Memory-Centric Computing

Solving Computing's Memory Problem

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

9 February 2026

EFCL Winter School Keynote Talk

SAFARI

ETH zürich

Computing's Memory Problem: Summary

- **Computing has a huge memory problem**
- Memory is responsible for most of the waste:
  - Performance bottlenecks
  - Energy consumption
  - Robustness problems
  - Monetary cost
  - Hardware real estate
  - …
- Problem becoming worse with more data-intensive workloads
- We can solve it by designing **memory-centric systems**
  - **Memory autonomously manages itself** → technology scaling
  - **Memory performs computation** → app & system scaling

Problem

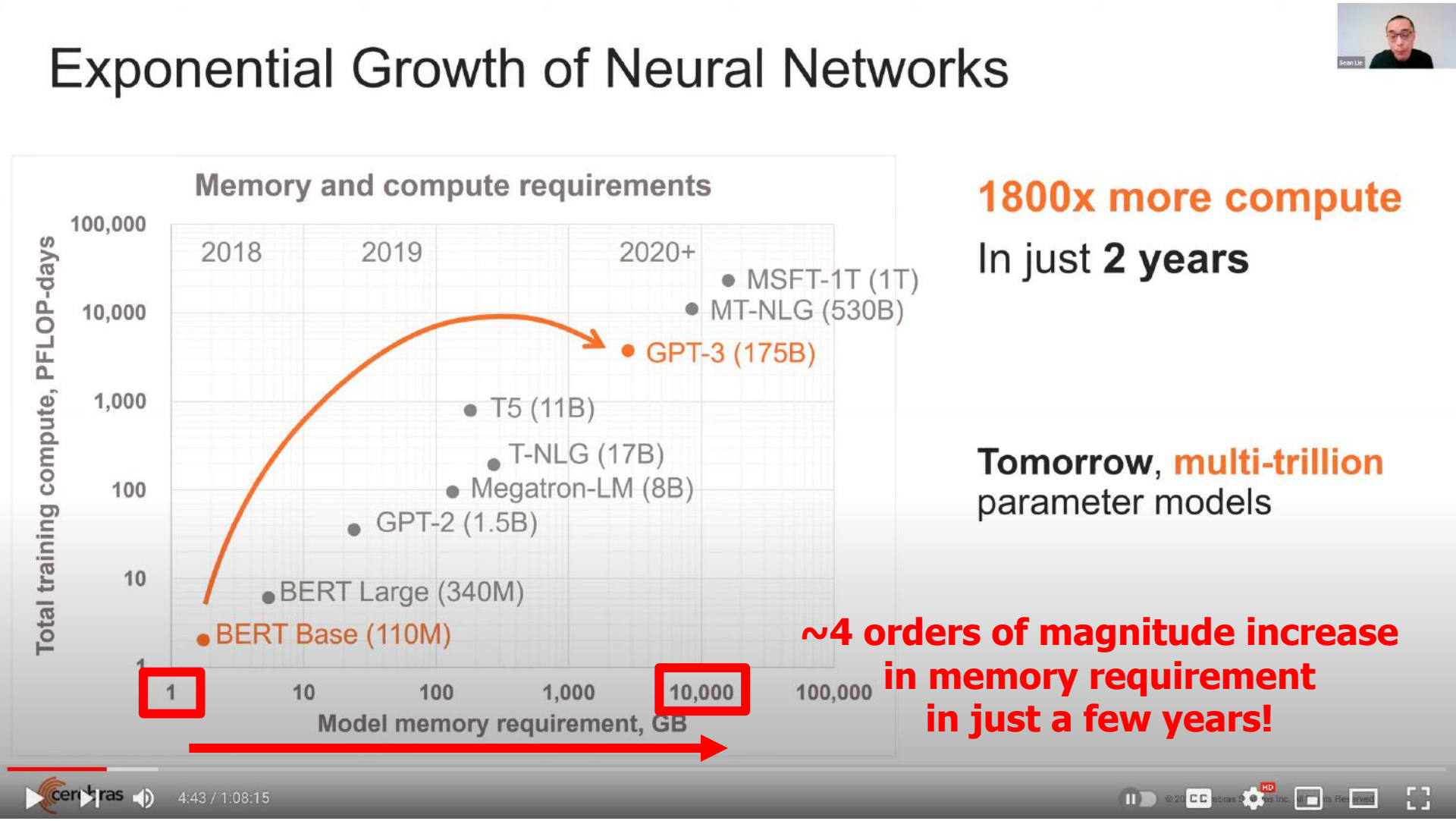
Computing

is Bottlenecked by Data

Data is Key for AI, ML, Genomics, …

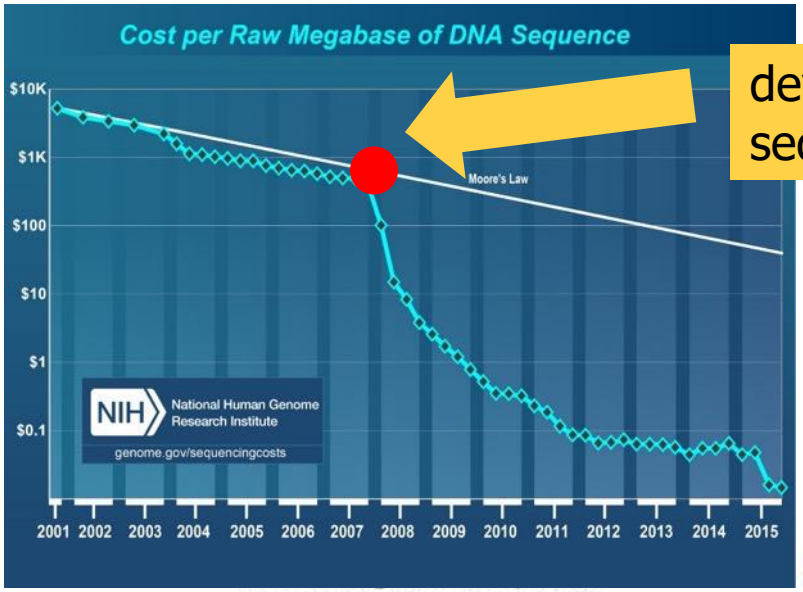
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
  - We can generate more than we can process
  - We need to perform more sophisticated analyses on more data

Huge Demand for Performance & Efficiency



Source: https://youtu.be/Bh13Idwcb0Q?t=283

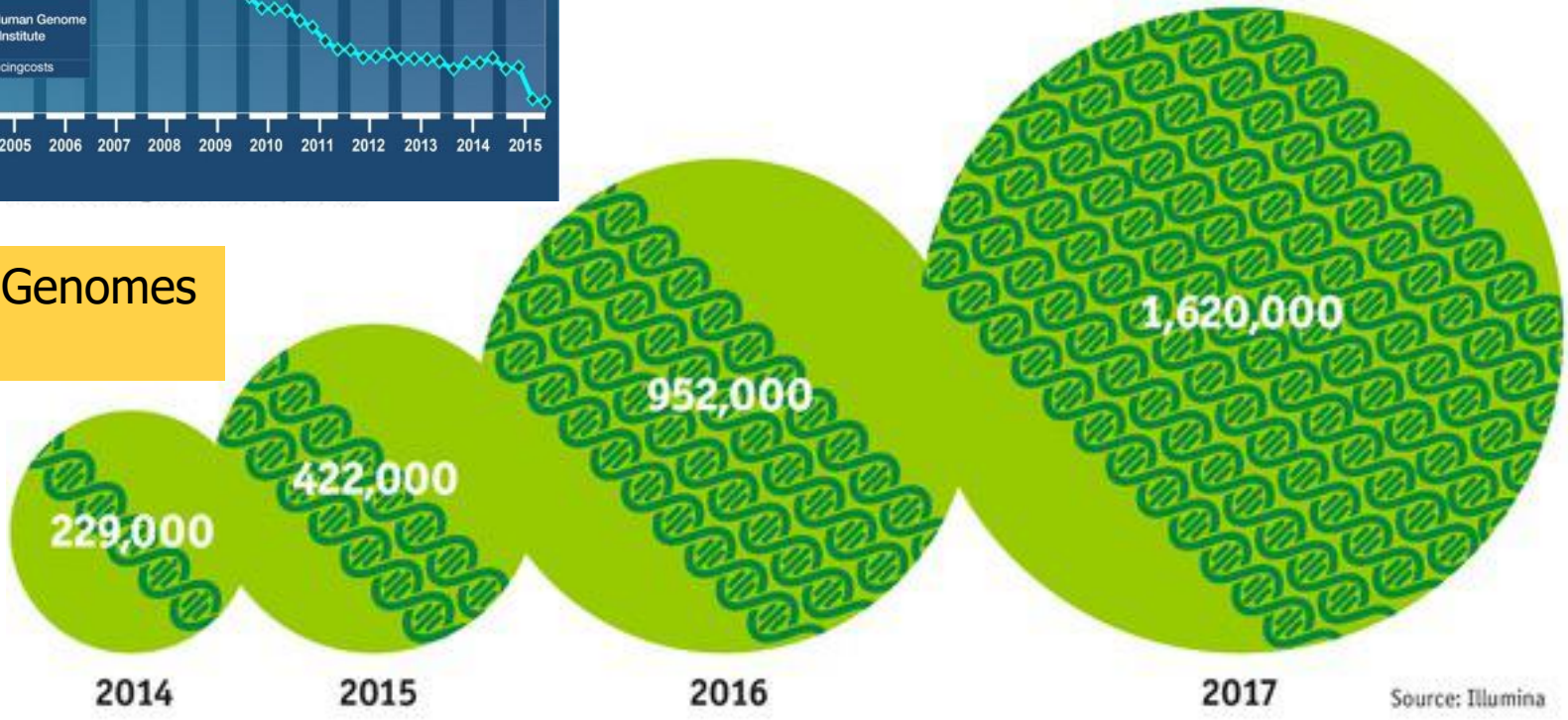
Huge Demand for Performance & Efficiency



development of new sequencing technologies

Oxford Nanopore MinION

Number of Genomes Sequenced



The Economist

http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped

The Problem

Data access is the major performance and energy bottleneck

Our current

design principles

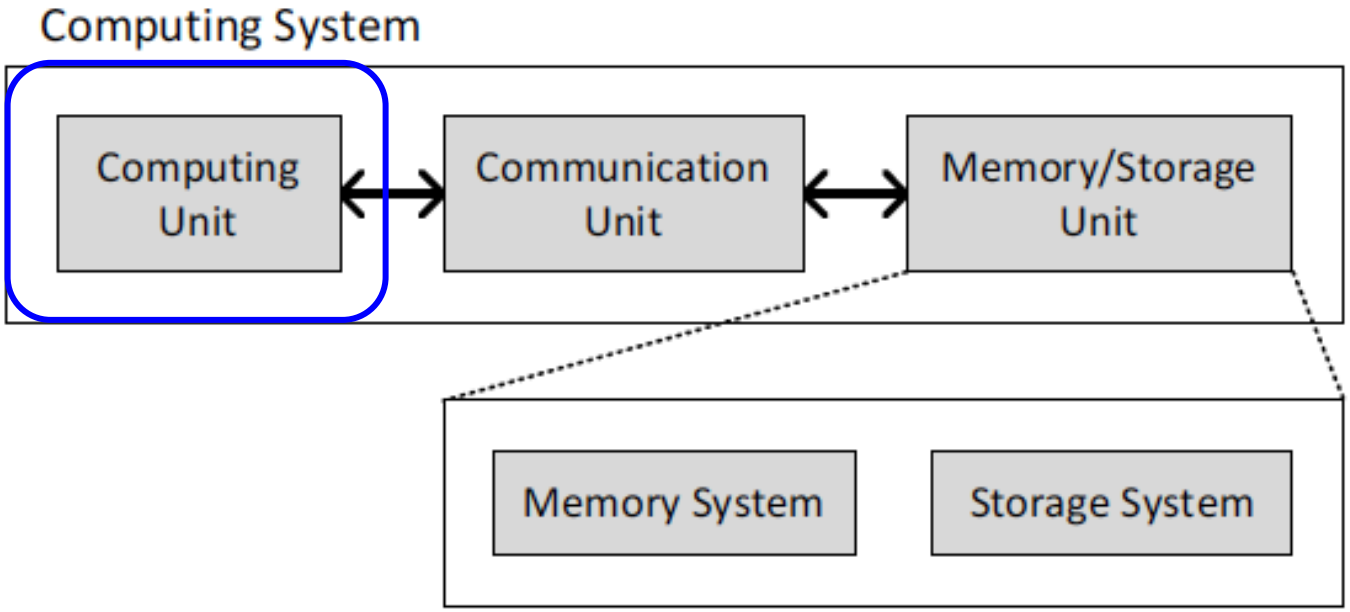
cause great energy waste

(and great performance loss)

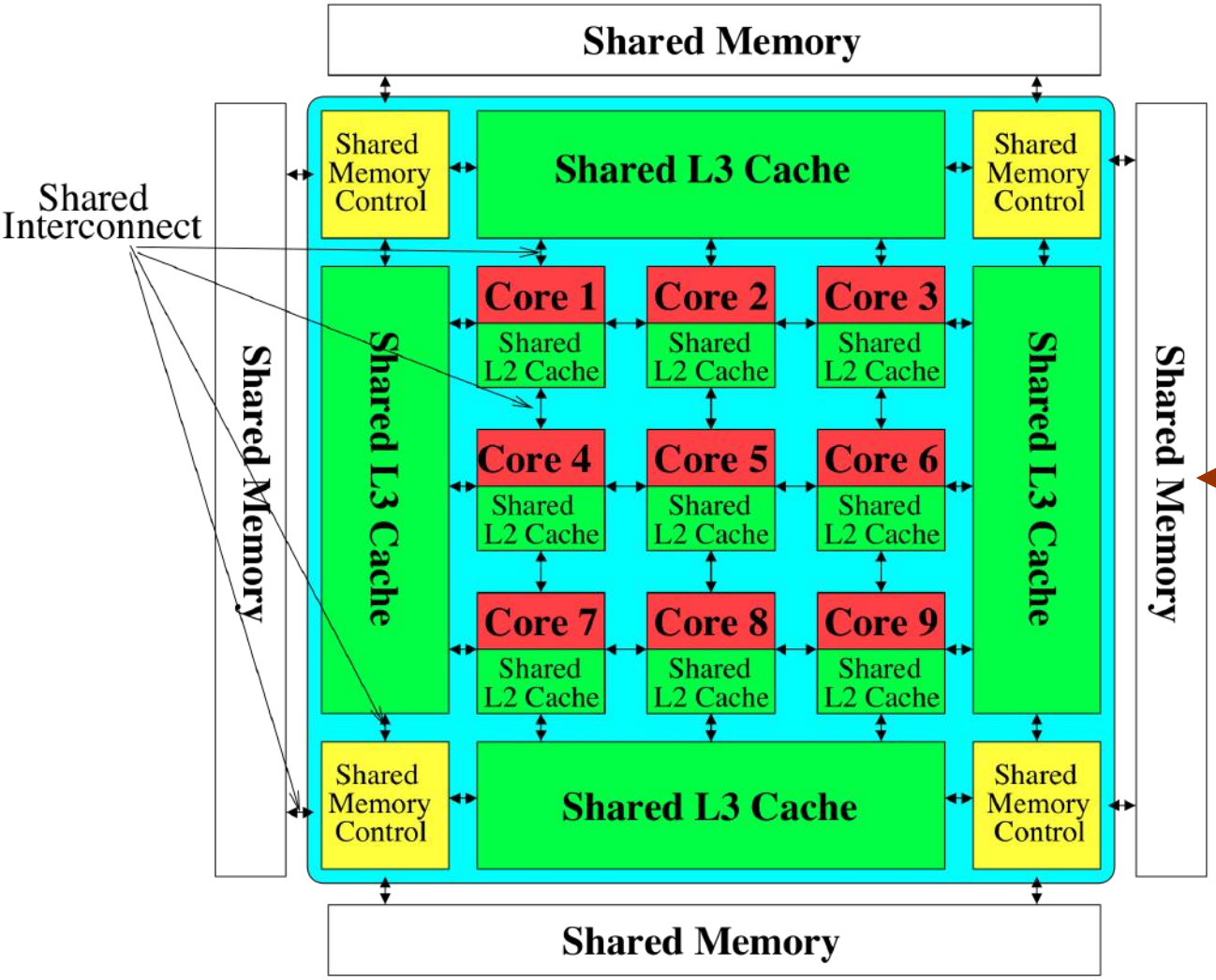
Today's Computing Systems

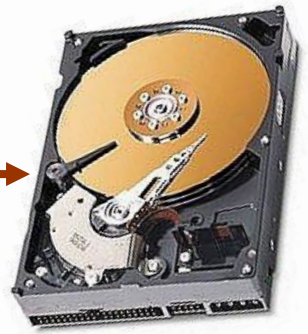
- Processor centric

- All data processed in the processor → at great system cost



Perils of Processor-Centric Design



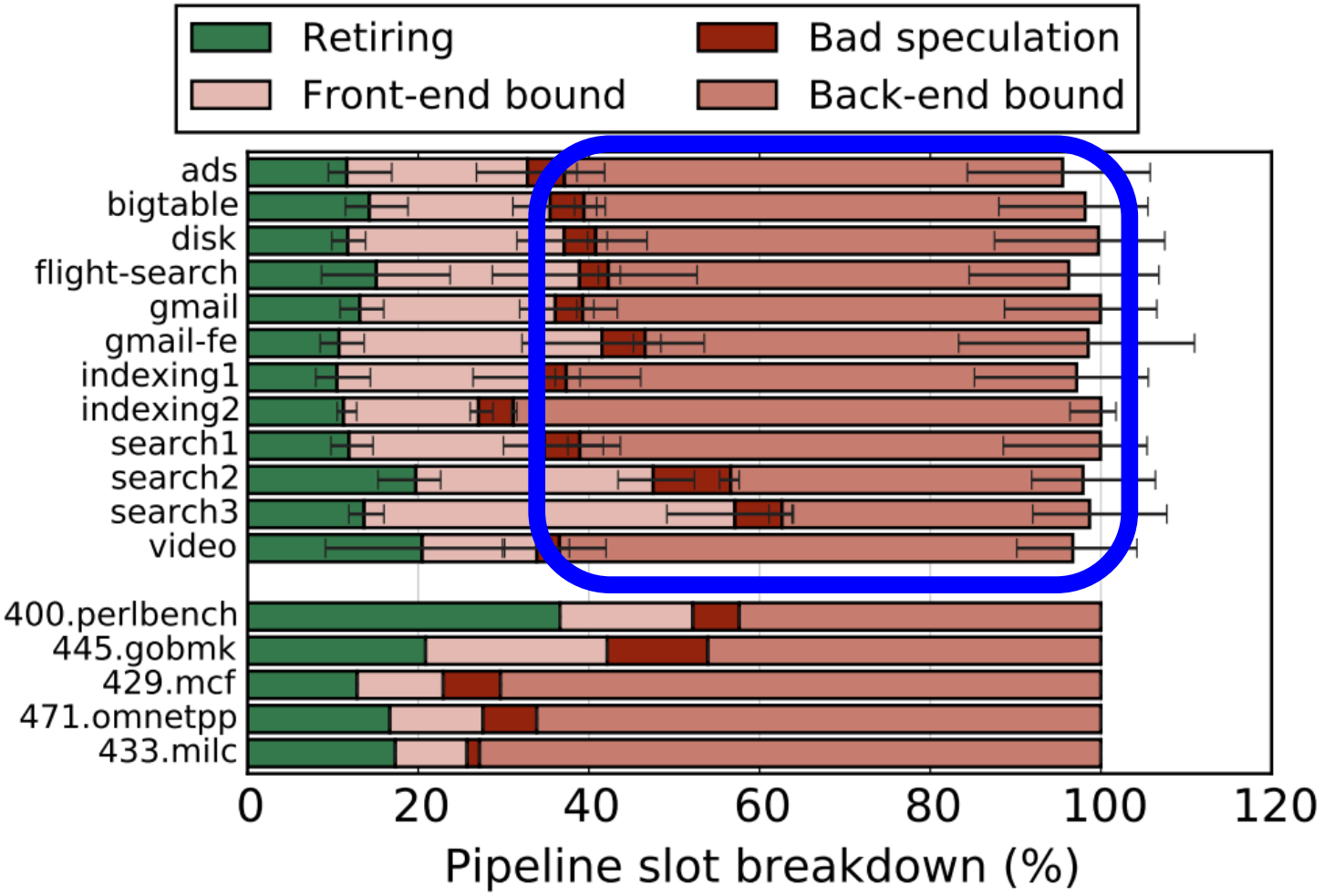


Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory

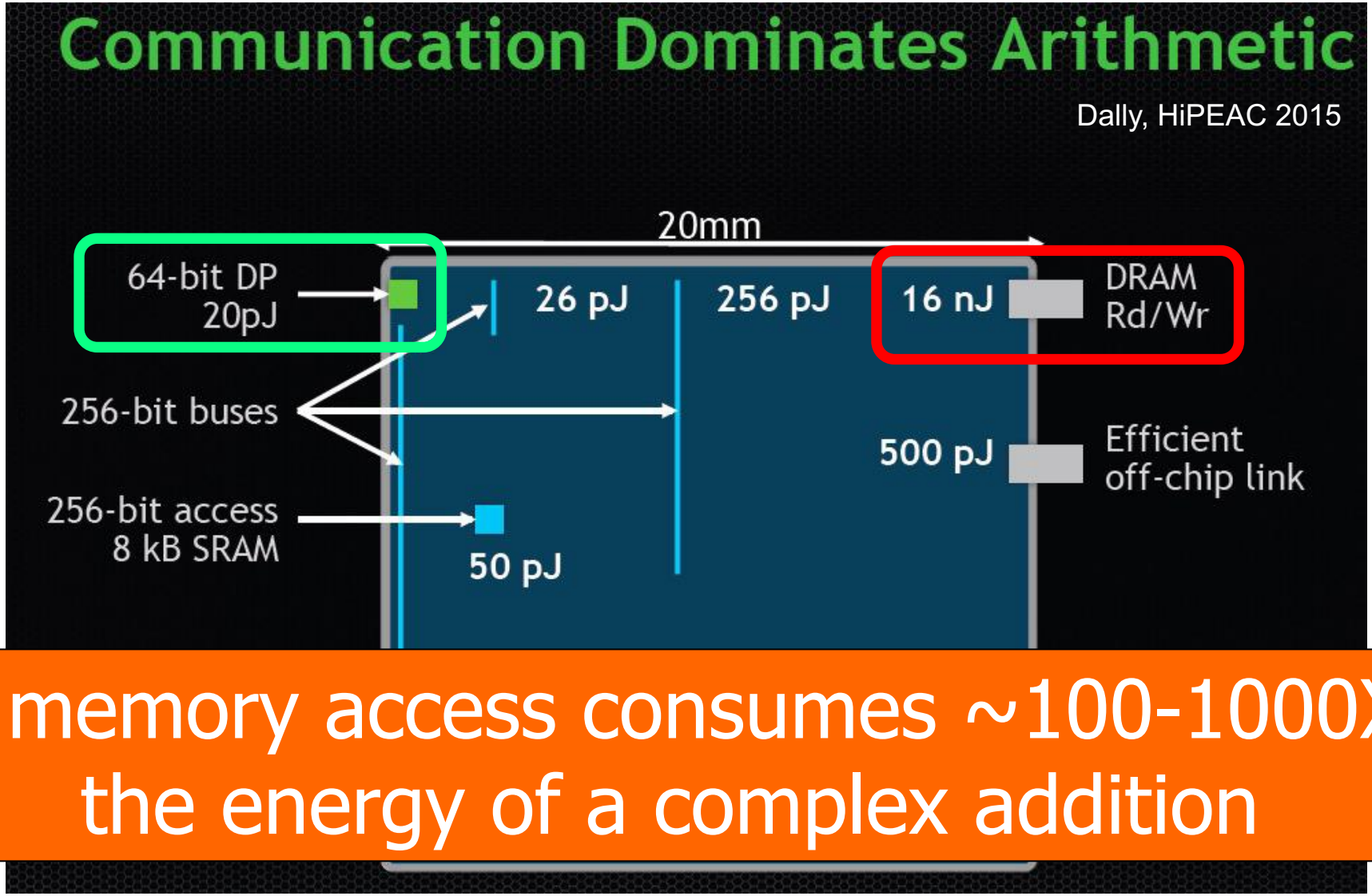
Processor-Centric System Performance

- All of Google's Data Center Workloads (2015):



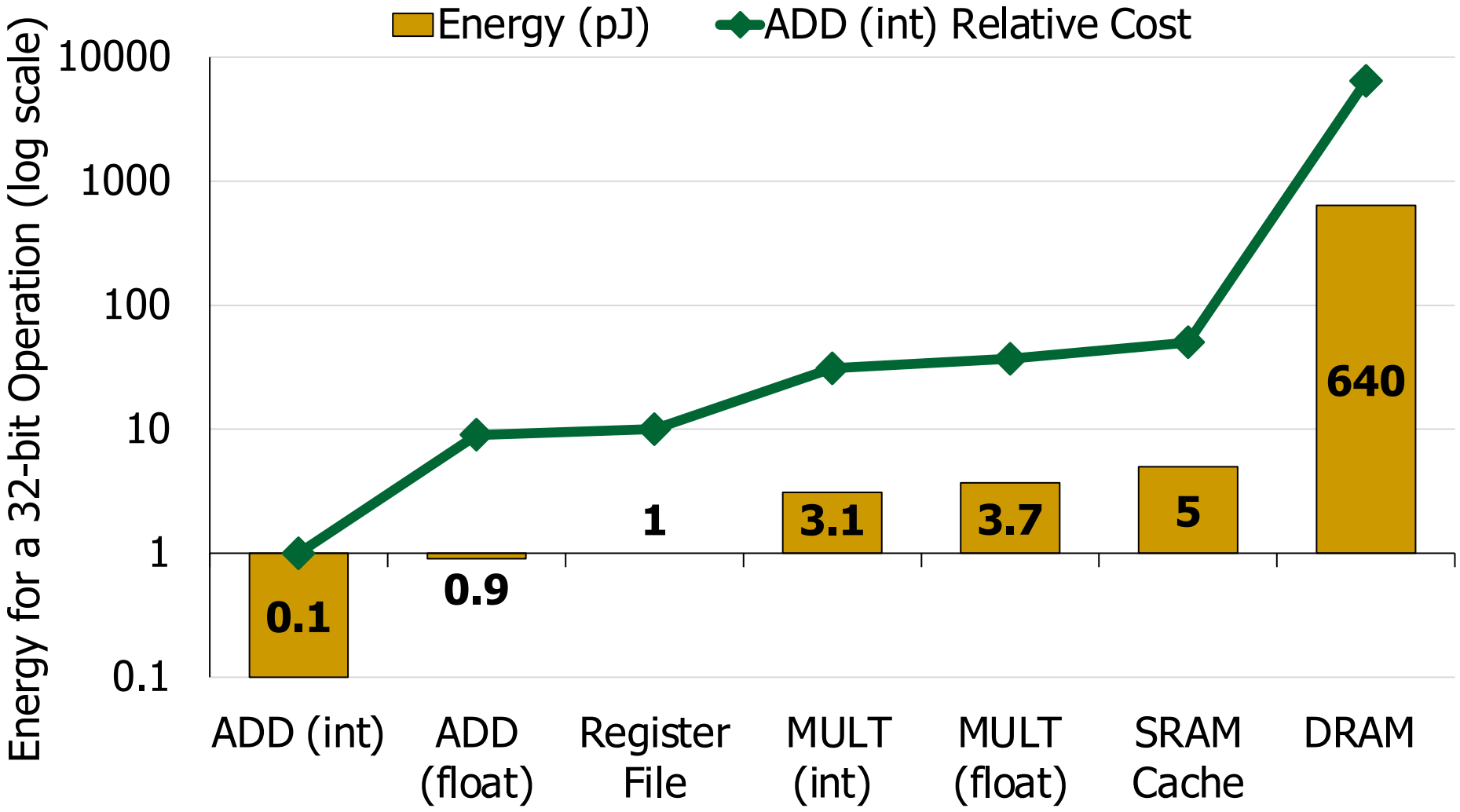
Kanev+, "Profiling a Warehouse-Scale Computer," ISCA 2015.

Data Movement vs. Computation Energy



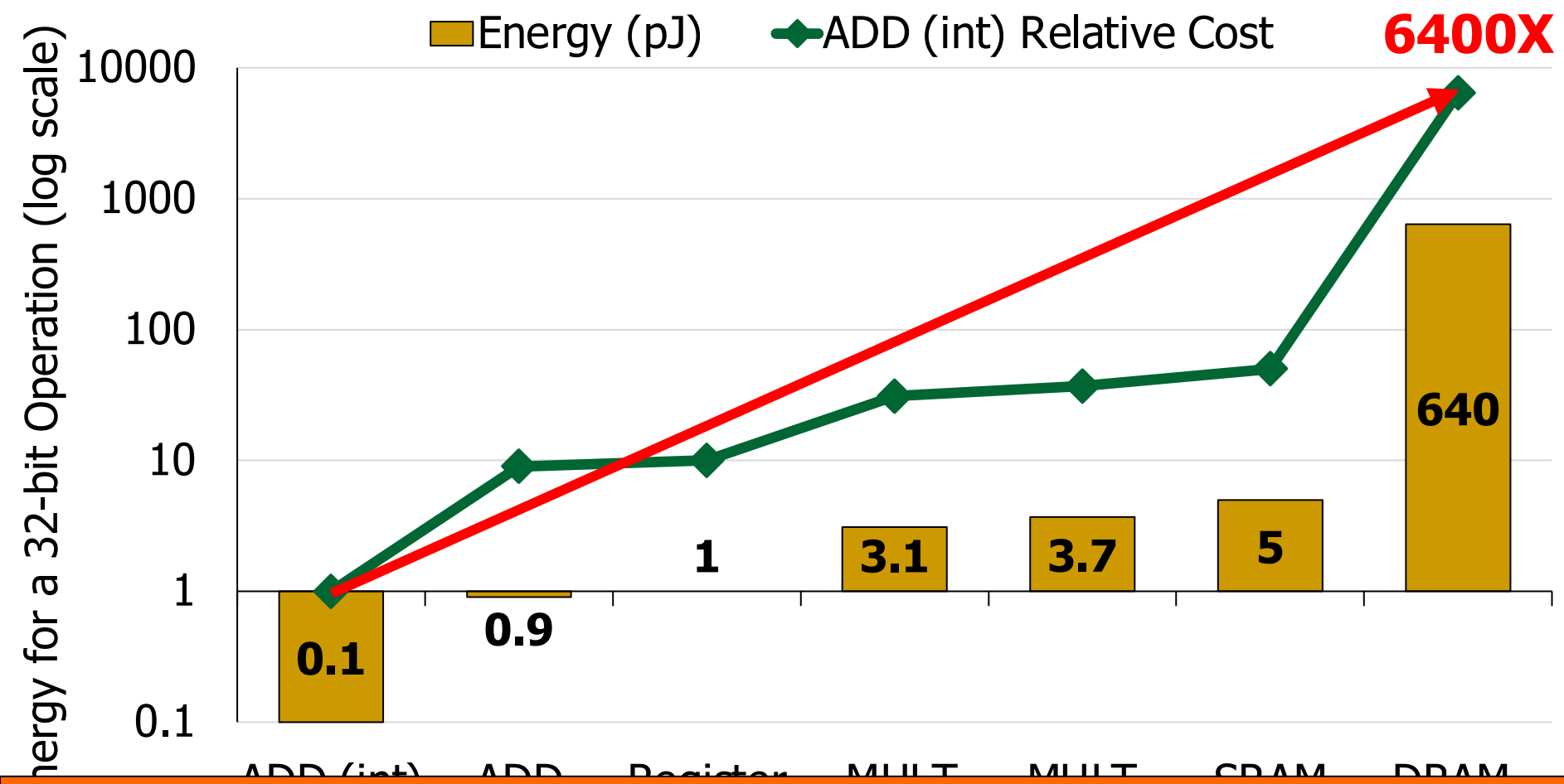
A memory access consumes ~100-1000X the energy of a complex addition

Data Movement vs. Computation Energy



Han+, "EIE: Efficient Inference Engine on Compressed Deep Neural Network," ISCA 2016.

Data Movement vs. Computation Energy



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

Energy Waste in 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.

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

Energy Waste in Accelerators

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

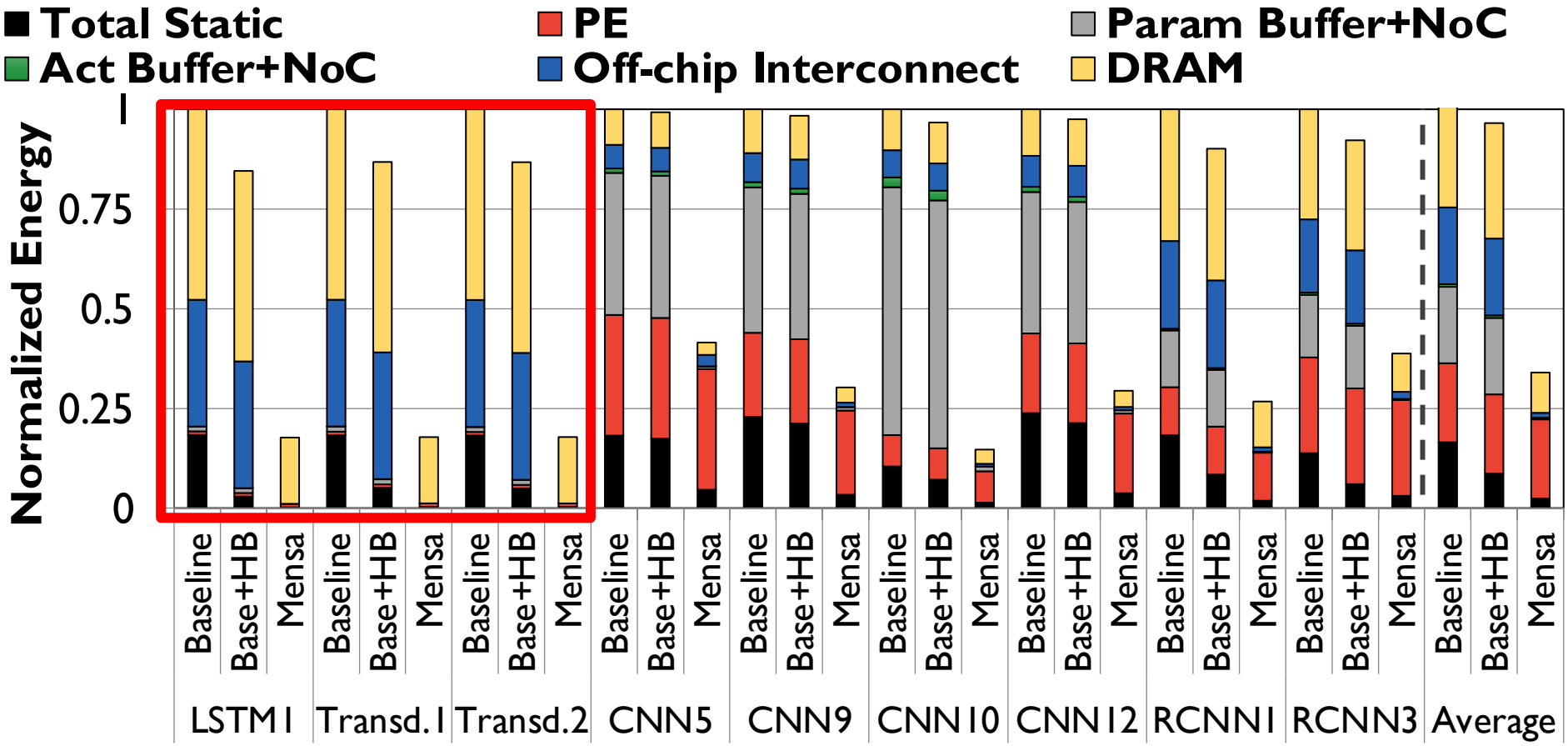
> 90% of the total system energy is spent on memory in large ML models

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

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

Energy Wasted on Data Movement



In LSTMs and Transducers used by Google,
>90% energy spent on off-chip interconnect and DRAM

https://arxiv.org/pdf/2109.14320

Fundamental Problem

Processing of data

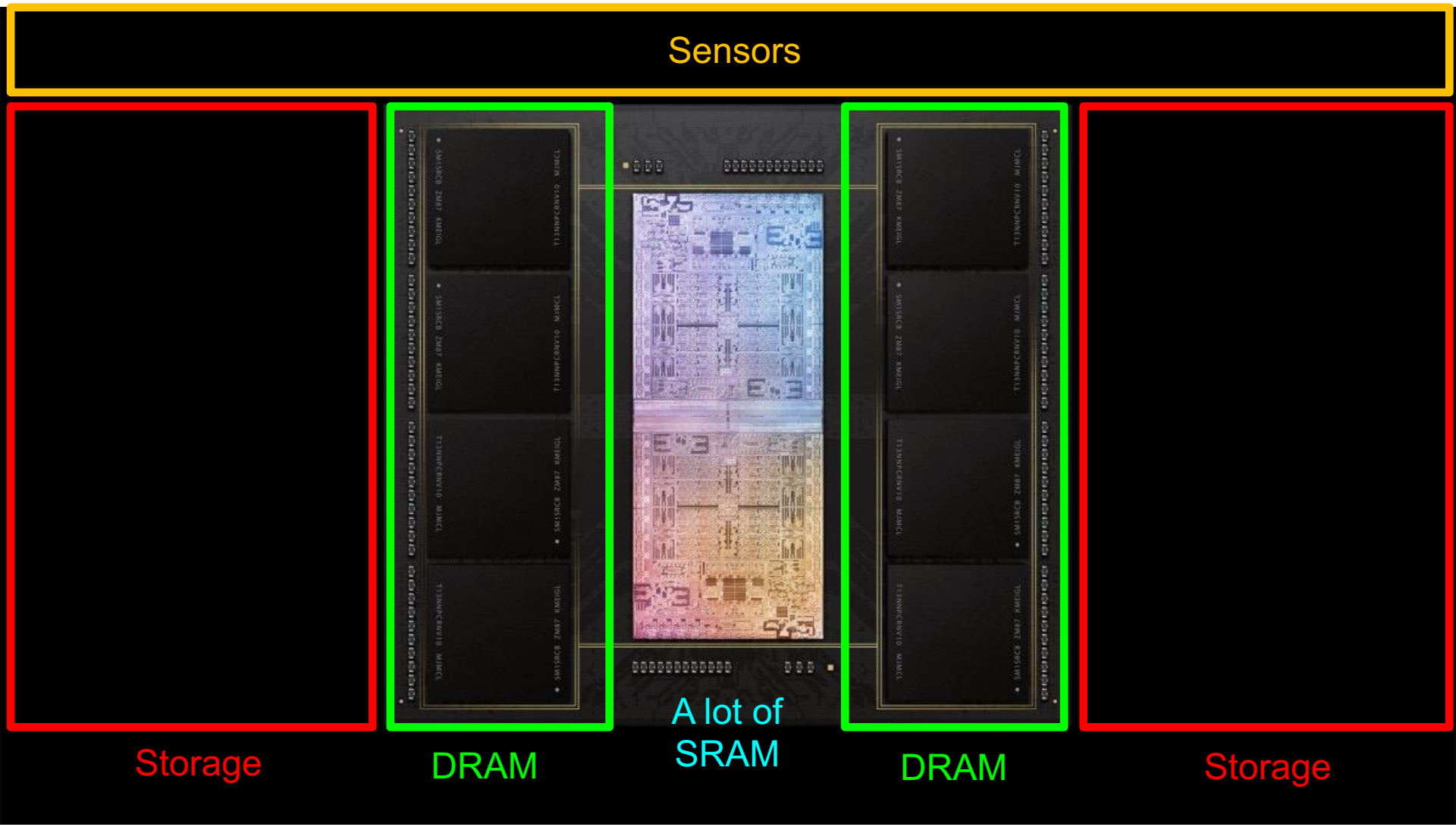
is performed

far away from the data

We Need A **Paradigm Shift** To …

- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric

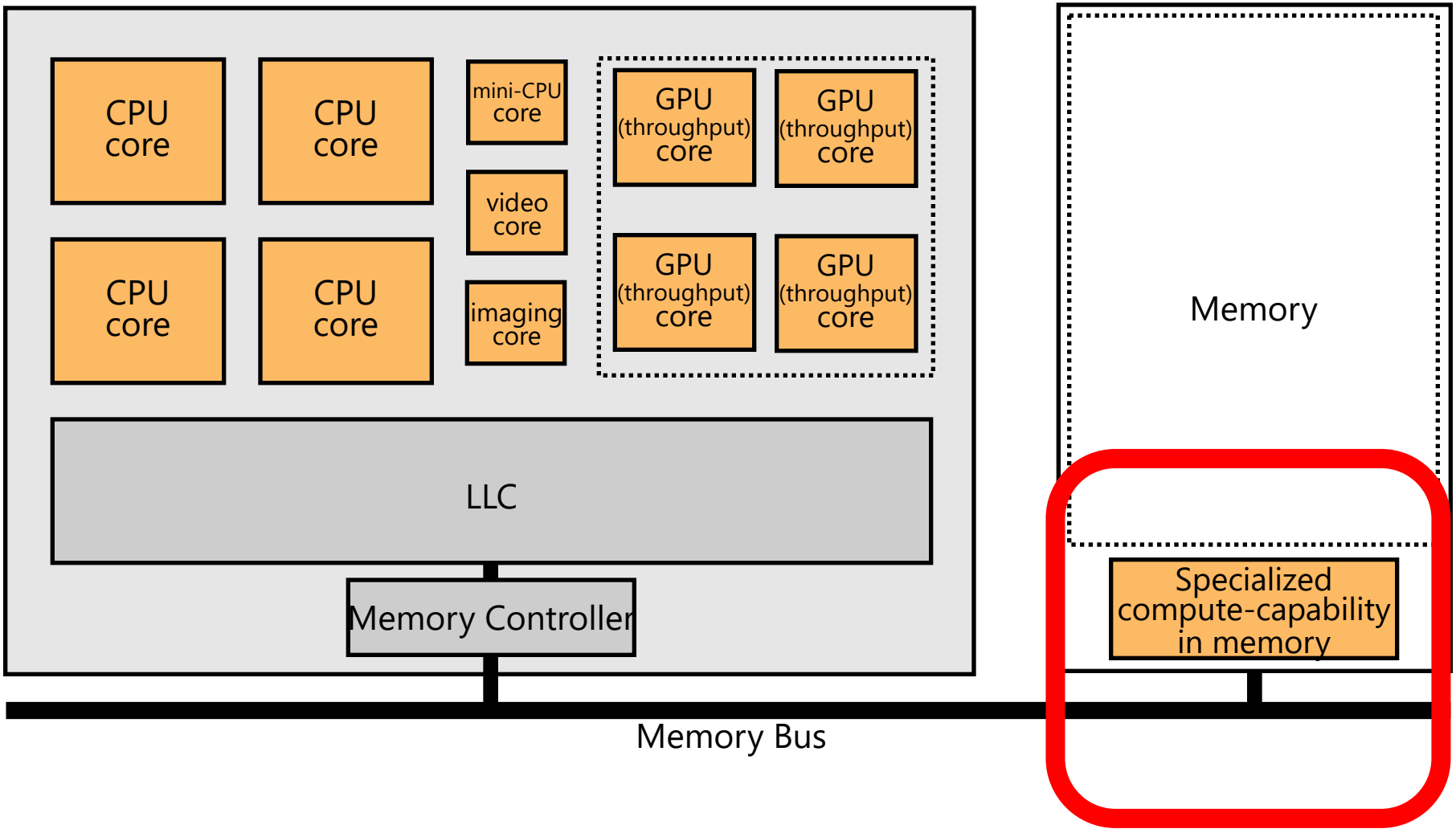
Process Data Where It Makes Sense



Apple M1 Ultra System (2022)

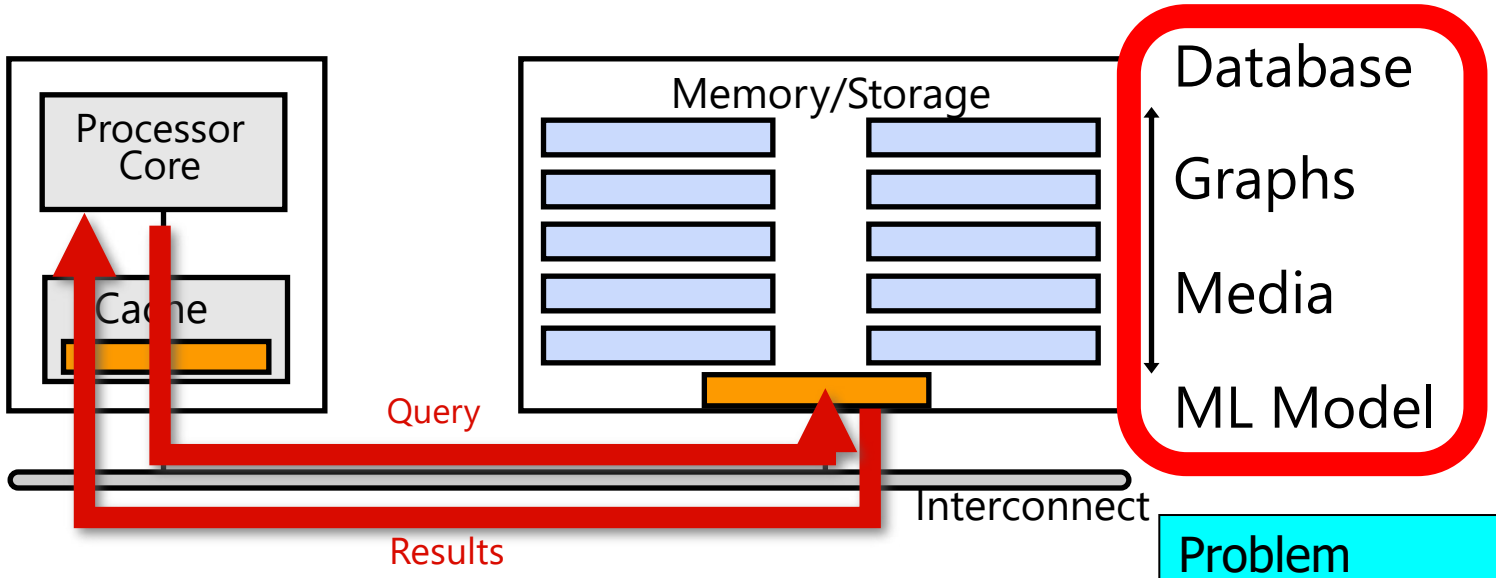
https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php

Memory as an Accelerator

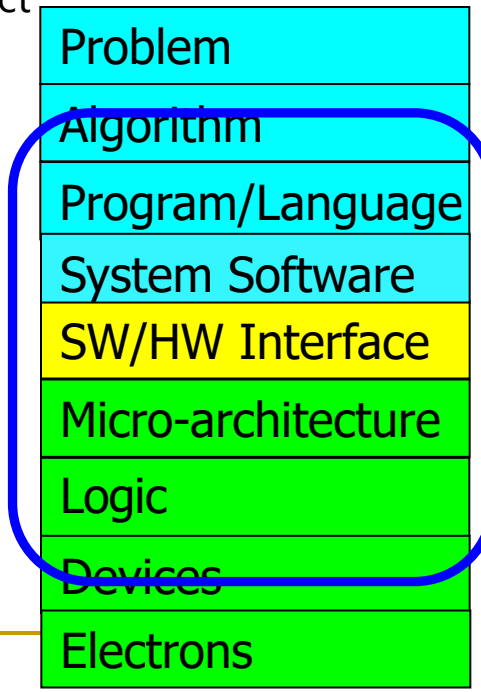


Memory similar to a "conventional" accelerator

Goal: Processing Inside Memory/Storage



- Many questions … How do we design the:
  - compute-capable memory & controllers?
  - processors & communication units?
  - software & hardware interfaces?
  - system software, compilers, languages?
  - algorithms & theoretical foundations?



Processing in/near Memory: An Old Idea

- Kautz, "Cellular Logic-in-Memory Arrays", IEEE TC 1969.

IEEE TRANSACTIONS ON COMPUTERS, VOL. C-18, NO. 8, AUGUST 1969

Cellular Logic-in-Memory Arrays

WILLIAM H. KAUTZ, MEMBER, IEEE

Abstract—As a direct consequence of large-scale integration, many advantages in the design, fabrication, testing, and use of digital circuitry can be achieved if the circuits can be arranged in a two-dimensional iterative, or cellular, array of identical elementary networks, or cells. When a small amount of storage is included in each cell, the same array may be regarded either as a logically enhanced memory array, or as a logic array whose elementary gates and connections can be "programmed" to realize a desired logical behavior.

In this paper the specific engineering features of such cellular logic-in-memory (CLIM) arrays are discussed, and one such special-purpose array, a cellular sorting array, is described in detail to illustrate how these features may be achieved in a particular design. It is shown how the cellular sorting array can be employed as a single-address, multiword memory that keeps in order all words stored within it. It can also be used as a content-addressed memory, a pushdown memory, a buffer memory, and (with a lower logical efficiency) a programmable array for the realization of arbitrary switching functions. A second version of a sorting array, operating on a different sorting principle, is also described.

Index Terms—Cellular logic, large-scale integration, logic arrays logic in memory, push-down memory, sorting, switching functions.

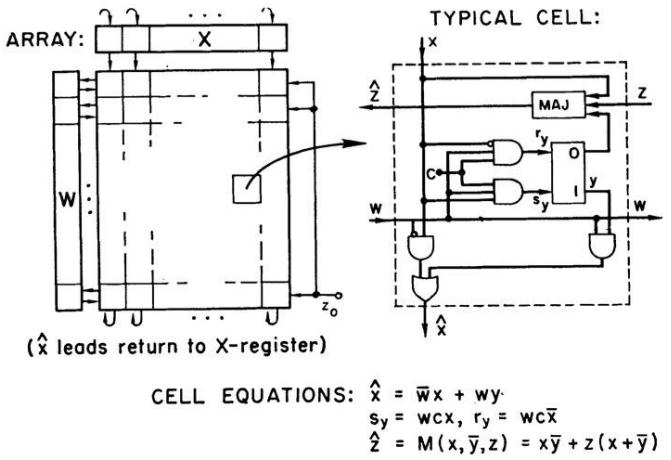


Fig. 1. Cellular sorting array I.

https://doi.org/10.1109/T-C.1969.222754

Processing in/near Memory: An Old Idea

- Stone, "A Logic-in-Memory Computer," IEEE TC 1970.

A Logic-in-Memory Computer

HAROLD S. STONE

Abstract—If, as presently projected, the cost of microelectronic arrays in the future will tend to reflect the number of pins on the array rather than the number of gates, the logic-in-memory array is an extremely attractive computer component. Such an array is essentially a microelectronic memory with some combinational logic associated with each storage element.

Why In-Memory Computation Today?

- **Huge demand from Applications & Systems**
  - Data access bottleneck
  - Energy & power bottlenecks
  - Data movement energy dominates computation energy
  - Need all at the same time: performance, energy, sustainability
  - We can improve all metrics by minimizing data movement

- **Huge problems with Memory Technology**
  - Memory technology scaling is not going well (e.g., RowHammer)
  - Many scaling issues demand intelligence in memory
  - Emerging technologies can enable new functions in memory

- **Designs are squeezed in the middle**

Approach & Takeaway

To Truly Solve Fundamental Problems,

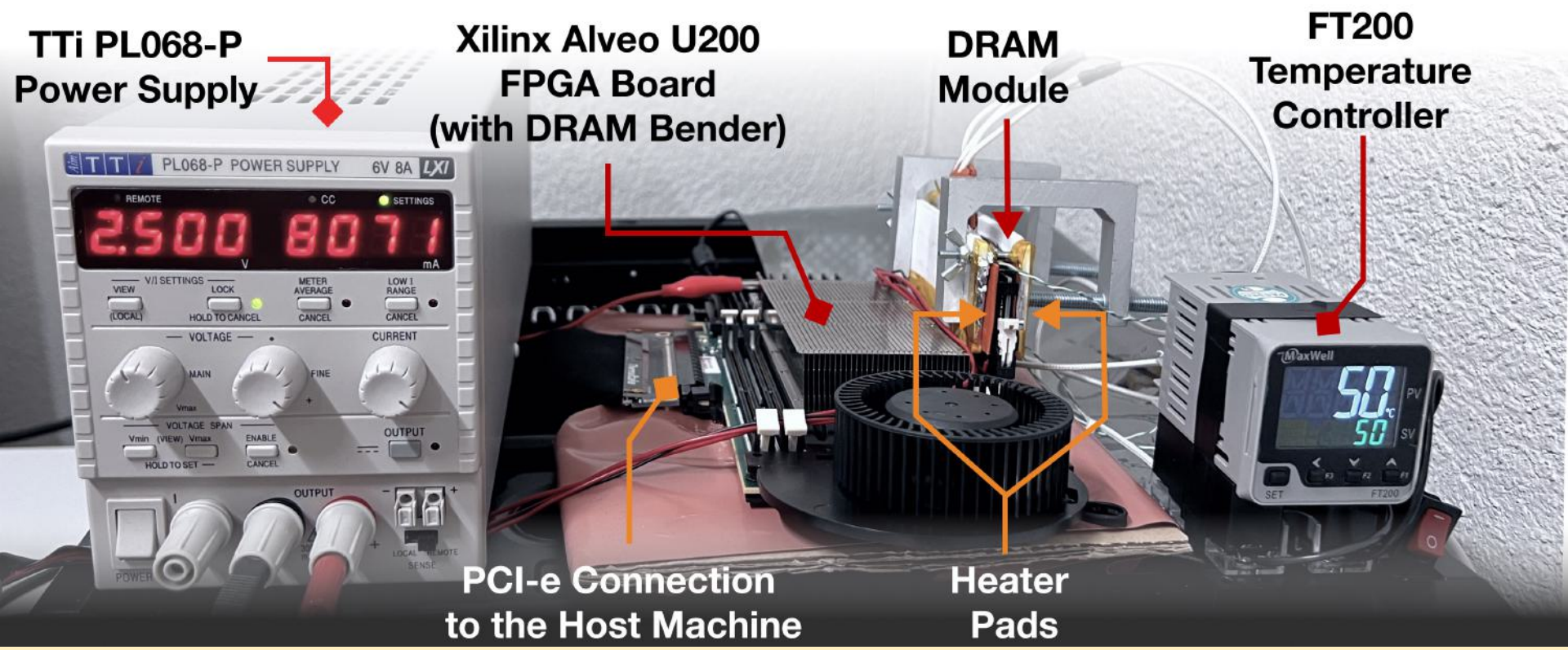
Defy Business as Usual

Ask: What would a smart 10-year-old kid do?

Memory Technology Scaling

Infrastructures to Understand Scaling Issues

DRAM Bender on a Xilinx Virtex UltraScale+ XCU200

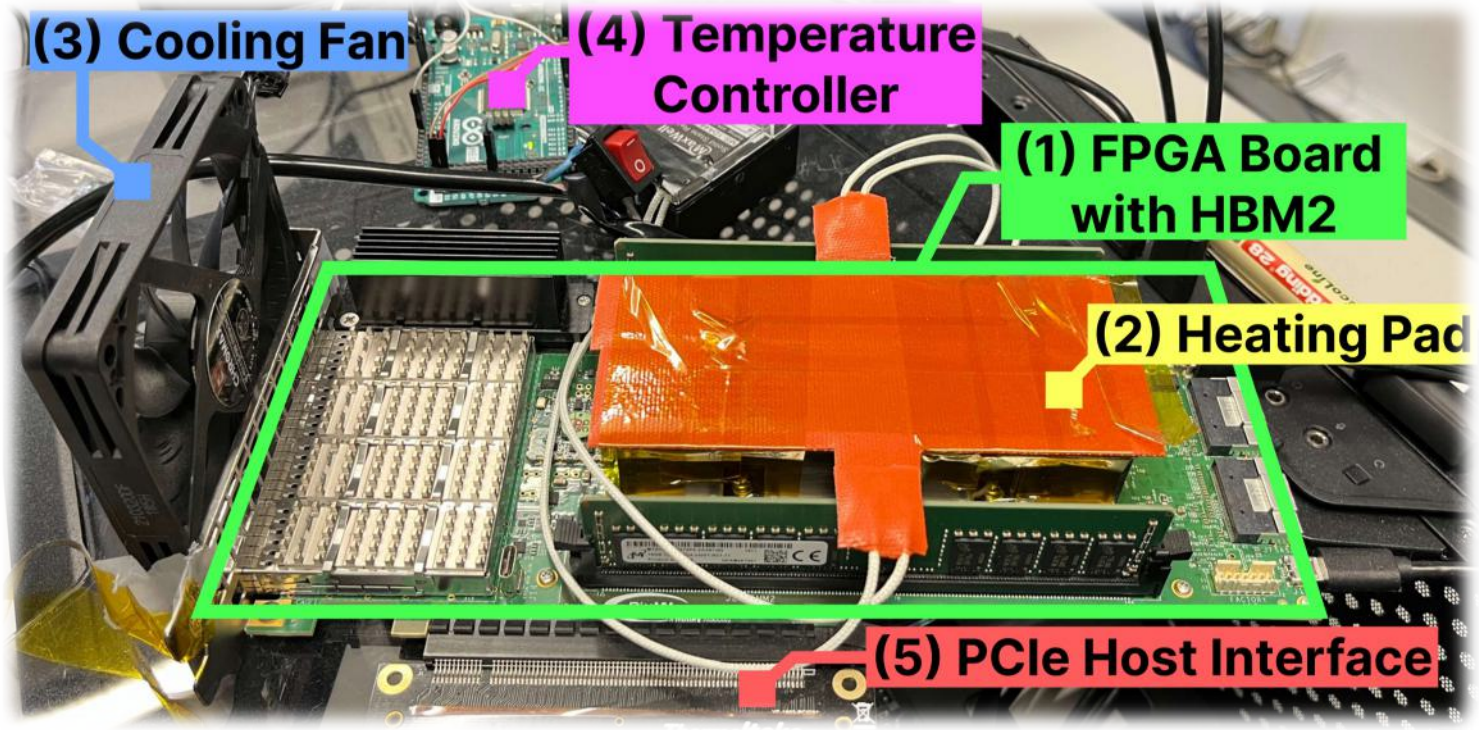


Fine-grained control over **DRAM commands**, **timing parameters (±1.5ns)**, **temperature (±0.5°C)**, and **voltage (±1mV)**

*Olgun et al., "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," in TCAD, 2023. [GitHub: https://github.com/CMU-SAFARI/DRAM-Bender]

HBM2 DRAM Testing Infrastructure

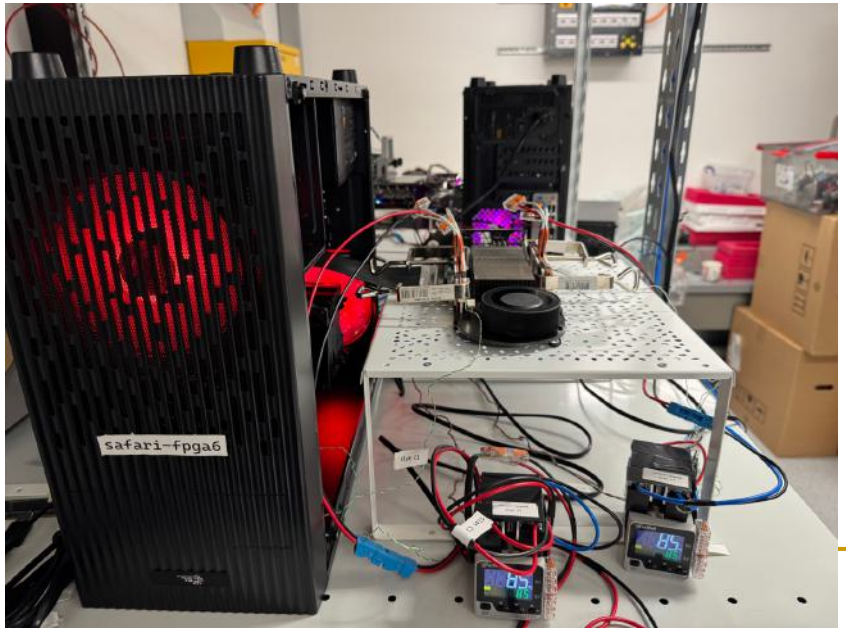
DRAM Bender on a Bittware XUPVVH



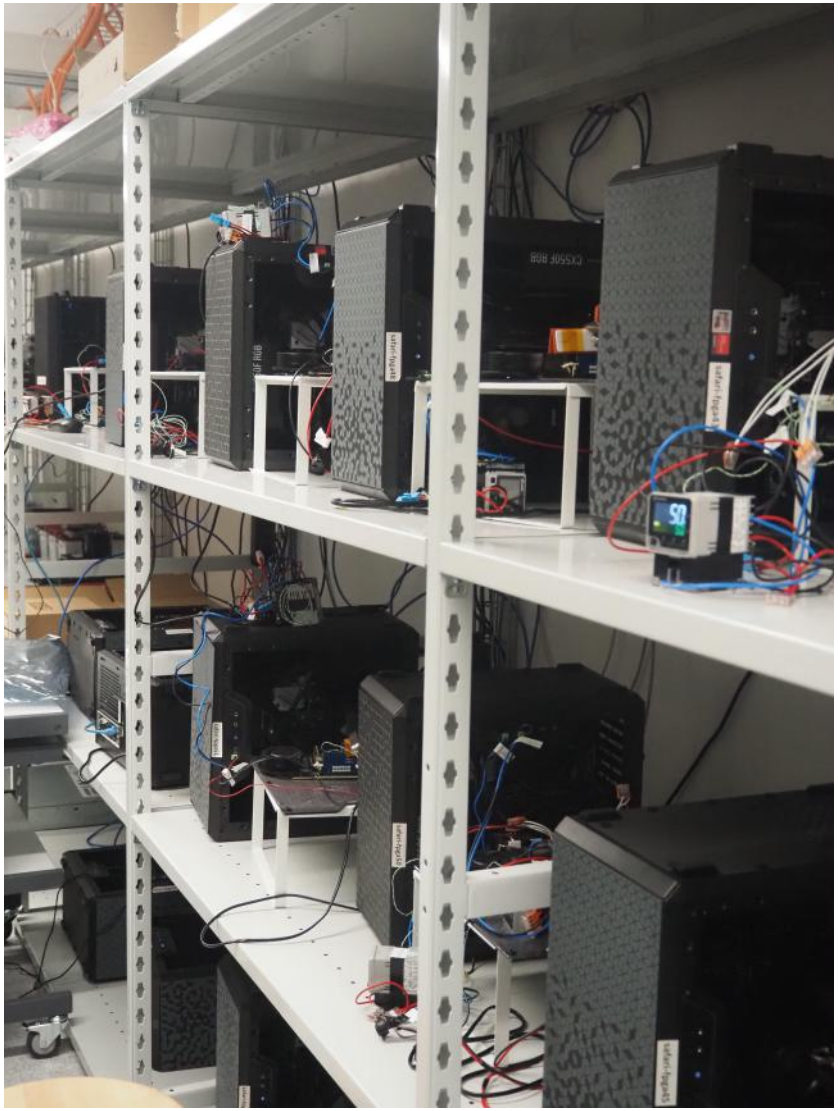
Fine-grained control over DRAM commands,
timing parameters (±1.67ns), and temperature (±0.5°C)

*Olgun et al., **"DRAM Bender: An Extensible and Versatile FPGA-Based Infrastructure to Easily Test State-of-the-Art DRAM Chips,"** TCAD, 2023. [GitHub: https://github.com/CMU-SAFARI/DRAM-Bender]

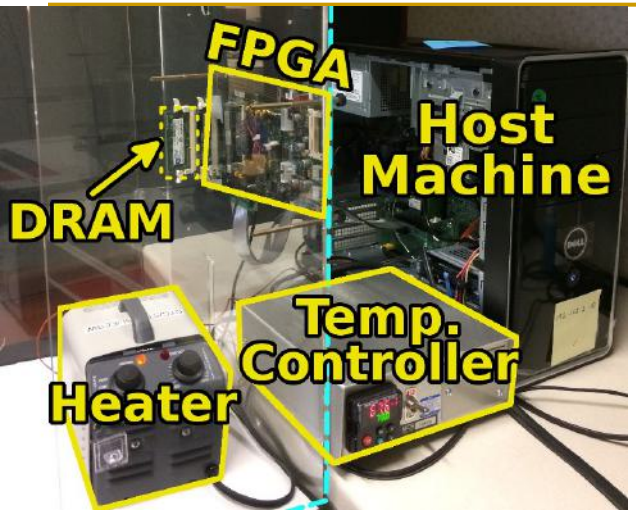
Laboratory for Understanding Memory



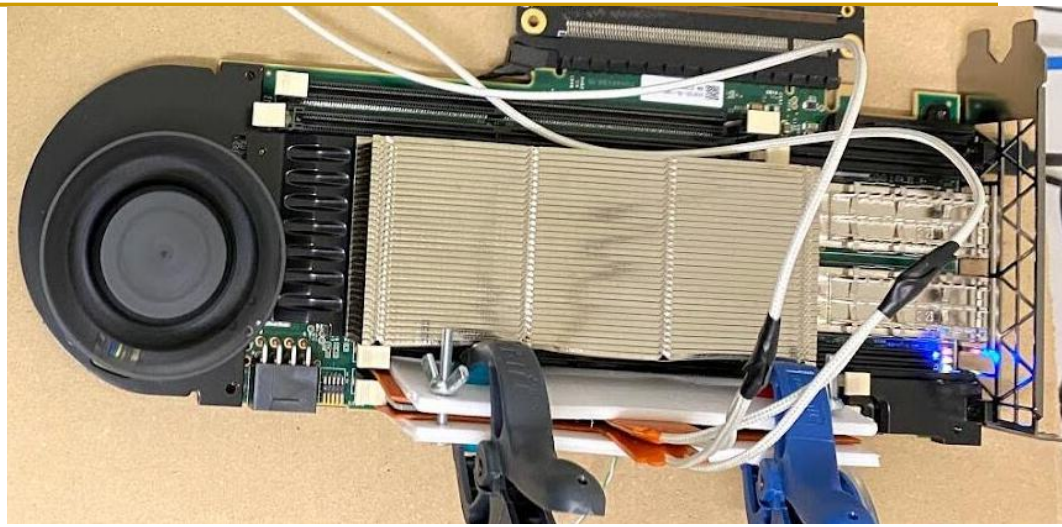
Laboratory for Understanding Memory



DRAM Testing Infrastructures (I)



DDR3 DRAM SODIMMs
Xilinx ML605

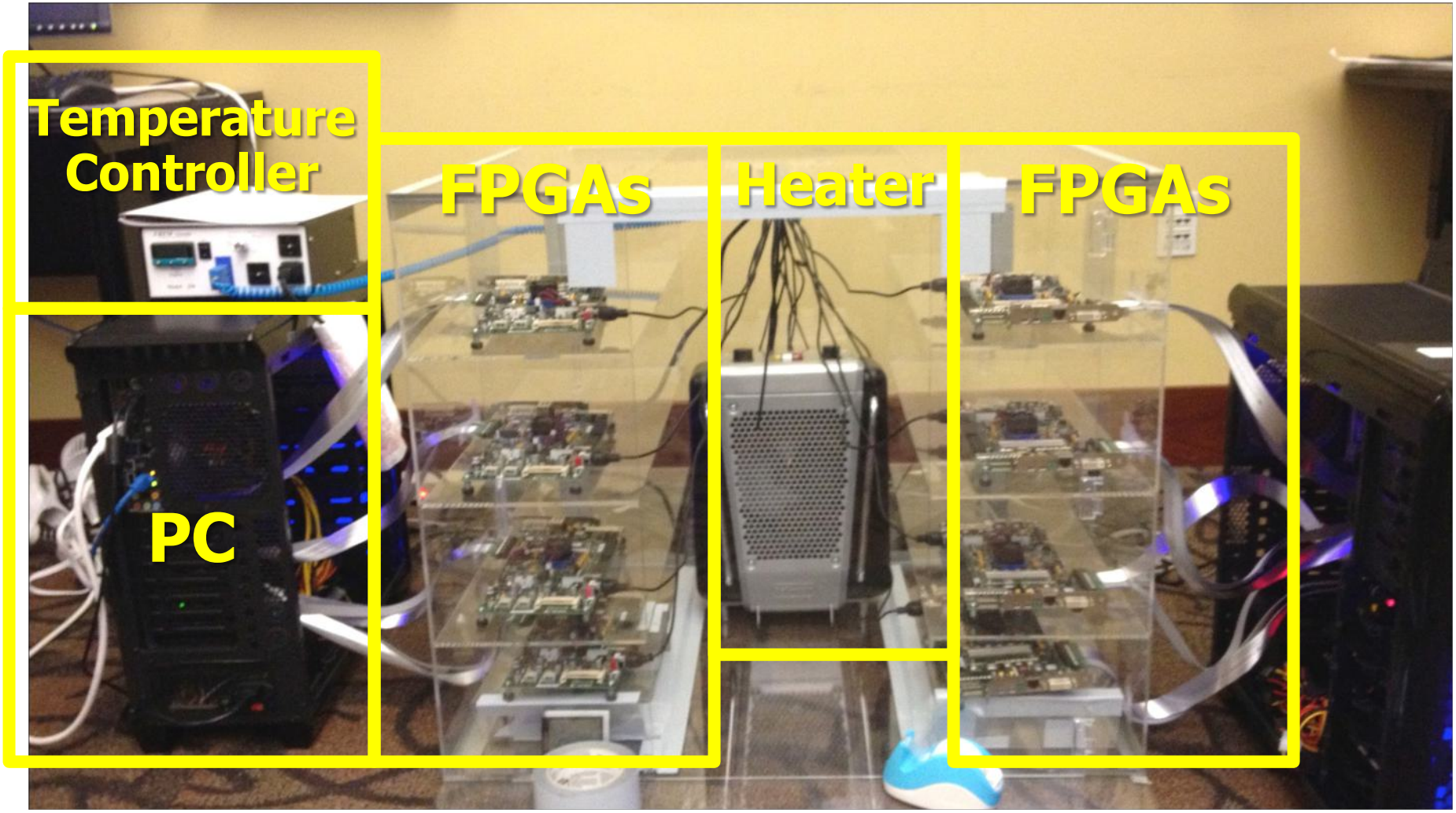


DDR4 DRAM R/UDIMMs
Xilinx Alveo U200

DDR4 DRAM (SODIMM)
Bittware XUSP3S

HBM2 DRAM Chips
Xilinx Alveo U50

DRAM Testing Infrastructures (II)



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

DRAM Testing Infrastructures (III)

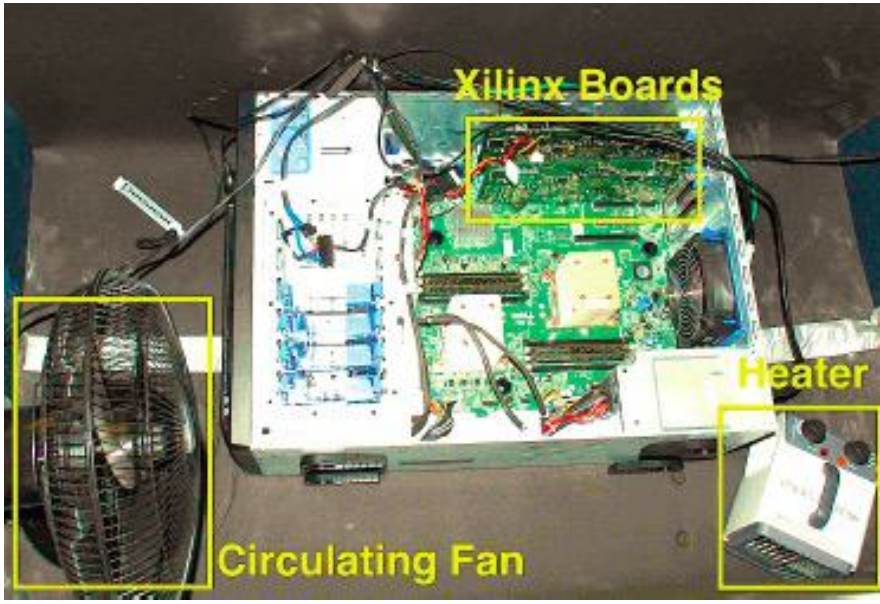
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)

An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)



SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,
  "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies"
  *Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (**HPCA**)*, Austin, TX, USA, February 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  [Full Talk Lecture (39 minutes)]
  [Source Code]

SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan[1,2,3] Nandita Vijaykumar[3] Samira Khan[4,3] Saugata Ghose[3] Kevin Chang[3]
Gennady Pekhimenko[5,3] Donghyuk Lee[6,3] Oguz Ergin[2] Onur Mutlu[1,3]

[1]*ETH Zürich* [2]*TOBB University of Economics & Technology* [3]*Carnegie Mellon University*
[4]*University of Virginia* [5]*Microsoft Research* [6]*NVIDIA Research*

DRAM Bender

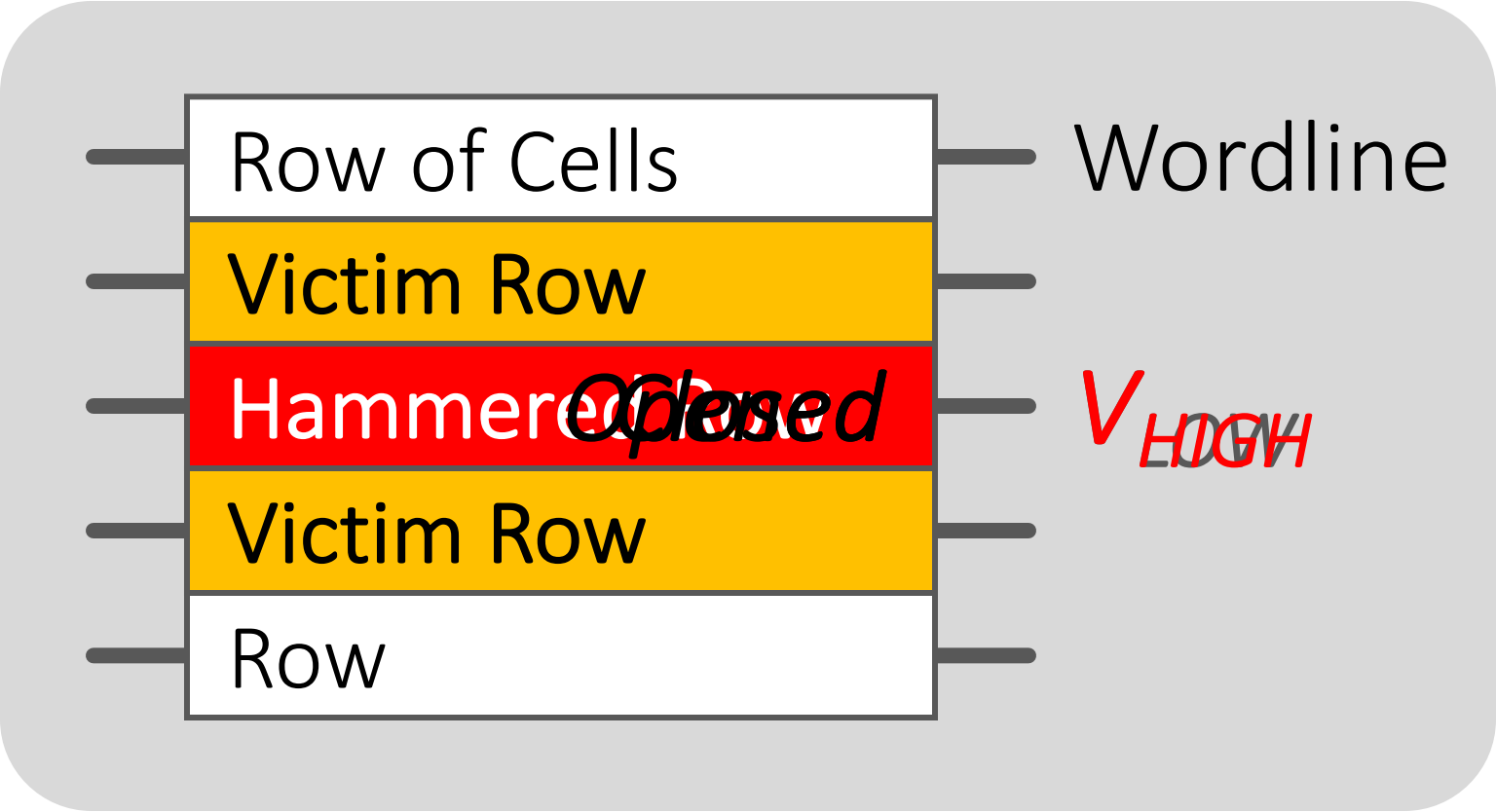
- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
  "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
  IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (***TCAD***), 2023.
  [Extended arXiv version]
  [DRAM Bender Source Code]
  [DRAM Bender Tutorial Video (43 minutes)]

DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul[§†]
Lois Orosa[§⊙] Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB ETÜ* [⊙]*Galician Supercomputing Center*


Rowhammer

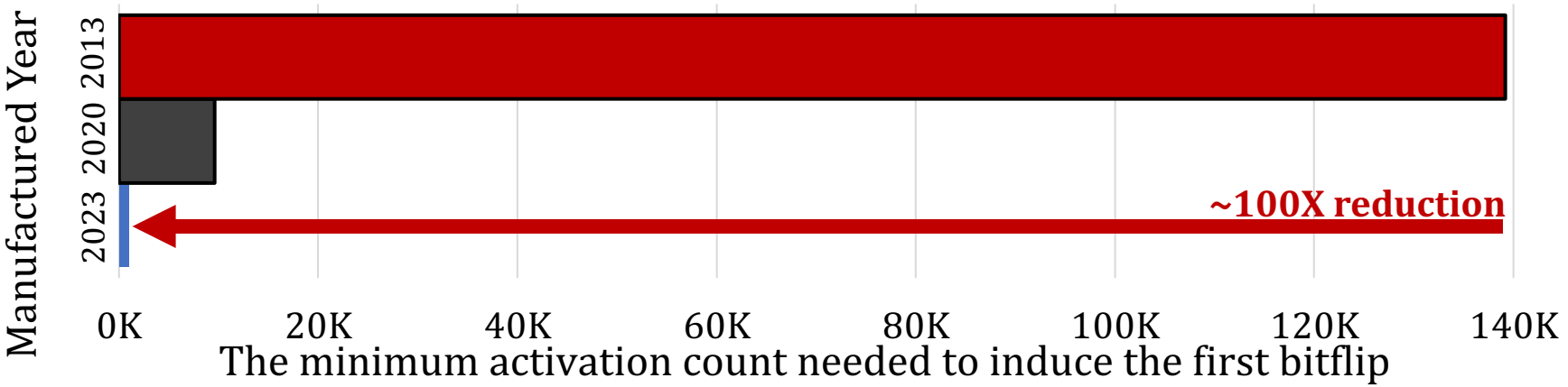
Modern DRAM is Prone to Disturbance Errors



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

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

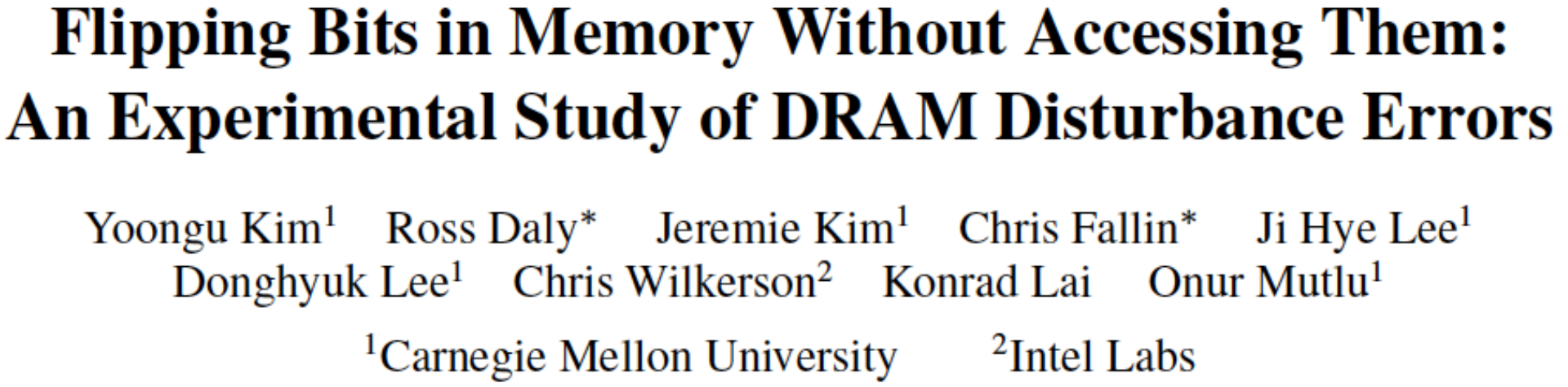
Read Disturbance Worsens with Scaling



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

RowHammer [ISCA 2014]

- 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] [Lecture Video (1 hr 49 mins), 25 September 2020]
  One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link). Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).



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

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

[1]Carnegie Mellon University [2]Intel Labs

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

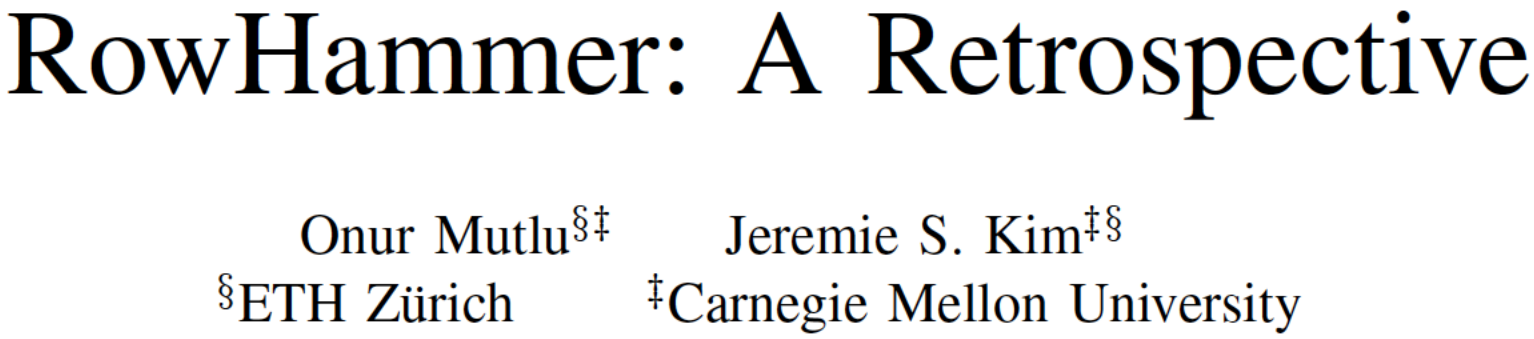
Exploiting the DRAM rowhammer bug to gain kernel privileges

Many RowHammer Security Exploits

- One can exploit RowHammer to
- Take over a system
- Read data they do not have access to
- Break out of virtual machine sandboxes
- Corrupt important data → render ML inference useless
- Steal secret data (e.g., crypto keys & ML model parameters)

A Long RowHammer Retrospective

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

A Short Retrospective @ 50 Years of ISCA

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

Onur Mutlu
ETH Zürich

***Abstract*—Our ISCA 2014 paper [1] provided the first scientific and detailed characterization, analysis, and real-system demonstration of what is now popularly known as the RowHammer phenomenon (or vulnerability) in modern commodity DRAM chips, which are used as main memory in almost all modern computing systems. It experimentally demonstrated that more than 80% of all DRAM modules we tested from the three major DRAM vendors were vulnerable to the RowHammer read disturbance phenomenon: one can predictably induce bitflips (i.e., data corruption) in real DRAM modules by repeatedly accessing a DRAM row and thus causing electrical disturbance to physically nearby rows. We showed that a simple unprivileged user-level program induced RowHammer bitflips in multiple real systems and suggested that a security attack can be built using this proof-of-concept to hijack control of the system or cause other harm. To solve the RowHammer problem, our paper examined seven different approaches (including a novel probabilistic approach that has very low cost), some of which influenced or were adopted in different industrial products.**

Many later works from various research communities examined RowHammer, building real security attacks, proposing new defenses, further analyzing the problem at various (e.g., device/circuit, architecture, and system) levels, and exploiting RowHammer for various purposes (e.g., to reverse-engineer DRAM chips). Industry has worked to mitigate the problem, changing both memory controllers and DRAM standards/chips. Two major DRAM vendors finally wrote papers on the topic in 2023, describing their current approaches to mitigate RowHammer. Research & development on RowHammer in both academia & industry continues to be very active and fascinating.

This short retrospective provides a brief analysis of our ISCA 2014 paper and its impact. We describe the circumstances that led to our paper, mention its influence on later works and products, describe the mindset change we believe it has helped enable in hardware security, and discuss our predictions for future.

I. Background and Circumstances

Our stumbling on the RowHammer problem and creation of its first scientific analysis happened as a result of a confluence of multiple factors. First, my group was working on DRAM technology scaling issues since late 2010. We were very interested in failure mechanisms that appear or worsen due to aggressive technology scaling. To study such issues (e.g., data retention errors [2]), we built an FPGA-based DRAM testing infrastructure [2] between 2011-2012, which we later open sourced as SoftMC [3, 4] and DRAM Bender [5, 6]. Second, around the same timeframe, we were investigating similar technology scaling issues in flash memory using real NAND flash chips [7, 8]. We knew read disturbance errors were significant in NAND flash memory [7–11] and were very interested in how prevalent they were in DRAM. Third, we were collaborating with Intel (e.g., [2]) to understand and solve DRAM technology scaling problems and build our DRAM infrastructure. Three of my students and I spent the summer of 2012 at Intel to work closely with our collaborators (two are co-authors): during this time, we finalized the calibration and stabilization of our infrastructure and had significant technical discussions and experimentation on DRAM scaling problems.

