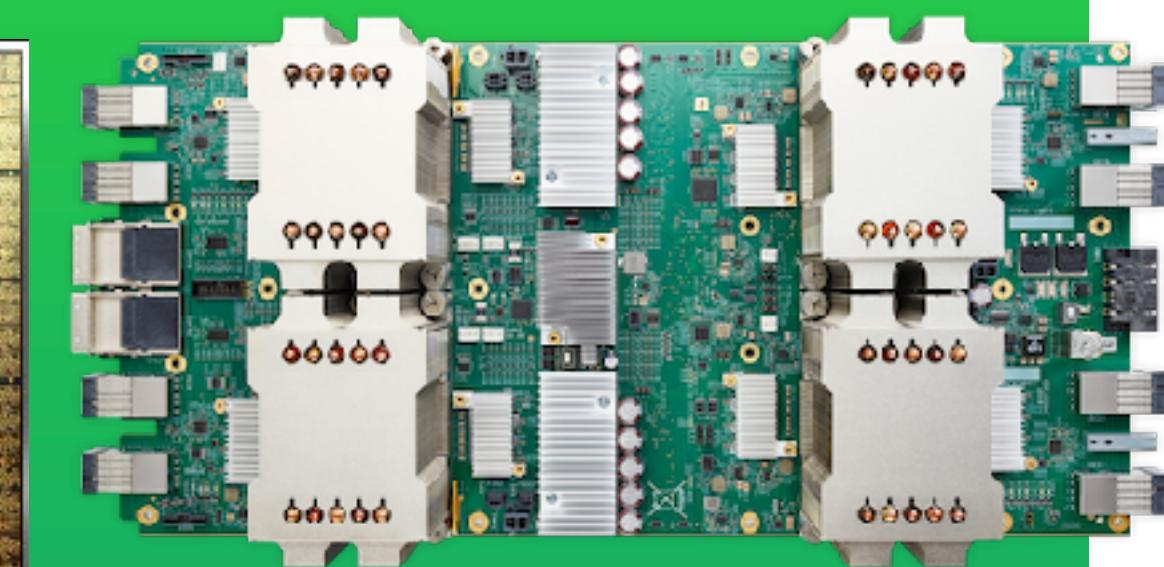
Thread-level parallelism

Hardware Accelerators



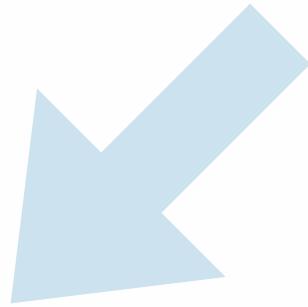
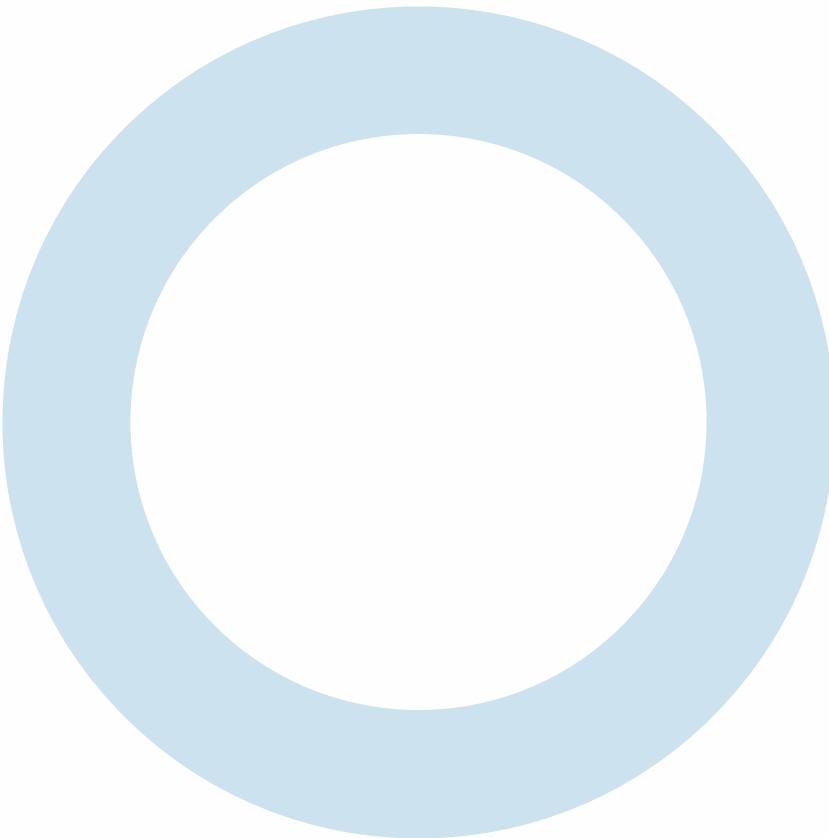
NVIDIA H100
Graphic Processing Units

Google
Tensor Processing Units



Data-level parallelism

GPUs are more power efficient per transistor than CPUs



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.

