First Day of CS203: Advanced Computer Architecture

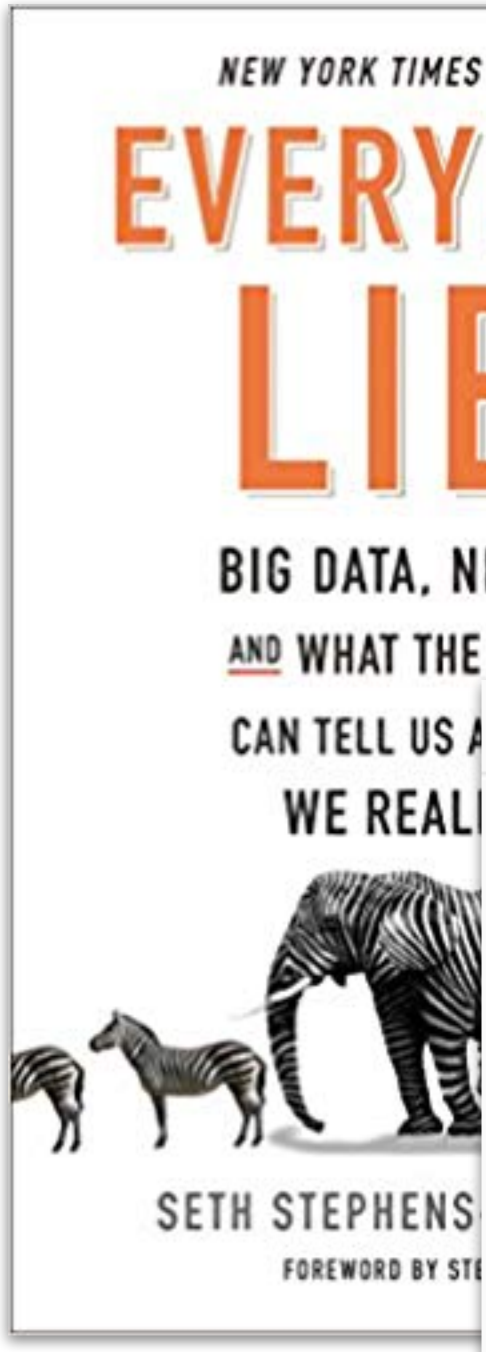
Hung-Wei Tseng

CS203: Let’s say something!

What’s your name?

What’s your favorite topic in computer science?

Why’re you taking CS203



NEW YORK TIMES BESTSELLER

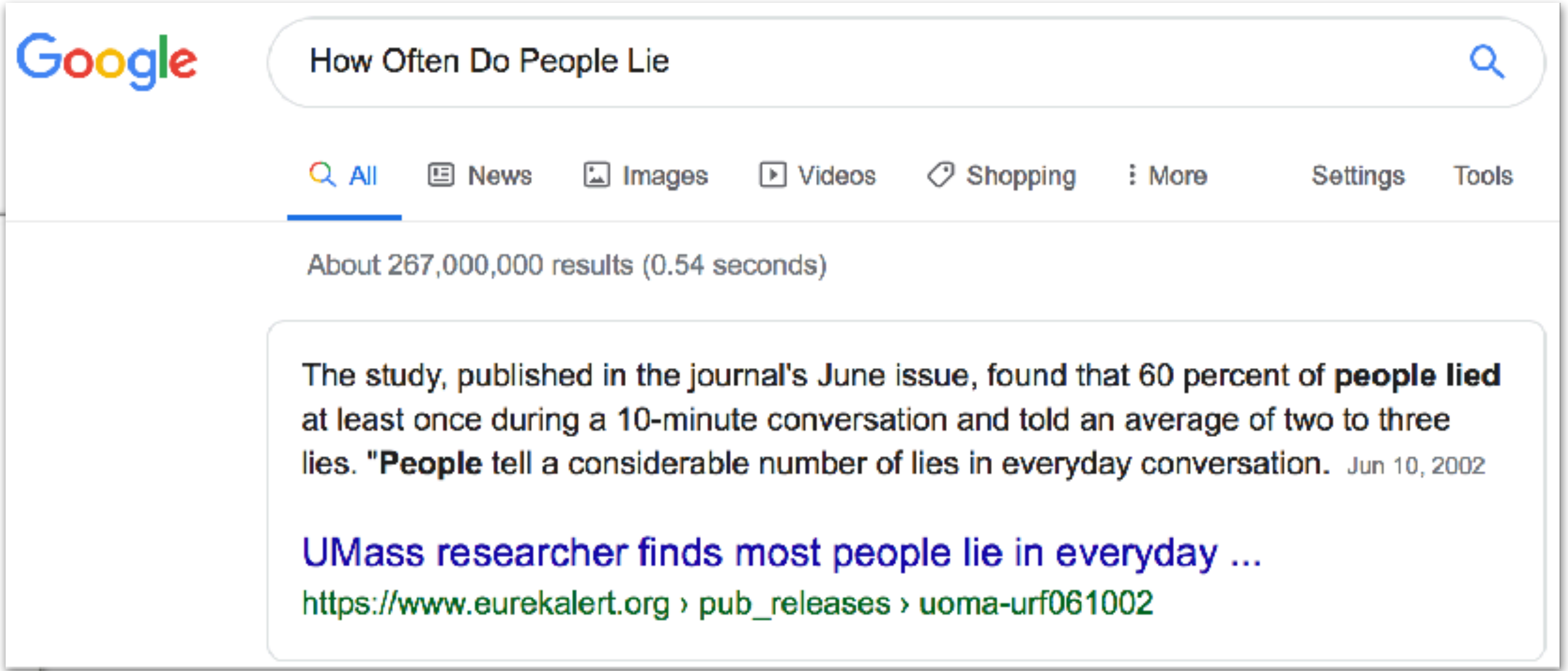
EVERYBODY LIES

BIG DATA, NEW DATA, AND WHAT THE INTERNET CAN TELL US A

WE REAL

SETH STEPHENS

FOREWORD BY STE



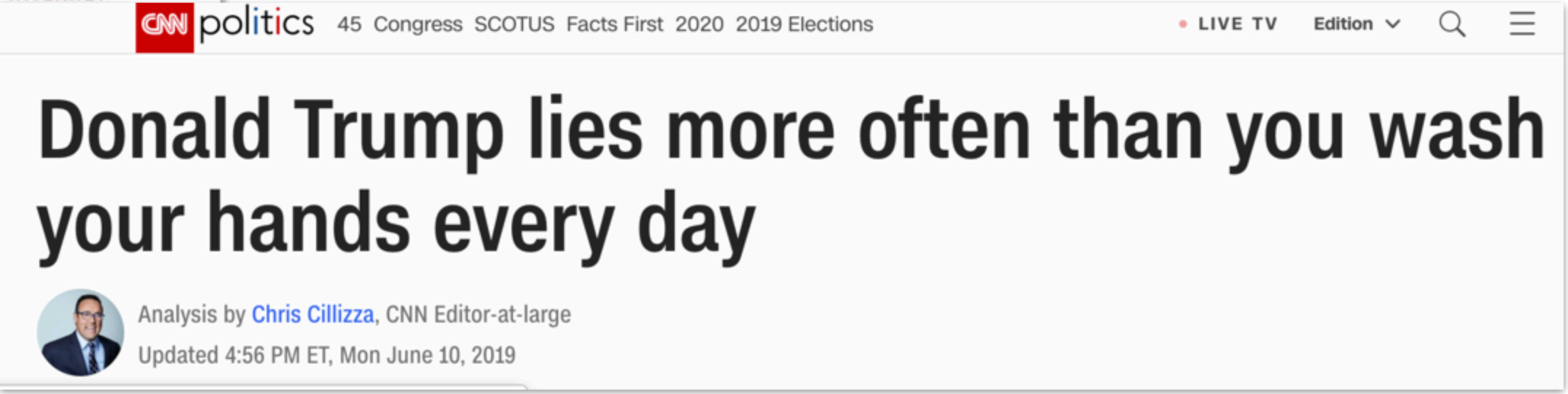
How Often Do People Lie

About 267,000,000 results (0.54 seconds)

The study, published in the journal's June issue, found that 60 percent of **people lied** at least once during a 10-minute conversation and told an average of two to three lies. "**People** tell a considerable number of lies in everyday conversation. Jun 10, 2002

UMass researcher finds most people lie in everyday ...

https://www.eurekalert.org › pub_releases › uoma-urf061002



Donald Trump lies more often than you wash your hands every day

Analysis by Chris Cillizza, CNN Editor-at-large

Updated 4:56 PM ET, Mon June 10, 2019

CS203: Let’s say something!

What’s your name?

What’s your favorite topic in computer science?

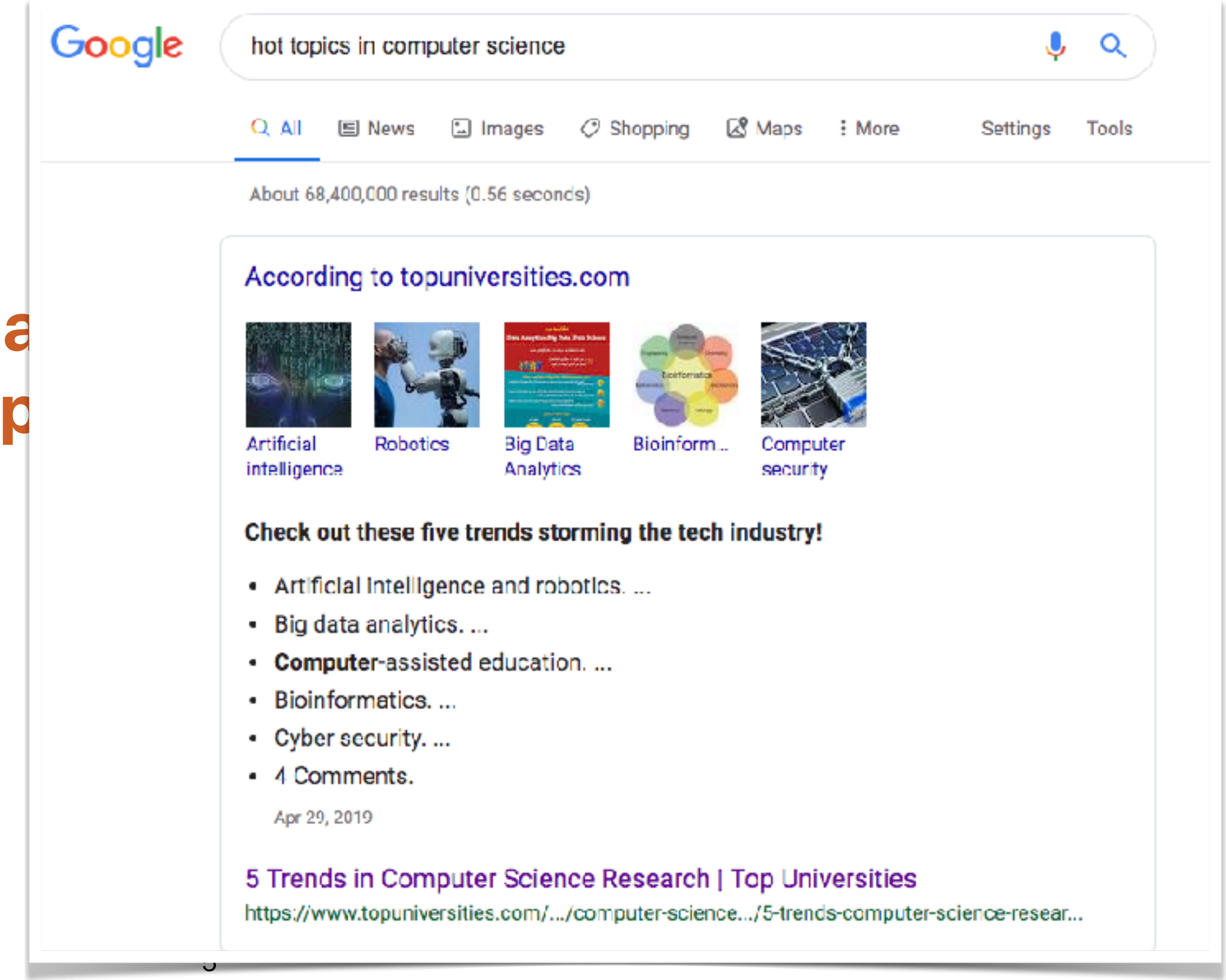
Why’re you taking CS203

CS203: Let’s say something!

What’s your name?

Wha

top



hot topics in computer science

About 68,400,000 results (0.56 seconds)

According to topuniversities.com

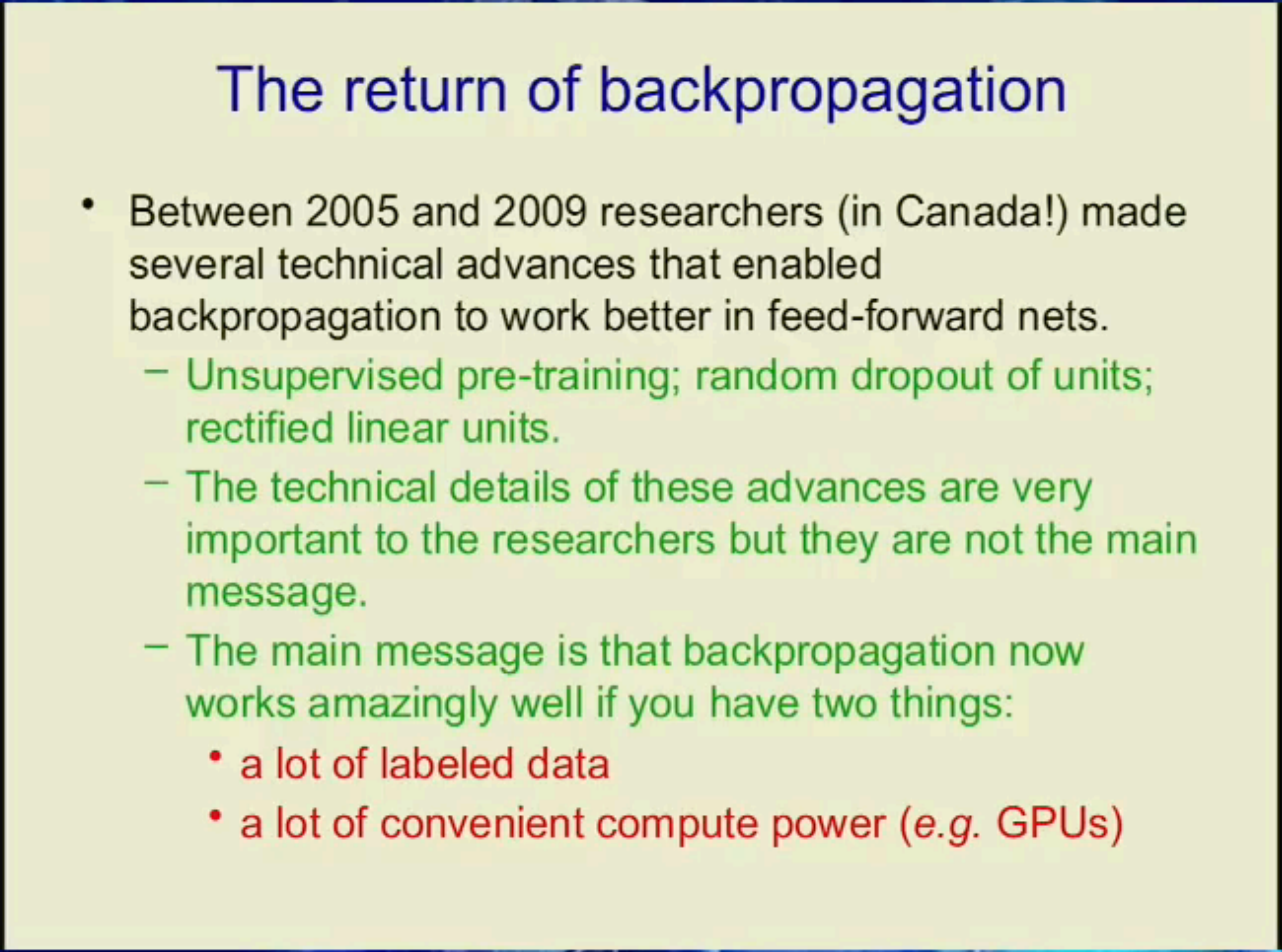
Check out these five trends storming the tech industry!

- Artificial Intelligence and robotics. ...
- Big data analytics. ...
- **Computer**-assisted education. ...
- Bioinformatics. ...
- Cyber security. ...
- 4 Comments.

Apr 29, 2019

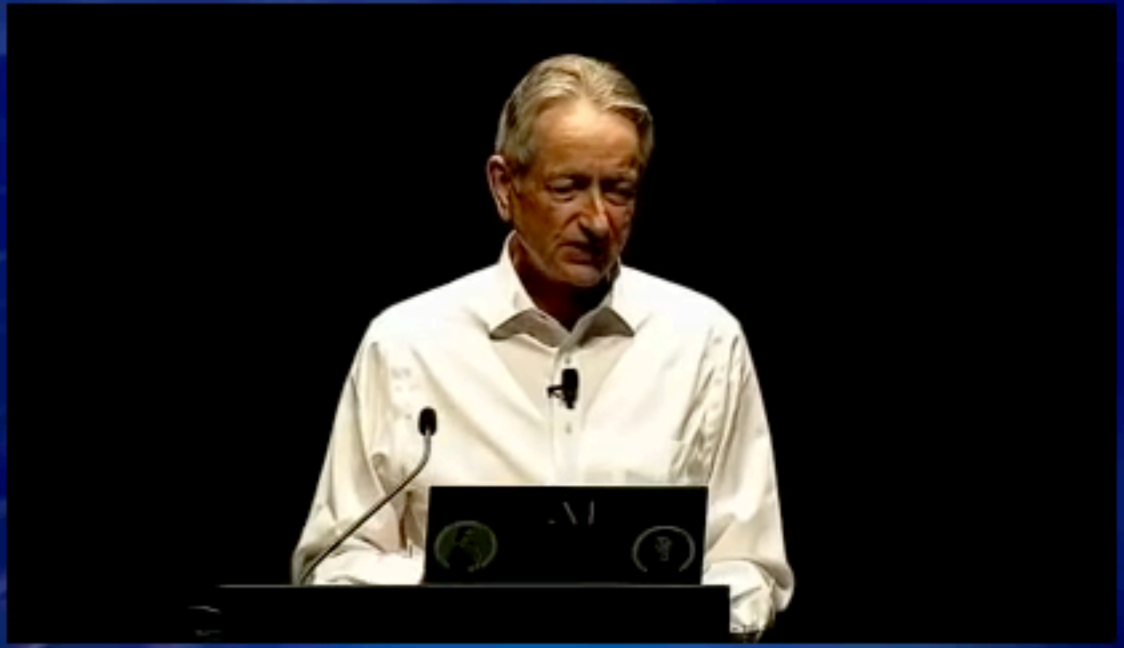
5 Trends in Computer Science Research | Top Universities

https://www.topuniversities.com/.../computer-science.../5-trends-computer-science-resear...