Although there was awareness of the RowHammer problem in industry in 2012 (see Footnote 1 in [1]), there was no comprehensive experimental analysis and detailed real-system demonstration of it. We believed it was critical to provide a rigorous scientific analysis using a wide variety of DRAM chips and scientifically establish major characteristics and prevalence of RowHammer. Hence, in the summer of 2012, we set out to use our DRAM testing infrastructure to analyze RowHammer. Our initial results showed how widespread the read disturbance problem was across the (at the time) recent DRAM chips we tested, so we studied the problem comprehensively and developed many solutions to it. The resulting paper was submitted to MICRO in May 2013 but was rejected. We strengthened the results, especially of the mitigation mechanisms and the number of tested chips, and made the analysis more comprehensive before it was accepted to ISCA 2014 (2 of the 6 reviewers still rejected it for interesting reasons).

II. Major Contribution and Influence

The major contribution of our paper is the exposure and detailed analysis of a fundamental hardware failure mechanism that breaks memory isolation in real systems and thus has huge implications on system reliability, security, and safety. Our paper is a comprehensive study of a major DRAM technology scaling problem, RowHammer, including its first scientific analysis, experimental characterization, real system demonstration, and solutions with their evaluation. To our knowledge, RowHammer is the first example of a hardware failure mechanism that creates a significant and widespread system security vulnerability [12–15], as our ISCA 2014 paper suggested.

Our work has had large influence on both industry & academia. Individual follow-on works are many to list here; we refer the reader to longer invited retrospectives we wrote [12–14]. We give major examples of influence, focusing on RowHammer's effect on the collective mindset of security research and major industry milestones related to RowHammer.

RowHammer Attacks & Mindset Shift in Hardware Security. Our demonstration that one can easily and predictably induce bitflips in commodity DRAM chips using a real user-level program enabled a major mindset shift in hardware security. It showed that general-purpose hardware is fallible in a very widespread manner and its problems are exploitable. Tens of works (see [13, 14]) built directly on our work to exploit RowHammer bitflips to develop many attacks that compromise system integrity and confidentiality, starting from the first RowHammer exploit by Google Project Zero in 2015 [16, 17] to recent works in 2022-2023 (e.g., [18, 19]). These attacks showed increasingly sophisticated ways by which an unprivileged attacker can exploit RowHammer bitflips to circumvent memory protection and gain complete control of a system (e.g., [16, 20–28]), gain access to confidential data (e.g., [18, 19, 29]), or maliciously destroy the safety and accuracy of a system, e.g., an otherwise accurate machine learning inference engine (e.g., [30, 31]). The mindset enabled by RowHammer bitflips caused a renewed interest in hardware security research, enticing many researchers to deeply understand hardware's inner workings and find new vulnerabilities. Thus, hardware security issues have become mainstream discussion in top security & architecture venues, some having sessions entitled RowHammer.

RowHammer Defenses. Tens of works proposed mitigations against RowHammer, some of which were inspired by the solutions we discussed in our ISCA 2014 paper. To date, the search for more efficient and low-cost RowHammer solutions continues. We refer the reader to our prior overview papers [13, 14, 32] and more recent works in 2023 (e.g., [33–35]).

RowHammer Analyses. Our paper initiated works at both architectural & circuit/device-levels to better understand RowHammer and reverse-engineer DRAM chips, to develop better models, defenses, and attacks (see [13, 14]). Our ISCA'20 work [36] revisited RowHammer, comprehensively analyzed of 1580 DRAM chips of three different types from at least two generations, showing that RowHammer has gotten much worse with technology scaling & existing solutions are not effective at future vulnerability levels.

Industry Reaction: Attacks, Analyses, and Mitigations. Folks developing industrial memory testing programs immediately included RowHammer tests, e.g., in memtest86 [37], citing our work. Industry needed to immediately protect RowHammer-vulnerable chips already in the field, so almost all system vendors increased refresh rates; a solution we examined in our paper and deemed costly for performance and energy, yet it was the only practical lever that could be used in the field. Apple publicly acknowledged our work in their security release [38] that announced higher refresh rates to mitigate RowHammer. Intel designed memory controllers that performed probabilistic activations (i.e., pTRR [39, 40]), similar to our PARA solution [1]. DRAM vendors modified the DRAM standard to introduce TRR (target row refresh) mechanisms [39] and claimed their new DDR4 chips to be RowHammer-free [39, 41]. This bold claim was later refuted by our TRRespass work [39] in 2020, which introduced the many-sided RowHammer attack to circumvent internal protection mechanisms added to the DRAM chips. Our later work, Uncovering TRR [41] showed that one can almost completely reverse-engineer and thus easily bypass RowHammer mitigations employed in all tested DRAM chips, i.e., RowHammer solutions in DRAM chips are broken. The analysis done by our two major works in 2020 [36, 39] caused the industry to reorganize the RowHammer task group at JEDEC, which produced two white papers on mitigating RowHammer [42, 43]. Nine years after our paper, in 2023, two major DRAM vendors, SK Hynix and Samsung, finally wrote papers [44, 45] on the RowHammer problem, describing their solutions. Several of these industry solutions build on the probabilistic & access-counter-based solution approaches our ISCA 2014 paper introduced.

Major Internet and cloud systems companies also took a deep interest in RowHammer as it can greatly impact their system security, dependability, and availability. Multiple works from Google, e.g., by Google Project Zero in 2015 [16, 17] and Half Double in 2021-2022 [46] directly built on our paper to demonstrate attacks in real systems. Researchers from Microsoft have developed deeper analyses of RowHammer [47], along with new RowHammer attacks [48] and defenses (e.g., [48–51]).

III. Summary and Future Outlook

Since 2012-2014, RowHammer vulnerability has become much worse due to technology scaling: without mitigation, one can now induce RowHammer bitflips with orders of magnitude smaller number of activations (e.g., ∼10K) and cause much higher rates of errors in cutting-edge DRAM chips [36, 41]. Sophisticated attacks are continuously developed to circumvent the mitigations employed in real DRAM chips. Fortunately, we have also come a long way in further understanding and better mitigating the RowHammer vulnerability. The industry is now (hopefully) fully aware of the importance of the problem and of avoiding bitflips. Unfortunately, an efficient and completely-secure solution is not found yet. The solution space poses a rich area of tradeoffs in terms of security, performance, power/energy, cost/complexity. All solutions forego some desirable properties in favor of others. As such, a critical direction for the future is to find solutions superior to what we have today. We believe system-DRAM cooperation [14, 52] will be important to enabling complete solutions. We also believe it is critical to deeply understand the properties of RowHammer under many different conditions so that we can develop effective solutions that work under all circumstances. Unfortunately, we do not yet fully understand many facets of RowHammer (see [14, 53–55]).

DRAM technology scaling will continue to create problems that will exacerbate the bitflips and the resulting robustness (i.e., safety/security/reliability) problems. Our ISCA 2023 paper on RowPress [55] provides the first scientific and detailed characterization, analysis, and real-system demonstration of yet another read disturbance mechanism in DRAM. What other fascinating problems will we see and can we completely solve them efficiently? Will we ever be free of bitflips at the system and application levels?

References

[1] Y. Kim *et al.*, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," in *ISCA*, 2014.
[2] J. Liu *et al.*, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms," *ISCA*, 2013.
[3] H. Hassan *et al.*, "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies," in *HPCA*, 2017.
[4] SoftMC Source Code. https://github.com/CMU-SAFARI/SoftMC.
[5] A. Olgun *et al.*, "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," *TCAD*, 2023.
[6] "DRAM Bender." https://github.com/CMU-SAFARI/DRAM-Bender.
[7] Y. Cai *et al.*, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," in *DATE*, 2012.
[8] Y. Cai *et al.*, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," *ITJ*, 2013.
[9] Y. Cai *et al.*, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation," in *ICCD*, 2013.
[10] Y. Cai *et al.*, "Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery," in *DSN*, 2015.
[11] Y. Cai *et al.*, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid-State Drives," *Proc. IEEE*, 2017.
[12] O. Mutlu, "The RowHammer Problem and Other Issues we may Face as Memory Becomes Denser," *DATE*, 2017.
[13] O. Mutlu and J. Kim, "RowHammer: A Retrospective," *IEEE TCAD Special Issue on Top Picks in Hardware and Embedded Security*, 2019.
[14] O. Mutlu *et al.*, "Fundamentally Understanding and Solving RowHammer," in *ASP-DAC*, 2023.
[15] T. Dullien, "Security, Moore's Law, and the Anomaly of Cheap Complexity," in *CCDCOE*, 2018, https://www.youtube.com/watch?v=q98foLaAfX8.
[16] M. Seaborn and T. Dullien, "Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges." http://googleprojectzero.blogspot.com.tr/2015/03/exploiting-dram-rowhammer-bug-to-gain.html, 2015.
[17] M. Seaborn and T. Dullien, "Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges," *Black Hat*, 2015.
[18] A. S. Rakin *et al.*, "DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories," in *S&P*, 2022.
[19] K. Mus *et al.*, "JolTik Recovering TLS Signing Keys via Rowhammer Faults," in *S&P*, 2023.
[20] D. Gruss *et al.*, "Rowhammer.js: A Remote Software-Induced Fault Attack in Javascript," in *DIMVA*, 2016.
[21] V. van der Veen *et al.*, "Drammer: Deterministic Rowhammer Attacks on Mobile Platforms," in *CCS*, 2016.
[22] Y. Xiao *et al.*, "One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation," in *USENIX Security*, 2016.
[23] K. Razavi *et al.*, "Flip Feng Shui: Hammering a Needle in the Software Stack," *USENIX Security*, 2016.
[24] A. Tatar *et al.*, "Throwhammer: Rowhammer Attacks Over the Network and Defenses," in *USENIX ATC*, 2018.
[25] M. Lipp *et al.*, "Nethammer: Inducing Rowhammer Faults Through Network Requests," arXiv:1805.04956, 2018.
[26] L. Cojocar *et al.*, "Exploiting Correcting Codes: On the Effectiveness of ECC Memory Against Rowhammer Attacks," in *S&P*, 2019.
[27] F. de Ridder *et al.*, "SMASH: Synchronized Many-Sided Rowhammer Attacks from JavaScript," in *USENIX Security*, 2021.
[28] P. Jattke *et al.*, "Blacksmith: Scalable Rowhammering in the Frequency Domain," in *S&P*, 2022.
[29] A. Kwong *et al.*, "RAMBleed: Reading Bits in Memory Without Accessing Them," in *S&P*, 2020.
[30] S. Hong *et al.*, "Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks," in *SS*, 2019.
[31] F. Yao *et al.*, "DeepHammer: Depleting the Intelligence of Deep Neural Networks Through Targeted Chain of Bit Flips," in *USENIX Security*, 2020.
[32] A. G. Yağlıkcı *et al.*, "BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows," in *HPCA*, 2021.
[33] M. Marazzi *et al.*, "ProTRR: Principled yet Optimal In-DRAM Target Row Refresh," in *S&P*, 2023.
[34] M. Wi *et al.*, "SHADOW: Preventing Row Hammer in DRAM with Intra-Subarray Row Shuffling," in *HPCA*. IEEE, 2023.
[35] J. Juffinger *et al.*, "CSI: Rowhammer-Cryptographic Security and Integrity against Rowhammer (to appear)," in *S&P*, 2023.
[36] S. Kim *et al.*, "Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques," in *ISCA*, 2020.
[37] PassMark Software, "MemTest86: The Original Industry Standard Memory Diagnostic Utility," http://www.memtest86.com/troubleshooting.htm, 2015.
[38] Apple Inc., "About the Security Content of Mac EFI Security Update 2015-001," https://support.apple.com/en-us/HT204934, 2015.
[39] P. Frigo *et al.*, "TRRespass: Exploiting the Many Sides of Target Row Refresh," in *S&P*, 2020.
[40] M. Kaczmarski, "Thoughts on Intel Xeon E5-2600 v2 Product Family Performance Optimisation – Component Selection Guidelines," http://infotazy.gda.pl/2014/pliki/prezentacje/d25x24-Kaczmarski-Optymalna.pdf, page 13, 2014.
[41] H. Hassan *et al.*, "Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications," in *MICRO*, 2021.
[42] JEDEC, *JEP300-1: Near-Term DRAM Level RowHammer Mitigation*, 2021.
[43] JEDEC, *JEP301-1: System Level RowHammer Mitigation*, 2021.
[44] W. Kim, "A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement," in *ISSCC*, 2023.
[45] S. Hong *et al.*, "DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm," arXiv:2302.03591, 2023.
[46] A. Kogler *et al.*, "Half-Double: Hammering From the Next Row Over," in *USENIX Security*, 2022.
[47] L. Cojocar *et al.*, "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers," in *S&P*, 2020.
[48] K. Loughlin *et al.*, "MOESI-Prime: Preventing Coherence-Induced Hammering in Commodity Workloads," in *ISCA*, 2022.
[49] T. Bennett *et al.*, "Panopticon: A Complete In-DRAM Rowhammer Mitigation," in *DRAMSec*, 2021.
[50] K. Loughlin *et al.*, "Stop! Hammer Time: Rethinking Our Approach to Rowhammer Mitigations," in *HotOS*, 2021.
[51] S. Saroiu and A. Wolman, "How to Configure Row-Sampling-Based Rowhammer Defenses," *DRAMSec*, 2022.
[52] O. Mutlu, "Memory Scaling: A Systems Architecture Perspective," in *IMW*, 2013.
[53] L. Orosa, "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses," in *MICRO*, 2021.
[54] A. G. Yağlıkcı *et al.*, "Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices," in *DSN*, 2022.
[55] H. Luo *et al.*, "RowPress: Amplifying Read Disturbance in Modern DRAM Chips," in *ISCA*, 2023.

https://arxiv.org/pdf/2306.16093

RowPress [ISCA 2023]

- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu, **"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"** *Proceedings of the 50th International Symposium on Computer Architecture (**ISCA**)*, Orlando, FL, USA, June 2023.
  [Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (3 minutes)]
  [RowPress Source Code and Datasets (Officially Artifact Evaluated with All Badges)]
  Officially artifact evaluated as available, reusable and reproducible. Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

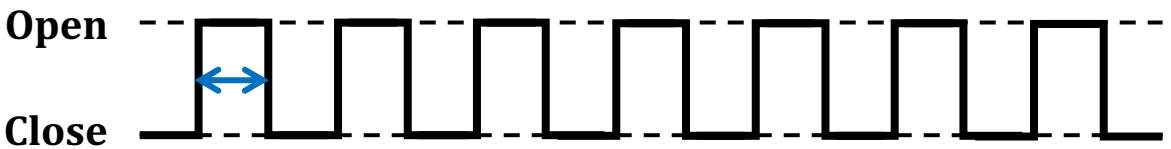
Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu
ETH Zürich

RowPress vs. RowHammer

Instead of using a high activation count,

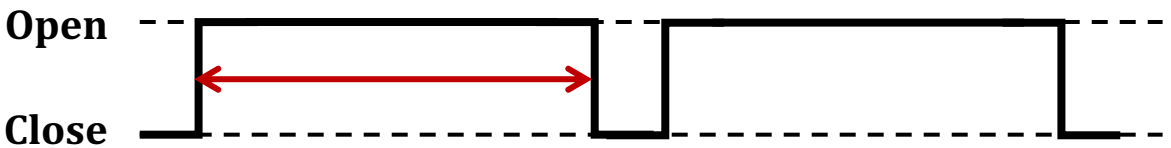
☞ increase the time that the aggressor row stays open

RowHammer Aggressor Row



36ns, 47K activations to induce bitflips

RowPress Aggressor Row

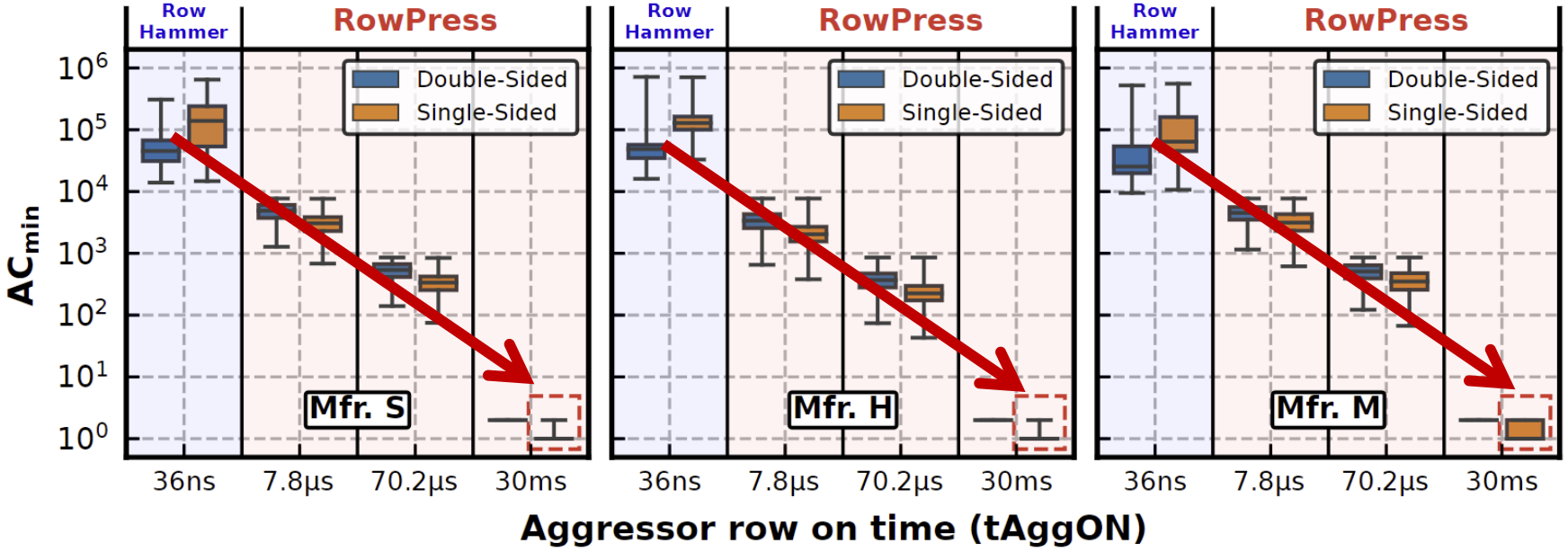


7.8µs, only 5K activations to induce bitflips

RowPress **reduces the number of activations** to induce a bitflip by **1-2 orders of magnitude**

Amplifies Read Disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**
- In extreme cases, activating a row **only once** induces bitflips

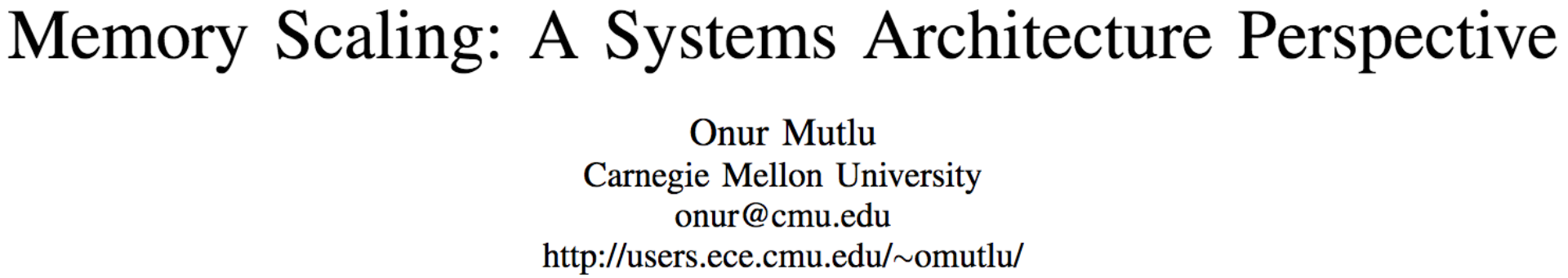


The Push from Circuits and Devices

Main Memory Needs Intelligent Controllers

An "Early" Position Paper [IMW 2013]

- Onur Mutlu,
  "Memory Scaling: A Systems Architecture Perspective"
  *Proceedings of the 5th International Memory Workshop (**IMW**)*, Monterey, CA, May 2013. Slides (pptx) (pdf)
  EETimes Reprint



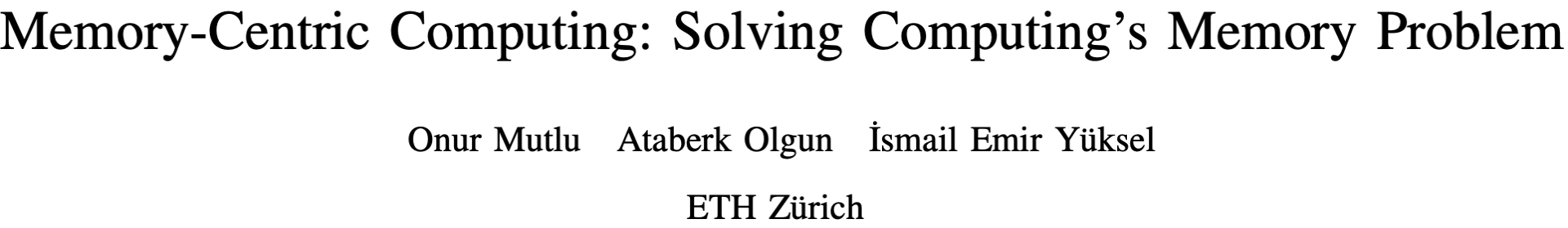
Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/

https://people.inf.ethz.ch/omutlu/pub/memory-scaling_memcon13.pdf

Updated Paper 12 Years Later [IMW 2025]

- Onur Mutlu, Ataberk Olgun, and İsmail Emir Yüksel, **"Memory-Centric Computing: Solving Computing's Memory Problem"**
  *Invited Paper in Proceedings of the 17th IEEE International Memory Workshop (**IMW**)*, Monterey, CA, USA, May 2025.
  [Slides (pptx) (pdf)]



Memory-Centric Computing: Solving Computing's Memory Problem

Onur Mutlu Ataberk Olgun İsmail Emir Yüksel

ETH Zürich

https://www.arxiv.org/pdf/2505.00458

Industry's Intelligent DRAM Controllers (I)

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

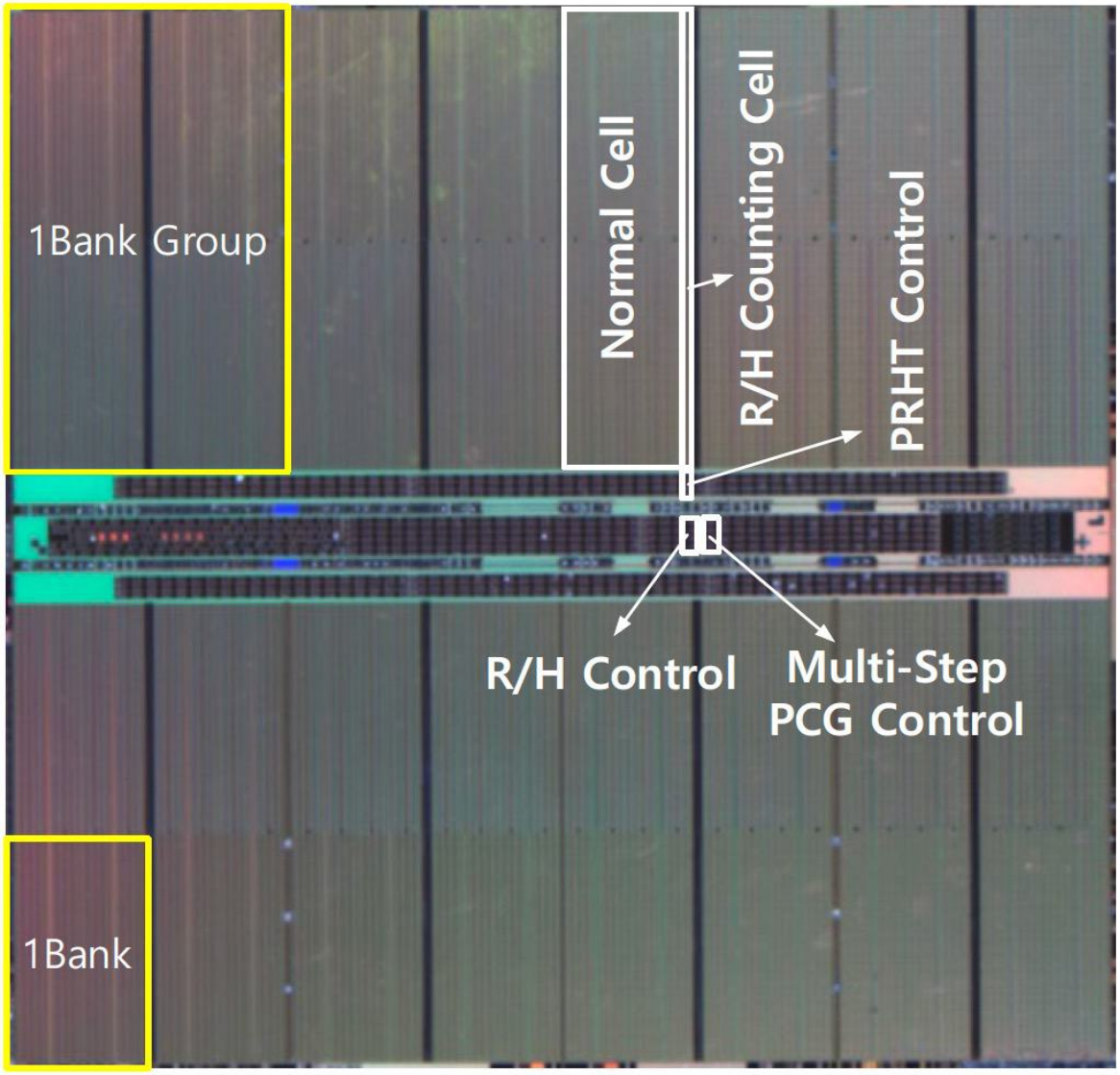
SK hynix Semiconductor, Icheon, Korea

Industry's Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's Intelligent DRAM Controllers (III)



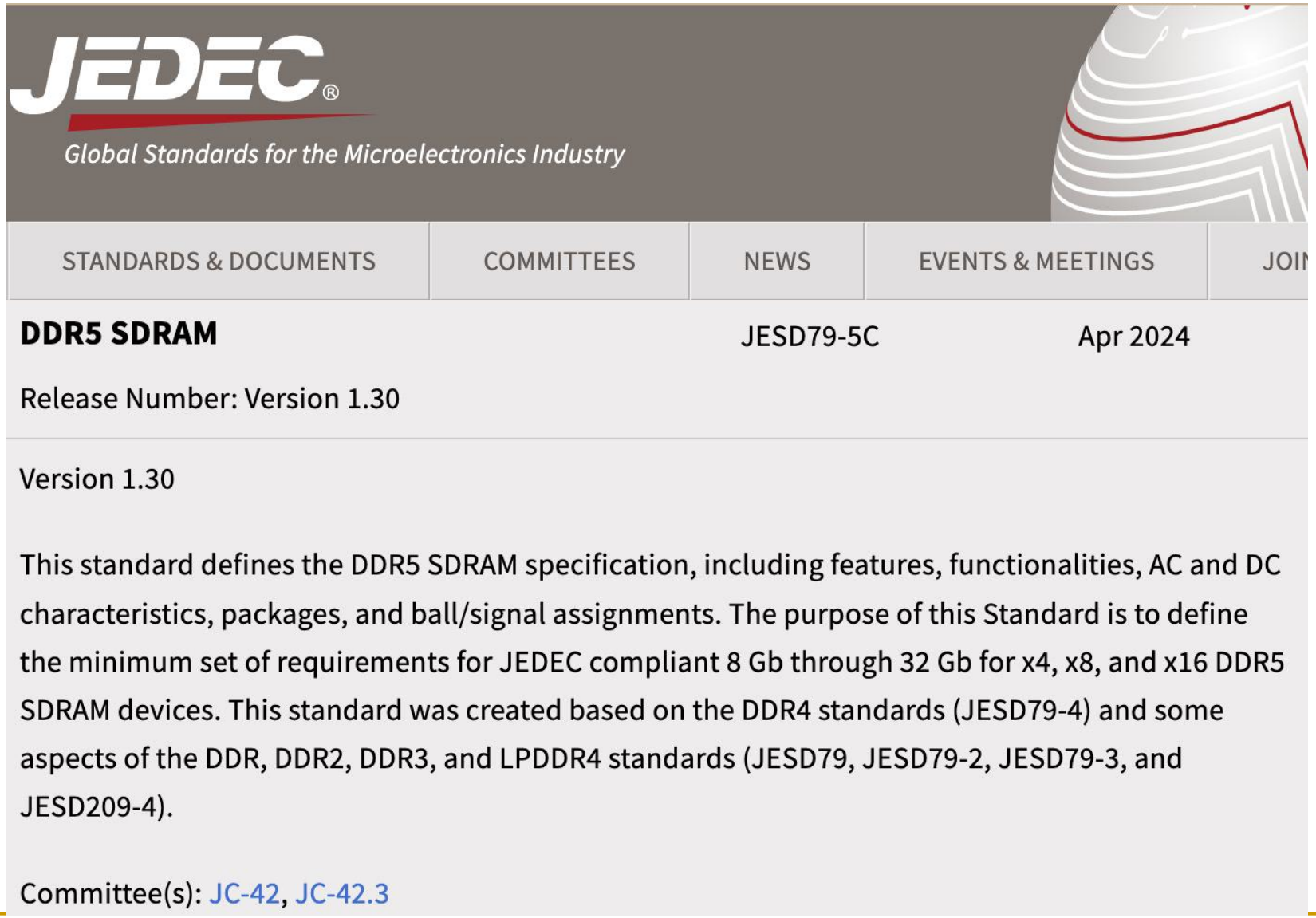
ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

Finally: Serious Changes in Standards (2024)



Are Solutions Good?

https://www.reddit.com/r/CrappyDesign/comments/arw0q8/now_this_this_is_poor_fencing/

Evaluation of Industry's Recent Solutions

- **Appears at DRAMSec 2024**

Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

https://arxiv.org/pdf/2406.19094

https://github.com/CMU-SAFARI/ramulator2

Evaluation of Industry's Recent Solutions

- Oguzhan Canpolat, Abdullah Giray Yaglikci, Geraldo Francisco de Oliveira, Ataberk Olgun, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Oguz Ergin, and Onur Mutlu,
  "Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance"
  *Proceedings of the 31st International Symposium on High-Performance Computer Architecture (**HPCA**)*, Las Vegas, NV, USA, March 2025.
  [Chronus Source Code (Officially Artifact Evaluated with All Badges)]
  Officially artifact evaluated as available, functional, and reproduced.

2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)

Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Nisa Bostancı[§] Ismail Emir Yuksel[§] Haocong Luo[§] Oğuz Ergin[‡†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [‡]*University of Sharjah*

https://arxiv.org/pdf/2502.12650

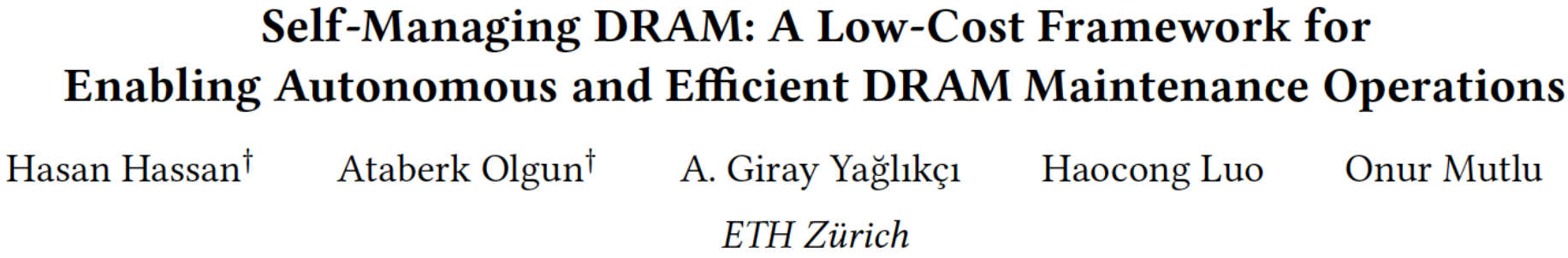
https://github.com/CMU-SAFARI/Chronus

Asking the Right Question

Can We Do Better?

Better Partitioning of DRAM & Controller

- Hasan Hassan, Ataberk Olgun, A. Giray Yaglikci, Haocong Luo, and Onur Mutlu,
  "Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations"
  *Proceedings of the 57th International Symposium on Microarchitecture (**MICRO**)*, Austin, TX, USA, November 2024.
  [Slides (pptx) (pdf)]
  [SelfManagingDRAM Source Code]



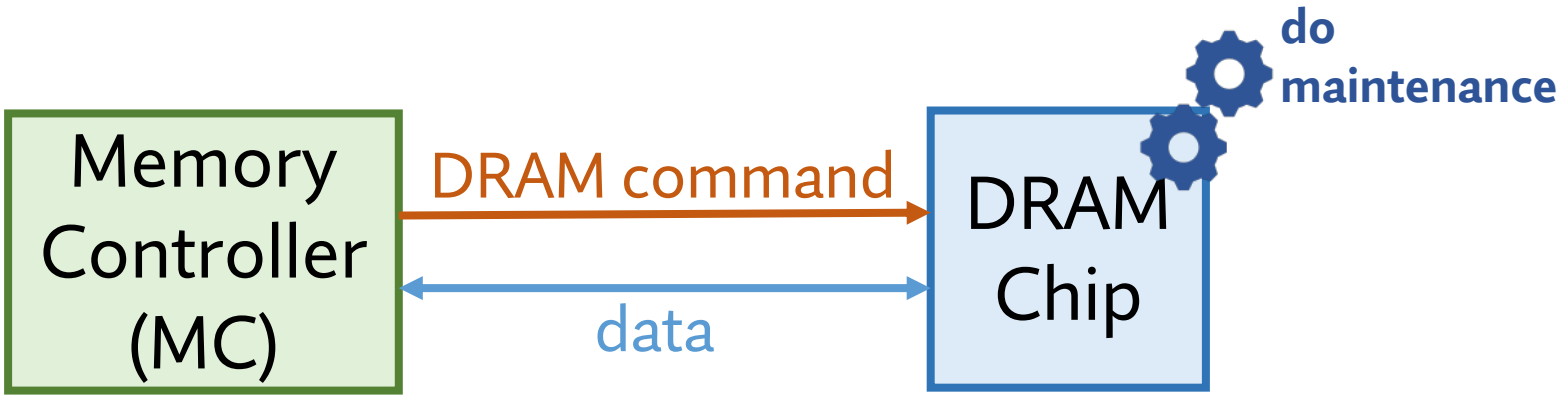
Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations

Hasan Hassan† Ataberk Olgun† A. Giray Yağlıkçı Haocong Luo Onur Mutlu

ETH Zürich

SMD Key Idea: Autonomous Maintenance

DRAM chip controls in-DRAM maintenance operations

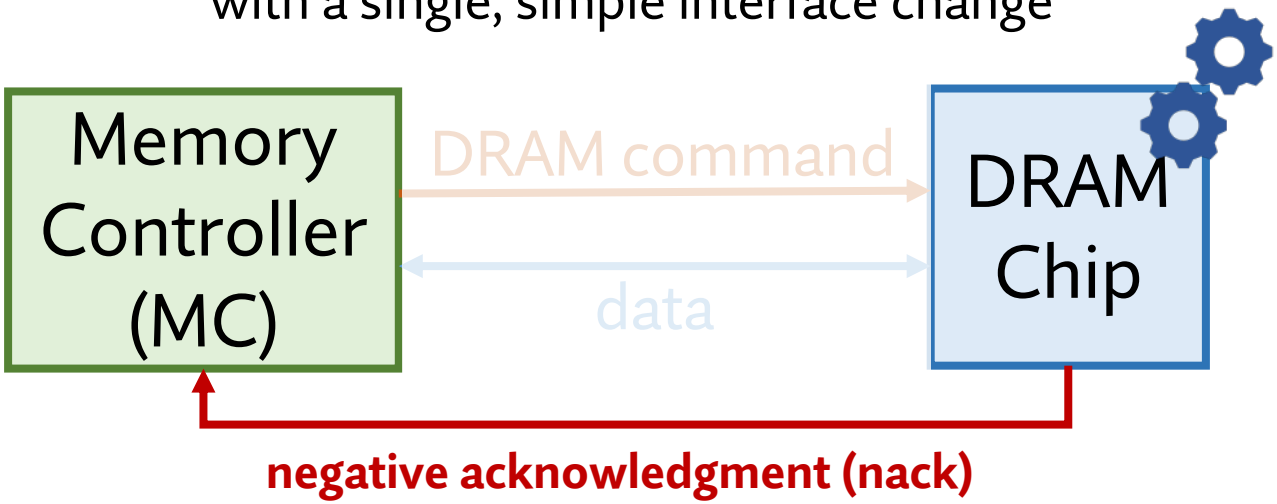


Enable implementing **new maintenance mechanisms** **without** modifying the standard and exposing **DRAM-internal proprietary** information

SMD Key Contribution

DRAM chip controls in-DRAM maintenance operations

with a single, simple interface change



orchestrates all access operations

performs its own maintenance autonomously

Partition the work nicely between the memory controller and the DRAM chip

SMD-Based Maintenance Mechanisms

DRAM Refresh

Fixed Rate (SMD-FR)

uniformly *refreshes* ***all*** *DRAM rows with a* ***fixed*** *refresh period*

Variable Rate (SMD-VR)

skips *refreshing rows that can* ***retain their data for longer*** *than the default refresh period*

RowHammer Protection

Probabilistic (SMD-PRP)

Performs ***neighbor row refresh*** *with* ***a small probability*** *on every row activation*

Deterministic (SMD-DRP)

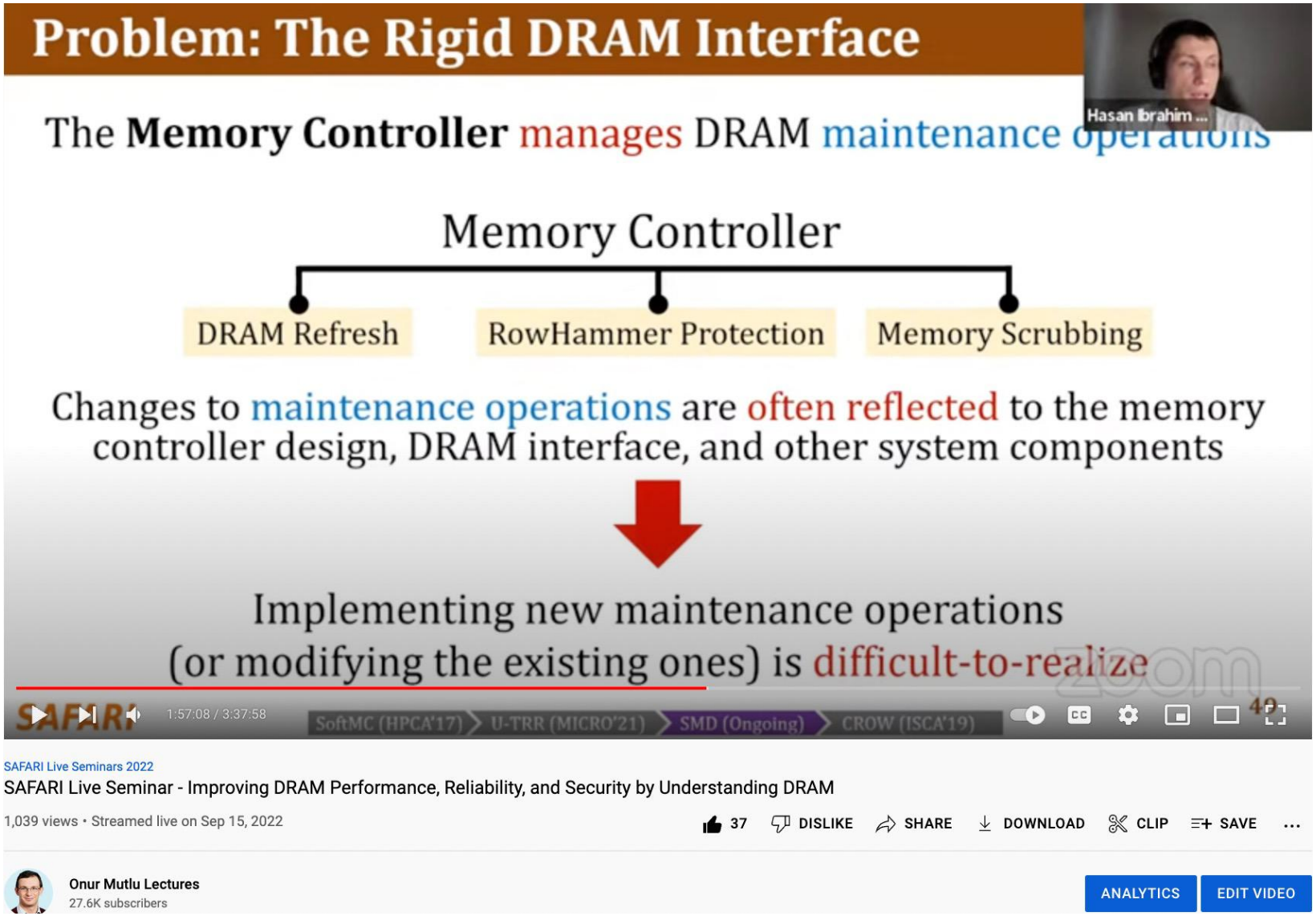
keeps track *of most* ***frequently activated*** *rows and performs* ***neighbor*** *row refresh when activation count threshold is exceeded*

Memory Scrubbing

Periodic Scrubbing (SMD-MS)

periodically ***scans*** *the* ***entire*** *DRAM for errors and corrects them*

Talk on Self-Managing DRAM



https://www.youtube.com/watch?v=mGa6-vpExbE

Self-Managing DRAM

- Hasan Hassan, Ataberk Olgun, A. Giray Yaglikci, Haocong Luo, and Onur Mutlu, **"Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations"** *Proceedings of the 57th International Symposium on Microarchitecture (**MICRO**)*, Austin, TX, USA, November 2024.
  [Slides (pptx) (pdf)]
  [SelfManagingDRAM Source Code]

Self-Managing DRAM: A Low-Cost Framework for Enabling Autonomous and Efficient DRAM Maintenance Operations

Hasan Hassan† Ataberk Olgun† A. Giray Yağlıkçı Haocong Luo Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2207.13358

Self-Managing DRAM: Brief History

Rejected 6 times:
2x MICRO, 2x ISCA, 2x HPCA

Acknowledgments

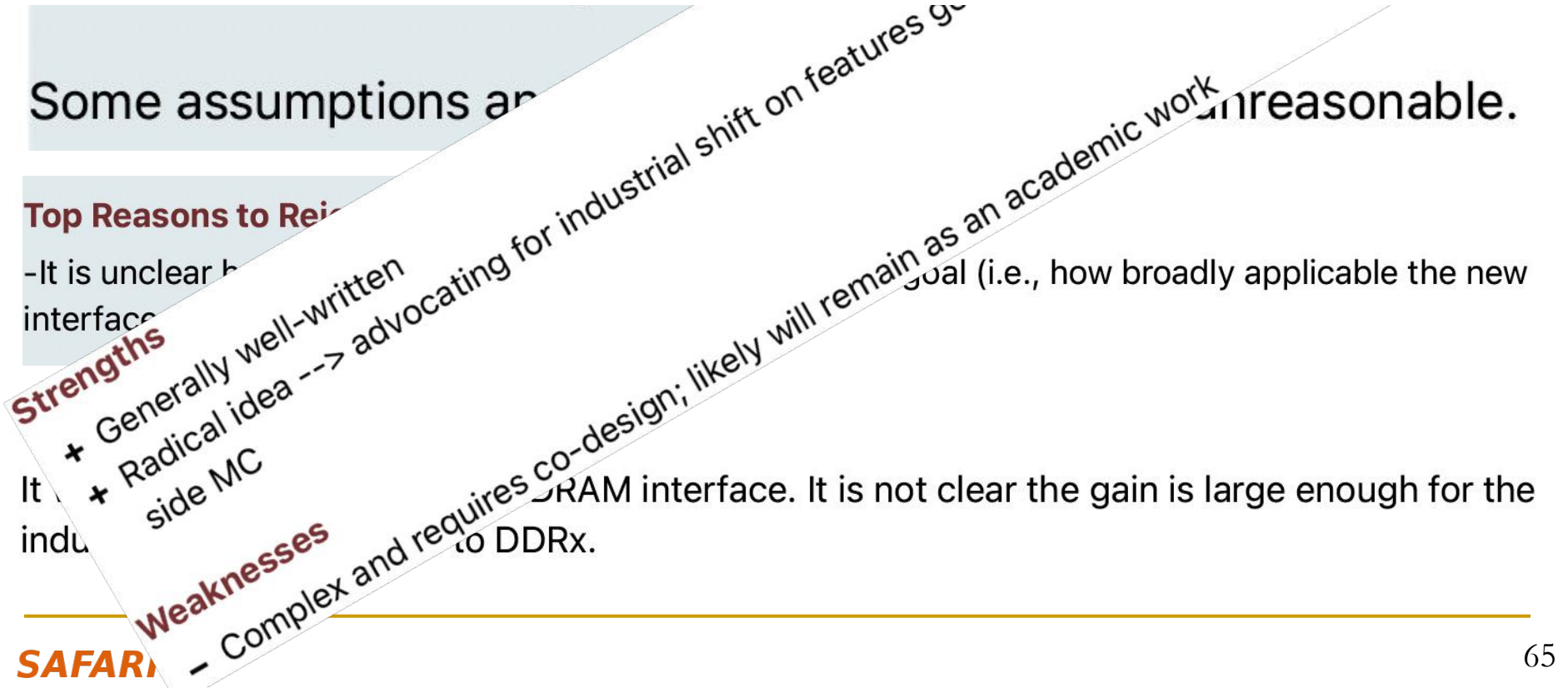
We thank the anonymous reviewers of MICRO 2022, HPCA 2023, ISCA 2023, MICRO 2023, HPCA 2024, ISCA 2024, and MICRO 2024 for the feedback. We thank the SAFARI Research Group members for their valuable and constructive feedback along with the stimulating scientific and intellectual environment they provide. We acknowledge the generous gift funding

https://arxiv.org/pdf/2207.13358

Are Reviewers Asking the Right Questions?

Rejected 6 times:

2x MICRO, 2x ISCA, 2x HPCA



Research Oriented Reviewers Also Exist

Strengths

- This is a good idea and a good way to introduce a layer of abstraction between DRAM and MC

Weaknesses

- There are some practical problems: for instance, it would be difficult to adopt in pin-limited use cases (like DDR). There also may be clock drift between DRAMs that needs to be accounted for. But these are not fundamental.

Accepted after 3.5 years

Approach & Takeaway

To Truly Solve Fundamental Problems,

Defy Business as Usual

Ask: What would a smart 10-year-old kid do?

Approach & Takeaway, Rephrased

Do Not Be Shackled
by Standards & Reviewers
(Business as Usual)

Ask: What could be the right (or a potentially better) thing to do?

Suggestion: Litmus Test

Our Litmus Test Should be to Efficiently Advance Scientific Endeavor

Ask: Is my review scientific? Does it rely on a crystal ball?

More in My MICRO 2025 Keynote Talk

Can We Do Better? - Keynote Talk at MICRO 2025 - Prof. Onur Mutlu

Onur Mutlu Lectures
58.5K subscribers

72 | Share | Save | Clip | Download

2,752 views Streamed live on Oct 21, 2025
Title: Can We Do Better?
Presenter: Prof. Onur Mutlu (https://people.inf.ethz.ch/omutlu/)
Date and Time: October 21, 2025, 08:00 AM (KST)

https://www.youtube.com/watch?v=Mk6j9zP3s9E

More to Come…

Can We Do Better?

There Is Still

a Whole Lot

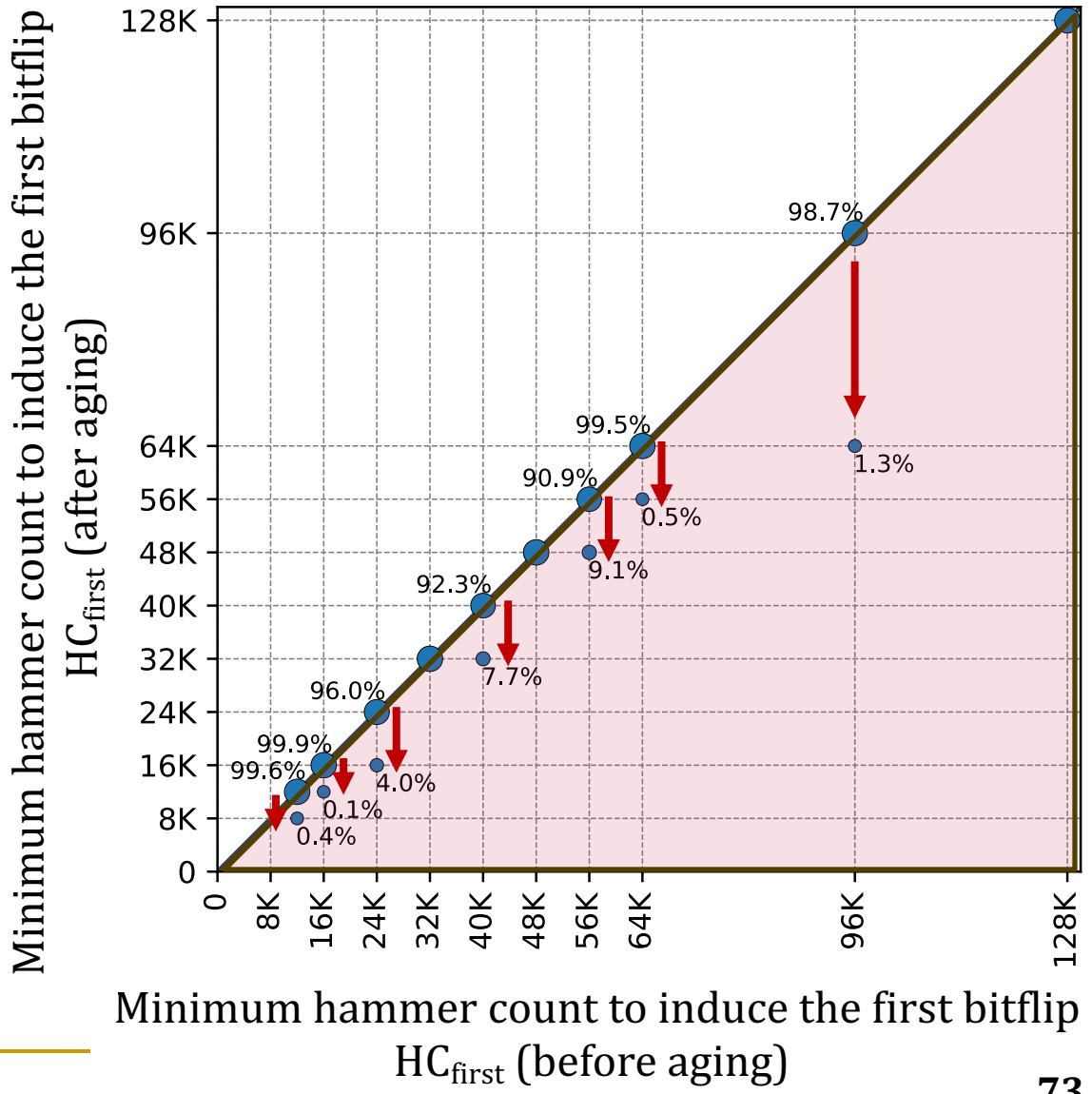
We Do Not Know

About DRAM

RowHammer Becomes Worse with Aging

Preliminary data on aging via 68-day of continuous hammering

Aging can lead to read disturbance bitflips at **smaller** hammer counts



RowHammer (Spatial Variation) Analysis (2024)

- **Appears at HPCA 2024**

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı Yahya Can Tuğrul Geraldo F. Oliveira
İsmail Emir Yüksel Ataberk Olgun Haocong Luo Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18652

Variable Read Disturbance (2025)

Key Takeaway

The Read Disturbance Threshold (RDT) of a row changes randomly and unpredictably over time

Accurately identifying RDT is challenging

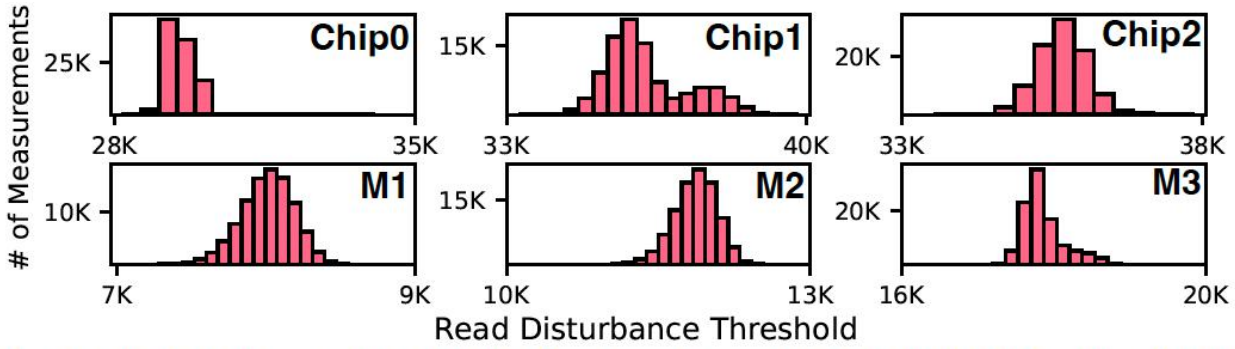


Fig. 2: **Read disturbance threshold of a row in each tested HBM2 chip (Chip0-2) and DDR4 module (M1-3) over 100,000 repeated measurements.** Adapted from [43].

https://arxiv.org/pdf/2406.13080v2

Variable Read Disturbance (2025)

- **Appears at HPCA 2025**

Variable Read Disturbance:
An Experimental Analysis of Temporal Variation in DRAM Read Disturbance

Ataberk Olgun† F. Nisa Bostancı† İsmail Emir Yüksel† Oğuzhan Canpolat† Haocong Luo†
Geraldo F. Oliveira† A. Giray Yağlıkçı† Minesh Patel‡ Onur Mutlu†
ETH Zurich† *Rutgers University‡*

Read disturbance threshold varies unpredictably over time

https://arxiv.org/pdf/2502.13075

ColumnDisturb (2025)

- **Appears at MICRO 2025**

ColumnDisturb: Understanding Column-based Read Disturbance in Real DRAM Chips and Implications for Future Systems

İsmail Emir Yüksel[1] Ataberk Olgun[1] F. Nisa Bostancı[1]
Haocong Luo[1] A. Giray Yağlıkçı[1,2] Onur Mutlu[1]
[1]ETH Zürich [2]CISPA

A single row activation flips bits across three DRAM subarrays (up to 3072 DRAM rows)

https://arxiv.org/pdf/2510.14750

PuDHammer (2025)

- **Appears at ISCA 2025**

PuDHammer: Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

İsmail Emir Yüksel Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo
F. Nisa Bostancı Mohammad Sadrosadati A. Giray Yağlıkçı Onur Mutlu

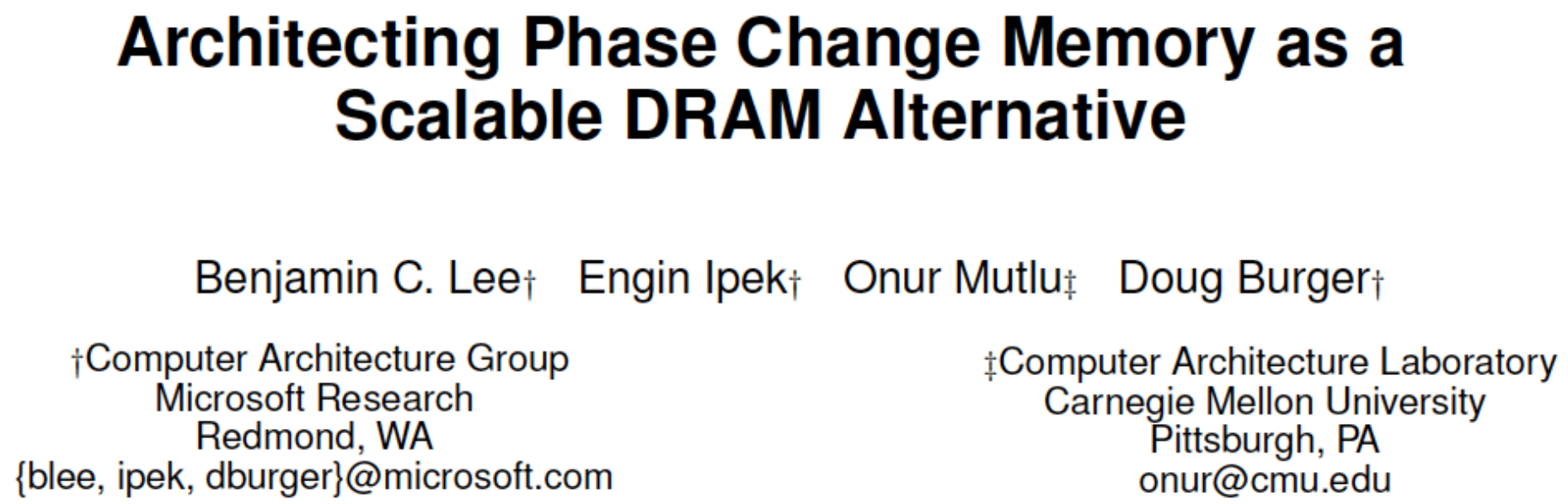
ETH Zürich

Read disturbance becomes much worse with processing using DRAM

https://arxiv.org/pdf/2506.12947

Emerging Memories Also Need Intelligent Controllers

- 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)
  One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight.
  2022 Persistent Impact Prize.



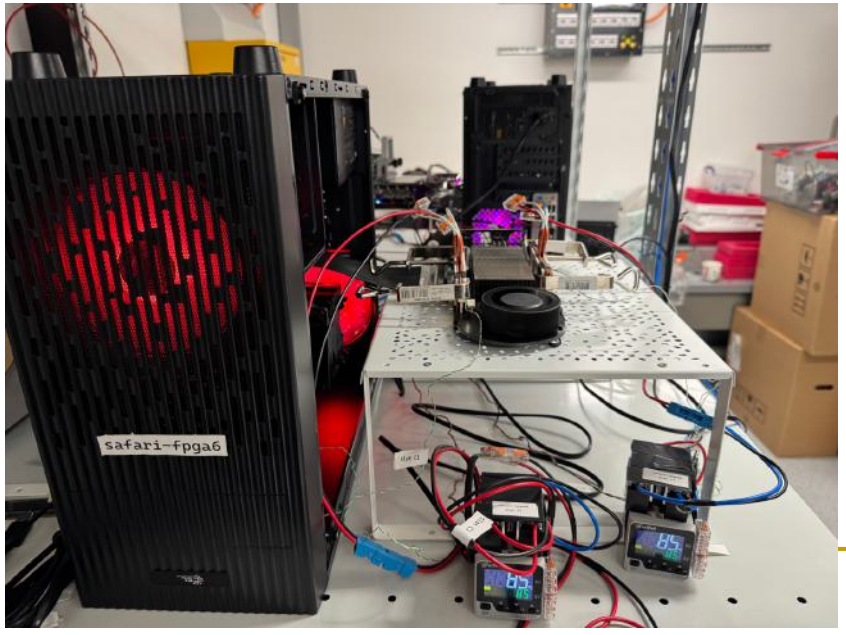
Architecting Phase Change Memory as a Scalable DRAM Alternative

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

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

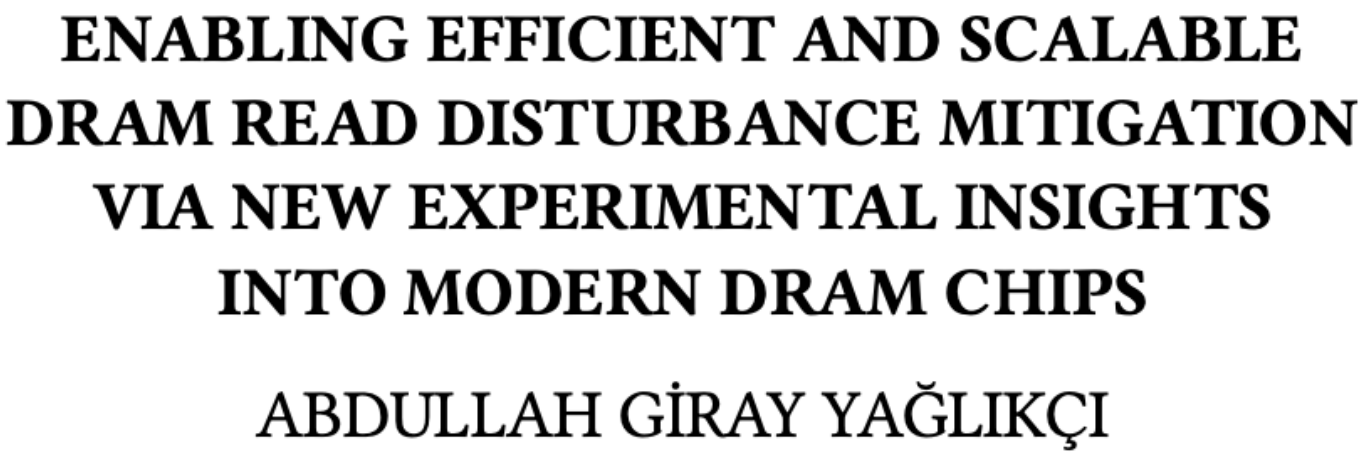
‡Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

Laboratory for Understanding Memory



A Very Recent PhD Thesis

- A. Giray Yaglikci, **"Enabling Efficient and Scalable DRAM Read Disturbance Mitigation via New Experimental Insights into Modern DRAM Chips,"** PhD Thesis, ETH Zürich, 2024.
  [Slides (pdf) (pptx)]
  [Thesis arXiv (abs) (pdf)]
  [SAFARI News]



https://arxiv.org/pdf/2408.15044.pdf

Read Disturbance Sessions @ HPCA 2025

HPCA 2025

2025 IEEE International Symposium on High-Performance Computer Architecture, 3/1/2025-3/5/2025, Las Vegas, NV, USA

Session 7A (*Acacia A and B*): Hammering the Odds – 1

Session Chair: *Gururaj Saileshwar (Toronto)*

- **Variable Read Disturbance: An Experimental Analysis of Temporal Variation in DRAM Read Disturbance**
  Ataberk Olgun (ETH Zürich), Nisa Bostanci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Giray Yaglikci (ETH Zürich), Geraldo F. Oliveira (ETH Zürich), Haocong Luo (ETH Zürich), Oguzhan Canpolat (ETH Zürich), Minesh Patel (Rutgers University), Onur Mutlu (ETH Zürich)
- **Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions**
  Yahya Can Tuğrul (TOBB ETÜ & ETH Zürich), Giray Yaglikci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Ataberk Olgun (ETH Zürich), Oğuzhan Canpolat (TOBB ETÜ & ETH Zürich), Nisa Bostanci (ETH Zürich), Mohammad Sadrosadati (ETH Zürich), Oguz Ergin (TOBB ETÜ), Onur Mutlu (ETH Zürich)
- **Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance**
  Oğuzhan Canpolat (TOBB ETÜ & ETH Zürich), Giray Yaglikci (ETH Zürich), Geraldo Francisco de Oliveira (ETH Zürich), Ataberk Olgun (ETH Zürich), Nisa Bostanci (ETH Zürich), Ismail Emir Yuksel (ETH Zürich), Haocong Luo (ETH Zürich), Oğuz Ergin (TOBB ETÜ), Onur Mutlu (ETH Zürich)

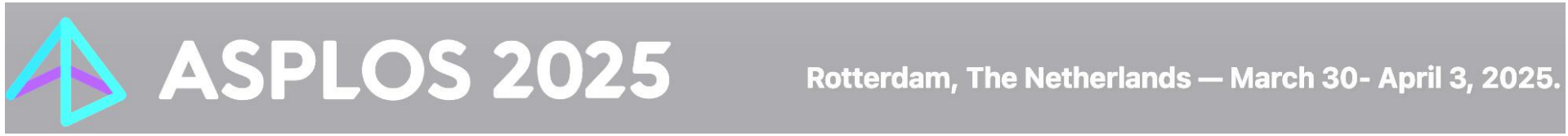
Session 8A (*Acacia A and B*): Hammering the Odds – 2

Session Chair: *Sudhanva Gurumurthi (AMD)*

- **AutoRFM: Scaling Low-Cost In-DRAM Trackers to Ultra-Low Rowhammer Thresholds**
  Moinuddin Qureshi (Georgia Tech)
- **DAPPER: A Performance-Attack-Resilient Tracker for RowHammer Defense**
  Jeonghyun Woo (The University of British Columbia (UBC)), Prashant J. Nair (The University of British Columbia (UBC))
- **QPRAC: Towards Secure and Practical PRAC-based Rowhammer Mitigation using Priority Queues**
  Jeonghyun Woo (The University of British Columbia (UBC)), Shaopeng (Chris) Lin (University of Toronto), Prashant J. Nair (The University of British Columbia (UBC)), Aamer Jaleel (NVIDIA), Gururaj Saileshwar (University of Toronto)

Tuesday, March 4th, 11am and 2pm

Read Disturbance Papers @ ASPLOS 2025



Session 4B: Memory & Storage +

LOCATION: VAN OLDENBARNEVELD

Marionette: A RowHammer Attack via Row Coupling

Seungmin Baek (Seoul National University), Minbok Wi (Seoul National University), Seonyong Park (Seoul National University), Hwayong Nam (Seoul National University), Michael Jaemin Kim (Seoul National University), Nam Sung Kim (University of Illinois), Jung Ho Ahn (Seoul National University)

Paper

MOAT: Securely Mitigating Rowhammer with Per-Row Activation Counters

Moinuddin Qureshi (Georgia Institute of Technology), Salman Qazi (Google)

Paper

HyperHammer: Breaking Free from KVM-Enforced Isolation

Wei Chen (Peking University), Zhi Zhang (University of Western Australia), Xin Zhang (Peking University), Qingni Shen (Peking University), Yuval Yarom (Ruhr University Bochum), Daniel Genkin (Georgia Institute of Technology), Chen Yan (Peking University), Zhe Wang (SKLP, Institute of Computing Technology, Chinese Academy of Sciences,Zhongguancun Laboratory)

Paper

Read Disturbance Session @ ISCA 2025

Session 5A: RowHammer

Location: Okuma Auditorium (Main)

Session Chair: TBA

08:30 AM – 08:50 AM

MoPAC: Efficiently Mitigating Rowhammer with Probabilistic Activation Counting

Suhas Vittal, Salman Qazi, Poulami Das, Moin Qureshi

08:50 AM – 09:10 AM

When Mitigations Backfire: Timing Channel Attacks and Defense for PRAC-Based Rowhammer Mitigations

Jeonghyun Woo, Joyce Qu, Gururaj Saileshwar, Prashant Nair

09:10 AM – 09:30 AM

PuDHammer: Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

Ismail Emir Yuksel, Akash Sood, Ataberk Olgun, O?uzhan Canpolat, Haocong Luo, Nisa Bostanci, Mohammad Sadrosadati, Giray Yaglikci, Onur Mutlu

09:30 AM – 09:50 AM

DREAM: Enabling Low-Overhead Rowhammer Mitigation via Directed Refresh Management

Hritvik Taneja, Moin Qureshi

Read Disturbance Papers @ DRAMSec 2025

Accepted papers

Softhammer: Exploiting Rowhammer Bit Flips without Crashing
Finn de Ridder, Patrick Jattke, Kaveh Razavi

Rubber Mallet: A Study of High Frequency Localized Bit Flips and Their Impact on Security
Andrew J. Adiletta, Zane Weissman, Fatemeh Khojasteh Dana, Berk Sunar, Shahin Tajik

CnC-PRAC: Coalesce, not Cache, Per Row Activation Counts for an Efficient in-DRAM Rowhammer Mitigation
Chris S. Lin, Jeonghyun Woo, Prashant J. Nair, Gururaj Saileshwar

A Simulation-based Evaluation Framework for Inter-VM RowHammer Mitigation Techniques
Hidemasa Kawasaki, Soramichi Akiyama

Sudoku: Decomposing DRAM Address Mapping into Component Functions
Minbok Wi, Seungmin Baek, Seonyong Park, Mattan Erez, Jung Ho Ahn

Counterpoint: One-Hot Counting for PRAC-Based RowHammer Mitigation
Shih-Lien Lu, Jeonghyun Woo, Prashant J. Nair

DRFM and the Art of Rowhammer Sampling
Salman Qazi, Moinuddin Qureshi

Keynote

Panel

Is PRAC a good solution to DRAM read disturbance? Are we missing anything? Can we (and should we) do much better (and hopefully not worse)?

Workshop chairs

- Onur Mutlu, ETH Zürich
- Kuljit Bains, NVIDIA

https://dramsec.ethz.ch/

https://www.youtube.com/watch?v=5KmKxFjPopM

RowHammer @ USENIX Security 2025

Rowhammer-Based Trojan Injection: One Bit Flip Is Sufficient for Backdooring DNNs

Xiang Li, Ying Meng, Junming Chen, Lannan Luo, and Qiang Zeng, *George Mason University*

Long Presentation

AVAILABLE MEDIA

GPUHammer: Rowhammer Attacks on GPU Memories are Practical

Chris S. Lin, Joyce Qu, and Gururaj Saileshwar, *University of Toronto*

Short Presentation

AVAILABLE MEDIA

Posthammer: Pervasive Browser-based Rowhammer Attacks with Postponed Refresh Commands

Finn de Ridder, Patrick Jattke, and Kaveh Razavi, *ETH Zurich*

Short Presentation

AVAILABLE MEDIA

Show details

ECC.fail: Mounting Rowhammer Attacks on DDR4 Servers with ECC Memory

Nureddin Kamadan and Walter Wang, *Georgia Tech;* Stephan van Schaik, *University of Michigan;* Christina Garman, *Purdue University;* Daniel Genkin, *Georgia Tech;* Yuval Yarom, *Ruhr University Bochum*

Short Presentation

AVAILABLE MEDIA

Achilles: A Formal Framework of Leaking Secrets from Signature Schemes via Rowhammer

Junkai Liang, *Peking University;* Zhi Zhang, *The University of Western Australia;* Xin Zhang and Qingni Shen, *Peking University;* Yansong Gao, *The University of Western Australia;* Xingliang Yuan, *The University of Melbourne;* Haiyang Xue and Pengfei Wu, *Singapore Management University;* Zhonghai Wu, *Peking University*

Long Presentation

McSee: Evaluating Advanced Rowhammer Attacks and Defenses via Automated DRAM Traffic Analysis

Patrick Jattke and Michele Marazzi, *ETH Zurich;* Flavien Solt, *UC Berkeley;* Max Wipfli, Stefan Gloor, and Kaveh Razavi, *ETH Zurich*

Long Presentation

AVAILABLE MEDIA

Show details

Not so Refreshing: Attacking GPUs using RFM Rowhammer Mitigation

Ravan Nazaraliyev and Yicheng Zhang, *University of California, Riverside;* Sankha Baran Dutta, *Brookhaven National Laboratory;* Andres Marquez and Kevin Barker, *Pacific Northwest National Laboratory;* Nael Abu-Ghazaleh, *University of California, Riverside*

Short Presentation

AVAILABLE MEDIA

Read Disturbance Papers @ MICRO 2025

Citadel: Rethinking Memory Allocation to Safeguard Against Inter-Domain Rowhammer Exploits

Anish Saxena, Walter Wang, Alexandros Daglis (Georgia Inst. of Technology)

ColumnDisturb: Understanding Column-based Read Disturbance in Real DRAM Chips and Implications for Future Systems

Ismail Emir Yuksel, Ataberk Olgun, Nisa Bostanci, Haocong Luo, Abdullah Giray Yaglikci, Onur Mutlu (ETH Zürich)

Understanding and Mitigating Covert and Side Channel Vulnerabilities Introduced by RowHammer Defenses

Nisa Bostanci (ETH Zürich); Oguzhan Canpolat (TOBB ETÜ and ETH Zurich); Ataberk Olgun, Ismail Emir Yuksel, Konstantinos Kanellopoulos, Mohammad Sadrosadati, Abdullah Giray Yaglikci, Onur Mutlu (ETH Zürich)

pHammer: Reviving RowHammer Attacks on New Architectures via Prefetching

Weijie Chen, Shan Tang, Yulin Tang (Huazhong Univ. of Science and Technology); Xiapu Luo (The Hong Kong Polytechnic Univ.); Yinqian Zhang (Southern Univ. of Science and Technology); Weizhong Qiang (Huazhong Univ. of Science and Technology)

Read Disturbance Papers @ HPCA 2026

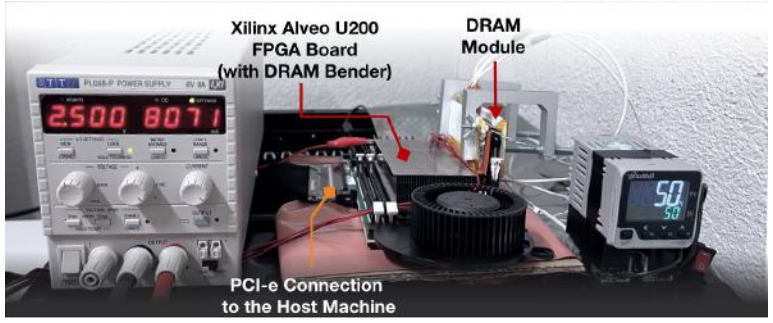
11:30 - 12:50 **DRAM Security and Reliability at Collaroy**

11:30	20m *Talk*	☆	**MIRZA: Efficiently Mitigating Rowhammer with Randomization and ALERT** Hritvik Taneja Georgia Tech, Ali Hajiabadi ETH Zurich, Michele Marazzi ABB Research, Kaveh Razavi ETH Zürich, Moinuddin K. Qureshi Georgia Tech
11:50	20m *Talk*	☆	**SALT: Track-and-Mitigate Subarrays, Not Rows, for Blast-Radius-Free Rowhammer Defense** Moinuddin K. Qureshi Georgia Tech
12:10	20m *Talk*	☆	**ReScue: Reliable and Secure CXL Memory** Chihun Song UIUC, Austin Antony Cruz UIUC, Michael Jaemin Kim Meta, Minbok Wi Seoul National University, Gaohan Ye UIUC, Kyungsan Kim Samsung Sung Kim UIUC

Tutorial on DRAM Bender & Ramulator

Tutorial on Ramulator & DRAM Bender:

Cutting-Edge Infrastructures for Real and Future Memory System Evaluation



R&DB:

A half-day tutorial @ ASPLOS 2026, Pittsburgh, USA
22nd March 2026 (Saturday)

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

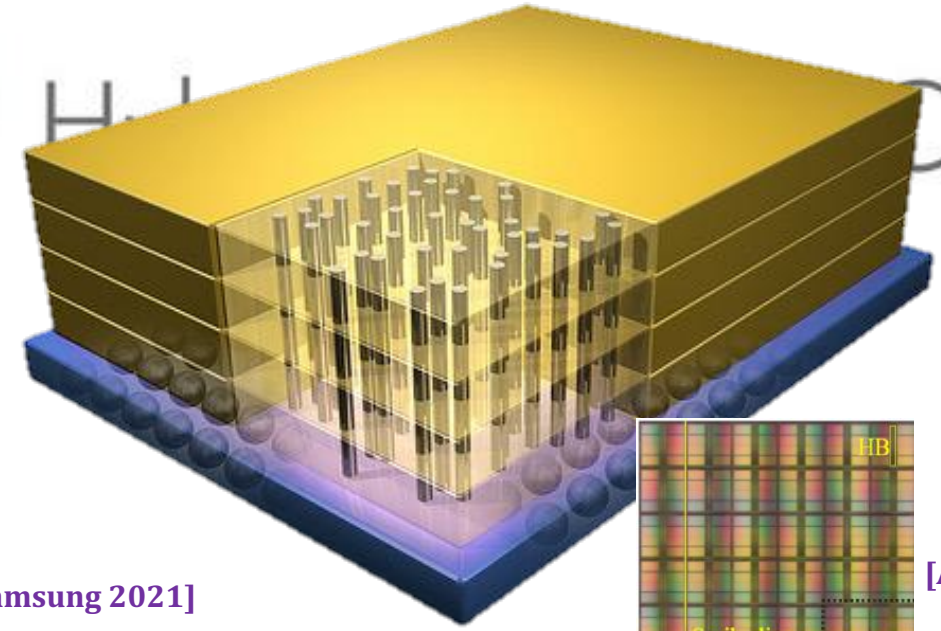
- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*

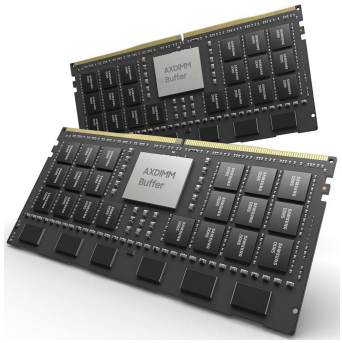
More information & call for presentation:
https://events.safari.ethz.ch/asplos26-ramulator-drambender/

Processing in Memory: Two Types

1. Processing **near** Memory
2. Processing **using** Memory

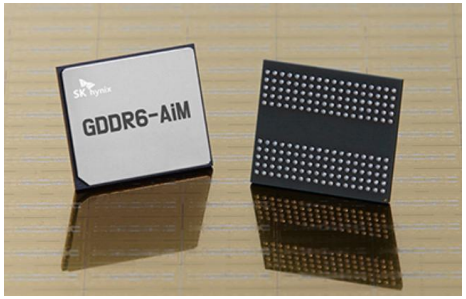
Processing-in-Memory Landscape Today



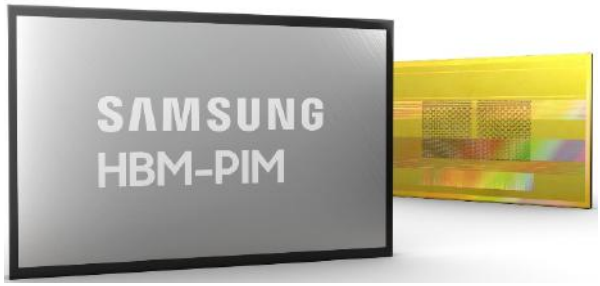


[Samsung 2021]

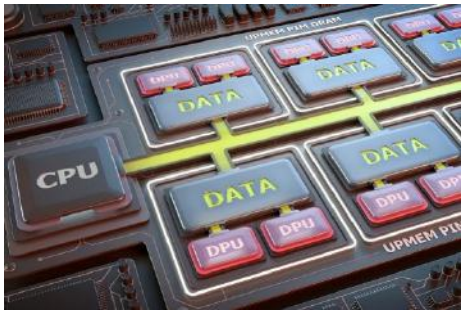
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

And, many other experimental chips and startups

Processing-in-Memory Landscape Today

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim, Soohong Ahn, Taeyoung Ahn, Seungyong Lee, Myunghyun Rhee, Jooyoung Kim, Kwangsik Shin, Donguk Moon, Euiseok Kim, and Kyoung Park

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to 1.9× better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

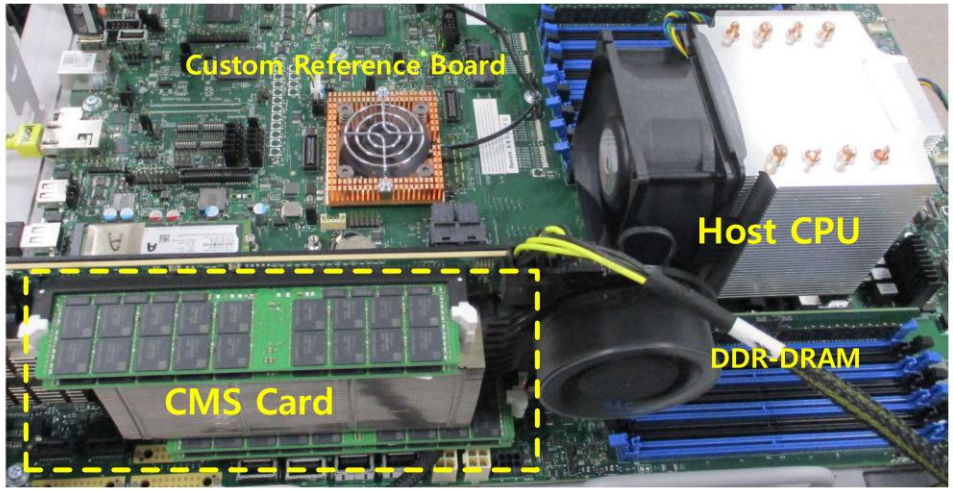


Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today

Samsung Processing in Memory Technology at Hot Chips 2023

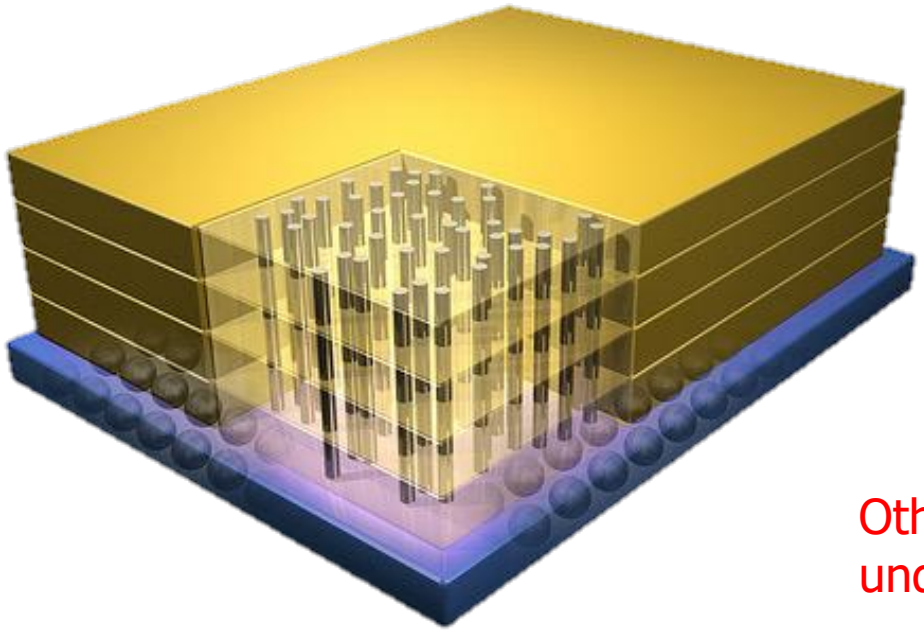
By **Patrick Kennedy** - August 28, 2023

Samsung PIM PNM For Transformer Based AI HC35_Page_24

https://www.servethehome.com/samsung-processing-in-memory-technology-at-hot-chips-2023/

Opportunity: 3D-Stacked Logic+Memory

Hybrid Memory Cube
CONSORTIUM



Memory

Logic

Other "True 3D" technologies under development

Tesseract: 3D-Stacked Processing Near Memory

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung 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 (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  Top Picks Honorable Mention by IEEE Micro.
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University [§]Oracle Labs [†]Carnegie Mellon University

A Short Retrospective @ 50 Years of ISCA

Retrospective: A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn[†] Sungpack Hong[‡] Sungjoo Yoo[▽] Onur Mutlu[§] Kiyoung Choi[▽]
[†]*Google DeepMind* [‡]*Oracle Labs* [§]*ETH Zürich* [▽]*Seoul National University*

***Abstract*—Our ISCA 2015 paper [1] provides a new programmable processing-in-memory (PIM) architecture and system design that can accelerate key data-intensive applications, with a focus on graph processing workloads. Our major idea was to completely rethink the system, including the programming model, data partitioning mechanisms, system support, instruction set architecture, along with near-memory execution units and their communication architecture, such that an important workload can be accelerated at a maximum level using a distributed system of well-connected near-memory accelerators. We built our accelerator system, Tesseract, using 3D-stacked memories with logic layers, where each logic layer contains general-purpose processing cores and cores communicate with each other using a message-passing programming model. Cores could be specialized for graph processing (or any other application to be accelerated).**

To our knowledge, our paper was the first to completely design a near-memory accelerator system from scratch such that it is both generally programmable and specifically customizable to accelerate important applications, with a case study on major graph processing workloads. Ensuing work in academia and industry showed that similar approaches to system design can greatly benefit both graph processing workloads and other applications, such as machine learning, for which ideas from Tesseract seem to have been influential.

This short retrospective provides a brief analysis of our ISCA 2015 paper and its impact. We briefly describe the major ideas and contributions of the work, discuss later works that built on it or were influenced by it, and make some educated guesses on what the future may bring on PIM and accelerator systems.

I. Background, Approach & Mindset

We started our research when 3D-stacked memories (e.g., [2–4]) were viable and seemed to have promise for building effective and practical processing-near-memory systems. Such near-memory systems could lead to improvements, but there was little to no research that examined how an accelerator could be completely (re-)designed using such near-memory technology, from its hardware architecture to its programming model and software system, and what the performance and energy benefits could be of such a re-design. We set out to answer these questions in our ISCA 2015 paper [1].

