

# Memory-Centric Computing

## Recent Advances in Processing-in-DRAM

Onur Mutlu

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

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

1 March 2025

Memory-Centric Computing Tutorial @ PPOPP

**SAFARI**

**ETH** zürich

Computing

is Bottlenecked by Data

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

---

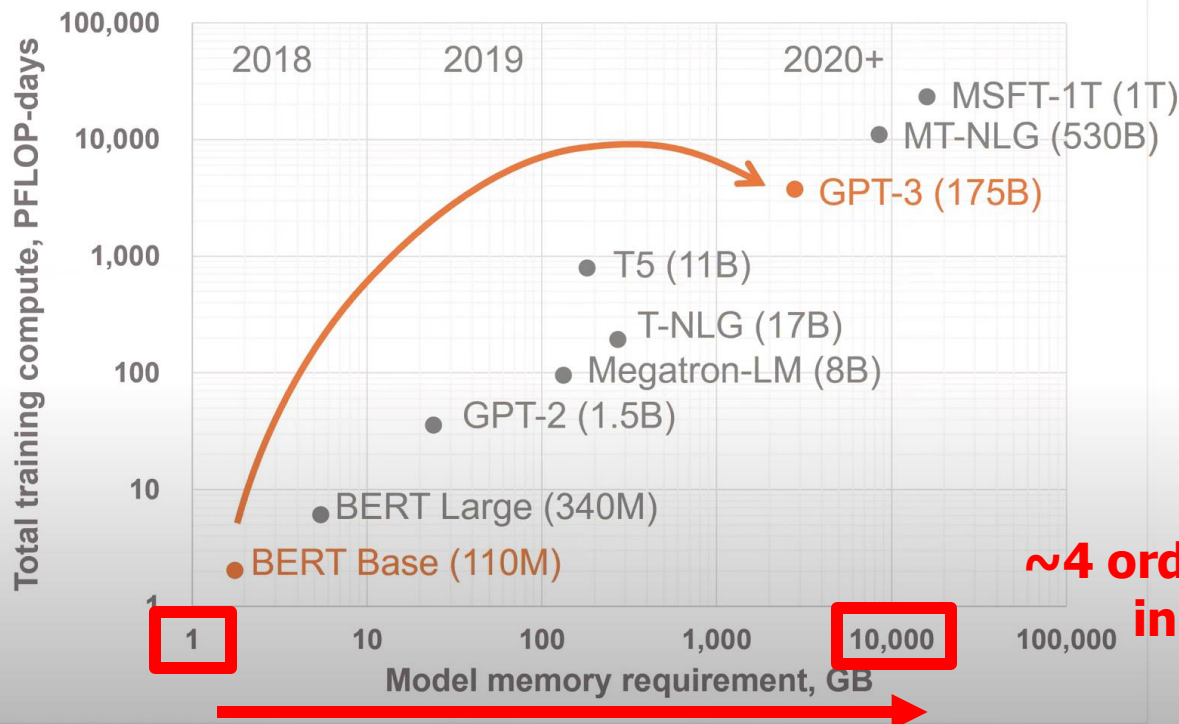
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
  - We can generate more than we can process
  - We need to perform more sophisticated analyses on more data

