

# *Introduction to Computer Architecture*

## Lecture 2: Tradeoffs, Metrics, Mindset

Pooyan Jamshidi

Week 2: January 16



**Engineering  
and Computing**

UNIVERSITY OF SOUTH CAROLINA

CSC212: Introduction to Computer Architecture | Spring 2024 | <https://pooyanjamshidi.github.io/csce212/>

[Slides are primarily based on those of Onur Mutlu for the Computer Architecture Course at CMU]

Many Interesting Things  
Are Happening Today  
in Computer Architecture

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**Performance**  
**Energy Efficiency**  
**Sustainability**  
**Specialized Accelerators**

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**Reliability**  
**Safety**  
**Security**  
**Privacy**

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**More Demanding Workloads**

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**New (Device) Technologies**

# The Problem

Computing

is Bottlenecked by Data

# Data is Key for AI, ML, Genomics, ...

---

- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
  - We can generate more than we can process



# Data is Key for Future Workloads

---



## In-memory Databases

[e.g., Redis]



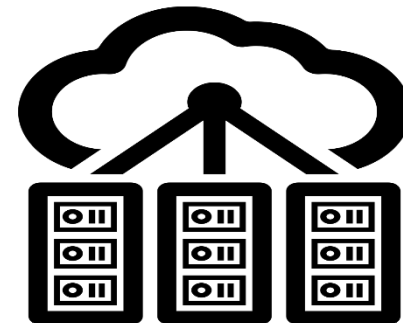
## Graph/Tree Processing

[Memgraph]



## In-Memory Data Analytics

[e.g., Spark, Hazelcast]



## Datacenter Workloads

[Google cloud]

---

# Data is Key for Future Workloads

---



**In-memory Databases**



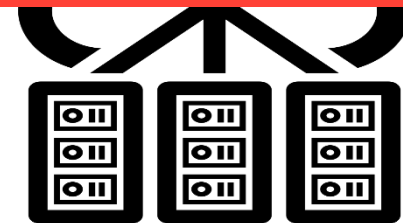
**Graph/Tree Processing**

Data → performance & energy bottleneck



**In-Memory Data Analytics**

[e.g., Spark, Hazelcast]



**Datacenter Workloads**

[Google cloud]

---

# Data is Key for Future Workloads

---



## Chrome

Google's web browser



## TensorFlow Mobile

Google's machine learning framework

# VP9



## Video Playback

Google's **video codec**

# VP9



## Video Capture

Google's **video codec**

# Data Overwhelms Modern Machines

---



**Chrome**



**TensorFlow Mobile**

Data → performance & energy bottleneck

**VP9**



**Video Playback**

Google's **video codec**

**VP9**



**Video Capture**

Google's **video codec**

# Data Movement Overwhelms Modern Machines

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, "[Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks](#)" *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Williamsburg, VA, USA, March 2018.

**62.7%** of the total system energy  
is spent on **data movement**

## Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

Daehyun Kim<sup>4,3</sup>

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

Onur Mutlu<sup>5,1</sup>

# Data Movement Overwhelms Accelerators

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,  
["Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"](#)  
*Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Virtual, September 2021.  
[[Slides \(pptx\)](#)] ([pdf](#))  
[[Talk Video](#) (14 minutes)]

**> 90% of the total system energy is spent on memory in large ML models**

## Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand<sup>†◇</sup>  
Geraldo F. Oliveira<sup>\*</sup>

Saugata Ghose<sup>‡</sup>  
Xiaoyu Ma<sup>§</sup>

Berkin Akin<sup>§</sup>  
Eric Shiu<sup>§</sup>

Ravi Narayanaswami<sup>§</sup>  
Onur Mutlu<sup>\*†</sup>

<sup>†</sup>Carnegie Mellon Univ.

<sup>◇</sup>Stanford Univ.

<sup>‡</sup>Univ. of Illinois Urbana-Champaign

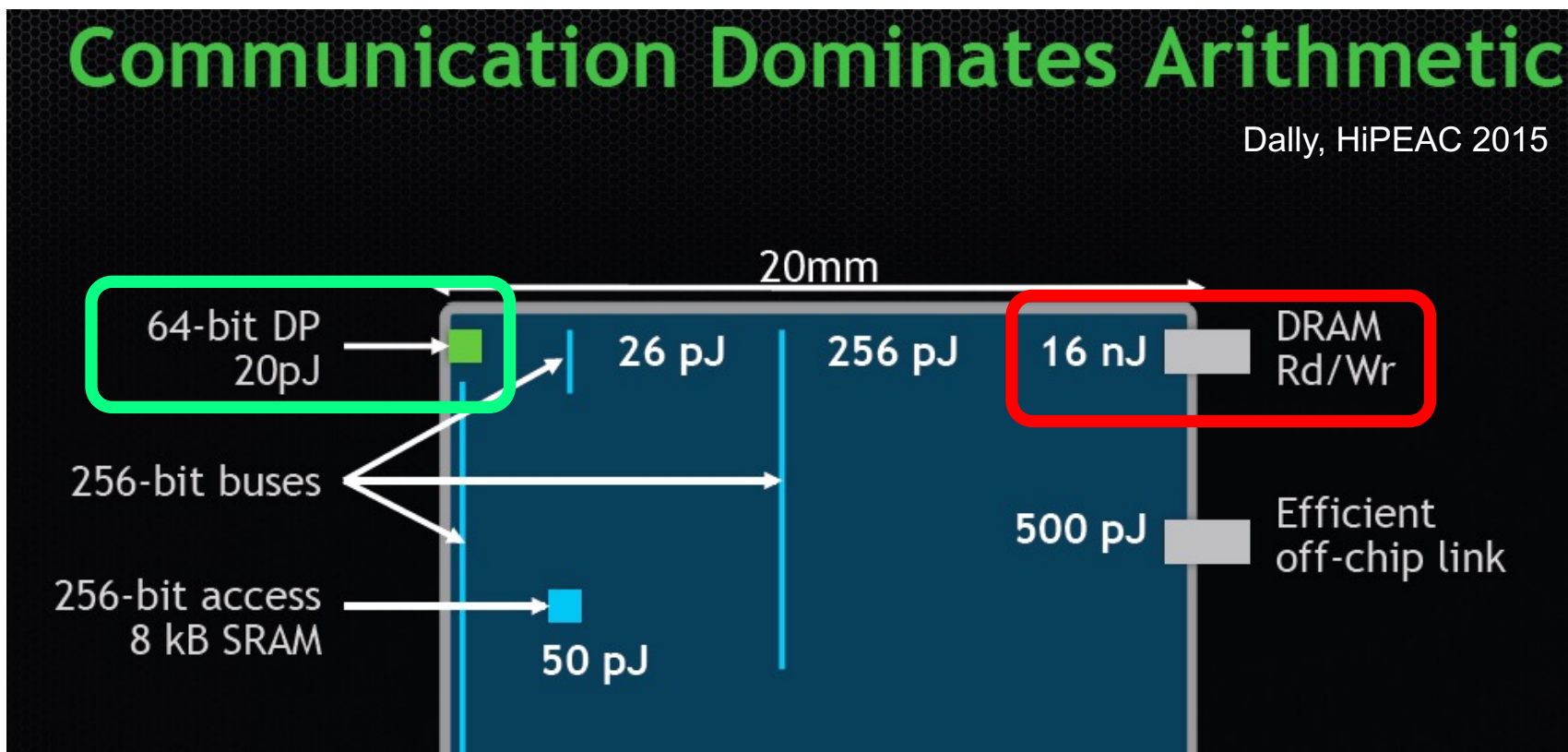
<sup>§</sup>Google

<sup>\*</sup>ETH Zürich

# Data Movement vs. Computation Energy

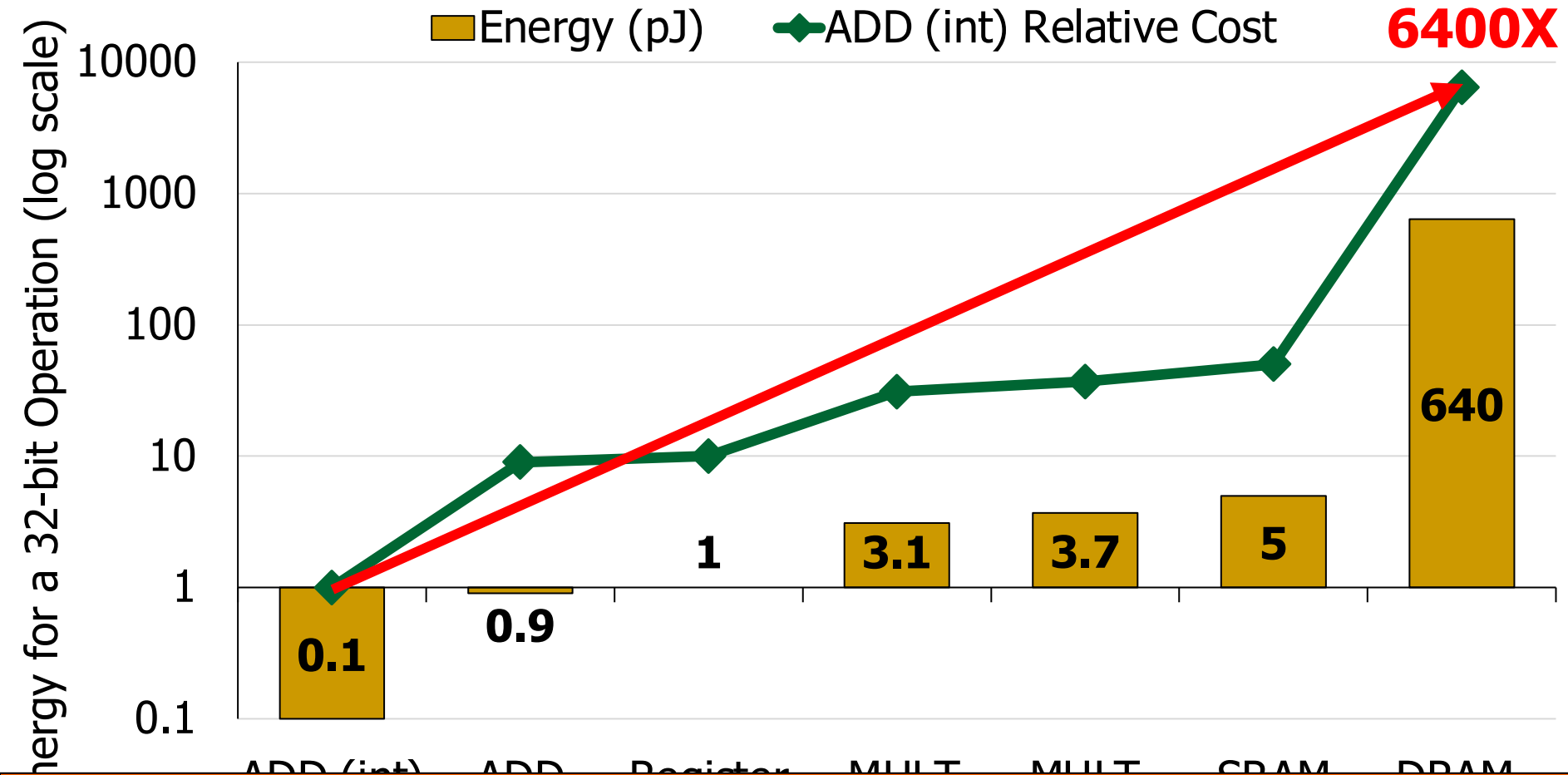
## Communication Dominates Arithmetic

Dally, HiPEAC 2015



A memory access consumes  $\sim 100-1000X$  the energy of a complex addition

# Data Movement vs. Computation Energy



A memory access consumes 6400X  
the energy of a simple integer addition



Many Interesting Things  
Are Happening Today  
in Computer Architecture

# Many Novel Concepts Investigated Today

---

- **New Computing Paradigms (Rethinking the Full Stack)**
  - Processing in Memory, Processing Near Data
  - Neuromorphic Computing, Quantum Computing
  - Fundamentally Secure and Dependable Computers
- **New Accelerators & Systems (Algorithm-Hardware Co-Designs)**
  - Artificial Intelligence & Machine Learning
  - Graph & Data Analytics, Vision, Video
  - Genome Analysis
- **New Memories, Storage Systems, Interconnects, Devices**
  - Non-Volatile Main Memory, Intelligent Memory Systems, Quantum
  - High-Speed Interconnects, Disaggregated Systems