We followed several major principles to design our accelerator from the ground up. We believe these principles are still important: a major contribution and influence of our work was in putting all of these together in a cohesive full-system design and demonstrating the large performance and energy benefits that can be obtained from such a design. We see a similar approach in many modern large-scale accelerator systems in machine learning today (e.g., [5–9]). Our principles are:

1. *Near-memory execution* to enable/exploit the high data access bandwidth modern workloads (e.g., graph processing) need and to reduce data movement and access latency.
2. *General programmability* so that the system can be easily adopted, extended, and customized for many workloads.
3. *Maximal acceleration capability* to maximize the performance and energy benefits. We set ourselves free from backward compatibility and cost constraints. We aimed to completely re-design the system stack. Our goal was to explore the maximal performance and energy efficiency benefits we can gain from a near-memory accelerator if we had complete freedom to change things as much as we needed. We contrast this approach to the *minimal intrusion* approach we also explored in a separate ISCA 2015 paper [10].
4. *Customizable to specific workloads*, such that we can maximize acceleration benefits. Our focus workload was graph analytics/processing, a key workload at the time and today. However, our design principles are not limited to graph processing and the system we built is customizable to other workloads as well, e.g., machine learning, genome analysis.
5. *Memory-capacity-proportional performance*, i.e., processing capability should proportionally grow (i.e., scale) as memory capacity increases and vice versa. This enables scaling of data-intensive workloads that need both memory and compute.
6. *Exploit new technology (3D stacking)* that enables tight integration of memory and logic and helps multiple above principles (e.g., enables customizable near-memory acceleration capability in the logic layer of a 3D-stacked memory chip).
7. *Good communication and scaling capability* to support scalability to large dataset sizes and to enable memory-capacity-proportional performance. To this end, we provided scalable communication mechanisms between execution cores and carefully interconnected small accelerator chips to form a large distributed system of accelerator chips.
8. *Maximal and efficient use of memory bandwidth* to supply the high-bandwidth data access that modern workloads need. To this end, we introduced new, specialized mechanisms for prefetching and a programming model that helps leverage application semantics for hardware optimization.

II. Contributions and Influence

We believe the major contributions of our work were 1) complete rethinking of how an accelerator system should be designed to enable maximal acceleration capability, and 2) the design and analysis of such an accelerator with this mindset and using the aforementioned principles to demonstrate its effectiveness in an important class of workloads.

One can find examples of our approach in modern large-scale machine learning (ML) accelerators, which are perhaps the most successful incarnation of scalable near-memory execution architectures. ML infrastructure today (e.g., [5–9]) consists of accelerator chips, each containing compute units and high-bandwidth memory tightly packaged together, and features scale-up capability enabled by connecting thousands of such chips with high-bandwidth interconnection links. The system-wide rethinking that was done to enable such accelerators and many of the principles used in such accelerators resemble our ISCA 2015 paper's approach.

The "memory-capacity-proportional performance" principle we explored in the paper shares similarities with how ML workloads are scaled up today. Similar to how we carefully sharded graphs across our accelerator chips to greatly improve effective memory bandwidth in our paper, today's ML workloads are sharded across a large number of accelerators by leveraging data/model parallelism and optimizing the placement to balance communication overheads and compute scalability [11, 12]. With the advent of large generative models requiring high memory bandwidth for fast training and inference, the scaling behavior where capacity and bandwidth are scaled together has become an essential architectural property to support modern data-intensive workloads.

The "maximal acceleration capability" principle we used in Tesseract provides much larger performance and energy improvements and better customization than the "minimalist" approach that our other ISCA 2015 paper on *PIM-Enabled Instructions* [10] explored: "minimally change" an existing system to incorporate (near-memory) acceleration capability to ease programming and keep costs low. So far, the industry has more widely adopted the maximal approach to overcome the pressing scaling bottlenecks of major workloads. The key enabler that bridges the programmability gap between the maximal approach favoring large performance & energy benefits and the minimal approach favoring ease of programming is compilation techniques. These techniques lower well-defined high-level constructs into lower-level primitives [12, 13]; our ISCA 2015 papers [1, 10] and a follow-up work [14] explore them lightly. We believe that a good programming model that enables large benefits coupled with support for it across the entire system stack (including compilers & hardware) will continue to be important for effective near-memory system and accelerator designs [14]. We also believe that the maximal versus minimal approaches that are initially explored in our two ISCA 2015 papers is a useful way of exploring emerging technologies (e.g., near-memory accelerators) to better understand the tradeoffs of system designs that exploit such technologies.

III. Influence on Later Works

Our paper was at the beginning of a proliferation of scalable near-memory processing systems designed to accelerate key applications (see [15] for many works on the topic). Tesseract has inspired many near-memory system ideas (e.g., [16–28]) and served as the de facto comparison point for such systems, including near-memory graph processing accelerators that built on Tesseract and improved various aspects of Tesseract. Since machine learning accelerators that use high-bandwidth memory (e.g., [5, 29]) and industrial PIM prototypes (e.g., [30–41]) are now in the market, near-memory processing is no longer an "eccentric" architecture it used to be when Tesseract was originally published.

Graph processing & analytics workloads remain as an important and growing class of applications in various forms, ranging from large-scale industrial graph analysis engines (e.g., [42]) to graph neural networks [43]. Our focus on large-scale graph processing in our ISCA 2015 paper increased attention to this domain in the computer architecture community, resulting in subsequent research on efficient hardware architectures for graph processing (e.g., [44–46]).

IV. Summary and Future Outlook

We believe that our ISCA 2015 paper's principled rethinking of system design to accelerate an important class of data-intensive workloads provided significant value and enabled/influenced a large body of follow-on works and ideas. We expect that such rethinking of system design for key workloads, especially with a focus on "maximal acceleration capability," will continue to be critical as pressing technology and application scaling challenges increasingly require us to think differently to substantially improve performance and energy (as well as other metrics). We believe the principles exploited in Tesseract are fundamental and they will remain useful and likely become even more important as systems become more constrained due to the continuously-increasing memory access and computation demands of future workloads. We also project that as hardware substrates for near-memory acceleration (e.g., 3D stacking, in-DRAM computation, NVM-based PIM, processing using memory [15]) evolve and mature, systems will take advantage of them even more, likely using principles similar to those used in the design of Tesseract.

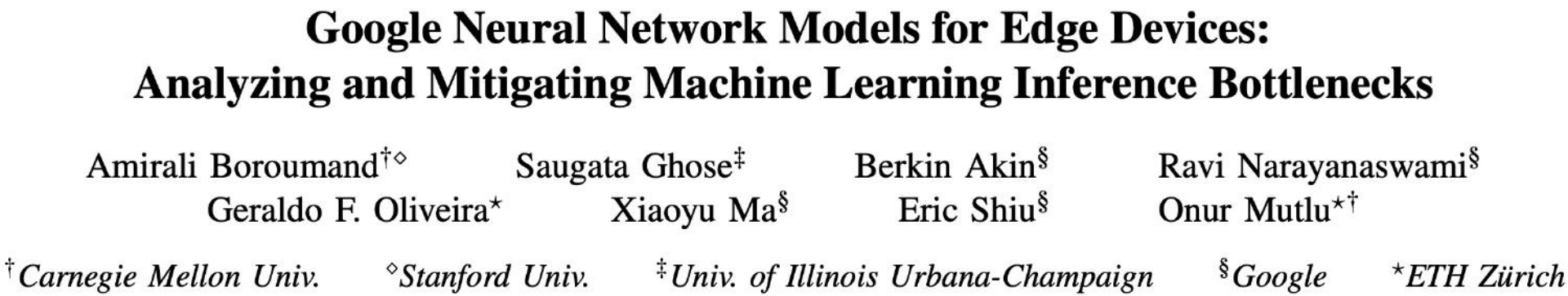
References

[1] J. Ahn *et al.*, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," in *ISCA*, 2015.
[2] Hybrid Memory Cube Consortium, "HMC Specification 1.1," 2013.
[3] J. Jeddeloh and B. Keeth, "Hybrid Memory Cube: New DRAM Architecture Increases Density and Performance," in *VLSIT*, 2012.
[4] JEDEC, "High Bandwidth Memory (HBM) DRAM," Standard No. JESD235, 2013.
[5] N. Jouppi *et al.*, "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embedding," in *ISCA*, 2023.
[6] J. Fowers *et al.*, "A Configurable Cloud-Scale DNN Processor for Real-Time AI," in *ISCA*, 2018.
[7] S. Lie, "Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning," in *IEEE Micro*, 2023.
[8] E. Talpes *et al.*, "The Microarchitecture of DOJO, Tesla's Exa-Scale Computer," in *IEEE Micro*, 2023.
[9] A. Ishii and R. Wells, "NVLink-Network Switch - NVIDIA's Switch Chip for High Communication-Bandwidth SuperPODs," in *Hot Chips*, 2022.
[10] J. Ahn *et al.*, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture," in *ISCA*, 2015.
[11] R. Pope *et al.*, "Efficiently Scaling Transformer Inference," in *MLSys*, 2023.
[12] D. Lepikhin *et al.*, "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding," in *ICLR*, 2021.
[13] S. Wang *et al.*, "Overlap Communication with Dependent Computation via Decomposition in Large Deep Learning Models," in *ASPLOS*, 2023.
[14] J. Ahn *et al.*, "AIM: Energy-Efficient Aggregation Inside the Memory Hierarchy," *ACM TACO*, vol. 13, no. 4, 2016.
[15] O. Mutlu *et al.*, "A Modern Primer on Processing in Memory," *Emerging Computing: From Devices to Systems*, 2021, https://arxiv.org/abs/2012.03112.
[16] M. Zhang *et al.*, "GraphP: Reducing Communication for PIM-Based Graph Processing with Efficient Data Partition," in *HPCA*, 2018.
[17] L. Song, "GraphR: Accelerating Graph Processing Using ReRAM," in *HPCA*, 2018.
[18] Y. Zhuo *et al.*, "GraphQ: Scalable PIM-Based Graph Processing," in *MICRO*, 2019.
[19] G. Dai *et al.*, "GraphH: A Processing-in-Memory Architecture for Large-Scale Graph Processing," *IEEE TCAD*, 2018.
[20] G. Li *et al.*, "GraphIA: An In-situ Accelerator for Large-scale Graph Processing ," in *MEMSYS*, 2018.
[21] S. Rheindt *et al.*, "NEMESYS: Near-Memory Graph Copy Enhanced System-Software," in *MEMSYS*, 2019.
[22] L. Belayneh and V. Bertacco, "GraphVine: Exploiting Multicast for Scalable Graph Analytics," in *DATE*, 2020.
[23] N. Challapalle *et al.*, "GaaS-X: Graph Analytics Accelerator Supporting Sparse Data Representation using Crossbar Architectures," in *ISCA*, 2020.
[24] M. Zhou *et al.*, "Ultra Efficient Acceleration for De Novo Genome Assembly via Near-Memory Computing," in *PACT*, 2021.
[25] X. Xie *et al.*, "SpaceA: Sparse Matrix Vector Multiplication on Processing-in-Memory Accelerator," in *HPCA*, 2021.
[26] M. Zhou *et al.*, "HyGraph: Accelerating Graph Processing with Hybrid Memory-Centric Computing," in *DATE*, 2021.
[27] M. Lenjani *et al.*, "Gearbox: A Case for Supporting Accumulation Dispatching and Hybrid Partitioning in PIM-based Accelerators," in *ISCA*, 2022.
[28] M. Orenes-Vera *et al.*, "Dalorex: A Data-Local Program Execution and Architecture for Memory-Bound Applications," in *HPCA*, 2023.
[29] J. Choquette, "Nvidia Hopper GPU: Scaling Performance," in *Hot Chips*, 2022.
[30] F. Devaux, "The True Processing In Memory Accelerator," in *Hot Chips*, 2019.
[31] J. Gómez-Luna *et al.*, "Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System," *IEEE Access*, 2022.
[32] J. Gomez-Luna *et al.*, "Evaluating Machine Learning Workloads on Memory-Centric Computing Systems," in *ISPASS*, 2023.
[33] S. Lee *et al.*, "Hardware Architecture and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product," in *ISCA*, 2021.
[34] Y.-C. Kwon *et al.*, "25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2 TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications," in *ISSCC*, 2021.
[35] L. Ke *et al.*, "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM," *IEEE Micro*, 2021.
[36] I. Oliveira, "Improving In-Memory Database Operations with Acceleration DIMM (AXDIMM)," in *DaMoN*, 2022.
[37] S. Lee *et al.*, "A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications," in *ISSCC*, 2022.
[38] D. Niu *et al.*, "184QPS/W 64Mb/mm^2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System," in *ISSCC*, 2022.
[39] Y. Kwon, "System Architecture and Software Stack for GDDR6-AiM," in *HCS*, 2022.
[40] G. Singh *et al.*, "FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications," *IEEE Micro*, 2021.
[41] G. Singh *et al.*, "Accelerating Weather Prediction using Near-Memory Reconfigurable Fabric," *ACM TRETS*, 2021.
[42] S. Hong *et al.*, "PGX.D: A Fast Distributed Graph Processing Engine," in *SC*, 2015.
[43] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *ICLR*, 2017.
[44] L. Nai *et al.*, "GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks," in *HPCA*, 2017.
[45] M. Besta *et al.*, "SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems," in *MICRO*, 2021.
[46] T. J. Ham *et al.*, "Graphicionado: A High-Performance and Energy-Efficient Accelerator for Graph Analytics," in *MICRO*, 2016.

https://arxiv.org/pdf/2306.16093

Accelerating ML Inference

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



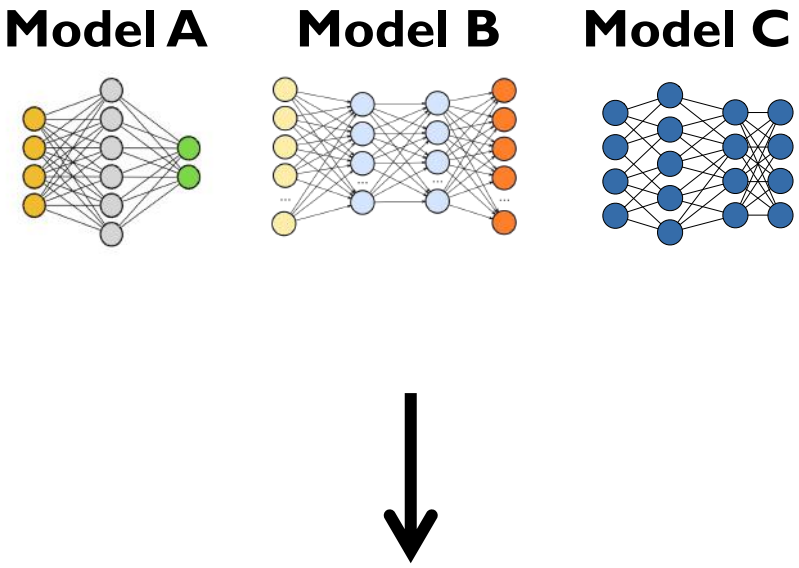
Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

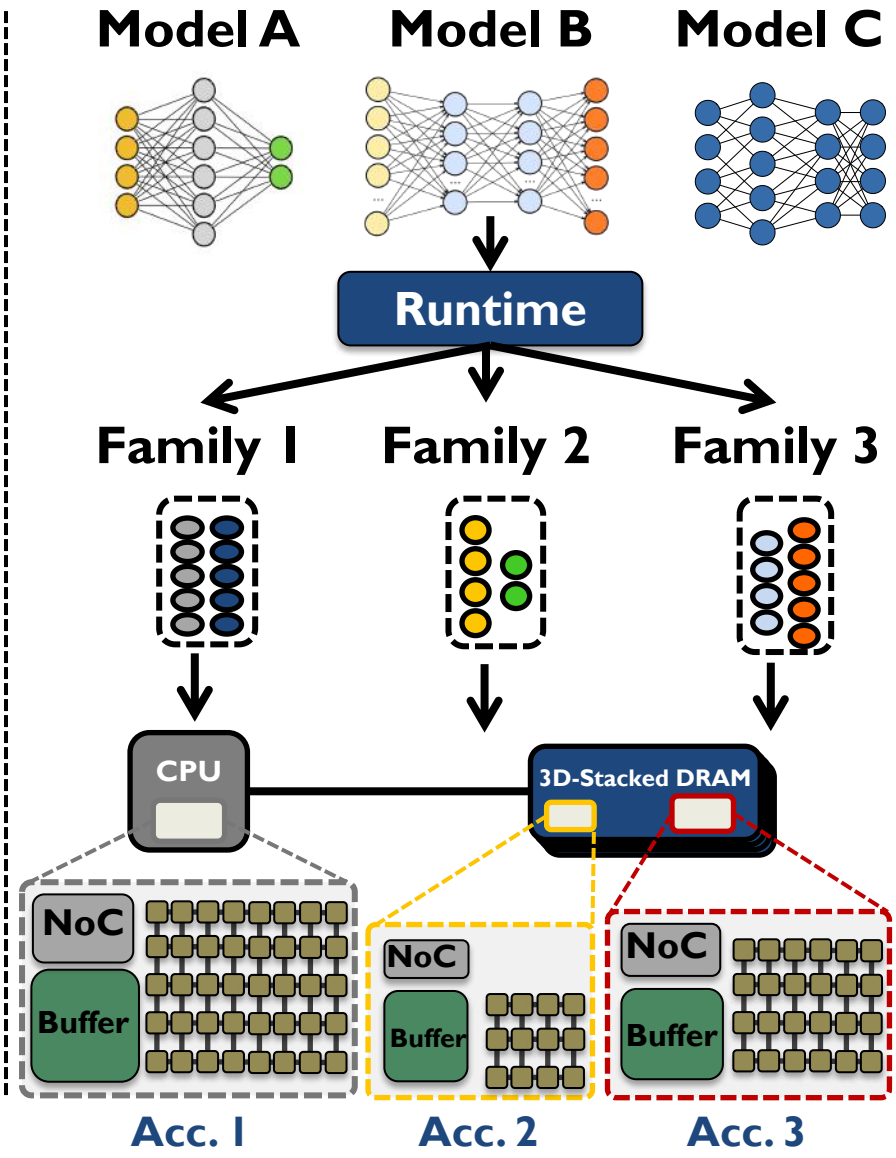
Mensa High-Level Overview

Edge TPU Accelerator



Monolithic Accelerator

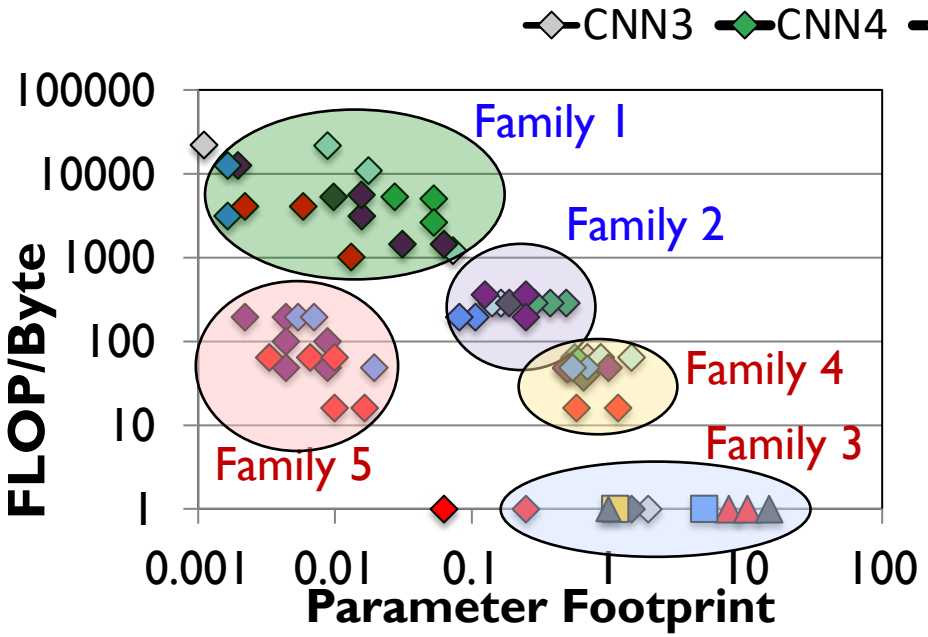
Mensa

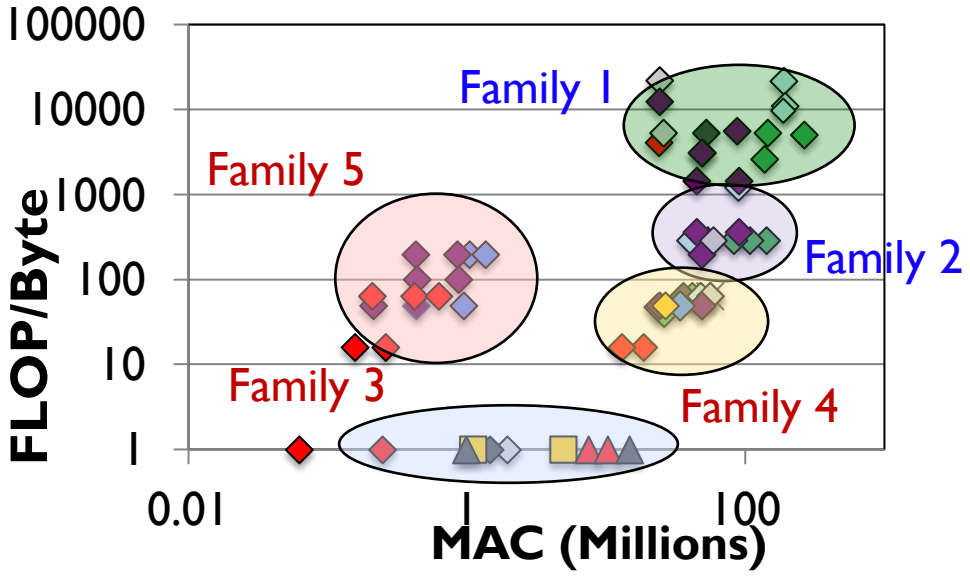


Heterogeneous Accelerators

Identifying Layer Families

Key observation: the majority of layers group into a small number of layer families

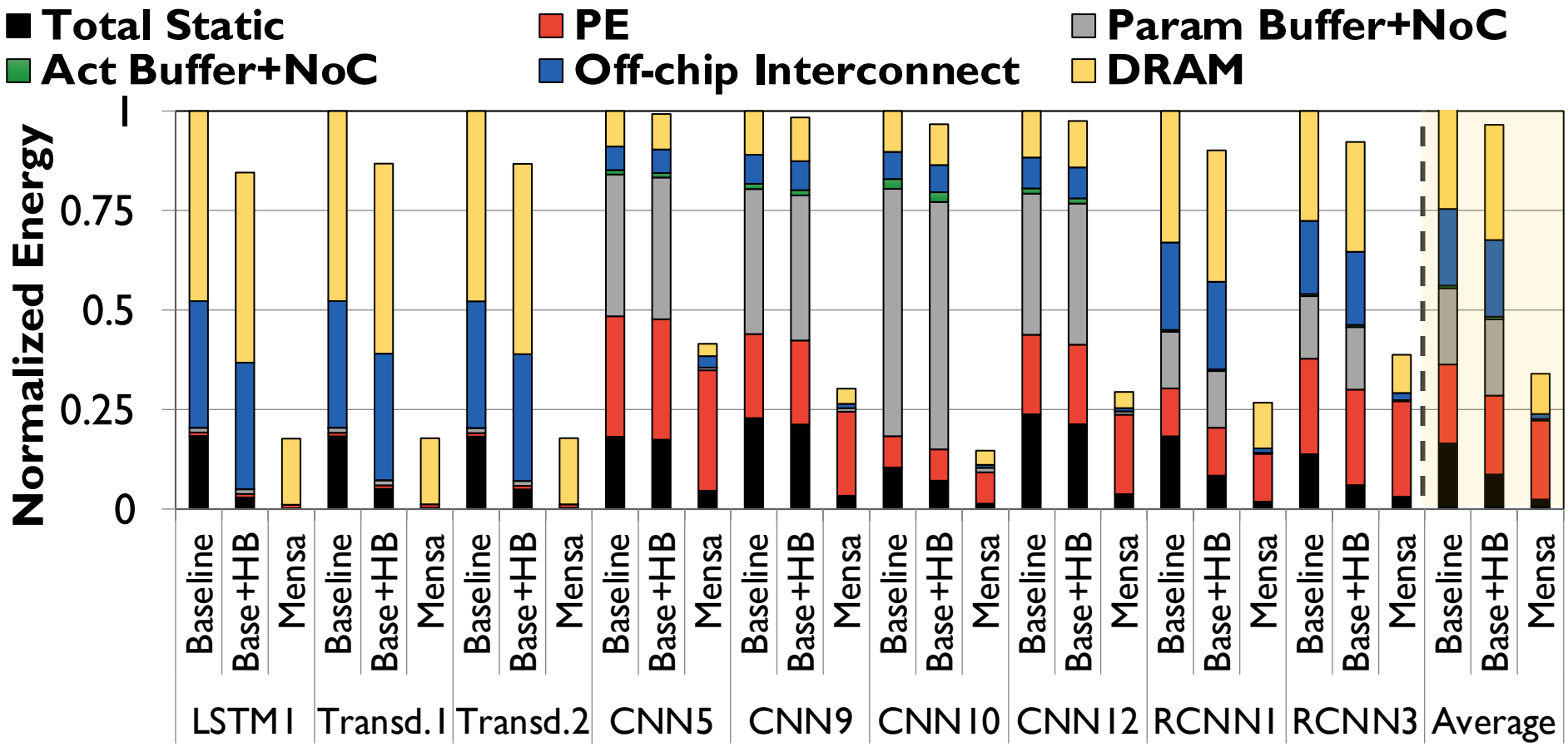




Families 1 & 2: low parameter footprint, high data reuse and MAC intensity
→ **compute-centric layers**

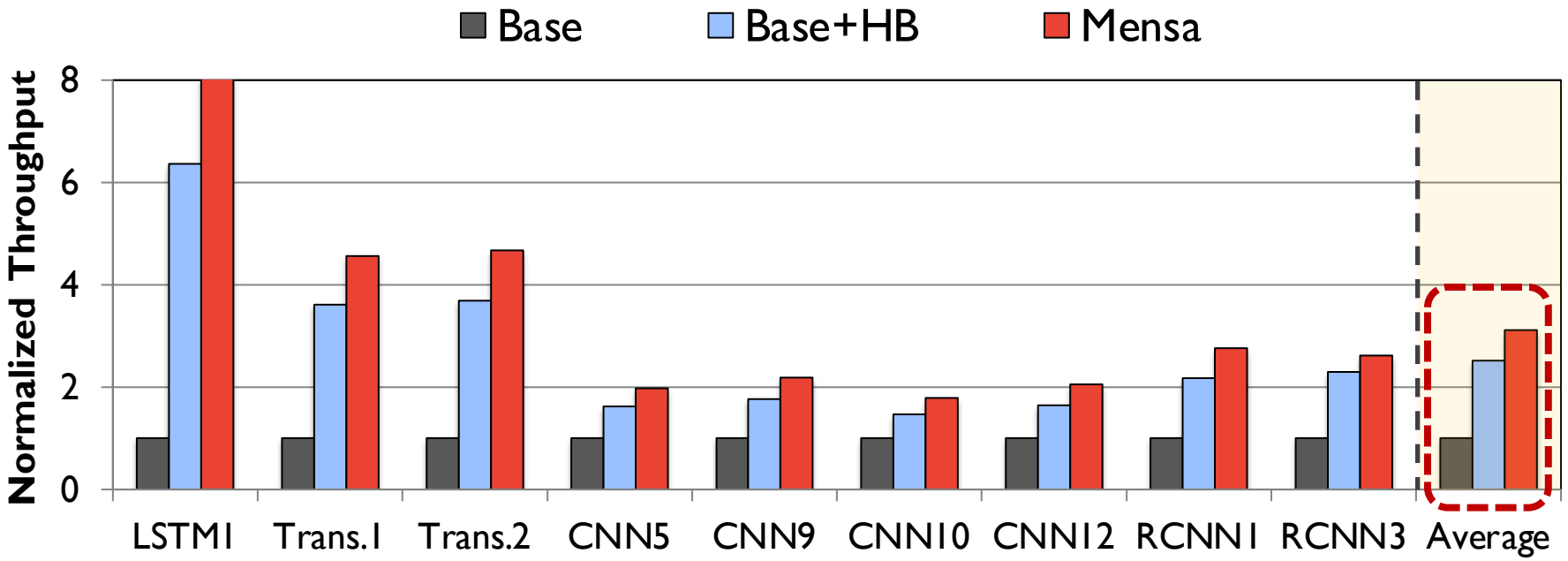
Families 3, 4 & 5: high parameter footprint, low data reuse and MAC intensity
→ **data-centric layers**

Mensa: Energy Reduction



Mensa-G reduces energy consumption by 3.0X compared to the baseline Edge TPU

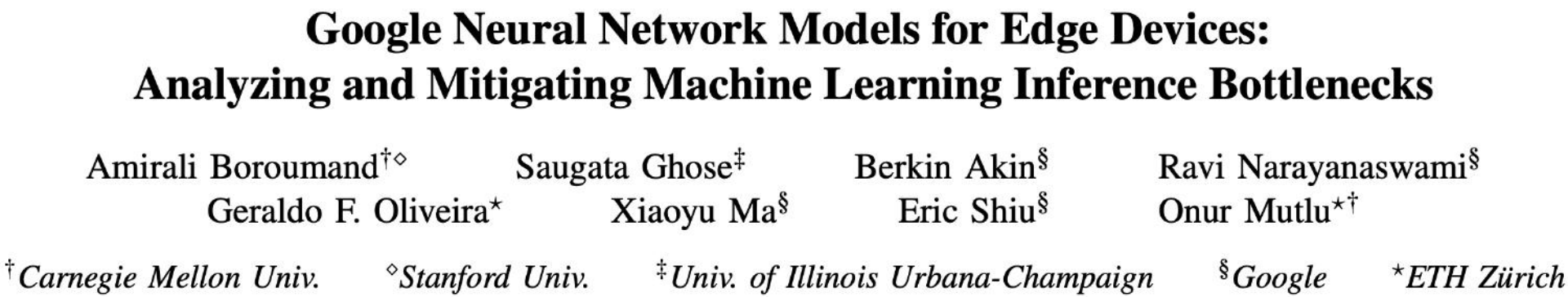
Mensa: Throughput Improvement



Mensa-G improves inference throughput by 3.1X compared to the baseline Edge TPU

Mensa: Highly-Efficient ML Inference

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



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

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

PAPI: Hybrid System for Near-Memory LLM Inference

- Yintao He, Haiyu Mao, Christina Giannoula, Mohammad Sadrosadati, Juan Gomez-Luna, Huawei Li, Xiaowei Li, Ying Wang, and Onur Mutlu, **"PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System,"** *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems* (***ASPLOS***), Rotterdam, Netherlands, April 2025.

PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System

Yintao He[1,2] Haiyu Mao[3,4] Christina Giannoula[5,6,4] Mohammad Sadrosadati[4]
Juan Gómez-Luna[7] Huawei Li[1,2] Xiaowei Li[1,2] Ying Wang[1] Onur Mutlu[4]
[1]SKLP, Institute of Computing Technology, CAS [2]University of Chinese Academy of Sciences [3] King's College London
[4]ETH Zürich [5]University of Toronto [6]Vector Institute [7] NVIDIA

https://arxiv.org/pdf/2502.15470

PAPI LLM Inference System [ASPLOS 2025]

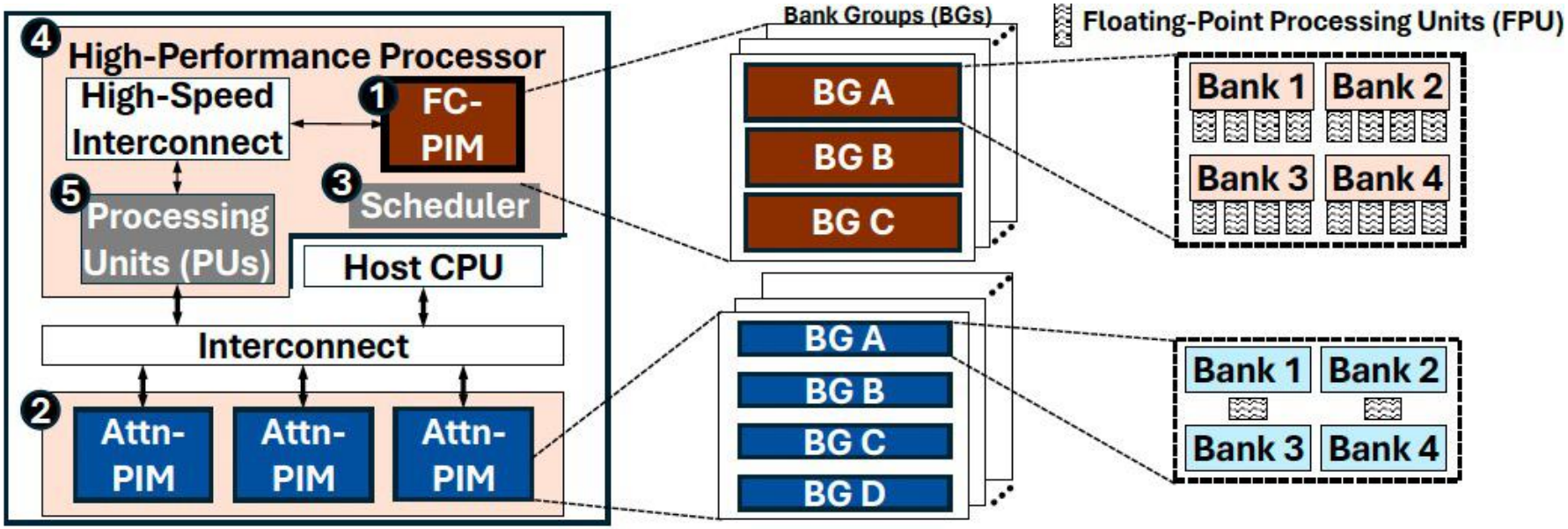


Fig. 5: **Overview of the PAPI LLM Inference System.** Adapted from [18].

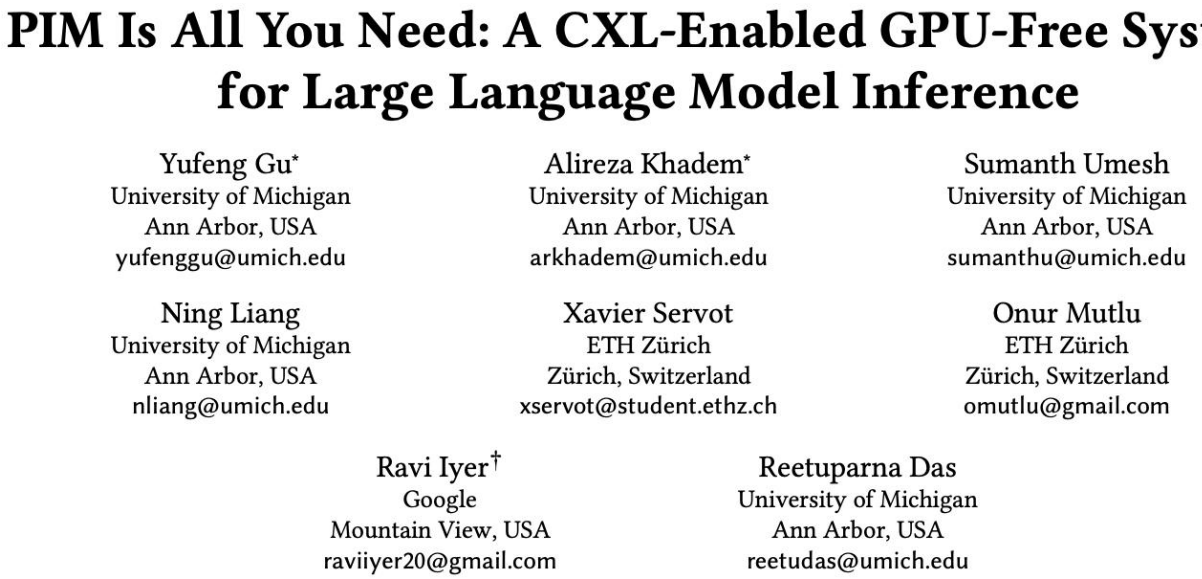
PAPI over best prior LLM decoding system

- 1.8× speedup
- 3.4× energy efficiency increase

https://arxiv.org/pdf/2502.15470

CENT: GPU-Free System for Near-Memory LLM Inference

- Yufeng Gu, Alireza Khadem, Sumanth Umesh, Ning Liang, Xavier Servot, Onur Mutlu, Ravi Iyer, and Reetuparna Das,
  "PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference,"
  *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Rotterdam, Netherlands, April 2025.
  Officially artifact evaluated as available, functional, and reproducible.



PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference

Yufeng Gu*
University of Michigan
Ann Arbor, USA
yufenggu@umich.edu

Alireza Khadem*
University of Michigan
Ann Arbor, USA
arkhadem@umich.edu

Sumanth Umesh
University of Michigan
Ann Arbor, USA
sumanthu@umich.edu

Ning Liang
University of Michigan
Ann Arbor, USA
nliang@umich.edu

Xavier Servot
ETH Zürich
Zürich, Switzerland
xservot@student.ethz.ch

Onur Mutlu
ETH Zürich
Zürich, Switzerland
omutlu@gmail.com

Ravi Iyer†
Google
Mountain View, USA
raviiyer20@gmail.com

Reetuparna Das
University of Michigan
Ann Arbor, USA
reetudas@umich.edu

https://arxiv.org/pdf/2502.07578

CENT LLM Inference System [ASPLOS 2025]

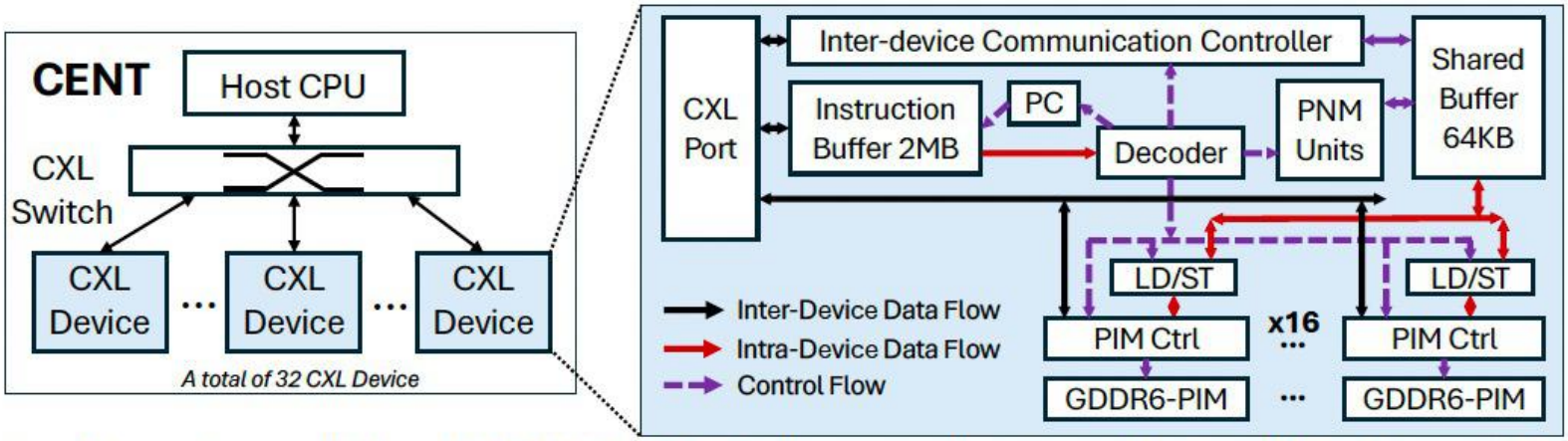


Fig. 6: **Overview of the CENT LLM Inference System.** Host CPU drives 32 CXL devices, each having a CXL controller, PNM units, and 16 GDDR6-PIM chips. The LLM inference task is partitioned between PNM units and GDDR6-PIM chips. CENT provides communication mechanisms within and across CXL devices to coordinate and scale computation. Adapted from [19].

CENT over best prior GPU LLM inference system

- 2.3× higher throughput
- 5.2× higher tokens per dollar
- 2.4× lower hardware cost

https://arxiv.org/pdf/2502.07578

Many Examples …

A Modern Primer on Processing-In-Memory

Onur Mutlu[a], Saugata Ghose[b], Juan Gómez-Luna[c], Rachata Ausavarungnirun[d],
Mohammad Sadrosadati[a], Geraldo F. Oliveira[a]

SAFARI Research Group

[a]*ETH Zürich*
[b]*University of Illinois Urbana-Champaign*
[c]*NVIDIA Research*
[d]*MangoBoost Inc.*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, Rachata Ausavarungnirun, Mohammad Sadrosadati, and Geraldo F. Oliveira,
"A Modern Primer on Processing in Memory"
Invited Book Chapter in ***Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann***, Springer, 2022.

https://arxiv.org/pdf/2012.03112

Processing in Memory: Two Types

1. Processing **near** Memory
2. Processing **using** Memory

Focus: Processing using DRAM

- We can natively support
  - Bulk bitwise COPY and INIT/ZERO
  - Bulk bitwise AND, OR, NOT, MAJ, NOR, NAND
  - True Random Number Generation; Physical Unclonable Functions
  - More complex computation using Lookup Tables
- At low cost
- Using analog computation capability of DRAM
  - Idea: activating (multiple) rows performs computation
    - Even in commodity off-the-shelf DRAM chips!

- 30X-257X performance and energy improvements

Seshadri+"RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

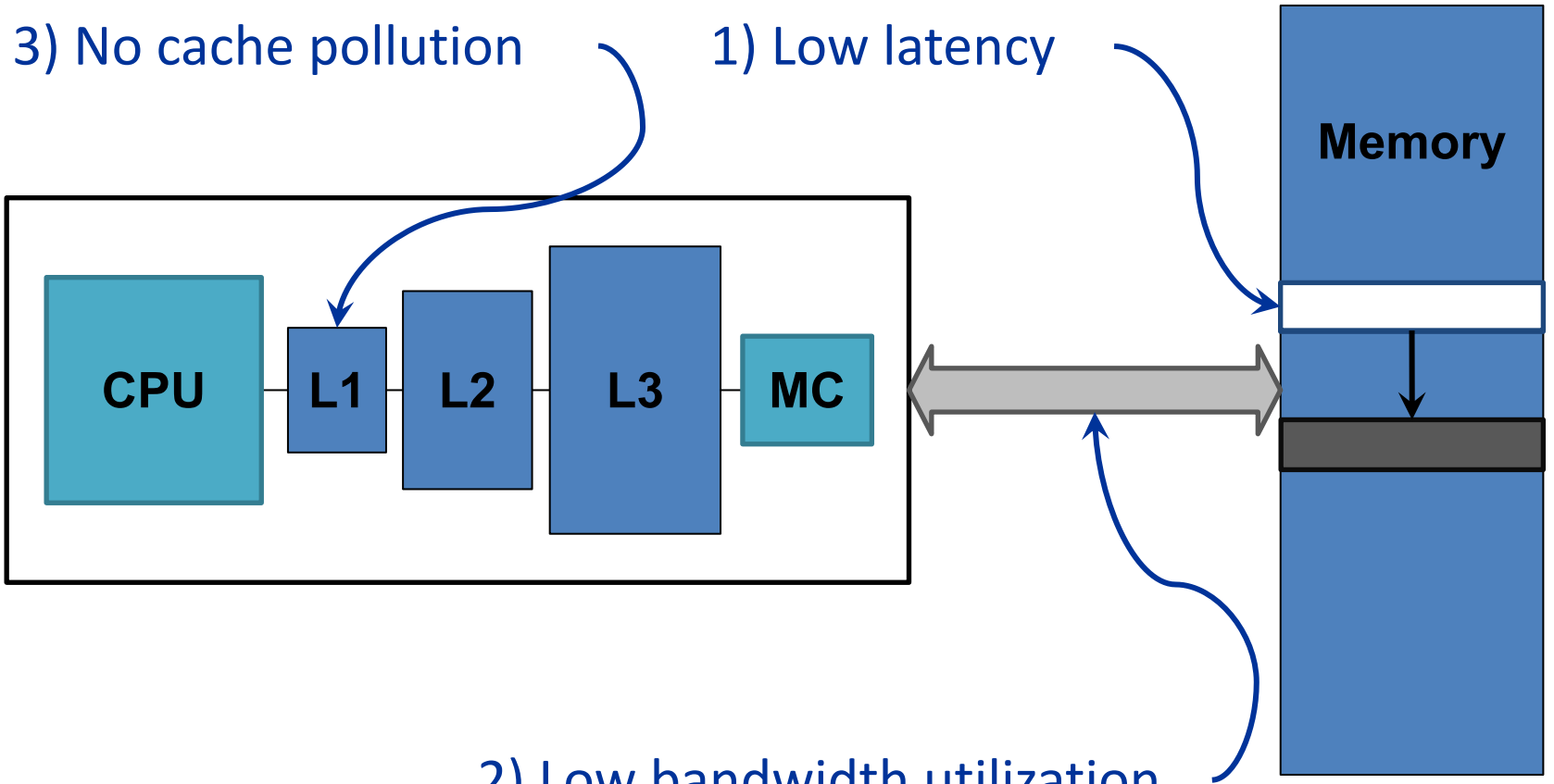
Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM", IEEE CAL 2015.

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.

Hajinazar+, "SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM," ASPLOS 2021.

Oliveira+, "MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Processing," HPCA 2024.

Future Systems: In-Memory Copy

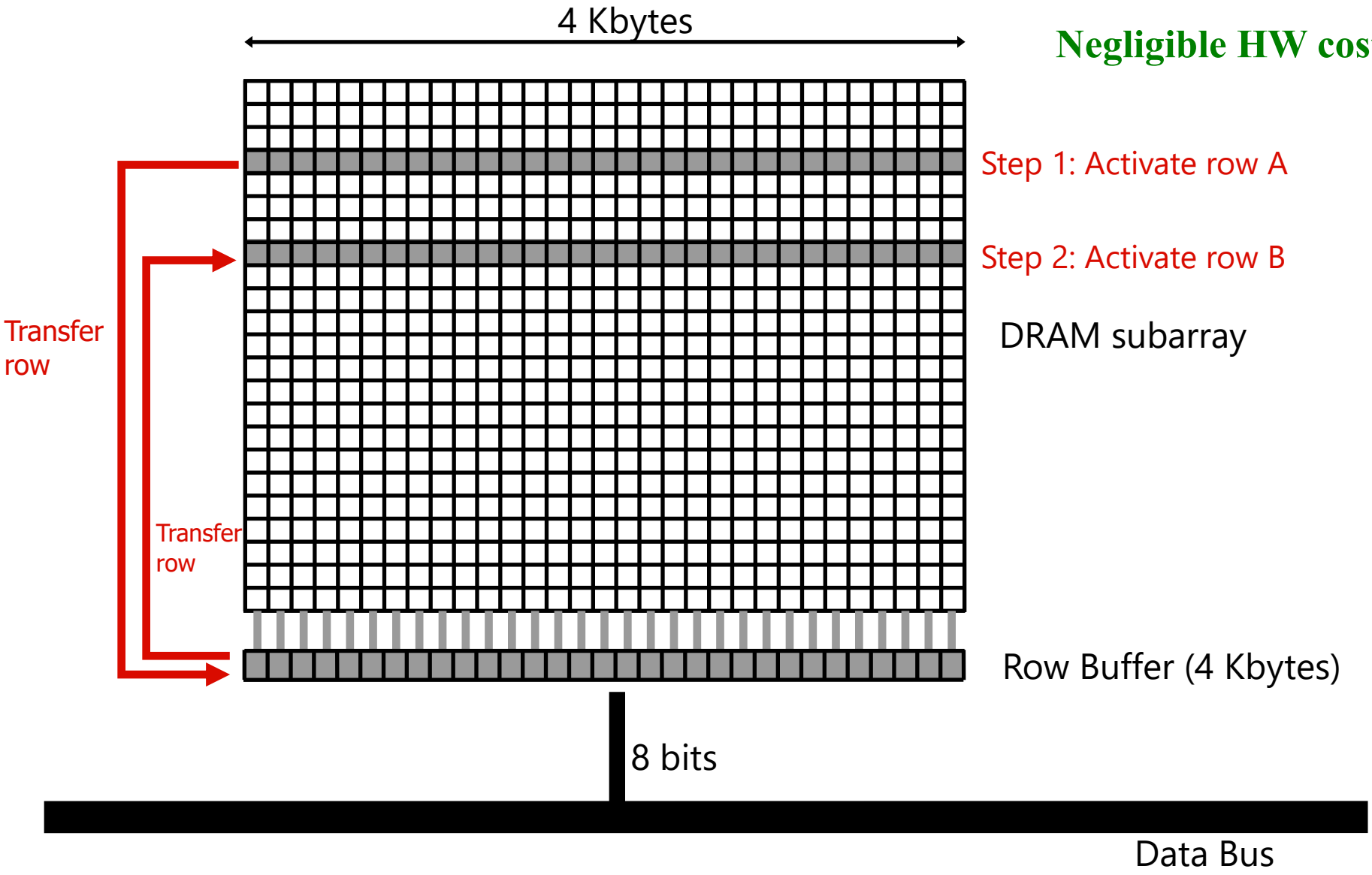


1046ns, 3.6uJ → 90ns, 0.04uJ

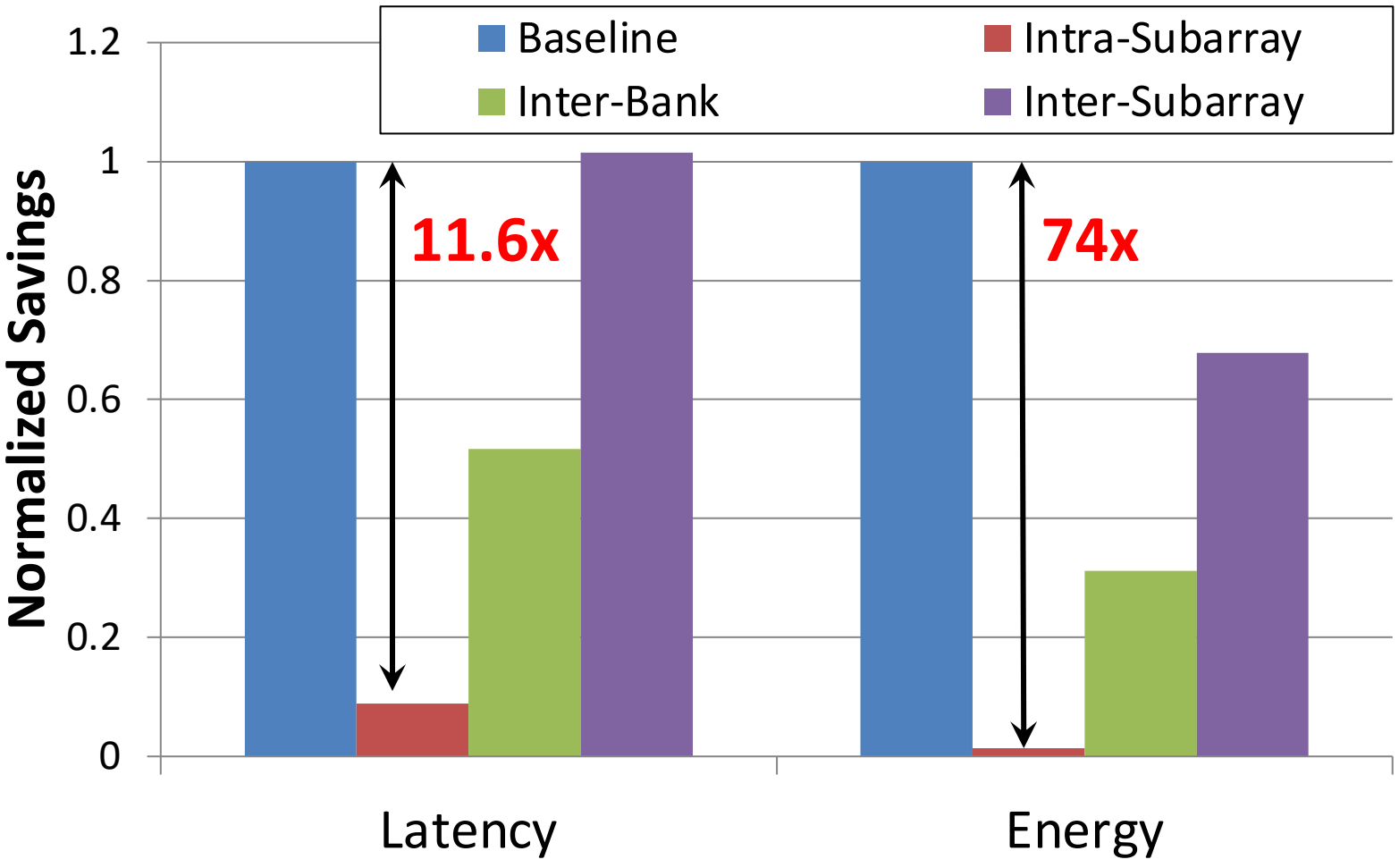
RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates

Negligible HW cost



RowClone: Latency and Energy Savings



Seshadri et al., "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

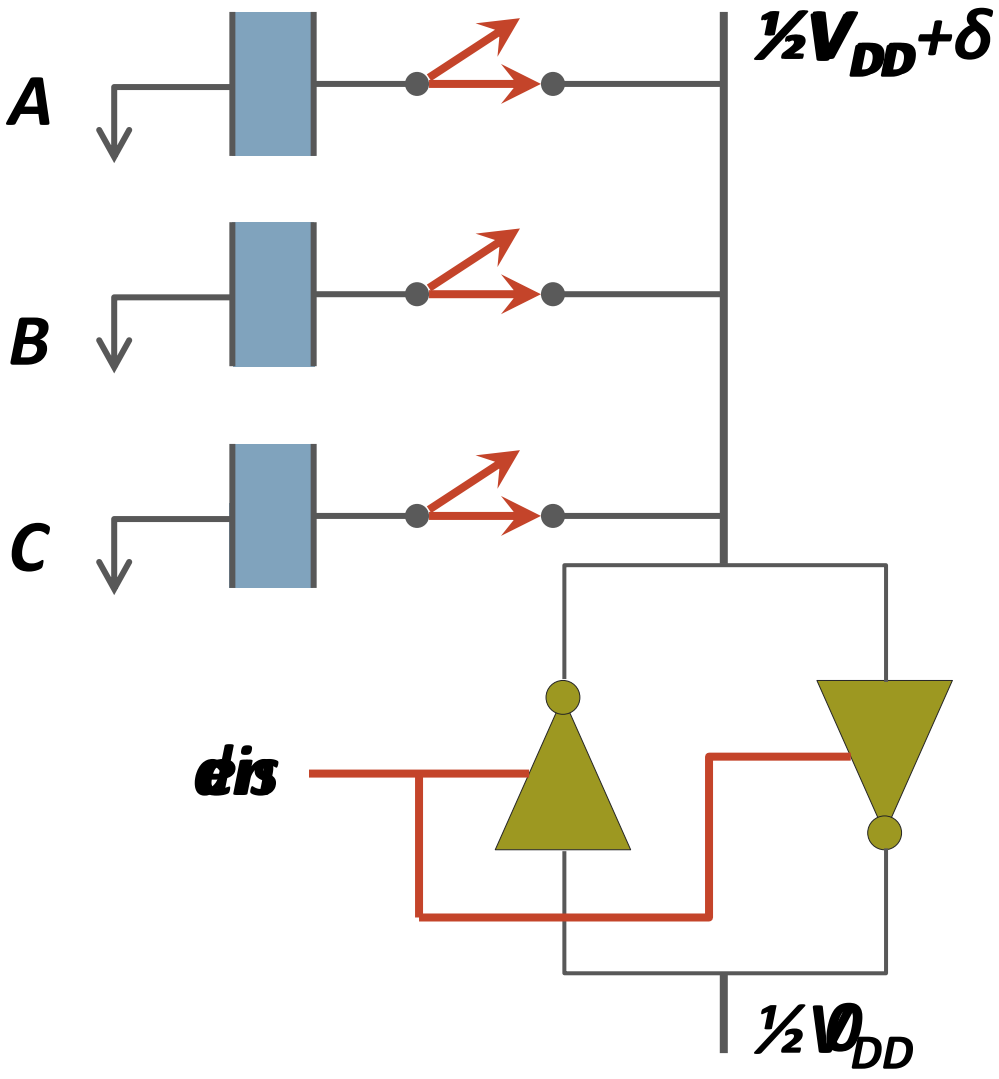
More on RowClone

- 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 vseshadr@cs.cmu.edu
Yoongu Kim yoongukim@cmu.edu
Chris Fallin* cfallin@c1f.net
Donghyuk Lee donghyuk1@cmu.edu
Rachata Ausavarungnirun rachata@cmu.edu
Gennady Pekhimenko gpekhime@cs.cmu.edu
Yixin Luo yixinluo@andrew.cmu.edu
Onur Mutlu onur@cmu.edu
Phillip B. Gibbons† phillip.b.gibbons@intel.com
Michael A. Kozuch† michael.a.kozuch@intel.com
Todd C. Mowry tcm@cs.cmu.edu
Carnegie Mellon University †Intel Pittsburgh

In-DRAM AND/OR: Triple Row Activation

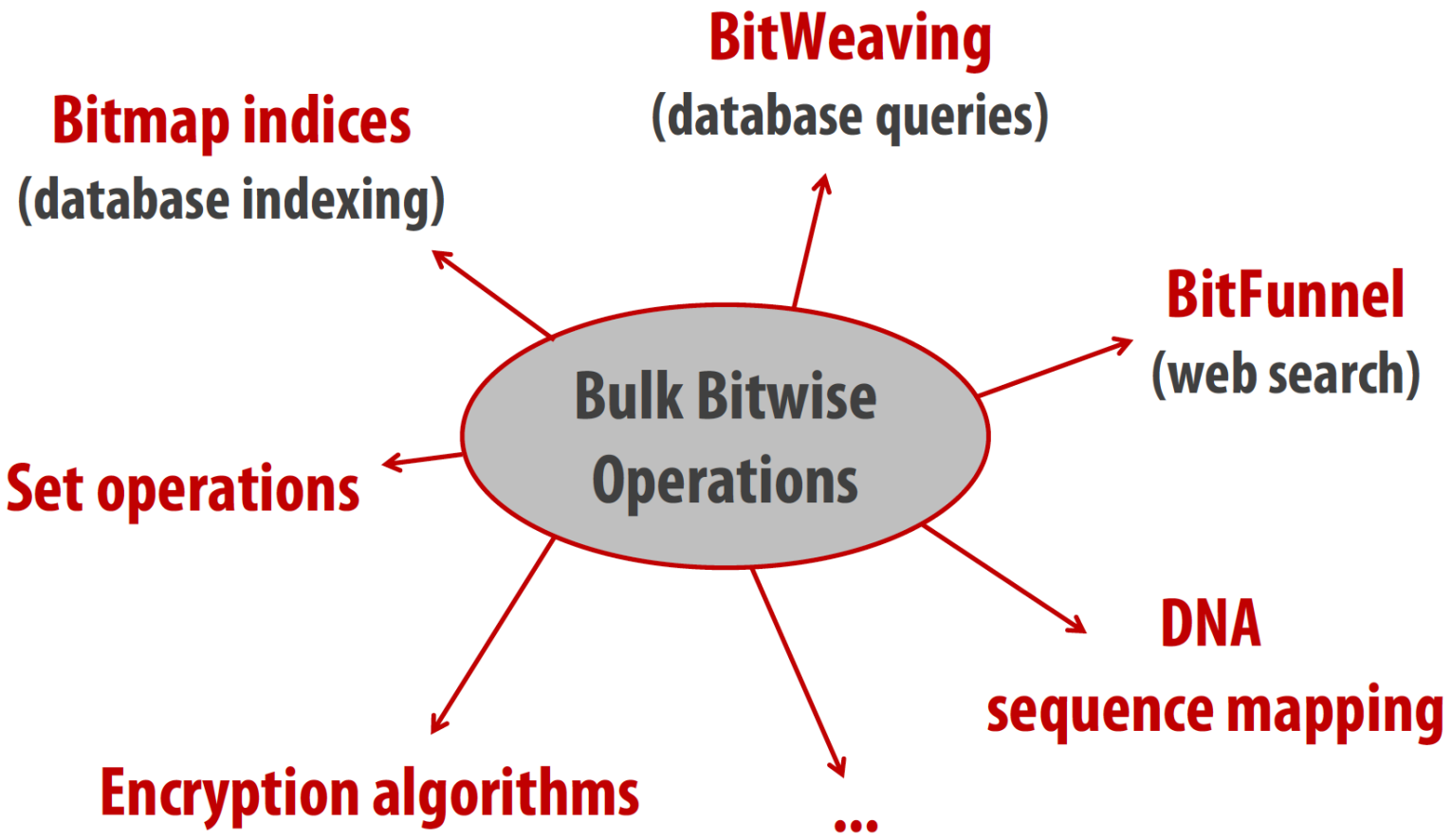


Final State
AB + BC + AC

C(A + B) + ~C(AB)

Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM", IEEE CAL 2015.

Bulk Bitwise Operations in Workloads



[1] Li and Patel, BitWeaving, SIGMOD 2013
[2] Goodwin+, BitFunnel, SIGIR 2017

In-DRAM Acceleration of Database Queries

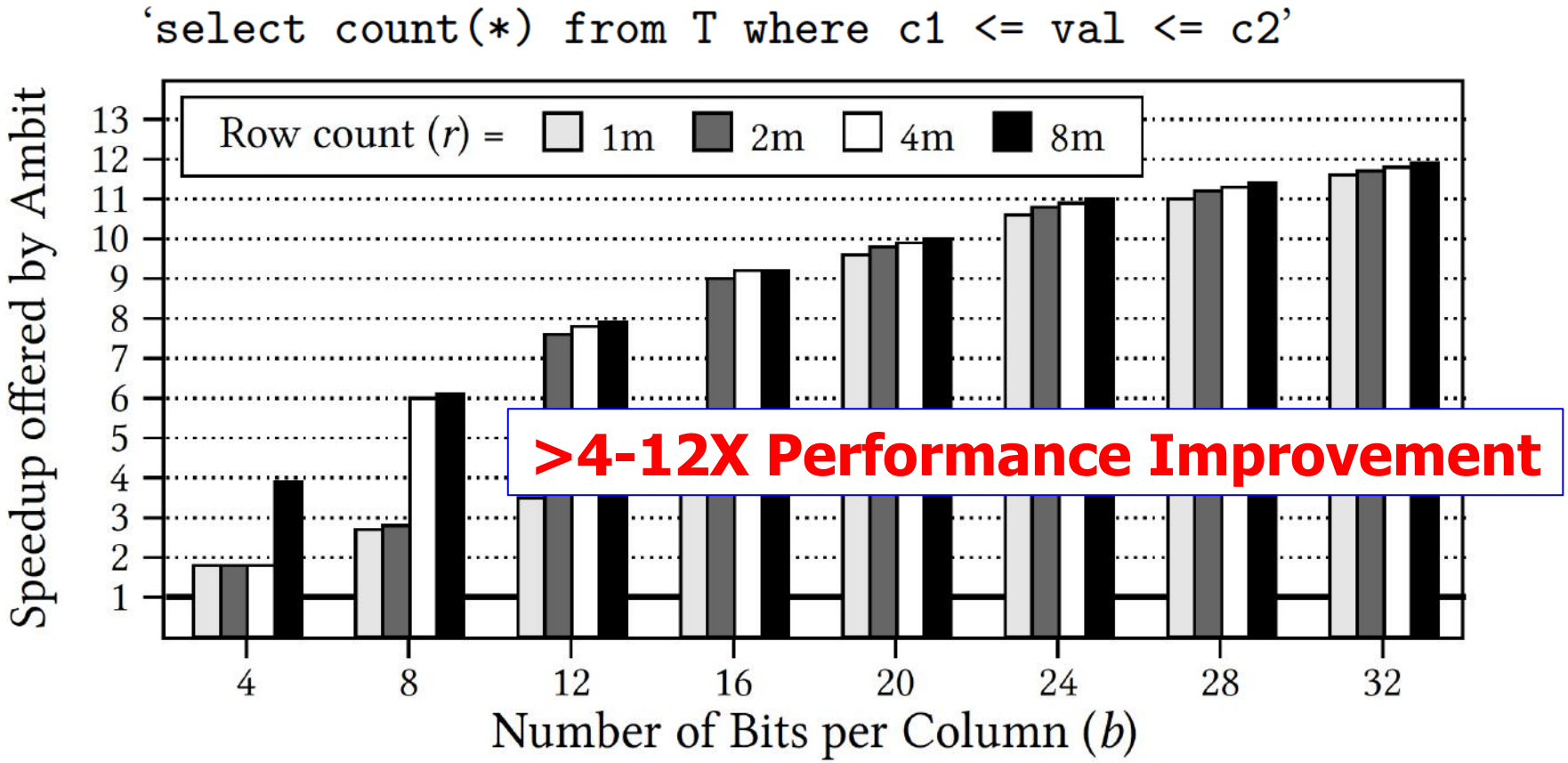
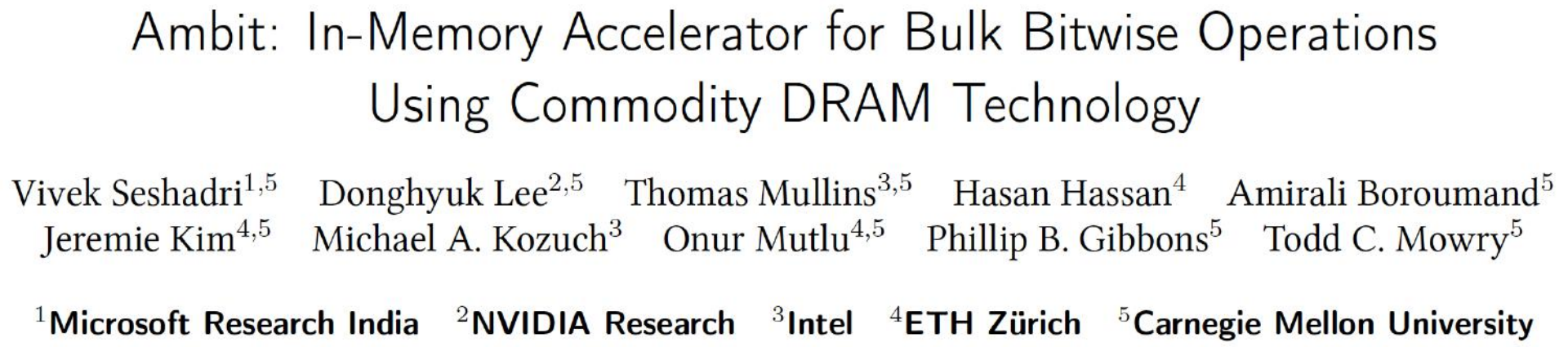


Figure 11: Speedup offered by Ambit over baseline CPU with SIMD for BitWeaving

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology," MICRO 2017.

More on Ambit

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
  "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology"
  *Proceedings of the 50th International Symposium on Microarchitecture (**MICRO**)*, Boston, MA, USA, October 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]


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

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

[1]Microsoft Research India [2]NVIDIA Research [3]Intel [4]ETH Zürich [5]Carnegie Mellon University

SIMDRAM Framework: Generalization

- 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[1,2] *Geraldo F. Oliveira[1] Sven Gregorio[1] João Dinis Ferreira[1]
Nika Mansouri Ghiasi[1] Minesh Patel[1] Mohammed Alser[1] Saugata Ghose[3]
Juan Gómez-Luna[1] Onur Mutlu[1]

[1]ETH Zürich [2]Simon Fraser University [3]University of Illinois at Urbana–Champaign

MIMDRAM: More Flexible Processing using DRAM

- **Appears at HPCA 2024** https://arxiv.org/pdf/2402.19080.pdf

MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Computing

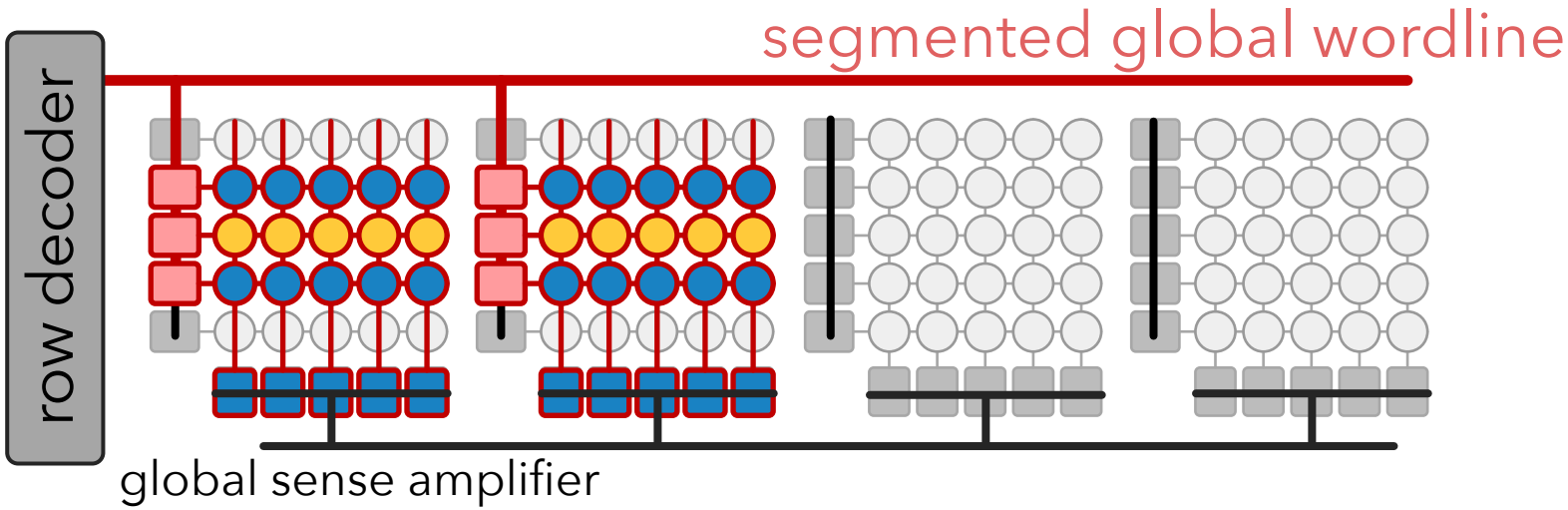
Geraldo F. Oliveira[†] Ataberk Olgun[†] Abdullah Giray Yağlıkçı[†] F. Nisa Bostancı[†]
Juan Gómez-Luna[†] Saugata Ghose[‡] Onur Mutlu[†]

[†] *ETH Zürich* [‡] *Univ. of Illinois Urbana-Champaign*

*Our **goal** is to design a flexible PUD system that overcomes the limitations caused by the large and rigid granularity of PUD. To this end, we propose MIMDRAM, a hardware/software co-designed PUD system that introduces new mechanisms to allocate and control only the necessary resources for a given PUD operation. The key idea of MIMDRAM is to leverage fine-grained DRAM (i.e., the ability to independently access smaller segments of a large DRAM row) for PUD computation. MIMDRAM exploits this key idea to enable a multiple-instruction multiple-data (MIMD) execution model in each DRAM subarray (and SIMD execution within each DRAM row segment).*

MIMDRAM: Key Idea

Enable narrower-width operations than a DRAM row



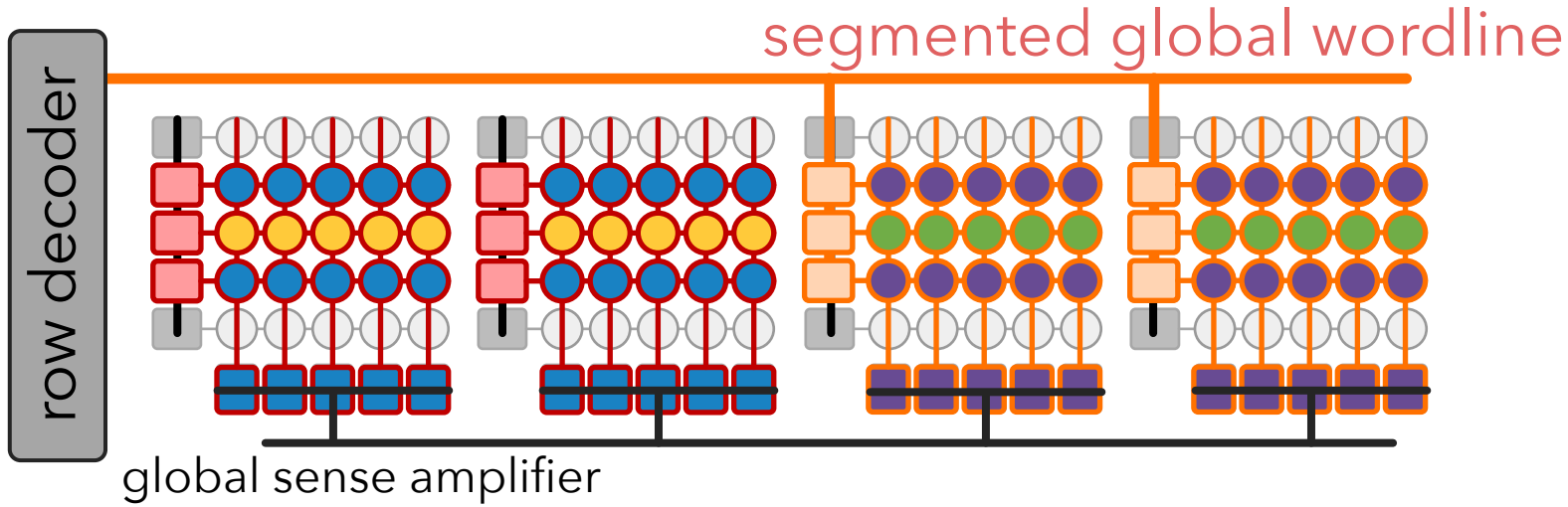
Use fine-grained DRAM for processing-using-DRAM:

1 Improves SIMD utilization

- for a single PUD operation, only access the DRAM mats with target data

MIMDRAM: Key Idea

Enable narrower-width operations than a DRAM row



Use fine-grained DRAM for processing-using-DRAM:

1 Improves SIMD utilization

- for a single PUD operation, only access the DRAM mats with target data
- for multiple PUD operations, execute independent operations concurrently
  → **multiple instruction, multiple data (MIMD) execution model**

Sectored DRAM

- Ataberk Olgun, F. Nisa Bostanci, Geraldo F. Oliveira, Yahya Can Tugrul, Rahul Bera, A. Giray Yaglikci, Hasan Hassan, Oguz Ergin, and Onur Mutlu, **"Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture"** *ACM Transactions on Architecture and Code Optimization* (***TACO***), [online] June 2024.
  [arXiv version]
  [ACM Digital Library version]

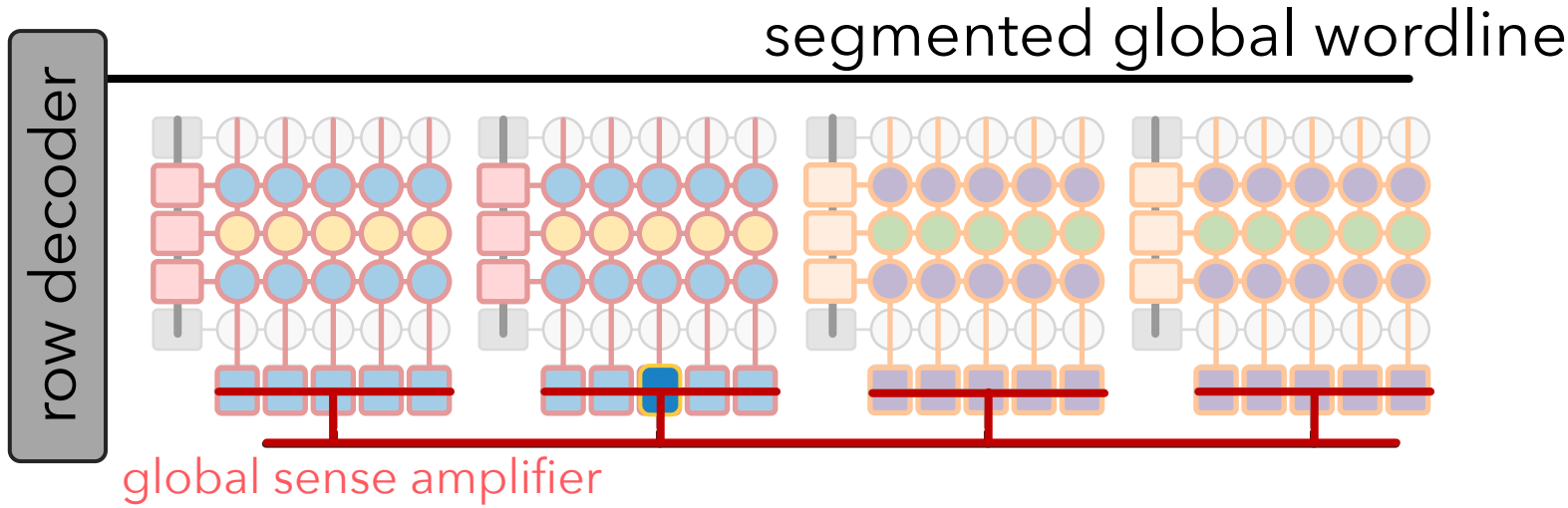
Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Ataberk Olgun[§] F. Nisa Bostancı[§†] Geraldo F. Oliveira[§] Yahya Can Tuğrul[§†] Rahul Bera[§]
A. Giray Yağlıkcı[§] Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

https://arxiv.org/pdf/2207.13795

MIMDRAM: Key Idea

Enable narrower-width operations than a DRAM row



Use fine-grained DRAM for processing-using-DRAM:

1 Improves SIMD utilization

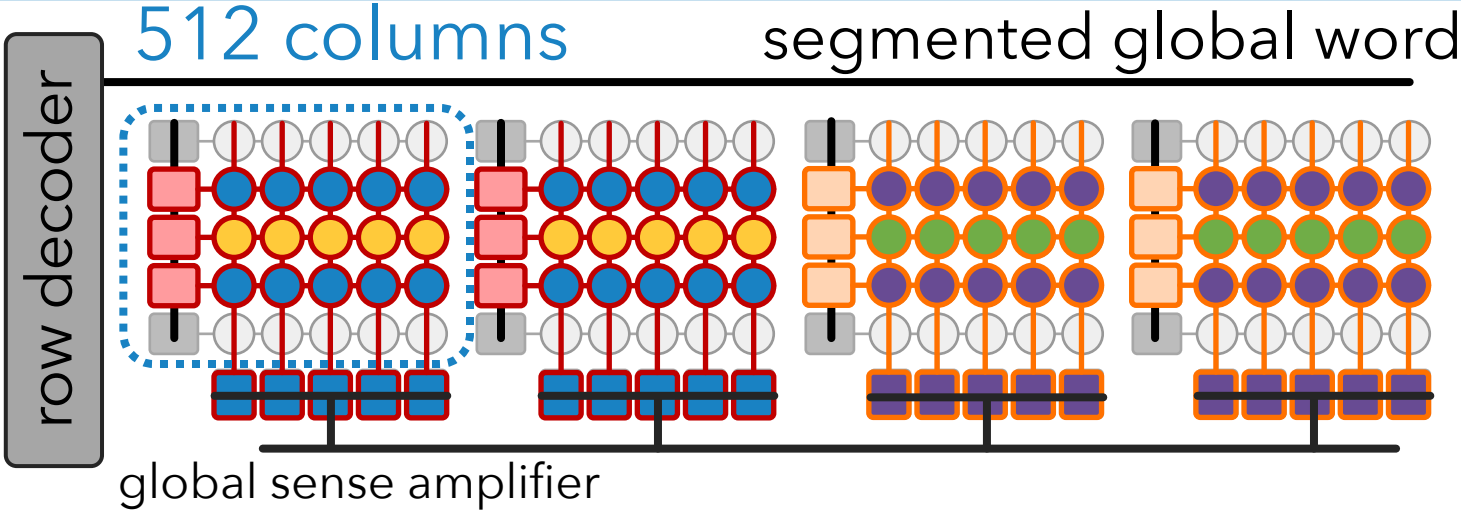
- for a single PUD operation, only access the DRAM mats with target data
- for multiple PUD operations, execute independent operations concurrently
  → **multiple instruction, multiple data (MIMD) execution model**

2 Enables low-cost interconnects for vector reduction

- global and local data buses can be used for inter-/intra-mat communication

MIMDRAM: Key Idea

Enable narrower-width operations than a DRAM row



Use fine-grained DRAM for processing-using-DRAM:

1 Improves SIMD utilization

- for a single PUD operation, only access the DRAM mats with target data
- for multiple PUD operations, execute independent operations concurrently
  → **multiple instruction, multiple data (MIMD) execution model**

2 Enables low-cost interconnects for vector reduction

- global and local data buses can be used for inter-/intra-mat communication

3 Eases programmability

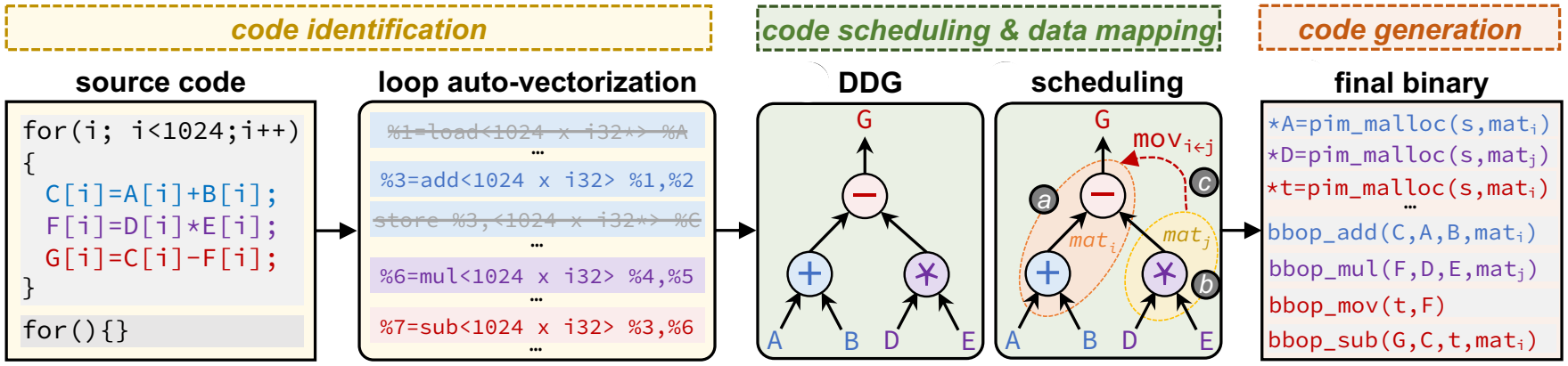
- SIMD parallelism in a DRAM mat is on par with vector ISAs' SIMD width

MIMDRAM:
Compiler Support

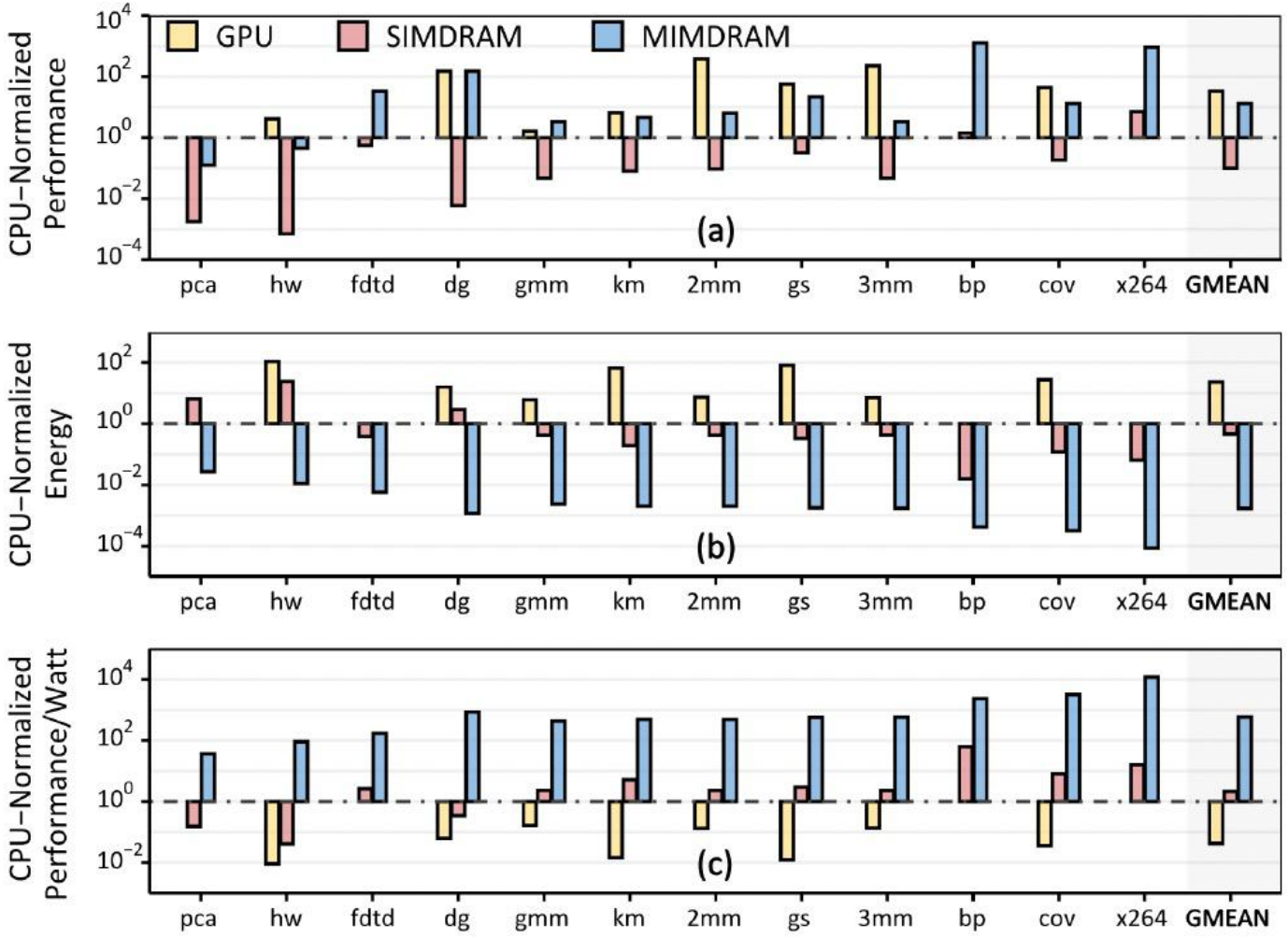
Goal

Transparently to programmer:
extract SIMD parallelism from an application, and
schedule PUD instructions while maximizing utilization

Three new LLVM-based passes targeting PUD execution



MIMDRAM Perf, Energy, Perf/Watt



582X and 13,612X the energy efficiency of CPU and GPU, respectively

Dynamic Precision & Flexible Arithmetic in PuD

- Geraldo Francisco, Mayank Kabra, Yuxin Guo, Kangqi Chen, A. Giray Yaglikci, Melina Soysal, Mohammad Sadrosadati, Joaquin Olivares, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu,
  "Proteus: Enabling High-Performance Processing-Using-DRAM via Dynamic Precision Bit-Serial Arithmetic"
  *Proceedings of the 37th ACM International Conference on Supercomputing (**ICS**)*, Salt Lake City, UT, USA, June 2025.