# Huge Demand for Performance & Efficiency

## Exponential Growth of Neural Networks



Memory and compute requirements

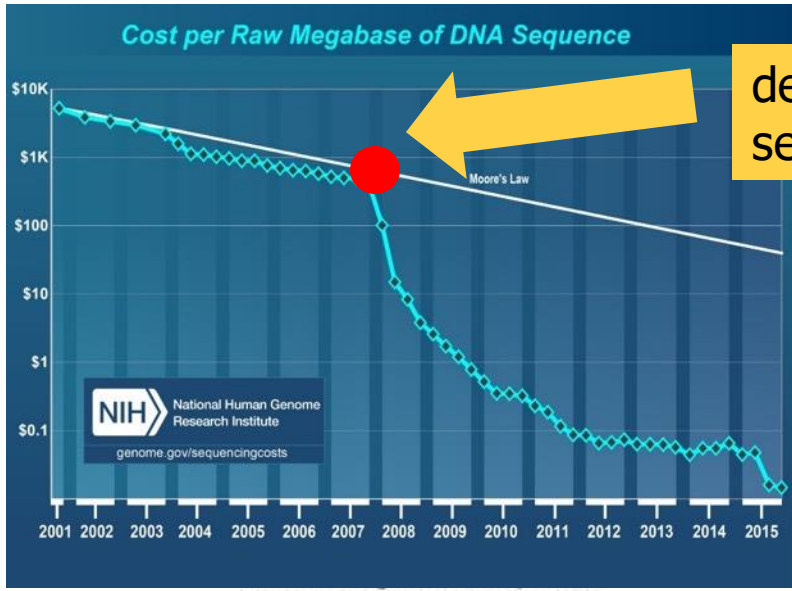


**1800x more compute**  
In just 2 years

Tomorrow, **multi-trillion** parameter models

**~4 orders of magnitude increase**  
in memory requirement  
in just a few years!

# Huge Demand for Performance & Efficiency

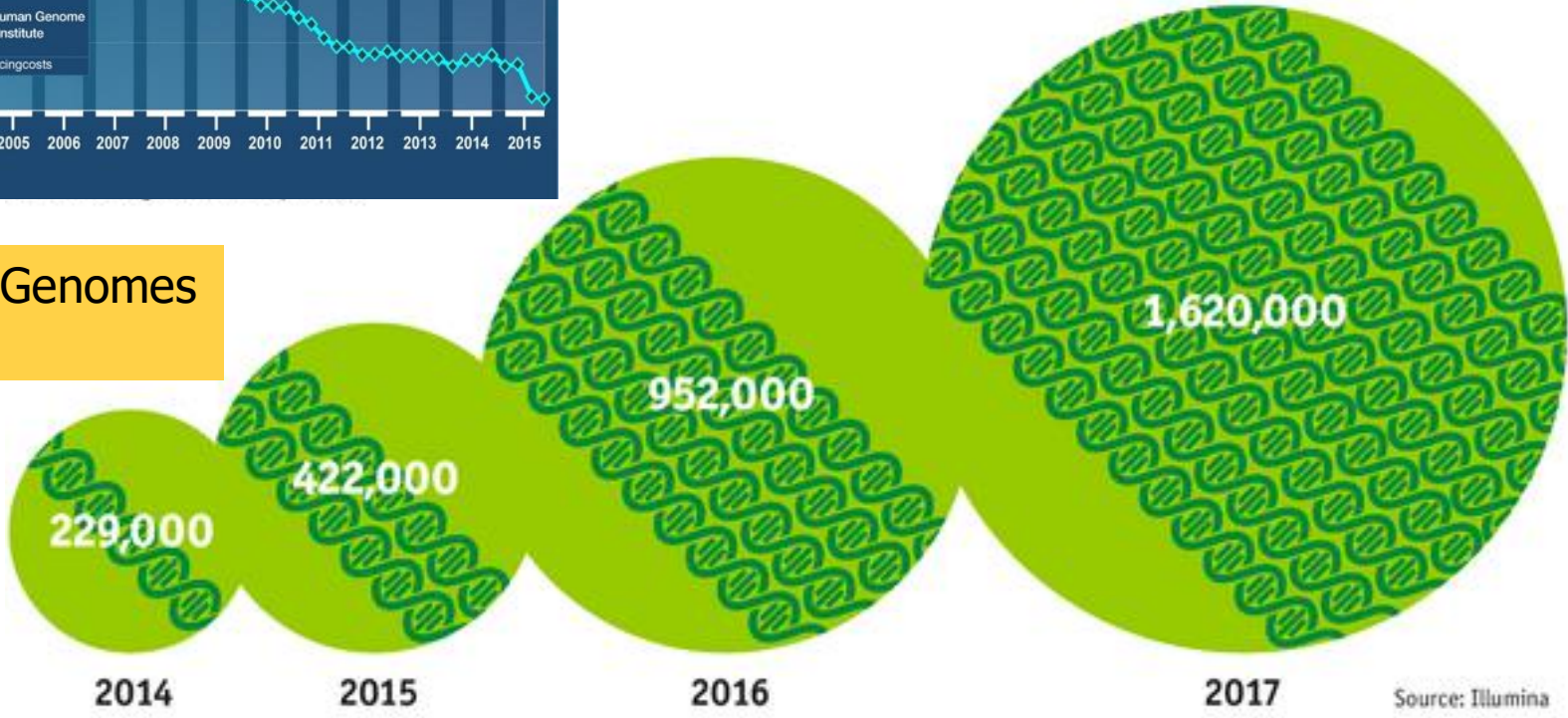