# Increasingly Demanding Applications

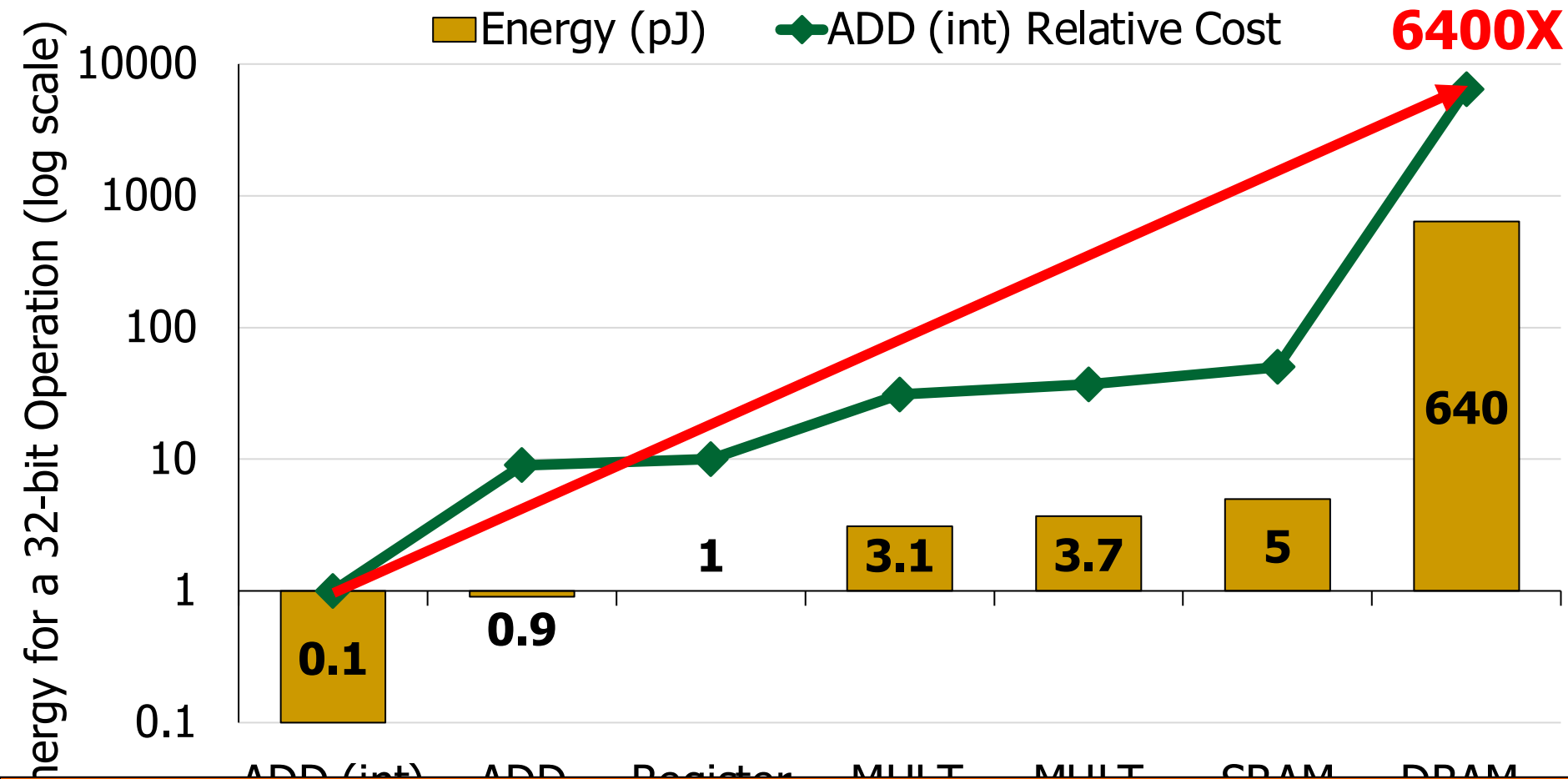
---

**Dream**

**and, they will come**

As applications push boundaries, computing platforms will become increasingly strained.

# Increasingly Diverging/Complex Tradeoffs

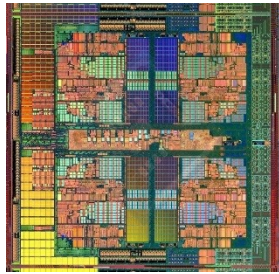


A memory access consumes 6400X  
the energy of a simple integer addition

# Increasingly Complex Systems

---

Past systems



Microprocessor



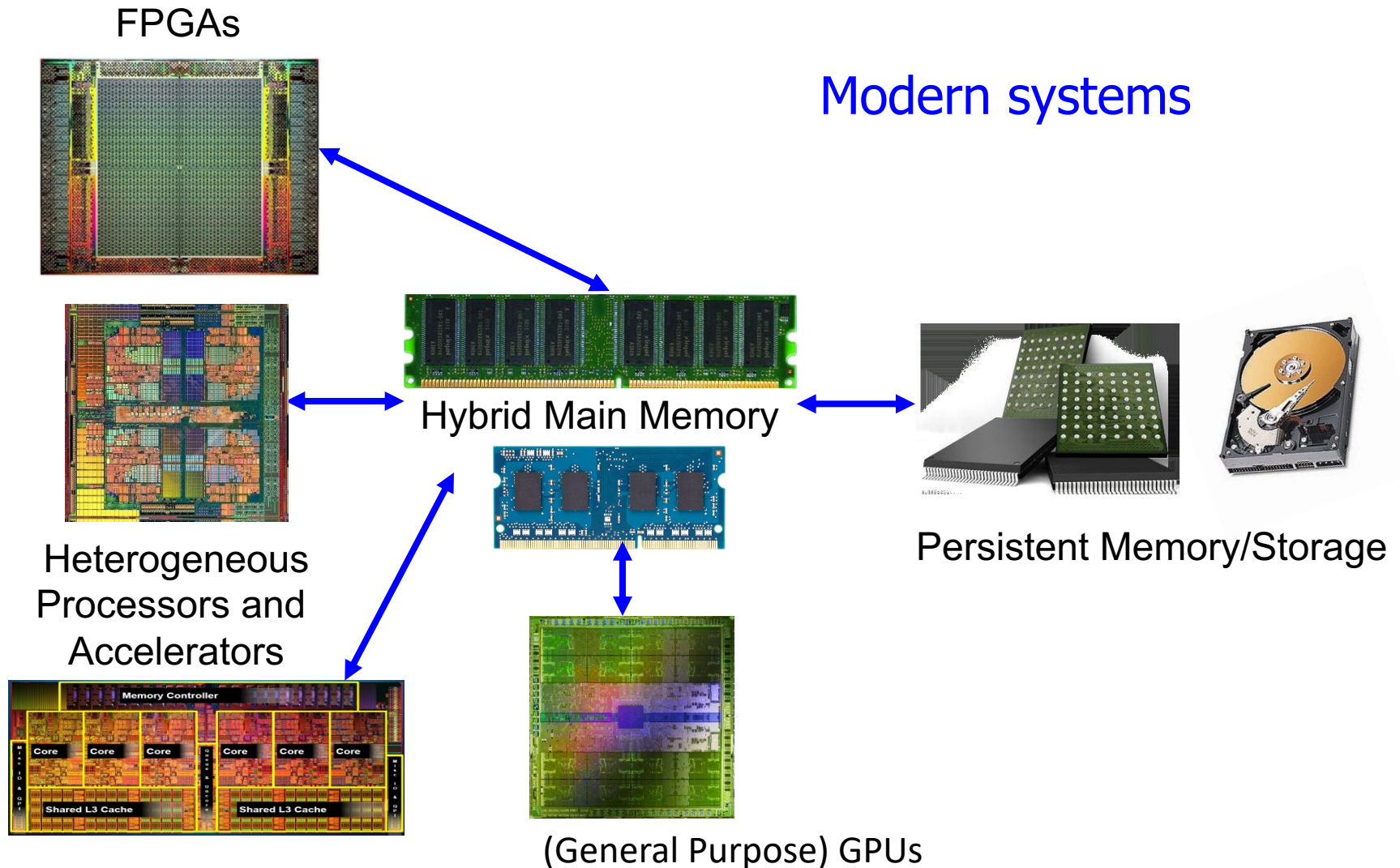
Main Memory



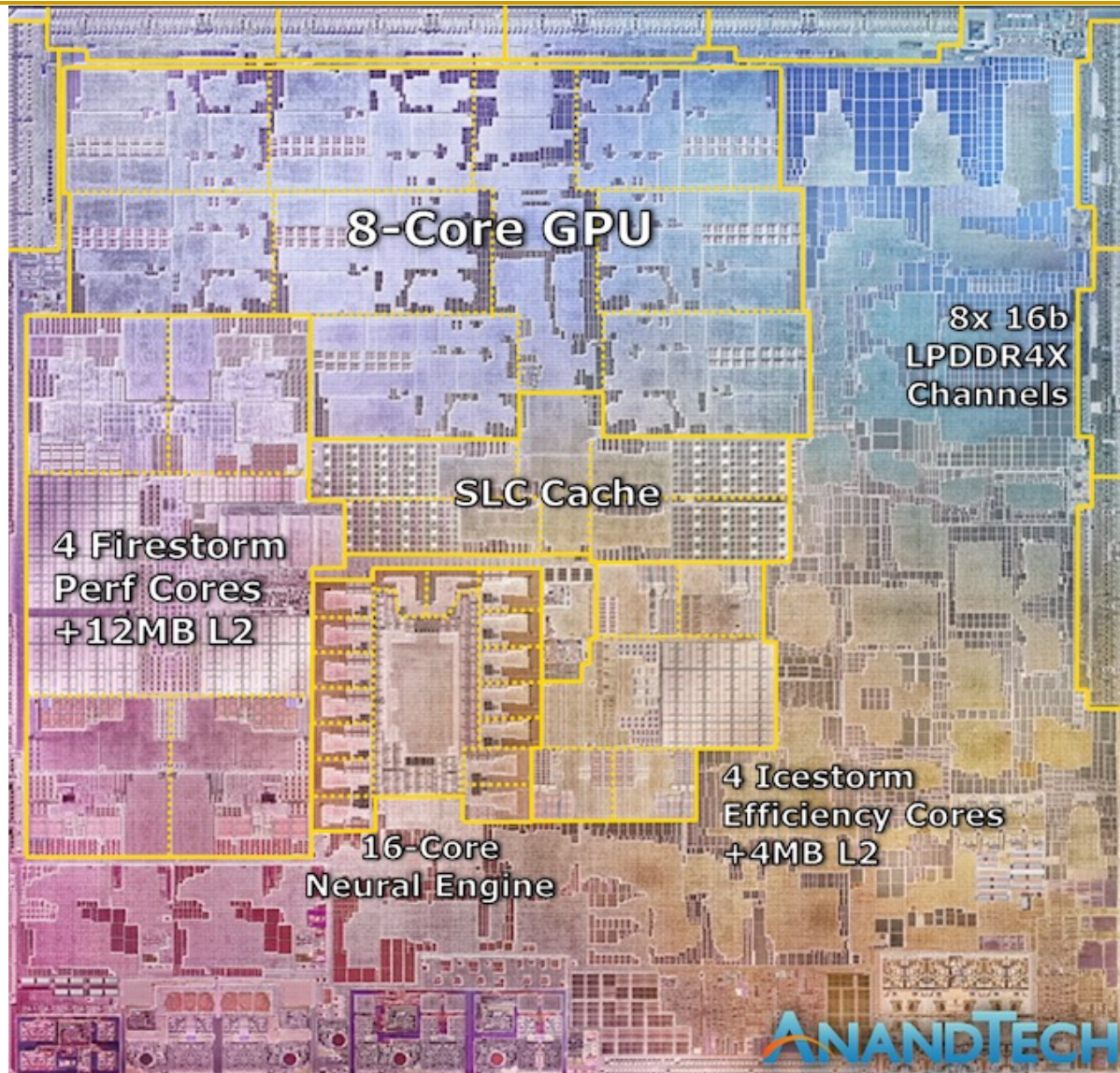
Storage (SSD/HDD)