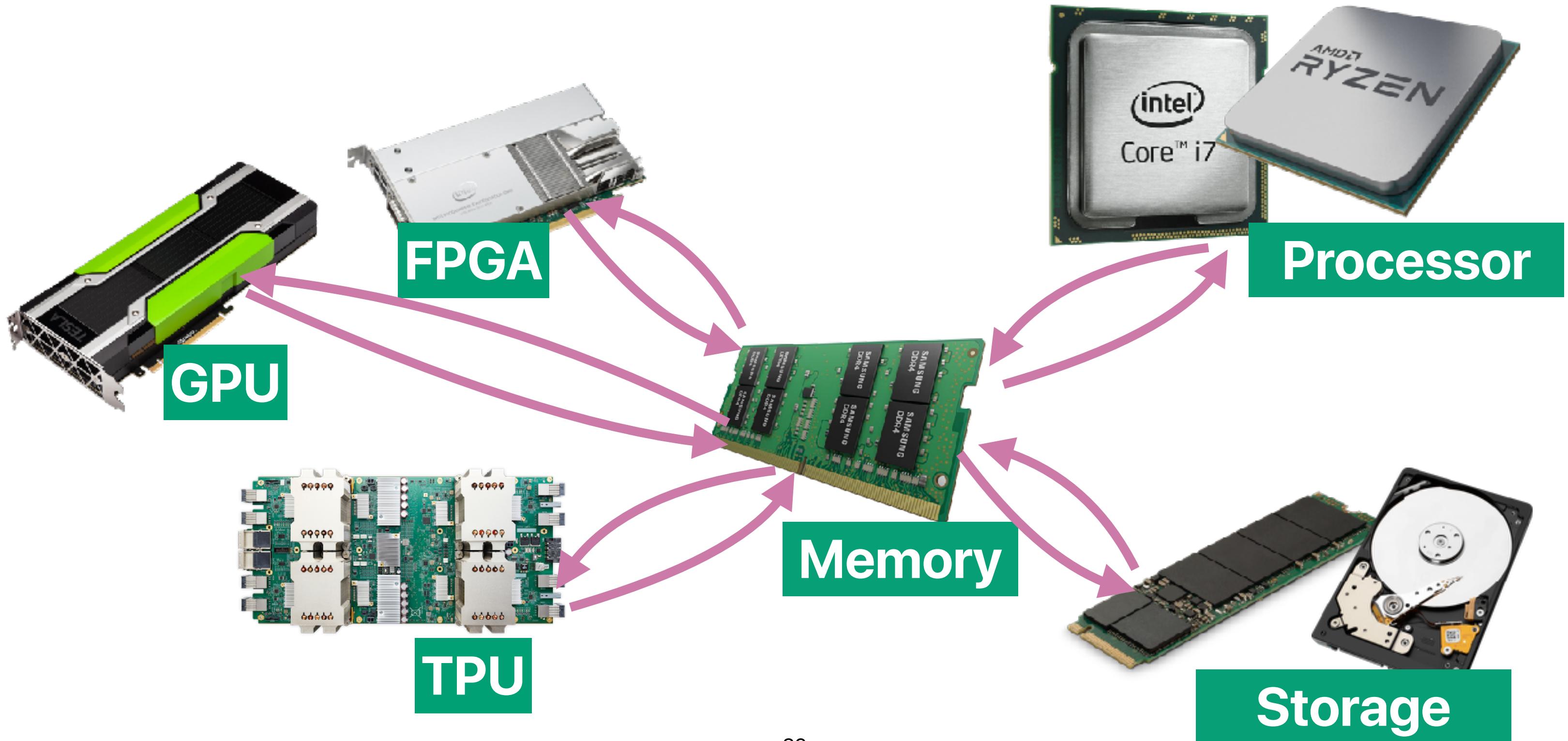
5.77 W/1B transistors

TSMC 4nm

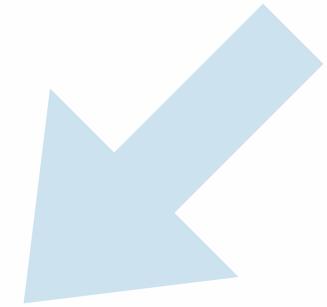
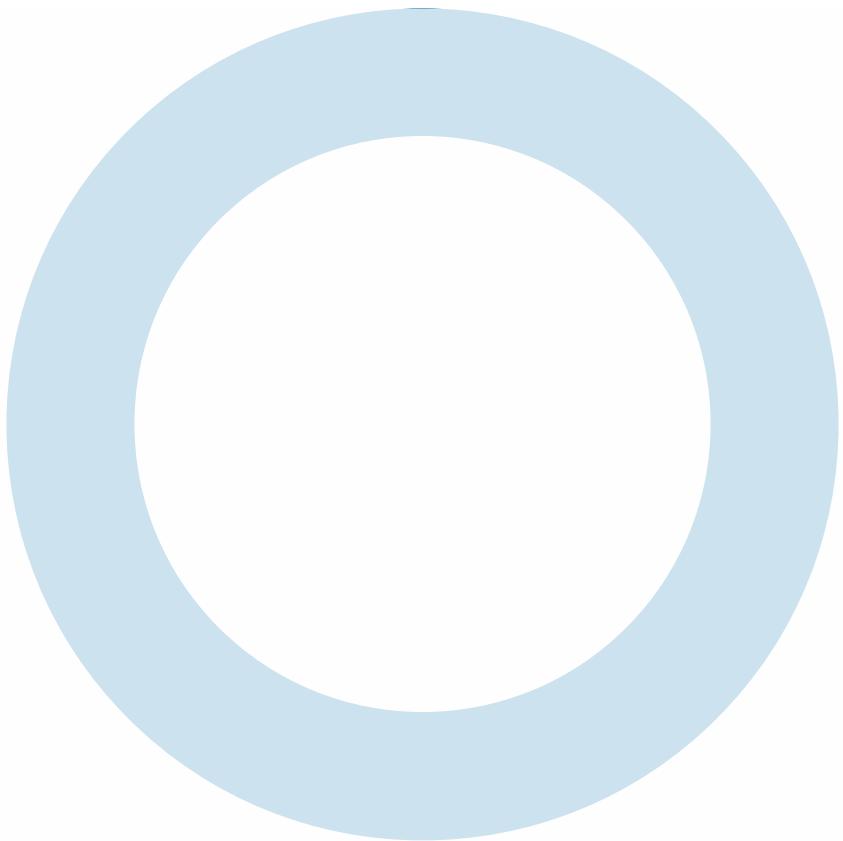
5.73 W/1B transistors