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)



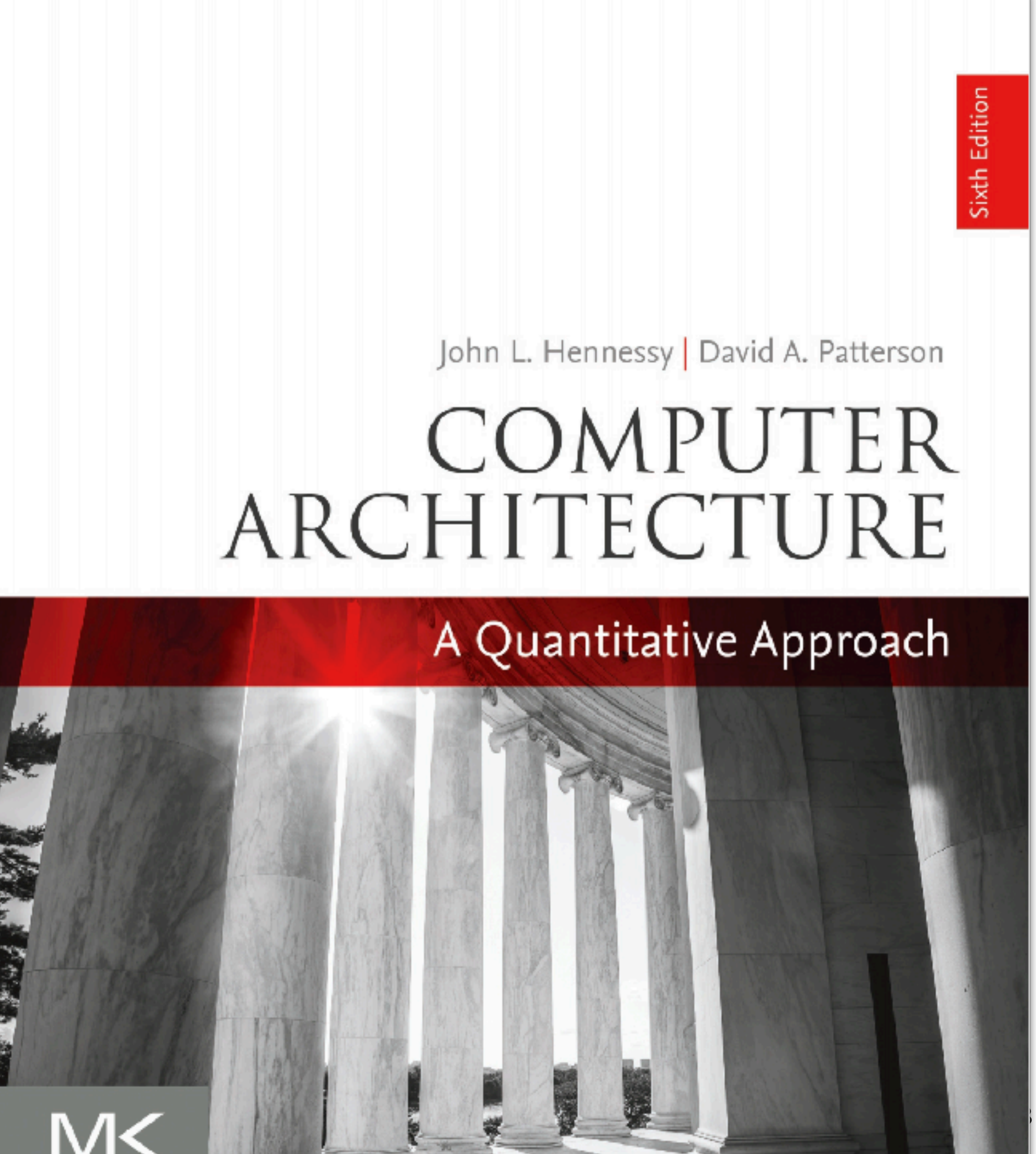
acm FCRC

2018 Turing Award

Hung-Wei Tseng

David Patterson

John Hennessy



Sixth Edition

John L. Hennessy | David A. Patterson

COMPUTER ARCHITECTURE

A Quantitative Approach

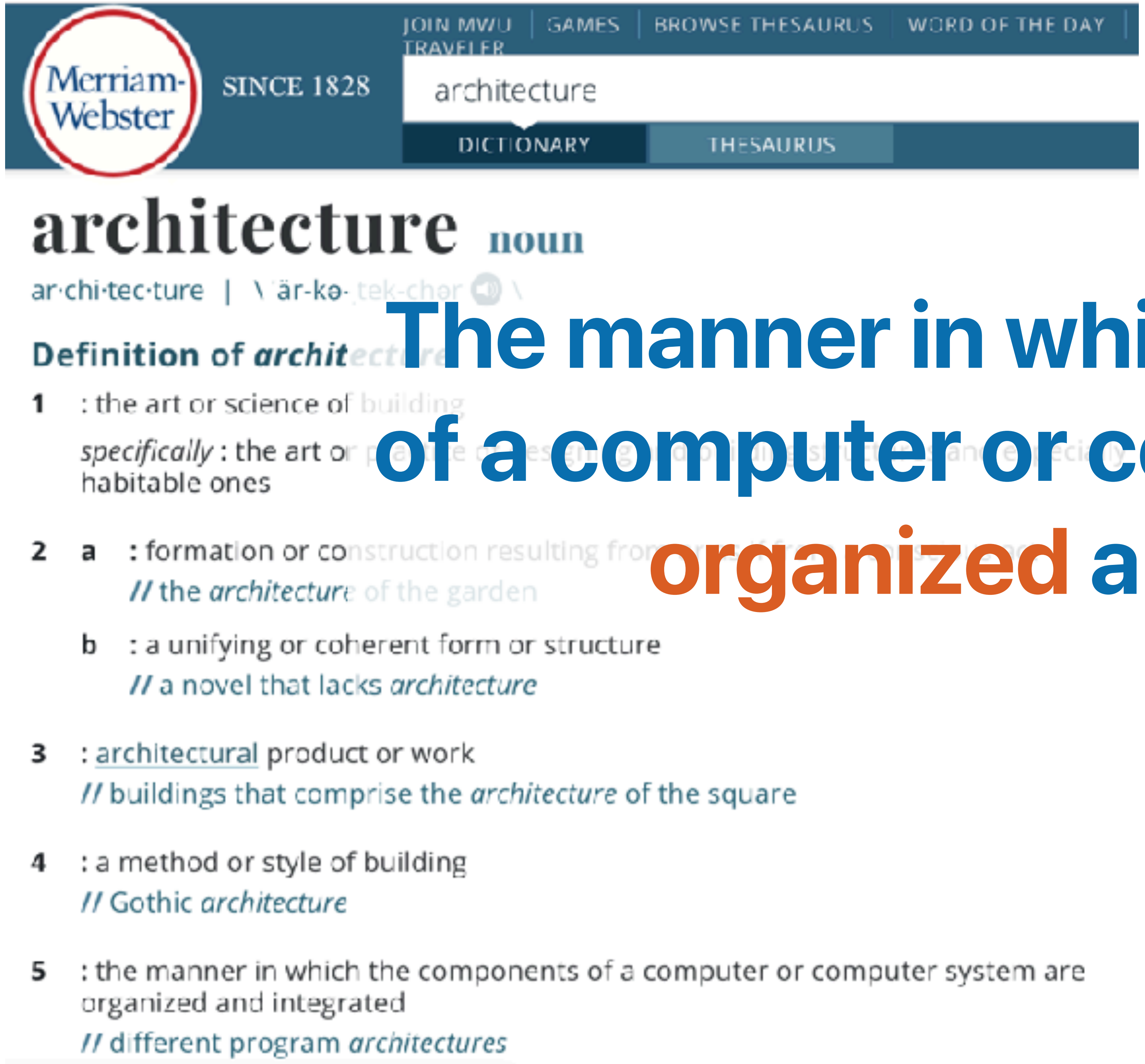
Computer Architecture

Enables

Deep Learning

Computer architecture also enables …

What’s computer architecture?



architecture noun

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

Definition of architecture

1 : the art or science of building

specifically : the art or practice of designing and building structures and especially habitable ones

2 a : formation or construction resulting from or as if from a conscious act
// 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

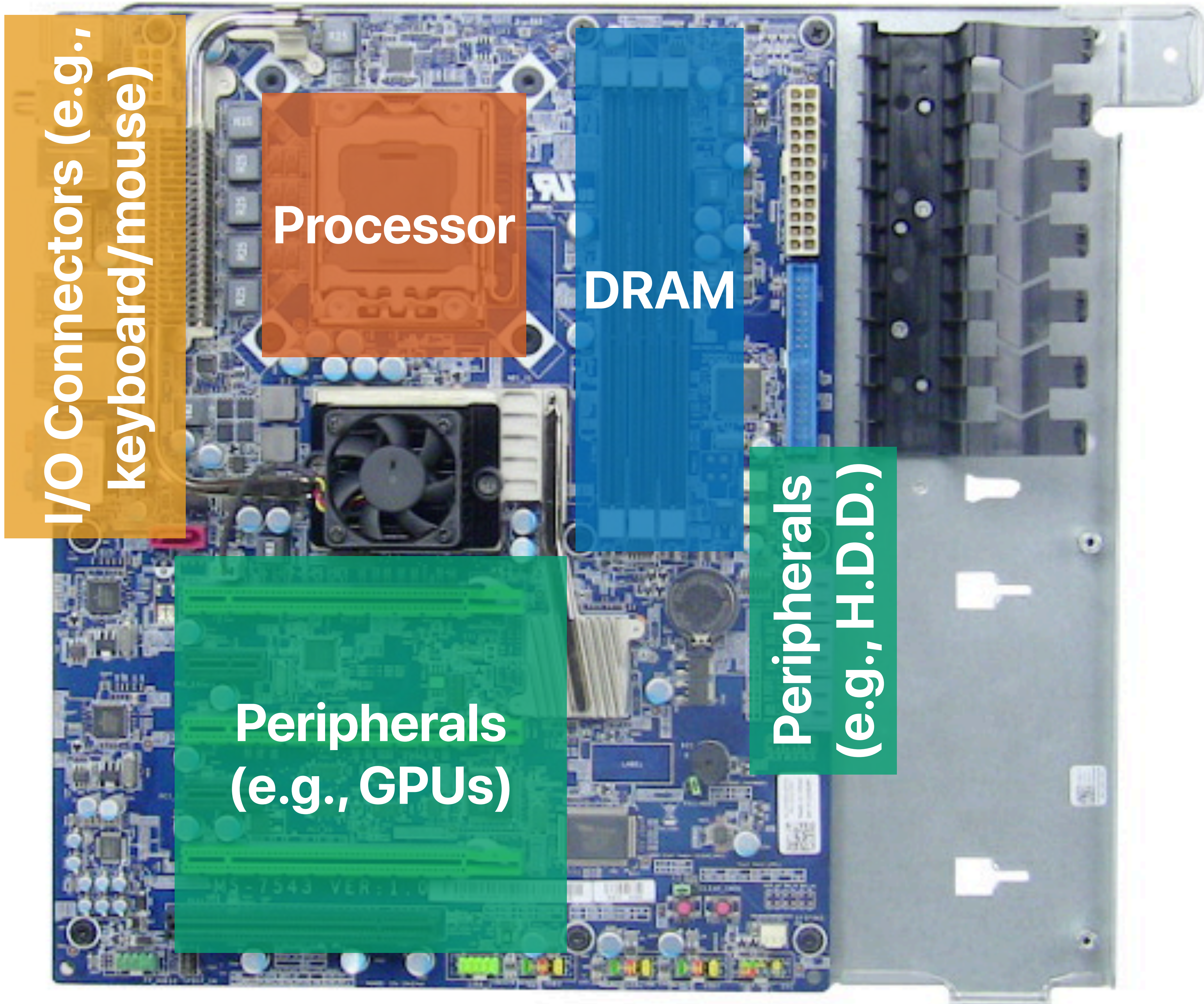
4 : a method or style of building
// Gothic *architecture*

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

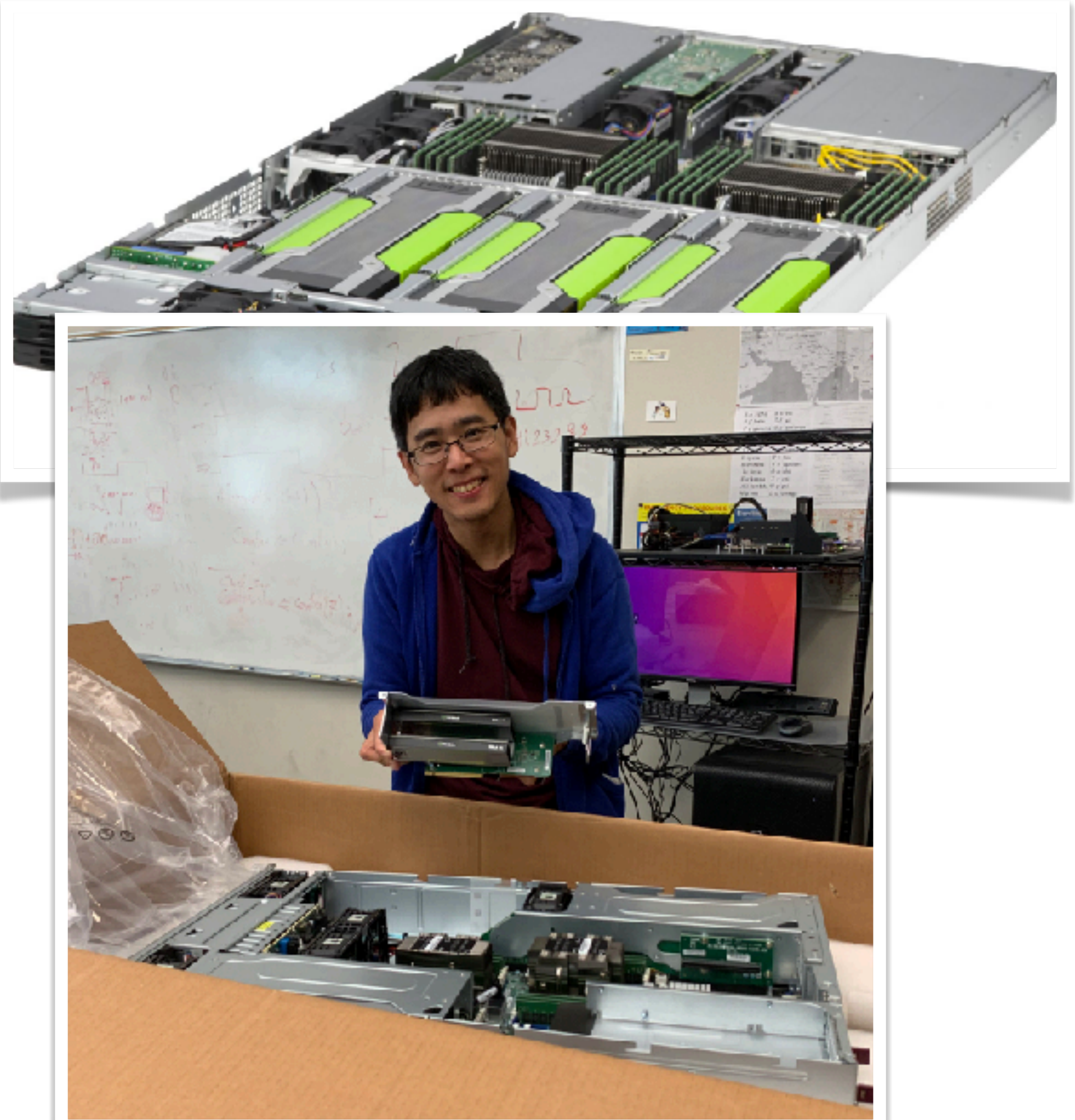
The manner in which the components of a computer or computer system are organized and integrated

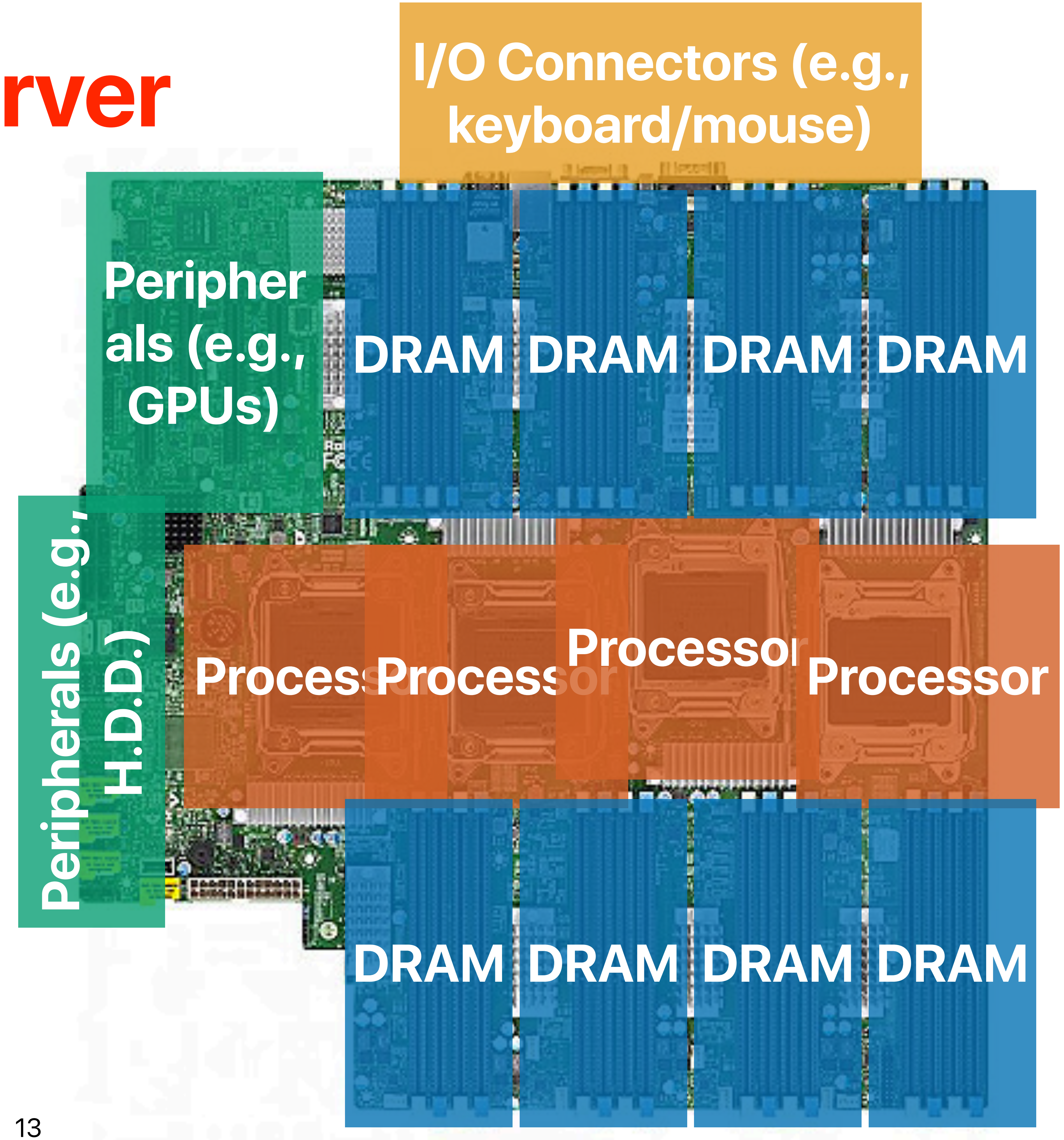
What’re those “components”?

Desktop Computer

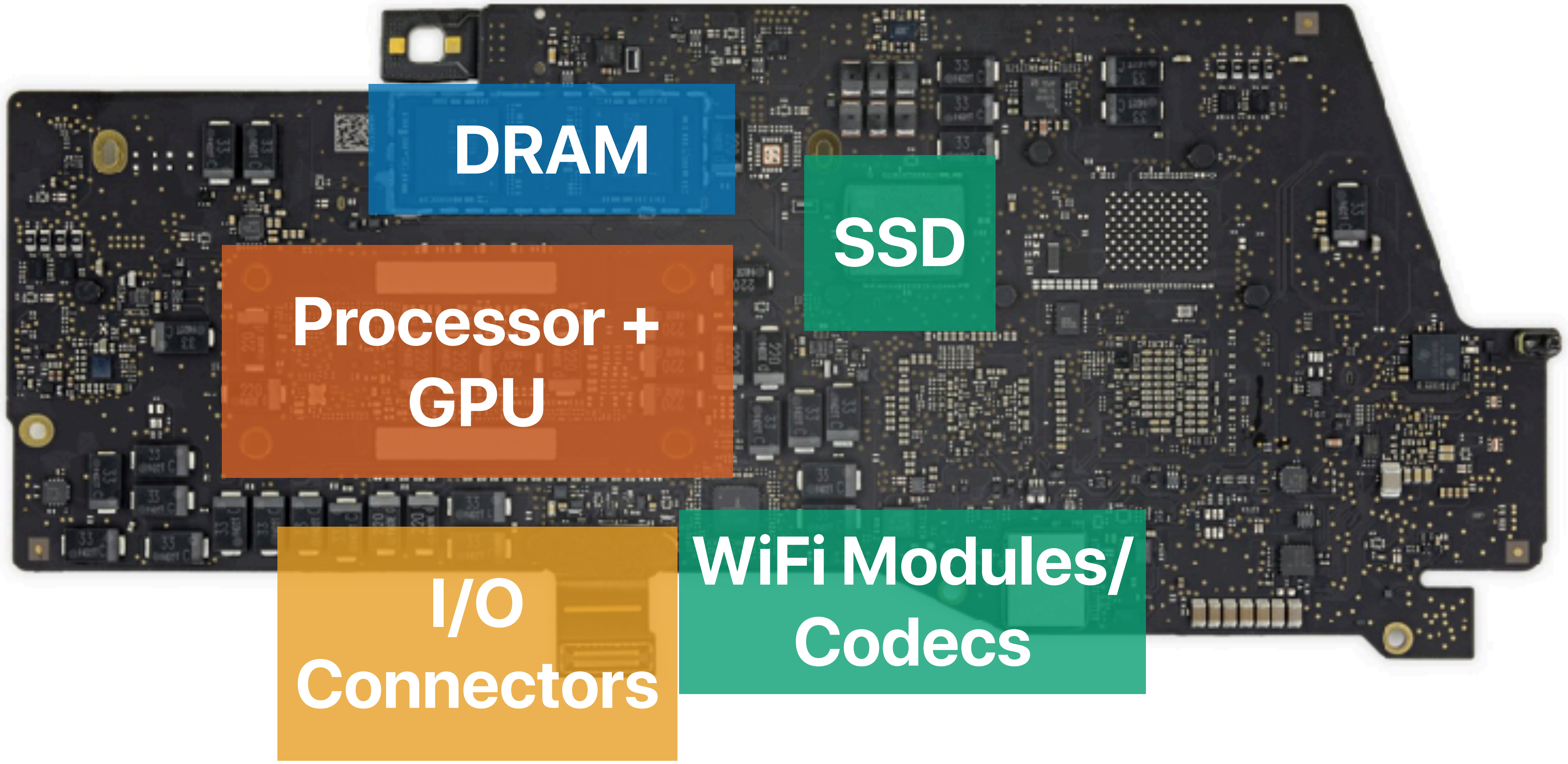

I/O Connectors (e.g., keyboard/mouse)
Processor
DRAM
Peripherals (e.g., GPUs)
Peripherals (e.g., H.D.D.)

