# Computer Architecture

## Lecture 1: Introduction and Basics

Prof. Onur Mutlu
ETH Zürich

Fall 2020

17 September 2020

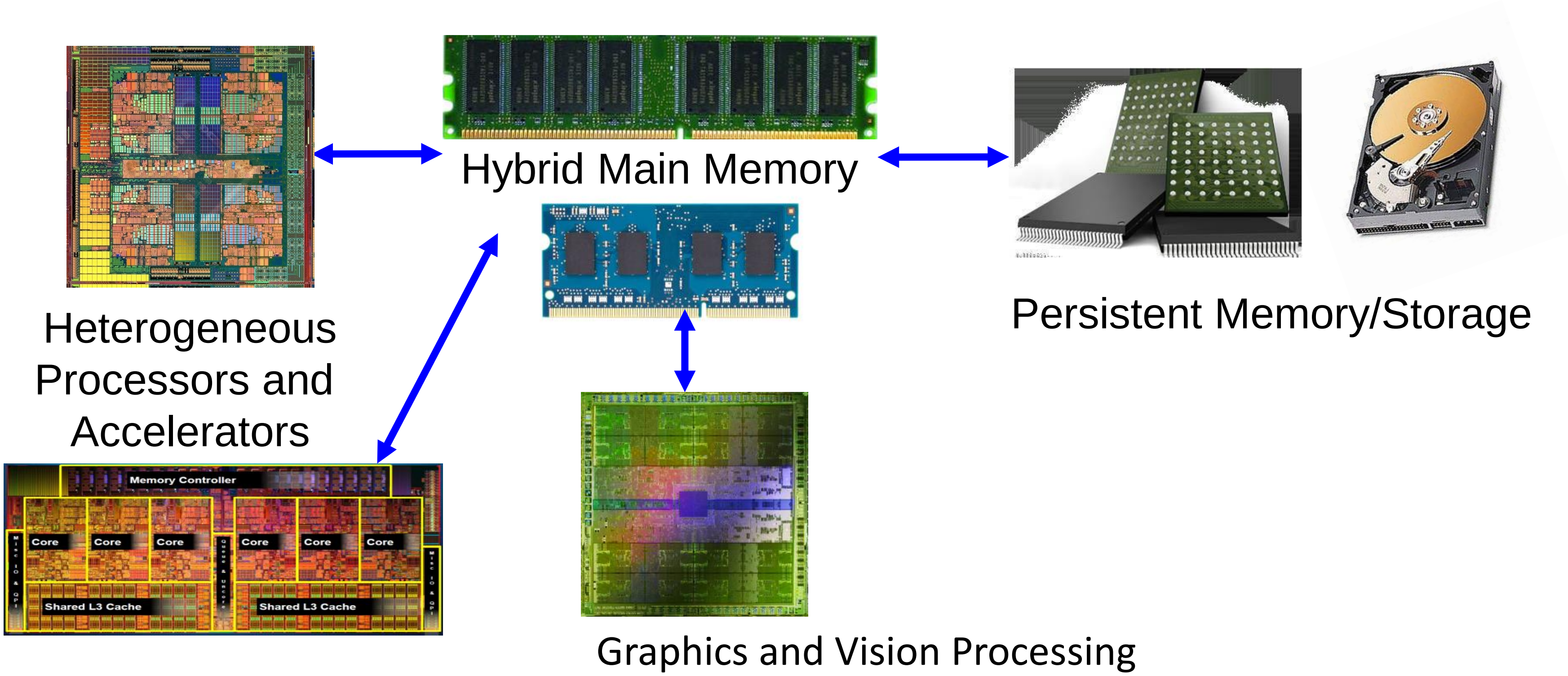
# Brief Self Introduction

- Onur Mutlu
  - Full Professor @ ETH Zurich ITET (INFK), since September 2015
  - Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-…
  - PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
  - https://people.inf.ethz.ch/omutlu/
  - omutlu@gmail.com (Best way to reach me)
  - https://people.inf.ethz.ch/omutlu/projects.htm

- Research and Teaching in:
  - Computer architecture, computer systems, hardware security, bioinformatics
  - Memory and storage systems
  - Hardware security, safety, predictability
  - Fault tolerance
  - Hardware/software cooperation
  - Architectures for bioinformatics, health, medicine
  - …

# Current Research Mission

***Computer architecture, HW/SW, systems, bioinformatics, security***




**Build fundamentally better architectures**

# Four Key Current Directions

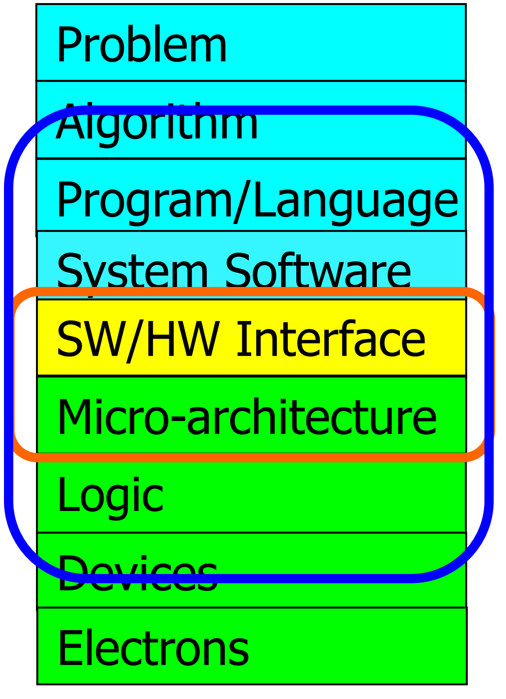
- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Architectures for AI/ML, Genomics, Medicine, Health

# The Transformation Hierarchy




**Computer Architecture (expanded view)**

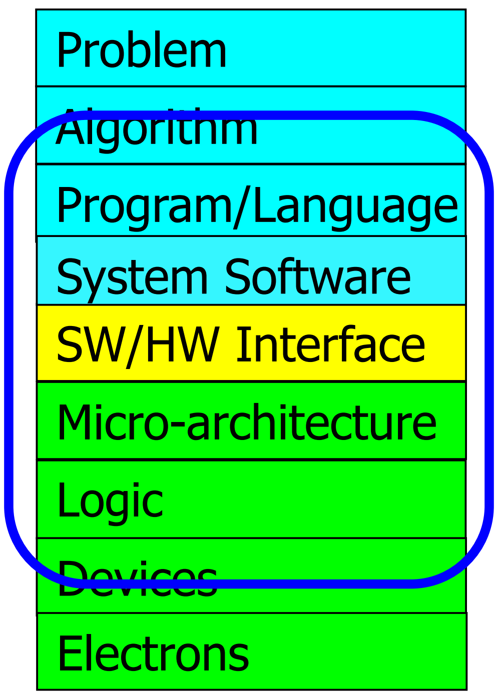
**Computer Architecture (narrow view)**

# Axiom

To achieve the highest energy efficiency and performance:

**we must take the expanded view**

of computer architecture

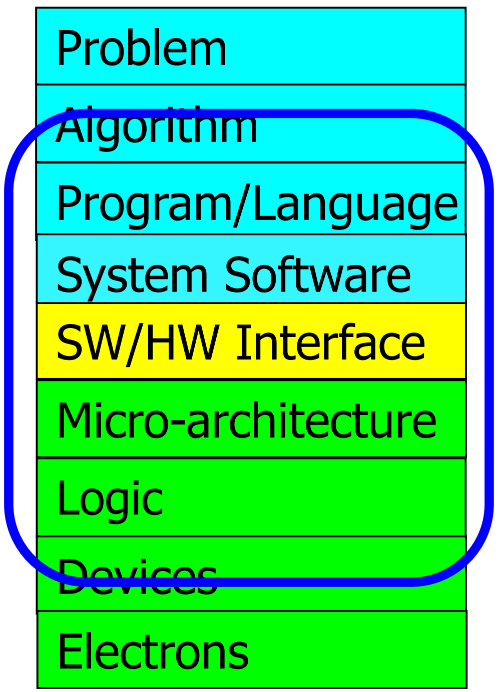



**Co-design across the hierarchy: Algorithms to devices**

**Specialize as much as possible within the design goals**

# Current Research Mission & Major Topics

## Build fundamentally better architectures




**Broad research spanning apps, systems, logic with architecture at the center**

- Data-centric arch. for low energy & high perf.
  - Proc. in Mem/DRAM, NVM, unified mem/storage
- Low-latency & predictable architectures
  - Low-latency, low-energy yet low-cost memory
  - QoS-aware and predictable memory systems
- Fundamentally secure/reliable/safe arch.
  - Tolerating all bit flips; patchable HW; secure mem
- Architectures for ML/AI/Genomics/Graph/Med
  - Algorithm/arch./logic co-design; full heterogeneity
- Data-driven and data-aware architectures
  - ML/AI-driven architectural controllers and design
  - Expressive memory and expressive systems

# Onur Mutlu's SAFARI Research Group

***Computer architecture, HW/SW, systems, bioinformatics, security, memory***

https://safari.ethz.ch/safari-newsletter-april-2020/


# Think BIG, Aim HIGH!

https://safari.ethz.ch

# Research & Teaching: Some Overview Talks

https://www.youtube.com/onurmutlulectures

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kgiZlSOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=1
- Enabling In-Memory Computation
  - https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=16
- Accelerating Genome Analysis
  - https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=9
- Rethinking Memory System Design
  - https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3
- Intelligent Architectures for Intelligent Machines
  - https://www.youtube.com/watch?v=n8Aj_A0WSg8&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=22

# An Interview on Research and Education

- Computing Research and Education (@ ISCA 2019)
  - https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz

- Maurice Wilkes Award Speech (10 minutes)
  - https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=15

# More Thoughts and Suggestions

- Onur Mutlu,
  **"Some Reflections (on DRAM)"**
  *Award Speech for ACM SIGARCH Maurice Wilkes Award, at the **ISCA** Awards Ceremony*, Phoenix, AZ, USA, 25 June 2019.
  [Slides (pptx) (pdf)]
  [Video of Award Acceptance Speech (Youtube; 10 minutes) (Youku; 13 minutes)]
  [Video of Interview after Award Acceptance (Youtube; 1 hour 6 minutes) (Youku; 1 hour 6 minutes)]
  [News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"]

- Onur Mutlu,
  **"How to Build an Impactful Research Group"**
  *57th Design Automation Conference Early Career Workshop (**DAC**)*, Virtual, 19 July 2020.
  [Slides (pptx) (pdf)]

# Why Study Computer Architecture?

# Computer Architecture

- is the science and art of designing computing platforms (hardware, interface, system SW, and programming model)

- to achieve a set of design goals
  - E.g., highest performance on earth on workloads X, Y, Z
  - E.g., longest battery life at a form factor that fits in your pocket with cost < $$$ CHF
  - E.g., best average performance across all known workloads at the best performance/cost ratio
  - …

  - Designing a supercomputer is different from designing a smartphone → But, many fundamental principles are similar

# Different Platforms, Different Goals


Source: http://www.sia-online.org (semiconductor industry association)

# Different Platforms, Different Goals



Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/

# Different Platforms, Different Goals

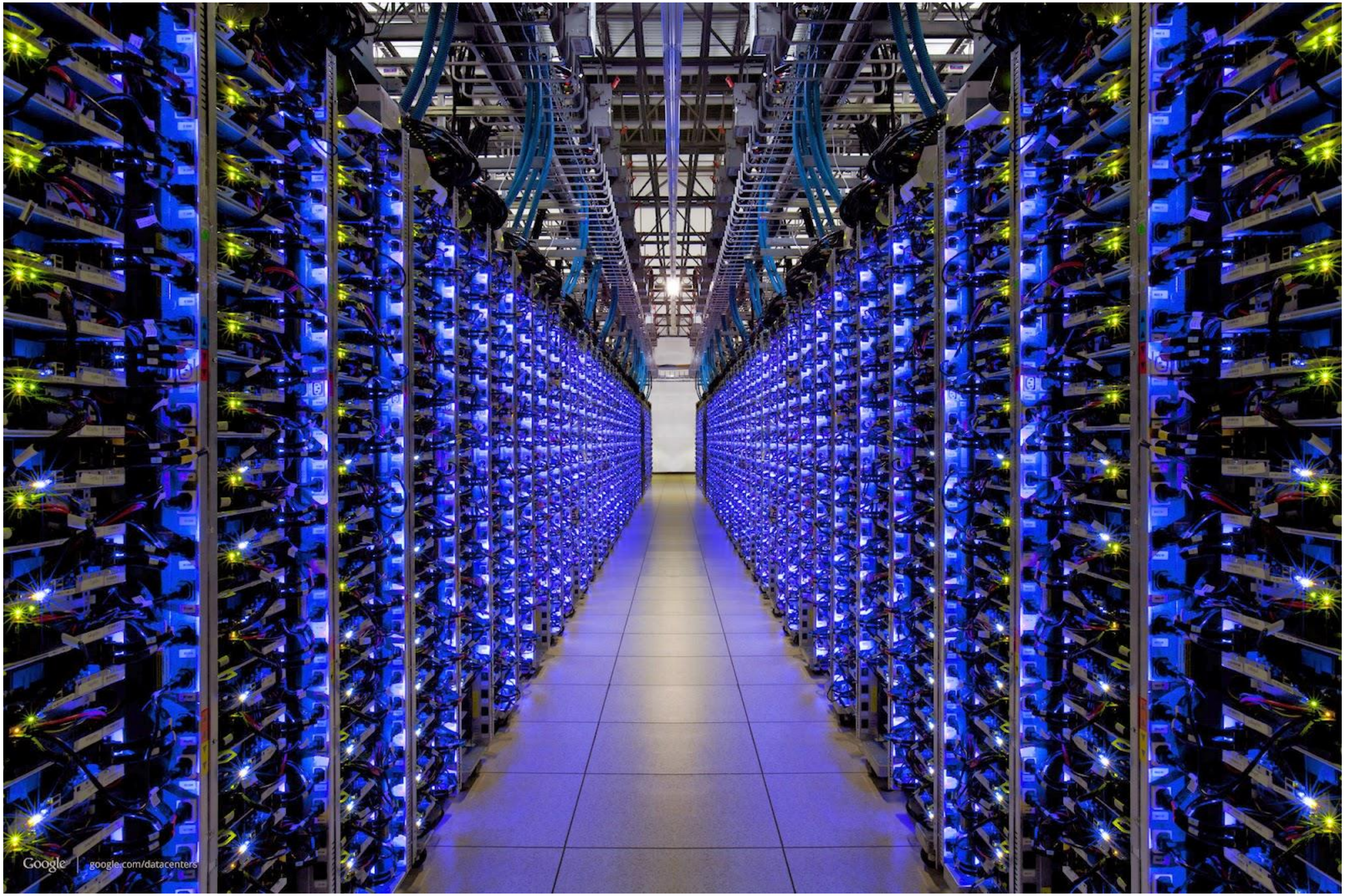

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

# Different Platforms, Different Goals

SAFARI

Source: http://sm.pcmag.com/pcmag_uk/photo/g/google-self-driving-car-the-guts/google-self-driving-car-the-guts_dwx8.jpg

# Different Platforms, Different Goals






Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg

# Different Platforms, Different Goals




Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg

# Different Platforms, Different Goals

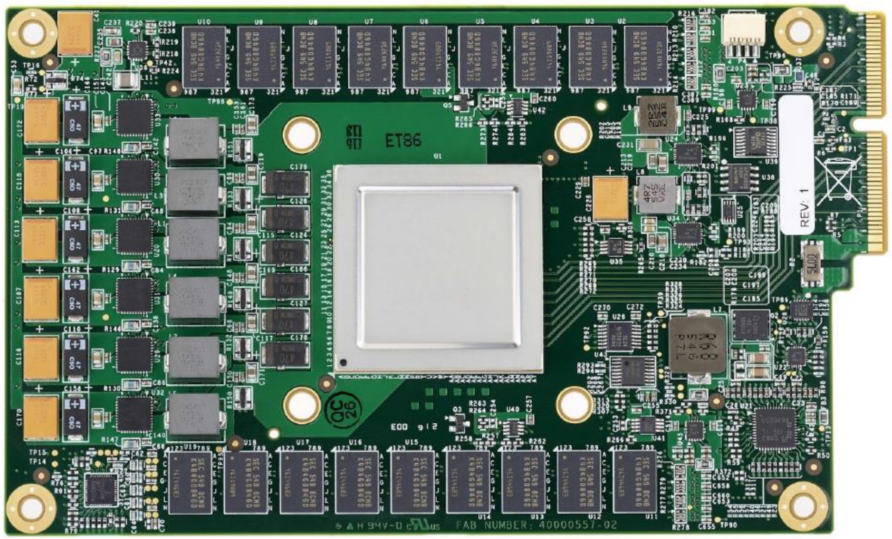


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

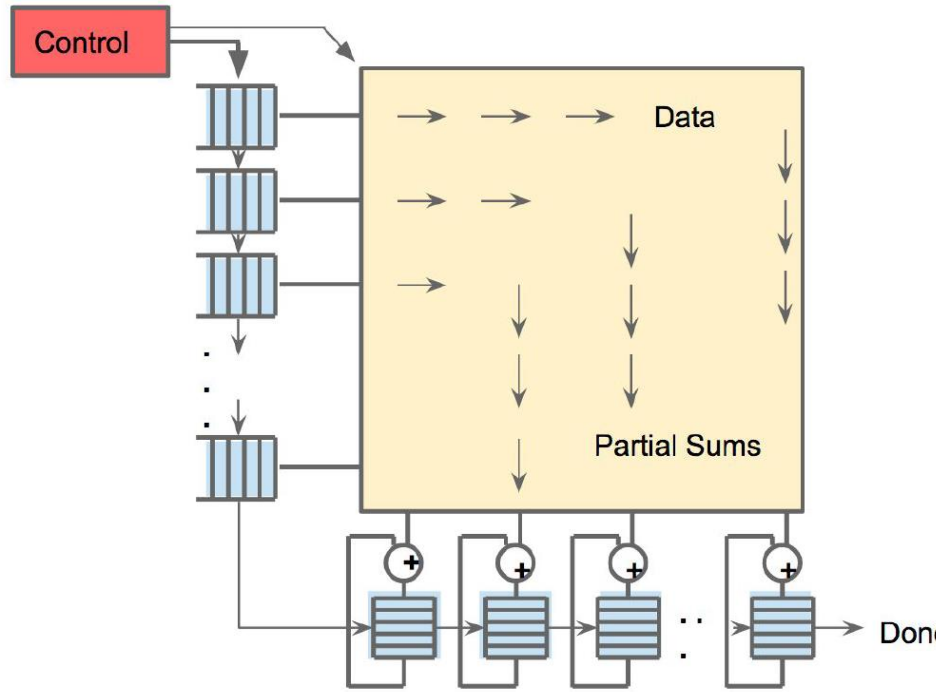
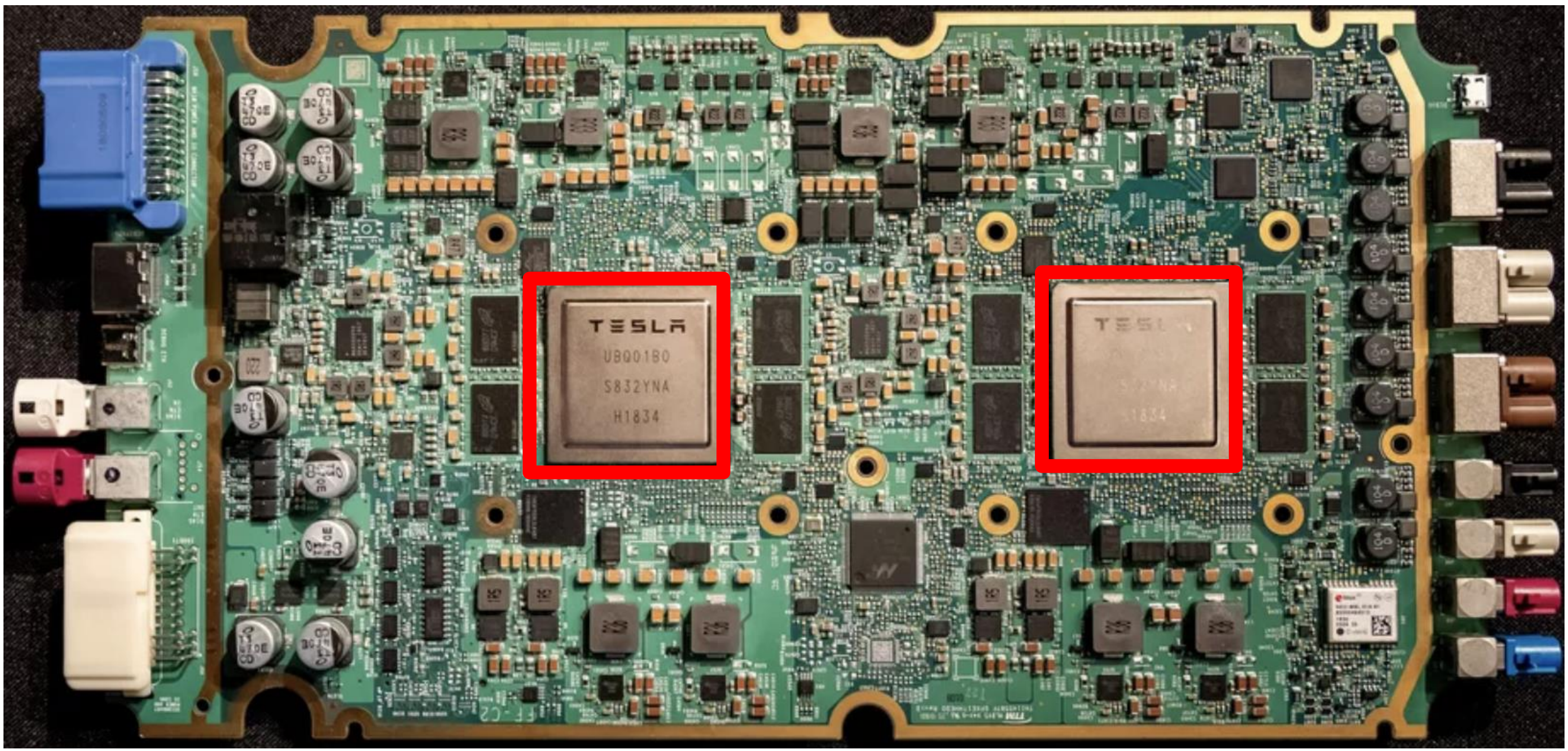