Intel 7nm

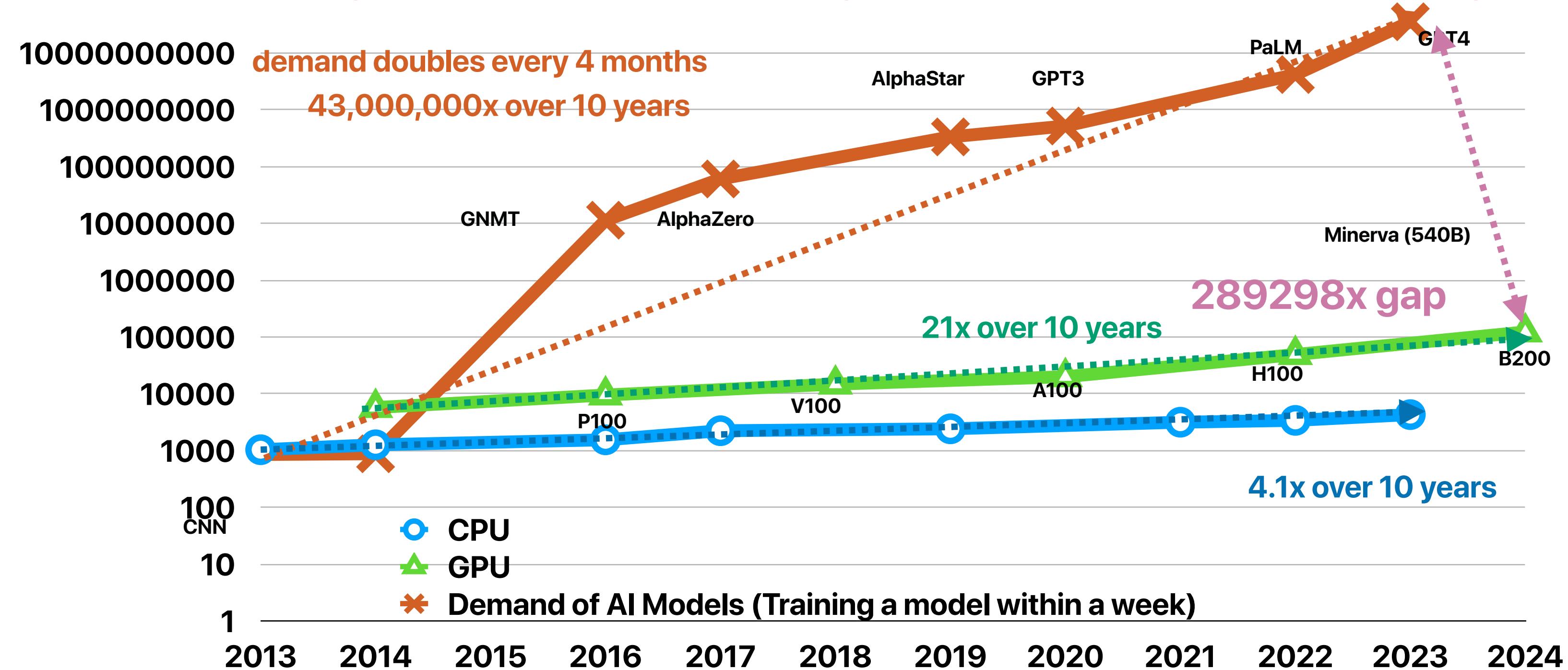
Heterogeneous Computer Architecture



**With 1000 NVIDIA's most high-end GPUs,
we can train a GPT-4 model within a month**



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



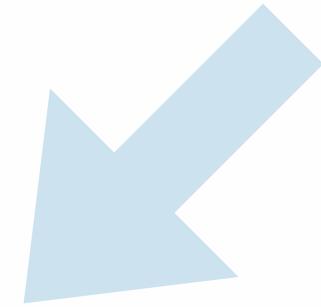
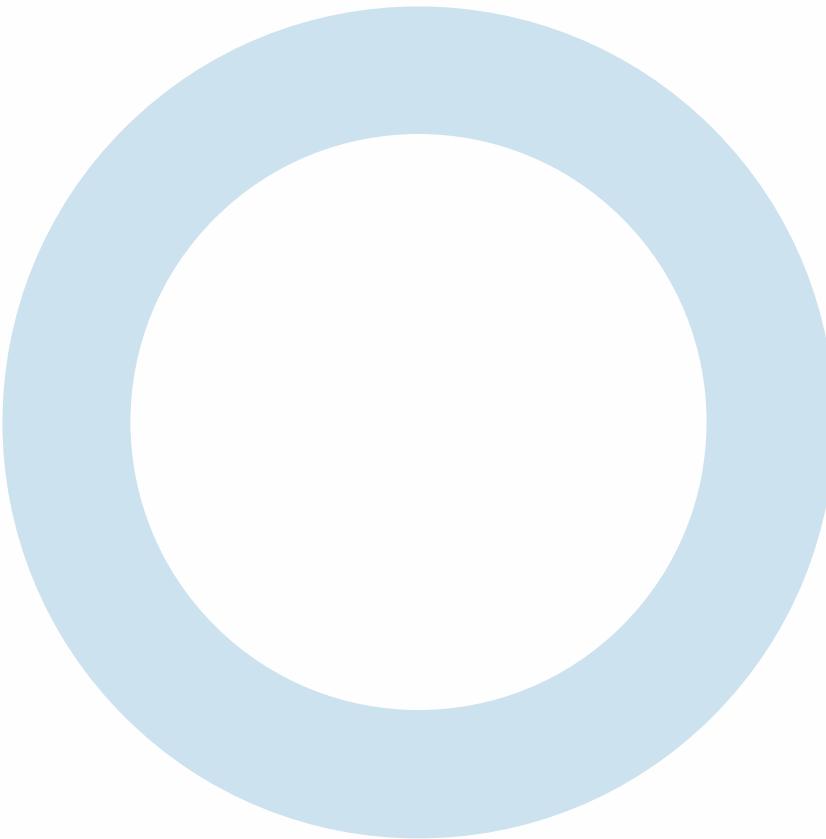
<https://ourworldindata.org/grapher/artificial-intelligence-training-computation>

