CS203 — Advanced Computer Architecture: Your First Step toward Performance Programming

Hung-Wei Tseng and Jinyoung Choi

Before we get there ...

What's your name?

What's your favorite topic in computer science?

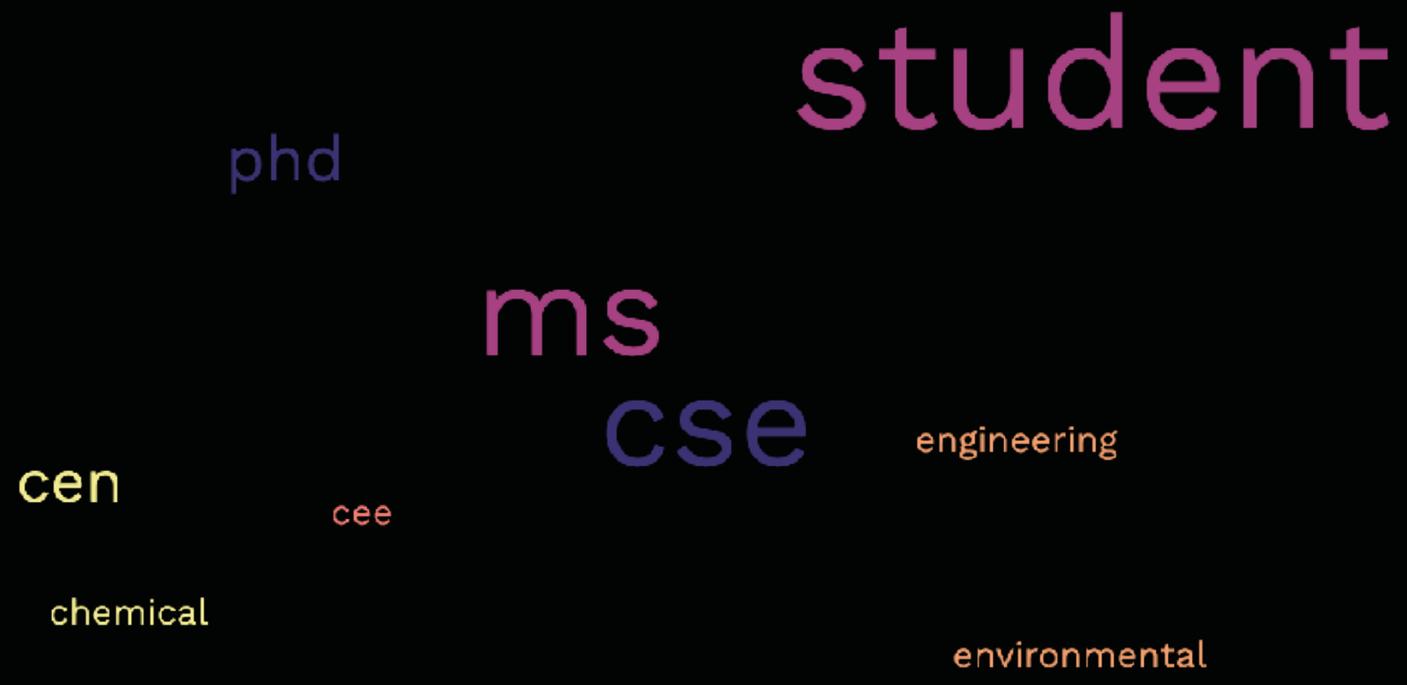
Why're you taking CS203

Background music: Before I Get There/ Songwriter(s):Bronze Radio Return, Chris Henderson/Performed by Bronze Radio Return

Before we get there …

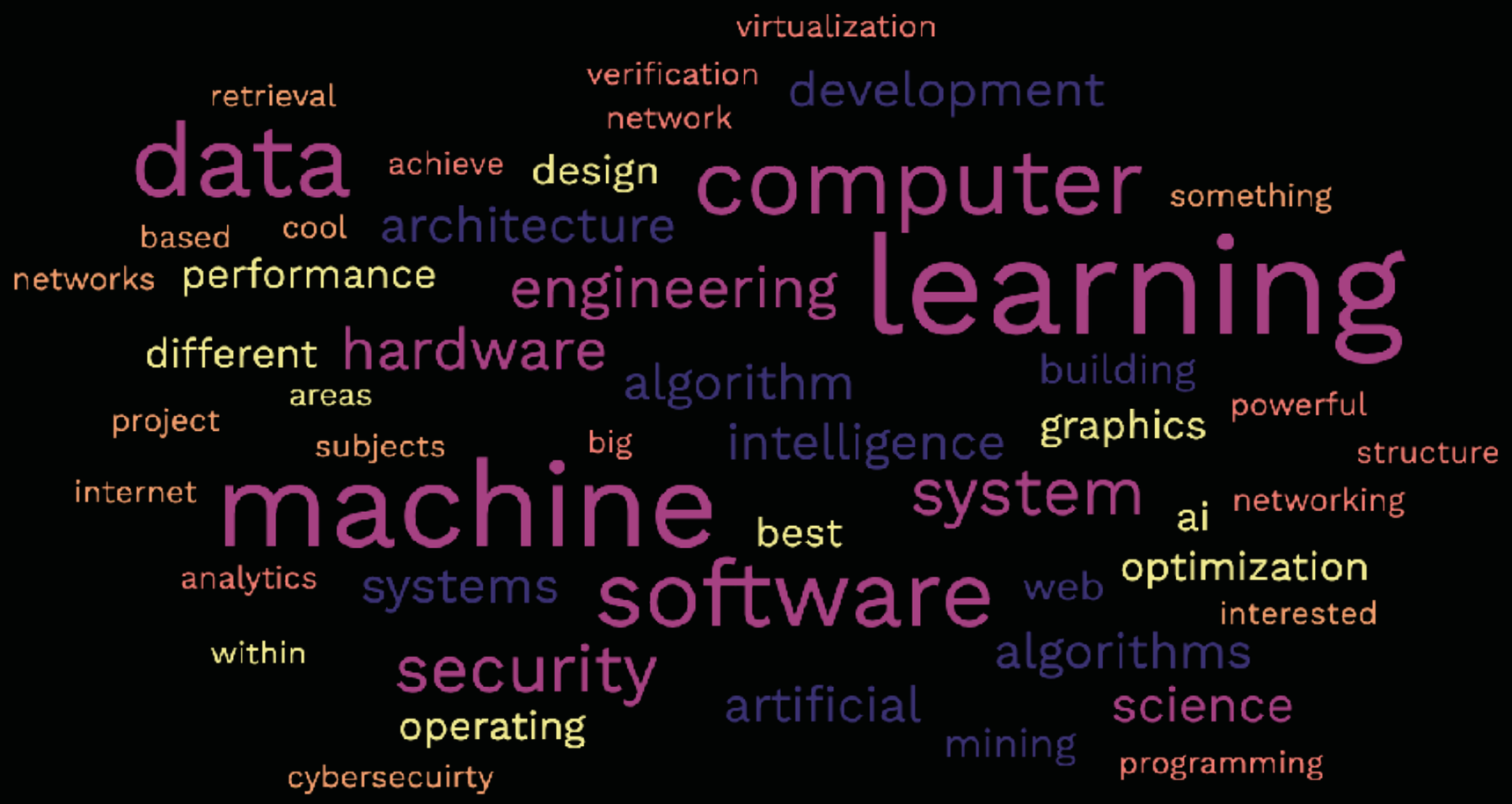
What's your name?

Why're you at UCR



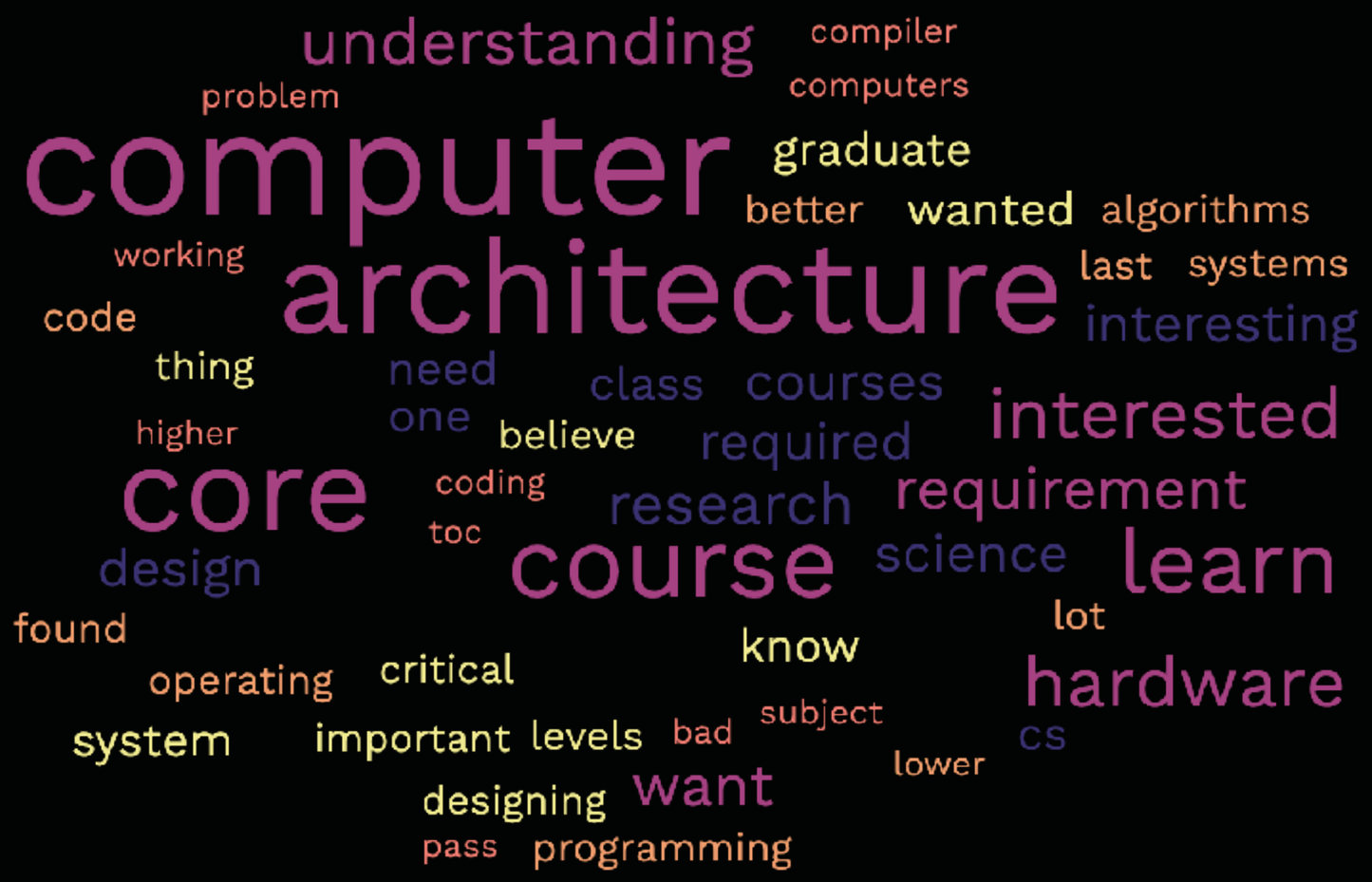
Before we get there …

What's your favorite topic in computer science?



Before we get there ...

Why're you taking CS203

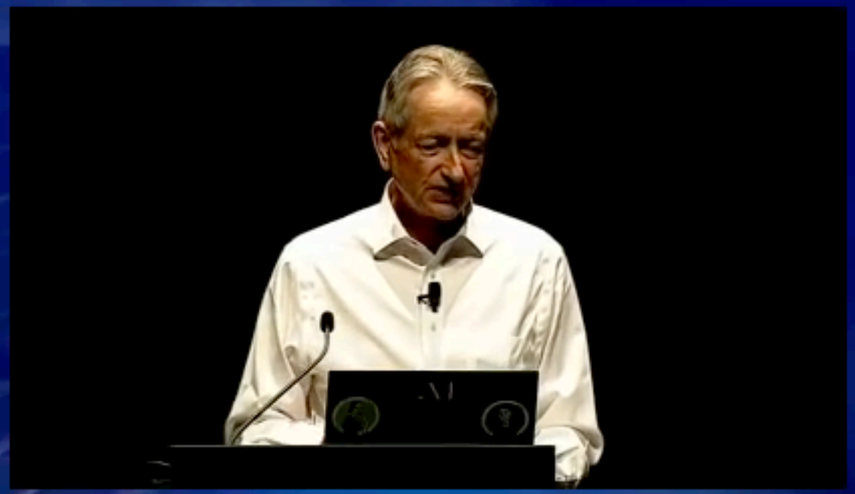


What do you care as a computer scientist/engineer?

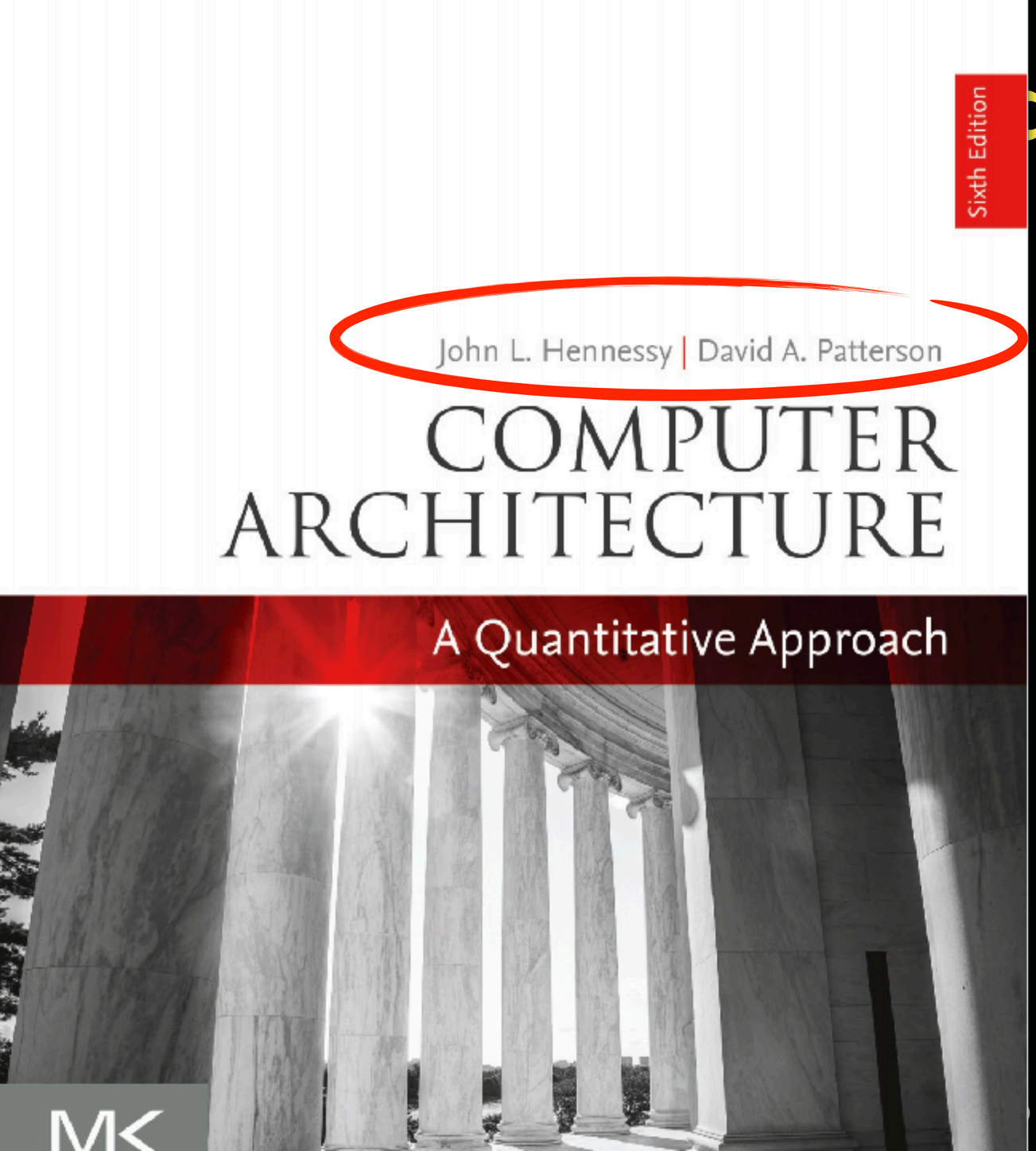
Algorithms
Data Structures
Computer Architecture
Software Engineering
Programming Languages
User Interfaces

The return of backpropagation

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



acm FCRC



d Patterson"?

