

# Computer Architecture

## Lecture 1: Introduction and Basics

Prof. Onur Mutlu

ETH Zürich

Fall 2021

30 September 2021

# 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](mailto:omutlu@gmail.com) (Best way to reach me)
- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>

## ■ Research and Teaching in:

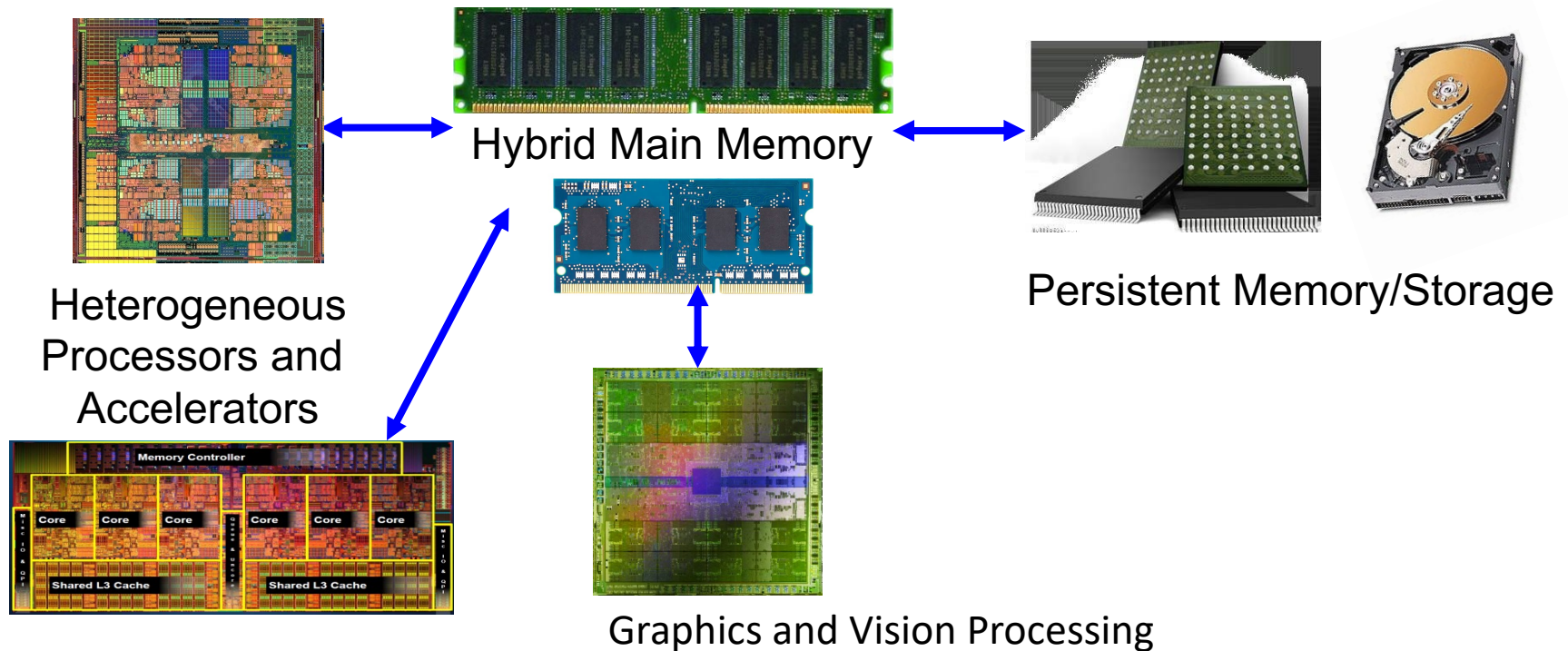
- ❑ Computer architecture, computer systems, hardware security, bioinformatics
- ❑ Memory and storage systems
- ❑ Hardware security, safety, predictability
- ❑ Fault tolerance
- ❑ Hardware/software cooperation
- ❑ Architectures for bioinformatics, health, medicine
- ❑ ...



# Current Research Mission

---

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



**Build fundamentally better architectures**

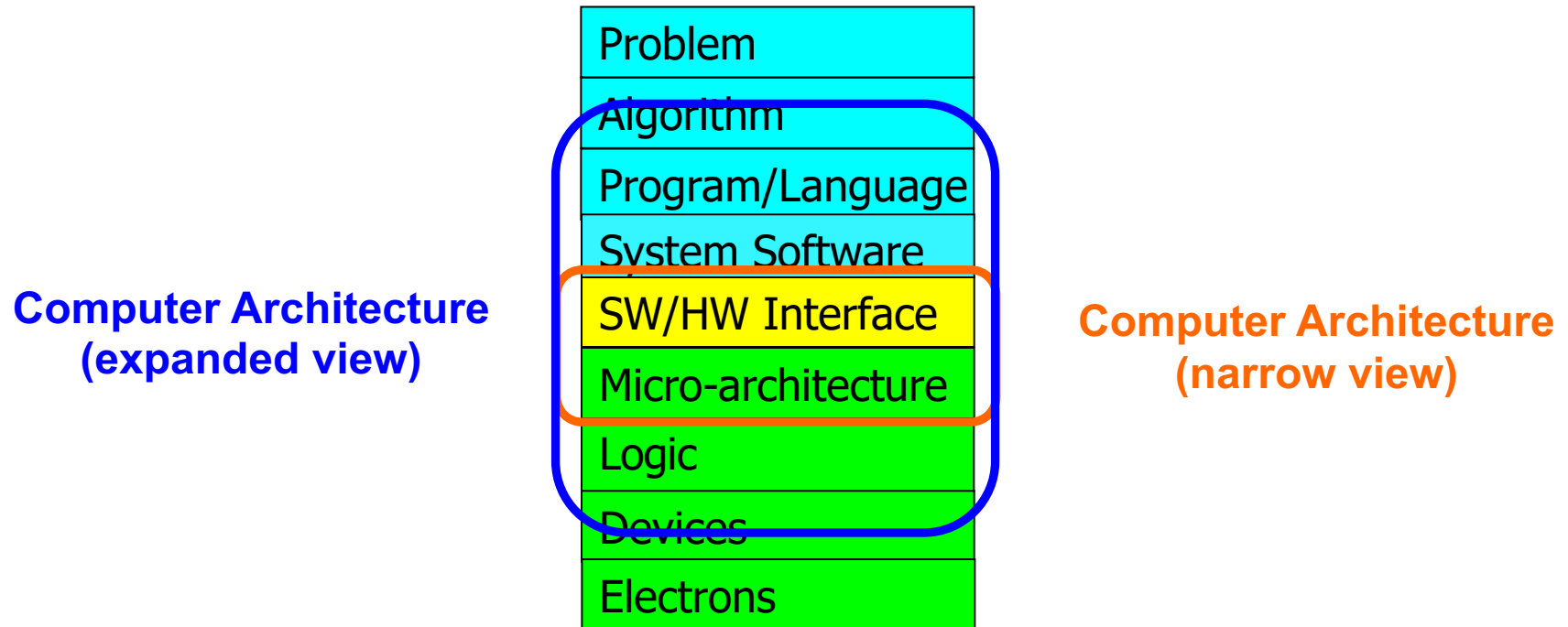
# Four Key Current Directions

---

- 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

---

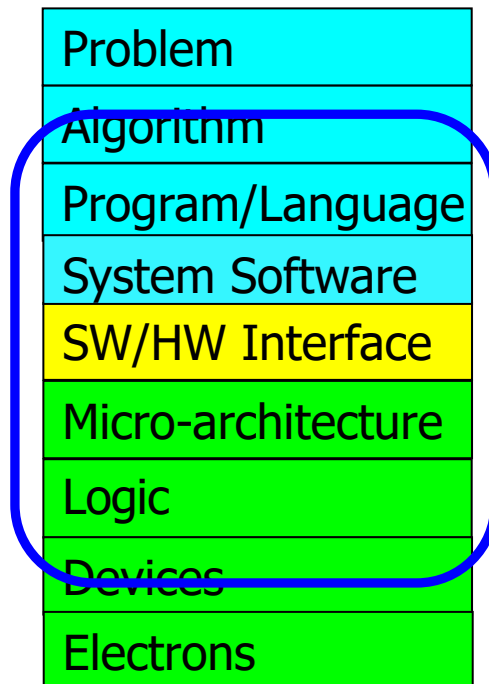


# Axiom

---

To achieve the highest **energy efficiency** and **performance**:

**we must take the expanded view**  
of computer architecture

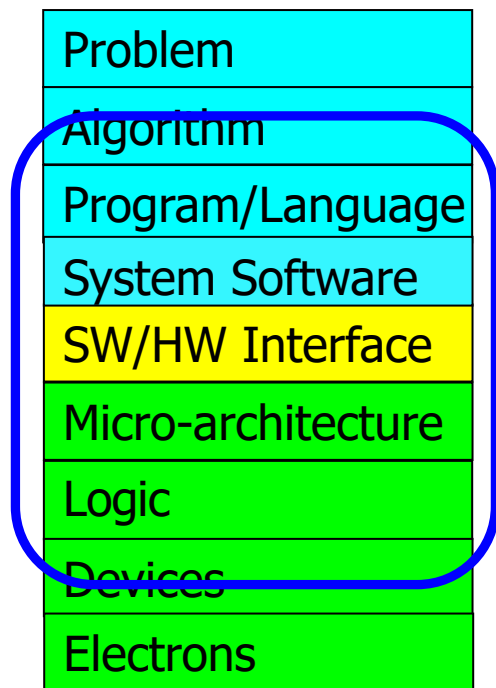


**Co-design across the hierarchy:**  
**Algorithms to devices**

**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

# SAFARI

*SAFARI Research Group*

*safari.ethz.ch*

Think BIG, Aim HIGH!

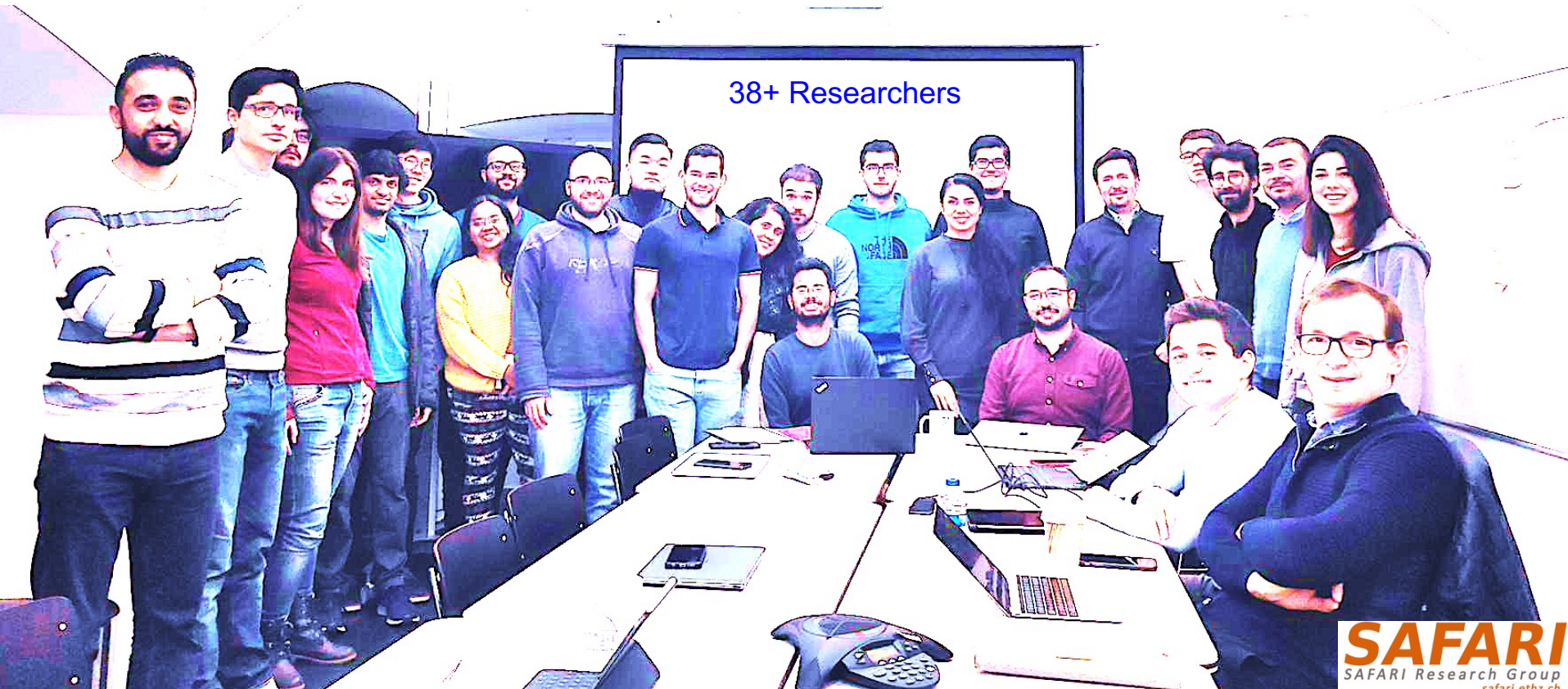
<https://safari.ethz.ch>



# 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!

**SAFARI**

<https://safari.ethz.ch>

# SAFARI Newsletter January 2021 Edition

- <https://safari.ethz.ch/safari-newsletter-january-2021/>



**SAFARI**  
SAFARI Research Group

Newsletter  
January 2021

*Think Big, Aim High, and  
Have a Wonderful 2021!*



Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 [here](#)). 2020 has



# SAFARI PhD and Post-Doc Alumni

---

- <https://safari.ethz.ch/safari-alumni/>
- Nastaran Hajinazar (ETH Zurich)
- Gagandeep Singh (ETH Zurich)
- Amirali Boroumand (Stanford Univ)
- Jeremie Kim (ETH Zurich)
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor)
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook)
- Mohammed Alser (ETH Zurich)
- Yixin Luo (Google)
- Kevin Chang (Facebook)
- Rachata Ausavarungnirun (KMUNTB, Assistant Professor)
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor)
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher)
- Yoongu Kim (Google)
- Lavanya Subramanian (Intel Labs → Facebook)
  
- Samira Khan (Univ. of Virginia, Assistant Professor)
- Saugata Ghose (Univ. of Illinois, Assistant Professor)
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)

# SAFARI Research Group: Introduction and Research

---

- Onur Mutlu,  
**"SAFARI Research Group: Introduction & Research"**  
*Talk at ETH Future Computing Laboratory Welcome  
Workshop (**EFCL**), Virtual, 6 July 2021.*  
[\[Slides \(pptx\) \(pdf\)\]](#)

# A Talk on Impactful Research & Teaching



The video player shows a presentation slide with the following content:

Applying to Grad School  
& Doing Impactful Research

Onur Mutlu  
[omutlu@gmail.com](mailto:omutlu@gmail.com)  
<https://people.inf.ethz.ch/omutlu>  
13 June 2020  
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

Logos at the bottom of the slide: SAFARI, ETH zürich, Carnegie Mellon.

Below the video player, the YouTube interface shows:

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu  
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures  
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021  
(<https://sites.google.com/wisc.edu/uar...>)

Engagement icons: 74 likes, 1 comment, SHARE, SAVE, and a menu icon.

Buttons: ANALYTICS, EDIT VIDEO

# Principle: Teaching and Research

---

...

Teaching drives Research

Research drives Teaching

...



Onur Mutlu Lectures

16.9K subscribers

CUSTOMIZE CHANNEL

MANAGE VIDEOS

HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

ABOUT



Popular uploads ▶ PLAY ALL

**How Computers Work (from the ground up)**

1:33:25

**Digital Design & Computer Architecture: Lecture 1:...**

49K views • 1 year ago

**Computer Architecture - Lecture 1: Introduction and...**

36K views • 3 years ago

**Computer Architecture - Lecture 1: Introduction and...**

31K views • 1 year ago

**Computer Architecture - Lecture 1: Introduction and...**

30K views • 8 months ago

**Design of Digital Circuits - Lecture 1: Introduction and...**

22K views • 2 years ago

**Computer Architecture - Lecture 2: Fundamentals,...**

17K views • 3 years ago

### First Course in Computer Architecture & Digital Design 2021-2013

**Livestream - Digital Design and Computer Architecture - ETH...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Digital Design & Computer Architecture - ETH Zürich...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Design of Digital Circuits - ETH Zürich - Spring 2019**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Design of Digital Circuits - ETH Zürich - Spring 2018**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Digital Circuits and Computer Architecture - ETH Zurich --...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Spring 2015 -- Computer Architecture Lectures --...**

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

### Advanced Computer Architecture Courses 2020-2012

**Computer Architecture - ETH Zürich - Fall 2020**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Computer Architecture - ETH Zürich - Fall 2019**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Computer Architecture - ETH Zürich - Fall 2018**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Computer Architecture - ETH Zürich - Fall 2017**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Fall 2015 - 740 Computer Architecture**

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

**Fall 2013 - 740 Computer Architecture - Carnegie Mellon**

Carnegie Mellon Computer Archite...

VIEW FULL PLAYLIST

### Special Courses on Memory Systems

**Memory Technology Lectures**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Champéry Winter School 2020 - Memory Systems and Memory...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**Perugia NiPS Summer School 2019**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**SAMOS Tutorial 2019 - Memory Systems**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**TU Wien 2019 - Memory Systems and Memory-Centric...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

**ACACES 2018 Lectures -- Memory Systems and Memory...**

Onur Mutlu Lectures

VIEW FULL PLAYLIST

Research Talks <https://www.youtube.com/onurmutlulectures>

# Online Courses & Lectures

---

## ■ **First Computer Architecture & Digital Design Course**

- Digital Design and Computer Architecture
- Spring 2021 Livestream Edition:  
[https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi\\_uej3aY39YB5pfW4SJ7LIN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

## ■ **Advanced Computer Architecture Course**

- Computer Architecture
- Fall 2020 Edition:  
<https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN>

# DDCA (Spring 2021)



Trace: · schedule

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

<https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule>

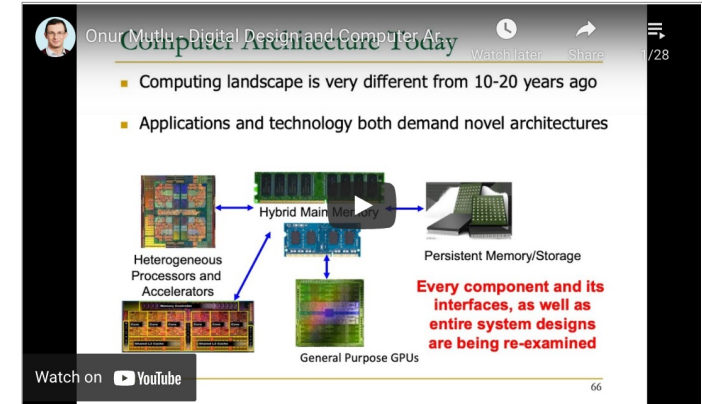
[https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi\\_uej3aY39YB5pfW4SJ7LIN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

## Bachelor's course

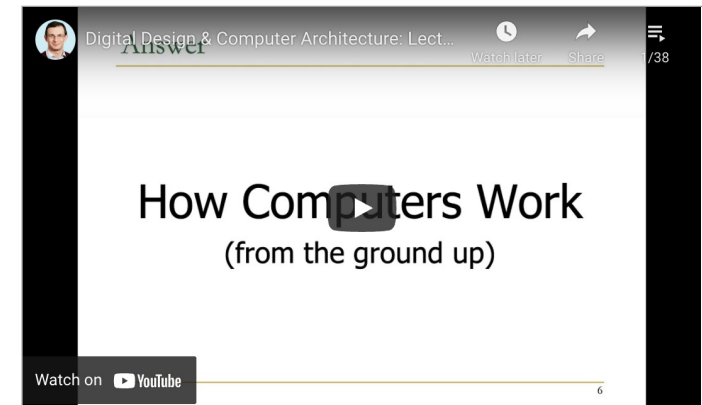
- ❑ 2<sup>nd</sup> semester at ETH Zurich
- ❑ Rigorous introduction into "How Computers Work"
- ❑ Digital Design/Logic
- ❑ Computer Architecture
- ❑ 10 FPGA Lab Assignments

## Lecture Video Playlist on YouTube

Livestream Lecture Playlist



Recorded Lecture Playlist



## Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics 02:00 (PDF) 02:00 (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset 02:00 (PDF) 02:00 (PPT)	Required		
			L2b: Mysteries in Computer Architecture 02:00 (PDF) 02:00 (PPT)	Required Suggested Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II 02:00 (PDF) 02:00 (PPT)	Required Suggested Mentioned		



# Comp Arch (Fall 2020)



Trace: start · schedule

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

Resources

- Computer Architecture FS19: Course Webpage
- Computer Architecture FS19: Lecture Videos
- Digitaltechnik SS20: Course Webpage
- Digitaltechnik SS20: Lecture Videos
- Moodle
- Piazza (Q&A)
- HotCRP
- Verilog Practice Website (HDLBits)

<https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule>

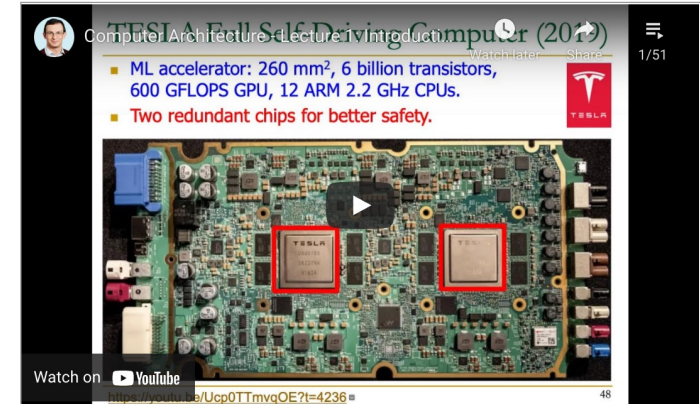
<https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN>

## Master's level course

- ❑ Taken by Bachelor's/Masters/PhD students
- ❑ Cutting-edge research topics + fundamentals in Computer Architecture
- ❑ 5 Simulator-based Lab Assignments
- ❑ Potential research exploration
- ❑ Many research readings

## Lecture Video Playlist on YouTube

Lecture Playlist



## Fall 2020 Lectures & Schedule

Week	Date	Lecture	Readings	Lab	HW
W1	17.09 Thu.	<b>L1: Introduction and Basics</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		HW 0 Out
		<b>L2a: Memory Performance Attacks</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested	Lab 1 Out	
	18.09 Fri.	<b>L2b: Data Retention and Memory Refresh</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
		<b>L2c: Course Logistics</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>			
W2	24.09 Thu.	<b>L3a: Introduction to Genome Sequence Analysis</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		HW 1 Out
		<b>L3b: Memory Systems: Challenges and Opportunities</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
	25.09 Fri.	<b>L4a: Memory Systems: Solution Directions</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
		<b>L4b: RowHammer</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
W3	01.10 Thu.	<b>L5a: RowHammer in 2020: TRRespass</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
		<b>L5b: RowHammer in 2020: Revisiting RowHammer</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described Suggested		
		<b>L5c: Secure and Reliable Memory</b> <a href="#">CORA (PDF)</a> <a href="#">PPT</a> <a href="#">YouTube</a> <a href="#">Video</a>	Described		




# Seminar (Spring'21)

■ [https://safari.ethz.ch/architecture\\_seminar/spring2021/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/spring2021/doku.php?id=schedule)

■ [https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi\\_awYdjmWVIUegsbY7TPGW4](https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi_awYdjmWVIUegsbY7TPGW4)

- Critical analysis course
  - ❑ Taken by Bachelor's/Masters/PhD students
  - ❑ Cutting-edge research topics + fundamentals in Computer Architecture
  - ❑ 20+ research papers, presentations, analyses



Seminar in Computer Architecture - Spring 2021

Recent Changes

Media Manager

Sitemap

Trace: [start](#) - [schedule](#)

Home

Materials

- Announcements
- Lectures/Schedule**
- Lecture Buzzwords
- Readings
- Sessions
- Papers
- Synthesis Report
- Homework

Past Course Materials

- Fall 2020
- Spring 2020
- Fall 2019
- Spring 2019

Resources

Computer Architecture

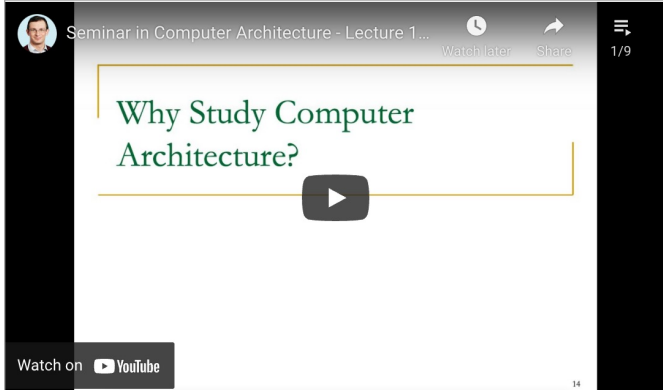
- Fall 2020
- Fall 2020: Lecture Videos
- Fall 2019
- Fall 2019: Lecture Videos
- Fall 2018
- Fall 2018: Lecture Videos

Digital Design and Computer Architecture

- Spring 2020
- Spring 2020: Lecture Videos
- Spring 2019
- Spring 2019: Lecture Videos

Lecture Video Playlist on YouTube

Lecture Playlist



Watch on YouTube

Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Assignments
W1	25.02 Thu.		<b>L1a: Introduction and Basics</b> <a href="#">PDF</a> <a href="#">PPT</a>	Suggested	
			Optional Lecture: Design Fundamentals <a href="#">PDF</a> <a href="#">PPT</a>		
			<b>L1b: Course Logistics</b> <a href="#">PDF</a> <a href="#">PPT</a>	Suggested	
W2	04.03 Thu.		<b>L2: Example Review: RowClone</b> <a href="#">PDF</a> <a href="#">PPT</a>	Suggested	
W3	11.03 Thu.		<b>L3: Example Review: Memory Channel Partitioning</b> <a href="#">PDF</a> <a href="#">PPT</a>	Suggested	
W4	18.03 Thu.		<b>L4: Example Review: GateKeeper</b> <a href="#">PDF</a> <a href="#">PPT</a>	Suggested	
W5	25.03 Thu.		<b>S1.1: Spectre Attacks: Exploiting Speculative Execution, S&amp;P 2019</b> <a href="#">PPT</a> <a href="#">PDF</a>	Mentioned	
			<b>S1.2: BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows, HPCA 2021</b> <a href="#">PPT</a> <a href="#">PDF</a>		
W6	01.04 Thu.		<b>S2.1: D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput, HPCA 2019</b> <a href="#">PPT</a> <a href="#">PDF</a>		
			<b>S2.2: ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs, MICRO 2019</b> <a href="#">PPT</a> <a href="#">PDF</a>	Mentioned	
W7	15.04 Thu.		<b>S3.1: PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture,</b>	Mentioned	

# Hands-On Projects & Seminars Courses

- [https://safari.ethz.ch/projects\\_and\\_seminars/doku.php](https://safari.ethz.ch/projects_and_seminars/doku.php)



SAFARI Project & Seminars Courses  
(Spring 2021)



[Recent Changes](#) [Media Manager](#) [Sitemap](#)

Trace: • [start](#)

[Home](#)

Projects

- [SoftMC](#)
- [Ramulator](#)
- [Accelerating Genomics](#)
- [Mobile Genomics](#)
- [Processing-in-Memory](#)
- [Heterogeneous Systems](#)
- [SSD Simulator](#)

start

## SAFARI Projects & Seminars Courses (Spring 2021)

Welcome to the wiki for Project and Seminar courses SAFARI offers.

### Courses we offer:

- Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments
- Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator
- Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
- Genome Sequencing on Mobile Devices
- Exploring the Processing-in-Memory Paradigm for Future Computing Systems
- Hands-on Acceleration on Heterogeneous Computing Systems
- Understanding and Designing Modern NAND Flash-Based Solid-State Drives (SSDs) by Building a Practical SSD Simulator

# Principle: Insight and Ideas

---

Focus on Insight

Encourage New Ideas

# Principle: Learning and Scholarship

---

Focus on  
learning and scholarship

# SAFARI Live Seminars (Past Talks)

## SAFARI Live Seminars in Computer Architecture

Dr. Juan Gómez Luna, ETH Zurich

Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

**SAFARI**  
SAFARI Research Group

12 Mon Jul 2021

## SAFARI Live Seminars in Computer Architecture

Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems

An Addition to Low Cost Per Memory Bit – How to Recognize it and What to Do About it

**SAFARI**  
SAFARI Research Group

19 Mo Jul 2021

## SAFARI Live Seminars in Computer Architecture

Geraldo F. Oliveira, ETH Zurich

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

**SAFARI**  
SAFARI Research Group

22 Do Jul 2021

**Near-Data Processing (2/2)**

**UPMEM (2019)** **Samsung HBM-PIM (2021)**

Near-DRAM-banks processing for general-purpose computing

Near-DRAM-banks processing for neural networks

0.9 TOPS compute throughput<sup>1</sup> 1.2 TFLOPS compute throughput<sup>2</sup>

The goal of Near-Data Processing (NDP) is to mitigate data movement

**SAFARI**

## SAFARI Live Seminars in Computer Architecture

Gennady Pekhimenko, University of Toronto

Efficient DNN Training at Scale: from Algorithms to Hardware

**SAFARI**  
SAFARI Research Group

5 Do Aug 2021

**DNN Training vs. Inference**

Step 1 - Forward Pass (makes a prediction)  
Step 2 - Backward Pass (calculates error gradients)

Generated in the forward pass Used in the backward pass

DNN training requires stashing feature maps for the backward pass (not required in inference)

## SAFARI Live Seminars in Computer Architecture

Jawad Haj-Yahya, Huawei Research Center Zurich

Power Management Mechanisms in Modern Microprocessors and Their Security Implications

**SAFARI**  
SAFARI Research Group

16 Mo Aug 2021

**Overview of a Modern SoC Architecture**

- 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
- Several voltage sources exist, and some of them are shared between domains
- IO controllers and engines, IO interconnect, memory controller, and DDRIO typically each has an independent clock

## SAFARI Live Seminars in Computer Architecture

Ataberk Olgun, TOBB & ETH Zurich

QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips

**SAFARI**  
SAFARI Research Group

15 Mi Sep 2021

**Using QUAC to Generate Random Values**

Use QUAC to activate DRAM rows that are initialized with conflicting data (e.g., two '1's and two '0's) to generate random values

**ACT** **PRE** **ACT**

**SAFARI** **kasirga**

## SAFARI Live Seminars in Computer Architecture

Minesh Patel, ETH Zurich

Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

**SAFARI**  
SAFARI Research Group

21 Tues Sep 2021

**Position Paper (Ongoing)** Arguing for increased transparency of DRAM reliability characteristics

**REAPER (ISCA'17)** Understand the basic properties of DRAM data-retention errors

**BEER (MICRO'20, best paper)** Determine exactly how on-die ECCs offuscate error characteristics

**HARP (MICRO'21)** Understand how errors appear and how to identify at-risk bits

**EIN (DSN'19, best paper)** Understand and recover the error characteristics beneath on-die ECC

## SAFARI Live Seminars in Computer Architecture

Christina Giannoula, National Technical University of Athens

Efficient Synchronization Support for Near-Data-Processing Architectures

**SAFARI**  
SAFARI Research Group

27 Mo Sep 2021

**NDP Synchronization Solution Space**

**Shared Memory**

- Hardware Cache Coherence
- Remote Atomics
- Specialized Hardware Support

**Message-passing**

- Software-based Schemes
- Specialized Hardware Support

**NDP Systems:** SynCron (HPCA'21)

# SAFARI Live Seminars (Upcoming Talk)

The image is a YouTube video player thumbnail. At the top left is a circular profile picture of a man. To its right is the text 'SAFARI Live Seminar: Security Implications of Power Managemen...'. At the top right are icons for 'Watch later' (clock) and 'Share' (arrow). The main title 'IChannels' is in large, bold, black font. Below it is the subtitle 'Exploiting Current Management Mechanisms to Create Covert Channels in Modern Processors' in a smaller, bold, black font. In the center is a play button icon over a video frame. Below the play button is the name 'Jawad Haj-Yahya' in blue, underlined text. Below that are the names of the speakers: 'Jeremie S. Kim', 'A. Giray Yağlıkçı', 'Ivan Puddu', 'Lois Orosa', 'Juan Gómez Luna', 'Mohammed Alser', and 'Onur Mutlu'. At the bottom are the logos for 'SAFARI' (in orange) and 'ETH zürich' (in black). At the bottom left is a 'Watch on YouTube' button.

SAFARI Live Seminar: Security Implications of Power Managemen...

**IChannels**

**Exploiting Current Management Mechanisms  
to Create Covert Channels in Modern Processors**

Jawad Haj-Yahya

Jeremie S. Kim   A. Giray Yağlıkçı   Ivan Puddu   Lois Orosa  
Juan Gómez Luna   Mohammed Alser   Onur Mutlu

**SAFARI**   **ETH** zürich

Watch on YouTube

## SAFARI Live Seminar: Jawad Haj-Yahya 4 October 2021

Posted on September 18, 2021 by ewent

Join us for our [SAFARI Live Seminar](#) with [Jawad Haj-Yahya](#).

**Monday, October 4 at 5:30 pm Zurich time (CEST)**

**Security Implications of Power Management Mechanisms In Modern Processors, Current Studies and Future Trends**

[Jawad Haj-Yahya](#), Huawei Research Center Zurich


<https://safari.ethz.ch/safari-seminar-series/>

# Open-Source Artifacts

**<https://github.com/CMU-SAFARI>**



# Open Source Tools: SAFARI GitHub



## SAFARI Research Group at ETH Zurich and Carnegie Mellon University


Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.


📍 ETH Zurich and Carnegie Mellon ... 🔗 <https://safari.ethz.ch/> ✉ [omutlu@gmail.com](mailto:omutlu@gmail.com)


[🏠 Overview](#) [💻 Repositories 55](#) [📦 Packages](#) [👤 People 40](#) [👥 Teams 1](#) [📁 Projects](#) [⚙ Settings](#)

### Pinned

Customize your pins

 **ramulator** Public ⋮  
A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...  
● C++ ☆ 250 🍴 130

 **prim-benchmarks** Public ⋮  
PRIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PRIM is developed to evaluate, analyze, and characterize the first publ...  
● C ☆ 18 🍴 8

 **DAMOV** Public ⋮  
DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processin...  
● C++ ☆ 12 🍴 1

### 📁 Repositories

Type ▾ Language ▾ Sort ▾ New

**Pythia**  
A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.  
● C++ ☆ 0 🍴 1 🔄 0 📄 0 Updated yesterday

**BurstLink**  
☆ 0 🍴 0 🔄 0 📄 0 Updated 21 days ago

<https://github.com/CMU-SAFARI/>

26



# Some Open Source Tools (I)

---


- Rowhammer – Program to Induce RowHammer Errors
  - <https://github.com/CMU-SAFARI/rowhammer>
- Ramulator – Fast and Extensible DRAM Simulator
  - <https://github.com/CMU-SAFARI/ramulator>
- MemSim – Simple Memory Simulator
  - <https://github.com/CMU-SAFARI/memsim>
- NOCulator – Flexible Network-on-Chip Simulator
  - <https://github.com/CMU-SAFARI/NOCulator>
- SoftMC – FPGA-Based DRAM Testing Infrastructure
  - <https://github.com/CMU-SAFARI/SoftMC>
- Other open-source software from my group
  - <https://github.com/CMU-SAFARI/>

# Some Open Source Tools (II)

---

- MQSim – A Fast Modern SSD Simulator
  - <https://github.com/CMU-SAFARI/MQSim>
- Mosaic – GPU Simulator Supporting Concurrent Applications
  - <https://github.com/CMU-SAFARI/Mosaic>
- IMPICA – Processing in 3D-Stacked Memory Simulator
  - <https://github.com/CMU-SAFARI/IMPICA>
- SMLA – Detailed 3D-Stacked Memory Simulator
  - <https://github.com/CMU-SAFARI/SMLA>
- HWASim – Simulator for Heterogeneous CPU-HWA Systems
  - <https://github.com/CMU-SAFARI/HWASim>
- Other open-source software from my group
  - <https://github.com/CMU-SAFARI/>

# More Open Source Tools (III)




## SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

📍 ETH Zurich and Carnegie Mellon ... 🔗 <https://safari.ethz.ch/> ✉ [omutlu@gmail.com](mailto:omutlu@gmail.com)


[🏠 Overview](#) [💻 Repositories 55](#) [📦 Packages](#) [👤 People 40](#) [👥 Teams 1](#) [📁 Projects](#) [⚙ Settings](#)

### Pinned

**ramulator** Public ⋮


A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

● C++ ☆ 250 🍴 130

**prim-benchmarks** Public ⋮

PRIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PRIM is developed to evaluate, analyze, and characterize the first publ...