# Increasingly Complex Systems



# Increasingly Complex Systems on Chip

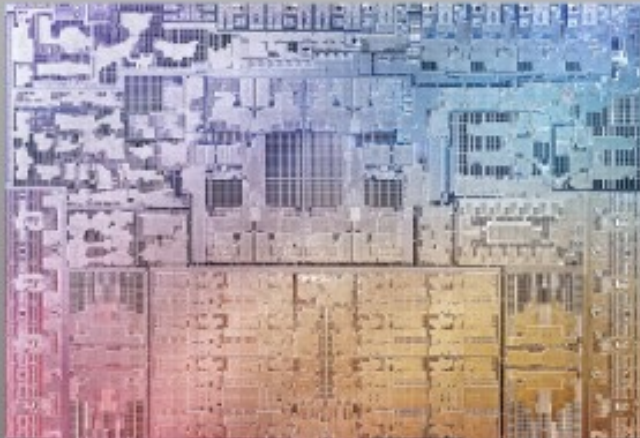


# Bigger and More Powerful Systems (2021)

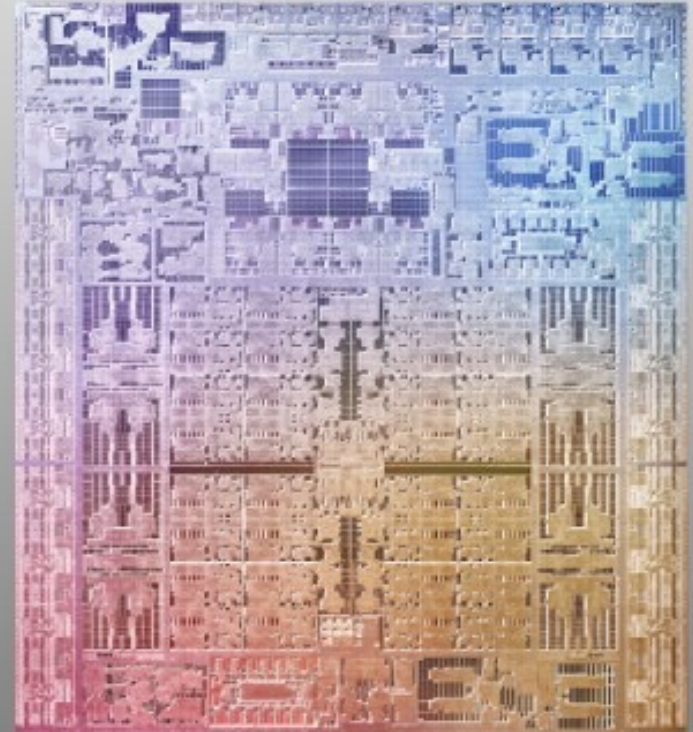
---



Apple M1



Apple M1 Pro

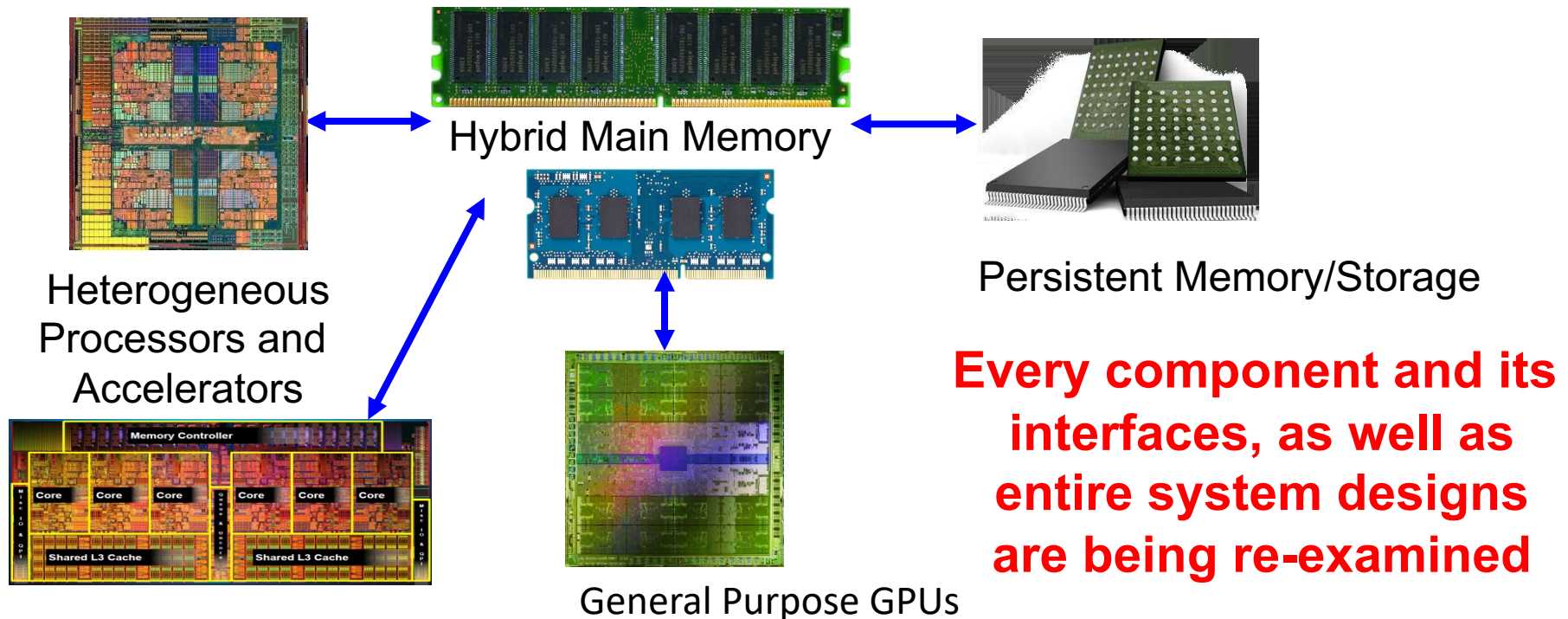


Apple M1 Max



# Computer Architecture Today

- Computing landscape is very different from 10-20 years ago
- Applications and technology both demand novel architectures



# Computer Architecture Today (II)

---

- You can revolutionize the way computers are built, if you understand both the hardware and the software (and change each accordingly)
- You can invent new paradigms for computation, communication, and storage
- Recommended book: Thomas Kuhn, “[The Structure of Scientific Revolutions](#)” (1962)
  - Pre-paradigm science: no clear consensus in the field
  - Normal science: dominant theory used to explain/improve things (business as usual); exceptions considered anomalies
  - Revolutionary science: underlying assumptions re-examined

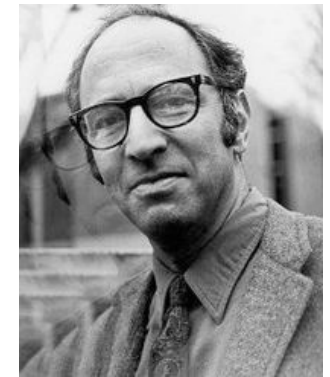
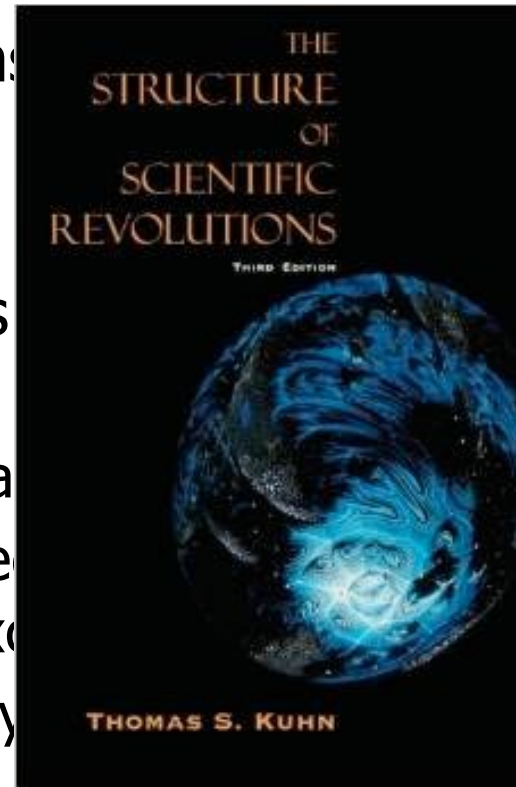
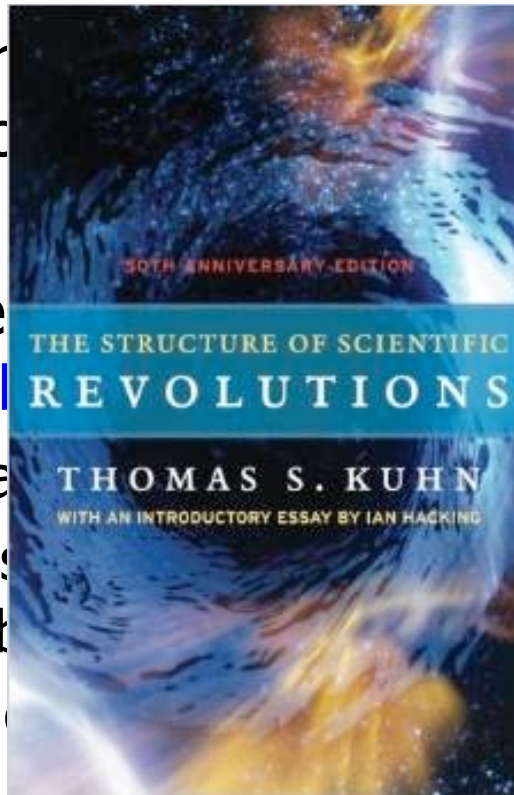
# Computer Architecture Today (II)

- You can revolutionize the way computers are built, if you understand both the hardware and the software (and change each accordingly)

- You can improve communication

- Recommended reading: **Scientific Revolutions**

- Pre-prepare
- Normal scientific things (the exception)
- Revolutionary



ure of  
eld  
improve  
anomalies  
examined

# Takeaways

---

- It is an exciting time to be understanding and designing computing architectures
- Many challenging and exciting problems in platform design
  - That no one has tackled (or thought about) before
  - That can have huge impact on the world's future
- Driven by huge hunger for data (Big Data), new applications (ML/AI, graph analytics, genomics), ever-greater realism, ...
  - We can easily collect more data than we can analyze/understand
- Driven by significant difficulties in keeping up with that hunger at the technology layer
  - Five walls: Energy, reliability, complexity, security, scalability

# Let's Start with Some Puzzles

a.k.a. Computer Architecture resembles Building Architecture

# What Is This?



# What About This?



What Do the Following  
Have in Common?



# What Do the Following

---

## Have in Common?

- **Neoclassical Architecture:**

- **South Carolina State House:** This building is a prime example of Neoclassical architecture with its large copper dome, Corinthian columns, and grand portico.
- **University of South Carolina's McKissick Museum:** Located on the historic Horseshoe, this building also exhibits Neoclassical features, with its symmetrical facade and columns.





What Do the Following  
Have in Common?

# What Do the Following

---

## Have in Common?

- **Antebellum Architecture:**

- **Robert Mills House:** A classic example of Antebellum architecture, known for its Greek Revival style with columns and a symmetrical facade.
- **Hampton-Preston Mansion:** Another Antebellum building, showcasing similar Greek Revival elements and a balanced, grand architectural style.





What Do the Following  
Have in Common?



# What Do the Following Have in Common?

---

- **Modern and Contemporary Designs:**
  - **Darla Moore School of Business, University of South Carolina:** Featuring modern architectural elements, glass facades, and a contemporary design.
  - **Columbia Metropolitan Convention Center:** Also modern in design, with extensive use of glass and a focus on sleek, clean lines.

# Gare do Oriente, Lisbon

---



# Milwaukee Art Museum



# Athens Olympic Stadium

---



# City of Arts and Sciences, Valencia

---



# Florida Polytechnic University (I)

---



# Oculus, New York City

---



What do All Those Have in  
Common?



# But First, Today's First Assignment

---

# Your First Comp. Architecture Assignment

---

- Go and find several pairs of buildings in Columbia that share a common architectural style.
- Appreciate the beauty & out-of-the-box, and creative thinking
- Think about tradeoffs in the design
  - Strengths, weaknesses, and goals of design
- Derive principles on your own for good design and innovation
- Due date: **Any time during or after this course**
  - Later during the course is better
  - Apply what you have learned in this course
  - Think out-of-the-box

# Aside: Evaluation Criteria for the Designs

---

- Functionality (Does it meet the specification?)
  - Reliability
  - Space requirement
  - Cost
  - Expandability
  - Comfort level of users
  - Happiness level of users
  - Aesthetics
  - Security
  - ...
- 
- How to evaluate goodness of design is always a critical question → "Performance" evaluation and metrics

# A Key Question

---

- How was Calavatra able to design especially his key buildings?
  - Can have many guesses
    - (Very) hard work, perseverance, dedication (over decades)
    - Experience
    - Creativity, Out-of-the-box thinking
    - A good understanding of past designs
    - Good judgment and intuition
    - Strong skill combination (math, architecture, art, engineering, ...)
    - Funding (\$\$\$\$), luck, initiative, entrepreneurialism
    - Strong understanding of and commitment to fundamentals
    - Principled design
    - ...
  - You will be exposed to and hopefully develop/enhance many of these skills in this course
-

# Principled Design

---

- “To me, there are **two overriding principles** to be found in nature which are most appropriate for building:
  - one is the **optimal use of material**,
  - the other **the capacity of organisms to change shape, to grow, and to move.**”
  - *Santiago Calatrava*
  
- “Calatrava's constructions are inspired by natural forms like plants, bird wings, and the human body.”