Computer Architecture

Enables

Deep Learning

2018 Turing Award

Hung-Wei Tseng

David Patterson

John Hennessy

Computer Architecture also enables..

What's "computer architecture"?

What's computer architecture?

Merriam-Webster SINCE 1828

JOIN MWU | GAMES | BROWSE THESAURUS | WORD OF THE DAY | TRAVELER

architecture

DICTIONARY | THESAURUS

architecture noun

ar·chi·tec·ture | \ ˈär-kə- tek-chər \

Definition of *archit*

1 : the art or science of building

specifically : the art or [illegible] habitable ones

2 a : formation or construction resulting from [illegible]

// 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

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

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): $O(n)$

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);
                }
            }
        }
    }
}
```

? ? ? ?

AIRBAG

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

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

CNBC

MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV WATCHLIST

If you're customer-obsessed.
We should talk.

ENTERPRISE

Google is making a big-time mov that should scare Nvidia

PUBLISHED WED, MAY 17 2017•1:03 PM EDT | UPDATED WED, MAY 17 2017•8:01 PM EDT

Jordan Novet
@JORDANNOVET

FORTUNE

Most Popular

TEC

A
C

BY JO

Febr

cnbc.com/2021/09/06/why-tesla-apple-google-and-facebook-are-designing-their-

CNBC

MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV WATCHLIST

GLOBAL X
by Mirae Asset
CLOU
PROSPECTUS > SEE INVESTME

TECH

Tech giants are rushing to develop their own chips — here's why

WIRED

BACKCHANNEL BUSINESS CULTURE GEAR IDEAS SCIENCE SIGN IN SUBSCRIBE

OTTERBOX
Explore More

BUSINESS 11.27.2019 08:11 PM

at Amazon: Its Own Chips for Cloud puting

ollows Google in designing a chip to make cloud computing more efficient; the chip n designs typically seen in smartphones.

CNBC

Search quotes, news & videos

MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV WATCHLIST PRO

You see a best-guess CRM revenue forecast.
We see a precise AI revenue forecast.
See more
C3 AI CRM
for Financial Services
Adobe C3.ai Microsoft
This is Enterprise AI.

TECH

Google reportedly plans to put its own chips in Chromebook laptops from 2023

PUBLISHED WED, SEP 1 2021•4:52 AM EDT

Sam Shead
@SAM_L_SHEAD

SHARE

KEY POINTS

- Google plans to use its own chips in Chromebooks and tablets that run on the company's Chrome operating system from around 2023.
- Google currently uses chips made by the likes of Intel and AMD to power Chromebooks.

/after-amazon-apple-and-google-facebook-to-develop-its-own-ml-chip/

After Amazon, Apple And Google, Facebook To Develop Its Own ML Chip

15/09/2021

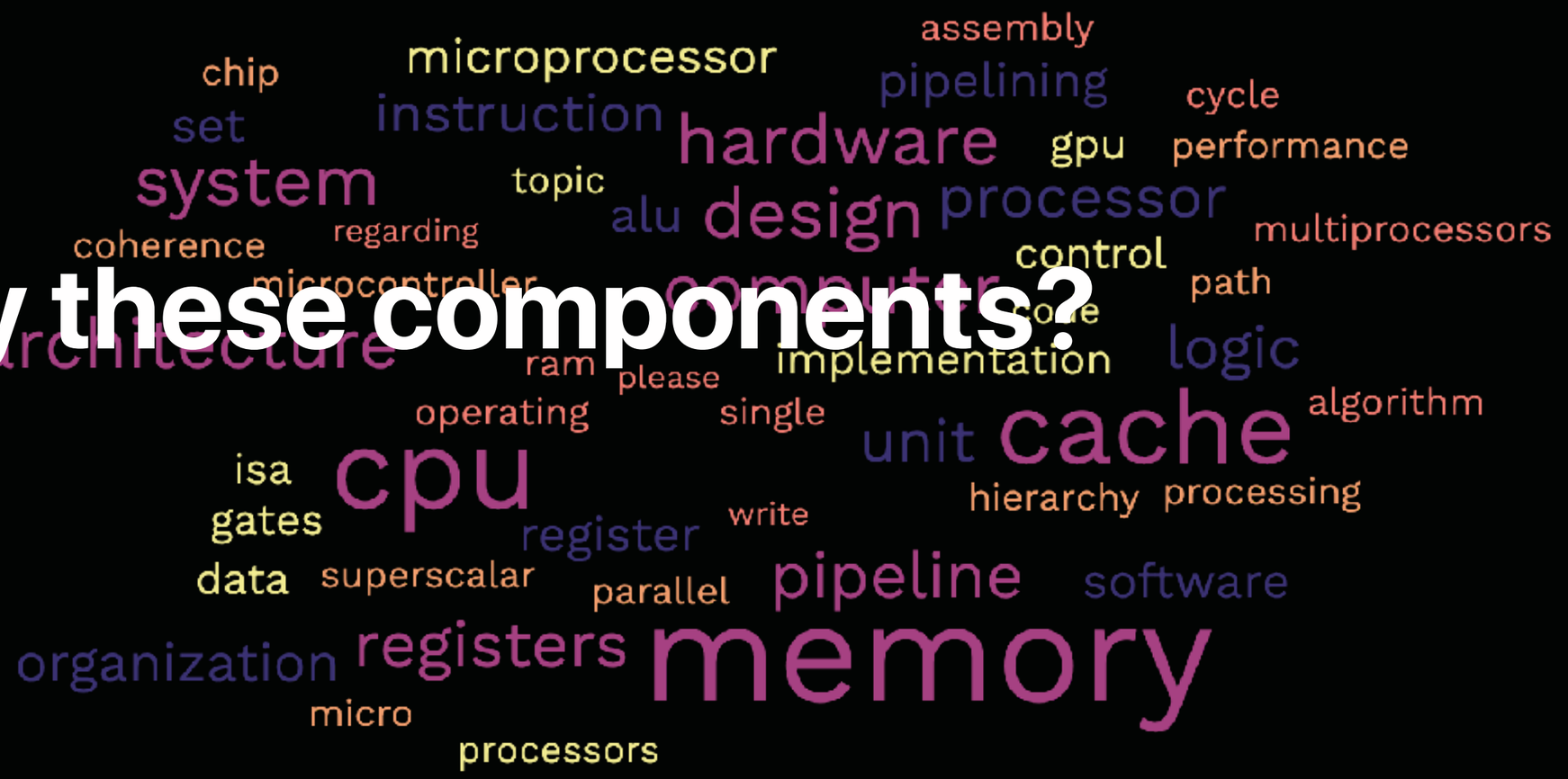
→ Facebook is building a machine learning chip to manage content recommendations for its users.

3:00