Proteus: Enabling High-Performance Processing-Using-DRAM with Dynamic Bit-Precision, Adaptive Data Representation, and Flexible Arithmetic

Geraldo F. Oliveira[†] Mayank Kabra[†] Yuxin Guo[‡] Kangqi Chen[†]
A. Giray Yağlıkçı[†] Melina Soysal[†] Mohammad Sadrosadati[†]
Joaquin O. Bueno[⋆] Saugata Ghose[▽] Juan Gómez-Luna[§] Onur Mutlu[†]

[†] *ETH Zürich* [‡] *Cambridge University* [⋆] *Universidad de Córdoba*
[▽] *Univ. of Illinois Urbana-Champaign* [§] *NVIDIA Research*

Dynamic Precision & Flexible Arithmetic in PuD

- Rejected from ASPLOS after satisfying all revision requirements

***Proteus*: En** **DRAM with**
Dynamic Bit-Pre **xible Arithmetic**

Geral Chen[†]
A. ati[†]
Joaquin C ur Mutlu[†]

[†] *ETH* *Córdoba*

Acknowledgments

We thank the anonymous reviewers of ASPLOS 2024, ISCA 2024, MICRO 2024, ASPLOS 2025, and ICS 2025 for their feedback. We thank the SAFARI Research Group members for providing a stimulating intellectual environment. We acknowledge the generous gifts from our industrial partners, including Google, Huawei, Intel, and Microsoft. This work is supported in part by the ETH Future Computing Laboratory (EFCL), Huawei ZRC Storage Team, Semiconductor Research Corporation, AI Chip Center for Emerging Smart Systems (ACCESS), sponsored by InnoHK funding, Hong Kong SAR, and European Union's Horizon programme for research and innovation [101047160 - BioPIM].

Capabilities of Off-The-Shelf Memory

Existing DRAM Chips

Are Already Quite Capable

Real Processing Using Memory Prototype

- End-to-end RowClone & TRNG using off-the-shelf DRAM chips
- Idea: Violate timing parameters to enable RowClone & TRNG

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§†] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§*]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

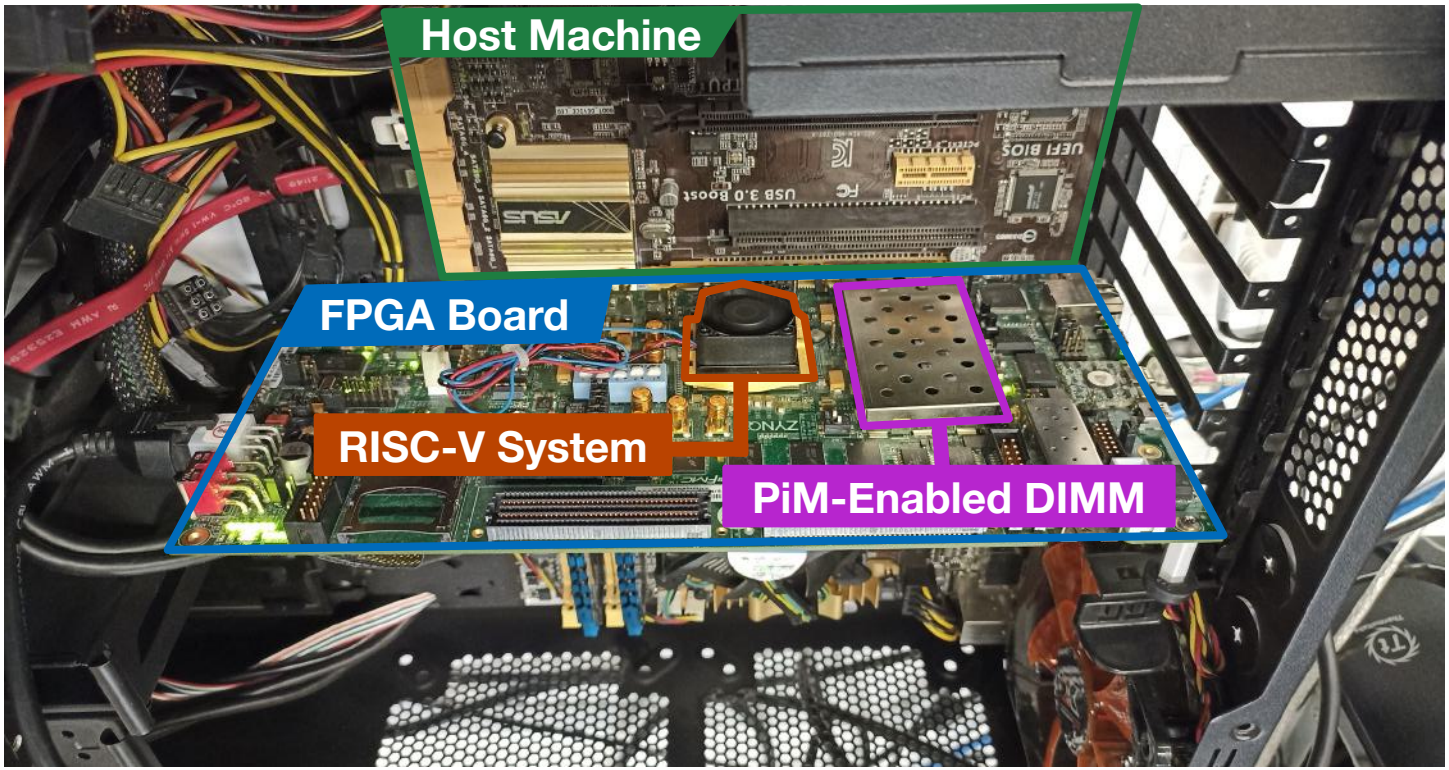
[§]ETH Zürich [†]TOBB ETÜ [*]BSC

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing-using-Memory Prototype



https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing-using-Memory Prototype

README.md

Building a PiDRAM Prototype

To build PiDRAM's prototype on Xilinx ZC706 boards, developers need to use the two sub-projects in this directory. `fpga-zynq` is a repository branched off of UCB-BAR's fpga-zynq repository. We use `fpga-zynq` to generate rocket chip designs that support end-to-end DRAM PuM execution. `controller-hardware` is where we keep the main Vivado project and Verilog sources for PiDRAM's memory controller and the top level system design.

Rebuilding Steps

1. Navigate into `fpga-zynq` and read the README file to understand the overall workflow of the repository
   - Follow the readme in `fpga-zynq/rocket-chip/riscv-tools` to install dependencies
2. Create the Verilog source of the rocket chip design using the `ZynqCopyFPGAConfig`
   - Navigate into zc706, then run `make rocket CONFIG=ZynqCopyFPGAConfig -j<number of cores>`
3. Copy the generated Verilog file (should be under zc706/src) and overwrite the same file in `controller-hardware/source/hdl/impl/rocket-chip`
4. Open the Vivado project in `controller-hardware/Vivado_Project` using Vivado 2016.2
5. Generate a bitstream
6. Copy the bitstream (system_top.bit) to `fpga-zynq/zc706`
7. Use the `./build_script.sh` to generate the new `boot.bin` under `fpga-images-zc706`, you can use this file to program the FPGA using the SD-Card
   - For details, follow the relevant instructions in `fpga-zynq/README.md`

You can run programs compiled with the RISC-V Toolchain supplied within the `fpga-zynq` repository. To install the toolchain, follow the instructions under `fpga-zynq/rocket-chip/riscv-tools`.

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx PHY IP we use in PiDRAM's memory controller due to licensing issues. We describe here how to regenerate them using Vivado 2016.2. First, you need to generate the IP RTL files:

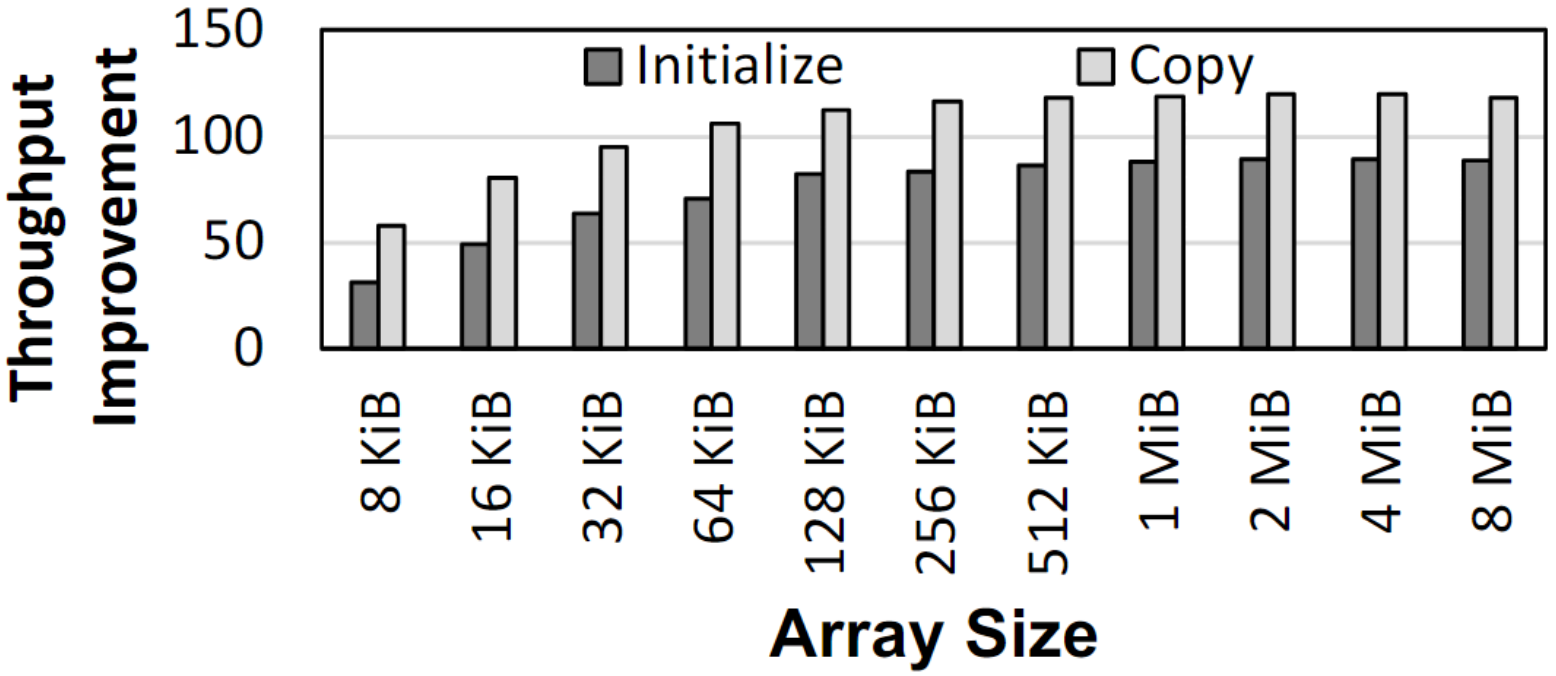
1- Open IP Catalog
2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Microbenchmark Copy/Initialization Throughput



In-DRAM Copy and Initialization improve throughput by 119x and 89x

More on PiDRAM

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
  "PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"
  ACM Transactions on Architecture and Code Optimization (***TACO***), March 2023.
  [arXiv version]
  Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (40 minutes)]
  [PiDRAM Source Code]

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

DRAM Chips Are Already (Quite) Capable!

- **Appears at HPCA 2024** https://arxiv.org/pdf/2402.18736.pdf

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

ETH Zürich

We experimentally demonstrate that COTS DRAM chips are capable of performing 1) functionally-complete *Boolean operations: NOT, NAND, and NOR and 2) many-input (i.e., more than two-input) AND and OR operations. We present an extensive characterization of* new *bulk bitwise operations in 256 off-the-shelf modern DDR4 DRAM chips. We evaluate the reliability of these operations using a metric called success rate: the fraction of correctly performed bitwise operations. Among our 19 new observations, we highlight four major results. First, we can perform the NOT operation on COTS DRAM chips with 98.37% success rate on average. Second, we can perform up to 16-input NAND, NOR, AND, and OR operations on COTS DRAM chips with high reliability (e.g., 16-input NAND, NOR, AND, and OR with average success rate of 94.94%, 95.87%, 94.94%, and 95.85%, respectively). Third, data pattern only slightly*

The Capability of COTS DRAM Chips

We **demonstrate** that **COTS DRAM chips:**

1 Can **copy one row into up to 31 other rows with >99.98% success rate**

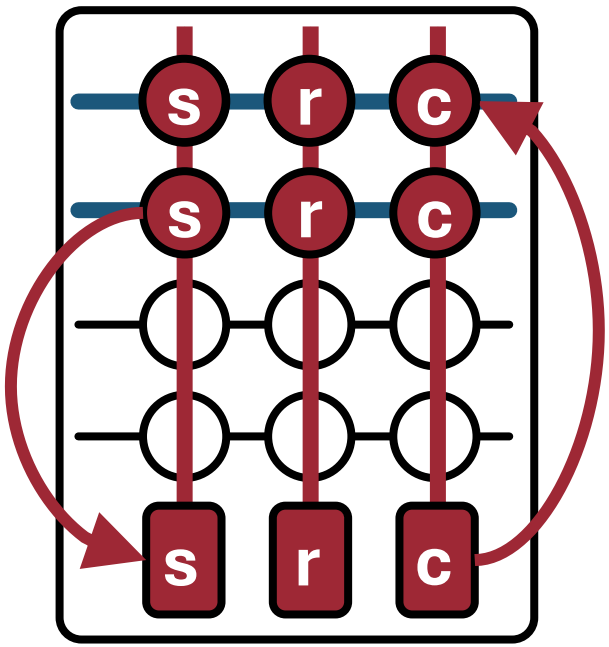
2 Can perform **NOT operation** with up to **32 output operands**

3 Can perform up to **16-input AND, NAND, OR, and NOR** operations

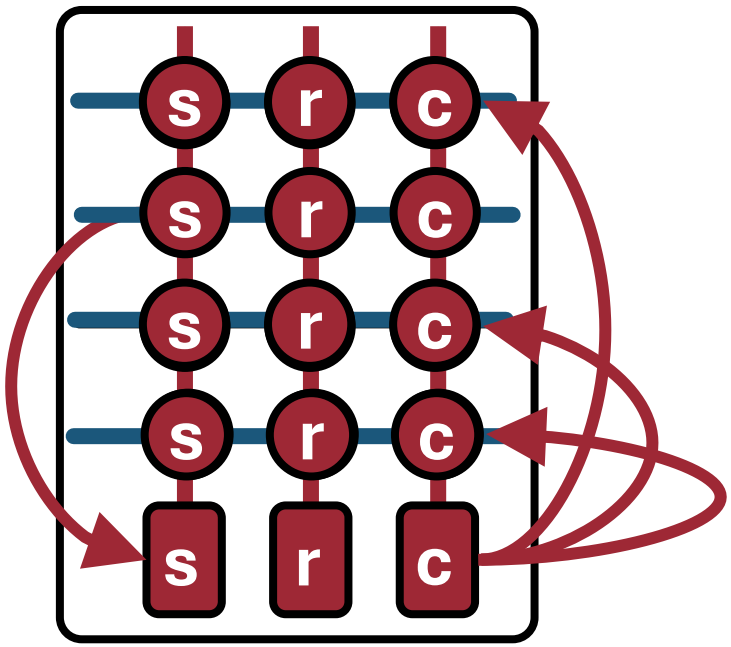
In-DRAM Multiple Row Copy (Multi-RowCopy)

Simultaneously activate many rows to
copy **one row's content** to **multiple destination rows**

RowClone

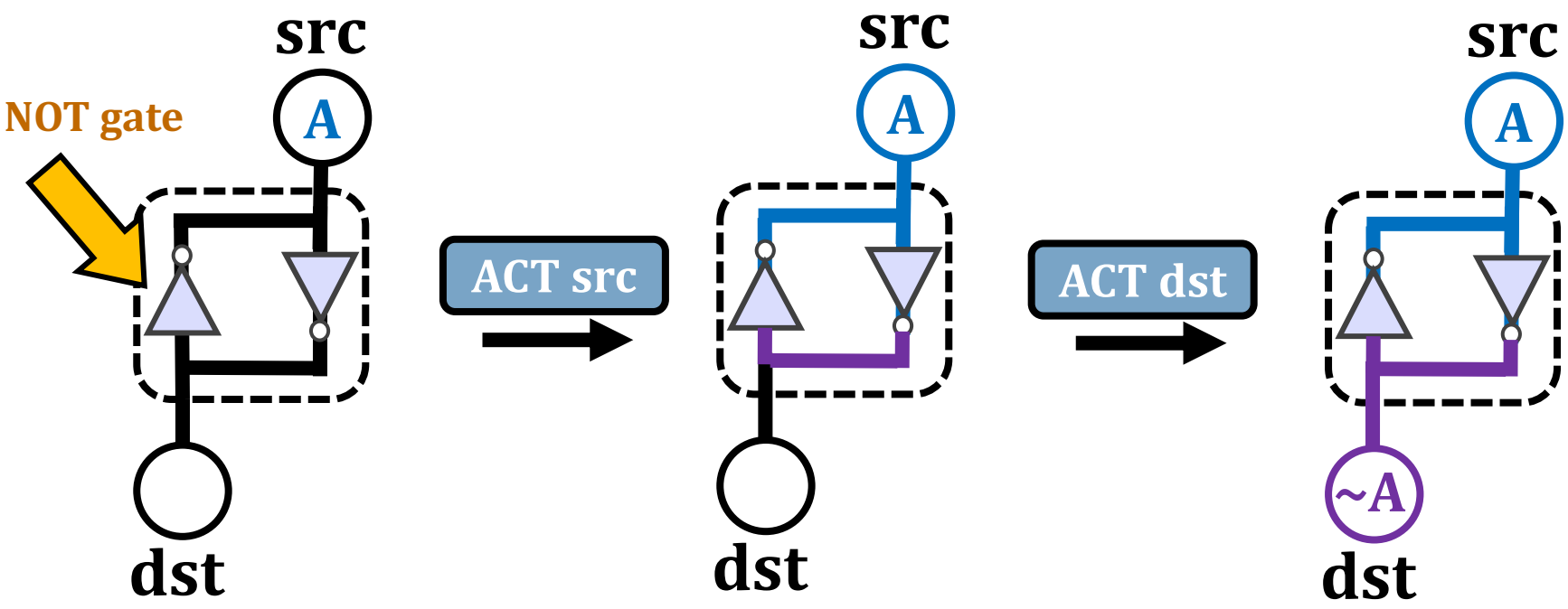


Multi-RowCopy



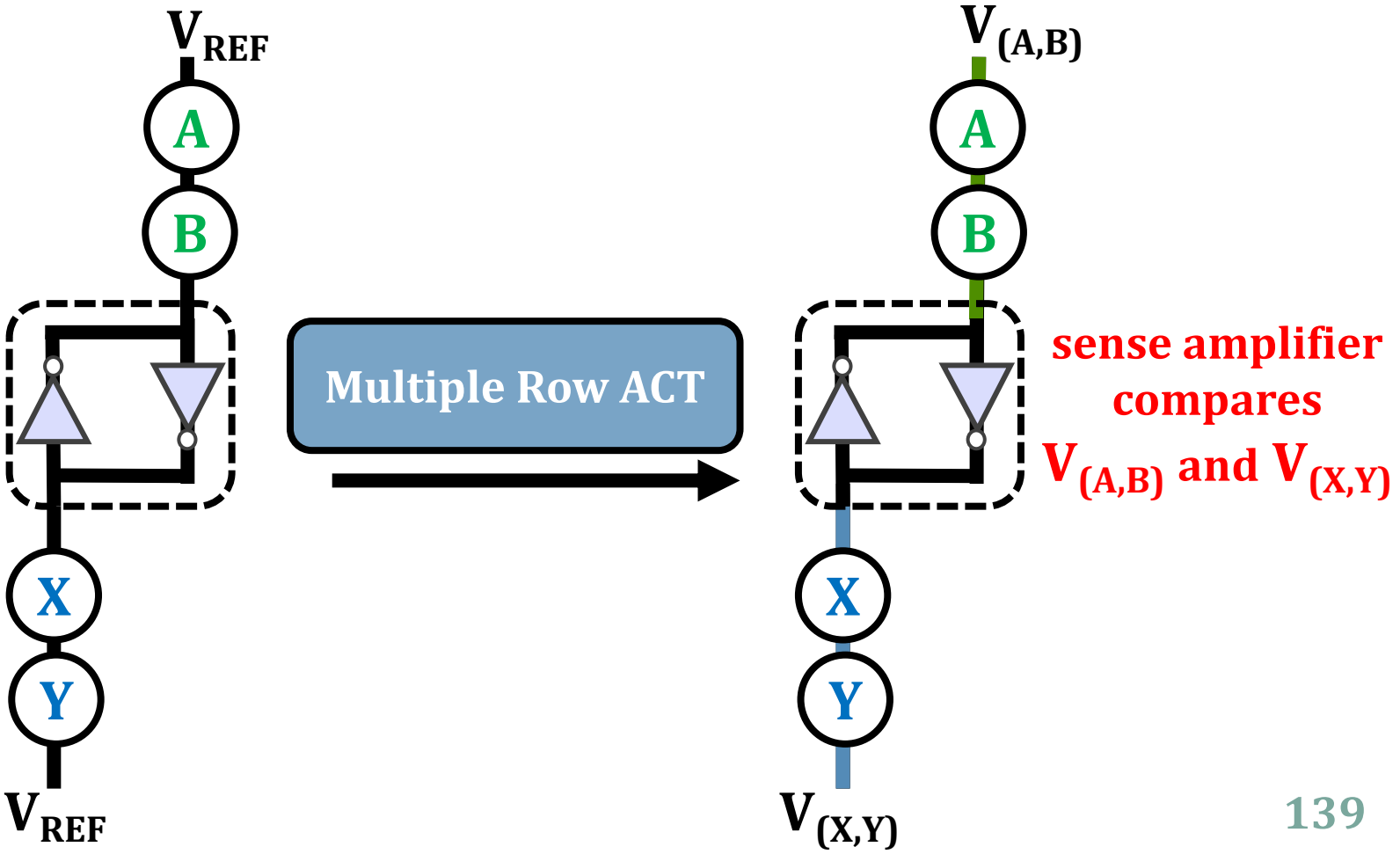
Key Idea: NOT Operation

Connect rows in neighboring subarrays
through **a NOT gate** by consecutively activating rows

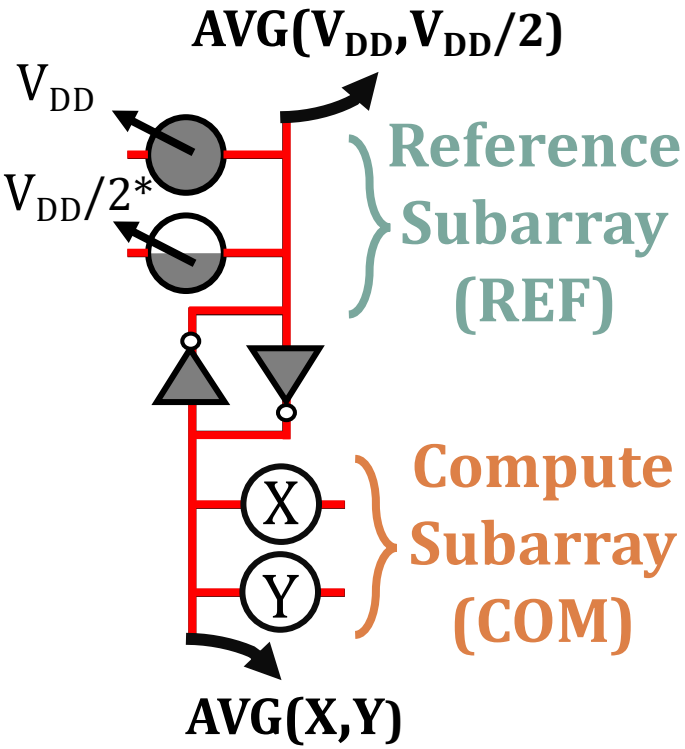


Key Idea: NAND, NOR, AND, OR

Manipulate the bitline voltage to express
a wide variety of functions using
simultaneous multi-row activation in neighboring subarrays



Two-Input AND and NAND Operations



V_{DD}=1 & GND = 0

X	Y	COM	REF
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0
		AND	NAND

*Gao et al., "FracDRAM: Fractional Values in Off-the-Shelf DRAM," in MICRO, 2022.

Many-Input AND, NAND, OR, and NOR Operations

We can express AND, NAND, OR, and NOR operations by carefully manipulating the **reference voltage**

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

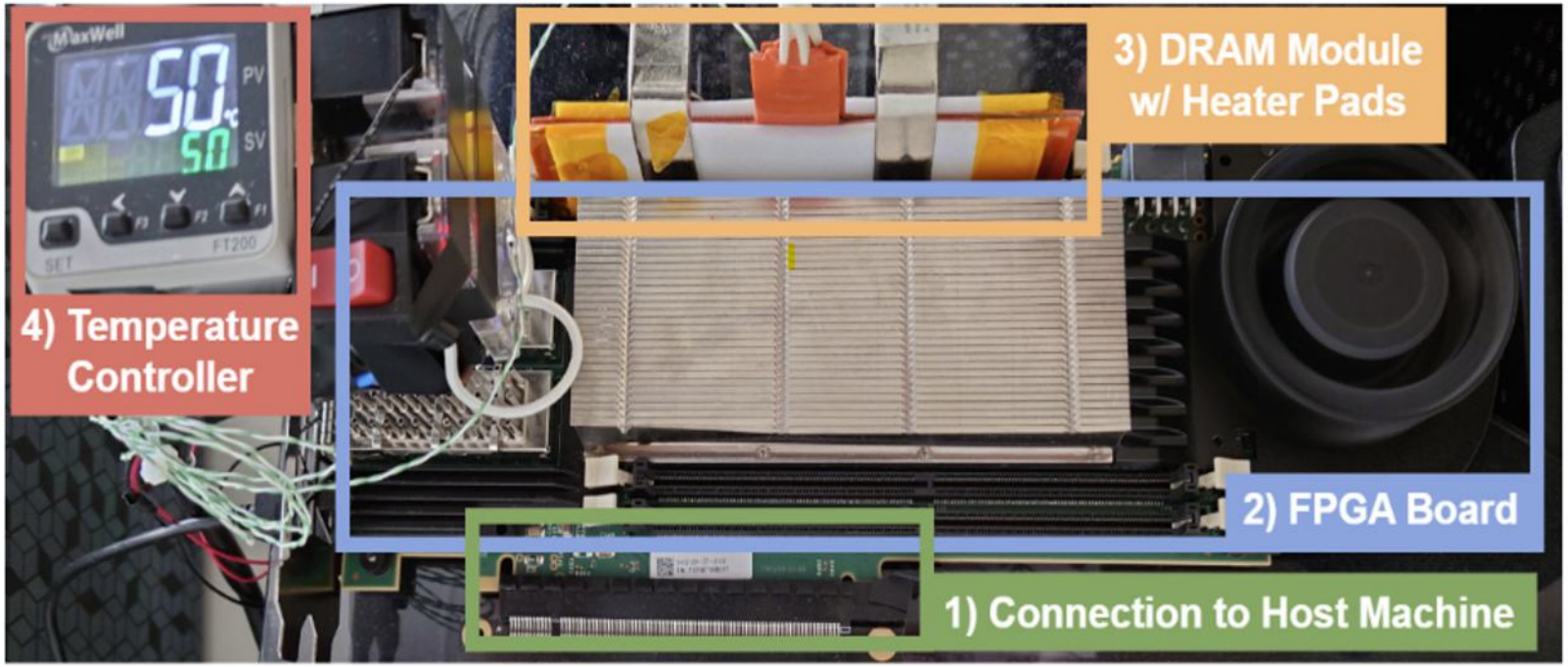
ETH Zürich

(More details in the paper)

https://arxiv.org/pdf/2402.18736.pdf

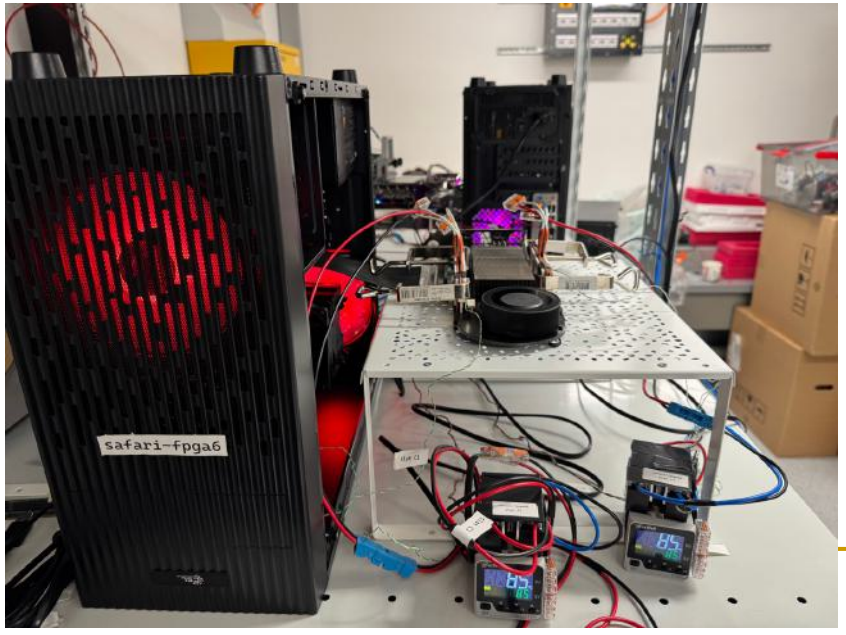
DRAM Testing Infrastructure

- Developed from DRAM Bender [Olgun+, TCAD'23]*
- Fine-grained control over DRAM commands, timings, and temperature



*Olgun et al., "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," TCAD, 2023.

Laboratory for Understanding Memory

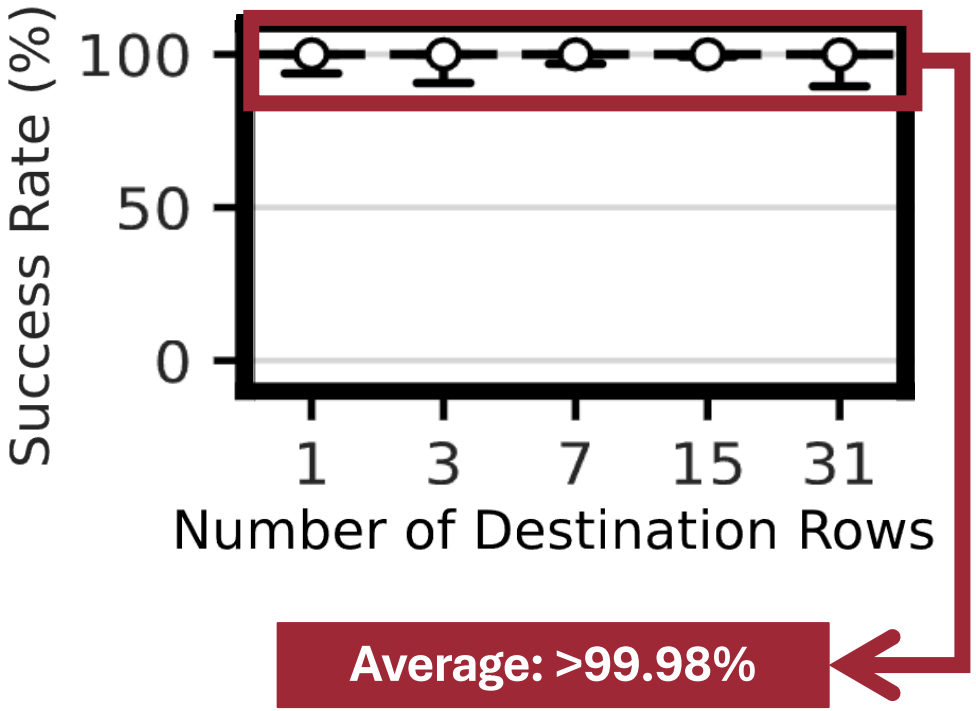


DRAM Chips Tested

- 256 DDR4 chips from two major DRAM manufacturers
- Covers different die revisions and chip densities

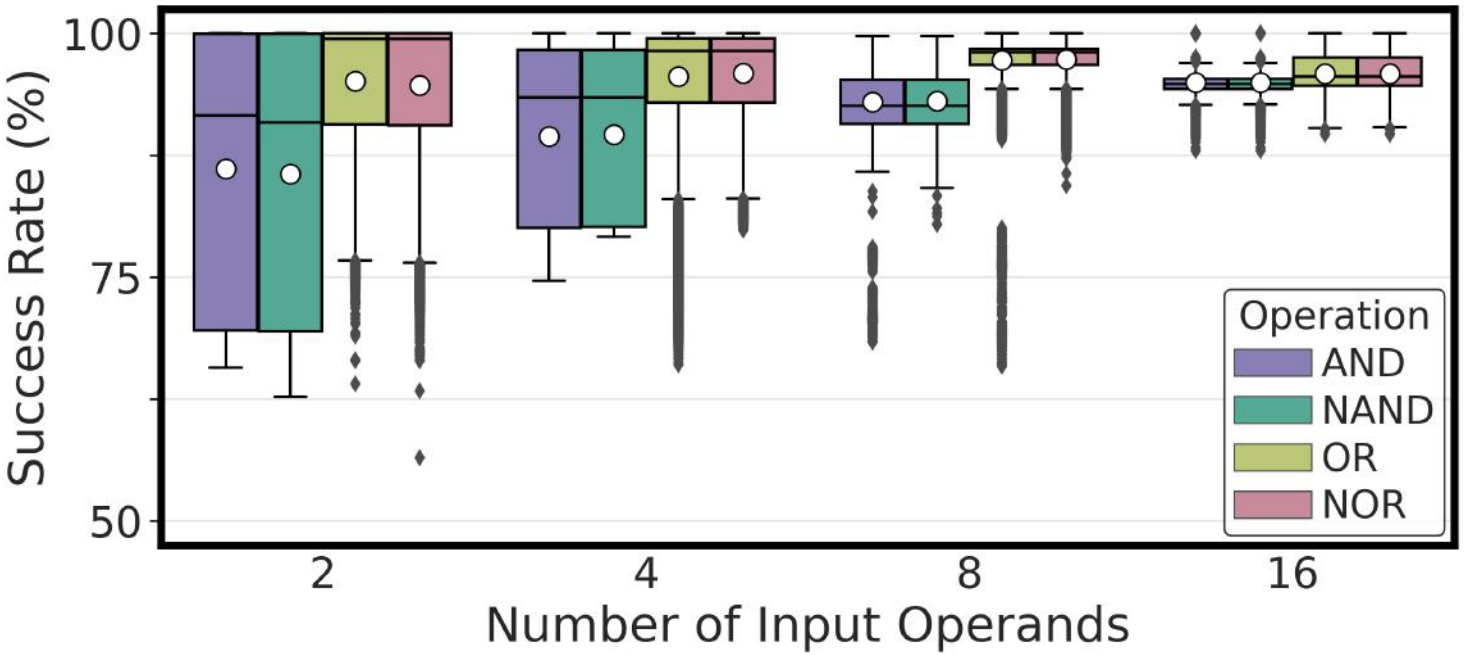
Chip Mfr.	#Modules (#Chips)	Die Rev.	Mfr. Date[a]	Chip Density	Chip Org.	Speed Rate
SK Hynix	9 (72)	M	N/A	4Gb	x8	2666MT/s
	5 (40)	A	N/A	4Gb	x8	2133MT/s
	1 (16)	A	N/A	8Gb	x8	2666MT/s
	1 (32)	A	18-14	4Gb	x4	2400MT/s
	1 (32)	A	16-49	8Gb	x4	2400MT/s
	1 (32)	M	16-22	8Gb	x4	2666MT/s
Samsung	1 (8)	F	21-02	4Gb	x8	2666MT/s
	2 (16)	D	21-10	8Gb	x8	2133MT/s
	1 (8)	A	22-12	8Gb	x8	3200MT/s

Robustness of Multi-RowCopy



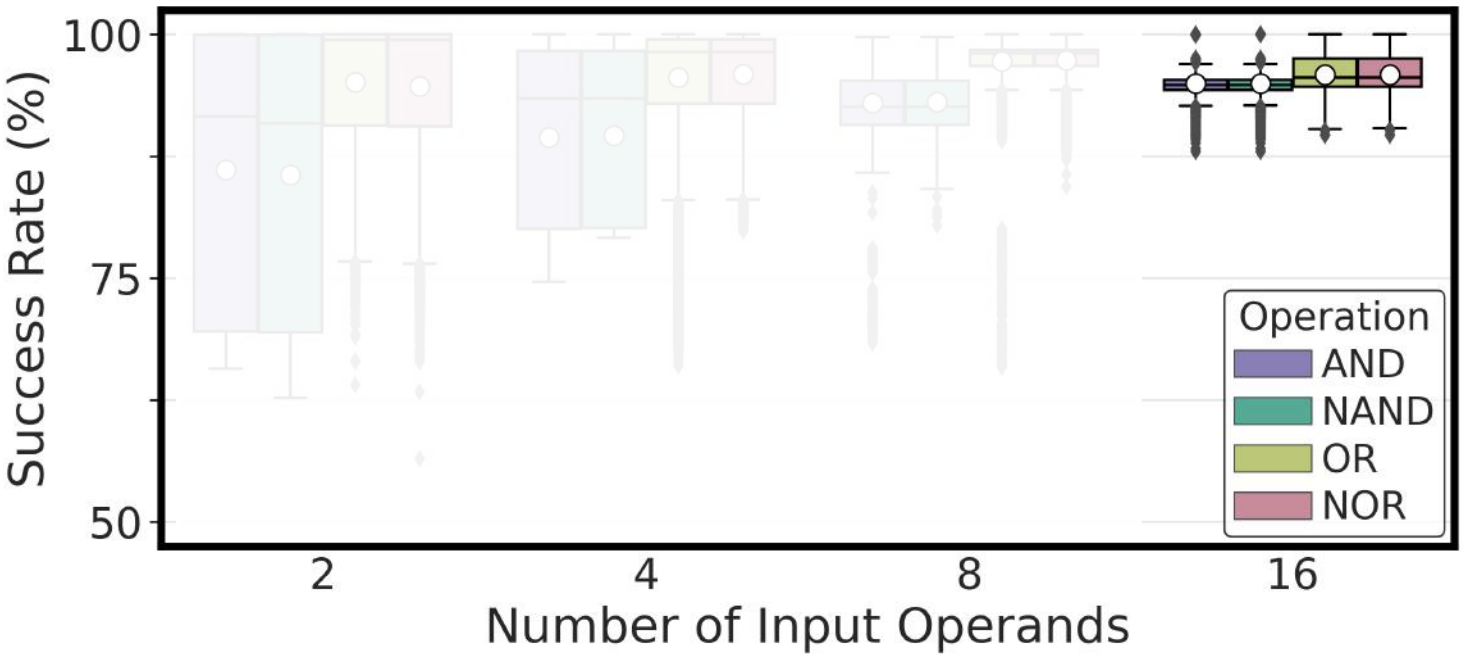
COTS DRAM chips can copy one row's content to up to 31 rows with a very high success rate

Performing AND, NAND, OR, and NOR



COTS DRAM chips can perform {2, 4, 8, 16}-input AND, NAND, OR, and NOR operations

Performing AND, NAND, OR, and NOR



COTS DRAM chips can perform 16-input AND, NAND, OR, and NOR operations with very high success rate (>94%)

More on Multi-Row Copy

- Ismail Emir Yuksel, Yahya Can Tugrul, F. Nisa Bostanci, Geraldo F. Oliveira, A. Giray Yaglikci, Ataberk Olgun, Melina Soysal, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,
  "Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis"
  *Proceedings of the 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Brisbane, Australia, June 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]
  [SiMRA-DRAM Source Code (Officially Artifact Evaluated with All Badges)]
  Officially artifact evaluated as both code and dataset available, reviewed and reproducible.

Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel[1] Yahya Can Tuğrul[1,2] F. Nisa Bostancı[1] Geraldo F. Oliveira[1]
A. Giray Yağlıkçı[1] Ataberk Olgun[1] Melina Soysal[1] Haocong Luo[1]
Juan Gómez-Luna[1] Mohammad Sadrosadati[1] Onur Mutlu[1]
[1]*ETH Zürich* [2]*TOBB University of Economics and Technology*

https://arxiv.org/pdf/2405.06081

More on Functionally-Complete DRAM

- Ismail Emir Yuksel, Yahya Can Tugrul, Ataberk Olgun, F. Nisa Bostanci, A. Giray Yaglikci, Geraldo F. Oliveira, Haocong Luo, Juan Gomez-Luna, Mohammad Sadrosadati, and Onur Mutlu,
  "Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis"
  *Proceedings of the 30th International Symposium on High-Performance Computer Architecture (**HPCA**)*, April 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]
  [FCDRAM Source Code]

Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

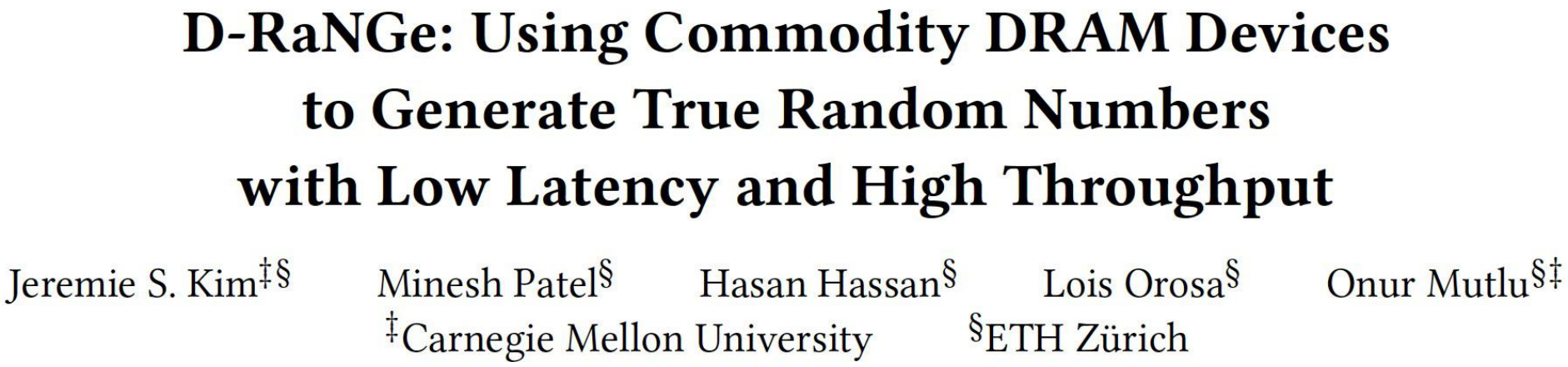
ETH Zürich

https://arxiv.org/pdf/2402.18736

What Else Can We Do Using Commodity Memories?

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.

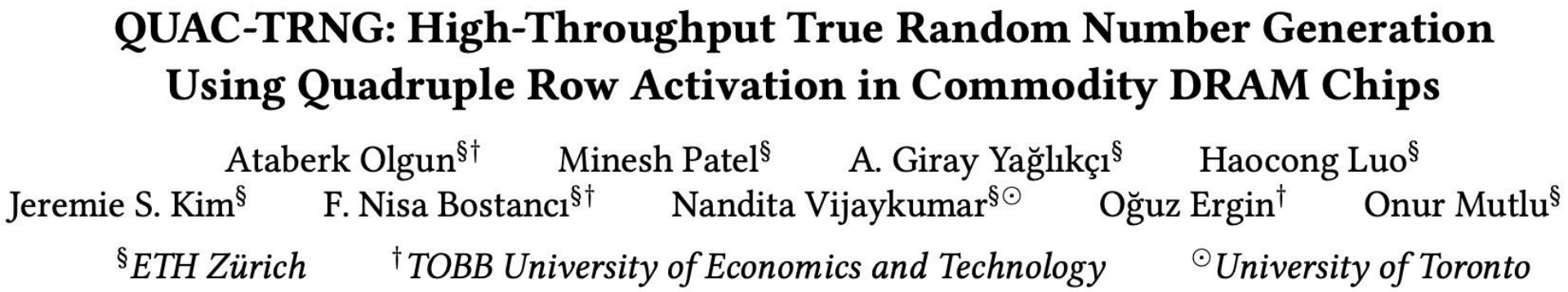


D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim[‡§] Minesh Patel[§] Hasan Hassan[§] Lois Orosa[§] Onur Mutlu[§‡]
[‡]Carnegie Mellon University [§]ETH Zürich

In-DRAM True Random Number Generation

- Ataberk Olgun, Minesh Patel, A. Giray Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu,
  "QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips"
  *Proceedings of the 48th International Symposium on Computer Architecture (**ISCA**)*, Virtual, June 2021.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Talk Video (25 minutes)]
  [SAFARI Live Seminar Video (1 hr 26 mins)]



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

Ataberk Olgun[§†] Minesh Patel[§] A. Giray Yağlıkçı[§] Haocong Luo[§]
Jeremie S. Kim[§] F. Nisa Bostancı[§†] Nandita Vijaykumar[§⊙] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [⊙]*University of Toronto*

In-DRAM TRNG: Recent Results

- N-row Activation
  - initialize cell values to sample random values in sense amplifiers

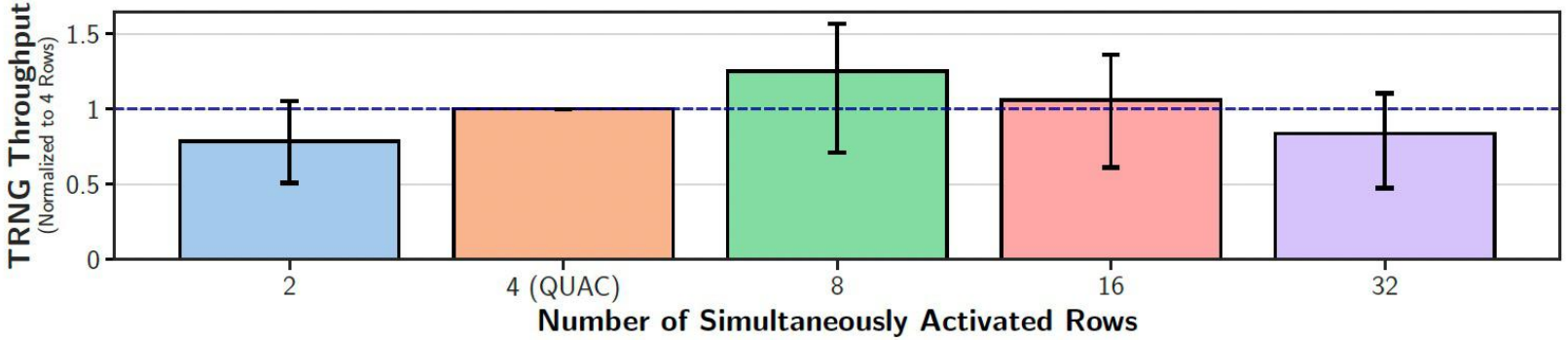


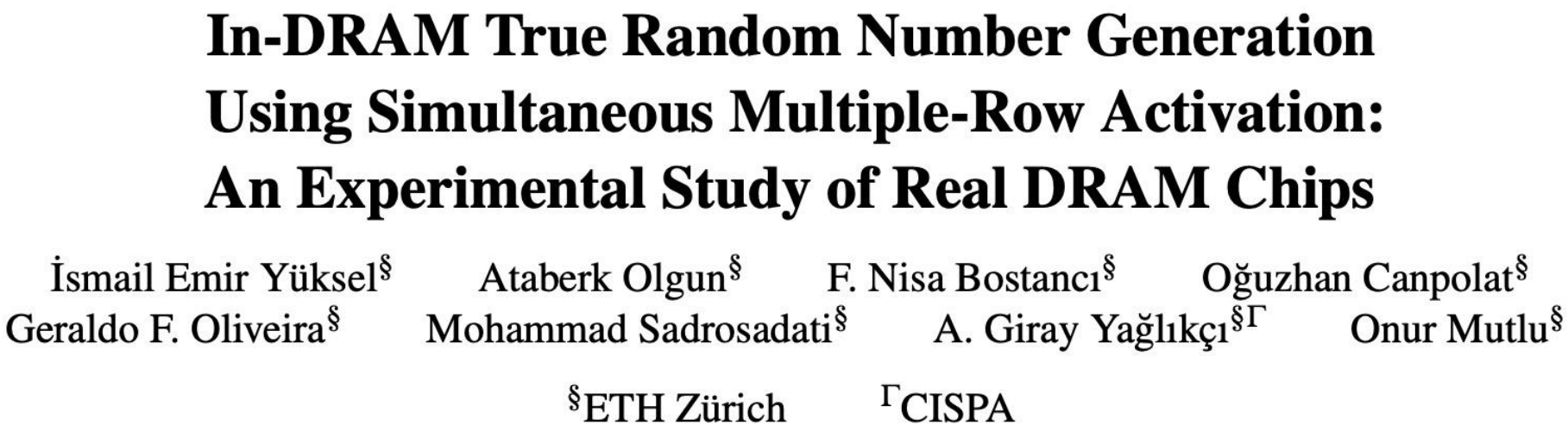
Fig. 11: **Throughput of generating true random numbers, as measured in 96 COTS DRAM chips using multiple-row activation, normalized to state-of-the-art DRAM-based TRNG, QUAC-TRNG (i.e., 4-row activation) [135].** Each error bar shows the range across all tested chips. We observe that random numbers that are generated with multiple-row activation and then post-processed with the SHA-256 function [221] pass *all* NIST STS tests [222], which means 2-, 4-, 8-, 16-, and 32-row activation generates high-quality true random bitstreams. On average, 8- and 16-row activation-based TRNG outperforms the state-of-the-art by 1.25× and 1.06×, respectively, while 2- and 32-row activation-based TRNG provides 0.69× and 0.84× the throughput of the state-of-the-art.

Mutlu+, "**Memory-Centric Computing: Recent Advances in Processing-in-DRAM**," IEDM 2024.

https://arxiv.org/pdf/2412.19275

In-DRAM True Random Number Generation

- Ismail Emir Yuksel, Ataberk Olgun, Nisa Bostancı, Oğuzhan Canpolat, Geraldo Francisco de Oliveira Junior, Mohammad Sadrosadati, Abdullah Giray Yağlıkçı, and Onur Mutlu, **"In-DRAM True Random Number Generation Using Simultaneous Multiple-Row Activation: An Experimental Study of Real DRAM Chips"**
  *Proceedings of the 43rd IEEE International Conference on Computer Design (**ICCD**)*, Dallas, TX, USA, November 2025.
  [Slides (pptx) (pdf)]



In-DRAM True Random Number Generation Using Simultaneous Multiple-Row Activation: An Experimental Study of Real DRAM Chips

İsmail Emir Yüksel[§] Ataberk Olgun[§] F. Nisa Bostancı[§] Oğuzhan Canpolat[§]
Geraldo F. Oliveira[§] Mohammad Sadrosadati[§] A. Giray Yağlıkçı[§Γ] Onur Mutlu[§]

[§]ETH Zürich [Γ]CISPA

https://arxiv.org/pdf/2510.20269

In-DRAM True Random Number Generation

- F. Nisa Bostanci, Ataberk Olgun, Lois Orosa, A. Giray Yaglikci, Jeremie S. Kim, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
  "DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators"
  *Proceedings of the 28th International Symposium on High-Performance Computer Architecture (**HPCA**)*, Virtual, April 2022.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]

DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators

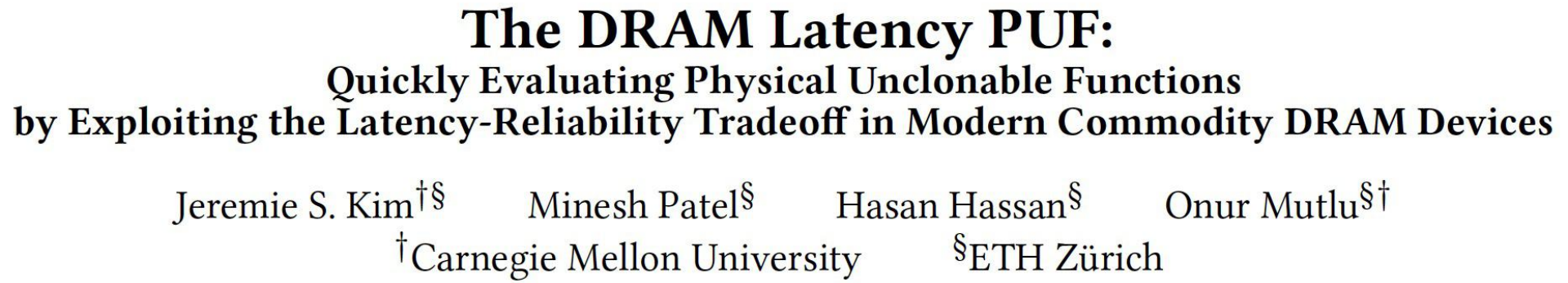
F. Nisa Bostancı[†§] Ataberk Olgun[†§] Lois Orosa[§] A. Giray Yağlıkçı[§]
Jeremie S. Kim[§] Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[†]*TOBB University of Economics and Technology* [§]*ETH Zürich*

https://arxiv.org/pdf/2201.01385.pdf

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[†§] Minesh Patel[§] Hasan Hassan[§] Onur Mutlu[§†]
[†]Carnegie Mellon University [§]ETH Zürich

In-DRAM Lookup-Table Based Execution

João Dinis Ferreira, Gabriel Falcao, Juan Gómez-Luna, Mohammed Alser, Lois Orosa, Mohammad Sadrosadati, Jeremie S. Kim, Geraldo F. Oliveira, Taha Shahroodi, Anant Nori, and Onur Mutlu,
"pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables"
*Proceedings of the 55th International Symposium on Microarchitecture (**MICRO**)*, Chicago, IL, USA, October 2022.
[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (26 minutes)]
[arXiv version]
[Source Code (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, reusable and reproducible.

pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables

João Dinis Ferreira[§] Gabriel Falcao[†] Juan Gómez-Luna[§] Mohammed Alser[§]
Lois Orosa[§▽] Mohammad Sadrosadati[§] Jeremie S. Kim[§] Geraldo F. Oliveira[§]
Taha Shahroodi[‡] Anant Nori[⋆] Onur Mutlu[§]
[§]*ETH Zürich* [†]*IT, University of Coimbra* [▽]*Galicia Supercomputing Center* [‡]*TU Delft* [⋆]*Intel*

https://arxiv.org/pdf/2104.07699.pdf

In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu,
  "Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory"
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***), Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (44 minutes)]
  [arXiv version]

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/pdf/2209.05566.pdf

In-Flash Homomorphic Encryption

- Mayank Kabra, Rakesh Nadig, Harshita Gupta, Rahul Bera, Manos Frouzakis, Vamanan Arulchelvan, Yu Liang, Haiyu Mao, Mohammad Sadrosadati, and Onur Mutlu, **"CIPHERMATCH: Accelerating Homomorphic Encryption based String Matching via Memory-Efficient Data Packing and In-Flash Processing"** *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Rotterdam, Netherlands, April 2025.
  [Slides (pptx) (pdf)]

CIPHERMATCH: Accelerating Homomorphic Encryption-Based String Matching via Memory-Efficient Data Packing and In-Flash Processing

Mayank Kabra† Rakesh Nadig† Harshita Gupta† Rahul Bera† Manos Frouzakis†
Vamanan Arulchelvan† Yu Liang† Haiyu Mao‡ Mohammad Sadrosadati† Onur Mutlu†
ETH Zurich† *King's College London‡*

https://arxiv.org/pdf/2503.08968

Processing in Memory: Two Types

1. Processing **near** Memory
2. Processing **using** Memory

PIM Review and Open Problems

A Modern Primer on Processing-In-Memory

Onur Mutlu[a], Saugata Ghose[b], Juan Gómez-Luna[c], Rachata Ausavarungnirun[d],
Mohammad Sadrosadati[a], Geraldo F. Oliveira[a]

SAFARI Research Group

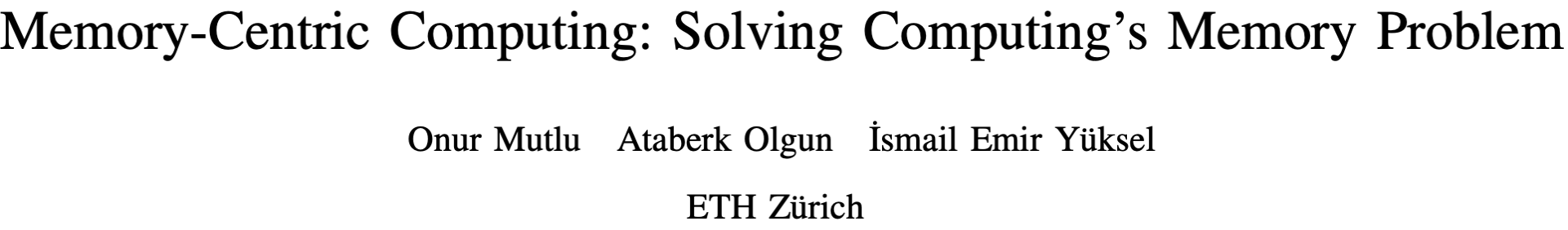
[a] *ETH Zürich*
[b] *University of Illinois Urbana-Champaign*
[c] *NVIDIA Research*
[d] *MangoBoost Inc.*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, Rachata Ausavarungnirun, Mohammad Sadrosadati, and Geraldo F. Oliveira,
"A Modern Primer on Processing in Memory"
Invited Book Chapter in ***Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann***, Springer, 2022.

https://arxiv.org/pdf/2012.03112

A Recent Short Paper [IMW 2025]

- Onur Mutlu, Ataberk Olgun, and İsmail Emir Yüksel, **"Memory-Centric Computing: Solving Computing's Memory Problem"**
  *Invited Paper in Proceedings of the 17th IEEE International Memory Workshop (**IMW**)*, Monterey, CA, USA, May 2025.
  [Slides (pptx) (pdf)]


Memory-Centric Computing: Solving Computing's Memory Problem

Onur Mutlu Ataberk Olgun İsmail Emir Yüksel

ETH Zürich

https://www.arxiv.org/pdf/2505.00458

Eliminating the Adoption Barriers

How to Enable Adoption of Processing in Memory

Potential Barriers to Adoption of PIM

1. **Applications** & **software** for PIM

2. Ease of **programming** (interfaces and compiler/HW support)

3. **System** and **security** support: coherence, synchronization, virtual memory, isolation, communication interfaces, …

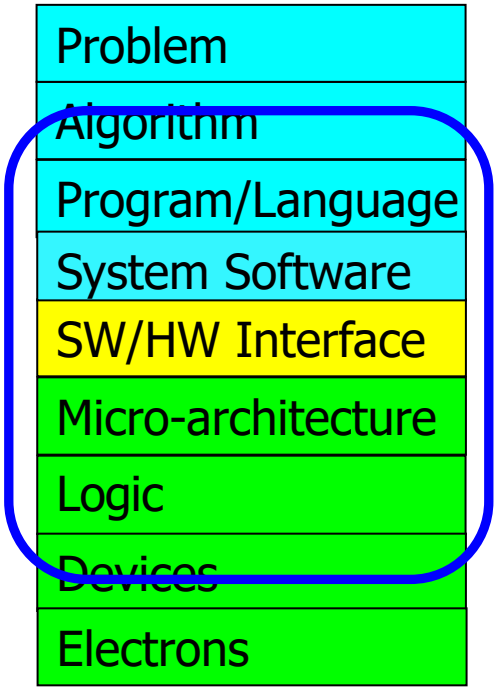
4. **Runtime** and **compilation** systems for adaptive scheduling, data mapping, access/sharing control, …

5. **Infrastructures** to assess benefits and feasibility

All can be solved with change of mindset

We Need to Revisit the Entire Stack

- With a **memory-centric mindset**



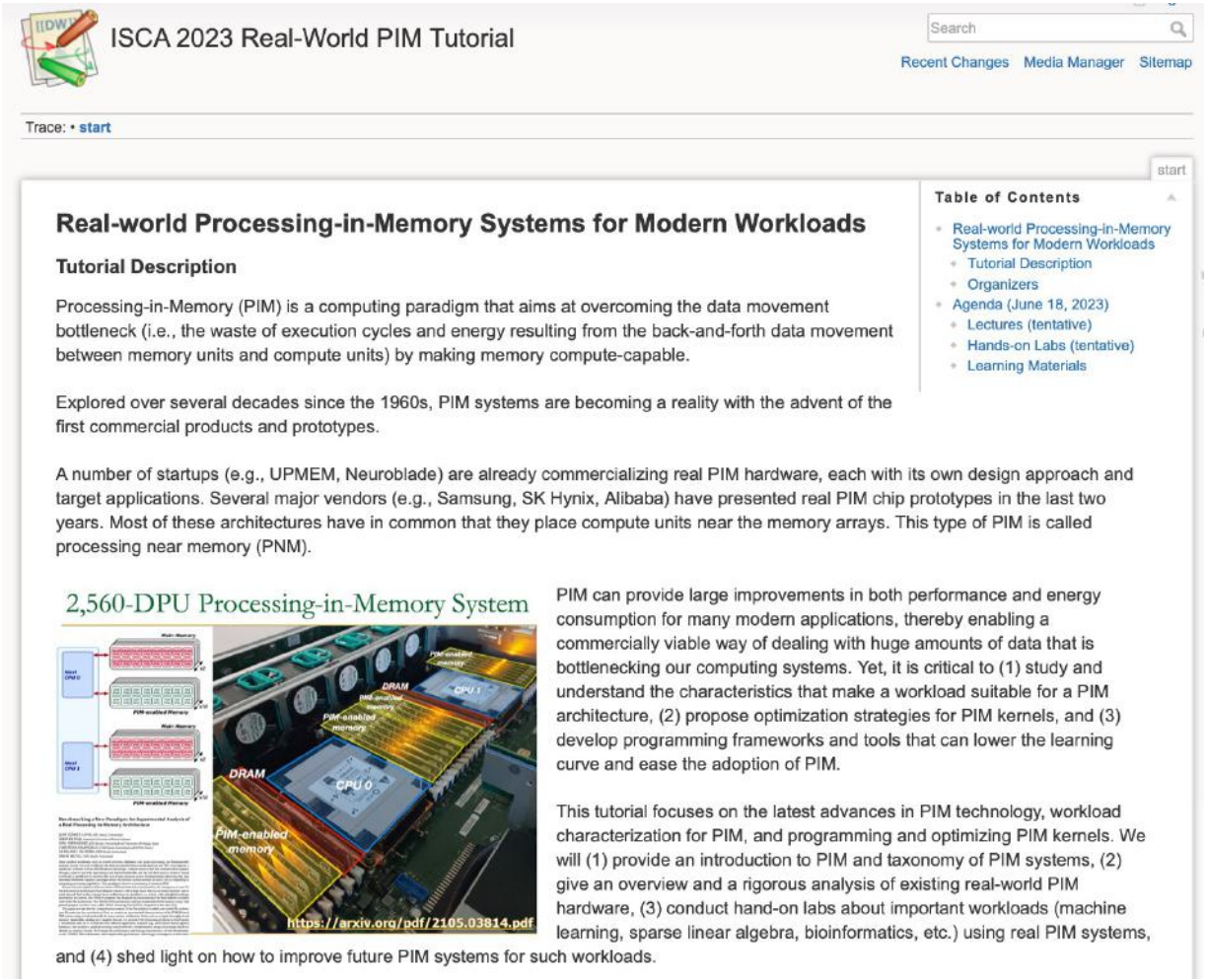
We can get there step by step

A Very Recent PhD Thesis

- https://safari.ethz.ch/geraldo-francisco-de-oliveira-junior-successfully-defends-his-phd/

PIM Tutorials [MICRO'23, ISCA'23, ASPLOS'23, HPCA'23, ISCA'24]

- Lectures + Hands-on labs + Invited talks



https://www.youtube.com/live/GIb5EgSrWk0

https://events.safari.ethz.ch/isca-pim-tutorial/

PIM Tutorial November 2024 Edition

MICRO 2024 - Tutorial on Memory-Centric Computing Systems

Saturday, November 2nd, Austin, Texas, USA

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/micro24-memorycentric-tutorial/

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities

https://www.youtube.com/watch?v=KV2MXvcBgb0

https://events.safari.ethz.ch/micro24-memorycentric-tutorial/

PIM Tutorial @ PPoPP/HPCA/CGO/CC

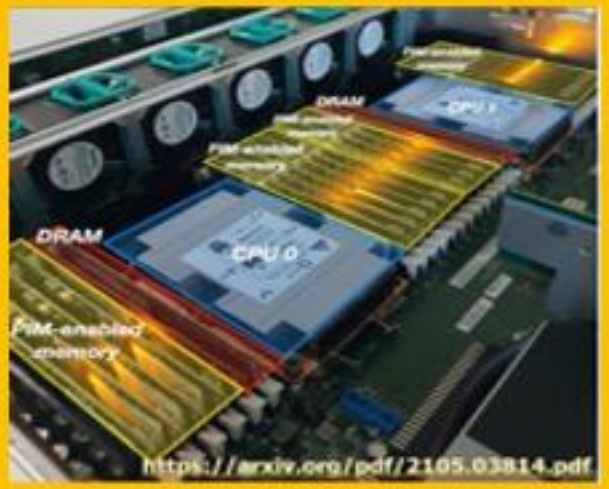
PPoPP 2025 - Tutorial on Memory-Centric Computing Systems

March 1st, Las Vegas, Nevada, USA

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/ppopp25-memorycentric-tutorial/

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities



https://www.youtube.com/live/NkDY6osus6g

https://events.safari.ethz.ch/ppopp25-memorycentric-tutorial/

PIM Tutorial/Workshop @ ASPLOS 2025

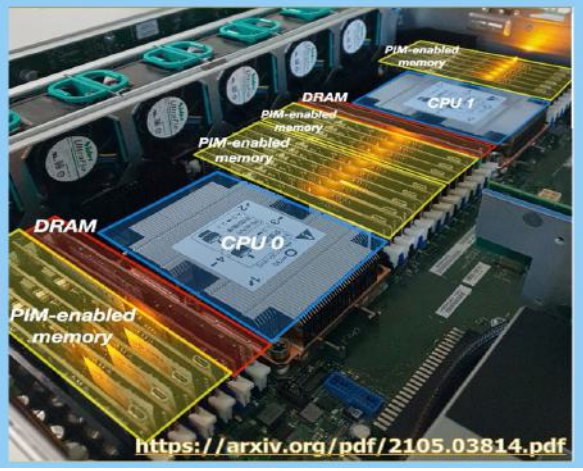
ASPLOS 2025 – 1st Workshop on Memory-Centric Computing Systems

Sunday, March 30th, Rotterdam, The Netherlands

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/asplos25-MCCSys/doku.php

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities



https://events.safari.ethz.ch/asplos25-MCCSys/doku.php

PIM Tutorial/Workshop @ ICS 2025

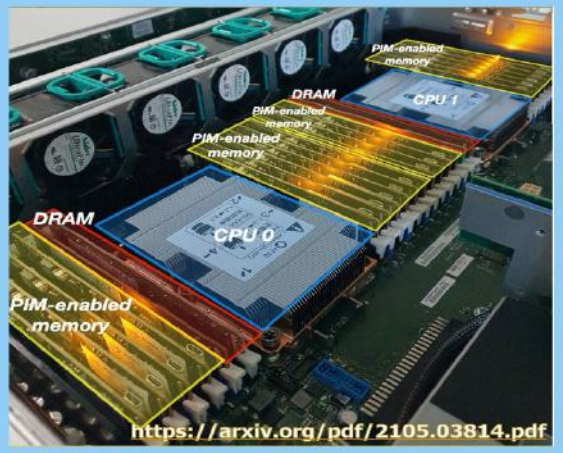
ICS 2025 – 2nd Workshop on Memory-Centric Computing Systems

Sunday, June 8th, Salt Lake City, USA

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/ics25-MCCSys/doku.php

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities



https://events.safari.ethz.ch/ics25-MCCSys/doku.php

PIM Tutorial/Workshop @ ISCA 2025

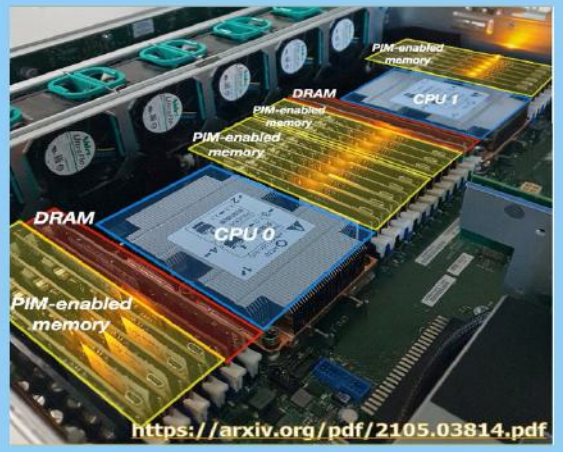
ISCA 2025 – 3rd Workshop on Memory-Centric Computing Systems

Saturday, 21st June, 2025, Tokyo, Japan

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/isca25-MCCSys/doku.php

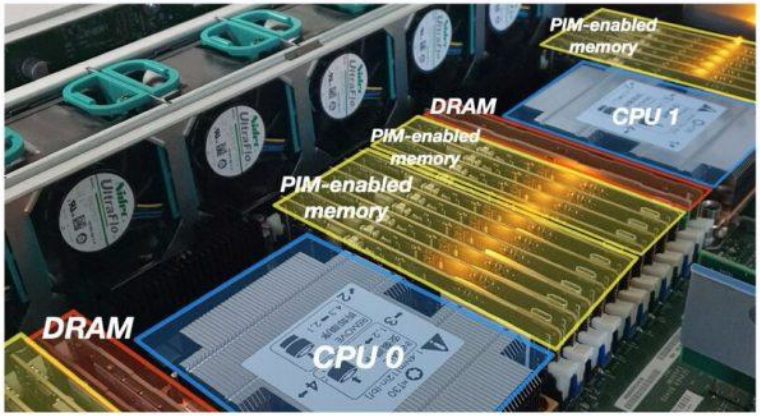
- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities



https://events.safari.ethz.ch/isca25-MCCSys/doku.php

4th MCCSys Workshop @ HPCA 2026

4th Workshop on Memory-Centric Computing Systems



MCCSys

A full-day workshop @ HPCA 2026, Sydney, Australia
1st February 2026 (Sunday)

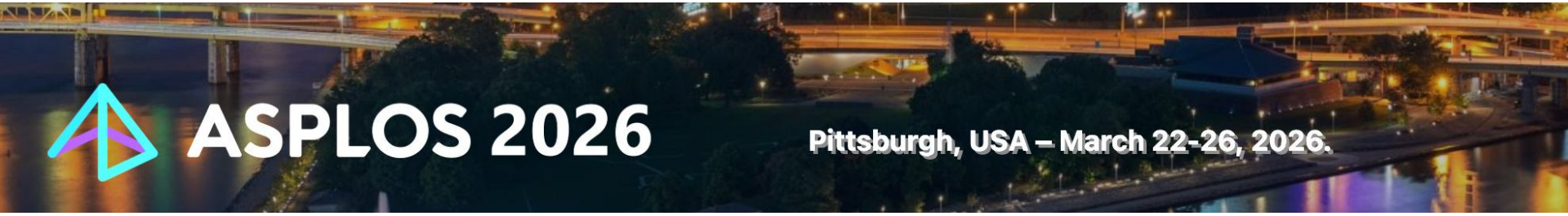
Organizers: Ismail Yuksel, Nisa Bostanci, Ataberk Olgun, Dr. Zhiheng Yue, Dr. Mohammad Sadrosadati, Dr. Geraldo F. Oliveira, Prof. Onur Mutlu

More information & call for presentation:

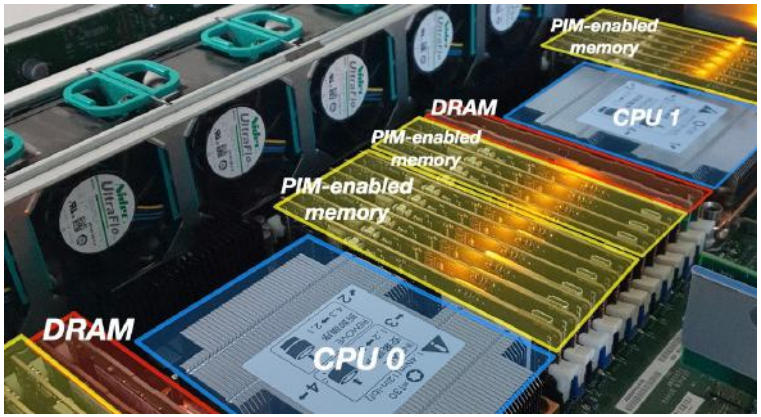
- Overview of PIM & PIM Taxonomy
- PIM in Memory & Storage
- Infrastructures for PIM Research
- Real-World PnM Systems
- PuM for Bulk Bitwise Operations
- Programming Techniques & Tools
- Research Challenges & Opportunities

https://events.safari.ethz.ch/hpca26-MCCSys/

https://events.safari.ethz.ch/hpca26-MCCSys/
https://www.youtube.com/watch?v=3wtg4sTp1qw



5th Workshop on Memory-Centric Computing Systems



MCCSys

A half-day workshop @ ASPLOS 2026
23rd March 2026 (Monday)

Organizers: Ismail Yuksel, Nisa Bostanci, Ataberk Olgun, Dr. Zhiheng Yue, Dr. Mohammad Sadrosadati, Dr. Geraldo F. Oliveira, Prof. Onur Mutlu

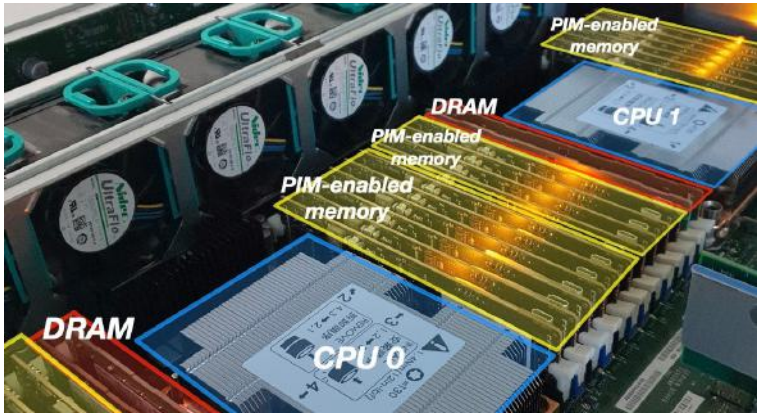
More information & call for presentations:

- **Overview of PIM & PIM Taxonomy**
- **PIM in Memory & Storage**
- **Infrastructures for PIM Research**
- **Real-World PnM Systems**
- **PuM for Bulk Bitwise Operations**
- **Programming Techniques & Tools**
- **Research Challenges & Opportunities**

SAFARI

https://events.safari.ethz.ch/asplos26-MCCSys/

6th Workshop on Memory-Centric Computing Systems



MCCSys

A full-day workshop @ ICS 2026
6th July 2026 (Monday)

Organizers: Ismail Yuksel, Nisa Bostanci, Ataberk Olgun, Dr. Zhiheng Yue, Dr. Mohammad Sadrosadati, Dr. Geraldo F. Oliveira, Prof. Onur Mutlu

More information & call for papers:

- **Overview of PIM & PIM Taxonomy**
- **PIM in Memory & Storage**
- **Infrastructures for PIM Research**
- **Real-World PnM Systems**
- **PuM for Bulk Bitwise Operations**
- **Programming Techniques & Tools**
- **Research Challenges & Opportunities**

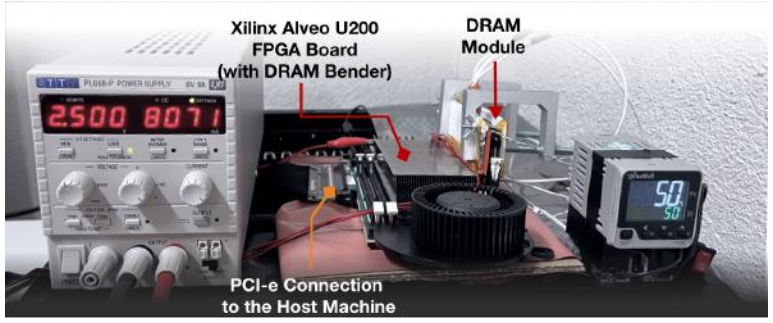
SAFARI

https://events.safari.ethz.ch/ics26-mccsys/ (Coming soon)

Tutorial on DRAM Bender & Ramulator

Tutorial on Ramulator & DRAM Bender:

Cutting-Edge Infrastructures for Real and Future Memory System Evaluation



R&DB:

A half-day tutorial @ ASPLOS 2026, Pittsburgh, USA
22nd March 2026 (Saturday)

Organizers: Nisa Bostanci, Ataberk Olgun, Ismail Yuksel, Haocong Luo, Prof. Onur Mutlu

- *Evaluating memory performance, emerging memory technologies, and architectural mechanisms*
- *Real DRAM device characterization studies*
- *New features, extensions, and enhancements*
- *Cross-layer research that combines memory system simulation with real-chip DRAM characterization*

More information & call for presentation:
https://events.safari.ethz.ch/asplos26-ramulator-drambender/

More in My MICRO 2025 Keynote Talk

Can We Do Better? - Keynote Talk at MICRO 2025 - Prof. Onur Mutlu

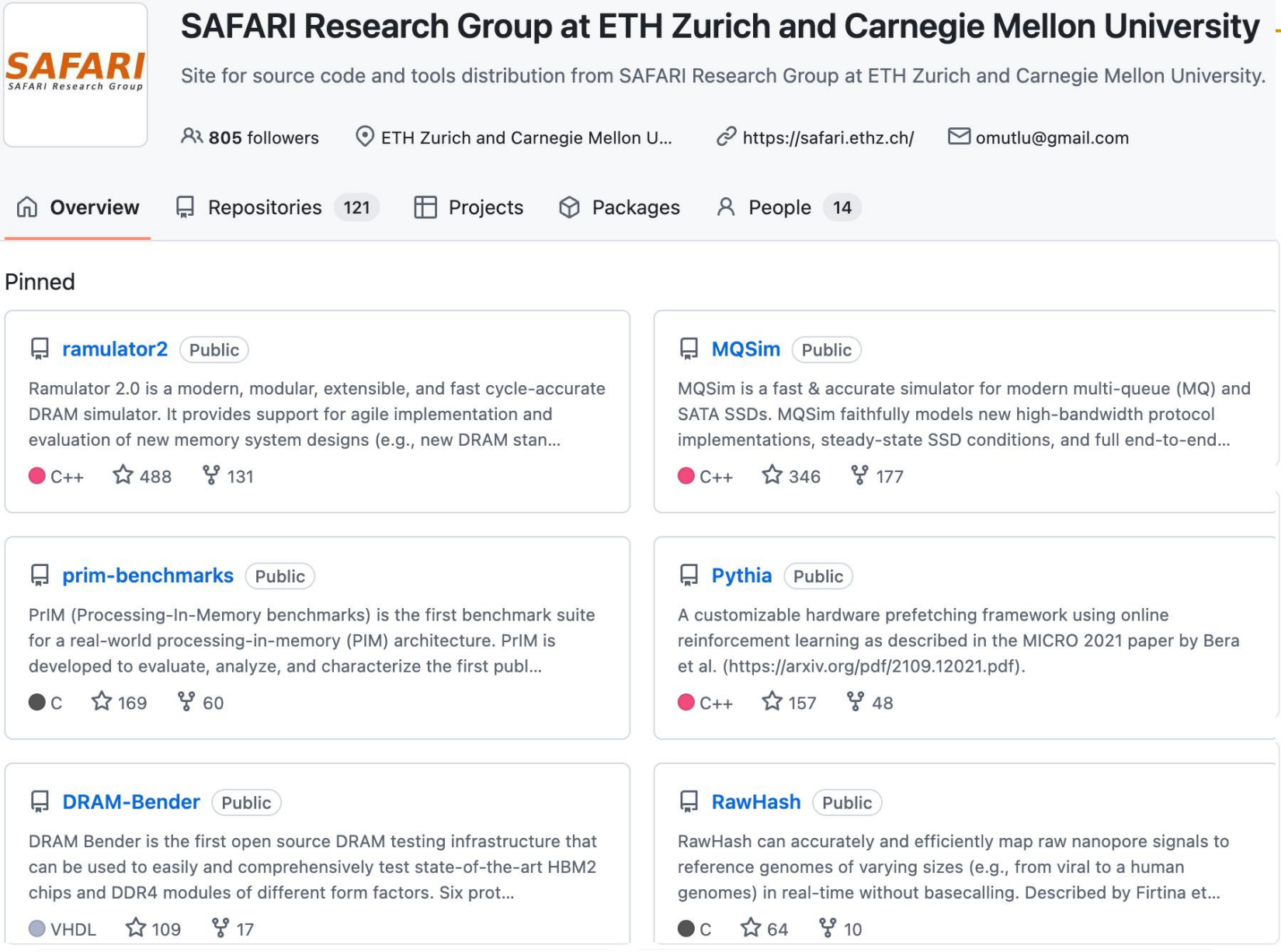
Onur Mutlu Lectures
58.5K subscribers

72 | Share | Save | Clip | Download | ...

2,752 views Streamed live on Oct 21, 2025
Title: Can We Do Better?
Presenter: Prof. Onur Mutlu (https://people.inf.ethz.ch/omutlu/)
Date and Time: October 21, 2025, 08:00 AM (KST)

https://www.youtube.com/watch?v=Mk6j9zP3s9E

Open Source Tools: SAFARI GitHub



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

Referenced Papers, Talks, Artifacts

- All are available at

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

  https://www.youtube.com/onurmutlulectures

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

Concluding Remarks

Summary

- Changing a paradigm is not easy
  - Copernican revolution did not happen overnight: ~1500-~1700
  - It took Brahe, Kepler, Galileo, and Newton, in addition to many
  - Longstanding and entrenched dogma had to be overcome
- We need to change the processor-centric paradigm
  - This can enable many benefits, not all quantifiable or predictable
  - It will **not** happen overnight
- Memory → **combined computation and storage substrate**
  - Can lead to **orders-of-magnitude energy & perf** improvements
  - **Unmodified DRAM chips are already capable of computation**
- We have a lot more exciting research to do & efficiently enable
  - We need to do research & design across the computing stack
  - With a proper mindset and infrastructure shift

Source: http://spectrum.ieee.org/image/MjYzMzAyMg.jpeg


Hilton

A Leading Architect

Eero Saarinen

45 languages

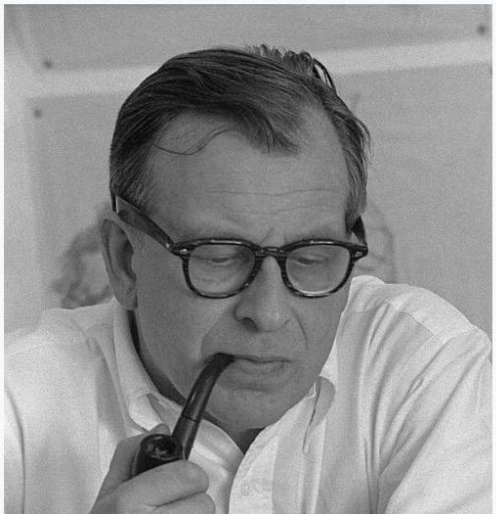
Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

Eero Saarinen (/ˈɛɪroʊ ˈsɑːrɪnən, ˈɛəroʊ -/, Finnish: [ˈeːro ˈsɑːrinen]; August 20, 1910 – September 1, 1961) was a Finnish-American architect and industrial designer. Saarinen's work includes the General Motors Technical Center; the Dulles International Airport Main Terminal; the TWA Flight Center at John F. Kennedy International Airport; the Vivian Beaumont Theater at Lincoln Center; the Gateway Arch; and the IBM Thomas J. Watson Research Center. During his career, Saarinen was elected a Fellow of the American Institute of Architects and served on the National Institute of Arts and Letters.