# Gare do Oriente, Lisbon, Revisited



Source: By Martín Gómez Tagle - Lisbon, Portugal, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=13764903>

Source: <http://www.arcspace.com/exhibitions/unsorted/santiago-calatrava/>

# A Principled Design

---

## Zoomorphic architecture

---

From Wikipedia, the free encyclopedia

**Zoomorphic architecture** is the practice of using animal forms as the inspirational basis and blueprint for architectural design. "While animal forms have always played a role adding some of the deepest layers of meaning in architecture, it is now becoming evident that a new strand of **biomorphism** is emerging where the meaning derives not from any specific representation but from a more general allusion to biological processes."<sup>[1]</sup>

Some well-known examples of Zoomorphic architecture can be found in the [TWA Flight Center](#) building in [New York City](#), by [Eero Saarinen](#), or the [Milwaukee Art Museum](#) by [Santiago Calatrava](#), both inspired by the form of a bird's wings.<sup>[3]</sup>

# What Does This Remind You Of?

---





# The Architect's Answer

---

## Design [\[ edit \]](#)

Calatrava said that the Oculus resembles a bird being released from a child's hand. The roof was originally designed to mechanically open to increase light and ventilation to the enclosed space. [Herbert Muschamp](#), architecture critic of *The New York Times*, compared the design to the [Bethesda Terrace and Fountain](#) in [Central Park](#), and wrote in 2004:

# Strengths and Praise

---

“ Santiago Calatrava's design for the World Trade Center PATH station should satisfy those who believe that buildings planned for ground zero must aspire to a spiritual dimension. Over the years, many people have discerned a metaphysical element in Mr. Calatrava's work. I hope New Yorkers will detect its presence, too. With deep appreciation, I congratulate the Port Authority for commissioning Mr. Calatrava, the great Spanish architect and engineer, to design a building with the power to shape the future of New York. It is a pleasure to report, for once, that public officials are not overstating the case when they describe a design as breathtaking.<sup>[43]</sup>”

# Design Constraints and Criticism

---

However, Calatrava's original soaring spike design was scaled back because of security issues. The *New York Times* observed in 2005:

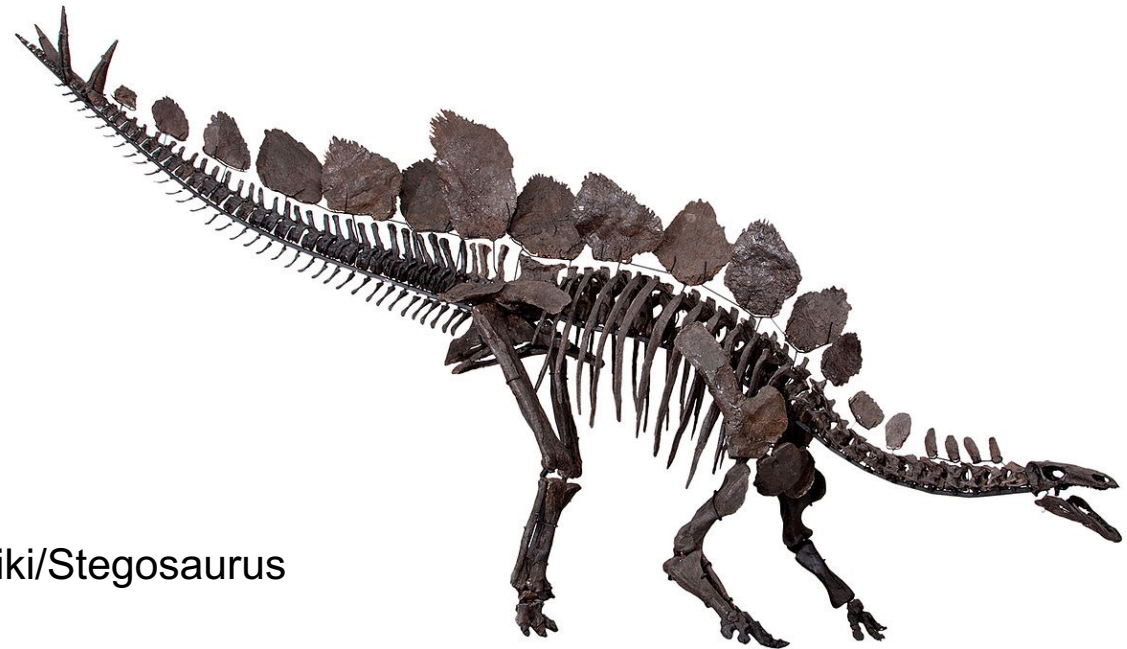
“ In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass.... [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural expressiveness. It may now evoke a slender *stegosaurus* more than it does a bird.<sup>[45]</sup> ”

# Stegosaurus

From Wikipedia, the free encyclopedia

For the *pachycephalosaurid* of a similar name, see *Stegoceras*.

**Stegosaurus** (/ˌstɛɡəˈsɔːrəs/<sup>[1]</sup>) is a genus of armored dinosaur. Fossils of this genus date to the Late Jurassic period, where they are found in Kimmeridgian to early Tithonian aged strata, between 155 and 150 million years ago, in the western United States and Portugal. Several



Source: <https://en.wikipedia.org/wiki/Stegosaurus>

Susannah Maidment et al. & Natural History Museum, London – Maidment SCR, Brassey C, Barrett PM (2015) The Postcranial Skeleton of an Exceptionally Complete Individual of the Plated Dinosaur *Stegosaurus stenops* (Dinosauria: Thyreophora) from the Upper Jurassic Morrison Formation of Wyoming, U.S.A. PLoS ONE 10(10): e0138352. doi:10.1371/journal.pone.0138352

# Design Constraints: Noone is Immune

---

However, Calatrava's original soaring spike design was scaled back because of security issues. The *New York Times* observed in 2005:

“ In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass.... [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural expressiveness. It may now evoke a slender *stegosaurus* more than it does a bird.<sup>[45]</sup> ”

The design was further modified in 2008 to eliminate the opening and closing roof mechanism because of budget and space constraints.<sup>[46]</sup>

The Transportation Hub has been dubbed "the world's most expensive transportation hub" for its massive cost for reconstruction—\$3.74 billion dollars.<sup>[48][58]</sup> By contrast, the proposed two-mile PATH extension

# The Lecture Was Slightly Different When I Was at CMU

# What Is This?



# Answer: Masterpiece of A Famous Architect

---

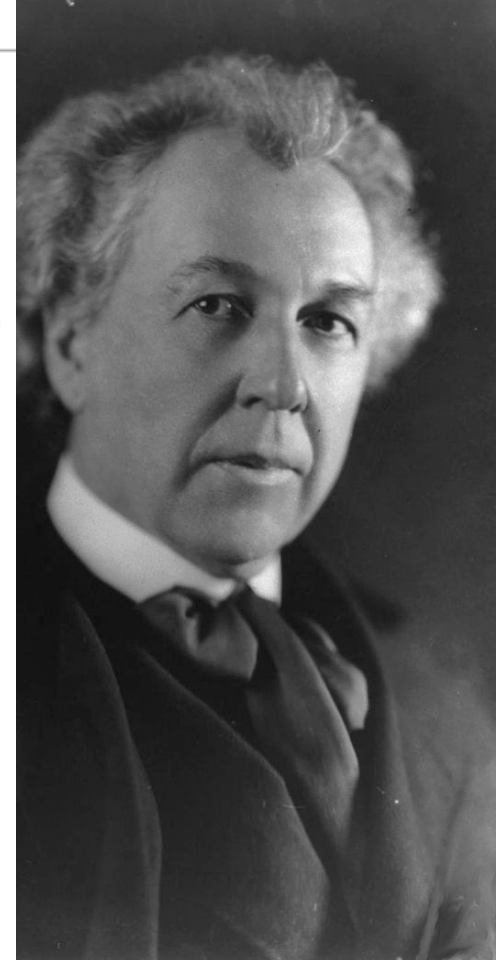
## Fallingwater

---

From Wikipedia, the free encyclopedia

**Fallingwater** or **Kaufmann Residence** is a house designed by architect [Frank Lloyd Wright](#) in 1935 in rural [southwestern Pennsylvania](#), 43 miles (69 km) southeast of [Pittsburgh](#).<sup>[4]</sup> The home was built partly over a waterfall on [Bear Run](#) in the Mill Run section of [Stewart Township, Fayette County, Pennsylvania](#), in the [Laurel Highlands](#) of the [Allegheny Mountains](#).

*Time* cited it after its completion as Wright's "most beautiful job";<sup>[5]</sup> it is listed among *Smithsonian's* Life List of 28 places "to visit before you die."<sup>[6]</sup> It was designated a [National Historic Landmark](#) in 1966.<sup>[3]</sup> In 1991, members of the [American Institute of Architects](#) named the house the "best all-time work of American architecture" and in 2007, it was ranked twenty-ninth on the [list of America's Favorite Architecture](#) according to the [AIA](#).





# Find The Differences of This and That

# This

---



# This

---



# That

---



# A Key Question

---

- How was Wright able to design his masterpiece?
  - Can have many guesses
    - (Very) hard work, perseverance, dedication (over decades)
    - Experience
    - Creativity, Out-of-the-box thinking
    - A good understanding of past designs
    - Good judgment and intuition
    - Strong skill combination (math, architecture, art, engineering, ...)
    - Funding (\$\$\$\$), luck, initiative, entrepreneurialism
    - Strong understanding of and commitment to fundamentals
    - Principled design
    - ...
  - You will be exposed to and hopefully develop/enhance many of these skills in this course
-

# A Quote from The Architect Himself

---

- “architecture [...] based upon **principle**, and not upon **precedent**”



# A Principled Design

---

## Organic architecture

---

From Wikipedia, the free encyclopedia

**Organic architecture** is a [philosophy](#) of [architecture](#) which promotes harmony between human habitation and the natural world through design approaches so sympathetic and well integrated with its site, that buildings, furnishings, and surroundings become part of a unified, interrelated composition.

A well-known example of organic architecture is [Fallingwater](#), the residence Frank Lloyd Wright designed for the Kaufmann family in rural Pennsylvania. Wright had many choices to locate a home on this large site, but chose to place the home directly over the waterfall and creek creating a close, yet noisy dialog with the rushing water and the steep site. The horizontal striations of stone masonry with daring [cantilevers](#) of colored beige concrete blend with native rock outcroppings and the wooded environment.

# A Key Question

---

- How was Wright able to design his masterpiece?
  - Can have many guesses
    - (Very) hard work, perseverance, dedication (over decades)
    - Experience
    - Creativity, Out-of-the-box thinking
    - A good understanding of past designs
    - Good judgment and intuition
    - Strong skill combination (math, architecture, art, engineering, ...)
    - Funding (\$\$\$\$), luck, initiative, entrepreneurialism
    - Strong understanding of and commitment to fundamentals
    - Principled design
    - ...
  - You will be exposed to and hopefully develop/enhance many of these skills in this course
-



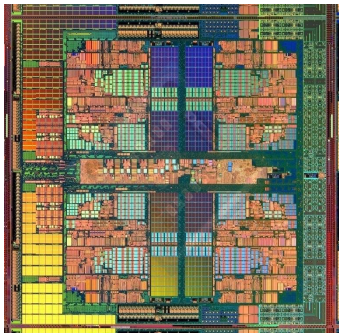
# Takeaways

---

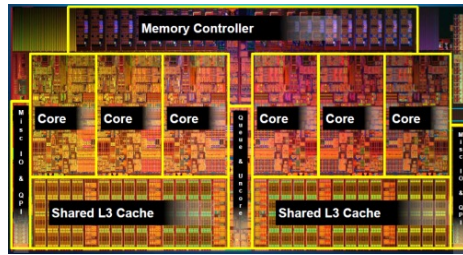
- It all starts from the basic building blocks and design principles
- And, knowledge of how to use, apply, enhance them
- Underlying technology might change (e.g., steel vs. wood)
  - but methods of taking advantage of technology bear resemblance
  - methods used for design depend on the principles employed

# The Same Applies to Processor Chips

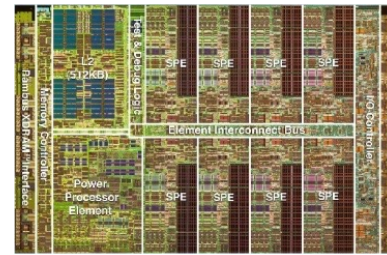
- There are **basic building blocks** and **design principles**



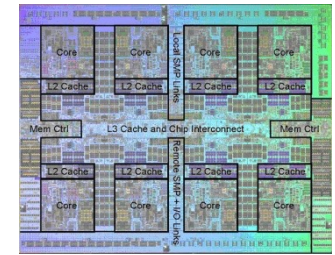
AMD Barcelona  
4 cores



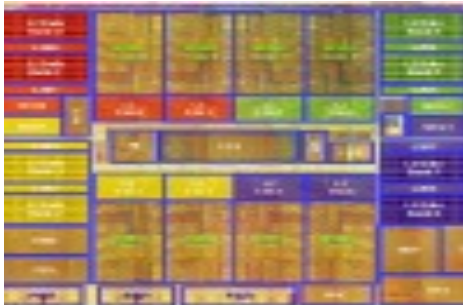
Intel Core i7  
8 cores



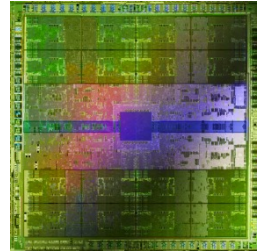
IBM Cell BE  
8+1 cores



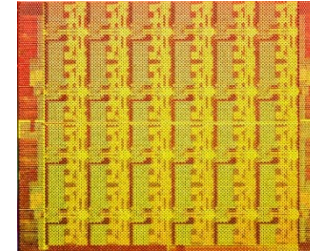
IBM POWER7  
8 cores



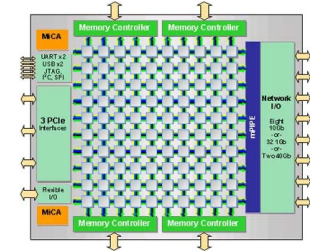
Sun Niagara II  
8 cores



Nvidia Fermi  
448 "cores"