How does a computer execute a program?

Your impression about computer architecture

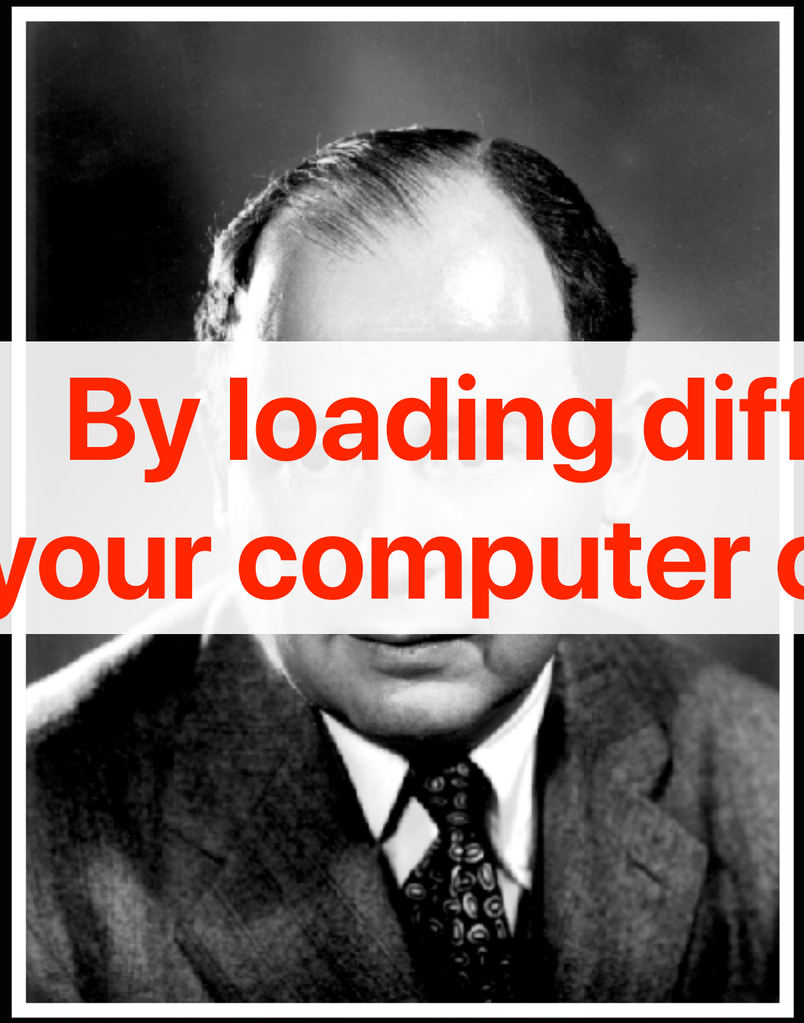
What's your impression about computer architecture?

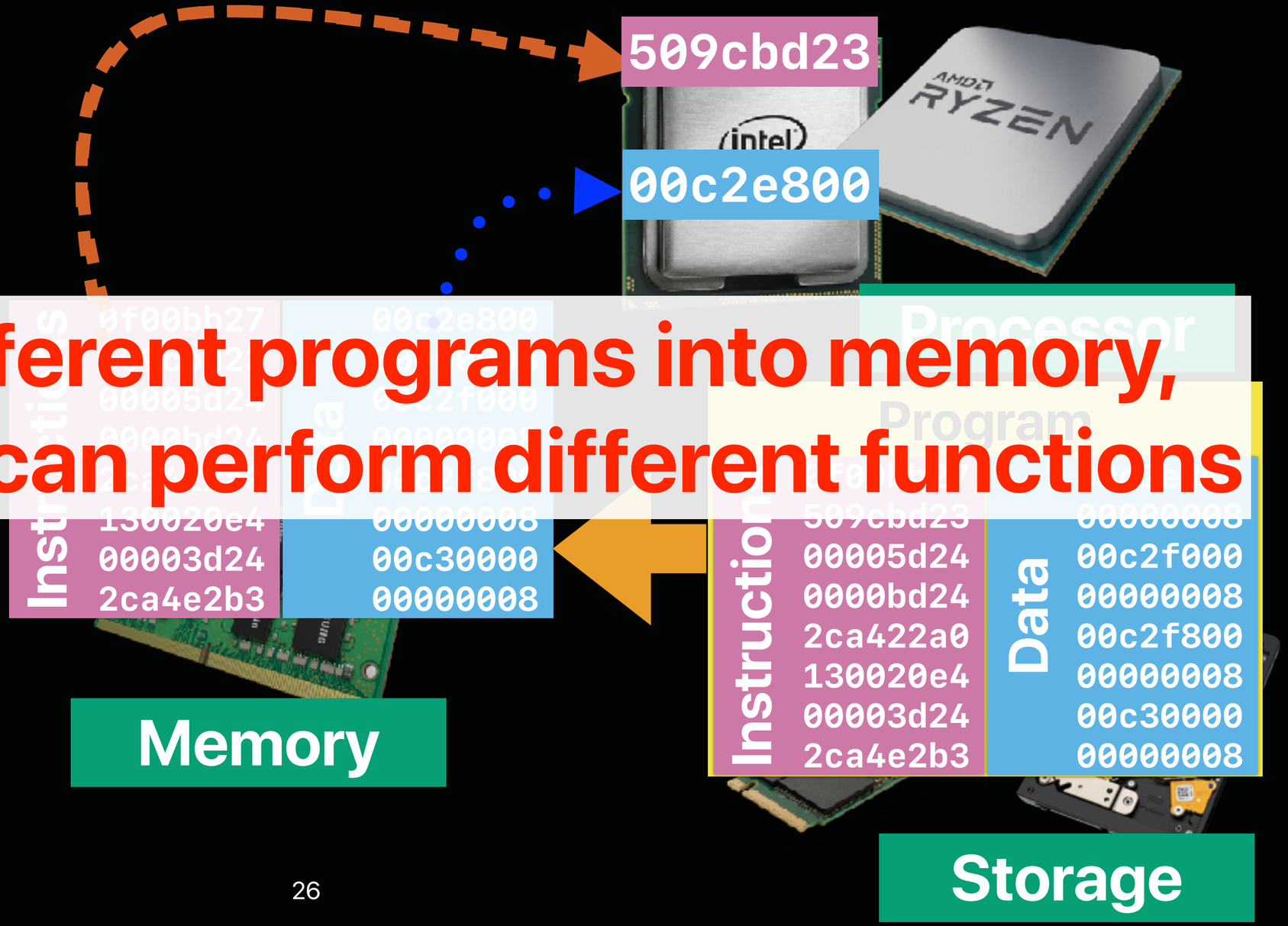


Why these components?

The big picture of “computer architecture"

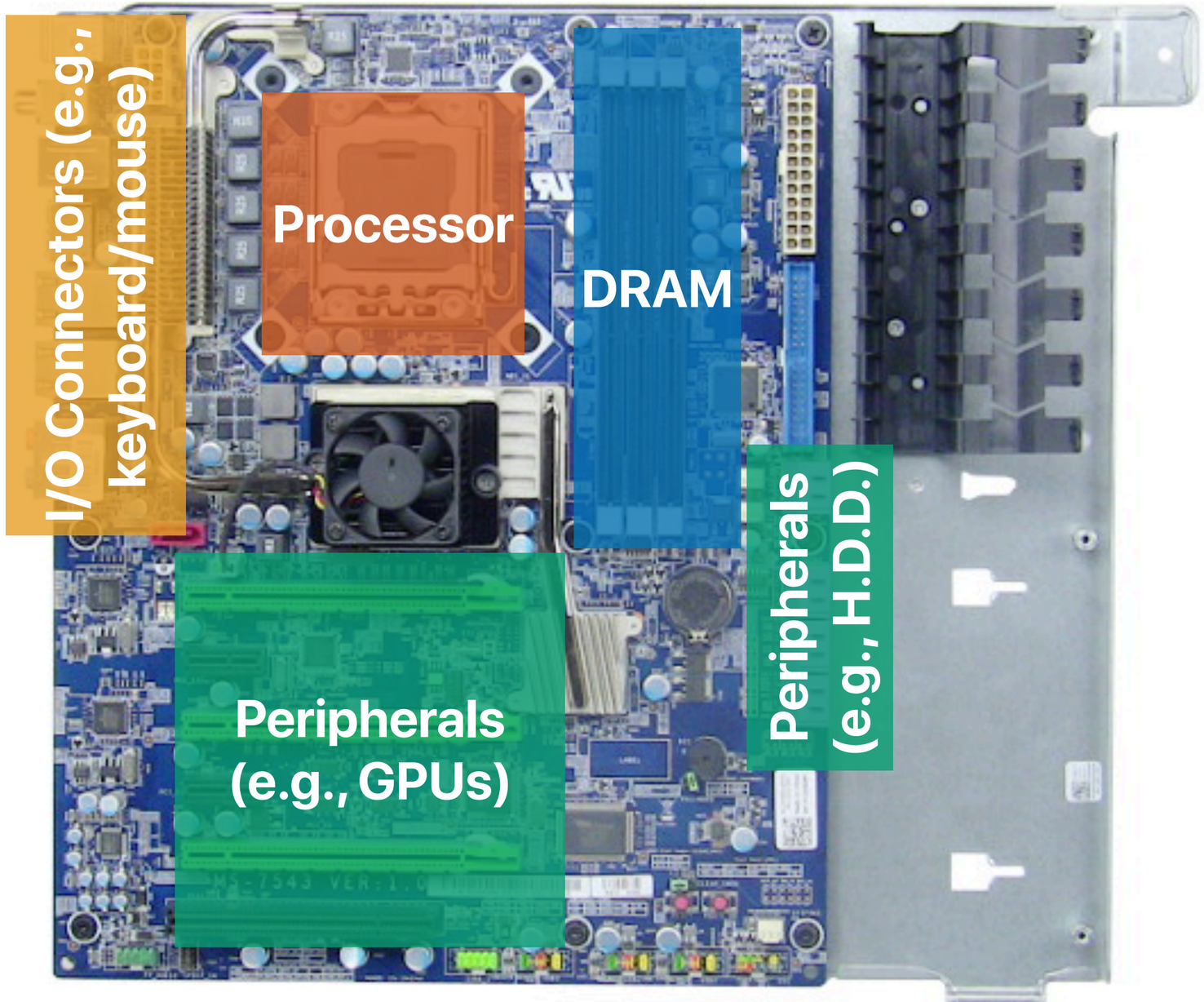
von Neumann architecture



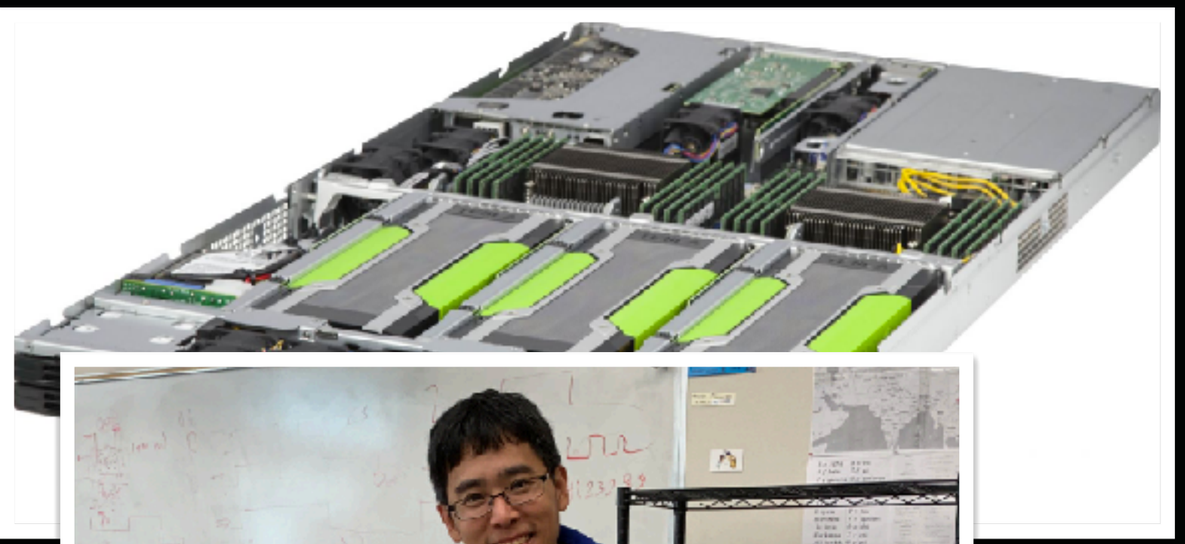


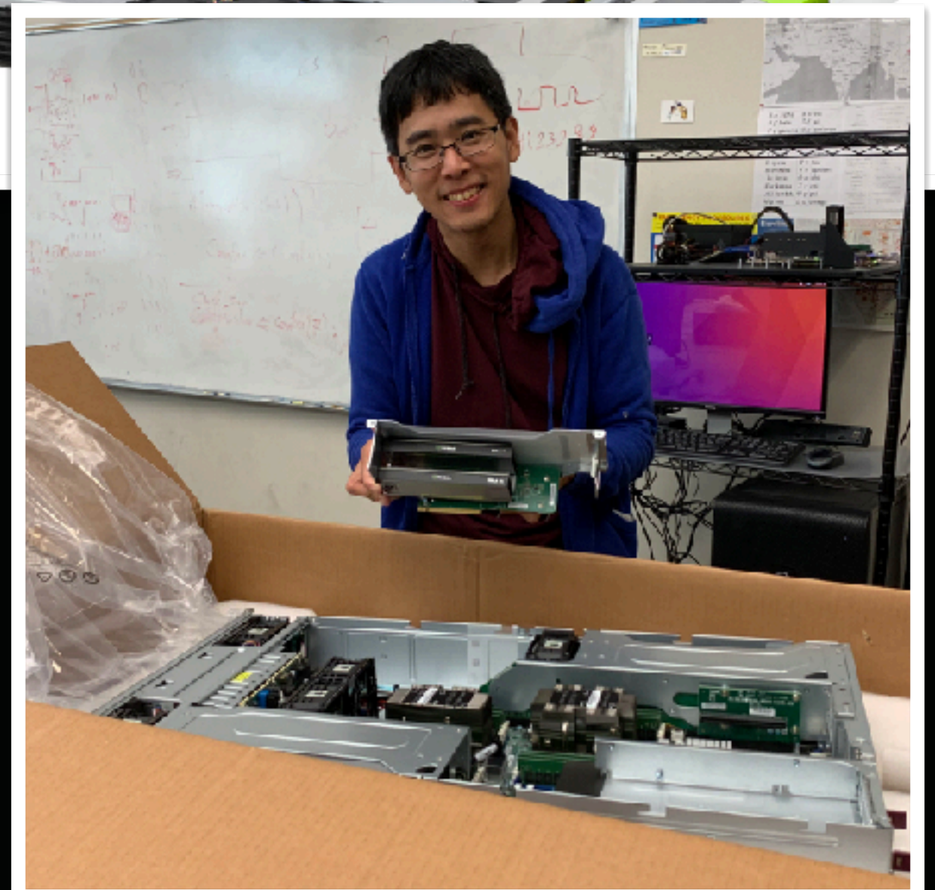
By loading different programs into memory, your computer can perform different functions

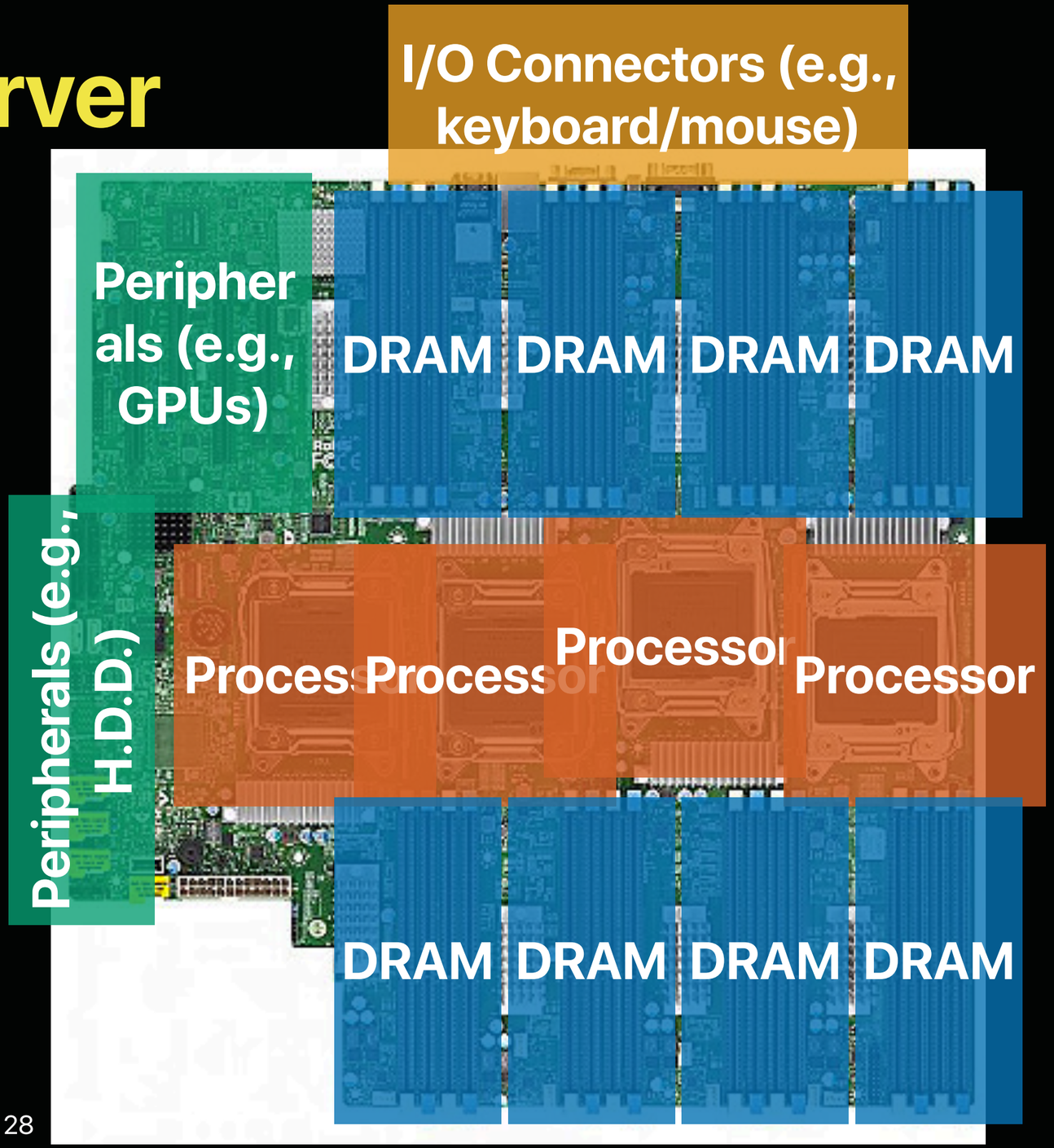
Desktop Computer



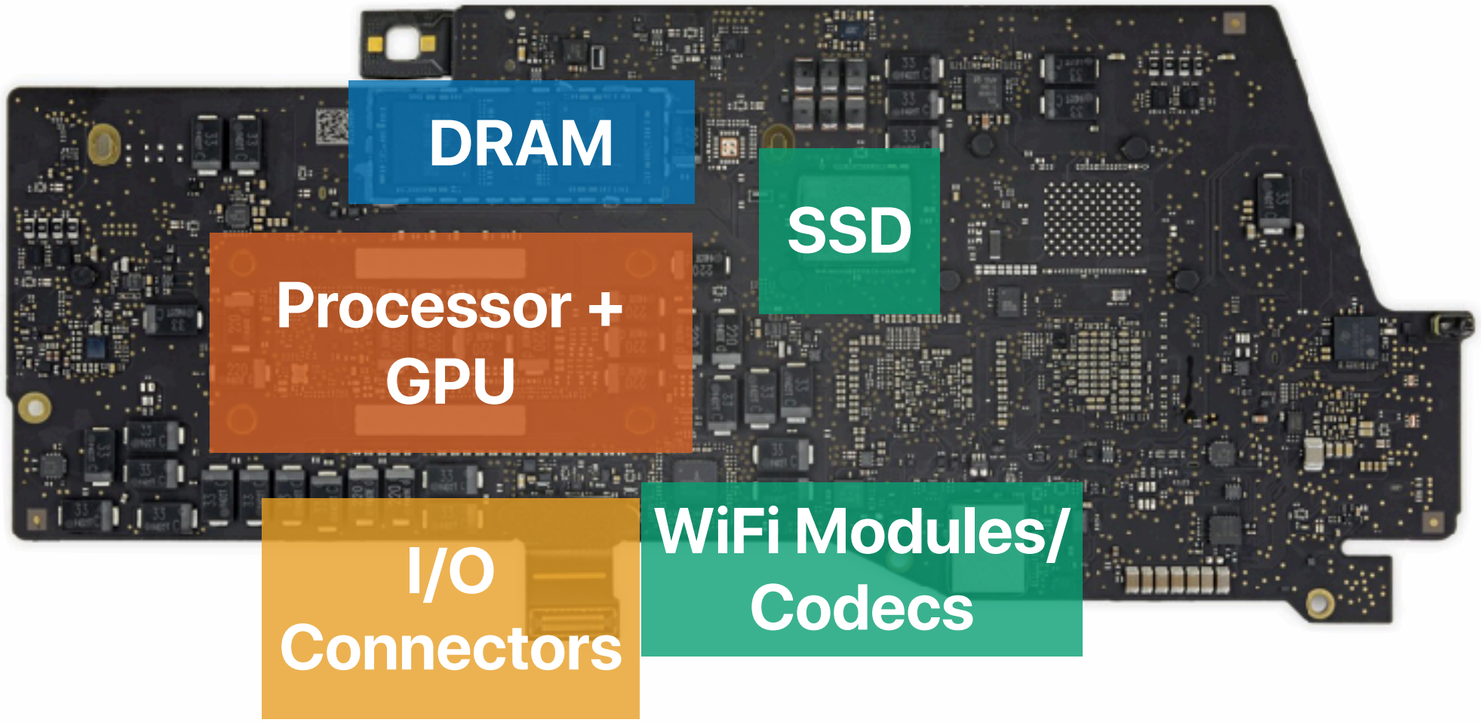
Server



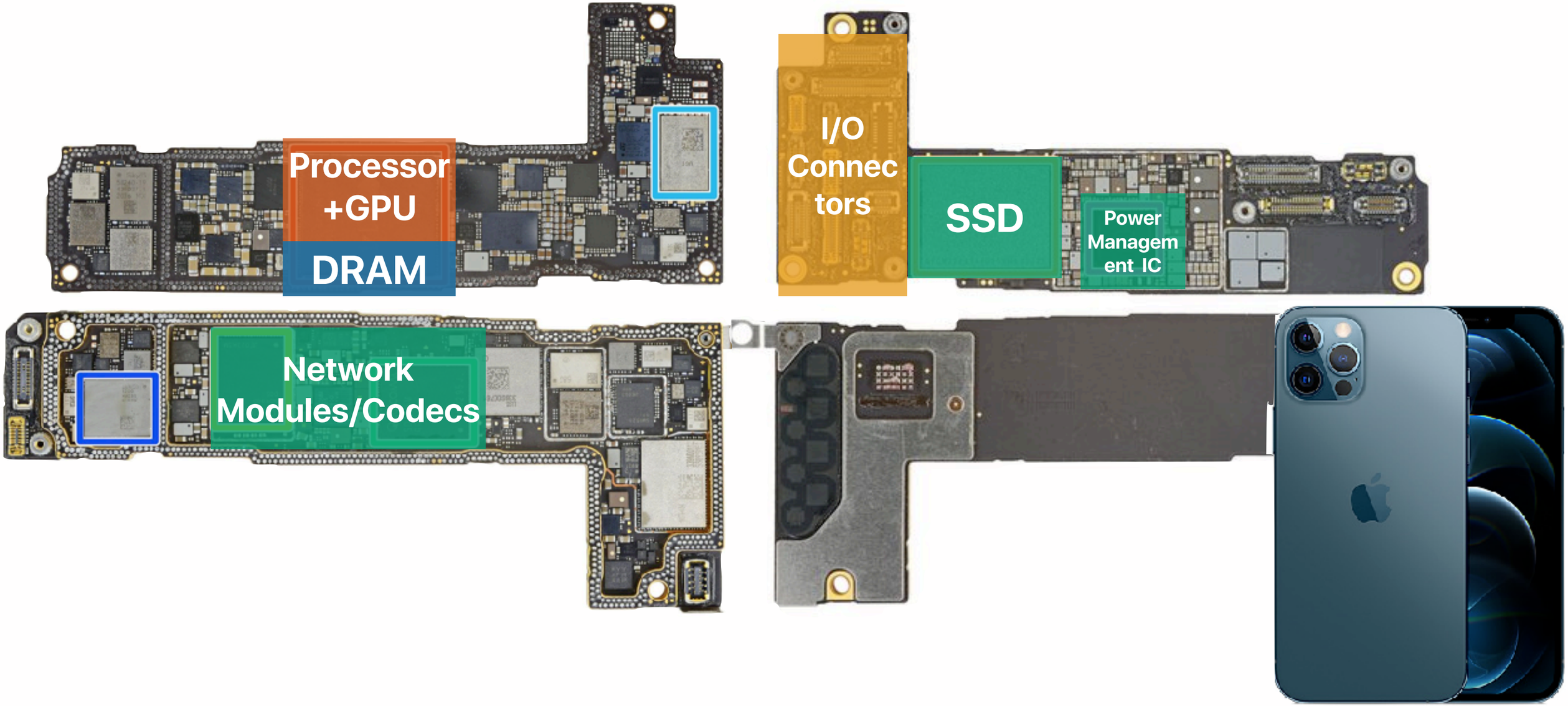



I/O Connectors (e.g., keyboard/mouse)
Peripherals (e.g., GPUs)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Processor
Processor
Processor
Processor
DRAM
DRAM
DRAM
DRAM

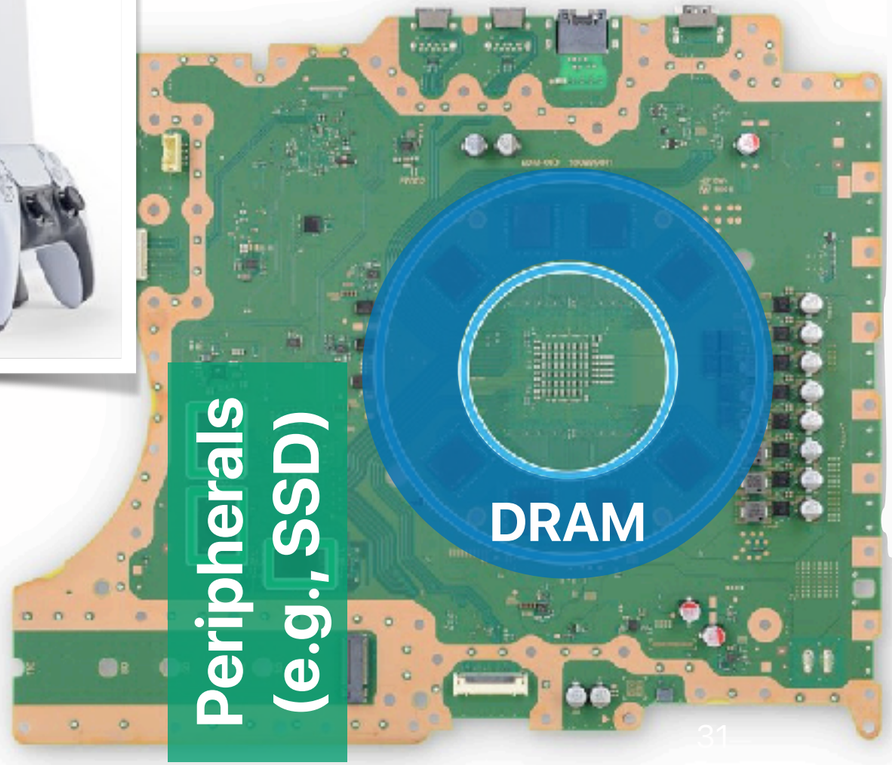
MacBook Pro 13″

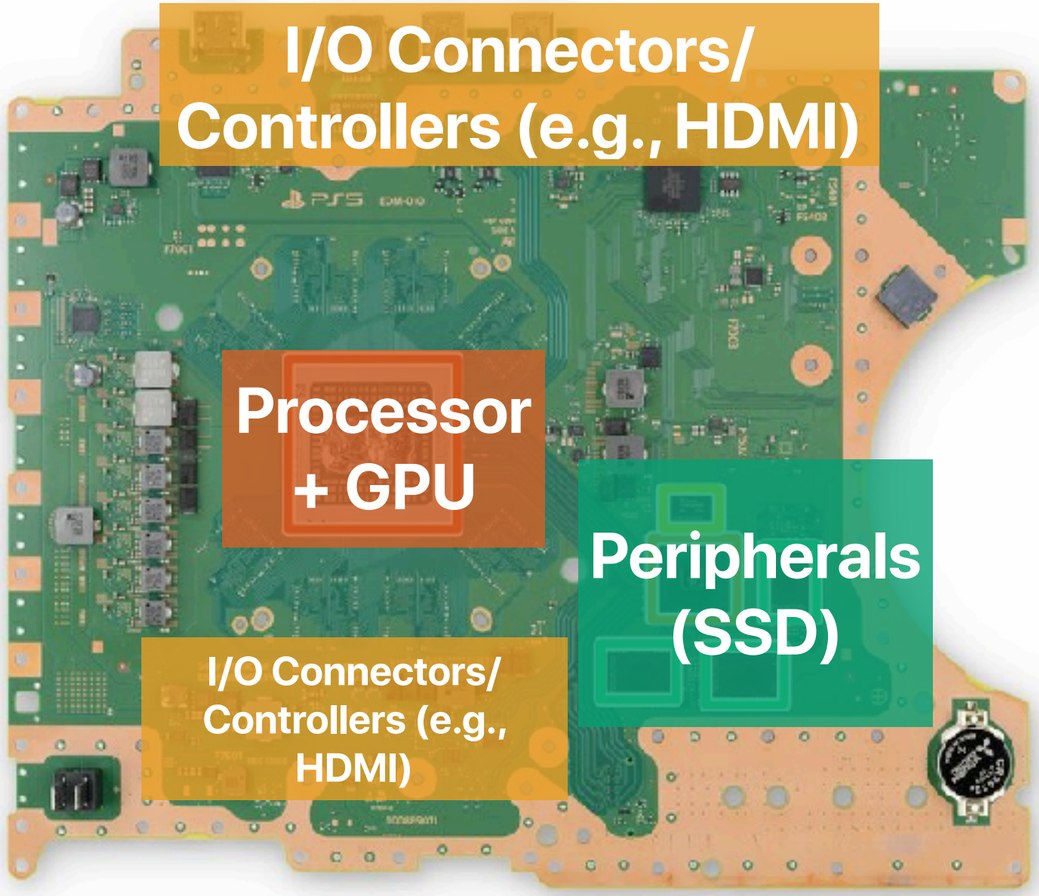


iPhone 13 Pro

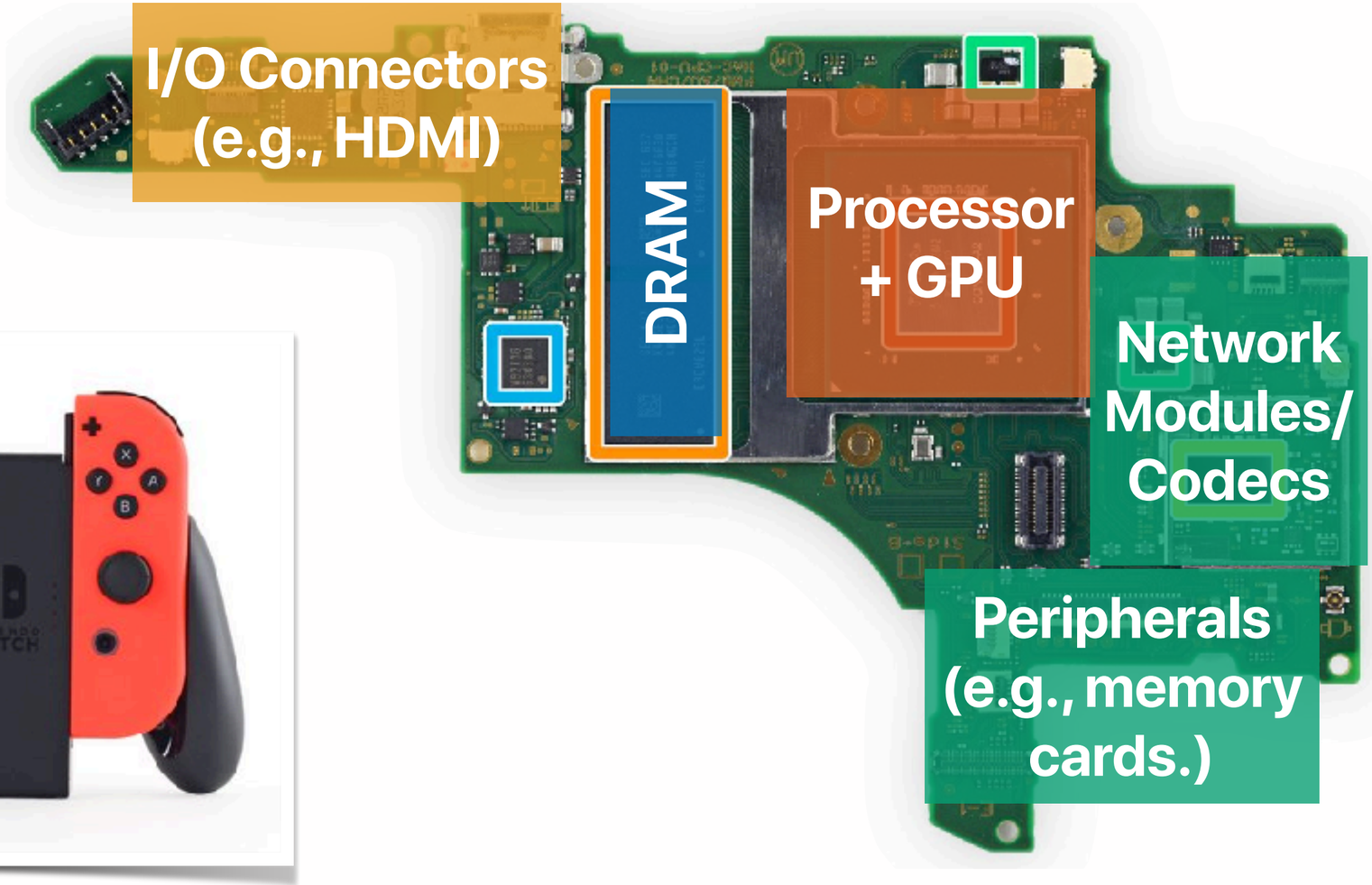


Play Station 5

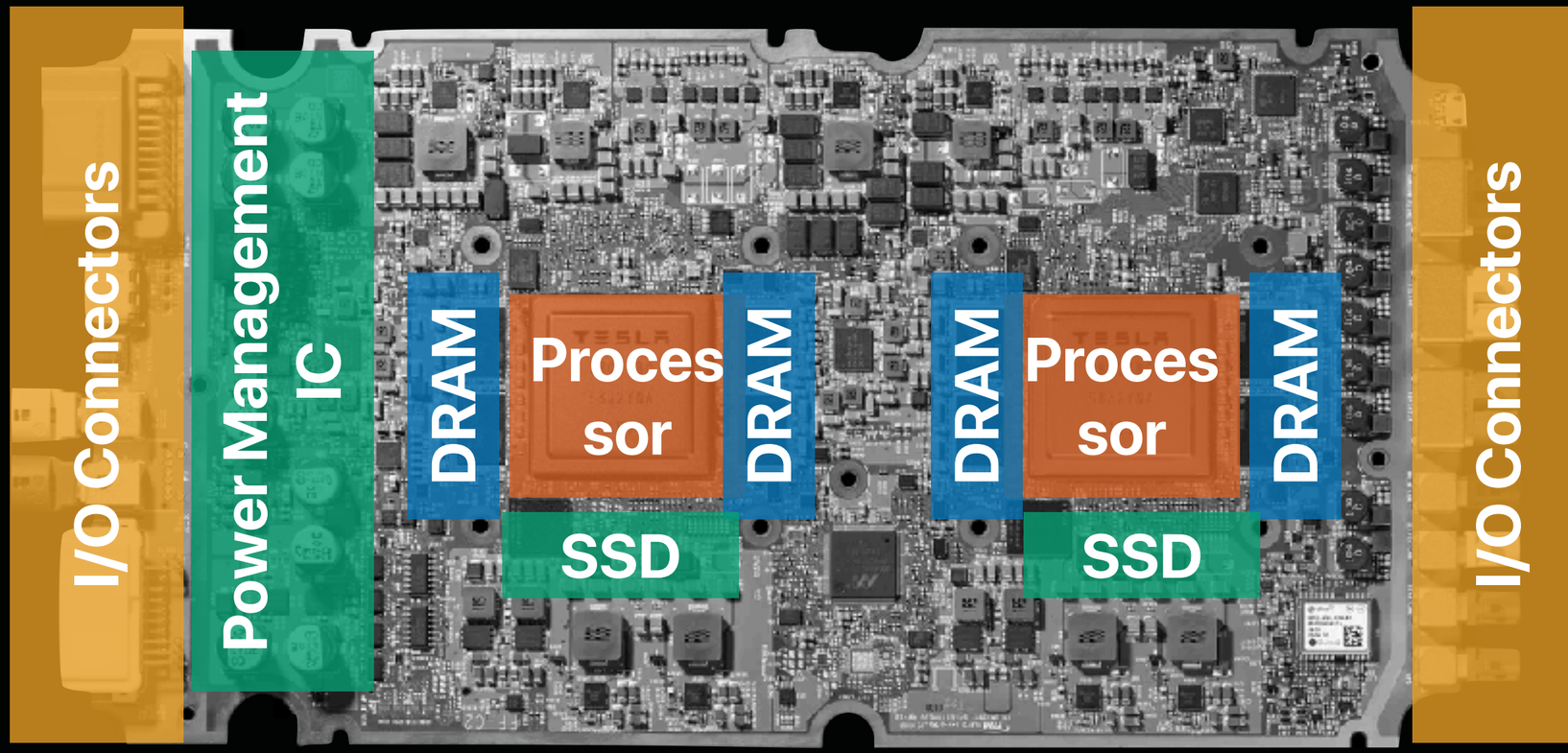




Nintendo Switch

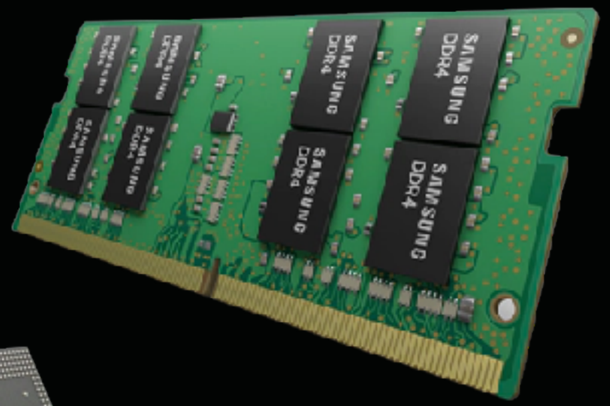


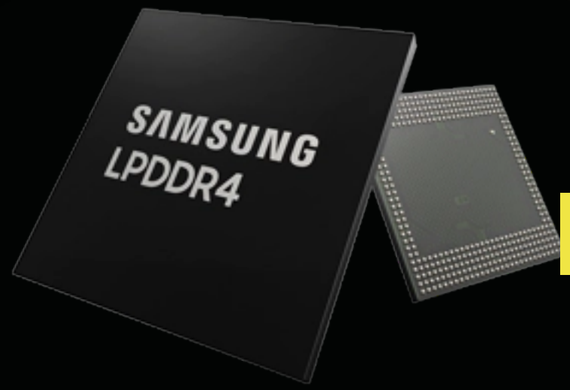
Tesla Model 3



Processors and memory modules are everywhere!

Processors





Memory

Challenges of von Neumann Architecture

Moore's Law

Present and future

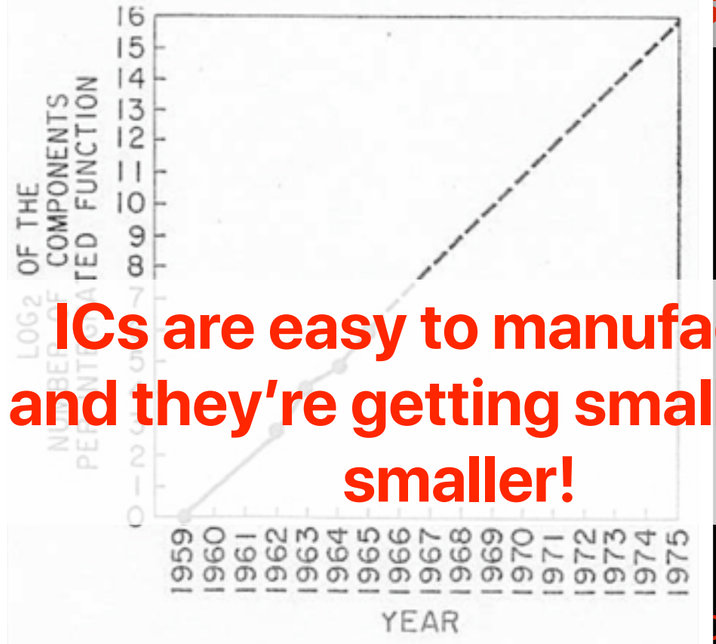
By integrated electronics, I mean
technologies which are referred to
tronics today as well as any additi
result in electronics functions suppli
to miniaturize electronics equipment
creasingly complex electronic functi
space with minimum weight. Seve
evolved, including microassembly
individual components, thin-film
semiconductor integrated circuits.

ICs are increasingly p

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!

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.

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

ICs are more reliable

Heat problem

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

Heat is a solvable issue

Two-mil squares

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

ICs are small

Moore's Law is a historic impo

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.

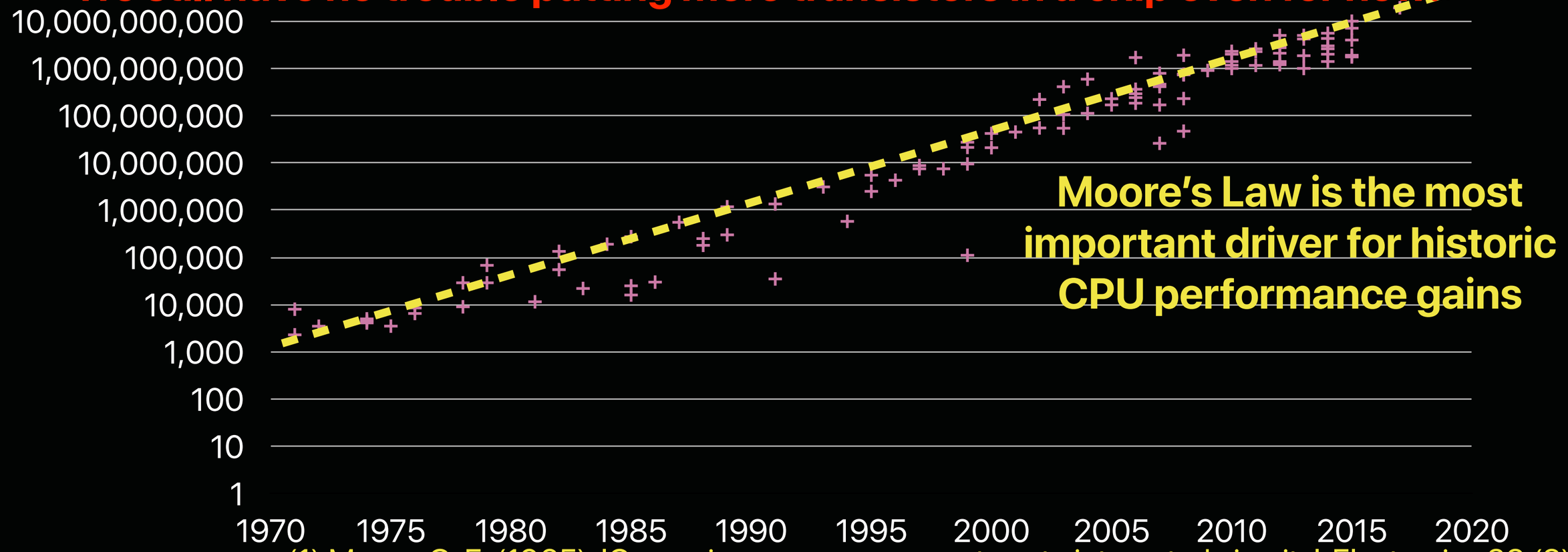
Designing ICs can be easy

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

Moore's Law

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

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



Moore's Law is the most important driver for historic CPU performance gains

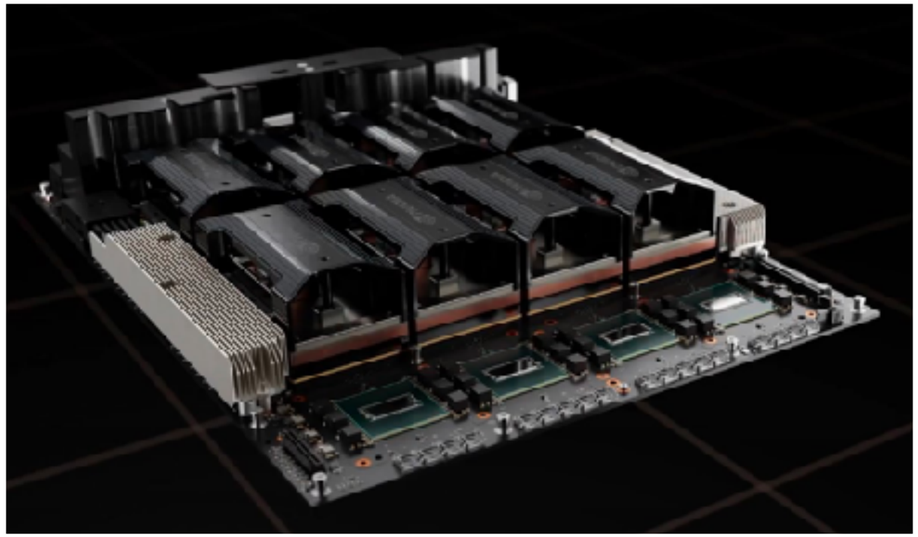
(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8) .

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

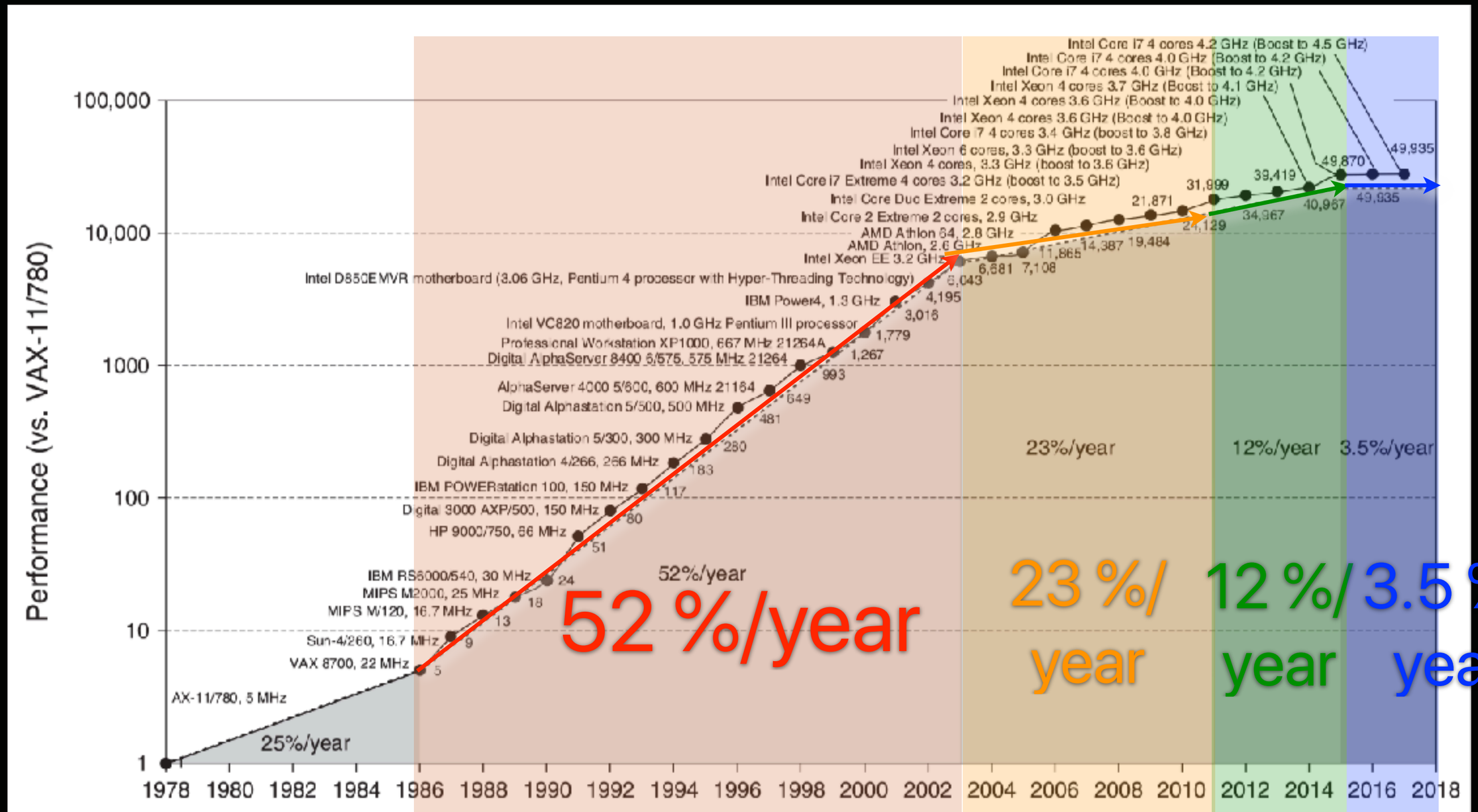
We still have no trouble cramming more transistors

NVIDIA Accelerator Specification Comparison			
	H100	A100 (80GB)	V100
FP32 CUDA Cores	16896	6912	5120