Server

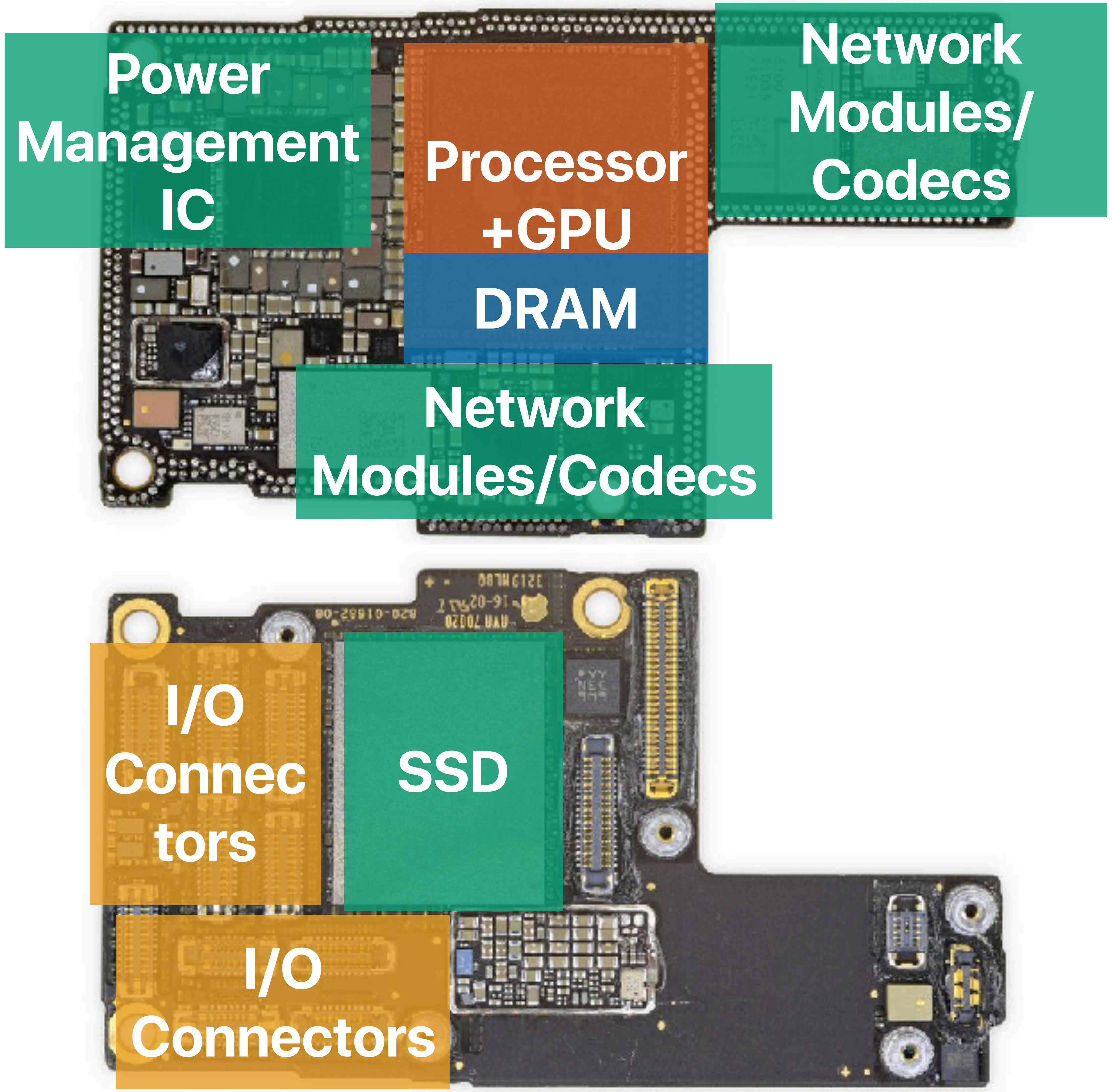



I/O Connectors (e.g., keyboard/mouse)
Peripherals (e.g., GPUs)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Processor
Processor
Processor
Processor
DRAM
DRAM
DRAM
DRAM

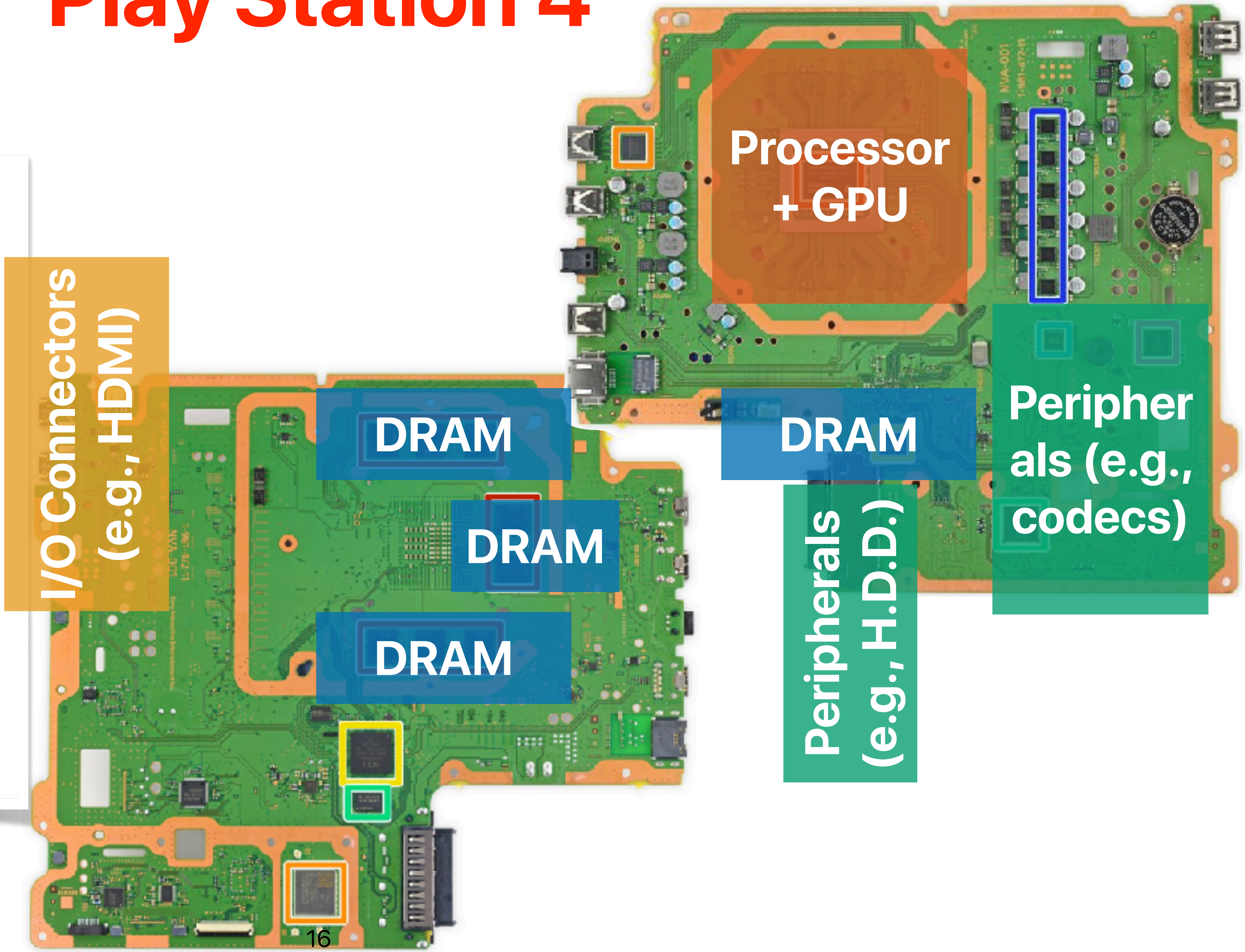
MacBook Pro 13"



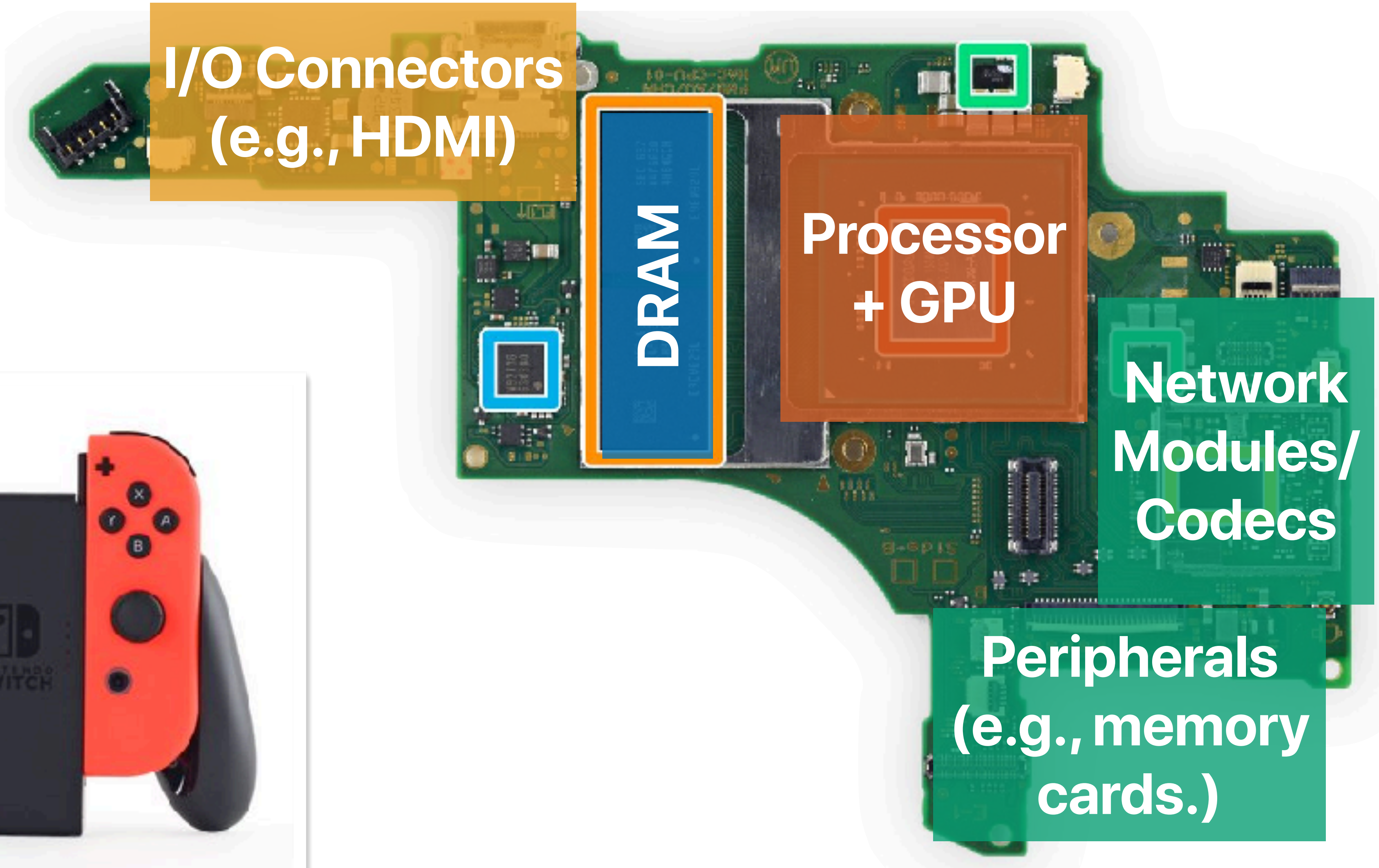
iPhone 11 Pro



Play Station 4


Processor + GPU
I/O Connectors (e.g., HDMI)
DRAM
DRAM
DRAM
DRAM
Peripherals (e.g., H.D.D.)
Peripherals (e.g., codecs)

Nintendo Switch



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

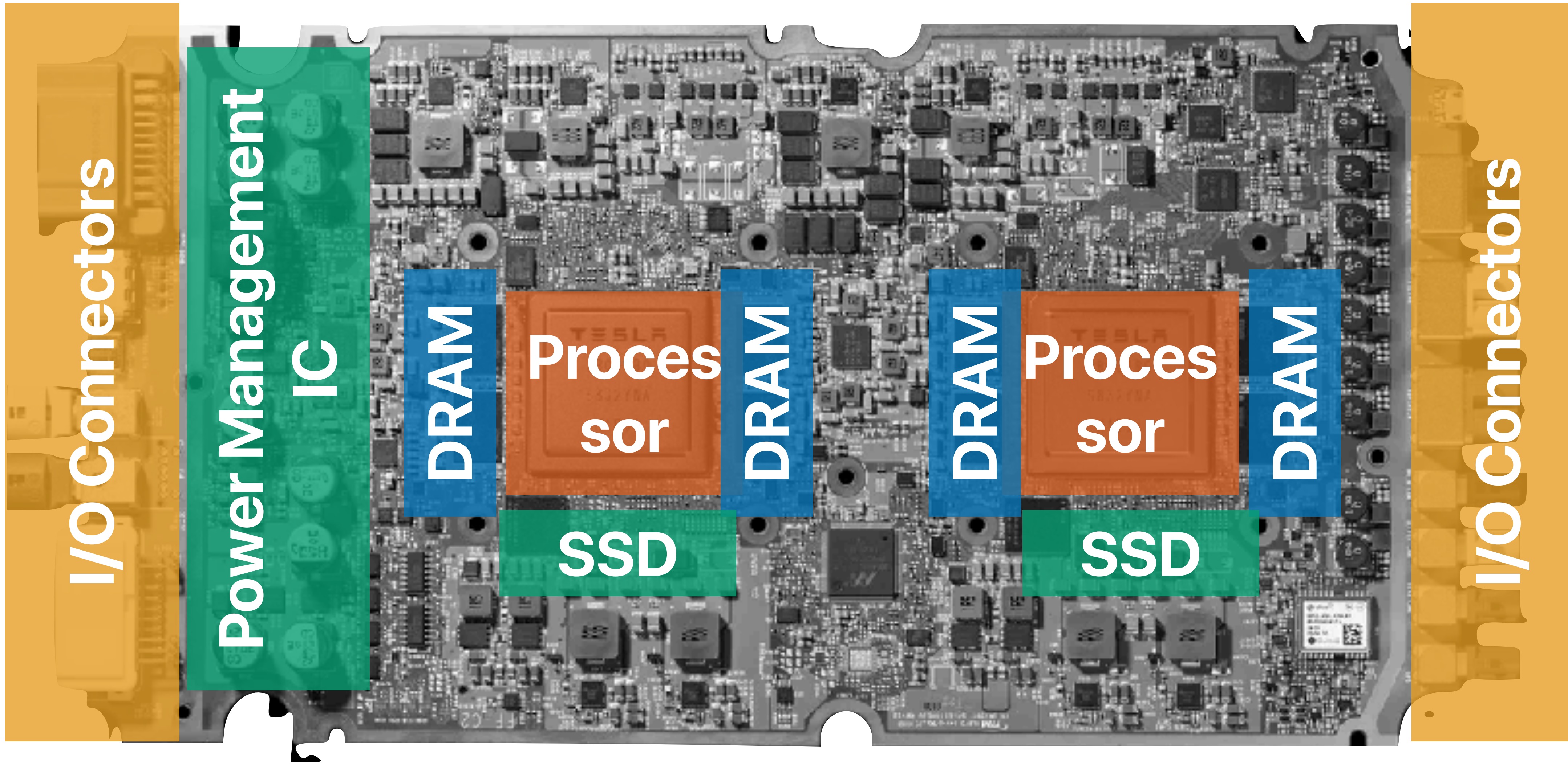
DRAM

Processor + GPU

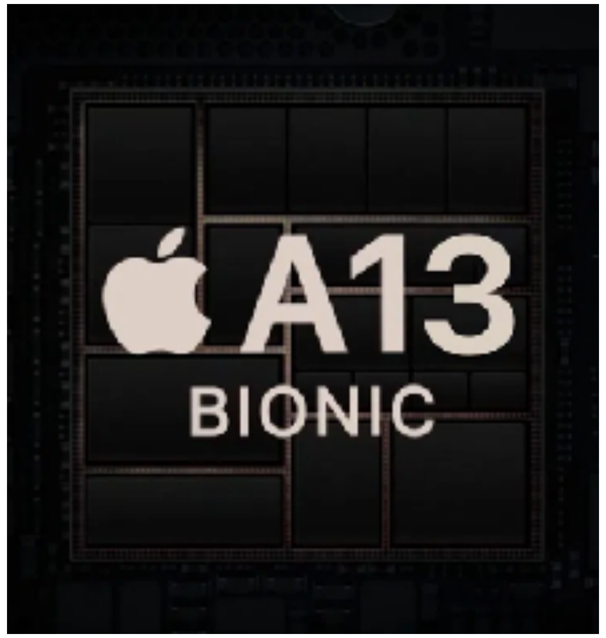
Network Modules/ Codecs

Peripherals (e.g., memory cards.)