● C ☆ 18 🍴 8

**DAMOV** Public ⋮

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processin...

● C++ ☆ 12 🍴 1

### 📁 Repositories

Type ▾ Language ▾ Sort ▾ New

**Pythia**

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.

● C++ ☆ 0 🍴 1 🔄 0 📄 0 Updated yesterday

**BurstLink**

☆ 0 🍴 0 🔄 0 📄 0 Updated 21 days ago

## Pythia

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.

machine-learning reinforcement-learning prefetcher cache-replacement  
branch-predictor champsim-simulator champsim-tracer

● C++ 1 0 0 0 Updated yesterday

## BurstLink

0 0 0 0 Updated 21 days ago

## MIG-7-PHY-DDR3-Controller

A DDR3 Controller that uses the Xilinx MIG-7 PHY to interface with DDR3 devices.

● Verilog 1 1 0 0 Updated on Aug 22

## Pythia-HDL

Implementation of Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning in Chisel HDL.

machine-learning scala reinforcement-learning chisel chisel3 firrtl hdl

● Scala 8 MIT 0 0 0 0 Updated on Jul 31

## HARP

Private

0 0 0 0 Updated on Jul 31

## EINSim

DRAM error-correction code (ECC) simulator incorporating statistical error properties and DRAM design characteristics for inferring pre-correction error characteristics using only the post-correction errors. Described in the 2019 DSN paper by Patel et al.: [https://people.inf.ethz.ch/omutlu/pub/understanding-and-modeling-in-DRAM-ECC\\_dsn19.pdf](https://people.inf.ethz.ch/omutlu/pub/understanding-and-modeling-in-DRAM-ECC_dsn19.pdf).

simulator reliability statistical-inference dram error-correcting-codes  
map-estimation error-correction

● C++ 8 MIT 0 5 0 0 Updated on Jul 29

## DAMOV

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at <https://arxiv.org/pdf/2105.03725.pdf>)

● C++ 1 12 1 0 Updated on Jul 13

## MetaSys

Metasys is the first open-source FPGA-based infrastructure with a prototype in a RISC-V core, to enable the rapid implementation and evaluation of a wide range of cross-layer software/hardware cooperative techniques techniques in real hardware. Described in our pre-print: <https://arxiv.org/abs/2105.08123>

1 0 0 0 Updated on Jul 9

## NATSA

NATSA is the first near-data-processing accelerator for time series analysis based on the Matrix Profile (SCRIMP) algorithm. NATSA exploits modern 3D-stacked High Bandwidth Memory (HBM) to enable efficient and fast matrix profile computation near memory. Described in ICCD 2020 by Fernandez et al.

[https://people.inf.ethz.ch/omutlu/pub/NATSA\\_time-...](https://people.inf.ethz.ch/omutlu/pub/NATSA_time-...)

accelerator hbm time-series-analysis matrix-profile near-data-processing  
scrimp

● C++ 1 4 0 0 Updated on Jun 28

## COVIDHunter

COVIDHunter: An accurate and flexible COVID-19 outbreak simulation model that forecasts the strength of future mitigation measures and the numbers of cases, hospitalizations, and deaths for a given day, while considering the potential effect of environmental conditions. Described by Alser et al. (preliminary version at <https://arxiv.org/abs/2...>

simulation epidemiology covid-19 covid-19-data covid-19-tracker  
reproduction-number covidhunter

● Swift 8 MIT 1 5 0 0 Updated on Jun 27

## prim-benchmarks

PRIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PRIM is developed to evaluate, analyze, and characterize the first publicly-available real-world PIM architecture, the UPMEM PIM architecture. Described by Gómez-Luna et al. (preliminary version at <https://arxiv.org/abs/2...>

● C 8 MIT 8 18 0 0 Updated on Jun 16

## SNP-Selective-Hiding

An optimization-based mechanism to selectively hide the minimum number of overlapping SNPs among the family members who participated in the genomic studies (i.e. GWAS). Our goal is to distort the dependencies among the family members in the original database for achieving better privacy without significantly degrading the data utility.

gwas genomics data-privacy differential-privacy genomic-data-analysis  
laplace-distribution genomic-privacy

● MATLAB 0 0 0 0 Updated on Jun 16

## SneakySnake

SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude) expedites sequence alignment calculation for both short and long reads. Described in the Bioinformatics (2020) by Alser et al.

<https://arxiv.org/abs...>

fpga gpu smith-waterman needleman-wunsch sequence-alignment  
long-reads minimap2

● VHDL 8 GPL-3.0 6 35 0 0 Updated on May 12

## ramulator

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the IEEE CAL 2015 paper by Kim et al. at [http://users.ece.cmu.edu/~omutlu/pub/ramulator\\_dram\\_simulator-ieee-cal15.pdf](http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf)

● C++ 8 MIT 130 250 49 4 Updated on May 11

## GenASM

Source code for the software implementations of the GenASM algorithms proposed in our MICRO 2020 paper: Senol Cali et. al., "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis" at <https://people.inf.ethz.ch/omutlu/pub/GenASM-approximate-string-matching-framework-for-genome-analys...>

approximate-string-matching read-mapping hw-sw-co-design  
read-alignment bitap-algorithm pre-alignment-filtering  
genome-sequence-analysis

● C 8 GPL-3.0 3 20 0 0 Updated on Mar 22

## AirLift

AirLift is a tool that updates mapped reads from one reference genome to another. Unlike existing tools, it accounts for regions not shared between the two reference genomes.

# Papers, Talks, Videos, Artifacts

---

- All are openly available at

<https://people.inf.ethz.ch/omutlu/projects.htm>

<http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en>

<https://www.youtube.com/onurmutlulectures>

<https://github.com/CMU-SAFARI/>

## Principle: Environment of Freedom

---

Create an environment  
that values

free exploration,  
openness, collaboration,  
hard work, creativity

# My Suggestions to You

Suggestion to Researchers: Principle: Passion

---

**Follow Your Passion**  
**(Do not get derailed  
by naysayers)**

---



# Principle: Build Infrastructure

---

Build Infrastructure to  
Enable Your Passion

# Principle: Work Hard

---

Work Hard to  
Enable Your Passion

Suggestion to Researchers: Principle: Resilience

---

**Be Resilient**

---

# Principle: Learning and Scholarship

---

Focus on  
learning and scholarship

# Principle: Learning and Scholarship

---

The quality of your work  
defines your impact

# Principle: Good Mindset, Goals & Focus

---

You can make a  
good impact  
on the world

# Research & Teaching: Some Overview Talks

---

<https://www.youtube.com/onurmutlulectures>

## ■ Future Computing Architectures

- [https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=1](https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=1)

## ■ Enabling In-Memory Computation

- [https://www.youtube.com/watch?v=njX\\_14584Jw&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=16](https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=16)

## ■ Accelerating Genome Analysis

- [https://www.youtube.com/watch?v=r7sn41IH-4A&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=41](https://www.youtube.com/watch?v=r7sn41IH-4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=41)

## ■ Rethinking Memory System Design

- [https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=3](https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=3)

## ■ Intelligent Architectures for Intelligent Machines

- [https://www.youtube.com/watch?v=c6\\_LgzuNdkw&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=25](https://www.youtube.com/watch?v=c6_LgzuNdkw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=25)

## ■ The Story of RowHammer

- [https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBjI&index=39](https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBjI&index=39)



# An Interview on Research and Education

---

- **Computing Research and Education (@ ISCA 2019)**
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi\\_4oP9LdL3cc8G6NIjD2Ydz](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\\_5MGV6EnXEJHnV2YFBJI&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&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 \(pptx\) \(pdf\)\]](#)

# More Thoughts and Suggestions (II)

---

- Onur Mutlu,  
**"Computer Architecture: Why Is It So Important and Exciting Today?"**  
Invited Lecture at *Izmir Institute of Technology (IYTE)*, Virtual, 16 October 2020.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (2 hours 12 minutes)]
  
- Onur Mutlu,  
**"Applying to Graduate School & Doing Impactful Research"**  
Invited Panel Talk at *the 3rd Undergraduate Mentoring Workshop, held with the 48th International Symposium on Computer Architecture (ISCA)*, Virtual, 18 June 2021.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (50 minutes)]

# A Talk on Impactful Research & Teaching



The video player shows a presentation slide with the title "Applying to Grad School & Doing Impactful Research" in a green serif font, enclosed in a thin gold border. Below the title, the speaker's name "Onur Mutlu" is listed, followed by his email "omutlu@gmail.com" and his website "https://people.inf.ethz.ch/omutlu". The date "13 June 2020" and the event "Undergraduate Architecture Mentoring Workshop @ ISCA 2021" are also displayed. At the bottom of the slide, the logos for "SAFARI", "ETH zürich", and "Carnegie Mellon" are shown. The video player interface includes a progress bar at 0:27 / 50:31, a small video thumbnail of the speaker in the top right corner, and a bottom bar with engagement metrics (74 likes, 1 comment), share and save buttons, and a description of the panel talk at the Undergraduate Architecture Mentoring Workshop at ISCA 2021.

Applying to Grad School  
& Doing Impactful Research

Onur Mutlu  
[omutlu@gmail.com](mailto:omutlu@gmail.com)  
<https://people.inf.ethz.ch/omutlu>  
13 June 2020  
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

SAFARI ETH zürich Carnegie Mellon

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu  
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures  
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021  
(<https://sites.google.com/wisc.edu/uar...>)

## **Richard Hamming**

### **“You and Your Research”**

Transcription of the  
Bell Communications Research Colloquium Seminar  
7 March 1986

<https://safari.ethz.ch/architecture/fall2021/lib/exe/fetch.php?media=youandyourresearch.pdf>

# How to Approach This Course

---

“Formative Experience”

# How to Approach This Course

---

“High investment,  
high return”



# How to Approach This Course

---

“Recorded lectures  
allowed me to go over  
the lectures when  
necessary”

# How to Approach This Course

---

“YouTube allows me to  
watch the lectures on  
my TV”

# How to Approach This Course

---

“The lecturer is very responsive to questions and remarks from students”

# How to Approach This Course

---

“Perhaps even better than in-person classes as questions can be asked asynchronously”

# How to Approach This Course

---

“Easy to understand  
course format with  
homework, labs, and  
lectures”

# How to Approach This Course

---

“Paper reviews +  
assignments + labs,  
a really great plan to  
learn in a  
comprehensive way”

# How to Approach This Course

---

“the course was  
fantastic and I would  
do it again at any  
time”



# How to Approach This Course

---

Learning experience

Long-term tradeoff  
analysis

Critical thinking &  
decision making

# How to Approach This Course

---

Concepts & Ideas

Fundamentals

Cutting-edge

Hands-on learning

# How to Approach This Course

---

Your mindset  
will determine  
what you  
get out of the course

# Required Reading on Mindset & More

---

**Richard Hamming**

**“You and Your Research”**

Transcription of the  
Bell Communications Research Colloquium Seminar  
7 March 1986

<https://safari.ethz.ch/architecture/fall2021/lib/exe/fetch.php?media=youandyourresearch.pdf>

# Required Reading on Mindset & More

---

If you really want to be a first-class scientist you need to know yourself, your weaknesses, your strengths, and your bad faults, like my egotism. How can you convert a fault to an asset? How can you convert a situation where you haven't got enough manpower to move into a direction when that's exactly what you need to do? I say again that I have seen, as I studied the history, the successful scientist changed the viewpoint and what was a defect became an asset.

In summary, I claim that some of the reasons why so many people who have greatness within their grasp don't succeed are: they don't work on important problems, they don't become emotionally involved, they don't try and change what is difficult to some other situation which is easily done but is still important, and they keep giving themselves alibis why they don't. They keep saying that it is a matter of luck. I've told you how easy it is; furthermore I've told you how to reform. Therefore, go forth and become great scientists!



# 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

---



# Different Platforms, Different Goals



# Different Platforms, Different Goals

---





# Different Platforms, Different Goals

---





# Different Platforms, Different Goals





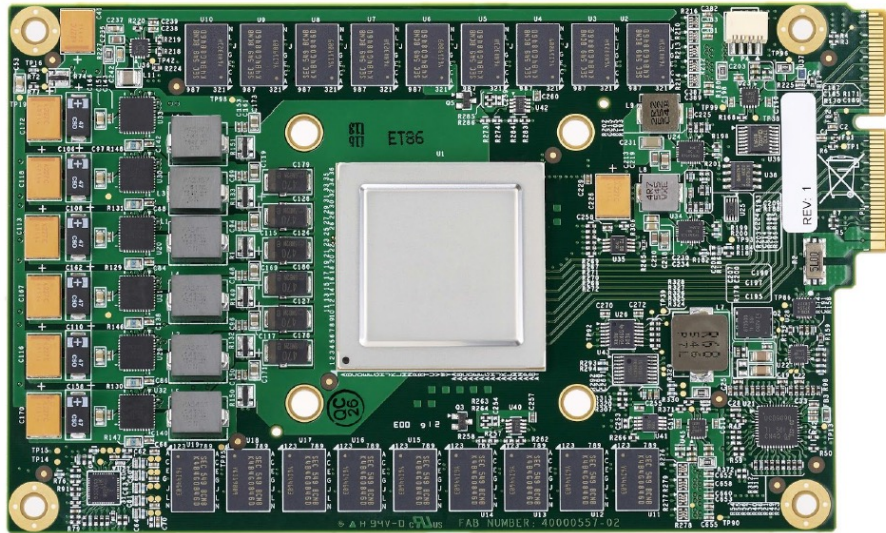
# Different Platforms, Different Goals

---

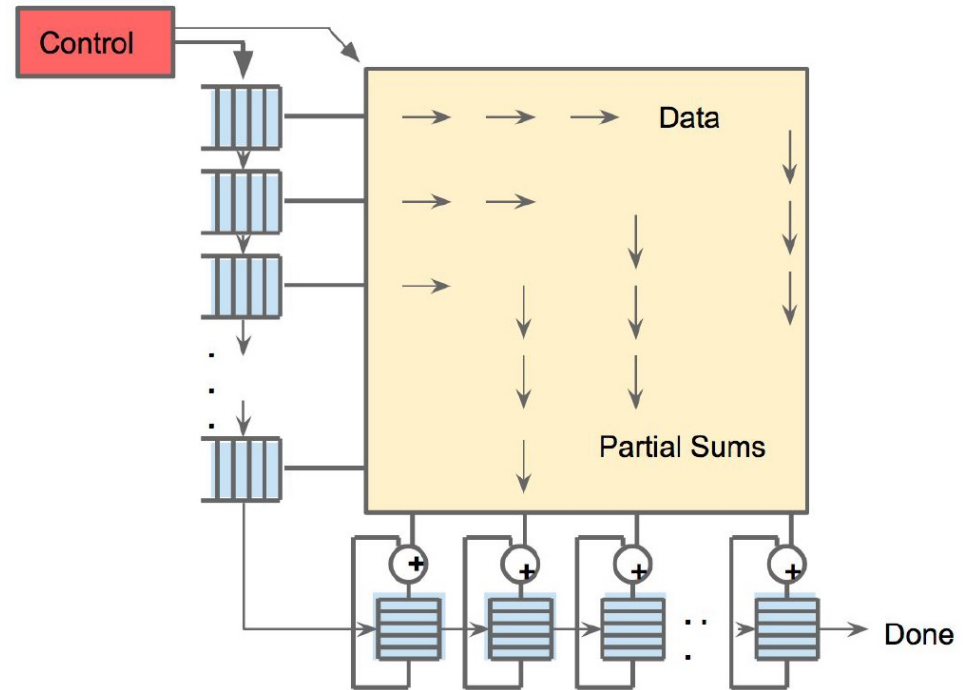


Jack Dongarra

# 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.

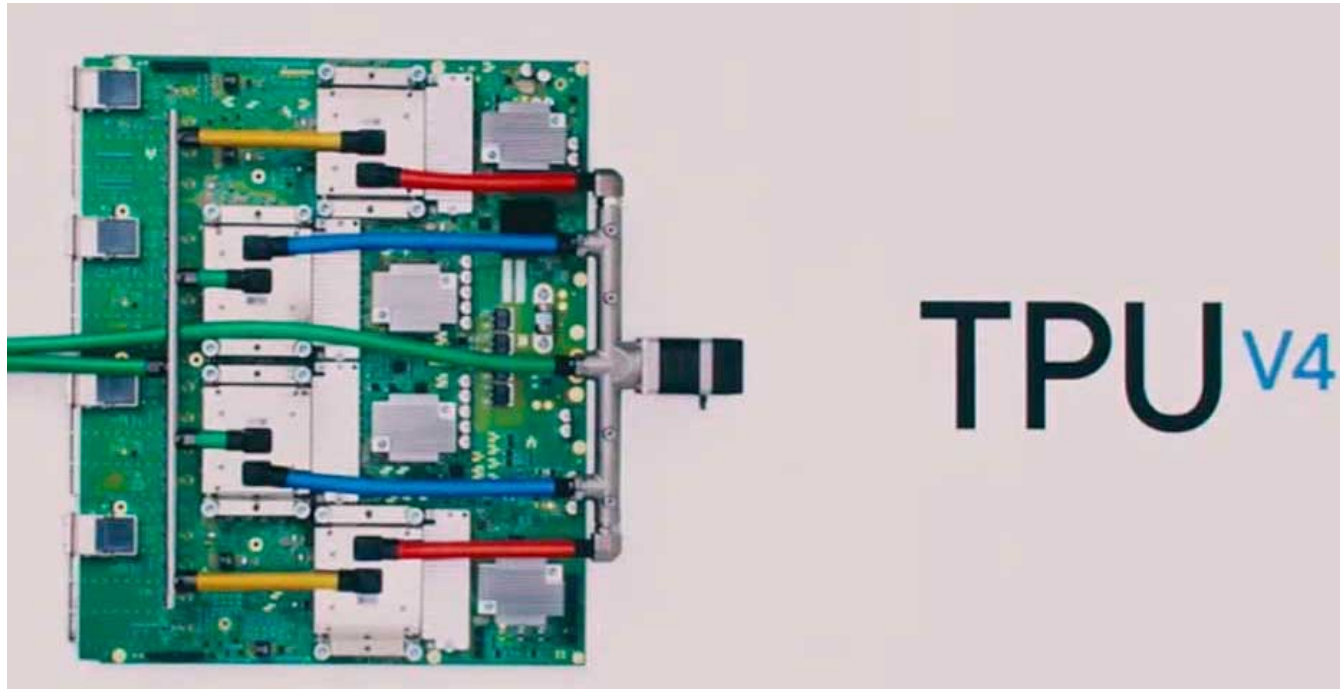


**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

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

# Different Platforms, Different Goals

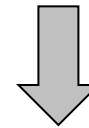
---



## 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



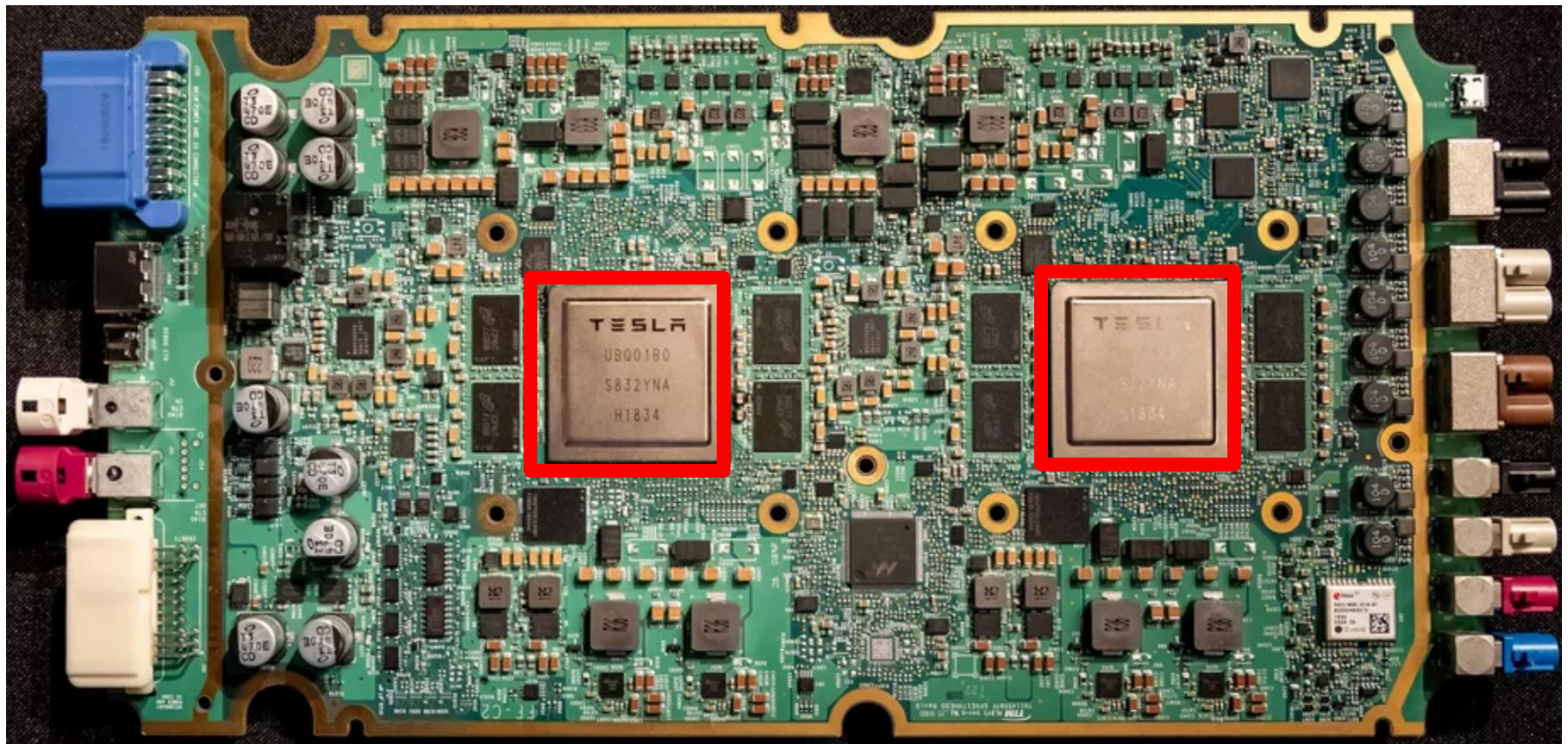
1 ExaFLOPS per board

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



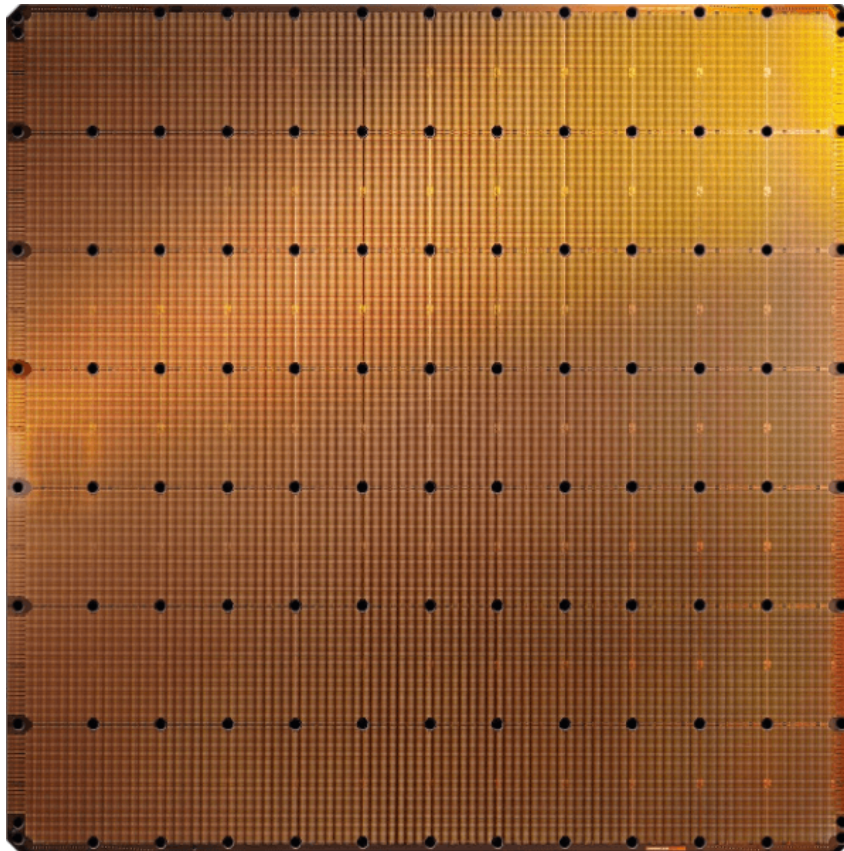
# Different Platforms, Different Goals

- 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.



# Different Platforms, Different Goals

---



**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/>



# Different Platforms, Different Goals

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#) IEEE Micro, August 2020.



MinION from ONT

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](https://doi.org/10.1109/MM.2020.3013728)

## FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications

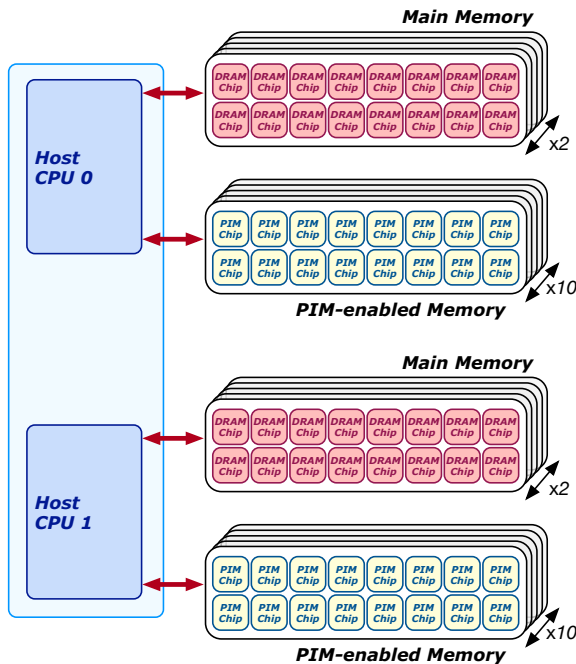
July-Aug. 2021, pp. 39-48, vol. 41

DOI Bookmark: [10.1109/MM.2021.3088396](https://doi.org/10.1109/MM.2021.3088396)



SmidgION from ONT

# Different Platforms, Different Goals



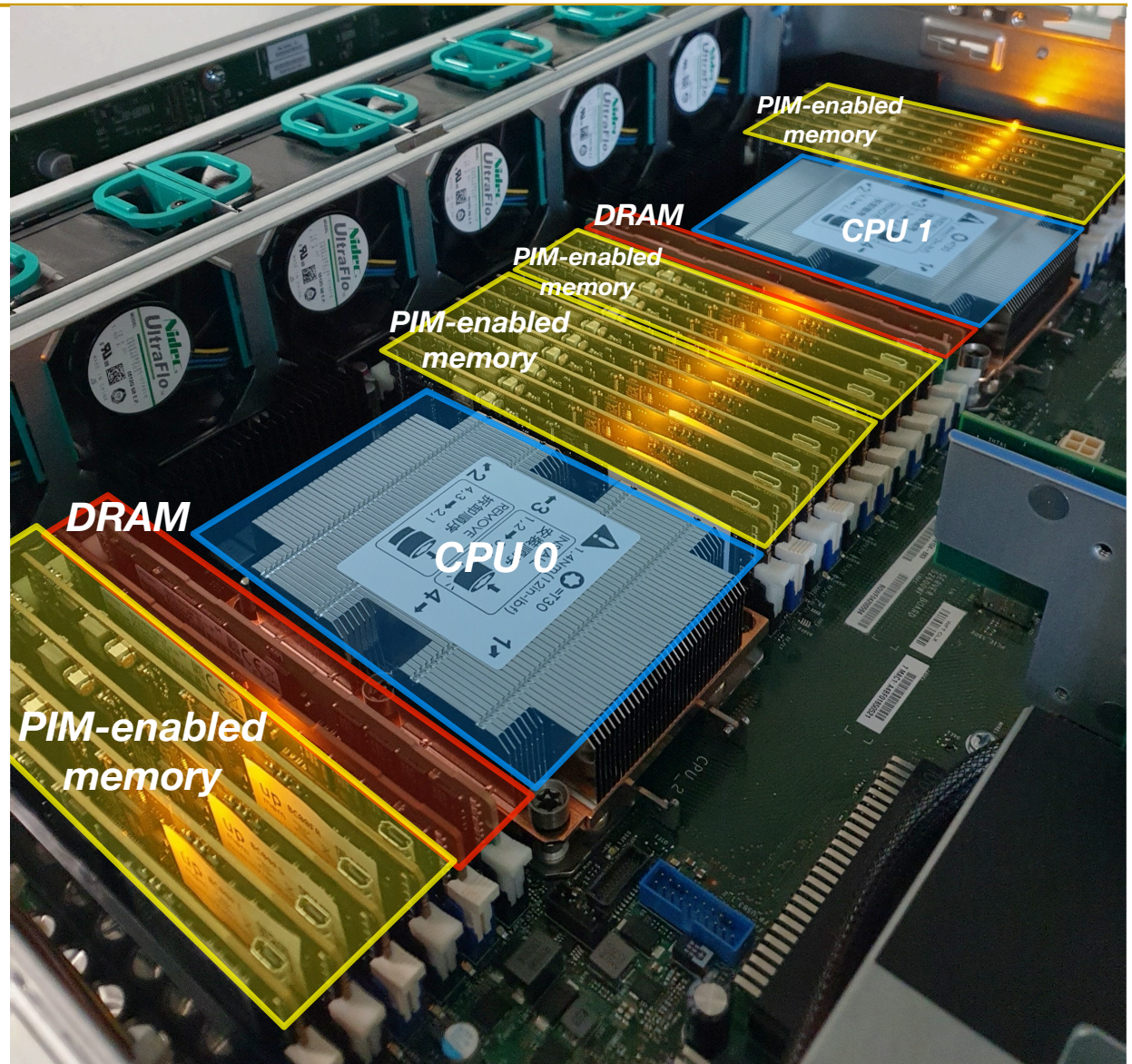
## Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

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 640 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>



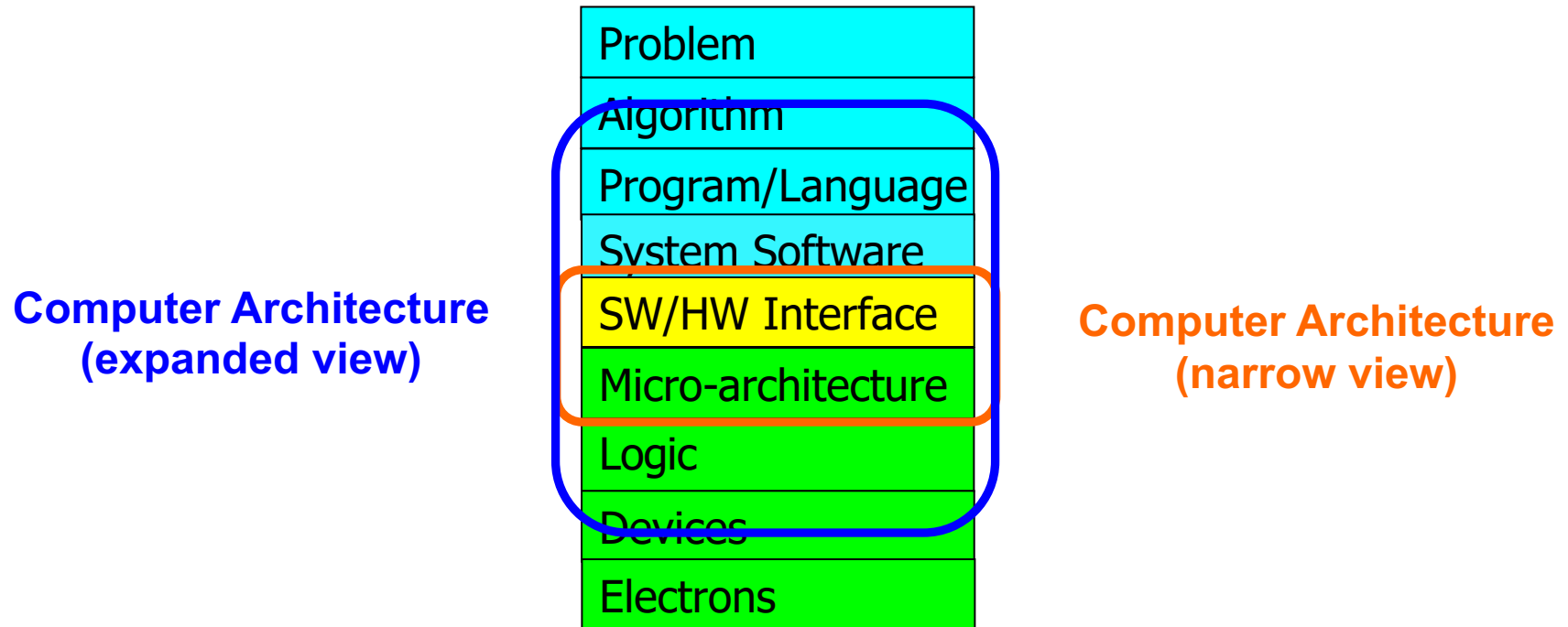
# What is Computer Architecture?