OpenAI's training FLOPS for GPT-4 is ~2.15e25, on ~25,000 A100s for 90 to 100 days at about 32% to 36% MFU.

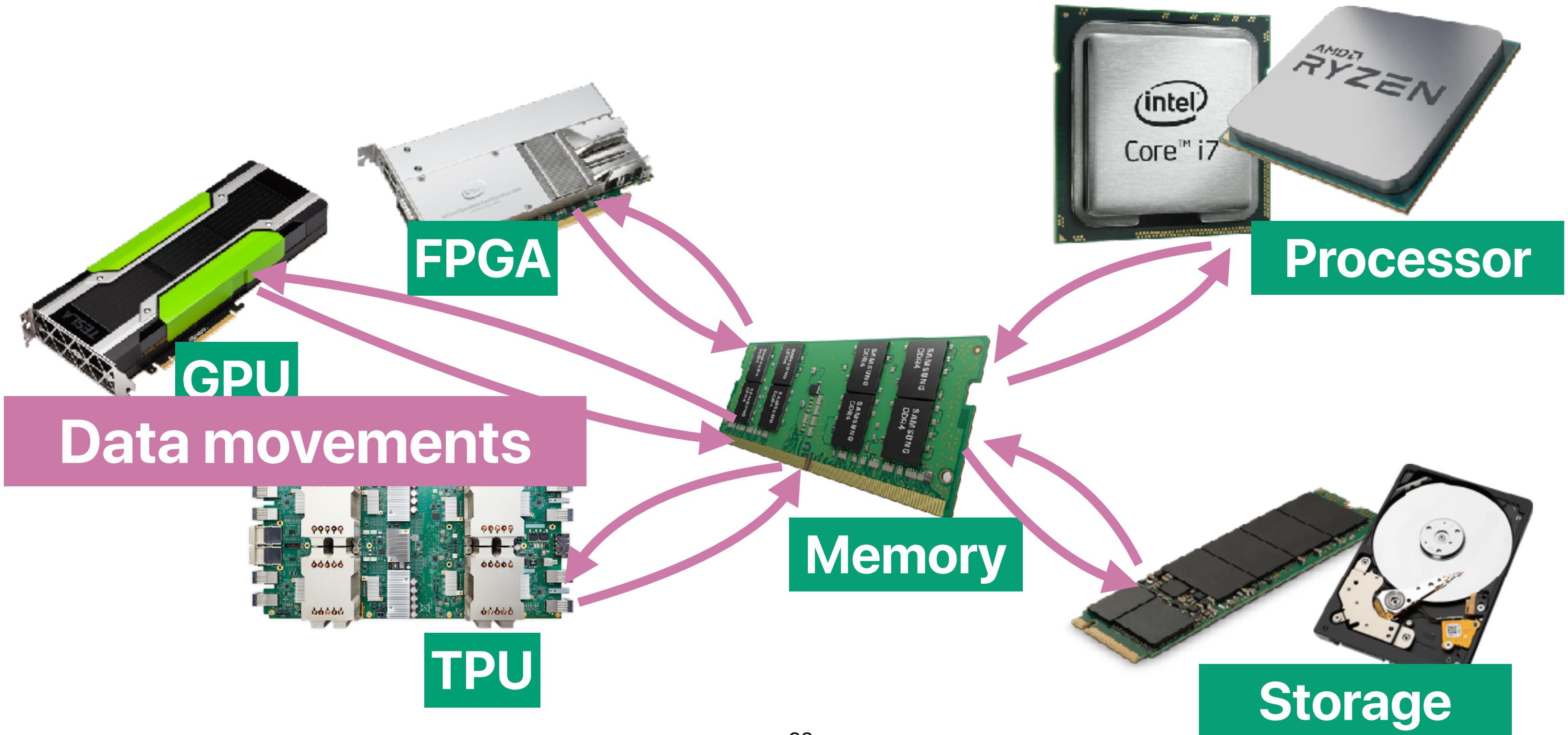
Part of this extremely low utilization is due to an absurd number of failures requiring checkpoints that needed to be restarted from. If their cost in the cloud was about \$1 per A100 hour, the training costs for this run alone would be about \$63 million.