**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

# Different Platforms, Different Goals

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



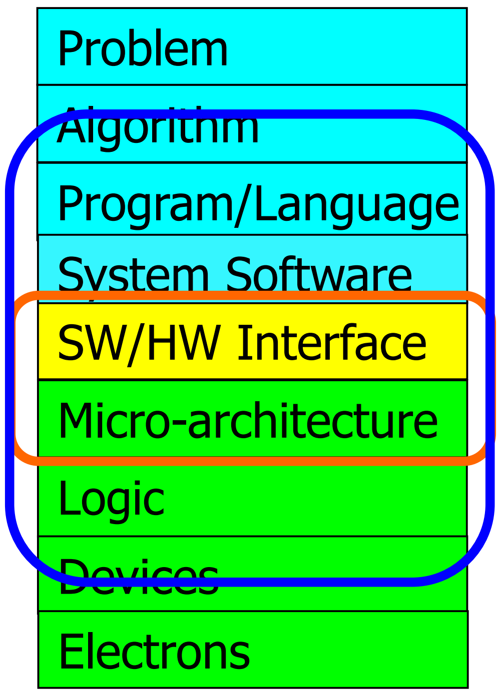

https://youtu.be/Ucp0TTmvqOE?t=4236

# What is Computer Architecture?

- The science and art of designing, selecting, and interconnecting hardware components and designing the hardware/software interface to create a computing system that meets functional, performance, energy consumption, cost, and other specific goals.

# The Transformation Hierarchy

Computer Architecture (expanded view)




Computer Architecture (narrow view)

# Why Study Computer Architecture?

- Enable better systems: make computers faster, cheaper, smaller, more reliable, …
  - By exploiting advances and changes in underlying technology/circuits

- Enable new applications
  - Life-like 3D visualization 20 years ago? Virtual reality?
  - Self-driving cars?
  - Personalized genomics? Personalized medicine?

- Enable better solutions to problems
  - Software innovation is built on trends and changes in computer architecture
    - > 50% performance improvement per year has enabled this innovation

- Understand why computers work the way they do

# Computer Architecture Today (I)

- Today is a very exciting time to study computer architecture

- Industry is in a large paradigm shift (to novel architectures) – many different potential system designs possible

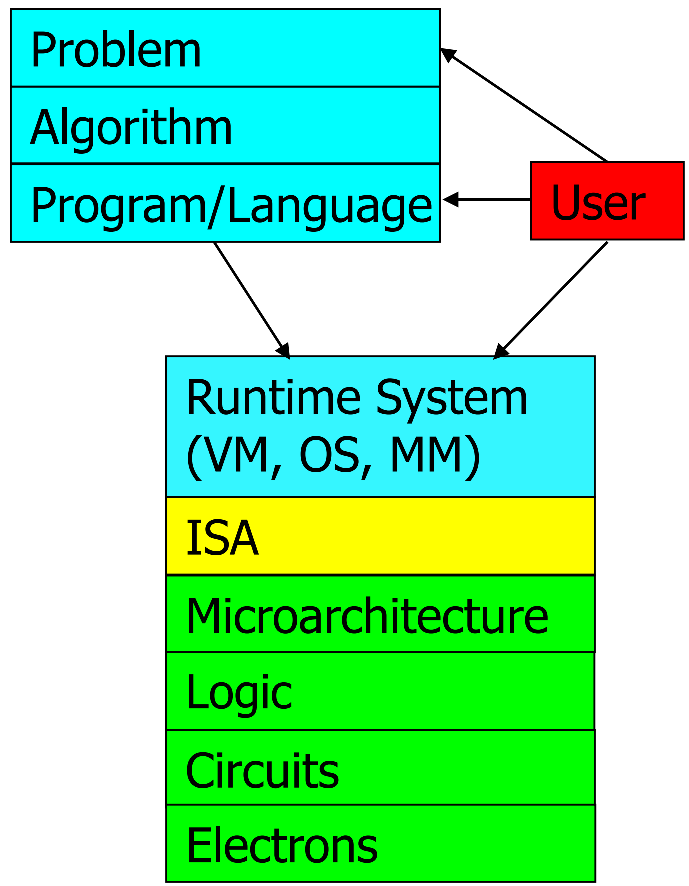
- Many difficult problems *motivating* and *caused by* the shift
  - Huge hunger for data and new data-intensive applications
  - Power/energy/thermal constraints
  - Complexity of design
  - Difficulties in technology scaling
  - Memory wall/gap
  - Reliability problems
  - Programmability problems
  - Security and privacy issues

- No clear, definitive answers to these problems

# Computer Architecture Today (II)

- These problems affect all parts of the computing stack – if we do not change the way we design systems

Many new demands from the top (Look Up)



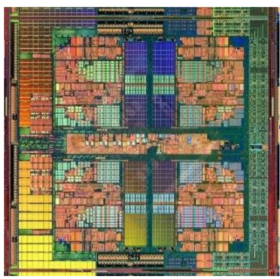

Fast changing demands and personalities of users (Look Up)

Many new issues at the bottom (Look Down)

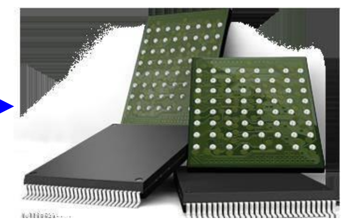
- No clear, definitive answers to these problems

# Computer Architecture Today (III)

- Computing landscape is very different from 10-20 years ago
- Both UP (software and humanity trends) and DOWN (technologies and their issues), FORWARD and BACKWARD, and the resulting requirements and constraints

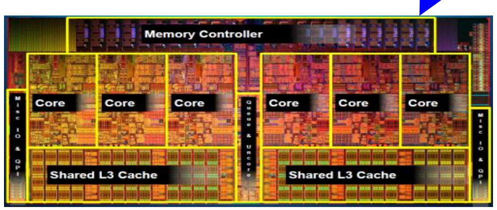


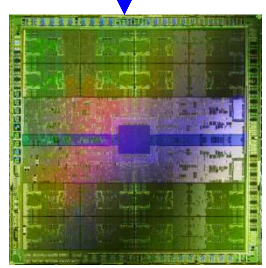
Hybrid Main Memory



Persistent Memory/Storage

Heterogeneous Processors and Accelerators



General Purpose GPUs

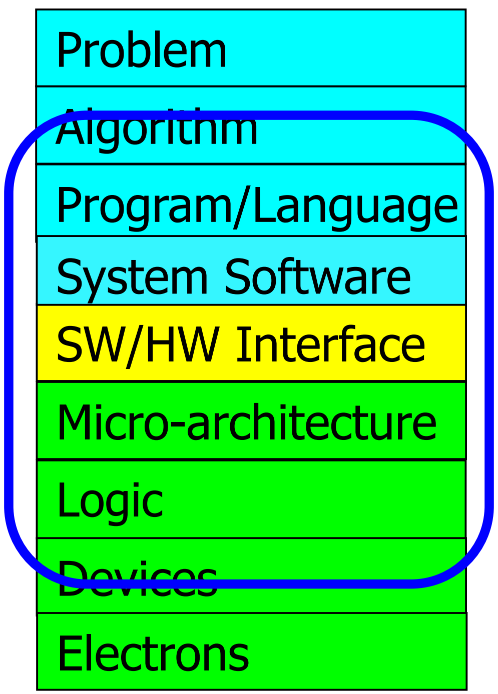
**Every component and its interfaces, as well as entire system designs are being re-examined**

# Axiom

To achieve the highest energy efficiency and performance:

**we must take the expanded view**

of computer architecture




**Co-design across the hierarchy: Algorithms to devices**

**Specialize as much as possible within the design goals**

# Historical: Opportunities at the Bottom

## There's Plenty of Room at the Bottom

From Wikipedia, the free encyclopedia

"**There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics**" was a lecture given by physicist Richard Feynman at the annual American Physical Society meeting at Caltech on December 29, 1959.[1] Feynman considered the possibility of direct manipulation of individual atoms as a more powerful form of synthetic chemistry than those used at the time. Although versions of the talk were reprinted in a few popular magazines, it went largely unnoticed and did not inspire the conceptual beginnings of the field. Beginning in the 1980s, nanotechnology advocates cited it to establish the scientific credibility of their work.

https://en.wikipedia.org/wiki/There%27s_Plenty_of_Room_at_the_Bottom

# Historical: Opportunities at the Top

REVIEW

## There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson[1], Neil C. Thompson[1,2,*], Joel S. Emer[1,3], Bradley C. Kuszmaul[1,†], Butler W. Lampson[1,4], ...
+ See all authors and affiliations

*Science* 05 Jun 2020:
Vol. 368, Issue 6495, eaam9744
DOI: 10.1126/science.aam9744

Much of the improvement in computer performance comes from decades of miniaturization of computer components, a trend that was foreseen by the Nobel Prize–winning physicist Richard Feynman in his 1959 address, "There's Plenty of Room at the Bottom," to the American Physical Society. In 1975, Intel founder Gordon Moore predicted the regularity of this miniaturization trend, now called Moore's law, which, until recently, doubled the number of transistors on computer chips every 2 years.

Unfortunately, semiconductor miniaturization is running out of steam as a viable way to grow computer performance—there isn't much more room at the "Bottom." If growth in computing power stalls, practically all industries will face challenges to their productivity. Nevertheless, opportunities for growth in computing performance will still be available, especially at the "Top" of the computing-technology stack: software, algorithms, and hardware architecture.

https://science.sciencemag.org/content/368/6495/eaam9744

# Axiom, Revisited

There is plenty of room both at the top and at the bottom

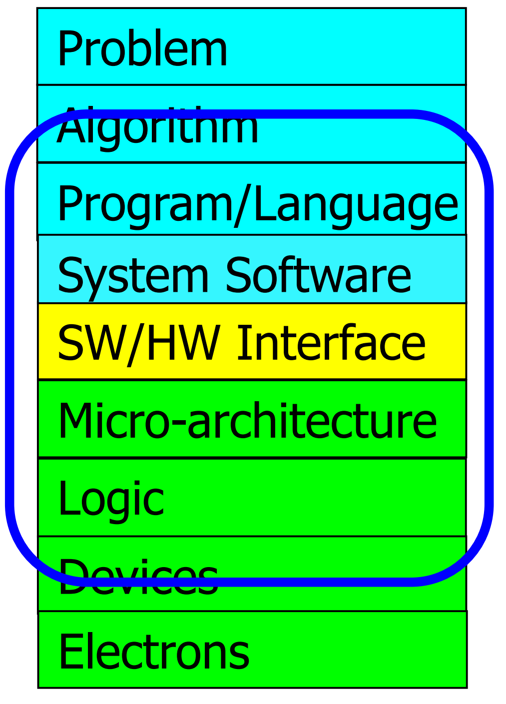
but much more so

when you communicate well between and optimize across

the top and the bottom.

# Hence the Expanded View

Computer Architecture
(expanded view)



# Some Cross-Layer Design Examples (Foreshadowing)

# Expressive (Memory) Interfaces

- Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu,
**"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"**
*Proceedings of the 45th International Symposium on Computer Architecture* (***ISCA***), Los Angeles, CA, USA, June 2018.
[Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video]

**A Case for Richer Cross-layer Abstractions:
Bridging the Semantic Gap with Expressive Memory**

Nandita Vijaykumar[†§] Abhilasha Jain[†] Diptesh Majumdar[†] Kevin Hsieh[†] Gennady Pekhimenko[‡]
Eiman Ebrahimi[ℵ] Nastaran Hajinazar[+] Phillip B. Gibbons[†] Onur Mutlu[§†]

[†]**Carnegie Mellon University** [‡]**University of Toronto** [ℵ]**NVIDIA**
[+]**Simon Fraser University** [§]**ETH Zürich**

# X-MeM Aids Many Optimizations

**Table 1: Summary of the example memory optimizations that XMem aids.**

| Memory optimization | Example semantics provided by XMem (described in §3.3) | Example Benefits of XMem |
|---|---|---|
| Cache management | *(i)* Distinguishing between data structures or pools of similar data; *(ii)* Working set size; *(iii)* Data reuse | Enables: *(i)* applying different caching policies to different data structures or pools of data; *(ii)* avoiding cache thrashing by *knowing* the active working set size; *(iii)* bypassing/prioritizing data that has no/high reuse. (§5) |
| Page placement in DRAM e.g., [23, 24] | *(i)* Distinguishing between data structures; *(ii)* Access pattern; *(iii)* Access intensity | Enables page placement at the *data structure* granularity to *(i)* isolate data structures that have high row buffer locality and *(ii)* spread out concurrently-accessed irregular data structures across banks and channels to improve parallelism. (§6) |
| Cache/memory compression e.g., [25–32] | *(i)* Data type: integer, float, char; *(ii)* Data properties: sparse, pointer, data index | Enables using a *different compression algorithm* for each data structure based on data type and data properties, e.g., sparse data encodings, FP-specific compression, delta-based compression for pointers [27]. |
| Data prefetching e.g., [33–36] | *(i)* Access pattern: strided, irregular, irregular but repeated (e.g., graphs), access stride; *(ii)* Data type: index, pointer | Enables *(i) highly accurate* software-driven prefetching while leveraging the benefits of hardware prefetching (e.g., by being memory bandwidth-aware, avoiding cache thrashing); *(ii)* using different prefetcher *types* for different data structures: e.g., stride [33], tile-based [20], pattern-based [34–37], data-based for indices/pointers [38, 39], etc. |
| DRAM cache management e.g., [40–46] | *(i)* Access intensity; *(ii)* Data reuse; *(iii)* Working set size | *(i)* Helps avoid cache thrashing by knowing working set size [44]; *(ii)* Better DRAM cache management via reuse behavior and access intensity information. |
| Approximation in memory e.g., [47–53] | *(i)* Distinguishing between pools of similar data; *(ii)* Data properties: tolerance towards approximation | Enables *(i)* each memory component to track how approximable data is (at a fine granularity) to inform approximation techniques; *(ii)* data placement in heterogeneous reliability memories [54]. |
| Data placement: NUMA systems e.g., [55, 56] | *(i)* Data partitioning across threads (i.e., relating data to threads that access it); *(ii)* Read-Write properties | Reduces the need for profiling or data migration *(i)* to co-locate data with threads that access it and *(ii)* to identify Read-Only data, thereby enabling techniques such as replication. |
| Data placement: hybrid memories e.g., [16, 57, 58] | *(i)* Read-Write properties (Read-Only/Read-Write); *(ii)* Access intensity; *(iii)* Data structure size; *(iv)* Access pattern | Avoids the need for profiling/migration of data in hybrid memories to *(i)* effectively manage the asymmetric read-write properties in NVM (e.g., placing Read-Only data in the NVM) [16, 57]; *(ii)* make tradeoffs between data structure "hotness" and size to allocate fast/high bandwidth memory [14]; and *(iii)* leverage row-buffer locality in placement based on access pattern [45]. |
| Managing NUCA systems e.g., [15, 59] | *(i)* Distinguishing pools of similar data; *(ii)* Access intensity; *(iii)* Read-Write or Private-Shared properties | *(i)* Enables using different cache policies for different data pools (similar to [15]); *(ii)* Reduces the need for reactive mechanisms that detect sharing and read-write characteristics to inform cache policies. |

# Expressive (Memory) Interfaces for GPUs

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu,
**"The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"**
*Proceedings of the 45th International Symposium on Computer Architecture* (***ISCA***), Los Angeles, CA, USA, June 2018.
[Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video]

**The Locality Descriptor:**
**A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs**

Nandita Vijaykumar[†§] Eiman Ebrahimi[‡] Kevin Hsieh[†]
Phillip B. Gibbons[†] Onur Mutlu[§†]

[†]**Carnegie Mellon University** [‡]**NVIDIA** [§]**ETH Zürich**

# Heterogeneous-Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu, **"Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"** *Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Atlanta, GA, June 2014. [Summary] [Slides (pptx) (pdf)] [Coverage on ZDNet]

**Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory**

Yixin Luo Sriram Govindan* Bikash Sharma* Mark Santaniello* Justin Meza
Aman Kansal* Jie Liu* Badriddine Khessib* Kushagra Vaid* Onur Mutlu
Carnegie Mellon University, yixinluo@cs.cmu.edu, {meza, onur}@cmu.edu
*Microsoft Corporation, {srgovin, bsharma, marksan, kansal, jie.liu, bkhessib, kvaid}@microsoft.com

# EDEN: Data-Aware Efficient DNN Inference

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,
  **"EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM"**
  *Proceedings of the 52nd International Symposium on Microarchitecture (**MICRO**)*, Columbus, OH, USA, October 2019.
  [Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Poster (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]
  [Full Talk Lecture (38 minutes)]

## EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM

Skanda Koppula Lois Orosa A. Giray Yağlıkçı
Roknoddin Azizi Taha Shahroodi Konstantinos Kanellopoulos Onur Mutlu
ETH Zürich