Intel SCC  
48 cores, networked



Tiler TILE Gx  
100 cores, networked

# The Same Applies to Computing Systems

---

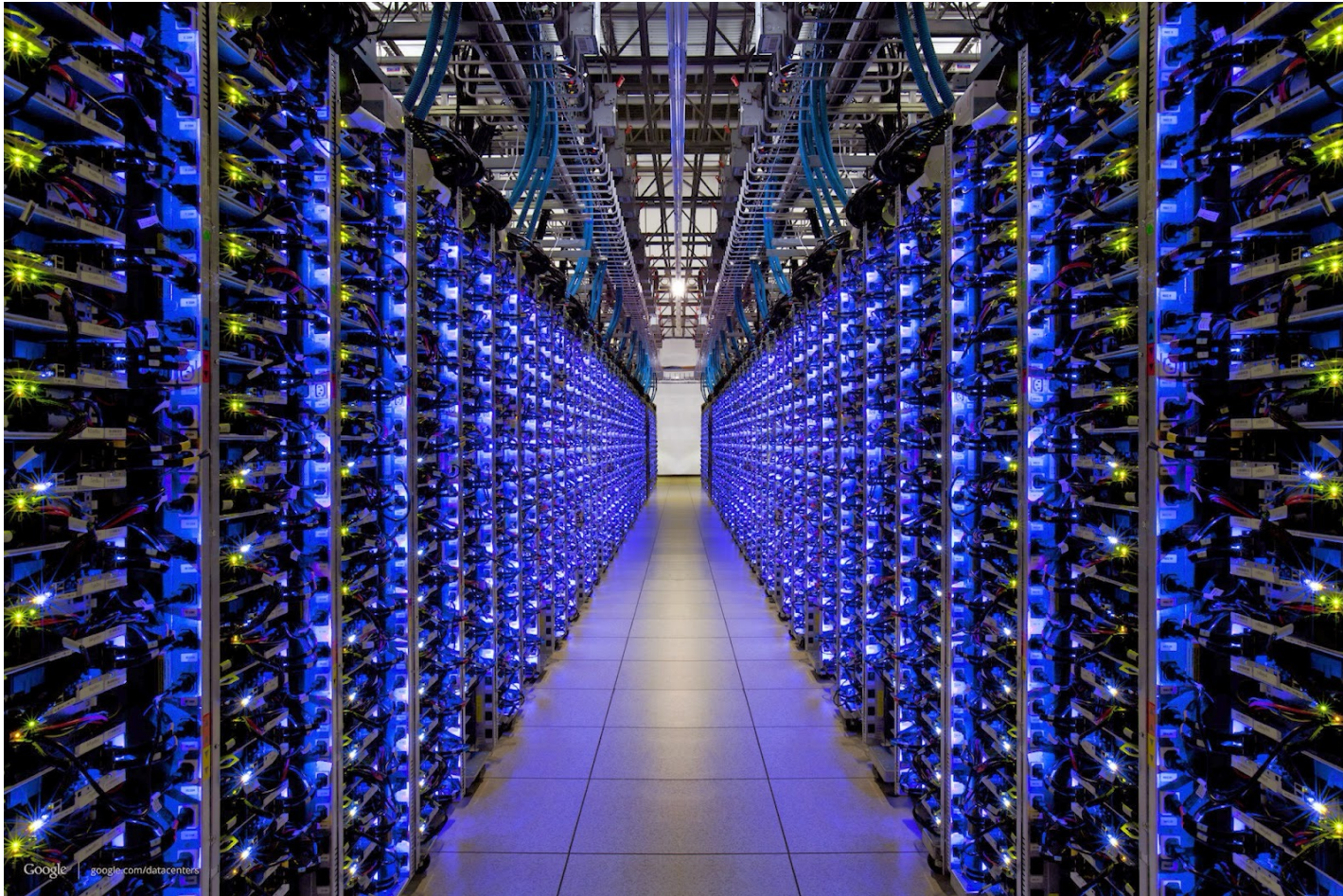
- There are **basic building blocks** and **design principles**



# The Same Applies to Computing Systems

---

- There are **basic building blocks** and **design principles**



# Different Platforms, Different Goals



# Different Platforms, Different Goals

---



# Different Platforms, Different Goals

---



# Different Platforms, Different Goals

---



Jack Dongara

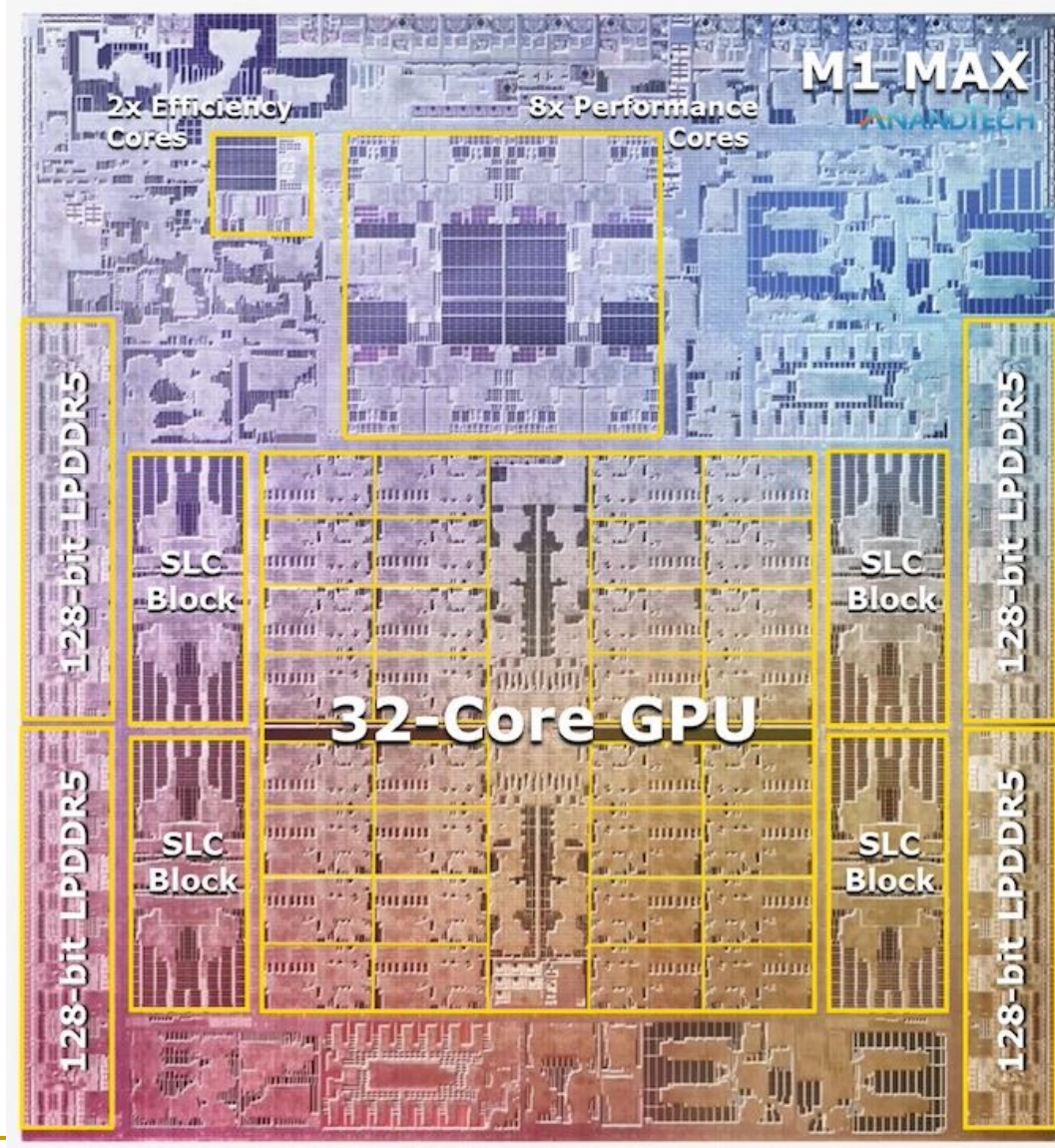


# Different Platforms, Different Goals

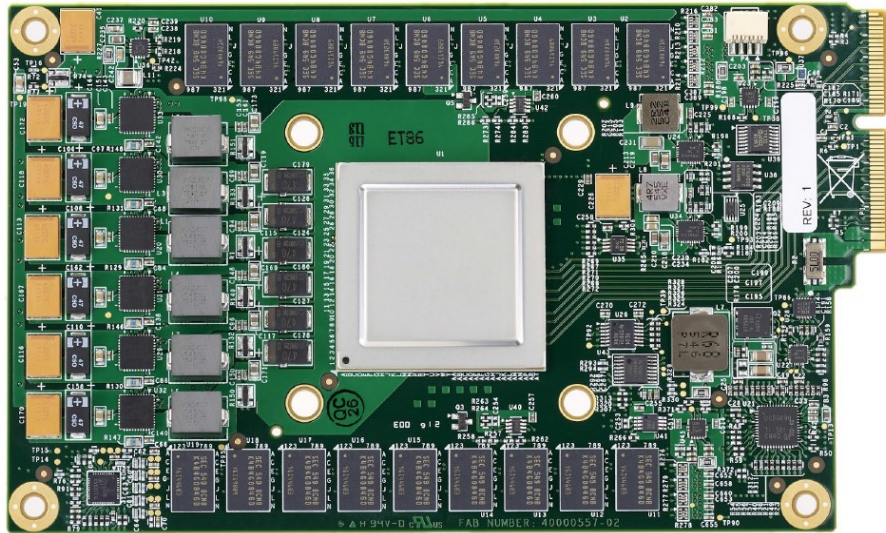
---



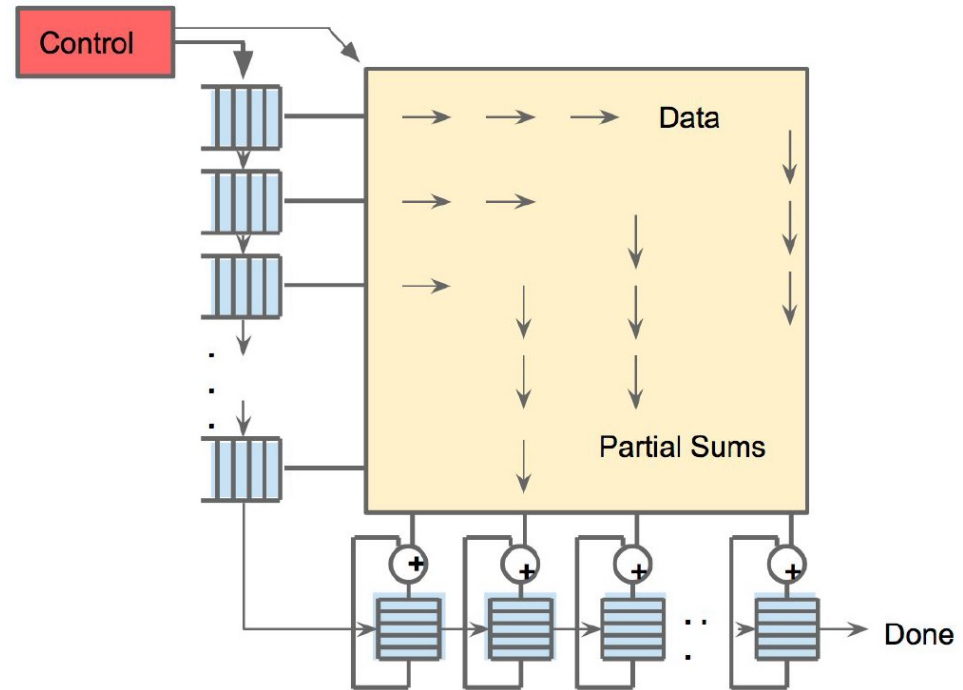
# Apple M1 Max System on Chip (2021)



# Google Tensor Processing Unit (~2016)



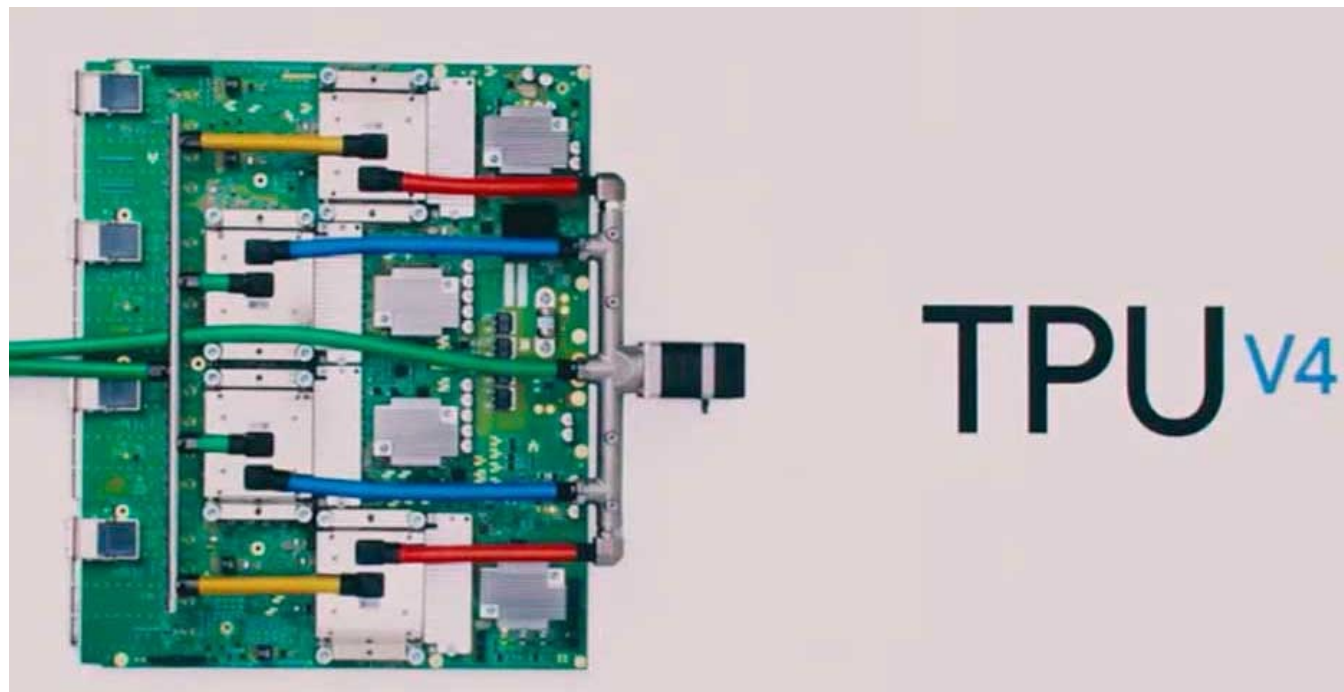
**Figure 3.** TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.



