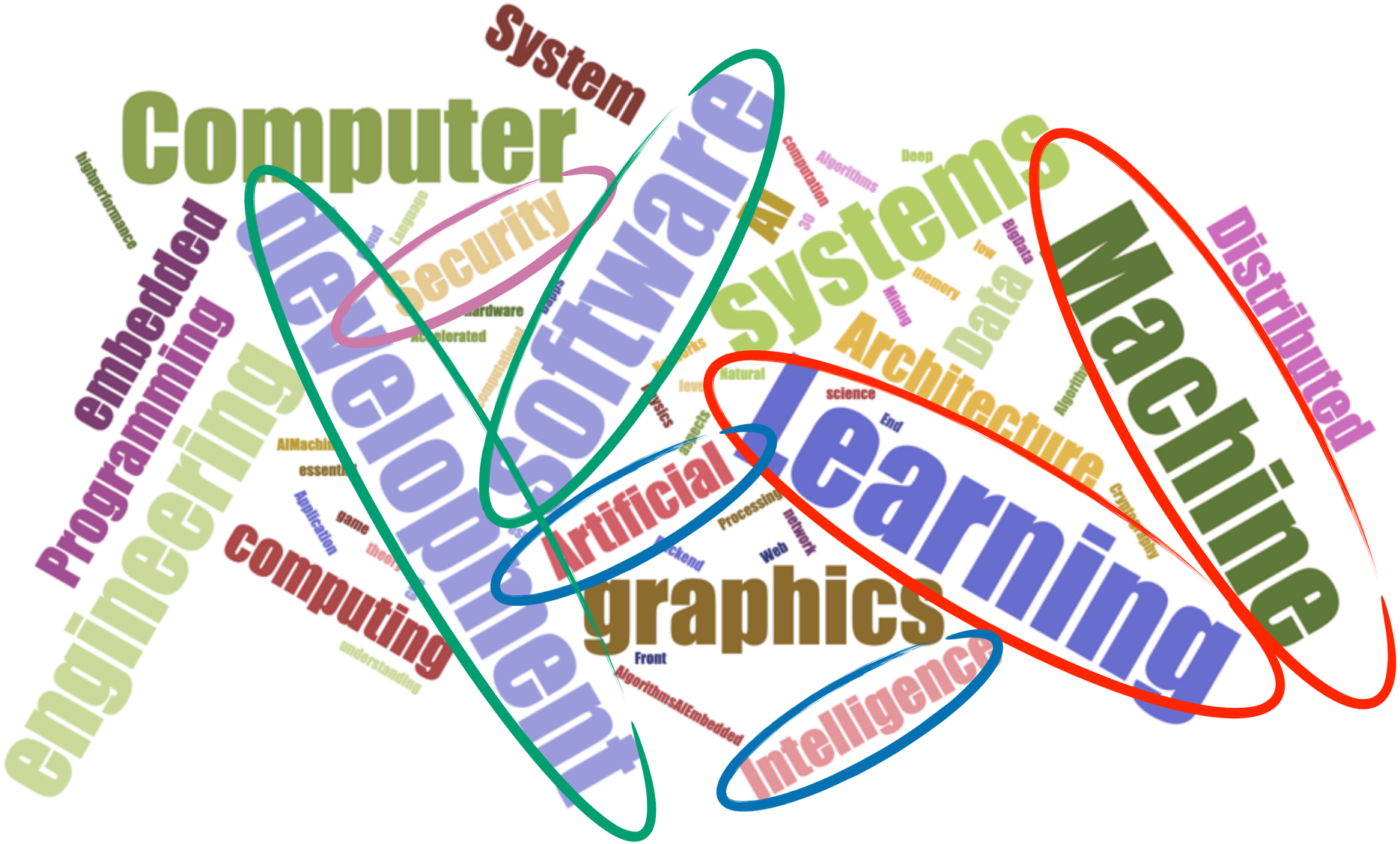
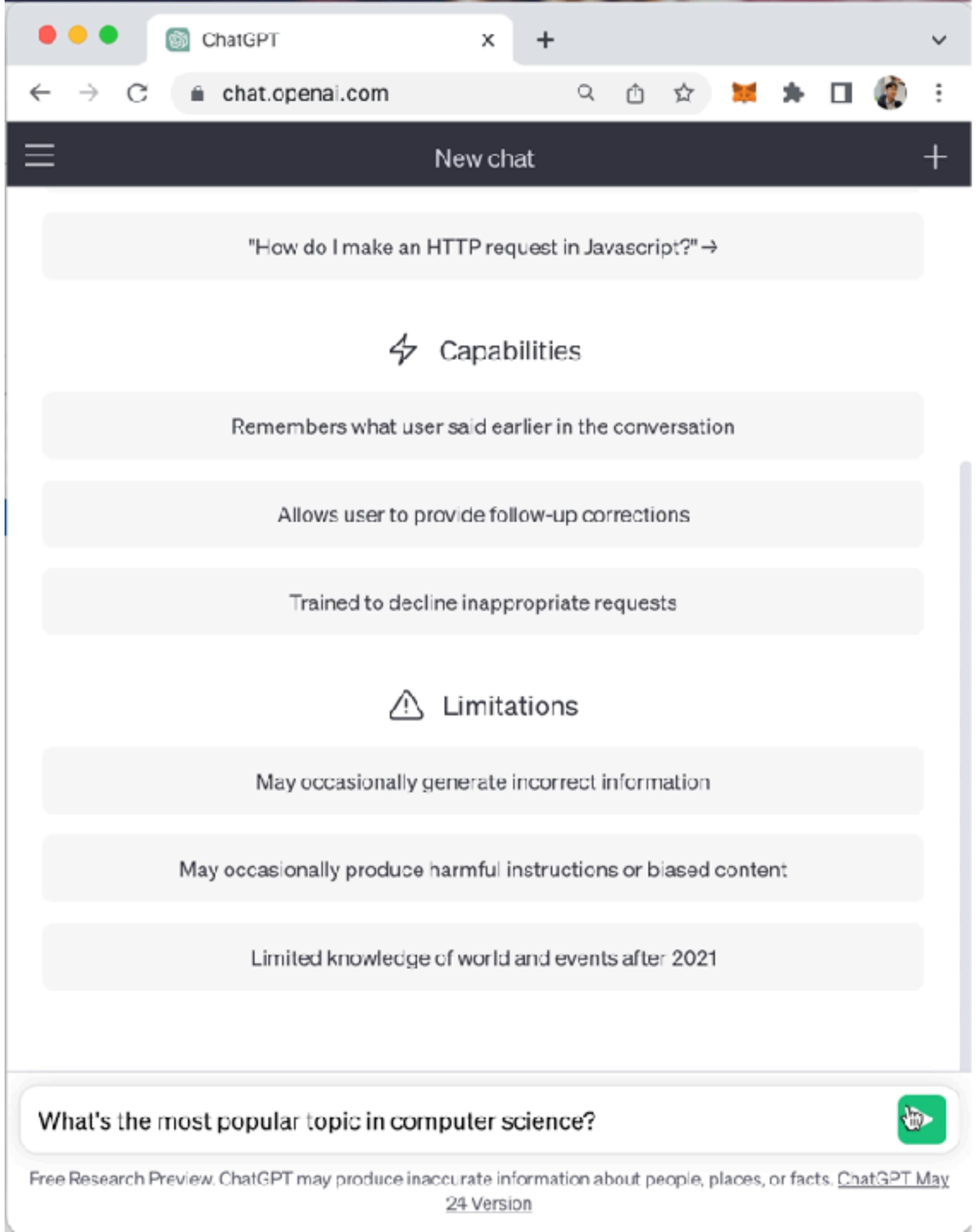
First Day of CS203: Advanced Computer Architecture (2023 Fall)

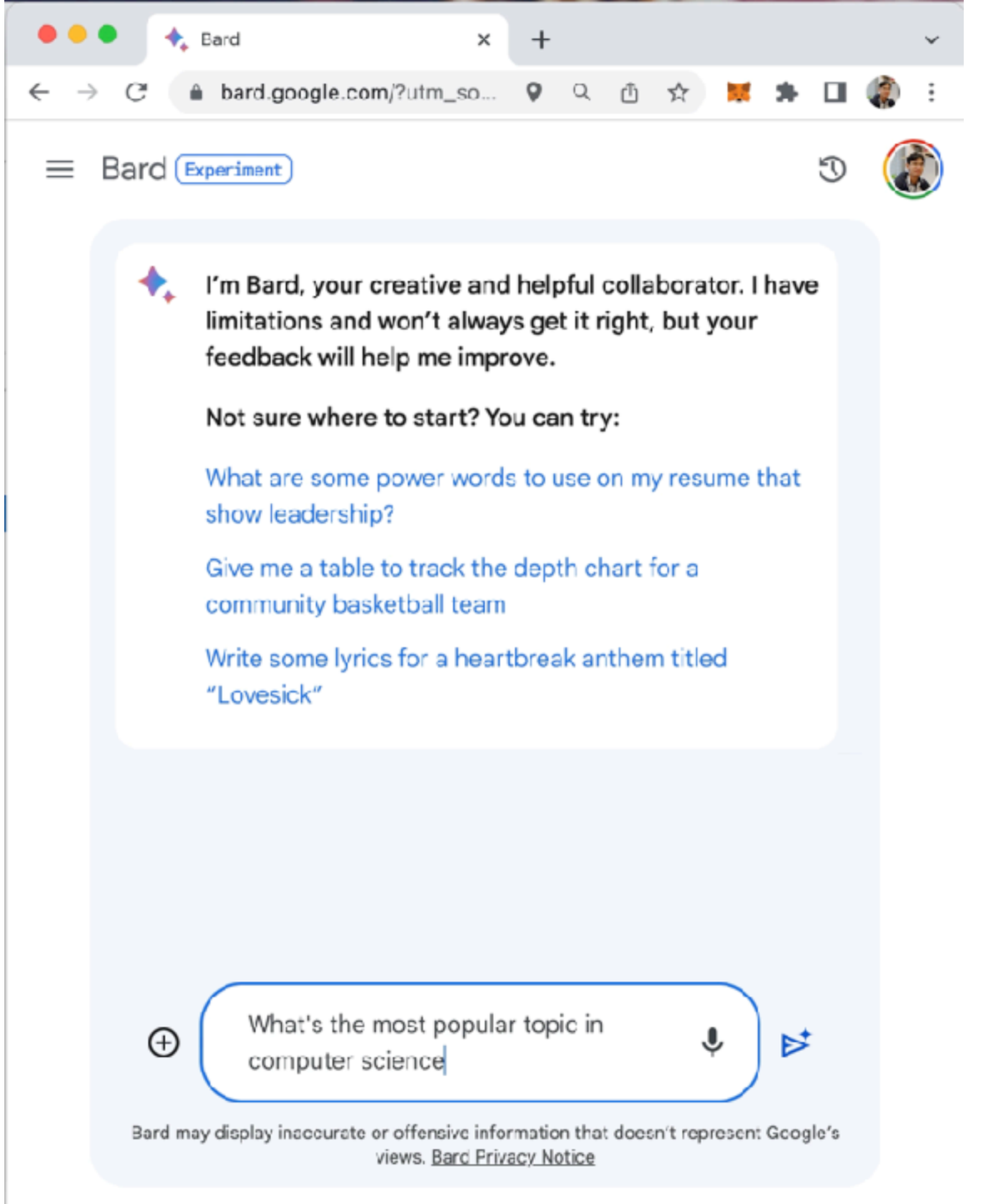
Hung-Wei Tseng

Why are we taking CS203

What are you interested




ChatGPT
chat.openai.com
New chat
"How do I make an HTTP request in Javascript?" →
Capabilities
Remembers what user said earlier in the conversation
Allows user to provide follow-up corrections
Trained to decline inappropriate requests
Limitations
May occasionally generate incorrect information
May occasionally produce harmful instructions or biased content
Limited knowledge of world and events after 2021
What's the most popular topic in computer science?
Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT May 24 Version


Bard
bard.google.com/?utm_so...
Bard Experiment
I'm Bard, your creative and helpful collaborator. I have limitations and won't always get it right, but your feedback will help me improve.
Not sure where to start? You can try:
What are some power words to use on my resume that show leadership?
Give me a table to track the depth chart for a community basketball team
Write some lyrics for a heartbreak anthem titled "Lovesick"
What's the most popular topic in computer science
Bard may display inaccurate or offensive information that doesn't represent Google's views. Bard Privacy Notice

What do you care as a computer scientist?

Algorithms & Data Structures

Software Engineering

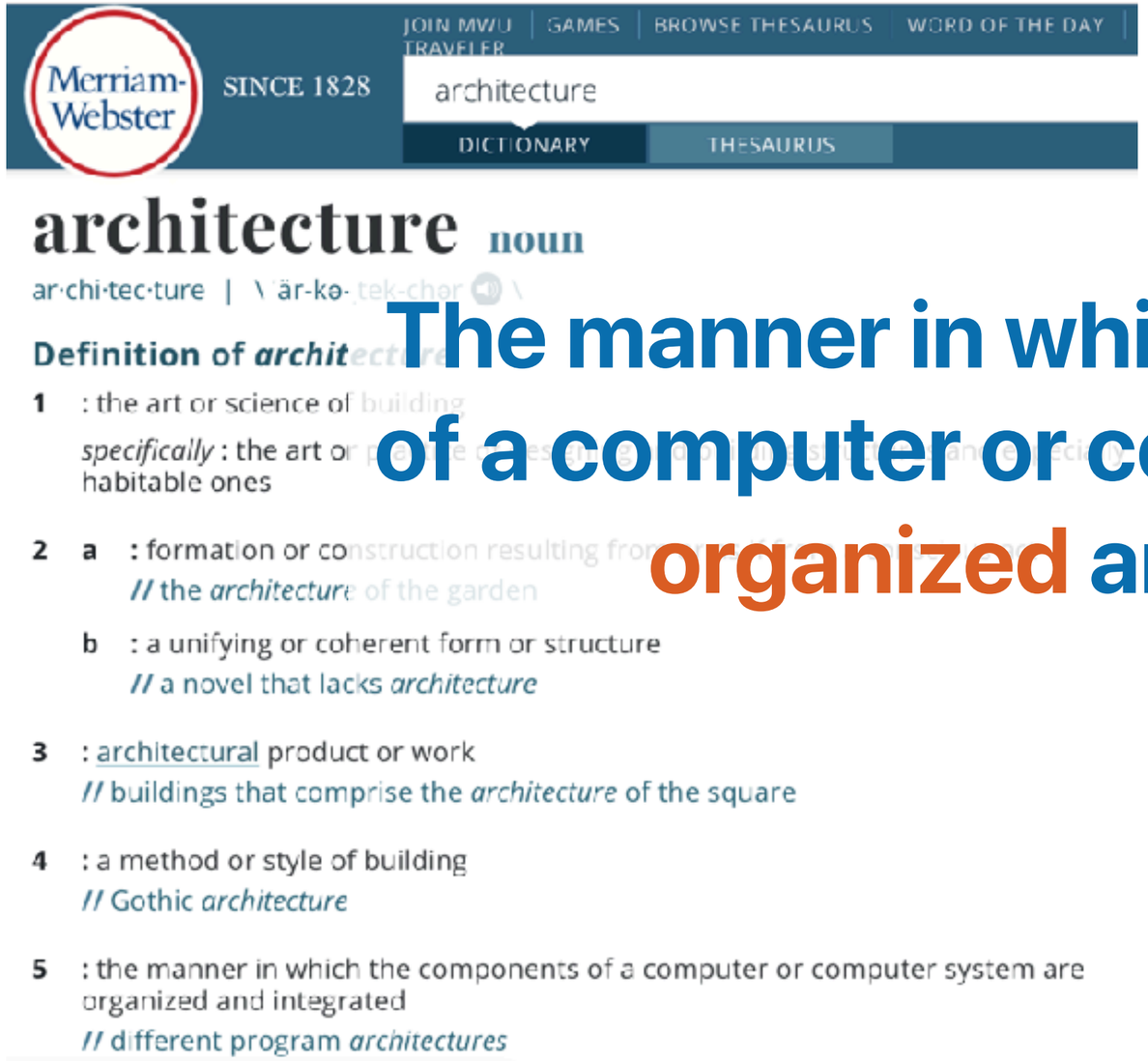
AI & ML

Computer Architecture

Programming Languages

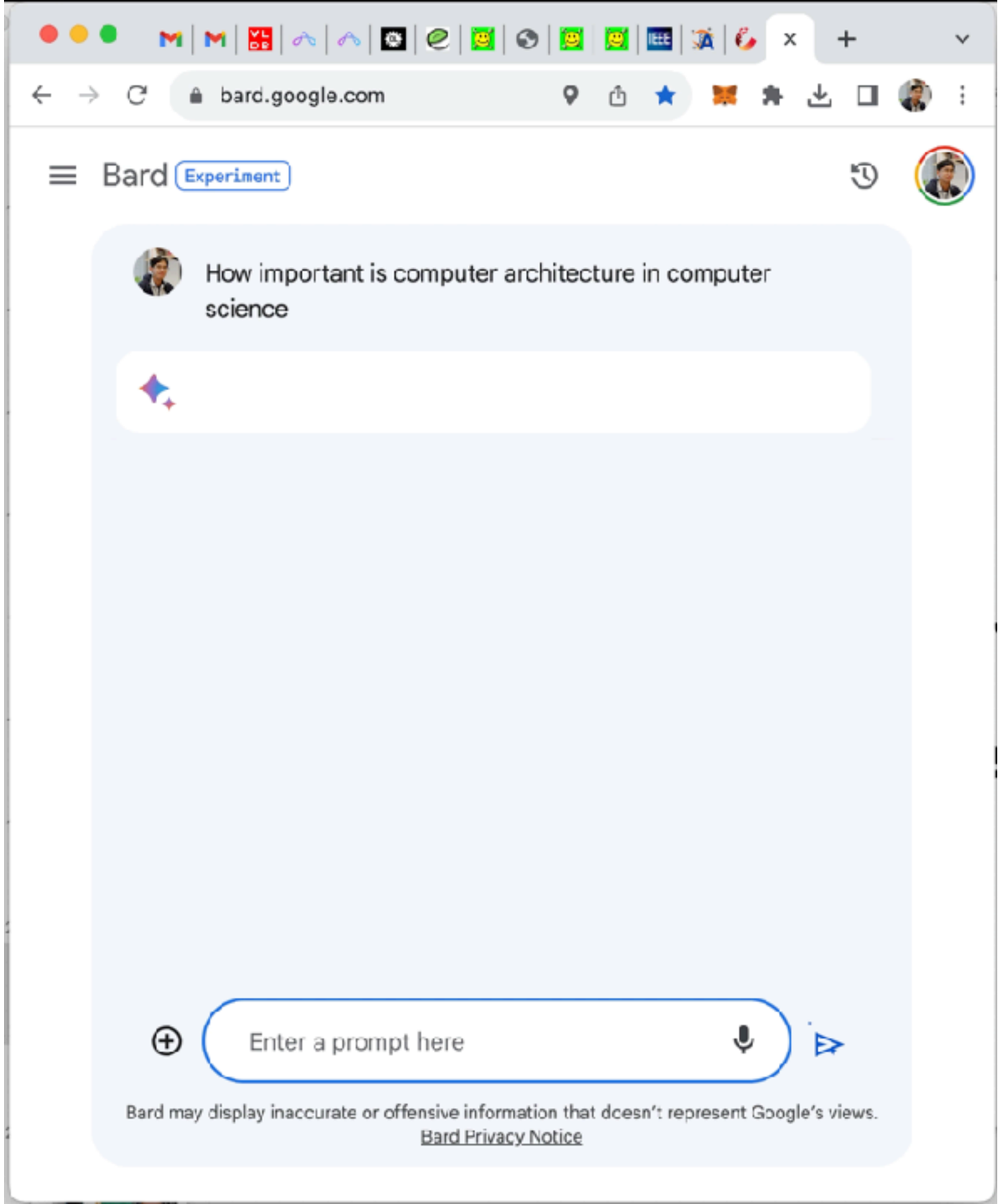
User Interfaces

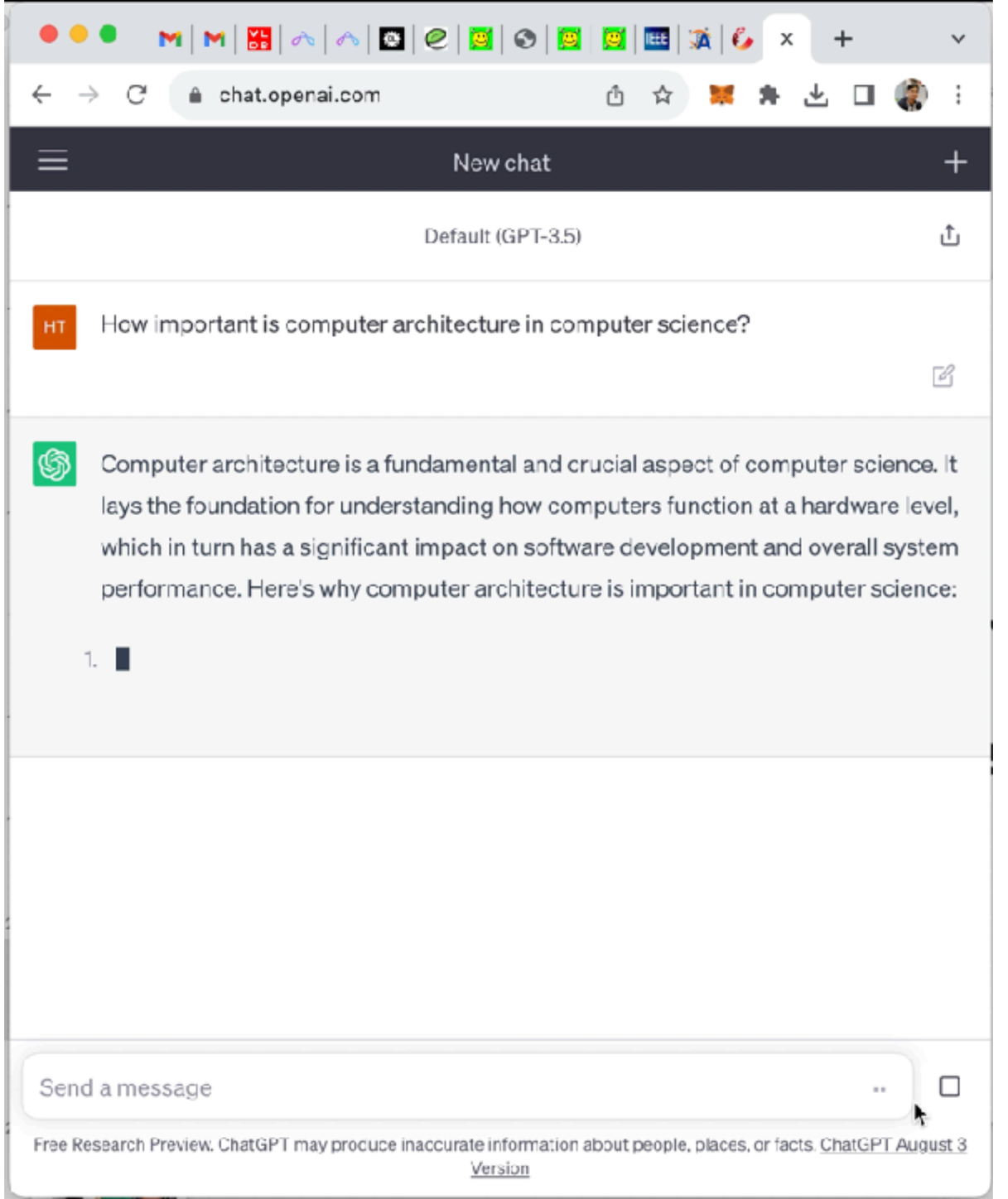
What's computer architecture?