# SMASH: SW/HW Indexing Acceleration

- Konstantinos Kanellopoulos, Nandita Vijaykumar, Christina Giannoula, Roknoddin Azizi, Skanda Koppula, Nika Mansouri Ghiasi, Taha Shahroodi, Juan Gomez-Luna, and Onur Mutlu,
  **"SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations"**
  *Proceedings of the 52nd International Symposium on Microarchitecture (**MICRO**)*, Columbus, OH, USA, October 2019.
  [Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Poster (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]
  [Full Talk Lecture (30 minutes)]

**SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations**

Konstantinos Kanellopoulos[1] Nandita Vijaykumar[2,1] Christina Giannoula[1,3] Roknoddin Azizi[1]
Skanda Koppula[1] Nika Mansouri Ghiasi[1] Taha Shahroodi[1] Juan Gomez Luna[1] Onur Mutlu[1,2]
[1]ETH Zürich [2]Carnegie Mellon University [3]National Technical University of Athens

# Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,
  **"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"**
  *Proceedings of the 47th International Symposium on Computer Architecture* (***ISCA***), Valencia, Spain, June 2020.
  [Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [ARM Research Summit Poster (pptx) (pdf)]
  [Talk Video (26 minutes)]
  [Lightning Talk Video (3 minutes)]

**The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework**

Nastaran Hajinazar[⋆†] Pratyush Patel[⋈] Minesh Patel[⋆] Konstantinos Kanellopoulos[⋆] Saugata Ghose[‡]
Rachata Ausavarungnirun[⊙] Geraldo F. Oliveira[⋆] Jonathan Appavoo[◇] Vivek Seshadri[▽] Onur Mutlu[⋆‡]

*[⋆]ETH Zürich [†]Simon Fraser University [⋈]University of Washington [‡]Carnegie Mellon University*
*[⊙]King Mongkut's University of Technology North Bangkok [◇]Boston University [▽]Microsoft Research India*

# Many Interesting Things Are Happening Today in Computer Architecture

# Many Interesting Things Are Happening Today in Computer Architecture

**Performance**
**and**
**Energy Efficiency**

# Intel Optane Persistent Memory (2019)

- Non-volatile main memory
- Based on 3D-XPoint Technology

# PCM as Main Memory: Idea in 2009

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger, **"Architecting Phase Change Memory as a Scalable DRAM Alternative"**
  *Proceedings of the 36th International Symposium on Computer Architecture* (***ISCA***), pages 2-13, Austin, TX, June 2009. Slides (pdf)

## Architecting Phase Change Memory as a Scalable DRAM Alternative


Benjamin C. Lee† Engin Ipek† Onur Mutlu‡ Doug Burger†

†Computer Architecture Group
Microsoft Research
Redmond, WA
{blee, ipek, dburger}@microsoft.com

‡Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

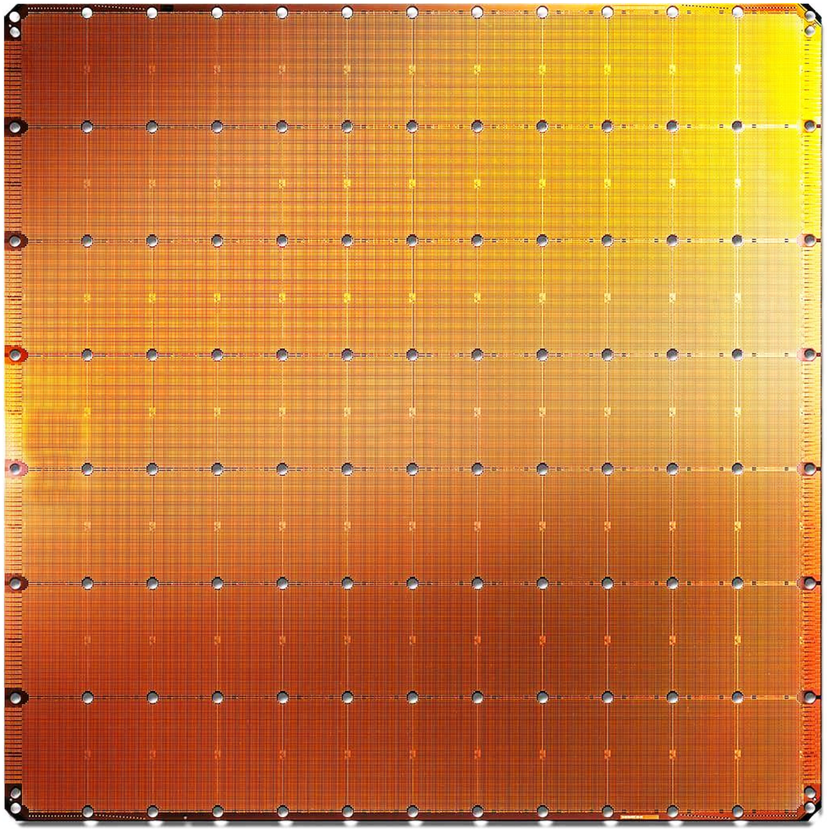
# PCM as Main Memory: Idea in 2009

- Benjamin C. Lee, Ping Zhou, Jun Yang, Youtao Zhang, Bo Zhao, Engin Ipek, Onur Mutlu, and Doug Burger, **"Phase Change Technology and the Future of Main Memory"** *IEEE Micro, Special Issue: Micro's Top Picks from 2009 Computer Architecture Conferences (**MICRO TOP PICKS**)*, Vol. 30, No. 1, pages 60-70, January/February 2010.

PHASE-CHANGE TECHNOLOGY AND THE FUTURE OF MAIN MEMORY

# Cerebras's Wafer Scale Engine (2019)

- The largest ML accelerator chip
- 400,000 cores



**Cerebras WSE**
1.2 Trillion transistors
46,225 mm²

**Largest GPU**
21.1 Billion transistors
815 mm²

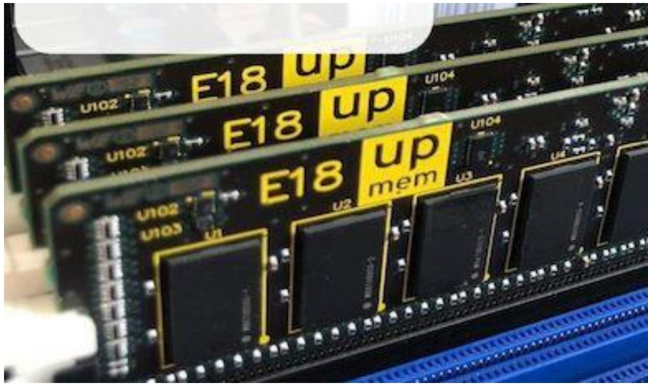
**NVIDIA** TITAN V

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

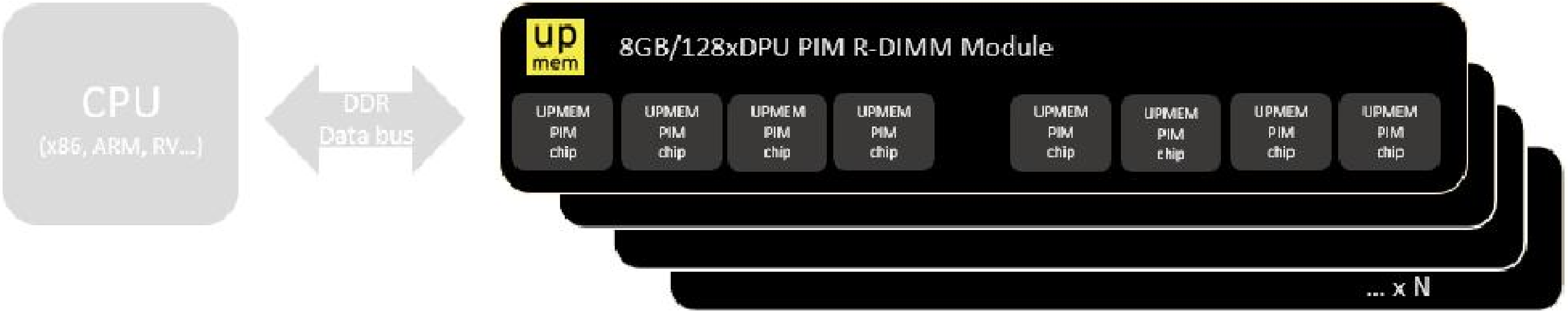
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

# 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.



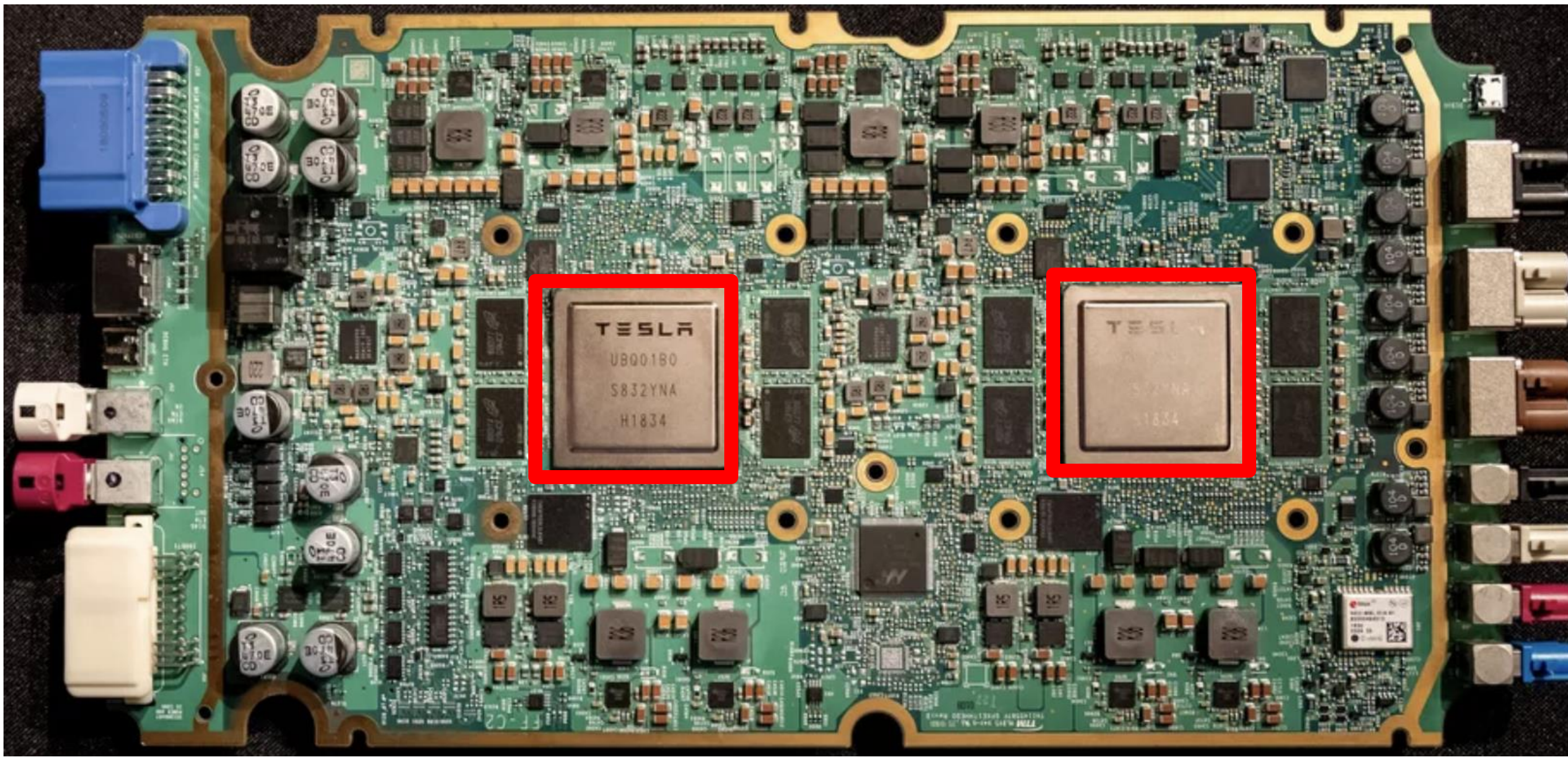

- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth




https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

# TESLA Full Self-Driving Computer (2019)

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




https://youtu.be/Ucp0TTmvqOE?t=4236

# Google TPU Generation I (~2016)

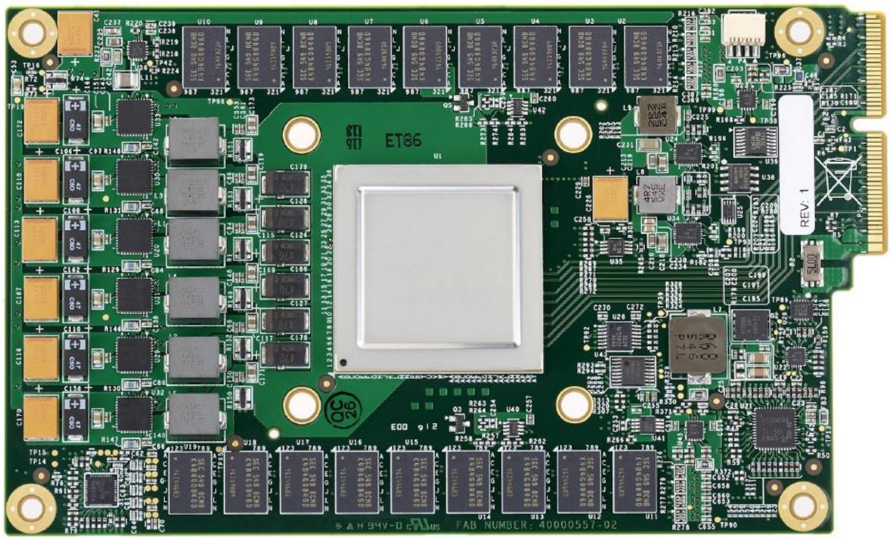


**Figure 3.** TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

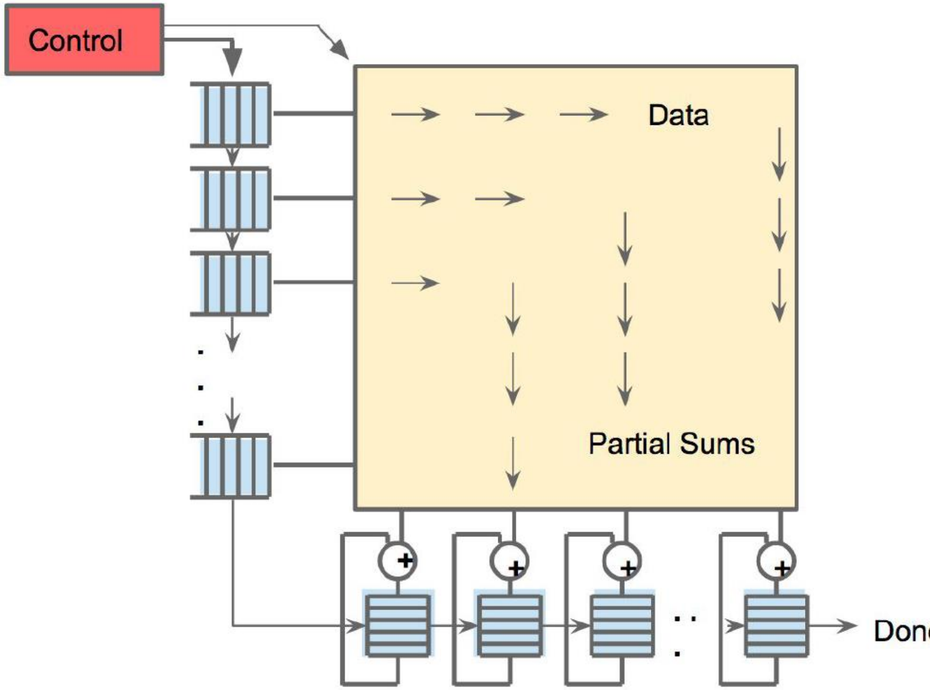



**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

# Google TPU Generation II (2017)

https://www.nextplatform.com/2017/05/17/first-depth-look-googles-new-second-generation-tpu/

4 TPU chips
vs 1 chip in TPU1

High Bandwidth Memory
vs DDR3

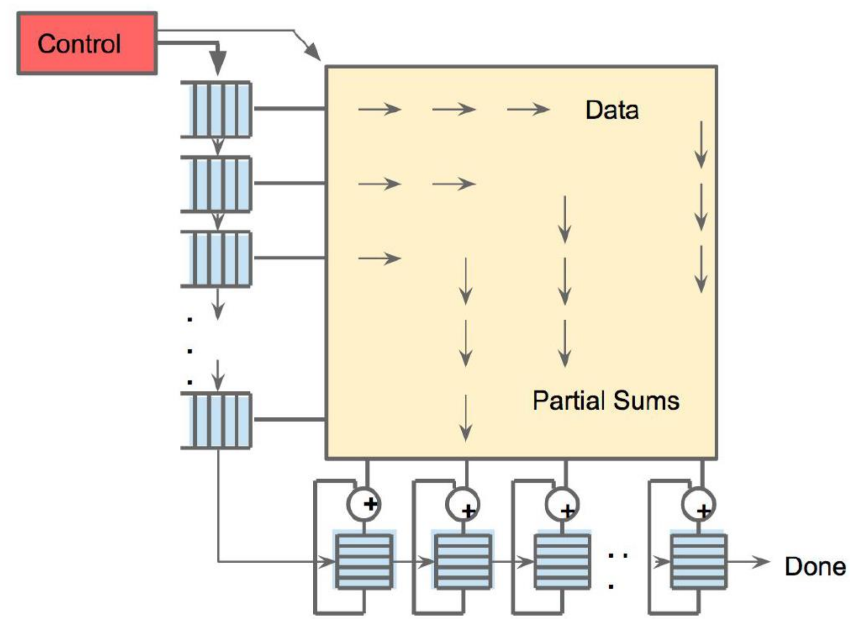
Floating point operations
vs FP16

45 TFLOPS per chip
vs 23 TOPS

Designed for training and inference
vs only inference

# An Example Modern Systolic Array: TPU (II)

As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the Unified Buffer [Kun80][Ram91][Ovt15b]. Figure 4 shows that data flows in from the left, and the weights are loaded from the top. A given 256-element multiply-accumulate operation moves through the matrix as a diagonal wavefront. The weights are preloaded, and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give the illusion that the 256 inputs are read at once, and that they instantly update one location of each of 256 accumulators. From a correctness perspective, software is unaware of the systolic nature of the matrix unit, but for performance, it does worry about the latency of the unit.




Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

# An Example Modern Systolic Array: TPU (III)

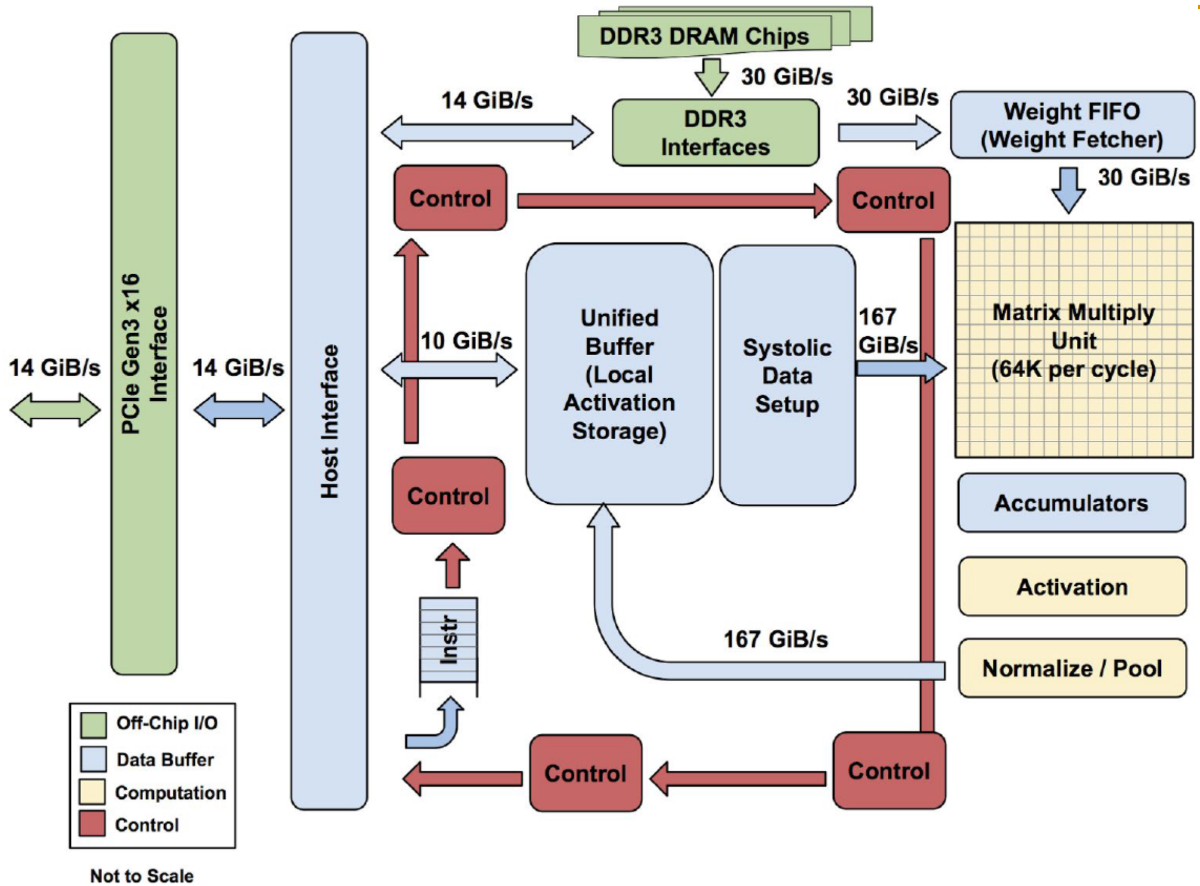


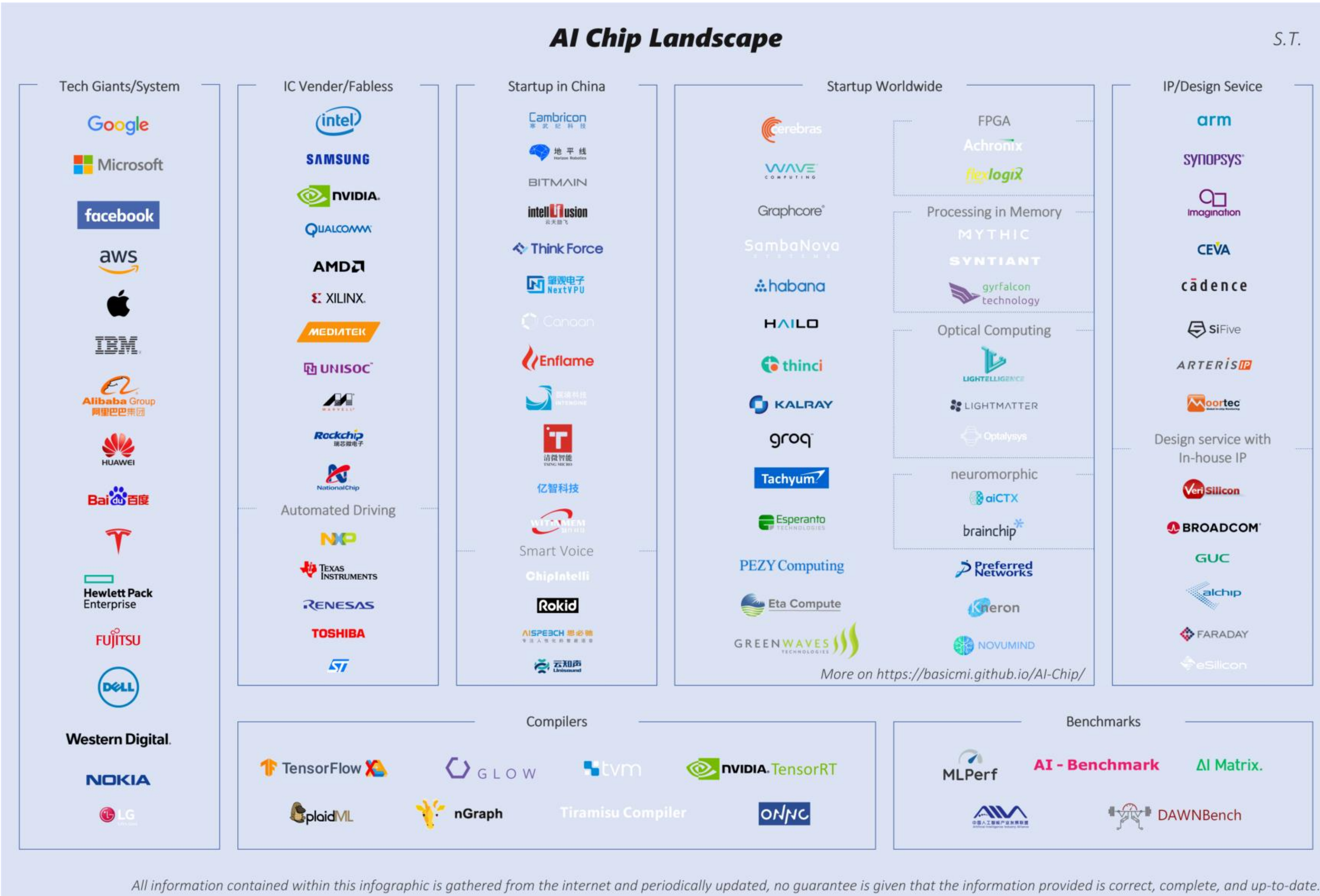

**Figure 1.** TPU Block Diagram. The main computation part is the yellow Matrix Multiply unit in the upper right hand corner. Its inputs are the blue Weight FIFO and the blue Unified Buffer (UB) and its output is the blue Accumulators (Acc). The yellow Activation Unit performs the nonlinear functions on the Acc, which go to the UB.

# Many (Other) AI/ML Chips

- Alibaba
- Amazon
- Facebook
- Google
- Huawei
- Intel
- Microsoft
- NVIDIA
- Tesla
- Many Others and Many Startups…

- **Many More to Come…**

# Many (Other) AI/ML Chips



# Many Interesting Things Are Happening Today in Computer Architecture

# Many Interesting Things Are Happening Today in Computer Architecture

**Reliability and Security**

# Security: RowHammer (2014)

# The Story of RowHammer

- One can predictably induce bit flips in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable

- First example of how a simple hardware failure mechanism can create a widespread system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

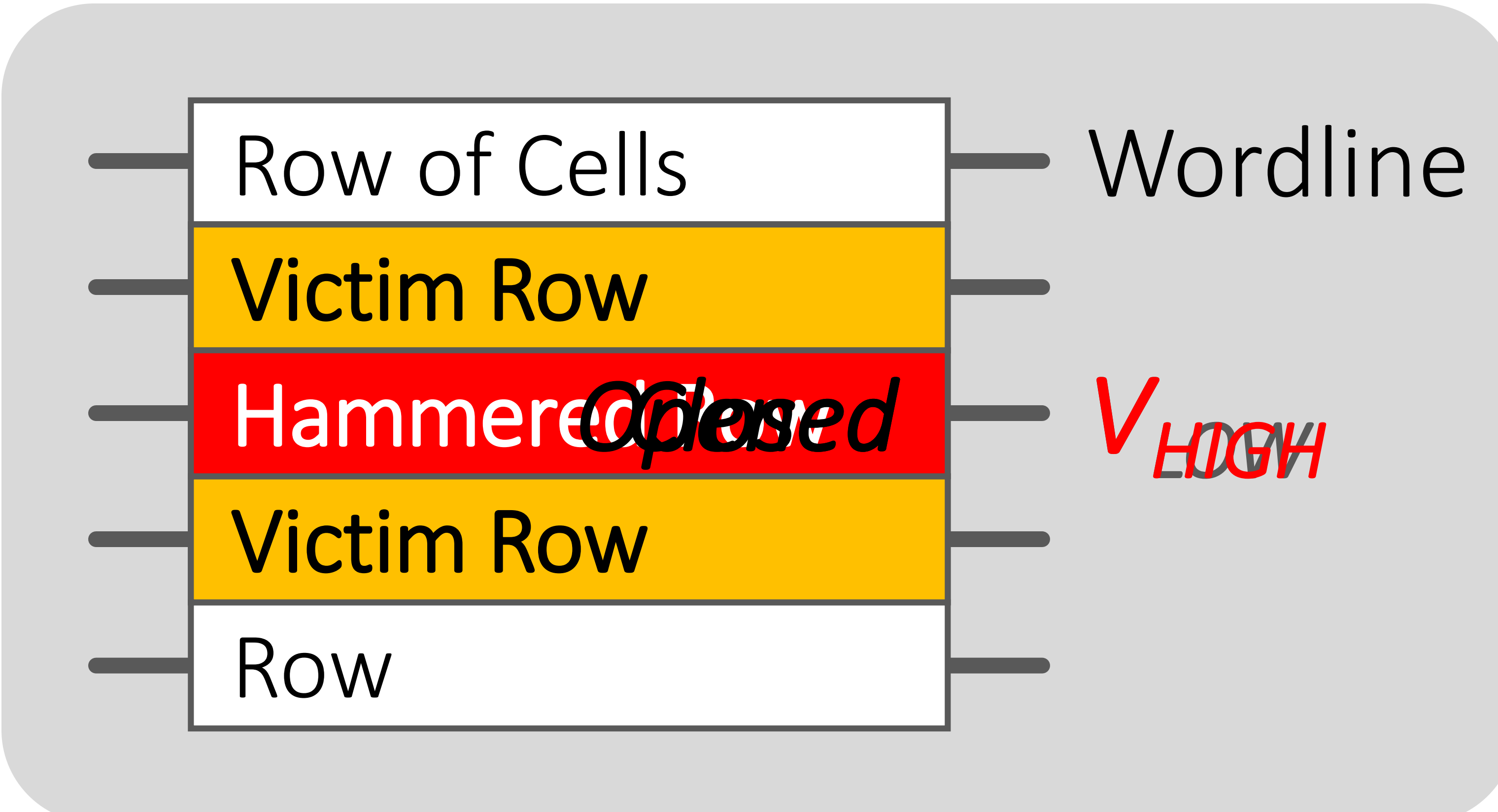
SHARE

SHARE 18276

TWEET

## FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# Modern DRAM is Prone to Disturbance Errors

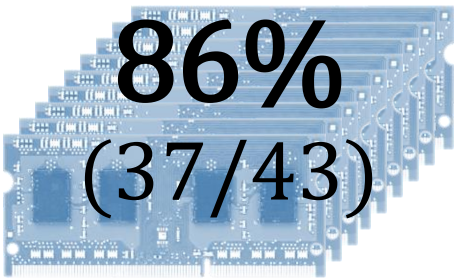



**Repeatedly reading** a row enough times (before memory gets refreshed) induces disturbance errors in **adjacent rows** in most real DRAM chips you can buy today

[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](), (Kim et al., ISCA 2014)

# Most DRAM Modules Are Vulnerable

## A company

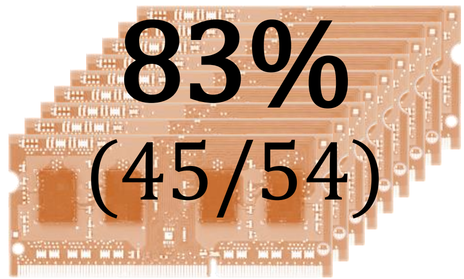



Up to

**$1.0\times10^{7}$**

errors

## B company

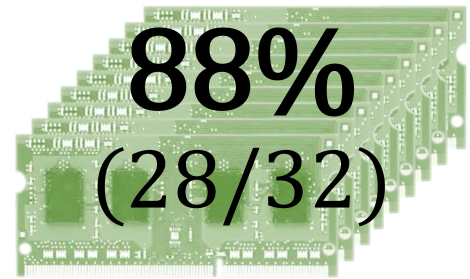



Up to

**$2.7\times10^{6}$**

errors

## C company




Up to

**$3.3\times10^{5}$**

errors

[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](), (Kim et al., ISCA 2014)

# One Can Take Over an Otherwise-Secure System

**Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors**

*Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology*

Project Zero

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

News and updates from the Project Zero team at Google

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn, 2015)

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

# Security: RowHammer (2014)


It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after

# RowHammer: Five Years Ago…

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
**"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"**
*Proceedings of the 41st International Symposium on Computer Architecture (**ISCA**)*, Minneapolis, MN, June 2014.
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]

## Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]Intel Labs

# RowHammer: Now and Beyond…

- Onur Mutlu and Jeremie Kim,
  **"RowHammer: A Retrospective"**
  *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (***TCAD***) *Special Issue on Top Picks in Hardware and Embedded Security*, 2019.
  [Preliminary arXiv version]

## RowHammer: A Retrospective

Onur Mutlu[§‡] Jeremie S. Kim[‡§]
[§]ETH Zürich [‡]Carnegie Mellon University

# RowHammer in 2020 (I)

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,
  **"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"**
  *Proceedings of the 47th International Symposium on Computer Architecture* (***ISCA***), Valencia, Spain, June 2020.
  [Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (20 minutes)]
  [Lightning Talk Video (3 minutes)]

**Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques**

Jeremie S. Kim[§†] Minesh Patel[§] A. Giray Yağlıkçı[§]
Hasan Hassan[§] Roknoddin Azizi[§] Lois Orosa[§] Onur Mutlu[§†]

[§]*ETH Zürich* [†]*Carnegie Mellon University*

# RowHammer in 2020 (II)

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,
**"TRRespass: Exploiting the Many Sides of Target Row Refresh"**
*Proceedings of the 41st IEEE Symposium on Security and Privacy* (***S&P***), San Francisco, CA, USA, May 2020.
[Slides (pptx) (pdf)]
[Talk Video (17 minutes)]
[Source Code]
[Web Article]
***Best paper award.***

# TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo[*†] Emanuele Vannacci[*†] Hasan Hassan[§] Victor van der Veen[¶]
Onur Mutlu[§] Cristiano Giuffrida[*] Herbert Bos[*] Kaveh Razavi[*]

[*]Vrije Universiteit Amsterdam [§]ETH Zürich [¶]Qualcomm Technologies Inc.

# RowHammer in 2020 (III)

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,
  **"Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"**
  *Proceedings of the 41st IEEE Symposium on Security and Privacy (**S&P**)*, San Francisco, CA, USA, May 2020.
  [Slides (pptx) (pdf)]
  [Talk Video (17 minutes)]

## Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers

Lucian Cojocar, Jeremie Kim[§†], Minesh Patel[§], Lillian Tsai[‡],
Stefan Saroiu, Alec Wolman, and Onur Mutlu[§†]
Microsoft Research, [§]ETH Zürich, [†]CMU, [‡]MIT

# Security: Meltdown and Spectre (2018)


Source: J. Masters, Redhat, FOSDEM 2018 keynote talk.

# Meltdown and Spectre

- Someone can steal secret data from the system even though
  - your program and data are perfectly correct and
  - your hardware behaves according to the specification and
  - there are no software vulnerabilities/bugs

- Why?
  - Speculative execution leaves traces of secret data in the processor's cache (internal storage)
    - It brings data that is not supposed to be brought/accessed if there was no speculative execution
  - A malicious program can inspect the contents of the cache to "infer" secret data that it is not supposed to access
  - A malicious program can actually force another program to speculatively execute code that leaves traces of secret data

# More on Meltdown/Spectre Vulnerabilities

# Project Zero

News and updates from the Project Zero team at Google

Wednesday, January 3, 2018

## Reading privileged memory with a side-channel

Posted by Jann Horn, Project Zero

We have discovered that CPU data cache timing can be abused to efficiently leak information out of mis-speculated execution, leading to (at worst) arbitrary virtual memory read vulnerabilities across local security boundaries in various contexts.

Source: https://googleprojectzero.blogspot.ch/2018/01/reading-privileged-memory-with-side.html

# Many Interesting Things Are Happening Today in Computer Architecture

# Many Interesting Things Are Happening Today in Computer Architecture

**More Demanding Workloads**

# New Genome Sequencing Technologies

**Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions**

Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

*Briefings in Bioinformatics*, bby017, https://doi.org/10.1093/bib/bby017

**Published:** 02 April 2018 **Article history ▾**


Oxford Nanopore MinION

Data → performance & energy bottleneck

# Why Do We Care? An Example

## 200 Oxford Nanopore sequencers have left UK for China, to support rapid, near-sample coronavirus sequencing for outbreak surveillance

Fri 31st January 2020

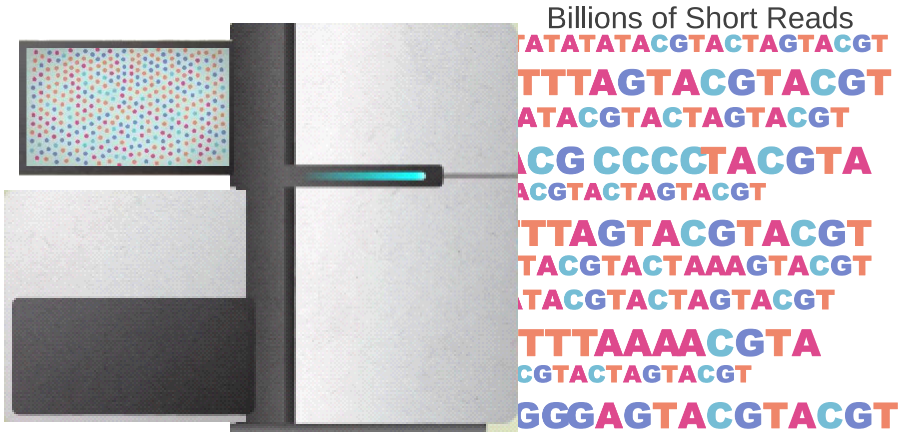
Following extensive support of, and collaboration with, public health professionals in China, Oxford Nanopore has shipped an additional 200 MinION sequencers and related consumables to China. These will be used to support the ongoing surveillance of the current coronavirus outbreak, adding to a large number of the devices already installed in the country.

Each MinION sequencer is approximately the size of a stapler, and can provide rapid sequence information about the coronavirus.


700Kg of Oxford Nanopore sequencers and consumables are on their way for use by Chinese scientists in understanding the current coronavirus outbreak.