Born in Hvitträsk, Finland, he was the son of Finnish architect Eliel Saarinen, and immigrated to the United States as a teenager. Saarinen grew up in Bloomfield Hills, Michigan, studying at the Cranbrook Academy of Art, where his father taught. Saarinen

Eero Saarinen



https://en.wikipedia.org/wiki/Eero_Saarinen

Perhaps More Importantly: An Enabler

Saarinen served on the jury for the Sydney Opera House commission in 1957 and was crucial in the selection of the now internationally known design by Jørn Utzon.[9] A jury which did not include Saarinen had discarded Utzon's design in the first round; Saarinen reviewed the discarded designs, recognized a quality in Utzon's design, and ultimately assured the commission of Utzon.[9]

The winner, announced in Sydney on 29 January 1957,[29] was Danish architect Jørn Utzon. Saarinen selected Utzon's distinctive design from a final cut of 30 rejects.[30] Utzon's design was inspired by natural shapes, most notably those of bird wings, clouds, shells, walnuts, rivers and palm leaves.[31]

https://en.wikipedia.org/wiki/Eero_Saarinen

Funding Acknowledgments

- Alibaba, AMD, ASML, Bytedance, Google, Facebook, Futurewei, Hi-Silicon, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware, Xilinx
- NSF
- NIH
- GSRC
- SRC
- CyLab
- EFCL
- SNSF
- HK ACCESS
- EU Horizon

Thank you!

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

Think BIG, Aim HIGH!

https://safari.ethz.ch

SAFARI Introduction & Research

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

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

SAFARI https://www.youtube.com/watch?v=mV2OuB2djEs

SAFARI Newsletter July 2024 Edition

- **https://safari.ethz.ch/safari-newsletter-july-2024/**

SAFARI Newsletter December 2025 Edition

- **https://people.inf.ethz.ch/omutlu/pub/safari-newsletter-december-2025.pdf**

Memory-Centric Computing

Solving Computing's Memory Problem

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

9 February 2026

EFCL Winter School Keynote Talk

SAFARI

ETH zürich

Backup Slides

Concluding Remarks

- **Goal: Enable computation capability in memory**

- We highlighted **major recent advances** in Processing-in-DRAM
  - Can lead to **orders-of-magnitude energy & perf** improvements
  - **Unmodified DRAM chips are already capable of computation**

- Memory should be designed as a **combined computation and storage substrate**
  - Not as an inactive storage substrate
  - Design mindset and flow should change

- Future of **truly memory-centric computing** is bright
  - We need to do research & design across the computing stack
  - With a proper mindset and infrastructure shift

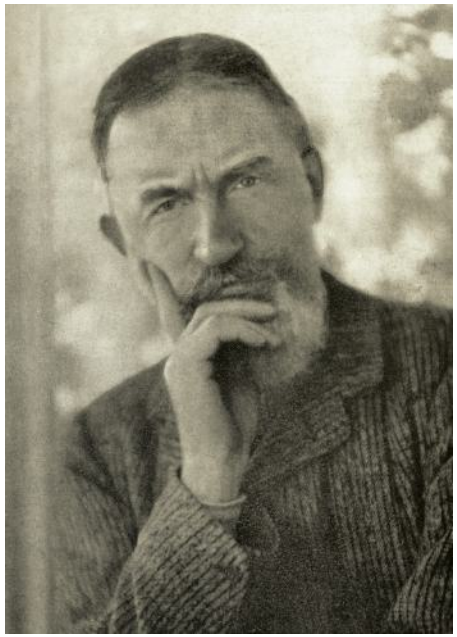
Food for Thought: Two Quotes

The reasonable man adapts himself to the world;

The unreasonable one persists in trying to adapt the world to himself.

Therefore, all progress depends on the unreasonable man.

George Bernard Shaw



Progress is impossible without change,
and those who cannot change their minds
cannot change anything.

And, One Poem

Whoso would be a man,
must be a nonconformist.
He who would gather mortal palms
must not be hindered by the name of goodness,
but must explore if it be goodness.
Nothing is at last sacred
but the integrity of your own mind.
Absolve you to yourself, and you
shall have the suffrage of the world.

Ralph Waldo Emerson

https://thelongvictorian.com/2016/04/27/quotes-wisdom-wit-ralph-waldo-emerson/

Adoption Issues

Adoption: How to Ease **Programmability**? (I)

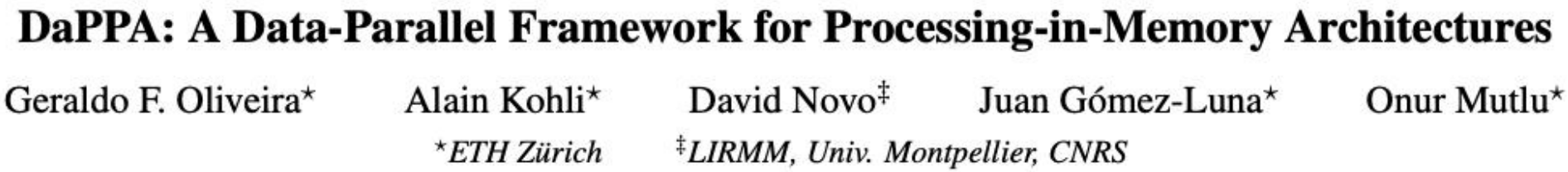
- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**
  *Proceedings of the 43rd International Symposium 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[‡] Eiman Ebrahimi[†] Gwangsun Kim[*] Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu[§‡] Stephen W. Keckler[†]
[‡]Carnegie Mellon University [†]NVIDIA [*]KAIST [§]ETH Zürich

Adoption: How to Ease **Programmability**? (II)

- Geraldo F. Oliveira, Alain Kohli, David Novo, Juan Gómez-Luna, Onur Mutlu, **"DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures,"** in *PACT SRC Student Competition,* Vienna, Austria, October 2023.



DaPPA: A Data-Parallel Framework for Processing-in-Memory Architectures

Geraldo F. Oliveira⋆ Alain Kohli⋆ David Novo‡ Juan Gómez-Luna⋆ Onur Mutlu⋆

⋆*ETH Zürich* ‡*LIRMM, Univ. Montpellier, CNRS*

Adoption: How to Ease **Programmability**? (III)

- Jinfan Chen, Juan Gómez-Luna, Izzat El Hajj, YuXin Guo, and Onur Mutlu,
  "SimplePIM: A Software Framework for Productive and Efficient Processing in Memory"
  *Proceedings of the 32nd International Conference on Parallel Architectures and Compilation Techniques (**PACT**)*, Vienna, Austria, October 2023.

SimplePIM: A Software Framework for Productive and Efficient Processing-in-Memory

Jinfan Chen[1] Juan Gómez-Luna[1] Izzat El Hajj[2] Yuxin Guo[1] Onur Mutlu[1]
[1]ETH Zürich [2]American University of Beirut

Adoption: How to Ease **Programmability**? (IV)

- 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"**
  IEEE Access, 8 September 2021.
  Preprint in ***arXiv***, 8 May 2021.
  [arXiv preprint]
  [IEEE Access version]
  [DAMOV Suite and Simulator Source Code]
  [SAFARI Live Seminar Video (2 hrs 40 mins)]
  [Short Talk Video (21 minutes)]

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, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

Adoption: How to Ease **Programmability**? (V)

- **Appears in IEEE TETC 2023**

ALP: Alleviating CPU-Memory Data Movement Overheads in Memory-Centric Systems

Nika Mansouri Ghiasi, Nandita Vijaykumar, Geraldo F. Oliveira, Lois Orosa, Ivan Fernandez, Mohammad Sadrosadati, Konstantinos Kanellopoulos, Nastaran Hajinazar, Juan Gómez Luna, Onur Mutlu

Abstract—Recent advances in memory technology have enabled near-data processing (NDP) to tackle main memory bottlenecks in modern systems. Prior works partition applications into segments (e.g., instructions, loops, functions) and execute memory-bound segments of the applications on NDP computation units, while mapping the cache-friendly application segments to host CPU cores that access a deeper cache hierarchy. Partitioning applications between NDP and host cores causes inter-segment data movement overhead, which is the overhead from moving data generated from one segment and used in the consecutive segments. This overhead can be large if the segments map to cores in different parts of the system (i.e., host and NDP). Prior works take two approaches to the inter-segment data movement overhead when partitioning applications between NDP and host cores. The first class of works maps segments to NDP or host cores based on the properties of each segment, neglecting the performance impact of the inter-segment data movement. Such partitioning techniques suffer from inter-segment data movement overhead. The second class of works maps segments to host or NDP cores based on the overall memory bandwidth savings of each segment (which depends on the memory bandwidth savings within each segment and the inter-segment data movement overhead between other segments). These works do not offload each segment to the best-fitting core if they incur high inter-segment data movement overhead. Therefore these works miss some of the potential NDP performance benefits. We show that mapping each segment (here basic block) to its best-fitting core based on the properties of each segment, assuming no inter-segment data movement, can provide substantial performance benefits. However, we show that the inter-segment data movement reduces this benefit significantly.

To this end, we introduce ALP, a new programmer-transparent technique to leverage the performance benefits of NDP by *alleviating* the performance impact of inter-segment data movement between host and memory and enabling efficient partitioning of applications between host and NDP cores. ALP alleviates the inter-segment data movement overhead by *proactively and accurately* transferring the required data between the segments mapped on host and NDP cores. This is based on the key observation that the instructions that generate the inter-segment data stay the same across different executions of a program on different input sets. ALP uses a compiler pass to identify these instructions and uses specialized hardware support to transfer data between the host and NDP cores at runtime. Using both the compiler and runtime information, ALP efficiently maps application segments to either host or NDP cores considering 1) the properties of each segment, 2) the inter-segment data movement overhead between different segments, and 3) whether this inter-segment data movement overhead can be alleviated proactively and in a timely manner. We evaluate ALP across a wide range of workloads and show on average 54.3% and 45.4% speedup compared to executing the application only on the host CPU or only the NDP cores, respectively.

https://arxiv.org/pdf/2212.06292

Adoption: How to Maintain **Coherence**? (I)

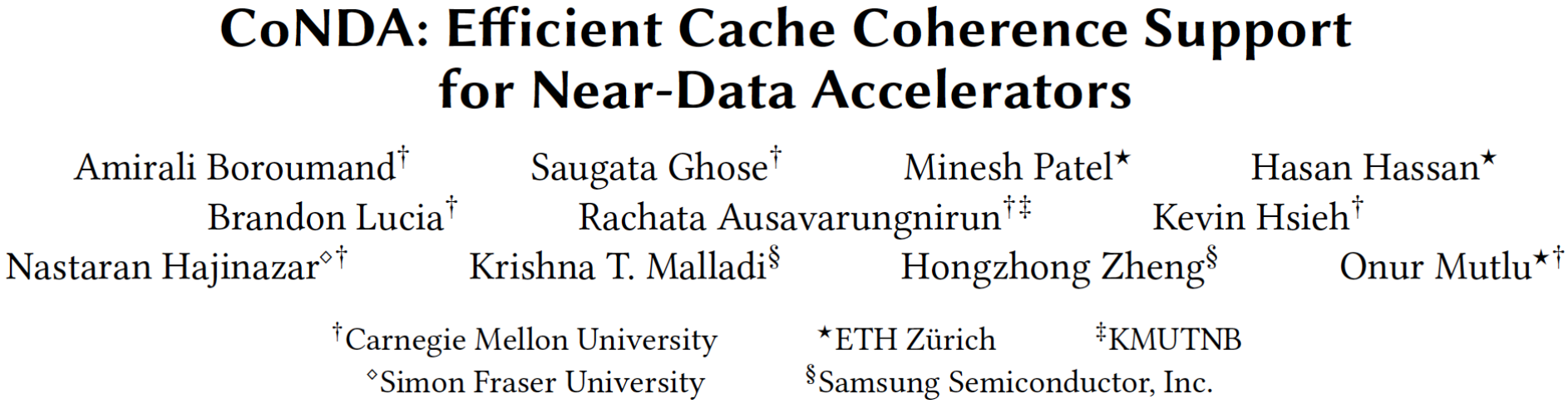
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
  "LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory"
  IEEE Computer Architecture Letters (***CAL***), June 2016.

LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory

Amirali Boroumand[†], Saugata Ghose[†], Minesh Patel[†], Hasan Hassan[†§], Brandon Lucia[†],
Kevin Hsieh[†], Krishna T. Malladi[*], Hongzhong Zheng[*], and Onur Mutlu[‡†]
[†]*Carnegie Mellon University* [*]*Samsung Semiconductor, Inc.* [§]*TOBB ETÜ* [‡]*ETH Zürich*

Adoption: How to Maintain **Coherence**? (II)

- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
  "CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators"
  *Proceedings of the 46th International Symposium on Computer Architecture (**ISCA**)*, Phoenix, AZ, USA, June 2019.



CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand[†] Saugata Ghose[†] Minesh Patel[⋆] Hasan Hassan[⋆]
Brandon Lucia[†] Rachata Ausavarungnirun[†‡] Kevin Hsieh[†]
Nastaran Hajinazar[◇†] Krishna T. Malladi[§] Hongzhong Zheng[§] Onur Mutlu[⋆†]

[†]Carnegie Mellon University [⋆]ETH Zürich [‡]KMUTNB
[◇]Simon Fraser University [§]Samsung Semiconductor, Inc.

Adoption: How to Support **Synchronization**?

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, 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.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Talk Video (21 minutes)]
  [Short Talk Video (7 minutes)]

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

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

Adoption: How to Support **Virtual Memory**?

- 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[†] Samira Khan[‡] Nandita Vijaykumar[†]
Kevin K. Chang[†] Amirali Boroumand[†] Saugata Ghose[†] Onur Mutlu[§†]
[†]*Carnegie Mellon University* [‡]*University of Virginia* [§]*ETH Zürich*

Adoption: Evaluation Infrastructures (I)

- 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"**
  IEEE Access, 8 September 2021.
  Preprint in ***arXiv***, 8 May 2021.
  [arXiv preprint]
  [IEEE Access version]
  [DAMOV Suite and Simulator Source Code]
  [SAFARI Live Seminar Video (2 hrs 40 mins)]
  [Short Talk Video (21 minutes)]

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, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

Adoption: Evaluation Infrastructures (II)

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
  "PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"
  ACM Transactions on Architecture and Code Optimization (***TACO***), March 2023.
  [arXiv version]
  Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (40 minutes)]
  [PiDRAM Source Code]

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

Adoption: Evaluation Infrastructures (III)

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
  "Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
  *Preprint on **arxiv***, August 2023.
  [arXiv version]
  [Ramulator 2.0 Source Code]

https://arxiv.org/pdf/2308.11030.pdf

https://github.com/CMU-SAFARI/ramulator2

What About Other Types of Memories?

In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu,
  "Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory"
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***), Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (44 minutes)]
  [arXiv version]

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

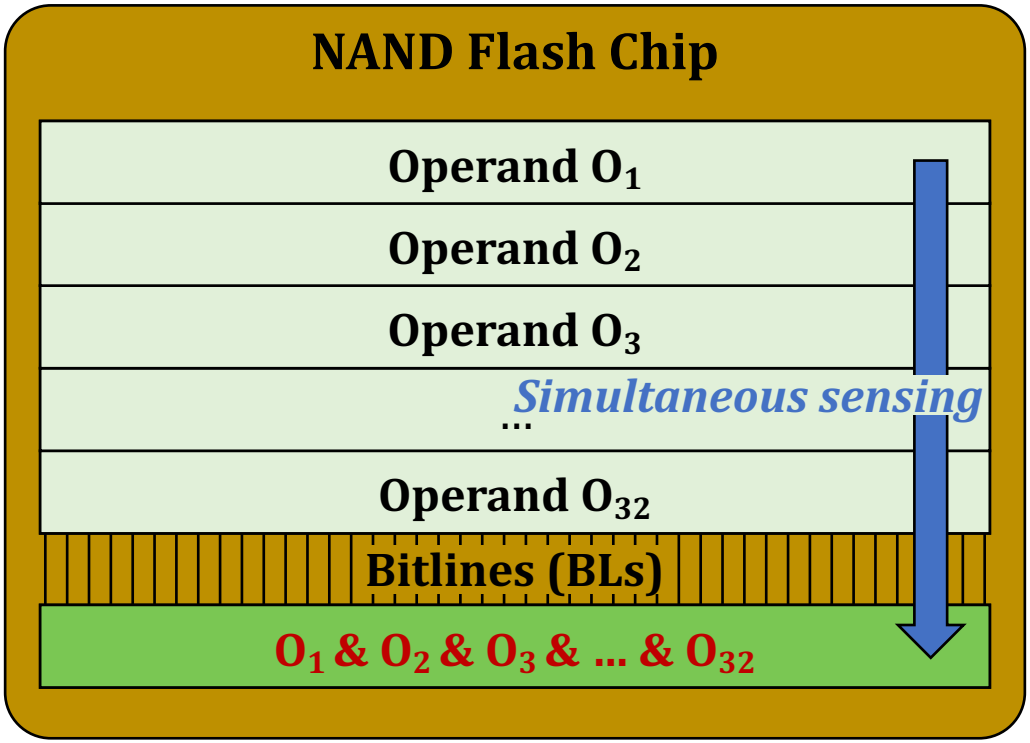
Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/pdf/2209.05566.pdf

Flash-Cosmos: Basic Ideas

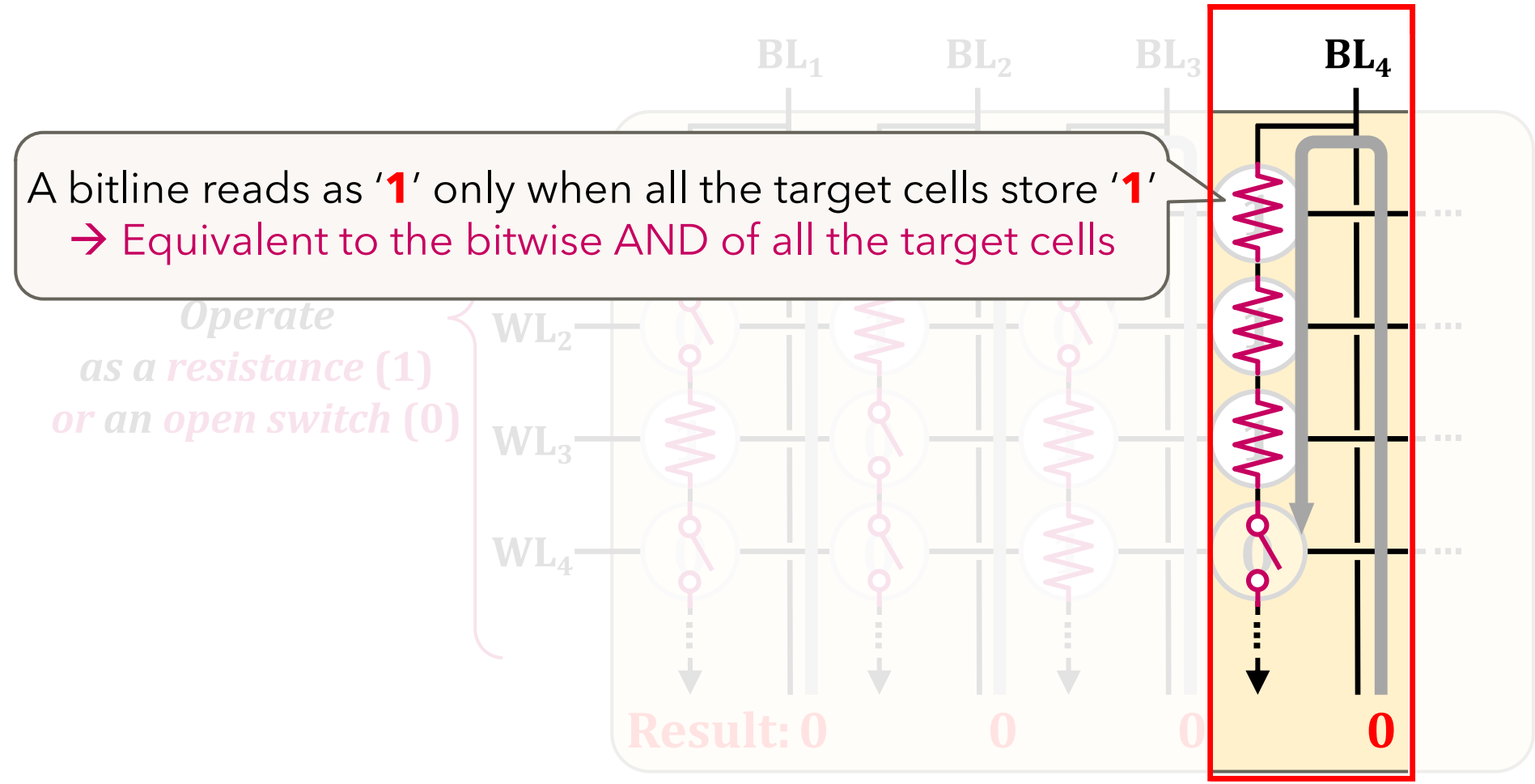
- Flash-Cosmos enables
  - Computation on multiple operands with a single sensing operation
  - Accurate computation results by eliminating raw bit errors in stored data



Multi-Wordline Sensing (MWS): Bitwise AND

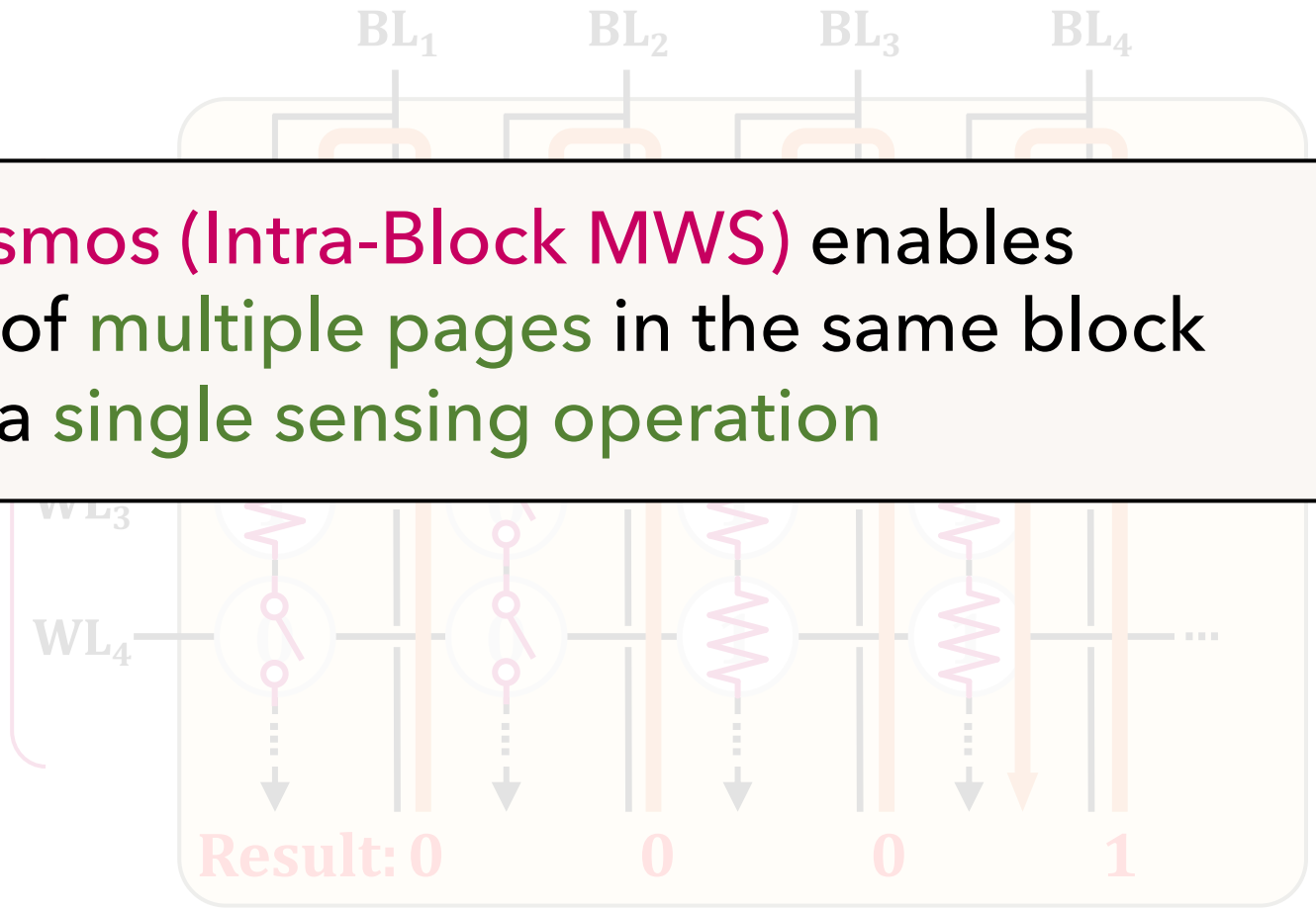
- **Intra-Block MWS**:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs

A bitline reads as '**1**' only when all the target cells store '**1**'
→ Equivalent to the bitwise AND of all the target cells



Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



Flash-Cosmos (Intra-Block MWS) enables bitwise AND of multiple pages in the same block via a single sensing operation

Other Types of Bitwise Operations

Flash-Cosmos also enables
other types of bitwise operations
(NOT/NAND/NOR/XOR/XNOR)
leveraging existing features of NAND flash memory

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/abs/2209.05566.pdf

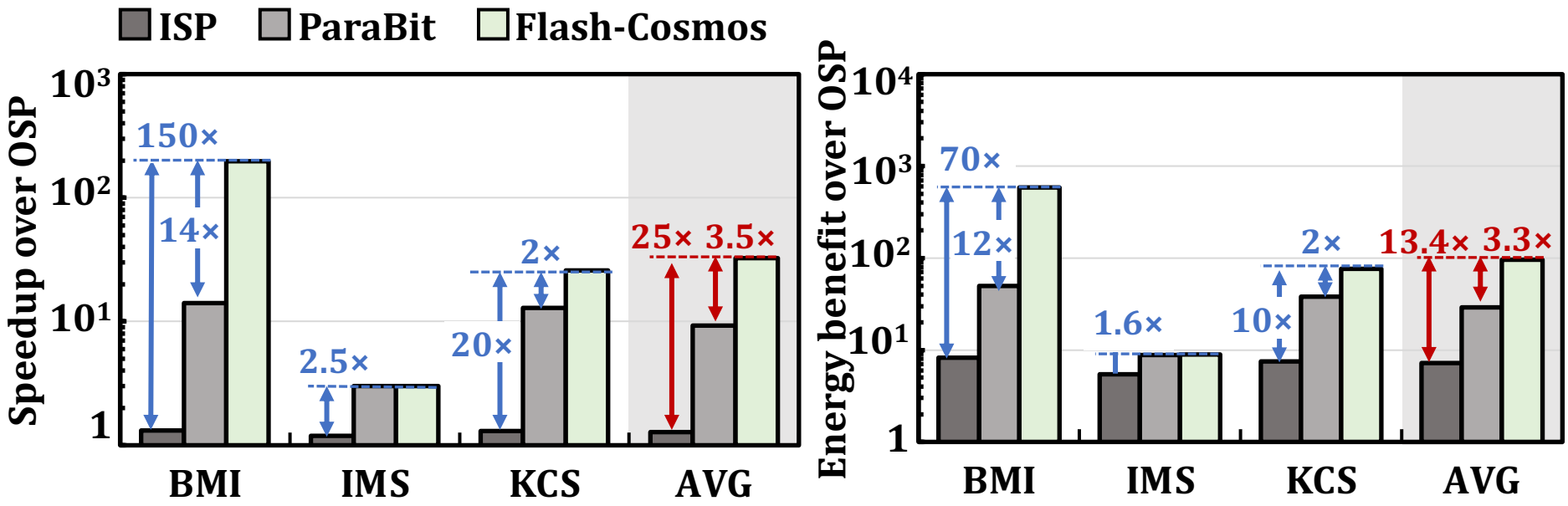
Results: Real-Device Characterization

No changes to the cell array
of commodity NAND flash chips

Can have many operands
(AND: up to 48, OR: up to 4)
with small increase in sensing latency (< 10%)

ESP significantly improves
the reliability of computation results
(no observed bit error in the tested flash cells)

Results: Performance & Energy



Flash-Cosmos provides significant performance & energy benefits over all the baselines

The larger the number of operands,
the higher the performance & energy benefits

Flash-Cosmos: In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu,
  "Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory"
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***), Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (44 minutes)]
  [arXiv version]

Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/pdf/2209.05566.pdf

PAPI LLM Inference System

[ASPLOS 2025]

PAPI: Hybrid System for Near-Memory LLM Inference

- Yintao He, Haiyu Mao, Christina Giannoula, Mohammad Sadrosadati, Juan Gomez-Luna, Huawei Li, Xiaowei Li, Ying Wang, and Onur Mutlu, **"PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System,"** *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems* (***ASPLOS***), Rotterdam, Netherlands, April 2025.

PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System

Yintao He[1,2] Haiyu Mao[3,4] Christina Giannoula[5,6,4] Mohammad Sadrosadati[4]
Juan Gómez-Luna[7] Huawei Li[1,2] Xiaowei Li[1,2] Ying Wang[1] Onur Mutlu[4]
[1]SKLP, Institute of Computing Technology, CAS [2]University of Chinese Academy of Sciences [3] King's College London
[4]ETH Zürich [5]University of Toronto [6]Vector Institute [7] NVIDIA

https://arxiv.org/pdf/2502.15470

PAPI's Key Idea

Enable **online dynamic task scheduling** in a **heterogeneous PIM-enabled architecture** via online identification of kernel properties in LLM decoding

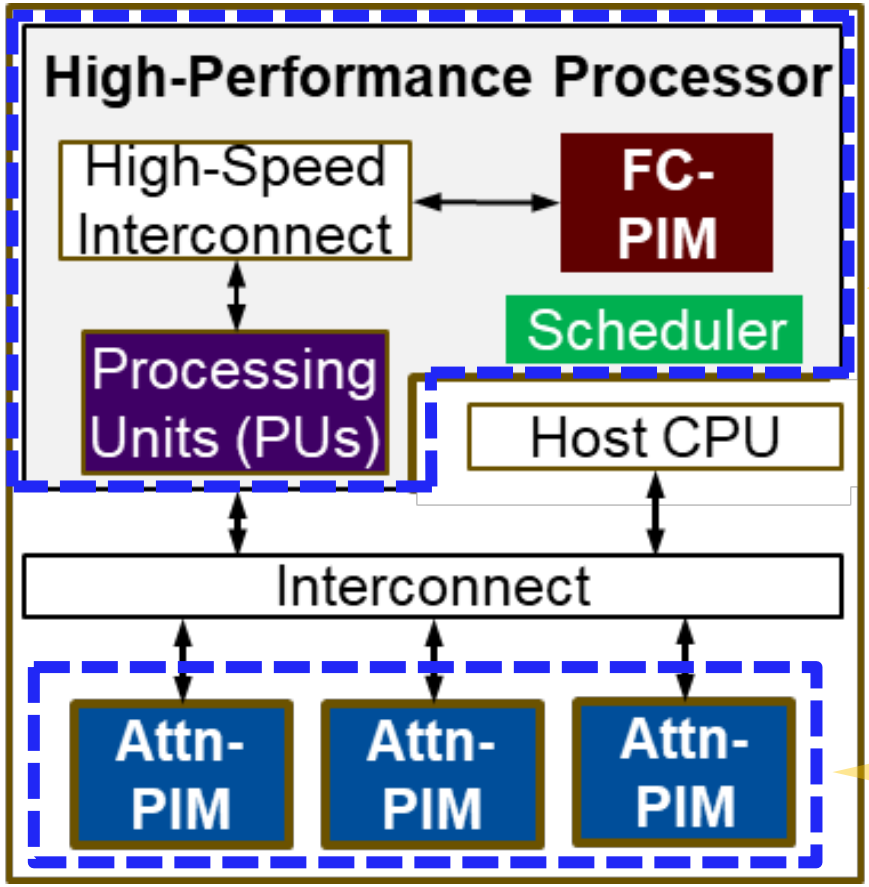
PAPI's Key Components

A new PIM-enabled computing system design

Hybrid PIM units
to cater to different parallelism levels of FC and attention kernels

Dynamic LLM kernel scheduling
to cater to dynamically changing parallelism levels

PAPI's Architecture

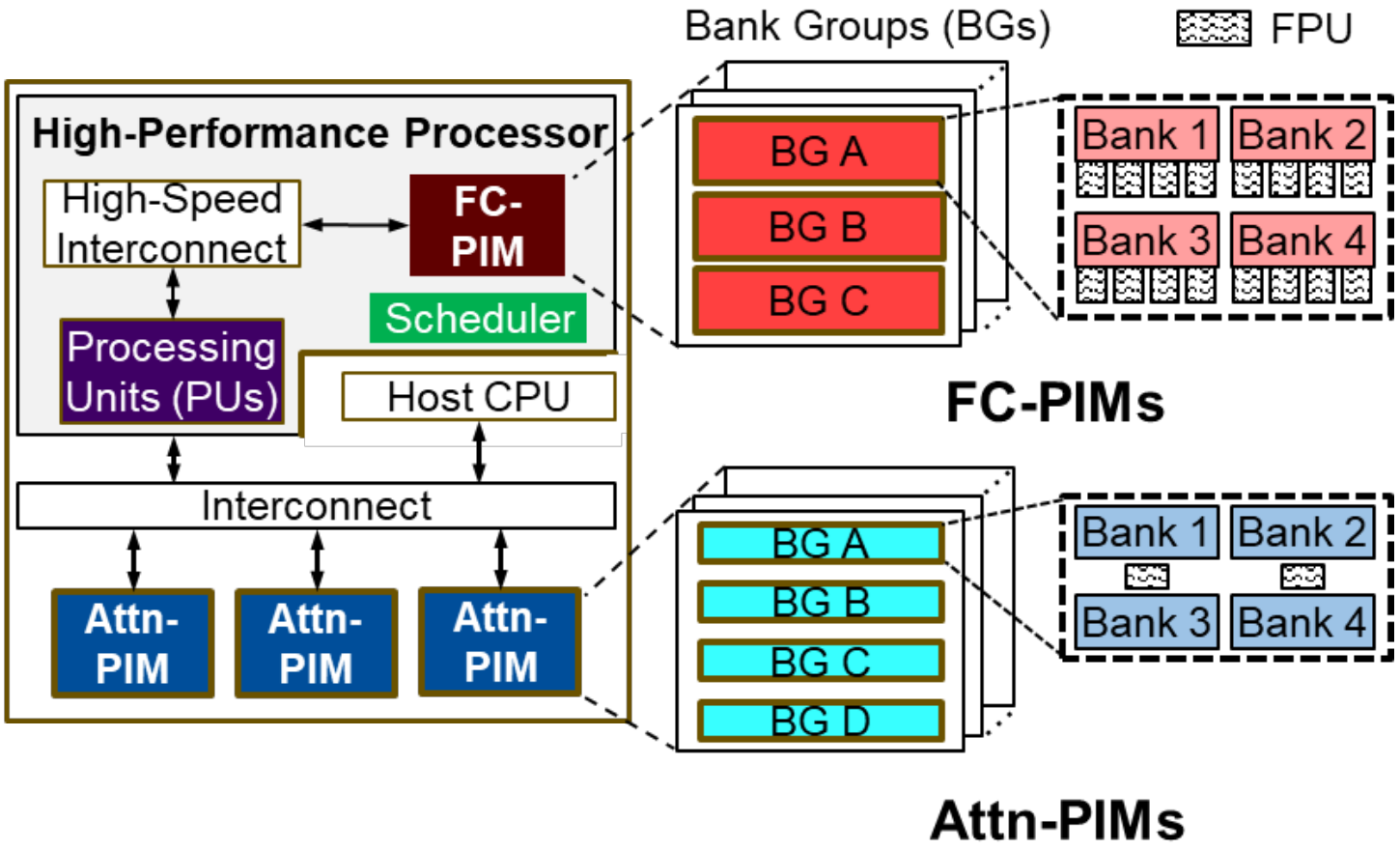


Handles memory-bound or compute-bound **FC kernels**

- Execution of FC kernels
- Dynamic scheduling

Handles memory-bound **attention kernels**

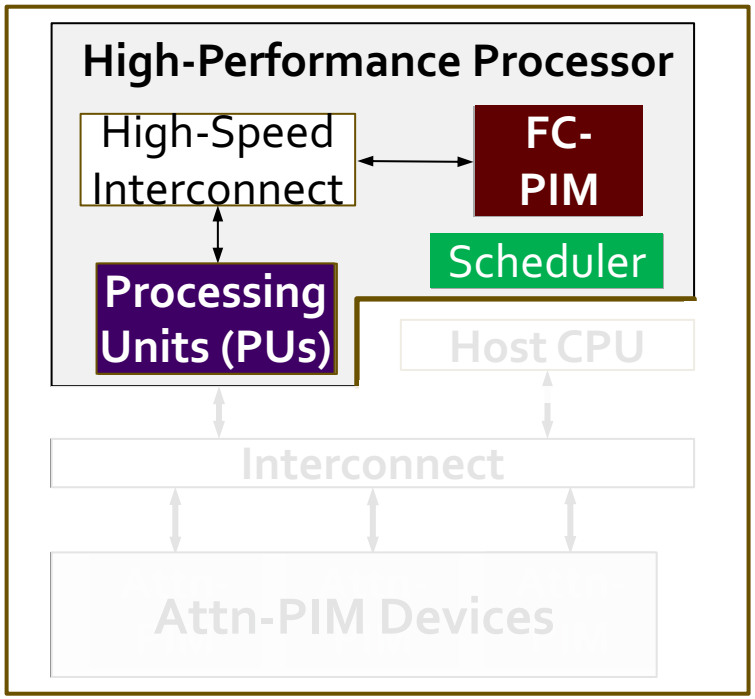
PAPI's Architecture



Hybrid PIM units handle memory-bound FC & attention kernels with **different computational and memory demands**

Outline

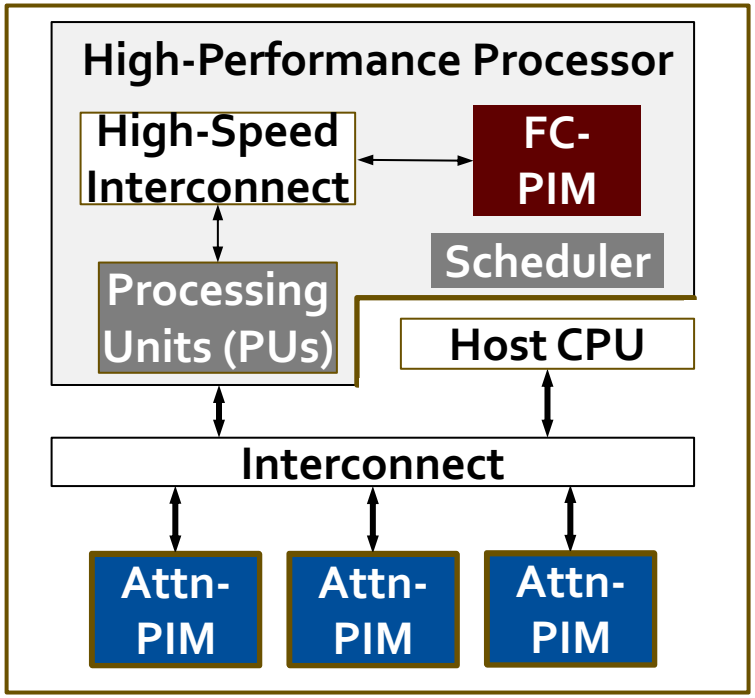
High-Performance Processor



When FC kernels are compute-bound:
Assign FC kernels to PUs

When FC kernels are memory-bound:
Assign FC kernels to FC-PIM

Hybrid PIM Units (I)



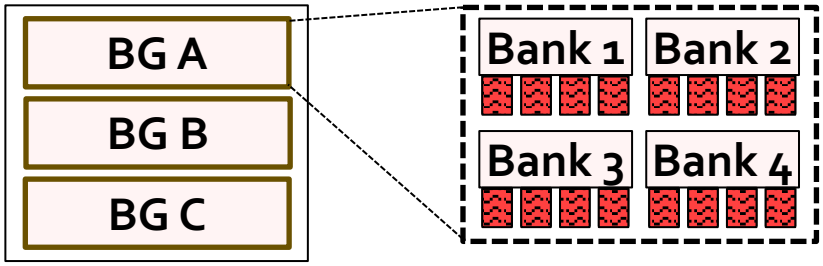
FC-PIM device placed in the High-Performance Processor

Attn-PIM devices store KV cache; **separated from** the High-Performance Processor

Hybrid PIM Units (II)

Floating-Point Processing Units (FPU)

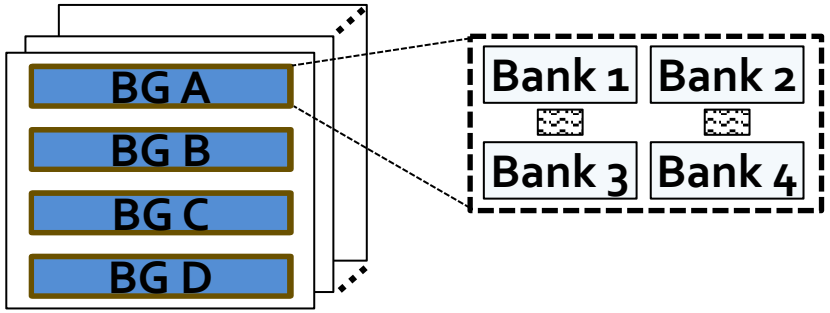
Bank Groups (BGs)



FC-PIM

More FPUs per Bank

Higher Computation Capability
to cater to FC kernels



Attn-PIMs

More Bank Groups per Stack

More Attn-PIM Devices

Higher Memory Capacity
to cater to attention kernels

PAPI Runtime Scheduler

Offline: identify memory-boundedness threshold

① Monitor Parallelism Levels

- RLP & TLP

② Arithmetic Intensity Predictor

- Estimate arithmetic intensity of FC kernels
- Compare with memory-boundedness threshold

③ Schedule the FC Kernels

- Map FC kernels to either FC-PIM or PUs

Evaluation Methodology

Performance and Energy Analysis:

- Simulation using AttAcc [ASPLOS'24] and Ramulator 2 [IEEE CAL'23]

Baselines:

- **AttAcc** [ASPLOS'24]
- **GPU+HBM-PIM** (NVIDIA A100 GPU + Samsung's HBM-PIM)
- **PIM-only** (PIM devices in AttAcc)

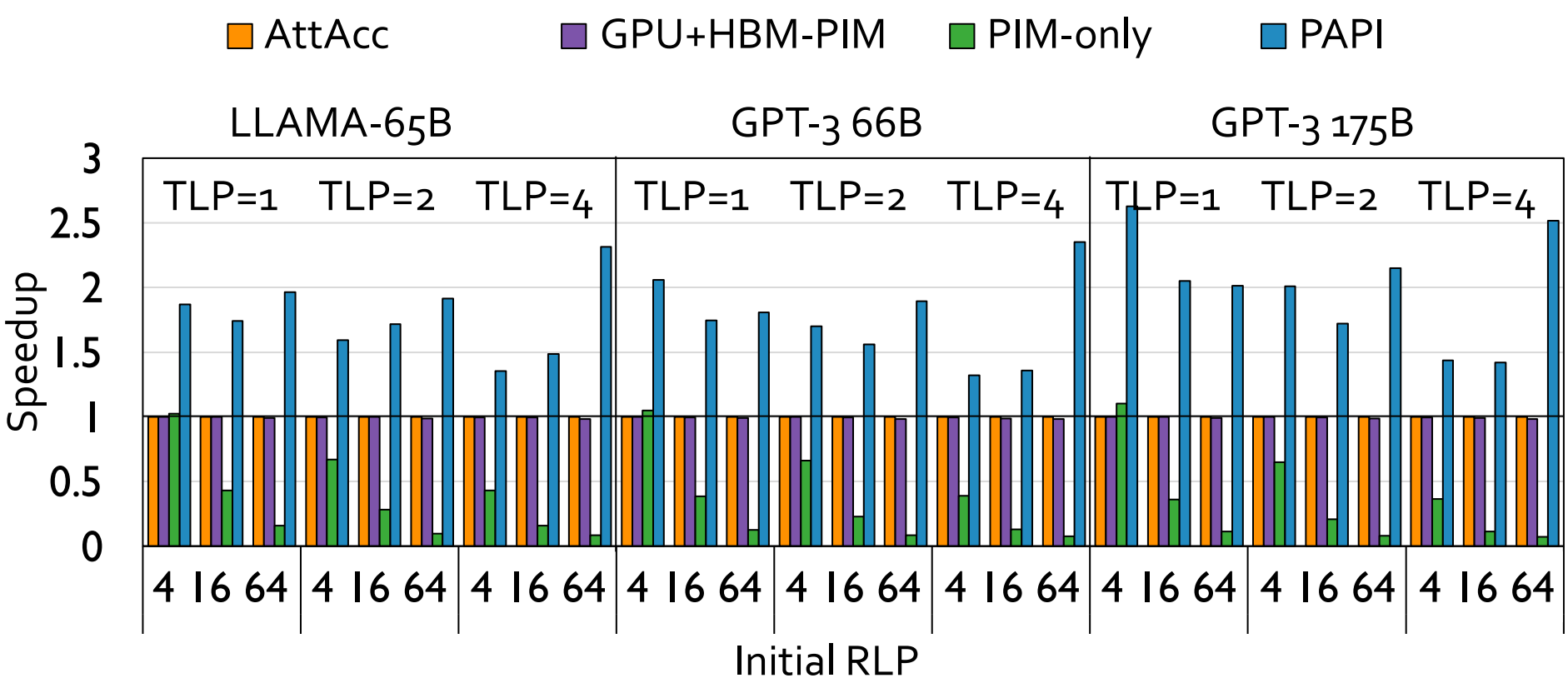
Workloads: Three transformer-based LLMs

- LLaMA-65B, GPT-3 66B, GPT-3 175B

Datasets: Dolly

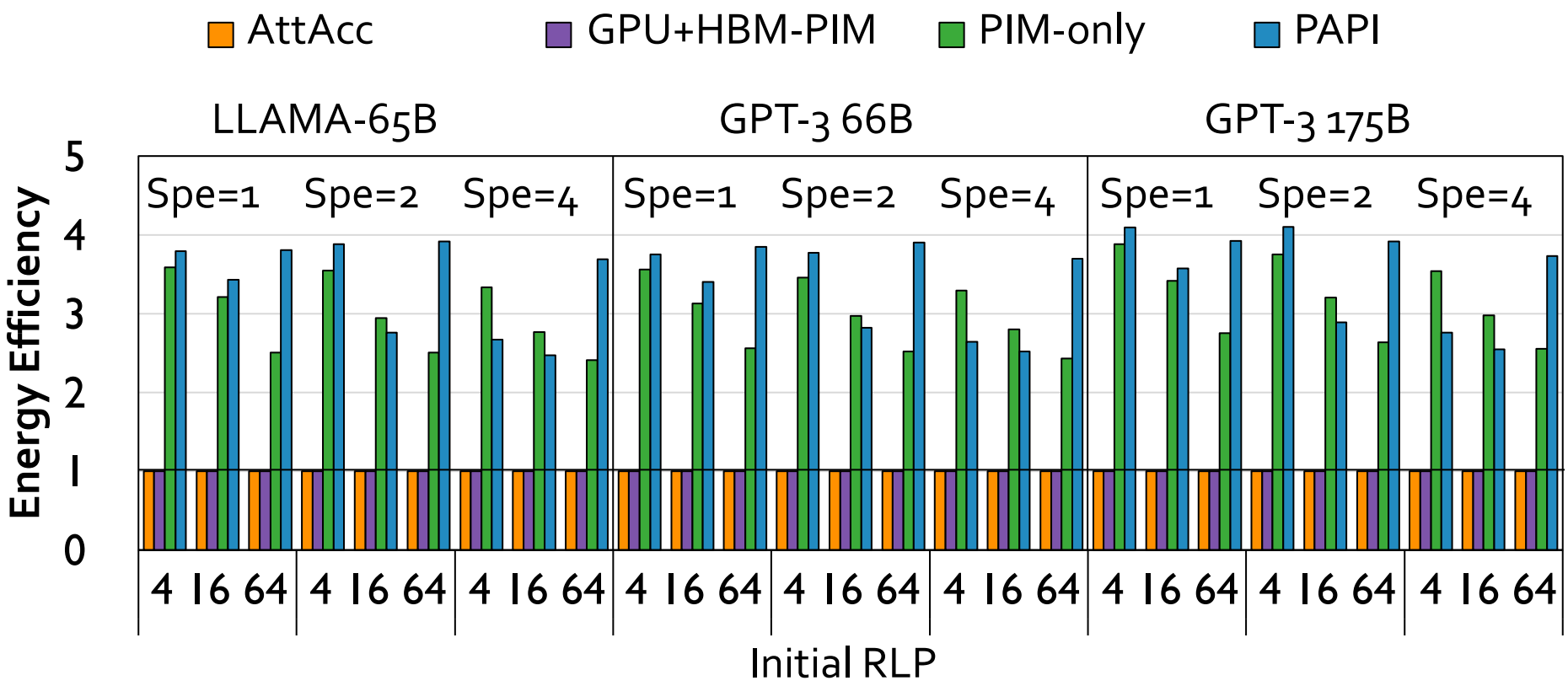
- Creative-writing tasks
- General-QA tasks

Performance Analysis



PAPI improves performance by 1.8X, 1.9X, and 11.1X compared to AttAcc, GPU+HBM-PIM, and PIM-only, respectively

Energy Analysis



PAPI improves energy efficiency by 3.4X, 3.4X, and 1.2X compared to AttAcc, GPU+HBM-PIM, and PIM-only, respectively

More in the Paper

- **Details on PAPI's implementation**
  - PAPI's heterogeneous architecture
  - PAPI's runtime scheduler
  - System integration
  - Data partitioning across PIM devices (both Attn-PIM & FC-PIM)

- **Detailed evaluation results**
  - PAPI's speedup across different RLP & TLP levels
  - Ablation study for PAPI's speedup

- **Area/power analysis**

More in the Paper

PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System

Yintao He[1,2] Haiyu Mao[3,4] Christina Giannoula[5,6,4] Mohammad Sadrosadati[4]
Juan Gómez-Luna[7] Huawei Li[1,2] Xiaowei Li[1,2] Ying Wang[1] Onur Mutlu[4]
[1]SKLP, Institute of Computing Technology, CAS [2]University of Chinese Academy of Sciences [3] King's College London
[4]ETH Zürich [5]University of Toronto [6]Vector Institute [7] NVIDIA

https://arxiv.org/pdf/2502.15470

Conclusion

Key Findings

1. LLM kernels have different computation and memory bandwidth demands across different RLP & TLP levels

2. Memory-bound kernels exhibit different computation demands depending on kernel type

3. LLM kernels have dynamically changing RLP and TLP levels

Conclusion

Key Contribution

PAPI

A new **PIM-enabled heterogeneous** system design that caters to **varying demands** of LLM kernels by scheduling them **dynamically** to computation-centric processing units and hybrid PIM units

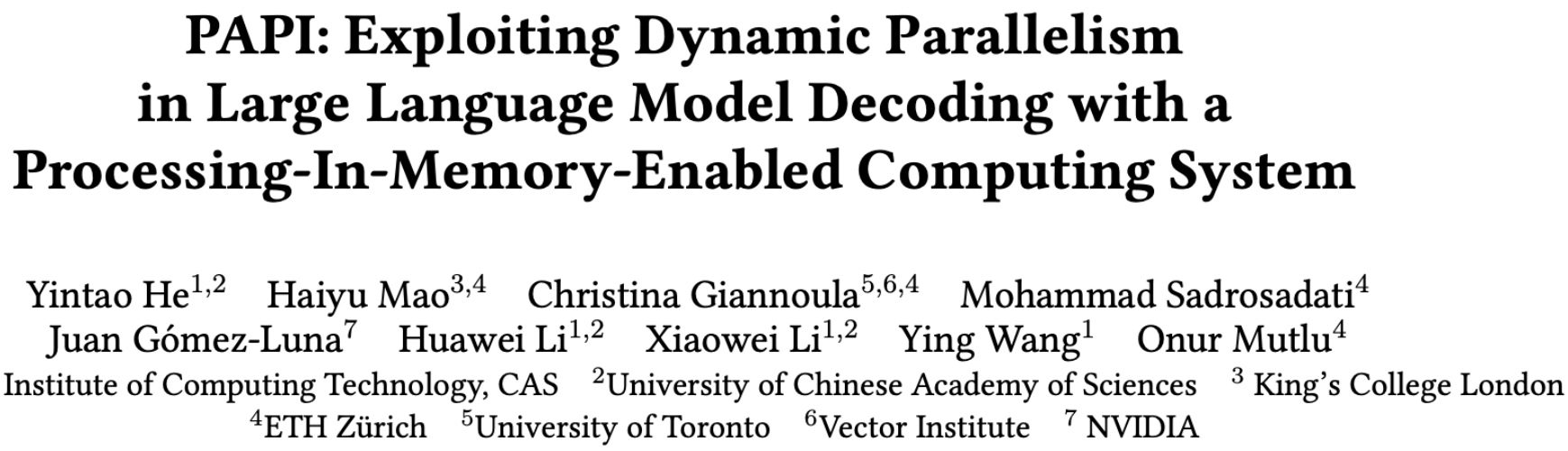
Key Results

PAPI largely improves both performance and energy efficiency over best prior LLM decoding system

- **1.8×** speedup
- **3.4×** energy efficiency increase

PAPI: Hybrid System for Near-Memory LLM Inference

- Yintao He, Haiyu Mao, Christina Giannoula, Mohammad Sadrosadati, Juan Gomez-Luna, Huawei Li, Xiaowei Li, Ying Wang, and Onur Mutlu, **"PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System,"** *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems* (***ASPLOS***), Rotterdam, Netherlands, April 2025.



PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System

Yintao He[1,2] Haiyu Mao[3,4] Christina Giannoula[5,6,4] Mohammad Sadrosadati[4]
Juan Gómez-Luna[7] Huawei Li[1,2] Xiaowei Li[1,2] Ying Wang[1] Onur Mutlu[4]
[1]SKLP, Institute of Computing Technology, CAS [2]University of Chinese Academy of Sciences [3] King's College London
[4]ETH Zürich [5]University of Toronto [6]Vector Institute [7] NVIDIA

https://arxiv.org/pdf/2502.15470

Workload Studies

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[‡] Eiman Ebrahimi[†] Gwangsun Kim[*] Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu[§‡] Stephen W. Keckler[†]
[‡]Carnegie Mellon University [†]NVIDIA [*]KAIST [§]ETH Zürich

Accelerating GPU Execution with PIM (II)

- Ashutosh Pattnaik, Xulong Tang, Adwait Jog, Onur Kayiran, Asit K. Mishra, Mahmut T. Kandemir, Onur Mutlu, and Chita R. Das, **"Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities"**
  *Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques (**PACT**)*, Haifa, Israel, September 2016.

Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities

Ashutosh Pattnaik[1] Xulong Tang[1] Adwait Jog[2] Onur Kayıran[3]
Asit K. Mishra[4] Mahmut T. Kandemir[1] Onur Mutlu[5,6] Chita R. Das[1]
[1]Pennsylvania State University [2]College of William and Mary
[3]Advanced Micro Devices, Inc. [4]Intel Labs [5]ETH Zürich [6]Carnegie Mellon University

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[†] Samira Khan[‡] Nandita Vijaykumar[†]
Kevin K. Chang[†] Amirali Boroumand[†] Saugata Ghose[†] Onur Mutlu[§†]
[†]*Carnegie Mellon University* [‡]*University of Virginia* [§]*ETH Zürich*

Accelerating Dependent Cache Misses

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt, **"Accelerating Dependent Cache Misses with an Enhanced Memory Controller"**
  *Proceedings of the 43rd International Symposium on Computer Architecture (**ISCA**)*, Seoul, South Korea, June 2016.
  [Slides (pptx) (pdf)]
  [Lightning Session Slides (pptx) (pdf)]

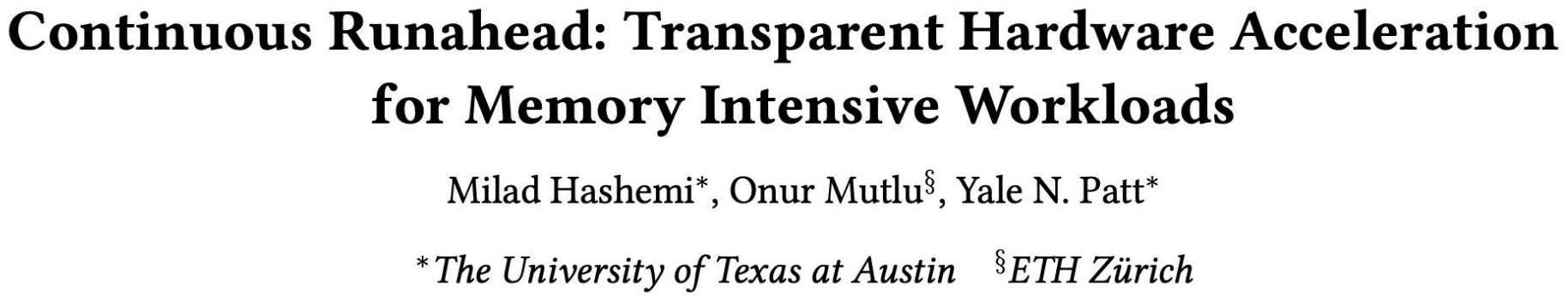
Accelerating Dependent Cache Misses with an Enhanced Memory Controller

Milad Hashemi*, Khubaib†, Eiman Ebrahimi‡, Onur Mutlu§, Yale N. Patt*

*The University of Texas at Austin †Apple ‡NVIDIA §ETH Zürich & Carnegie Mellon University

Accelerating Runahead Execution

- Milad Hashemi, Onur Mutlu, and Yale N. Patt,
  "Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads"
  *Proceedings of the 49th International Symposium on Microarchitecture (**MICRO**)*, Taipei, Taiwan, October 2016.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pdf)] [Poster (pptx) (pdf)]
  Best paper session.



Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

Milad Hashemi*, Onur Mutlu§, Yale N. Patt*

*The University of Texas at Austin §ETH Zürich

Accelerating Climate Modeling

- 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[a,b,c] Dionysios Diamantopoulos[c] Christoph Hagleitner[c] Juan Gómez-Luna[b]
Sander Stuijk[a] Onur Mutlu[b] Henk Corporaal[a]
[a]Eindhoven University of Technology [b]ETH Zürich [c]IBM Research Europe, Zurich

Accelerating DNA Read Mapping

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

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

Accelerating Approximate String Matching

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

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

Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,
  "SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
  *Proceedings of the 49th International Symposium on Computer Architecture (**ISCA**)*, New York, June 2022.
  [arXiv version]

SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali[1] Konstantinos Kanellopoulos[2] Joël Lindegger[2] Zülal Bingöl[3]
Gurpreet S. Kalsi[4] Ziyi Zuo[5] Can Firtina[2] Meryem Banu Cavlak[2] Jeremie Kim[2]
Nika Mansouri Ghiasi[2] Gagandeep Singh[2] Juan Gómez-Luna[2] Nour Almadhoun Alserr[2]
Mohammed Alser[2] Sreenivas Subramoney[4] Can Alkan[3] Saugata Ghose[6] Onur Mutlu[2]

[1]Bionano Genomics [2]ETH Zürich [3]Bilkent University [4]Intel Labs
[5]Carnegie Mellon University [6]University of Illinois Urbana-Champaign

Accelerating Basecalling + Read Mapping

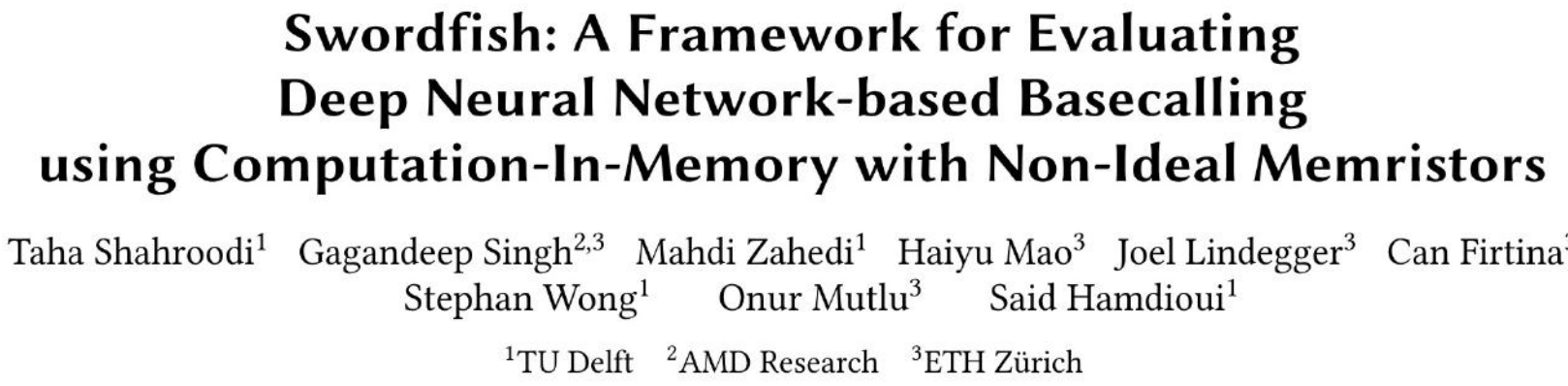
- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu,
  "GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping"
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***),
  Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (25 minutes)]
  [arXiv version]

GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping

Haiyu Mao[1] Mohammed Alser[1] Mohammad Sadrosadati[1] Can Firtina[1] Akanksha Baranwal[1]
Damla Senol Cali[2] Aditya Manglik[1] Nour Almadhoun Alserr[1] Onur Mutlu[1]
[1]*ETH Zürich* [2]*Bionano Genomics*

Accelerating Basecalling

- Taha Shahroodi, Gagandeep Singh, Mahdi Zahedi, Haiyu Mao, Joel Lindegger, Can Firtina, Stephan Wong, Onur Mutlu, and Said Hamdioui,
  "Swordfish: A Framework for Evaluating Deep Neural Network-based Basecalling using Computation-In-Memory with Non-Ideal Memristors"
  *Proceedings of the 56th International Symposium on Microarchitecture (**MICRO**)*, Toronto, ON, Canada, November 2023.
  [Slides (pptx) (pdf)]
  [arXiv version]



Swordfish: A Framework for Evaluating Deep Neural Network-based Basecalling using Computation-In-Memory with Non-Ideal Memristors

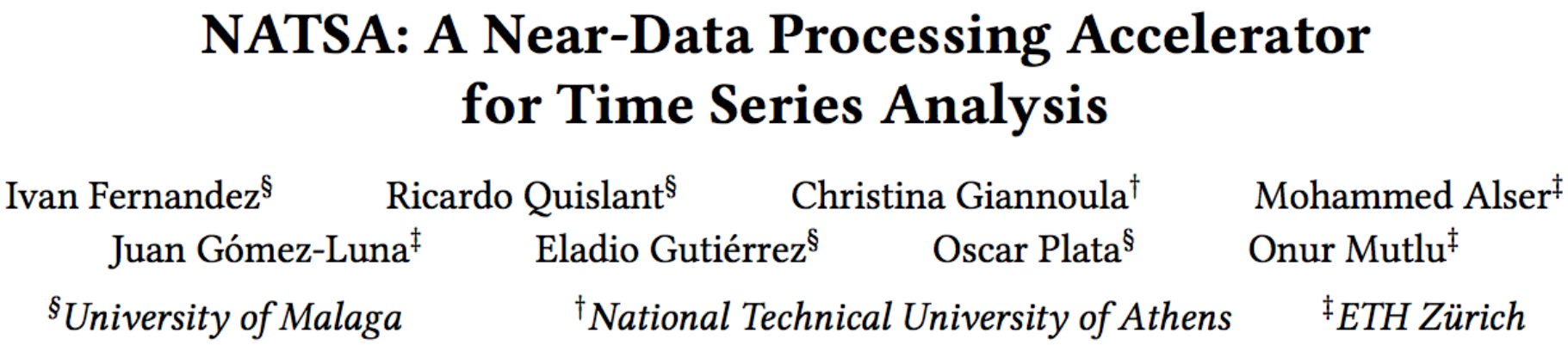
Taha Shahroodi[1] Gagandeep Singh[2,3] Mahdi Zahedi[1] Haiyu Mao[3] Joel Lindegger[3] Can Firtina[3]
Stephan Wong[1] Onur Mutlu[3] Said Hamdioui[1]

[1]TU Delft [2]AMD Research [3]ETH Zürich

https://arxiv.org/pdf/2310.04366

Accelerating Time Series Analysis (I)

- Ivan Fernandez, Ricardo Quislant, 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[§] Ricardo Quislant[§] Christina Giannoula[†] Mohammed Alser[‡]
Juan Gómez-Luna[‡] Eladio Gutiérrez[§] Oscar Plata[§] Onur Mutlu[‡]

[§]*University of Malaga* [†]*National Technical University of Athens* [‡]*ETH Zürich*

https://arxiv.org/pdf/2010.02079

Accelerating Time Series Analysis (II)

- Ivan Fernandez, Christina Giannoula, Aditya Manglik, Ricardo Quislant, Nika Mansouri Ghiasi, Juan Gomez Luna, Eladio Gutierrez, Oscar Plata and Onur Mutlu,
  "MATSA: An MRAM-Based Energy-Efficient Accelerator for Time Series Analysis"
  IEEE Access, March 2024.
  [arXiv version]
  [IEEE Access version]

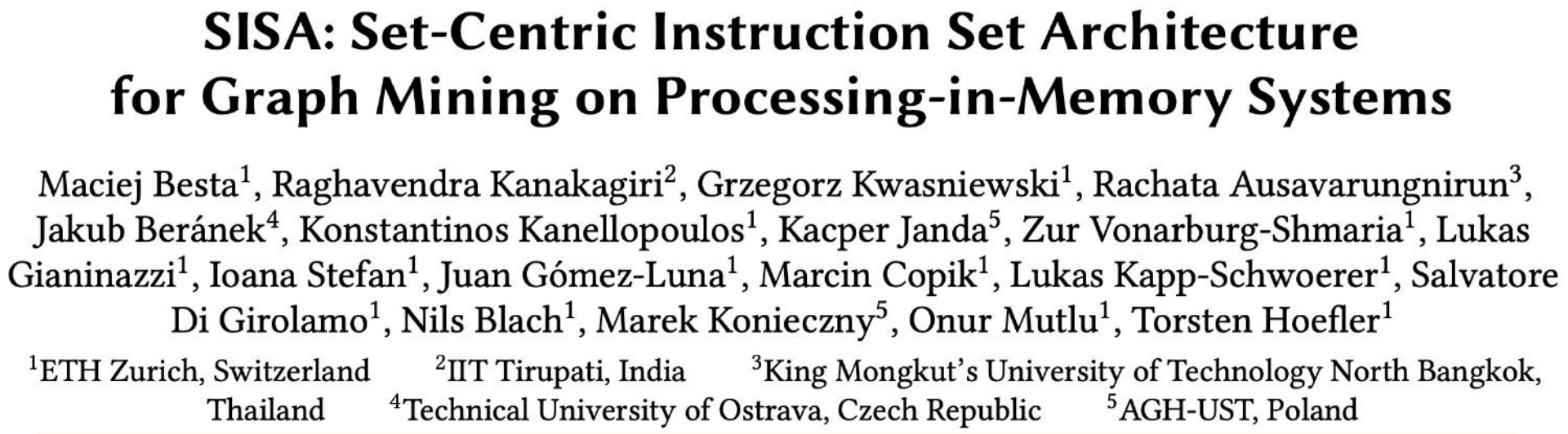
Accelerating Time Series Analysis via Processing using Non-Volatile Memories

Ivan Fernandez[§†¶] *Christina Giannoula[†‡] *Aditya Manglik[†] Ricardo Quislant[§] Nika Mansouri Ghiasi[†]
Juan Gómez-Luna[†] Eladio Gutierrez[§] Oscar Plata[§] Onur Mutlu[†]
[§]University of Malaga [†]ETH Zürich [¶]Barcelona Supercomputing Center [‡]National Technical University of Athens

https://arxiv.org/pdf/2211.04369

Accelerating Graph Pattern Mining

- Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungnirun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,
  "SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems"
  Proceedings of the 54th International Symposium on Microarchitecture (***MICRO***), Virtual, October 2021.
  [Slides (pdf)]
  [Talk Video (22 minutes)]
  [Lightning Talk Video (1.5 minutes)]
  [Full arXiv version]



SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems

Maciej Besta[1], Raghavendra Kanakagiri[2], Grzegorz Kwasniewski[1], Rachata Ausavarungnirun[3], Jakub Beránek[4], Konstantinos Kanellopoulos[1], Kacper Janda[5], Zur Vonarburg-Shmaria[1], Lukas Gianinazzi[1], Ioana Stefan[1], Juan Gómez-Luna[1], Marcin Copik[1], Lukas Kapp-Schwoerer[1], Salvatore Di Girolamo[1], Nils Blach[1], Marek Konieczny[5], Onur Mutlu[1], Torsten Hoefler[1]

[1]ETH Zurich, Switzerland [2]IIT Tirupati, India [3]King Mongkut's University of Technology North Bangkok, Thailand [4]Technical University of Ostrava, Czech Republic [5]AGH-UST, Poland

https://arxiv.org/pdf/2104.07582

Accelerating HTAP Database Systems

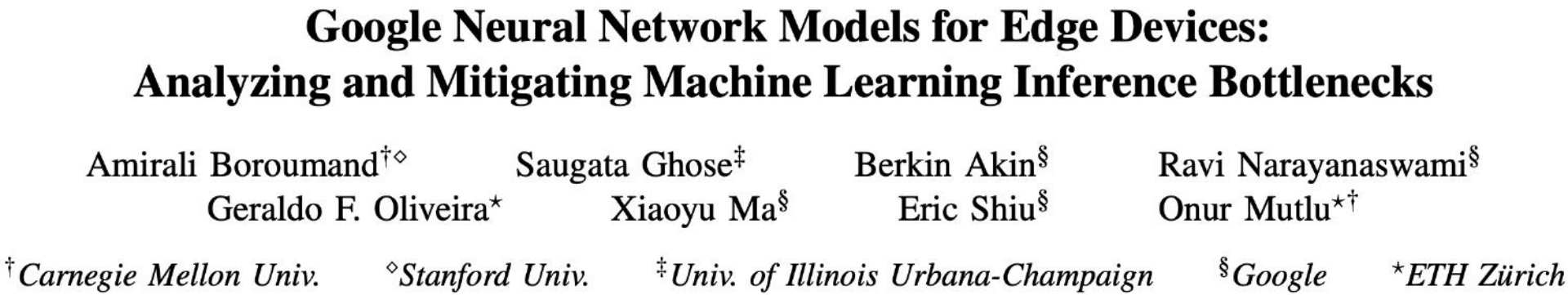
- Amirali Boroumand, Saugata Ghose, Geraldo F. Oliveira, and Onur Mutlu, **"Polynesia: Enabling High-Performance and Energy-Efficient Hybrid Transactional/Analytical Databases with Hardware/Software Co-Design"** *Proceedings of the 38th International Conference on Data Engineering* (***ICDE***), Virtual, May 2022.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]

Polynesia: Enabling High-Performance and Energy-Efficient Hybrid Transactional/Analytical Databases with Hardware/Software Co-Design

Amirali Boroumand[†] Saugata Ghose[◇] Geraldo F. Oliveira[‡] Onur Mutlu[‡]
[†]*Google* [◇]*Univ. of Illinois Urbana-Champaign* [‡]*ETH Zürich*

Accelerating ML Inference

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



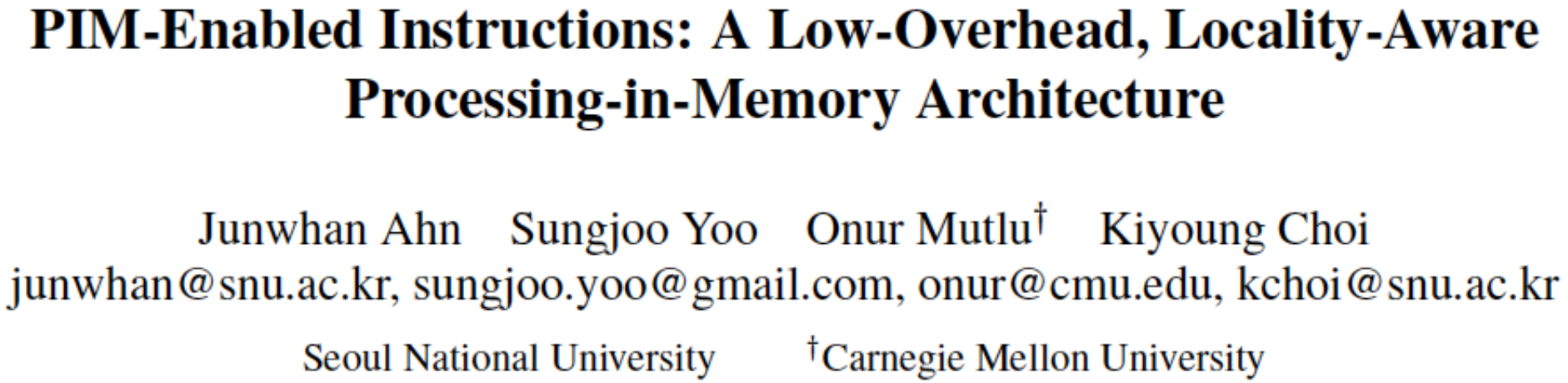
Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

Accelerating Data-Intensive Workloads

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoung 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)]

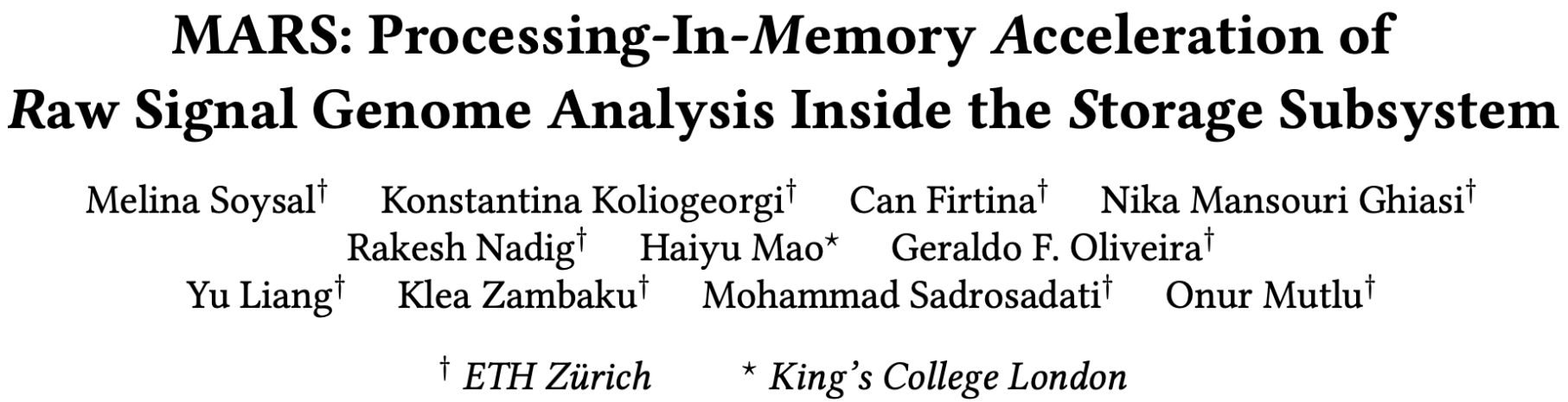


PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture

Junwhan Ahn Sungjoo Yoo Onur Mutlu† Kiyoung Choi
junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University †Carnegie Mellon University

Accelerating Raw Signal Genome Analysis

- Melina Soysal, Konstantina Koliogeorgi, Can Firtina, Nika Mansouri Ghiasi, Rakesh Nadig, Haiyu Mao, Geraldo Francisco, Yu Liang, Klea Zambaku, Mohammad Sadrosadati, and Onur Mutlu,
  "MARS: Processing-In-Memory Acceleration of Raw Signal Genome Analysis Inside the Storage Subsystem"
  *Proceedings of the 37th ACM International Conference on Supercomputing (**ICS**)*, Salt Lake City, UT, USA, June 2025.



MARS: Processing-In-*Memory* *Acceleration* of *Raw* Signal Genome Analysis Inside the *Storage* Subsystem

Melina Soysal[†] Konstantina Koliogeorgi[†] Can Firtina[†] Nika Mansouri Ghiasi[†]
Rakesh Nadig[†] Haiyu Mao[⋆] Geraldo F. Oliveira[†]
Yu Liang[†] Klea Zambaku[†] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†] *ETH Zürich* [⋆] *King's College London*

https://arxiv.org/pdf/2506.10931

Accelerating Retrieval Augmented Generation

- Kangqi Chen, Rakesh Nadig, Andreas Kosmas Kakolyris, Manos Frouzakis, Nika Mansouri Ghiasi, Yu Liang, Haiyu Mao, Jisung Park, Mohammad Sadrosadati, and Onur Mutlu,
  "REIS: A High-Performance and Energy-Efficient Retrieval System with In-Storage Processing"
  *Proceedings of the 52nd Annual International Symposium on Computer Architecture (**ISCA**)*, Tokyo, Japan, June 2025.

REIS: A High-Performance and Energy-Efficient Retrieval System with In-Storage Processing

Kangqi Chen[1] Andreas Kosmas Kakolyris[1] Rakesh Nadig[1] Manos Frouzakis[1]
Nika Mansouri Ghiasi[1] Yu Liang[1] Haiyu Mao[1,2]
Jisung Park[3] Mohammad Sadrosadati[1] Onur Mutlu[1]

ETH Zürich[1] King's College London[2] POSTECH[3]

https://arxiv.org/pdf/2506.16444

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[◇] **Mohammed Alser**[◇] **Damla Senol Cali**[⋈]
Dionysios Diamantopoulos[▽] **Juan Gómez-Luna**[◇]
Henk Corporaal[⋆] **Onur Mutlu**[◇⋈]

[◇]*ETH Zürich* [⋈]*Carnegie Mellon University*
[⋆]*Eindhoven University of Technology* [▽]*IBM Research Europe*

Security Issues & Benefits

Security Issues in Processing in Memory

- Does PIM make security better or easier?
- Does PIM make security worse?
- Many interesting questions here
- Some recent papers:
  - Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System **[IISWC 2023]**
  - Amplifying Main Memory-Based Timing Covert and Side Channels using Processing-in-Memory Operations **[arxiv 2024]**

Homomorphic Operations on Real PIM Systems

- Harshita Gupta, Mayank Kabra, Juan Gómez-Luna, Konstantinos Kanellopoulos, and Onur Mutlu,
  "Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System"
  *Proceedings of the 2023 IEEE International Symposium on Workload Characterization Poster Session (**IISWC**)*, Ghent, Belgium, October 2023.
  [arXiv version]
  [Lightning Talk Slides (pptx) (pdf)]
  [Poster (pptx) (pdf)]

Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra* Juan Gómez-Luna Konstantinos Kanellopoulos Onur Mutlu
ETH Zürich

PIM Amplifies Covert & Side Channels

- Nisa Bostanci, Konstantinos Kanellopoulos, Ataberk Olgun, A. Giray Yaglikci, Ismail Emir Yuksel, Nika Mansouri Ghiasi, Zulal Bingol, Mohammad Sadrosadati, and Onur Mutlu,
  "Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory"
  *Proceedings of the 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Naples, Italy, June 2025.
  Officially artifact evaluated as available, reviewed, and reproduced.
  [IMPACT Source Code]

Revisiting Main Memory-Based Covert and Side Channel Attacks in the Context of Processing-in-Memory

F. Nisa Bostancı[†*] Konstantinos Kanellopoulos[†*] Ataberk Olgun[†]
A. Giray Yağlıkçı[†] İsmail Emir Yüksel[†] Nika Mansouri Ghiasi[†]
Zülal Bingöl[†‡] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†]ETH Zürich [‡]Bilkent University

https://arxiv.org/pdf/2404.11284

A Talk on Security of PIM Systems

https://www.youtube.com/watch?v=UjE9hygFXEM

Real PIM Systems

Eliminating the Adoption Barriers

Processing-in-Memory in the Real World

PIM Tutorial at ISCA 2024

ISCA 2024 Memory-Centric Computing Systems Tutorial

Saturday, June 29, Buenos Aires, Argentina

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/isca24-memorycentric-tutorial/

- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities

https://www.youtube.com/watch?v=KV2MXvcBgb0

https://events.safari.ethz.ch/isca24-memorycentric-tutorial

PIM Tutorial at MICRO 2024

MICRO 2024 - Tutorial on Memory-Centric Computing Systems

Saturday, November 2nd, Austin, Texas, USA

Organizers: Geraldo F. Oliveira, Dr. Mohammad Sadrosadati, Ataberk Olgun, Professor Onur Mutlu

Program: https://events.safari.ethz.ch/micro24-memorycentric-tutorial/

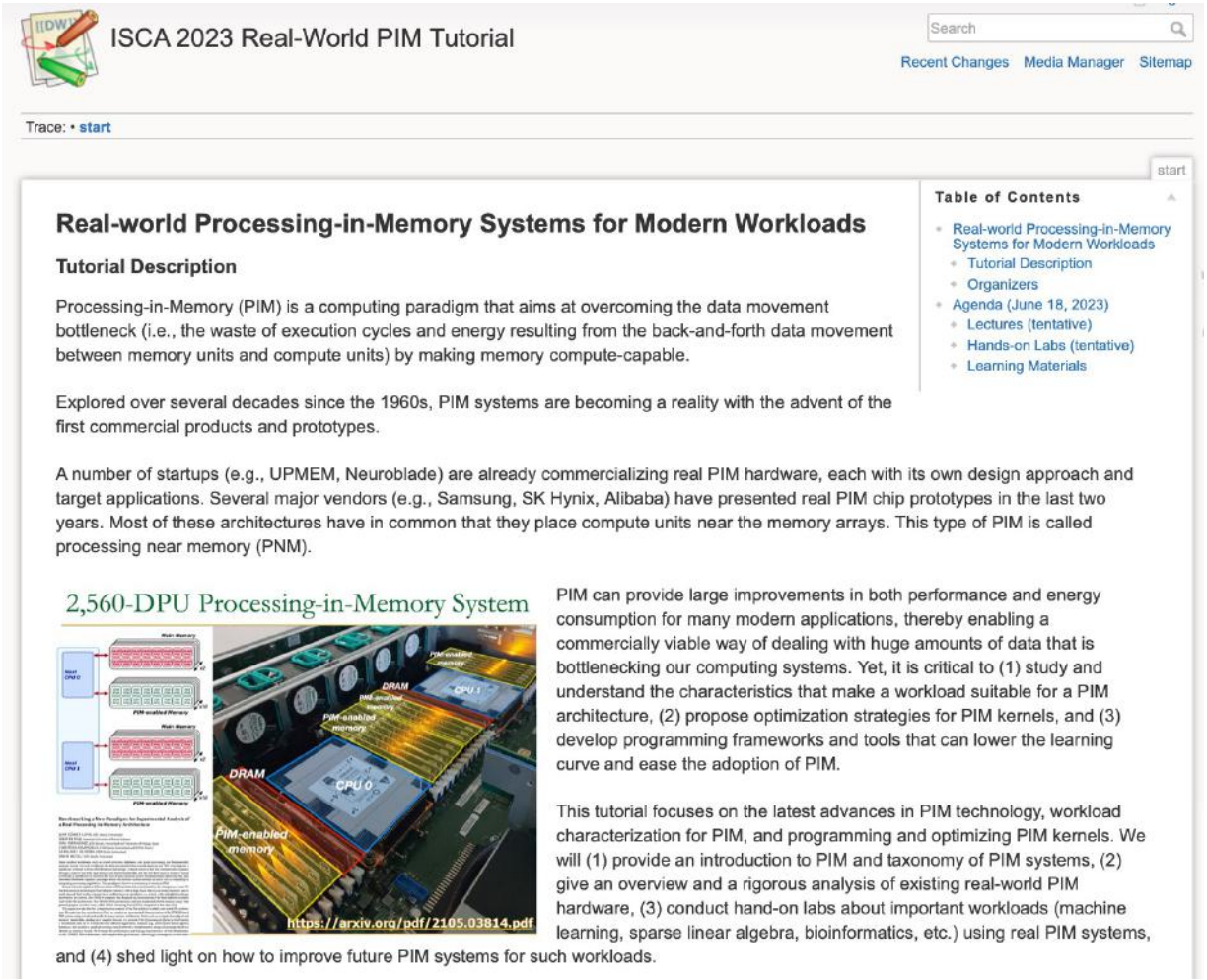
- Overview of PIM | PIM taxonomy
- PIM in memory & storage
- Real-world PNM systems
- PUM for bulk bitwise operations
- Programming techniques & tools
- Infrastructures for PIM Research
- Research challenges & opportunities

https://www.youtube.com/watch?v=KV2MXvcBgb0

https://events.safari.ethz.ch/micro24-memorycentric-tutorial/

PIM Tutorials [MICRO'23, ISCA'23, ASPLOS'23, HPCA'23, ISCA'24]

- Lectures + Hands-on labs + Invited talks



https://www.youtube.com/live/GIb5EgSrWk0

https://events.safari.ethz.ch/isca-pim-tutorial/

Real PIM Tutorial [ISCA 2023]

- June 18: Lectures + Hands-on labs + Invited talks

Tutorial Materials

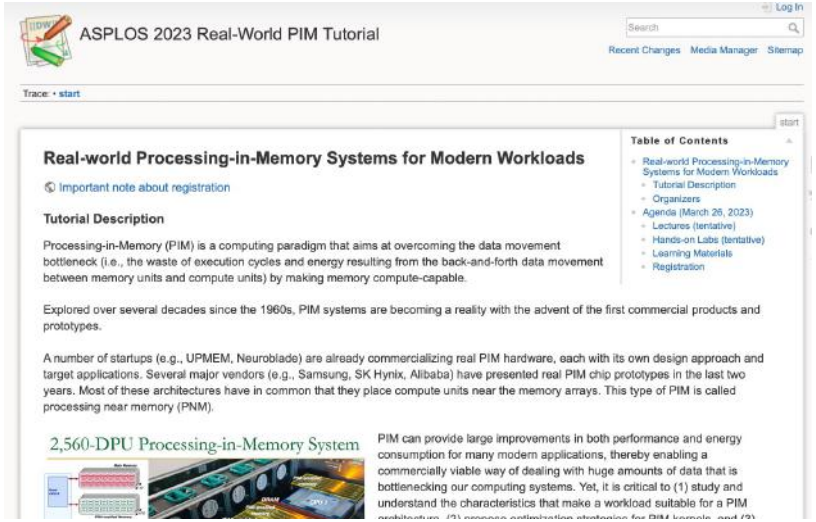
Time	Speaker	Title	Materials
8:55am-9:00am	Dr. Juan Gómez Luna	Welcome & Agenda	(PDF) (PPT)
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:20am-11:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures / Programming General-purpose PIM	(PDF) (PPT)
11:20am-11:50am	Prof. Izzat El Hajj	High-throughput Sequence Alignment using Real Processing-in-Memory Systems	(PDF) (PPT)
11:50am-12:30pm	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication for Real Processing-In-Memory Systems	(PDF) (PPT)
2:00pm-2:45pm	Dr. Sukhan Lee	Introducing Real-world HBM-PIM Powered System for Memory-bound Applications	(PDF) (PPT)
2:45pm-3:30pm	Dr. Juan Gómez Luna / Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components / PUM Prototypes: PiDRAM	(PDF) (PPT) (PDF) (PPT)
4:00pm-4:40pm	Dr. Juan Gómez Luna	Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT)
4:40pm-5:20pm	Dr. Juan Gómez Luna	Adoption Issues: How to Enable PIM?	(PDF) (PPT)
5:20pm-5:30pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

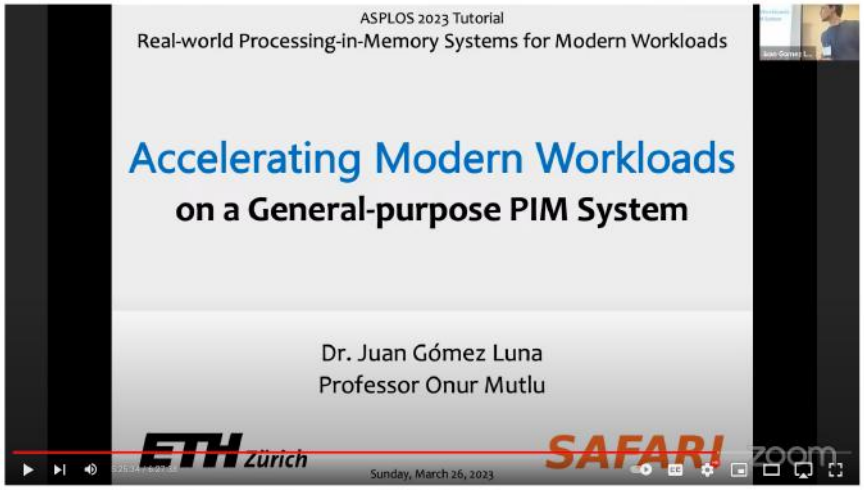
https://www.youtube.com/live/GIb5EgSrWk0

https://events.safari.ethz.ch/isca-pim-tutorial/

Real PIM Tutorial [ASPLOS 2023]

- March 26: Lectures + Hands-on labs + Invited talks





ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures — Subscribed — 33 — Share — Clip — Save

views Streamed 7 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
LOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads
://events.safari.ethz.ch/asplos-...