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

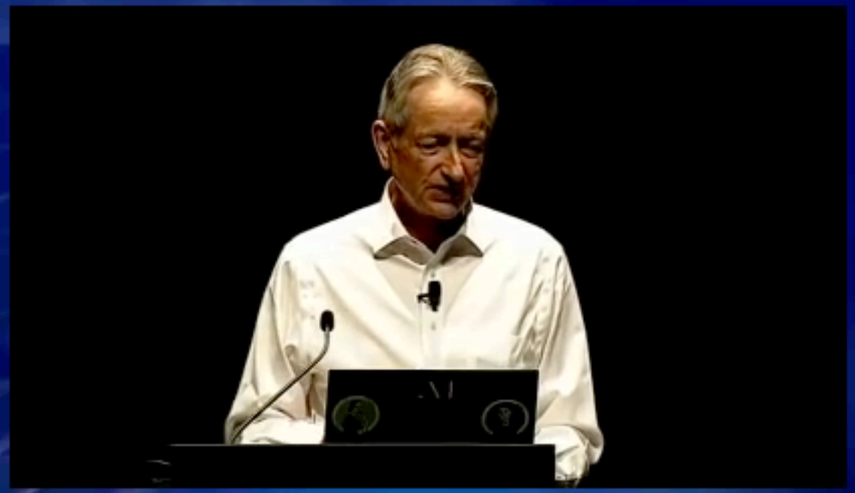
Computer Architects — Hidden Figures


bard.google.com
Bard
Experiment
How important is computer architecture in computer science
Enter a prompt here
Bard may display inaccurate or offensive information that doesn't represent Google's views.
Bard Privacy Notice

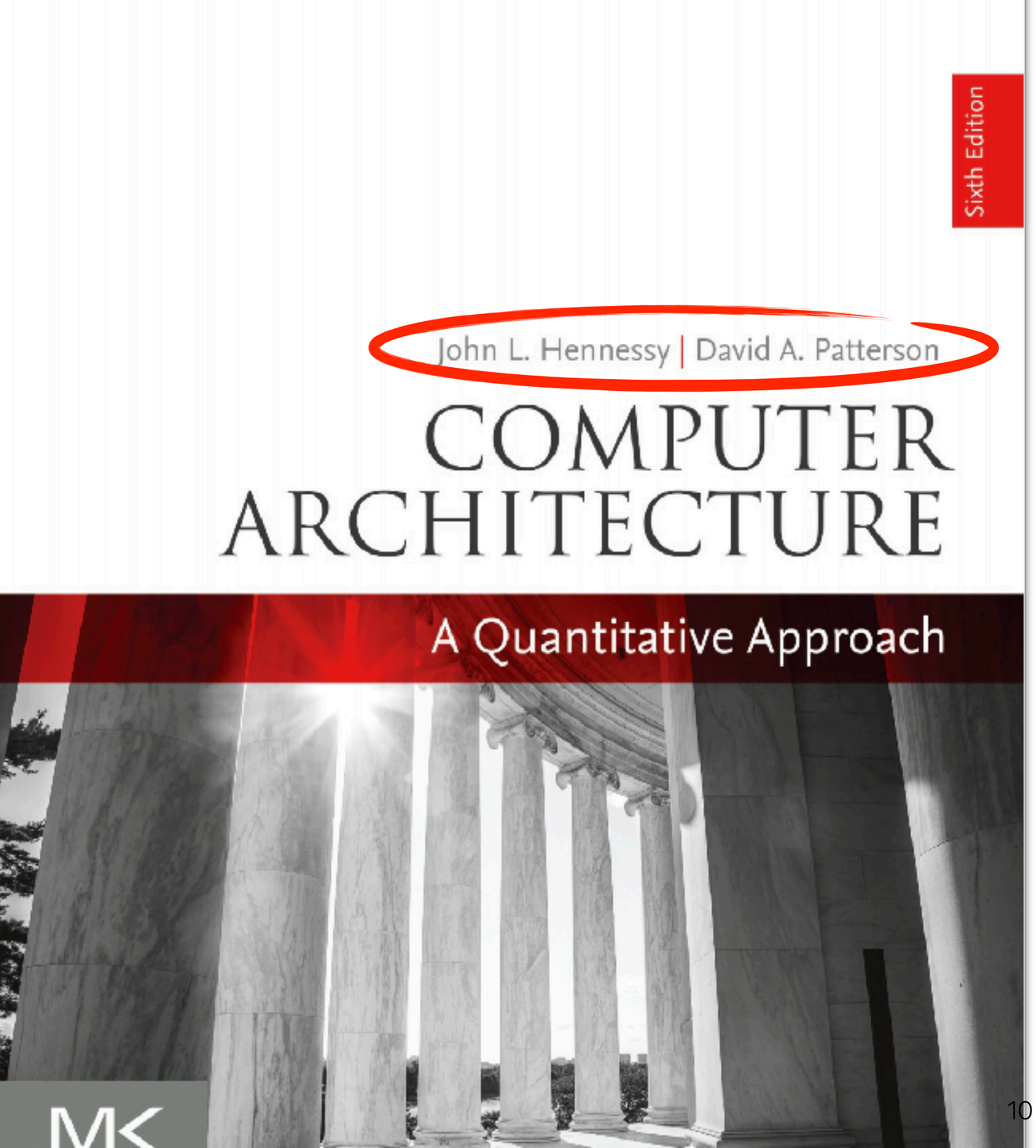

chat.openai.com
New chat
Default (GPT-3.5)
HT
How important is computer architecture in computer science?
Computer architecture is a fundamental and crucial aspect of computer science. It lays the foundation for understanding how computers function at a hardware level, which in turn has a significant impact on software development and overall system performance. Here's why computer architecture is important in computer science:
1.
Send a message
Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT August 3 Version

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)



acm FCRC



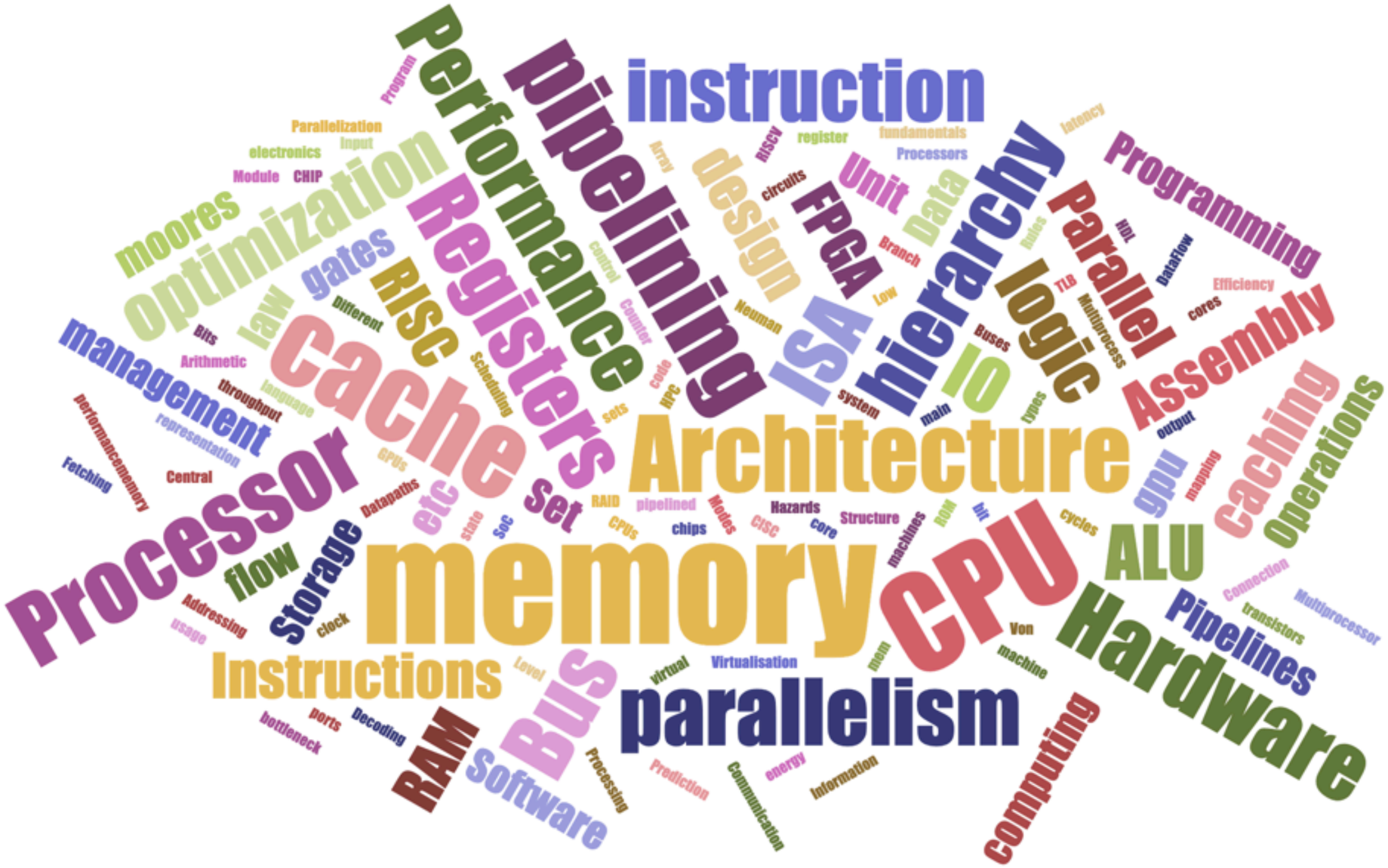
Computer Architecture

Enables

Deep Learning

How much do we understand “Computer Architectures” now

Your impression about Computer Architecture



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

Moore’s Law is current slowing/discontinuing

Programs with lower computational complexities are more efficient

Program performance scales with its main algorithm complexity

Algorithm complexity is less important if we have unlimited parallelism

Memory operations are more important for large matrix multiplications on modern computers

Efficient programs should leverage more “bit-wise” operations

Slowing down a running program can extend the battery life

GPUs are going to replace CPUs

AI Accelerators (e.g., TPUs) are less precise than CPUs

Why CS203 — Computers are not working as what “theorists” imagine

? ? ? ?

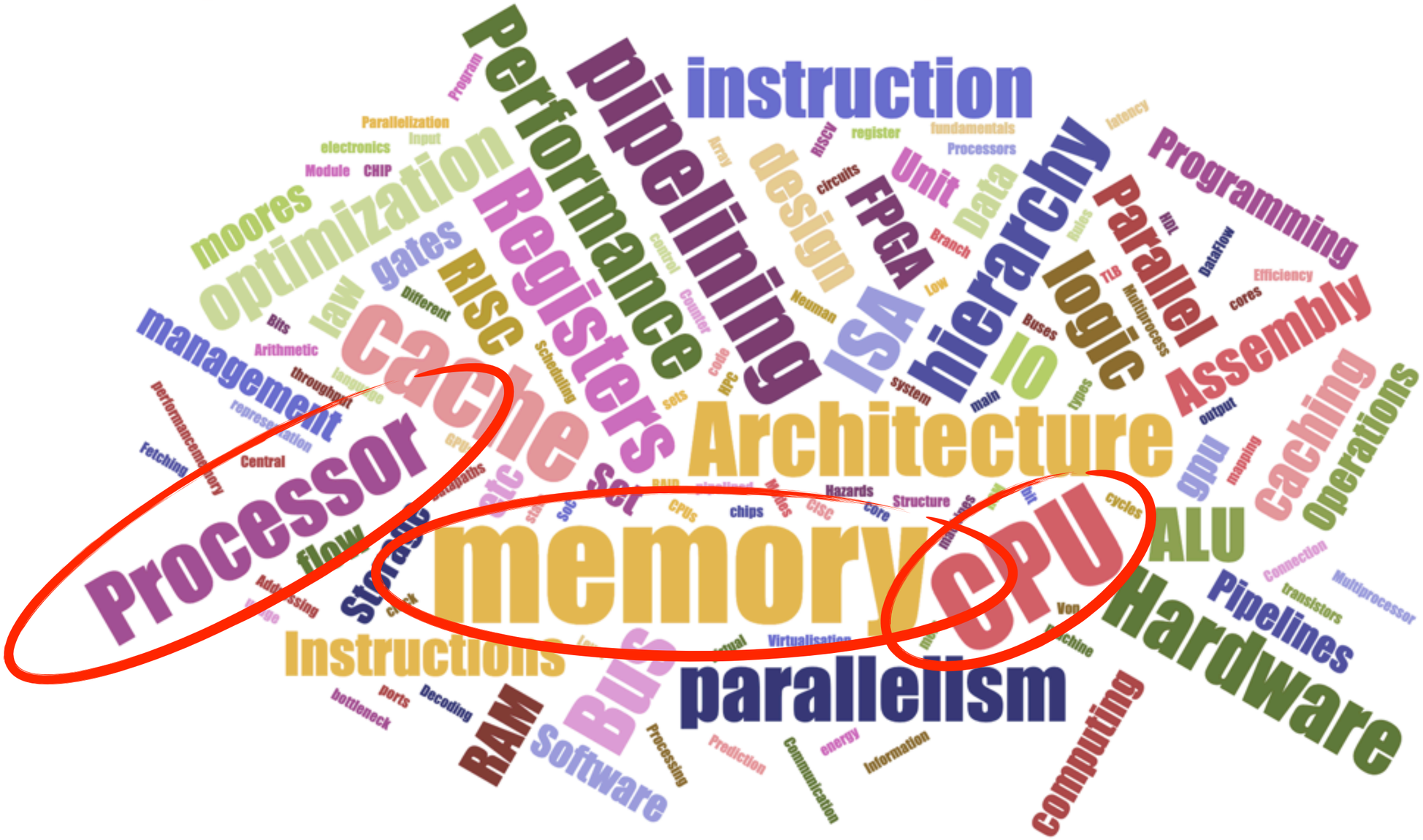
Thinking about the washlet

Or a Tesla

So you need to care “Computer Architecture”

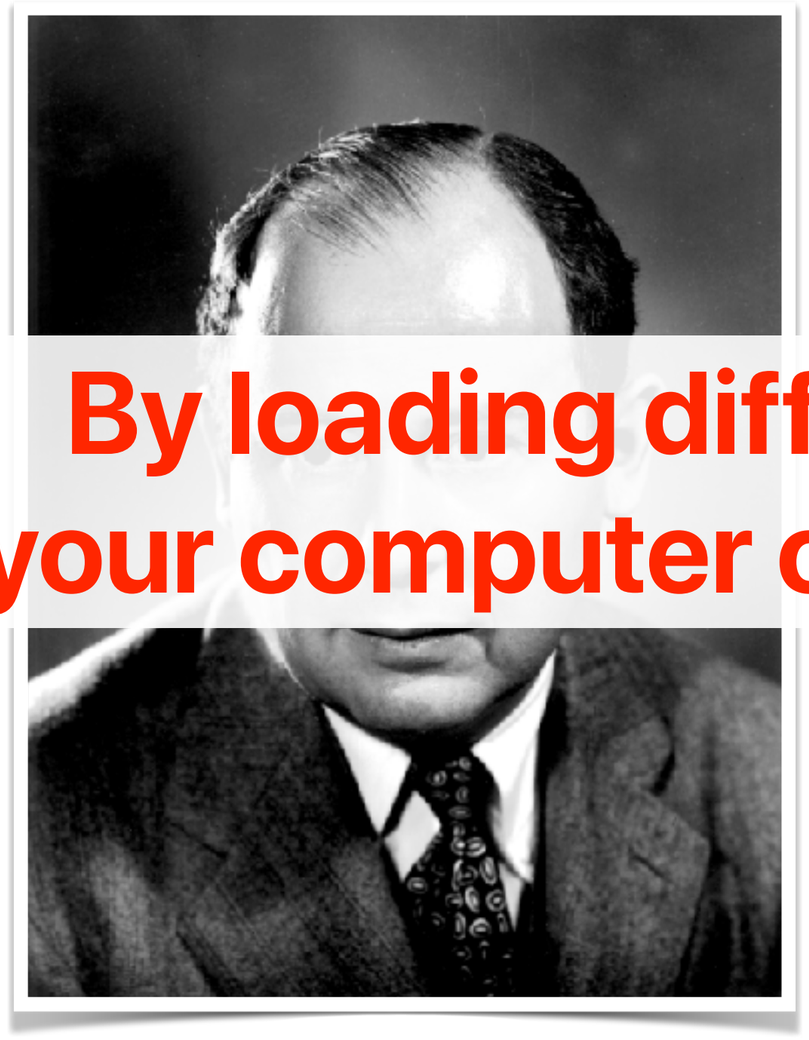
Processors and memories are essential for most modern general-purpose computers

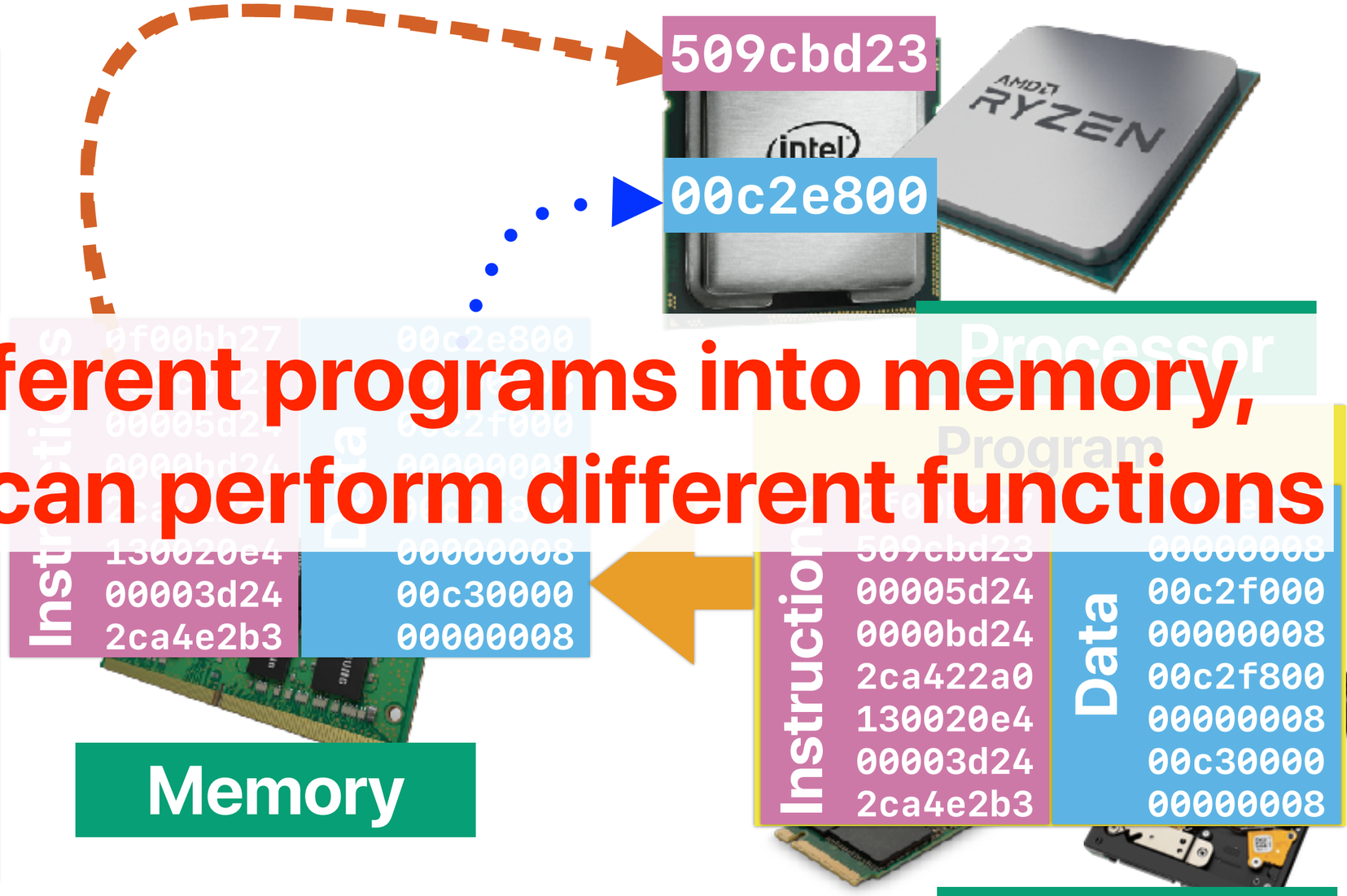
Your impression about Computer Architecture