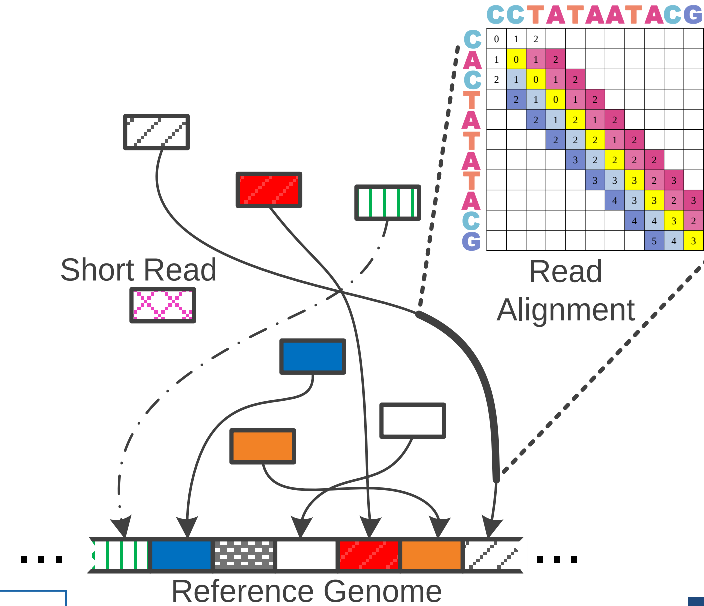


Source: https://nanoporetech.com/about-us/news/200-oxford-nanopore-sequencers-have-left-uk-china-support-rapid-near-sample




**1 Sequencing**

# Genome Analysis




**Read Mapping 2**

## Data → performance & energy bottleneck

```
read4:              CGCTTCCAT
read5:                   CCATGACGC
read6:                 TTCCATGAC
```

**3 Variant Calling**

**Scientific Discovery 4**

# GateKeeper: FPGA-Based Alignment Filtering

- Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan
  **"GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"**
  ***Bioinformatics***, [published online, May 31], 2017.
  [Source Code]
  [Online link at Bioinformatics Journal]

## GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

Mohammed Alser ✉, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu ✉, Can Alkan ✉

*Bioinformatics*, Volume 33, Issue 21, 1 November 2017, Pages 3355–3363,
https://doi.org/10.1093/bioinformatics/btx342
**Published:** 31 May 2017 **Article history** ▾

# In-Memory DNA Sequence Analysis

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu, **"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"**
  ***BMC Genomics***, 2018.
  *Proceedings of the 16th Asia Pacific Bioinformatics Conference (**APBC**)*, Yokohama, Japan, January 2018.
  arxiv.org Version (pdf)

## GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies

Jeremie S. Kim[1,6]*, Damla Senol Cali[1], Hongyi Xin[2], Donghyuk Lee[3], Saugata Ghose[1], Mohammed Alser[4], Hasan Hassan[6], Oguz Ergin[5], Can Alkan[4]* and Onur Mutlu[6,1]*

*From* The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

# GenASM: Fast Approximate String Matching

## GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis

Damla Senol Cali[†⋈] Gurpreet S. Kalsi[⋈] Zülal Bingöl[▽] Can Firtina[◇] Lavanya Subramanian[‡] Jeremie S. Kim[◇†]
Rachata Ausavarungnirun[⊙] Mohammed Alser[◇] Juan Gomez-Luna[◇] Amirali Boroumand[†] Anant Nori[⋈]
Allison Scibisz[†] Sreenivas Subramoney[⋈] Can Alkan[▽] Saugata Ghose[⋆†] Onur Mutlu[◇†▽]

[†]*Carnegie Mellon University* [⋈]*Processor Architecture Research Lab, Intel Labs* [▽]*Bilkent University* [◇]*ETH Zürich*
[‡]*Facebook* [⊙]*King Mongkut's University of Technology North Bangkok* [⋆]*University of Illinois at Urbana–Champaign*

# More on Genome Analysis: Another Lecture

- Onur Mutlu,
  **"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**
  *Keynote talk at 2nd Workshop on Accelerator Architecture in Computational Biology and Bioinformatics (**AACBB**)*, Washington, DC, USA, February 2019.
  [Slides (pptx)(pdf)]
  [Video]

## Accelerating Genome Analysis

### A Primer on an Ongoing Journey

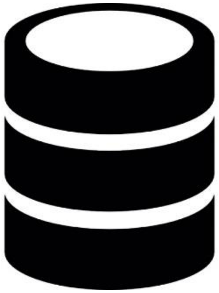
Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
16 February 2019
AACBB Keynote Talk

SAFARI ETH zürich Carnegie Mellon

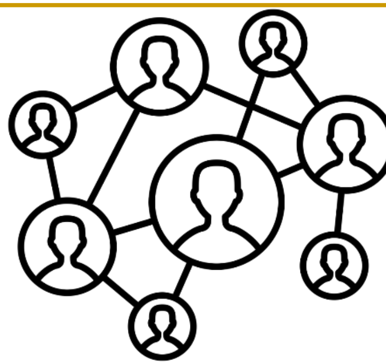


https://www.youtube.com/watch?v=hPnSmfwu2-A

# Data Overwhelms Modern Machines



**In-memory Databases**

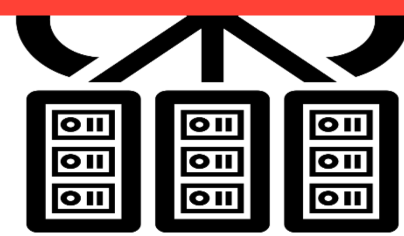


**Graph/Tree Processing**

Data → performance & energy bottleneck

**In-Memory Data Analytics**

[Clapp+ (**Intel**), IISWC'15;
Awan+, BDCloud'15]



**Datacenter Workloads**

[Kanev+ (**Google**), ISCA'15]

# Data Overwhelms Modern Machines

**Chrome**

**TensorFlow Mobile**

Data → performance & energy bottleneck

**Video Playback**

Google's **video codec**

**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[1] Saugata Ghose[1] Youngsok Kim[2]
Rachata Ausavarungnirun[1] Eric Shiu[3] Rahul Thakur[3] Daehyun Kim[4,3]
Aki Kuusela[3] Allan Knies[3] Parthasarathy Ranganathan[3] Onur Mutlu[5,1]

# 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
  - Fundamentally Secure and Dependable Computers

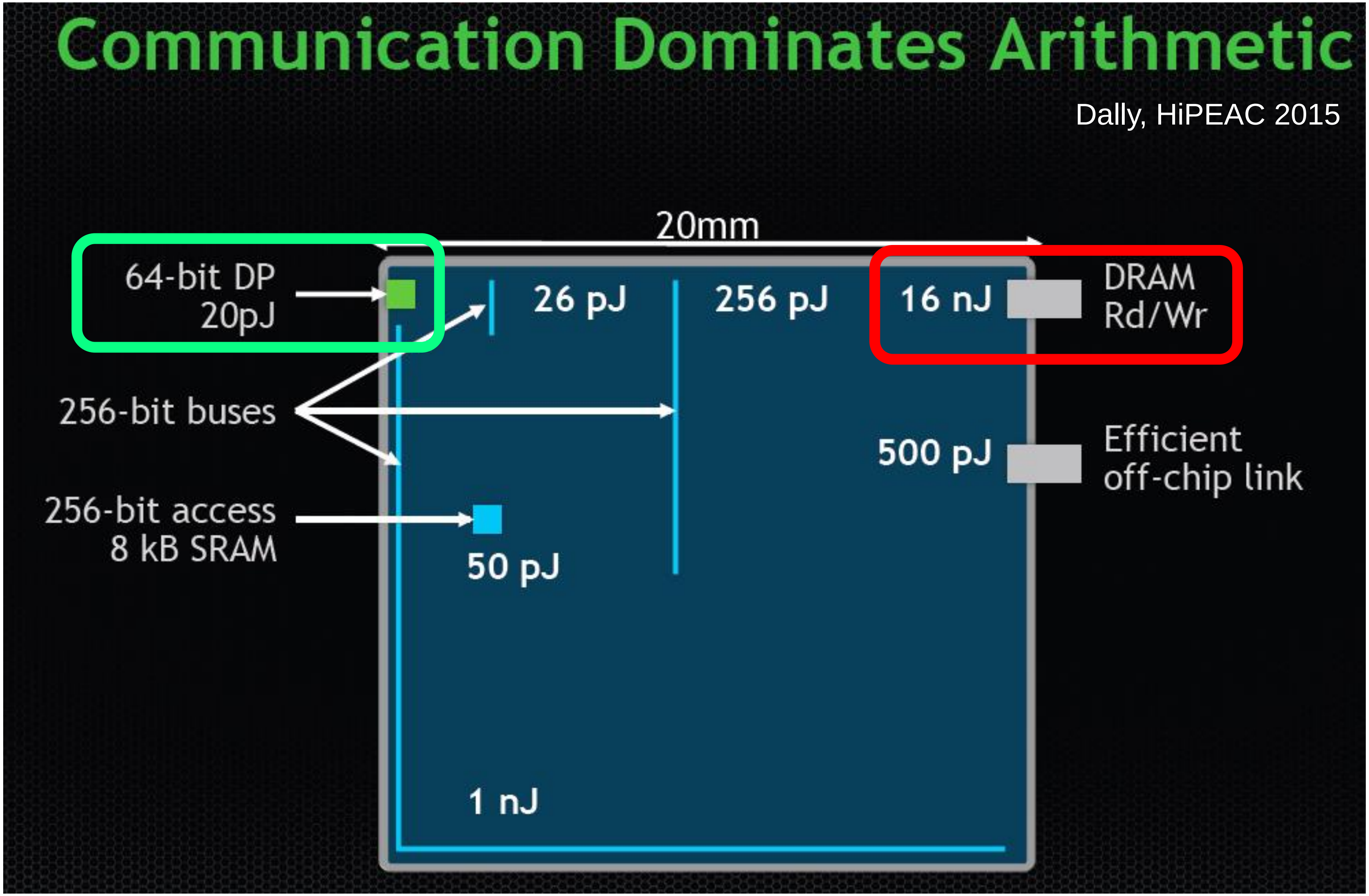
- New Accelerators (Algorithm-Hardware Co-Designs)
  - Artificial Intelligence & Machine Learning
  - Graph Analytics
  - Genome Analysis

- New Memories and Storage Systems
  - Non-Volatile Main Memory
  - Intelligent Memory

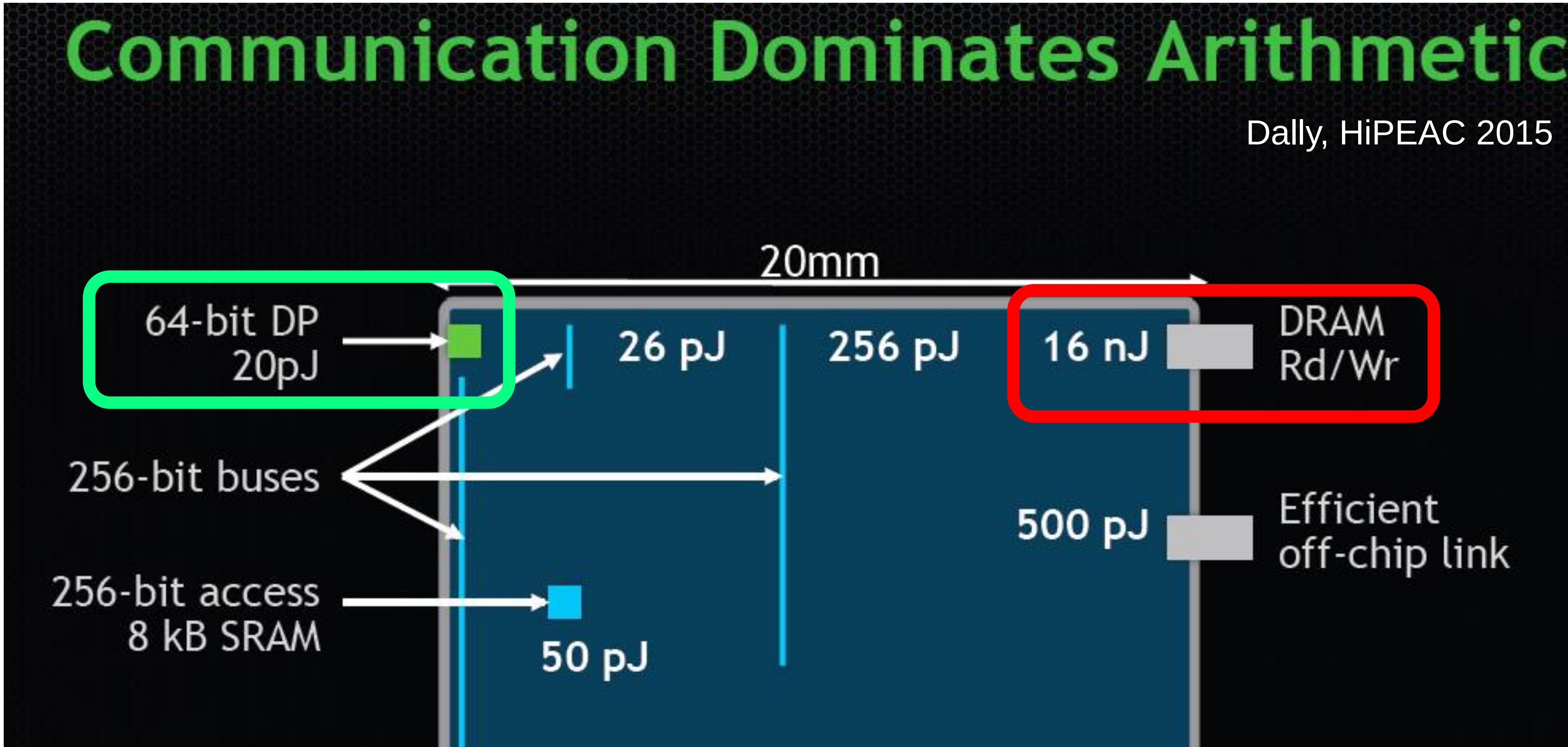
# Increasingly Demanding Applications

- Dream, and they will come

# Increasingly Diverging/Complex Tradeoffs



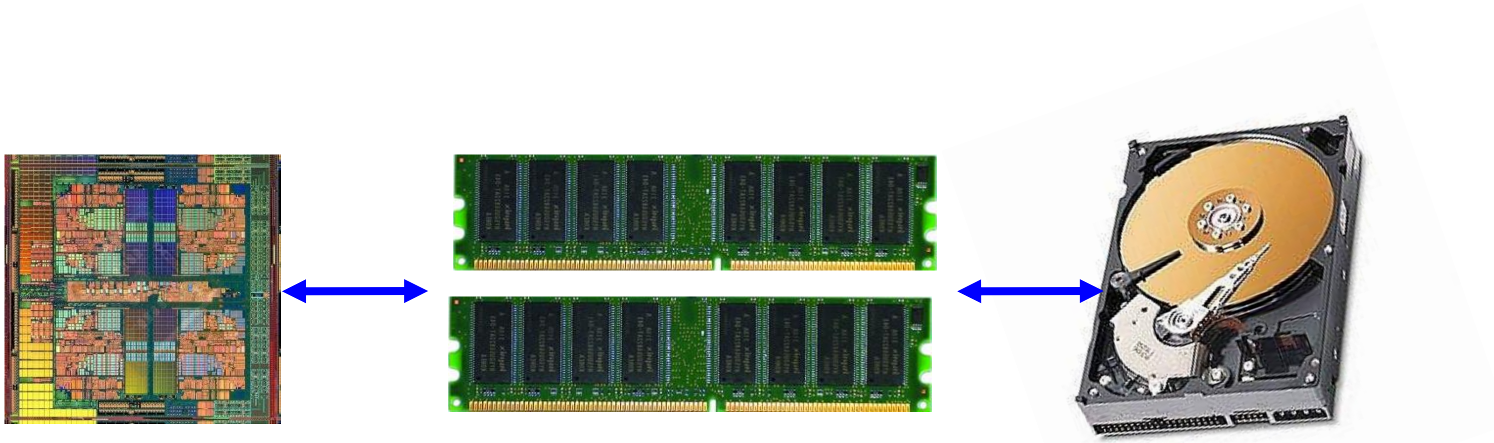
# Increasingly Diverging/Complex Tradeoffs




**A memory access consumes ~1000X the energy of a complex addition**

# Increasingly Complex Systems

## Past systems

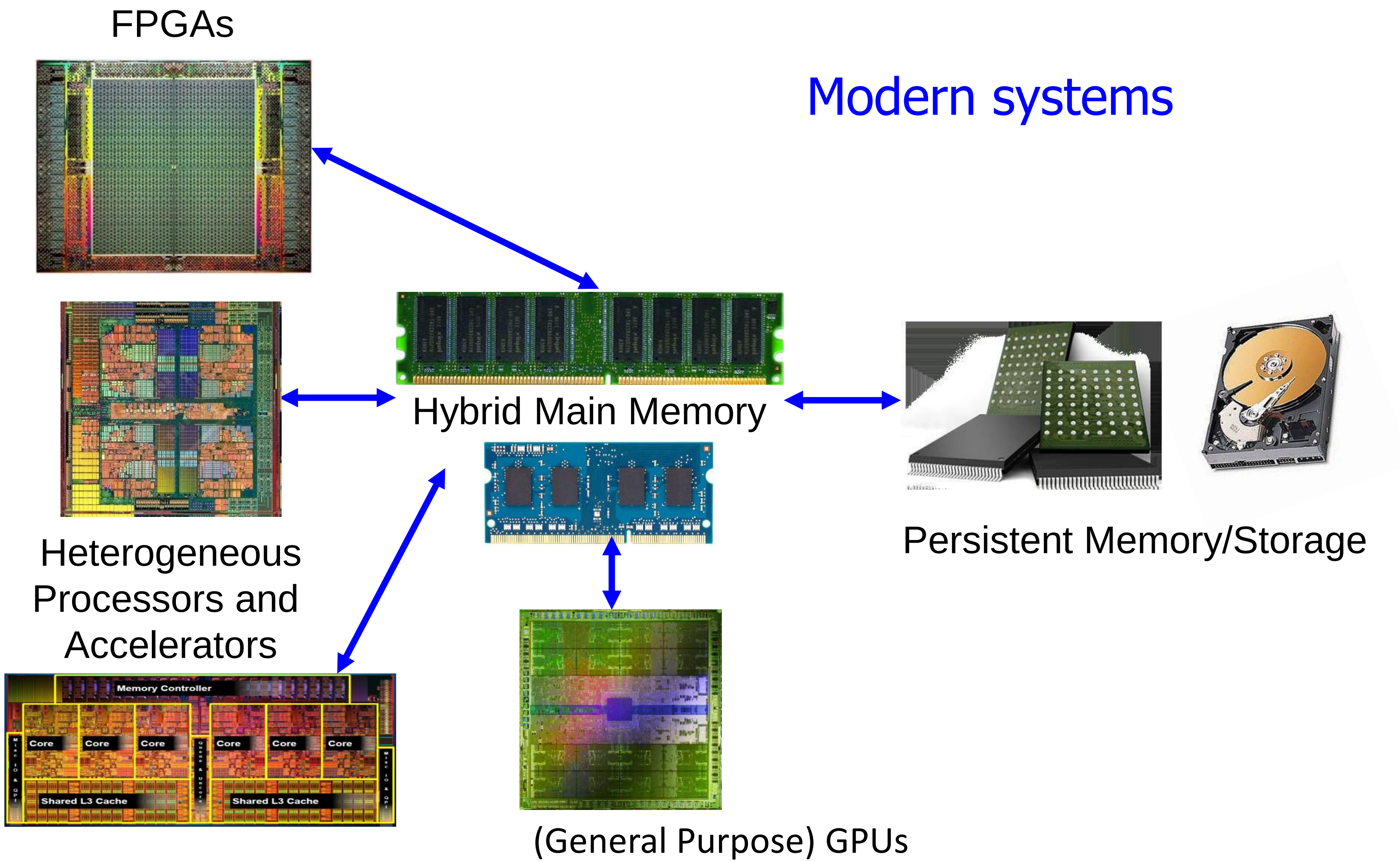


Microprocessor

Main Memory

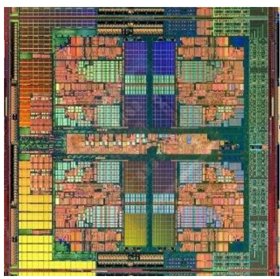
Storage (SSD/HDD)