Tensor Cores	528	432	640
Boost Clock	~1.78GHz (Not Finalized)	1.41GHz	1.53GHz
Memory Clock	4.8Gbps HBM3	3.2Gbps HBM2e	1.75Gbps HBM2
Memory Bus Width	5120-bit	5120-bit	4096-bit
Memory Bandwidth	3TB/sec	2TB/sec	900GB/sec
VRAM	80GB	80GB	16GB/32GB
FP32 Vector	60 TFLOPS	19.5 TFLOPS	15.7 TFLOPS
FP64 Vector	30 TFLOPS	9.7 TFLOPS (1/2 FP32 rate)	7.8 TFLOPS (1/2 FP32 rate)
INT8 Tensor	2000 TOPS	624 TOPS	N/A
FP16 Tensor	1000 TFLOPS	312 TFLOPS	125 TFLOPS
TF32 Tensor	500 TFLOPS	156 TFLOPS	N/A
FP64 Tensor	60 TFLOPS	19.5 TFLOPS	N/A
Interconnect	NVLink 4 18 Links (900GB/sec)	NVLink 3 12 Links (600GB/sec)	NVLink 2 6 Links (300GB/sec)
GPU	GH100 (814mm2)	GA100 (826mm2)	GV100 (815mm2)
Transistor Count	80B	54.2B	21.1B
TDP	700W	400W	300W/350W
Manufacturing Process	TSMC 4N	TSMC 7N	TSMC 12nm FFN
Interface	SXM5	SXM4	SXM2/SXM3
Architecture	Hopper	Ampere	Volta

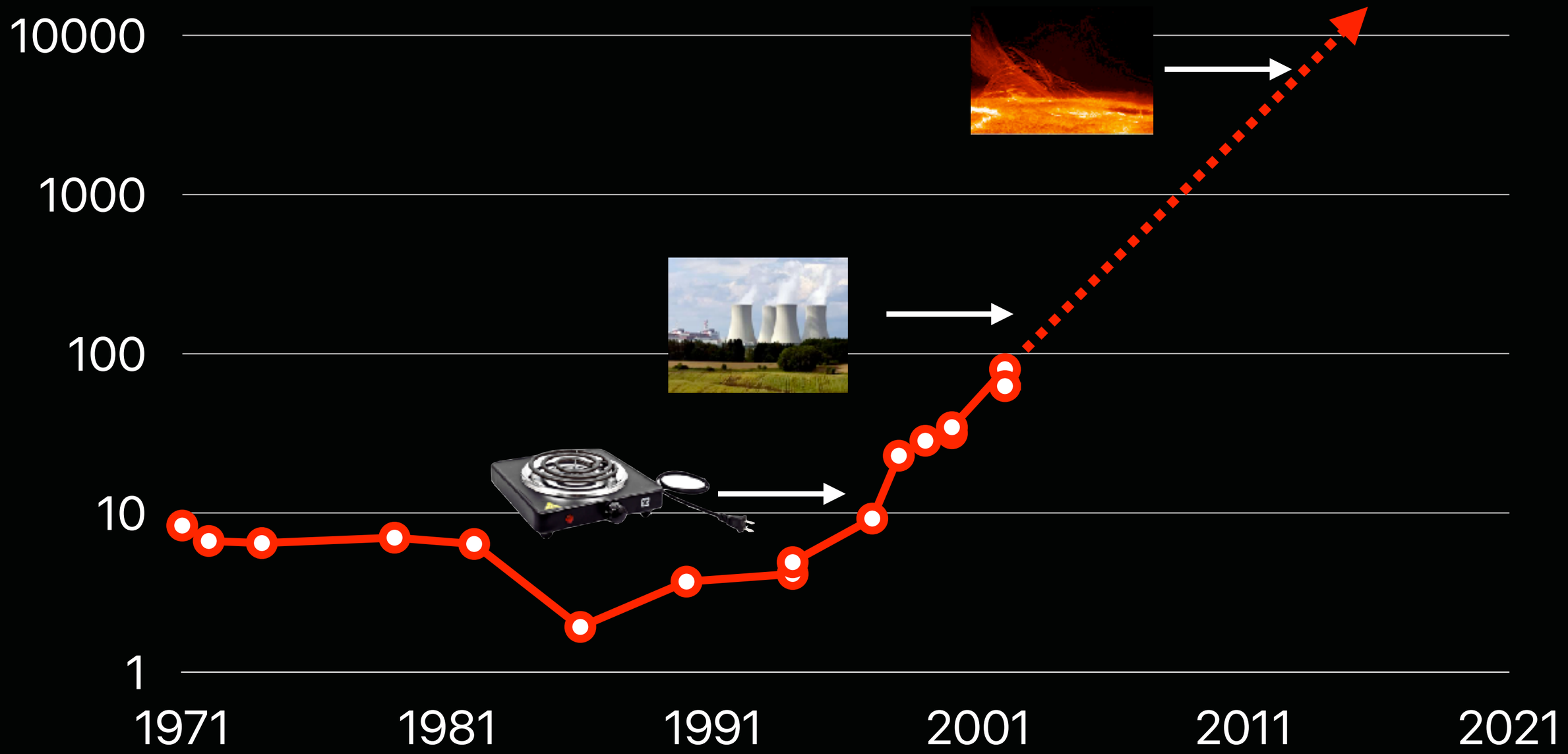
https://www.workstationspecialist.com/product/nvidia-tesla-a100/

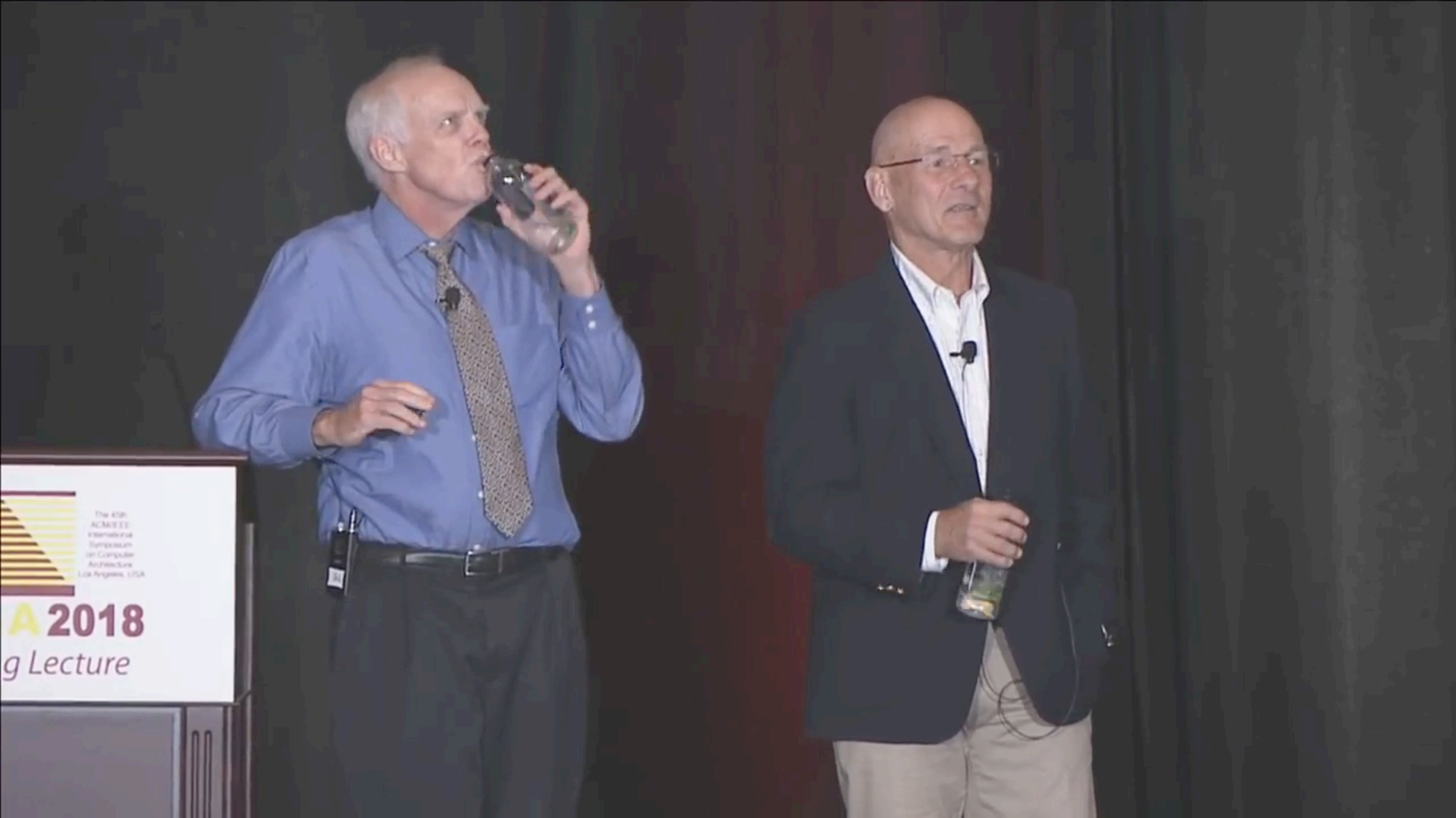


CPU performance scaling slows down

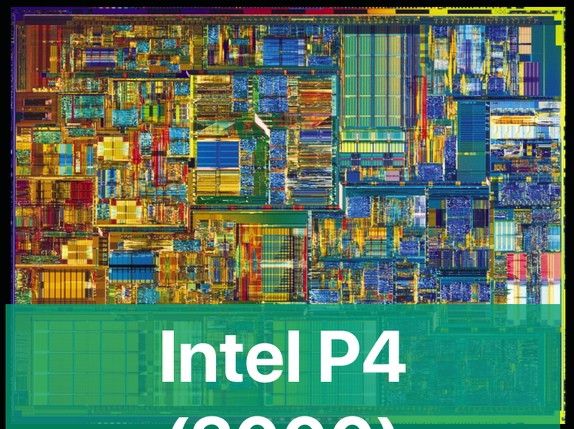


Power Density of Processors

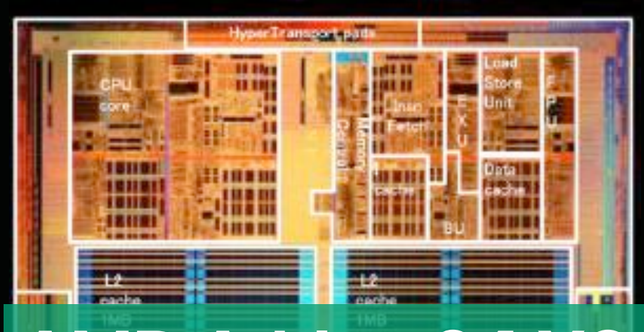



The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA
A 2018
g Lecture

The era of chip multiprocessors

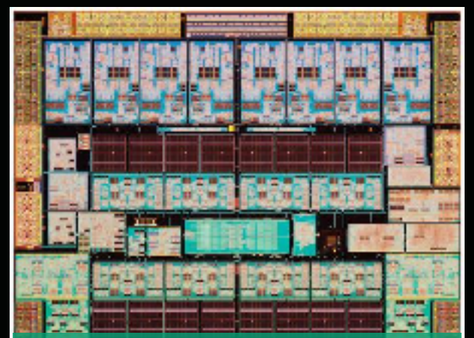


Intel P4
(2000)
1 core

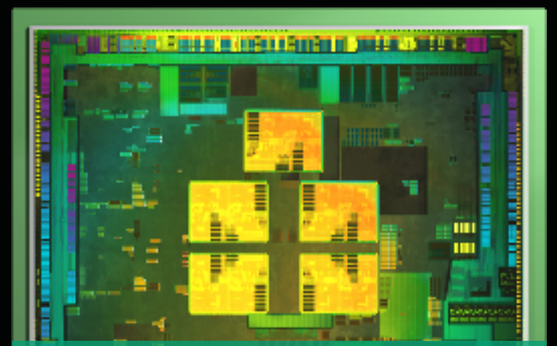


AMD Athlon 64 X2
(2005)
2 cores

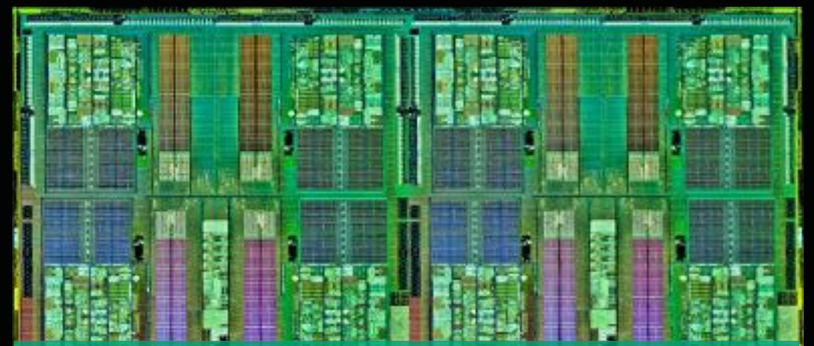
Intel Nahalem
(2010)
4 cores



SPARC T3
(2010)
16 cores

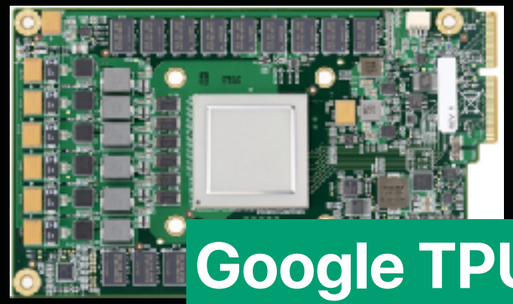


Nvidia Tegra 3
(2011)
5 cores

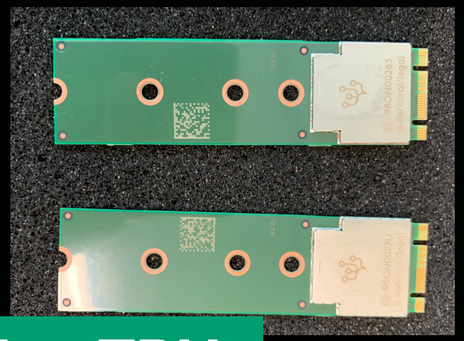


AMD Zambezi
(2011)
16 cores

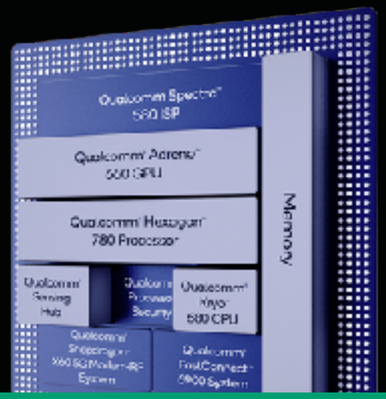
The explosion of AI/ML accelerators



Google TPUv1



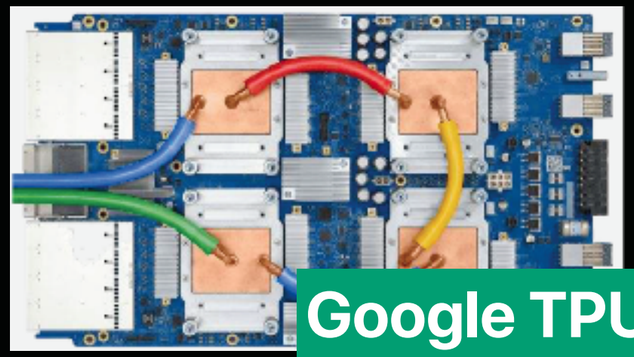
Edge TPUs



Qualcomm Hexagon

Google TPUv2

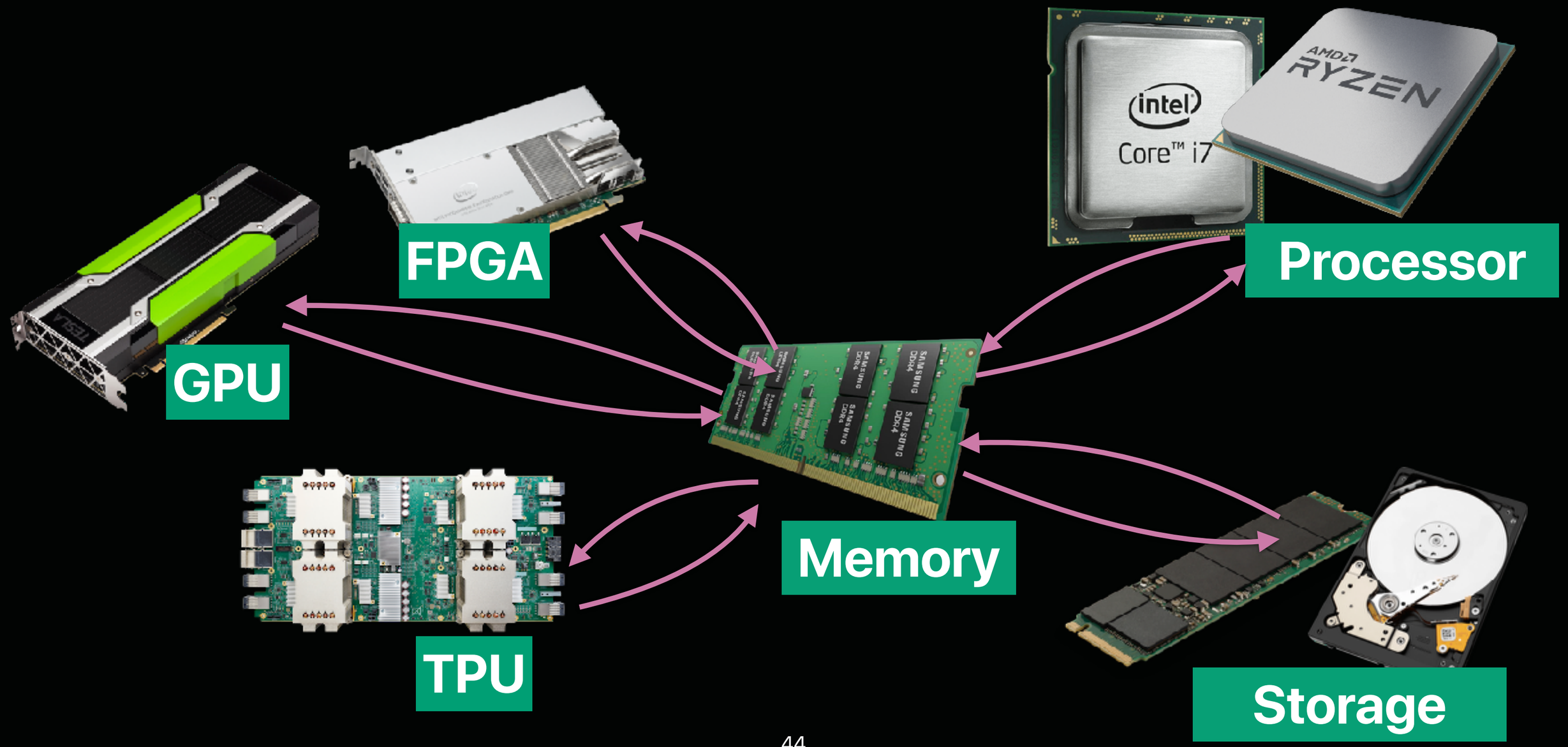
Apple Neural Engines



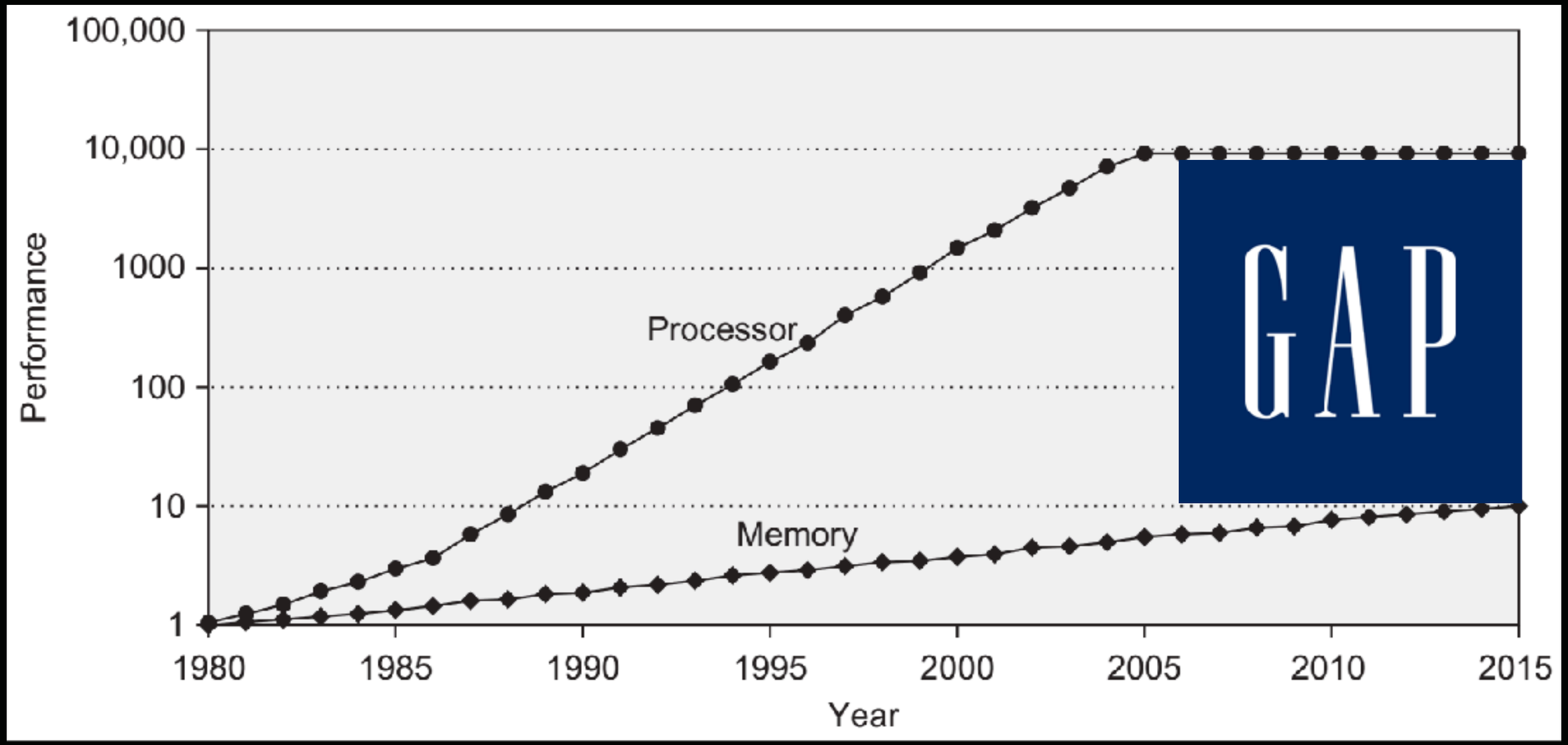
Google TPUv3

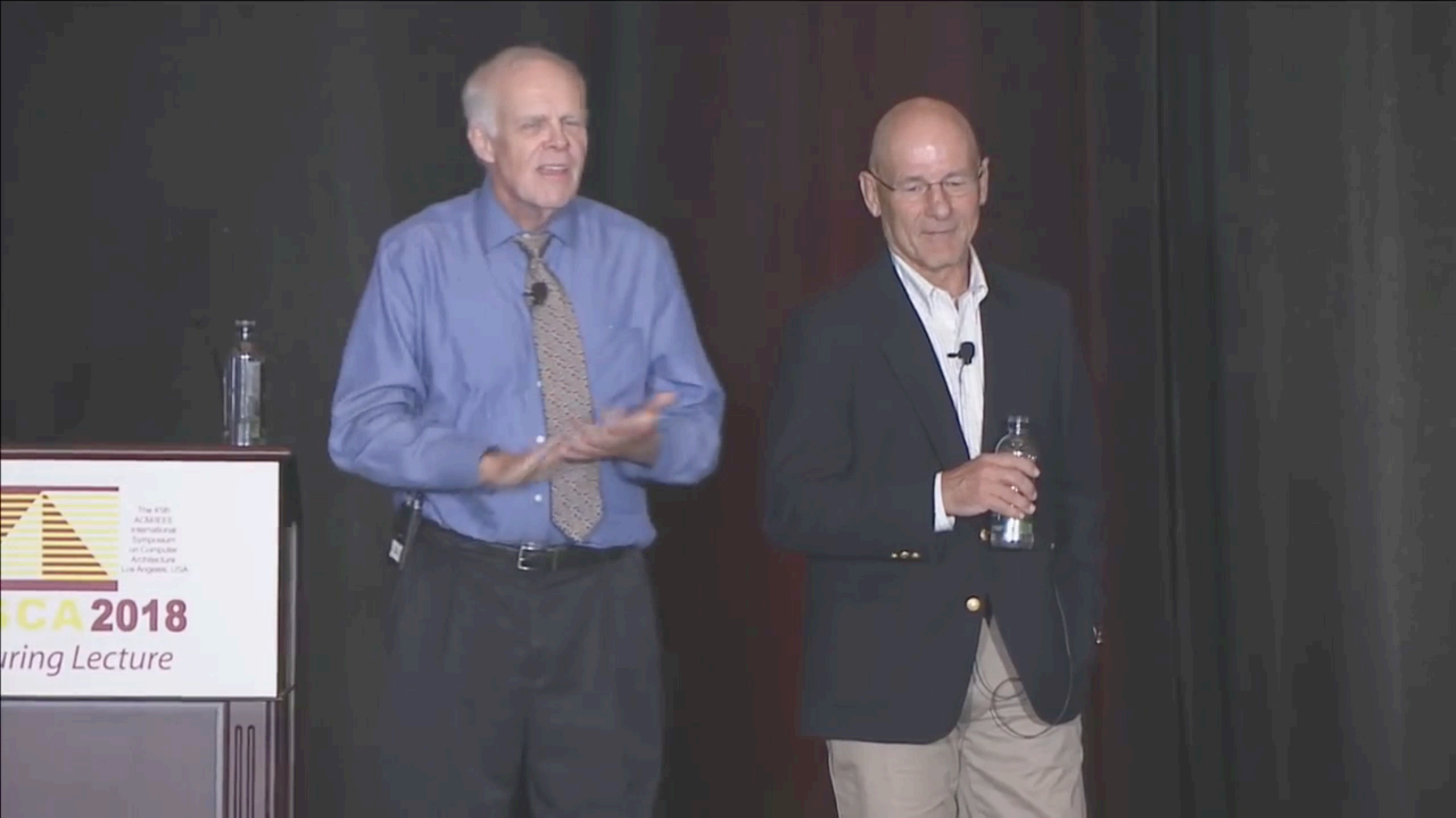
NVIDIA
Tensor Cores

Heterogeneous computer architecture



Processor/memory performance gap




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

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

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

	Topic	Reading	Due
9/26/2022	Introduction	- G.E. Moore. Cramming More Components Onto Integrated Circuits. Electronics, pp. 114–117, April 19, 1965. – Chapter 1.1-1.6	
9/28/2022	Performance Evaluation (I)	– Chapter 1.3 & 1.8-1.9	Reading Quiz #1
10/3/2022	Cancelled		
10/6/2022	Performance Evaluation (II)	– M. D. Hill and M. R. Marty, "Amdahl's Law in the Multicore Era," Computer, vol. 41, no. 7, pp. 33-38, July 2008. – V. Sze, Y. -H. Chen, T. -J. Yang and J. S. Emer. How to Evaluate Deep Neural Network Processors: TOPS/W (Alone) Considered Harmful. In IEEE Solid-State Circuits Magazine, vol. 12, no. 3, pp. 28-41, Summer 2020.	Reading Quiz #2