Big Picture: the Von Neumann Architecture

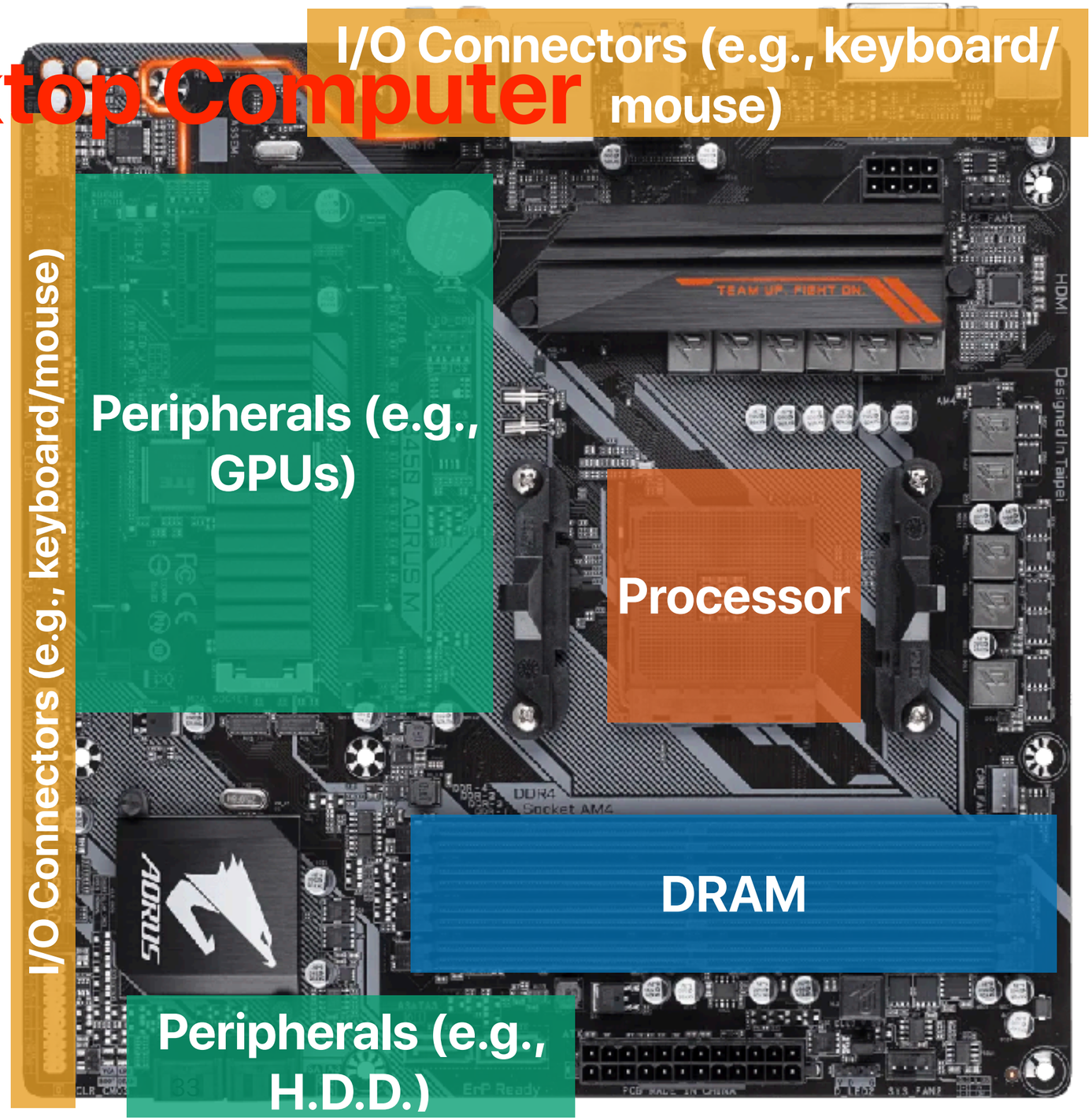
von Neumman Architecture



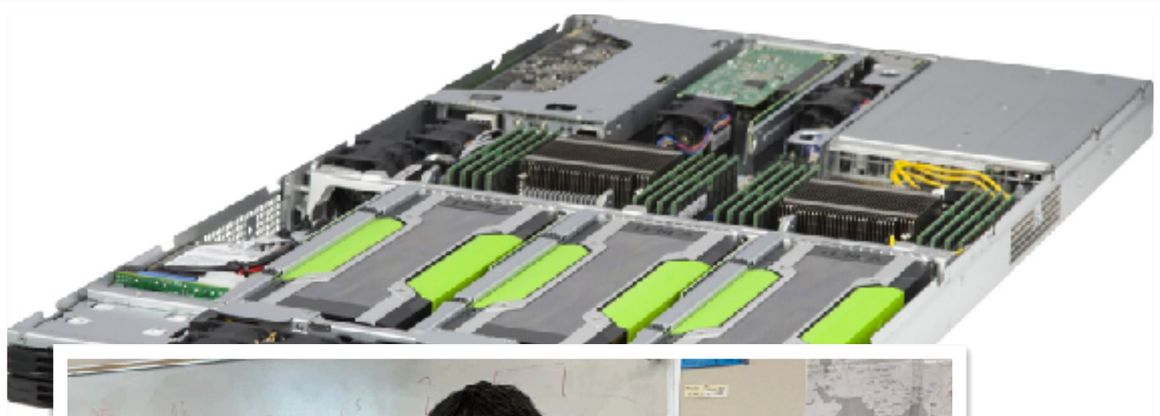


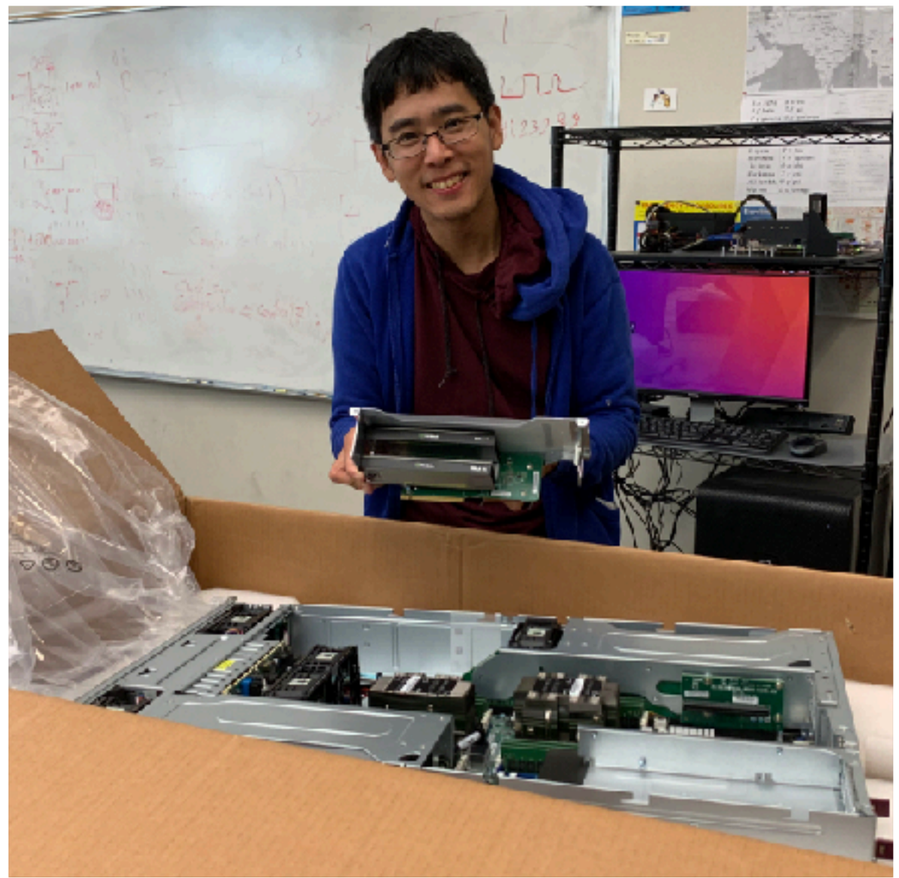
By loading different programs into memory, your computer can perform different functions

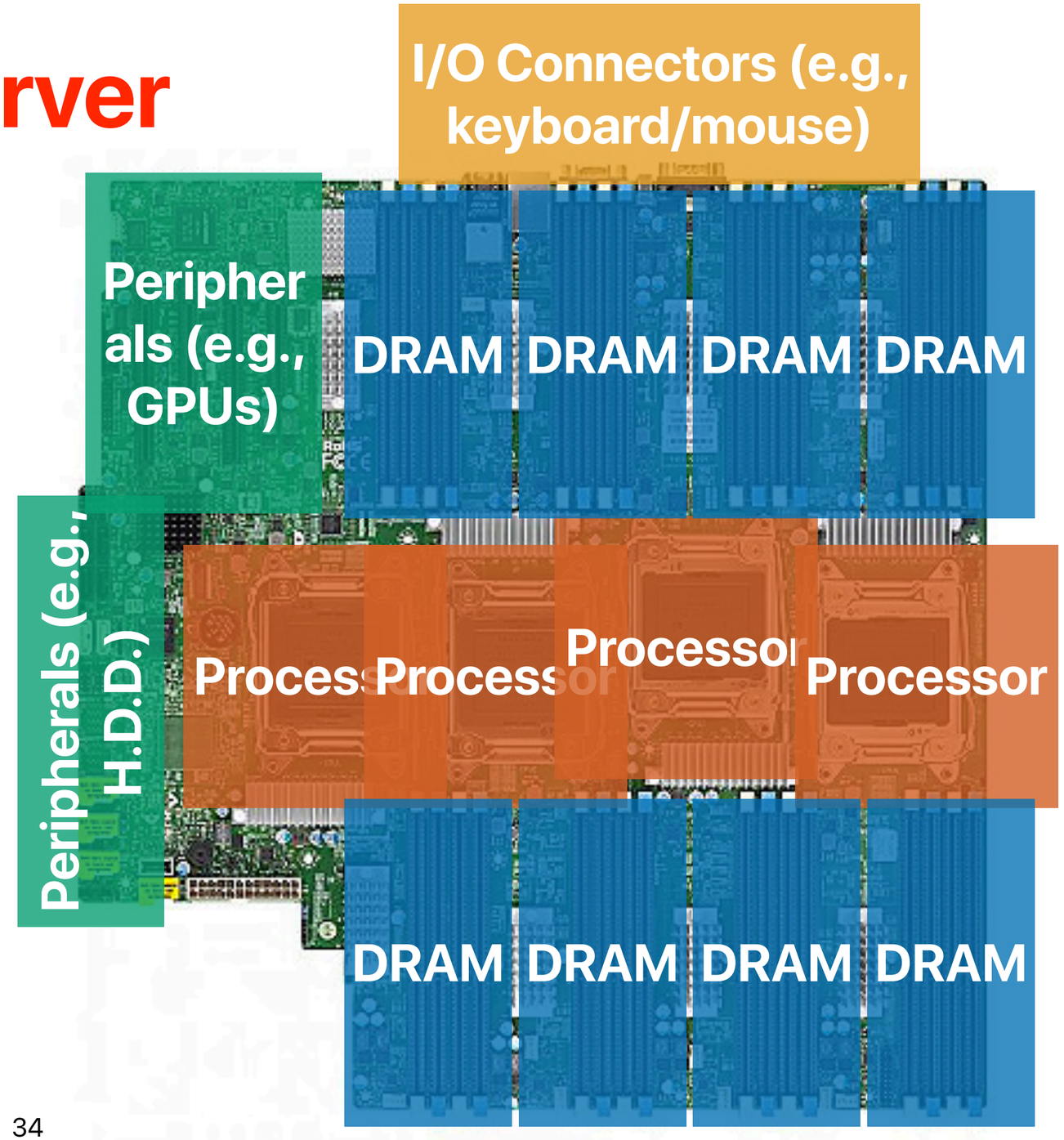
Desktop Computer



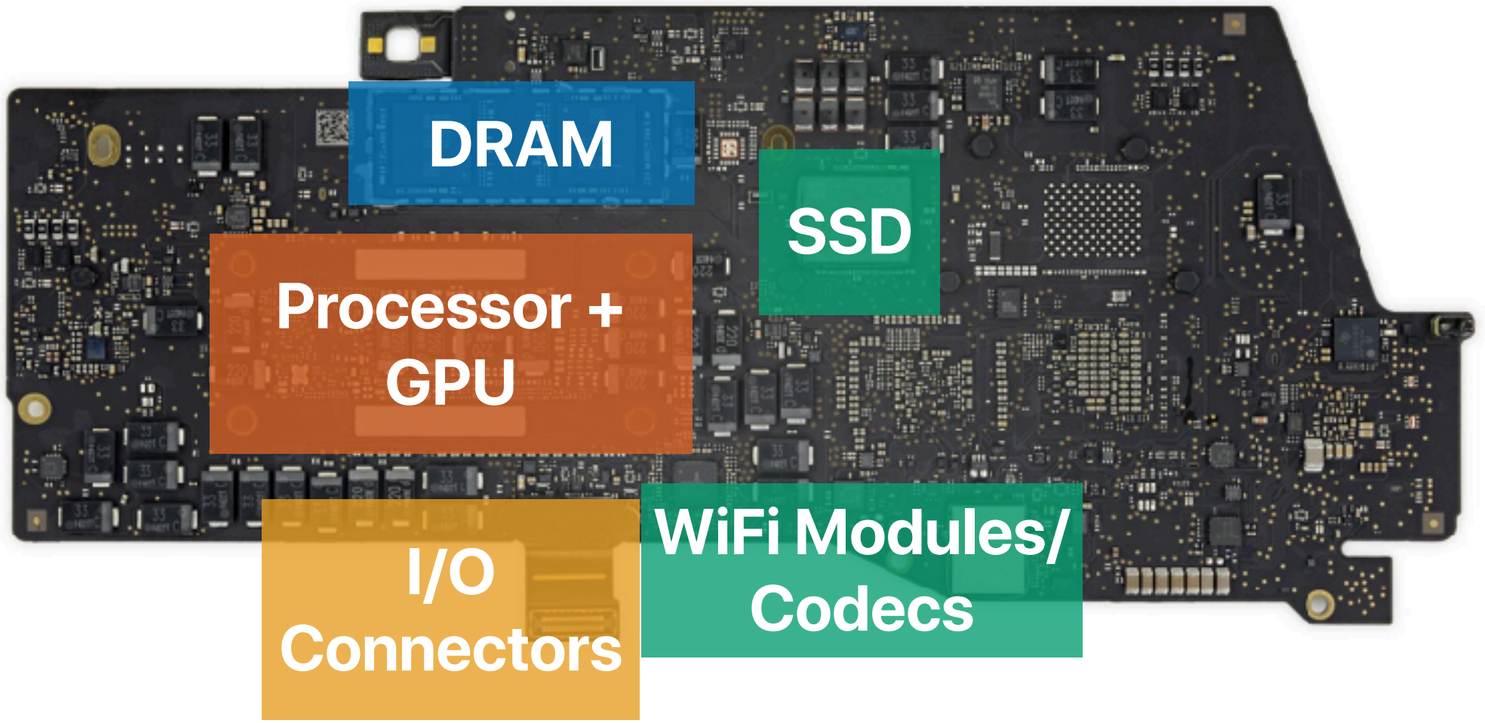
Server





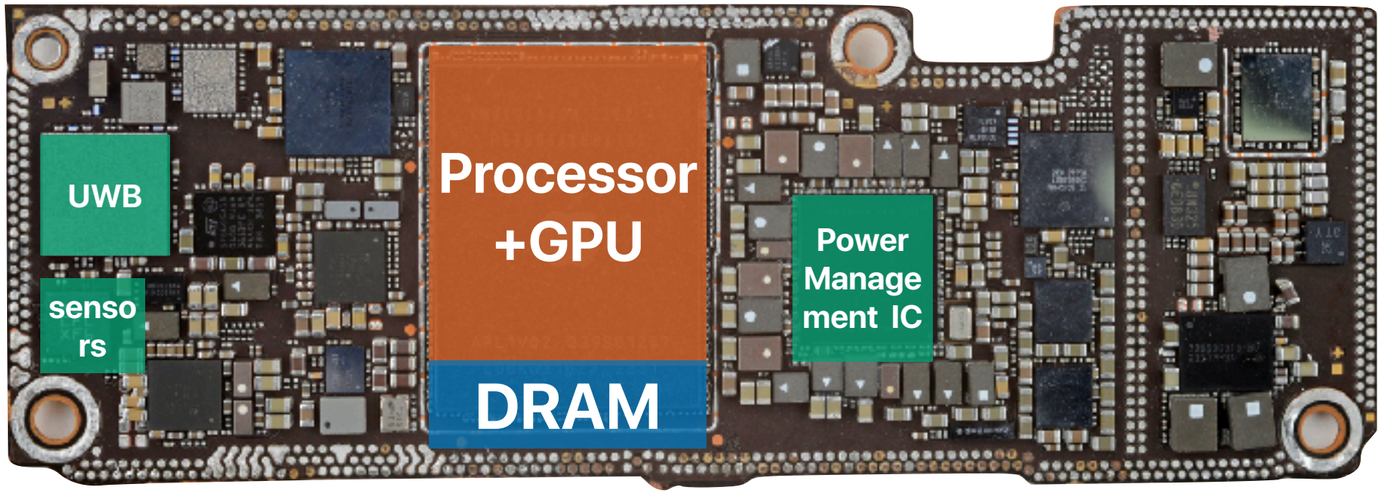

I/O Connectors (e.g., keyboard/mouse)
Peripherals (e.g., GPUs)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Processor
Processor
Processor
Processor
DRAM
DRAM
DRAM
DRAM

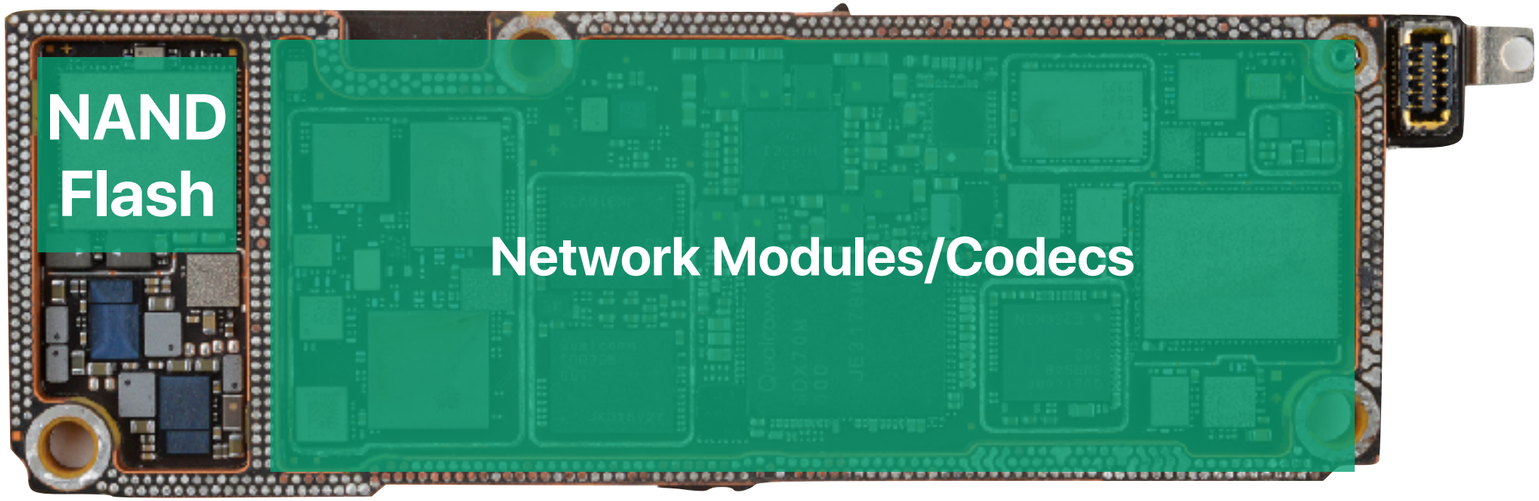
MacBook Pro 13″



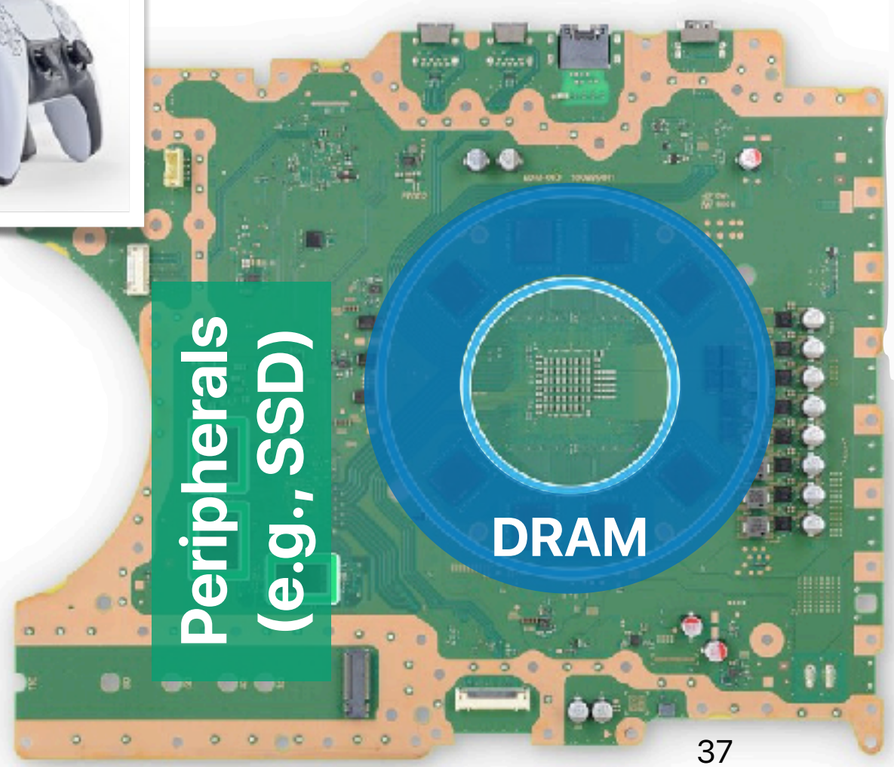
www.ifixit.com/News/82867/iphone-15-teardown-reveals-software-lockdown