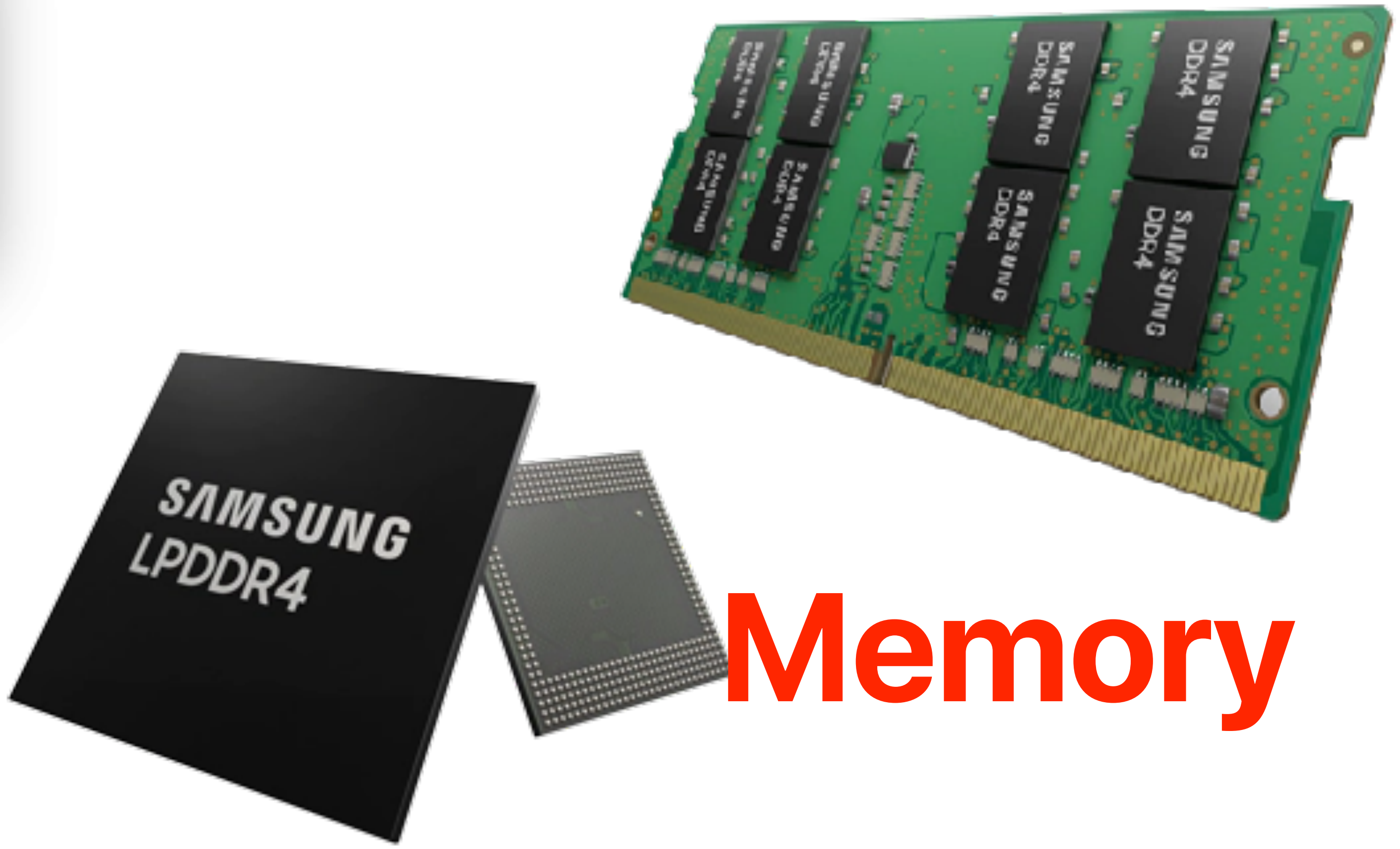
Tesla Model 3



Processors and memory modules are everywhere!



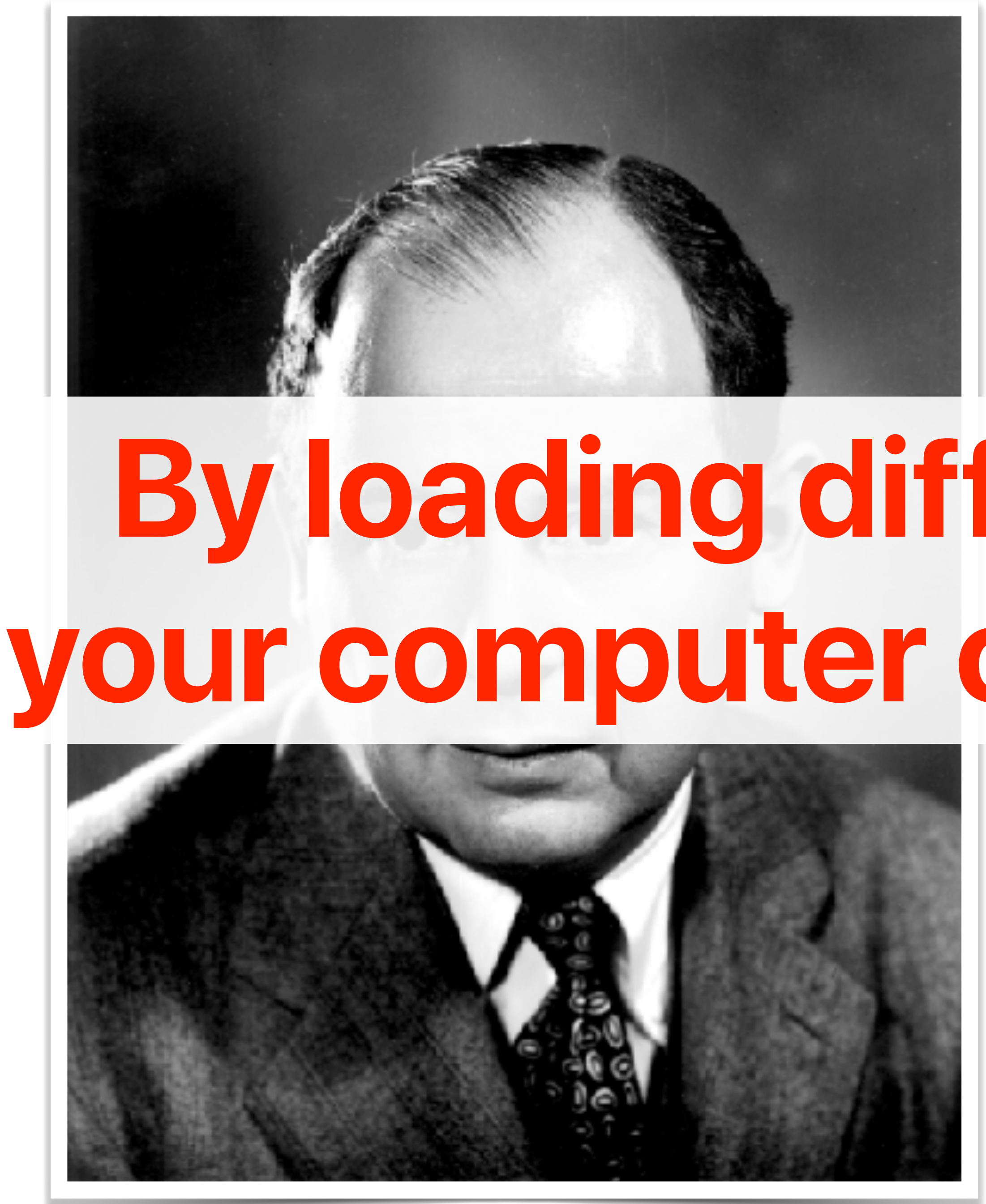
Processors

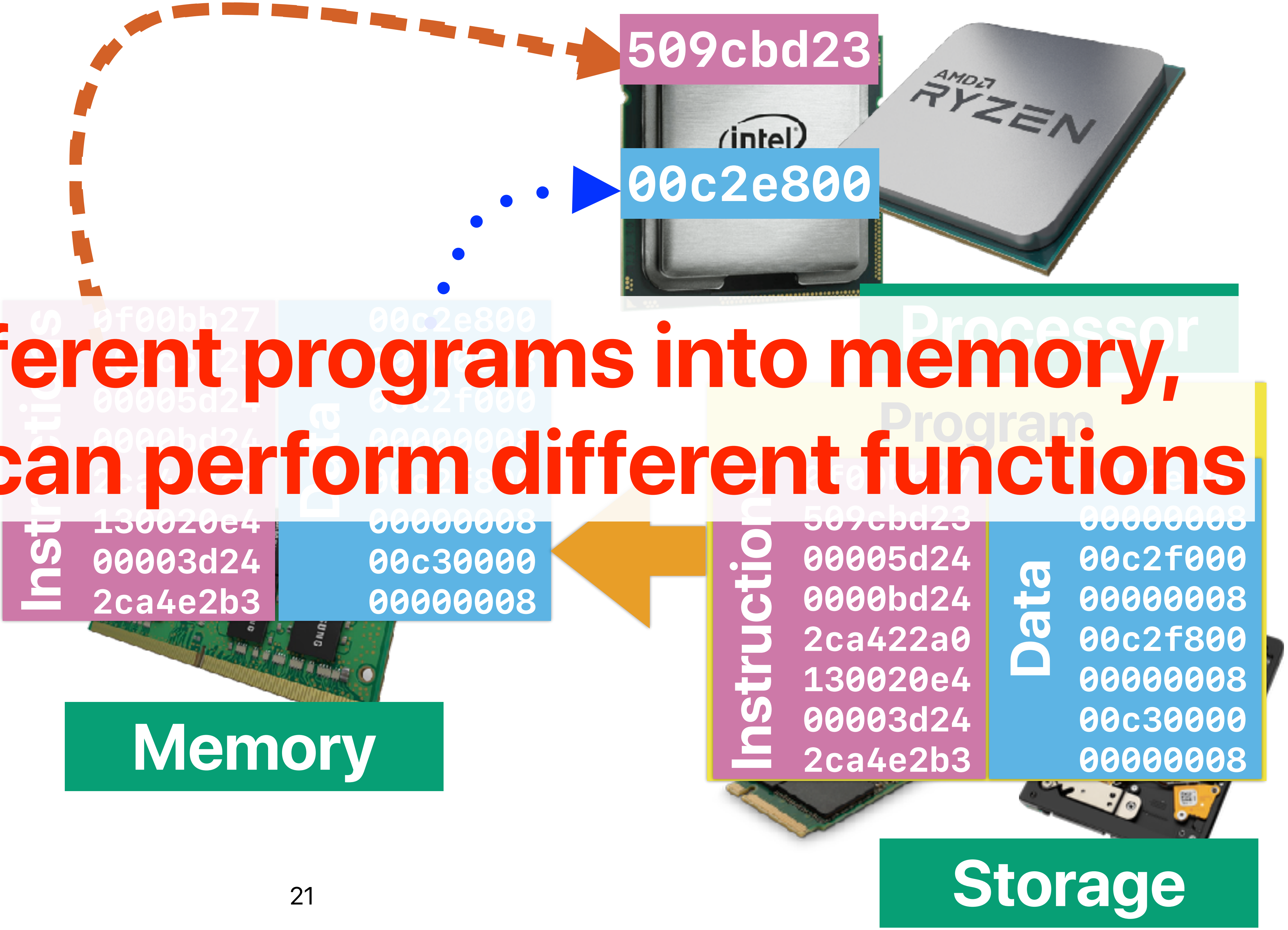


Memory

Why are “Processor” & “Memory” everywhere?

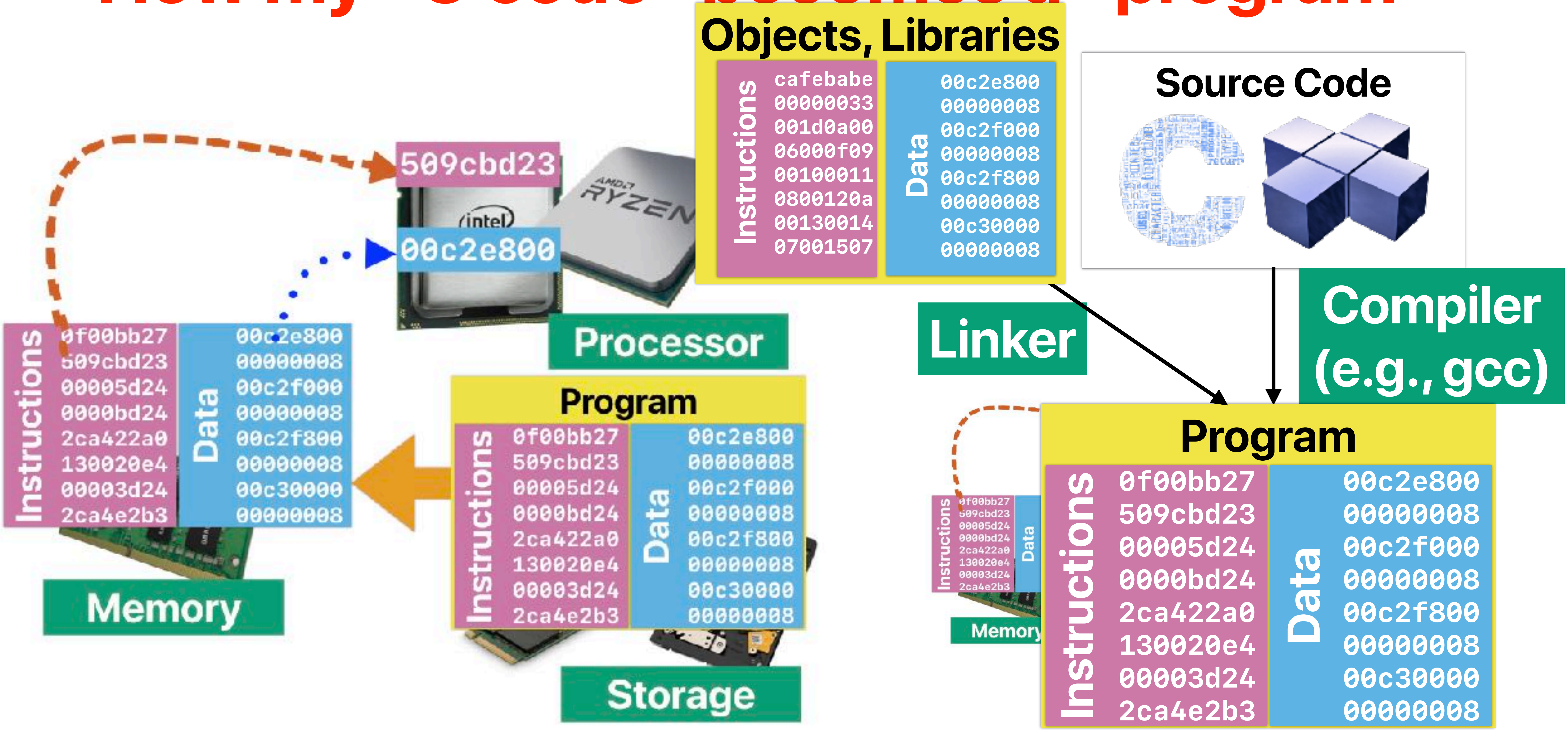
von Neumman Architecture





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

How my “C code” becomes a “program”



Objects, Libraries

Instructions	Data
cafebabe	00c2e800
00000033	00000008
001d0a00	00c2f000
06000f09	00000008
00100011	00c2f800
0800120a	00000008
00130014	00c30000
07001507	00000008

Source Code

Compiler (e.g., gcc)

Linker

Program

Instructions	Data
0f00bb27	00c2e800
509cbd23	00000008
00005d24	00c2f000
0000bd24	00000008
2ca422a0	00c2f800
130020e4	00000008
00003d24	00c30000
2ca4e2b3	00000008

Storage

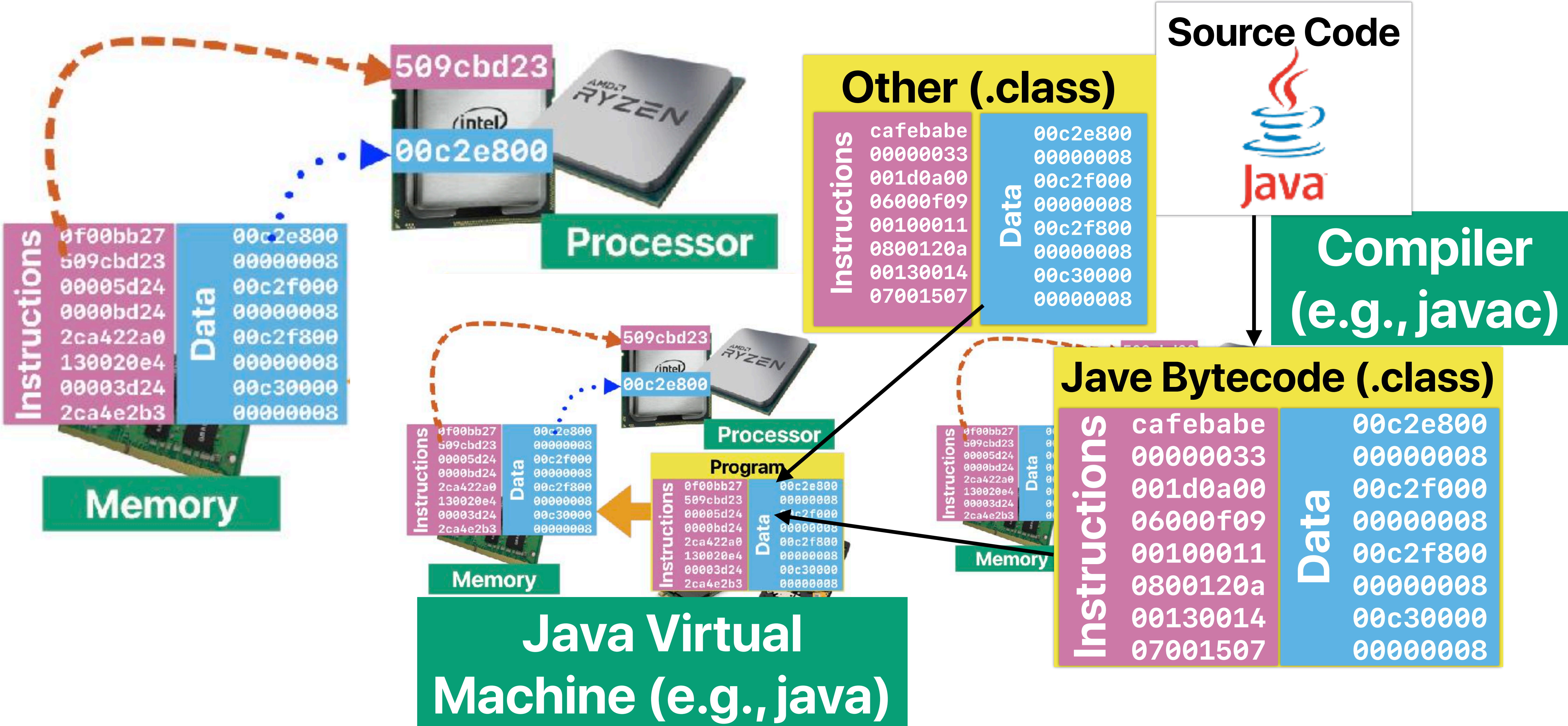
Memory

Processor

509cbd23

00c2e800

How my “Java code” becomes a “program”



Source Code

Compiler (e.g., javac)

Other (.class)

Instructions
```
cafebabe
00000033
001d0a00
06000f09
00100011
0800120a
00130014
07001507
```

Data
```
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008
```

Jave Bytecode (.class)

Instructions
```
cafebabe
00000033
001d0a00
06000f09
00100011
0800120a
00130014
07001507
```

Data
```
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008
```

Java Virtual Machine (e.g., java)

Program

Processor

Memory

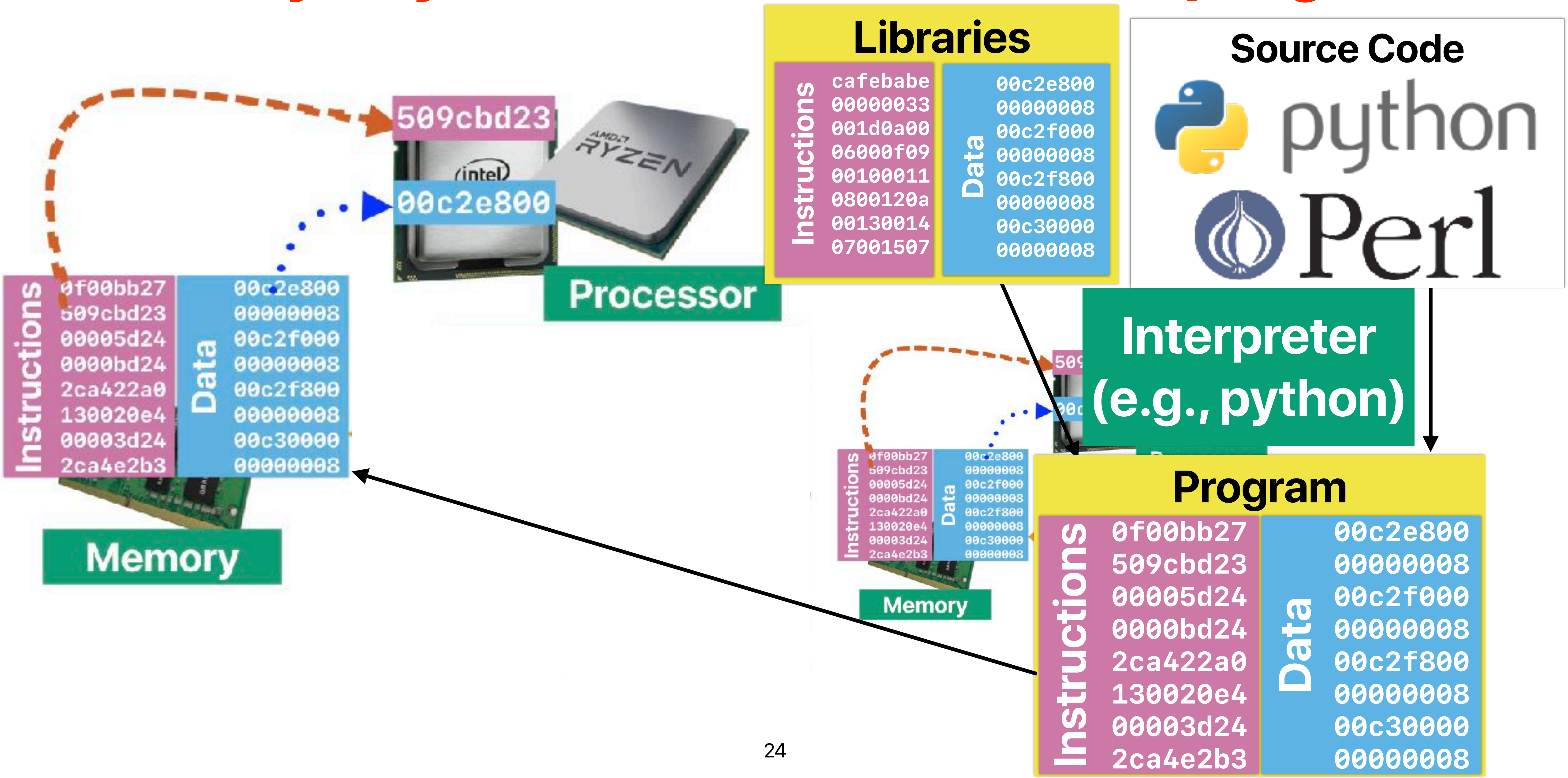
Instructions
```
0f00bb27
509cbd23
00005d24
0000bd24
2ca422a0
130020e4
00003d24
2ca4e2b3
```

Data
```
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008
```

509cbd23

00c2e800

How my “Python code” becomes a “program”



Libraries

Instructions
cafebabe
00000033
001d0a00
06000f09
00100011
0800120a
00130014
07001507

Data
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008

Source Code

python

Perl

Interpreter (e.g., python)

Program

Instructions
0f00bb27
509cbd23
00005d24
0000bd24
2ca422a0
130020e4
00003d24
2ca4e2b3

Data
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008

Processor

509cbd23

00c2e800

Memory

Instructions
0f00bb27
509cbd23
00005d24
0000bd24
2ca422a0
130020e4
00003d24
2ca4e2b3

Data
00c2e800
00000008
00c2f000
00000008
00c2f800
00000008
00c30000
00000008

Challenges of von Neumann Architecture

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
i
to miniaturize electronics equipment
creasingly complex electronic functi
space with minimum weight. Sever
evolved, including microassembly
individual components, thin-film s
semiconductor integrated circuits.