10/10/2022	Performance Evaulation (III)	- (Optional) Andrew Davison, Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers. in Supercomputer Review, MTBF 1995	Assignment #0 Due 10/7
10/12/2022	Memory Hierarchy (1): The	– Appendix B.1-B.3	Reading Quiz #3
10/17/2022	Memory Hierarchy (2)	– Appendix B.1-B.3, 2.3	Assignment #1 Due 10/14
10/19/2022	Memory Hierarchy (3):	– Chapter 2.3 – Norman P. Jouppi. 1990. Improving direct-mapped cache performance by the addition of a small fully-associative cache and	Reading Quiz #4
10/24/2022	Memory Hierarchy (4):		
10/26/2022	Virtual Memory	– Chapter B.4 & B.5, 2.4	Assignment #2 Due 10/28
10/31/2022	Midterm		
11/02/2022	Basic Processor Design &	– Chapter 3.3 – Appendix C.1 – C.4	Reading Quiz #5
11/07/2022	Advanced Branch Prediction	– M. Evers, S. J. Patel, R. S. Chappell and Y. N. Patt, "An analysis of correlation and predictability: what makes two-level branch predictors work," Proceedings. 25th Annual International Symposium on Computer Architecture (Cat. No.98CB36235), Barcelona,	Reading Quiz #6
11/9/2022	OOO Scheduling	– Chapter 3.4	Reading Quiz #7
11/14/2022	OOO Scheduling	– K. C. Yeager, "The MIPS R10000 superscalar microprocessor," in IEEE Micro, vol. 16, no. 2, pp. 28-41, April 1996. – R. E. Kessler, "The Alpha 21264 microprocessor," in IEEE Micro, vol. 19, no. 2, pp. 24-36, March-April 1999.	
11/16/2022	OOO Scheduling		Assignment #3 Due 11/18
11/21/2022	SMT & Chip Multiprocessors	– Chapter 3.11 – Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor, Dean M. Tullsen,	Reading Quiz #8
11/23/2022	Chip Multiprocessors	– The case for a single-chip multiprocessor, Kunle Olukotun, Basem A. Nayfeh, Lance Hammond, Ken Wilson, and Kunyung Chang, SIGOPS OSR Vol. 31(S): 2-11, 1996.	
11/28/2022	Dark Silicon	– Chapter 1.7 – H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam and D. Burger. Dark Silicon and the End of Multicore Scaling. In IEEE	Reading Quiz #9
11/30/2022	TPU, FPGA	– Norm P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaman Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordie MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Cristian Tomasiello, Bob Todd, Ben Seshasena, Jonathan Vaitsas, Vinod Vasudevan, Emad Wachsman, Walter Wang, Shibin Yan, Cliff Zhou, Doe Hyun Yoon, Ed Lucero, Omar Lacy, Gregory Mathias, Anthony Nguyen, Michael Patterson, Alan Perico, Pardeep Subhedar, Cedric Valescuza, Ashish Narasimhan, Gaurav Agrawal, Minan Bayiates, Matt Dau, Poorna Kale, Beth Gianos, Erick Tuttle, Eric Wilcox, and Hyo Eun Yoon, "A domain-specific supercomputer for training deep neural networks," Communications of the ACM, 63.7, 67-78, 2020. – Kevin M. Lepak, Gordon B. Bell, and Mikko H. Lipasti. Silicon Implementation of the PowerPC Microprocessor. In HPCA, 1998. – Ronny Ronen, Avi Mendelson, Kattalin Lai, Shih-Lien Lu, Fred Pollack, and John