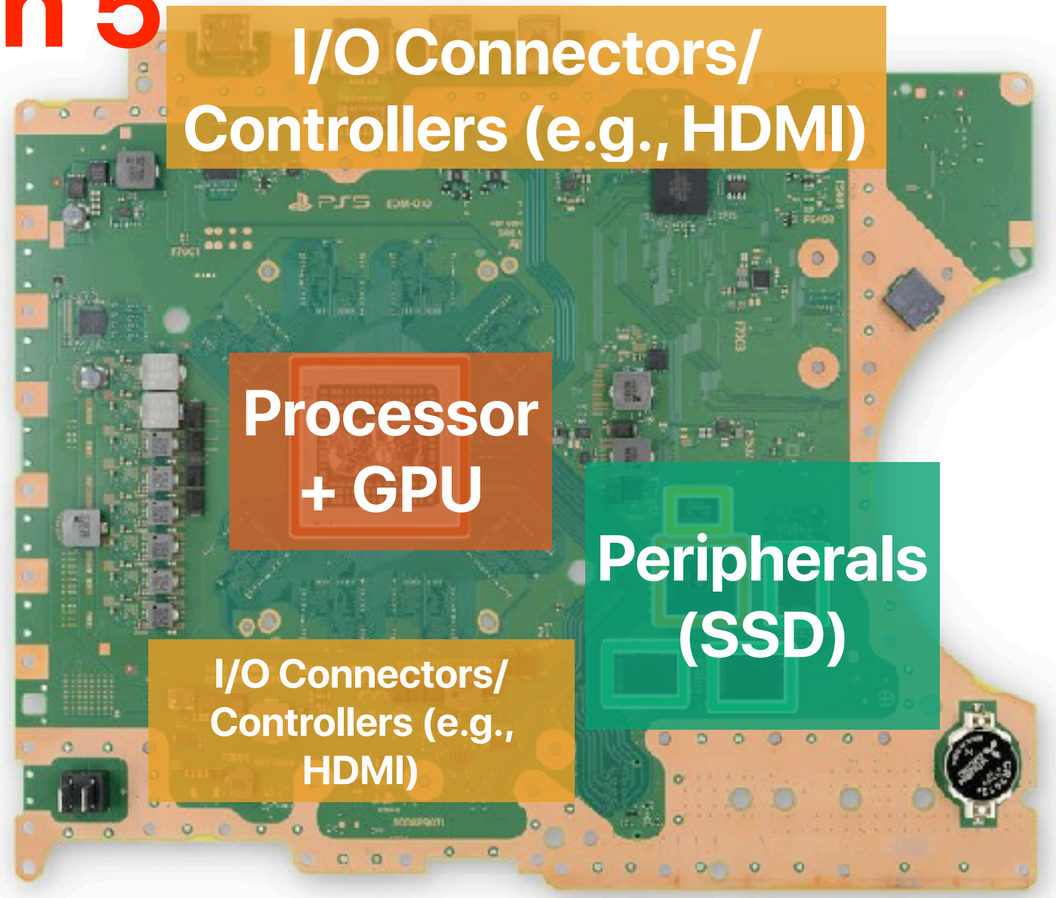
iPhone 15 Pro



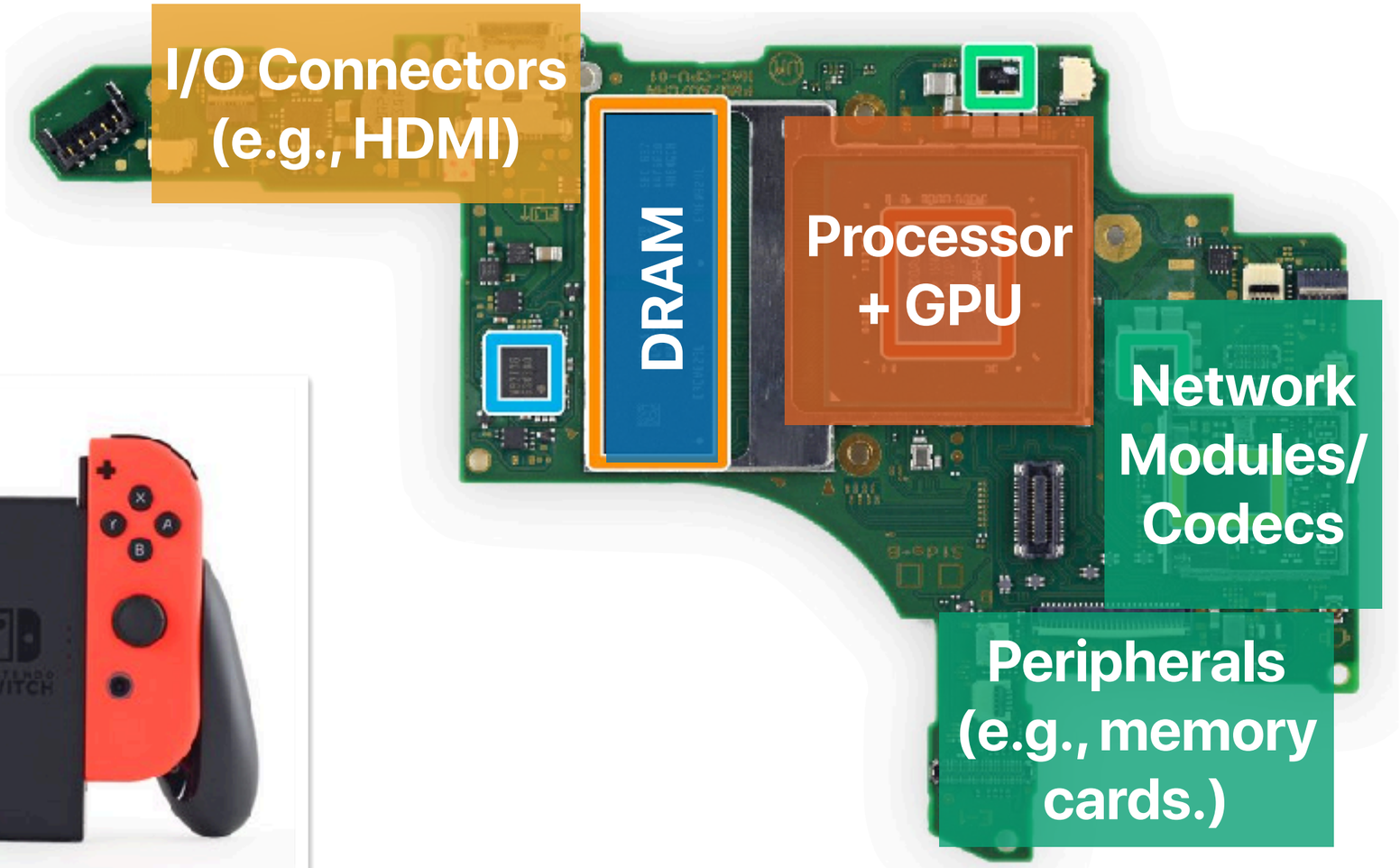


Play Station 5

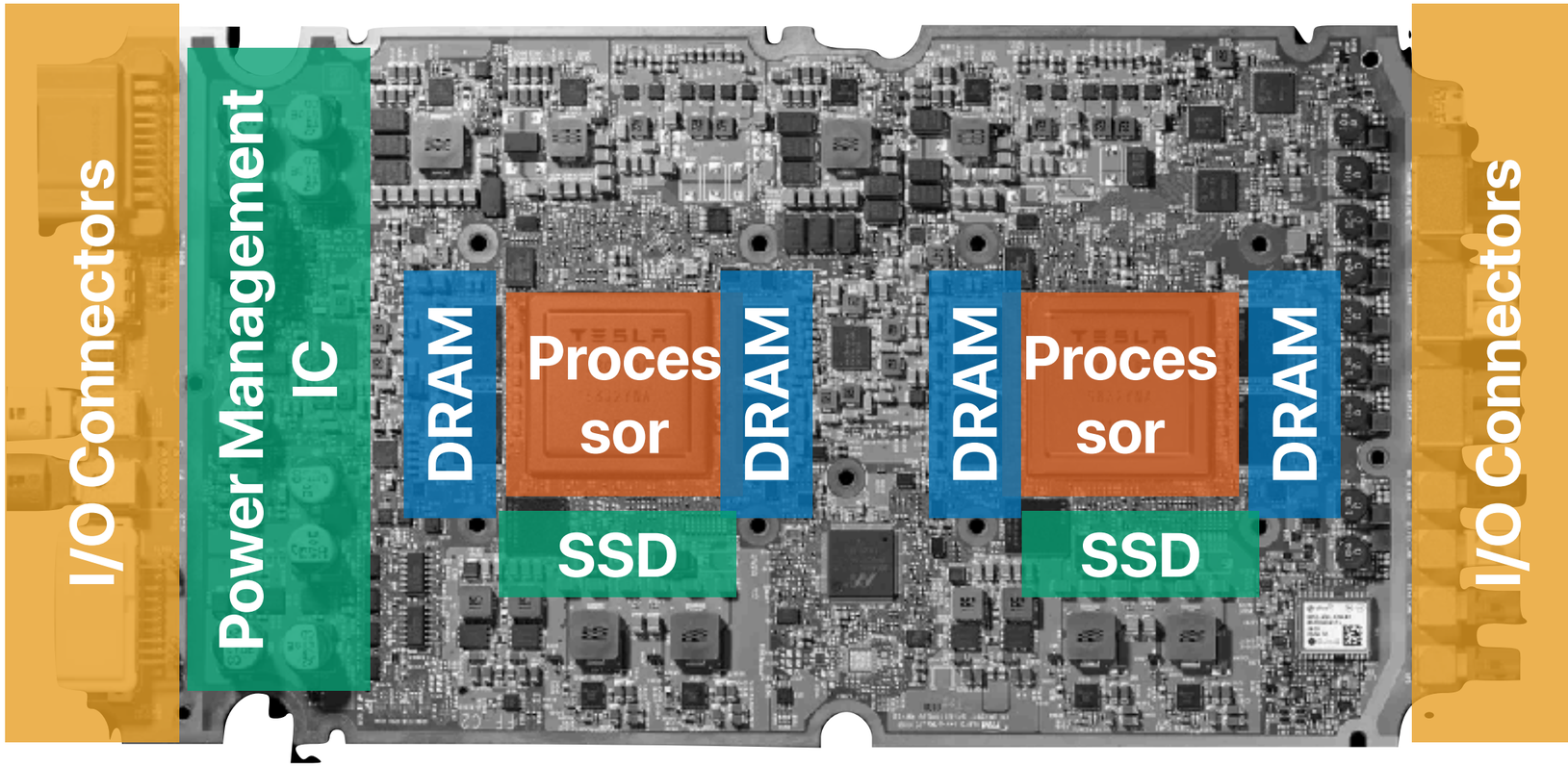




Nintendo Switch

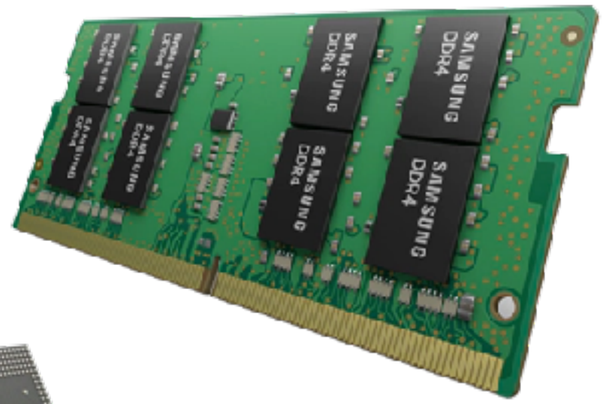


Tesla Model 3



Processors and memory modules are everywhere!

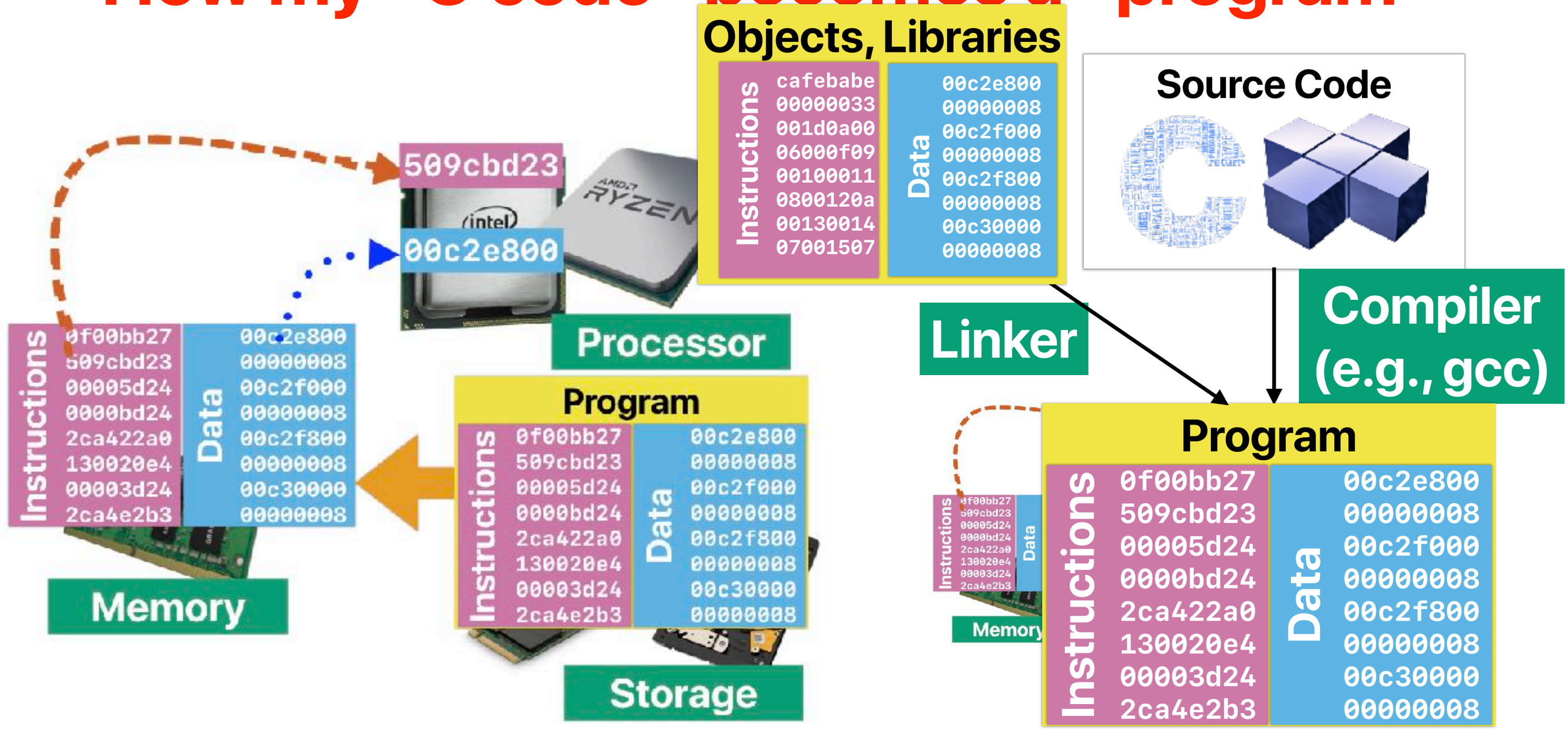
Processors



Memory

Processors & Memory are every where!

How my "C 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  ....UH...E......
 0010 45fc0000 0000c745 f8000000 00eb1e8b  E......E........
 0020 45f84898 488d1485 00000000 488d0500  E.H.H.......H...
 0030 0000008b 04020145 fc8345f8 01837df8  .......E..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 474e5500  ............GNU.
 0010 020000c0 04000000 03000000 00000000  ................
Contents of section .eh_frame:
 0000 14000000 00000000 017a5200 01781001  .........zR..x..
 0010 1b0c0708 90010000 1c000000 1c000000  ................
 0020 00000000 4a000000 00450e10 8602430d  ....J....E....C.
 0030 0602410c 07080000                    ..A.....
```

Instructions

Data

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 000000c7  ....UH...E......
 0010 45fc0000 0000c745 f8000000 00eb1e8b  E......E........
 0020 45f84898 488d1485 00000000 488d0500  E.H.H.......H...
 0030 0000008b 04020145 fc8345f8 01837df8  .......E..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 474e5500  ............GNU.
 0010 020000c0 04000000 03000000 00000000  ................
Contents of section .eh_frame:
 0000 14000000 00000000 017a5200 01781001  .........zR..x..
 0010 1b0c0708 90010000 1c000000 1c000000  ................
 0020 00000000 4a000000 00450e10 8602430d  ....J....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 01000000 02000000
 0030 03000000 04000000 05000000 06000000
 0040 07000000 08000000 09000000 0a000000
```

Data

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

control flow operations

logical operations

memory access

arithmetic operations

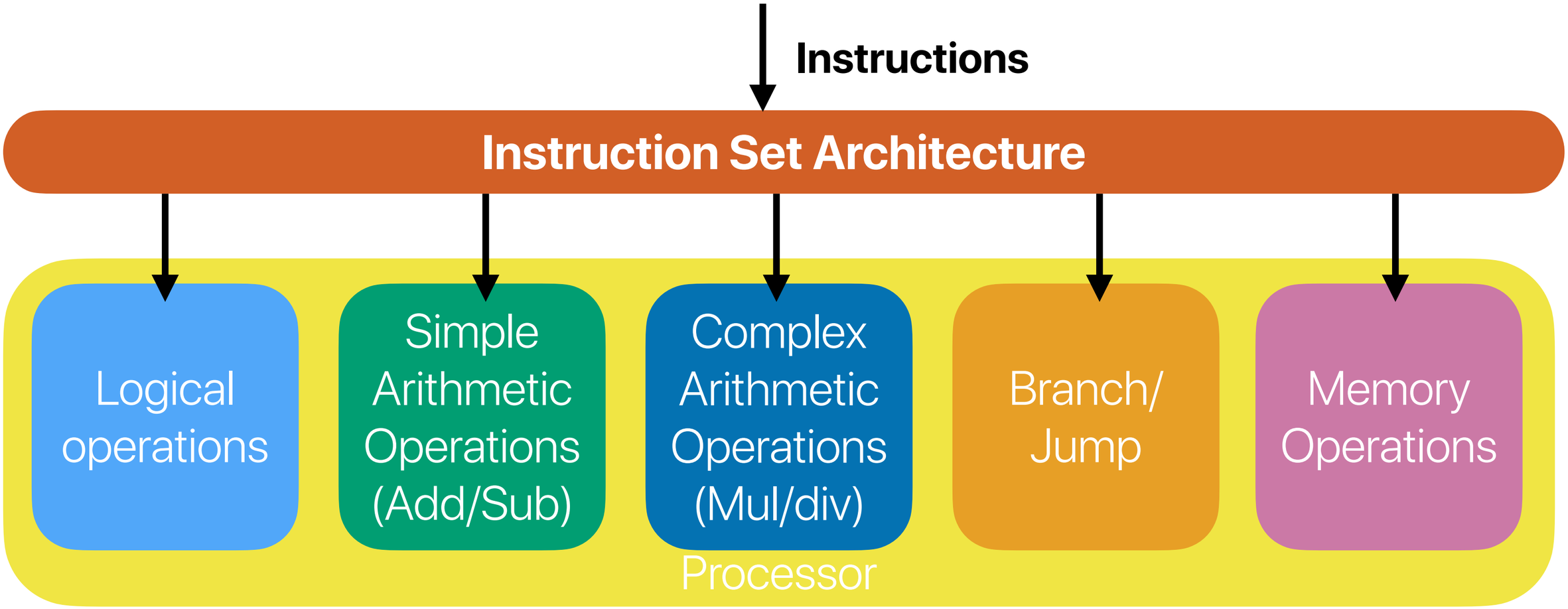
```
main:
.LFB0:
    endbr64
    pushq   %rbp
    movq %rsp, %rbp
    movl $0, -8(%rbp)
    movl $0, -4(%rbp)
    movl $0, -8(%rbp)
    jmp  .L2
.L3:
    movl -8(%rbp), %eax
    cltq
    leaq 0(,%rax,4), %rdx
    leaq A(%rip), %rax
    movl (%rdx,%rax),
%eax
    addl %eax, -4(%rbp)
    addl $1, -8(%rbp)
.L2:
    cmpl $19, -8(%rbp)
    jle  .L3
    movl $0, %eax
    popq %rbp
    ret
```