ICs are increasingly p

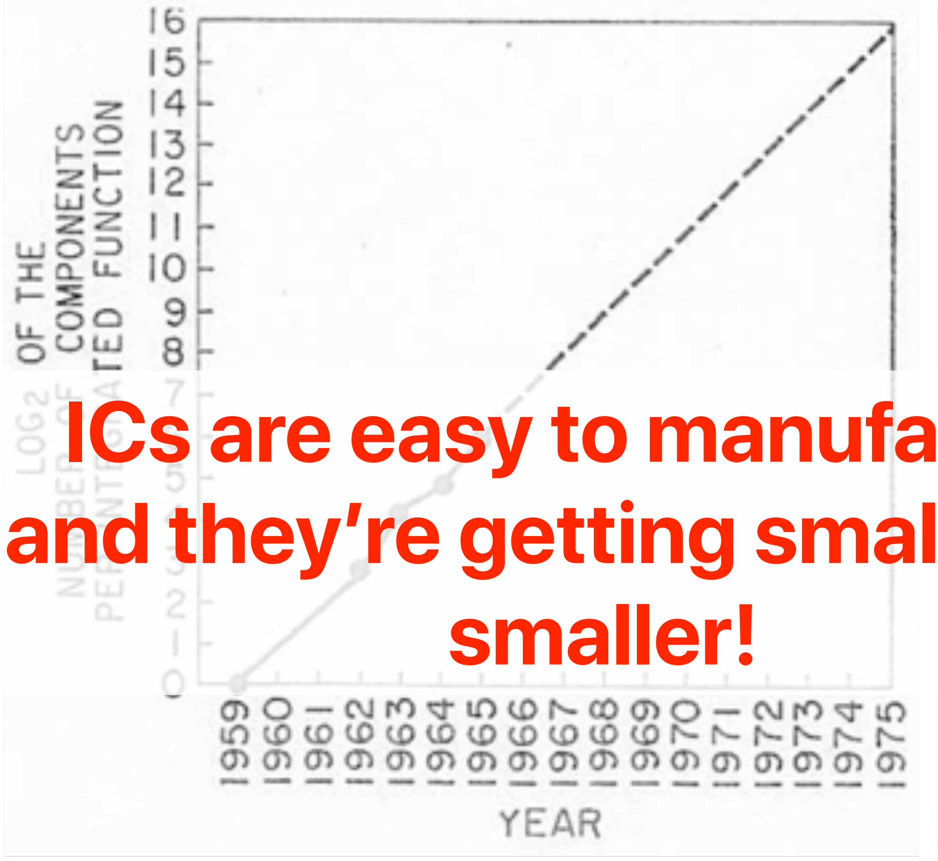
The establishment

Integrated electronics is established
for new
and
vable
llo, fo
he reli
that
ilure as the

stablish

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-per formance transistors can be built on centers two thousandths of an inch apart. Such a two-mil squar 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 circuits. The lack of large-value capacitors and
to integrated electronics in the linear area.

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
impo
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.

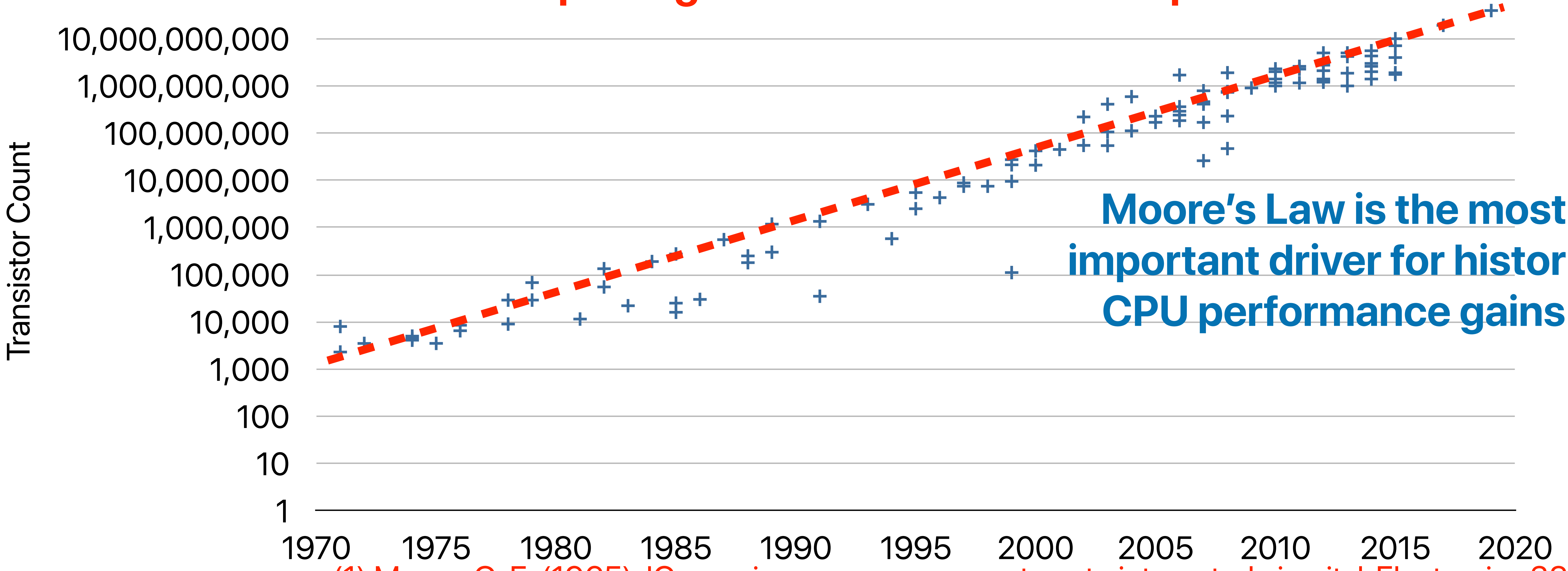
Designing ICs can be easy

[1] Mo mponents 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.

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

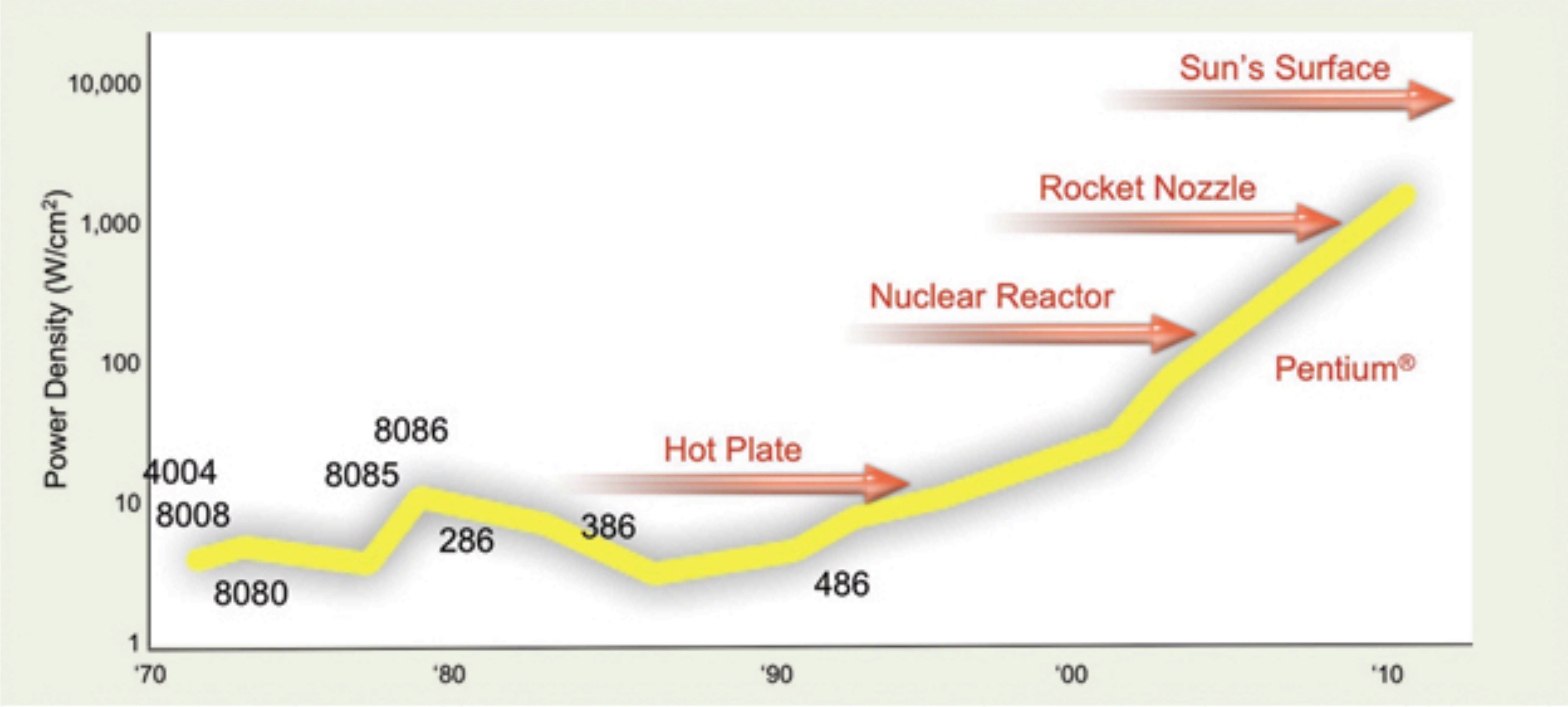
(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8) .

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

Moore’s Law still alive, but not that useful. Because …

CPU Architecture Today

Heat becoming un unmanageable problem



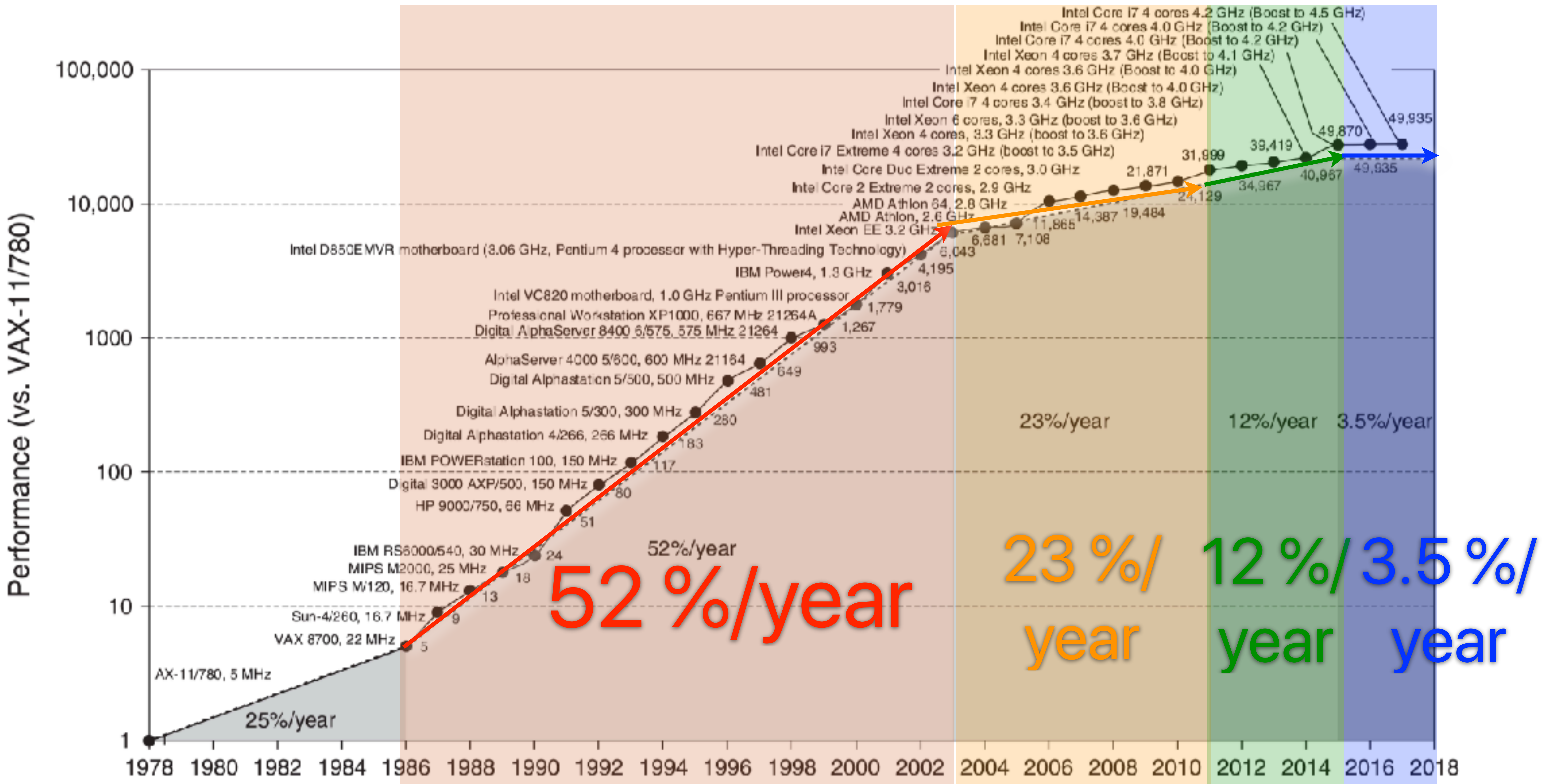
https://www.cadalyst.com/files/cadalyst/nodes/2008/6351/i1.jpg

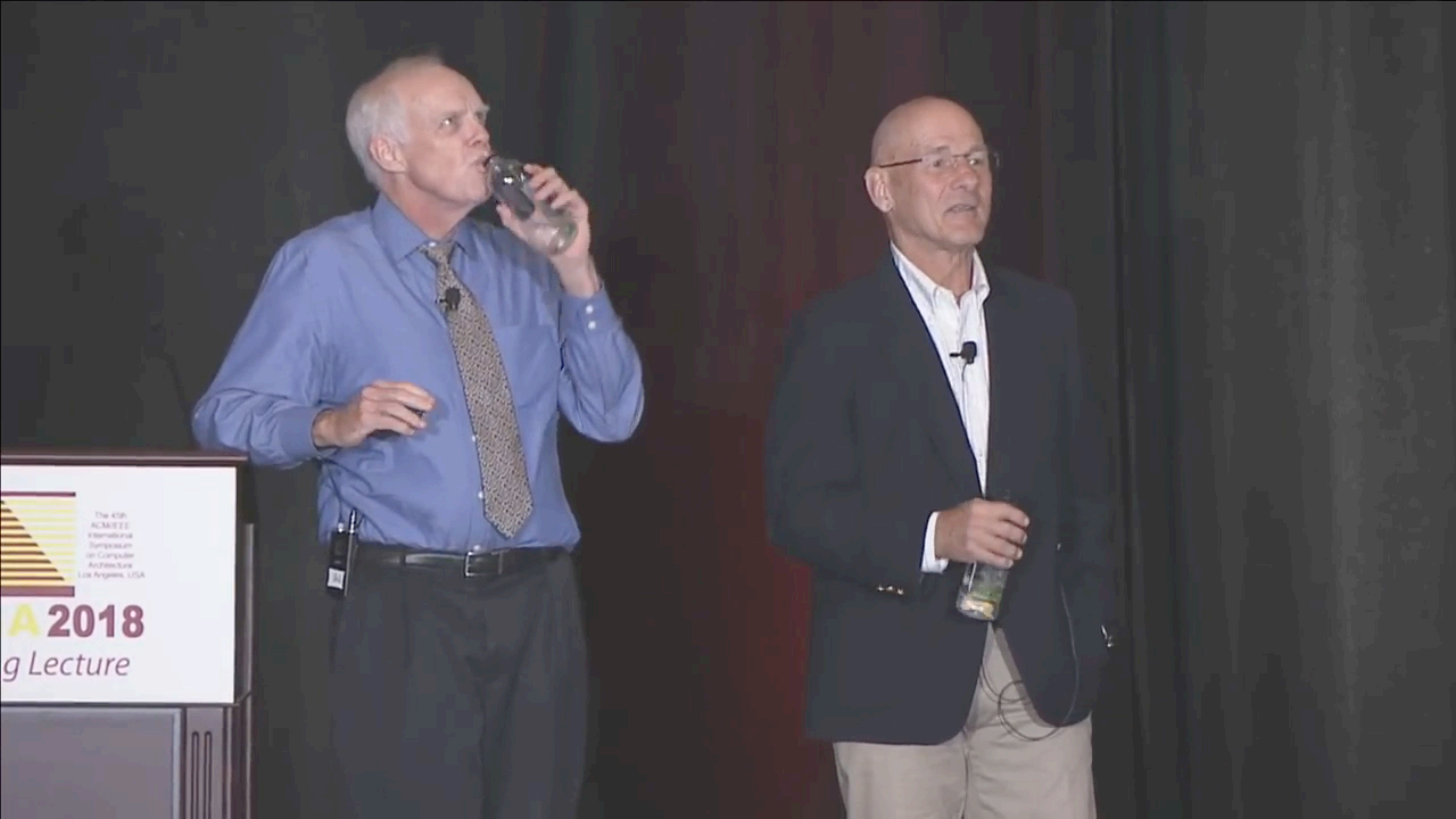
Figure 1. In CPU architecture today, heat is becoming an unmanageable problem.