Shen. Coming Challenges in Microarchitecture and Architecture. In IEEE, 2001. – Ronny Ronen, Avi Mendelson, Kattalin Lai, Shih-Lien Lu, Fred Pollack, and John Shen. Coming Challenges in Microarchitecture and Architecture. In IEEE, 2001. – Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. Memory Hierarchy and Prefetching for Energy-Scalable Tiled Chip Multiprocessors. In SBAC-PAD, 2010. – Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek Hower, Tushar Krishna, Sooraj Sardashti, Rathijit Sen, Korey Sewell, Muhammad Shoaib, Nilay Vaish, Mark D. Hill, and David A. Wood. The gem5 simulator. In SIGARCH Computer Architecture News, 2011. – Nathan L. Binkert, Ronald G. Dreslinski, Lisa R. Hsu, Kevin T. Lim, Ali G. Saidi, and Steven K. Reinhardt. The M5 simulator: Modeling networked systems. In MOBS, 2006. – Jason Lowe-Power, Abel Mulugeta, Mohyee Mikkami, Dan Sunderland, Matt Poremba, Yanghui Li, Ayaz Akram, Marjan Fariborz, Bradford Beckmann, Matt Sinclair, Matt Wilcox, Matthew Poremba, Cathy Mauzy, Anthony Gutierrez, Lena Olson, Matt Decandido, Muhammad Hamdi, Qiongqiong Zhou, Hongil Yoon, Carlos Herrera, Chun Chen, Ver Kazakova, Glenn Correll, Chris Hsieh, Nathanael Premillieu, Paul Rosenfeld, Marisa Vega, Timothy Ramirez, Tony Mysselwhite, Erfan Azarkhish, Fabien Chaix, Matheus Cavalcante, Pouya Fotouhi, Matthew Hoseklis, Baxishi Olippuzha, Matthew Poremba, Yasuko Eckert, Xiangyu Dong, Wendy Elsasser, Jagi Gal-On, Marco Elver, Jeff Furlong, Nathan Gab極, Anthony Jaleel, Malek Musleh, Seth Pugsley, Vilas Sridharan, Lisa Hsu, Kaustav Goradia, Nathan Tallent, Brad Beckmann, Marc Orr, Shaizeen Aga, Jieming Yin, Daniel Sorin, Shan Li, Alan Ji, Erik Hallnor, Tuan Ta, Tiyasa Mitra Mandal, Blake Hechtman, Alex Dutu, Noah Beck, Kyle Roarty, Mark Hildebrand, Brandon Potter, Abdul Mujeeb, Shaoni Mukherjee, Johnathan Alsop, Timothy Sorenson, Matthew Lyle Fronsdal, Izhak Bratchel, Anthony Mayer, Broderick Gardner, Brad Integlia, Yuan Hsi Chen, Matt Evanusa, William Wu, Eric Hoster, Yunsup Lee, Colin Bruce, Shaunak Shah, Daniel Hower, Joseph Gross, Steve Reinhardt, Jason Agron, Ivan Porres, Ilkka Rauta, Olli-Pekka Mäkitalo, Kalle Valo, Shamel Sunshine, Olli Pitkanen, Tuan Dinh Tran, Anthony Battaglia, Matthew Haney, Kaeden Hamilton, Shunde Zhang, Paul Rosenfeld, Matt Sinclair, Christopher Ramirez, Kaz Kylheku, Anthony Gutierrez, Marco Elver, Anthony Jaleel, Matt Horsnell, Geoffrey Blake

CS203: reinventing your Learning eXperience

Active learning: in-person lecture experience

Conventional lectures

Me
I expect the lecture to be...
You

Peer instruction

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

Read **Think** **Discuss**

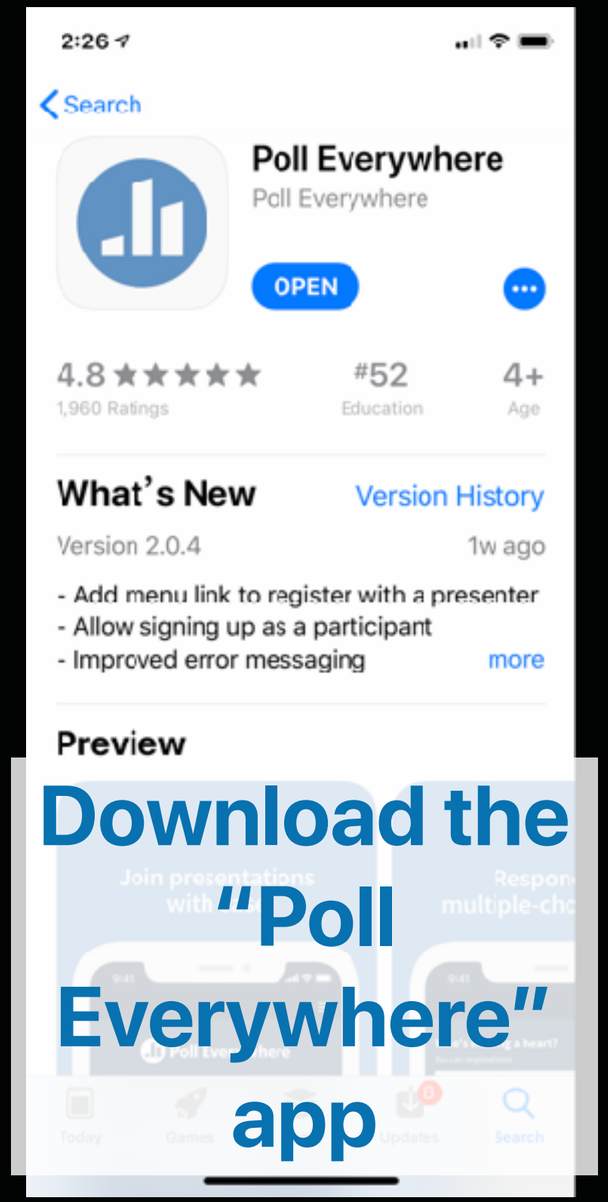
Peer instruction

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

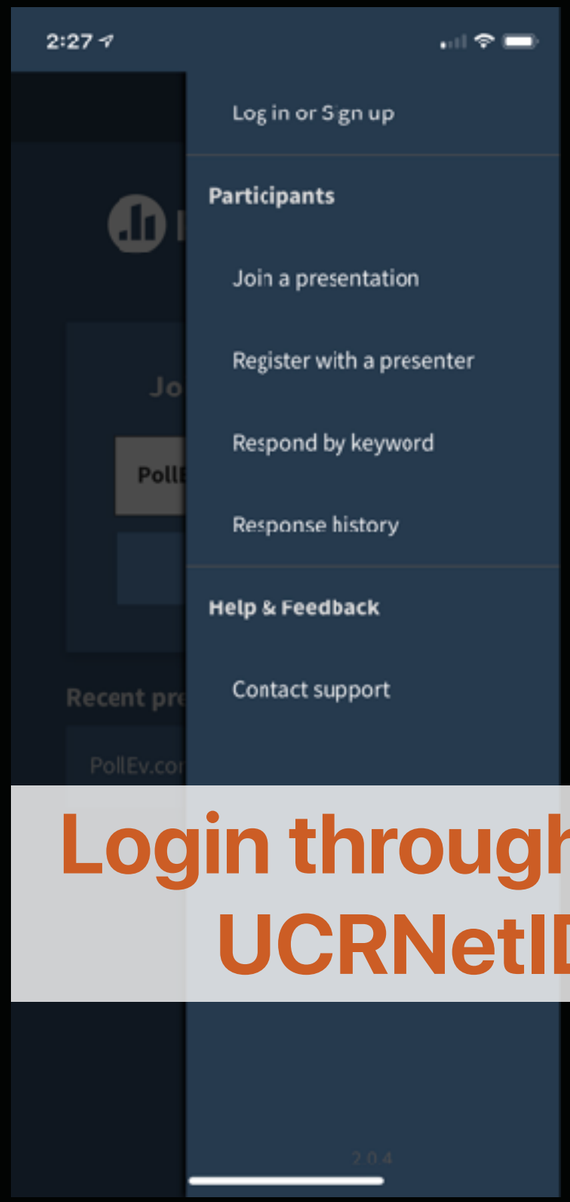
Why reading quizzes?

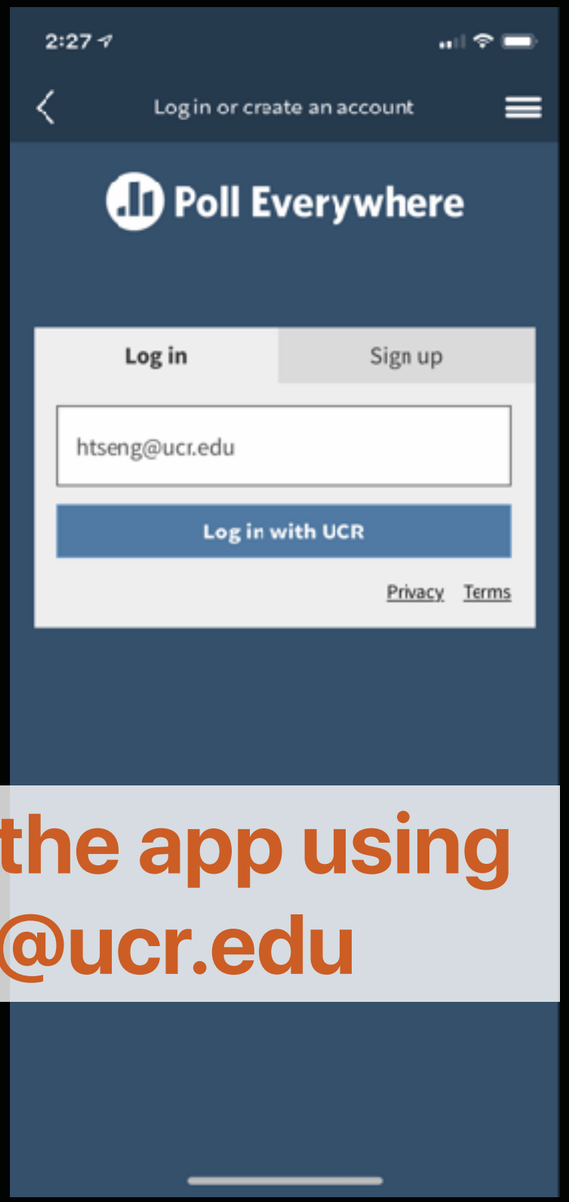