Compiler

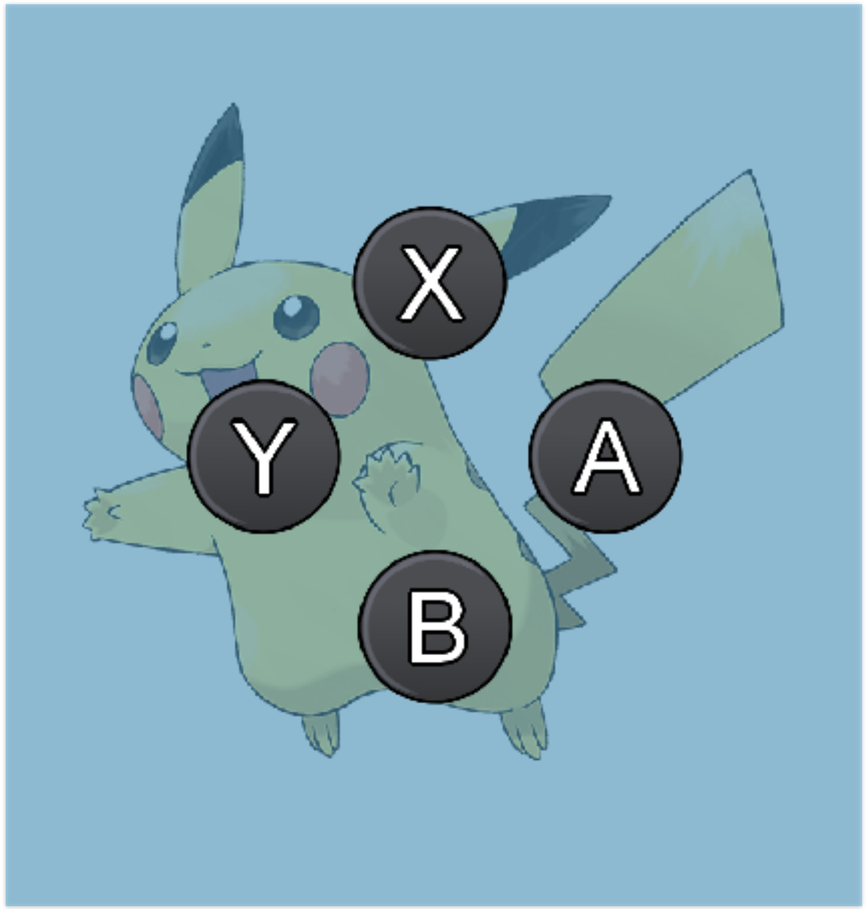
```
Contents of section .text:
 0000 f30f1efa 554889e5 c745f800 000000c7
 0010 45fc0000 0000c745 f8000000 00eb1e8b
 0020 45f84898 488d1425 00000000 488d0500
 0030 0000008b 04020145 fc8345f8 01837df8
 0040 137edcb8 00000000 5dc3
```

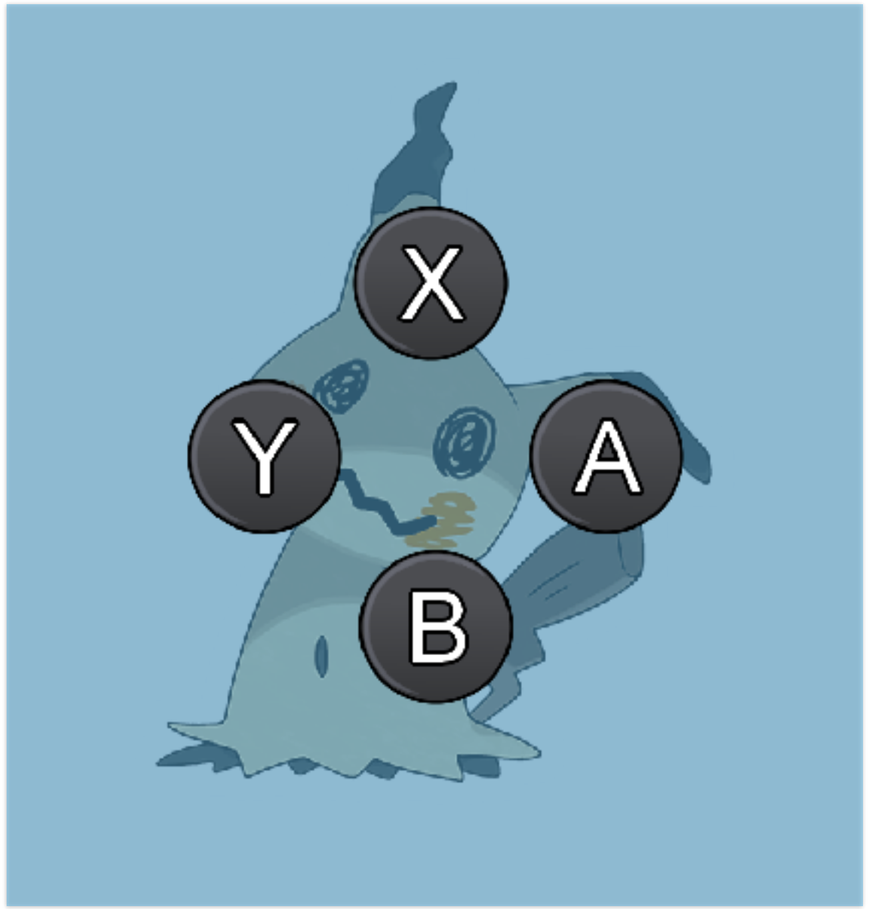
Instructions

Microprocessor — a collection of functional units

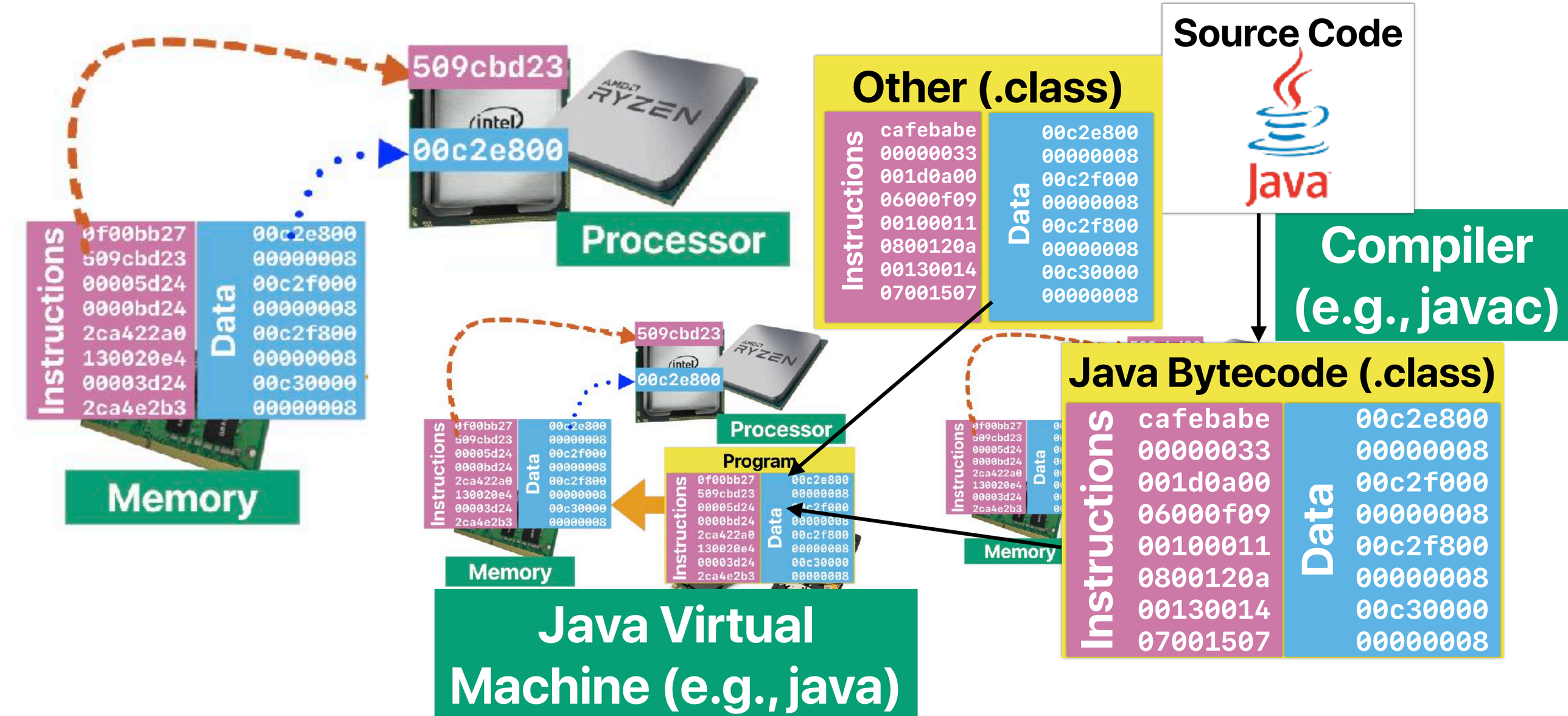


ISA — the “abstraction” of processor features

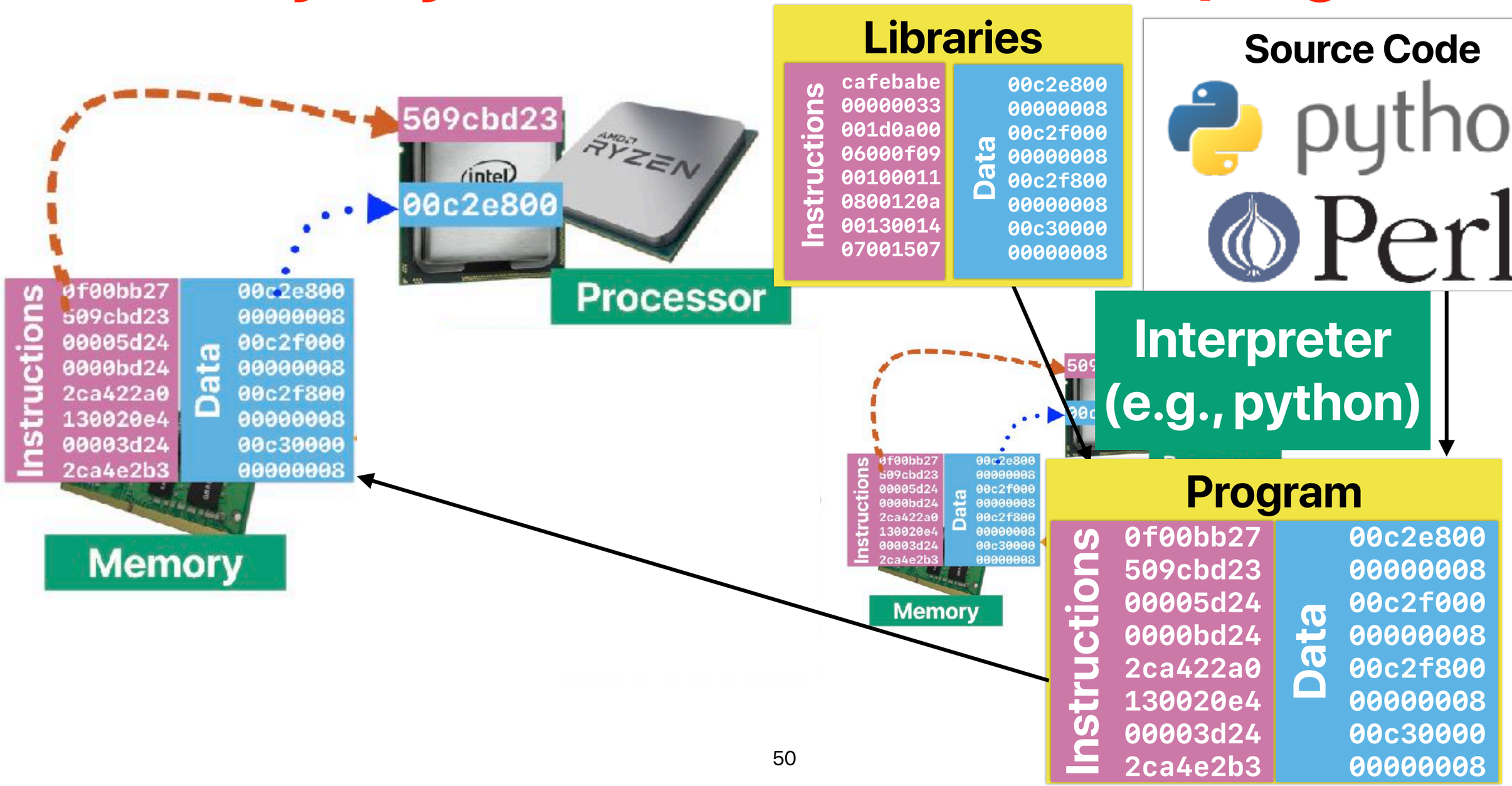




How my “Java code” becomes a “program”



How my "Python code" becomes a "program"



Programs with lower computational complexities are more efficient

Demo

```
if(option)
    std::sort(data, data + arraySize);

for (unsigned c = 0; c < arraySize*1000; ++c) {
    int t = std::rand();

    (data[c%arraySize] >= t)
            sum ++;
    }
}
```

$O(nlog_2n)$

$O(n)$

if option is set to 1: $O(nlog_2n)$

otherwise, O(n): $O(n)$

Algorithm complexity is less important if we have unlimited parallelism

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

Quick Sort

$O(nlog_2n)$

Bitonic Sort

$O(nlog_2^2n)$

```
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);
                }
            }
        }
    }
}
```