<https://www.linkedin.com/pulse/supposed-leak-gpt4-architecture-alvaro-duran-tovar/>

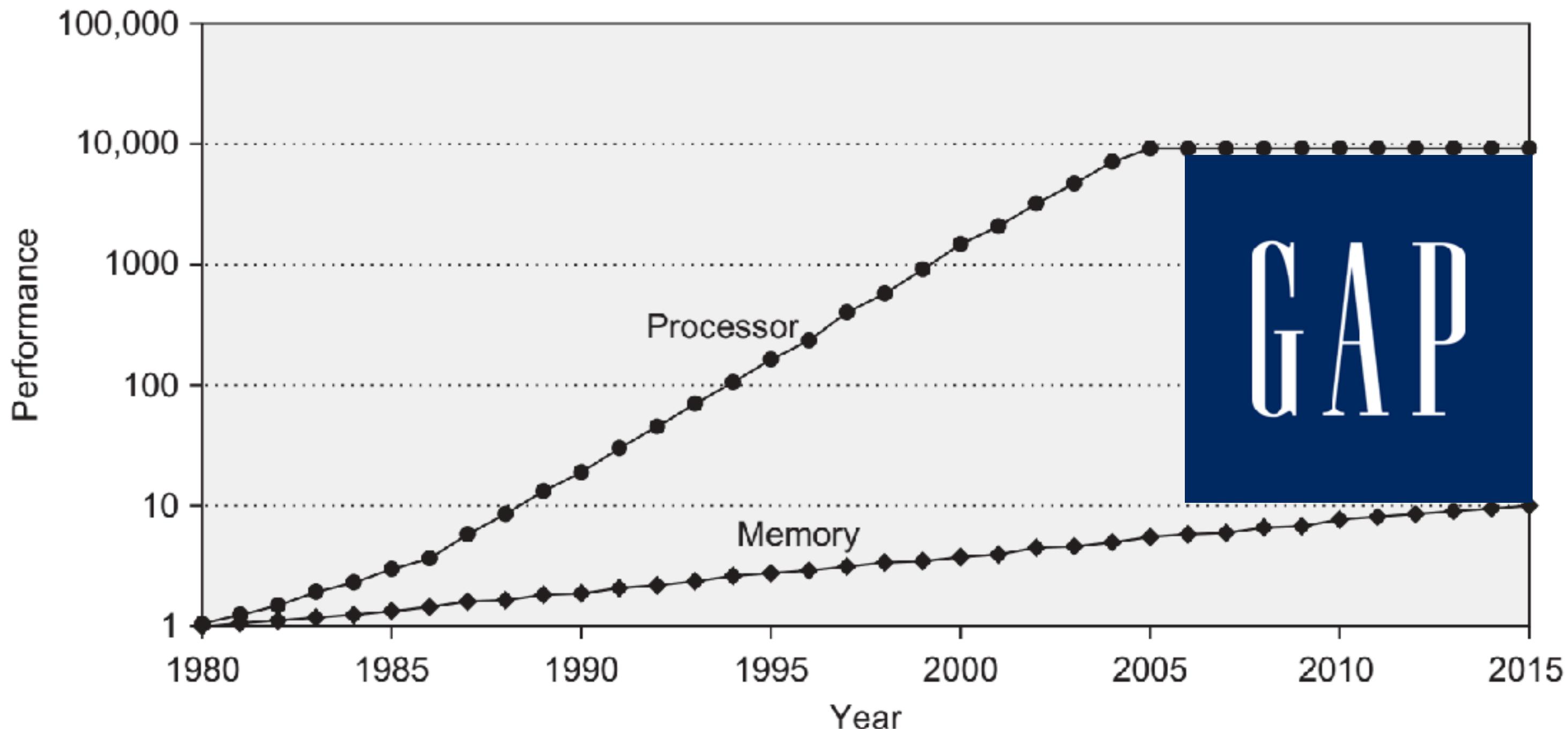
GPUs are going to replace CPUs



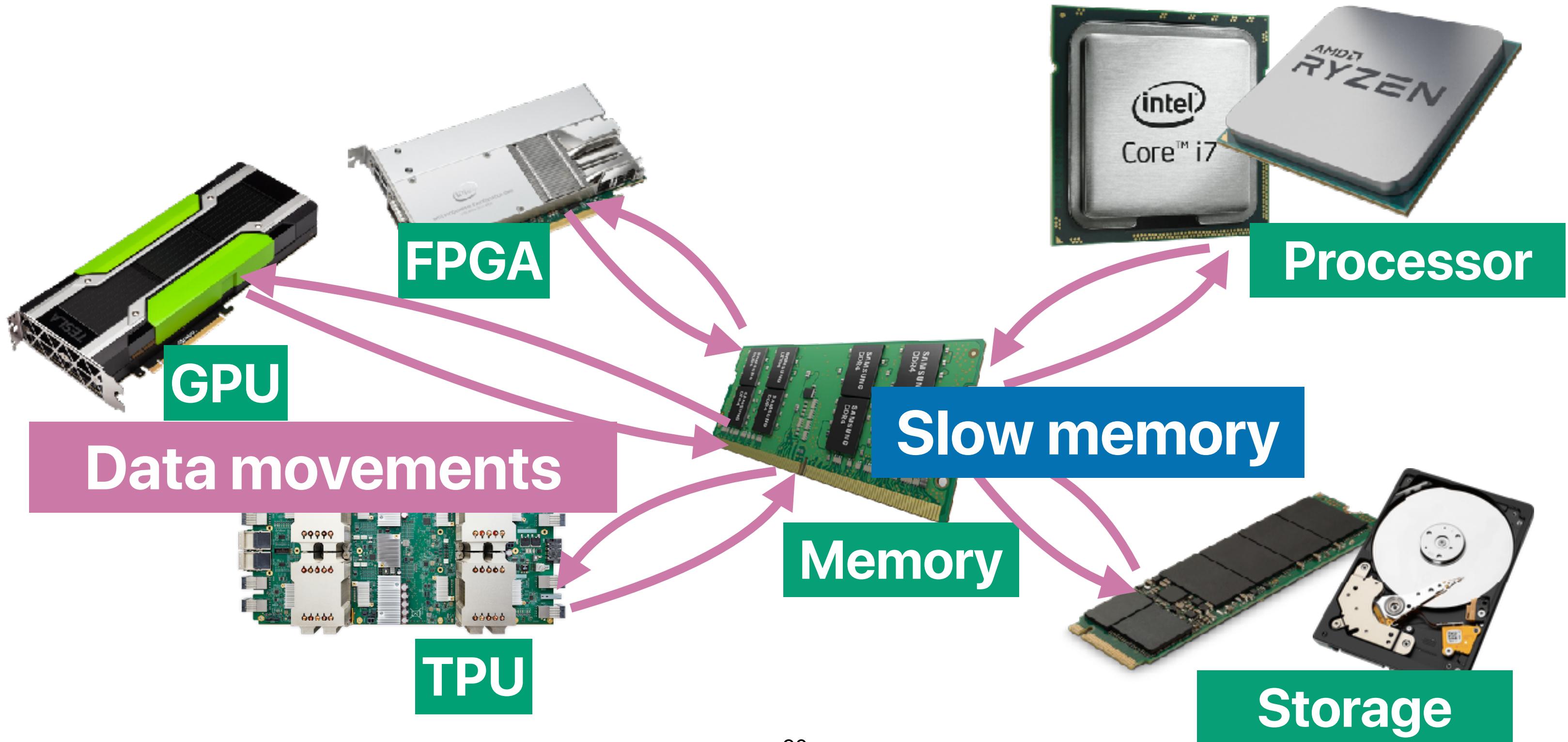
Heterogeneous Computer Architecture



Performance gap between Processor/Memory



Heterogeneous Computer Architecture



Take-aways: Why CS203?

- Processors/Memories are essential for modern computer systems
- Algorithm complexity does not work well on “real” computers
- Memory operations are more expensive and unpredictable than others
- Moore’s Law continues, but Dennard Scaling discontinues
- We have to rethink about programming as computers become more parallel and heterogeneous

What's going to be in the class?

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	Readings (Required)	Preview Slides	Slides & Demo	Reading Quiz Due	Assignment Due
04/02/2024	Introduction	- G.E. Moore. Cramming More Components				
04/04/2024	Performance Evaluation (I)	- Chapter 1.3 & 1.8-1.9			Reading Quiz #1	
04/09/2024	Performance Evaluation (II)	- M. D. Hill and M. R. Marty. Amdahl's Law in the Modern Era			Reading Quiz #2	
04/11/2024	Performance Evaluation (III)	(Optional) Andrew Davison, "Twelve Ways to Fool the System"				Assignment #1 Due
04/16/2024	Performance Evaluation (IV)	- Appendix B.1-B.3			Reading Quiz #3	
04/18/2024	Memory Hierarchy (1): The Basics	- Appendix B.1-B.3				Assignment #2 Due
04/23/2024	Memory Hierarchy (2)	- Chapter 2.3			Reading Quiz #4	
04/25/2024	Memory Hierarchy (3)					