Tutorial Materials

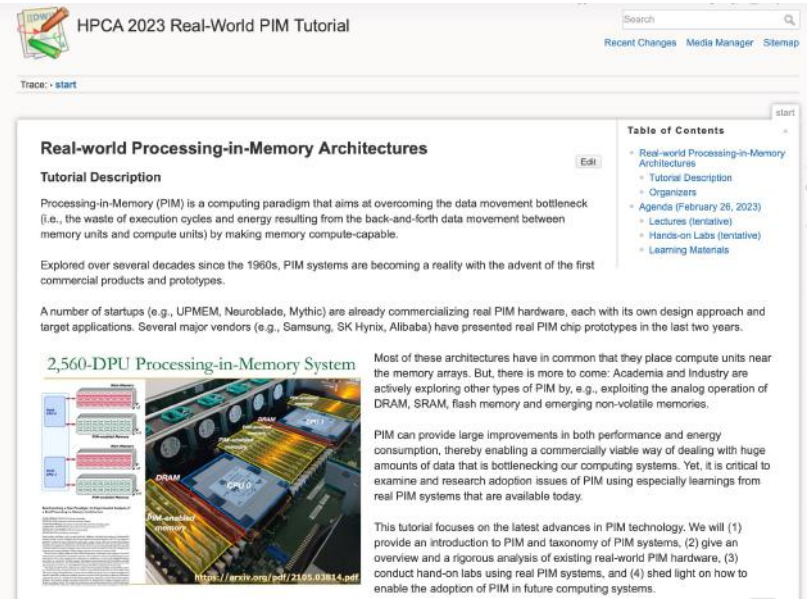
Time	Speaker	Title	Materials
9:00am-10:20am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
10:40am-12:00pm	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
1:40pm-2:20pm	Prof. Alexandra (Sasha) Fedorova (UBC)	Processing in Memory in the Wild	(PDF) (PPT)
2:20pm-3:20pm	Dr. Juan Gómez Luna & Ataberk Olgun	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT) (PDF) (PPT)
3:40pm-4:10pm	Dr. Juan Gómez Luna	Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System	(PDF) (PPT) (PDF) (PPT)
4:10pm-4:50pm	Dr. Yongkee Kwon & Eddy (Chanwook) Park (SK Hynix)	System Architecture and Software Stack for GDDR6-AiM	(PDF) (PPT)
4:50pm-5:00pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

https://www.youtube.com/watch?v=oYCaLcT0Kmo

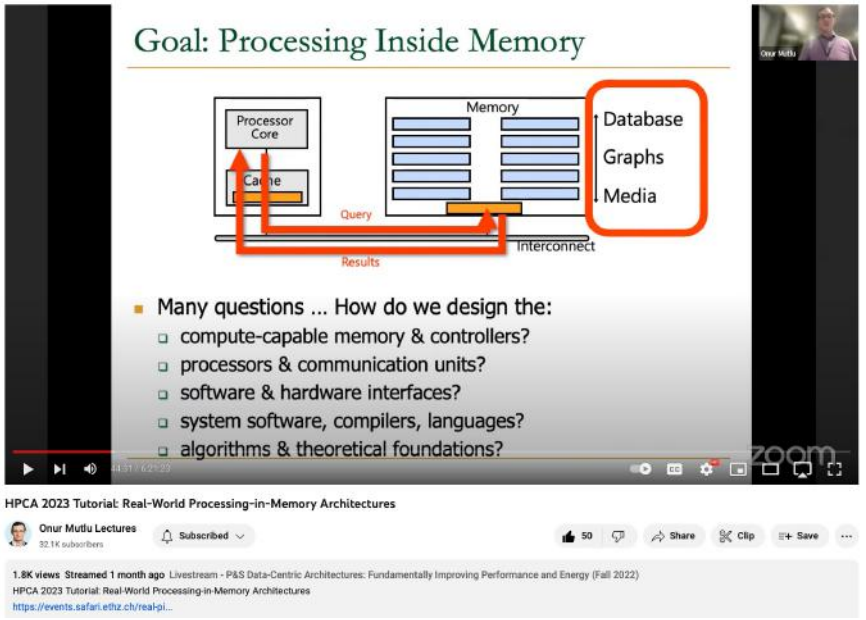
https://events.safari.ethz.ch/asplos-pim-tutorial/

Real PIM Tutorial [HPCA 2023]

- February 26: Lectures + Hands-on labs + Invited Talks



Time	Speaker	Title	Materials
8:00am-8:40am	Prof. Onur Mutlu	Memory-Centric Computing	(PDF) (PPT)
8:40am-10:00am	Dr. Juan Gómez Luna	Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM	(PDF) (PPT)
10:20am-11:00am	Dr. Dimin Niu	A 3D Logic-to-DRAM Hybrid Bonding Process-Near-Memory Chip for Recommendation System	
11:00am-11:40am	Dr. Christina Giannoula	SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures	(PDF) (PPT)
1:30pm-2:10pm	Dr. Juan Gómez Luna	Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components	(PDF) (PPT)
2:10pm-2:50pm	Dr. Manuel Le Gallo	Deep Learning Inference Using Computational Phase-Change Memory	
2:50pm-3:30pm	Dr. Juan Gómez Luna	PIM Adoption Issues: How to Enable PIM Adoption?	(PDF) (PPT)
3:40pm-5:40pm	Dr. Juan Gómez Luna	Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture	(Handout) (PDF) (PPT)

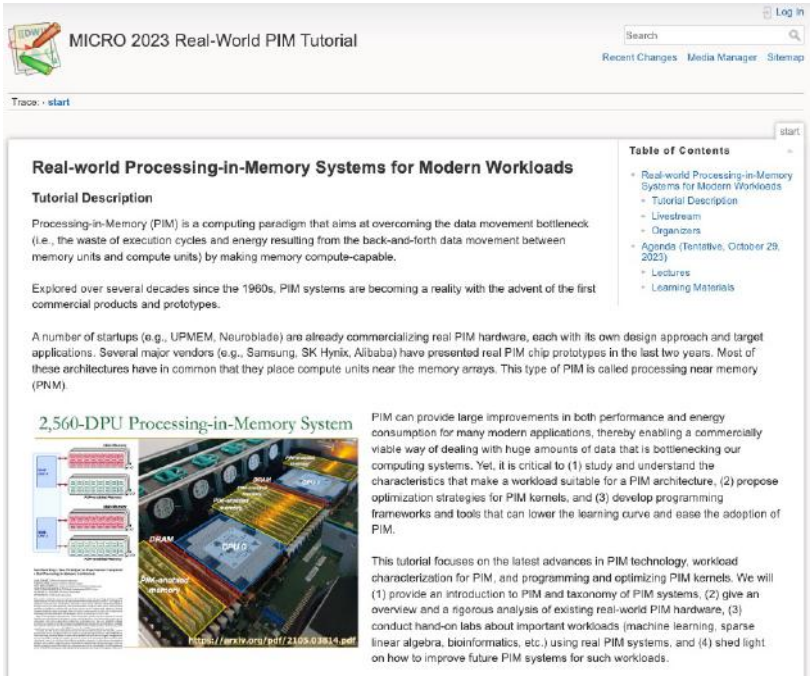


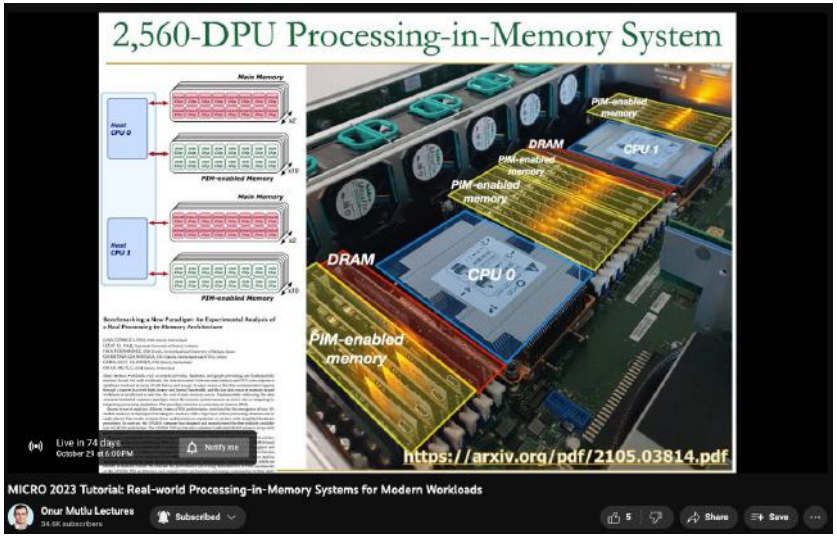
https://www.youtube.com/watch?v=f5-nT1tbz5w

https://events.safari.ethz.ch/real-pim-tutorial/

Real PIM Tutorial [MICRO 2023]

- **October 29**: Lectures + Hands-on labs + Invited talks





Agenda (Tentative, October 29, 2023)

Lectures

1. Introduction: PIM as a paradigm to overcome the data movement bottleneck.
2. PIM taxonomy: PNM (processing near memory) and PUM (processing using memory).
3. General-purpose PNM: UPMEM PIM.
4. PNM for neural networks: Samsung HBM-PIM, SK Hynix AiM.
5. PNM for recommender systems: Samsung AxDIMM, Alibaba PNM.
6. PUM prototypes: PiDRAM, SRAM-based PUM, Flash-based PUM.
7. Other approaches: Neuroblade, Mythic.
8. Adoption issues: How to enable PIM?
9. Hands-on labs: Programming a real PIM system.

https://www.youtube.com/watch?v=ohUooNSIxOI

https://events.safari.ethz.ch/micro-pim-tutorial

FPGA-based Processing Near Memory

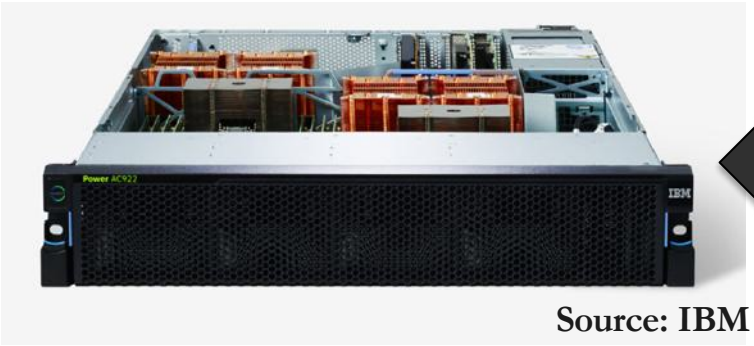
- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, **"FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"**
  IEEE Micro (***IEEE MICRO***), 2021.

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

Gagandeep Singh[◇] **Mohammed Alser**[◇] **Damla Senol Cali**[⋈]
Dionysios Diamantopoulos[▽] **Juan Gómez-Luna**[◇]
Henk Corporaal[⋆] **Onur Mutlu**[◇⋈]

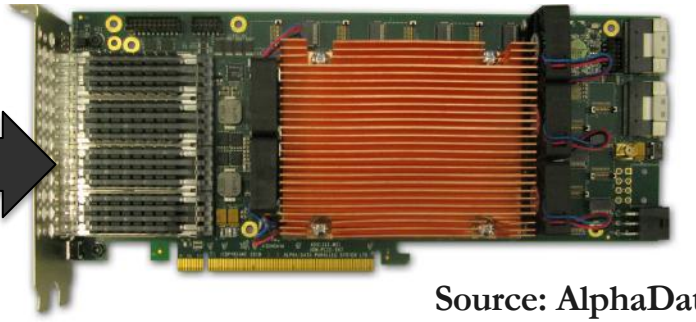
[◇]*ETH Zürich* [⋈]*Carnegie Mellon University*
[⋆]*Eindhoven University of Technology* [▽]*IBM Research Europe*

Near-Memory Acceleration using FPGAs



IBM POWER9 CPU

OCAPI

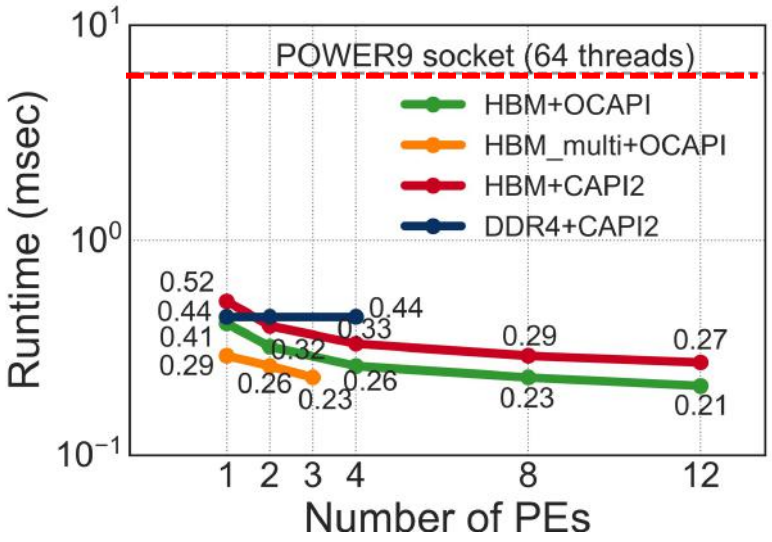


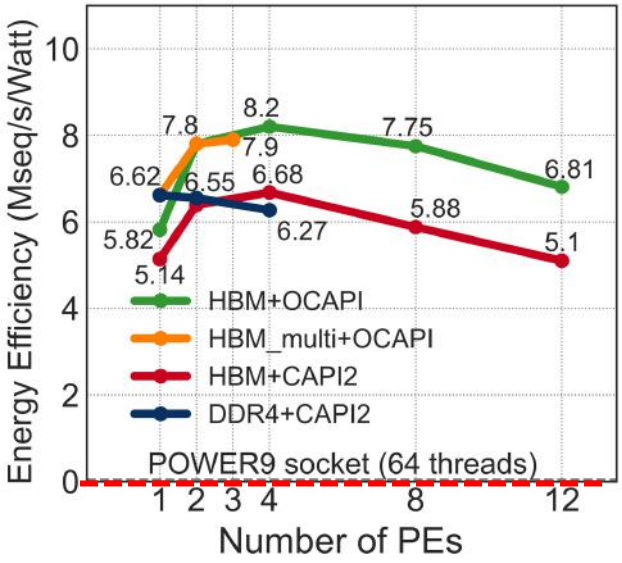
HBM-based FPGA board

Near-HBM FPGA-based accelerator

Two communication technologies: **CAPI2** and **OCAPI**
Two memory technologies: **DDR4** and **HBM**
Two workloads: **Weather Modeling** and **Genome Analysis**

Performance & Energy Greatly Improve





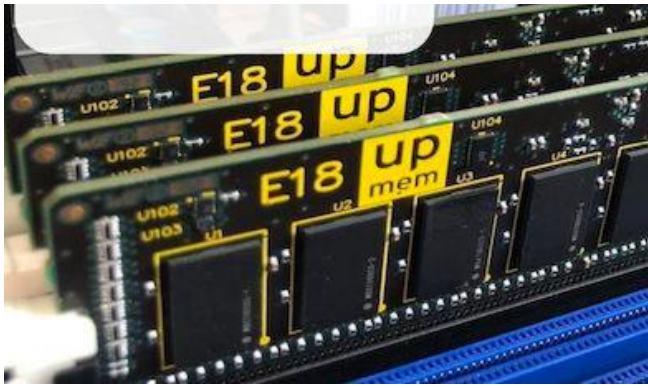
5-27× performance vs. a 16-core (64-thread) IBM POWER9 CPU

12-133× energy efficiency vs. a 16-core (64-thread) IBM POWER9 CPU

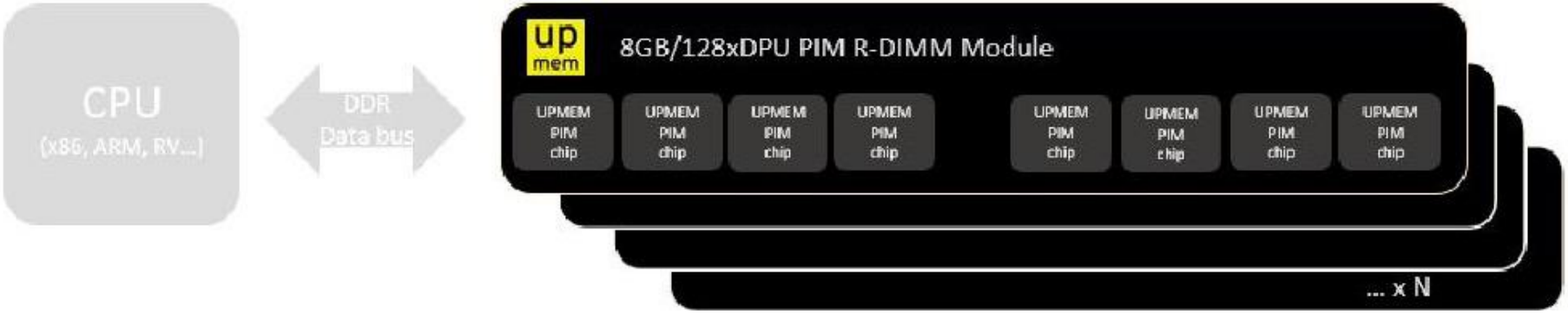
HBM alleviates memory bandwidth contention vs. DDR4

UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.



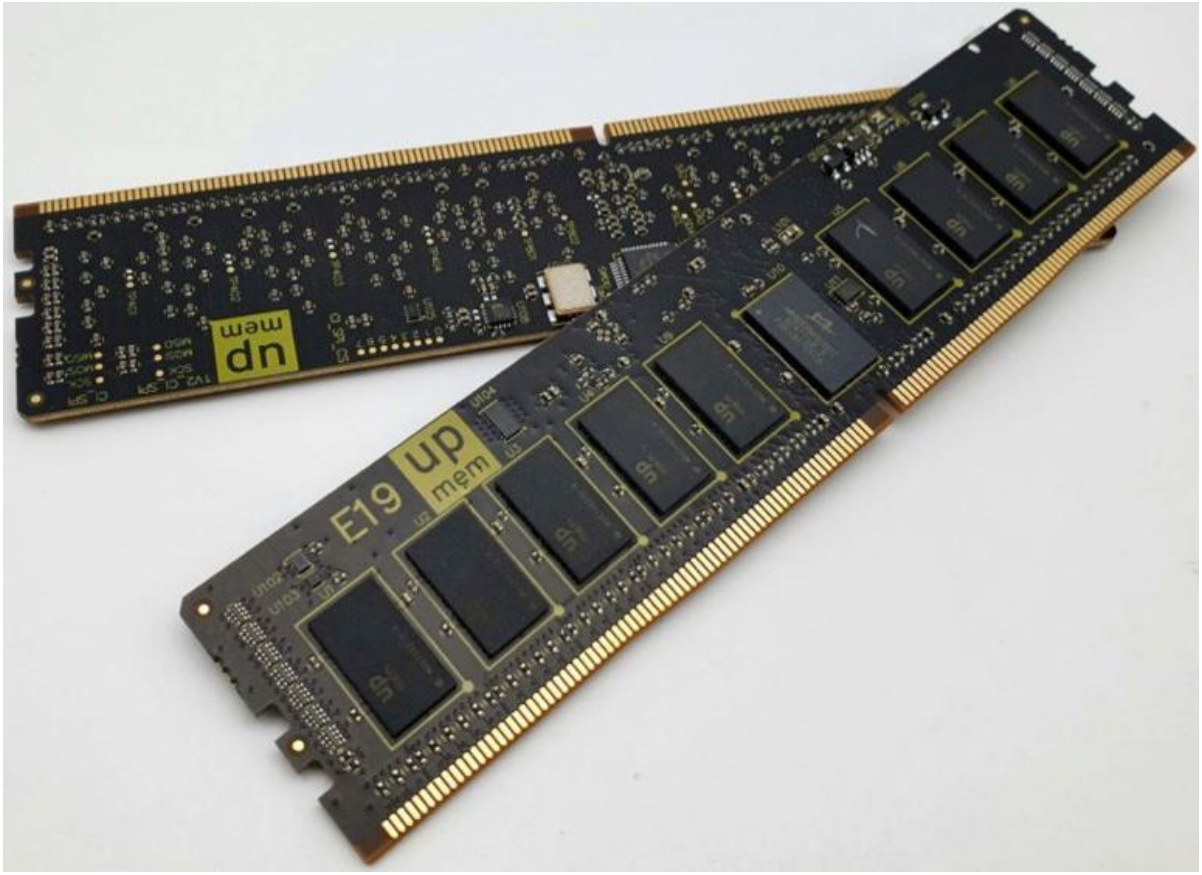
- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth



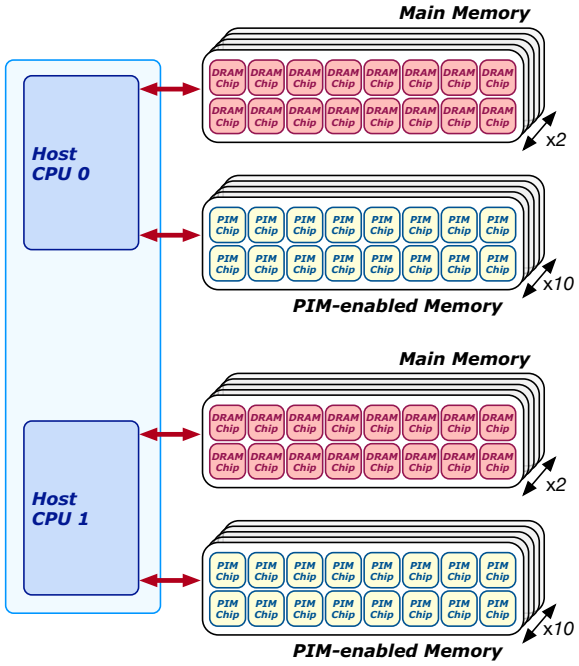
https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

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



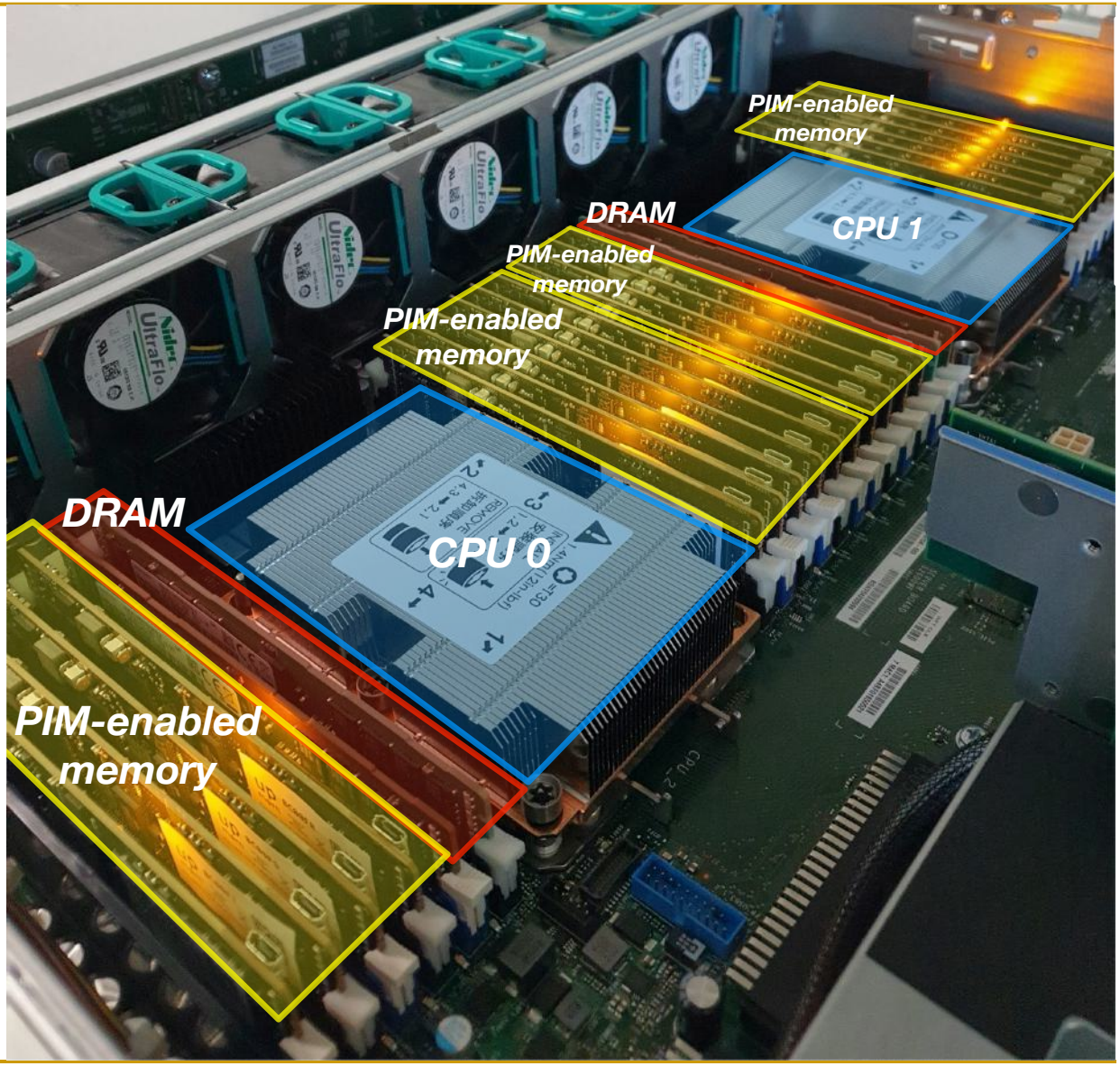
Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

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*).

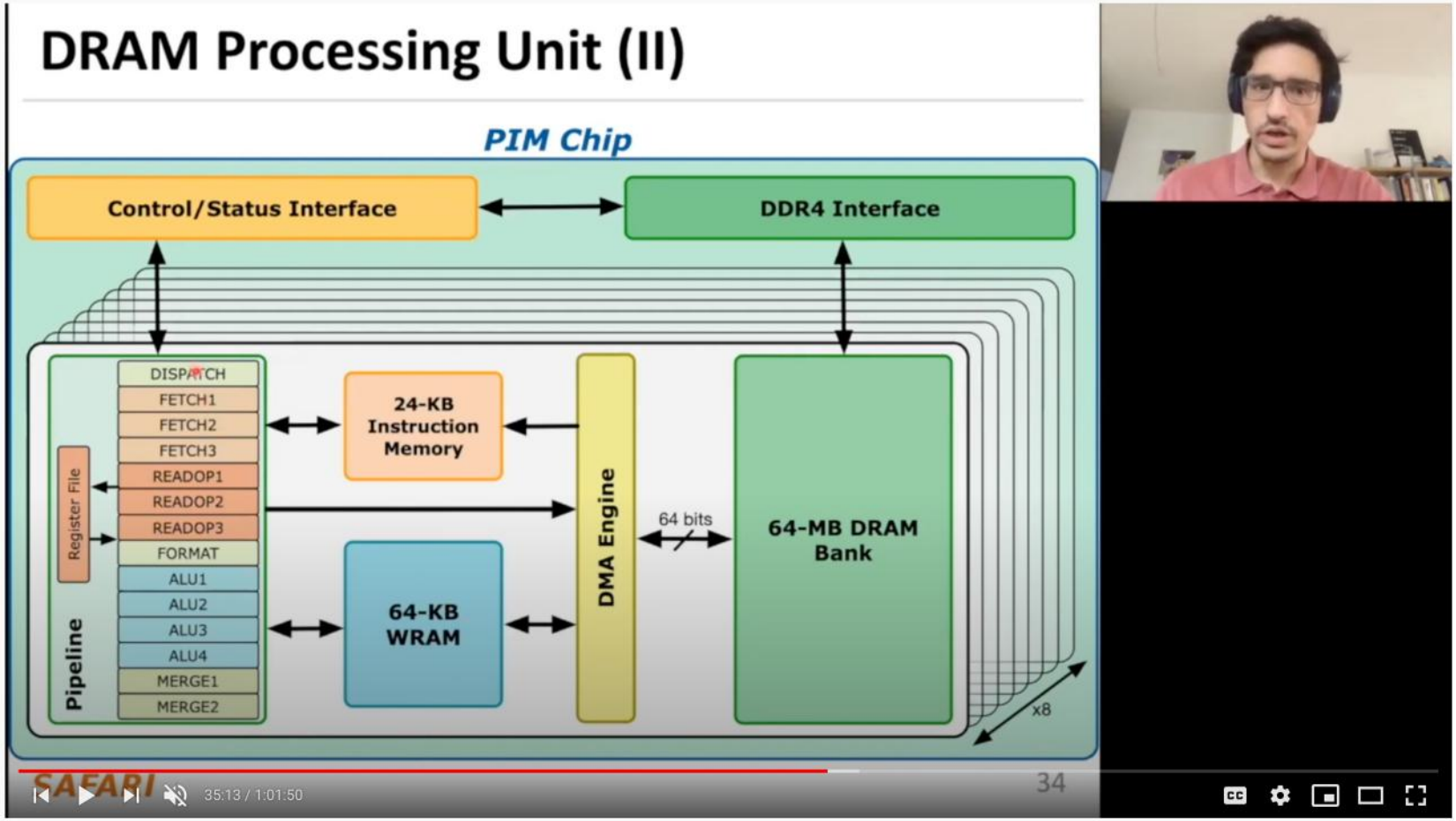
Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware 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 2,556 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

More on the UPMEM PIM System



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.

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

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

Recent SRC TECHCON Presentation

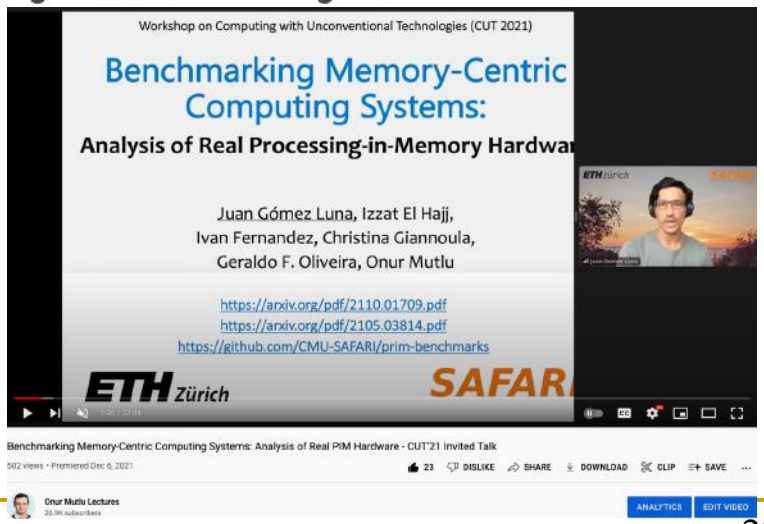
- Dr. Juan Gomez-Luna
  - Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
  - Based on two major works
    - https://arxiv.org/pdf/2105.03814.pdf
    - https://arxiv.org/pdf/2207.07886.pdf

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-In-Memory Hardware

Year: 2021, Pages: 1-7
DOI Bookmark: 10.1109/IGSC54211.2021.9651614

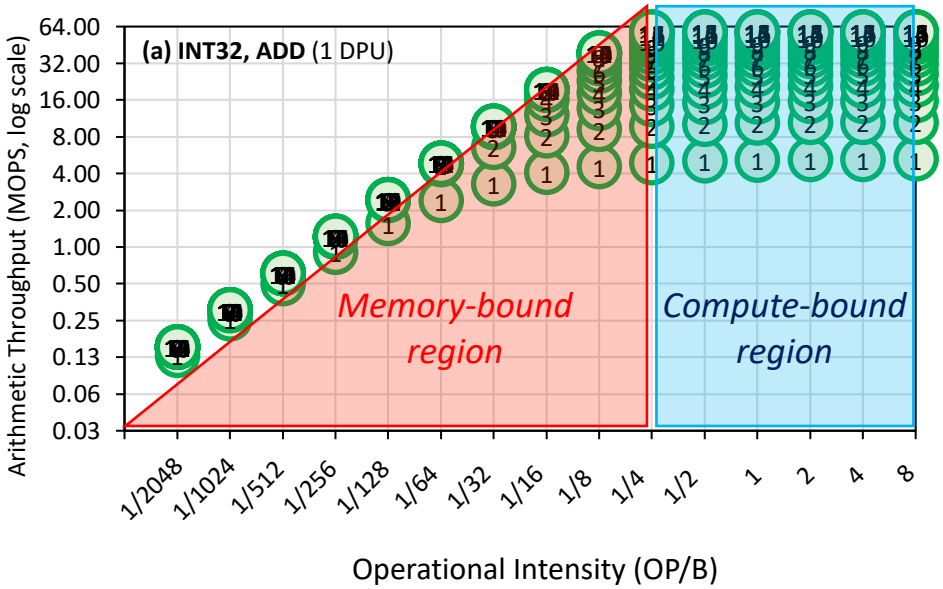
Authors

Juan Gómez-Luna, ETH Zürich
Izzat El Hajj, American University of Beirut
Ivan Fernandez, University of Malaga
Christina Giannoula, National Technical University of Athens
Geraldo F. Oliveira, ETH Zürich
Onur Mutlu, ETH Zürich



https://www.youtube.com/watch?v=nphV36SrysA

Key Takeaway 1



The throughput saturation point is as low as ¼ OP/B, i.e., 1 integer addition per every 32-bit element fetched

KEY TAKEAWAY 1

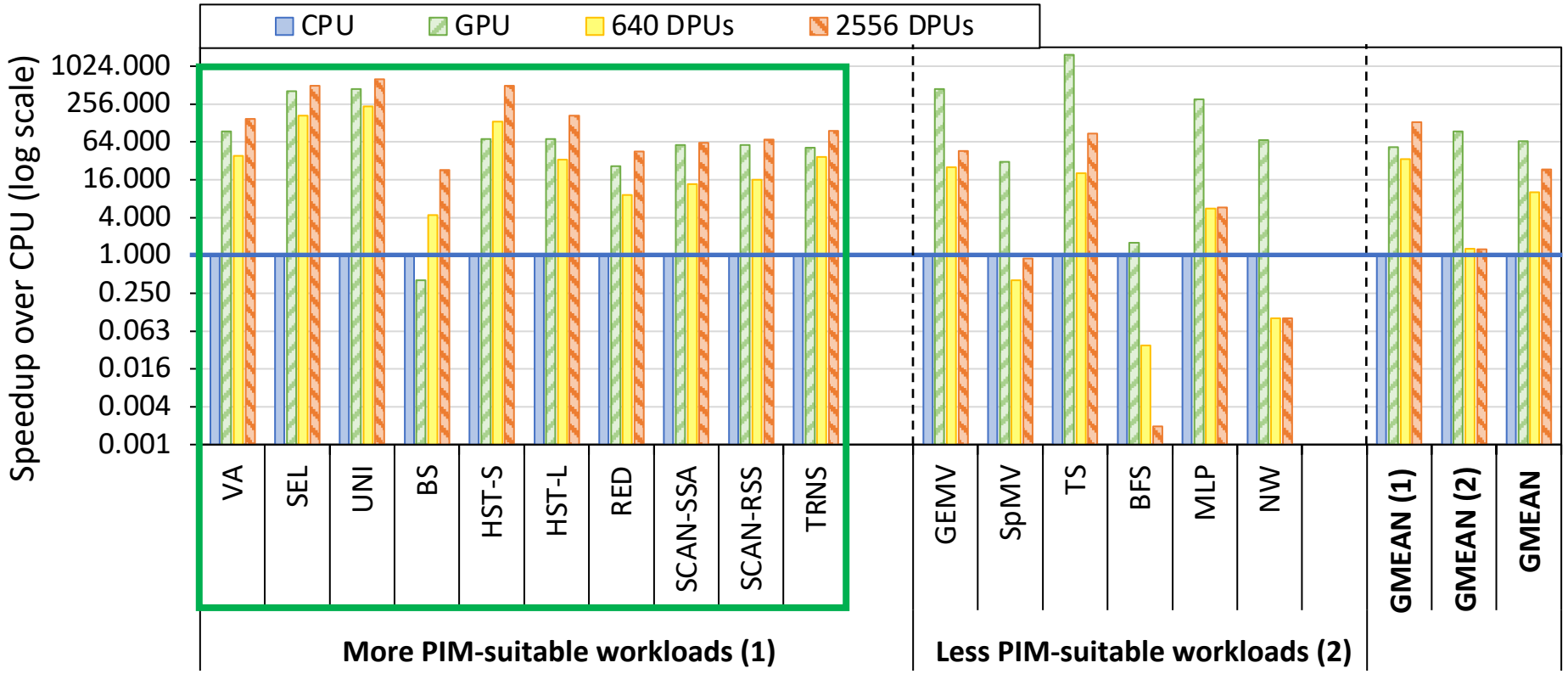
The UPMEM PIM architecture is fundamentally compute bound.
As a result, **the most suitable workloads are memory-bound.**

Key Takeaway 2

Table 4: Evaluated CPU, GPU, and UPMEM-based PIM Systems.

System	Process Node	Processor Cores			Memory		TDP
		Total Cores	Frequency	Peak Performance	Capacity	Total Bandwidth	
Intel Xeon E3-1225 v6 CPU [241]	14 nm	4 (8 threads)	3.3 GHz	26.4 GFLOPS*	32 GB	37.5 GB/s	73 W
NVIDIA Titan V GPU [277]	14 nm	80 (5,120 SIMD lanes)	1.2 GHz	12,288.0 GFLOPS	12 GB	652.8 GB/s	250 W
2,556-DPU PIM System	2x nm	2,556[9]	350 MHz	894.6 GOPS	159.75 GB	1.7 TB/s	383 W†
640-DPU PIM System	2x nm	640	267 MHz	170.9 GOPS	40 GB	333.75 GB/s	96 W†

*Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.
†Estimated TDP = $\frac{\text{Total DPUs}}{\text{DPUs/chip}}$ × 1.2 W/chip [199].



KEY TAKEAWAY 2

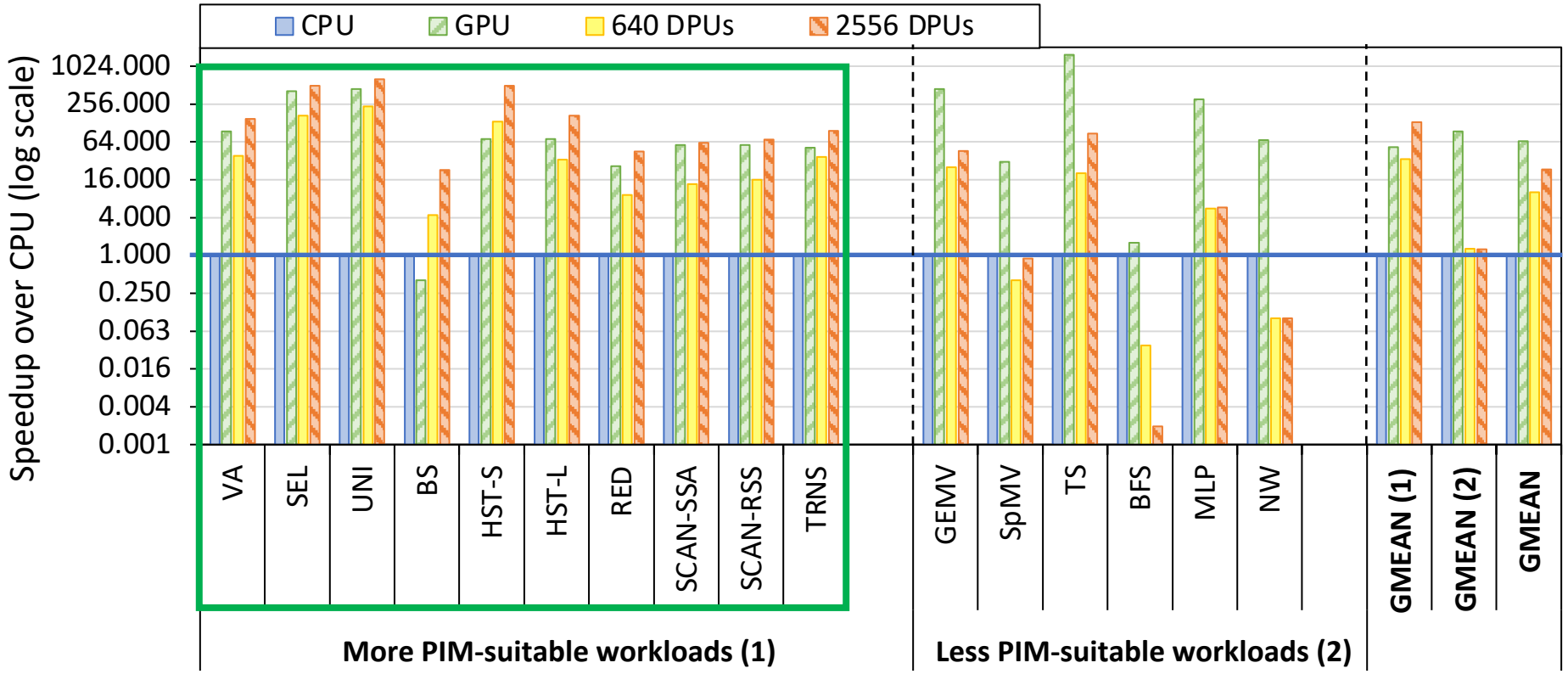
The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).

Key Takeaway 3

Table 4: Evaluated CPU, GPU, and UPMEM-based PIM Systems.

System	Process Node	Processor Cores			Memory		TDP
		Total Cores	Frequency	Peak Performance	Capacity	Total Bandwidth	
Intel Xeon E3-1225 v6 CPU [241]	14 nm	4 (8 threads)	3.3 GHz	26.4 GFLOPS*	32 GB	37.5 GB/s	73 W
NVIDIA Titan V GPU [277]	14 nm	80 (5,120 SIMD lanes)	1.2 GHz	12,288.0 GFLOPS	12 GB	652.8 GB/s	250 W
2,556-DPU PIM System	2x nm	2,556[9]	350 MHz	894.6 GOPS	159.75 GB	1.7 TB/s	383 W†
640-DPU PIM System	2x nm	640	267 MHz	170.9 GOPS	40 GB	333.75 GB/s	96 W†

*Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.
†Estimated TDP = $\frac{\text{Total DPUs}}{\text{DPUs/chip}}$ × 1.2 W/chip [199].



KEY TAKEAWAY 3

The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).

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

el1goluj@gmail.com

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

ETH zürich

SAFARI

UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
  "Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware"
  *Invited Paper at Workshop on Computing with Unconventional Technologies (**CUT**)*, Virtual, October 2021.
  [arXiv version]
  [PrIM Benchmarks Source Code]
  [Slides (pptx) (pdf)]
  [Talk Video (37 minutes)]
  [Lightning Talk Video (3 minutes)]


Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware

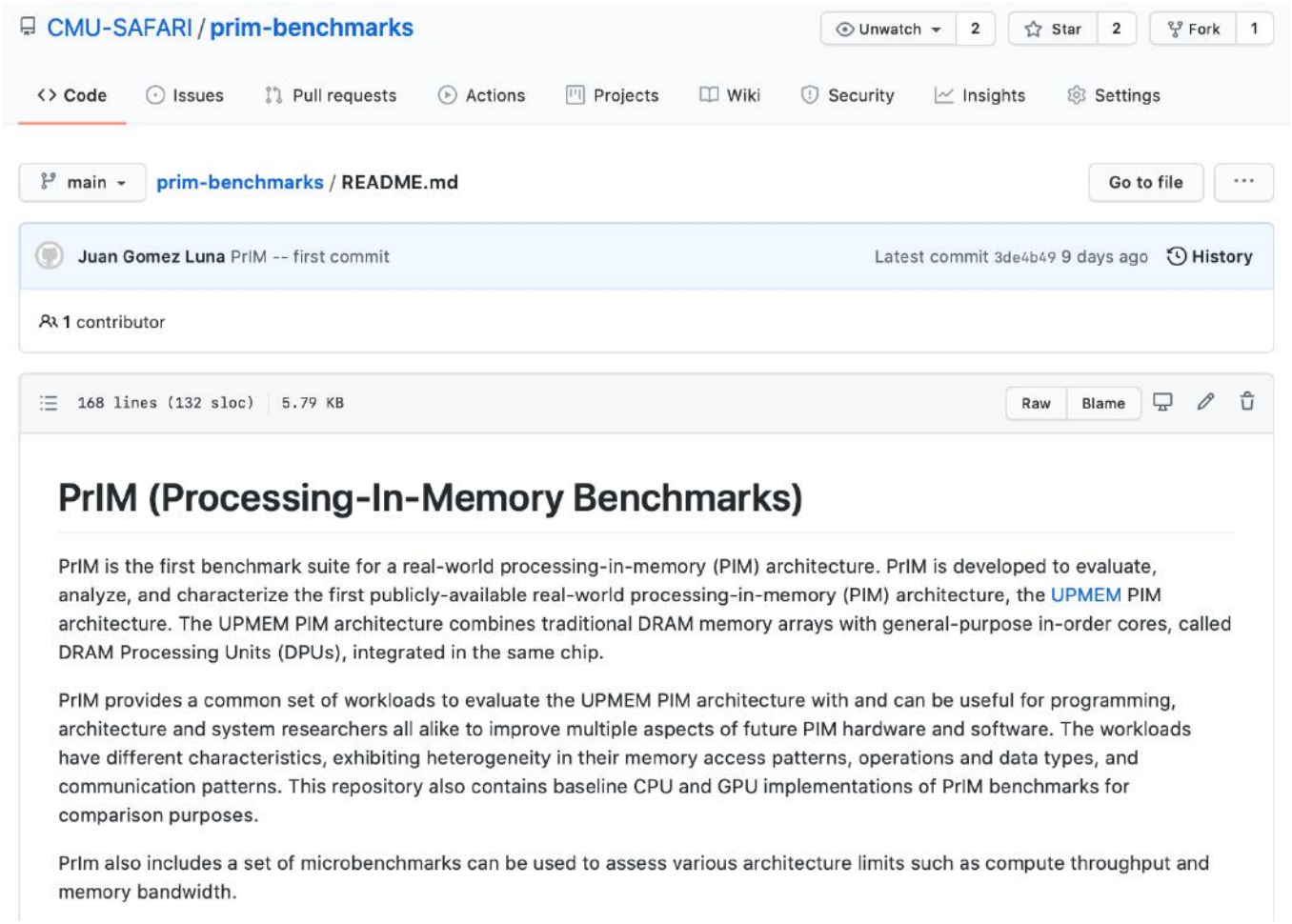
Juan Gómez-Luna, ETH Zürich
Izzat El Hajj, American University of Beirut
Ivan Fernandez, University of Malaga
Christina Giannoula, National Technical University of Athens
Geraldo F. Oliveira, ETH Zürich
Onur Mutlu, ETH Zürich

PrIM Benchmarks: Application Domains

Domain	Benchmark	Short name
Dense linear algebra	Vector Addition	VA
	Matrix-Vector Multiply	GEMV
Sparse linear algebra	Sparse Matrix-Vector Multiply	SpMV
Databases	Select	SEL
	Unique	UNI
Data analytics	Binary Search	BS
	Time Series Analysis	TS
Graph processing	Breadth-First Search	BFS
Neural networks	Multilayer Perceptron	MLP
Bioinformatics	Needleman-Wunsch	NW
Image processing	Image histogram (short)	HST-S
	Image histogram (large)	HST-L
Parallel primitives	Reduction	RED
	Prefix sum (scan-scan-add)	SCAN-SSA
	Prefix sum (reduce-scan-scan)	SCAN-RSS
	Matrix transposition	TRNS

PrIM Benchmarks are Open Source

- All microbenchmarks, benchmarks, and scripts
- https://github.com/CMU-SAFARI/prim-benchmarks



Understanding a Modern PIM Architecture

Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System

JUAN GÓMEZ-LUNA[1], IZZAT EL HAJJ[2], IVAN FERNANDEZ[1,3], CHRISTINA GIANNOULA[1,4], GERALDO F. OLIVEIRA[1], AND ONUR MUTLU[1]

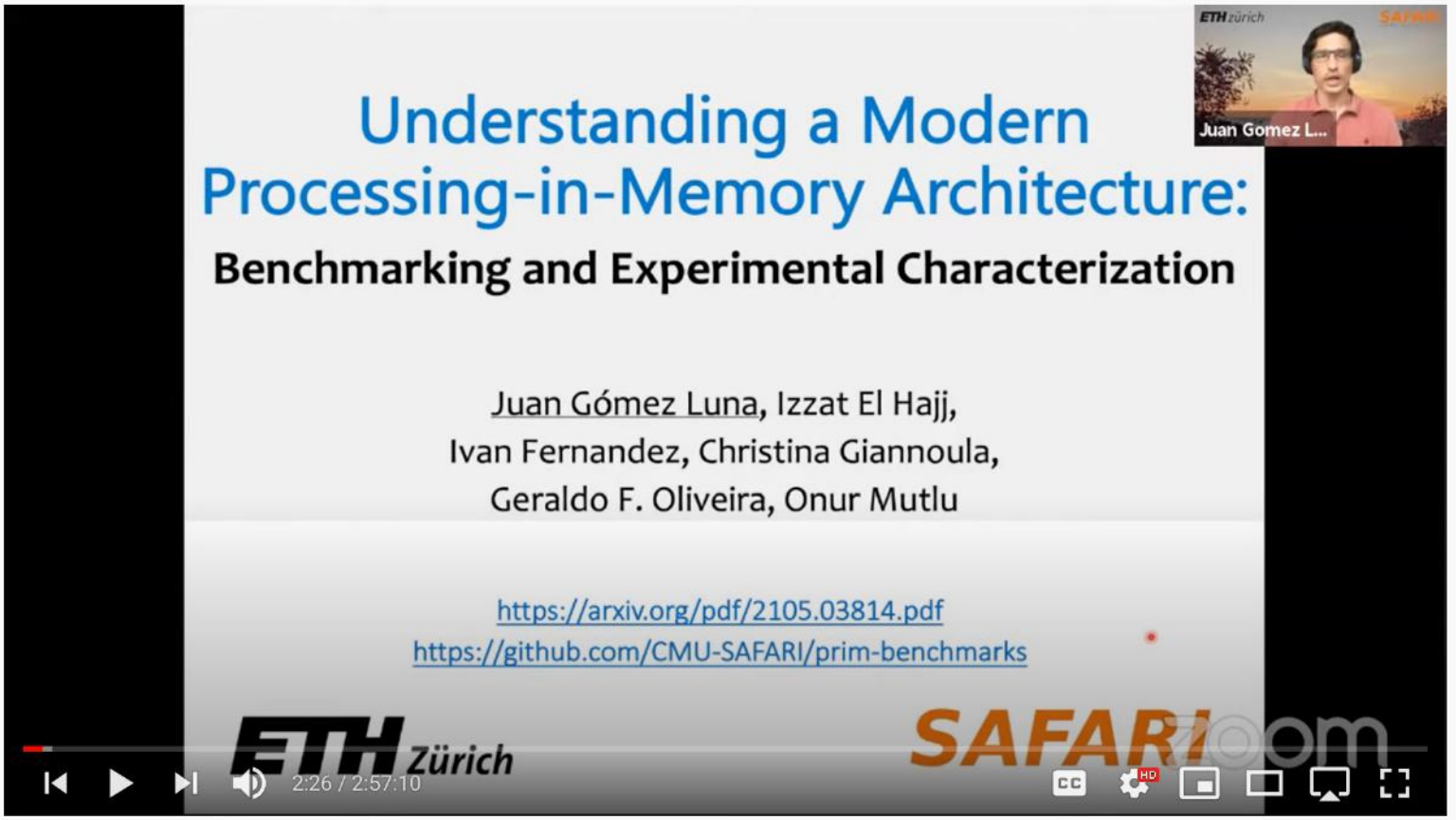
[1]ETH Zürich
[2]American University of Beirut
[3]University of Malaga
[4]National Technical University of Athens

Corresponding author: Juan Gómez-Luna (e-mail: juang@ethz.ch).

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

https://github.com/CMU-SAFARI/prim-benchmarks

Understanding a Modern PIM Architecture



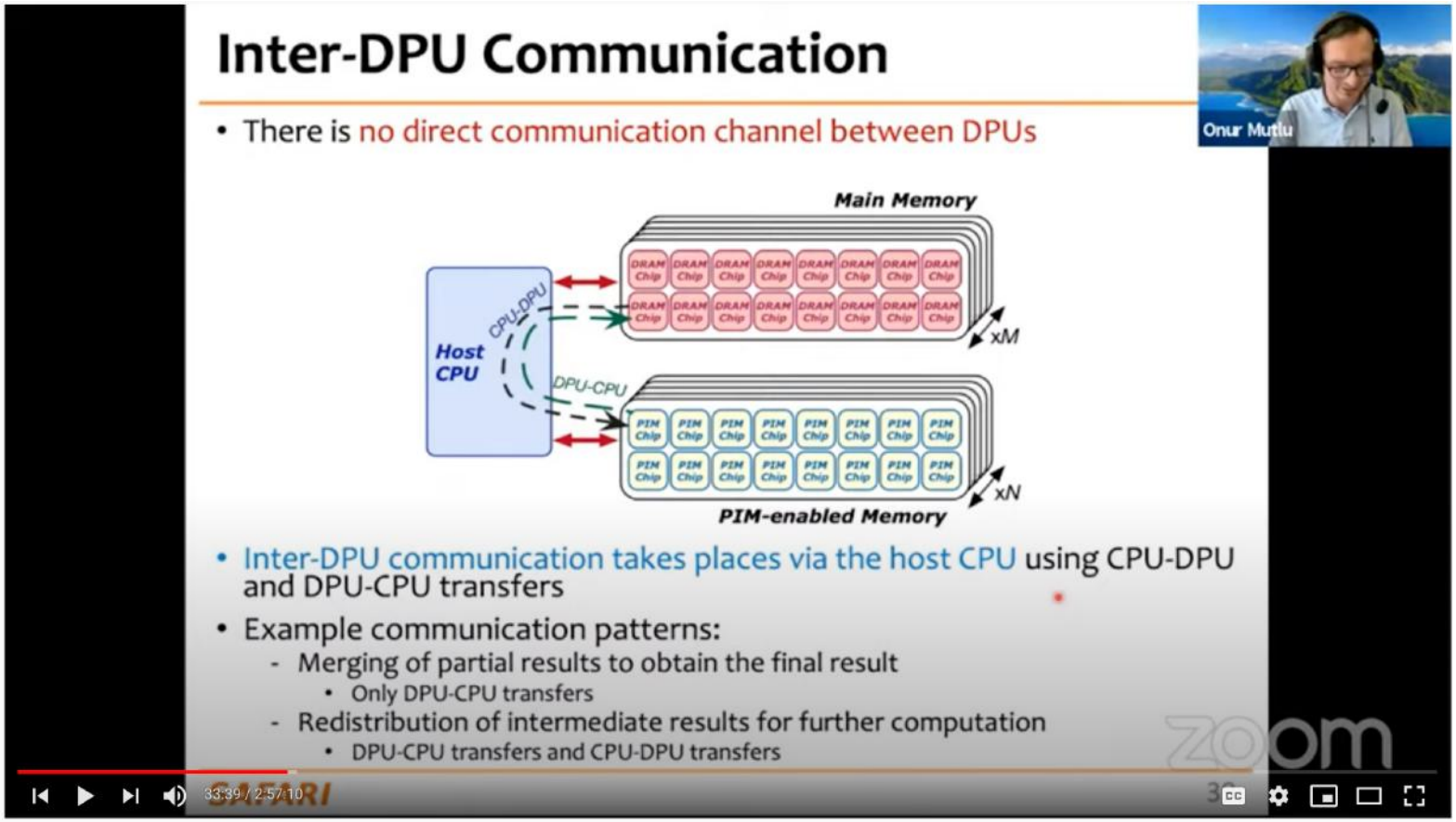
SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture

2,579 views • Streamed live on Jul 12, 2021 93 0 SHARE SAVE ...

Onur Mutlu Lectures
18.7K subscribers
SUBSCRIBED

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine



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

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

81 0 SHARE SAVE ...

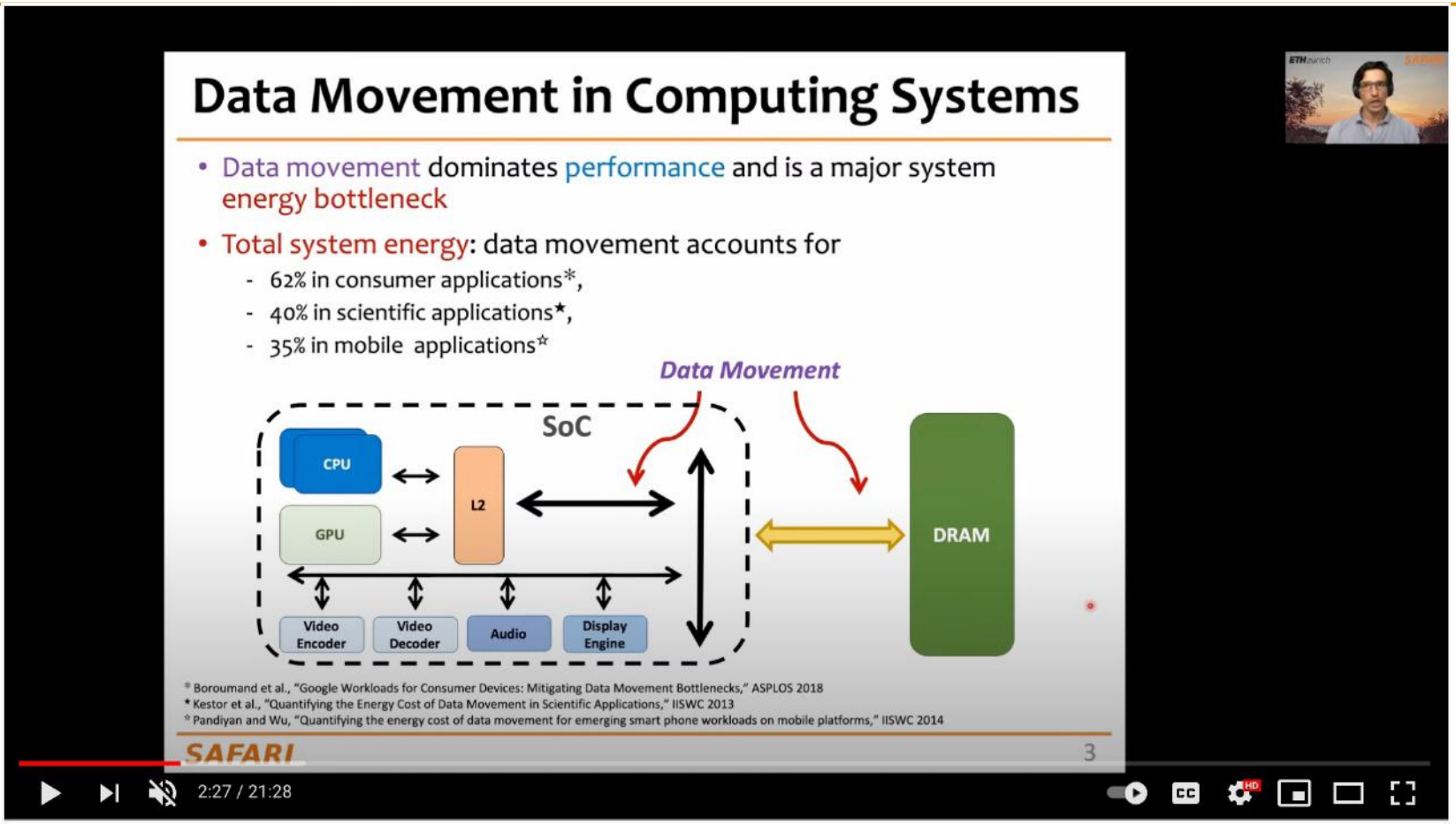
Onur Mutlu Lectures
17.6K subscribers

ANALYTICS EDIT VIDEO

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

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine



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

ML Training on a Real PIM System

Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]
[1]ETH Zürich [2]UPMEM

An Experimental Evaluation of Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]
[1]ETH Zürich [2]UPMEM

Short version: https://arxiv.org/pdf/2206.06022.pdf

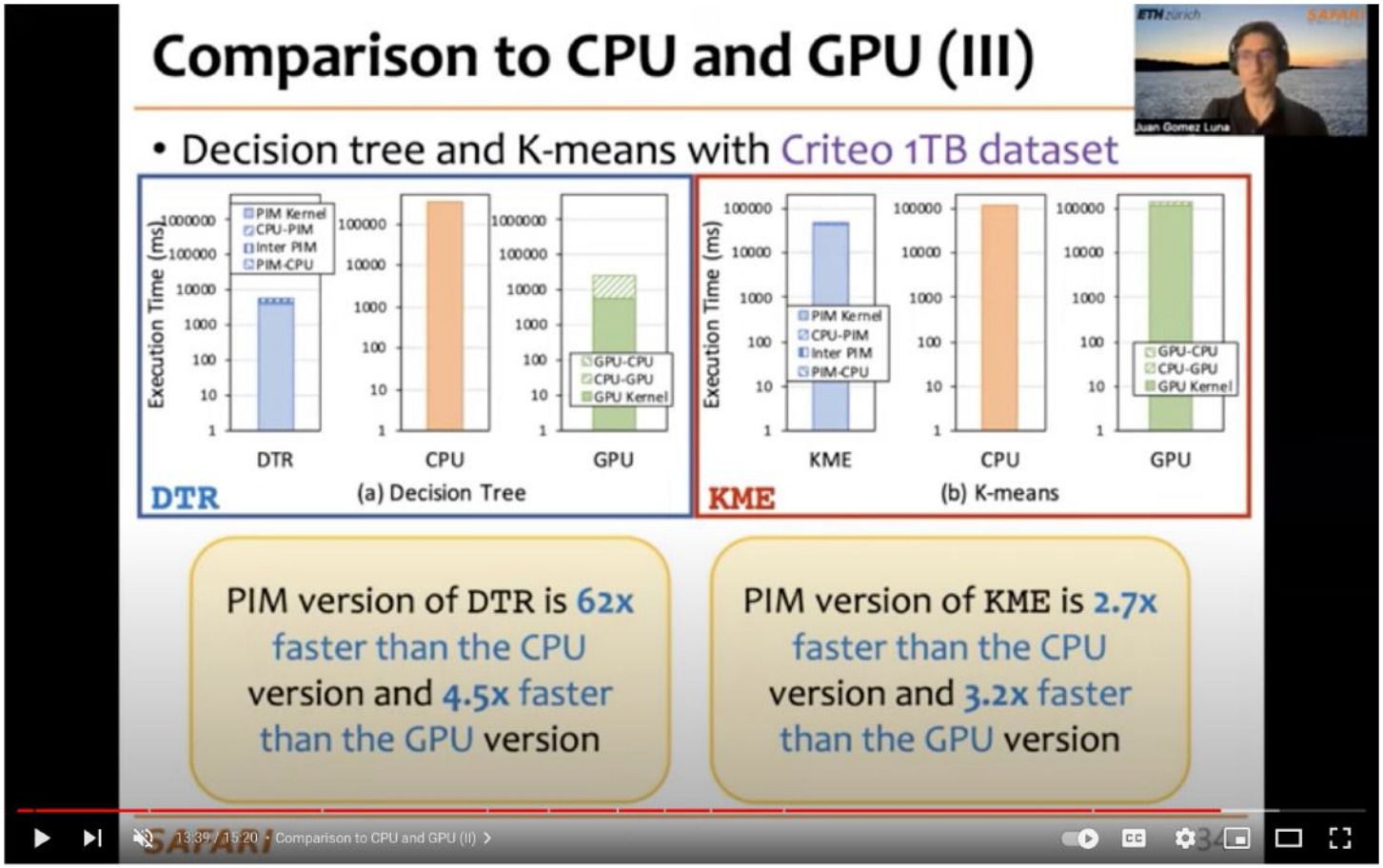
Long version: https://arxiv.org/pdf/2207.07886.pdf

https://www.youtube.com/watch?v=qeukNs5XI3g&t=11226s

ML Training on a Real PIM System

- **Need to optimize data representation**
  - (1) fixed-point
  - (2) quantization
  - (3) hybrid precision
- **Use lookup tables (LUTs) to implement complex functions (e.g., sigmoid)**
- **Optimize data placement & layout for streaming**
- **Large speedups:** 2.8X/27X vs. CPU, 1.3x/3.2x vs. GPU

ML Training on Real PIM Talk Video



Machine Learning Training on Memory-centric Computing Systems, Juan Gómez-Luna for ISPASS 2023

Onur Mutlu Lectures
32.9K subscribers
Analytics | Edit video | 9 | Share | Download | Clip | Save

242 views 11 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
Evaluating Machine Learning Workloads on Memory-centric Computing Systems

https://www.youtube.com/watch?v=60pkaI5AeM4

ML Training on Real PIM Systems

- Juan Gómez Luna, Yuxin Guo, Sylvan Brocard, Julien Legriel, Remy Cimadomo, Geraldo F. Oliveira, Gagandeep Singh, and Onur Mutlu,
  "Evaluating Machine Learning Workloads on Memory-Centric Computing Systems"
  *Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Raleigh, North Carolina, USA, April 2023.
  [arXiv version, 16 July 2022.]
  [PIM-ML Source Code]
  Best paper session.

An Experimental Evaluation of Machine Learning Training on a Real Processing-in-Memory System

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]
[1]ETH Zürich [2]UPMEM

https://github.com/CMU-SAFARI/pim-ml

https://arxiv.org/pdf/2207.07886.pdf

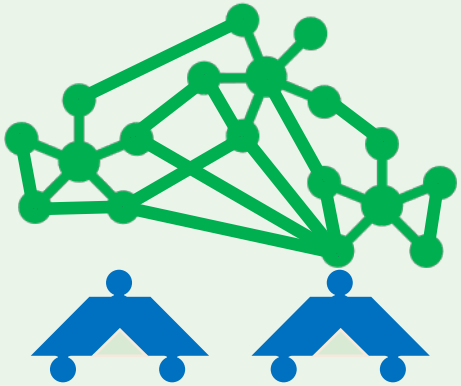
SpMV Multiplication on Real PIM Systems

- Appears at SIGMETRICS 2022

SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Systems

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and National Technical University of Athens, Greece
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
NECTARIOS KOZIRIS, National Technical University of Athens, Greece
GEORGIOS GOUMAS, National Technical University of Athens, Greece
ONUR MUTLU, ETH Zürich, Switzerland

https://arxiv.org/pdf/2201.05072.pdf
https://github.com/CMU-SAFARI/SparseP



SparseP

Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures

Christina Giannoula

Ivan Fernandez, Juan Gomez-Luna,

Nectarios Koziris, Georgios Goumas, Onur Mutlu

SAFARI ETH zürich National Technical University of Athens CSLab UNIVERSIDAD DE MÁLAGA

SparseP: Key Contributions

1. Efficient SpMV kernels for current & future PIM systems
   - SparseP library = 25 SpMV kernels
     - Compression, data types, data partitioning, synchronization, load balancing

SparseP is Open-Source

SparseP: https://github.com/CMU-SAFARI/SparseP

2. Comprehensive analysis of SpMV on the first commercially-available real PIM system

   - 26 sparse matrices
   - Comparisons to state-of-the-art CPU and GPU systems
   - Recommendations for software, system and hardware designers

Recommendations for Architects and Programmers

Full Paper: https://arxiv.org/pdf/2201.05072.pdf

SparseP Talk Video

Processing-in-Memory Course: Lecture 11: SpMV on a Real PIM Architecture - Spring 2022

149 views • Streamed live on May 19, 2022 12 DISLIKE SHARE DOWNLOAD CLIP SAVE ...

Onur Mutlu Lectures
25K subscribers

ANALYTICS EDIT VIDEO

https://www.youtube.com/watch?v=5kaOsJKlGrE

More on SparseP

Christina Giannoula, Ivan Fernandez, Juan Gomez-Luna, Nectarios Koziris, Georgios Goumas, and Onur Mutlu,
"SparseP: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures"
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Mumbai, India, June 2022.
[Extended arXiv Version]
[Abstract]
[Slides (pptx) (pdf)]
[Long Talk Slides (pptx) (pdf)]
[SparseP Source Code]
[Talk Video (16 minutes)]
[Long Talk Video (55 minutes)]

***SparseP*: Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Systems**

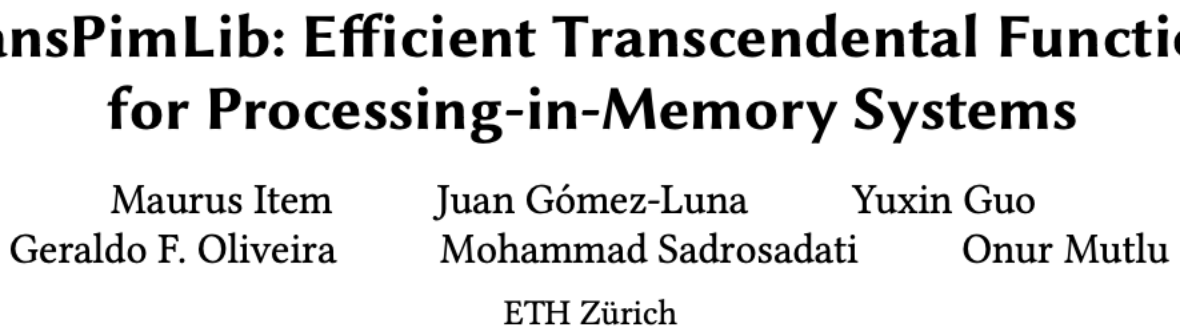
CHRISTINA GIANNOULA, ETH Zürich, Switzerland and National Technical University of Athens, Greece
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
NECTARIOS KOZIRIS, National Technical University of Athens, Greece
GEORGIOS GOUMAS, National Technical University of Athens, Greece
ONUR MUTLU, ETH Zürich, Switzerland

https://github.com/CMU-SAFARI/SparseP

https://arxiv.org/pdf/2201.05072.pdf

Transcendental Functions on Real PIM Systems

- Maurus Item, Juan Gómez Luna, Yuxin Guo, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu,
  "TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems"
  *Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Raleigh, North Carolina, USA, April 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [TransPimLib Source Code]
  [Talk Video (17 minutes)]



https://github.com/CMU-SAFARI/transpimlib

https://arxiv.org/pdf/2304.01951.pdf

Sequence Alignment on Real PIM Systems

- Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, and Izzat El Hajj,
  "A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems"
  Bioinformatics, [published online on] 27 March 2023.
  [Online link at Bioinformatics Journal]
  [arXiv preprint]
  [AiM Source Code]

A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab [1] Amir Nassereldine [1] Mohammed Alser [2] Juan Gómez Luna [2]
Onur Mutlu [2] Izzat El Hajj [1]

[1]American University of Beirut [2]ETH Zürich

https://github.com/CMU-SAFARI/alignment-in-memory

https://arxiv.org/pdf/2208.01243.pdf

Summary

- Sequence alignment on traditional systems is limited by the **memory bandwidth bottleneck**
- **Processing-in-memory (PIM)** overcomes this bottleneck by placing cores near the memory
- Our framework, **Alignment-in-Memory (AIM)**, is a PIM framework that supports multiple alignment algorithms (NW, SWG, GenASM, WFA)
  - Implemented on UPMEM, the first real PIM system
- Results show **substantial speedups over both CPUs (1.8X-28X) and GPUs (1.2X-2.7X)**
- AIM is available at:
  - https://github.com/CMU-SAFARI/alignment-in-memory

Better Sequence Alignment on Real PIM Systems

- Alejandro Alonso-Marín, Ivan Fernandez, Quim Aguado-Puig, Juan Gómez-Luna, Santiago Marco-Sola, Onur Mutlu, and Miquel Moreto,
  "BIMSA: Accelerating Long Sequence Alignment Using Processing-In-Memory"
  Bioinformatics, [published online on] 21 October 2024.
  [Online link at Bioinformatics Journal]
  [biorXiv version]
  [BIMSA Source Code]

BIMSA: Accelerating Long Sequence Alignment Using Processing-In-Memory

Alejandro Alonso-Marín[1,2,3*], Ivan Fernandez[1,4], Quim Aguado-Puig[1,3,5], Juan Gómez-Luna[6], Santiago Marco-Sola[1,2], Onur Mutlu[7], Miquel Moreto[1,4]

[1]Computer Sciences Department, Barcelona Supercomputing Center, Barcelona, 08034, Spain.
[2]Department of Computer Science, Universitat Politècnica de Catalunya, Barcelona, 08034, Spain.
[3]Department of Electronic Engineering, Universitat Politècnica de Catalunya, Barcelona, 08034, Spain.
[4]Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, 08034, Spain.
[5]Departament d'Arquitectura de Computadors i Sistemes Operatius, Universitat Autònoma de Barcelona, Barcelona, 08193, Spain.
[6]NVIDIA Corporation, Santa Clara, California, US.
[7]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8006, Switzerland.

https://arxiv.org/pdf/2208.01243.pdf

Homomorphic Operations on Real PIM Systems

- Harshita Gupta, Mayank Kabra, Juan Gómez-Luna, Konstantinos Kanellopoulos, and Onur Mutlu,
  "Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System"
  *Proceedings of the 2023 IEEE International Symposium on Workload Characterization Poster Session (**IISWC**)*, Ghent, Belgium, October 2023.
  [arXiv version]
  [Lightning Talk Slides (pptx) (pdf)]
  [Poster (pptx) (pdf)]

Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra* Juan Gómez-Luna Konstantinos Kanellopoulos Onur Mutlu
ETH Zürich

Accelerating Reinforcement Learning

- Kailash Gogineni, Sai Santosh Dayapule, Juan Gomez-Luna, Karthikeya Gogineni, Peng Wei, Tian Lan, Mohammad Sadrosadati, Onur Mutlu, Guru Venkataramani, **"SwiftRL: Towards Efficient Reinforcement Learning on Real Processing-In-Memory Systems"**
  *Proceedings of the 2024 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Indianapolis, Indiana, May 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]

SwiftRL: Towards Efficient Reinforcement Learning on Real Processing-In-Memory Systems

Kailash Gogineni[1] Sai Santosh Dayapule[1] Juan Gómez-Luna[2] Karthikeya Gogineni[3]
Peng Wei[1] Tian Lan[1] Mohammad Sadrosadati[2] Onur Mutlu[2] Guru Venkataramani[1]
[1]George Washington University, USA [2]ETH Zürich, Switzerland [3]Independent

https://arxiv.org/pdf/2405.03967

Accelerating ML Training on Real PIM Systems

- Steve Rhyner, Haocong Luo, Juan Gómez-Luna, Mohammad Sadrosadati, Jiawei Jiang, Ataberk Olgun, Harshita Gupta, Ce Zhang, and Onur Mutlu,
  "PIM-Opt: Demystifying Distributed Optimization Algorithms on a Real-World Processing-In-Memory System"
  *Proceedings of the 33rd International Conference on Parallel Architectures and Compilation Techniques (**PACT**)*, Long Beach, CA, USA, October 2024.
  [Slides (pptx) (pdf)]
  [PIM-Opt Source Code]
  [arXiv version]

Analysis of Distributed Optimization Algorithms on a Real Processing-In-Memory System

Steve Rhyner[1] Haocong Luo[1] Juan Gómez-Luna[2] Mohammad Sadrosadati[1]
Jiawei Jiang[3] Ataberk Olgun[1] Harshita Gupta[1] Ce Zhang[4] Onur Mutlu[1]
[1]ETH Zurich [2]NVIDIA [3]Wuhan University [4]University of Chicago

Accelerating ML Training on Real PIM Systems

- **Appears at PACT 2024**

8. Conclusion

We evaluate and train ML models on large-scale datasets with centralized parallel optimization algorithms on a *real-world* PIM architecture. We show the importance of carefully *choosing* the distributed optimization algorithm that fits PIM and analyze tradeoffs. We demonstrate that *commercial* general-purpose PIM systems can be a viable alternative for many ML training workloads on large-scale datasets to processor-centric architectures. Our results demonstrate the necessity of adapting PIM architectures to enable inter-DPU communication to overcome scalability challenges for many ML training workloads and discuss decentralized parallel SGD optimization algorithms as a potential solution.

https://arxiv.org/pdf/2404.07164

Accelerating GNNs on Real PIM Systems

- Christina Giannoula, Peiming Yang, Ivan Fernandez, Jiacheng Yang, Sankeerth Durvasula, Yu Xin Li, Mohammad Sadrosadati, Juan Gomez Luna, Onur Mutlu, and Gennady Pekhimenko,
  "PyGim: An Efficient Graph Neural Network Library for Real Processing-In-Memory Architectures"
  *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Stony Brook, NY, USA, June 2025.
  [PyGim Source Code]

PyGim: An Efficient Graph Neural Network Library for Real Processing-In-Memory Architectures

CHRISTINA GIANNOULA, University of Toronto, Canada, ETH Zürich, Switzerland, Vector Institute, Canada, and CentML, Canada
PEIMING YANG, University of Toronto, Canada
IVAN FERNANDEZ, Barcelona Supercomputing Center, Spain, Universitat Politècnica de Catalunya, Spain, and ETH Zürich, Switzerland
JIACHENG YANG, University of Toronto, Canada and Vector Institute, Canada
SANKEERTH DURVASULA, University of Toronto, Canada and Vector Institute, Canada
YU XIN LI, University of Toronto, Canada
MOHAMMAD SADROSADATI, ETH Zürich, Switzerland
JUAN GOMEZ LUNA, NVIDIA, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland
GENNADY PEKHIMENKO, University of Toronto, Canada, Vector Institute, Canada, and CentML,

https://arxiv.org/pdf/2402.16731

Accelerating GNNs on Real PIM Systems

- **https://arxiv.org/pdf/2402.16731**

Graph Neural Networks (GNNs) are emerging models to analyze graph-structure data. The GNN execution involves both compute-intensive and memory-intensive kernels. The memory-intensive kernels dominate execution time, because they are significantly bottlenecked by data movement between memory and processors. Processing-In-Memory (PIM) systems can alleviate this data movement bottleneck by placing simple processors near or inside memory arrays. To this end, we investigate the potential of PIM systems to alleviate the data movement bottleneck in GNNs, and introduce PyGim, an efficient and easy-to-use GNN library for real PIM systems. We propose intelligent parallelization techniques for memory-intensive kernels of GNNs tailored for real PIM systems, and develop an easy-to-use Python API for them. PyGim employs a cooperative GNN execution, in which the compute- and memory-intensive kernels are executed in processor-centric and memory-centric computing systems, respectively, to fully exploit the hardware capabilities. PyGim integrates a lightweight tuner that configures the parallelization strategy of the memory-intensive kernel of GNNs to provide high system performance, while also enabling high programming ease. We extensively evaluate PyGim on a real-world PIM system that has 16 PIM DIMMs with 1992 PIM cores connected to a Host CPU. In GNN inference, we demonstrate that it outperforms prior state-of-the-art PIM works by on average 4.38× (up to 7.20×), and the state-of-the-art PyTorch implementation running on Host (on Intel Xeon CPU) by on average 3.04× (up to 3.44×). PyGim improves energy efficiency by 2.86× (up to 3.68×) and 1.55× (up to 1.75×) over prior PIM and PyTorch Host schemes, respectively. In memory-intensive kernel of GNNs, PyGim provides 11.6× higher resource utilization in PIM system than that of PyTorch library (optimized CUDA implementation) in GPU systems. Our work provides useful recommendations for software, system and hardware designers. PyGim is publicly and freely available at https://github.com/CMU-SAFARI/PyGim to facilitate the widespread use of PIM systems in GNNs.