---

- 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

---



# 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)

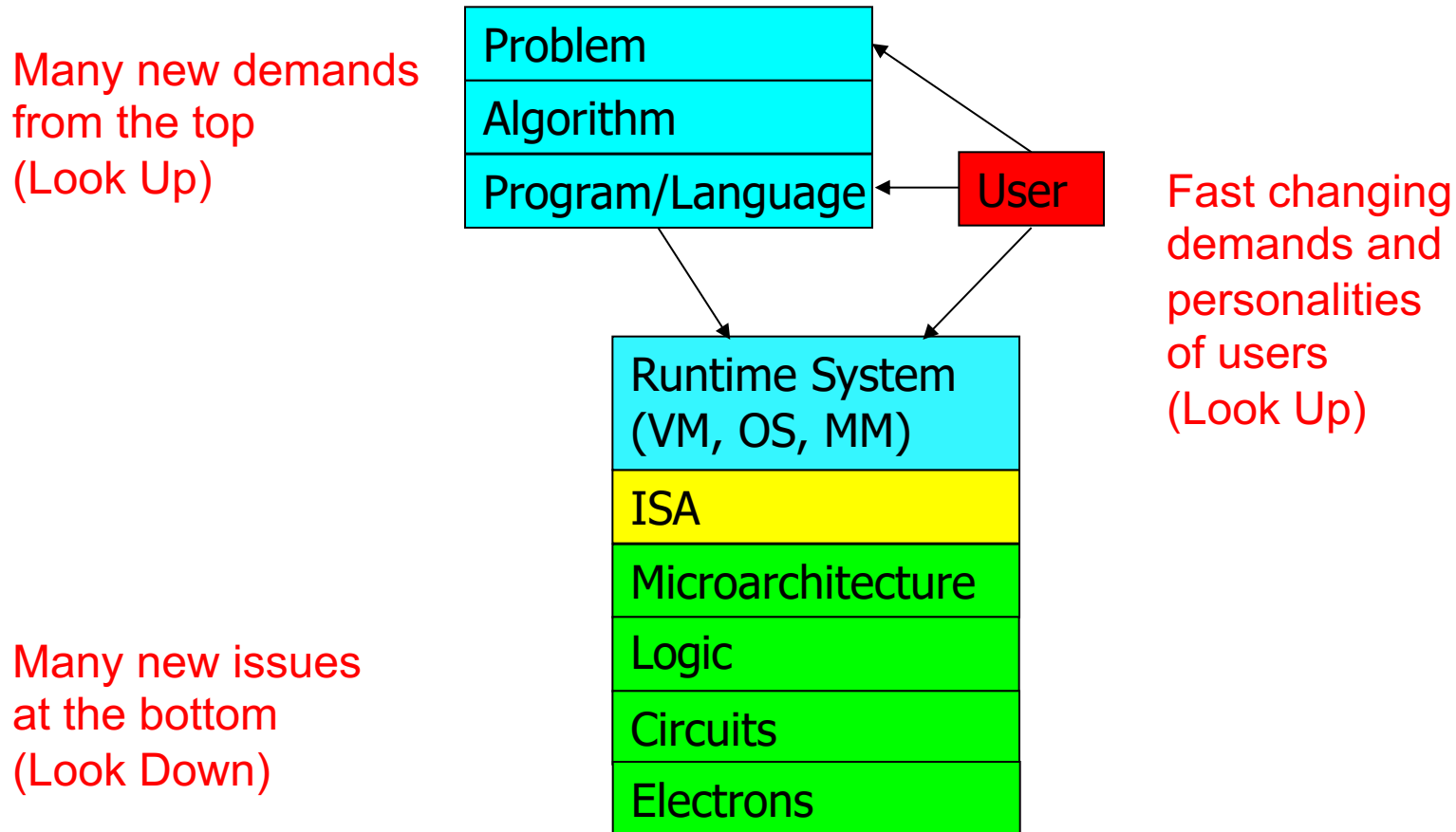
---

- 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 bottleneck
    - Reliability problems
    - Programmability problems
    - Security and privacy issues
  - No clear, definitive answers to these problems
-



# Computer Architecture Today (II)

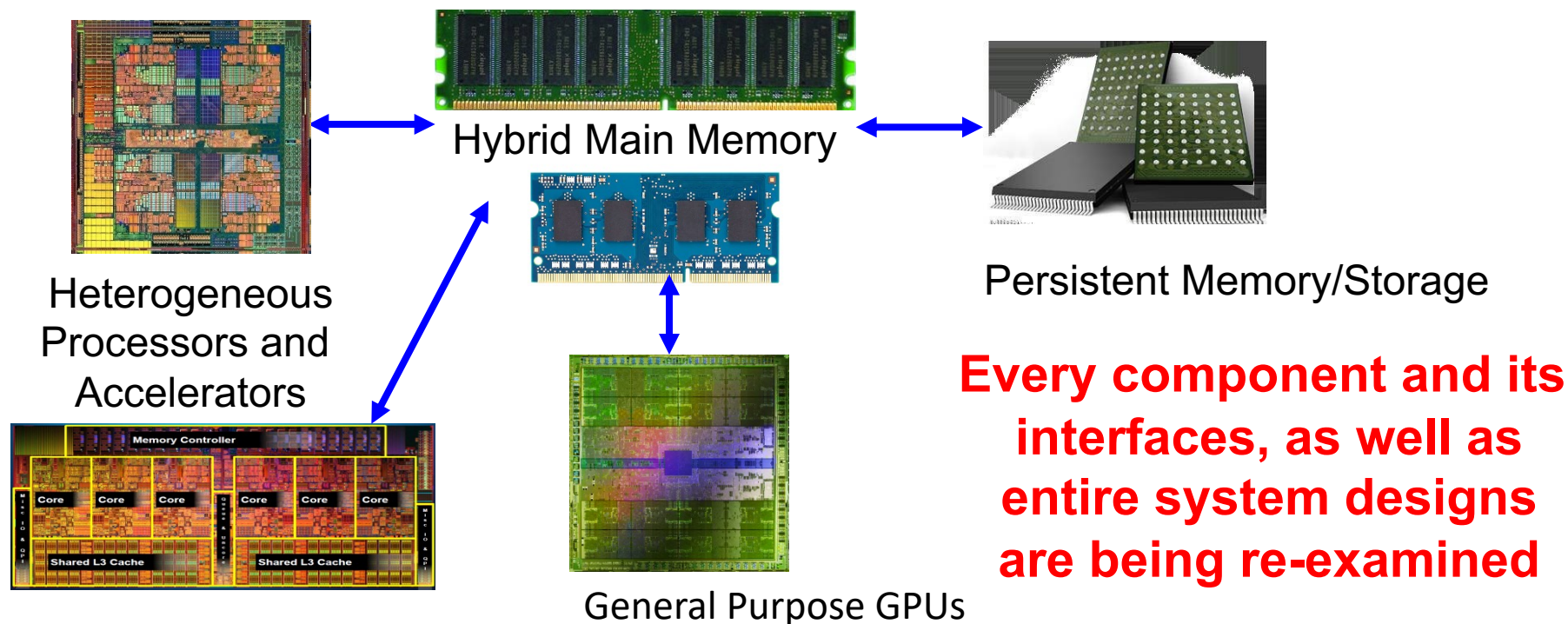
- These problems affect all parts of the computing stack – if we do not change the way we design systems



- No clear, definitive answers to these problems

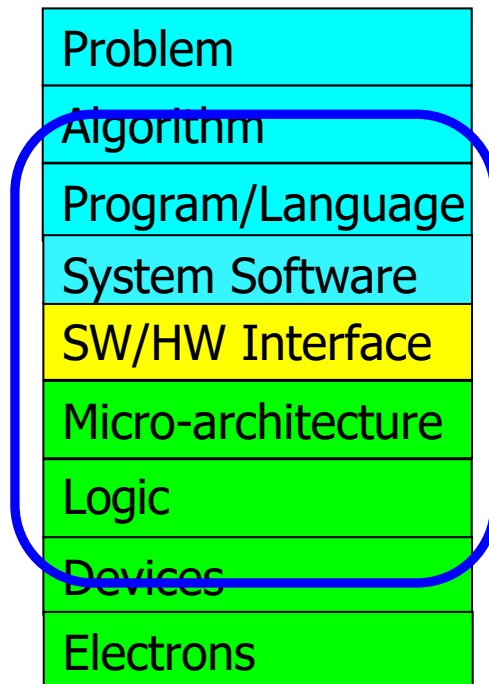
# Computer Architecture Today (III)

- 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



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.<sup>[1]</sup> 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.

# Historical: Opportunities at the Bottom (II)

---

## There's Plenty of Room at the Bottom

---

From Wikipedia, the free encyclopedia

Feynman considered some ramifications of a general ability to manipulate matter on an atomic scale. He was particularly interested in the possibilities of denser computer circuitry, and microscopes that could see things much smaller than is possible with scanning electron microscopes. These ideas were later realized by the use of the scanning tunneling microscope, the atomic force microscope and other examples of scanning probe microscopy and storage systems such as Millipede, created by researchers at IBM.

Feynman also suggested that it should be possible, in principle, to make nanoscale machines that "arrange the atoms the way we want", and do chemical synthesis by mechanical manipulation.

He also presented the possibility of "swallowing the doctor", an idea that he credited in the essay to his friend and graduate student Albert Hibbs. This concept involved building a tiny, swallowable surgical robot.



# 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<sup>1</sup>,  Neil C. Thompson<sup>1,2,\*</sup>,  Joel S. Emer<sup>1,3</sup>,  Bradley C. Kuszmaul<sup>1,†</sup>, Butler W. Lampson<sup>1,4</sup>,  ...

+ 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.

# Axiom, Revisited

---

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

but **much more so**

when you

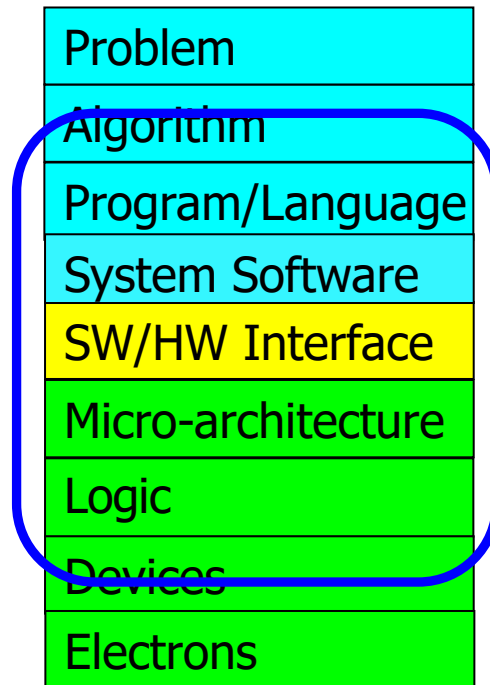
**communicate well between and optimize across**

**the top and the bottom**

# Hence the Expanded View

---

**Computer Architecture  
(expanded view)**





# Some Cross-Layer Design Examples (Foreshadowing)

# 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<sup>1</sup> Nandita Vijaykumar<sup>2,1</sup> Christina Giannoula<sup>1,3</sup> Roknoddin Azizi<sup>1</sup>  
Skanda Koppula<sup>1</sup> Nika Mansouri Ghiasi<sup>1</sup> Taha Shahroodi<sup>1</sup> Juan Gomez Luna<sup>1</sup> Onur Mutlu<sup>1,2</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Carnegie Mellon University

<sup>3</sup>National Technical University of Athens

# GenASM: HW/SW Approximate String Matching Accelerator

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, 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, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**  
*Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.*  
[[Lighting Talk Video](#) (1.5 minutes)]  
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (18 minutes)]  
[[Slides \(pptx\)](#) ([pdf](#))]

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

Damla Senol Cali<sup>†</sup>✕ Gurpreet S. Kalsi<sup>✕</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◇</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<sup>◇†</sup>  
Rachata Ausavarungnirun<sup>○</sup> Mohammed Alser<sup>◇</sup> Juan Gomez-Luna<sup>◇</sup> Amirali Boroumand<sup>†</sup> Anant Nori<sup>✕</sup>  
Allison Scibisz<sup>†</sup> Sreenivas Subramoney<sup>✕</sup> Can Alkan<sup>▽</sup> Saugata Ghose<sup>\*†</sup> Onur Mutlu<sup>◇†▽</sup>  
<sup>†</sup>Carnegie Mellon University   <sup>✕</sup>Processor Architecture Research Lab, Intel Labs   <sup>▽</sup>Bilkent University   <sup>◇</sup>ETH Zürich  
<sup>‡</sup>Facebook   <sup>○</sup>King Mongkut's University of Technology North Bangkok   <sup>\*</sup>University of Illinois at Urbana-Champaign

# SW/HW Climate Modeling Accelerator

---

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,  
**"NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"**  
*Proceedings of the 30th International Conference on Field-Programmable Logic and Applications (FPL)*, Gothenburg, Sweden, September 2020.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]  
[[Talk Video](#) (23 minutes)]  
***Nominated for the Stamatis Vassiliadis Memorial Award.***

## NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling

Gagandeep Singh<sup>a,b,c</sup>    Dionysios Diamantopoulos<sup>c</sup>    Christoph Hagleitner<sup>c</sup>    Juan Gómez-Luna<sup>b</sup>  
Sander Stuijk<sup>a</sup>    Onur Mutlu<sup>b</sup>    Henk Corporaal<sup>a</sup>  
<sup>a</sup>Eindhoven University of Technology    <sup>b</sup>ETH Zürich    <sup>c</sup>IBM Research Europe, Zurich



# HW/SW Time Series Analysis Accelerator

---

- Ivan Fernandez, Ricardo Quisiant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,  
**"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"**  
*Proceedings of the 38th IEEE International Conference on Computer Design (ICCD)*, Virtual, October 2020.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Talk Video](#) (10 minutes)]  
[[Source Code](#)]

## NATSA: A Near-Data Processing Accelerator for Time Series Analysis

Ivan Fernandez<sup>§</sup>

Ricardo Quisiant<sup>§</sup>

Christina Giannoula<sup>†</sup>

Mohammed Alser<sup>‡</sup>

Juan Gómez-Luna<sup>‡</sup>

Eladio Gutiérrez<sup>§</sup>

Oscar Plata<sup>§</sup>

Onur Mutlu<sup>‡</sup>

<sup>§</sup>*University of Malaga*

<sup>†</sup>*National Technical University of Athens*

<sup>‡</sup>*ETH Zürich*

# FPGA-based Processing Near Memory

---

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, ["FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"](#) *IEEE Micro* (**IEEE MICRO**), 2021.

## FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications

Gagandeep Singh<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>★</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

<sup>★</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# Accelerating Genome Analysis

---

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#)  
[IEEE Micro \(IEEE MICRO\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.  
[[Slides \(pptx\)\(pdf\)](#)]  
[[Talk Video \(1 hour 2 minutes\)](#)]

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**

ETH Zürich

**Zülal Bingöl**

Bilkent University

**Damla Senol Cali**

Carnegie Mellon University

**Jeremie Kim**

ETH Zurich and Carnegie Mellon University

**Saugata Ghose**

University of Illinois at Urbana–Champaign and  
Carnegie Mellon University

**Can Alkan**

Bilkent University

**Onur Mutlu**

ETH Zurich, Carnegie Mellon University, and  
Bilkent University



# Graph Processing Accelerator w/ PIM

---

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi,  
**"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"**  
*Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.  
[Slides (pdf)] [Lightning Session Slides (pdf)]

## A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn   Sungpack Hong<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoun Choi  
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University   <sup>§</sup>Oracle Labs   <sup>†</sup>Carnegie Mellon University

# Processing in Memory for Mobile Workloads

---

- 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.*

## 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>

# Accelerating Linked Data Structures

---

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,  
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)  
*Proceedings of the 34th IEEE International Conference on Computer Design (ICCD)*, Phoenix, AZ, USA, October 2016.

## Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation

Kevin Hsieh<sup>†</sup> Samira Khan<sup>‡</sup> Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup> Amirali Boroumand<sup>†</sup> Saugata Ghose<sup>†</sup> Onur Mutlu<sup>§†</sup>  
<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

# 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<sup>†§</sup> Abhilasha Jain<sup>†</sup> Diptesh Majumdar<sup>†</sup> Kevin Hsieh<sup>†</sup> Gennady Pekhimenko<sup>‡</sup>  
Eiman Ebrahimi<sup>⌘</sup> Nastaran Hajinazar<sup>†</sup> Phillip B. Gibbons<sup>†</sup> Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>University of Toronto

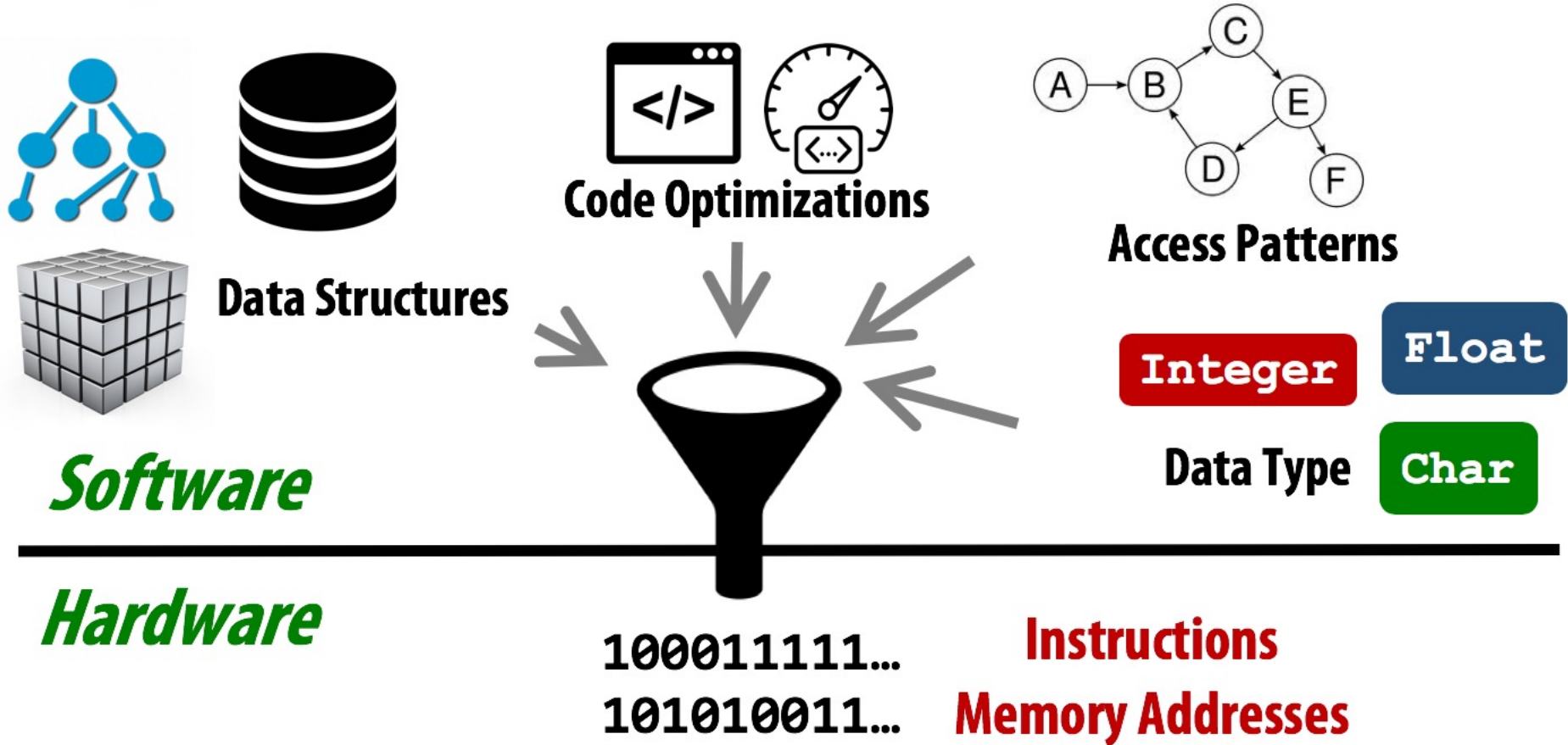
<sup>⌘</sup>NVIDIA

<sup>†</sup>Simon Fraser University

<sup>§</sup>ETH Zürich

# One Problem: Limited SW/HW Communication

## Higher-level information is not visible to HW

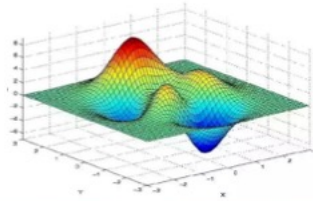




# A Solution: More Expressive Interfaces

**Performance**

**Software**



**Functionality**



**ISA  
Virtual Memory**

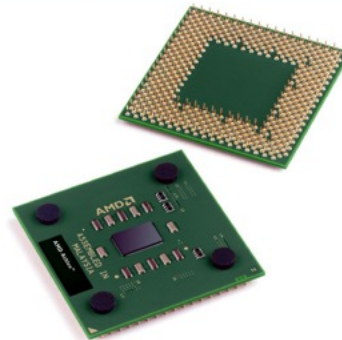
**Higher-level  
Program  
Semantics**

**Expressive  
Memory  
“XMem”**

**Hardware**



wiseGEEK





# 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 <i>knowing</i> 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 <i>data structure</i> 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 <i>different compression algorithm</i> 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) <i>highly accurate</i> software-driven prefetching while leveraging the benefits of hardware prefetching (e.g., by being memory bandwidth-aware, avoiding cache thrashing); (ii) using different prefetcher <i>types</i> 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<sup>†§</sup>      Eiman Ebrahimi<sup>‡</sup>      Kevin Hsieh<sup>†</sup>  
Phillip B. Gibbons<sup>†</sup>      Onur Mutlu<sup>§†</sup>  
<sup>†</sup>Carnegie Mellon University      <sup>‡</sup>NVIDIA      <sup>§</sup>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](#)]

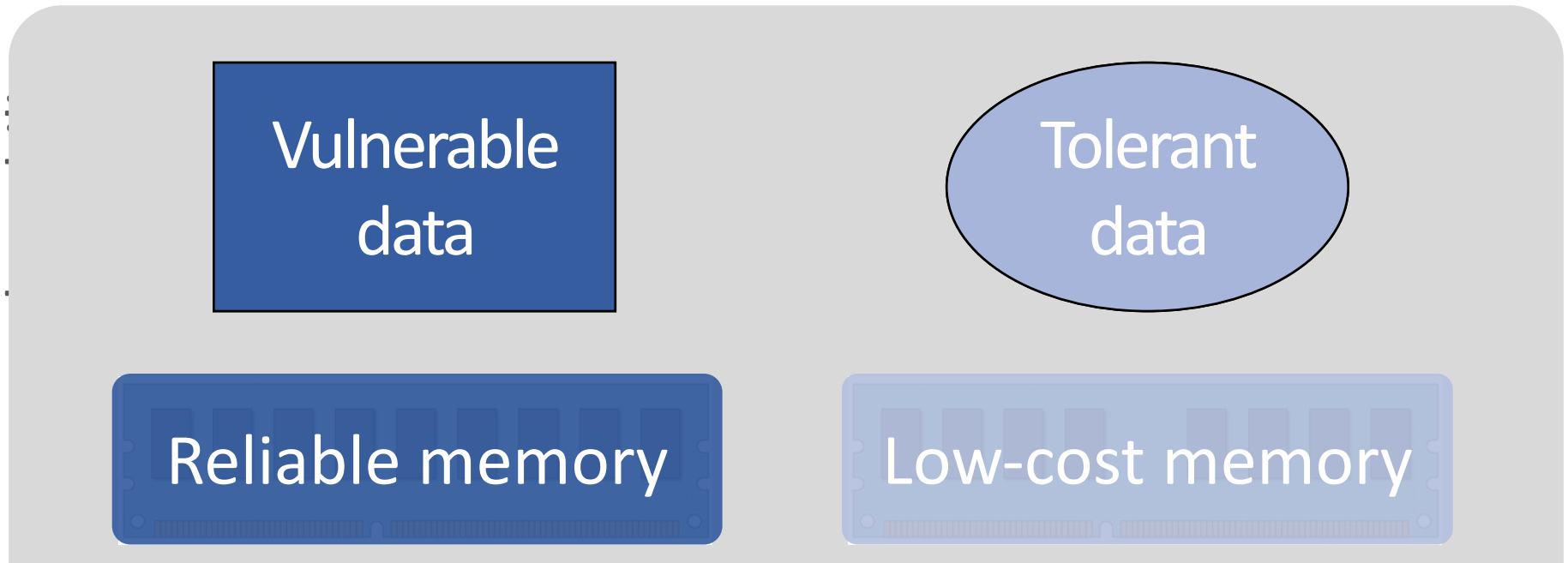
## Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

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, bknessib, kvaid}@microsoft.com

# Exploiting Memory Error Tolerance with Hybrid Memory Systems



On Microsoft's Web Search workload

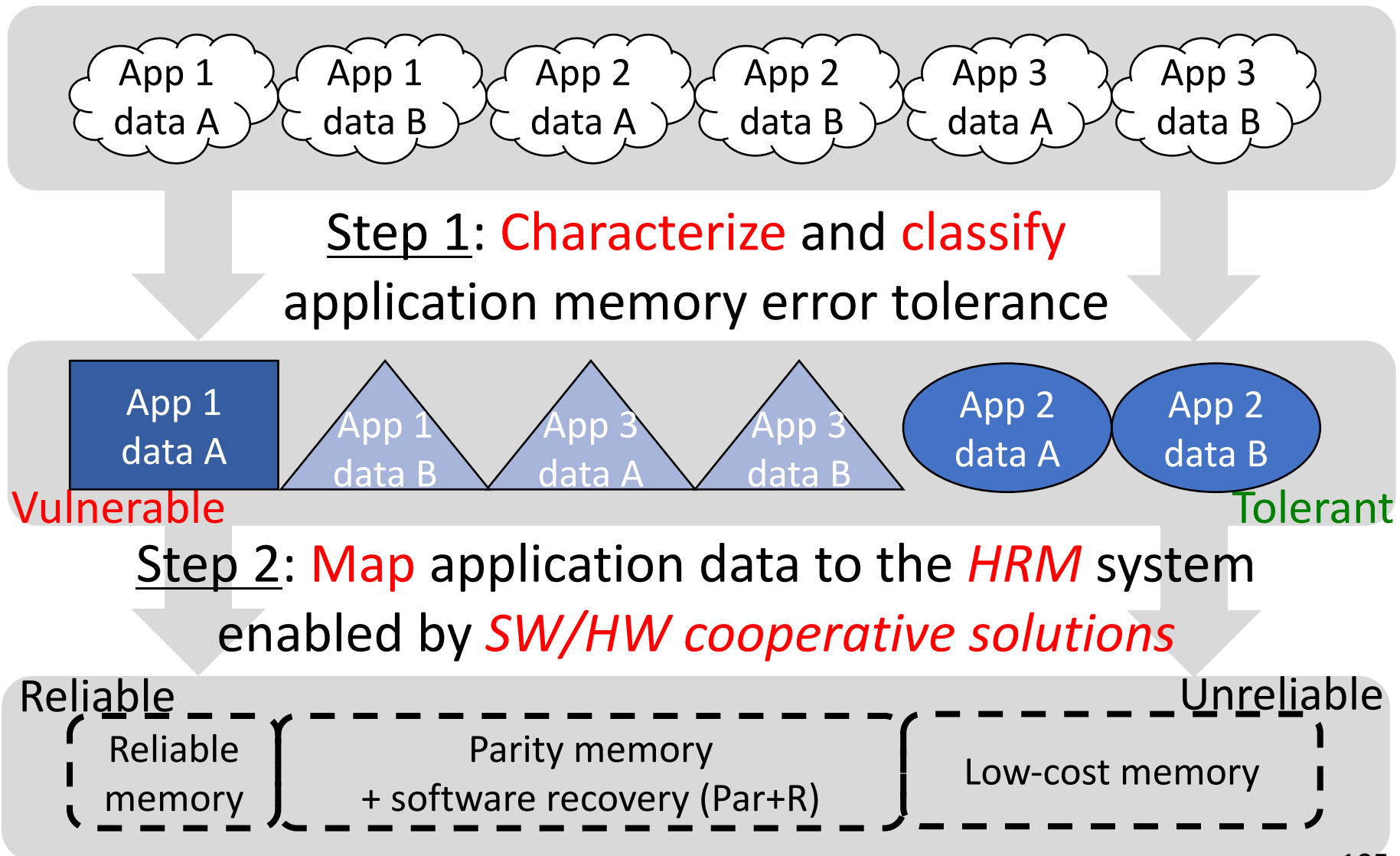
Reduces server hardware **cost** by **4.7 %**

Achieves single server **availability** target of **99.90 %**

**Heterogeneous-Reliability Memory** [DSN 2014]



# Heterogeneous-Reliability Memory



# 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<sup>\*†</sup> Pratyush Patel<sup>✕</sup> Minesh Patel<sup>\*</sup> Konstantinos Kanellopoulos<sup>\*</sup> Saugata Ghose<sup>‡</sup>  
Rachata Ausavarungnirun<sup>⊙</sup> Geraldo F. Oliveira<sup>\*</sup> Jonathan Appavoo<sup>◇</sup> Vivek Seshadri<sup>▽</sup> Onur Mutlu<sup>\*‡</sup>

<sup>\*</sup>ETH Zürich   <sup>†</sup>Simon Fraser University   <sup>✕</sup>University of Washington   <sup>‡</sup>Carnegie Mellon University  
<sup>⊙</sup>King Mongkut's University of Technology North Bangkok   <sup>◇</sup>Boston University   <sup>▽</sup>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<sup>†</sup> Engin Ipek<sup>†</sup> Onur Mutlu<sup>‡</sup> Doug Burger<sup>†</sup>

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

<sup>‡</sup>Computer Architecture Laboratory  
Carnegie Mellon University  
Pittsburgh, PA  
onur@cmu.edu

# PCM as Main Memory: Idea in 2009

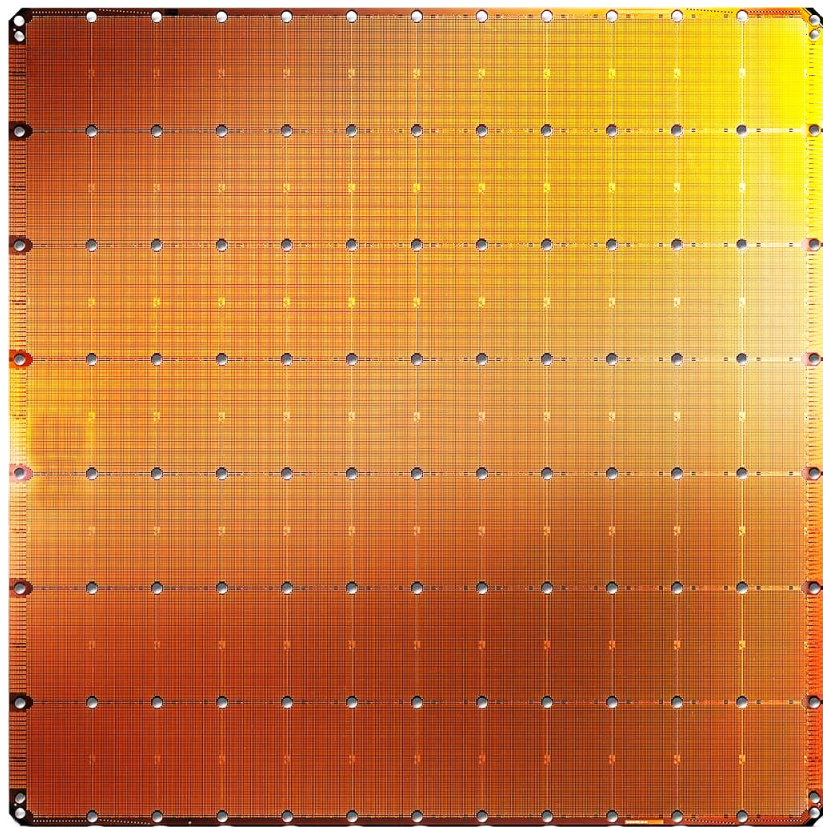
---

- 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)

---



## **Cerebras WSE**

1.2 Trillion transistors

46,225 mm<sup>2</sup>

- The largest ML accelerator chip
- 400,000 cores



## **Largest GPU**

21.1 Billion transistors

815 mm<sup>2</sup>

NVIDIA TITAN V

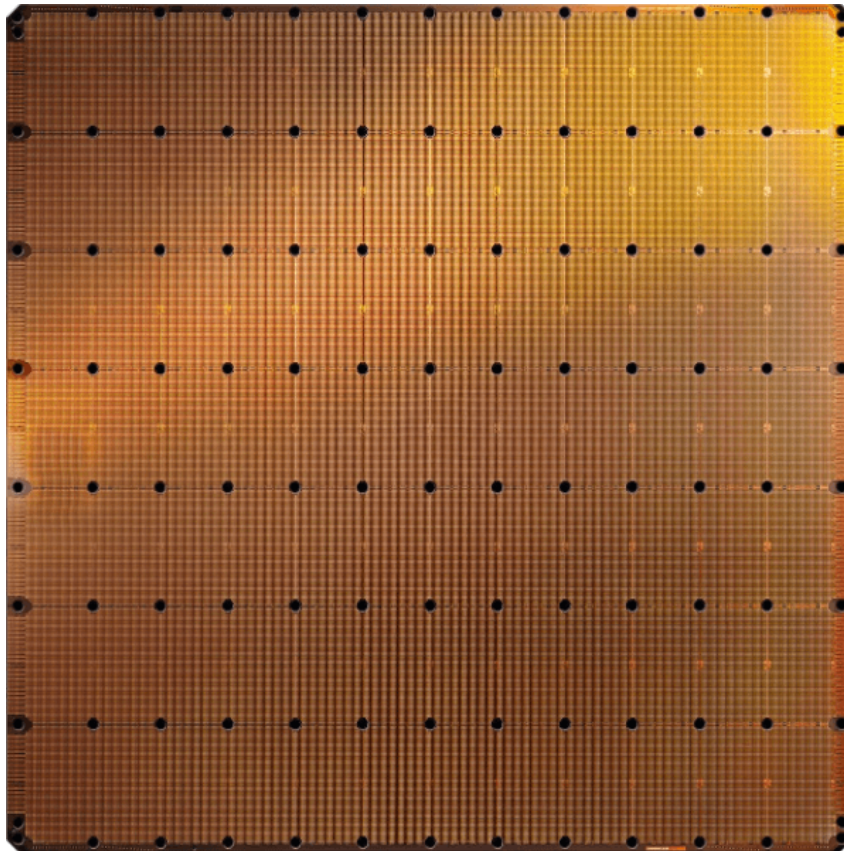
<https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning>

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



# Cerebras's Wafer Scale 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>

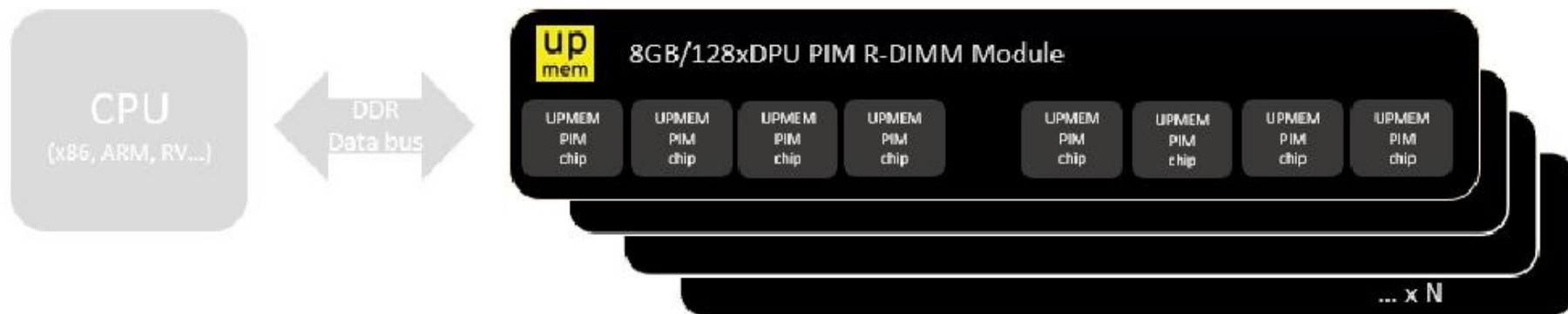
NVIDIA Ampere GA100

<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/>

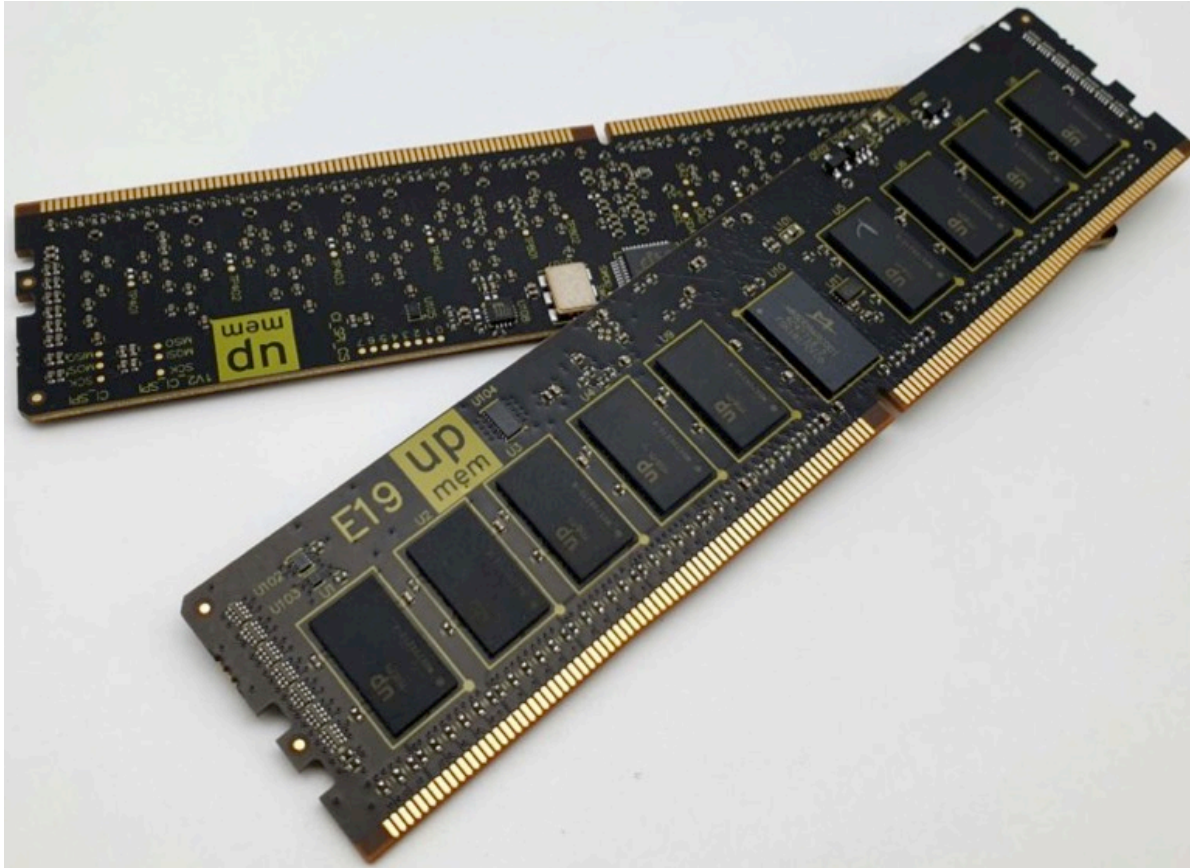
# UPMEM Processing-in-DRAM Engine (2019)