04/30/2024	Memory Hierarchy (4): Optimizing Cache Performance	- Chapter B.4 & B.5, 2.4			Reading Quiz #5	
05/02/2024	Virtual Memory					Assignment #3 Due
05/07/2024	Midterm					
05/09/2024	Basic Processor Design & Branch Prediction	- Chapter 3.3			Reading Quiz #6	
05/14/2024	Branch Prediction	- M. Evers, S. J. Patel, R. S. Chappell and Y. N. Chen, "Branch Prediction: A Survey"			Reading Quiz #7	
05/16/2024	Data hazards	- Chapter 3.4				
05/21/2024	Data hazards & OOO Scheduling	- D. Suggs, M. Subramony and D. Bouvier, "The Impact of Data Hazards on Out-of-order Scheduling"			Reading Quiz #8	
05/23/2024	OOO Scheduling					Assignment #4 Due
05/28/2024	Programming Modern Processors	- Chapter 3.11			Reading Quiz #9	
05/30/2024	Chip Multiprocessors & Modern Processor Architectures	- D. Suggs, M. Subramony and D. Bouvier, "The Impact of Chip Multiprocessing on Processor Architectures"				
06/04/2024	Parallel architectures & Dark Silicon	- Chapter 17				Assignment #5 Due
06/06/2024	TPU, FPGA	Adrian McLaughlin, Elie Chang, Andrew Chien, "The TPU: A Special-Purpose Accelerator for Machine Learning"				
06/14/2024	Subject to change		According to the previous schedule	94	Check due dates here	

Performance

Memory

Processor

You need to complete the
reading of assigned reading
materials

According to the previous schedule

94

Logistics

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: <https://intra.engr.ucr.edu/~htseng/>
- E-mail: cs203@escalab.org
- Office Hours: W 2p-4p @ WCH 406
- 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