Dynamic/Active Power

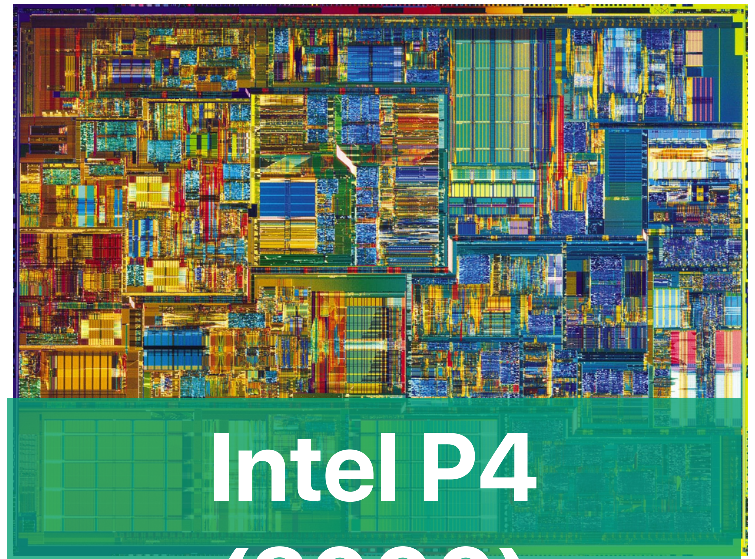
- The power consumption due to the switching of transistor states
- Dynamic power per transistor
  $P_{dynamic} \sim \alpha \times C \times \boxed{V^2 \times f} \times N$
  - α: average switches per cycle
  - C: capacitance
  - V: voltage
  - f: frequency, usually linear with V
  - N: the number of transistors

CPU is important but...

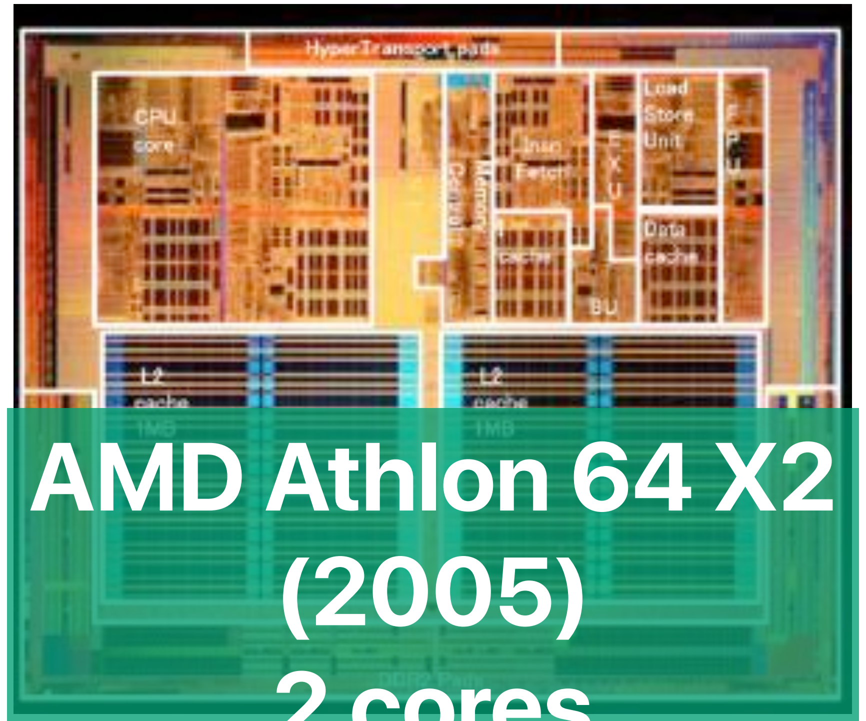



The 45th
ACM/IEEE
International
Symposium
on Computer
Architecture
Los Angeles, USA
A 2018
g Lecture

Multicore processors

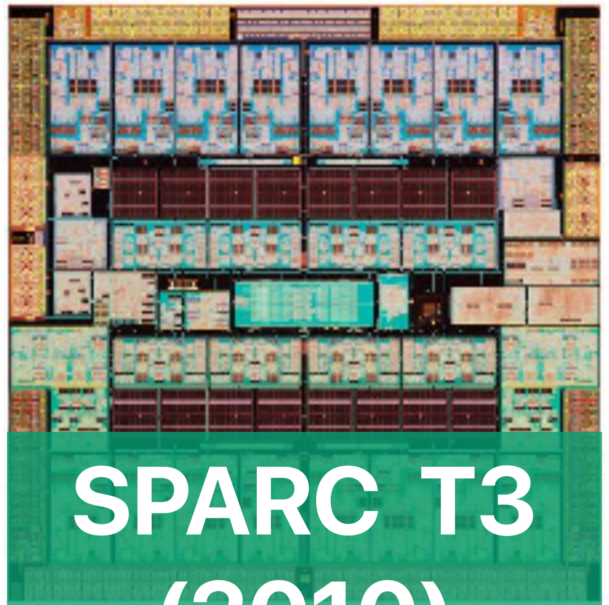


Intel P4
(2000)
1 core

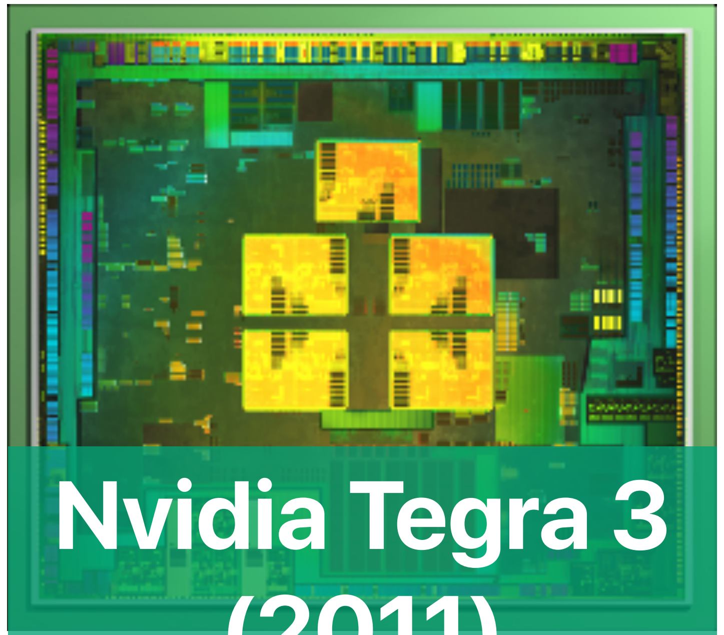


AMD Athlon 64 X2
(2005)
2 cores

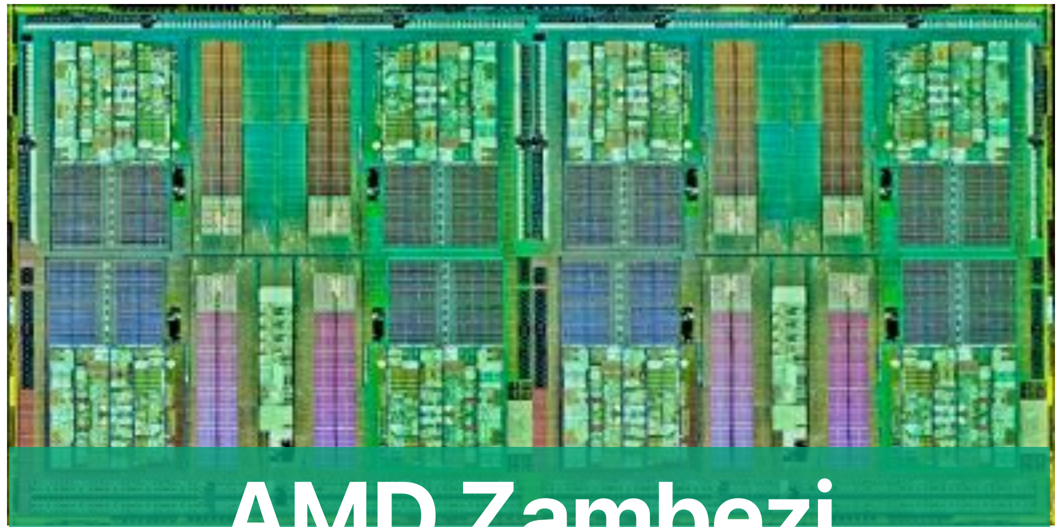
Intel Nahalem
(2010)
4 cores



SPARC T3
(2010)
16 cores



Nvidia Tegra 3
(2011)
5 cores



AMD Zambezi
(2011)
16 cores

Dennardian Broken

- Given a scaling factor S

Parameter	Relation	Classical Scaling	Leakage Limited
Power Budget		1	1
Chip Size		1	1
Vdd (Supply Voltage)		1/S	1
Vt (Threshold Voltage)	1/S	1/S	1
tex (oxide thickness)		1/S	1/S
W, L (transistor		1/S	1/S
Cgate (gate capacitance)	WL/tox	1/S	1/S
Isat (saturation current)	WVdd/tox	1/S	1
F (device frequency)	Isat/(CgateVdd)	S	S
D (Device/Area)	1/(WL)	S^2	S^2
p (device power)	IsatVdd	$1/S^2$	1
P (chip power)	Dp	1	S^2
U (utilization)	1/P	1	$1/S^2$

Static/Leakage Power

- The power consumption due to leakage — transistors do not turn all the way off during no operation
- Becomes the **dominant** factor in the most advanced process technologies.

$$P_{leakage} \sim N \times V \times e^{-V_t}$$

  - N: number of transistors
  - V: voltage
  - V_t: threshold voltage where transistor conducts (begins to switch)

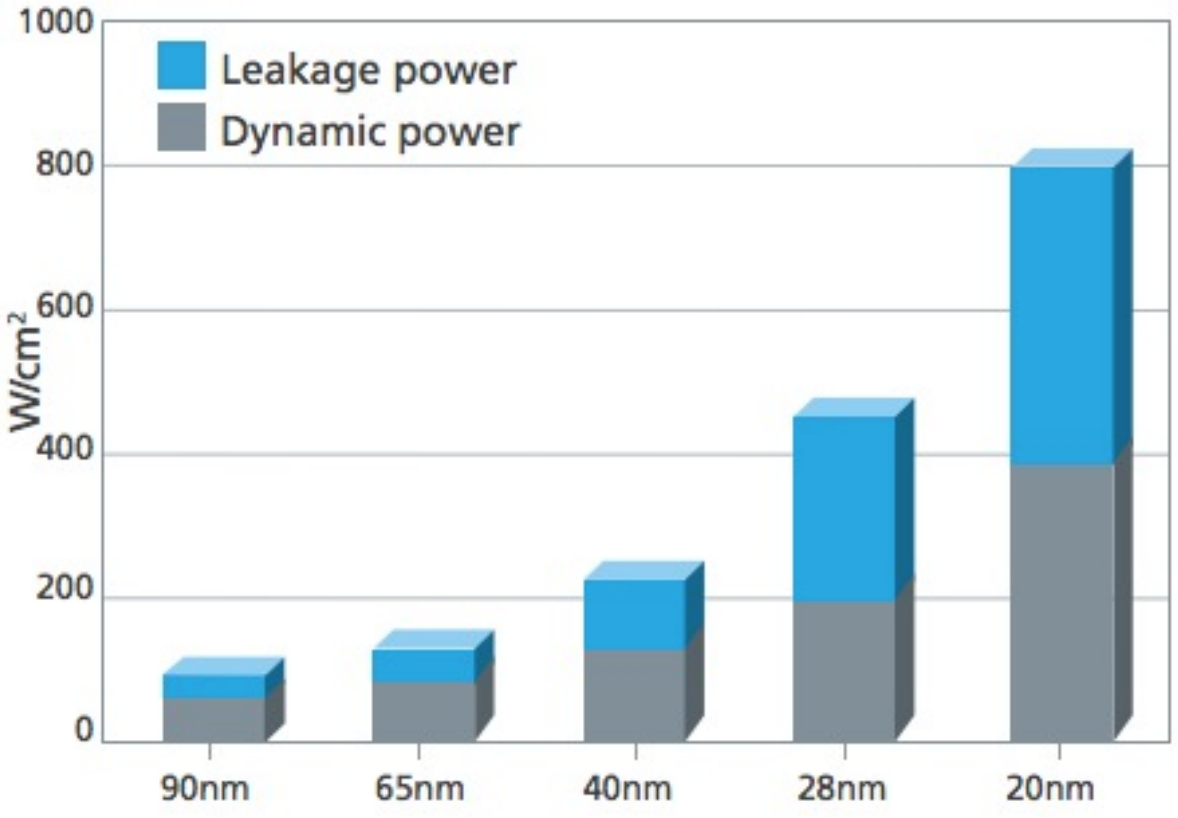


Figure 1: Leakage power becomes a growing problem as demands for more performance and functionality drive chipmakers to nanometer-scale process nodes (Source: IBS).

Power consumption to light on all transistors

Chip

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

=49W

Dennardian Scaling

Chip

0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

=50W

Dennardian Broken

Chip

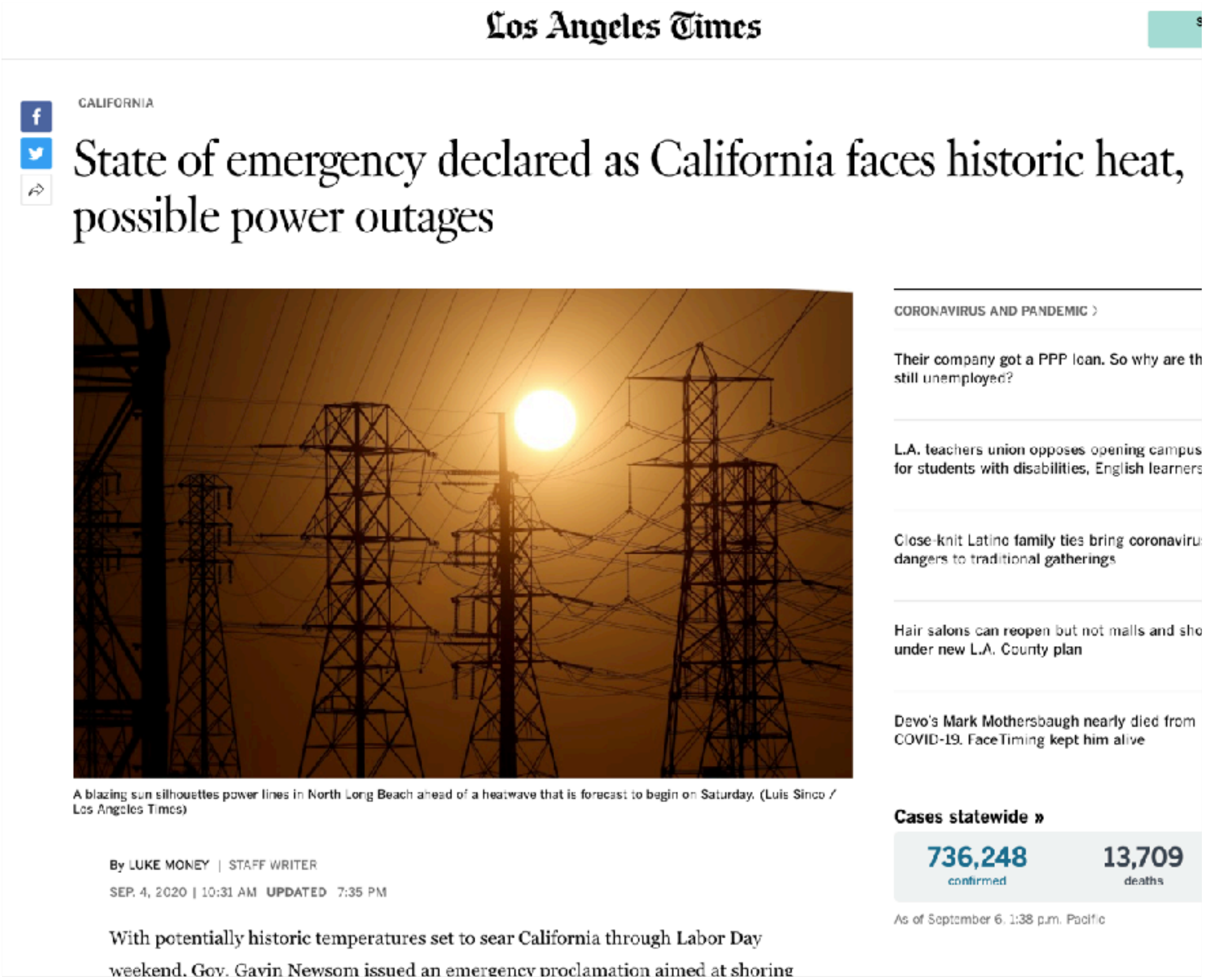
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

On ~ 50W

Off ~ 0W

Dark!

=100W!



Los Angeles Times

CALIFORNIA

State of emergency declared as California faces historic heat, possible power outages

A blazing sun silhouettes power lines in North Long Beach ahead of a heatwave that is forecast to begin on Saturday. (Luis Sinco / Los Angeles Times)

By LUKE MONEY | STAFF WRITER

SEP. 4, 2020 | 10:31 AM UPDATED 7:35 PM

With potentially historic temperatures set to sear California through Labor Day weekend, Gov. Gavin Newsom issued an emergency proclamation aimed at shoring

CORONAVIRUS AND PANDEMIC ›

Their company got a PPP loan. So why are th still unemployed?

L.A. teachers union opposes opening campus for students with disabilities, English learners

Close-knit Latino family ties bring coronaviru dangers to traditional gatherings

Hair salons can reopen but not malls and sho under new L.A. County plan

Devo's Mark Mothersbaugh nearly died from COVID-19. FaceTiming kept him alive

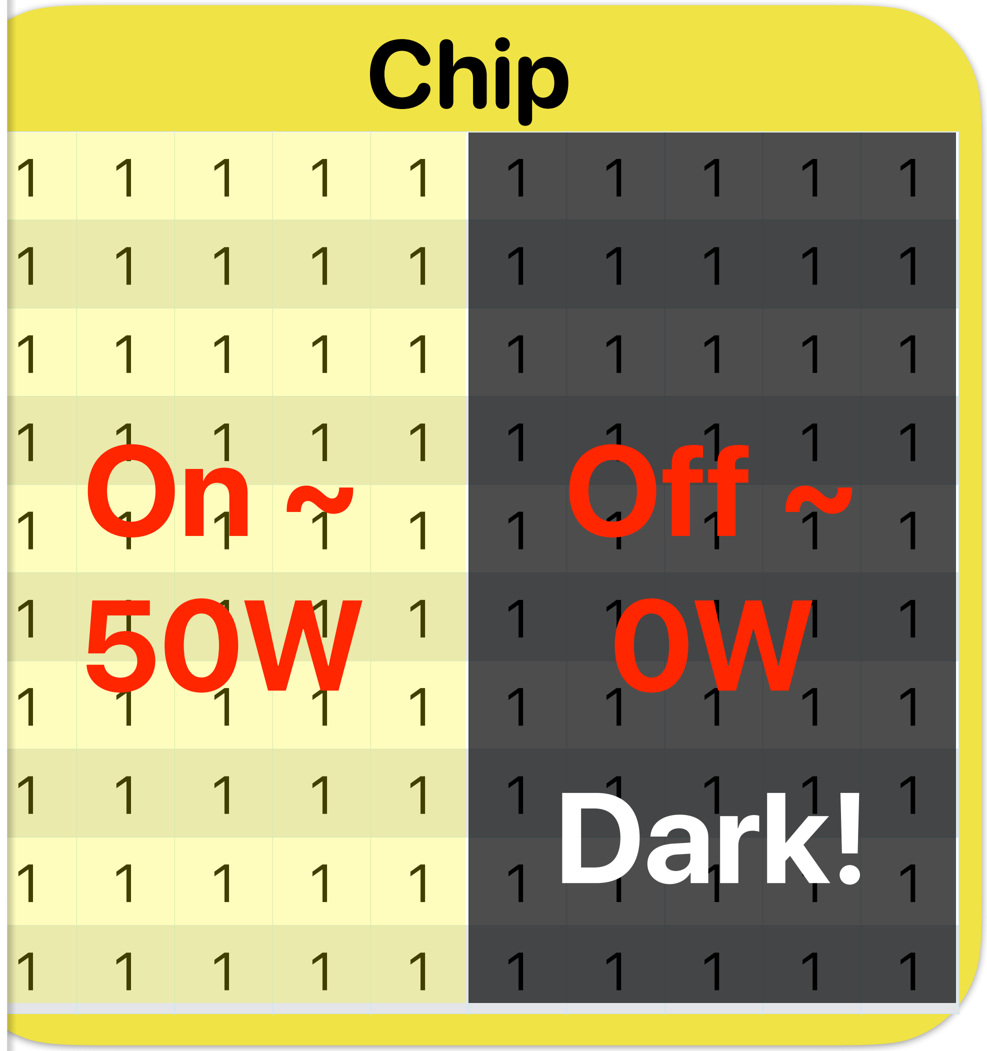
Cases statewide »