- **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



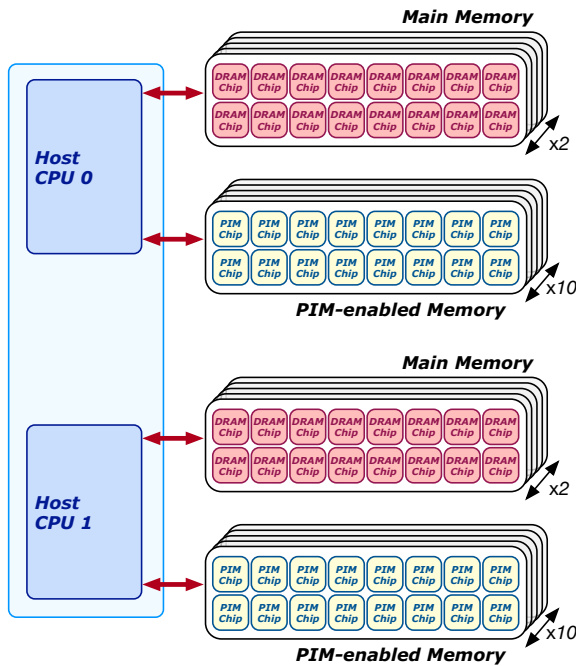
# UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz





# 2,560-DPU Processing-in-Memory System



## 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)*.

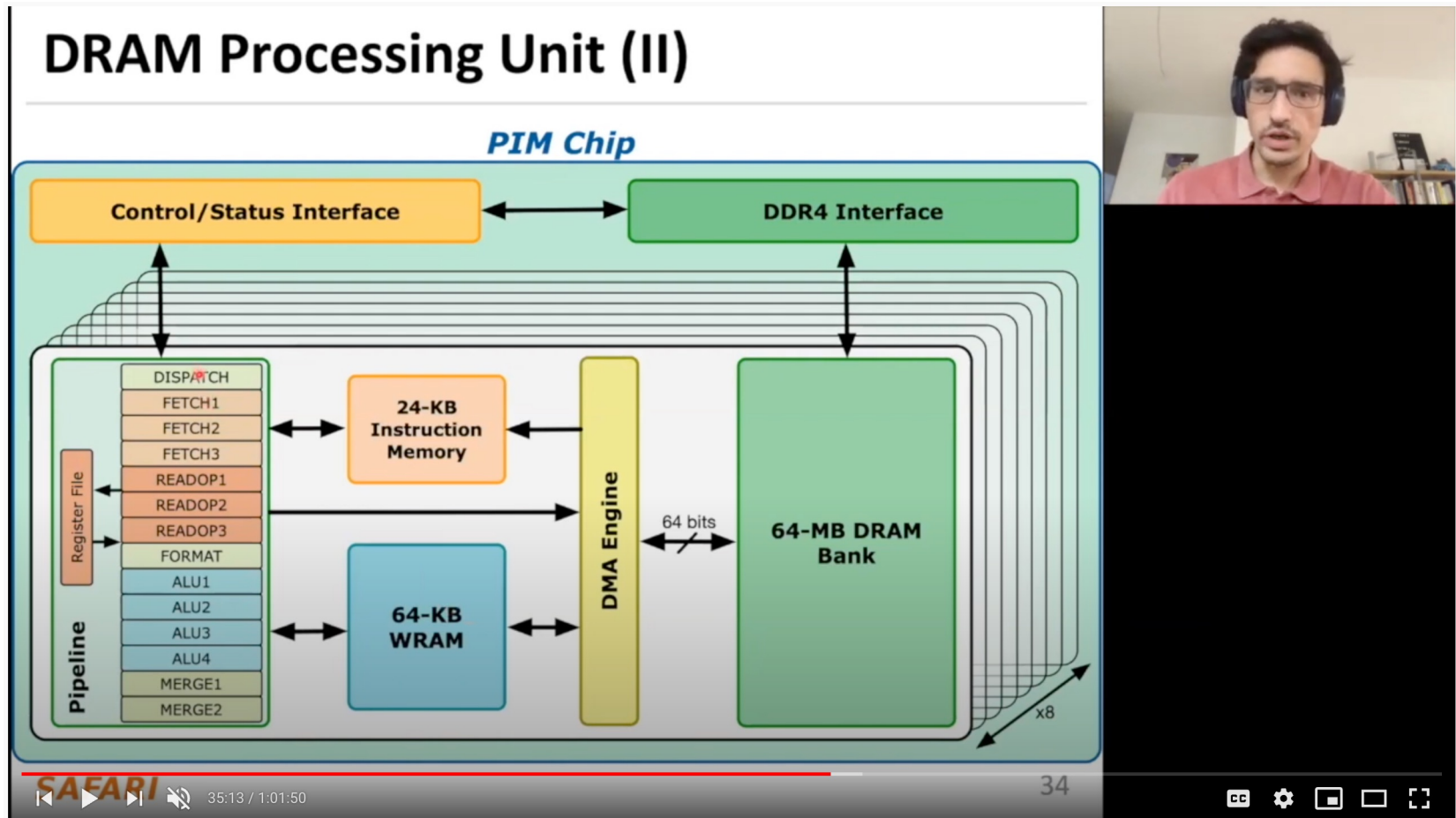
Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware 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 640 and 2,560 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>

# More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

Computer Architecture - Lecture 12d: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



**Onur Mutlu Lectures**  
16.7K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26>



# Experimental Analysis of the UPMEM PIM Engine

---

## 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 640 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.



# Understanding a Modern PIM Architecture



The video player shows a lecture titled "Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization". The speaker is Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu. The video is from the "SAFARI Live Seminar" series. The player interface includes a progress bar at 2:26 / 2:57:10, a video quality selector set to HD, and a "SUBSCRIBED" button. The video player also displays the "ETH Zürich" and "SAFARI" logos.

**Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization**

Juan Gómez Luna, Izzat El Hajj,  
Ivan Fernandez, Christina Giannoula,  
Geraldo F. Oliveira, Onur Mutlu

<https://arxiv.org/pdf/2105.03814.pdf>  
<https://github.com/CMU-SAFARI/prim-benchmarks>

ETH Zürich SAFARI

SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

2,579 views • Streamed live on Jul 12, 2021

93 0 SHARE + SAVE ...

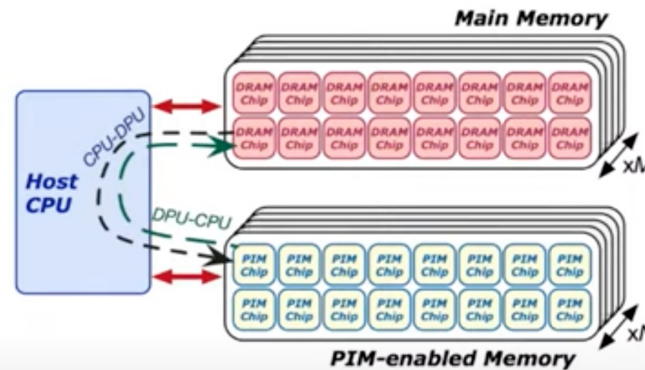
 **Onur Mutlu Lectures**  
18.7K subscribers

SUBSCRIBED

# More on Analysis of the UPMEM PIM Engine

## Inter-DPU Communication

- There is **no direct communication channel between DPUs**



- Inter-DPU communication takes place via the host CPU using CPU-DPU and DPU-CPU transfers
- Example communication patterns:
  - Merging of partial results to obtain the final result
    - Only DPU-CPU transfers
  - Redistribution of intermediate results for further computation
    - DPU-CPU transfers and CPU-DPU transfers



SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

1,868 views • Streamed live on Jul 12, 2021

81 0 SHARE SAVE ...



Onur Mutlu Lectures  
17.6K subscribers

Talk Title: Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization  
Dr. Juan Gómez-Luna, SAFARI Research Group, D-ITET, ETH Zurich

ANALYTICS

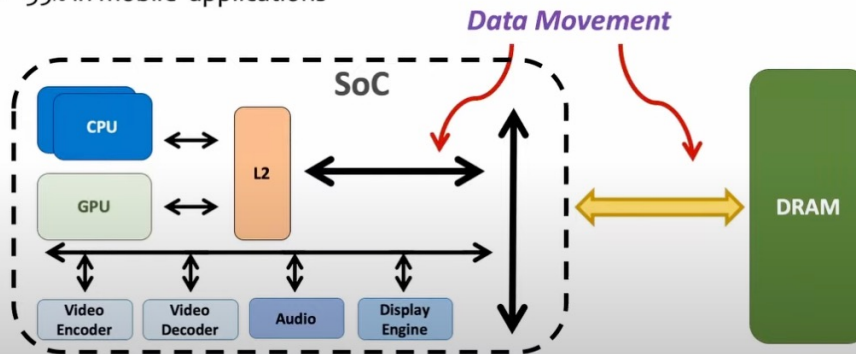
EDIT VIDEO

[https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi\\_tOTAYm--dYByNPL7JhwR9](https://www.youtube.com/watch?v=D8Hjy2IU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9)

# More on Analysis of the UPMEM PIM Engine

## Data Movement in Computing Systems

- **Data movement** dominates **performance** and is a major system **energy bottleneck**
- **Total system energy**: data movement accounts for
  - 62% in consumer applications\*,
  - 40% in scientific applications\*,
  - 35% in mobile applications\*



\* Boroumand et al., "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018

\* Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," IISWC 2013

\* Pandiyan and Wu, "Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms," IISWC 2014

SAFARI

3

Understanding a Modern Processing-in-Memory Arch: Benchmarking & Experimental Characterization; 21m

3,482 views • Premiered Jul 25, 2021

38

0

SHARE

SAVE

...



Onur Mutlu Lectures

17.9K subscribers

ANALYTICS

EDIT VIDEO

[https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8\\_VVChACnON4sfh2bJ5IrD&index=159](https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159)

# FPGA-based Processing Near Memory

---

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, ["FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"](#) *IEEE Micro* (**IEEE MICRO**), to appear, 2021.

## FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications

Gagandeep Singh<sup>◇</sup> Mohammed Alser<sup>◇</sup> Damla Senol Cali<sup>✕</sup>

Dionysios Diamantopoulos<sup>▽</sup> Juan Gómez-Luna<sup>◇</sup>

Henk Corporaal<sup>\*</sup> Onur Mutlu<sup>◇✕</sup>

<sup>◇</sup>*ETH Zürich*    <sup>✕</sup>*Carnegie Mellon University*

<sup>\*</sup>*Eindhoven University of Technology*    <sup>▽</sup>*IBM Research Europe*

# Samsung Function-in-Memory DRAM (2021)



## Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021

Audio



Share



*The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%*

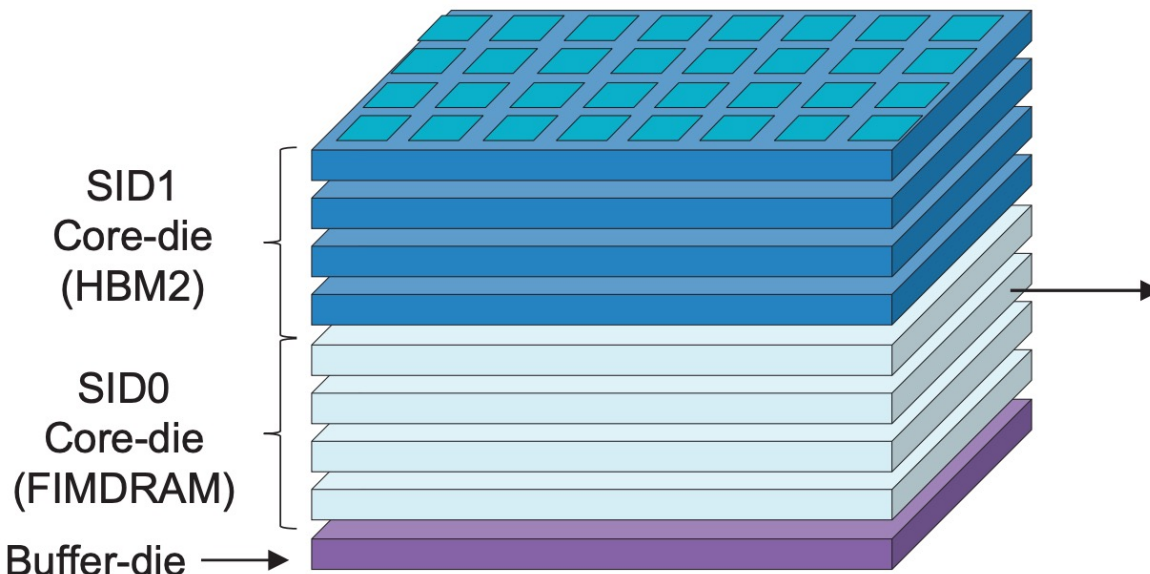
Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power – the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, "Our groundbreaking HBM-PIM is the industry's first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications."



# 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

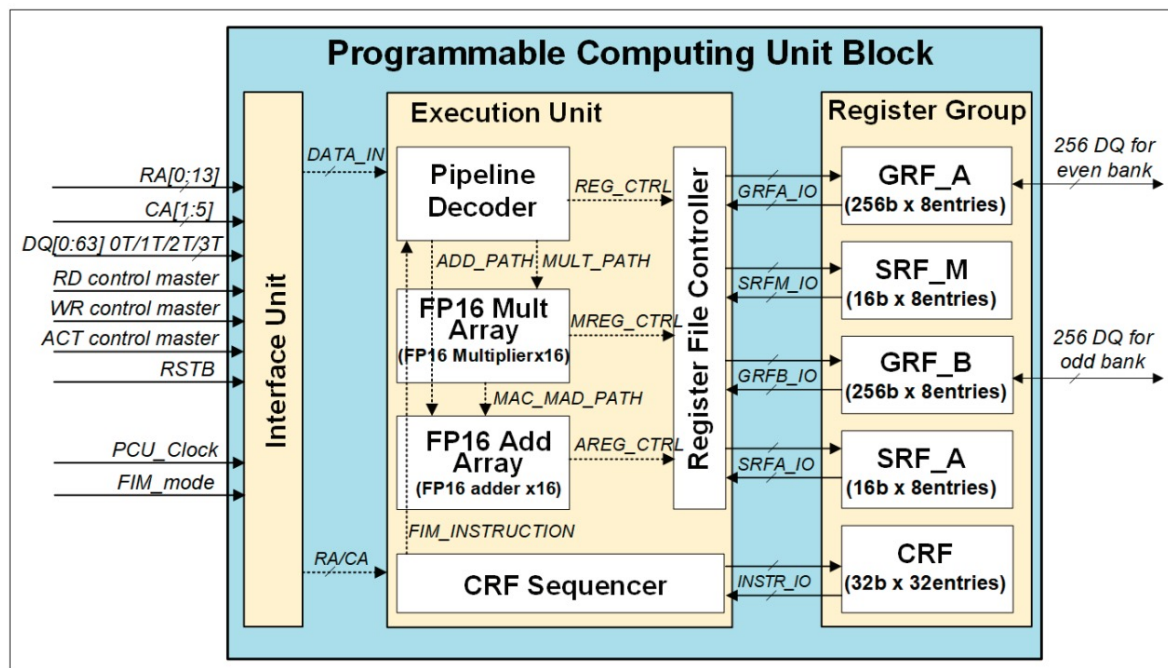


# Samsung Function-in-Memory DRAM (2021)

## Programmable Computing Unit

### ■ Configuration of PCU block

- Interface unit to control data flow
- Execution unit to perform operations
- Register group
  - 32 entries of CRF for instruction memory
  - 16 GRF for weight and accumulation
  - 16 SRF to store constants for MAC operations



[Block diagram of PCU in FIMDRAM]

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>, HyungMin 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>3</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 Function-in-Memory DRAM (2021)

[Available instruction list for FIM operation]

Type	CMD	Description
Floating Point	ADD	FP16 addition
	MUL	FP16 multiplication
	MAC	FP16 multiply-accumulate
	MAD	FP16 multiply and add
Data Path	MOVE	Load or store data
	FILL	Copy data from bank to GRFs
Control Path	NOP	Do nothing
	JUMP	Jump instruction
	EXIT	Exit instruction

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>, HyungMin 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>1</sup>, Seungwoo Seo<sup>1</sup>, JoonHo Song<sup>1</sup>, Jaeyoun Youn<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

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

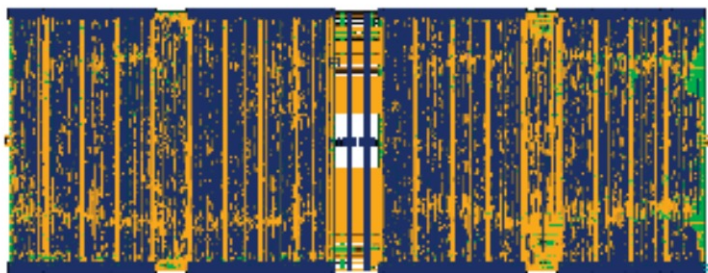
<sup>2</sup>Samsung Electronics, San Jose, CA

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

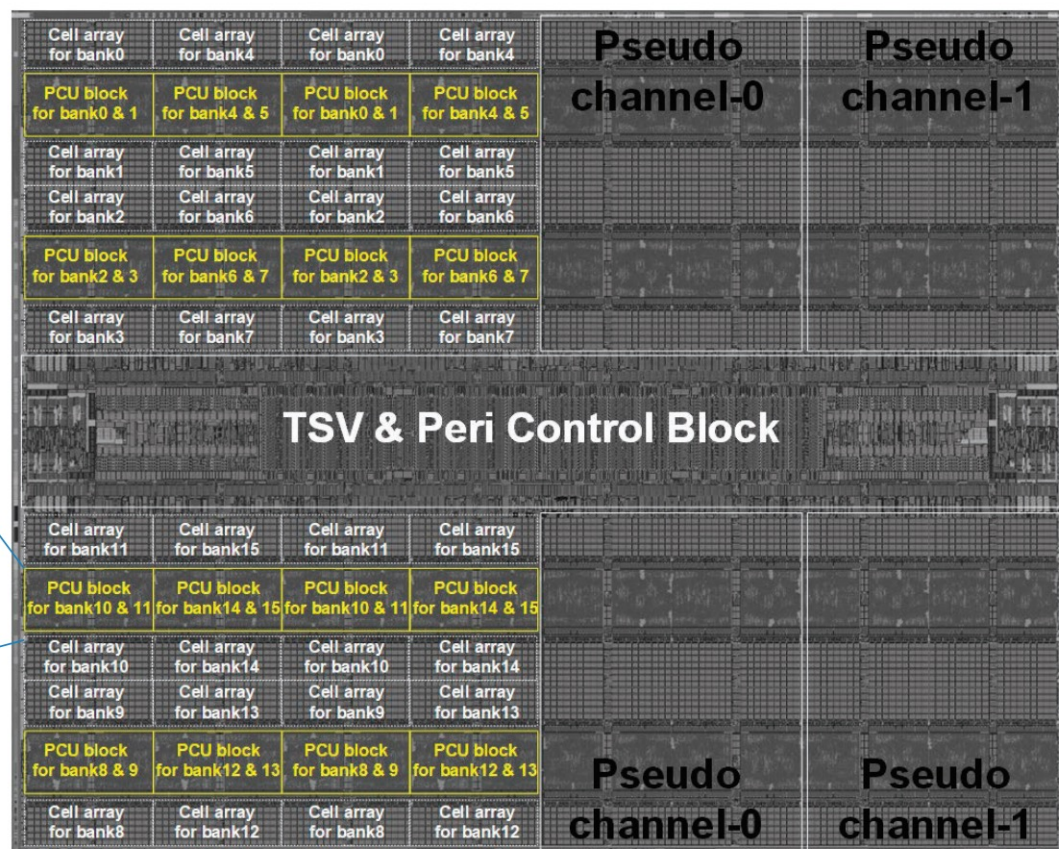
# Samsung Function-in-Memory DRAM (2021)

## Chip Implementation

- Mixed design methodology to implement FIMDRAM
  - Full-custom + Digital RTL



[Digital RTL design for PCU block]



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>, Jongyeon Choi<sup>1</sup>, Hyun-Sung Shim<sup>1</sup>, Jin Kim<sup>1</sup>, BengSeng Phuah<sup>1</sup>, HyounMin Kim<sup>1</sup>, Myeong Jun Song<sup>1</sup>, Ahn Chai<sup>1</sup>, Daeho Kim<sup>1</sup>, SooYoung Kim<sup>1</sup>, Eun-Bong Kim<sup>1</sup>, David Wang<sup>2</sup>, Shintraeng Kang<sup>3</sup>, Yulwan Ro<sup>3</sup>, Seungwoo Seo<sup>3</sup>, JoonHo Song<sup>3</sup>, Jaeyoun Yoon<sup>1</sup>, Kyomin Sohn<sup>1</sup>, Nam Sung Kim<sup>1</sup>

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

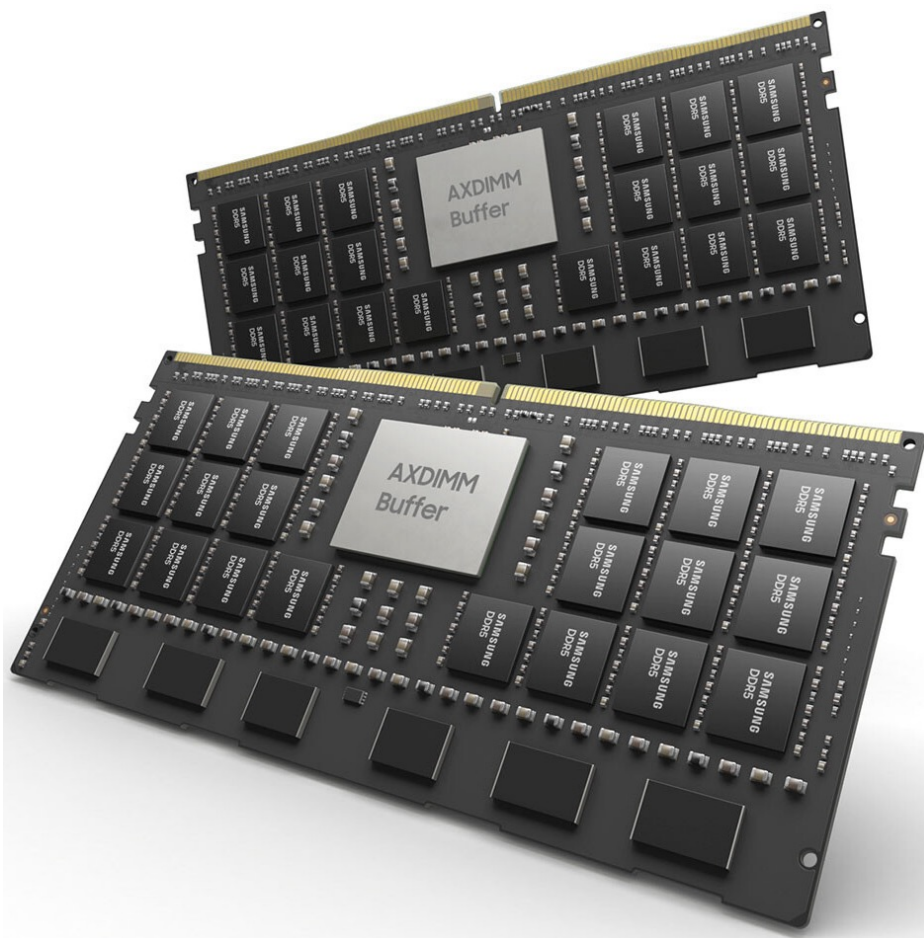
<sup>2</sup>Samsung Electronics, San Jose, CA

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

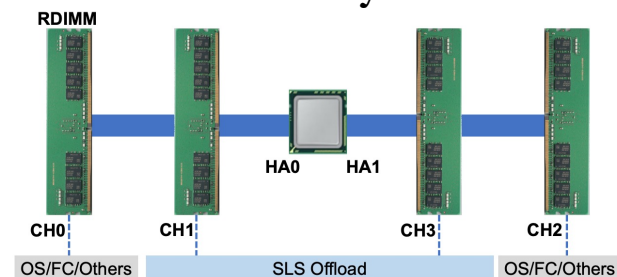


# Samsung AxDIMM (2021)

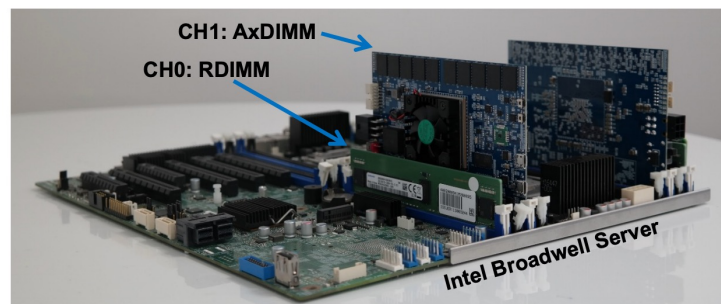
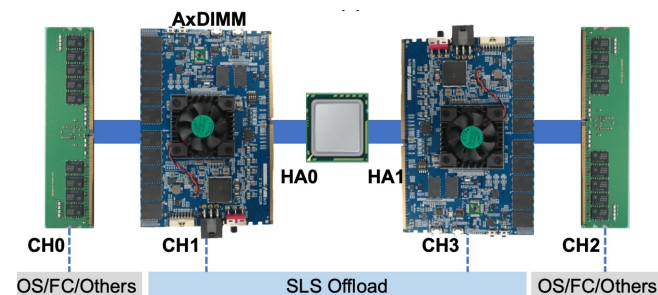
- DDR5-PIM
  - DLRM recommendation system



Baseline System



AxDIMM System



# Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory

# Specialized Processing in Memory (2015)

---

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoun Choi,  
**"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"**  
*Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.  
[Slides (pdf)] [Lightning Session Slides (pdf)]

## A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn   Sungpack Hong<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoun Choi  
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

<sup>§</sup>Oracle Labs

<sup>†</sup>Carnegie Mellon University



# Simple Processing in Memory (2015)

---

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoungh Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015.  
[[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

## **PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture**

Junwhan Ahn   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   Kiyoungh Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

<sup>†</sup>Carnegie Mellon University

# Processing in Memory on Mobile Devices

---

- 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.*

## 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>

# Efficient Synchronization for NDP

---

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, and Onur Mutlu,  
**"SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"**  
*Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA)*, Virtual, February-March 2021.

## **SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures**

Christina Giannoula<sup>†‡</sup> Nandita Vijaykumar<sup>\*‡</sup> Nikela Papadopoulou<sup>†</sup> Vasileios Karakostas<sup>†</sup> Ivan Fernandez<sup>§‡</sup>  
Juan Gómez-Luna<sup>‡</sup> Lois Orosa<sup>‡</sup> Nectarios Koziris<sup>†</sup> Georgios Goumas<sup>†</sup> Onur Mutlu<sup>‡</sup>  
<sup>†</sup>*National Technical University of Athens*    <sup>‡</sup>*ETH Zürich*    <sup>\*</sup>*University of Toronto*    <sup>§</sup>*University of Malaga*

# Accelerating GPU Execution with PIM (I)

---

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

*Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.*

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Session Slides \(pptx\)](#) ([pdf](#))]

## Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems

Kevin Hsieh<sup>‡</sup> Eiman Ebrahimi<sup>†</sup> Gwangsun Kim<sup>\*</sup> Niladrish Chatterjee<sup>†</sup> Mike O'Connor<sup>†</sup>  
Nandita Vijaykumar<sup>‡</sup> Onur Mutlu<sup>§‡</sup> Stephen W. Keckler<sup>†</sup>

<sup>‡</sup>Carnegie Mellon University <sup>†</sup>NVIDIA <sup>\*</sup>KAIST <sup>§</sup>ETH Zürich

# Accelerating Linked Data Structures

---

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,  
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)  
*Proceedings of the 34th IEEE International Conference on Computer Design (ICCD)*, Phoenix, AZ, USA, October 2016.

## Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation

Kevin Hsieh<sup>†</sup> Samira Khan<sup>‡</sup> Nandita Vijaykumar<sup>†</sup>  
Kevin K. Chang<sup>†</sup> Amirali Boroumand<sup>†</sup> Saugata Ghose<sup>†</sup> Onur Mutlu<sup>§†</sup>  
<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*



# DAMOV Analysis Methodology & Workloads

---

## DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Data movement between the CPU and main memory is a first-order obstacle against improving performance, scalability, and energy efficiency in modern systems. Computer systems employ a range of techniques to reduce overheads tied to data movement, spanning from traditional mechanisms (e.g., deep multi-level cache hierarchies, aggressive hardware prefetchers) to emerging techniques such as Near-Data Processing (NDP), where some computation is moved close to memory. Prior NDP works investigate the root causes of data movement bottlenecks using different profiling methodologies and tools. However, there is still a lack of understanding about the key metrics that can identify different data movement bottlenecks and their relation to traditional and emerging data movement mitigation mechanisms. Our goal is to methodically identify potential sources of data movement over a broad set of applications and to comprehensively compare traditional compute-centric data movement mitigation techniques (e.g., caching and prefetching) to more memory-centric techniques (e.g., NDP), thereby developing a rigorous understanding of the best techniques to mitigate each source of data movement.

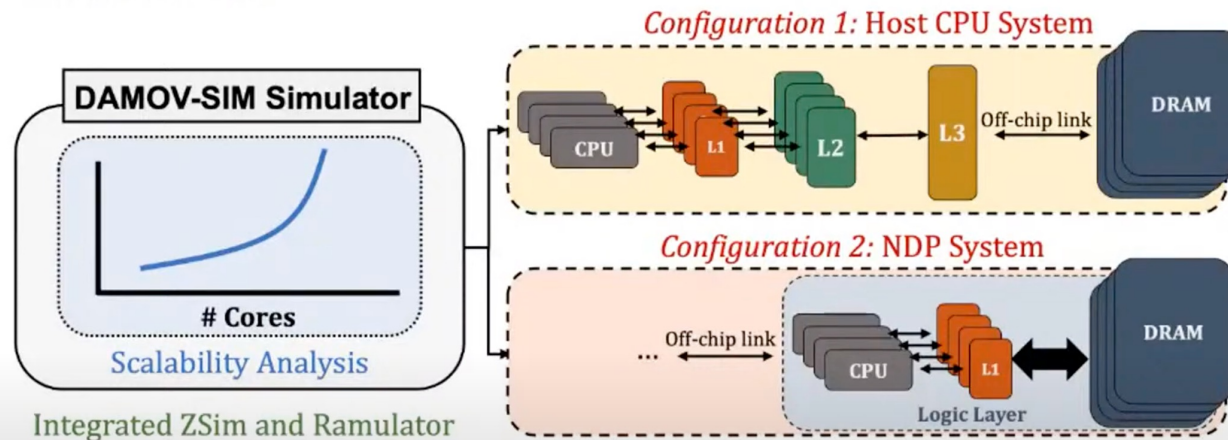
With this goal in mind, we perform the first large-scale characterization of a wide variety of applications, across a wide range of application domains, to identify fundamental program properties that lead to data movement to/from main memory. We develop the first systematic methodology to classify applications based on the sources contributing to data movement bottlenecks. From our large-scale characterization of 77K functions across 345 applications, we select 144 functions to form the first open-source benchmark suite (DAMOV) for main memory data movement studies. We select a diverse range of functions that (1) represent different types of data movement bottlenecks, and (2) come from a wide range of application domains. Using NDP as a case study, we identify new insights about the different data movement bottlenecks and use these insights to determine the most suitable data movement mitigation mechanism for a particular application. We open-source DAMOV and the complete source code for our new characterization methodology at <https://github.com/CMU-SAFARI/DAMOV>.



# More on DAMOV Analysis Methodology & Workloads

## Step 3: Memory Bottleneck Classification (2/)

- **Goal:** identify the specific sources of data movement bottlenecks



- **Scalability Analysis:**
  - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
  - 3D-stacked memory as main memory

DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV>

SAFARI Live Seminar: DAMOV: A New Methodology & Benchmark Suite for Data Movement Bottlenecks

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures  
17.7K subscribers

ANALYTICS

EDIT VIDEO

SAFARI

[https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi\\_tOTAYm--dYByNPL7JhWR9&index=3](https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhWR9&index=3)

# DAMOV is Open-Source

- We open-source our benchmark suite and our toolchain

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About



DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at <https://arxiv.org/pdf/2105.03725.pdf>)

Readme

Releases

No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)

Languages



omutlu Update README.md

ce1b4ea 17 days ago 5 commits

simulator

Cleaning

19 days ago

README.md

Update README.md

17 days ago

get\_workloads.sh

DAMOV -- first commit

19 days ago

README.md



## DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including [BWA](#), [Chai](#), [Darknet](#), [GASE](#), [Hardware Effects](#), [Hashjoin](#), [HPCC](#), [HPCG](#), [Ligra](#), [PARSEC](#), [Parboil](#), [PolyBench](#), [Phoenix](#), [Rodinia](#), [SPLASH-2](#), [STREAM](#).

DAMOV-SIM

DAMOV  
Benchmarks

# More on DAMOV

---

- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,  
**"DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"**  
*Preprint in [arXiv](#), 8 May 2021.*  
[[arXiv preprint](#)]  
[[DAMOV Suite and Simulator Source Code](#)]

# Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory

# In-DRAM Processing (2013)

---

- Vivek Seshadri et al., “[\*\*Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology\*\*](#),” MICRO 2017.

## Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

Vivek Seshadri<sup>1,5</sup> Donghyuk Lee<sup>2,5</sup> Thomas Mullins<sup>3,5</sup> Hasan Hassan<sup>4</sup> Amirali Boroumand<sup>5</sup>  
Jeremie Kim<sup>4,5</sup> Michael A. Kozuch<sup>3</sup> Onur Mutlu<sup>4,5</sup> Phillip B. Gibbons<sup>5</sup> Todd C. Mowry<sup>5</sup>

<sup>1</sup>Microsoft Research India   <sup>2</sup>NVIDIA Research   <sup>3</sup>Intel   <sup>4</sup>ETH Zürich   <sup>5</sup>Carnegie Mellon University

# In-DRAM Bulk Bitwise Execution (2017)

---

- Vivek Seshadri and Onur Mutlu,  
**"In-DRAM Bulk Bitwise Execution Engine"**  
*Invited Book Chapter in Advances in Computers*, to appear  
in 2020.  
[[Preliminary arXiv version](#)]

## In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri  
Microsoft Research India  
visesha@microsoft.com

Onur Mutlu  
ETH Zürich  
onur.mutlu@inf.ethz.ch



# SIMDRAM Framework

---

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, [\*\*"SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"\*\*](#) *Proceedings of the 26th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Virtual, March-April 2021.  
[[2-page Extended Abstract](#)]  
[[Short Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Short Talk Video](#) (5 mins)]  
[[Full Talk Video](#) (27 mins)]

## SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM

*Nastaran Hajinazar <sup>1,2</sup>	*Geraldo F. Oliveira <sup>1</sup>	Sven Gregorio <sup>1</sup>	João Dinis Ferreira <sup>1</sup>
Nika Mansouri Ghiasi <sup>1</sup>	Minesh Patel <sup>1</sup>	Mohammed Alser <sup>1</sup>	Saugata Ghose <sup>3</sup>
	Juan Gómez-Luna <sup>1</sup>	Onur Mutlu <sup>1</sup>	

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

<sup>3</sup>University of Illinois at Urbana–Champaign

# Bulk Data Copy and Initialization in DRAM

---

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry,  
**"RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization"**  
*Proceedings of the 46th International Symposium on Microarchitecture (MICRO)*, Davis, CA, December 2013. [[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)]

## RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization

Vivek Seshadri      Yoongu Kim      Chris Fallin\*      Donghyuk Lee  
vseshadr@cs.cmu.edu    yoongukim@cmu.edu    cfallin@c1f.net    donghyuk1@cmu.edu

Rachata Ausavarungnirun      Gennady Pekhimenko      Yixin Luo  
rachata@cmu.edu      gpekhime@cs.cmu.edu      yixinluo@andrew.cmu.edu

Onur Mutlu      Phillip B. Gibbons†      Michael A. Kozuch†      Todd C. Mowry  
onur@cmu.edu    phillip.b.gibbons@intel.com    michael.a.kozuch@intel.com    tcm@cs.cmu.edu

Carnegie Mellon University    †Intel Pittsburgh

# LISA: Increasing Connectivity in DRAM

---

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,  
**"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"**  
*Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA)*, Barcelona, Spain, March 2016.  
[[Slides \(pptx\)](#)] [[pdf](#)]  
[[Source Code](#)]

## Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang<sup>†</sup>, Prashant J. Nair<sup>\*</sup>, Donghyuk Lee<sup>†</sup>, Saugata Ghose<sup>†</sup>, Moinuddin K. Qureshi<sup>\*</sup>, and Onur Mutlu<sup>†</sup>

<sup>†</sup>Carnegie Mellon University    <sup>\*</sup>Georgia Institute of Technology

# FIGARO: Fine-Grained In-DRAM Copy

---

- Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu,  
**"FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching"**  
*Proceedings of the 53rd International Symposium on Microarchitecture (MICRO)*, Virtual, October 2020.

## FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching

Yaohua Wang<sup>\*</sup> Lois Orosa<sup>†</sup> Xiangjun Peng<sup>⊙\*</sup> Yang Guo<sup>\*</sup> Saugata Ghose<sup>◇‡</sup> Minesh Patel<sup>†</sup>  
Jeremie S. Kim<sup>†</sup> Juan Gómez Luna<sup>†</sup> Mohammad Sadrosadati<sup>§</sup> Nika Mansouri Ghiasi<sup>†</sup> Onur Mutlu<sup>†‡</sup>

<sup>\*</sup>National University of Defense Technology <sup>†</sup>ETH Zürich <sup>⊙</sup>Chinese University of Hong Kong

<sup>◇</sup>University of Illinois at Urbana–Champaign <sup>‡</sup>Carnegie Mellon University <sup>§</sup>Institute of Research in Fundamental Sciences

# Network-On-Memory: Fast Inter-Bank Copy

---

- Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Modarressi, Rachata Ausavarungnirun, Mohammad Sadrosadati, Onur Mutlu, and Masoud Daneshtalab,  
["NoM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories"](#)  
[IEEE Computer Architecture Letters](#) (**CAL**), to appear in 2020.

## NOm: NETWORK-ON-MEMORY FOR INTER-BANK DATA TRANSFER IN HIGHLY-BANKED MEMORIES

Seyyed Hossein SeyyedAghaei Rezaei<sup>1</sup>  
Mohammad Sadrosadati<sup>3</sup>

Mehdi Modarressi<sup>1,3</sup>  
Onur Mutlu<sup>4</sup>

Rachata Ausavarungnirun<sup>2</sup>  
Masoud Daneshtalab<sup>5</sup>

<sup>1</sup>University of Tehran

<sup>2</sup>King Mongkut's University of Technology North Bangkok

<sup>3</sup>Institute for Research in Fundamental Sciences

<sup>4</sup>ETH Zürich

<sup>5</sup>Mälardalens University



# In-DRAM Physical Unclonable Functions

---

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,  
**"The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"**  
*Proceedings of the 24th International Symposium on High-Performance Computer Architecture (HPCA)*, Vienna, Austria, February 2018.  
[[Lightning Talk Video](#)]  
[[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)]  
[[Full Talk Lecture Video](#) (28 minutes)]

## The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions

by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim<sup>†§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich



# In-DRAM True Random Number Generation

---

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, **"D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"**

*Proceedings of the 25th International Symposium on High-Performance Computer Architecture (HPCA), Washington, DC, USA, February 2019.*

[[Slides \(pptx\)](#) ([pdf](#))]

[[Full Talk Video](#) (21 minutes)]

[[Full Talk Lecture Video](#) (27 minutes)]

***Top Picks Honorable Mention by IEEE Micro.***

## D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim<sup>‡§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§‡</sup>

<sup>‡</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich

# Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory

# PIM Review and Open Problems

---

## A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirun<sup>d</sup>

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

<sup>b</sup>*Carnegie Mellon University*

<sup>c</sup>*University of Illinois at Urbana-Champaign*

<sup>d</sup>*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,

**"A Modern Primer on Processing in Memory"**

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, to be published in 2021.*

# A Modern Primer on Processing in Memory

Onur Mutlu<sup>a,b</sup>, Saugata Ghose<sup>b,c</sup>, Juan Gómez-Luna<sup>a</sup>, Rachata Ausavarungnirund<sup>d</sup>

SAFARI Research Group

<sup>a</sup>ETH Zürich

<sup>b</sup>Carnegie Mellon University

<sup>c</sup>University of Illinois at Urbana-Champaign

<sup>d</sup>King Mongkut's University of Technology North Bangkok

---

## Abstract

Modern computing systems are overwhelmingly designed to move data to computation. This design choice goes directly against at least three key trends in computing that cause performance, scalability and energy bottlenecks: (1) data access is a key bottleneck as many important applications are increasingly data-intensive, and memory bandwidth and energy do not scale well, (2) energy consumption is a key limiter in almost all computing platforms, especially server and mobile systems, (3) data movement, especially off-chip to on-chip, is very expensive in terms of bandwidth, energy and latency, much more so than computation. These trends are especially severely-felt in the data-intensive server and energy-constrained mobile systems of today.

At the same time, conventional memory technology is facing many technology scaling challenges in terms of reliability, energy, and performance. As a result, memory system architects are open to organizing memory in different ways and making it more intelligent, at the expense of higher cost. The emergence of 3D-stacked memory plus logic, the adoption of error correcting codes inside the latest DRAM chips, proliferation of different main memory standards and chips, specialized for different purposes (e.g., graphics, low-power, high bandwidth, low latency), and the necessity of designing new solutions to serious reliability and security issues, such as the RowHammer phenomenon, are an evidence of this trend.

This chapter discusses recent research that aims to practically enable computation close to data, an approach we call *processing-in-memory* (PIM). PIM places computation mechanisms in or near where the data is stored (i.e., inside the memory chips, in the logic layer of 3D-stacked memory, or in the memory controllers), so that data movement between the computation units and memory is reduced or eliminated. While the general idea of PIM is not new, we discuss motivating trends in applications as well as memory circuits/technology that greatly exacerbate the need for enabling it in modern computing systems. We examine at least two promising new approaches to designing PIM systems to accelerate important data-intensive applications: (1) *processing using memory* by exploiting analog operational properties of DRAM chips to perform massively-parallel operations in memory, with low-cost changes, (2) *processing near memory* by exploiting 3D-stacked memory technology design to provide high memory bandwidth and low memory latency to in-memory logic. In both approaches, we describe and tackle relevant cross-layer research, design, and adoption challenges in devices, architecture, systems, and programming models. Our focus is on the development of in-memory processing designs that can be adopted in real computing platforms at low cost. We conclude by discussing work on solving key challenges to the practical adoption of PIM.

**Keywords:** memory systems, data movement, main memory, processing-in-memory, near-data processing, computation-in-memory, processing using memory, processing near memory, 3D-stacked memory, non-volatile memory, energy efficiency, high-performance computing, computer architecture, computing paradigm, emerging technologies, memory scaling, technology scaling, dependable systems, robust systems, hardware security, system security, latency, low-latency computing



<b>1 Introduction</b>	<b>2</b>
<b>2 Major Trends Affecting Main Memory</b>	<b>4</b>
<b>3 The Need for Intelligent Memory Controllers to Enhance Memory Scaling</b>	<b>6</b>
<b>4 Perils of Processor-Centric Design</b>	<b>9</b>
<b>5 Processing-in-Memory (PIM): Technology Enablers and Two Approaches</b>	<b>12</b>
5.1 New Technology Enablers: 3D-Stacked Memory and Non-Volatile Memory . . .	12
5.2 Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM) . . . . .	13
<b>6 Processing Using Memory (PUM)</b>	<b>14</b>
6.1 RowClone . . . . .	14
6.2 Ambit . . . . .	15
6.3 Gather-Scatter DRAM . . . . .	17
6.4 In-DRAM Security Primitives . . . . .	17
<b>7 Processing Near Memory (PNM)</b>	<b>18</b>
7.1 Tesseract: Coarse-Grained Application-Level PNM Acceleration of Graph Processing . . . . .	19
7.2 Function-Level PNM Acceleration of Mobile Consumer Workloads . . . . .	20
7.3 Programmer-Transparent Function-Level PNM Acceleration of GPU Applications . . . . .	21
7.4 Instruction-Level PNM Acceleration with PIM-Enabled Instructions (PEI) . .	21
7.5 Function-Level PNM Acceleration of Genome Analysis Workloads . . . . .	22
7.6 Application-Level PNM Acceleration of Time Series Analysis . . . . .	23
<b>8 Enabling the Adoption of PIM</b>	<b>24</b>
8.1 Programming Models and Code Generation for PIM . . . . .	24
8.2 PIM Runtime: Scheduling and Data Mapping . . . . .	25
8.3 Memory Coherence . . . . .	27
8.4 Virtual Memory Support . . . . .	27
8.5 Data Structures for PIM . . . . .	28
8.6 Benchmarks and Simulation Infrastructures . . . . .	29
8.7 Real PIM Hardware Systems and Prototypes . . . . .	30
8.8 Security Considerations . . . . .	30
<b>9 Conclusion and Future Outlook</b>	<b>31</b>

Main memory, built using the Dynamic Random Access Memory (DRAM) technology, is a major component in nearly all computing systems, including servers, cloud platforms, mobile/embedded devices, and sensor systems. Across all of these systems, the data working set sizes of modern applications are rapidly growing, while the need for fast analysis of such data is increasing. Thus, main memory is becoming an increasingly significant bottleneck across a wide variety of computing systems and applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Alleviating the main memory bottleneck requires the memory capacity, energy, cost, and performance to all scale in an efficient manner across technology generations. Unfortunately, it has become increasingly difficult in recent years, especially the past decade, to scale all of these dimensions [1, 2, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49], and thus the main memory bottleneck has been worsening.

A major reason for the main memory bottleneck is the high energy and latency cost associated with *data movement*. In modern computers, to perform any operation on data that resides in main memory, the processor must retrieve the data from main memory. This requires the memory controller to issue commands to a DRAM module across a relatively slow and power-hungry off-chip bus (known as the *memory channel*). The DRAM module sends the requested data across the memory channel, after which the data is placed in the caches and registers. The CPU can perform computation on the data once the data is in its registers. Data movement from the DRAM to the CPU incurs long latency and consumes a significant amount of energy [7, 50, 51, 52, 53, 54]. These costs are often exacerbated by the fact that much of the data brought into the caches is *not reused* by the CPU [52, 53, 55, 56], providing little benefit in return for the high latency and energy cost.

The cost of data movement is a fundamental issue with the *processor-centric* nature of contemporary computer systems. The CPU is considered to be the master in the system, and computation is performed only in the processor (and accelerators). In contrast, data storage and communication units, including the main memory, are treated as unintelligent workers that are incapable of computation. As a result of this processor-centric design paradigm, data moves a lot in the system between the computation units and communication/ storage units so that computation can be done on it. With the increasingly *data-centric* nature of contemporary and emerging appli-

# PIM Review and Open Problems (II)

---

## A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose<sup>†</sup>   Amirali Boroumand<sup>†</sup>   Jeremie S. Kim<sup>†§</sup>   Juan Gómez-Luna<sup>§</sup>   Onur Mutlu<sup>§†</sup>

<sup>†</sup>*Carnegie Mellon University*

<sup>§</sup>*ETH Zürich*

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,

**"Processing-in-Memory: A Workload-Driven Perspective"**

*Invited Article in IBM Journal of Research & Development, Special Issue on Hardware for Artificial Intelligence, to appear in November 2019.*

[Preliminary arXiv version]



# A Tutorial on PIM

---

- Onur Mutlu,

## **"Memory-Centric Computing Systems"**

Invited Tutorial at *66th International Electron Devices Meeting (IEDM)*, Virtual, 12 December 2020.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Executive Summary Slides \(pptx\)](#) ([pdf](#))]

[[Tutorial Video](#) (1 hour 51 minutes)]

[[Executive Summary Video](#) (2 minutes)]

[[Abstract and Bio](#)]

[[Related Keynote Paper from VLSI-DAT 2020](#)]

[[Related Review Paper on Processing in Memory](#)]

<https://www.youtube.com/watch?v=H3sEaINPBOE>

# Memory-Centric Computing Systems



Onur Mutlu

[omutlu@gmail.com](mailto:omutlu@gmail.com)

<https://people.inf.ethz.ch/omutlu>

12 December 2020

IEDM Tutorial

**SAFARI**

**ETH** zürich

Carnegie Mellon



0:06 / 1:51:05



IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

48 0 SHARE SAVE ...



Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/onurmutlulectures>

156

# Detailed Lectures on PIM (I)

---

- **Computer Architecture, Fall 2020, Lecture 6**
  - **Computation in Memory** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=oGcZAGwfEUE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=12>
- **Computer Architecture, Fall 2020, Lecture 7**
  - **Near-Data Processing** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=13>
- **Computer Architecture, Fall 2020, Lecture 11a**
  - **Memory Controllers** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=20>
- **Computer Architecture, Fall 2020, Lecture 12d**
  - **Real Processing-in-DRAM with UPMEM** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25>

# Detailed Lectures on PIM (II)

---

- **Computer Architecture, Fall 2020, Lecture 15**
  - **Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=AIE1rD9G\\_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28](https://www.youtube.com/watch?v=AIE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28)
- **Computer Architecture, Fall 2020, Lecture 16a**
  - **Opportunities & Challenges of Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29>
- **Computer Architecture, Fall 2020, Guest Lecture**
  - **In-Memory Computing: Memory Devices & Applications** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=wNmQqHiEZnk&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=41>

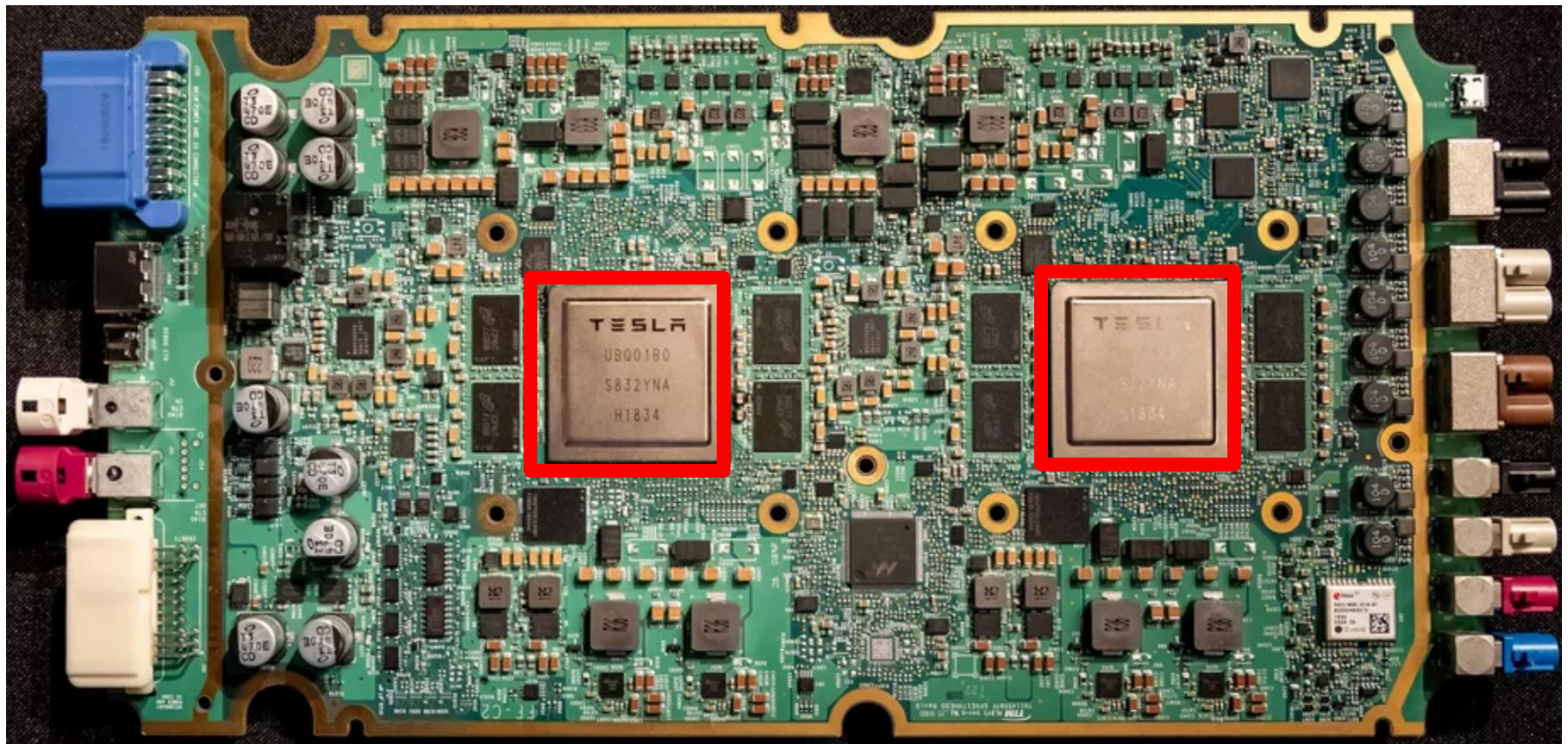
Many Interesting Things  
Are Happening Today  
in Computer Architecture

**Performance  
and  
Energy Efficiency**

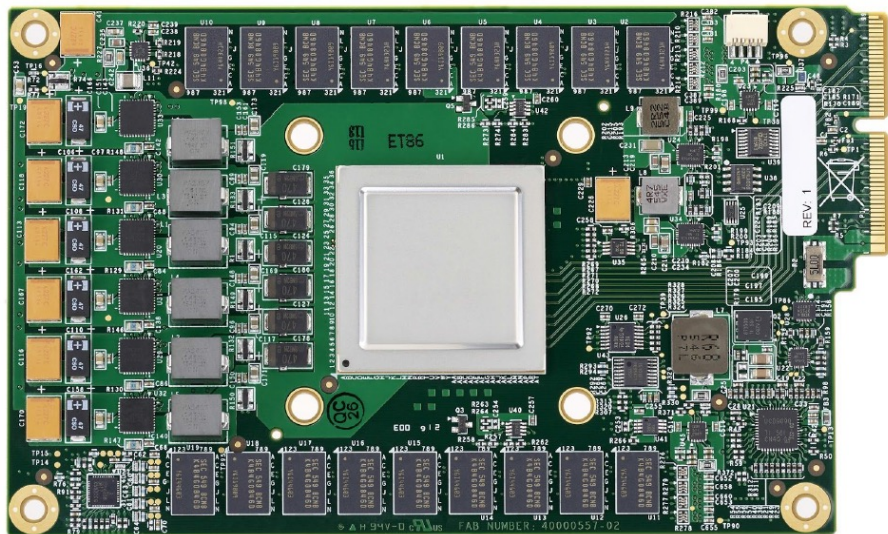


# TESLA Full Self-Driving Computer (2019)

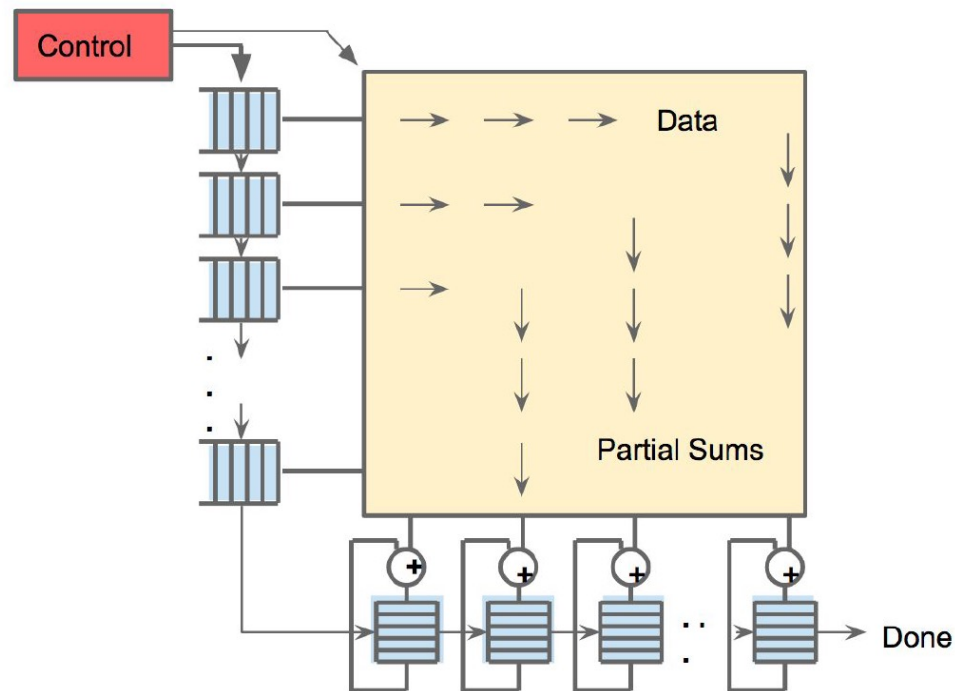
- ML accelerator: 260 mm<sup>2</sup>, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



# 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.



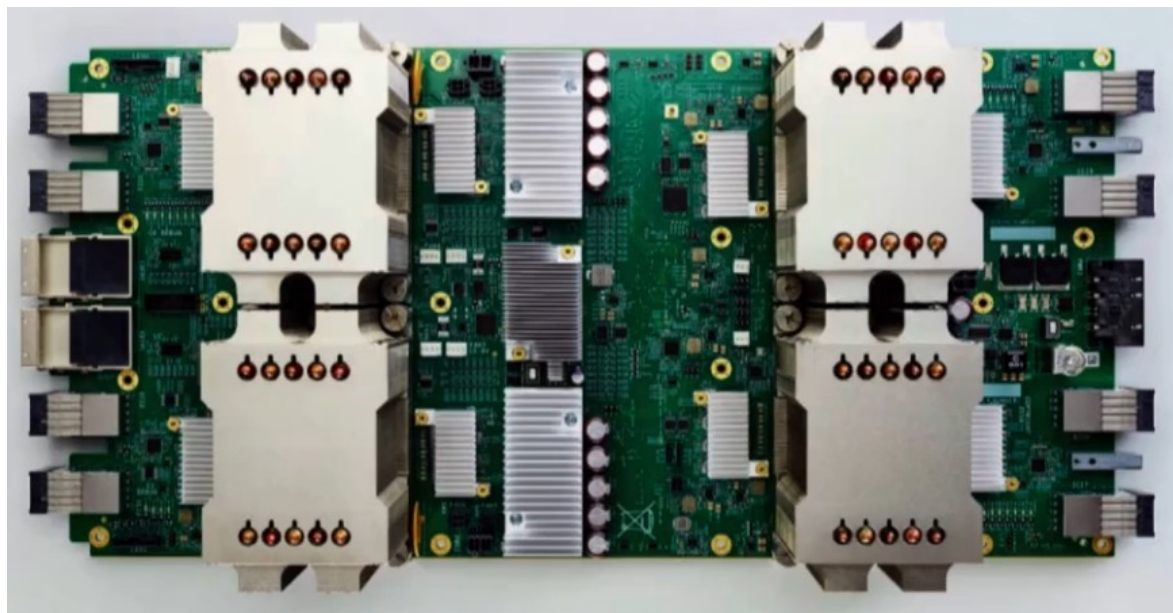
**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., “In-Datcenter 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

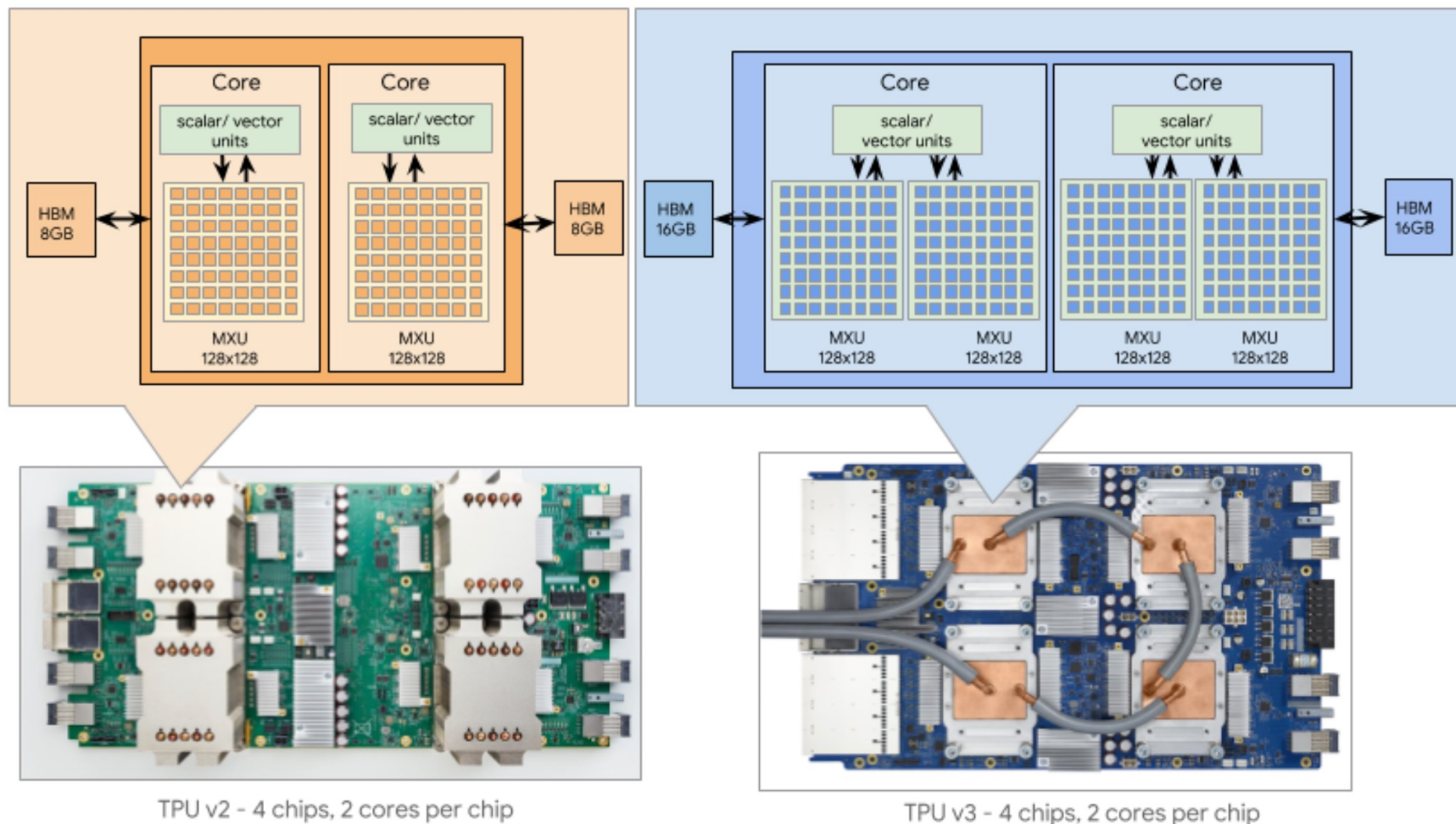
High Bandwidth Memory  
vs DDR3

Floating point operations  
vs FP16

45 TFLOPS per chip  
vs 23 TOPS

Designed for **training**  
and **inference**  
vs only inference

# Google TPU Generation III (2019)



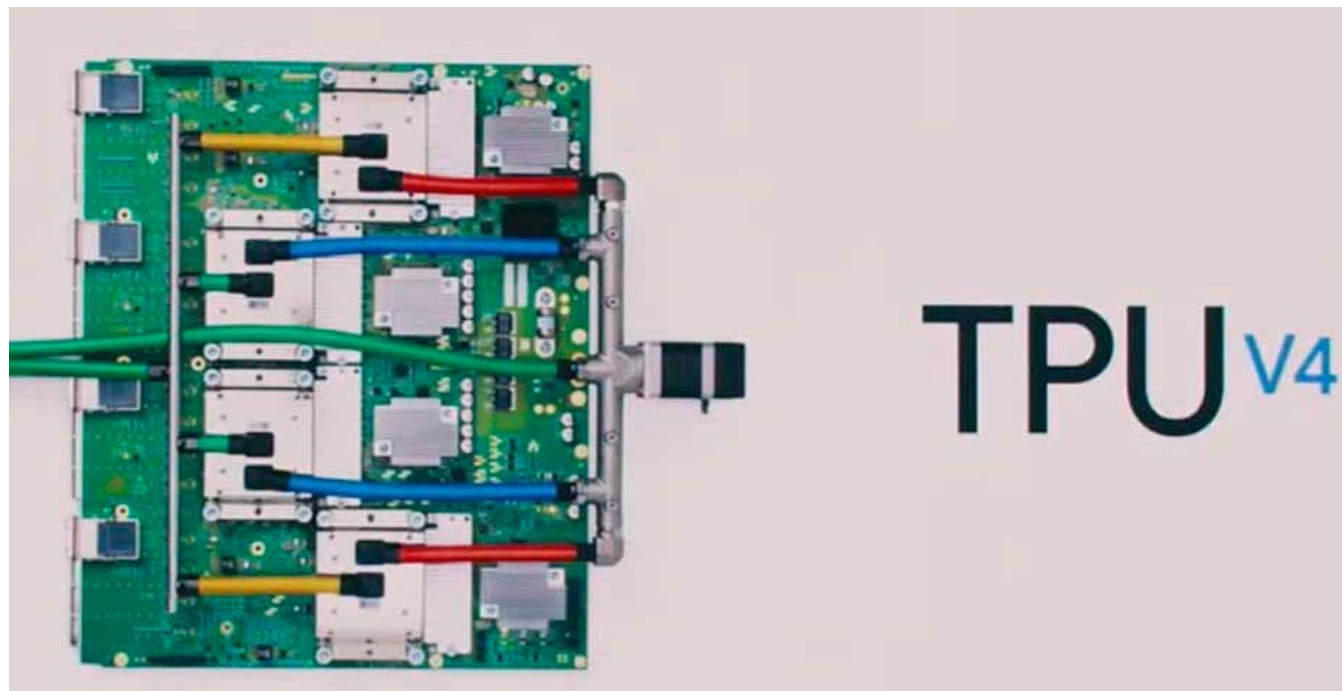
32GB HBM per chip  
vs 16GB HBM in TPU2

4 Matrix Units per chip  
vs 2 Matrix Units in TPU2

90 TFLOPS per chip  
vs 45 TFLOPS in TPU2

# Google TPU Generation IV (2021)

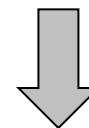
---



## 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



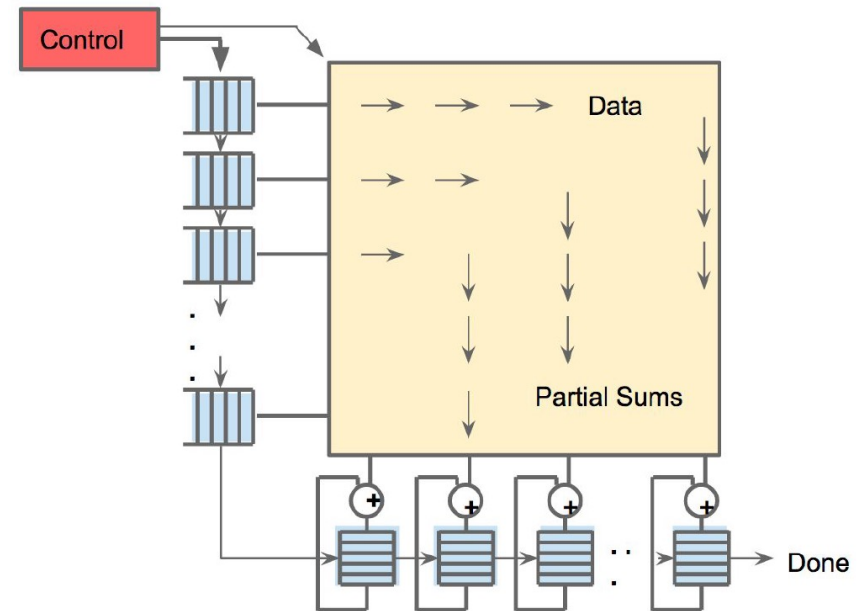
1 ExaFLOPS per board

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



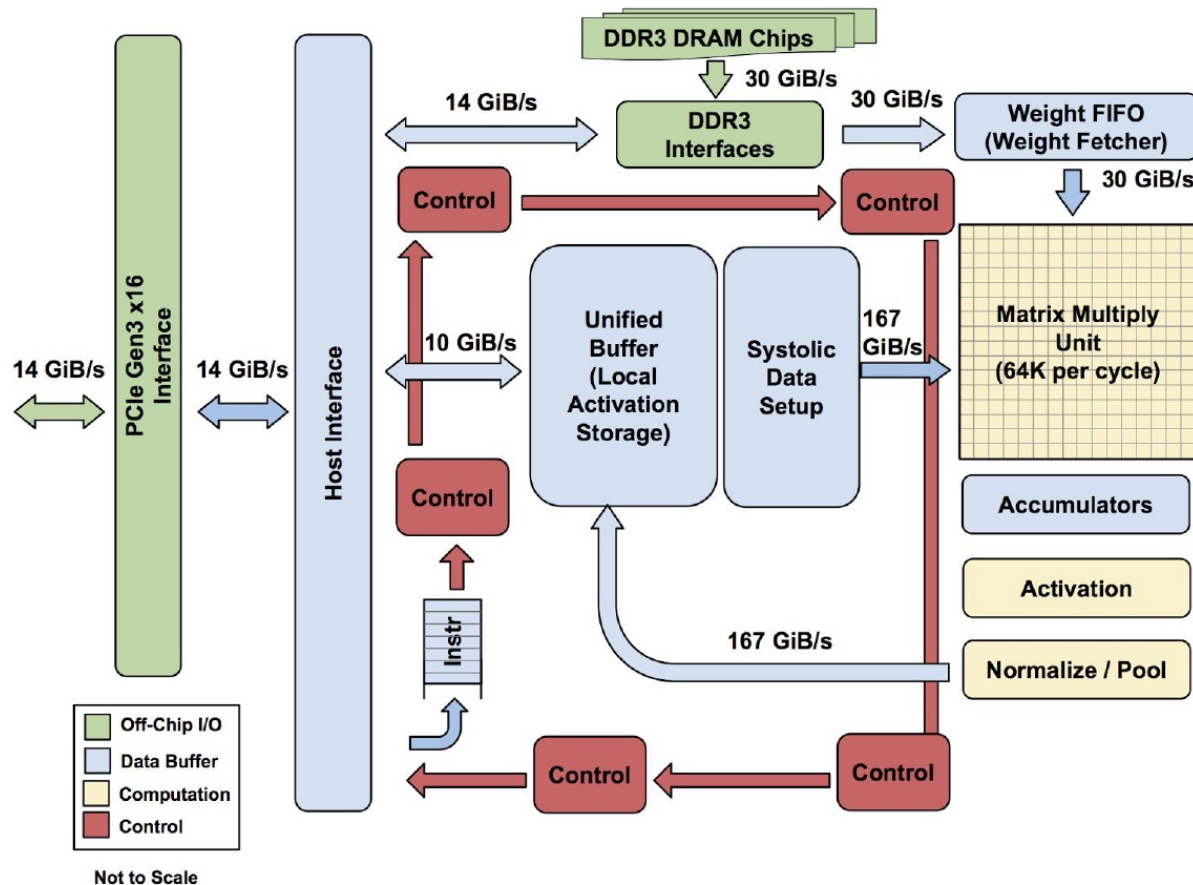
# An Example Modern Systolic Array: TPU (II)