Teaching Assistant — Nurlan Nazaraliyev

- Office hours: M 2p-4p @ TBA
- E-mail: cs203@escalab.org



Course website

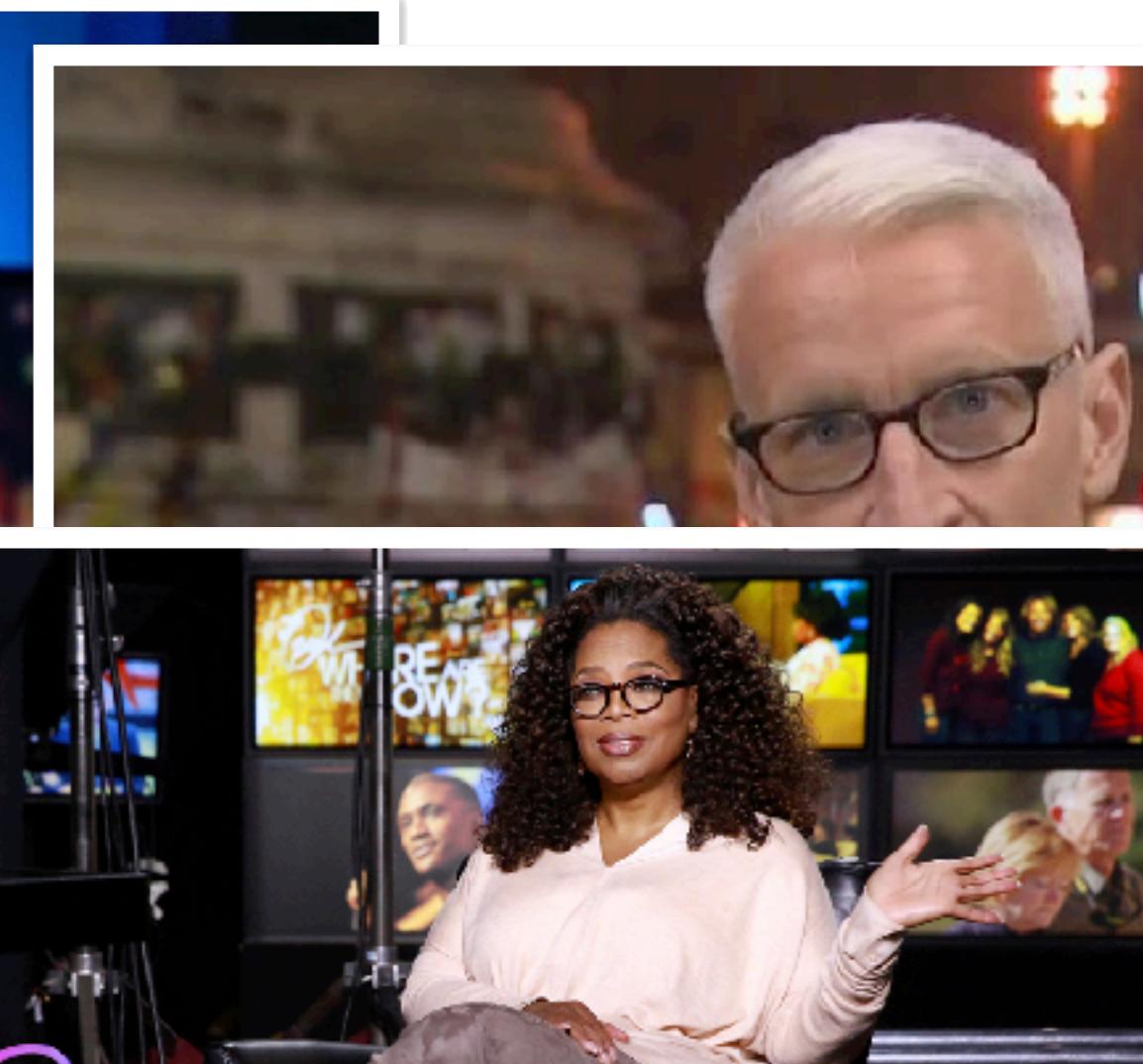
- <https://www.escalab.org/classes/cs203-2024sp>
- Calendar
- Schedule
- Slides
 - Preview — for the ease of note taking
 - Release — the actual slides
- My Grade
 - You may lookup your grades in the section

Summary of course resources

- Lectures:
 - In-person @
 - Repository on **Youtube**: <https://www.youtube.com/profusagi>
- Schedule, slides on **course webpage**:
<https://www.escalab.org/classes/cs203-2024sp>
- Discussion on **piazza**:
https://piazza.com/ucr/spring2024/cs_203_001_24s/home
- Reading quizzes, assignments, grading on **gradescope**:
<https://www.gradescope.com/courses/757338>
- Office Hours & Locations
https://calendar.google.com/calendar/u/0/r?cid=c_1b9678f28a16f7b598fcdd2f884dcf6ef27702c4c29b5715e61e6f1e121e2589@group.calendar.google.com

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 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
 - Computer Architecture: A Quantitative Approach 7th Edition by John Hennessy and David Patterson
 - AND other assigned materials
- Reading quizzes:
 - On gradescope
 - Due before the lecture, usually once a week. Check the schedule on our webpage
 - No time limitation until the deadline
 - Only one shot!!!
 - No make up reading quizzes — we will drop your lowest one at least

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

About the time of the Lecture — Setup Poll Everywhere

The image consists of four vertical screenshots of the Poll Everywhere mobile application, showing different stages of setup:

- Screenshot 1 (Top Left):** The app's listing on the App Store. It shows a blue icon with a bar chart, the title "Poll Everywhere", the subtitle "Poll Everywhere", a "OPEN" button, a 4.8-star rating, 1,960 ratings, #52 in Education category, and 4+ age rating. Below this is a "What's New" section for Version 2.0.4 (1w ago), which includes a list of changes: "Add menu link to register with a presenter", "Allow signing up as a participant", and "Improved error messaging". There is also a "more" link.
- Screenshot 2 (Top Middle):** The app's main menu screen. It features a sidebar with "Participants" (selected), "Join a presentation", "Register with a presenter", "Respond by keyword", and "Response history". The main area has sections for "Help & Feedback" (with "Contact support") and "Recent presentations" (listing "PollEv.com/hungweitseng").
- Screenshot 3 (Top Right):** The login screen. It has "Log in" and "Sign up" buttons. A text input field contains "htseng@ucr.edu", and a "Log in with UCR" button is below it. At the bottom are "Privacy" and "Terms" links.
- Screenshot 4 (Bottom Right):** The presentation selection screen. It shows a large "Join" button above a URL "PollEv.com/hungweitseng". Below this is another "Join" button and a "Recent presentations" section with the same URL listed again.