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];
    }
  }
}
```

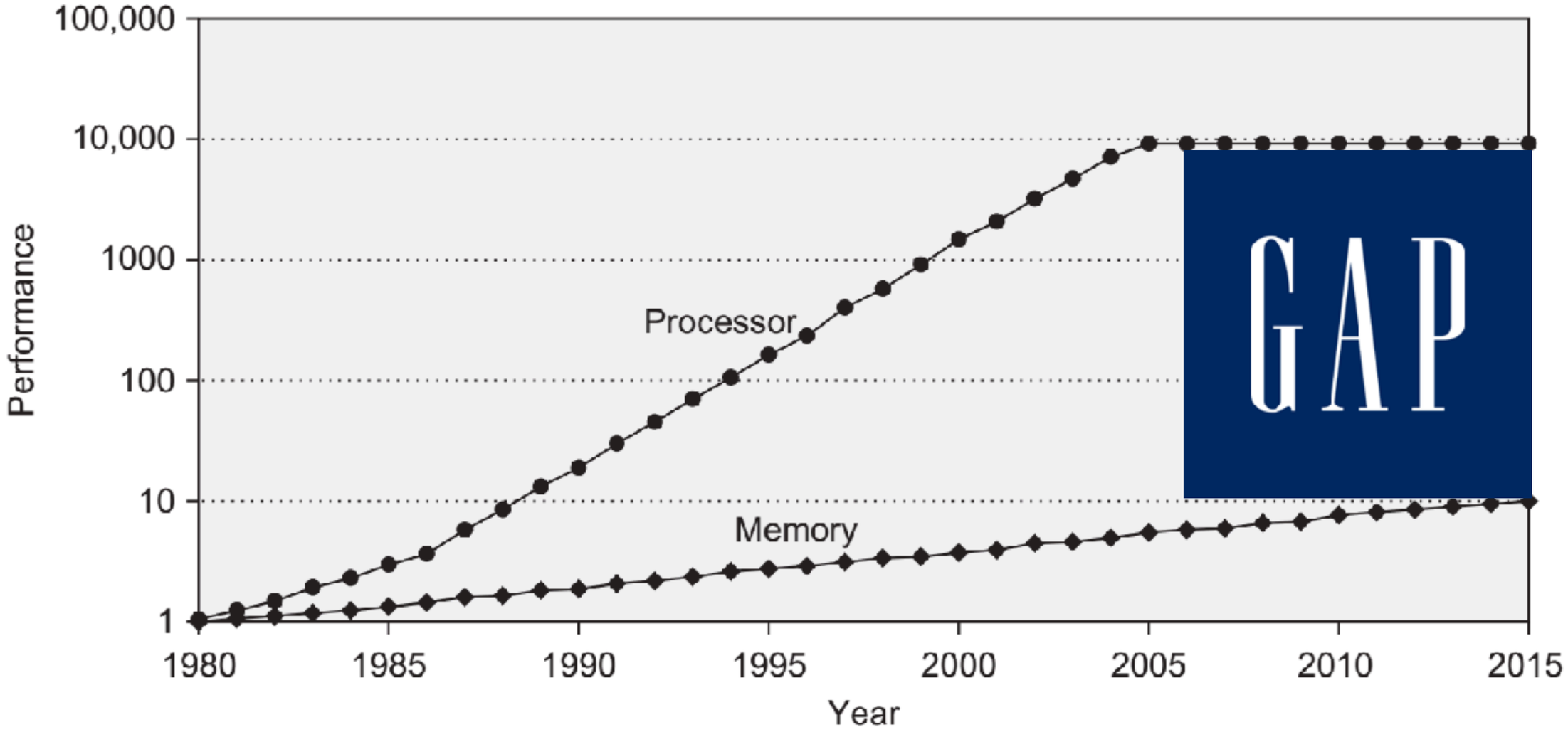
Algorithm class tells you it's $O(n^3)$

If n=1024, it takes about 2 sec

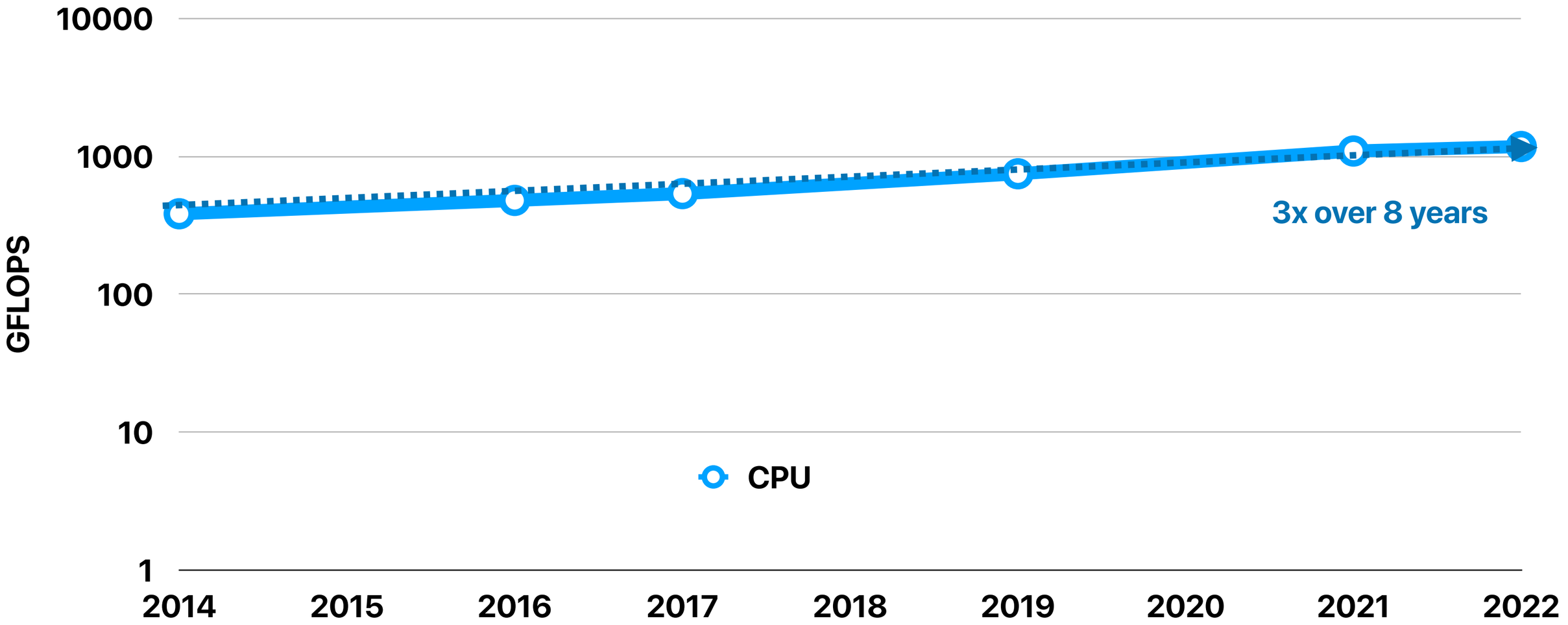
How long is it take when n=2048?

Memory operations are more important for large matrix multiplications on modern computers

Performance gap between Processor/Memory



Processor performance does not improve anymore



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 tronics today as well as any additi result in electronics functions suppli as irreducible units. These technolo to miniaturize electronics equipment creasingly complex electronic functi space with minimum weight. Sever evolved, including microassembly individual components, thin-film semiconductor integrated circuits.

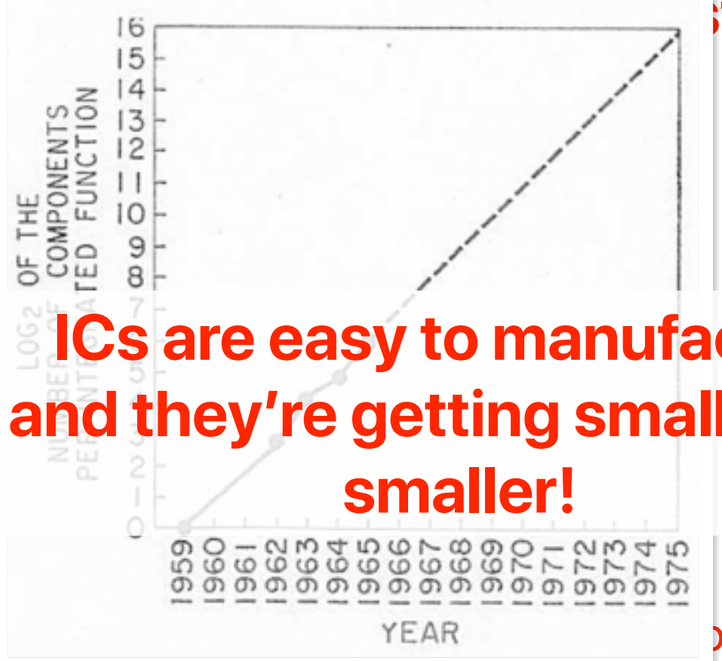
ICs are increasingly p

The establishment

Integrated electronics is established

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!

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 o

ICs are small

Reliability coun

In almost e demonstrated h 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

Linear circuitry

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

ICs are widely applicable

Heat problem

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

Heat is a solvable issue

Moore's Law is the most 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 design automation procedures could translate from any special engineering.

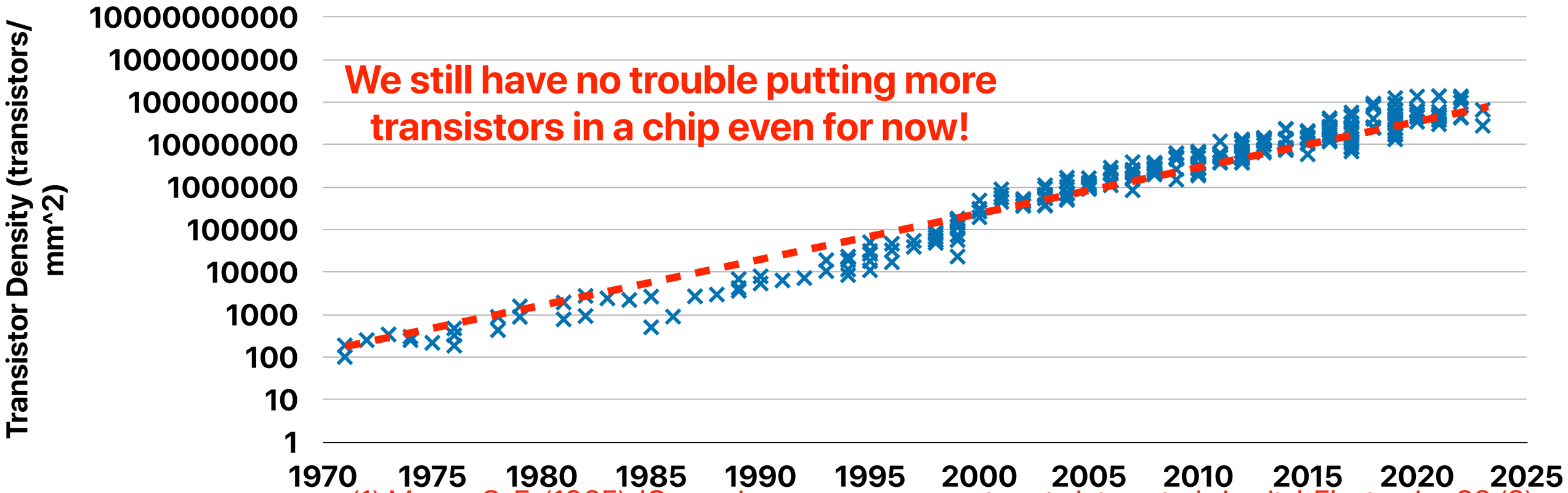
Designing ICs can be easy

(1) Mo omponents onto integrated circuits', Electronics 38 (8) .

Moore's Law is current slowing/discontinuing

Moore's Law[1]

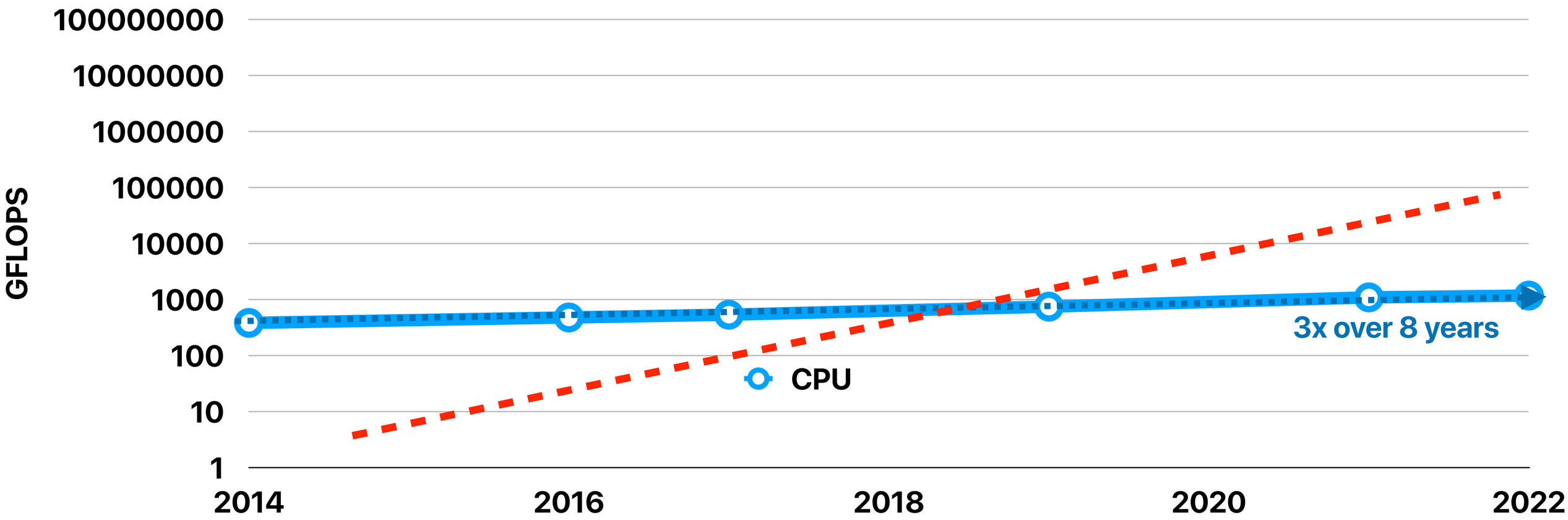
- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.
- Moore's Law "was" the most important driver for historic CPU performance gains



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

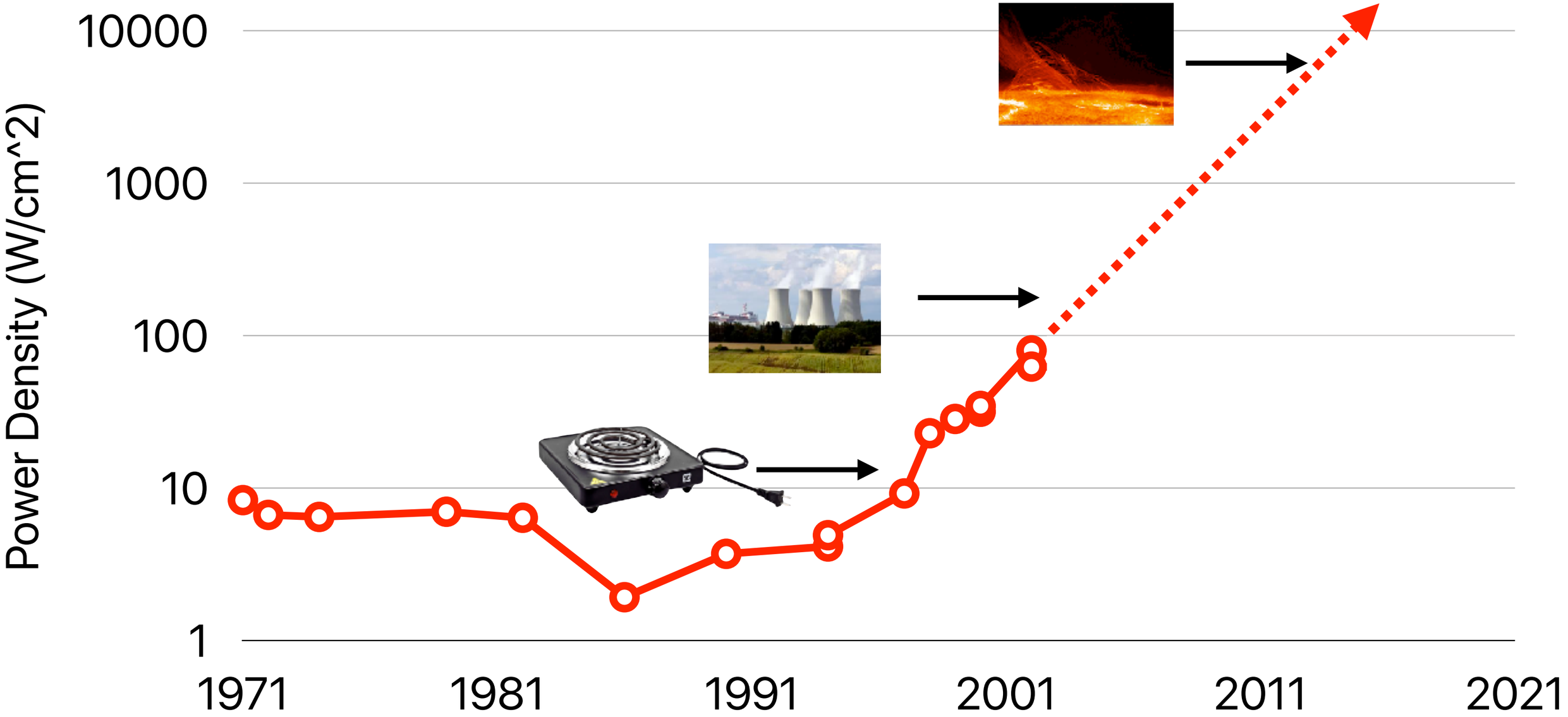
Plot based on https://en.wikipedia.org/wiki/Transistor_count by Hung-Wei Tseng

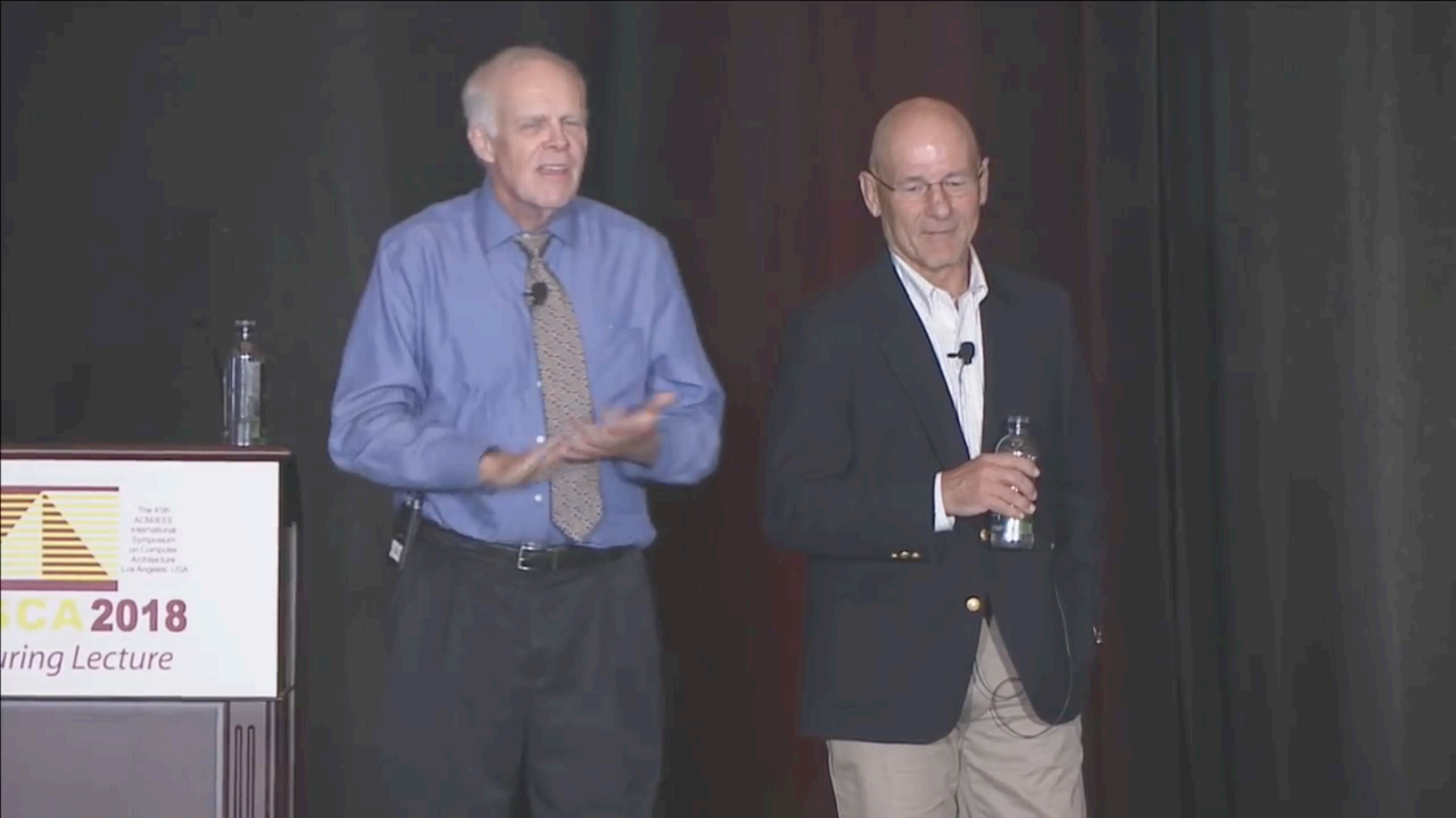
CPU Performance v.s. Moore's Law

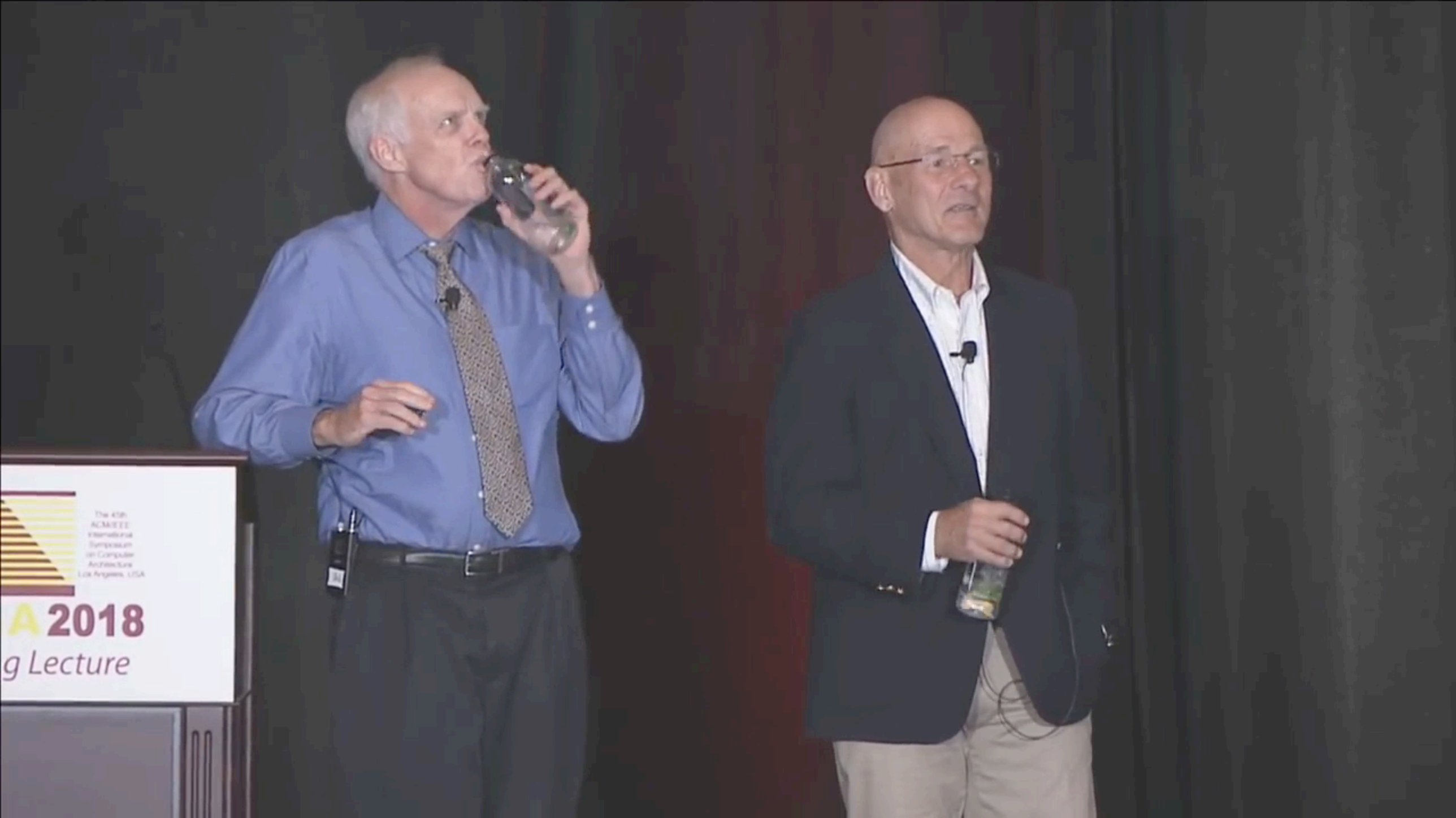


https://ourworldindata.org/grapher/artificial-intelligence-training-computation

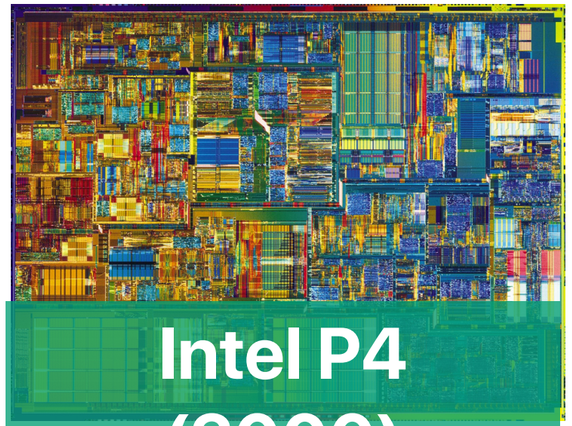
Power Density of Processors



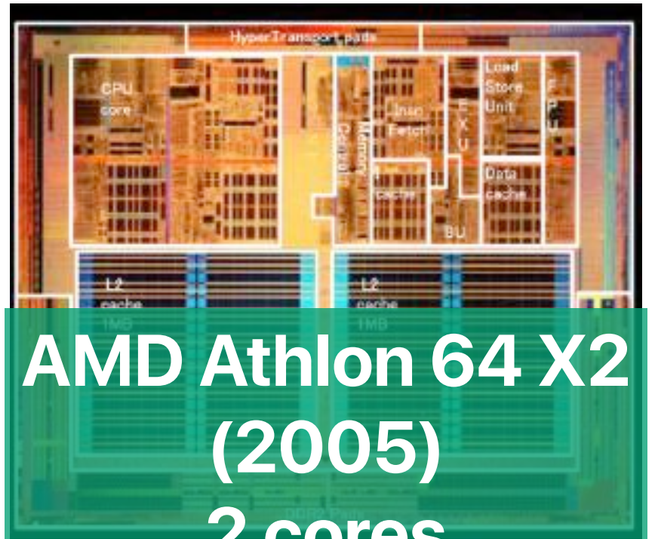

The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA
CA 2018
uring Lecture


The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA
A 2018
g Lecture

Multicore processors

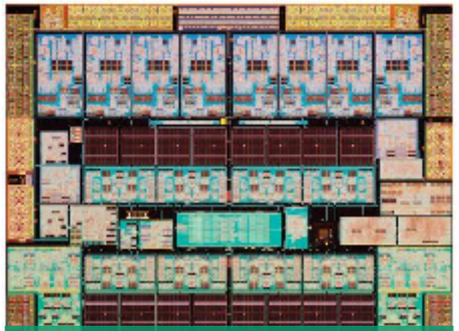


Intel P4
(2000)
1 core

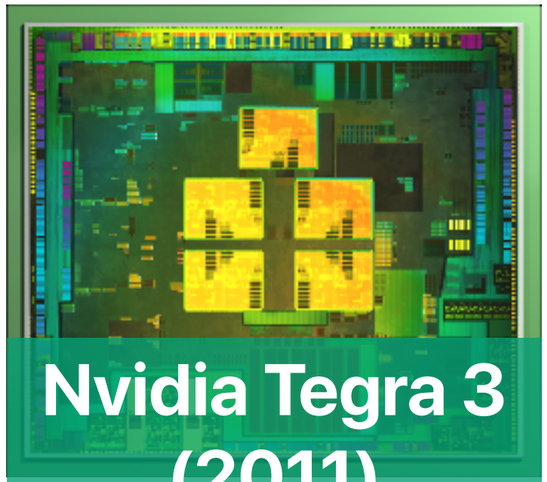


AMD Athlon 64 X2
(2005)
2 cores

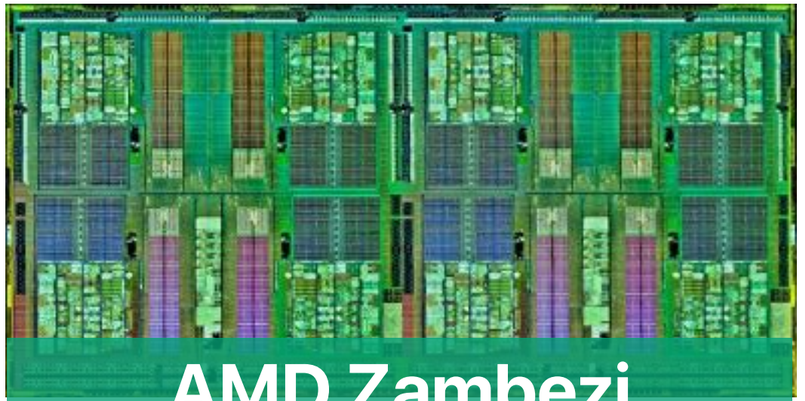
Intel Nahalem
(2010)
4 cores



SPARC T3
(2010)
16 cores

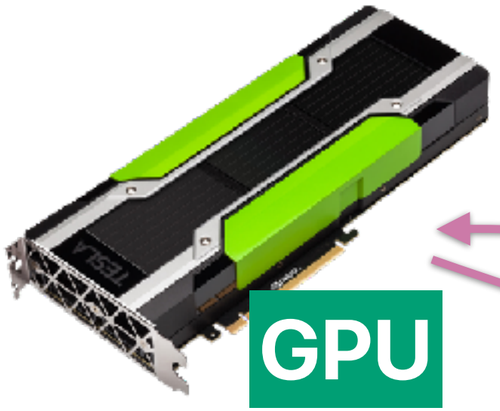


Nvidia Tegra 3
(2011)
5 cores



AMD Zambezi
(2011)
16 cores

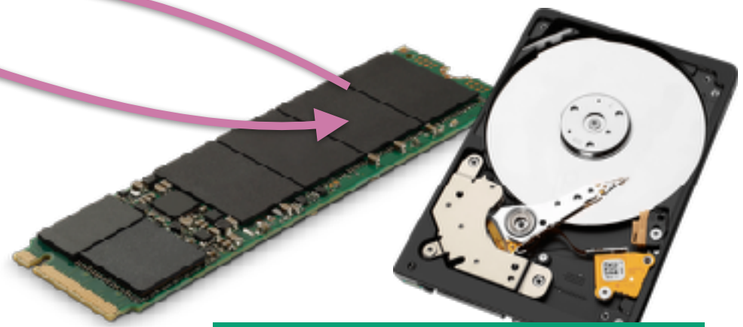
GPGPU Computer Architecture



GPU

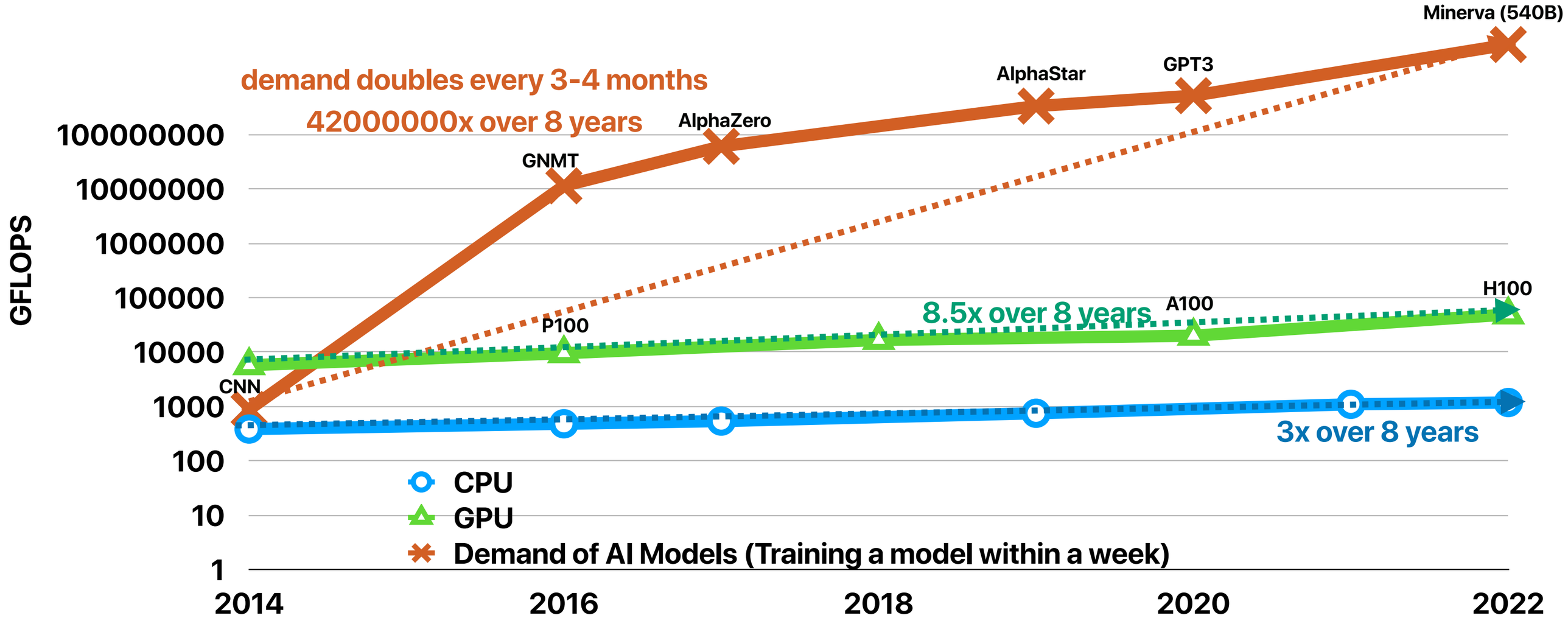
Processor

Memory



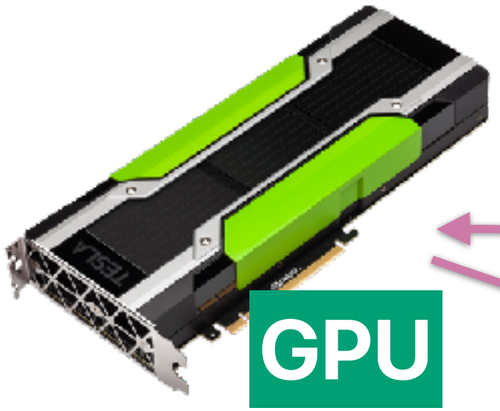
Storage

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



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

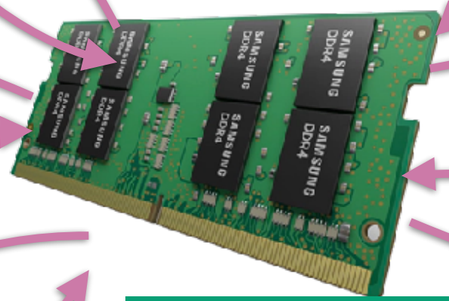
Heterogeneous Computer Architecture



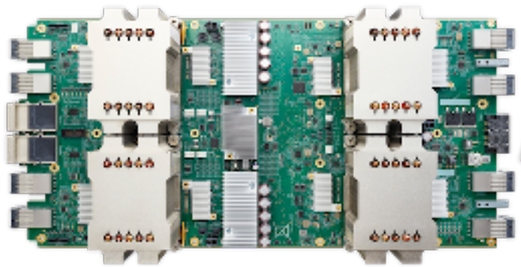
GPU

FPGA

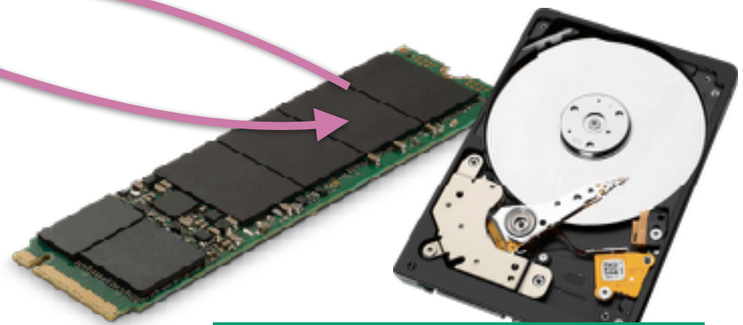
Processor



Memory

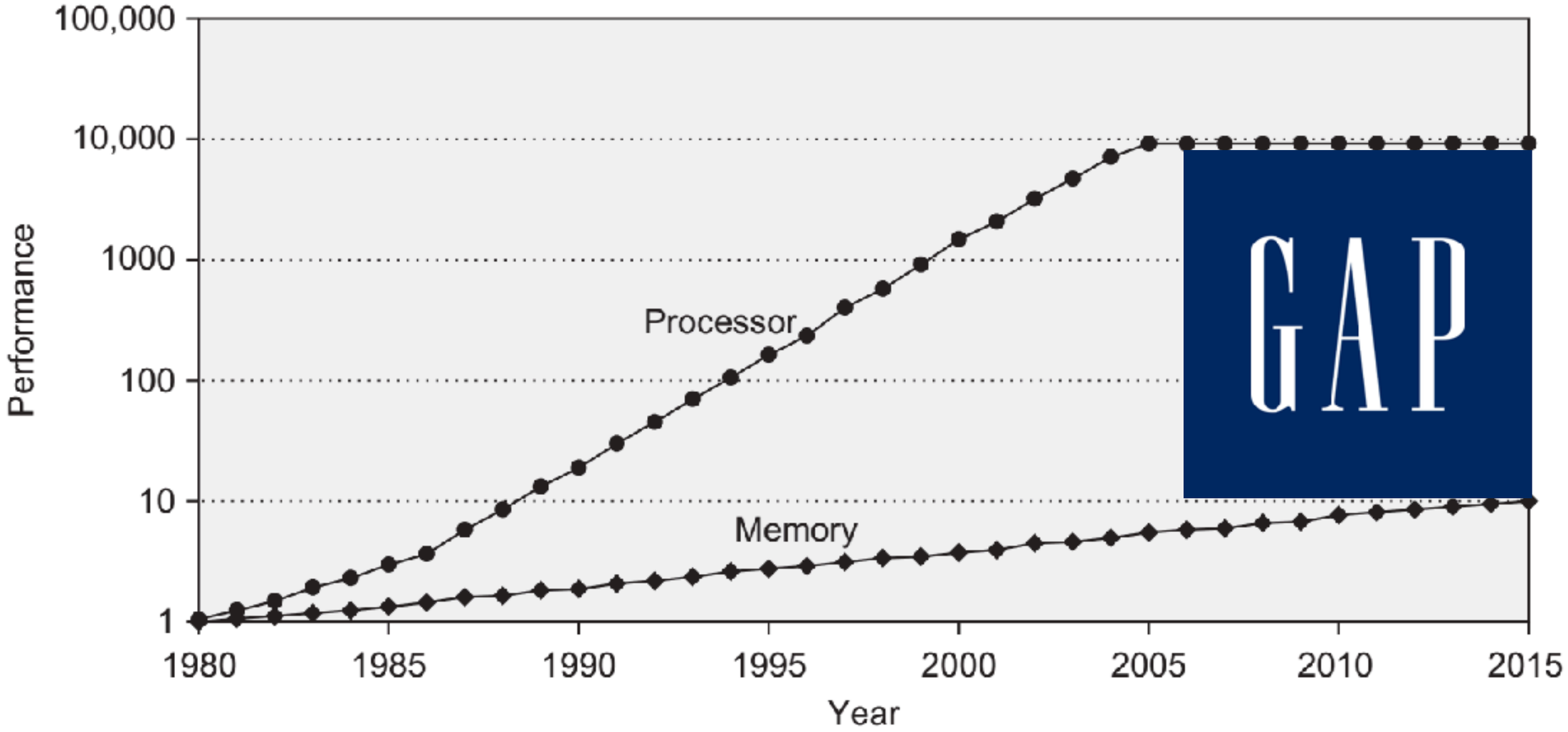


TPU

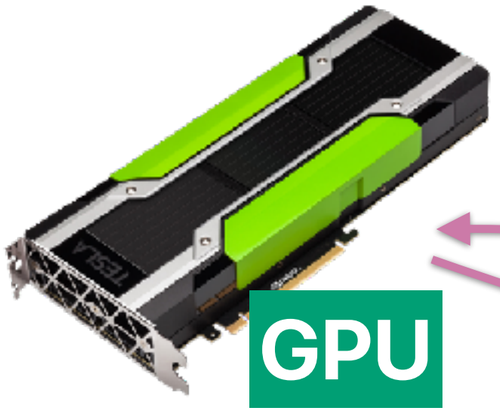


Storage

Performance gap between Processor/Memory



Heterogeneous Computer Architecture



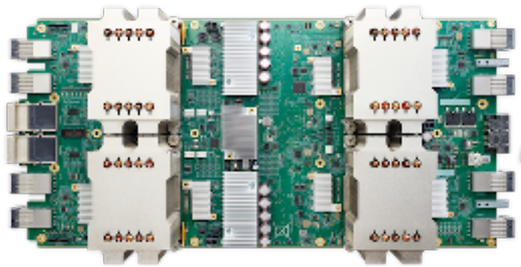
GPU

FPGA

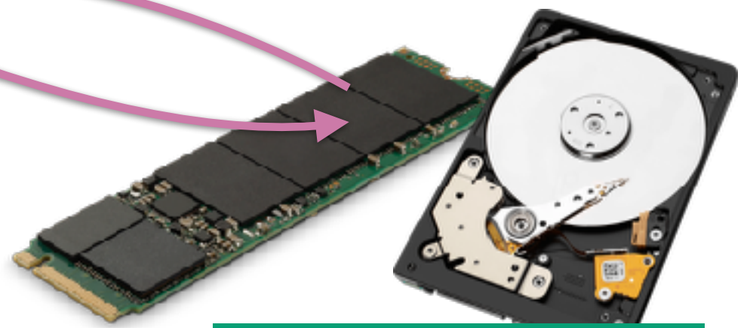
Processor

Data movements

Slow memory



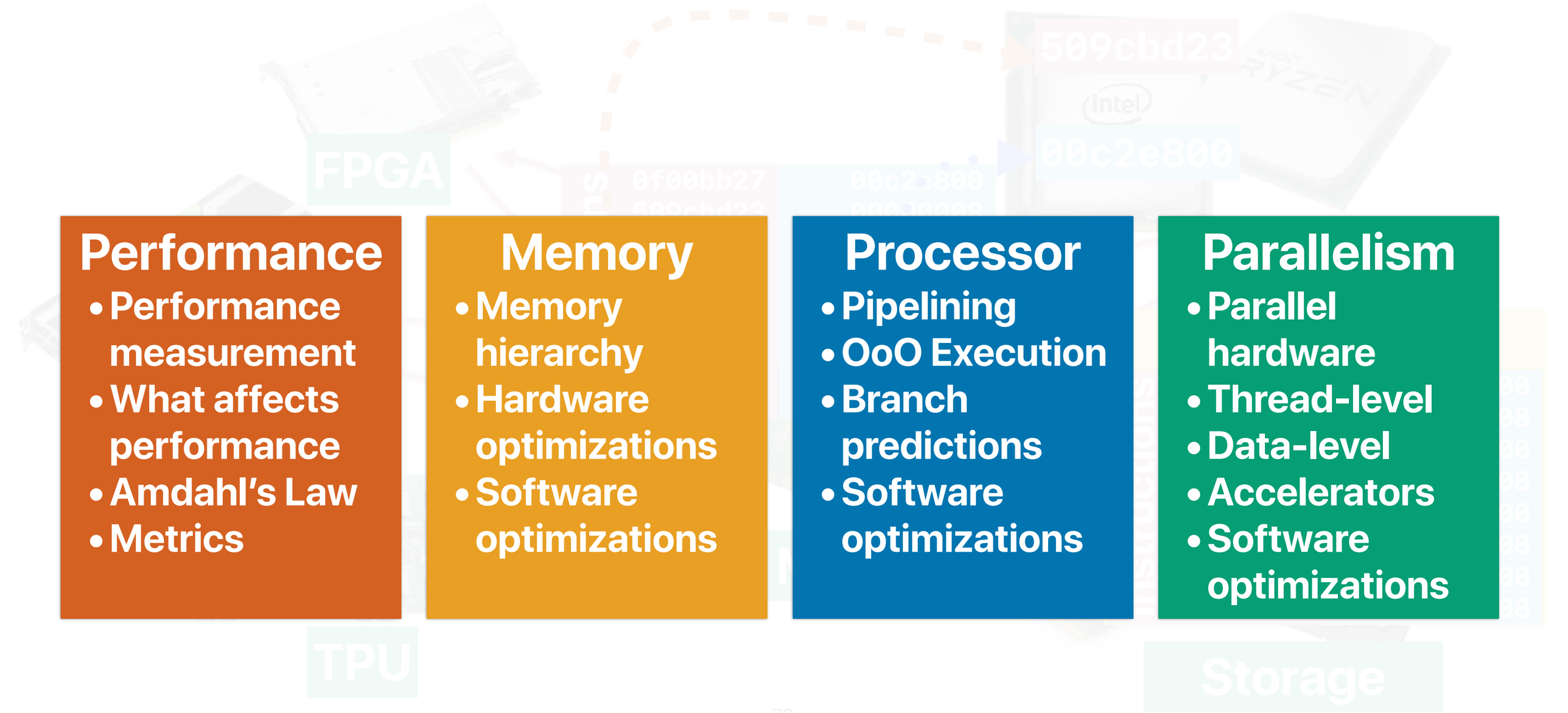
TPU



Storage

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

Tentative Schedule

Date	Topic	Readings (Required)	Preview Slides	Slides & Demo	Reading Quiz Due	Assignment Due