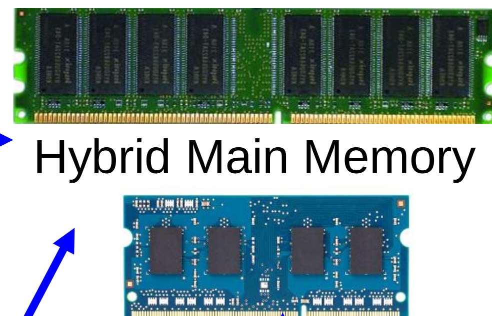
# Increasingly Complex Systems



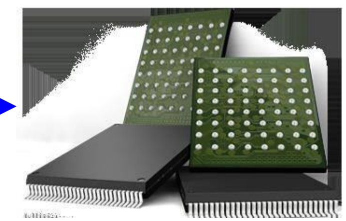
# Computer Architecture Today

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



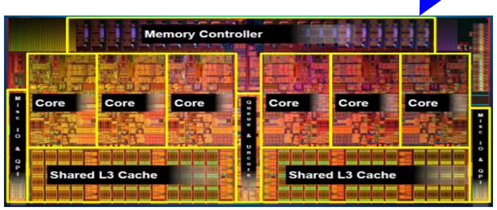


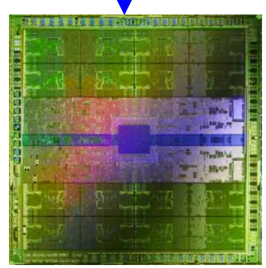
Hybrid Main Memory



Persistent Memory/Storage

Heterogeneous Processors and Accelerators



General Purpose GPUs

**Every component and its interfaces, as well as entire system designs are being 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)

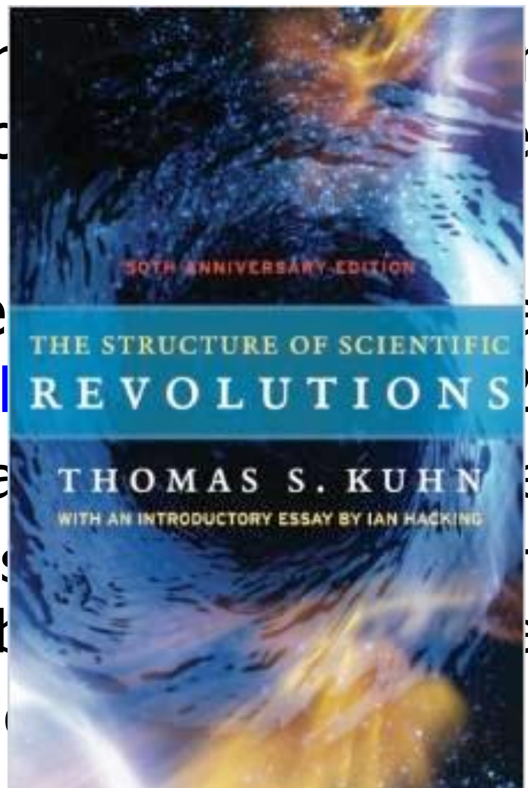
- You can invent new paradigms for computation, communication, and storage

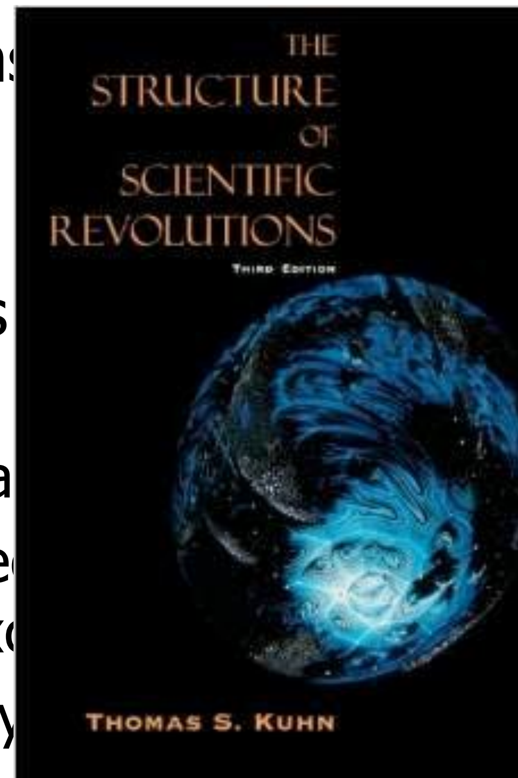
- Recommended book: Thomas Kuhn, "The Structure of Scientific Revolutions" (1962)
  - Pre-paradigm science: no clear consensus in the field
  - Normal science: dominant theory used to explain/improve things (business as usual); exceptions considered anomalies
  - Revolutionary science: underlying assumptions 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 ir ... ms communic ... e




- Recomme ... as ... ure of Scientific I ... 2)
  - Pre-para ... ea ... ield
  - Normal ... ne ... improve things (b ... ex ... anomalies
  - Revoluti ... rly ... -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 Fundamentals

# Question: What Is This?


Source: By Toni_V, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=4087256

# Answer: The First Major Piece of a Famous Architect

- **Bahnhof Stadelhofen:** "The train station has several of the features that became signatures of his work; straight lines and right angles are rare."
- ETH Alumnus, PhD in Civil Engineering

**Santiago Calatrava Valls** (born 28 July 1951) is a Spanish architect, structural engineer, sculptor and painter, particularly known for his bridges supported by single leaning pylons, and his railway stations, stadiums, and museums, whose sculptural forms often resemble living organisms.[1] His best-known works include the Milwaukee Art Museum, the Turning Torso tower in Malmo, Sweden, the Margaret Hunt Hill Bridge in Dallas, Texas, and the Museum of Tomorrow in Rio de Janeiro,

Source: By 準建築人手札網站 Forgemind ArchiMedia - Flickr: IMG_2489.JPG, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=31493356, https://en.wikipedia.org/wiki/Santiago_Calatrava

# Compare To This

Source: http://cookiemagik.deviantart.com/art/Train-station-207266944

# Question 2: What Is This?

Source: https://www.dezeen.com/2016/08/29/santiago-calatrava-oculus-world-trade-center-transportation-hub-new-york-photographs-hufton-crow/

# Answer: Masterpiece of a Famous Architect

## 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:

Source: https://en.wikipedia.org/wiki/World_Trade_Center_station_(PATH)

# 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.[43]

Source: https://en.wikipedia.org/wiki/World_Trade_Center_station_(PATH)

# 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.[45]

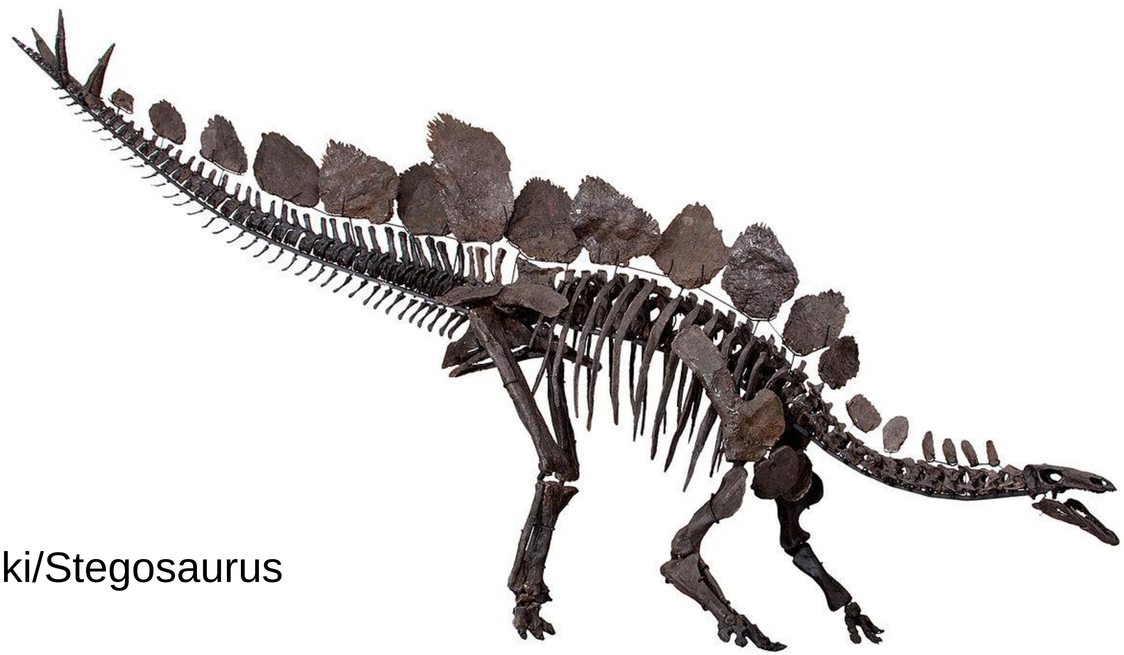
Source: https://en.wikipedia.org/wiki/World_Trade_Center_station_(PATH)

# *Stegosaurus*

From Wikipedia, the free encyclopedia

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

***Stegosaurus*** (/ˌstɛɡəˈsɔːrəs/[1]) 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:

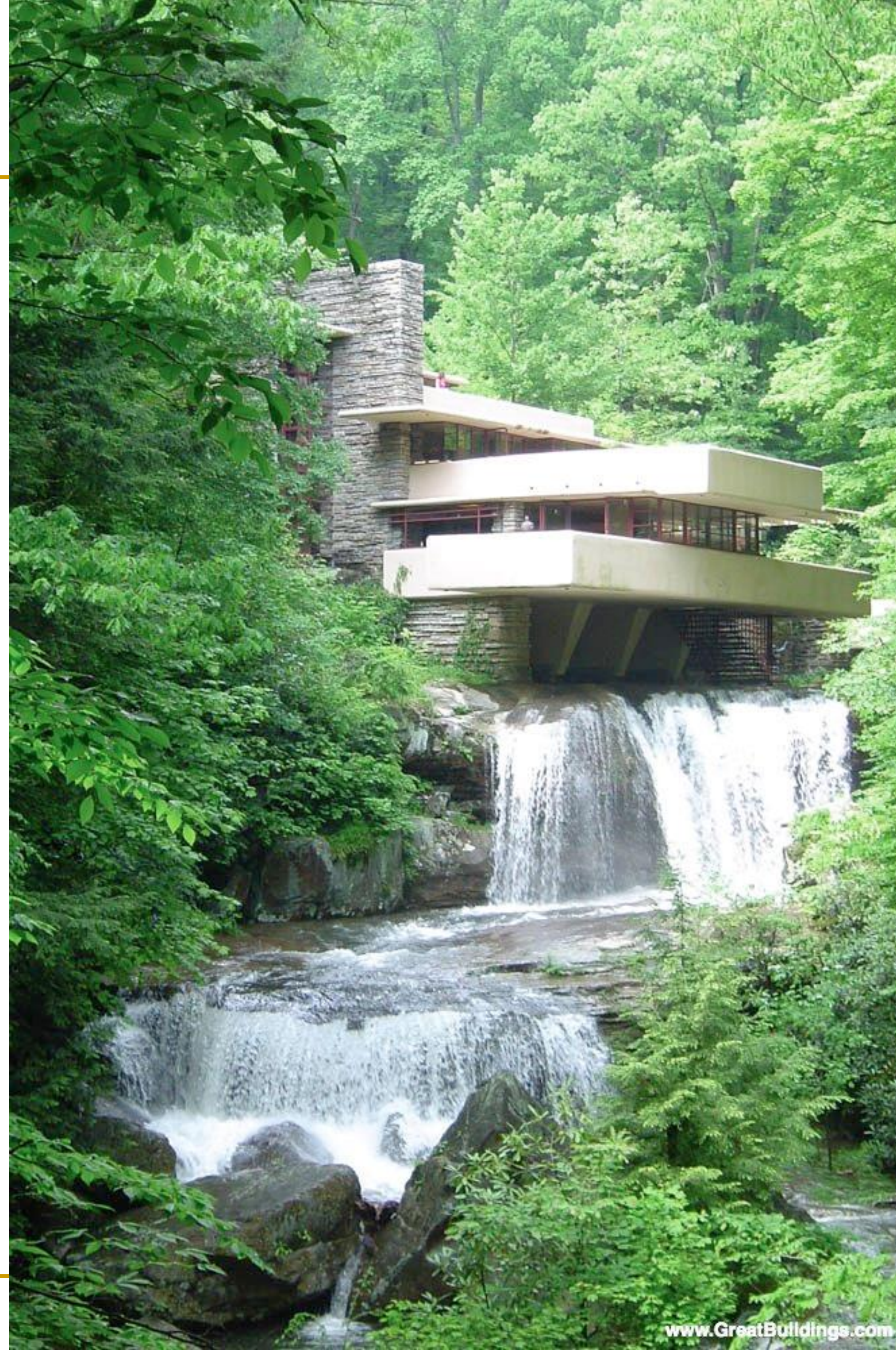
> In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass.... [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural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

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

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

Source: https://en.wikipedia.org/wiki/World_Trade_Center_station_(PATH)

# Question: What Is This?


www.GreatBuildings.com

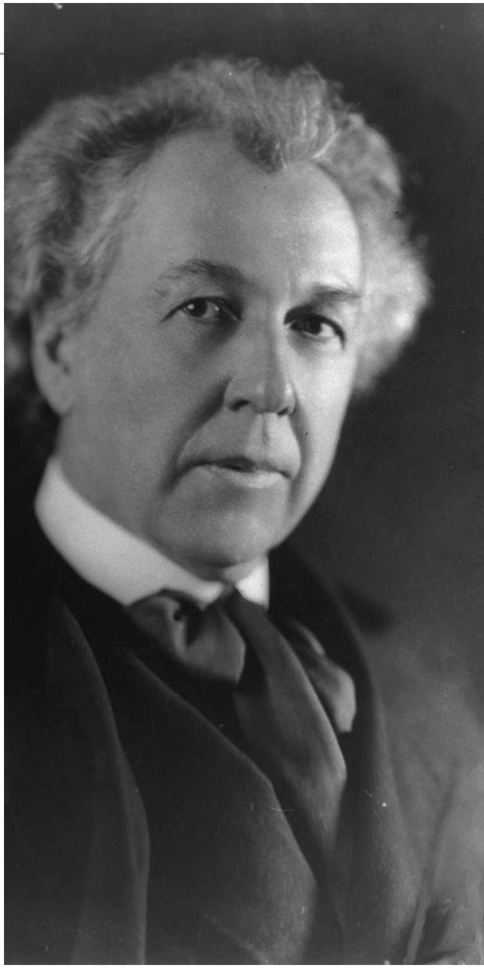
# Answer: Masterpiece of Another Famous Architect

# 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.[4] 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";[5] it is listed among *Smithsonian*'s Life List of 28 places "to visit before you die."[6] It was designated a National Historic Landmark in 1966.[3] 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.



Source: https://en.wikipedia.org/wiki/Fallingwater

# Your First Comp Arch Assignment

- Go and visit Bahnhof Stadelhofen
  - Extra credit: Repeat for Oculus
  - Extra+ credit: Repeat for Fallingwater

- Appreciate the beauty & out-of-the-box and creative thinking
- Think about tradeoffs in the design of the Bahnhof
  - Strengths, weaknesses, goals of design
- Derive principles on your own for good design and innovation

- Due date: **Any time during this course**
  - Later during the course is better
  - Apply what you have learned in this course
  - Think out-of-the-box

# But First, Today's First Assignment

- Find The Differences Of This and That

# Find The Differences of This and That

# This


Source: By Toni_V from Zurich, Switzerland - Stadelhofen2, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=4087256

# That


Source: http://cookiemagik.deviantart.com/art/Train-station-207266944

# Many Tradeoffs Between Two Designs

- You can list them after you complete the first assignment…

# 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
- …

- How to evaluate goodness of design is always a critical question.

# A Key Question

- How was Calavatra able to design especially his key buildings?
- Can have many guesses
  - (Ultra) 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."

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

# 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."[1]

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.[3]

# What Does This Remind You Of?

Source: https://www.dezeen.com/2016/08/29/santiago-calatrava-oculus-world-trade-center-transportation-hub-new-york-photographs-hufton-crow/

# What About This?

Source: De Galván - Puente del Alamillo.jpg on Enciclopedia.us.es, GFDL, https://commons.wikimedia.org/w/index.php?curid=15026095

# A Quote from The Other Famous Architect

- “architecture […] based upon principle, and not upon precedent” (Frank Lloyd Wright)

Source: http://www.fallingwater.org/

# A Principled Design

## Organic architecture

From Wikipedia, the free encyclopedia

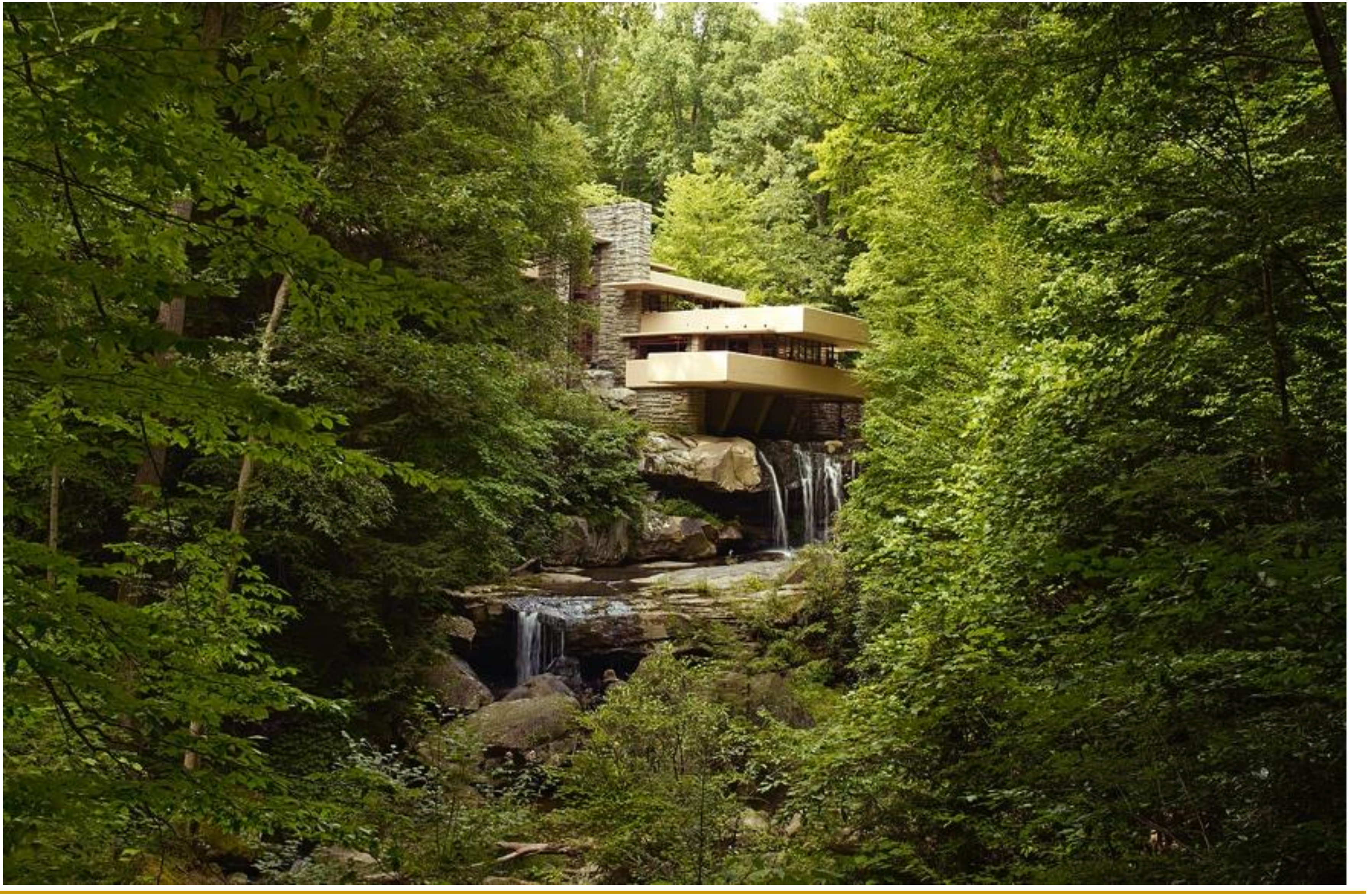
**Organic architecture** is a philosophy of architecture which promotes harmony between human habitation and the natural world through design approaches so sympathetic and well integrated with its site, that buildings, furnishings, and surroundings become part of a unified, interrelated composition.