Text overlay: In the center of the image, there is overlaid text in orange and pink colors:

**Login through the app using
UCSD_username@ucsd.edu**

**Join
PollEv.com/
hungweitseng**

Jupyter notebook based assignment — learning & reviewing by practicing

Course Infrastructure: Jupyter Notebook

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



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

For every assignment

- Watch this video first!!! <https://youtu.be/cSISDCfrUmk>
- Go to course home page:
<https://www.escalab.org/classes/cs203-2024sp>
- Click invitation link for the upcoming assignment
- Log into <https://www.escalab.org/datahub>
- Select this option and click “Launch Environment”
- Open a terminal
- Clone your starter repo.
- Open up Assignment.ipynb
- Follow the instructions

Course Infrastructure: Github and github classroom

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

In each assignment

- You should expect 20-30 questions per assignment (except for the 1st one).
- Correctness
 - Demonstrate mastery
 - Give the right answer — earn points
- Completeness
 - “forcing function” to get you to engage with the material
 - Give an answer — earn points
 - We will grade ~5 of these at random per assignment.
- Optional
 - Optional material for interested students.
 - Give the right answer — earn a sense of personal accomplishment

Submitting your assignment

- All assignments will mostly be submitted via gradescope
- Please ensure your answers show complete/necessary explanations.
 - Considering its an interview
 - We don't regrade unless there is an obvious error — thinking about this — can you request a "re-interview"?
- Assignments
 - Everything autograded
 - Submitted via github on gradescope
 - You will receive a feedback a few minutes later — so act early
 - Unlimited submission before the deadline
- Post-lab survey
 - Embedded in the assignment as a google form.

Examines

We still have examines

- To verify if you really have the concept/high-level ideas **in mind**
- Midterm — 5/7, closed-book, no cheatsheets allowed
- Final — 6/14, closed-book, no cheatsheets allowed
 - 2 Questions for MSCS comprehensive exams

Grading Breakdown

	In-person session
Reading Quizzes	15% Drop lowest 2
Participation	Count as one assignment
Assignments	25% Drop lowest 1
Midterm	25% In-person, closed book
Final	35% In-person — closed book

Academic Honesty

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

Hey, I need help...

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

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

- I'm going to miss class
 - Sorry to miss you! Please watch my youtube channel!
<https://www.youtube.com/profusagi>
 - In-depth class concept question (e.g., what's the difference between pass-by-value vs. pass-by-reference)
 - Class or Discussion (piazza)
- I can't login to escalab.org/datahub
 - E-mail cs203ucr@googlegroups.com

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

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

Why...

- Do I really need the textbook
 - Again, we need to prepare you for lectures
 - Textbook helps to make sure we're all on the same page when we talk about something

Do you think this tells you about ChatGPT and your life?



**Generative AI is still not reliable &
that's why "you" still need to learn!**

Course agreement

- I have reviewed the schedule and policies listed on the course website: <https://www.escalab.org/classes/cs203-2024sp>
- I understand that all deadlines in this class are hard deadlines with no extensions possible. It is my duty to follow the published schedule and carefully plan my time to fulfill the deadlines.
- I understand that CS203 is an Advanced Computer Architecture class for graduate students. My duty is to ensure that I grasp the prerequisites well, including undergraduate computer architecture class and C/C++ programming languages. If I am not confident I can catch up with the prerequisites, I should take an undergraduate computer architecture class instead.
- I understand that this class requires intensive readings in research papers and the assigned textbook.
- I understand that this class requires participation and discussion. If I participate in peer instruction questions remotely, it is considered cheating.
- I understand this class requires programming assignments using the C/C++ programming language. It is my responsibility to learn how to program in C/C++. I am also responsible for designing the architecture, implementation details, and tests for all coding assignments.
- I understand that the instructor and course staff reserve the right to refuse to answer inappropriate questions (e.g., directly telling if an answer is correct).
- I understand that I am responsible for tracking the latest schedule, information, grades, and materials from our course website, e-mails from the course staff, and the Piazza forum. If there exists any conflict between the submission deadlines, **the course website** is the authority to follow.
- I understand that *the collaboration is allowed* on assignments of this course. If I discuss with other students, have to list their names in my assignment, and even with that, plagiarism or any text that may be considered plagiarism by human beings or AI assisted tools is prohibited and will be reported to the UCR Student Conduct & Academic Integrity Programs.
- I understand that the plagiarism detection will be treated seriously. Any cheating and plagiarism cases will be referred to the UCR Student Conduct & Academic Integrity Programs.
- I understand that *copying code snippets from websites, including StackOverflow are considered plagiarism.*
- I know this class will use tools based on MOSS (<https://theory.stanford.edu/~aiken/moss/>) to detect plagiarism. If the similarity score is higher than 98%, we will consider this assignment involving plagiarism and lead to an F in the final grade.
- I should drop the course if I disagree with any of the above rules.



By clicking this box, you are agreeing to the Terms and Conditions of CS 203, Spring 2024

Announcements

- Check our website
- Login to Gradescope
 - Reading quiz due **Thursday before the lecture**
- Assignment #1 due **next Thursday**
 - **Already released! Check our website for the link**
 - You cannot start your assignment if you did not submit the course agreement — you'll need to wait for 24—48 hours before you can start working on the assignment.
 - Please act early
 - We auto-grade everything assignment (both writing and programming part). You will get early feedbacks before the deadline
 - Unlimited submissions before the deadline
- Login to piazza
- Setup your poll everywhere App

Computer Science & Engineering

203

つづく