- We need to prepare you for peer instruction activities and discussions!
- Reading assignments from
  - Computer Architecture: A Quantitative Approach (6th Edition) by John Hennessy and David Patterson
  - Papers
- Reading quizzes:
  - On eLearn
  - Due before the lecture, usually once a week. Check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

About the time of the Lecture — Setup Poll Everywhere

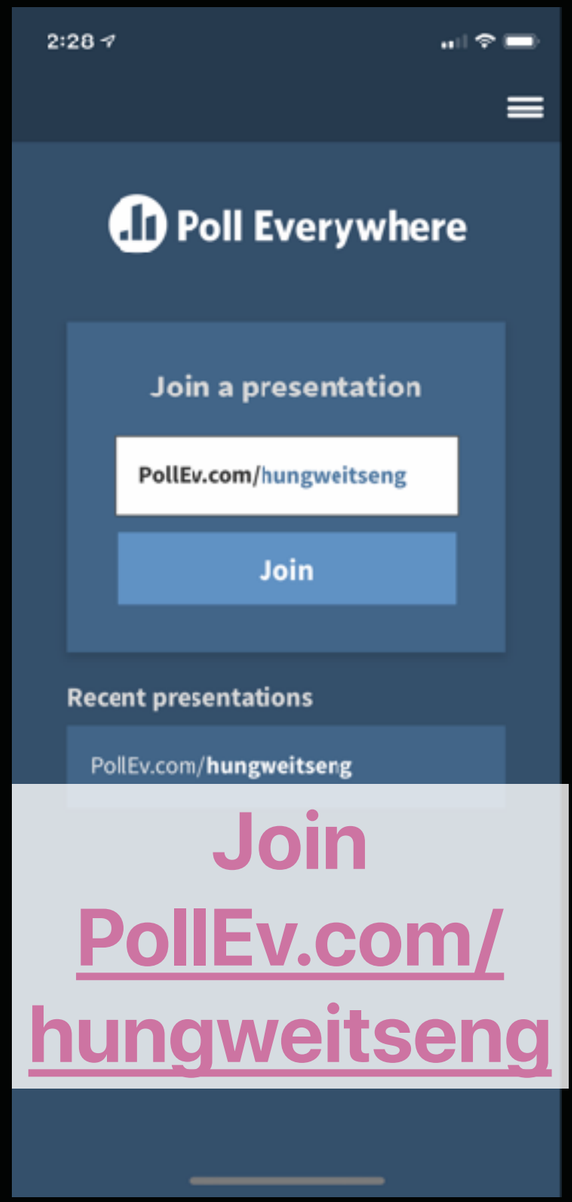


Download the "Poll Everywhere" app





Login through the app using UCRNetID@ucr.edu



Join PollEv.com/hungweitseng

Grading breakdown

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

Jupyter notebook based assignment — learning & reviewing by practicing

For every assignment

- Go to course home page: https://www.escalab.org/classes/cs203-2022fa/
- Click invitation link for the upcoming assignment
- Connect to UCR VPN
- Log into https://www.escalab.org/datahub
- Select this option and click "Launch Environment"
- Open a terminal
- Clone your starter repo.
- Open up Assignment.ipynb
- Follow the instructions

Course Infrastructure: Github and github classroom

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

Course Infrastructure: Bare metal Servers in The Cloud

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

Course Infrastructure: Jupyter Notebook

- A large part of each lab is done in a Jupyter Notebook
  - Jupyter Notebook is a web-based, interactive computing environment
  - It's good for collecting and visualizing data
- 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

File Edit View Run Kernel Tabs Settings Help

ASSIGNMENT.IPYNB

1. Assignment 1: The Performance Equation

2. FAQ and Updates

3. Browser Compatibility

4. About Assigments In This Class

4.1. How To Succeed On the Assignments

4.2. Getting Help

4.3. Posting Answerable Questions on Piazza

4.4. Keeping Your Lab Up-to-Date

4.5. How To Use This Document

4.5.1. Running Code

4.5.2. Telling What The Notebook is Doing

4.5.3. What to Do Jupyter Notebook It Gets Stuck

4.5.4. Common Errors and Non-Errors

4.5.5. Useful Tips

Launcher | Assignment.ipynb

Markdown | Python 3 (ipykernel)

3200MHz/1550Mhz ratio

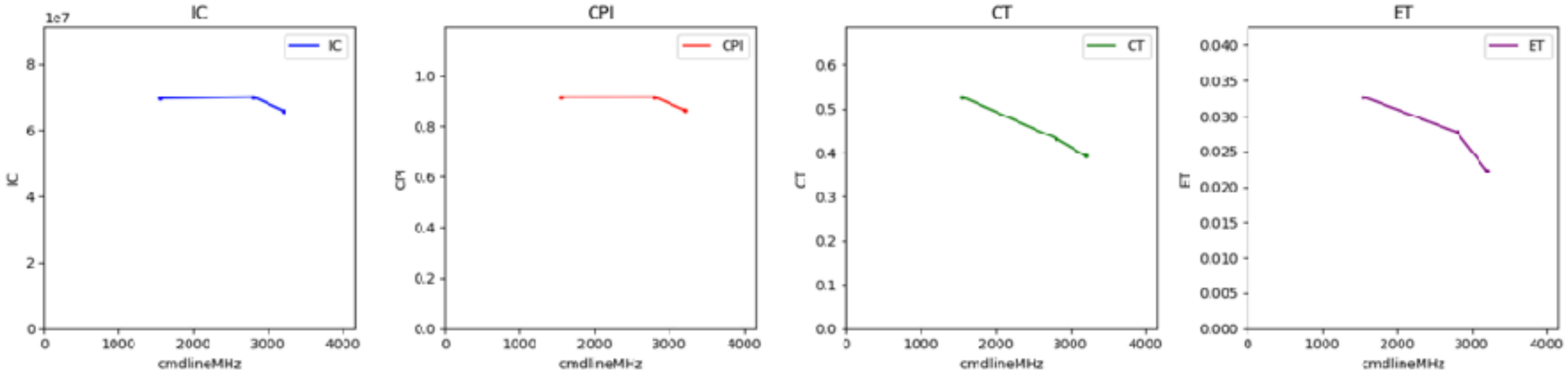
[71]:
```
!cs203 job run './microbench.exe -o cycle_time.csv -M 1550 2800 3200  -f baseline_int -r 50'
```

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

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

[72]:

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



Simple 0 0 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 Assignment.ipynb

In each assignment

- There are 20-30 questions per lab.
- Correctness
  - Demonstrate mastery
  - Give the right answer — earn points
- • Completeness
  - "forcing function" to get you to engage with the material
  - Give an answer — earn points
  - We will grade ~5 of these at random per assignment.