736,248 confirmed

13,709 deaths

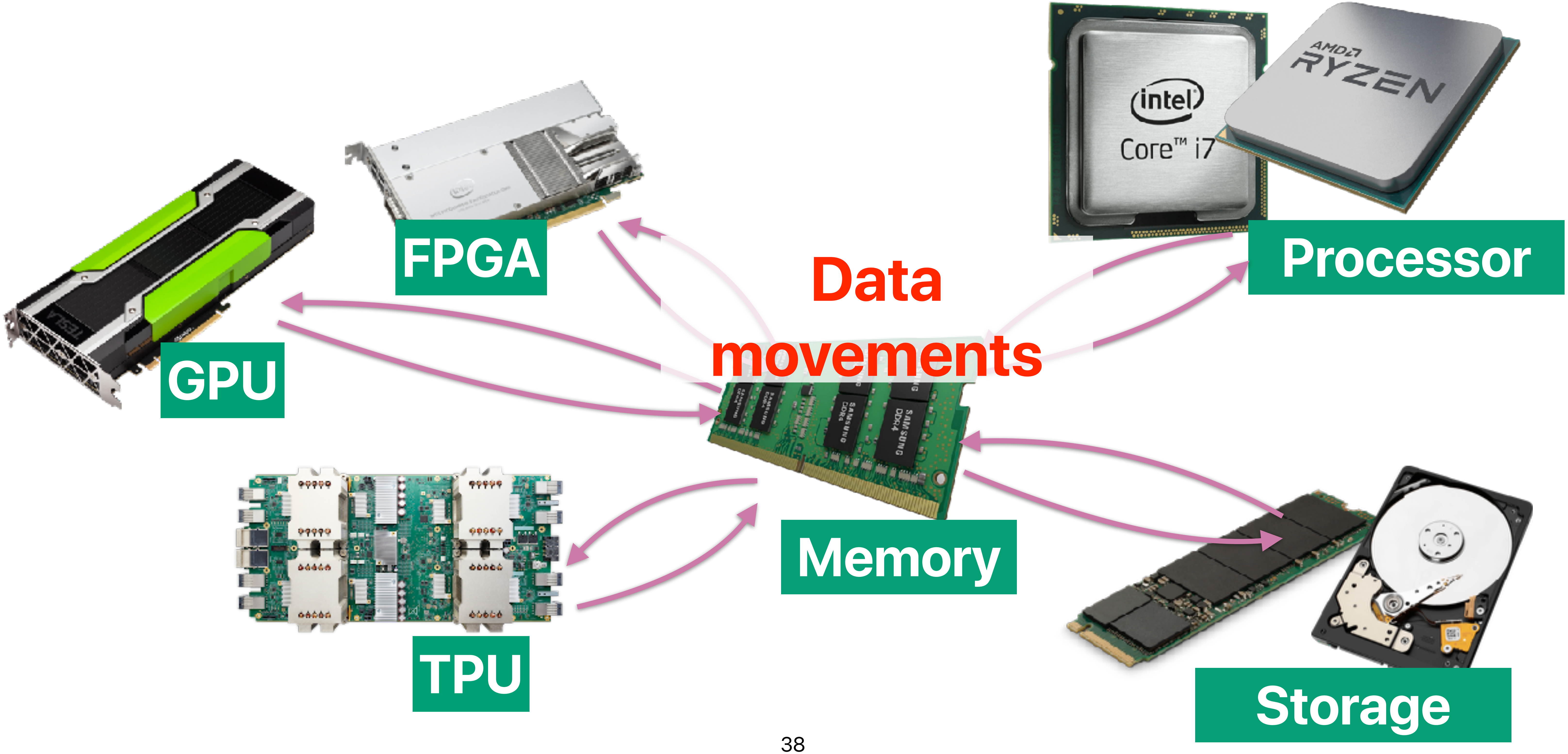
As of September 6, 1:38 p.m. Pacific

Dennardian Broken

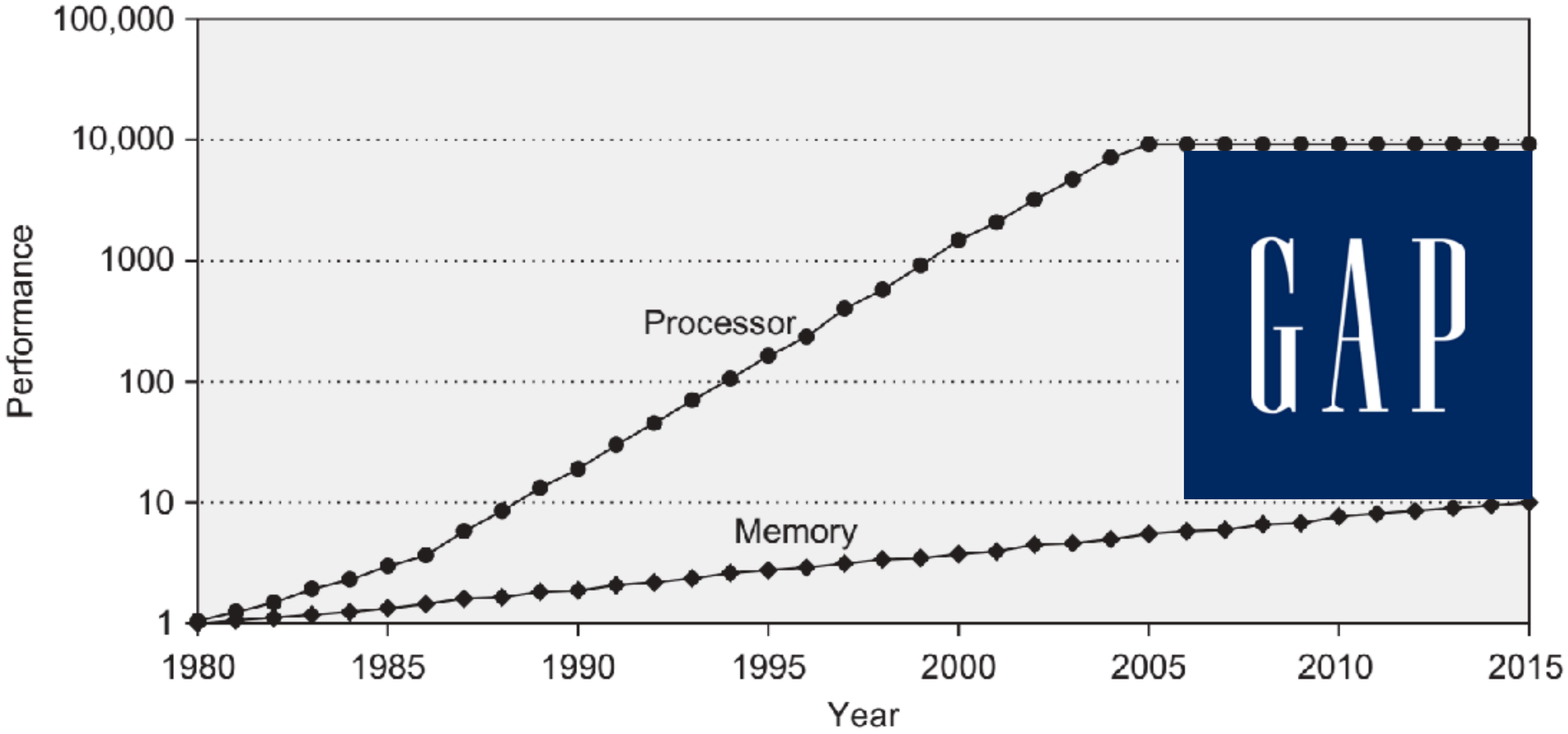


1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

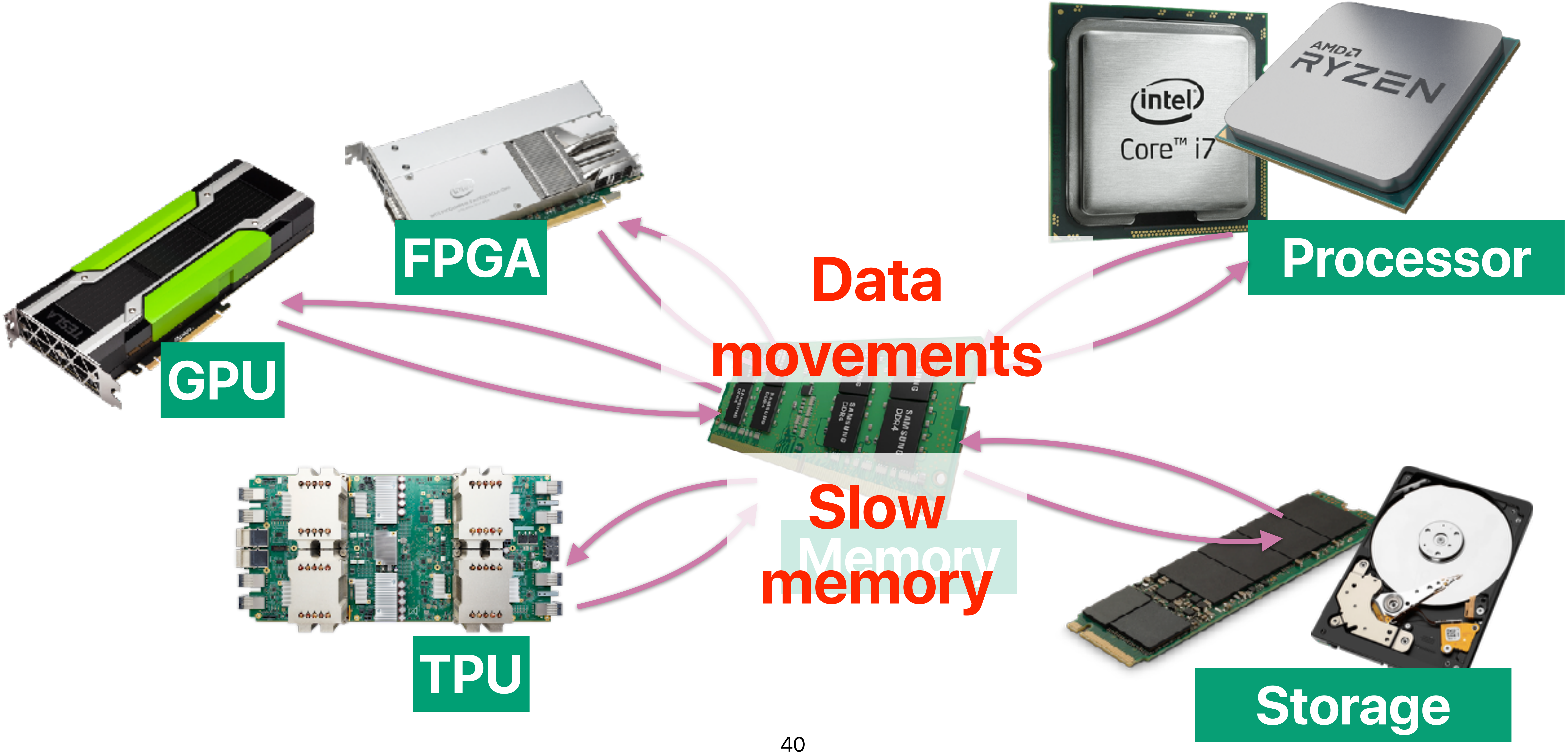
Heterogeneous Computer Architecture



Performance gap between Processor/Memory



Heterogeneous Computer Architecture



FPGA

Processor

GPU

Data movements

Memory

Slow memory

TPU

Storage

MAKE ARCHITECTURE
GREAT AGAIN
#MAGA
BIDEN
Unite for a Better Architecture
APPROVED BY JOE BIDEN. PAID FOR BY BIDEN FOR PRESIDENT.

Why should I care about “Computer Architecture”

What do you care when you're writing a program?

Algorithms

Data Structures

Software Engineering

Computer Architecture

Programming Languages

User Interfaces

Demo (1)