9/28/2023	Introduction	- G.E. Moore. Cramming More Components				
10/03/2023	Performance Evaluation (I)	– Chapter 1.3 & 1.8-1.9			Reading Quiz #1	
10/05/2023	Performance Evaluation (II)	– M. D. Hill and M. R. Marty, Amdahl's Law in the				Assignment #1 Due
10/10/2023	Performance Evaluation (III)	(Optional) Andrew Davison, Twelve Ways to Fool			Reading Quiz #2	
10/12/2023	Memory Hierarchy (1): The Basics	– Appendix B.1-B.3				Assignment #2 Due
10/17/2023	Memory Hierarchy (2)	– Appendix B.1-B.3			Reading Quiz #3	
10/19/2023	Memory Hierarchy (3)	– Chapter 2.3				
10/24/2023	Memory Hierarchy (4): Optimizing				Reading Quiz #4	
10/26/2023	Virtual Memory	– Chapter B.4 & B.5, 2.4				Assignment #3 Due
10/31/2023	Midterm					
11/02/2023	Basic Processor Design & Branch	– Chapter 3.3			Reading Quiz #5	
11/07/2023	Branch Prediction	– M. Evers, S. J. Patel, R. S. Chappell and Y. N.				
11/09/2023	Data hazards	– Chapter 3.4			Reading Quiz #6	
11/14/2023	Data hazards & OOO Scheduling	– D. Suggs, M. Subramony and D. Bouvier, "The				
11/16/2023	OOO Scheduling				Reading Quiz #7	
11/21/2023	Programming Modern Processors	– Chapter 3.11				Assignment #4 Due
11/28/2023	Chip Multiprocessors & Modern	– D. Suggs, M. Subramony and D. Bouvier, "The			Reading Quiz #8	
11/30/2023	Parallel architectures & Dark Silicon	– Chapter 1.7				
12/05/2023	TPU, FPGA	– Adrian M. Caulfield, Eric S. Chung, Andrew				Assignment #5 Due
12/07/2023	Final Exam (Part I)					
12/14/2023	Final Exam (Part II)	According to the campus schedule				

Performance

Memory

Processor

Parallelism

You need to complete the reading of assigned reading materials

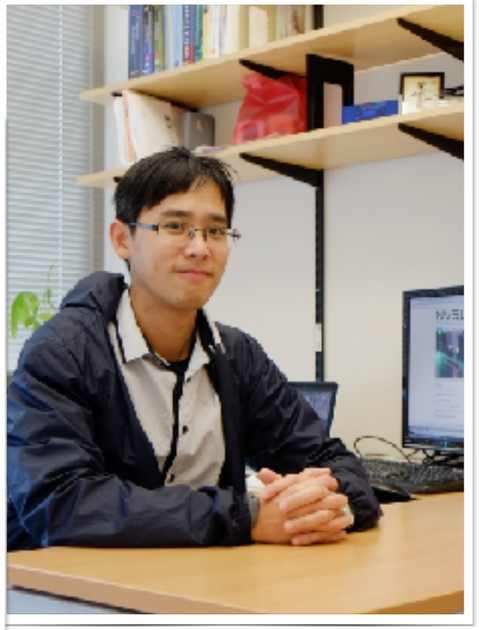
Subject to change

Check due dates here

Logistics

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: https://intra.engr.ucr.edu/~htseng/
- E-mail: cs203ucr@googlegroups.com
- 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 — Teng-Hung “Kimbo” Chen

- Office hours: M 3p-5p @ TBA
- E-mail: cs203ucr@googlegroups.com
- Fun fact: Kimbo can solve Rubik's Cube.

Course website

- https://www.escalab.org/classes/cs203-2023fa/
- Calendar
- Schedule
- Slides
  - Preview — for the ease of note taking
  - Release — the actual slides
- My Grade
  - You may lookup your grades in the section

CS203: Advanced Computer Architecture (2023 Fall)

Lecture: TuTh 3:30p – 4:50p

Where: Student Success Center | Room 230

Schedule and Slides | Assignments | Lookups | My Grade

Instructor

Hung-Wei Tseng

email: cs203ucr@groups.ucr.edu

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

Teaching Assistant

Office Hours:

Other important links

- Assignment & Reading quizzes submission through Gradescope
- Discussion Forum on Piazza
- Lecture recordings: Youtube Channel
- Calendar: Google Calendar

CS203 2023 Fall

Summary of course resources

- Lectures:
  - In-person @ SSC 235
  - Repository on **Youtube**: https://www.youtube.com/profusagi
- Schedule, slides on **course webpage**: https://www.escalab.org/classes/cs203-2023fa/
- Discussion on **piazza**: https://piazza.com/class/lms1flx0rnd5cf
- Reading quizzes, assignments, grading on **gradescope**: https://www.gradescope.com/courses/627794
- Office Hours & Locations https://calendar.google.com/calendar/u/0/r?cid=c_1b9678f28a16f7b598fcdd2f884dcf6ef27702c4c29b5715e61e6f1e121e2589@group.calendar.google.com

Learning eXperience

We're back in-person (still have recordings)

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

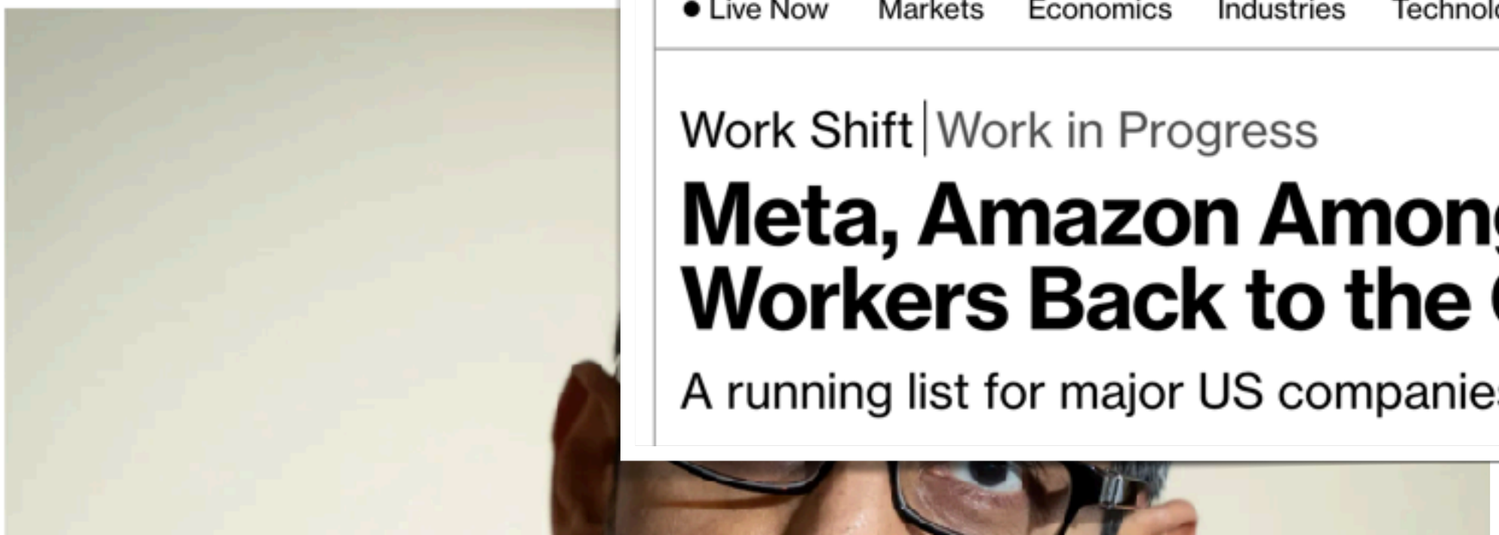
INE RANKINGS MAGAZINE NEWSLETTERS PODCASTS MORE SEARCH SIGN IN

TECH · GOOGLE

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

BY CHRISTIAAN HETZNER

March 7, 2023 at 4:56 AM PST



Bloomberg US Edition Sign In

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

Work Shift | Work in Progress

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