- Optional
  - Optional material for interested students.
  - Give the right answer — earn a sense of personal accomplishment

Course Infrastructure: cse142

- ‘cs203’ is a command line tool you’ll use to run your code.
  - It can do several things.
  - The most common is something like
- cs203 job run “echo hello world”
- What this does:
  - Gathers up all the files in the directory
  - Ships them to the bare metal machine in the cloud
  - Runs your command
  - Gathers up any files that changed.
  - Ships them back to you.
- It looks mostly like “echo hello world” ran locally.

Submitting your assignment

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

Quick walk through of your 0th assignment

Warnings

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

Logistics

Course resource

- Lectures:
  - MW 9:30a—10:50a @ SSC 329
- Living streaming, video recording https://www.youtube.com/profusagi
- Schedule, slides on **course webpage**: https://www.escalab.org/classes/cs203-2022fa/
- Discussion on **piazza**: https://piazza.com/class/l7ar19cgxtt4oq
- Reading quizzes, homework submissions on **eLearn**: https://elearn.ucr.edu

The website

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

Instructor — Prof. Usagi (a.k.a. Hung-Wei Tseng)

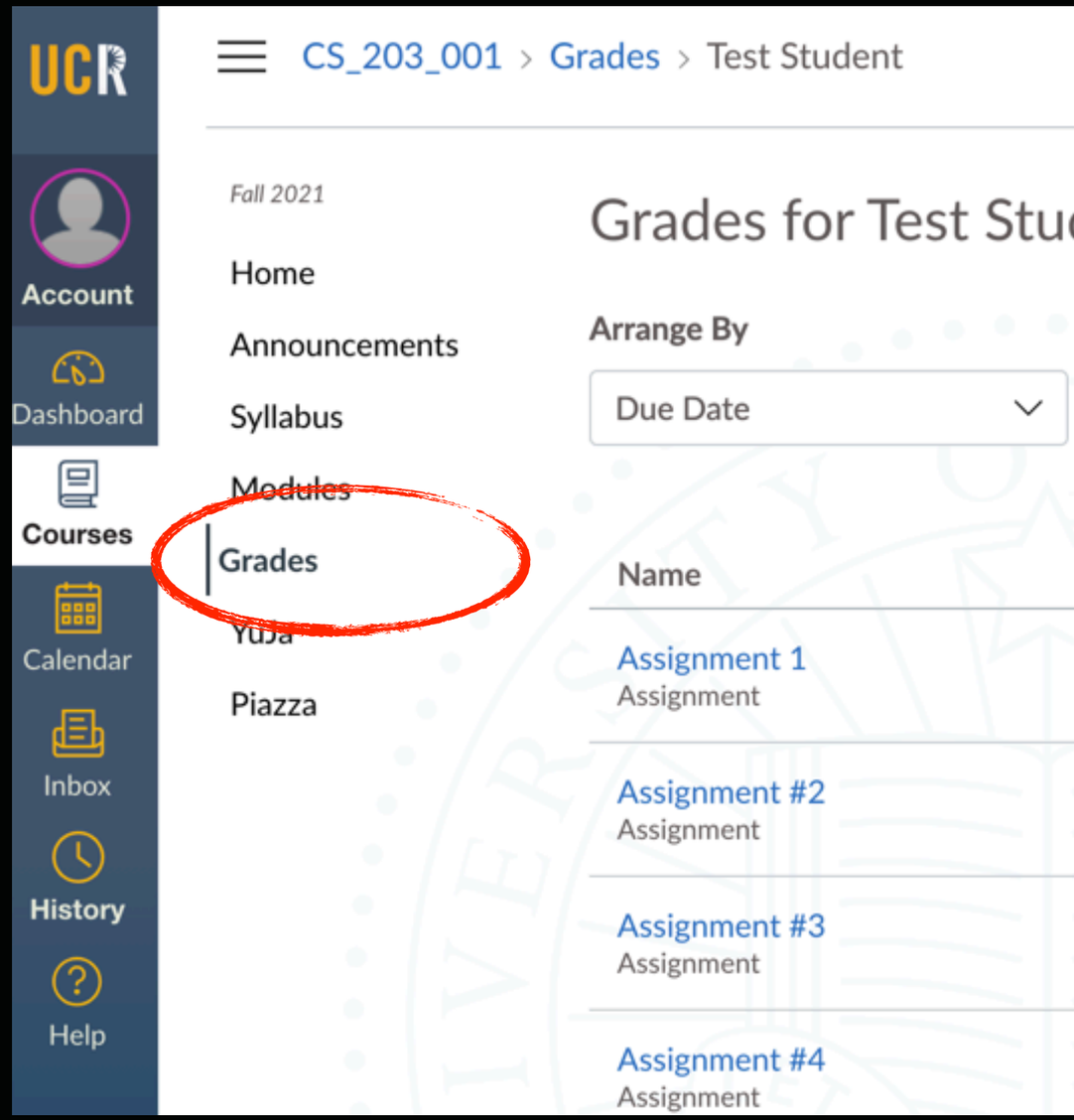
- Website: https://intra.engr.ucr.edu/~htseng/
- E-mail: htseng @ ucr.edu
- PhD in **Computer Science**, University of California, San Diego
- Research Interests
  - General-purpose computing on AI/ML/NN accelerators
  - Intelligent storage devices & near-data processing
  - Or anything else fun — we have an OpenUVR project recently
- Office hour: TuW 11a-12p @ WCH 406 or through Zoom (will share the link on eLearn)

Teaching Assistant — Jinyoung Choi

- Office hours: M 3p-4p & Th 2p-3p
- E-mail: jchoi264 @ ucr.edu

Grading

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



Academic Honesty

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

Hey, I need help…

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

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

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

Hey, I need help… (part 3)

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

Hey, I need help … (part 4)

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

Why…

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

Term of Service

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

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

UC San Diego
NC STATE UNIVERSITY
UC RIVERSIDE
2012 Summer
2016 Fall
You
2022 Fall
2014 Summer
2017 Sp
2021 Fall
2022 Spring
2017 Fal
2019 Fall
2016 Spring
2018 Spring
Questions & Comments?
2021 Spring
2021 Winte
2016 Summer
2018 Fall
2020 Summ
2019 Spring
UC ZOOM
2019 Summer I
2019 Summer II
2020 Spring

Computer
Science &
Engineering

203

つづく