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.

# Another View

Source: https://roadtrippers.com/stories/falling-water

# Yet Another View



Source: By Carol M. Highsmith - http://www.loc.gov/pictures/collection/highsm/item/2010630255/, Public Domain, https://commons.wikimedia.org/w/index.php?curid=29385254


www.GreatBuildings.com

# Major High-Level Goals of This Course

- Understand the principles
- Understand the precedents

- Based on such understanding:
  - Enable you to evaluate tradeoffs of different designs and ideas
  - Enable you to develop principled designs
  - Enable you to develop novel, out-of-the-box designs

- The focus is on:
  - Principles, precedents, and how to use them for new designs

- In Computer Architecture

# Role of the (Computer) Architect

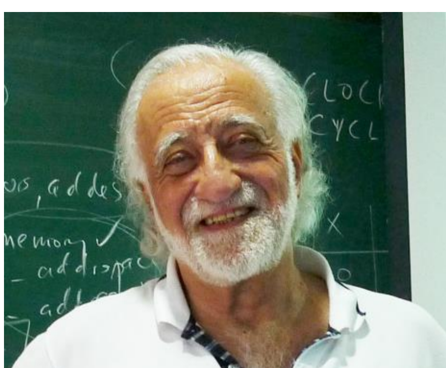
## Role of the Architect

-- Look Backward (Examine old code)

-- Look forward (Listen to the dreamers)

-- Look Up (Nature of the problems)

-- Look Down (Predict the future of technology)



from Yale Patt's lecture notes

# Role of The (Computer) Architect

- Look backward (to the past)
  - Understand tradeoffs and designs, upsides/downsides, past workloads. Analyze and evaluate the past.
- Look forward (to the future)
  - Be the dreamer and create new designs. Listen to dreamers.
  - Push the state of the art. Evaluate new design choices.
- Look up (towards problems in the computing stack)
  - Understand important problems and their nature.
  - Develop architectures and ideas to solve important problems.
- Look down (towards device/circuit technology)
  - Understand the capabilities of the underlying technology.
  - Predict and adapt to the future of technology (you are designing for N years ahead). Enable the future technology.

# Takeaways

- Being an architect is not easy
- You need to consider **many** things in designing a new system + have good intuition/insight into ideas/tradeoffs

- But, it is fun and can be very rewarding
- And, enables a great future
  - E.g., many scientific and everyday-life innovations would not have been possible without architectural innovation that enabled very high performance systems
  - E.g., your mobile phones
  - E.g., self-driving vehicles

- This course will enable you to become a good computer architect

# So, I Hope You Are Here for This

Comp. Systems

"C" as a model of computation

Programmer's view of how a computer system works

- How does an assembly program end up executing as digital logic?
- What happens in-between?
- How is a computer designed using logic gates and wires to satisfy specific goals?

*Architect/microarchitect's view: How to design a computer that meets system design goals.*

*Choices critically affect both the SW programmer and the HW designer*

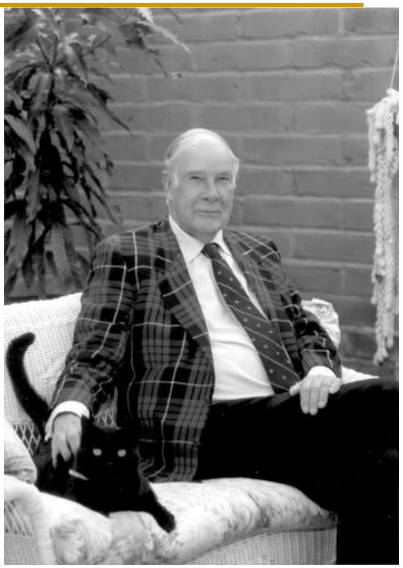
HW designer's view of how a computer system works

Digital logic as a model of computation

Digital Design

# Levels of Transformation

"The purpose of computing is [to gain] insight" (*Richard Hamming*)
*We gain and generate insight by solving problems*
*How do we ensure problems are solved by electrons?*

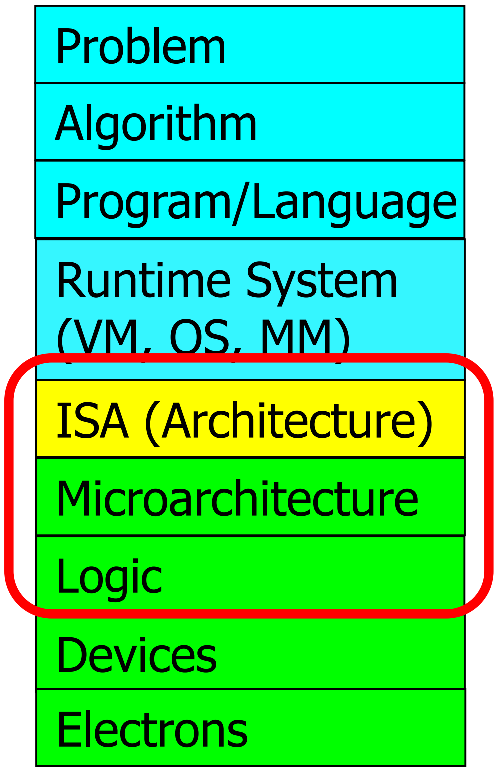


Algorithm

Step-by-step procedure that is **guaranteed to terminate** where **each step is precisely stated** and **can be carried out by a computer**

- **Finiteness**
- **Definiteness**
- **Effective computability**

Many algorithms for the same problem




ISA
(Instruction Set Architecture)

Interface/contract between SW and HW.

What the programmer assumes hardware will satisfy.

Microarchitecture
An implementation of the ISA

Digital logic circuits
Building blocks of micro-arch (e.g., gates)

# Aside: A Famous Work By Hamming

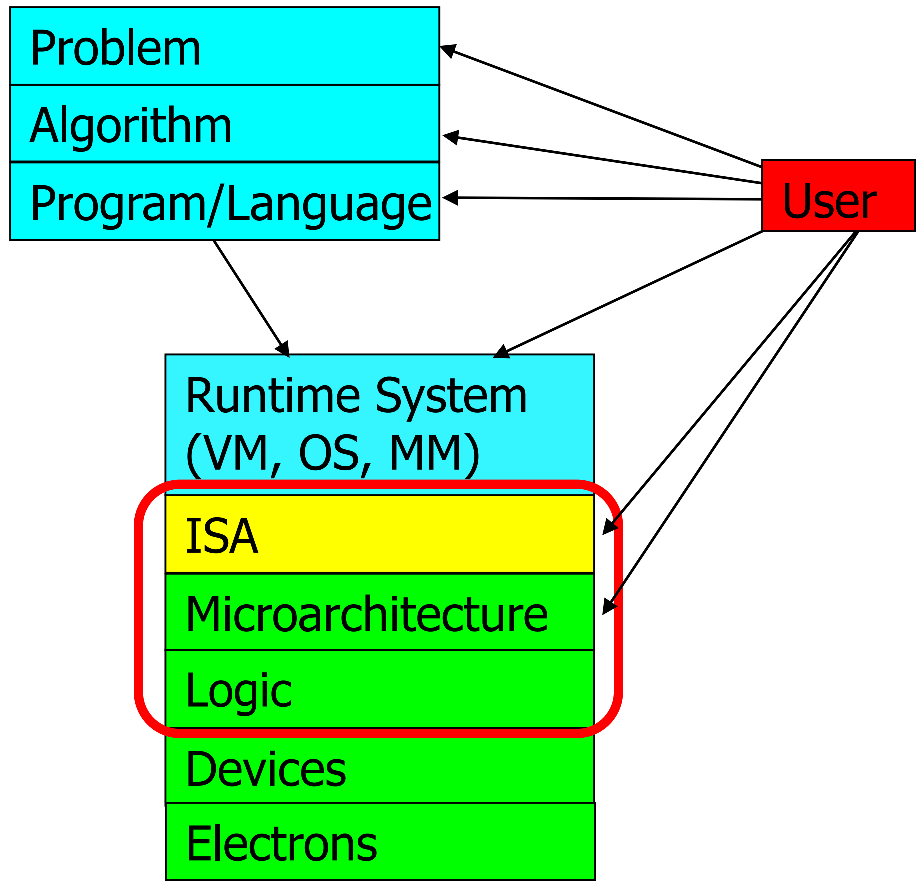
- Hamming, "Error Detecting and Error Correcting Codes," Bell System Technical Journal 1950.

- Introduced the concept of Hamming distance
  - number of locations in which the corresponding symbols of two equal-length strings is different
- Developed a theory of codes used for error detection and correction

- Also see:
  - Hamming, "You and Your Research," Talk at Bell Labs, 1986.
  - http://www.cs.virginia.edu/~robins/YouAndYourResearch.html

# Levels of Transformation, Revisited

- A user-centric view: computer designed for users




- The entire stack should be optimized for user

# The Power of Abstraction

- Levels of transformation create abstractions
  - Abstraction: A higher level only needs to know about the interface to the lower level, not how the lower level is implemented
  - E.g., high-level language programmer does not really need to know what the ISA is and how a computer executes instructions

- Abstraction improves productivity
  - No need to worry about decisions made in underlying levels
  - E.g., programming in Java vs. C vs. assembly vs. binary vs. by specifying control signals of each transistor every cycle

- Then, why would you want to know what goes on underneath or above?

# Crossing the Abstraction Layers

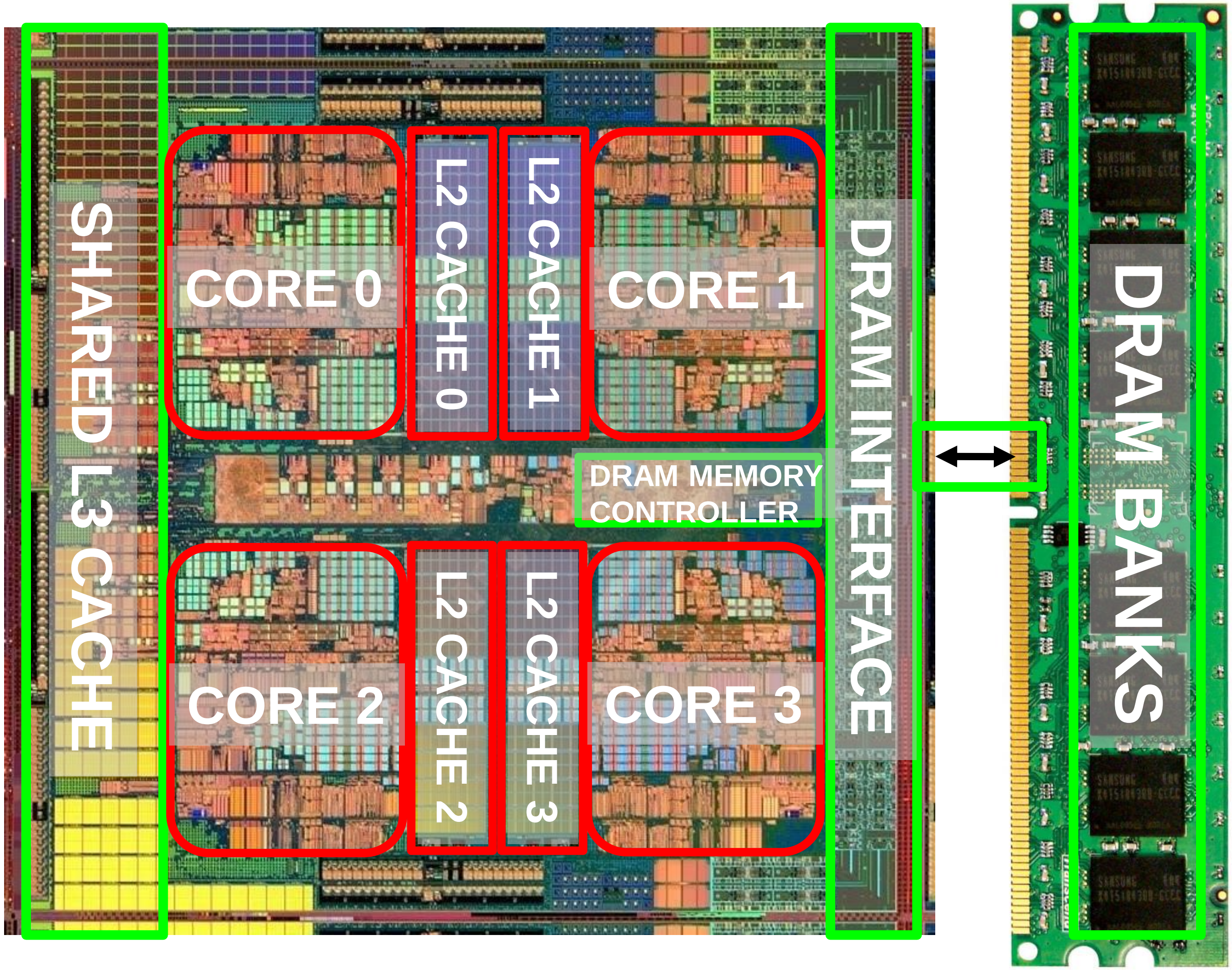
- As long as everything goes well, not knowing what happens underneath (or above) is not a problem.
- What if
  - The program you wrote is running slow?
  - The program you wrote does not run correctly?
  - The program you wrote consumes too much energy?
  - Your system just shut down and you have no idea why?
  - Someone just compromised your system and you have no idea how?
- What if
  - The hardware you designed is too hard to program?
  - The hardware you designed is too slow because it does not provide the right primitives to the software?
- What if
  - You want to design a much more efficient and higher performance system?

# Crossing the Abstraction Layers

- Two key goals of this course are
  - to understand how a processor works underneath the software layer and how decisions made in hardware affect the software/programmer
  - to enable you to be comfortable in making design and optimization decisions that cross the boundaries of different layers and system components

# An Example: Multi-Core Systems

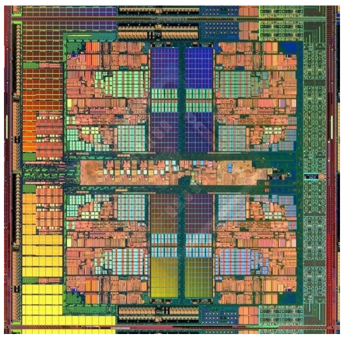
Multi-Core Chip



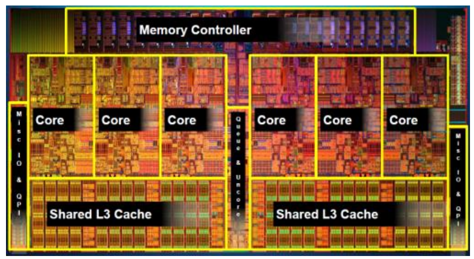

*Die photo credit: AMD Barcelona

# A Trend: Many Cores on Chip

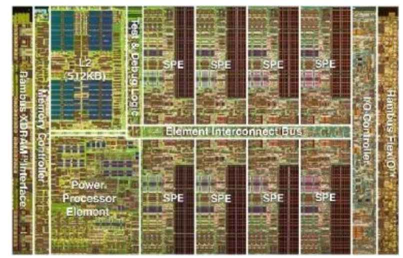
- Simpler and lower power than a single large core
- Parallel processing on single chip → faster, new applications



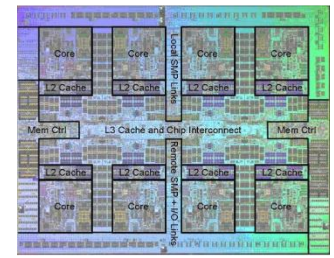
AMD Barcelona
4 cores



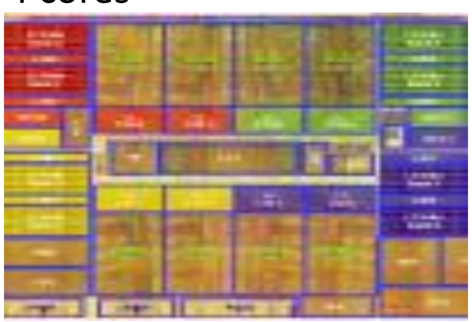

Intel Core i7
8 cores



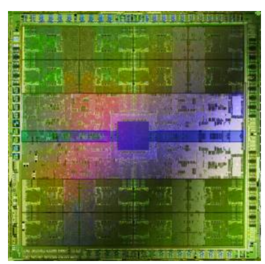

IBM Cell BE
8+1 cores



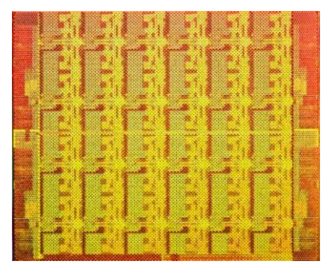

IBM POWER7
8 cores



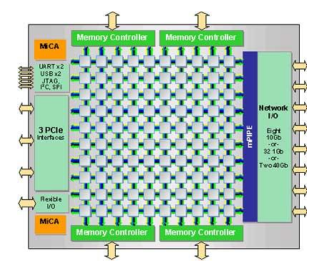
Sun Niagara II
8 cores



Nvidia Fermi
448 “cores”



Intel SCC
48 cores, networked



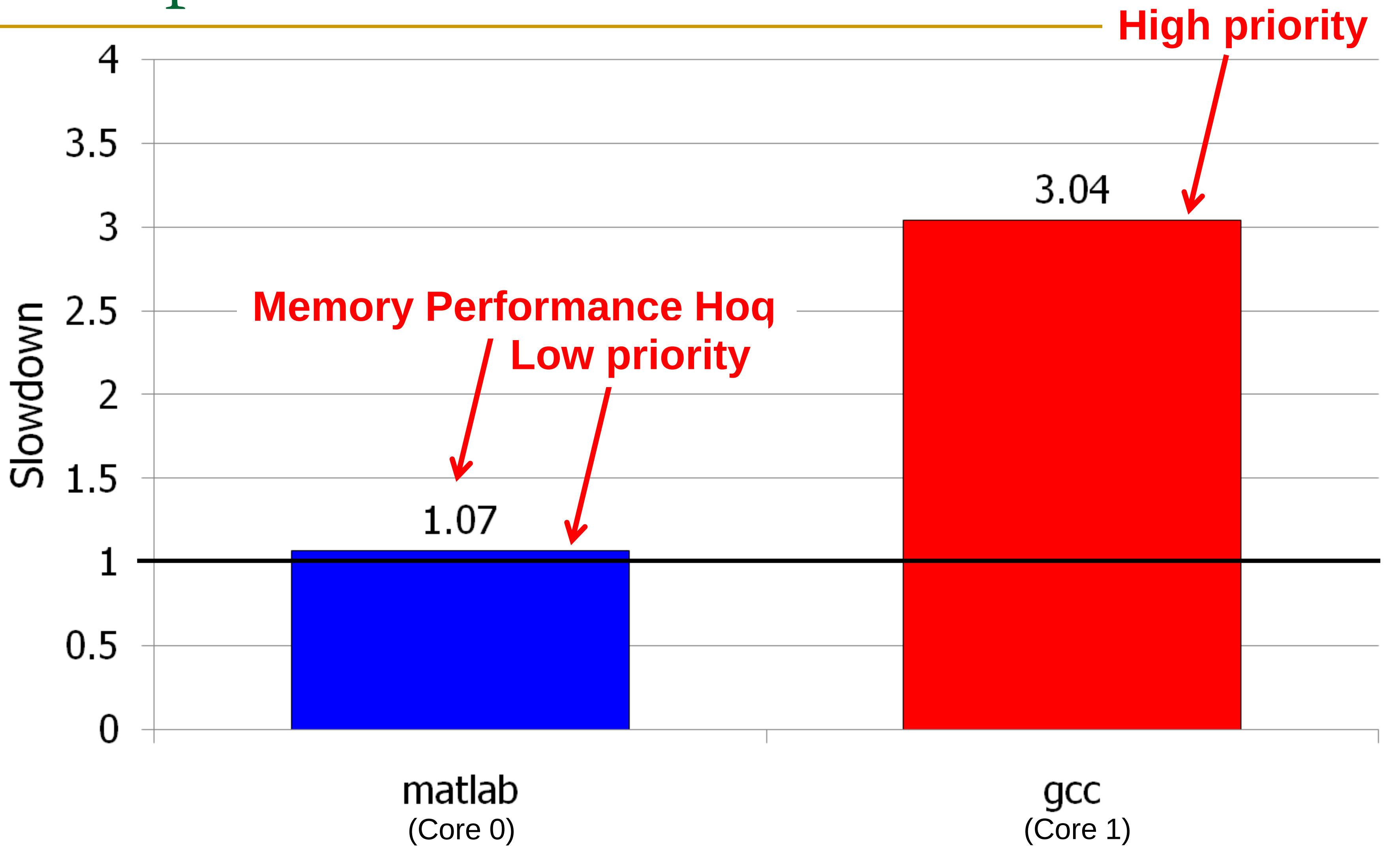

Tilera TILE Gx
100 cores, networked

# Many Cores on Chip

- What we want:
  - N times the system performance with N times the cores

- What do we get today?