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%

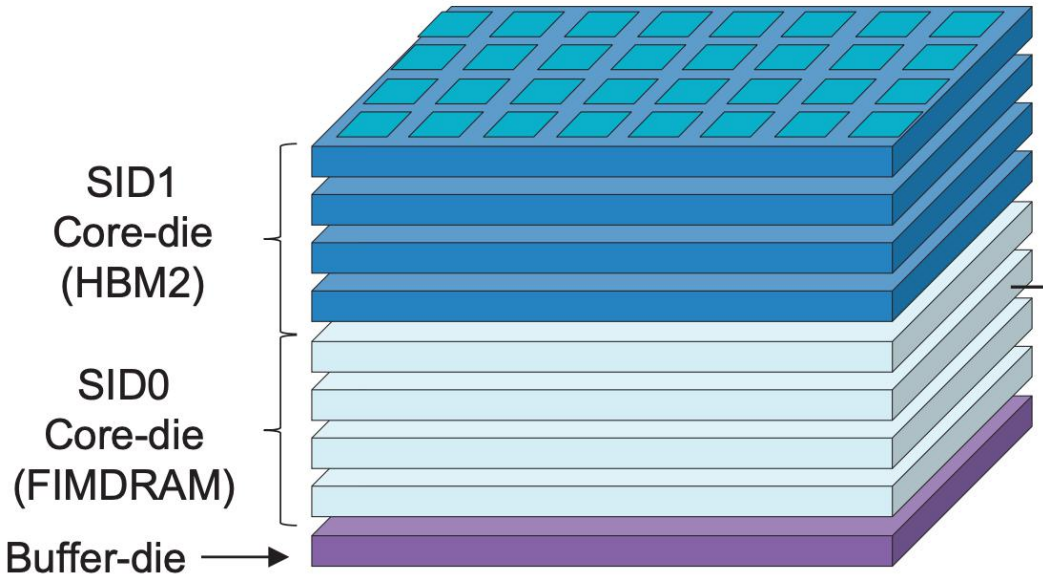
Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power – the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing 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."

https://news.samsung.com/global/samsung-develops-industrys-first-high-bandwidth-memory-with-ai-processing-power

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

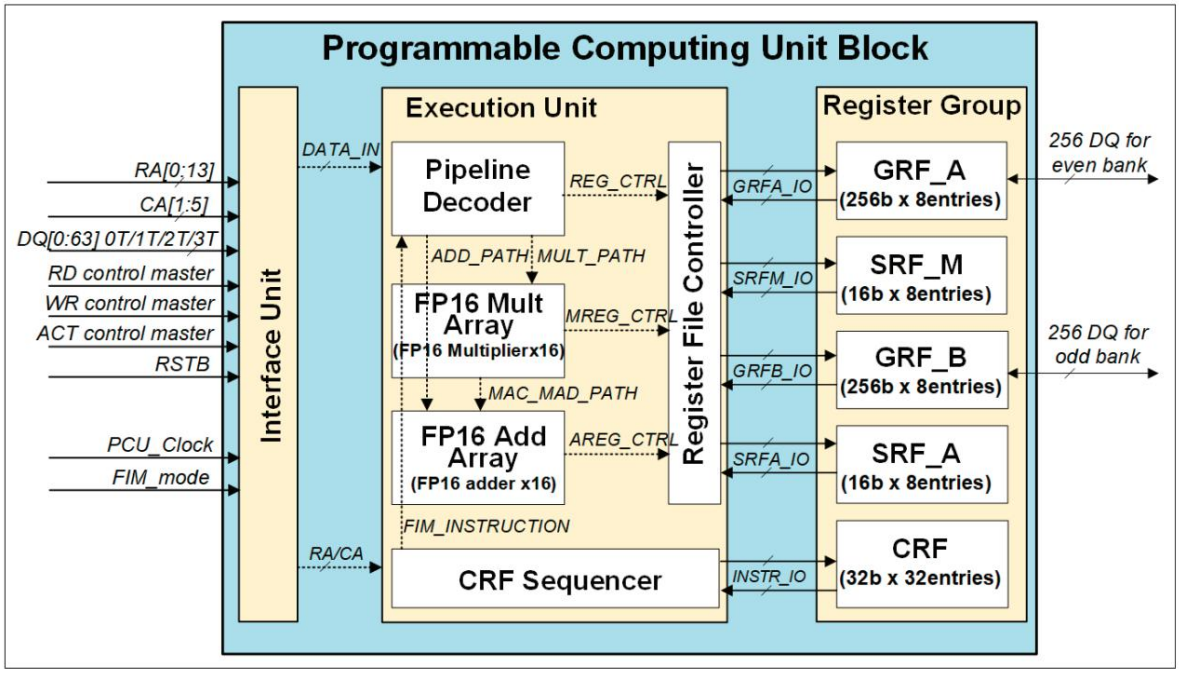
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1],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

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

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]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

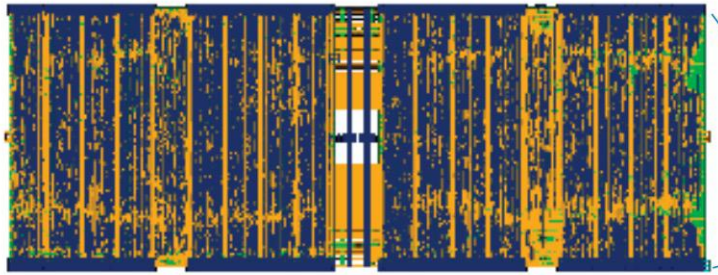
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1],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

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

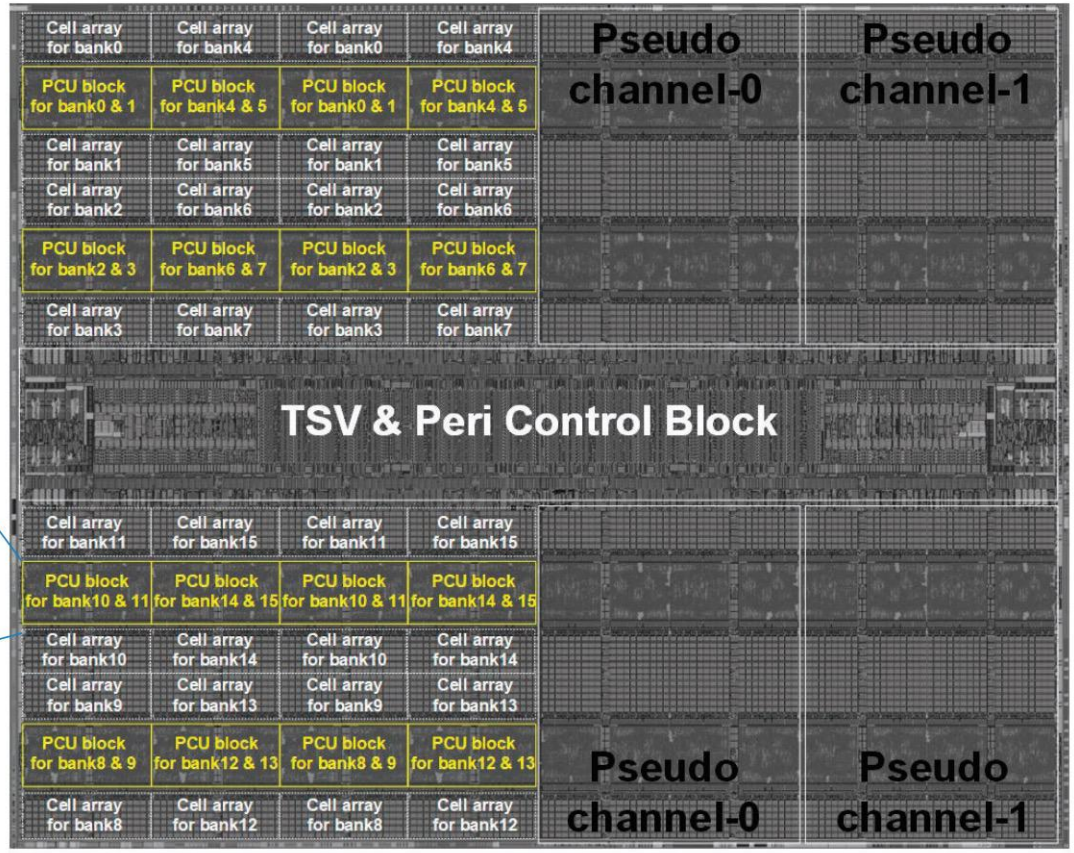
Samsung Function-in-Memory DRAM (2021)

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

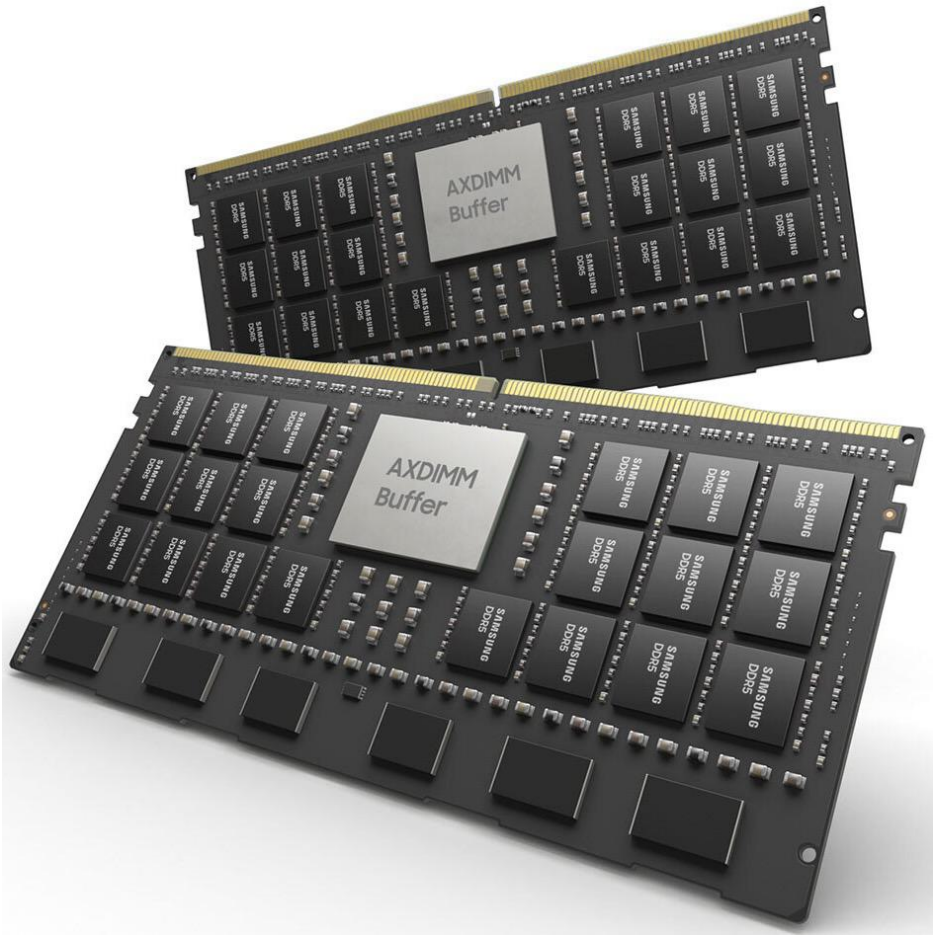
25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

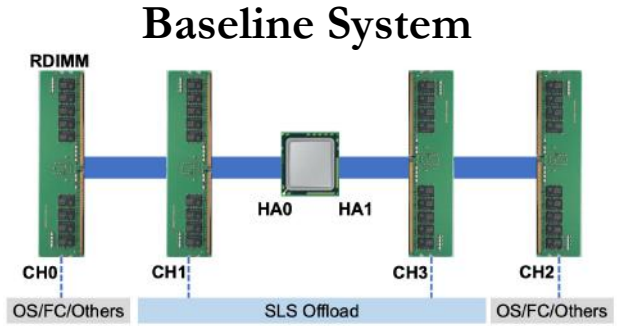
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1],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

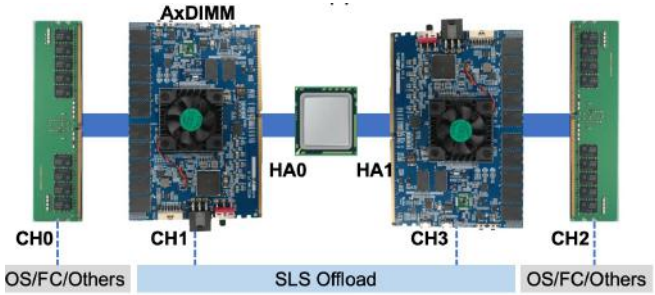
[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

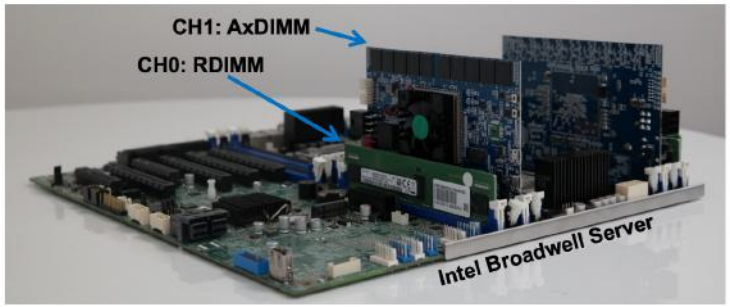
Samsung AxDIMM (2021)

- DDRx-PIM
  - DLRM recommendation system









Ke et al. "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM", IEEE Micro (2021)

SK Hynix Accelerator-in-Memory (2022)

SKhynix NEWSROOM ENG

INSIGHT SK hynix STORY PRESS CENTER MULTIMEDIA Search

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022

Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

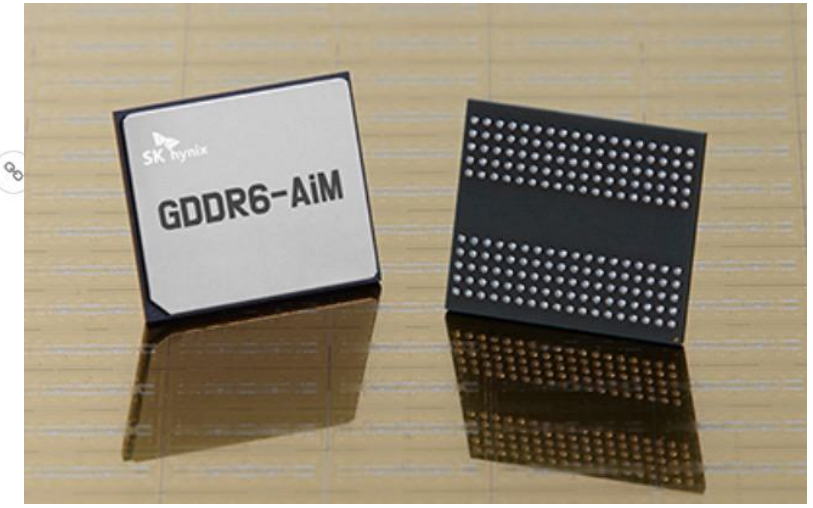
*PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world's most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

*ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of "Intelligent Silicon for a Sustainable World"

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.



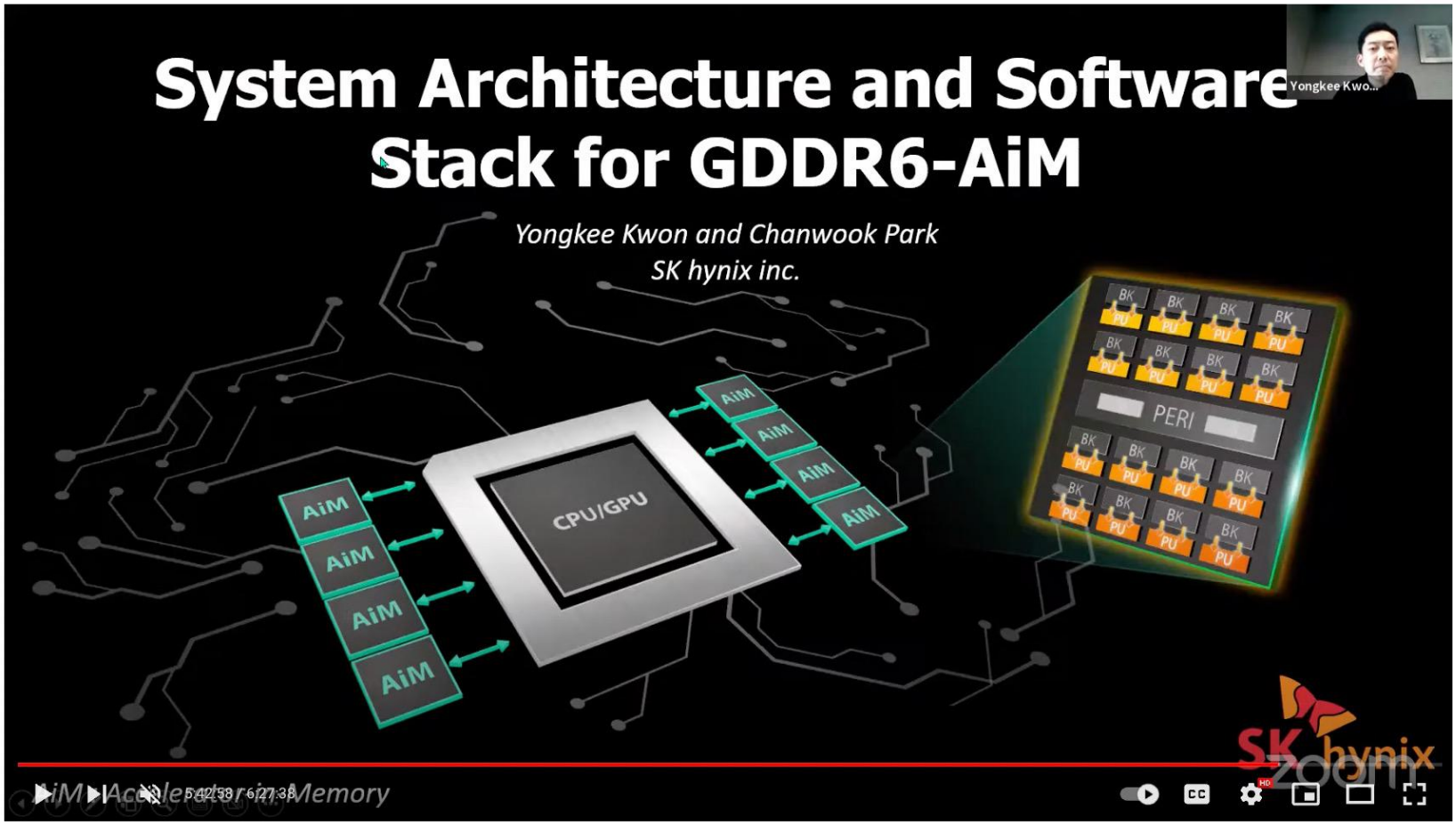
11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

Seongju Lee, SK hynix, Icheon, Korea

In Paper 11.1, SK Hynix describes an 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

https://news.skhynix.com/sk-hynix-develops-pim-next-generation-ai-accelerator/

SK Hynix Accelerator-in-Memory (2022)



ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

Onur Mutlu Lectures
32.1K subscribers

Analytics | Edit video | 33 | Share | Download | Clip | Save | ...

1,146 views Streamed live on Mar 26, 2023 Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads
https://events.safari.ethz.ch/asplos-...

https://www.youtube.com/watch?v=oYCaLcT0Kmo

AliBaba PIM Recommendation System (2022)

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

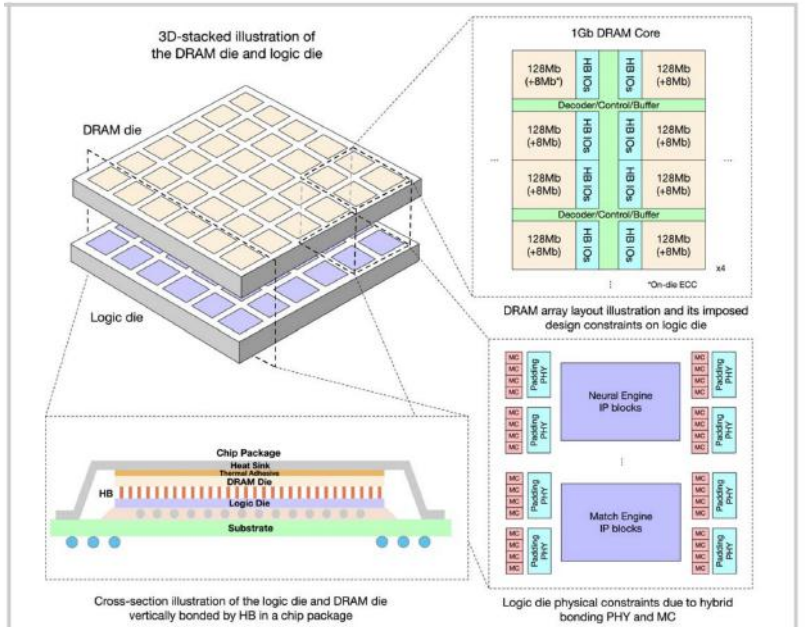
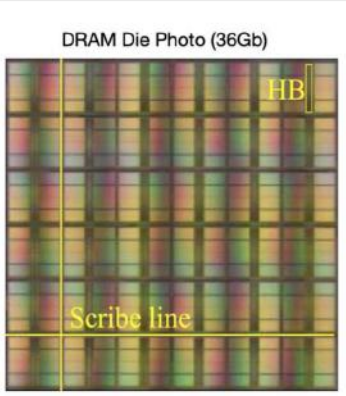
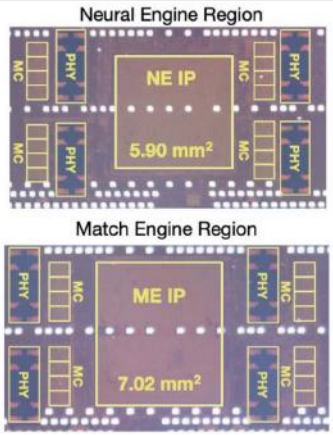


Figure 29.1.2: Illustration of 3D-stacked chip, cross-illustration of package, DRAM array layout and design blocks on logic die.





DRAM Die		
Technology	25nm	
Area	Total*	602.22 mm²
	Neural Engine	32 mm²
	Match Engine	32 mm²
Voltage	1.1 V	
Frequency (max)	150 MHz	
Power	300 mW per 1Gb	
Bandwidth**	153.60 GB/s / 1.38 TB/s	

* Total die size of test chip
** Memory bandwidth of NE and ME / Total bandwidth of DRAM die

Logic Die		
Technology	55nm	
Area	Total*	602.22 mm²
	Neural Engine	5.90 mm²
	Match Engine	7.02 mm²
# of MC	16 per IP	
Voltage	1.2 V	
Frequency	300 MHz	
Power	977.70 mW	
Precision	INT8	

Figure 29.1.7: Die micrographs of DRAM die, NE and ME. Detailed specifications of DRAM die and logic die.

29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

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

SK Hynix CXL Processing Near Memory (2023)

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim, Soohong Ahn, Taeyoung Ahn, Seungyong Lee, Myunghyun Rhee, Jooyoung Kim, Kwangsik Shin, Donguk Moon, Euiseok Kim, and Kyoung Park

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to 1.9× better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

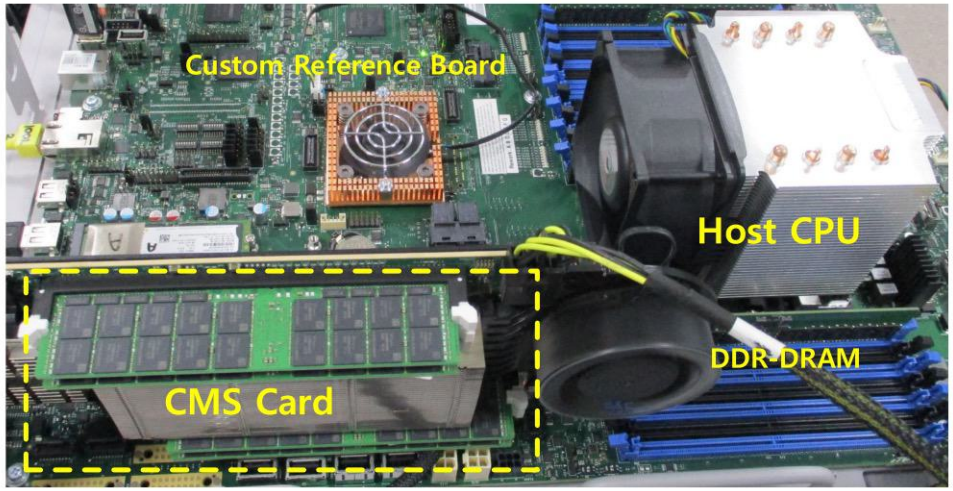


Fig. 6. FPGA prototype of proposed CMS card.

Samsung CXL Processing Near Memory (2023)

Samsung Processing in Memory Technology at Hot Chips 2023

By **Patrick Kennedy** - August 28, 2023

Samsung PIM PNM For Transformer Based AI HC35_Page_24

https://www.servethehome.com/samsung-processing-in-memory-technology-at-hot-chips-2023/

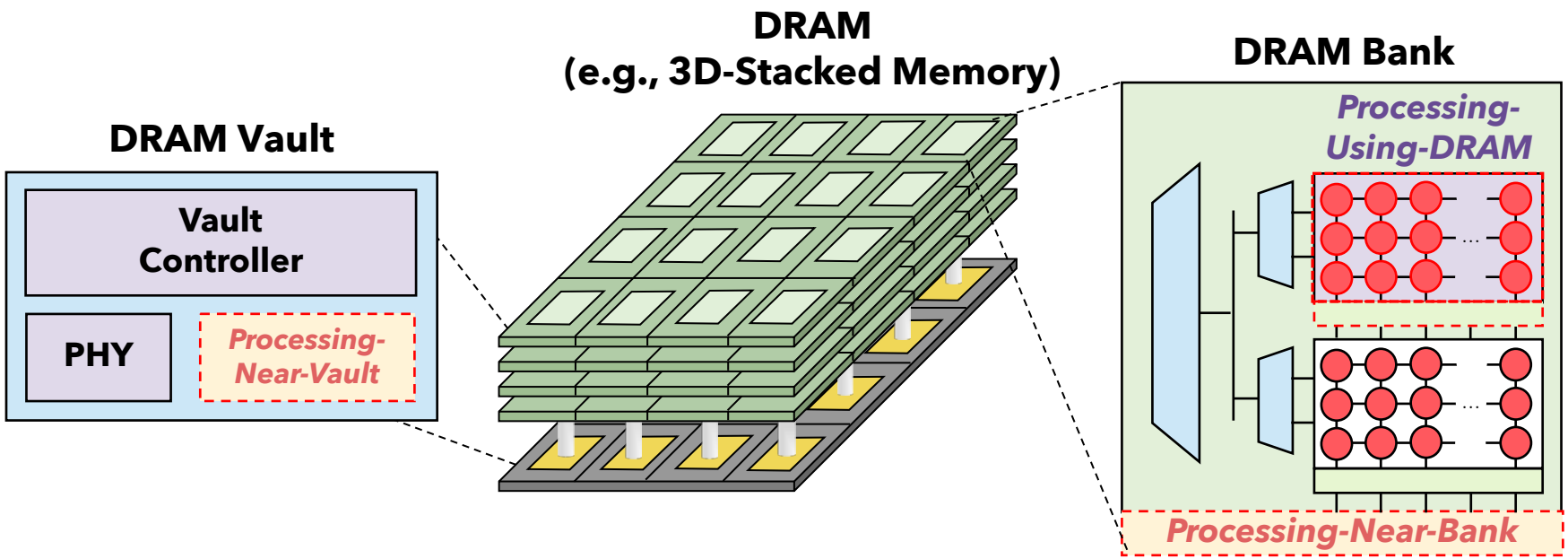
Processing in Memory: Two Types

1. Processing **near** Memory
2. Processing **using** Memory

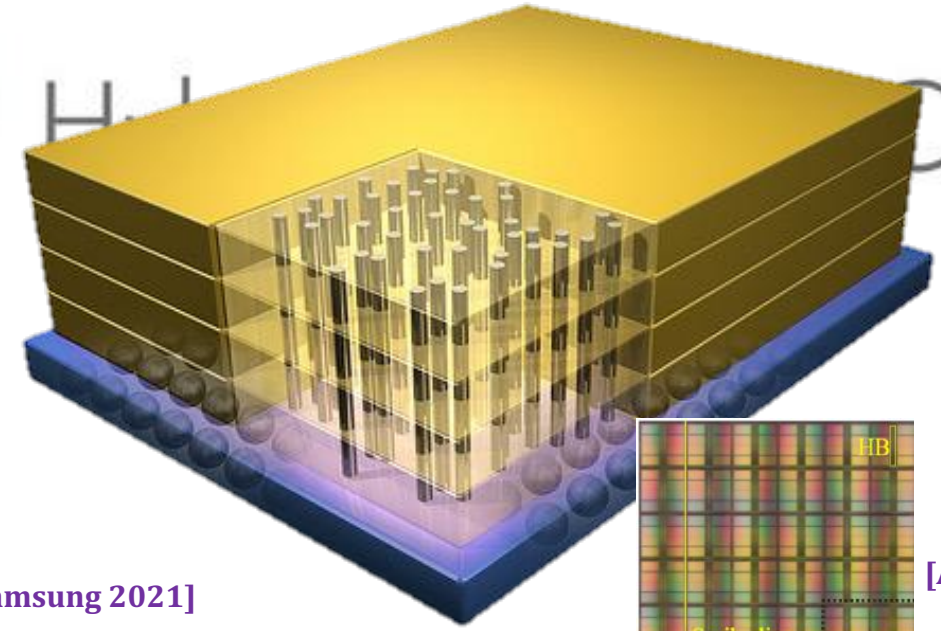
Processing-in-Memory: Two Types

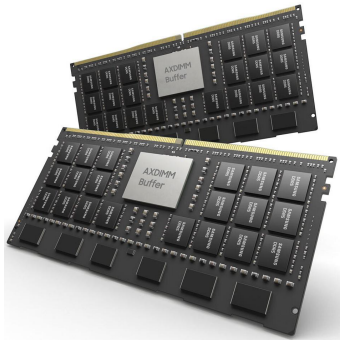
1 **Processing-Near-Memory**: Computation logic is added to the same die as memory or to the logic layer of 3D-stacked memory

2 **Processing-Using-Memory**: uses the operational principles of memory cells & circuitry to perform computation



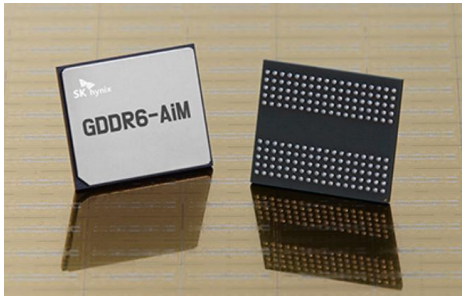
Processing-in-Memory Landscape Today



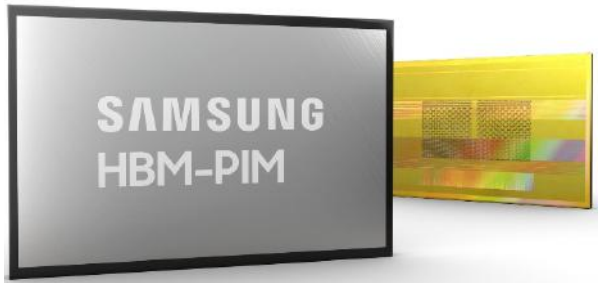


[Samsung 2021]

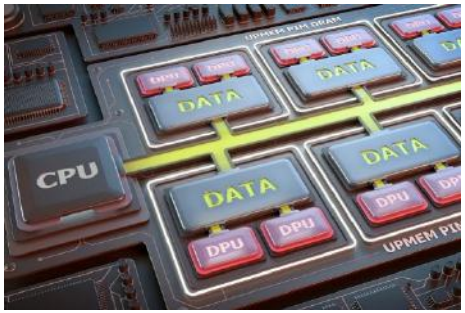
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

And, many other experimental chips and startups

Processing-in-Memory Landscape Today

IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim, Soohong Ahn, Taeyoung Ahn, Seungyong Lee, Myunghyun Rhee, Jooyoung Kim, Kwangsik Shin, Donguk Moon, Euiseok Kim, and Kyoung Park

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to 1.9× better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

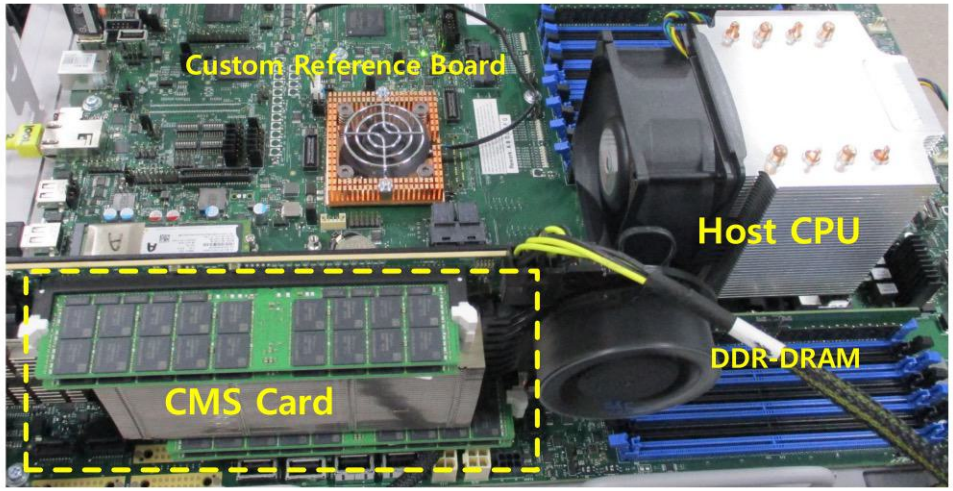


Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today

Samsung Processing in Memory Technology at Hot Chips 2023

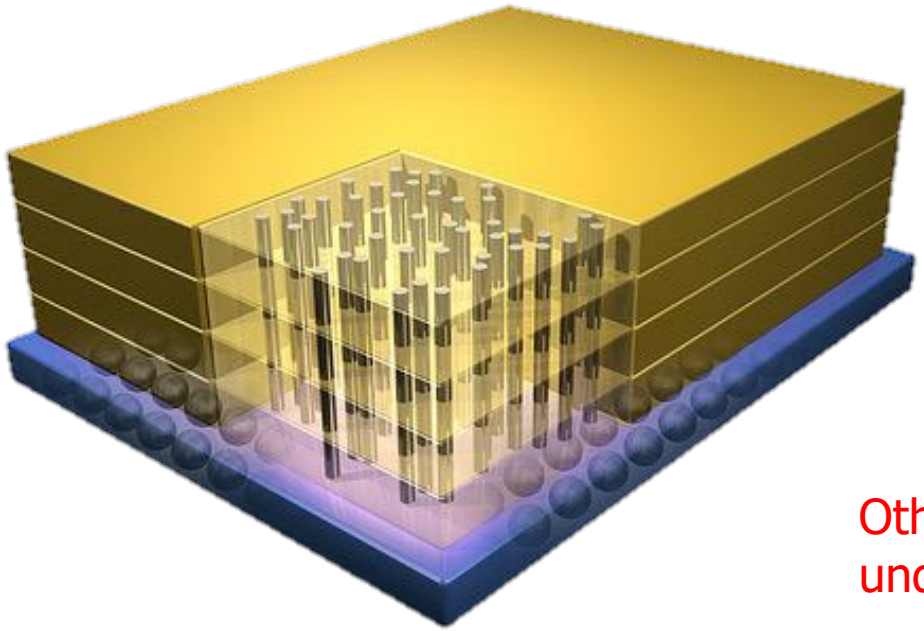
By **Patrick Kennedy** - August 28, 2023

Samsung PIM PNM For Transformer Based AI HC35_Page_24

https://www.servethehome.com/samsung-processing-in-memory-technology-at-hot-chips-2023/

Opportunity: 3D-Stacked Logic+Memory

Hybrid Memory Cube
CONSORTIUM



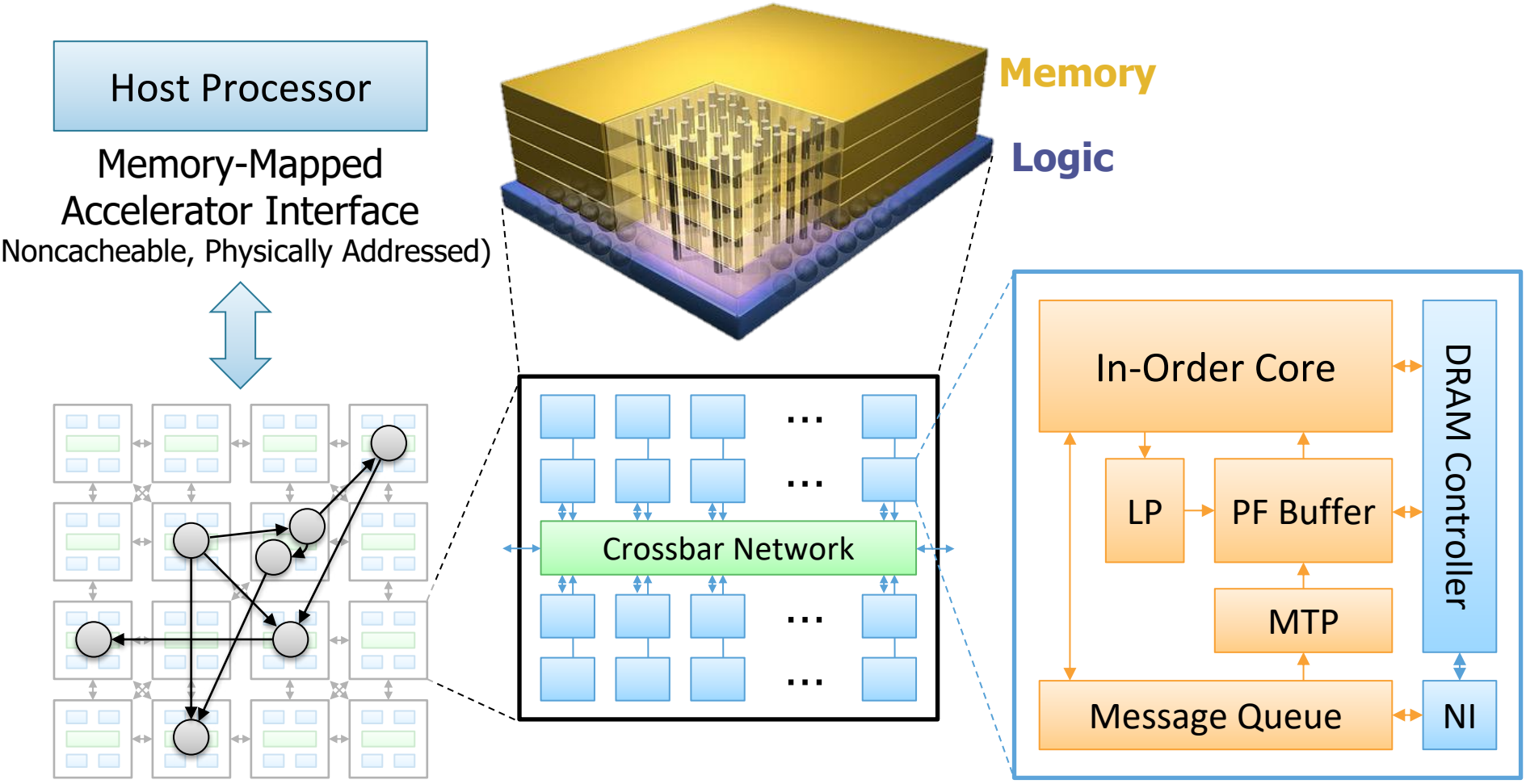
Memory

Logic

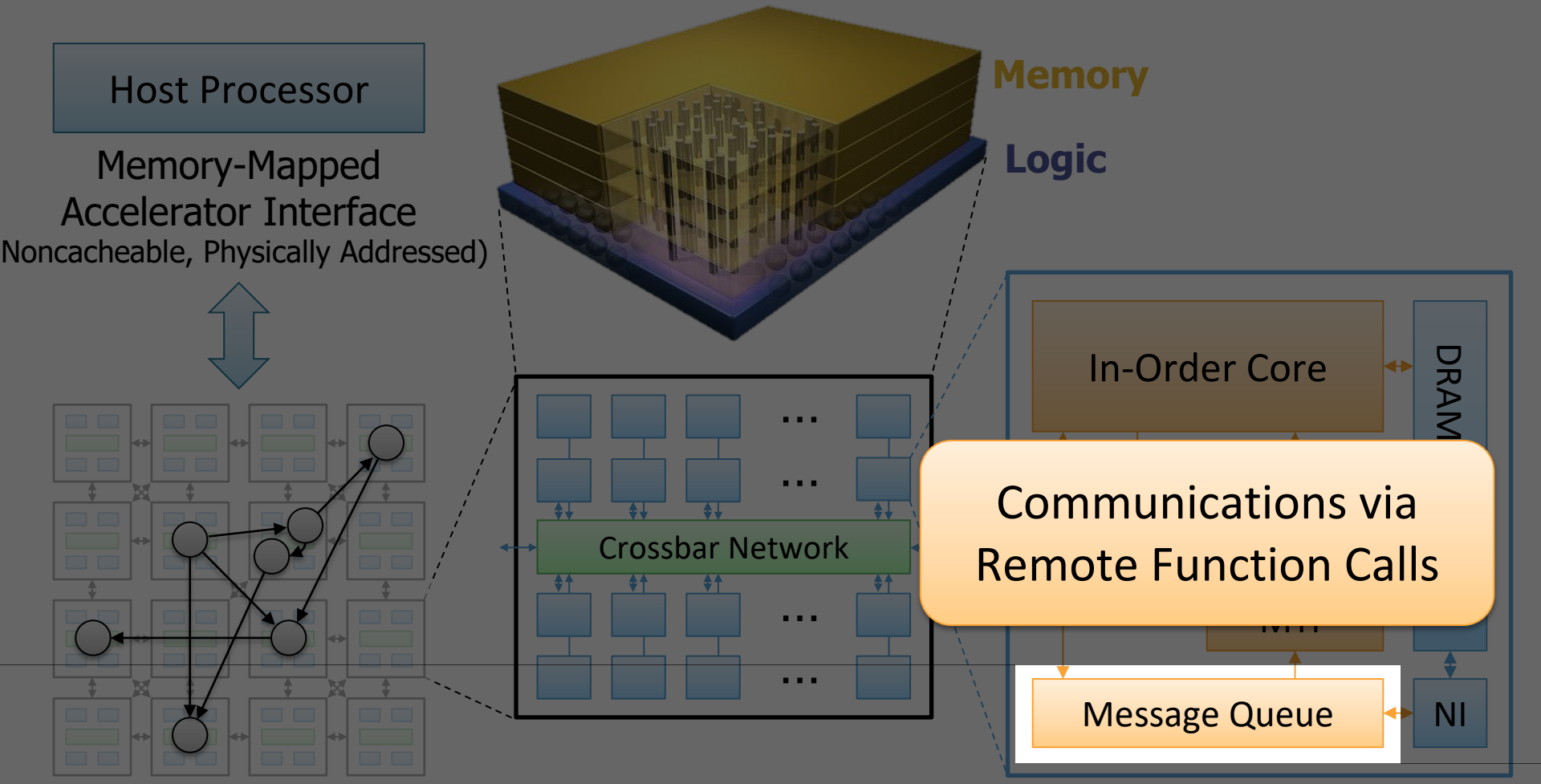
Other "True 3D" technologies under development

Tesseract System for Graph Processing

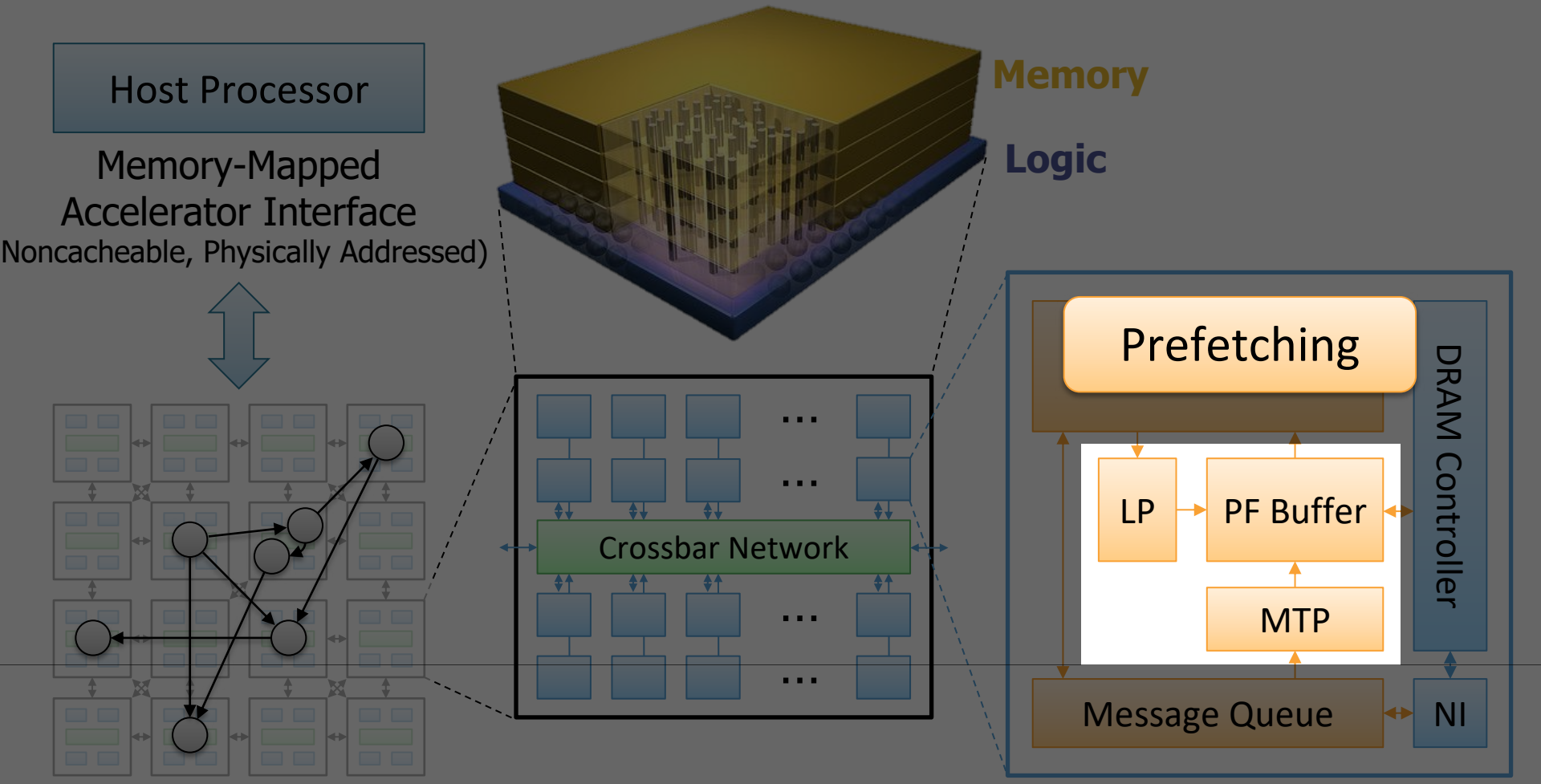
Interconnected set of 3D-stacked memory+logic chips with simple cores



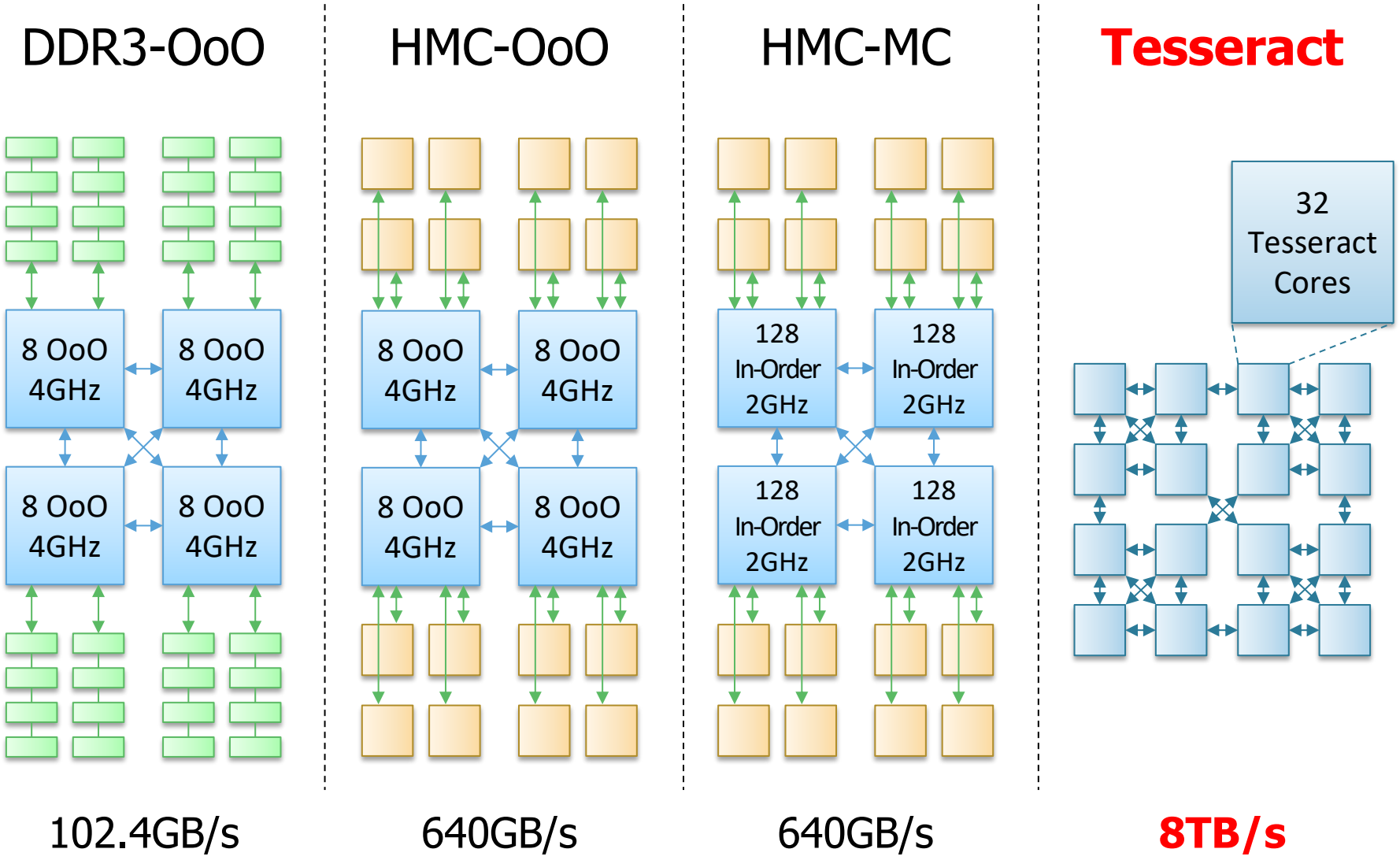
Tesseract System for Graph Processing



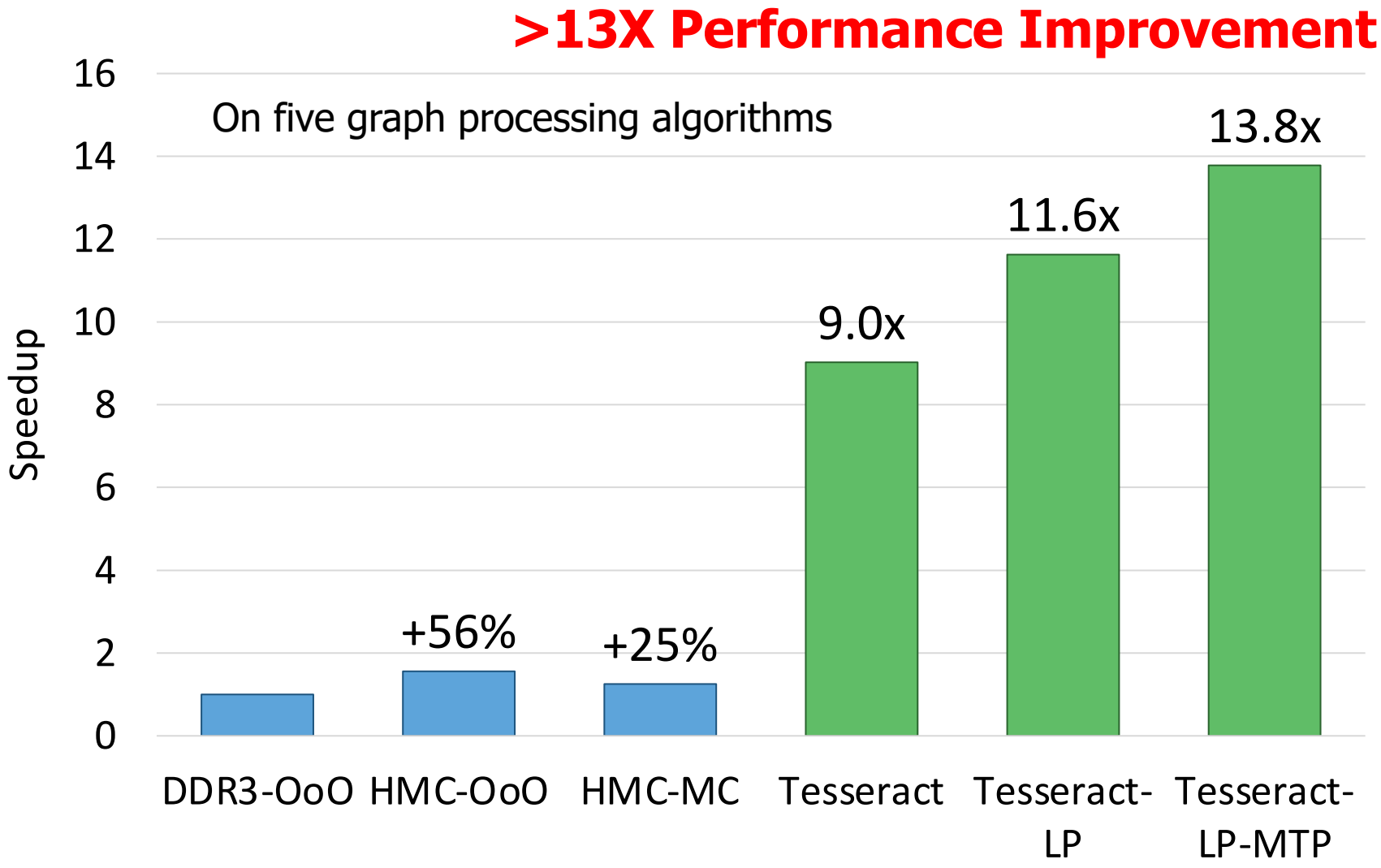
Tesseract System for Graph Processing



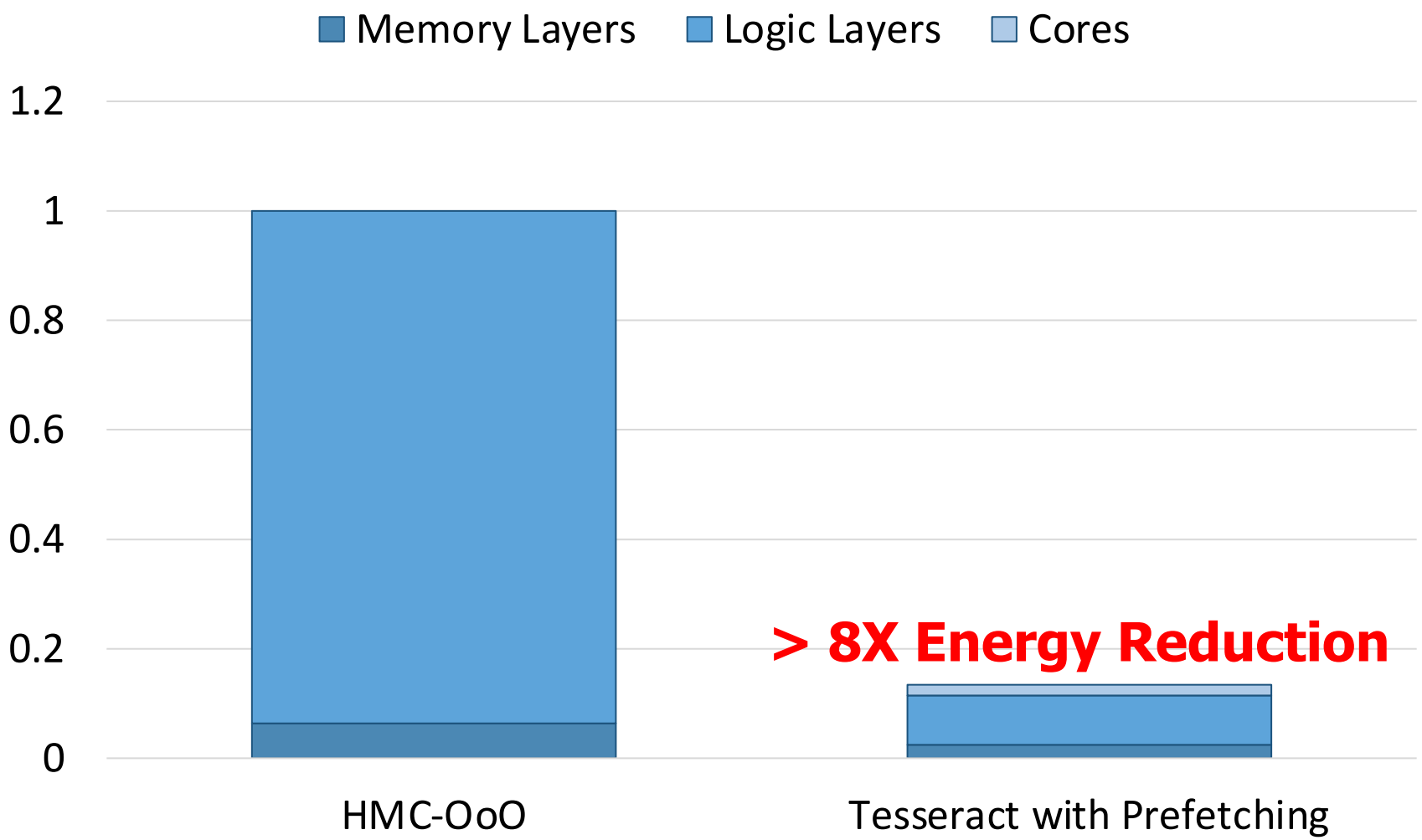
Evaluated Systems



Tesseract Graph Processing Performance



Tesseract Graph Processing System Energy



More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung 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 (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  Top Picks Honorable Mention by IEEE Micro.
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University [§]Oracle Labs [†]Carnegie Mellon University

A Short Retrospective @ 50 Years of ISCA

Retrospective: A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

Junwhan Ahn[†] Sungpack Hong[‡] Sungjoo Yoo[▽] Onur Mutlu[§] Kiyoung Choi[▽]
[†]*Google DeepMind* [‡]*Oracle Labs* [§]*ETH Zürich* [▽]*Seoul National University*

***Abstract*—Our ISCA 2015 paper [1] provides a new programmable processing-in-memory (PIM) architecture and system design that can accelerate key data-intensive applications, with a focus on graph processing workloads. Our major idea was to completely rethink the system, including the programming model, data partitioning mechanisms, system support, instruction set architecture, along with near-memory execution units and their communication architecture, such that an important workload can be accelerated at a maximum level using a distributed system of well-connected near-memory accelerators. We built our accelerator system, Tesseract, using 3D-stacked memories with logic layers, where each logic layer contains general-purpose processing cores and cores communicate with each other using a message-passing programming model. Cores could be specialized for graph processing (or any other application to be accelerated).**

To our knowledge, our paper was the first to completely design a near-memory accelerator system from scratch such that it is both generally programmable and specifically customizable to accelerate important applications, with a case study on major graph processing workloads. Ensuing work in academia and industry showed that similar approaches to system design can greatly benefit both graph processing workloads and other applications, such as machine learning, for which ideas from Tesseract seem to have been influential.

This short retrospective provides a brief analysis of our ISCA 2015 paper and its impact. We briefly describe the major ideas and contributions of the work, discuss later works that built on it or were influenced by it, and make some educated guesses on what the future may bring on PIM and accelerator systems.

I. Background, Approach & Mindset

We started our research when 3D-stacked memories (e.g., [2–4]) were viable and seemed to have promise for building effective and practical processing-near-memory systems. Such near-memory systems could lead to improvements, but there was little to no research that examined how an accelerator could be completely (re-)designed using such near-memory technology, from its hardware architecture to its programming model and software system, and what the performance and energy benefits could be of such a re-design. We set out to answer these questions in our ISCA 2015 paper [1].

We followed several major principles to design our accelerator from the ground up. We believe these principles are still important: a major contribution and influence of our work was in putting all of these together in a cohesive full-system design and demonstrating the large performance and energy benefits that can be obtained from such a design. We see a similar approach in many modern large-scale accelerator systems in machine learning today (e.g., [5–9]). Our principles are:

1. *Near-memory execution* to enable/exploit the high data access bandwidth modern workloads (e.g., graph processing) need and to reduce data movement and access latency.
2. *General programmability* so that the system can be easily adopted, extended, and customized for many workloads.
3. *Maximal acceleration capability* to maximize the performance and energy benefits. We set ourselves free from backward compatibility and cost constraints. We aimed to completely re-design the system stack. Our goal was to explore the maximal performance and energy efficiency benefits we can gain from a near-memory accelerator if we had complete freedom to change things as much as we needed. We contrast this approach to the *minimal intrusion* approach we also explored in a separate ISCA 2015 paper [10].
4. *Customizable to specific workloads*, such that we can maximize acceleration benefits. Our focus workload was graph analytics/processing, a key workload at the time and today. However, our design principles are not limited to graph processing and the system we built is customizable to other workloads as well, e.g., machine learning, genome analysis.
5. *Memory-capacity-proportional performance*, i.e., processing capability should proportionally grow (i.e., scale) as memory capacity increases and vice versa. This enables scaling of data-intensive workloads that need both memory and compute.
6. *Exploit new technology (3D stacking)* that enables tight integration of memory and logic and helps multiple above principles (e.g., enables customizable near-memory acceleration capability in the logic layer of a 3D-stacked memory chip).
7. *Good communication and scaling capability* to support scalability to large dataset sizes and to enable memory-capacity-proportional performance. To this end, we provided scalable communication mechanisms between execution cores and carefully interconnected small accelerator chips to form a large distributed system of accelerator chips.
8. *Maximal and efficient use of memory bandwidth* to supply the high-bandwidth data access that modern workloads need. To this end, we introduced new, specialized mechanisms for prefetching and a programming model that helps leverage application semantics for hardware optimization.

II. Contributions and Influence

We believe the major contributions of our work were 1) complete rethinking of how an accelerator system should be designed to enable maximal acceleration capability, and 2) the design and analysis of such an accelerator with this mindset and using the aforementioned principles to demonstrate its effectiveness in an important class of workloads.

One can find examples of our approach in modern large-scale machine learning (ML) accelerators, which are perhaps the most successful incarnation of scalable near-memory execution architectures. ML infrastructure today (e.g., [5–9]) consists of accelerator chips, each containing compute units and high-bandwidth memory tightly packaged together, and features scale-up capability enabled by connecting thousands of such chips with high-bandwidth interconnection links. The system-wide rethinking that was done to enable such accelerators and many of the principles used in such accelerators resemble our ISCA 2015 paper's approach.

The "memory-capacity-proportional performance" principle we explored in the paper shares similarities with how ML workloads are scaled up today. Similar to how we carefully sharded graphs across our accelerator chips to greatly improve effective memory bandwidth in our paper, today's ML workloads are sharded across a large number of accelerators by leveraging data/model parallelism and optimizing the placement to balance communication overheads and compute scalability [11, 12]. With the advent of large generative models requiring high memory bandwidth for fast training and inference, the scaling behavior where capacity and bandwidth are scaled together has become an essential architectural property to support modern data-intensive workloads.

The "maximal acceleration capability" principle we used in Tesseract provides much larger performance and energy improvements and better customization than the "minimalist" approach that our other ISCA 2015 paper on *PIM-Enabled Instructions* [10] explored: "minimally change" an existing system to incorporate (near-memory) acceleration capability to ease programming and keep costs low. So far, the industry has more widely adopted the maximal approach to overcome the pressing scaling bottlenecks of major workloads. The key enabler that bridges the programmability gap between the maximal approach favoring large performance & energy benefits and the minimal approach favoring ease of programming is compilation techniques. These techniques lower well-defined high-level constructs into lower-level primitives [12, 13]; our ISCA 2015 papers [1, 10] and a follow-up work [14] explore them lightly. We believe that a good programming model that enables large benefits coupled with support for it across the entire system stack (including compilers & hardware) will continue to be important for effective near-memory system and accelerator designs [14]. We also believe that the maximal versus minimal approaches that are initially explored in our two ISCA 2015 papers is a useful way of exploring emerging technologies (e.g., near-memory accelerators) to better understand the tradeoffs of system designs that exploit such technologies.

III. Influence on Later Works

Our paper was at the beginning of a proliferation of scalable near-memory processing systems designed to accelerate key applications (see [15] for many works on the topic). Tesseract has inspired many near-memory system ideas (e.g., [16–28]) and served as the de facto comparison point for such systems, including near-memory graph processing accelerators that built on Tesseract and improved various aspects of Tesseract. Since machine learning accelerators that use high-bandwidth memory (e.g., [5, 29]) and industrial PIM prototypes (e.g., [30–41]) are now in the market, near-memory processing is no longer an "eccentric" architecture it used to be when Tesseract was originally published.

Graph processing & analytics workloads remain as an important and growing class of applications in various forms, ranging from large-scale industrial graph analysis engines (e.g., [42]) to graph neural networks [43]. Our focus on large-scale graph processing in our ISCA 2015 paper increased attention to this domain in the computer architecture community, resulting in subsequent research on efficient hardware architectures for graph processing (e.g., [44–46]).

IV. Summary and Future Outlook

We believe that our ISCA 2015 paper's principled rethinking of system design to accelerate an important class of data-intensive workloads provided significant value and enabled/influenced a large body of follow-on works and ideas. We expect that such rethinking of system design for key workloads, especially with a focus on "maximal acceleration capability," will continue to be critical as pressing technology and application scaling challenges increasingly require us to think differently to substantially improve performance and energy (as well as other metrics). We believe the principles exploited in Tesseract are fundamental and they will remain useful and likely become even more important as systems become more constrained due to the continuously-increasing memory access and computation demands of future workloads. We also project that as hardware substrates for near-memory acceleration (e.g., 3D stacking, in-DRAM computation, NVM-based PIM, processing using memory [15]) evolve and mature, systems will take advantage of them even more, likely using principles similar to those used in the design of Tesseract.

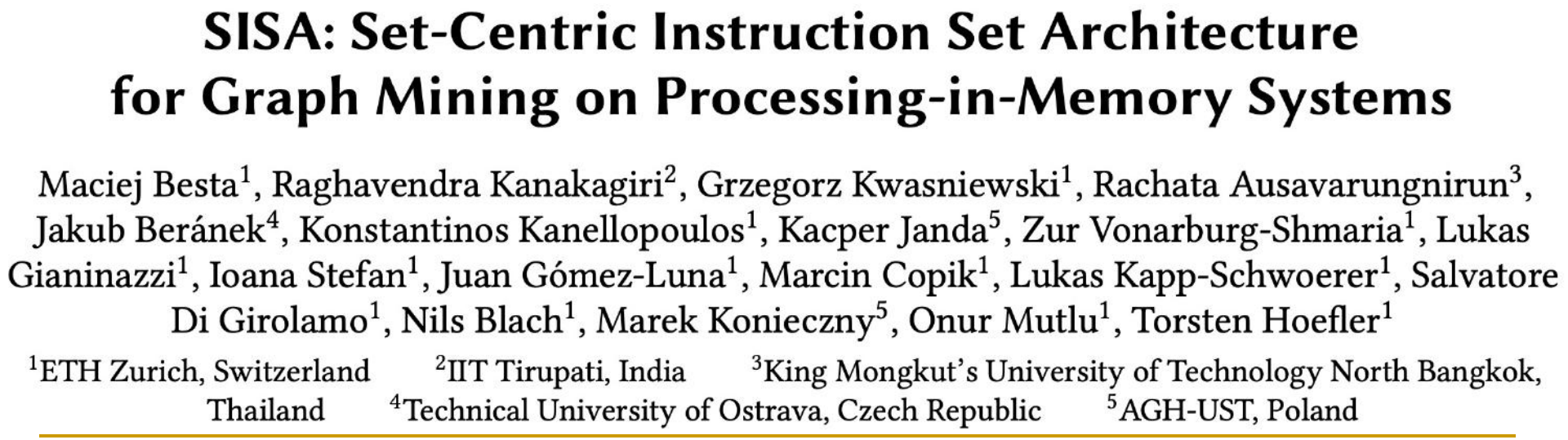
References

[1] J. Ahn *et al.*, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," in *ISCA*, 2015.
[2] Hybrid Memory Cube Consortium, "HMC Specification 1.1," 2013.
[3] J. Jeddeloh and B. Keeth, "Hybrid Memory Cube: New DRAM Architecture Increases Density and Performance," in *VLSIT*, 2012.
[4] JEDEC, "High Bandwidth Memory (HBM) DRAM," Standard No. JESD235, 2013.
[5] N. Jouppi *et al.*, "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embedding," in *ISCA*, 2023.
[6] J. Fowers *et al.*, "A Configurable Cloud-Scale DNN Processor for Real-Time AI," in *ISCA*, 2018.
[7] S. Lie, "Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning," in *IEEE Micro*, 2023.
[8] E. Talpes *et al.*, "The Microarchitecture of DOJO, Tesla's Exa-Scale Computer," in *IEEE Micro*, 2023.
[9] A. Ishii and R. Wells, "NVLink-Network Switch - NVIDIA's Switch Chip for High Communication-Bandwidth SuperPODs," in *Hot Chips*, 2022.
[10] J. Ahn *et al.*, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture," in *ISCA*, 2015.
[11] R. Pope *et al.*, "Efficiently Scaling Transformer Inference," in *MLSys*, 2023.
[12] D. Lepikhin *et al.*, "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding," in *ICLR*, 2021.
[13] S. Wang *et al.*, "Overlap Communication with Dependent Computation via Decomposition in Large Deep Learning Models," in *ASPLOS*, 2023.
[14] J. Ahn *et al.*, "AIM: Energy-Efficient Aggregation Inside the Memory Hierarchy," *ACM TACO*, vol. 13, no. 4, 2016.
[15] O. Mutlu *et al.*, "A Modern Primer on Processing in Memory," *Emerging Computing: From Devices to Systems*, 2021, https://arxiv.org/abs/2012.03112.
[16] M. Zhang *et al.*, "GraphP: Reducing Communication for PIM-Based Graph Processing with Efficient Data Partition," in *HPCA*, 2018.
[17] L. Song, "GraphR: Accelerating Graph Processing Using ReRAM," in *HPCA*, 2018.
[18] Y. Zhuo *et al.*, "GraphQ: Scalable PIM-Based Graph Processing," in *MICRO*, 2019.
[19] G. Dai *et al.*, "GraphH: A Processing-in-Memory Architecture for Large-Scale Graph Processing," *IEEE TCAD*, 2018.
[20] G. Li *et al.*, "GraphIA: An In-situ Accelerator for Large-scale Graph Processing ," in *MEMSYS*, 2018.
[21] S. Rheindt *et al.*, "NEMESYS: Near-Memory Graph Copy Enhanced System-Software," in *MEMSYS*, 2019.
[22] L. Belayneh and V. Bertacco, "GraphVine: Exploiting Multicast for Scalable Graph Analytics," in *DATE*, 2020.
[23] N. Challapalle *et al.*, "GaaS-X: Graph Analytics Accelerator Supporting Sparse Data Representation using Crossbar Architectures," in *ISCA*, 2020.
[24] M. Zhou *et al.*, "Ultra Efficient Acceleration for De Novo Genome Assembly via Near-Memory Computing," in *PACT*, 2021.
[25] X. Xie *et al.*, "SpaceA: Sparse Matrix Vector Multiplication on Processing-in-Memory Accelerator," in *HPCA*, 2021.
[26] M. Zhou *et al.*, "HyGraph: Accelerating Graph Processing with Hybrid Memory-Centric Computing," in *DATE*, 2021.
[27] M. Lenjani *et al.*, "Gearbox: A Case for Supporting Accumulation Dispatching and Hybrid Partitioning in PIM-based Accelerators," in *ISCA*, 2022.
[28] M. Orenes-Vera *et al.*, "Dalorex: A Data-Local Program Execution and Architecture for Memory-Bound Applications," in *HPCA*, 2023.
[29] J. Choquette, "Nvidia Hopper GPU: Scaling Performance," in *Hot Chips*, 2022.
[30] F. Devaux, "The True Processing In Memory Accelerator," in *Hot Chips*, 2019.
[31] J. Gómez-Luna *et al.*, "Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System," *IEEE Access*, 2022.
[32] J. Gomez-Luna *et al.*, "Evaluating Machine Learning Workloads on Memory-Centric Computing Systems," in *ISPASS*, 2023.
[33] S. Lee *et al.*, "Hardware Architecture and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product," in *ISCA*, 2021.
[34] Y.-C. Kwon *et al.*, "25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2 TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications," in *ISSCC*, 2021.
[35] L. Ke *et al.*, "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM," *IEEE Micro*, 2021.
[36] D. Lee *et al.*, "Improving In-Memory Database Operations with Acceleration DIMM (AxDIMM)," in *DaMoN*, 2022.
[37] S. Lee *et al.*, "A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications," in *ISSCC*, 2022.
[38] D. Niu *et al.*, "184QPS/W 64Mb/mm^2 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System," in *ISSCC*, 2022.
[39] Y. Kwon, "System Architecture and Software Stack for GDDR6-AiM," in *HCS*, 2022.
[40] G. Singh *et al.*, "FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications," *IEEE Micro*, 2021.
[41] G. Singh *et al.*, "Accelerating Weather Prediction using Near-Memory Reconfigurable Fabric," *ACM TRETS*, 2021.
[42] S. Hong *et al.*, "PGX.D: A Fast Distributed Graph Processing Engine," in *SC*, 2015.
[43] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *ICLR*, 2017.
[44] L. Nai *et al.*, "GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks," in *HPCA*, 2017.
[45] M. Besta *et al.*, "SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems," in *MICRO*, 2021.
[46] T. J. Ham *et al.*, "Graphicionado: A High-Performance and Energy-Efficient Accelerator for Graph Analytics," in *MICRO*, 2016.

https://arxiv.org/pdf/2306.16093

Accelerating Graph Pattern Mining

- Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungnirun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,
  "SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems"
  Proceedings of the 54th International Symposium on Microarchitecture (***MICRO***), Virtual, October 2021.
  [Slides (pdf)]
  [Talk Video (22 minutes)]
  [Lightning Talk Video (1.5 minutes)]
  [Full arXiv version]



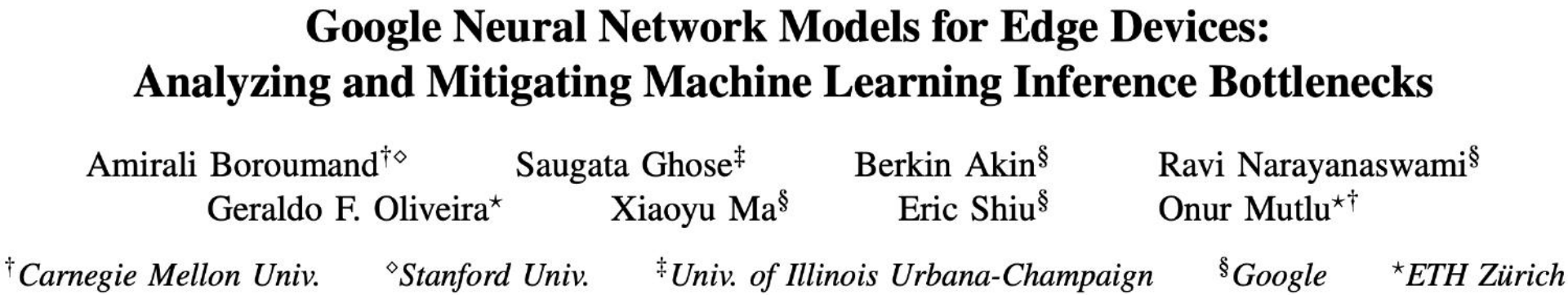
SISA: Set-Centric Instruction Set Architecture for Graph Mining on Processing-in-Memory Systems

Maciej Besta[1], Raghavendra Kanakagiri[2], Grzegorz Kwasniewski[1], Rachata Ausavarungnirun[3], Jakub Beránek[4], Konstantinos Kanellopoulos[1], Kacper Janda[5], Zur Vonarburg-Shmaria[1], Lukas Gianinazzi[1], Ioana Stefan[1], Juan Gómez-Luna[1], Marcin Copik[1], Lukas Kapp-Schwoerer[1], Salvatore Di Girolamo[1], Nils Blach[1], Marek Konieczny[5], Onur Mutlu[1], Torsten Hoefler[1]

[1]ETH Zurich, Switzerland [2]IIT Tirupati, India [3]King Mongkut's University of Technology North Bangkok, Thailand [4]Technical University of Ostrava, Czech Republic [5]AGH-UST, Poland

Accelerating Machine Learning Inference

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



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

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models

Speech Recognition

6 RNN Transducers

2 LSTMs

Language Translation

Face Detection

13 CNN

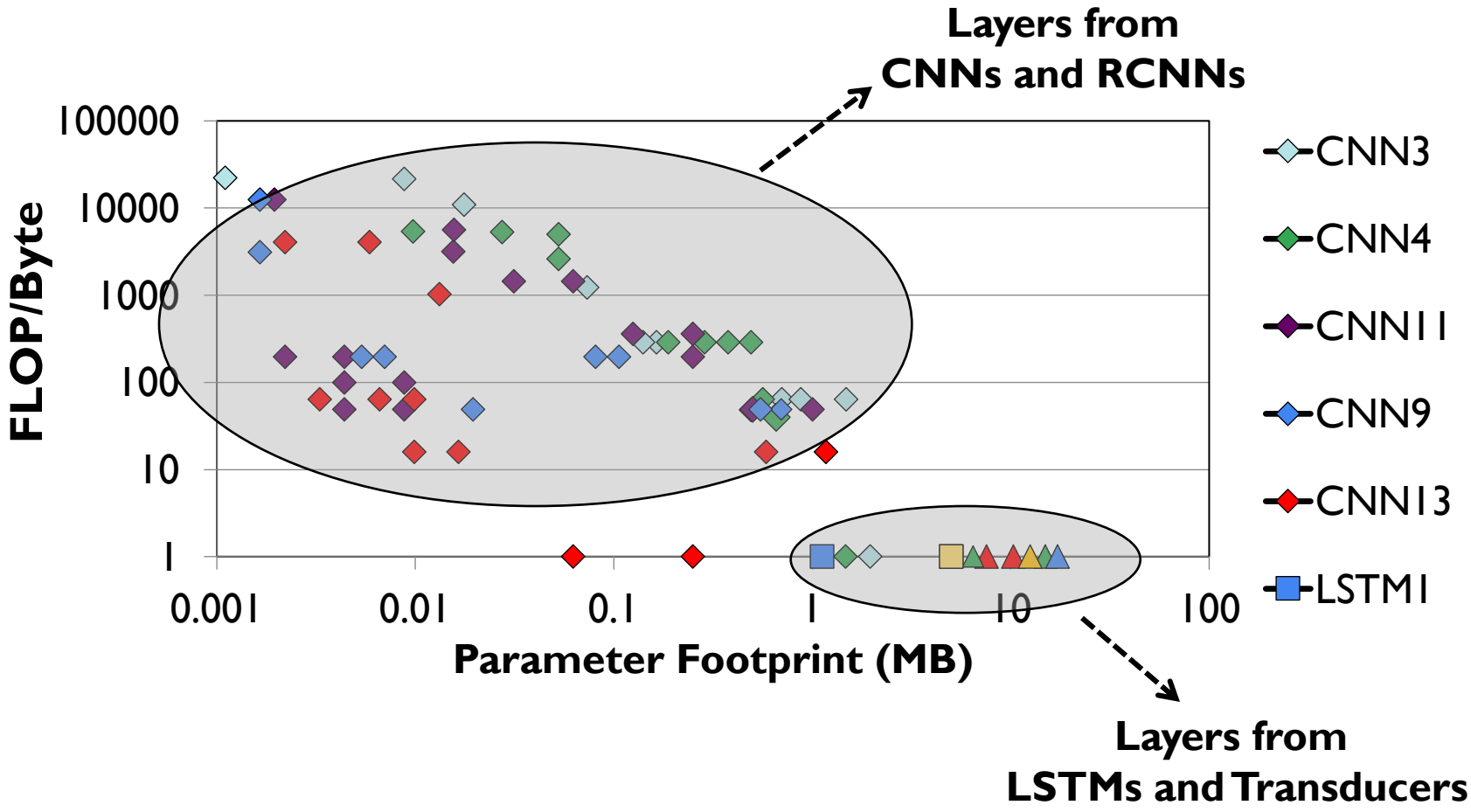
Google Edge TPU

3 RCNN

Image Captioning

Diversity Across the Models

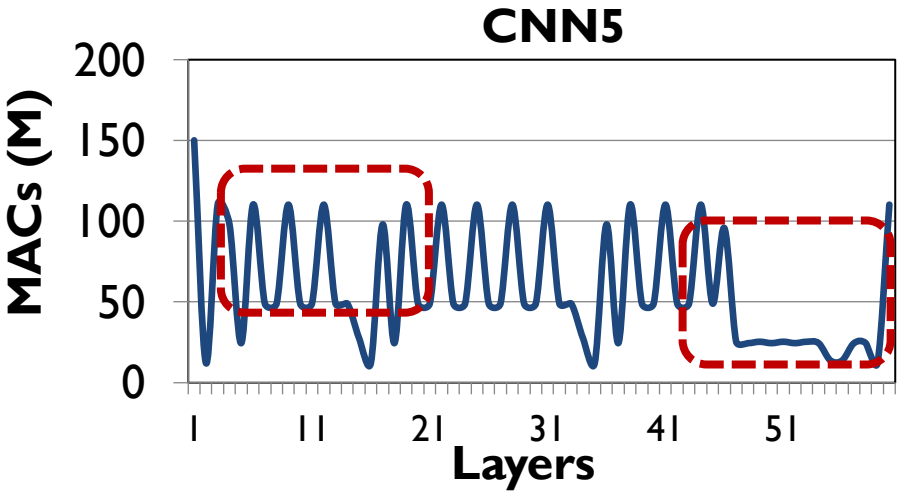
Insight 1: there is significant variation in terms of layer characteristics across the models

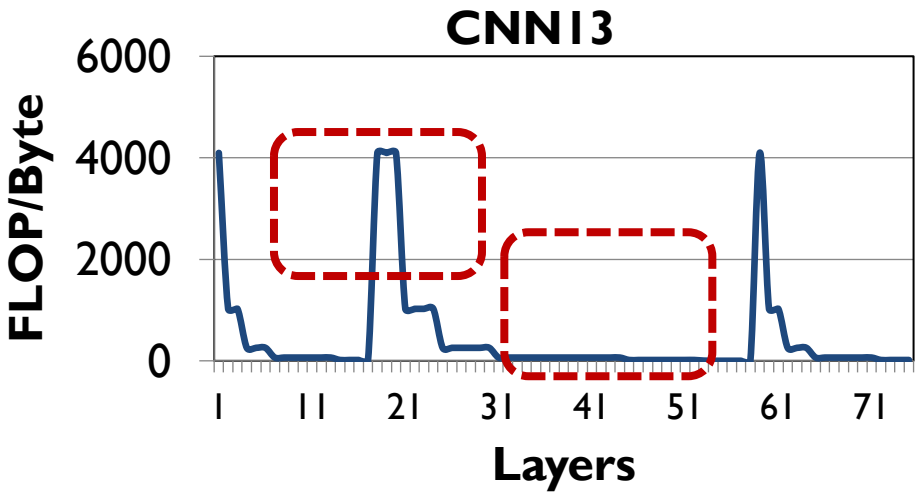


Diversity Within the Models

Insight 2: even within each model, layers exhibit significant variation in terms of layer characteristics

For example, our analysis of edge CNN models shows:



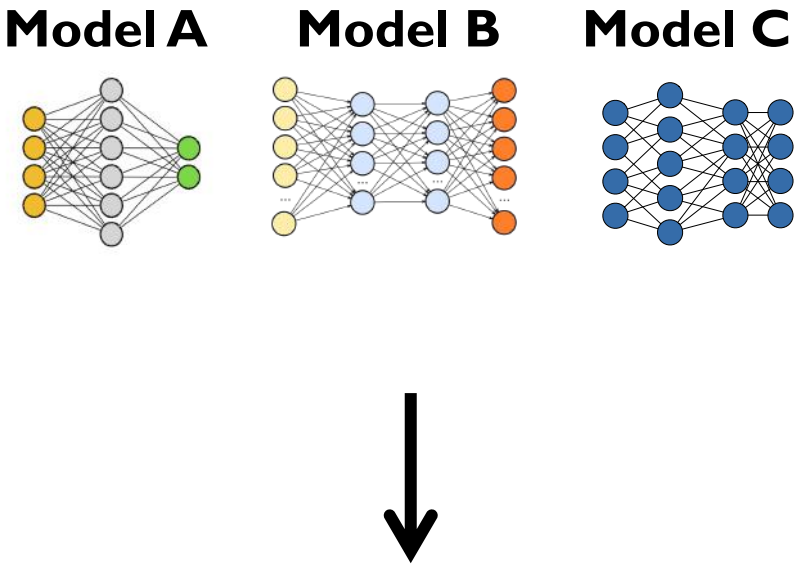


Variation in MAC intensity: up to 200x across layers

Variation in FLOP/Byte: up to 244x across layers

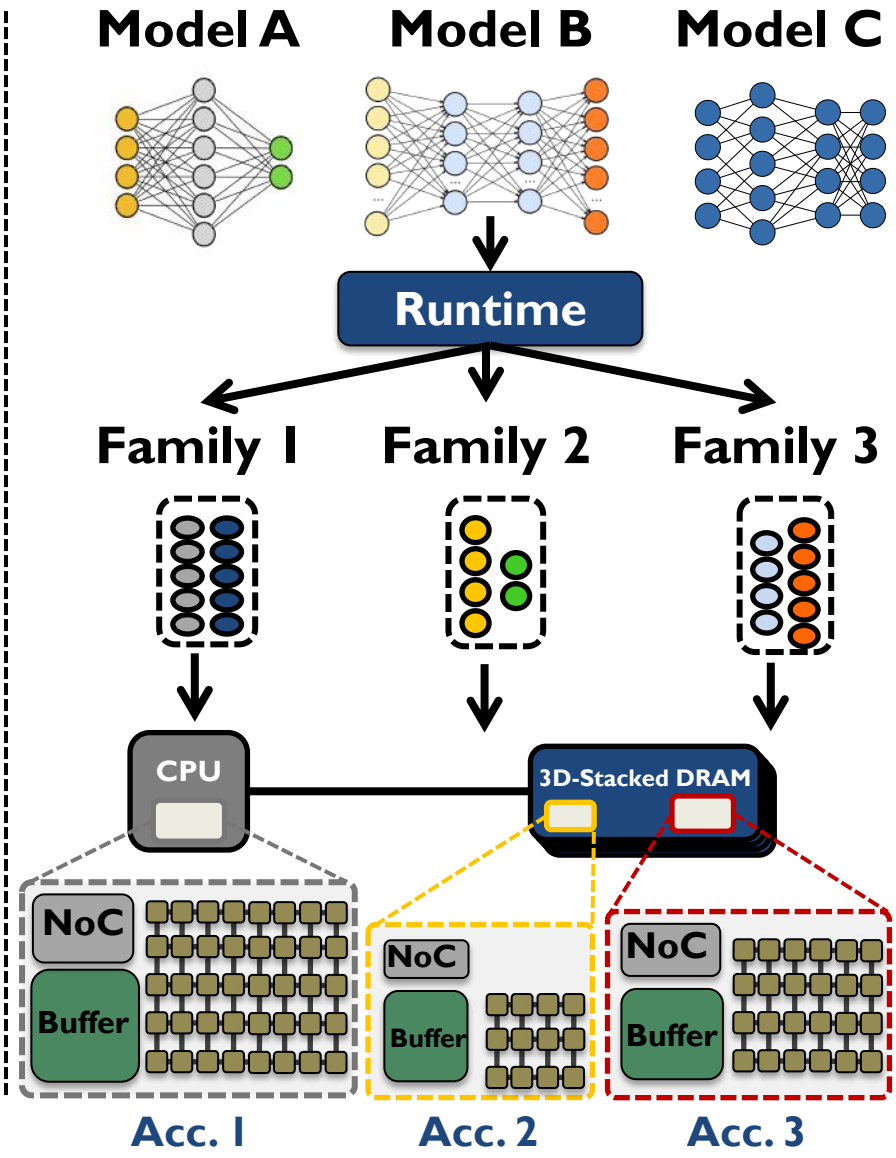
Mensa High-Level Overview

Edge TPU Accelerator



Monolithic Accelerator

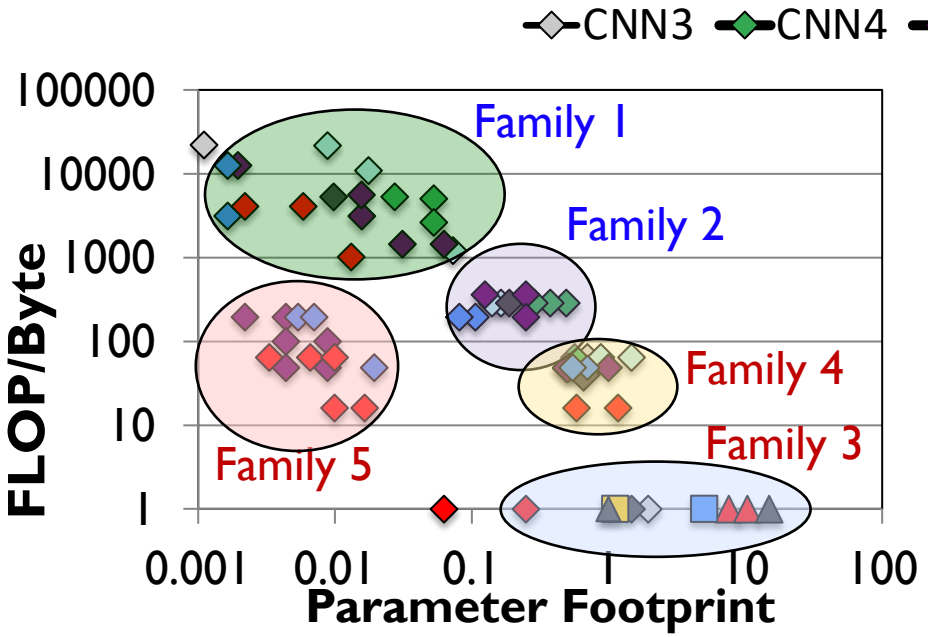
Mensa

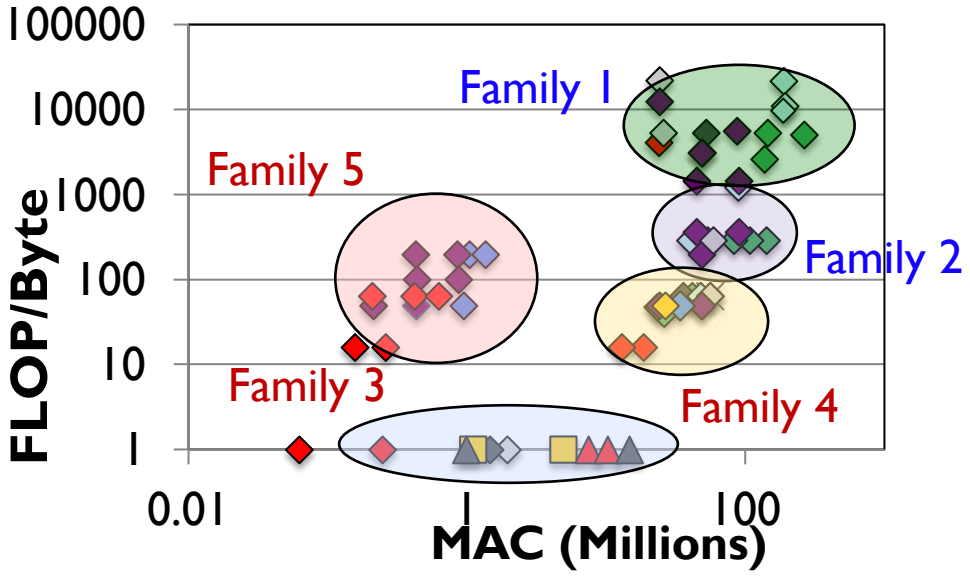


Heterogeneous Accelerators

Identifying Layer Families

Key observation: the majority of layers group into a small number of layer families

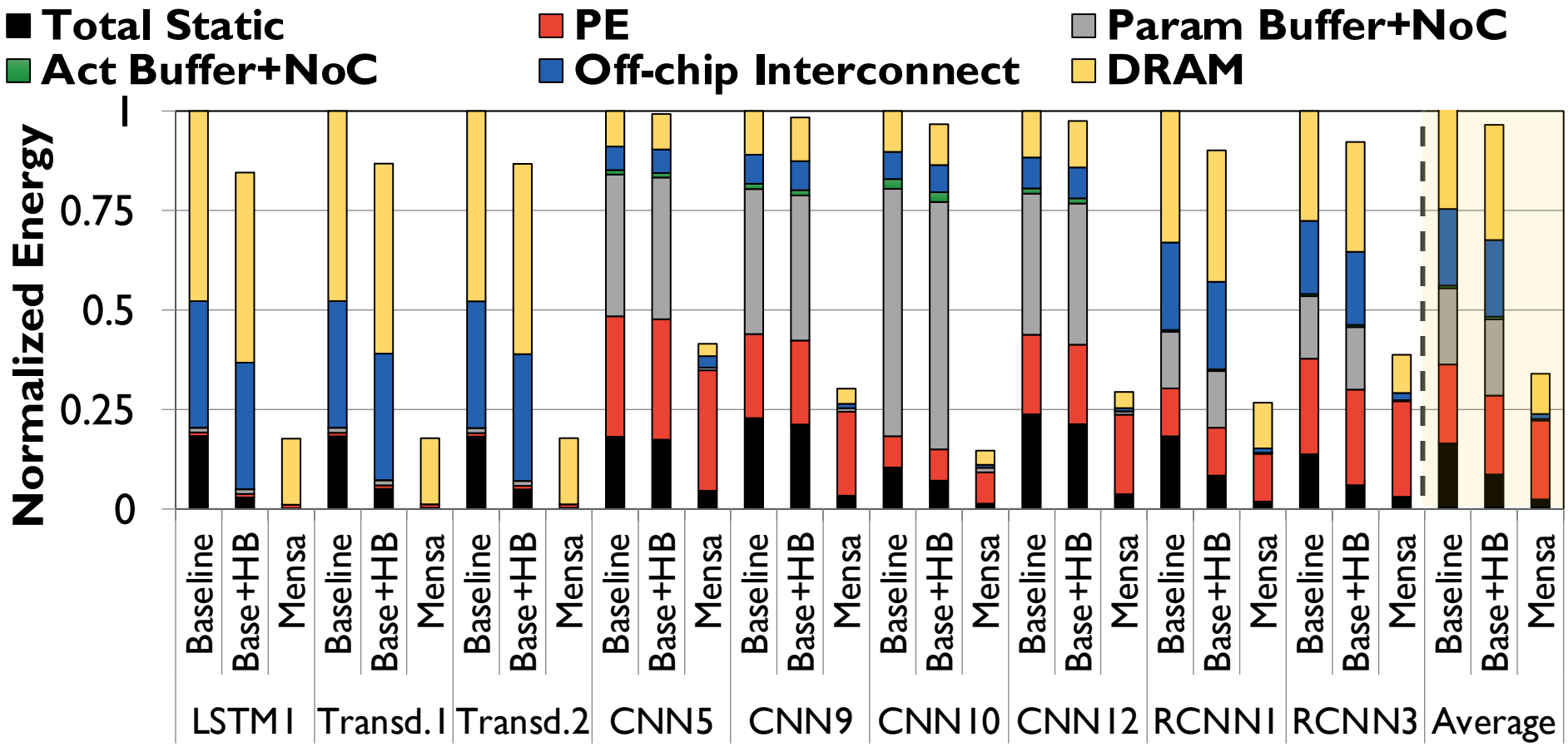




Families 1 & 2: low parameter footprint, high data reuse and MAC intensity
→ **compute-centric layers**

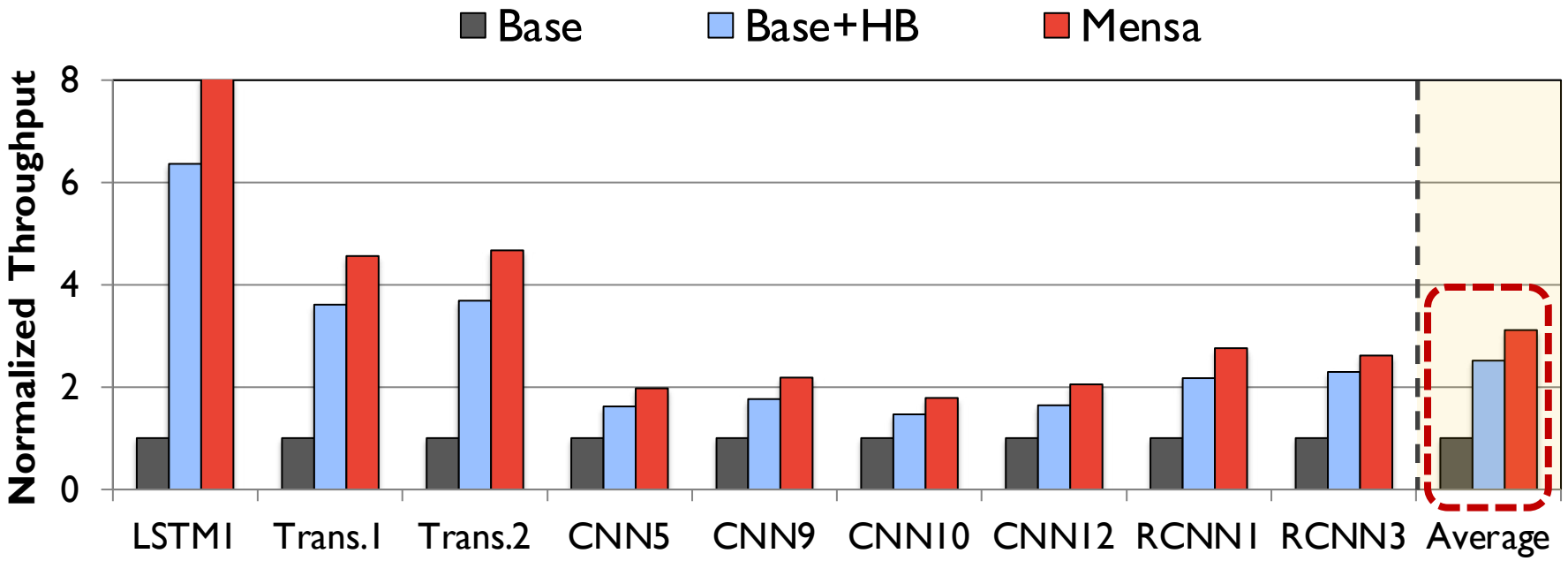
Families 3, 4 & 5: high parameter footprint, low data reuse and MAC intensity
→ **data-centric layers**

Mensa: Energy Reduction



Mensa-G reduces energy consumption by 3.0X compared to the baseline Edge TPU

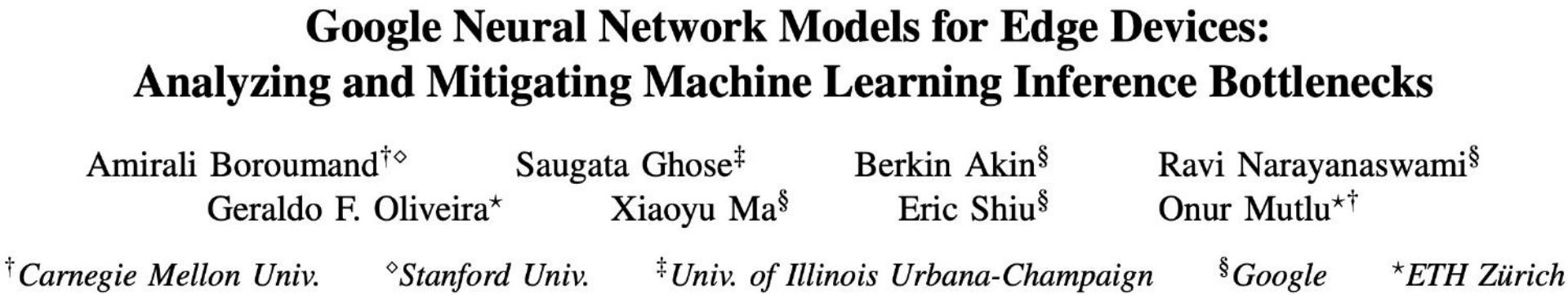
Mensa: Throughput Improvement



Mensa-G improves inference throughput by 3.1X compared to the baseline Edge TPU

Mensa: Highly-Efficient ML Inference

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



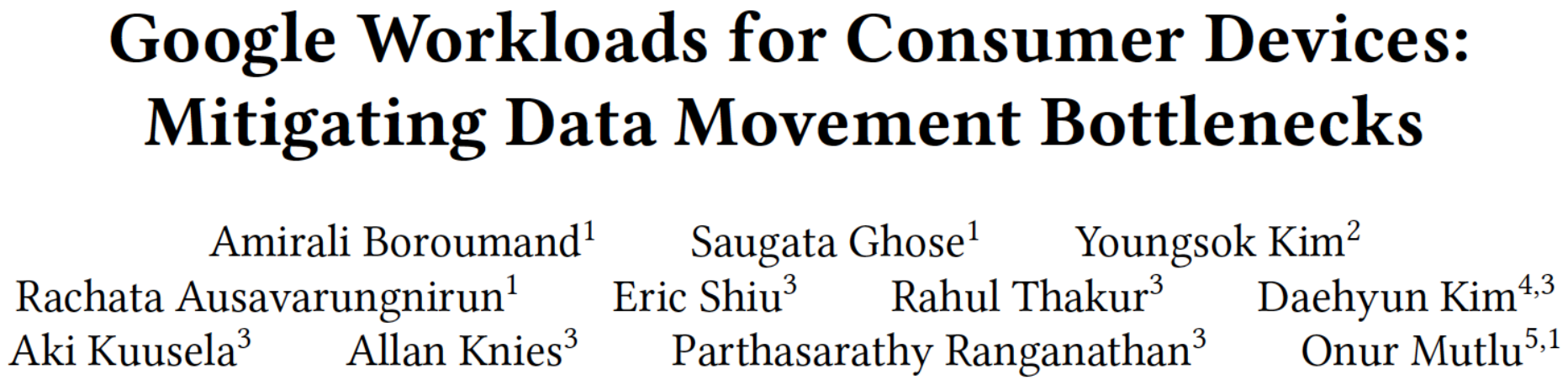
Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

Accelerating 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.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]
  [Lightning Talk Video (2 minutes)]
  [Full Talk Video (21 minutes)]



Accelerating DNA Read Mapping

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

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

In-Storage Genomic Data Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
  "GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
  *Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Virtual, February-March 2022.
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi[1] Jisung Park[1] Harun Mustafa[1] Jeremie Kim[1] Ataberk Olgun[1]
Arvid Gollwitzer[1] Damla Senol Cali[2] Can Firtina[1] Haiyu Mao[1] Nour Almadhoun Alserr[1]
Rachata Ausavarungnirun[3] Nandita Vijaykumar[4] Mohammed Alser[1] Onur Mutlu[1]

[1]ETH Zürich [2]Bionano Genomics [3]KMUTNB [4]University of Toronto

In-Storage Metagenomics [ISCA 2024]

- Nika Mansouri Ghiasi, Mohammad Sadrosadati, Harun Mustafa, Arvid Gollwitzer, Can Firtina, Julien Eudine, Haiyu Mao, Joel Lindegger, Meryem Banu Cavlak, Mohammed Alser, Jisung Park, and Onur Mutlu,
  "MegIS: High-Performance and Low-Cost Metagenomic Analysis with In-Storage Processing"
  *Proceedings of the 51st Annual International Symposium on Computer Architecture (**ISCA**)*, Buenos Aires, Argentina, July 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]

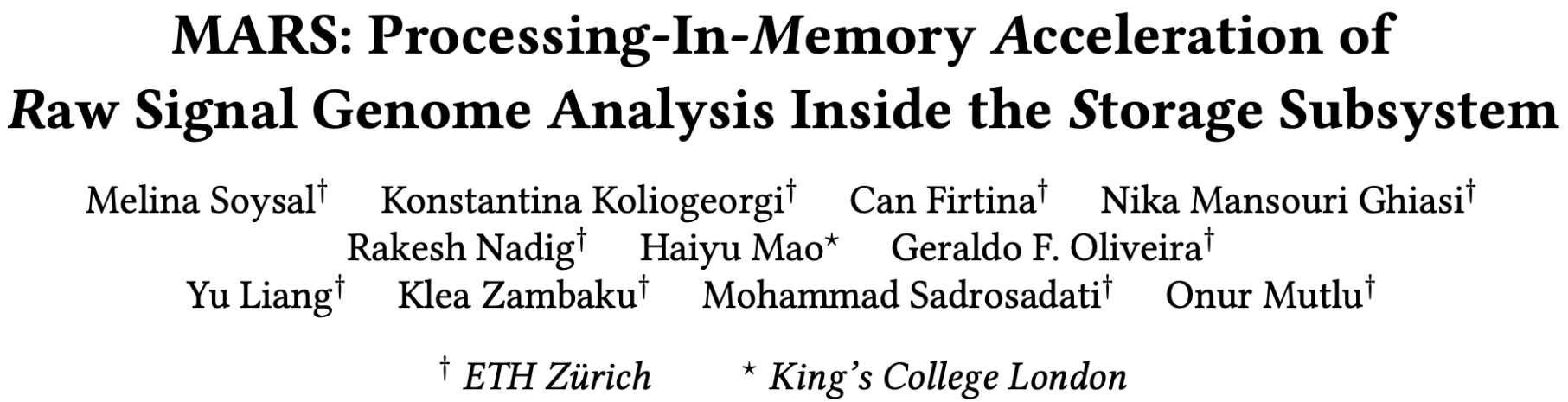
MegIS: High-Performance, Energy-Efficient, and Low-Cost Metagenomic Analysis with In-Storage Processing

Nika Mansouri Ghiasi[1] Mohammad Sadrosadati[1] Harun Mustafa[1] Arvid Gollwitzer[1]
Can Firtina[1] Julien Eudine[1] Haiyu Mao[1] Joël Lindegger[1] Meryem Banu Cavlak[1]
Mohammed Alser[1] Jisung Park[2] Onur Mutlu[1]
[1]ETH Zürich [2]POSTECH

https://arxiv.org/pdf/2406.19113

Accelerating Raw Signal Genome Analysis

- Melina Soysal, Konstantina Koliogeorgi, Can Firtina, Nika Mansouri Ghiasi, Rakesh Nadig, Haiyu Mao, Geraldo Francisco, Yu Liang, Klea Zambaku, Mohammad Sadrosadati, and Onur Mutlu,
  "MARS: Processing-In-Memory Acceleration of Raw Signal Genome Analysis Inside the Storage Subsystem"
  *Proceedings of the 37th ACM International Conference on Supercomputing (**ICS**)*, Salt Lake City, UT, USA, June 2025.



MARS: Processing-In-*Memory* *Acceleration* of *Raw* Signal Genome Analysis Inside the *Storage* Subsystem

Melina Soysal[†] Konstantina Koliogeorgi[†] Can Firtina[†] Nika Mansouri Ghiasi[†]
Rakesh Nadig[†] Haiyu Mao[⋆] Geraldo F. Oliveira[†]
Yu Liang[†] Klea Zambaku[†] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†] *ETH Zürich* [⋆] *King's College London*

https://arxiv.org/pdf/2506.10931

Accelerating Retrieval Augmented Generation

- Kangqi Chen, Rakesh Nadig, Andreas Kosmas Kakolyris, Manos Frouzakis, Nika Mansouri Ghiasi, Yu Liang, Haiyu Mao, Jisung Park, Mohammad Sadrosadati, and Onur Mutlu,
  "REIS: A High-Performance and Energy-Efficient Retrieval System with In-Storage Processing"
  *Proceedings of the 52nd Annual International Symposium on Computer Architecture (**ISCA**)*, Tokyo, Japan, June 2025.

REIS: A High-Performance and Energy-Efficient Retrieval System with In-Storage Processing

Kangqi Chen[1] Andreas Kosmas Kakolyris[1] Rakesh Nadig[1] Manos Frouzakis[1]
Nika Mansouri Ghiasi[1] Yu Liang[1] Haiyu Mao[1,2]
Jisung Park[3] Mohammad Sadrosadati[1] Onur Mutlu[1]

ETH Zürich[1] King's College London[2] POSTECH[3]

Many More Examples …

A Modern Primer on Processing-In-Memory

Onur Mutlu[a], Saugata Ghose[b], Juan Gómez-Luna[c], Rachata Ausavarungnirun[d],
Mohammad Sadrosadati[a], Geraldo F. Oliveira[a]

SAFARI Research Group

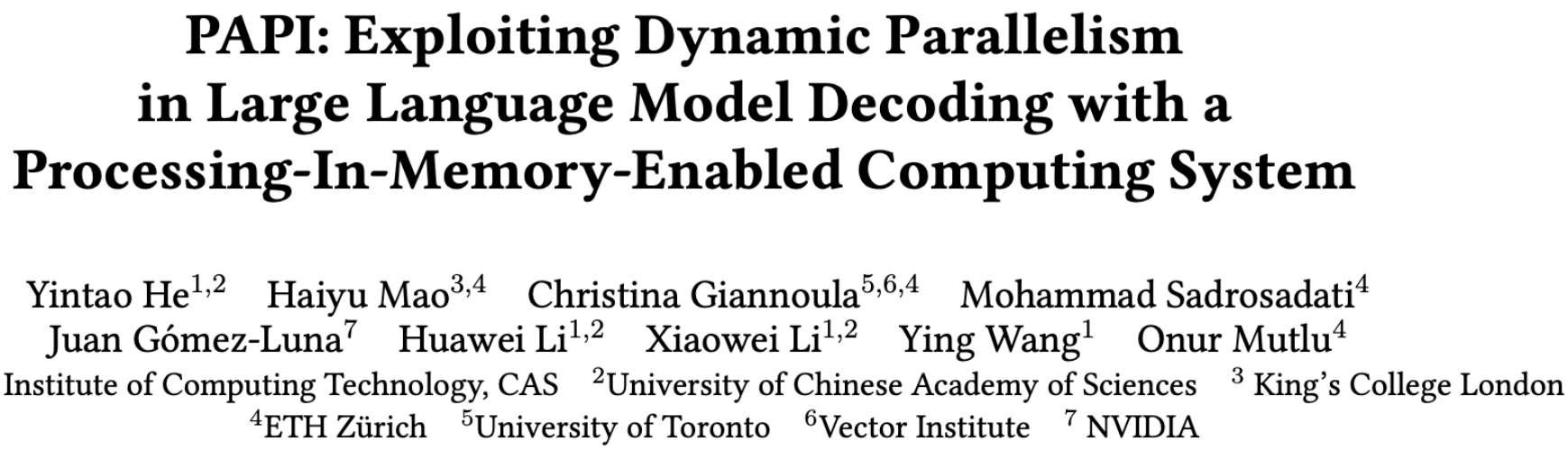
[a] *ETH Zürich*
[b] *University of Illinois Urbana-Champaign*
[c] *NVIDIA Research*
[d] *MangoBoost Inc.*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, Rachata Ausavarungnirun, Mohammad Sadrosadati, and Geraldo F. Oliveira,
"A Modern Primer on Processing in Memory"
Invited Book Chapter in ***Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann***, Springer, 2022.

https://arxiv.org/pdf/2012.03112

PAPI: Hybrid System for Near-Memory LLM Inference

- Yintao He, Haiyu Mao, Christina Giannoula, Mohammad Sadrosadati, Juan Gomez-Luna, Huawei Li, Xiaowei Li, Ying Wang, and Onur Mutlu, **"PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System,"** *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems* (***ASPLOS***), Rotterdam, Netherlands, April 2025.



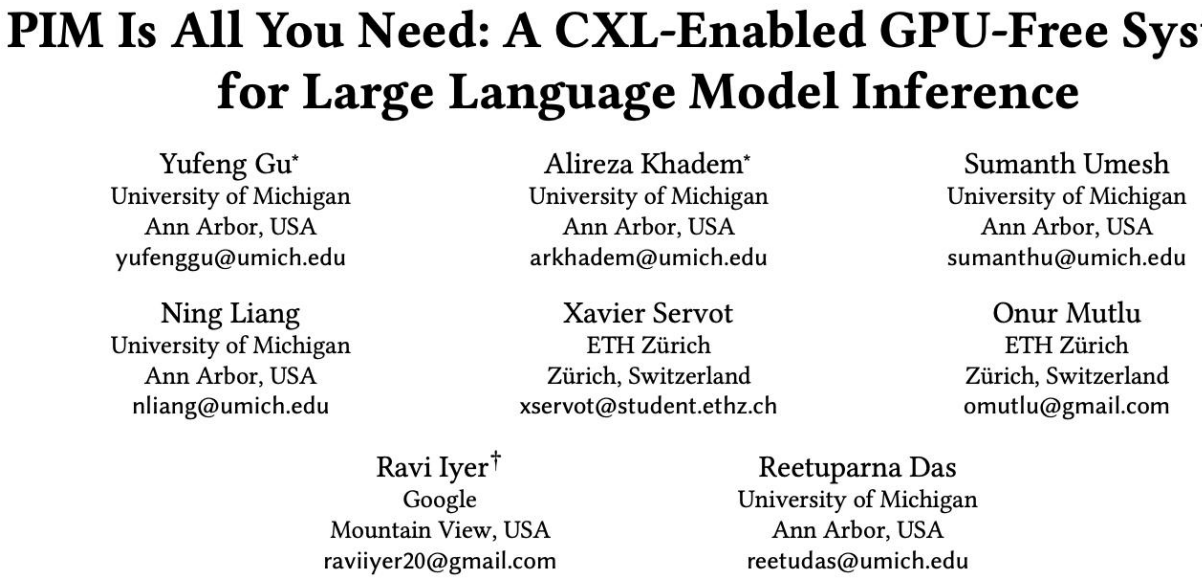
PAPI: Exploiting Dynamic Parallelism in Large Language Model Decoding with a Processing-In-Memory-Enabled Computing System

Yintao He[1,2] Haiyu Mao[3,4] Christina Giannoula[5,6,4] Mohammad Sadrosadati[4]
Juan Gómez-Luna[7] Huawei Li[1,2] Xiaowei Li[1,2] Ying Wang[1] Onur Mutlu[4]
[1]SKLP, Institute of Computing Technology, CAS [2]University of Chinese Academy of Sciences [3] King's College London
[4]ETH Zürich [5]University of Toronto [6]Vector Institute [7] NVIDIA

https://arxiv.org/pdf/2502.15470

CENT: GPU-Free System for Near-Memory LLM Inference

- Yufeng Gu, Alireza Khadem, Sumanth Umesh, Ning Liang, Xavier Servot, Onur Mutlu, Ravi Iyer, and Reetuparna Das,
  "PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference,"
  Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (***ASPLOS***), Rotterdam, Netherlands, April 2025.
  Officially artifact evaluated as available, functional, and reproducible.



PIM Is All You Need: A CXL-Enabled GPU-Free System for Large Language Model Inference

Yufeng Gu*
University of Michigan
Ann Arbor, USA
yufenggu@umich.edu

Alireza Khadem*
University of Michigan
Ann Arbor, USA
arkhadem@umich.edu

Sumanth Umesh
University of Michigan
Ann Arbor, USA
sumanthu@umich.edu

Ning Liang
University of Michigan
Ann Arbor, USA
nliang@umich.edu

Xavier Servot
ETH Zürich
Zürich, Switzerland
xservot@student.ethz.ch

Onur Mutlu
ETH Zürich
Zürich, Switzerland
omutlu@gmail.com

Ravi Iyer†
Google
Mountain View, USA
raviiyer20@gmail.com

Reetuparna Das
University of Michigan
Ann Arbor, USA
reetudas@umich.edu

https://arxiv.org/pdf/2502.07578

PAPI LLM Inference System [ASPLOS 2025]

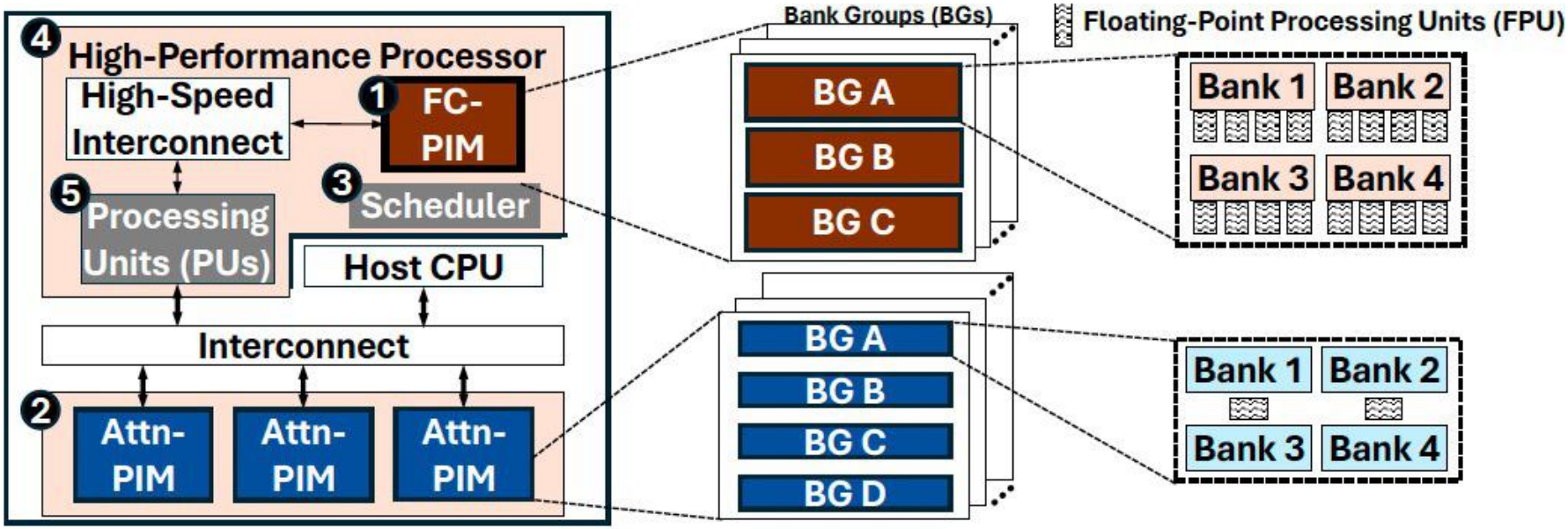


Fig. 5: **Overview of the PAPI LLM Inference System.** Adapted from [18].

PAPI over best prior LLM decoding system

- 1.8× speedup
- 3.4× energy efficiency increase

https://arxiv.org/pdf/2502.15470

CENT LLM Inference System [ASPLOS 2025]

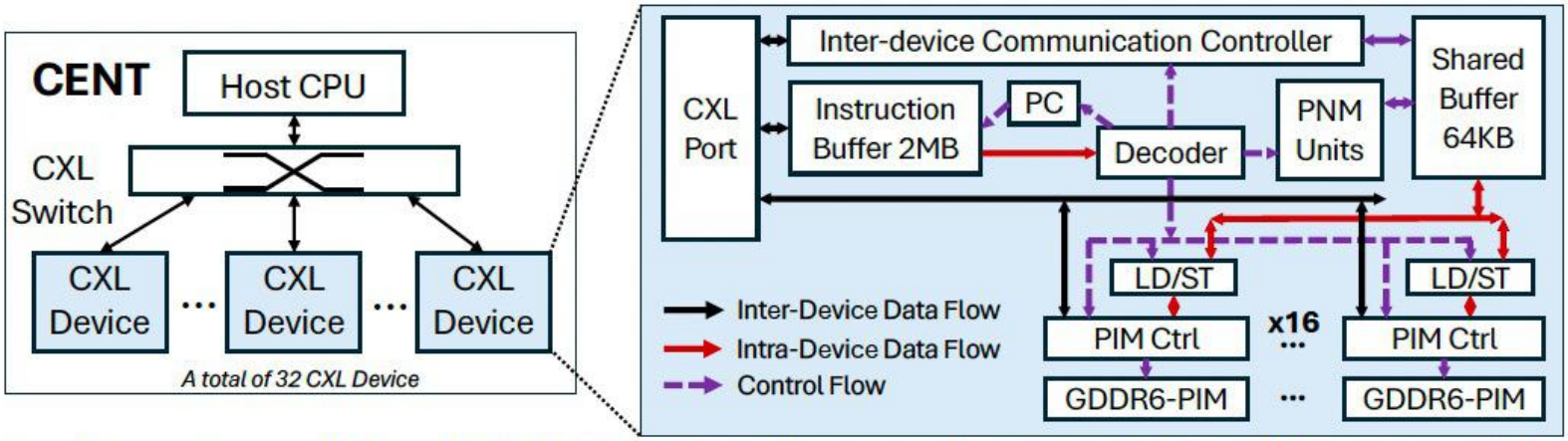


Fig. 6: **Overview of the CENT LLM Inference System.** Host CPU drives 32 CXL devices, each having a CXL controller, PNM units, and 16 GDDR6-PIM chips. The LLM inference task is partitioned between PNM units and GDDR6-PIM chips. CENT provides communication mechanisms within and across CXL devices to coordinate and scale computation. Adapted from [19].

CENT over best prior GPU LLM inference system

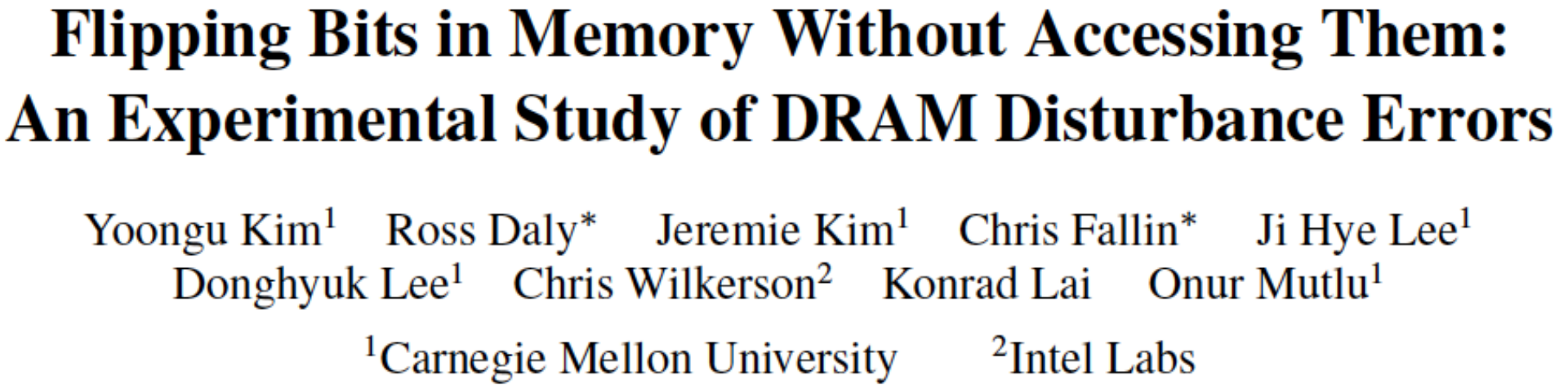
- 2.3× higher throughput
- 5.2× higher tokens per dollar
- 2.4× lower hardware cost

https://arxiv.org/pdf/2502.07578

RowHammer Review History

Original RowHammer Paper [ISCA'14]

- 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] [Lecture Video (1 hr 49 mins), 25 September 2020]
  One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link). Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).



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

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

[1]Carnegie Mellon University [2]Intel Labs

Some More Historical Perspective

- RowHammer: first example of a circuit failure mechanism causing a widespread system security vulnerability

- It led to large body of work in security attacks, mitigations, architectural solutions, analyses, device-level works, …
- It led to large industrial effort: DDR4, DDR5 and other standards already modified in major ways

- Work building on RowHammer continues
  - See many top venues in 2020-2026
  - Dedicated RowHammer sessions in top conferences

- Initially, it was dismissed by some reviewers
  - **Rejected** from MICRO 2013

Initial RowHammer Reviews (MICRO 2013)

#66 Disturbance Errors in DRAM: Demonstration, Characterization, and Prevention

Rejected (R2) 863kB Friday 31 May 2013 2:00:53pm PDT

b9bf06021da54cddf4cd0b3565558a181868b972

You are an **author** of this paper.

+ Abstract

We demonstrate the vulnerability of commodity DRAM chips to disturbance errors. By repeatedly reading from one DRAM address, we show that it is possible to corrupt the data stored [more]

+ Authors

Y. Kim, R. Daly, J. Lee, J. Kim, C. Fallin, C. WIlkerson, O. Mutlu [details]

Keywords: DRAM; errors

+ Topics

	OveMer	Nov	WriQua	RevExp
Review #66A	1	4	4	4
Review #66B	5	4	5	3
Review #66C	2	3	5	4
Review #66D	1	2	3	4
Review #66E	4	4	4	3
Review #66F	2	4	4	3

Reviewer A -- Security is Not "Realistic"

Review #66A Modified Friday 5 Jul 2013 3:59:18am PDT Plain text

OVERALL MERIT (?)

1. Reject

PAPER SUMMARY

This work tests and studies the disturbance problem in DRAM arrays in isolation.

PAPER STRENGTHS

+ Many results and observations.
+ Insights on how the may happen

PAPER WEAKNESSES

- Whereas they show disturbance may happen in DRAM array, authors don't show it can be an issue in realistic DRAM usage scenario
- Lacks architectural/microarchitectural impact on the DRAM disturbance analysis

NOVELTY (?)

4. New contribution.

WRITING QUALITY (?)

4. Well-written

Reviewer A -- Security is Not "Realistic"

COMMENTS FOR AUTHORS

I found the paper very well written and organized, easy to understand. The topic is interesting and relevant.
However, I'm not fully convinced that the disturbance problem is going to be an issue in a realistic DRAM usage scenario (main memory with caches). In that scenarion the 64ms refresh interval might be enough. Overall, the work presented, the experimenation and the results are not enough to justify/claim that disturbance may be an issue for future systems, and that microarchitectural solutions are required.

I really encourage the authors to address this issue, to run the new set of experiments; if the results are positive, the work is great and will be easily accepted in a top notch conference. Test scenario in the paper (open-read-close a row many times consecutively) that is used to create disturbances is not likely to show up in a realistic usage scenario (check also rebuttal question).

Rebuttal to Reviewer A

_____WILL IT AFFECT REAL WORKLOADS ON REAL SYSTEMS? (A, E)_____

Malicious workloads and pathological access-patterns can bypass/thrash the cache
and access the same DRAM row a very large number of times. While these workloads
may not be common, they are just as real. Using non-temporal

Reviewer A -- Demands

To make sure that correct information and messages are given to the research community, it would be good if the conclusions drawn in the paper were verified with the actual DRAM manufacturers, although I see that it can be difficult to do. In addition, knowing the technology node of each tested DRAM would make the paper stronger and would avoid speculative guesses.

REVIEWER EXPERTISE (?)

4. Expert in area, with highest confidence in review.

Reviewer C – No Architectural Content

Review #66C Modified Friday 12 Jul 2013 7:38:57am PDT

Plain text

Overall merit (?)

2. Weak reject

Paper summary

This paper presents a rigorous study of DRAM module errors which are observed to be caused through repeated access to the same address in the DRAMs.

Paper strengths

The paper's measurement methodology is outstanding, and the authors very thoroughly dive into different test scenarios, to isolate the circumstances under which the observed errors take place.

Paper weaknesses

This is an excellent test methodology paper, but there is no micro-architectural or architectural content.

Novelty (?)

3. Incremental improvement.

Writing quality (?)

5. Outstanding

Questions to address in the rebuttal

My primary concern with this paper is that it doesn't have (micro-)architectural content, and may not spur on future work.

Reviewer C -- Leave It to DRAM Vendors

COMMENTS FOR AUTHORS

This is an extremely well-written analysis of DRAM behavior, and the authors are to be commended on establishing a robust and flexible characterization platform and methodology.

That being said, disturb errors have occurred repeatedly over the course of DRAM's history (which the authors do acknowledge). History has shown that particular disturbances, and in particular hammer errors, are short-lived, and are quickly solved by DRAM manufacturers. Historically, once these these types of errors occur at a particular lithography node/DRAM density, they must be solved by the DRAM manufacturers, because even if a solution for a systemic problem could be asserted for particular markets (e.g., server, where use of advanced coding techniques, extra chips, etc. is acceptable), there will always be significant DRAM chip volume in single-piece applications (e.g., consumer devices, etc.) where complex architectural solutions aren't an option. The authors have identified a contemporary disturb sensitivity in DRAMs, but as non-technologists, our community can generally only observe, not correct, such problems.

REVIEWER EXPERTISE (?)

4. Expert in area, with highest confidence in review.

Reviewer D -- Nothing New in RowHammer

Review #66D Modified Thursday 18 Jul 2013 12:51pm PDT Plain text

Overall merit (?)

1. Reject

Reviewer expertise (?)

4. Expert in area, with highest confidence in review.

Paper summary

The authors demonstrate that repeated activate-precharge operations on one wordline of a DRAM can disturb a few cells on adjacent wordlines. They showed that such a behavior can be caused for most DRAMs and all DRAMs of recent manufacture they tested.

Paper strengths

DRAM errors are getting more likely with newer generations and it is necessary to investigate their cause and mitigation in computer systems, as such the paper addresses a subtopic of a relevant problem.

Paper weaknesses

The mechanism investigated by the authors is one of many well known disturb mechanisms. The paper does not discuss the root causes to sufficient depth and the importance of this mechanism compared to others. Overall the length of the sections restating known information is much too long in relation to new work.

Novelty (?)

2. Insignificant novelty. Virtually all of the ideas are published or known.

Writing quality (?)

3. Adequate

ISCA 2014 Submission

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

N **Accepted**  639kB 21 Nov 2013 10:53:11pm CST |
f039be2735313b39304ae1c6296523867a485610

You are an **author** of this paper.

+ 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 [more]

+ AUTHORS

Y. Kim, R. Daly, J. Kim, J. Lee, C. Fallin, C. Wilkerson, O. Mutlu [details]

+ TOPICS

	OveMer	Nov	WriQua	RevConAnd
Review #41A	8	4	5	3
Review #41B	7	4	4	3
Review #41C	6	4	4	3
Review #41D	2	2	5	4
Review #41E	3	2	3	3
Review #41F	7	4	4	3

Reviewer D – Already Done on Youtube

Review #41D Modified 19 Feb 2014 8:47:24pm CST Plain text

OVERALL MERIT (?)

2. Reject

PAPER SUMMARY

The authors
1) characterize disturbance error in commodity DRAM
2) identify the root cause such errors (but it's already a well know problem in DRAM community).
3) propose a simple architectural technique to mitigate such errors.

PAPER STRENGTHS

The authors demonstrated the problem using the real systems

PAPER WEAKNESSES

1) The disturbance error (a.k.a coupling or cross-talk noise induced error) is a known problem to the DRAM circuit community.
2) What you demonstrated in this paper is so called DRAM row hammering issue - you can even find a Youtube video showing this! - http://www.youtube.com/watch?v=i3-gOSnBcdo
2) The architectural contribution of this study is too insignificant.

Novelty (?)

2. Insignificant novelty. Virtually all of the ideas are published or known.

Writing Quality (?)

5. Outstanding

Reviewer Confidence and Expertise (?)

4. Expert in area, with highest confidence in review.

Questions for Authors

1. There are other sources of disturbance errors How can you guarantee the errors observed by you are not from such errors?

2. You did you best on explaining why we have much fewer 1->0 error but not quite satisfied. Any other explanation?

3. Can you elaborate why we have more disturbed cells over rounds while you claim that disturbed cells are not weak cells? I'm sure this is related to device again issues

Detailed Comments

This is a well written and executed paper (in particular using real systems), but I have many concerns:

1) this is a well-known problem to the DRAM community (so no novelty there); in DRAM community people use

Reviewer D Continued…

2) what you did to incur disturbance is is so called "row hammering" issues - please see http://www.youtube.com/watch?v=i3-gQSnBcdo - a demonstration video for capturing this problem...

3) the relevance of this paper to ISCA. I feel that this paper (most part) is more appropriate to conferences like International Test Conference (ITC) or VLSI Test Symposium or Dependable Systems and Networks (DSN) at most. This is because the authors mainly dedicated the effort to the DRAM circuit characterization and test method in my view while the architectural contribution is very weak - I'm not even sure this can be published to these venues since it's a well known problem! I also assume techniques proposed to minimize disturbance error in STT-RAM and other technology can be employed here as well.

Rebuttal to Reviewer D

_____Reviewer D (Comments)_____

- 1. As we acknowledge in the paper, it is true that different
types of DRAM coupling phenomena have been known to the DRAM
circuits/testing community. However, there is a clear
distinction between circuits/testing techniques confined to the
foundry versus characterization/solution of a problem out in
the *field*. The three citations (from 10+ years ago) do *not*
demonstrate that disturbance errors exist in DIMMs sold then or
now. They do *not* provide any real data (only simulated ones),
let alone a large-scale characterization across many DIMMs from
multiple manufacturers. They do *not* construct an attack on
real systems, and they do *not* provide any solutions. Finally,
our paper *already* references all three citations, or their
more relevant equivalents. (The second/third citations provided
by the reviewer are on bitline-coupling, whereas we cite works
from the same authors on wordline-coupling [2, 3, 37].)

- 2. We were aware of the video from Teledyne (a test equipment
company) and have *already* referenced slides from the same
company [36]. In terms of their content regarding "row hammer",
the video and the slides are identical: all they mention is
that "aggressive row activations can corrupt adjacent rows".
(They then advertise how their test equipment is able to
capture a timestamped DRAM access trace, which can then be
post-processed to identify when the number of activations
exceeds a user-set threshold.) Both the video and slides do
not say that this is a real problem affecting DIMMs on the
market now. They do *not* provide any quantitative data, *nor*
real-system demonstration, *nor* solution.

Reviewer E

Review #41E Modified 7 Feb 2014 11:08:04pm CST Plain text

OVERALL MERIT (?)

3. Weak Reject

PAPER SUMMARY

This paper studies the row disturbance problem in DRAMs. The paper includes a thorough quantitative characterization of the problem and a qualitative discussion of the source of the problem and potential solutions.

PAPER STRENGTHS

+ The paper provides a detailed quantitative characterization of the "row hammering" problem in memories.

PAPER WEAKNESSES

- Row Hammering appears to be well-known, and solutions have already been proposed by industry to address the issue.

- The paper only provides a qualitative analysis of solutions to the problem. A more robust evaluation is really needed to know whether the proposed solution is necessary.

NOVELTY (?)

2. Insignificant novelty. Virtually all of the ideas are published or known.

WRITING QUALITY (?)

3. Adequate

REVIEWER CONFIDENCE AND EXPERTISE (?)

3. Knowledgeable in area, and significant confidence in

but there are numerous mentions of hammering in the literature, and clearly industry has studied this problem for many years. In particular, Intel has a patent application on a memory controller technique that addresses this exact problem, with priority date June 2012:

http://www.google.com/patents/WO2014004748A1?cl=en

The patent application details sound very similar to solution 6 in this paper, so a more thorough comparison with solution 7 seems mandatory.

My overall feeling is that while the reliability characterization is important and interesting, a better target audience for the characterization work would be in a testing/reliability venue. The most interesting part of this paper from the ISCA point of view are the proposed solutions, but all of these are discussed in a very qualitative manner. My preference would be to see a much shorter characterization section with a much stronger and quantitative evaluation and comparison of the proposed solutions.

Rebuttal to Reviewer

Nevertheless, we were able to induce a large number of DRAM
disturbance errors on all the latest Intel/AMD platforms that we
tested: Haswell, Ivy Bridge, Sandy Bridge, and Piledriver. (At
the time of submission, we had tested only Sandy Bridge.)
Importantly, the patents do *not* provide quantitative characterization

nor real-system demonstration.

[R1] "Row Hammer Refresh Command." US20140006703 A1
[R2] "Row Hammer Condition Monitoring." US20140006704 A1

_____Reviewer E (Comments)_____

After our paper was submitted, two patents that had been filed by

Intel were made public (one is mentioned by the reviewer [R1]).
Together, the two patents describe what we posed as the *sixth*
potential solution in our paper (Section 8). Essentially, the
memory controller maintains a table of counters to track the
number of activations to recently activated rows [R2]. And if one
of the counters exceeds a certain threshold, the memory controller notifies the DRAM chips using a special command [R1].
The DRAM chips would then refresh an entire "region" of rows that
includes both the aggressor and its victim(s) [R1]. For the
patent [R1] to work, DRAM manufacturers must cooperate and
implement this special command. (It is a convenient way of
circumventing the opacity in the logical-physical mapping. If
implemented, the same command can also be used for our *seventh*
solution.) The limitation of this *sixth* solution is the storage
overhead of the counters and the extra power required to associatively search through them on every activation (Section
8). That is why we believe our *seventh* solution to be more
attractive. We will cite the patents and include a more concrete
comparison between the two solutions.

As Evaluators of Scientific Work …

Can We Do Better?

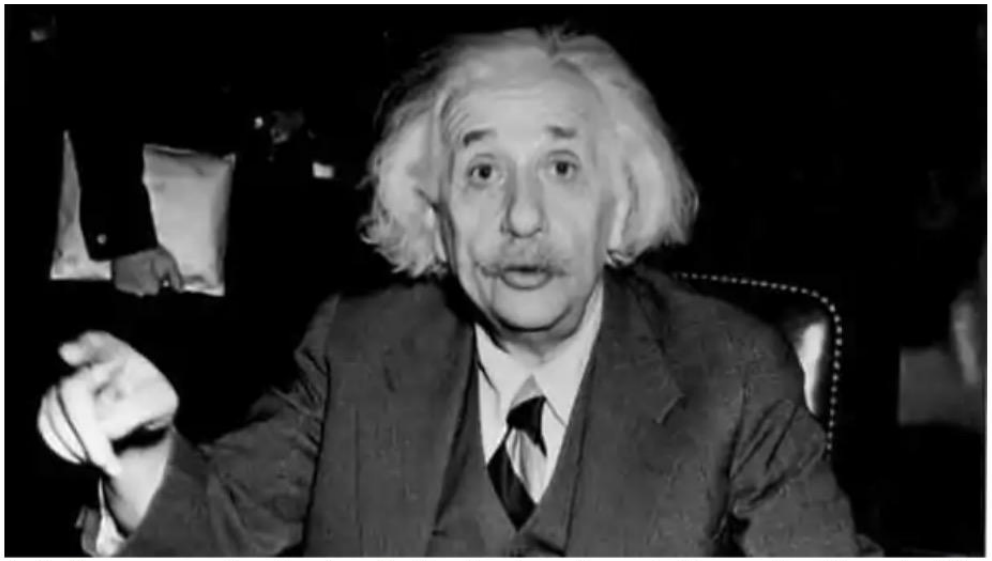
Suggestions to Reviewers

- Be fair; you do not know it all (past, present, future)
- Be open-minded; you do not know it all (past, present, future)
- Be accepting of diverse research methods: there is no single way of doing research or writing papers
- Be constructive, not destructive; also accept more
- Do **not** have double standards… suppress your biases…

Do not block or delay progress for non-reasons

A Fun Reading: Food for Thought

- https://www.livemint.com/science/news/could-einstein-get-published-today-11601014633853.html



A similar process of professionalization has transformed other parts of the scientific landscape. (Central Press/Getty Images)

THE WALL STREET JOURNAL.

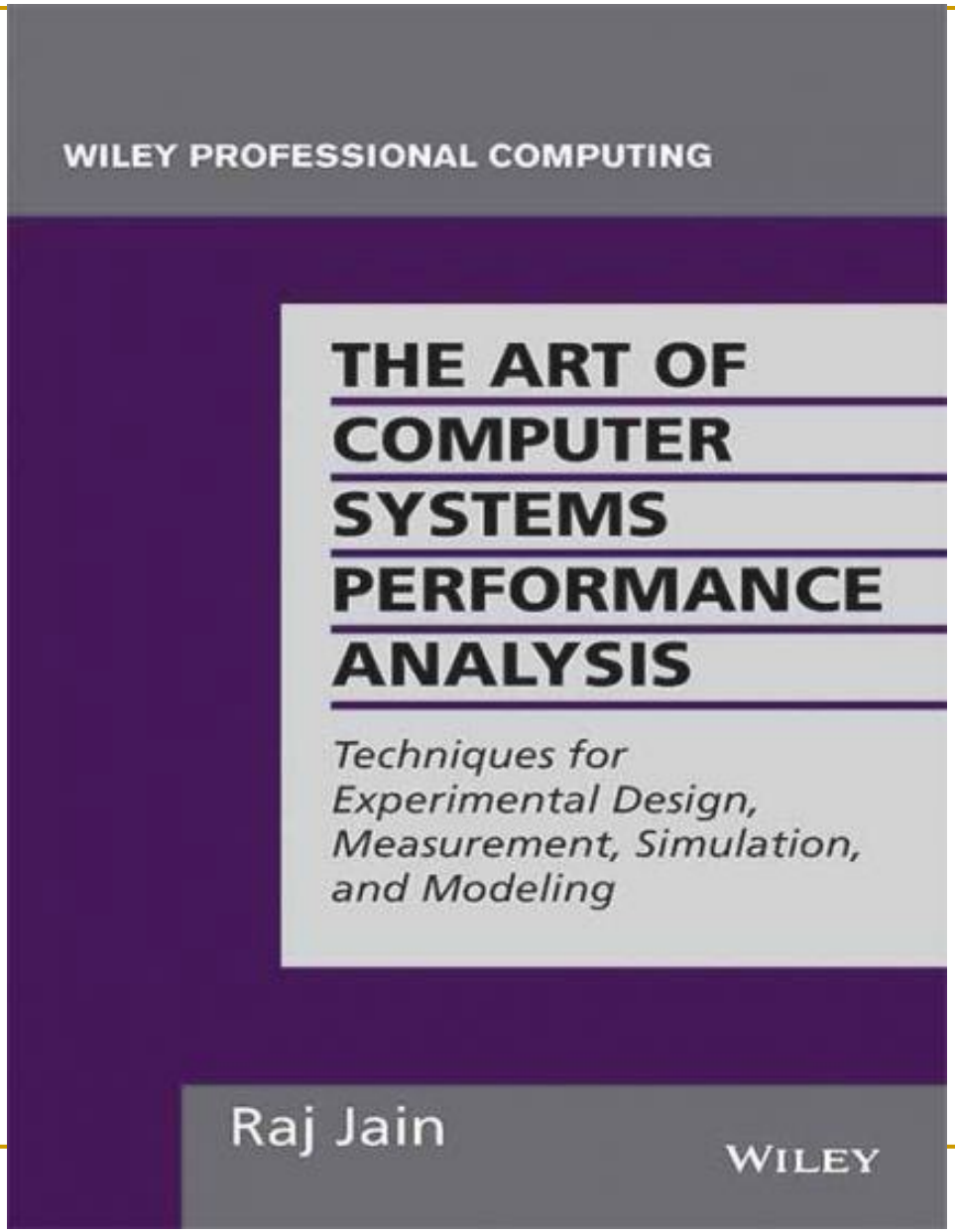
Could Einstein get published today?

3 min read . Updated: 25 Sep 2020, 11:51 AM IST

The Wall Street Journal

Scientific journals and institutions have become more professionalized over the last century, leaving less room for individual style

Aside: A Recommended Book



Raj Jain, "The Art of Computer Systems Performance Analysis," Wiley, 1991.

10.8 DECISION MAKER'S GAMES

Even if the performance analysis is correctly done and presented, it may not be enough to persuade your audience—the decision makers—to follow your recommendations. The list shown in Box 10.2 is a compilation of reasons for rejection heard at various performance analysis presentations. You can use the list by presenting it immediately and pointing out that the reason for rejection is not new and that the analysis deserves more consideration. Also, the list is helpful in getting the competing proposals rejected!

There is no clear end of an analysis. Any analysis can be rejected simply on the grounds that the problem needs more analysis. This is the first reason listed in Box 10.2. The second most common reason for rejection of an analysis and for endless debate is the workload. Since workloads are always based on the past measurements, their applicability to the current or future environment can always be questioned. Actually workload is one of the four areas of discussion that lead a performance presentation into an endless debate. These "rat holes" and their relative sizes in terms of time consumed are shown in Figure 10.26. Presenting this cartoon at the beginning of a presentation helps to avoid these areas.

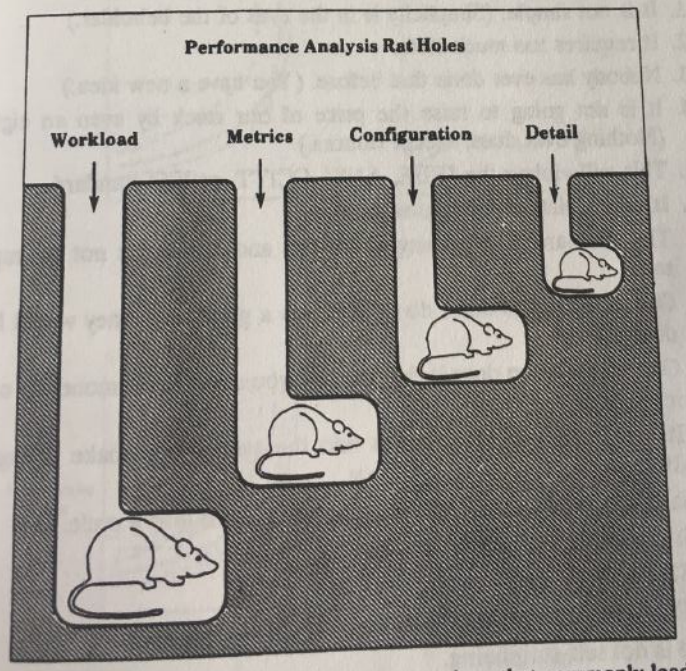
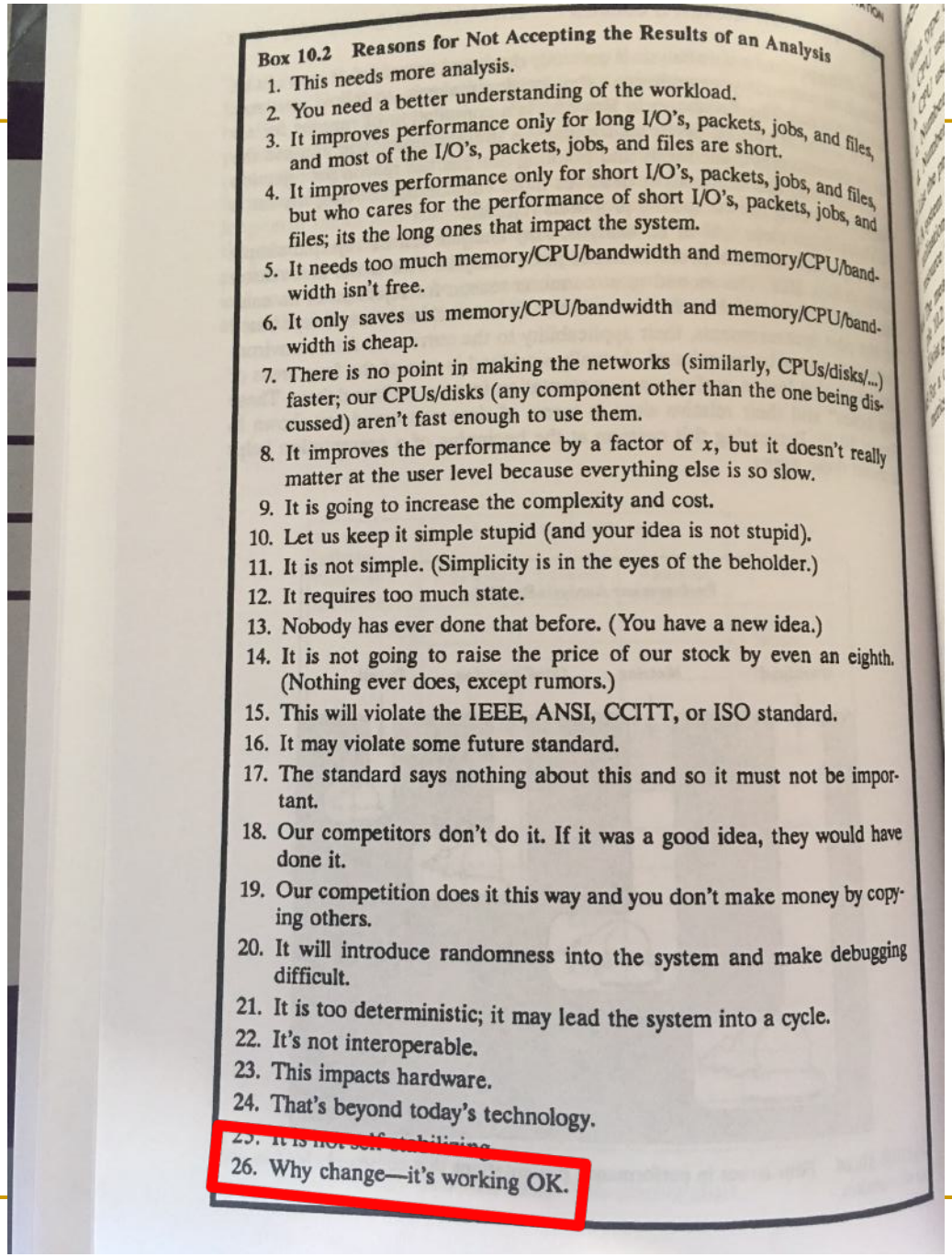


FIGURE 10.26 Four issues in performance presentations that commonly lead to endless discussion.

Raj Jain, "The Art of Computer Systems Performance Analysis," Wiley, 1991.



Box 10.2 Reasons for Not Accepting the Results of an Analysis

1. This needs more analysis.
2. You need a better understanding of the workload.
3. It improves performance only for long I/O's, packets, jobs, and files, and most of the I/O's, packets, jobs, and files are short.
4. It improves performance only for short I/O's, packets, jobs, and files, but who cares for the performance of short I/O's, packets, jobs, and files; its the long ones that impact the system.
5. It needs too much memory/CPU/bandwidth and memory/CPU/bandwidth isn't free.
6. It only saves us memory/CPU/bandwidth and memory/CPU/bandwidth is cheap.
7. There is no point in making the networks (similarly, CPUs/disks/...) faster; our CPUs/disks (any component other than the one being discussed) aren't fast enough to use them.
8. It improves the performance by a factor of x, but it doesn't really matter at the user level because everything else is so slow.
9. It is going to increase the complexity and cost.
10. Let us keep it simple stupid (and your idea is not stupid).
11. It is not simple. (Simplicity is in the eyes of the beholder.)
12. It requires too much state.
13. Nobody has ever done that before. (You have a new idea.)
14. It is not going to raise the price of our stock by even an eighth. (Nothing ever does, except rumors.)
15. This will violate the IEEE, ANSI, CCITT, or ISO standard.
16. It may violate some future standard.
17. The standard says nothing about this and so it must not be important.
18. Our competitors don't do it. If it was a good idea, they would have done it.
19. Our competition does it this way and you don't make money by copying others.
20. It will introduce randomness into the system and make debugging difficult.
21. It is too deterministic; it may lead the system into a cycle.
22. It's not interoperable.
23. This impacts hardware.
24. That's beyond today's technology.
25. It is not self-stabilizing.
26. Why change—it's working OK.

Raj Jain, "The Art of Computer Systems Performance Analysis," Wiley, 1991.

Reviews **After** the Paper Was Published (I)

Review #54B Modified 23 Dec 2014 12:31:15pm PST Plain text

I poked around a bit and DRAM vendors have already solved this problem. DRAM row hammering appears to be a known problem.

Reviews **After** the Paper Was Published (II)

Review #54D Modified 1 Jan 2015 4:13:18pm PST Plain text

CHANCE OF IMPACT (?)

3. Minor impact

OVERALL MERIT (?)

2. Weak reject (Happy to discuss but unlikely to be chosen.)

COMMENTS FOR AUTHOR

Interesting paper for those interested in DRAM issues. I wonder if it is possible to gain an insight into why this happens.

I seem to remember that, during the presentation at ISCA, it was pointed out that DRAM manufacturers have already fixed the
problem. So where is the novelty and long term impact?

Reviews **After** the Paper Was Published (III)

Review #54E Modified 4 Jan 2015 4:40:44am PST Plain text

SHORT PAPER SUMMARY

This paper identifies a new class of DRAM errors called "disturbance" errors. The authors provide a characterization of such errors using DRAM chips dating back to 2008 and show that the disturbance error incidence is a relatively recent phenomenon (after 2010). Finally, the authors explore a set of possible mitigation solutions, while advocating one of them, called PARA (probabilistic adjacent row activation).

CHANCE OF IMPACT (?)

3. Minor impact

OVERALL MERIT (?)

2. Weak reject (Happy to discuss but unlikely to be chosen.)

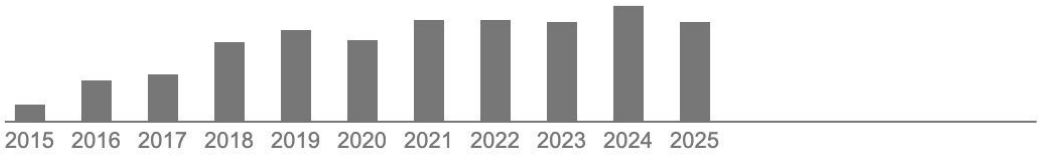
COMMENTS FOR AUTHOR

The authors should be given due credit for identifying and characterizing an emerging new class of DRAM errors. However, it is not clear if this class of errors is significant enough in the future, given the many other modes of failure that DRAM vendors and users are primarily concerned with. As a reader of this paper, I could not but get the feeling that this is an interesting new DRAM error class, but could not find convincing arguments from the paper as to why this would constitute one of the key, first-order error behaviors affecting DRAMS of the future. The mitigation solution offered is simple and effective (I like it); but I was not convinced that this paper will be cited in the future as one that opened up a brand new area of research and consequent use in practice.

A Delayed Rebuttal

Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors

Authors	Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, Onur Mutlu
Publication date	2014
Conference	ISCA 2014
Volume	42
Issue	3
Pages	361-372
Publisher	IEEE Press
Description	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 scales down to smaller dimensions, it becomes more difficult to prevent DRAM cells from electrically interacting with each other. In this paper, we expose the vulnerability of commodity DRAM chips to disturbance errors. By reading from the same address in DRAM, we show that it is possible to corrupt data in nearby addresses. More specifically, activating the same row in DRAM corrupts data in nearby rows. We demonstrate this phenomenon on Intel and AMD systems using a malicious program that generates many DRAM accesses. We induce errors in most DRAM modules (110 out of 129) from three major DRAM manufacturers. From this we conclude that many deployed systems are …
Total citations	Cited by 1777



Suggestions to Reviewers

- Be fair; you do not know it all (past, present, future)
- Be open-minded; you do not know it all (past, present, future)
- Be accepting of diverse research methods: there is no single way of doing research or writing papers
- Be constructive, not destructive; also accept more
- Do **not** have double standards… suppress your biases…

Do not block or delay progress for non-reasons

Suggestion to Community

We Need to Fix the Reviewer Accountability & Closed-Mindedness Problem

Takeaway

Main Memory Needs Intelligent Controllers

Research Community Needs

Accountable & Open-Minded Reviewers

Suggestions for Patching the System

- Accept papers liked a lot by some reviewers
- **Not** based on majority voting
- A majority agreement should **not** be necessary
- Avoid increasing negativity and bias
  - E.g., "Anti-champion" is a bad idea
  - We already have enough negativity in the system
- Litmus test: Our goal is to advance science efficiently
  - Not have "everyone" agree on acceptance

As Students and Producers of Science …

Can We Do Better?

Suggestion to Researchers: Principle: Passion

Follow Your Passion

(Do not get derailed

by naysayers

& toxic comments)

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: Work Hard

Work Hard to
Enable Your Passion

Principle: Good Mindset, Goals & Focus

You can make a good (great) impact on the world

And, you should not be derailed

Suggested Reading

Richard Hamming

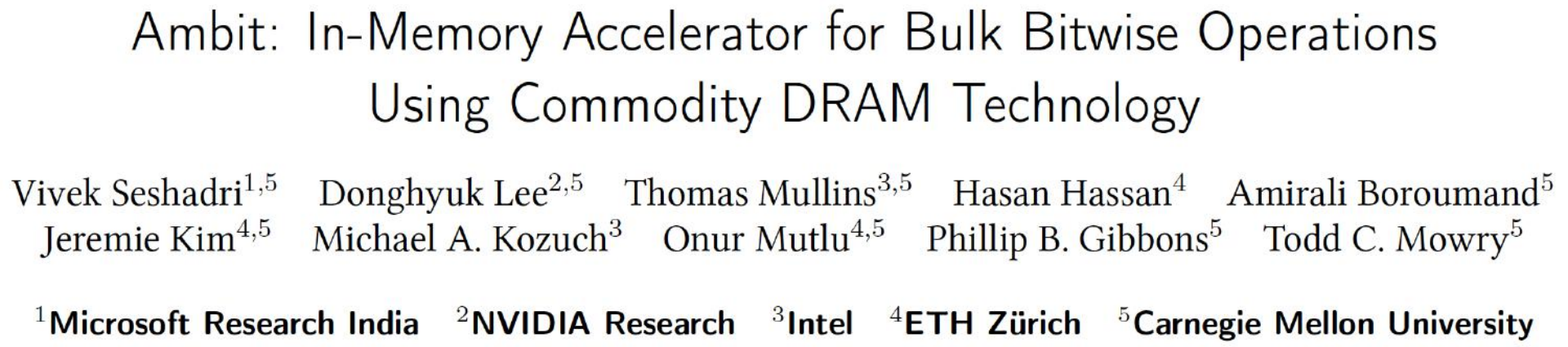
``You and Your Research''

Transcription of the
Bell Communications Research Colloquium Seminar
7 March 1986

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

Remember Ambit?

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
  "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology"
  *Proceedings of the 50th International Symposium on Microarchitecture (**MICRO**)*, Boston, MA, USA, October 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]


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

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

[1]Microsoft Research India [2]NVIDIA Research [3]Intel [4]ETH Zürich [5]Carnegie Mellon University

Remember Ambit?

Rejected 4 times:
2x ISCA, 1x MICRO, 1X HPCA

Proceedings of the [illegible]

Microar

[Slides (otx) (pdf)]

Ambi itions

Vivek Seshadri li Boroumand[5]
Jeremie Kim[4] d C. Mowry[5]

[1]Microsoft Re llon University

Acknowledgments

We thank the reviewers of ISCA 2016/2017, MICRO 2016/2017, and HPCA 2017 for their valuable comments. We thank the members of the SAFARI group and PDL for their feedback. We acknowledge the generous support of our industrial partners, especially Google, Huawei, Intel, Microsoft, Nvidia, Samsung, Seagate, and VMWare. This work was supported in part by NSF, SRC, and the Intel Science and Technology Center for Cloud Computing. A preliminary version of this work was published in IEEE CAL [99], which introduced bulk bitwise AND/OR in DRAM, and in ADCOM [96], which introduced the idea of *processing using memory*. An earlier pre-print of this paper was posted on arxiv.org [101].

Ambit

- First work on charge-sharing based bitwise ops in DRAM
  - Extends and completes our IEEE CAL 2015 paper
- **Disruptive** -- spans algorithms to circuits/devices
  - Requires hardware/software cooperation for adoption
- Led to a large amount of work in DRAM and NVM
  - The work continues to build
- Initially, it was dismissed by many reviewers
  - Rejected from 4 conferences!

ISCA 2016: Rejected

Buddy RAM: Fast and Efficient Bulk Bitwise Operations Using DRAM

Rejected 2006kB 23 Nov 2015 11:30:23pm EST ·
7f7234da178e644380275ce12a4f539ef45c4418

You are an **author** of this paper.

▶ **Abstract**

Many data structures (e.g., database bitmap indices) rely on fast bitwise operations on large bit vectors to achieve high performance. Unfortunately, the throughput of such bulk [more]

▶ **Authors**

V. Seshadri, D. Lee, T. Mullins, A. Boroumand, J. Kim, M. Kozuch, O. Mutlu, P. Gibbons, T. Mowry [details]

▶ **Topics and Options**

	RelISC	OveMerPos	RevConAnd	Nov	WriQua
Review #171A	3	4	4	2	3
Review #171B	2	4	3	3	4
Review #171C	3	4	4	2	3
Review #171D	3	5	2	2	3
Review #171E	2	3	2	3	3

MICRO 2016: Rejected

Submission (1662kB) 10 Apr 2016 9:32:31pm EDT ·
e518c6a8916109492574858db80a6184fe61ca0c

▶ **Abstract**

Certain widely-used data structures (e.g., bitmap indices) rely on

[more]

▼ **Authors**

Vivek Seshadri (CMU)
<vseshadr@cs.cmu.ed
Donghyuk Lee (NVIDIA Re
<donghyuk1@cmu.edu
Thomas Mullins (Intel)
<thomas.p.mullins@int
Amirali Boroumand (CMU)
Jeremie Kim (CMU)
Michael A. Kozuch (Intel)
<michael.a.kozuch@in
Onur Mutlu (CMU/ETH)
<omutlu@gmail.com>
Phillip B. Gibbons (CMU)
<gibbons@cs.cmu.edu
Todd C. Mowry (CMU) <tc

▶ **Topics**

Rejected · You are an **author** of this paper.

	PosRebOve	OveMer	RevExp	Nov	WriQua
Review #249A	2	2	4	3	3
Review #249B	4	4	3	3	5
Review #249C	2	3	4	2	3
Review #249D	5	5	2	3	3
Review #249E	5	5	2	2	3
Review #249F	3	3	3	3	4

HPCA 2017: Rejected

1)~significantly improves the performance of queries in applications that use bitmap indices for fast analytics, and 2)~makes bit vectors more attractive than red-black trees to represent sets. We believe Buddy can trigger programmers to redesign applications to use bitwise operations with the goal of achieving high performance and efficiency.

Rejected · You are an **author** of this paper.

	OveMer	RevExp	WriQua	ExpMet	Nov
Review #119A	1	2	3	2	2
Review #119B	4	1	4	4	3
Review #119C	4	4	4	4	4
Review #119D	3	1	4	4	3
Review #119E	3	2	5	4	4

1 Comment: *Response* (V. Seshadri)

ISCA 2017: Rejected

Rejected

Submission 19 Nov 2016 12:03:02am EST

3eea263e35e53552851cabc5225162776f809eaa

▶ **Abstract**

Bitwise operations are an important component of

[more]

▶ **Authors**

V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. Kozuch, O. Mutlu, P. Gibbons, T. Mowry [details]

▶ **Topics and Options**

	PosRebOve	OveMer	Nov	WriQua	RevExp
Review #162A	1	2	2	4	5
Review #162B	2	2	3	3	3
Review #162C	4	4	3	4	4
Review #162D	3	3	3	4	4
Review #162E	4	4	3	4	3

Ambit Sounds Good, No?

Review from ISCA 2016

Paper summary

The paper proposes to extend DRAM to include bulk, bit-wise logical
operations directly between rows within the DRAM.

Strengths

- Very clever/novel idea.

- Great potential speedup and efficiency gains.

Weaknesses

- Probably won't ever be built. Not practical to assume DRAM manufacturers with change DRAM in this way.

Very Interesting and Novel, ….. BUT …

Comments for the authors

I found this idea very interesting and novel. In particular, while there have been many works proposing moving computation closer to
memory, I'm not aware of any work which proposes to leverage the DRAM
rows themselves to implement the computation. The benefits to this
approach are large in that no actual logic is used to implement the
logical functions. Further the operation occurs in parallel across
the whole row, a huge degree of data parallelism.

… This Will Never Get Implemented

- The biggest problem with the work is that it underestimates the
  difficulty in modifying DRAM process for benefit in only a subset of
  applications which do bulk bitwise operations. In particular, I find
  it hard to believe that the commodity DRAM industry will incorporate
  this into their standard DRAM process. DRAM process is, at this
  point, a highly optimized, extremely tuned endeavor. Adding this
  kind of functionally will have a big impact on DRAM cost. The
  performance benefit on the subset of applications isn't enough to
  justify the higher costs this will incur and this will never get
  implemented.

Another Review

Another Review from ISCA 2016

Strengths

The proposed mechanisms effectively exploit the operation of the DRAM to perform efficient bitwise operations across entire rows of the DRAM.

Weaknesses

This requires a modification to the DRAM that will only help this type of bitwise operation. It seems unlikely that something like that will be adopted.

… This Will Never Get Implemented

Comments for the authors

This paper shows that DRAM could be modified to support bitwise operations directly within the DRAM itself. The performance advantages are compelling for situations in which bulk bitwise operations matter.

However, I am not really convinced that any DRAM manufacturer would really consider modifying the DRAM in this way. It beneifts one specific type of operation, and while that is important for some applications, it is not really a general-purpose operation. It is not like the STL library would be changed to use this for its implementation of sets.

Yet Another Review

Yet Another Review from ISCA 2016

Weaknesses

The core novelty of Buddy RAM is almost all circuits-related (by exploiting sense amps). I do not find architectural innovation even though the circuits technique benefits architecturally by mitigating memory bandwidth and relieving cache resources within a subarray. The only related part is the new ISA support for bitwise operations at DRAM side and its induced issue on cache coherence.

This paper suits better to be peer-reviewed and published in a circuit conference or with a fabricated chip in ISSCC.

A Review from HPCA 2017: REJECT

Review #119A

Paper summary

Paper proposes DRAM technology changes (inverts, etc) to implement bit-wise operations directly on DRAM rows.

Overall merit

1. Reject

Post-response overall merit

Unknown

Reviewer expertise

2. I have passing familiarity with this area

Writing quality

3. Adequate

Experimental methodology

2. Poor

Novelty

2. Incremental improvement

Strengths

Seems like a new idea. Processor-in-Memory (PIM) ideas have resurged.

Weaknesses

#119 - HPCA23

* Impractical. Too many implications on ISA, DRAM design, and coherence protocols.
* Unlikely to benefit real-world computations.
* Evaluation did not consider full-program performance.

Comments for author

I am skeptical this would benefit real-world computations. I've never seen real-world program profiles with hot functions or instructions that are bit-wise operations.

On the other hand, I *have* seen system profiles that show non-trivial time zeroing pages. Suggest re-tooling your work to support page zeroing and evaluating that with a full-system simulation. Take a look at when/why the Linux kernel zeroes pages. You might be surprised at the possible impact.

A Review from ISCA 2017

Review #162A Updated 28 Jan 2017 5:16:50am EST

Plain text

Post rebuttal overall merit
1. Reject

Overall merit
2. Weak reject

Novelty
2. Incremental improvement

Writing quality
4. Well-written

Reviewer expertise
5. This is my area

Paper summary
This paper proposes in-DRAM bit-wise operations by activating more than one word lines (and cells connected to the wordiness). Basically, it's a charge-based computation where the difference in charge stored cells connected to the same bit line is used for the logic operation.

Strengths
- conceptually a very interesting proposal (but practically not sure).
- consider various aspects including the interaction between processors and RAM (although there isn't any new contribution and rather use the same proposal as prior work).

Weaknesses
- negative impact on the regularity of DRAM array design (and associated overhead evaluation seems to be very weak.
- significantly increase the testing cost

Comments to authors
This is an interesting proposal and well presented paper. However, I have some concerns regarding the evaluation (especially related to circuit level issues).

Especially, I feel that the variation related modeling and evaluation are weak as there are multiple sources of variations such as access transistors and sense-amp mismatches, minor defects in either access transistors and/or capacitor that can manifest in this particular proposed operation scenarios. That is, the authors oversimplify the variation modeling, which I believe failed to convince me this will work in practice. Also, the area overhead analysis sounds hand-waivy. I totally understand the difficulty of DRAM overhead analysis but also we must pursue more precise ways of evaluating the area impact as DRAM is very cost-sensitive.

Another Review from ISCA 2017

Review #162B Updated 1 Feb 2017 6:50:31pm EST

Plain text

Post rebuttal overall merit
2. Weak reject

Overall merit
2. Weak reject

Novelty
3. New contribution

Writing quality
3. Adequate

Reviewer expertise
3. I know the material, but am not an expert

Paper summary
This paper proposes performing bulk bit-wise operations at DRAM. They leverage analog operation of DRAM, and add some extra

circuits to do bit-wise operations at row granularity.

Strengths
The idea of handling bit-wise operations in memory is interesting.

Weaknesses
Not motivated well.
Not convinced the possible gains worth all the complexity.
Not convinced if the proposal is applicable in real world applications that do bit-wise operations on different data granularity.

Comments to authors
* The paper lacks motivation. The authors talk about how common bit-wise operations are. However, they do not provide any stat on how often these operations are being used, and more importantly, on what data granularity.

* Although bit-wise operations are common in some applications, they are not necessarily done at large granularity. For example, many applications do bit-wise operations at small 64-byte (or even smaller) entities. For such cases, this paper requires copying two whole rows to some temporary rows, and doing the operation on those rows. Please explain how you handle such cases, and what the benefits would be.

* What happens if the user does bit-wise operation on two 8-byte data, and want to store it in a third block?

* What happens if both operands are located in one row?

* The main issue with this work is that it requires flushing blocks out of caches to do the bit-wise operations. Imagine you have blocks A and B in the cache, as discussed in section 6.2.3., the proposal would flush them out of caches (not sure how?), writes

ISCA 2017 Summary: Nitpicks

@A1 6 Mar 2017

This paper was discussed both online and at the PC meeting. Reviewers were uniformly positive about the novelty of the proposed Buddy-RAM design. However, reviewers were also concerned about the feasibility of the design. During the post-rebuttal and PC discussion, the main concerns raised were (1) the impact of process variation on the design's functional correctness;

#162 - ISCA 2017

(2) the potential reliability issues that arise due to the lack of ECC/CRC mechanisms; and (3) the impact on DRAM testing cost.

Specifically on point (1), some reviewers raised concerns about the limitations of the simulations performed to address variability: "Monte-Carlo cannot capture tail distribution of cell failures. Also Monte-Carlo cannot capture random correlated WID process variation issues (only some random uncorrelated variations)."

Given these concerns, the PC ultimately decided to reject the paper. We hope that this feedback is useful in preparing a future version of the paper.

The Reviewer Accountability Problem

Acknowledgments

We thank the reviewers of ISCA 2016/2017, MICRO 2016/2017, and HPCA 2017 for their valuable comments. We

MICRO 2017: Accepted

Accepted

Submission (1837kB) 4 Apr 2017 11:33:57pm EDT ·
7420f9f02c549bcca0dc6216a5e9887dffe0d422

Revision (1852kB) 14 Jun 2017 4:16am EDT ·
22f0d123a22cf04960928e0ac43d972b5a33a848

▼Abstract

Many important applications trigger bitwise operations on large bit vectors (bulk bitwise operations). In fact, recent

▶Authors

V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. Kozuch, O. Mutlu, P. Gibbons, T. Mowry [details]

	PosResOve	OveMer	RevExp	Nov	PotImp	WriQua	ImpRev
Review #347A	4	3	5	4	3	4	2
Review #347B	3	4	5	4	3	4	3
Review #347C		2	5	3	2	4	4
Review #347D	3	4	4	4	4	4	2
Review #347E	3	3	4	3	3	3	4
Review #347F	3	2	4	3	3	4	1

1 Comment: *Rebuttal Response (V. Seshadri)*

Some Questions to Ponder

- Do all research ideas need to be commercially viable?
  - If yes, when? Immediately? How can one predict the future?
- Can a "non-viable" idea become viable?
  - Or inspire something else that could be useful?
- Does a paper have to solve all problems with an idea?
  - Or can it do enough to enable fast & efficient progress?
- Are all reviews bias free and hidden/visible-COI free?
  - Or should we simply accept more works to enable fast progress?
- What do we lose from being more positive and accepting?

Suggestion: Litmus Test

Our Litmus Test Should be to Efficiently Advance Scientific Endeavor

Ask: Is my review scientific? Does it rely on a crystal ball?

Suggestion to Researchers: Principle: Resilience

Be Resilient

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

Principle: Learning and Scholarship

Focus on

learning and scholarship

Suggestion: Litmus Test

Our Litmus Test Should be to Efficiently Advance Scientific Endeavor

Ask: Is my review scientific? Does it rely on a crystal ball?