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

for (unsigned c = 0; c < arraySize*1000; ++c) {
        if (data[c%arraySize] >= INT_MAX/2)
            sum ++;
    }
}
```

$O(nlog_2n)$

$O(n)$

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

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

Demo (2) — merge sort v.s. bitonic sort

Merge Sort

$O(nlog_2n)$

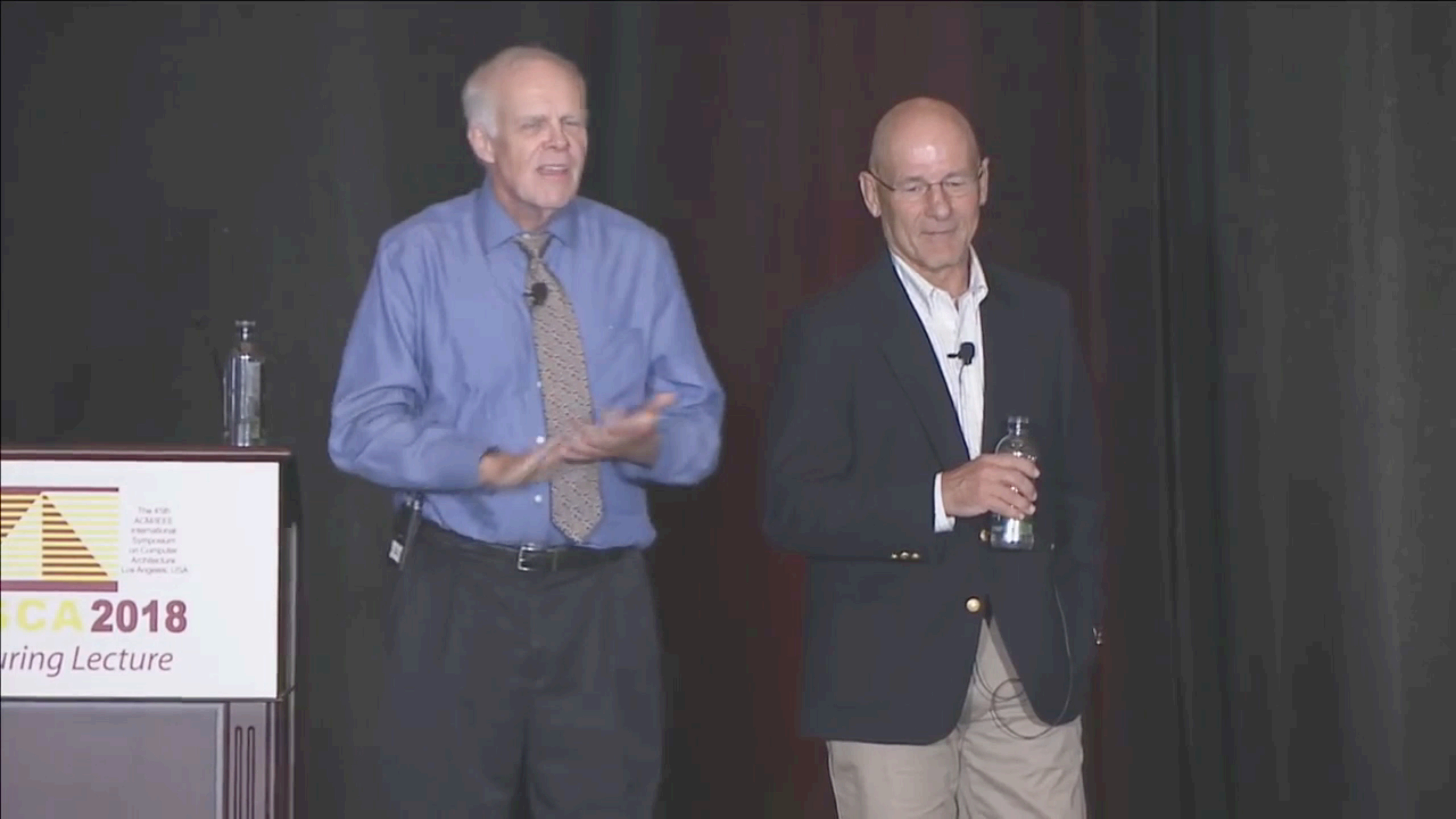
Bitonic Sort

$O(nlog_2^2n)$

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


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

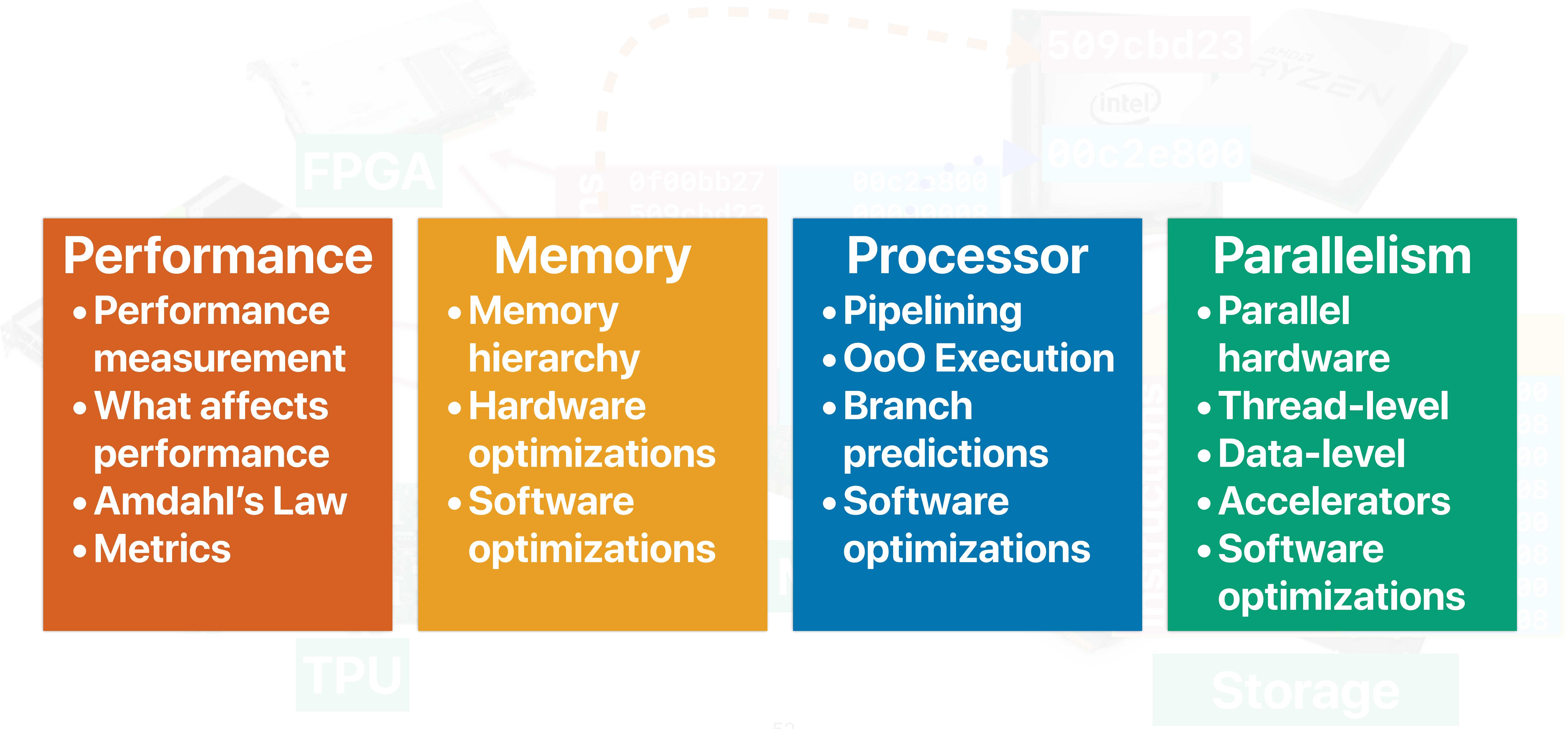
? ? ? ?

Thinking about the washlet

Or a Tesla

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

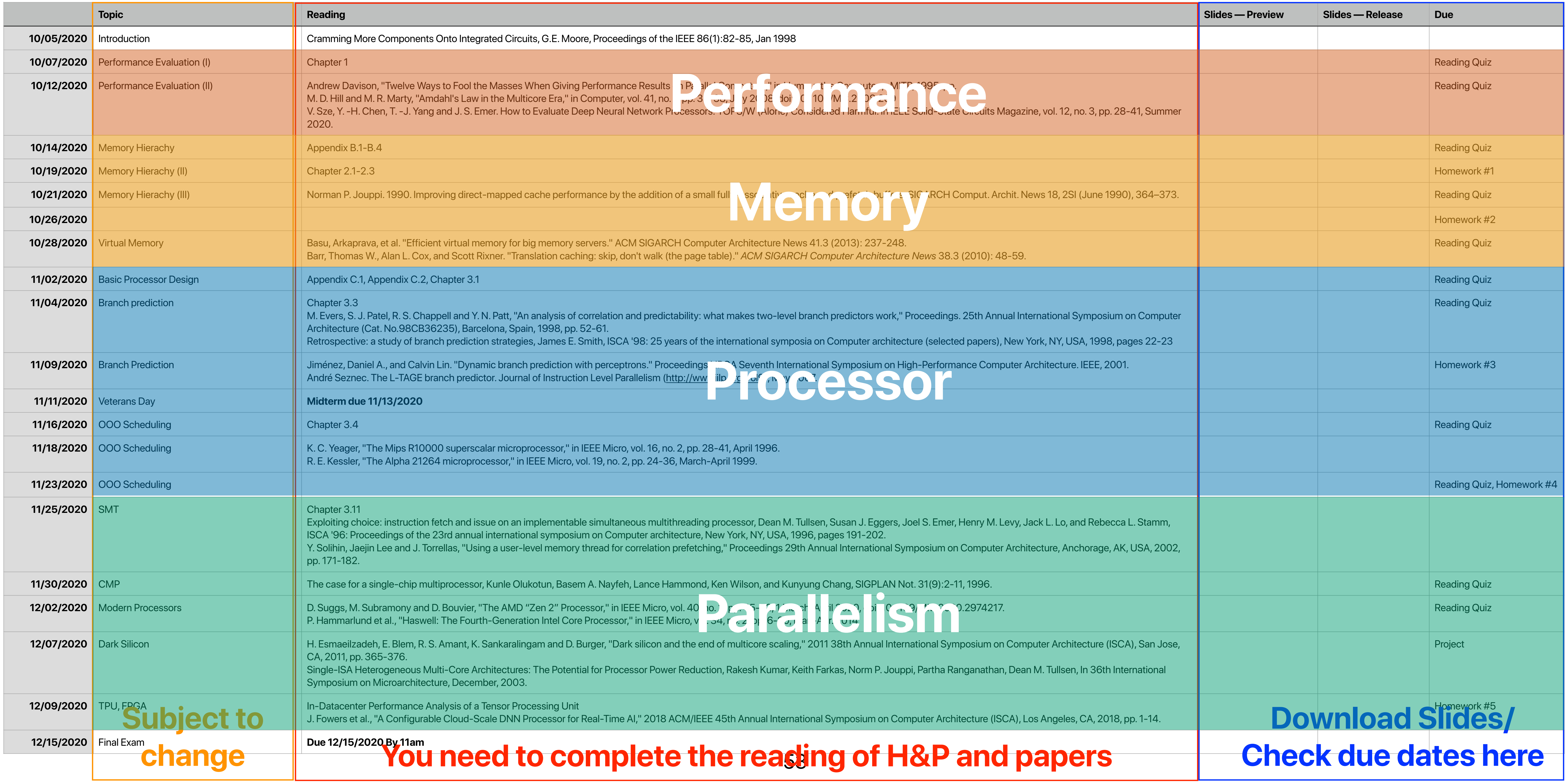
Processor

- Pipelining
- OoO Execution
- Branch predictions
- Software optimizations

Parallelism

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

Tentative Schedule (on Website)



	Topic	Reading	Slides — Preview	Slides — Release	Due
10/05/2020	Introduction	Cramming More Components Onto Integrated Circuits, G.E. Moore, Proceedings of the IEEE 86(1):82-85, Jan 1998			
10/07/2020	Performance Evaluation (I)	Chapter 1			Reading Quiz
10/12/2020	Performance Evaluation (II)	Andrew Davison, "Twelve Ways to Fool the Masses When Giving Performance Results [illegible] M. D. Hill and M. R. Marty, "Amdahl's Law in the Multicore Era," in Computer, vol. 41, no. [illegible] V. Sze, Y. -H. Chen, T. -J. Yang and J. S. Emer. How to Evaluate Deep Neural Network Pr[illegible] Solid-State Circuits Magazine, vol. 12, no. 3, pp. 28-41, Summer 2020.			Reading Quiz
10/14/2020	Memory Hierachy	Appendix B.1-B.4			Reading Quiz
10/19/2020	Memory Hierachy (II)	Chapter 2.1-2.3			Homework #1
10/21/2020	Memory Hierachy (III)	Norman P. Jouppi. 1990. Improving direct-mapped cache performance by the addition of a small full[illegible] SIGARCH Comput. Archit. News 18, 2SI (June 1990), 364–373.			Reading Quiz
10/26/2020					Homework #2
10/28/2020	Virtual Memory	Basu, Arkaprava, et al. "Efficient virtual memory for big memory servers." ACM SIGARCH Computer Architecture News 41.3 (2013): 237-248. Barr, Thomas W., Alan L. Cox, and Scott Rixner. "Translation caching: skip, don't walk (the page table)." *ACM SIGARCH Computer Architecture News* 38.3 (2010): 48-59.			Reading Quiz
11/02/2020	Basic Processor Design	Appendix C.1, Appendix C.2, Chapter 3.1			Reading Quiz
11/04/2020	Branch prediction	Chapter 3.3 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, Spain, 1998, pp. 52-61. Retrospective: a study of branch prediction strategies, James E. Smith, ISCA '98: 25 years of the international symposia on Computer architecture (selected papers), New York, NY, USA, 1998, pages 22-23			Reading Quiz
11/09/2020	Branch Prediction	Jiménez, Daniel A., and Calvin Lin. "Dynamic branch prediction with perceptrons." Proceedings [illegible]A Seventh International Symposium on High-Performance Computer Architecture. IEEE, 2001. André Seznec. The L-TAGE branch predictor. Journal of Instruction Level Parallelism (http://ww[illegible]			Homework #3
11/11/2020	Veterans Day	**Midterm due 11/13/2020**			
11/16/2020	OOO Scheduling	Chapter 3.4			Reading Quiz
11/18/2020	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.			
11/23/2020	OOO Scheduling				Reading Quiz, Homework #4
11/25/2020	SMT	Chapter 3.11 Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor, Dean M. Tullsen, Susan J. Eggers, Joel S. Emer, Henry M. Levy, Jack L. Lo, and Rebecca L. Stamm, ISCA '96: Proceedings of the 23rd annual international symposium on Computer architecture, New York, NY, USA, 1996, pages 191-202. Y. Solihin, Jaejin Lee and J. Torrellas, "Using a user-level memory thread for correlation prefetching," Proceedings 29th Annual International Symposium on Computer Architecture, Anchorage, AK, USA, 2002, pp. 171-182.			
11/30/2020	CMP	The case for a single-chip multiprocessor, Kunle Olukotun, Basem A. Nayfeh, Lance Hammond, Ken Wilson, and Kunyung Chang, SIGPLAN Not. 31(9):2-11, 1996.			Reading Quiz
12/02/2020	Modern Processors	D. Suggs, M. Subramony and D. Bouvier, "The AMD "Zen 2" Processor," in IEEE Micro, vol. 40[illegible].2974217. P. Hammarlund et al., "Haswell: The Fourth-Generation Intel Core Processor," in IEEE Micro, v[illegible]			Reading Quiz
12/07/2020	Dark Silicon	H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam and D. Burger, "Dark silicon and the end of multicore scaling," 2011 38th Annual International Symposium on Computer Architecture (ISCA), San Jose, CA, 2011, pp. 365-376. Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction, Rakesh Kumar, Keith Farkas, Norm P. Jouppi, Partha Ranganathan, Dean M. Tullsen, In 36th International Symposium on Microarchitecture, December, 2003.			Project
12/09/2020	TPU, FPGA	In-Datacenter Performance Analysis of a Tensor Processing Unit J. Fowers et al., "A Configurable Cloud-Scale DNN Processor for Real-Time AI," 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), Los Angeles, CA, 2018, pp. 1-14.			Homework #5
12/15/2020	Final Exam	**Due 12/15/2020 By 11am**			

Performance

Memory

Processor

Parallelism

Subject to change

You need to complete the reading of H&P and papers

Download Slides/
Check due dates here

Learning eXperience

Most lectures today ...

I expect the lecture to be...

You

Me

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 polls in Zoom to express your opinion
  - Group **discussion**
    - Breakout rooms based on your residential colleges!
    - Use polls in Zoom to express your group's opinion
  - Whole-classroom **discussion** — we would like to hear from you

Read **Think** **Discuss**

Your tasks

- Login/discussion in iLearn 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 iLearn (15%) — will drop the lowest
- Homework throughout the course. (15%) — will drop the lowest
  - Help to practice the concepts from each topic
  - Come to class — counted as an assignment
- Project (10%)
- Midterm (20%)
- Cumulative final (35%)

Background music: We're Not Gonna Take It/ Songwriter(s): Dee Snider/Performed by Twisted Sister

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 iLearn
  - 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 one or two lowest at least

Why attend live sessions and discuss?

- 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
  - It’s going to be fun!
- You will be GET CREDIT for your efforts to learn in class
  - By answering questions with polls within Zoom
  - Answer **50%** of the clicker questions in class, get full credits for a homework assignment
  - The best group — the group with the most correct answers after group discussions, will receive a USD 5 amazon gift card for each of its members

Why still assignments and term project?

- Human beings' memories are volatile and vague
- Assignments
  - Let you practice again the concepts learned from the lectures
  - The best way to prepare for midterm and final
  - Publish on the website, submit through iLearn
- Project
  - Let you get a feeling how you can apply the knowledge learned in class to "real-life" applications/program
  - C/C++ programming
  - Individual project
  - It's going to be a "contest" — the winner will have a prize

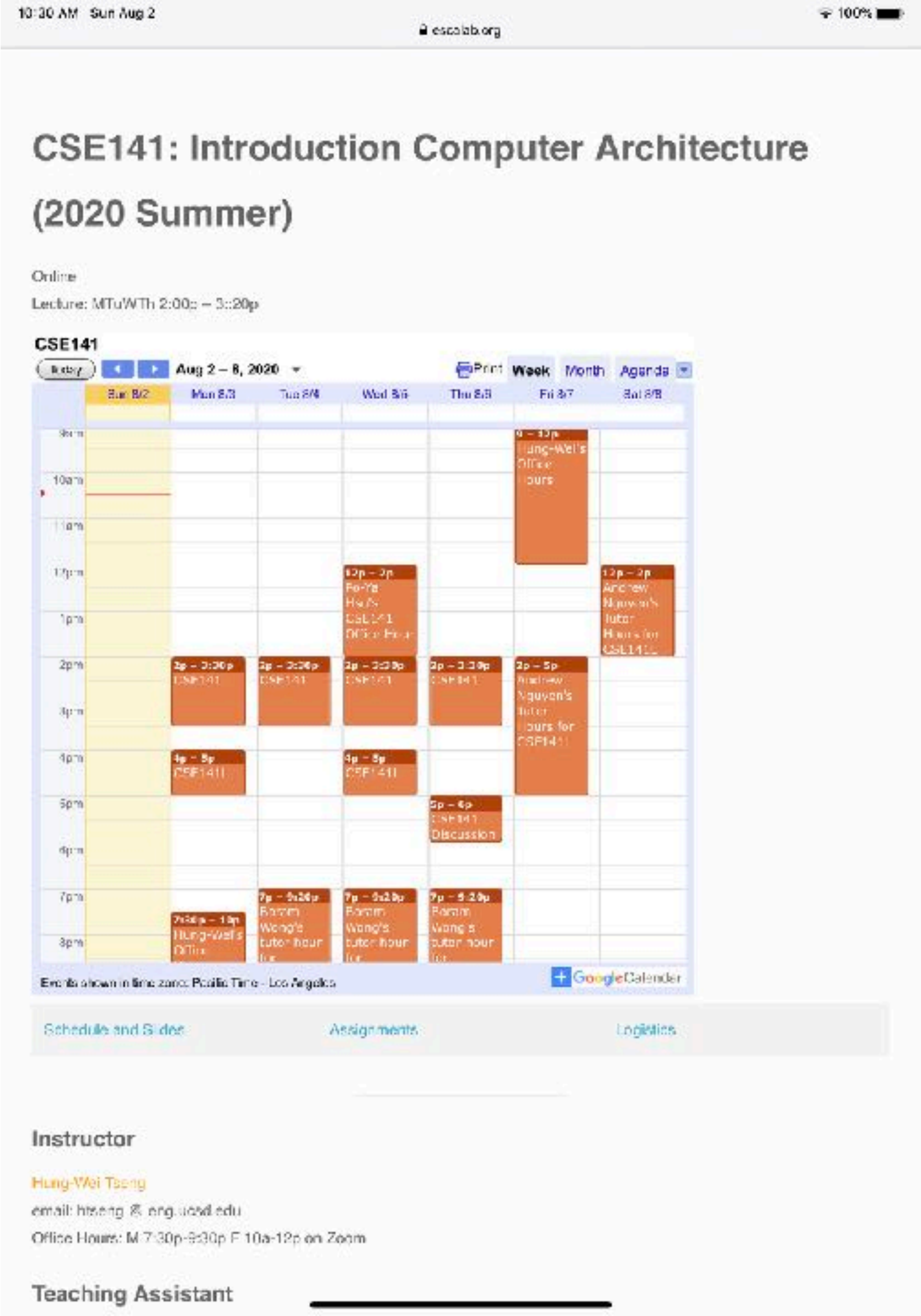
Logistics

Course resource

- Lectures:
  - Live on Zoom — please check your e-mail/iLearn for the link
  - Live on Youtube (you can only watch): https://www.youtube.com/profusagi
  - Repository on Youtube: https://www.youtube.com/profusagi
- Schedule, slides on **course webpage**: https://www.escalab.org/classes/cs203-2020fa/
- Discussion on **piazza**: https://piazza.com/class/kffrohnk4kw6vo
- Reading quizzes, homework submissions on **iLearn**: https://ilearn.ucr.edu

The website

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



10:30 AM Sun Aug 2

escalab.org

100%

CSE141: Introduction Computer Architecture (2020 Summer)

Online

Lecture: MTuWTh 2:00p – 3:20p

CSE141

	Sun 8/2	Mon 8/3	Tue 8/4	Wed 8/5	Thu 8/6	Fri 8/7	Sat 8/8
9am						9 – 12p Hung-Wei's Office Hours	
10am							
11am							
12pm				12p – 2p Po-Ya Hsu's CSE141 Office Hour			12p – 2p Andrew Nguyen's Tutor Hours for CSE141…
1pm							
2pm		2p – 3:20p CSE141	2p – 3:20p CSE141	2p – 3:20p CSE141	2p – 3:20p CSE141	2p – 5p Andrew Nguyen's Tutor Hours for CSE141L	
3pm							
4pm		4p – 5p CSE141L		4p – 5p CSE141L			
5pm					5p – 6p CSE141 Discussion		
6pm							
7pm		7:30p – 10p Hung-Wei's Office	7p – 9:20p Bonnim Wong's tutor hour for	7p – 9:20p Bonnim Wong's tutor hour for	7p – 9:20p Bonnim Wong's tutor hour for		
8pm							

Events shown in time zone: Pacific Time - Los Angeles

Schedule and Slides Assignments Logistics

Instructor

Hung-Wei Tseng

email: htseng @ eng.ucsd.edu

Office Hours: M 7:30p-9:30p F 10a-12p on Zoom

Teaching Assistant

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

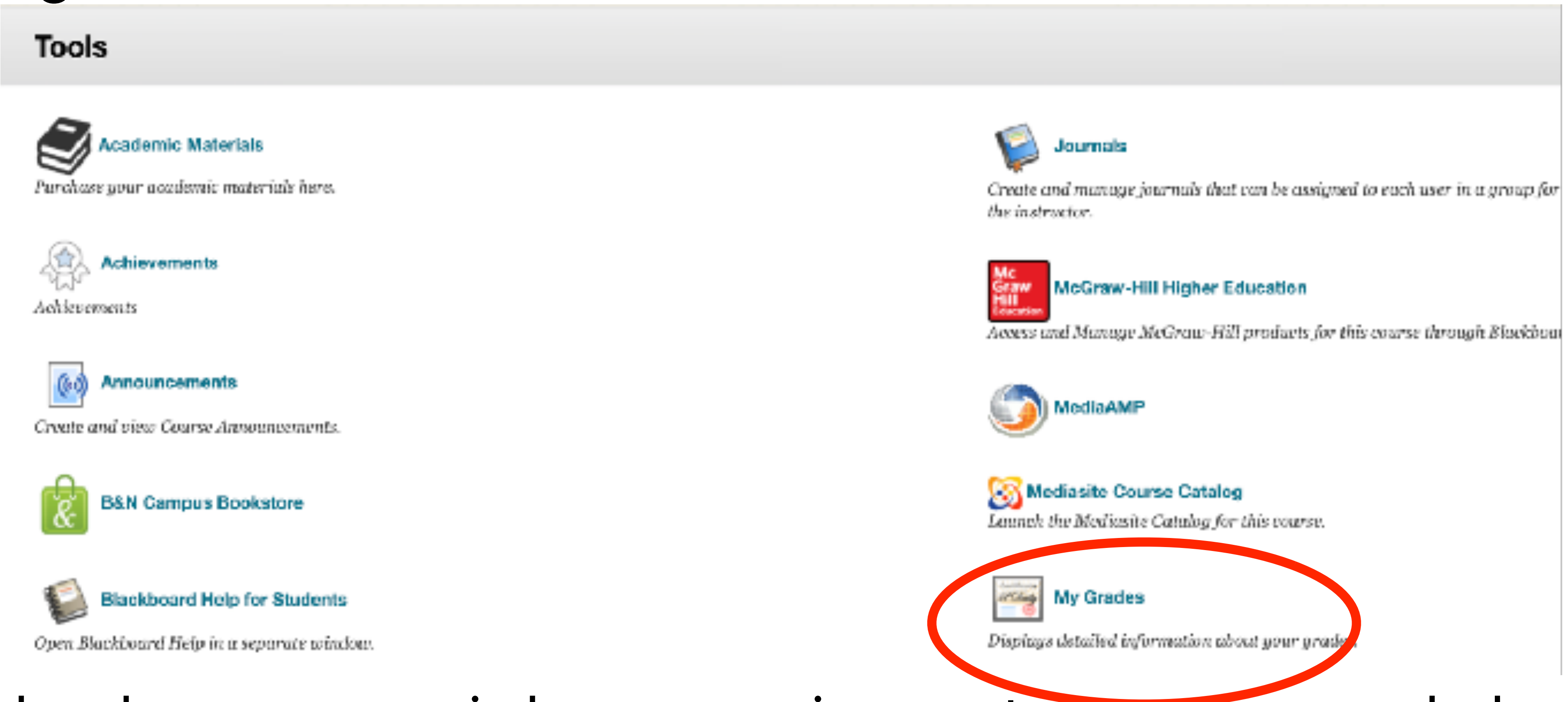
- Website:
  https://intra.engr.ucr.edu/~htseng/
- E-mail: htseng @ ucr.edu
- PhD in **Computer Science**,
  University of California, San Diego
- Research Interests
  - Intelligent storage devices
  - Non-volatile memory based systems
  - Near-data processing
  - Or anything could accelerate applications
- Office hour:
  M 8p-9p and W 2p-3p on Zoom

Teaching Assistant — Quan Fan

- Office hours: Fridays 1p-3p on Zoom
- E-mail: qfan005 @ ucr.edu

Grading

- You can see your grades on iLearn.



Tools

Academic Materials
Purchase your academic materials here.

Achievements
Achievements

Announcements
Create and view Course Announcements.

B&N Campus Bookstore

Blackboard Help for Students
Open Blackboard Help in a separate window.

Journals
Create and manage journals that can be assigned to each user in a group for the instructor.

McGraw-Hill Higher Education
Access and Manage McGraw-Hill products for this course through Blackbou

MediaAMP

Mediasite Course Catalog
Launch the Mediasite Catalog for this course.

My Grades
Displays detailed information about your grade

- 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.

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 2020.**

UC San Diego

NC STATE UNIVERSITY

UC RIVERSIDE

2012 Summer

2019 Summer I

2016 Fall

2019 Fall

2014 Summer

2019 Summer II

2017 Spri

2020 Winter

2017 Fall

UC zoom

2016 Spring

2018 Spring

You
2020 Fall

2016 Summer

2018 Fall

2019 Spring

2020 Summer

2020 Spring

Q & A

Announcements

- Login piazza, iLearn
- Check our website — where you can find the slides, the schedule, the syllabus, the complete schedule of classes
- Reading quiz due this Wednesday before class

Computer
Science &
Engineering

203

つづく