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

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**

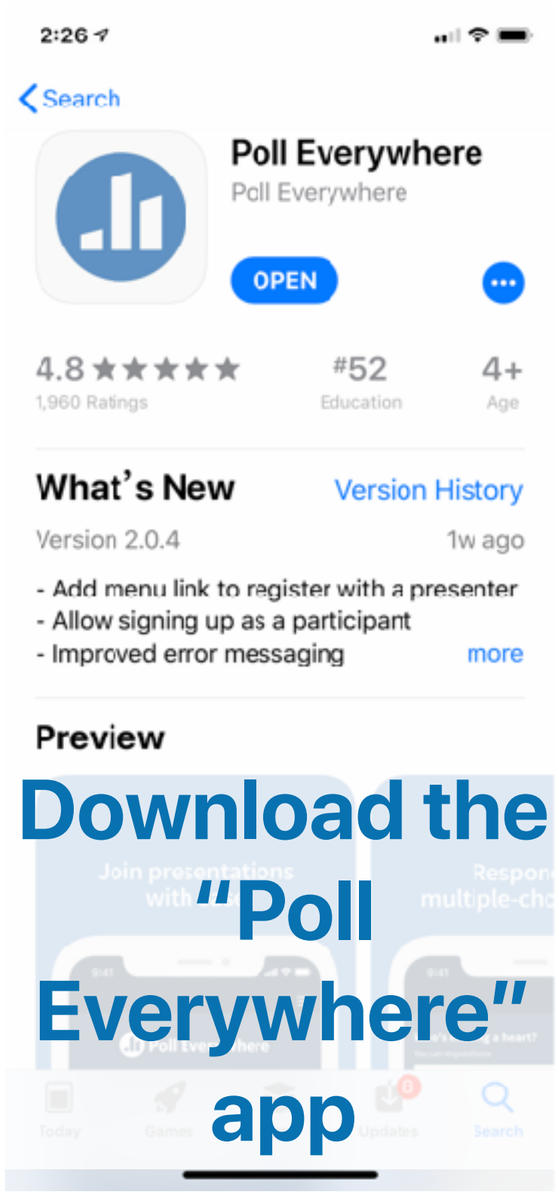
Before lectures: reading quizzes

- We need to prepare you for peer instruction activities and discussions!
- Reading assignments from
  - 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
  - No make up reading quizzes — we will drop your lowest one at least

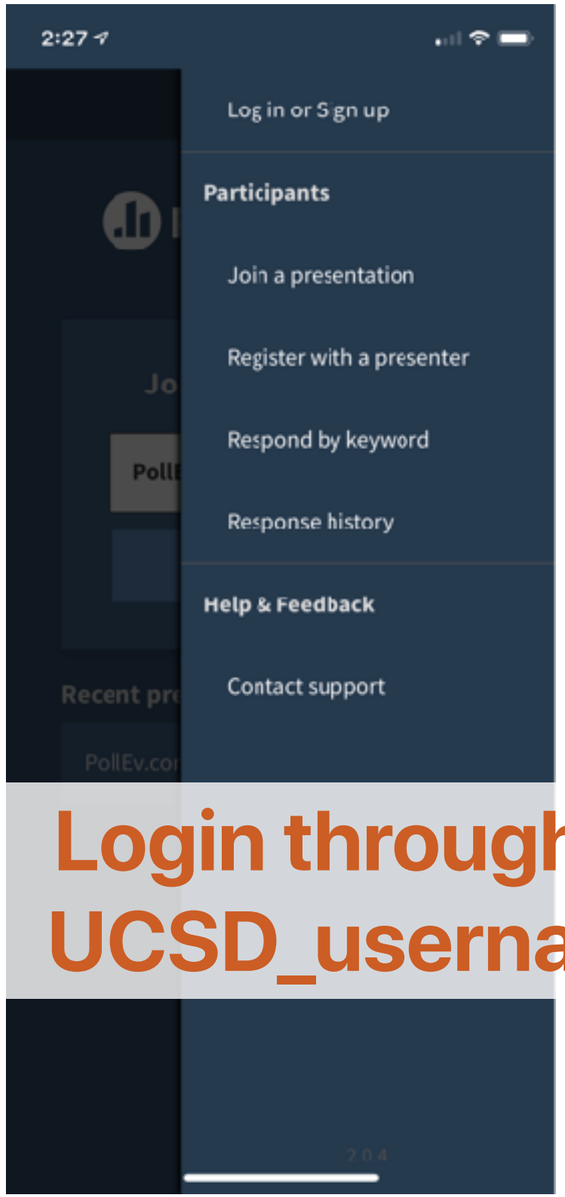
Peer instruction

- 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

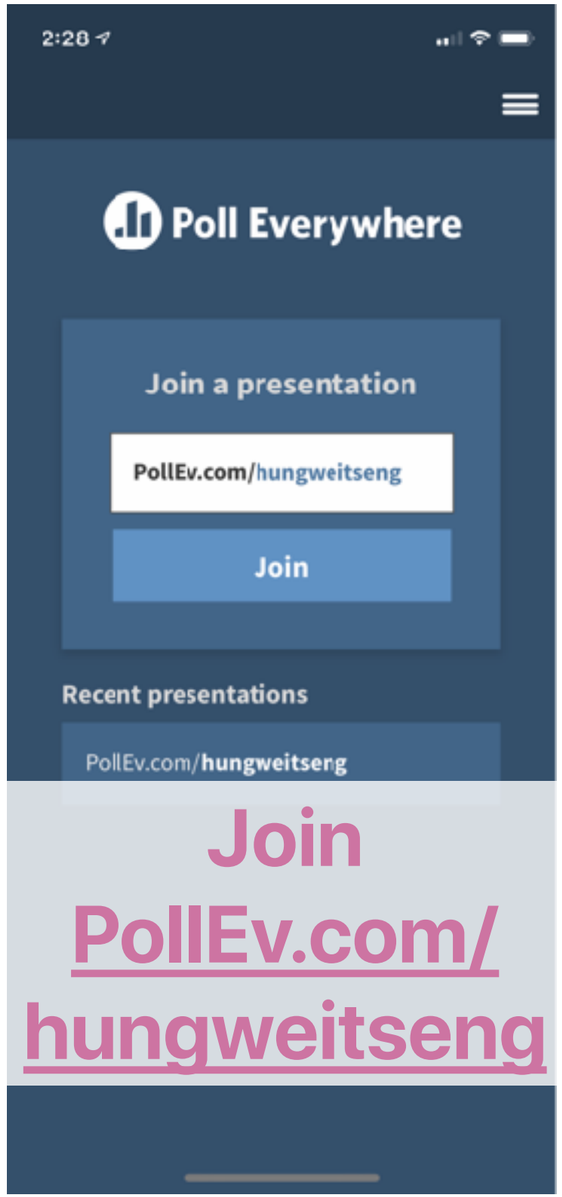
About the time of the Lecture — Setup Poll Everywhere



Download the "Poll Everywhere" app



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

For every assignment

- Watch this video first!!! https://youtu.be/cSlSDCfrUmk
- Go to course home page: https://www.escalab.org/classes/cs203-2023fa
- 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"?
- Jupyter Notebook PDFs
  - PDF submission
  - Submitted via upload on gradescope
- Programming assignments
  - Autograded
  - Submitted via github on gradescope
- 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 — 10/31, closed-book, no cheatsheets allowed
- Final
  - Part I — 12/7: In-person, closed-book, including CSMS comprehensive examine questions
  - Part II—12/14: Online, open-book, open-ended questions to finish in 3 hours

Grading Breakdown

	In-person session
Reading Quizzes	15% Drop lowest 1
Participation	Count as one assignment
Assignments	25% Drop lowest 1
Midterm	25% In-person, closed book
Final	35% Part I & Comprehensive Examine: In-person — closed book Part II: Take home, open 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…

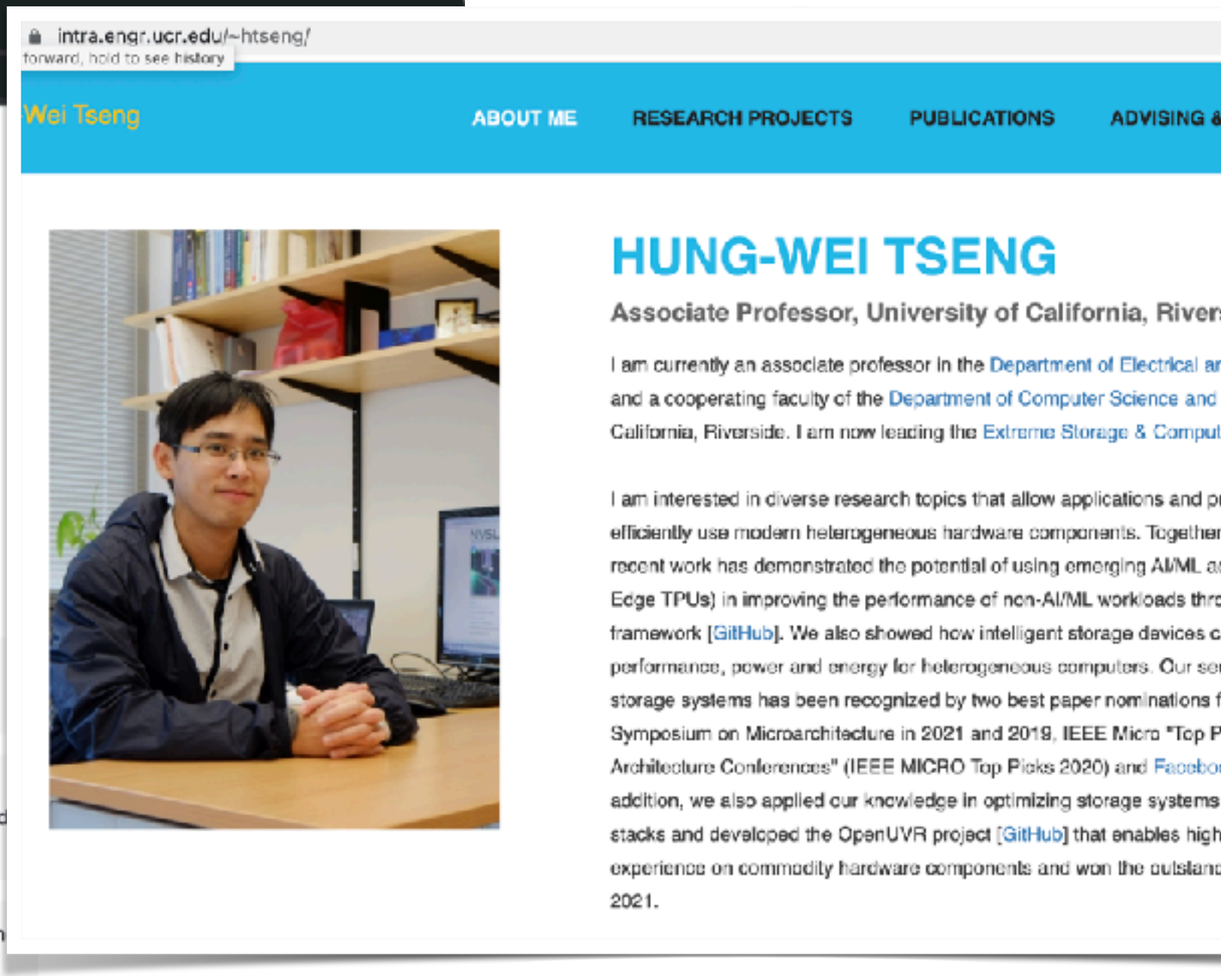
- Do I really need the textbook
  - Again, we need to prepare you for lectures
  - Textbook helps to make sure we're all on the same page when we talk about something

New chat

ChatGPT

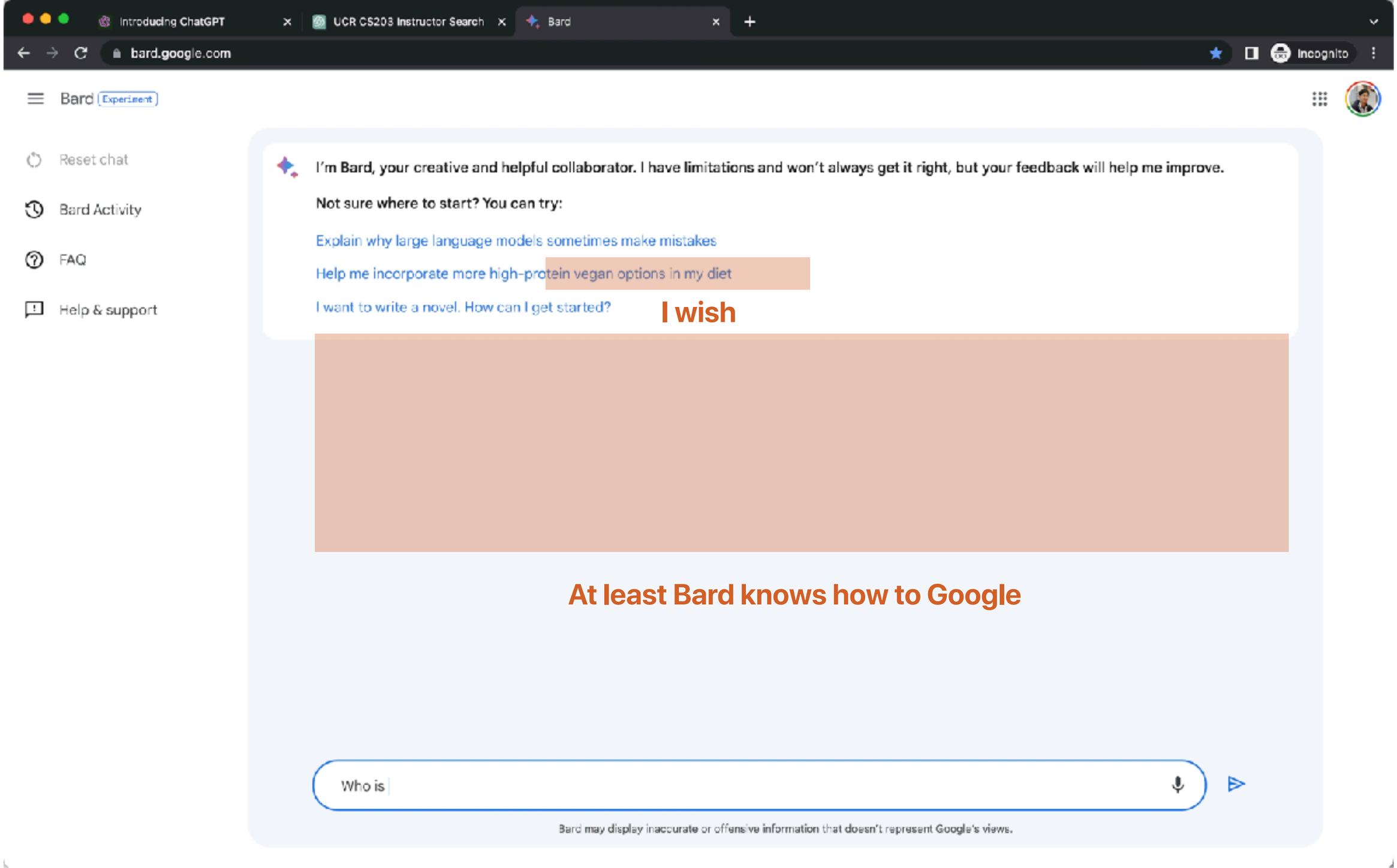
Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce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

Not true



Who

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


Introducing ChatGPT
UCR CS203 Instructor Search
Bard
bard.google.com
Incognito
Bard Experiment
Reset chat
Bard Activity
FAQ
Help & support
I'm Bard, your creative and helpful collaborator. I have limitations and won't always get it right, but your feedback will help me improve.
Not sure where to start? You can try:
Explain why large language models sometimes make mistakes
Help me incorporate more high-protein vegan options in my diet
I want to write a novel. How can I get started?
I wish
At least Bard knows how to Google
Who is
Bard may display inaccurate or offensive information that doesn't represent Google's views.

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/cs203-2023fa
- 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 th[illegible] assignment, a[illegible] plagiarism or any text that may be considered plagiarism by human beings or AI-assisted tools is prohibited and will be reported to the [illegible] Programs.
- I understand th[illegible] incident to the UCR Student Conduct & Academic Integrity Programs.
- I understand th[illegible] 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.

If you haven't submitted, you cannot access the datahub service.
We won't count any effort you made until we confirmed your agreement.
Please submit your agreement as soon as possible and an e-mail justifying your late submission.

☑ **By clicking this box, you are agreeing to the Terms and Conditions of CS 203, Fall 2023.**


WHITE HO
WASHINGTON
Q & A
NBC NEWS
NOW

Announcements

- Check our website
- Login to Gradescope
  - Reading quiz due **Thursday before the lecture**
  - You cannot start your assignment if you did not submit the course agreement
- Assignment #1 due **next Thursday**
- Login to piazza

Computer
Science &
Engineering

203

つづく