**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., “In-Datcenter Performance Analysis of a Tensor Processing Unit”, ISCA 2017.

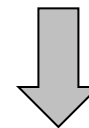
# Google TPU Generation IV (2021)



## New ML applications (vs. TPU3):

- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

250 TFLOPS per chip in 2021  
vs 90 TFLOPS in TPU3

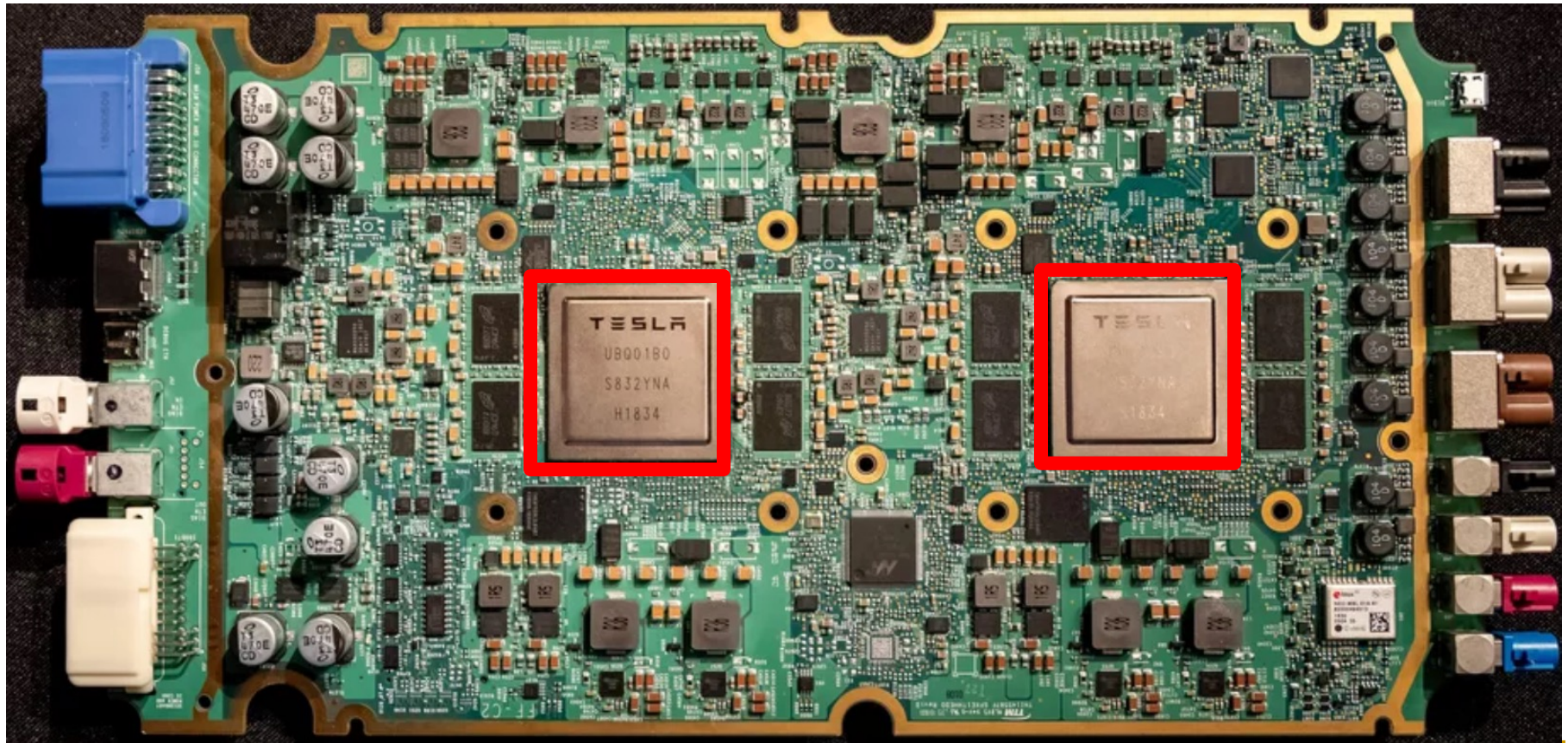


1 ExaFLOPS per board

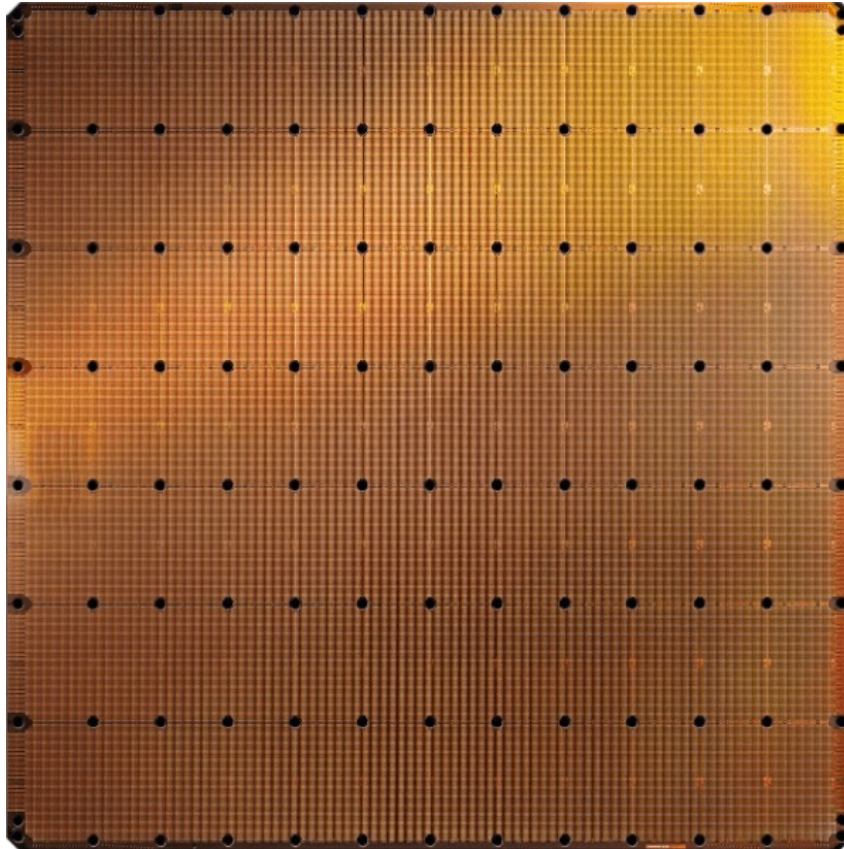
<https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests>

# TESLA Full Self-Driving Computer (2019)

- ML accelerator: 260 mm<sup>2</sup>, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



# Cerebras's Wafer Scale ML Engine-2 (2021)



**Cerebras WSE-2**  
2.6 Trillion transistors  
46,225 mm<sup>2</sup>

- The largest ML accelerator chip (2021)
- 850,000 cores



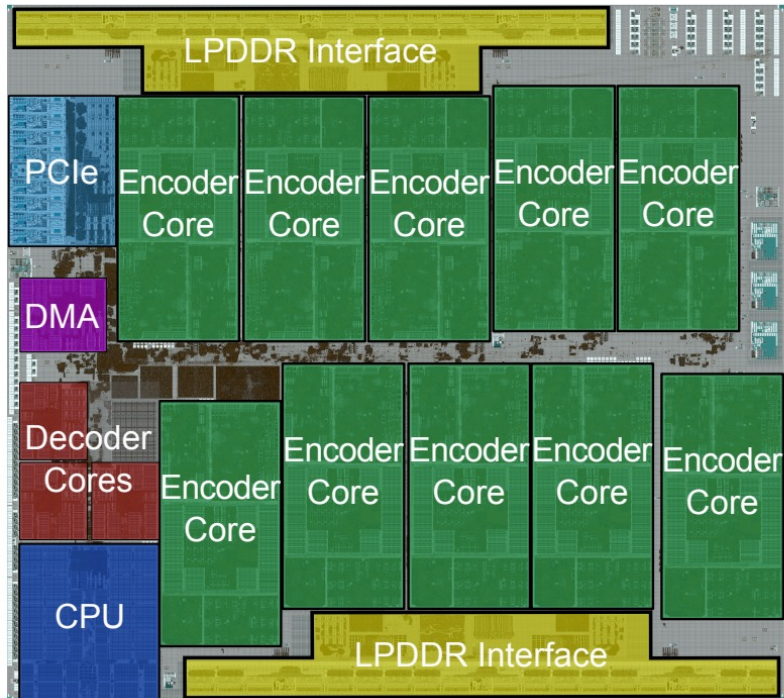
**Largest GPU**  
54.2 Billion transistors  
826 mm<sup>2</sup>  
NVIDIA Ampere GA100

<https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning>

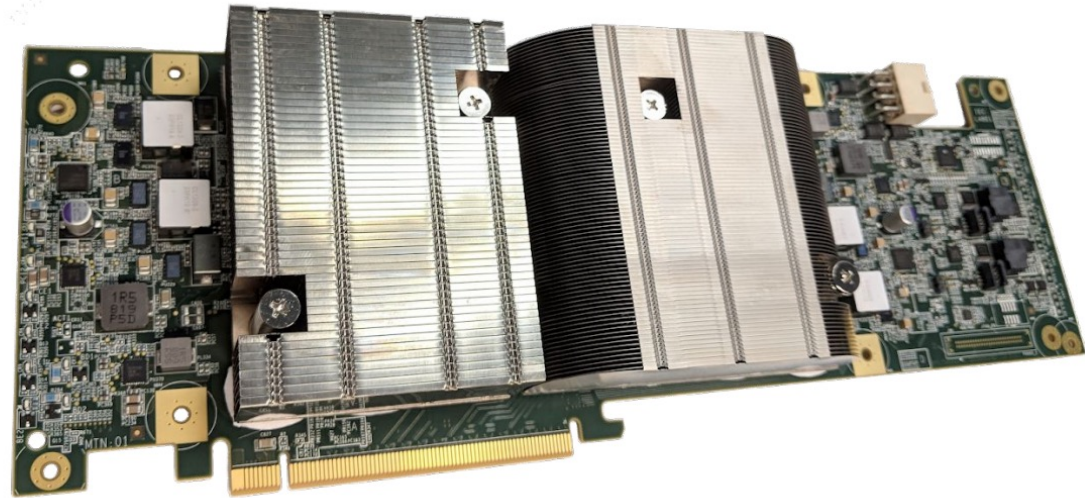
<https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/>

# Google's Video Coding Unit (2021)

Warehouse-Scale Video Acceleration: Co-design and Deployment in the Wild



**(a) Chip floorplan**

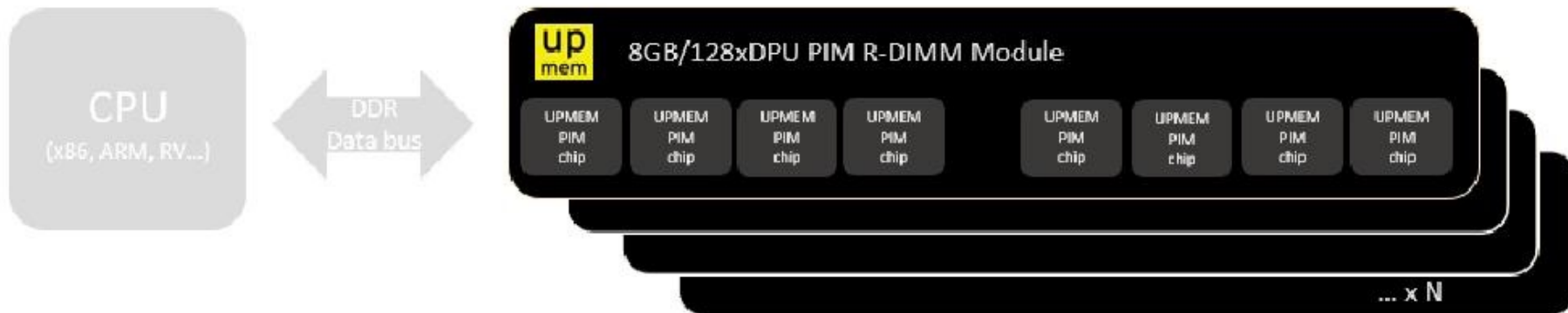


**(b) Two chips on a PCBA**

**Figure 5: Pictures of the VCU**

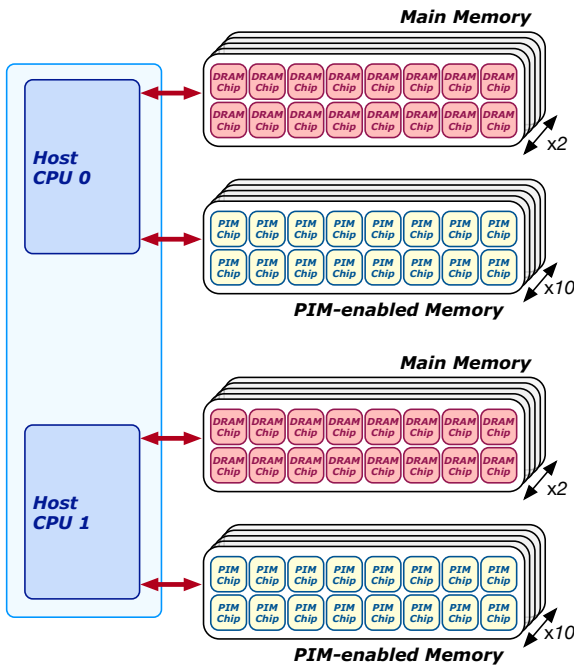
# UPMEM Processing-in-DRAM Engine (2019)