# Unexpected Slowdowns in Multi-Core




Moscibroda and Mutlu, "Memory performance attacks: Denial of memory service in multi-core systems," USENIX Security 2007.

# Three Questions

- Can you figure out why the applications slow down if you do not know the underlying system and how it works?
- Can you figure out why there is a disparity in slowdowns if you do not know how the system executes the programs?
- Can you fix the problem without knowing what is happening "underneath"?

# Three Questions: Rephrased & Concise

- Why is there any slowdown?
- Why is there a disparity in slowdowns?
- How can we solve the problem if we do not want that disparity?

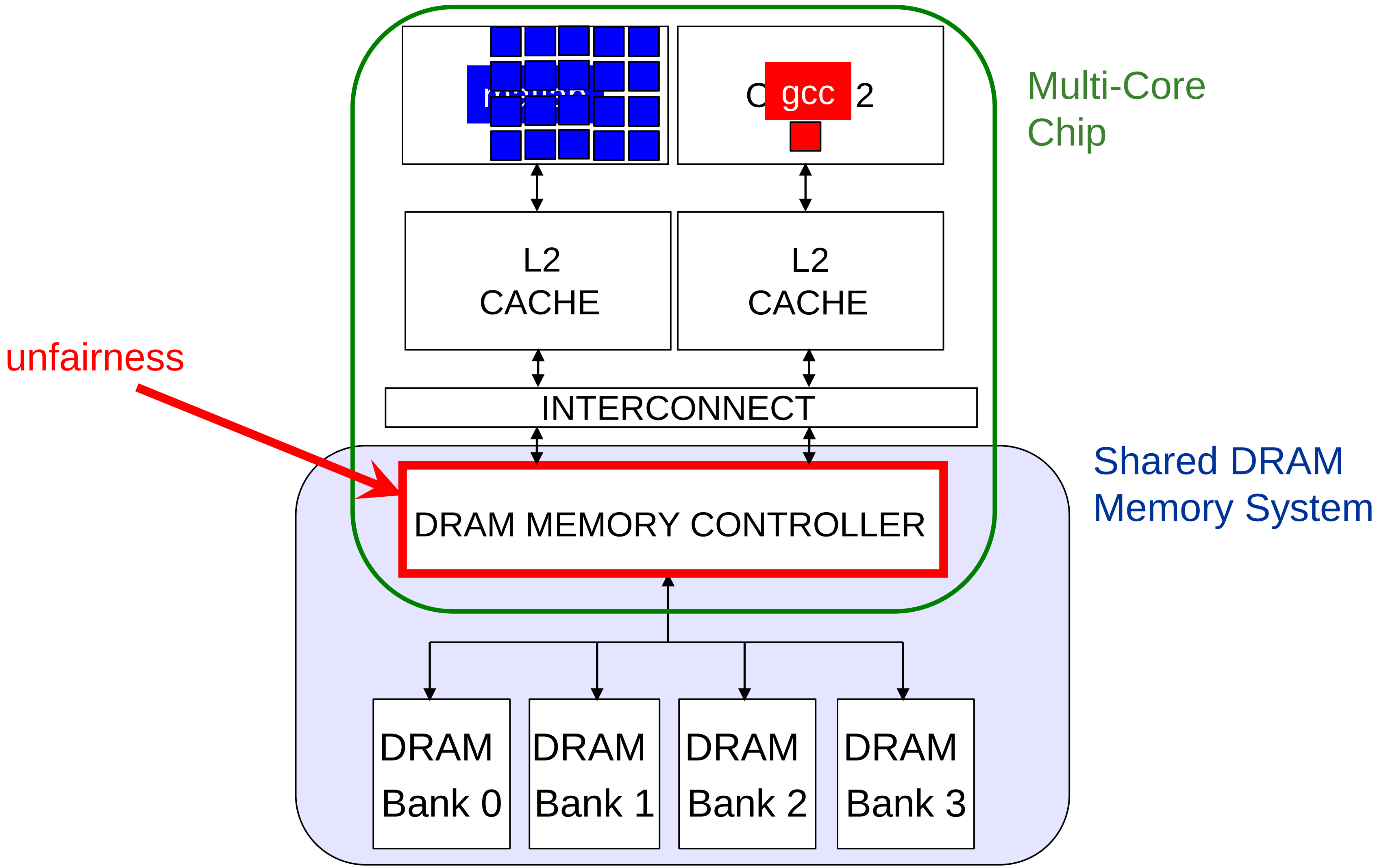
# Why Is This Important?

- We want to execute applications in parallel in multi-core systems → consolidate more and more (for efficiency)
  - Cloud computing
  - Mobile phones
  - Automotive systems

- We want to mix different types of applications together
  - those requiring QoS guarantees (e.g., video, pedestrian detection)
  - those that are important but less so
  - those that are less important

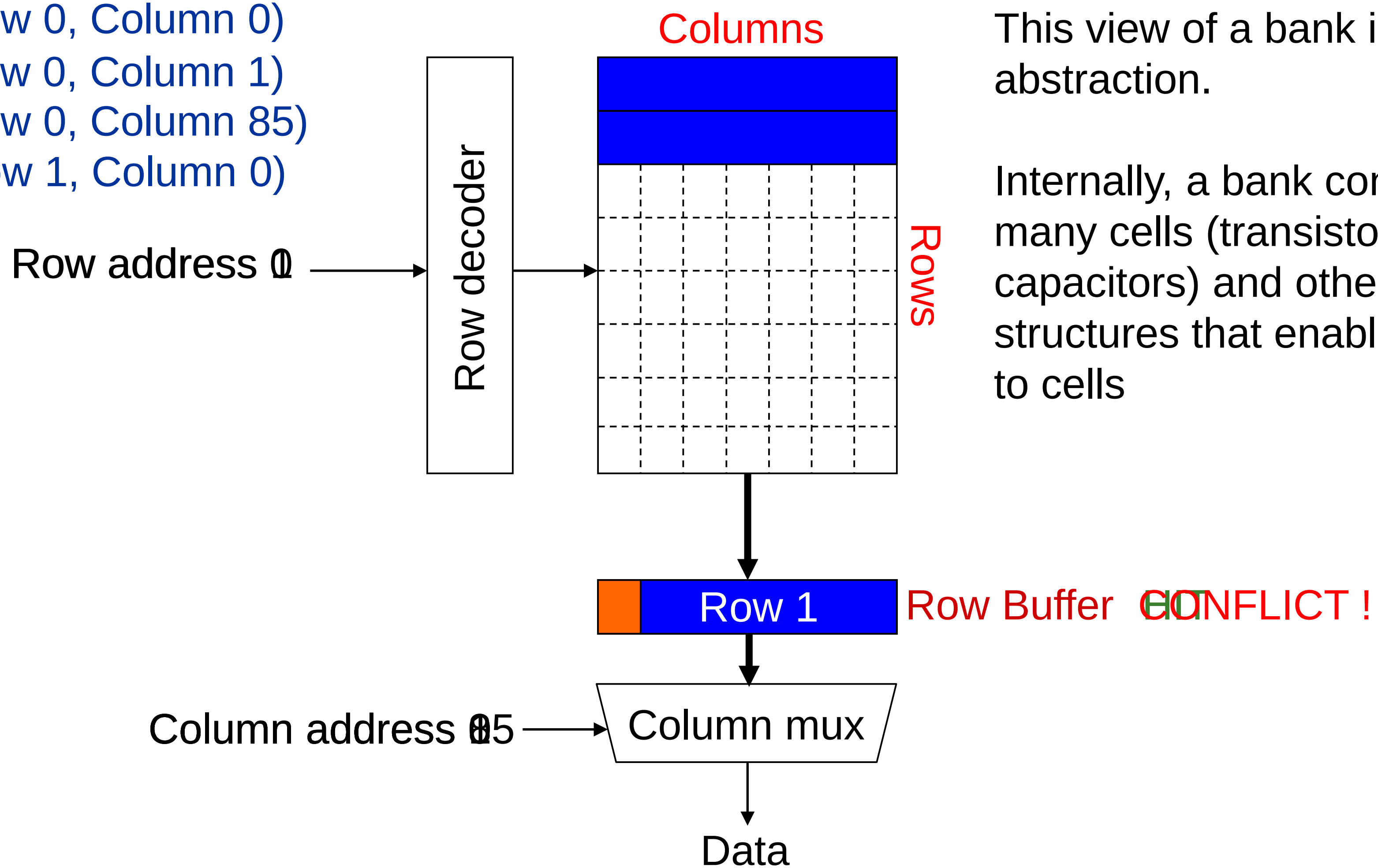
- We want the system to be controllable **and** high performance

# Why the Disparity in Slowdowns?



# Digging Deeper: DRAM Bank Operation

Access Address:
(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)




This view of a bank is an abstraction.

Internally, a bank consists of many cells (transistors & capacitors) and other structures that enable access to cells

# DRAM Controllers

- A row-conflict memory access takes significantly longer than a row-hit access
- Current controllers take advantage of this fact
- Commonly used scheduling policy (FR-FCFS) [Rixner 2000]*
  (1) Row-hit first: Service row-hit memory accesses first
  (2) Oldest-first: Then service older accesses first
- This scheduling policy aims to maximize DRAM throughput

*Rixner et al., "Memory Access Scheduling," ISCA 2000.
*Zuravleff and Robinson, "Controller for a synchronous DRAM …," US Patent 5,630,096, May 1997.

# The Problem

- Multiple applications share the DRAM controller
- DRAM controllers designed to maximize DRAM data throughput

- DRAM scheduling policies are unfair to some applications
  - Row-hit first: unfairly prioritizes apps with high row buffer locality
    - Threads that keep on accessing the same row
  - Oldest-first: unfairly prioritizes memory-intensive applications

- DRAM controller vulnerable to denial of service attacks
  - Can write programs to exploit unfairness

# A Memory Performance Hog

```
// initialize large arrays A, B

for (j=0; j<N; j++) {
    index = j*linesize;      streaming
    A[index] = B[index];     (in sequence)
    …
}
```

```
// initialize large arrays A, B

for (j=0; j<N; j++) {
    index = rand();      random
    A[index] = B[index];
    …
}
```

**STREAM**

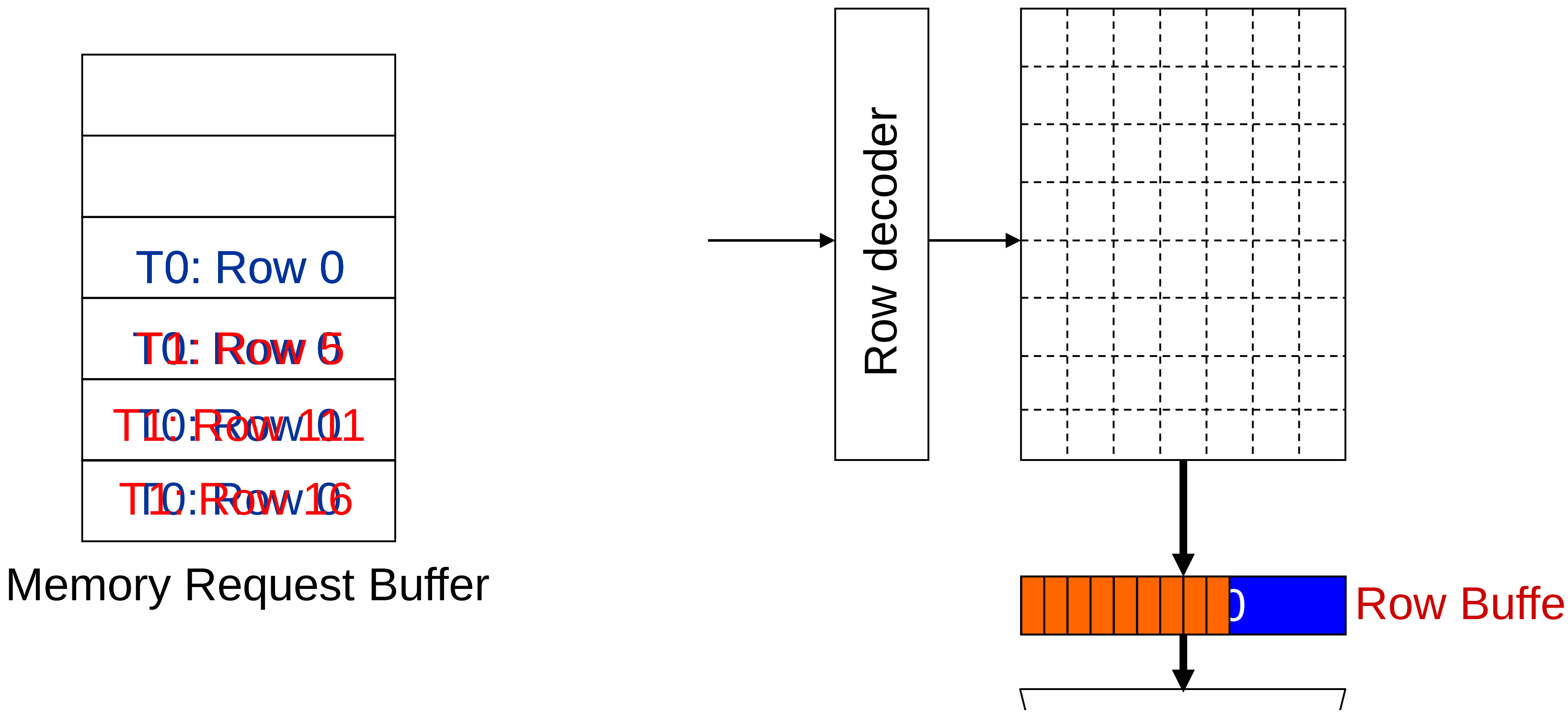
- Sequential memory access
- Very high row buffer locality (96% hit rate)
- Memory intensive

**RANDOM**

- Random memory access
- Very low row buffer locality (3% hit rate)
- Similarly memory intensive

Moscibroda and Mutlu, "Memory Performance Attacks," USENIX Security 2007.

# What Does the Memory Hog Do?




Row size: 8KB, request size: 64B

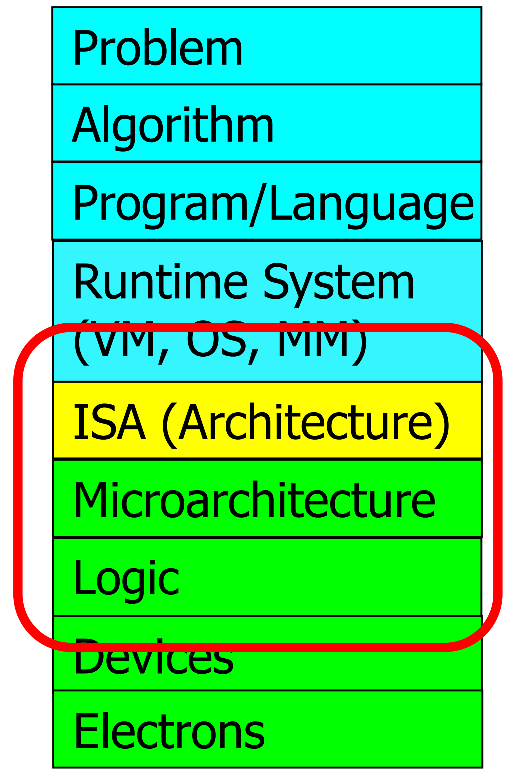
128 (8KB/64B) requests of STREAM serviced before a single request of RANDOM

Moscibroda and Mutlu, "Memory Performance Attacks," USENIX Security 2007.

# Now That We Know What Happens Underneath

- How would you solve the problem?

- What is the right place to solve the problem?
  - Programmer?
  - System software?
  - Compiler?
  - Hardware (Memory controller)?
  - Hardware (DRAM)?
  - Circuits?




- Two other goals of this course:
  - Enable you to think critically
  - Enable you to think broadly

# Reading on Memory Performance Attacks

- Thomas Moscibroda and Onur Mutlu,
  **"Memory Performance Attacks: Denial of Memory Service in Multi-Core Systems"**
  *Proceedings of the 16th USENIX Security Symposium* (***USENIX SECURITY***), pages 257-274, Boston, MA, August 2007. Slides (ppt)

- One potential reading for your Homework 1 assignment

**Memory Performance Attacks:**
**Denial of Memory Service in Multi-Core Systems**

*Thomas Moscibroda    Onur Mutlu*
*Microsoft Research*
*{moscitho,onur}@microsoft.com*

# If You Are Interested … Further Readings

- Onur Mutlu and Thomas Moscibroda,
  **"Stall-Time Fair Memory Access Scheduling for Chip Multiprocessors"**
  *Proceedings of the 40th International Symposium on Microarchitecture (**MICRO**)*, pages 146-158, Chicago, IL, December 2007. Slides (ppt)

- Onur Mutlu and Thomas Moscibroda,
  **"Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems"**
  *Proceedings of the 35th International Symposium on Computer Architecture (**ISCA**)* [Slides (ppt)]

- Sai Prashanth Muralidhara, Lavanya Subramanian, Onur Mutlu, Mahmut Kandemir, and Thomas Moscibroda,
  **"Reducing Memory Interference in Multicore Systems via Application-Aware Memory Channel Partitioning"**
  *Proceedings of the 44th International Symposium on Microarchitecture (**MICRO**)*, Porto Alegre, Brazil, December 2011. Slides (pptx)

# Takeaway

Breaking the abstraction layers (between components and transformation hierarchy levels)

and knowing what is underneath

enables you to **understand** and **solve** problems

# Another Example

- DRAM Refresh

# Computer Architecture
## Lecture 1: Introduction and Basics

Prof. Onur Mutlu
ETH Zürich
Fall 2020
17 September 2020