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)



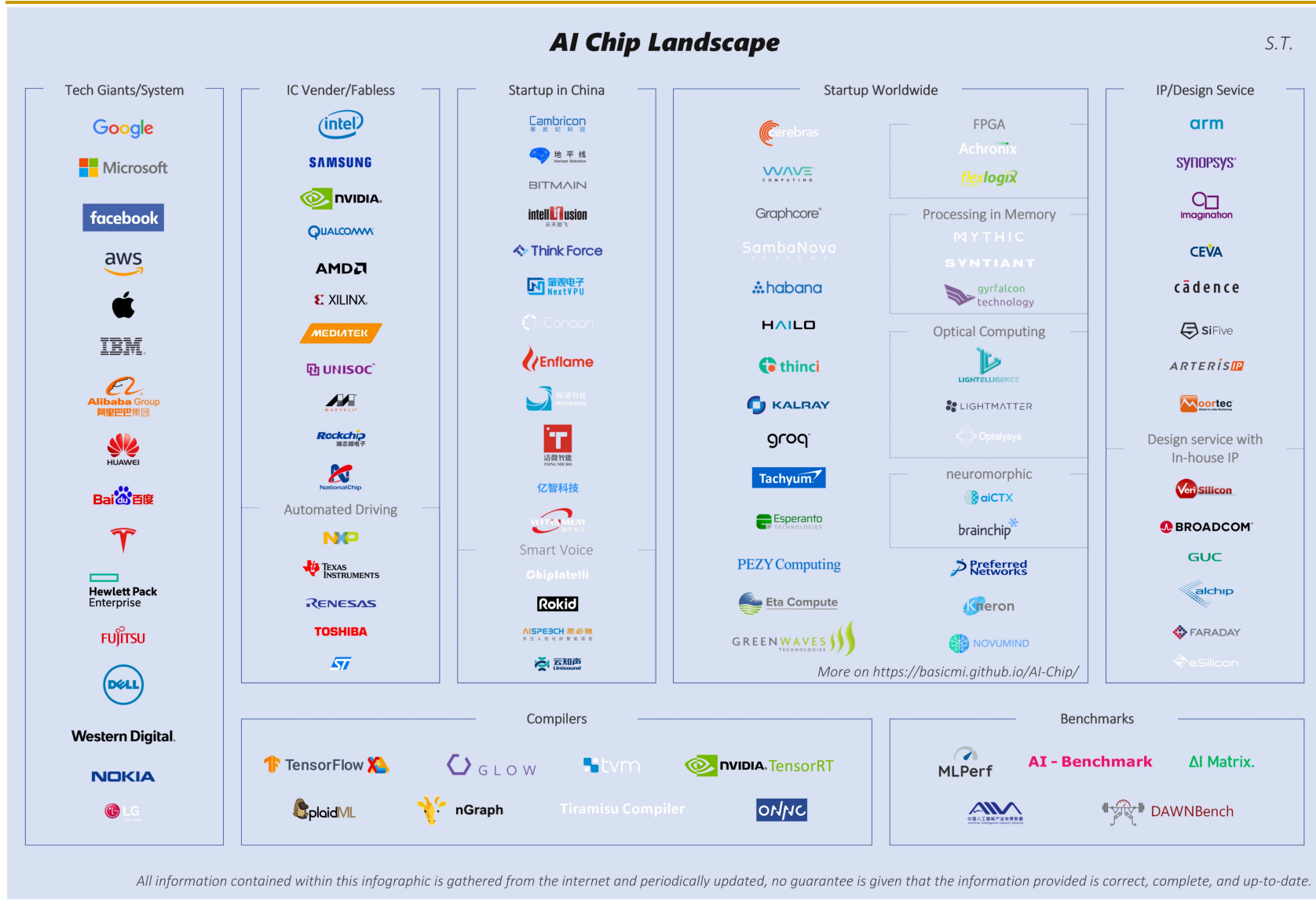
**Figure 1.** TPU Block Diagram. The main computation part is the yellow Matrix Multiply unit in the upper right hand corner. Its inputs are the blue Weight FIFO and the blue Unified Buffer (UB) and its output is the blue Accumulators (Acc). The yellow Activation Unit performs the nonlinear functions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

# Many (Other) AI/ML Chips

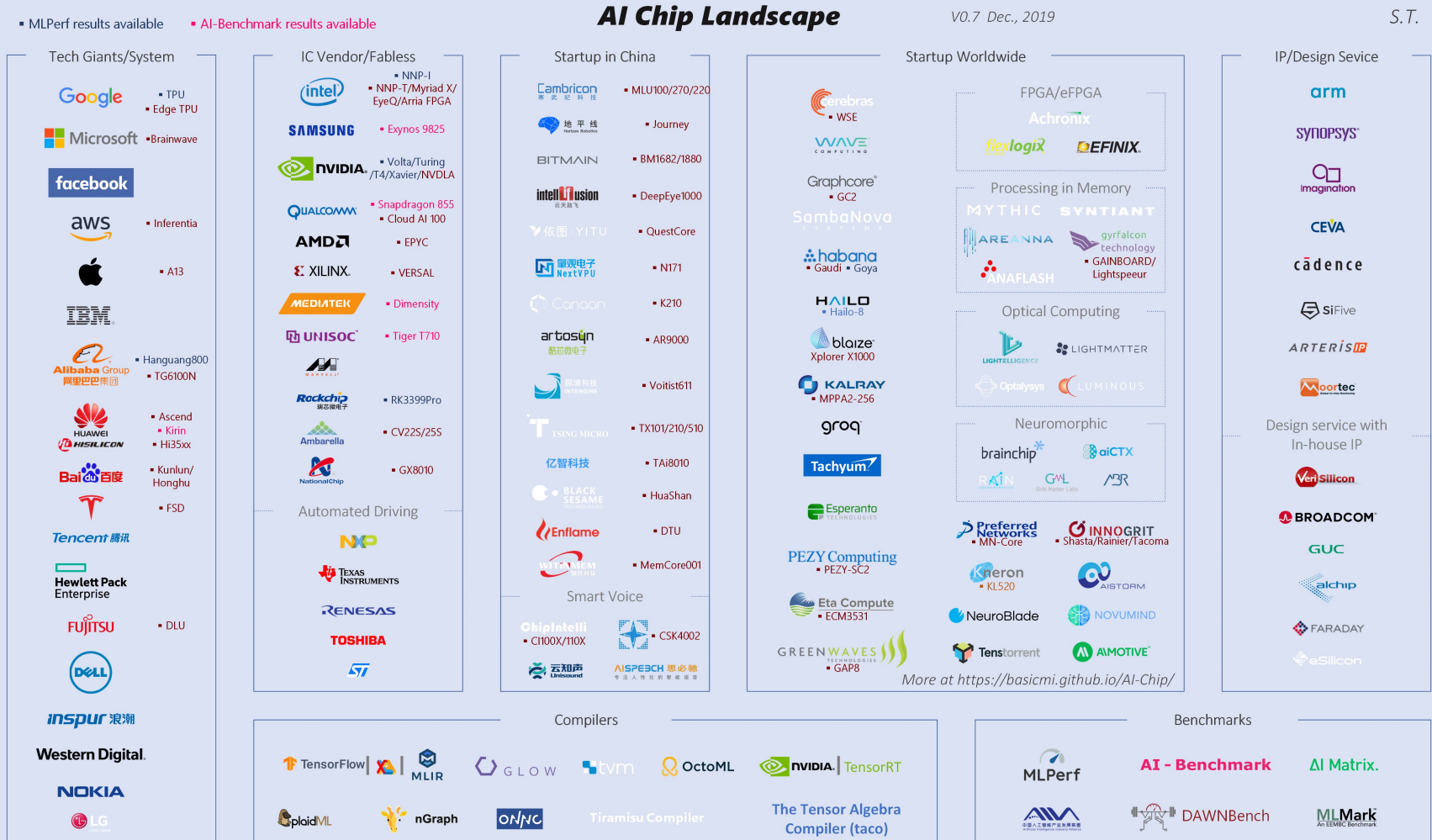
---

- Alibaba
- Amazon
- Facebook
- Google
- Huawei
- Intel
- Microsoft
- NVIDIA
- Tesla
- Many Others and Many Startups...
- **Many More to Come...**

# Many (Other) AI/ML Chips (2019)



# Many (Other) AI/ML Chips (2021)



All information contained within this infographic is gathered from the internet and periodically updated, no guarantee is given that the information provided is correct, complete, and up-to-date.



# Computer Architecture

## Lecture 1: Introduction and Basics

Prof. Onur Mutlu

ETH Zürich

Fall 2021

30 September 2021

**Further Slides for Your Own Study  
(May Be Covered in Future Lectures)**

# Many Interesting Things Are Happening Today in Computer Architecture

# Many Interesting Things Are Happening Today in Computer Architecture

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

# 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



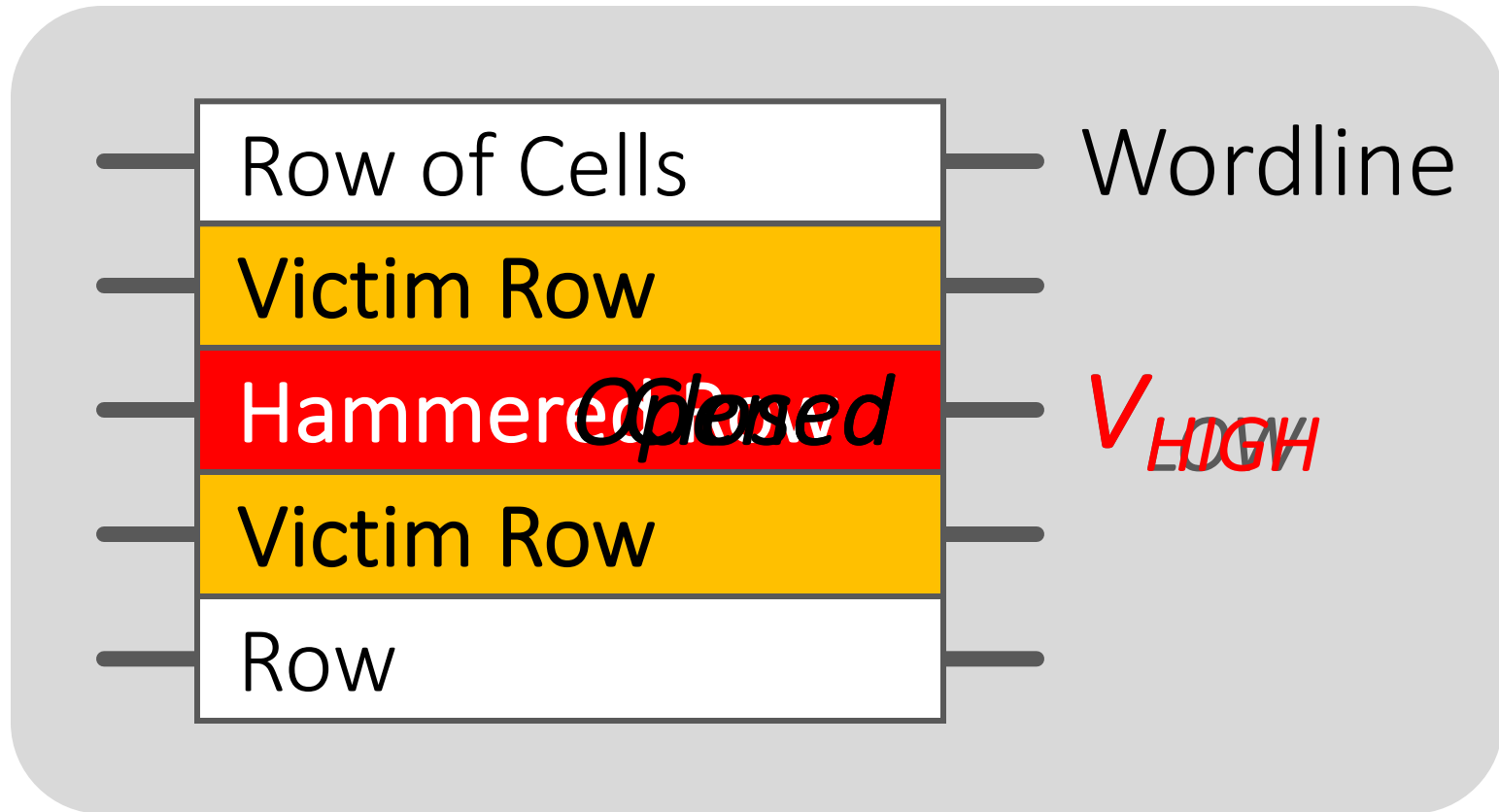
SHARE  
18276



TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

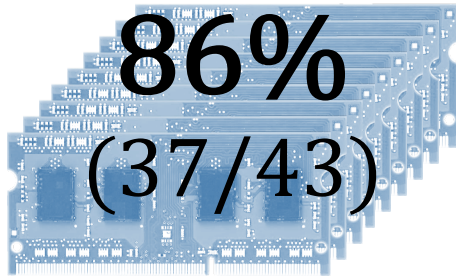
# Modern DRAM is Prone to Disturbance Errors



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**

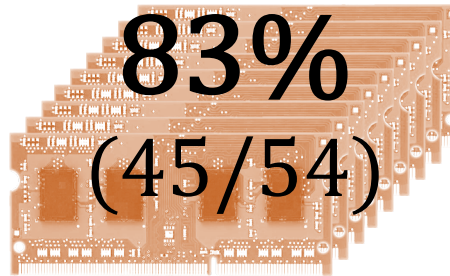
# Most DRAM Modules Are Vulnerable

A company



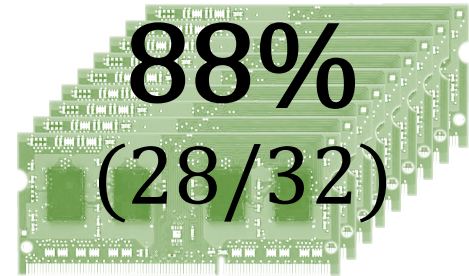
Up to  
 $1.0 \times 10^7$   
errors

B company



Up to  
 $2.7 \times 10^6$   
errors

C company



Up to  
 $3.3 \times 10^5$   
errors

# 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



# More Security Implications (I)

**“We can gain unrestricted access to systems of website visitors.”**

www.iaik.tugraz.at ■

Not there yet, but ...



ROOT privileges for web apps!

29

Daniel Gruss (@lavados), Clémentine Maurice (@BloodyTangerine),  
December 28, 2015 — 32c3, Hamburg, Germany



GATED  
COMMUNITIES

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA'16)

# More Security Implications (II)

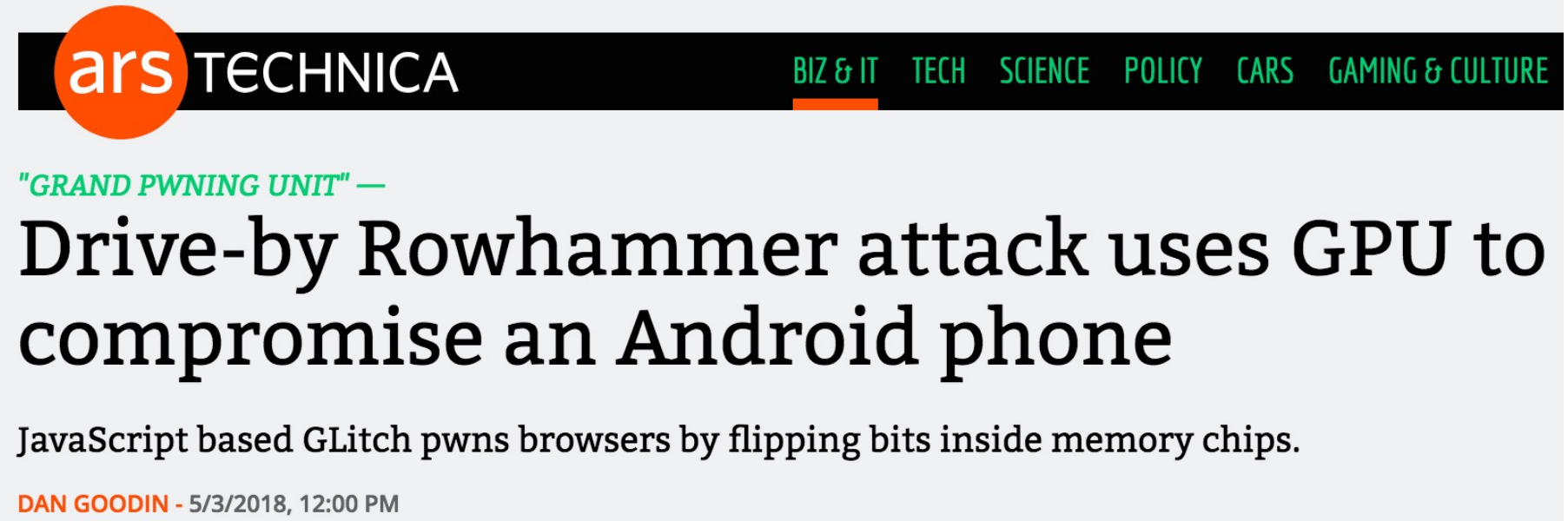
**"Can gain control of a smart phone deterministically"**



Drammer: Deterministic Rowhammer  
Attacks on Mobile Platforms, CCS'16<sup>181</sup>

# More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface



The image is a screenshot of an Ars Technica article. At the top, the Ars Technica logo is on the left, and a navigation bar with links for BIZ & IT, TECH, SCIENCE, POLICY, CARS, and GAMING & CULTURE is on the right. Below the navigation bar, the article title "Drive-by Rowhammer attack uses GPU to compromise an Android phone" is displayed in large, bold black text. Above the title, the subtitle "GRAND PWINING UNIT" — is shown in green. Below the title, a summary line reads "JavaScript based GLitch pwns browsers by flipping bits inside memory chips." At the bottom left of the article preview, the author "DAN GOODIN" and the date "5/3/2018, 12:00 PM" are listed.

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

"GRAND PWINING UNIT" —

## Drive-by Rowhammer attack uses GPU to compromise an Android phone

JavaScript based GLitch pwns browsers by flipping bits inside memory chips.

DAN GOODIN - 5/3/2018, 12:00 PM

## Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

Pietro Frigo  
Vrije Universiteit  
Amsterdam  
p.frigo@vu.nl

Cristiano Giuffrida  
Vrije Universiteit  
Amsterdam  
giuffrida@cs.vu.nl

Herbert Bos  
Vrije Universiteit  
Amsterdam  
herbertb@cs.vu.nl

Kaveh Razavi  
Vrije Universiteit  
Amsterdam  
kaveh@cs.vu.nl

# More Security Implications (IV)

## ■ Rowhammer over RDMA (I)



TECHNICA

BIZ & IT

TECH

SCIENCE

POLICY

CARS

GAMING & CULTURE

THROWHAMMER —

# Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM

## Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar  
*VU Amsterdam*

Radhesh Krishnan  
*VU Amsterdam*

Elias Athanasopoulos  
*University of Cyprus*

Cristiano Giuffrida  
*VU Amsterdam*

Herbert Bos  
*VU Amsterdam*

Kaveh Razavi  
*VU Amsterdam*



# More Security Implications (V)

---

## ■ Rowhammer over RDMA (II)



**Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests**



## **Nethammer: Inducing Rowhammer Faults through Network Requests**

Moritz Lipp  
Graz University of Technology

Daniel Gruss  
Graz University of Technology

Misiker Tadesse Aga  
University of Michigan

Clémentine Maurice  
Univ Rennes, CNRS, IRISA

Michael Schwarz  
Graz University of Technology

Lukas Raab  
Graz University of Technology

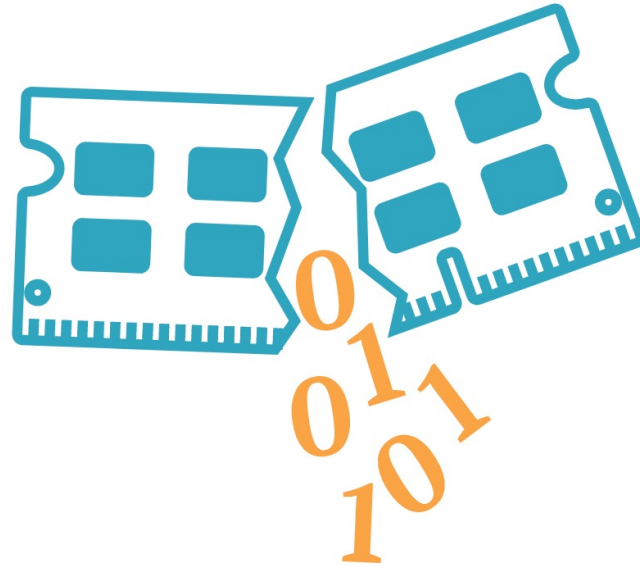
Lukas Lamster  
Graz University of Technology



# More Security Implications (VI)

---

- IEEE S&P 2020



RAMBleed

## RAMBleed: Reading Bits in Memory Without Accessing Them

Andrew Kwong  
*University of Michigan*  
[ankwong@umich.edu](mailto:ankwong@umich.edu)

Daniel Genkin  
*University of Michigan*  
[genkin@umich.edu](mailto:genkin@umich.edu)

Daniel Gruss  
*Graz University of Technology*  
[daniel.gruss@iaik.tugraz.at](mailto:daniel.gruss@iaik.tugraz.at)

Yuval Yarom  
*University of Adelaide and Data61*  
[yval@cs.adelaide.edu.au](mailto:yval@cs.adelaide.edu.au)

# More Security Implications (VII)

---

## ■ USENIX Security 2019

### **Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks**

Sanghyun Hong, Pietro Frigo<sup>†</sup>, Yiğitcan Kaya, Cristiano Giuffrida<sup>†</sup>, Tudor Dumitraş

*University of Maryland, College Park*

*<sup>†</sup>Vrije Universiteit Amsterdam*



#### **A Single Bit-flip Can Cause Terminal Brain Damage to DNNs**

*One specific bit-flip in a DNN's representation leads to accuracy drop over 90%*

Our research found that a specific bit-flip in a DNN's bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...

[Read More](#)

# More Security Implications (VIII)

## ■ USENIX Security 2020

### DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

Fan Yao  
University of Central Florida  
fan.yao@ucf.edu

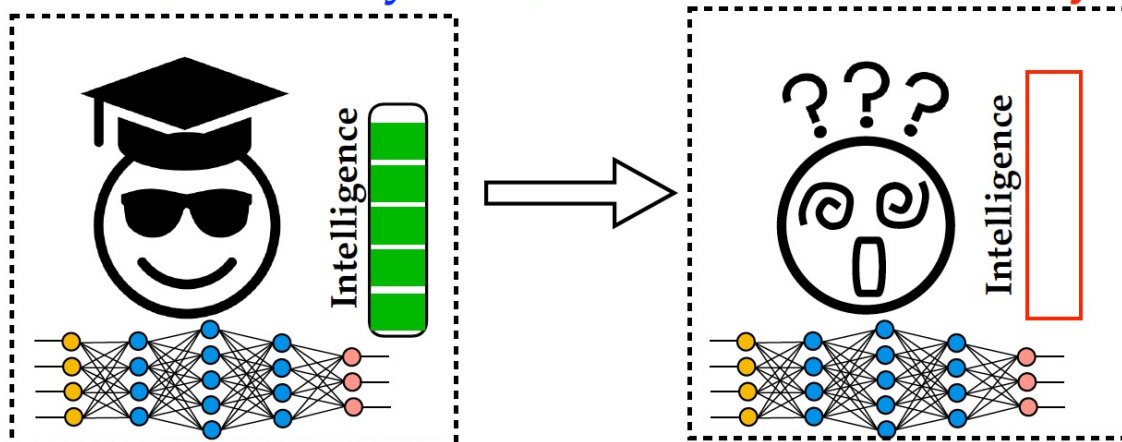
Adnan Siraj Rakin  
Arizona State University  
asrakin@asu.edu

Deliang Fan  
Arizona State University  
dfan@asu.edu

Degrade the inference accuracy to the level of Random Guess

Example: ResNet-20 for CIFAR-10, 10 output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**



# RowHammer: Seven 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<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

<sup>1</sup>Carnegie Mellon University   <sup>2</sup>Intel Labs



# RowHammer: 2019 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](#)]  
[[Slides from COSADE 2019 \(pptx\)](#)]  
[[Slides from VLSI-SOC 2020 \(pptx\) \(pdf\)](#)]  
[[Talk Video](#) (1 hr 15 minutes, with Q&A)]

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>      Jeremie S. Kim<sup>‡§</sup>  
<sup>§</sup>ETH Zürich      <sup>‡</sup>Carnegie Mellon University



# RowHammer in 2020

# 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<sup>§†</sup>      Minesh Patel<sup>§</sup>      A. Giray Yağlıkçı<sup>§</sup>  
Hasan Hassan<sup>§</sup>      Roknoddin Azizi<sup>§</sup>      Lois Orosa<sup>§</sup>      Onur Mutlu<sup>§†</sup>  
<sup>§</sup>*ETH Zürich*      <sup>†</sup>*Carnegie Mellon University*

# Key Takeaways from 1580 Chips

- **Newer DRAM chips are more vulnerable to RowHammer**
- There are chips today whose weakest cells fail after **only 4800 hammers**
- Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in **more rows** and 2) **farther away** from the victim row.
- **Existing mitigation mechanisms are NOT effective**

# 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](#)]  
[[Lecture Slides \(pptx\)](#)] [[pdf](#)]  
[[Talk Video](#)] (17 minutes)  
[[Lecture Video](#)] (59 minutes)  
[[Source Code](#)]  
[[Web Article](#)]  
***Best paper award.***  
***Pwnie Award 2020 for Most Innovative Research.*** [Pwnie Awards 2020](#)

## TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo<sup>\*†</sup>   Emanuele Vannacci<sup>\*†</sup>   Hasan Hassan<sup>§</sup>   Victor van der Veen<sup>¶</sup>  
Onur Mutlu<sup>§</sup>   Cristiano Giuffrida<sup>\*</sup>   Herbert Bos<sup>\*</sup>   Kaveh Razavi<sup>\*</sup>

# 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<sup>§†</sup>, Minesh Patel<sup>§</sup>, Lillian Tsai<sup>‡</sup>,  
Stefan Saroiu, Alec Wolman, and Onur Mutlu<sup>§†</sup>  
Microsoft Research, <sup>§</sup>ETH Zürich, <sup>†</sup>CMU, <sup>‡</sup>MIT



# BlockHammer Solution in 2021

---

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

## **"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"**

*Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.*

[[Slides \(pptx\)](#) ([pdf](#))]

[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (22 minutes)]

[[Short Talk Video](#) (7 minutes)]

## **BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows**

A. Giray Yağlıkçı<sup>1</sup> Minesh Patel<sup>1</sup> Jeremie S. Kim<sup>1</sup> Roknoddin Azizi<sup>1</sup> Ataberk Olgun<sup>1</sup> Lois Orosa<sup>1</sup>  
Hasan Hassan<sup>1</sup> Jisung Park<sup>1</sup> Konstantinos Kanellopoulos<sup>1</sup> Taha Shahroodi<sup>1</sup> Saugata Ghose<sup>2</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>University of Illinois at Urbana–Champaign

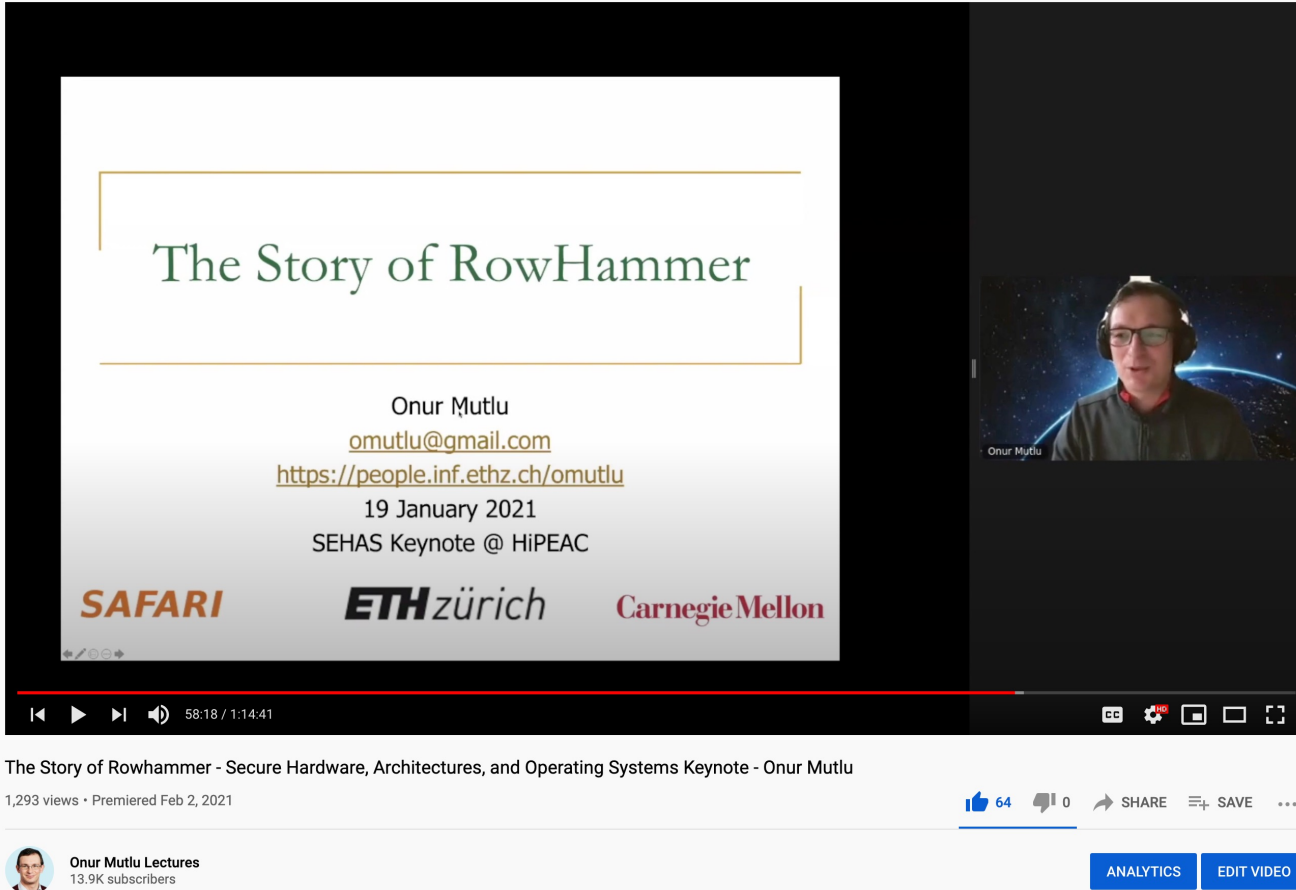
# Detailed Lectures on RowHammer

---

- **Computer Architecture, Fall 2020, Lecture 4b**
  - RowHammer (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=KDy632z23UE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=8>
- **Computer Architecture, Fall 2020, Lecture 5a**
  - RowHammer in 2020: TRRespass (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=pwRw7QqK\\_qA&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=9](https://www.youtube.com/watch?v=pwRw7QqK_qA&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=9)
- **Computer Architecture, Fall 2020, Lecture 5b**
  - RowHammer in 2020: Revisiting RowHammer (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=gR7XR-Eepcg&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=10>
- **Computer Architecture, Fall 2020, Lecture 5c**
  - Secure and Reliable Memory (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=HvswnsfG3oQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=11>

# The Story of RowHammer Lecture ...

- Onur Mutlu,  
**"The Story of RowHammer"**  
Keynote Talk at *Secure Hardware, Architectures, and Operating Systems Workshop (SeHAS)*, held with *HiPEAC 2021 Conference*, Virtual, 19 January 2021.  
[Slides (pptx)] (pdf)  
[Talk Video] (1 hr 15 minutes, with Q&A)]



The video player shows a presentation slide titled "The Story of RowHammer" by Onur Mutlu. The slide includes contact information: [omutlu@gmail.com](mailto:omutlu@gmail.com), <https://people.inf.ethz.ch/omutlu>, and the date 19 January 2021. It also mentions "SEHAS Keynote @ HiPEAC". Logos for SAFARI, ETH zürich, and Carnegie Mellon are at the bottom. The video player interface shows a progress bar at 58:18 / 1:14:41 and a video feed of Onur Mutlu on the right. Below the player, the video title is "The Story of Rowhammer - Secure Hardware, Architectures, and Operating Systems Keynote - Onur Mutlu", with 1,293 views and a premiere date of Feb 2, 2021. The channel "Onur Mutlu Lectures" has 13.9K subscribers. Interaction buttons for likes (64), comments (0), share, save, and analytics are visible.

The Story of RowHammer

Onur Mutlu  
[omutlu@gmail.com](mailto:omutlu@gmail.com)  
<https://people.inf.ethz.ch/omutlu>  
19 January 2021  
SEHAS Keynote @ HiPEAC

SAFARI ETH zürich Carnegie Mellon

58:18 / 1:14:41

The Story of Rowhammer - Secure Hardware, Architectures, and Operating Systems Keynote - Onur Mutlu

1,293 views • Premiered Feb 2, 2021

64 0 SHARE SAVE ...

Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS EDIT VIDEO



Rowhammer

# Two Upcoming RowHammer Papers at MICRO 2021

---

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, Onur Mutlu,  
**"A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"**  
*MICRO 2021*

## **A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses**

Lois Orosa\*  
ETH Zürich

A. Giray Yağlıkçı\*  
ETH Zürich

Haocong Luo  
ETH Zürich

Ataberk Olgun  
ETH Zürich, TOBB ETÜ

Jisung Park  
ETH Zürich

Hasan Hassan  
ETH Zürich

Minesh Patel  
ETH Zürich

Jeremie S. Kim  
ETH Zürich

Onur Mutlu  
ETH Zürich



# Two Upcoming RowHammer Papers at MICRO 2021

---

- Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, Onur Mutlu,

## **"Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications"**

*MICRO 2021*

## **Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications**

Hasan Hassan<sup>†</sup>

Yahya Can Tuğrul<sup>†‡</sup>

Jeremie S. Kim<sup>†</sup>

Victor van der Veen<sup>σ</sup>

Kaveh Razavi<sup>†</sup>

Onur Mutlu<sup>†</sup>

<sup>†</sup>*ETH Zürich*

<sup>‡</sup>*TOBB University of Economics & Technology*

<sup>σ</sup>*Qualcomm Technologies Inc.*

RowHammer is still  
an open problem

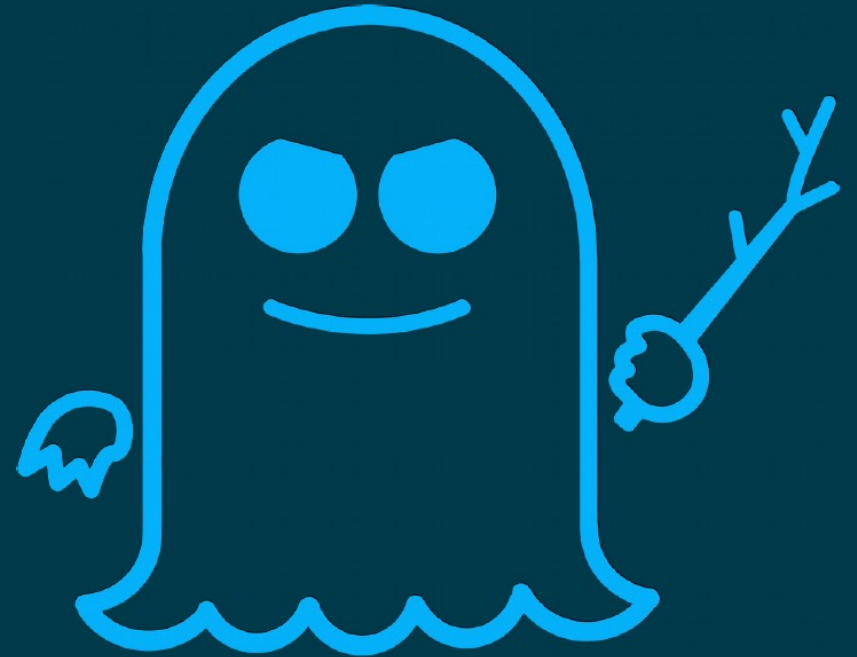
Security by obscurity  
is likely not a good solution

# Security: Meltdown and Spectre (2018)

---



**MELTDOWN**



**SPECTRE**

# 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.



# Many Interesting Things Are Happening Today in Computer Architecture

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**More Demanding Workloads**

# Increasingly Demanding Applications

---

Dream

and, they will come

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

# 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.

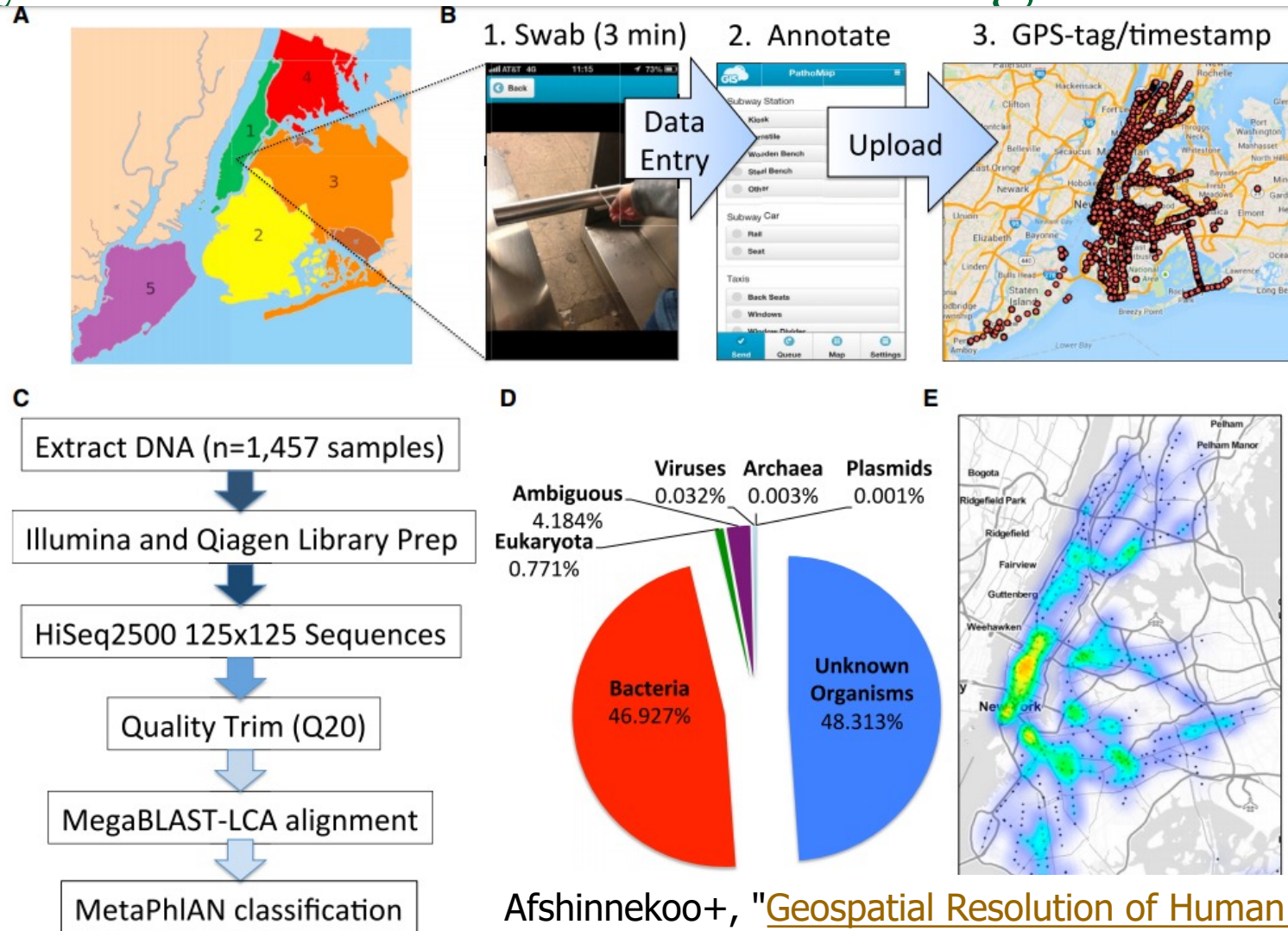


# Population-Scale Microbiome Profiling





# City-Scale Microbiome Profiling



**Figure 1. The Metagenome of New York City**

(A) The five boroughs of NYC include (1) Manhattan (green)

(B) The collection from the 466 subway stations of NYC across the 24 subway lines involved three main steps: (1) collection with Copan Elution swabs, (2) data entry into the database, and (3) uploading of the data. An image is shown of the current collection database, taken from <http://pathomap.giscloud.com>.

(C) Workflow for sample DNA extraction, library preparation, sequencing, quality trimming of the FASTQ files, and alignment with MegaBLAST and MetaPhlAn to discern taxa present

Afshinneko+, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics", Cell Systems, 2015

# Example: Rapid Surveillance of Ebola Outbreak

**Figure 1: Deployment of the portable genome surveillance system in Guinea.**



Quick+, "Real-time, portable genome sequencing for Ebola surveillance", *Nature*, 2016

# High-Throughput Genome Sequencers



Illumina MiSeq



Pacific  
Biosciences  
Sequel II

Oxford  
Nanopore  
PromethION



Oxford Nanopore MinION



Illumina NovaSeq 6000



Pacific Biosciences RS II



Oxford  
Nanopore  
SmidgION

**... and more! All produce data with different properties.**



# High-Throughput Genome Sequencers

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu  
[“Accelerating Genome Analysis: A Primer on an Ongoing Journey”](#) IEEE Micro, August 2020.



MinION from ONT

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](https://doi.org/10.1109/MM.2020.3013728)

## FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications

July-Aug. 2021, pp. 39-48, vol. 41

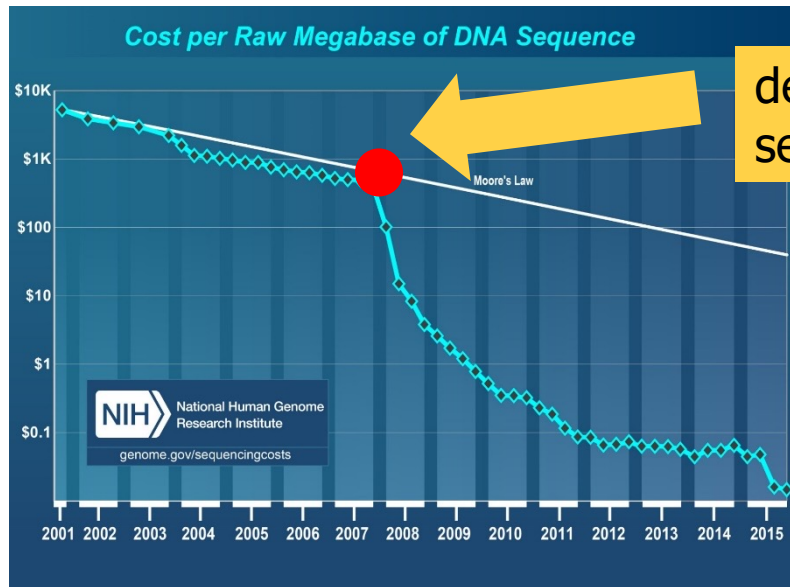
DOI Bookmark: [10.1109/MM.2021.3088396](https://doi.org/10.1109/MM.2021.3088396)



SmidgION from ONT

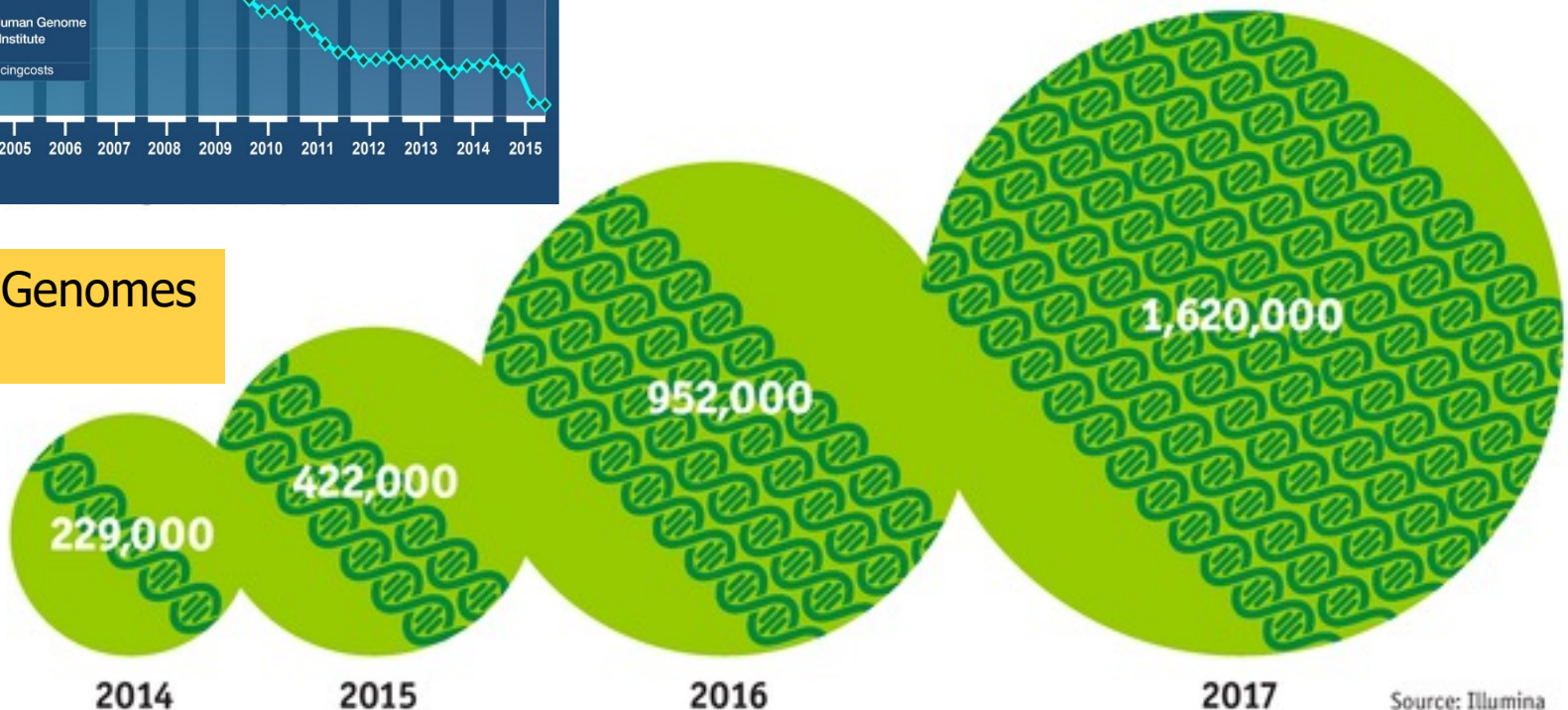


# The Genomic Era

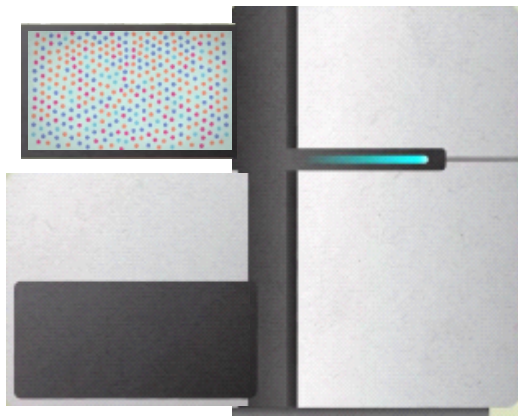


development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

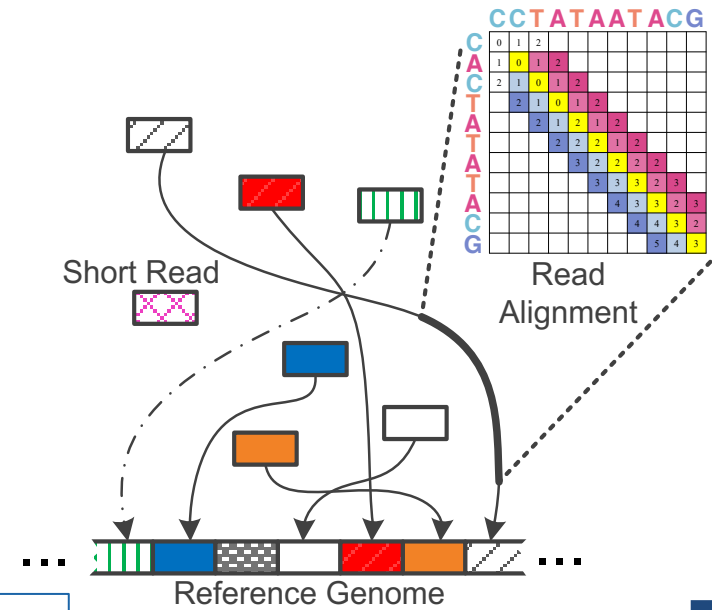


The Economist



Billions of Short Reads

ATATATACGTACTAGTACGT  
 TTTAGTACGTACGT  
 ATACGTACTAGTACGT  
 CGCCCCTACGTA  
 ACGTACTAGTACGT  
 TTAGTACGTACGT  
 TACGTACTAAAGTACGT  
 TACGTACTAGTACGT  
 TTTAAACGTA  
 CGTACTAGTACGT  
 GGGAGTACGTACGT



## 1 Sequencing

# Genome Analysis

## 2 Read Mapping

Data → performance & energy bottleneck

read4: CGCTTCCAT  
 read5: CCATGACGC  
 read6: TTCCATGAC



## 3 Variant Calling

## 4 Scientific Discovery

# Software Acceleration: Eliminate Useless Work

---

- Download the source code and try for yourself
  - [Download link to FastHASH](http://www.biomedcentral.com/1471-2164/14/S1/S13)

Xin *et al.* *BMC Genomics* 2013, **14**(Suppl 1):S13  
<http://www.biomedcentral.com/1471-2164/14/S1/S13>



**PROCEEDINGS**

**Open Access**

## Accelerating read mapping with FastHASH

Hongyi Xin<sup>1</sup>, Donghyuk Lee<sup>1</sup>, Farhad Hormozdiari<sup>2</sup>, Samihan Yedkar<sup>1</sup>, Onur Mutlu<sup>1\*</sup>, Can Alkan<sup>3\*</sup>

*From* The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)  
Vancouver, Canada. 21-24 January 2013

# Shifted Hamming Distance: SIMD Acceleration

---

<https://github.com/CMU-SAFARI/Shifted-Hamming-Distance>

*Bioinformatics*, 31(10), 2015, 1553–1560

doi: 10.1093/bioinformatics/btu856

Advance Access Publication Date: 10 January 2015

Original Paper

OXFORD

---

Sequence analysis

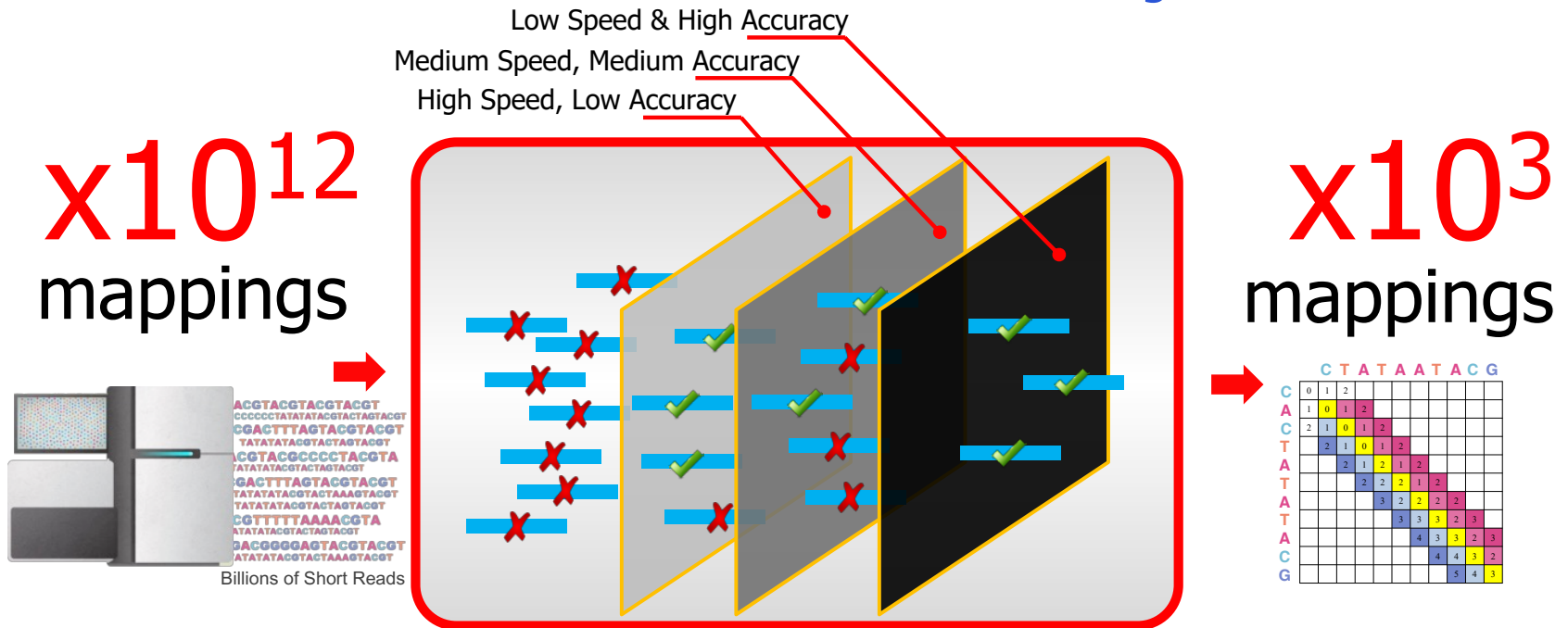
## **Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping**

Hongyi Xin<sup>1,\*</sup>, John Greth<sup>2</sup>, John Emmons<sup>2</sup>, Gennady Pekhimenko<sup>1</sup>,  
Carl Kingsford<sup>3</sup>, Can Alkan<sup>4,\*</sup> and Onur Mutlu<sup>2,\*</sup>

Xin+, "[Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping](#)", **Bioinformatics 2015.**

---

# GateKeeper: FPGA-Based Alignment Filtering



- 1** High throughput DNA sequencing (HTS) technologies
- 2** Read Pre-Alignment Filtering  
Fast & Low False Positive Rate
- 3** Read Alignment  
Slow & Zero False Positives



# 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.  
[[Slides \(pptx\) \(pdf\)](#)]  
[[Source Code](#)]  
[[arxiv.org Version \(pdf\)](#)]  
[[Talk Video at AACBB 2019](#)]

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

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

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

# Shouji (障子) [Alser+, Bioinformatics 2019]

---

Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,  
**"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"**  
*Bioinformatics*, [published online, March 28], 2019.

[\[Source Code\]](#)

[\[Online link at Bioinformatics Journal\]](#)

*Bioinformatics*, 2019, 1–9

doi: 10.1093/bioinformatics/btz234

Advance Access Publication Date: 28 March 2019

Original Paper

OXFORD

---

Sequence alignment

## Shouji: a fast and efficient pre-alignment filter for sequence alignment

Mohammed Alser<sup>1,2,3,\*</sup>, Hasan Hassan<sup>1</sup>, Akash Kumar<sup>2</sup>, Onur Mutlu<sup>1,3,\*</sup>  
and Can Alkan<sup>3,\*</sup>

<sup>1</sup>Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, <sup>2</sup>Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and <sup>3</sup>Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

\*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

# SneakySnake [Alser+, Bioinformatics 2020]

---

Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu,  
**"SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment  
Filter for CPUs, GPUs, and FPGAs"**

**Bioinformatics**, to appear in 2020.

**[Source Code]**

**[Online link at Bioinformatics Journal]**

*Bioinformatics*

doi.10.1093/bioinformatics/xxxxxx

Advance Access Publication Date: Day Month Year

Manuscript Category

OXFORD

---

Subject Section

## **SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs**

**Mohammed Alser<sup>1,2,\*</sup>, Taha Shahroodi<sup>1</sup>, Juan Gómez-Luna<sup>1,2</sup>,  
Can Alkan<sup>4,\*</sup>, and Onur Mutlu<sup>1,2,3,4,\*</sup>**

<sup>1</sup>Department of Computer Science, ETH Zurich, Zurich 8006, Switzerland

<sup>2</sup>Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland

<sup>3</sup>Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh 15213, PA, USA

<sup>4</sup>Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

# GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, 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, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"**  
*Proceedings of the 53rd International Symposium on Microarchitecture (MICRO)*, Virtual, October 2020.  
[[Lighting Talk Video](#) (1.5 minutes)]  
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (18 minutes)]  
[[Slides \(pptx\)](#) ([pdf](#))]

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

Damla Senol Cali<sup>†⌘</sup> Gurpreet S. Kalsi<sup>⌘</sup> Zülal Bingöl<sup>▽</sup> Can Firtina<sup>◇</sup> Lavanya Subramanian<sup>‡</sup> Jeremie S. Kim<sup>◇†</sup>  
Rachata Ausavarungnirun<sup>○</sup> Mohammed Alser<sup>◇</sup> Juan Gomez-Luna<sup>◇</sup> Amirali Boroumand<sup>†</sup> Anant Nori<sup>⌘</sup>  
Allison Scibisz<sup>†</sup> Sreenivas Subramoney<sup>⌘</sup> Can Alkan<sup>▽</sup> Saugata Ghose<sup>\*†</sup> Onur Mutlu<sup>◇†▽</sup>  
<sup>†</sup>Carnegie Mellon University   <sup>⌘</sup>Processor Architecture Research Lab, Intel Labs   <sup>▽</sup>Bilkent University   <sup>◇</sup>ETH Zürich  
<sup>‡</sup>Facebook   <sup>○</sup>King Mongkut's University of Technology North Bangkok   <sup>\*</sup>University of Illinois at Urbana-Champaign



# Future of Genome Sequencing & Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu  
["Accelerating Genome Analysis: A Primer on an Ongoing Journey"](#) IEEE Micro, August 2020.



MinION from ONT

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

Sept.-Oct. 2020, pp. 65-75, vol. 40

DOI Bookmark: [10.1109/MM.2020.3013728](https://doi.org/10.1109/MM.2020.3013728)

## FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications

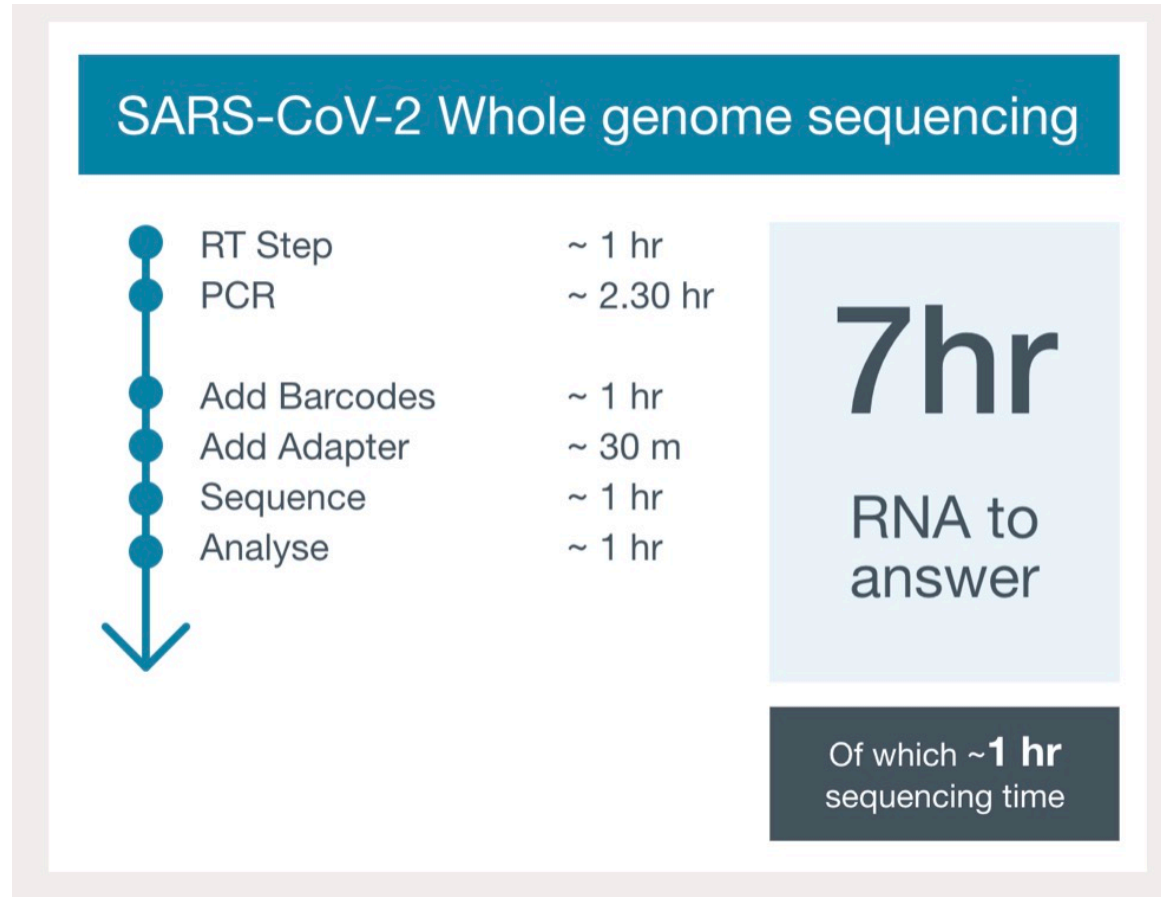
July-Aug. 2021, pp. 39-48, vol. 41

DOI Bookmark: [10.1109/MM.2021.3088396](https://doi.org/10.1109/MM.2021.3088396)



SmidgION from ONT

# COVID-19 Nanopore Sequencing (I)



• From ONT (<https://nanoporetech.com/covid-19/overview>)

# COVID-19 Nanopore Sequencing (II)

## How are scientists using nanopore sequencing to research COVID-19?



Samples  
are collected

**Validated SARS-CoV-2  
RT-PCR test performed**



SARS-CoV-2 positive samples



SARS-CoV-2 negative samples:  
used as negative controls

**How can this be used?**  
Genomic epidemiology: analyse variants  
& mutation rate, track spread of virus,  
identify clusters of transmission

**What are the results?**  
From RNA to full  
SARS-CoV-2 consensus  
sequence in ~7 hours

**How?**  
Targeted amplification of  
SARS-CoV-2 genome + multiplexed,  
rapid nanopore sequencing

**Targeted SARS-CoV-2  
nanopore sequencing**



**Metagenomic  
nanopore sequencing**

**How?**  
1 x RNA metagenomic  
sequencing run  
1 x DNA metagenomic  
sequencing run

**What are the results?**  
RNA: data for RNA viruses (including  
SARS-CoV-2) + microbial transcripts  
DNA: data for bacteria + DNA viruses

**How can this be used?**  
Characterise co-infecting bacteria  
& viruses, identify any correlation  
of risk factors, research potential  
future treatment implications

**SARS-CoV-2 Direct RNA whole  
genome sequencing:** assess  
viral genome in its native RNA  
form and the effect of base  
modifications

**Immune repertoire:** assess  
response of the immune system to  
SARS-CoV-2 infection by  
sequencing of full-length immune  
cell receptor genes and transcripts

**Whole human genome  
sequencing:** investigate what  
might cause different responses  
to the virus in different people  
based on their genome

**What's next?**



Find out more at [nanoporetech.com/covid19](https://nanoporetech.com/covid19)

MinION™

GridION™

PromethION™

Oxford Nanopore Technologies, the Wheel icon, GridION, PromethION and MinION are registered trademarks of Oxford Nanopore Technologies in various countries. © 2020 Oxford Nanopore Technologies. All rights reserved. Oxford Nanopore Technologies' products are currently for research use only. IG\_1061[EN]\_V1\_03Apr2020

• From ONT (<https://nanoporetech.com/covid-19/overview>)

# Accelerating Genome Analysis: Overview

---

- Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,  
[\*\*"Accelerating Genome Analysis: A Primer on an Ongoing Journey"\*\*](#)  
[\*IEEE Micro\* \(\*\*IEEE MICRO\*\*\)](#), Vol. 40, No. 5, pages 65-75, September/October 2020.  
[[Slides \(pptx\)\(pdf\)](#)]  
[[Talk Video \(1 hour 2 minutes\)](#)]

## Accelerating Genome Analysis: A Primer on an Ongoing Journey

**Mohammed Alser**

ETH Zürich

**Zülal Bingöl**

Bilkent University

**Damla Senol Cali**

Carnegie Mellon University

**Jeremie Kim**

ETH Zurich and Carnegie Mellon University

**Saugata Ghose**

University of Illinois at Urbana–Champaign and  
Carnegie Mellon University

**Can Alkan**

Bilkent University

**Onur Mutlu**

ETH Zurich, Carnegie Mellon University, and  
Bilkent University

# More on Fast Genome Analysis ...

- Onur Mutlu,  
**"Accelerating Genome Analysis: A Primer on an Ongoing Journey"**  
*Invited Lecture at [Technion](#), Virtual, 26 January 2021.*  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (1 hour 37 minutes, including Q&A)]  
[[Related Invited Paper \(at IEEE Micro, 2020\)](#)]

The video player displays a slide titled "Insight: Shifting a String Helps Similarity Search". The slide content includes the text "7 matches 1 mismatch" and a diagram showing the alignment of the strings "I STANBUL" and "I STNBUL". The diagram uses green arrows to indicate matches (I, S, T, N, B, U, L) and a red arrow to indicate a mismatch (A vs empty space). The video player interface shows a progress bar at 46:08 / 1:37:37, a video thumbnail of Onur Mutlu, and a list of video recommendations.