- **Processing in DRAM Engine**
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.
- Replaces **standard DIMMs**
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth





# Different Platforms, Different Goals



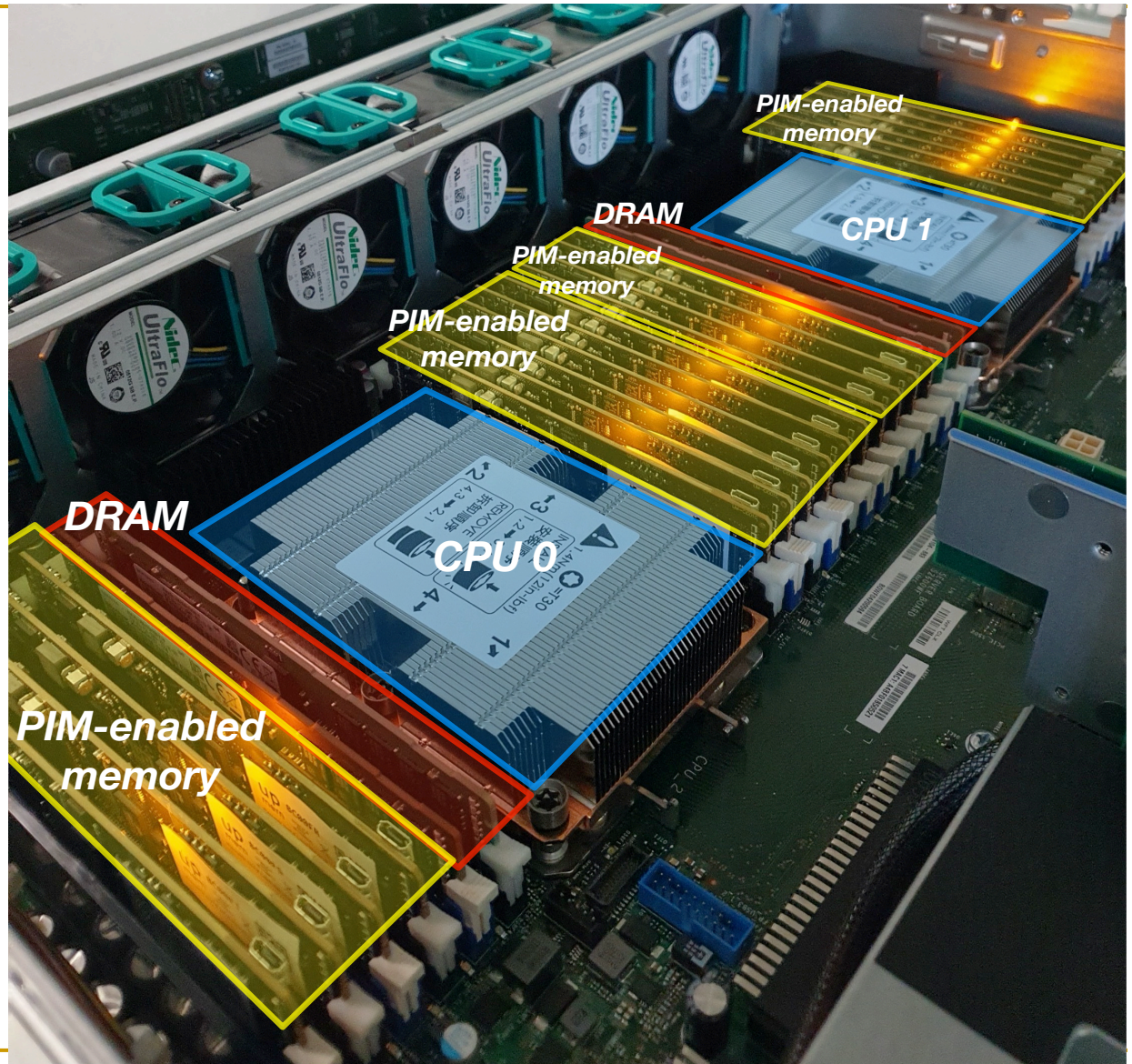
## Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland  
 IZZAT EL HAJJ, American University of Beirut, Lebanon  
 IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain  
 CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece  
 GERALDO F. OLIVEIRA, ETH Zürich, Switzerland  
 ONUR MUTLU, ETH Zürich, Switzerland

Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this *data movement bottleneck* requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as *processing-in-memory (PIM)*.

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called *DRAM Processing Units (DPUs)*, integrated in the same chip.

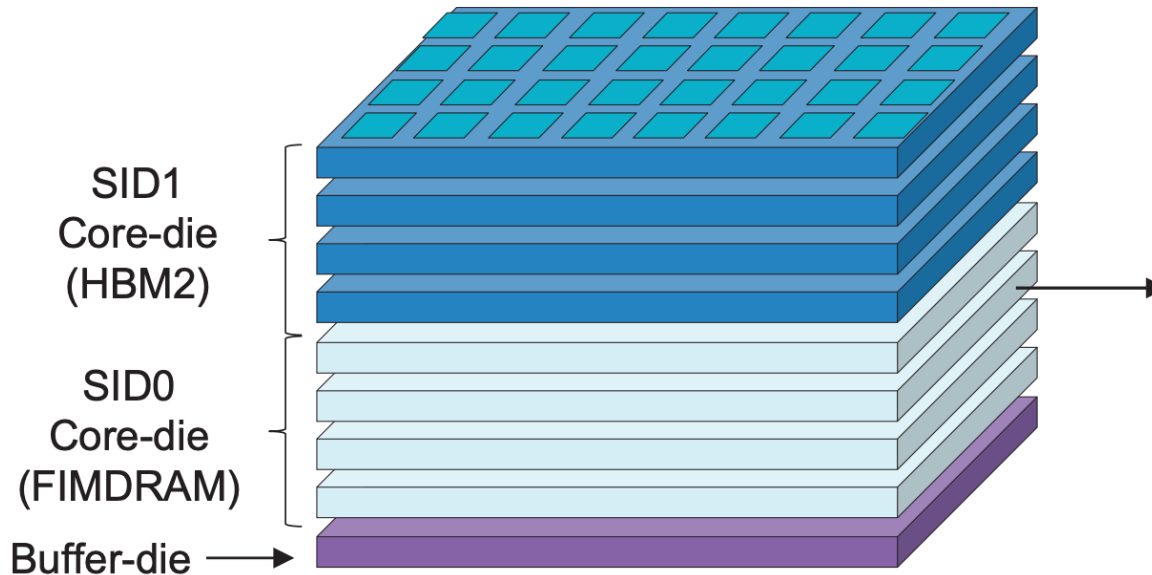
This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM (Processing-In-Memory benchmarks)*, a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 440 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.



<https://arxiv.org/pdf/2105.03814.pdf>

# Samsung Function-in-Memory DRAM (2021)

## ■ FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

### Chip Specification

128DQ / 8CH / 16 banks / BL4

32 PCU blocks (1 FIM block/2 banks)

1.2 TFLOPS (4H)

**FP16 ADD /  
Multiply (MUL) /  
Multiply-Accumulate (MAC) /  
Multiply-and- Add (MAD)**

ISSCC 2021 / SESSION 25 / DRAM / 25.4

**25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications**

Young-Cheon Kwon<sup>1</sup>, Suk Han Lee<sup>1</sup>, Jaehoon Lee<sup>1</sup>, Sang-Hyuk Kwon<sup>1</sup>, Je Min Ryu<sup>1</sup>, Jong-Pil Son<sup>1</sup>, Seongil O<sup>1</sup>, Hak-Soo Yu<sup>1</sup>, Haesuk Lee<sup>1</sup>, Soo Young Kim<sup>1</sup>, Youngmin Cho<sup>1</sup>, Jin Guk Kim<sup>1</sup>, Jongyoon Choi<sup>1</sup>, Hyun-Sung Shin<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyoungMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Choi<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shinhaeng Kang<sup>1</sup>, Yuhwan Ro<sup>3</sup>, Seungwoo Seo<sup>3</sup>, JoonHo Song<sup>3</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

<sup>1</sup>Samsung Electronics, Hwaseong, Korea

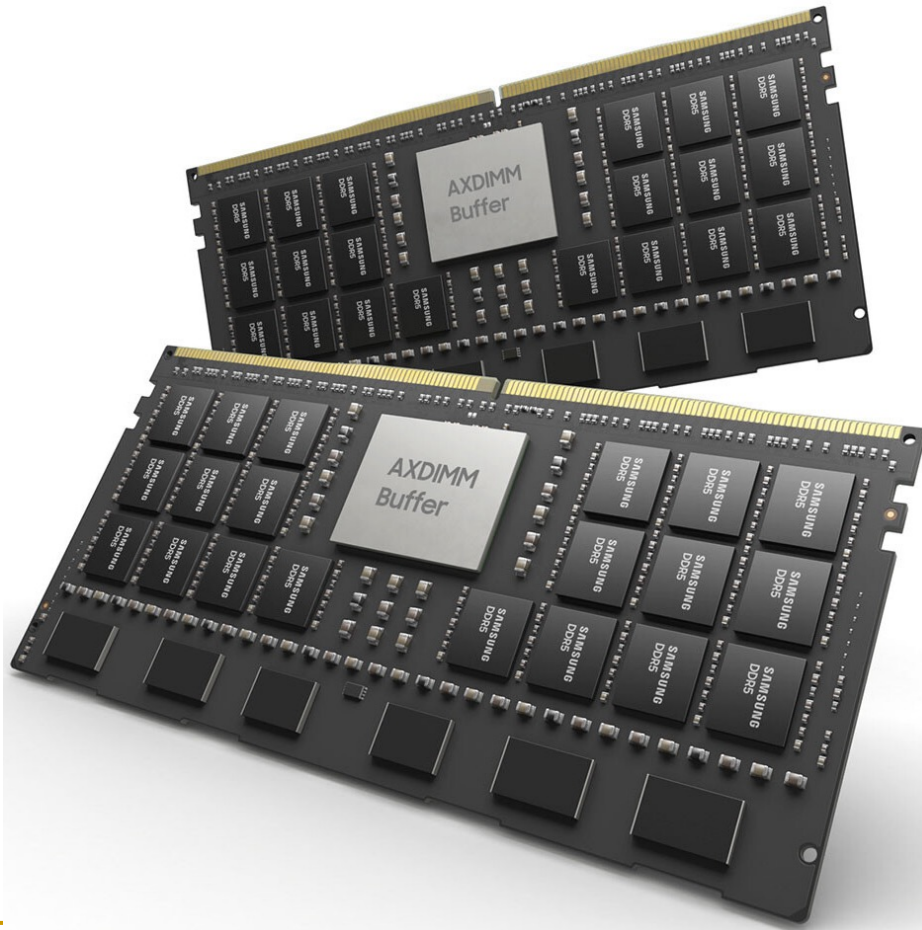
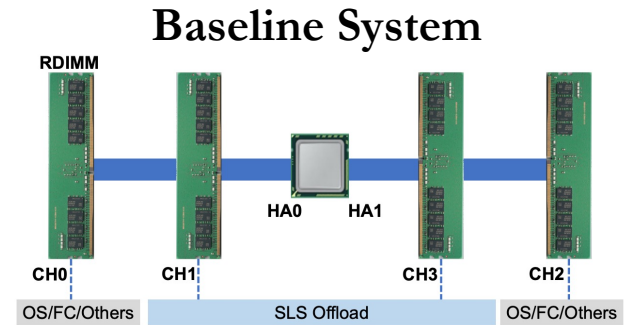
<sup>2</sup>Samsung Electronics, San Jose, CA

<sup>3</sup>Samsung Electronics, Suwon, Korea

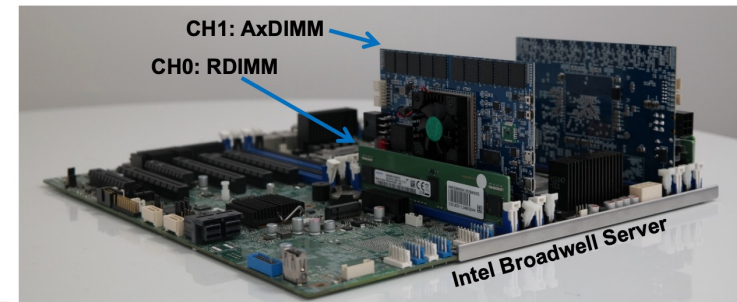
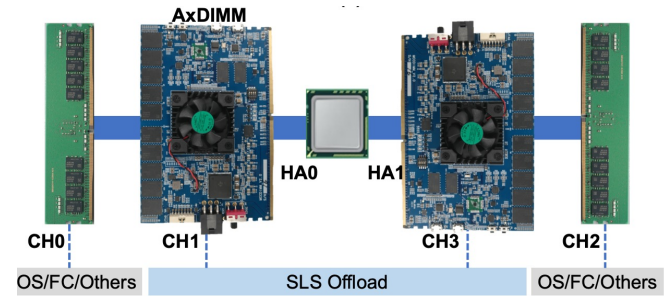
# Samsung AxDIMM (2021)