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

566 views · Premiered Feb 6, 2021

Onur Mutlu Lectures  
13.9K subscribers

ANALYTICS EDIT VIDEO



# Detailed Lectures on Genome Analysis

---

- **Computer Architecture, Fall 2020, Lecture 3a**
  - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5>
- **Computer Architecture, Fall 2020, Lecture 8**
  - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14>
- **Computer Architecture, Fall 2020, Lecture 9a**
  - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15>
- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
  - **Accelerating Genomics** (ETH Zürich, Fall 2020)
  - <https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId>

Many Interesting Things  
Are Happening Today  
in Computer Architecture

**More Demanding Workloads**

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

[Mao+, EuroSys'12;  
Clapp+ (Intel), IISWC'15]



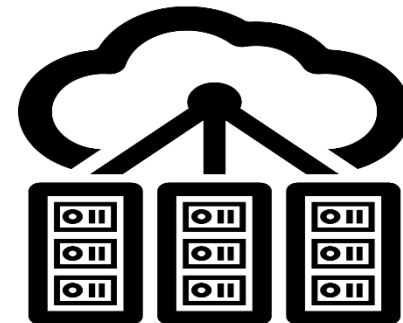
## In-Memory Data Analytics

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



## Graph/Tree Processing

[Xu+, IISWC'12; Umuroglu+, FPL'15]



## Datacenter Workloads

[Kanev+ (Google), ISCA'15]



# Data Overwhelms Modern Machines

---



**In-memory Databases**



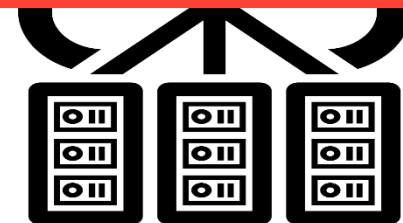
**Graph/Tree Processing**

**Data → performance & energy bottleneck**



**In-Memory Data Analytics**

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



**Datacenter Workloads**

[Kanev+ (Google), ISCA'15]

# Data is Key for Future Workloads



**Chrome**

Google's web browser



**TensorFlow Mobile**

Google's machine learning  
framework



**Video Playback**

Google's **video codec**



**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>

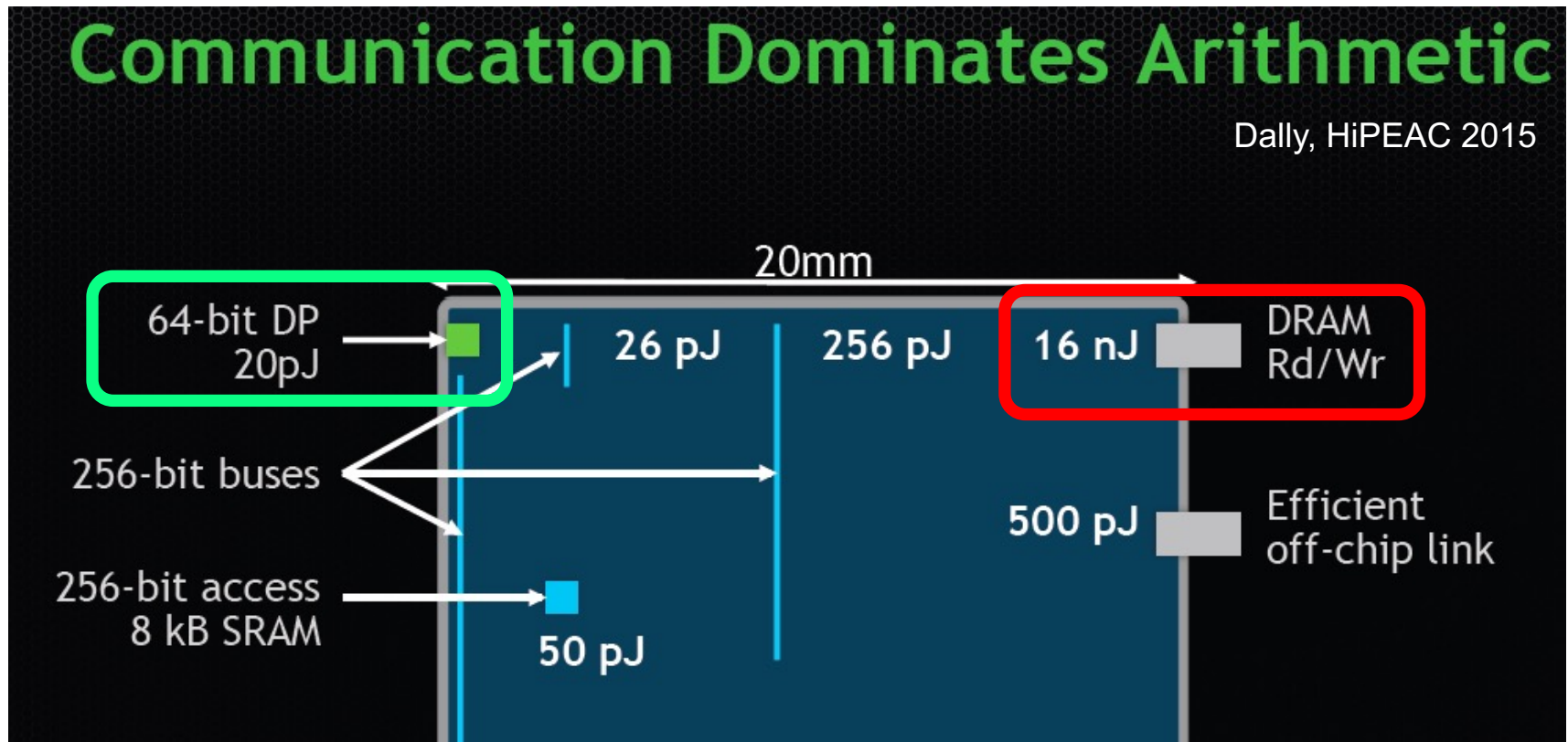
Parthasarathy Ranganathan<sup>3</sup>

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

# Data Movement vs. Computation Energy

## Communication Dominates Arithmetic

Dally, HiPEAC 2015



A memory access consumes  $\sim 100-1000\times$  the energy of a complex 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
  - ❑ 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

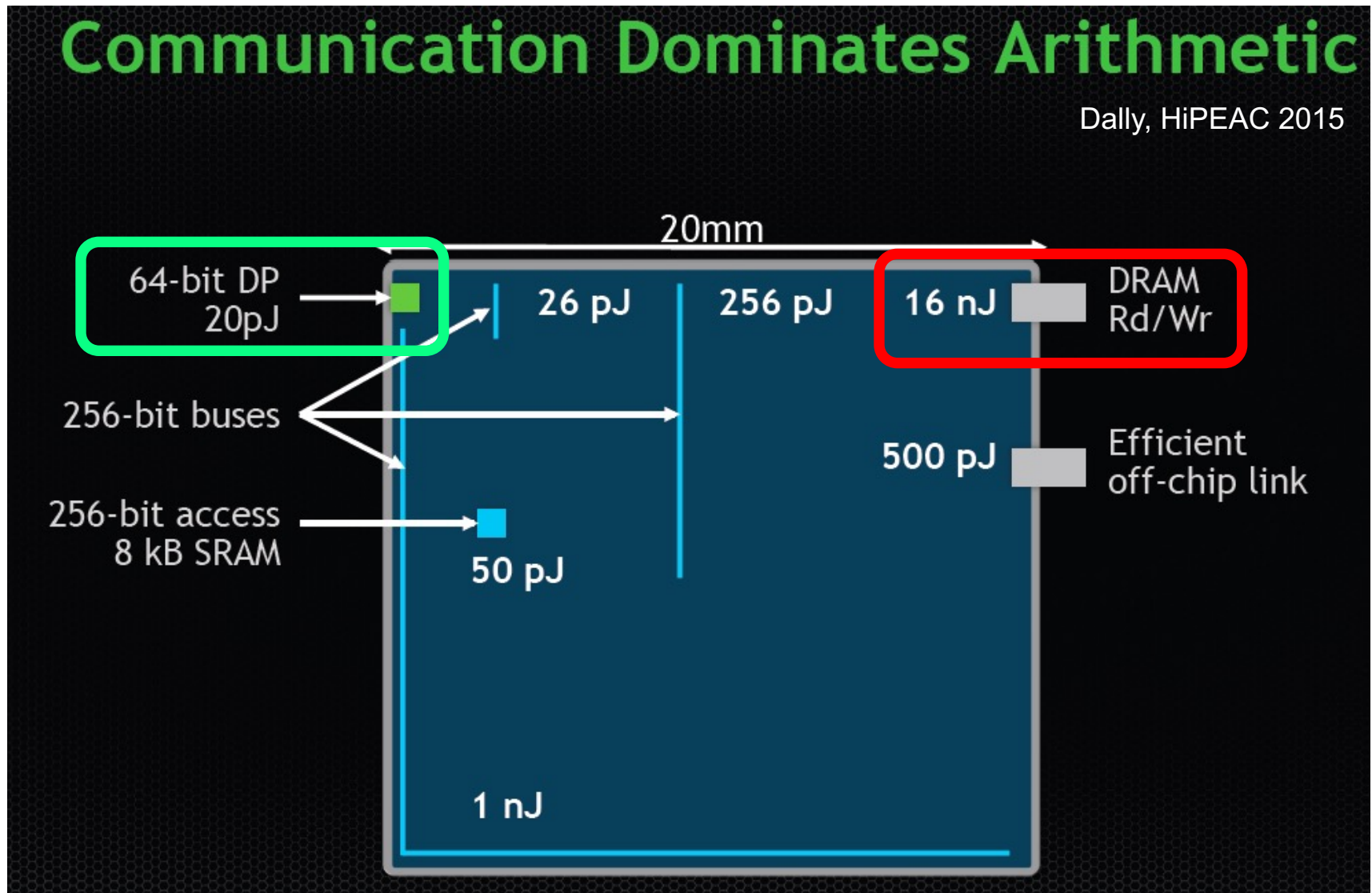
and, they will come

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

# Increasingly Diverging/Complex Tradeoffs

## Communication Dominates Arithmetic

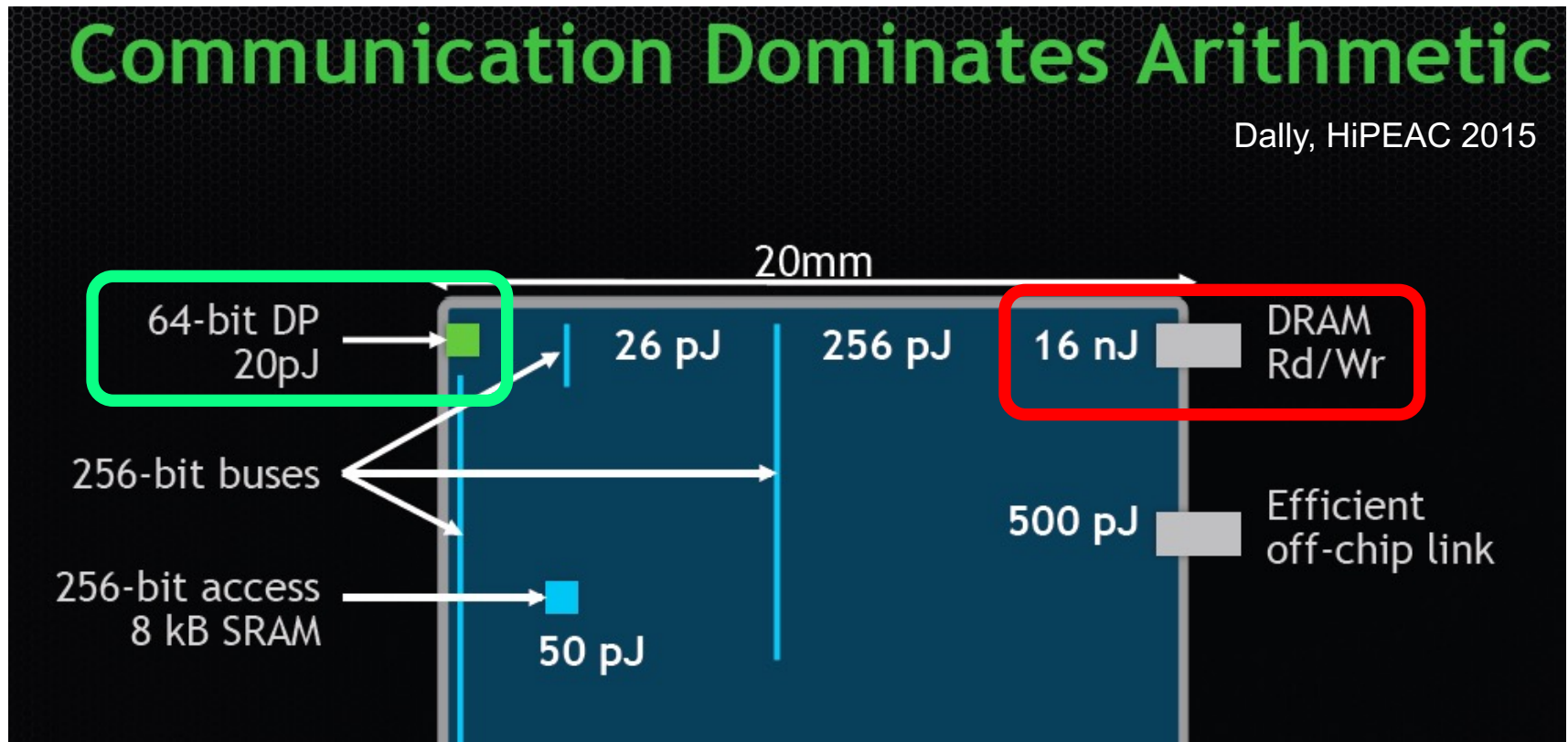
Dally, HiPEAC 2015



# Increasingly Diverging/Complex Tradeoffs

## Communication Dominates Arithmetic

Dally, HiPEAC 2015



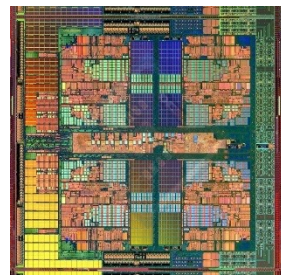
A memory access consumes  $\sim 1000\times$  the energy of a complex addition



# Increasingly Complex Systems

---

## Past systems



Microprocessor



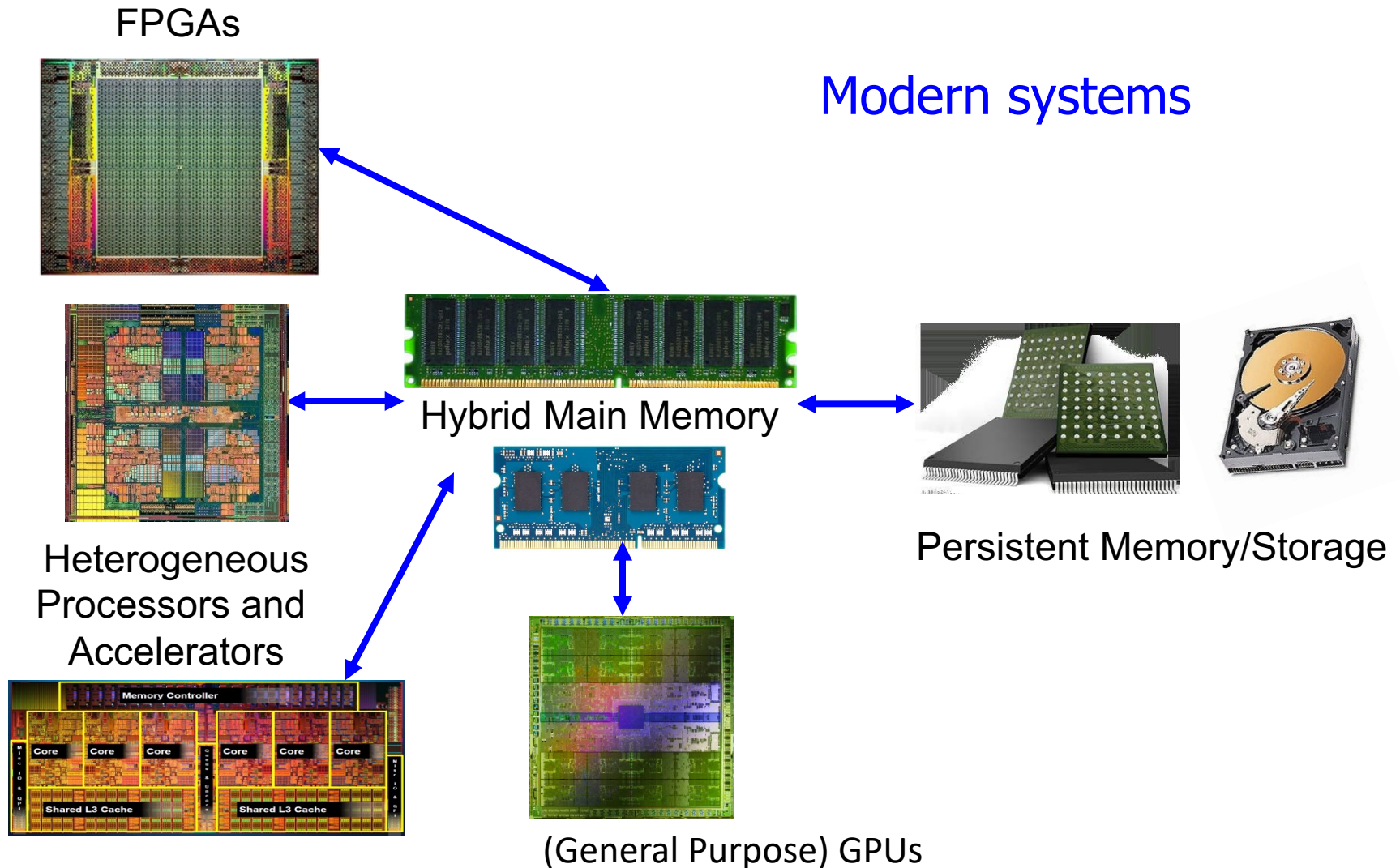
Main Memory



Storage (SSD/HDD)

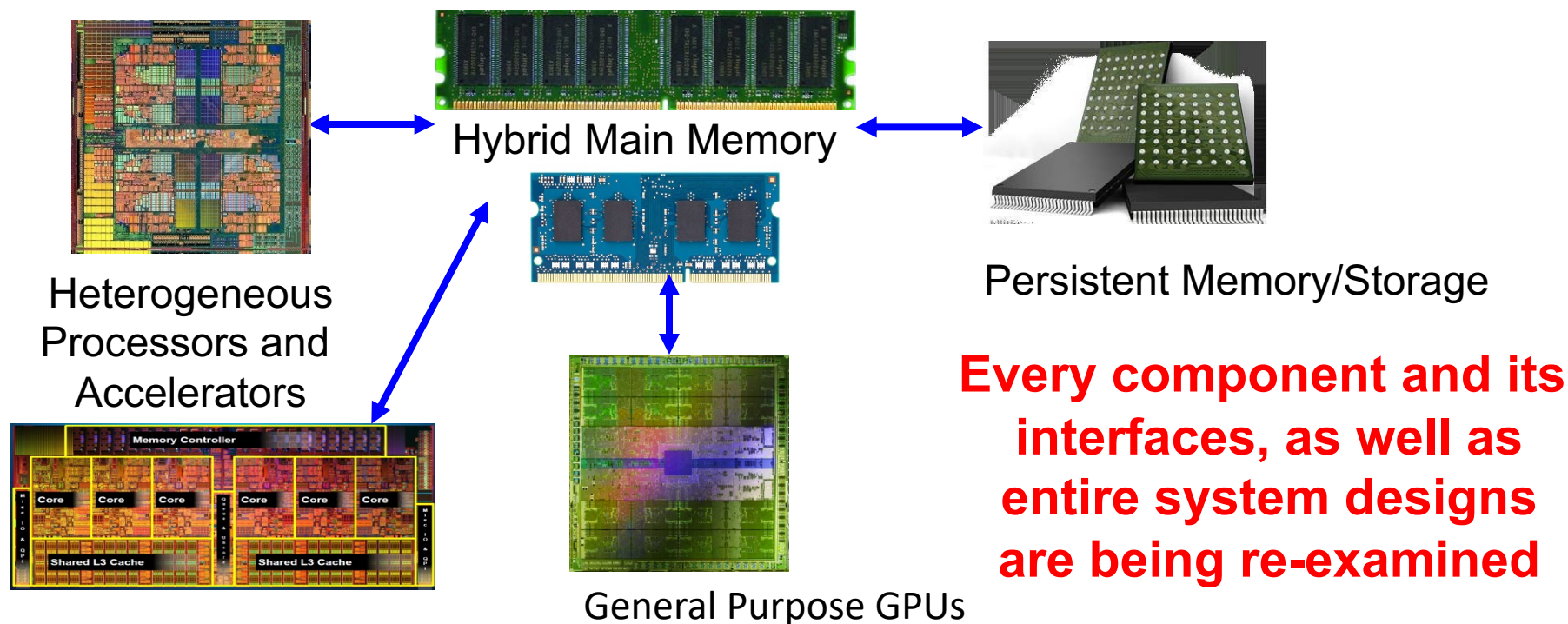


# Increasingly Complex Systems



# Computer Architecture Today

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



# Computer Architecture Today (II)

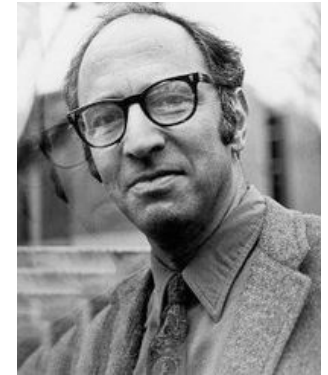
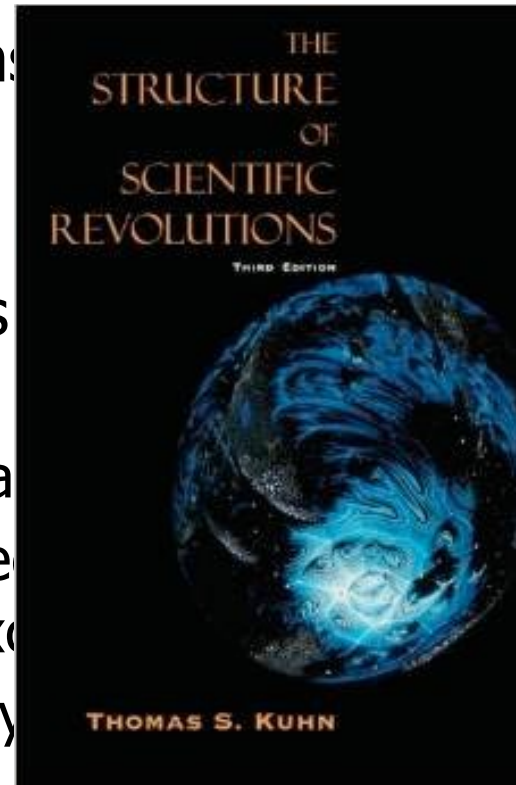
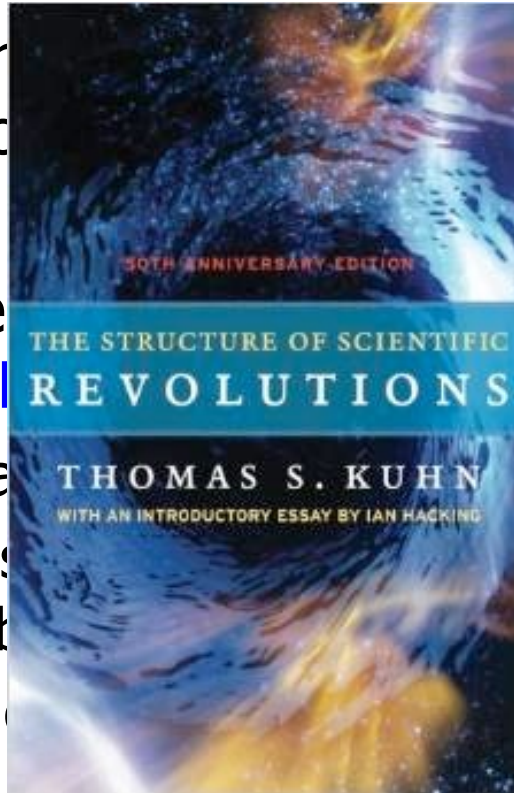
---

- 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** by Thomas S. Kuhn
  - Pre-para
  - Normal s
  - things (b
  - Revolution



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 Fundamentals

# Question: What Is This?





# 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.<sup>[1]</sup> 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](#),



# Compare To This

---





# Question 2: What Is This?



# 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:



# 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:

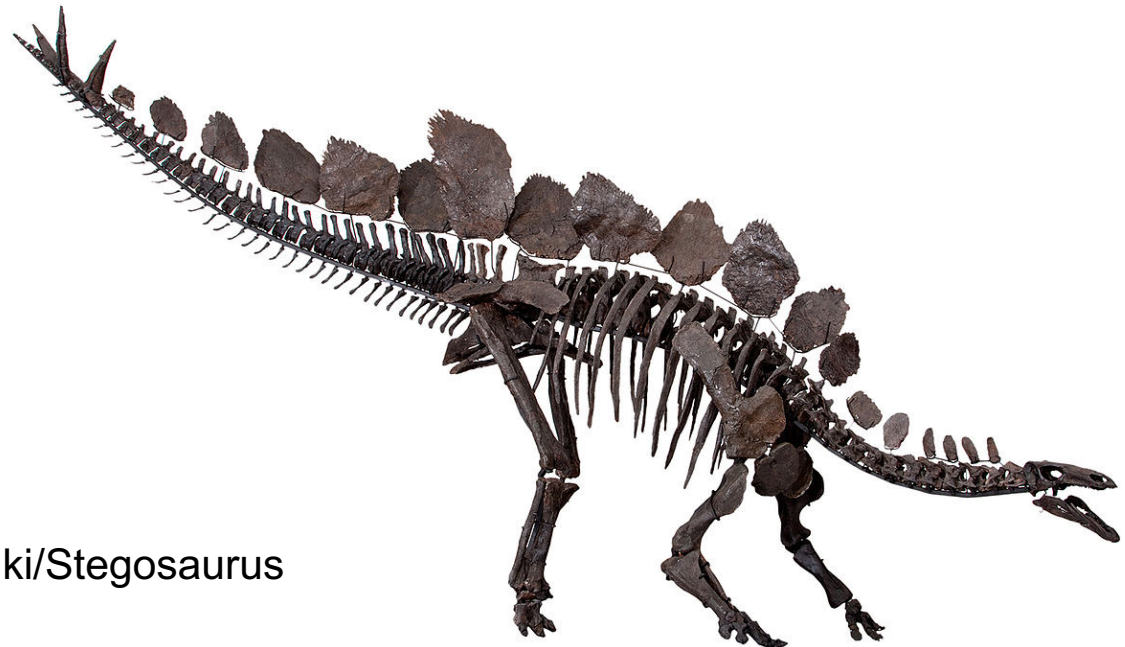
“ In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass.... [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural 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

# Question: What Is This?

---







# Answer: Masterpiece of Another Famous Architect

---

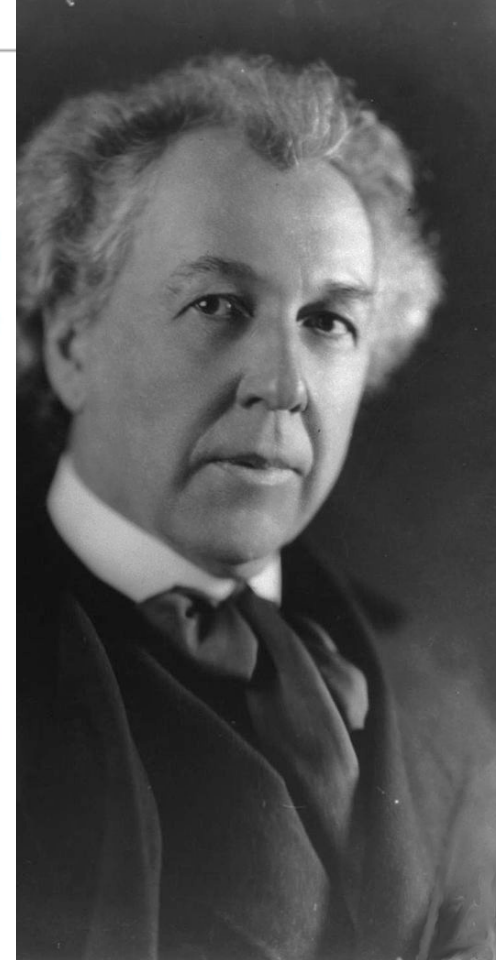
## Fallingwater

---

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](#).<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.





# 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





# That

---



# 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.”

# 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?

---





# What About This?

---





# Milwaukee Art Museum

---





# Athens Olympic Stadium

---





# City of Arts and Sciences, Valencia

---





# Florida Polytechnic University (I)

---





# Oculus, New York City

---





# A Quote from The Other Famous Architect

---

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





# 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





# Yet Another View







# Major High-Level Goals of This Course

---

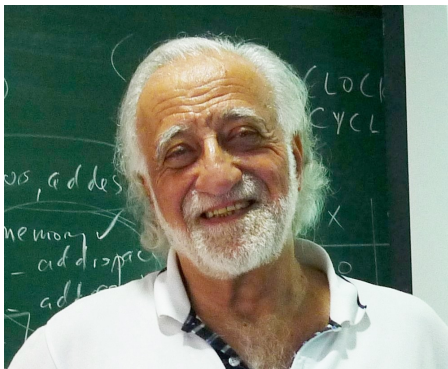
- Understand the principles
- Understand the precedents
- Based on such understanding:
  - Enable you to evaluate tradeoffs of different designs and ideas
  - Enable you to develop principled designs
  - Enable you to develop novel, out-of-the-box designs
- The focus is on:
  - Principles, precedents, 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

- 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?



## Digital Design

“C” as a model of computation

Programmer’s view of how a computer system works

*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

# 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

## Microarchitecture

An implementation of the ISA

Problem

Algorithm

Program/Language

Runtime System  
(VM, OS, MM)

ISA (Architecture)

Microarchitecture

Logic

Devices

Electrons

## ISA

(Instruction Set Architecture)

Interface/contract between  
SW and HW.

What the programmer  
assumes hardware will  
satisfy.



## Digital logic circuits

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

# Aside: An Important Work By Hamming

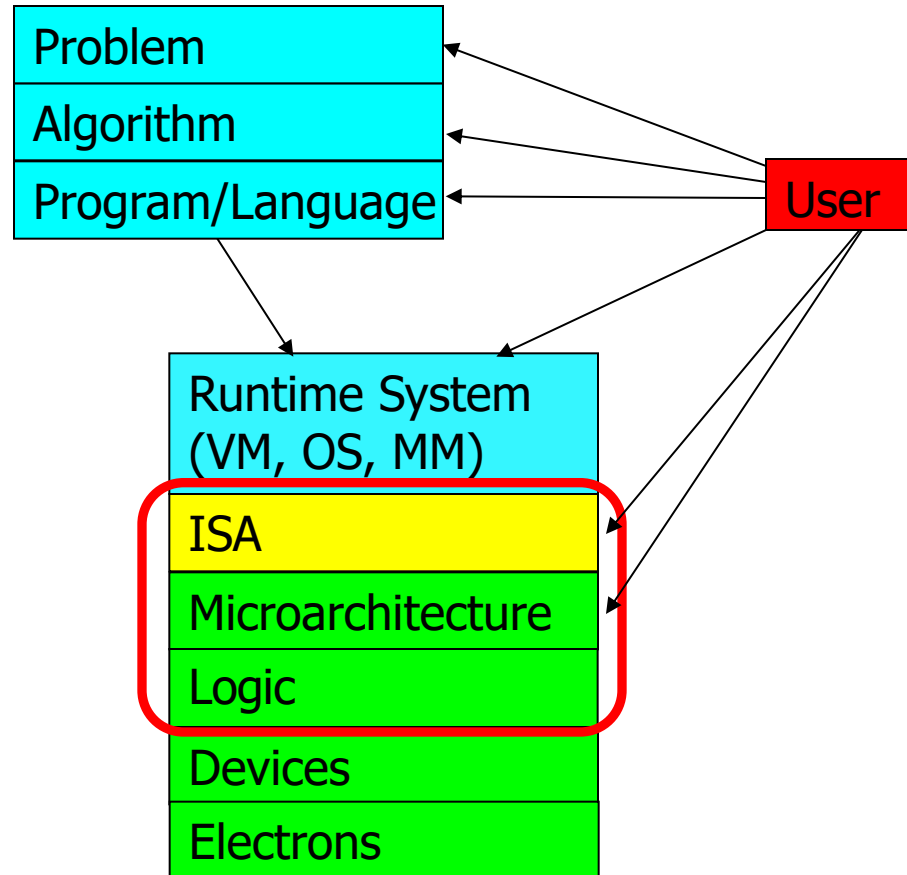
---

- 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?

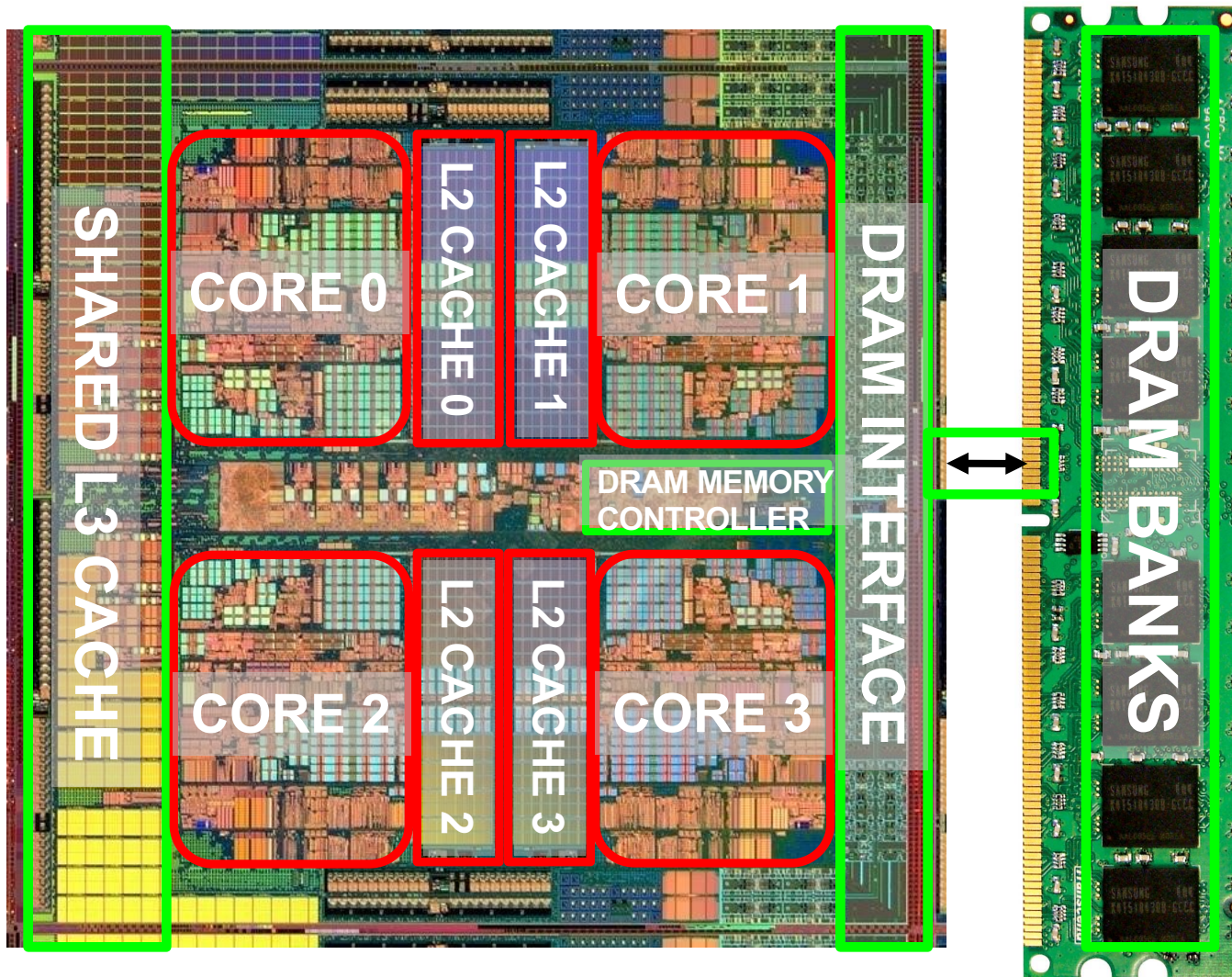
# Crossing the Abstraction Layers

---

- 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

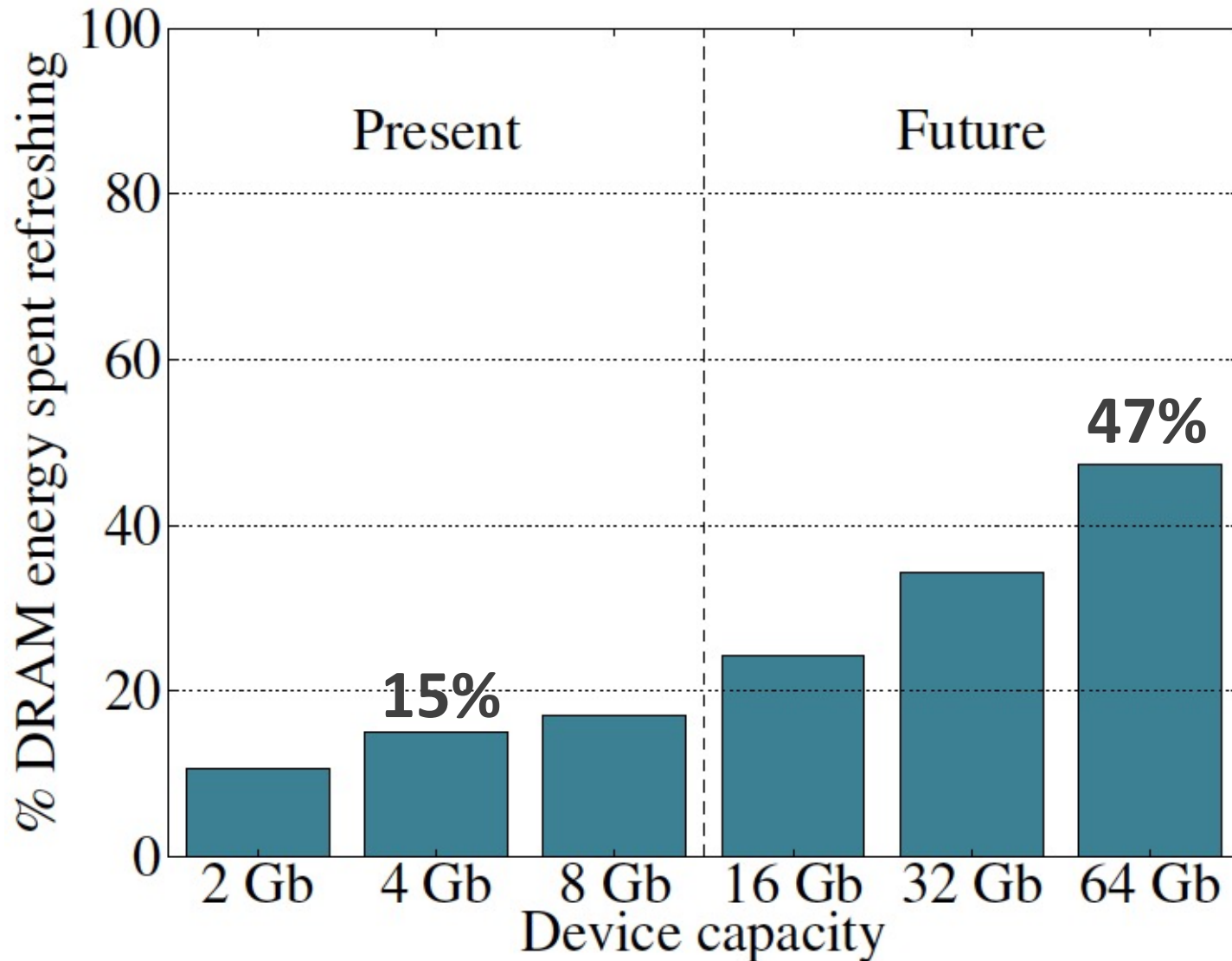
# An Example: Multi-Core Systems

Multi-Core  
Chip



\*Die photo credit: AMD Barcelona

# Another Example: Memory Refresh





# Computer Architecture

## Lecture 1: Introduction and Basics

Prof. Onur Mutlu

ETH Zürich

Fall 2021

30 September 2021