- DDRx-PIM

Baseline system

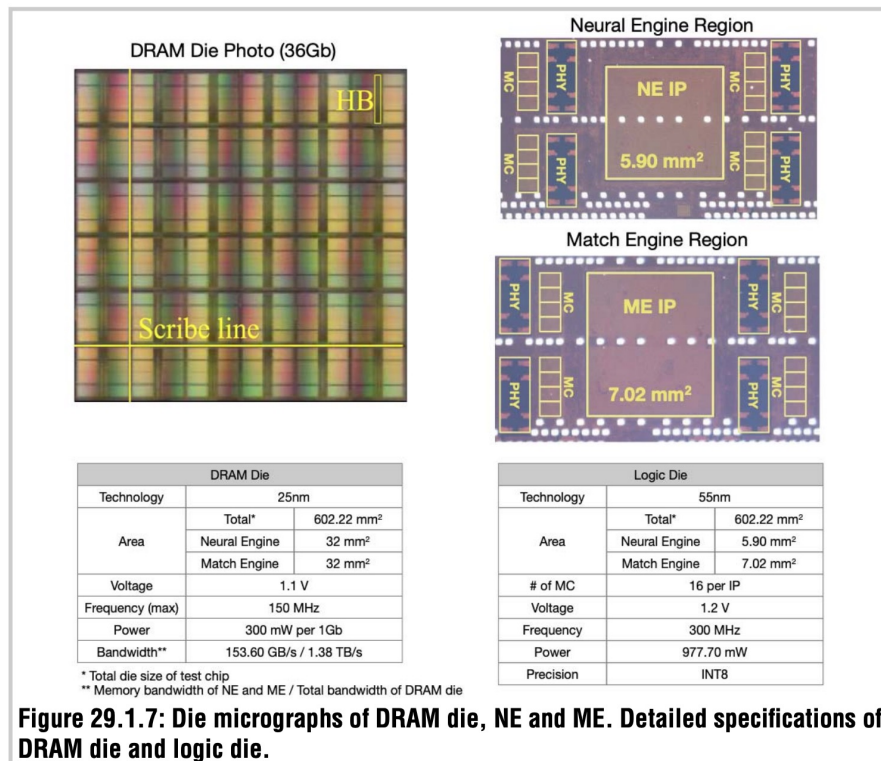
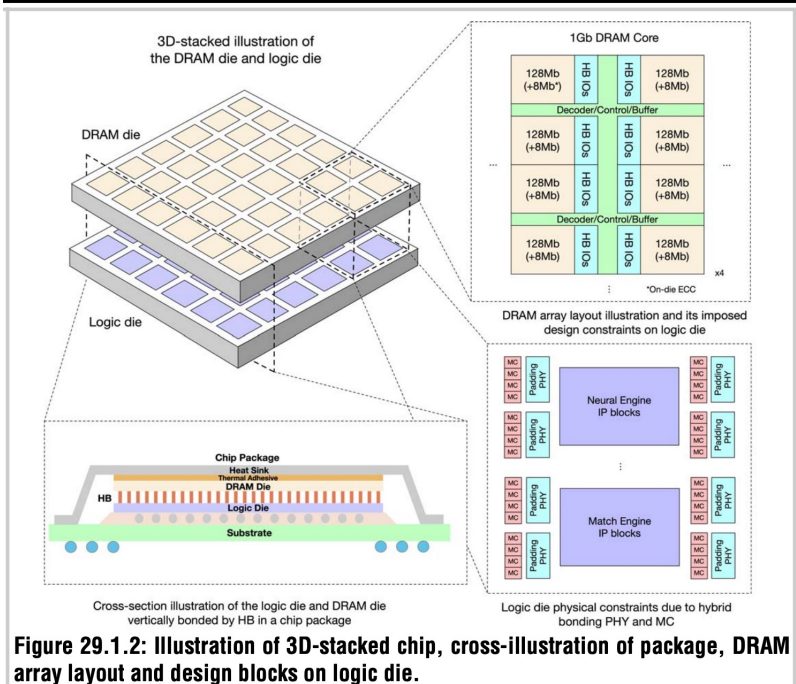


## AxDIMM System



# AliBaba PIM Recommendation System (2022)

ISSCC 2022 / February 24, 2022 / 8:30 AM



## 29.1 184QPS/W 64Mb/mm<sup>2</sup> 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu<sup>1</sup>, Shuangchen Li<sup>1</sup>, Yuhao Wang<sup>1</sup>, Wei Han<sup>1</sup>, Zhe Zhang<sup>2</sup>, Yijin Guan<sup>2</sup>, Tianchan Guan<sup>3</sup>, Fei Sun<sup>1</sup>, Fei Xue<sup>1</sup>, Lide Duan<sup>1</sup>, Yuanwei Fang<sup>1</sup>, Hongzhong Zheng<sup>1</sup>, Xiping Jiang<sup>4</sup>, Song Wang<sup>4</sup>, Fengguo Zuo<sup>4</sup>, Yubing Wang<sup>4</sup>, Bing Yu<sup>4</sup>, Qiwei Ren<sup>4</sup>, Yuan Xie<sup>1</sup>

# Recall: Takeaways

---

- It all starts from the basic building blocks and design principles
- And, knowledge of how to use, apply, enhance them
- Underlying technology might change (e.g., steel vs. wood)
  - but methods of taking advantage of technology bear resemblance
  - methods used for design depend on the principles employed

# Basic Building Blocks

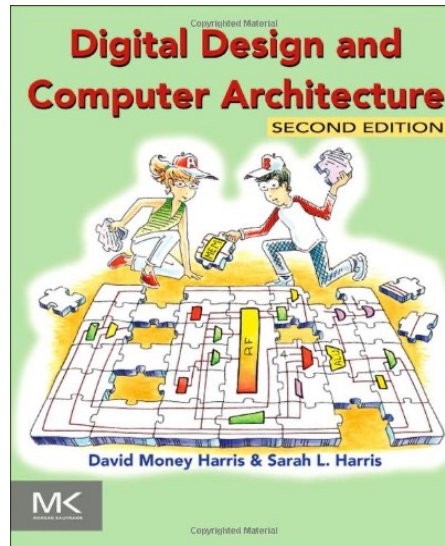
---

- Electrons
- Transistors
- Logic Gates
- Combinational Logic Circuits
- Sequential Logic Circuits
  - Storage Elements and Memory
- ...
- Cores
- Caches
- Interconnect
- Memories
- ...

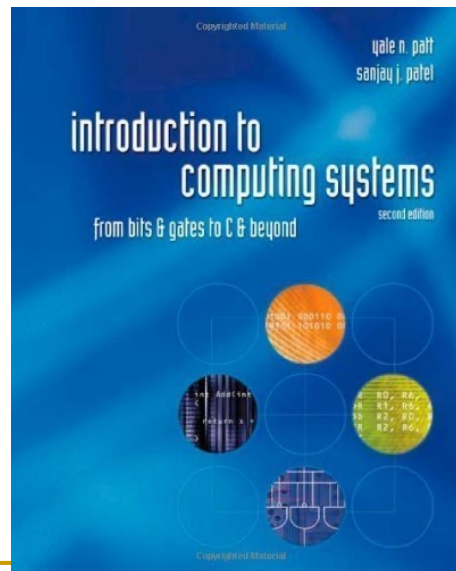
# Reading Assignments for This Week

---

- Chapter 1 in Harris & Harris



- Chapters 1-2 in Patt and Patel



- Supplementary Lecture Slides on Binary Numbers

# Recall: High-Level Goals of This Course

---

- In Digital Design & Computer Architecture
  - Understand the basics
  - Understand the principles (of design)
  - Understand the precedents
  - Based on such understanding:
    - learn how a modern computer works underneath
    - evaluate tradeoffs of different designs and ideas
    - implement a principled design (a simple microprocessor)
    - learn to systematically debug increasingly complex systems
    - Hopefully enable you to develop novel, out-of-the-box designs
  - The focus is on basics, principles, precedents, and how to use them to create/implement good designs
-



# Recall: Why These Goals?

---

- Because you are here for a Computer Science degree
- **Regardless of your future direction**, learning the principles of digital design & computer architecture will be useful to
  - design better hardware
  - design better software
  - design better systems
  - make better tradeoffs in design
  - understand why computers behave the way they do
  - solve problems better
  - think “in parallel”
  - think critically
  - ...

# Course Info and Logistics

# If You Need Help

---

- Post your question on Moodle Q&A Forum
  - <https://moodle-app2.let.ethz.ch/course/view.php?id=19395>
  - We will create a forum on Moodle for each activity
  - **Preferred** for **technical** questions
  
- Write an e-mail to:
  - [digitaltechnik@lists.inf.ethz.ch](mailto:digitaltechnik@lists.inf.ethz.ch)
  - The instructor and all assistants will receive this e-mail
  
- Come to office hours
  - We will provide office locations & Zoom links
  - TBD

# Where to Get Up-to-date Course Info?

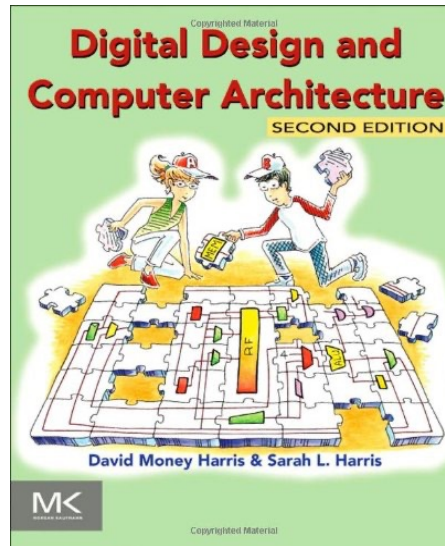
---

- Website:
  - ❑ <https://pooyanjamshidi.github.io/csce212/>
  - ❑ Lecture slides and (videos)
  - ❑ Readings
  - ❑ Course schedule, handouts, FAQs
  - ❑ Software
  - ❑ Any other useful information for the course
  - ❑ Check frequently for announcements and due dates
  - ❑ This is your single point of access to all resources
  
- TA

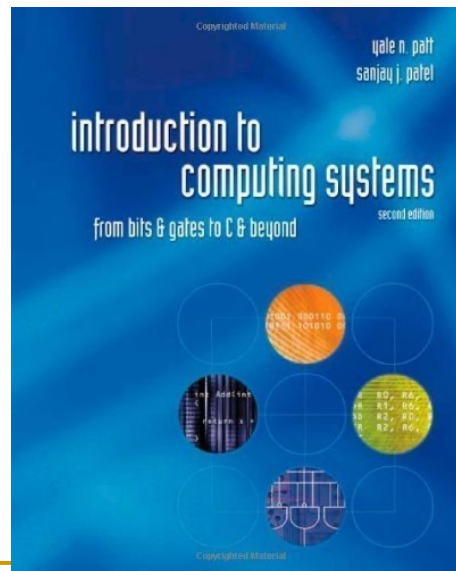
# Reading Assignments for This Week

---

- Chapter 1 in Harris & Harris



- Chapters 1-2 in Patt and Patel (encouraged)



# Reading Assignments for Next Week

---

- Combinational Logic chapters from both books
  - Harris and Harris, Chapter 2
  - Patt and Patel, Chapter 3
- Check the course website for all future readings
  - Required
  - Recommended
  - Mentioned

# Future Lectures and Assignments

---

- You can also anticipate (and plan for) future lectures and assignments based on Spring 2023 schedule:
  - <https://pooyanjamshidi.github.io/csce212/lectures/>





# Takeaways

---

- It is an exciting time to be understanding and designing computing architectures
- Many challenging and exciting problems in platform design
  - That no one has tackled (or thought about) before
  - That can have huge impact on the world's future
- Driven by huge hunger for data (Big Data), new applications (ML/AI, graph analytics, genomics), ever-greater realism, ...
  - We can easily collect more data than we can analyze/understand
- Driven by significant difficulties in keeping up with that hunger at the technology layer
  - Five walls: Energy, reliability, complexity, security, scalability

# Major High-Level Goals of This Course

---

## In Computer Architecture

- Understand the **basics**
- Understand the **principles (of design)**
- Understand the **precedents**
- Based on such understanding:
  - learn how a modern computer works underneath
  - evaluate tradeoffs of different designs and ideas
  - implement a principled design (a simple microprocessor)
  - **Hopefully enable you to** develop novel, out-of-the-box designs
- The focus is on basics, principles, precedents, and how to use them to create/implement good designs, tradeoffs are important!

# Why These Goals?

---

- Because you are here for a Computer Science degree
- **Regardless of your future direction**, learning the principles of computer architecture will be useful to
  - design better systems (software + hardware)
  - make better tradeoffs in design
  - understand why computers behave the way they do
  - solve problems better
  - think “in parallel”
  - think critically
  - ...

# I presume you all know the number systems?

---

- Binary Number
- Hexadecimal Numbers
- Bits, Bytes, Words
- least significant bit (lsb), most significant bit (msb)
- Least Significant Byte (LSB), Most Significant Byte (MSB)
- KB, MB, GB, TB
- Binary Addition
- Signed Binary Numbers