development of new sequencing technologies



Oxford Nanopore MinION

Number of Genomes Sequenced



The Economist

Source: Illumina

# The Problem

---

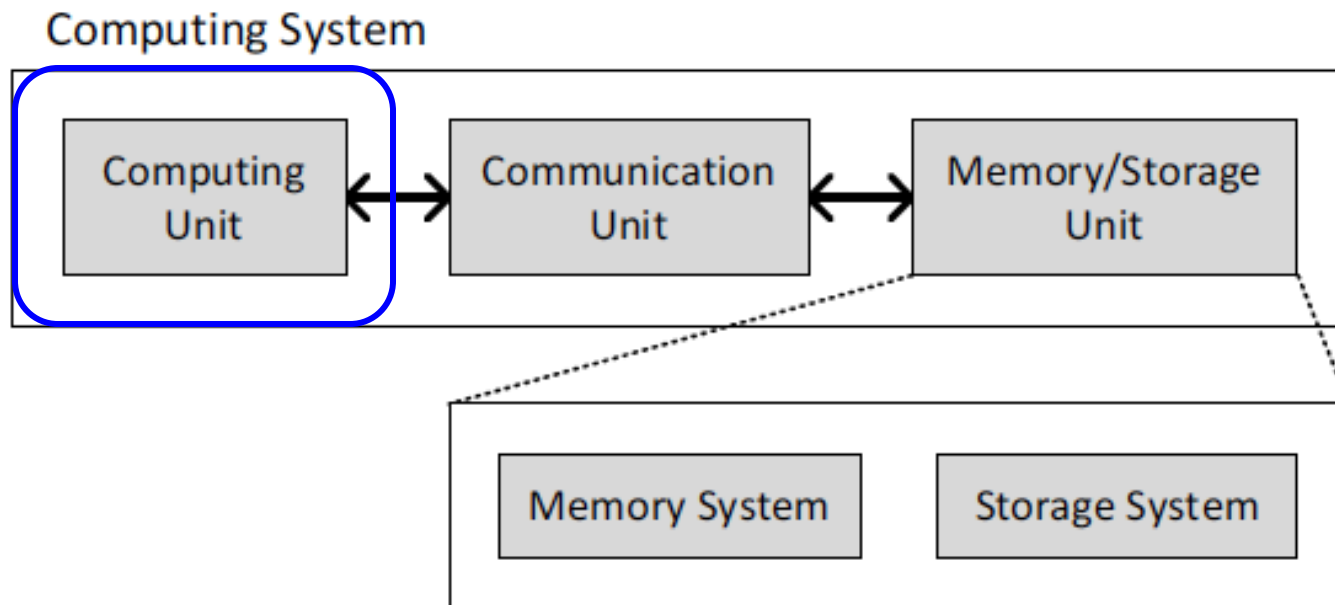
Data access is the major performance and energy bottleneck

Our current  
design principles  
cause great energy waste  
(and great performance loss)

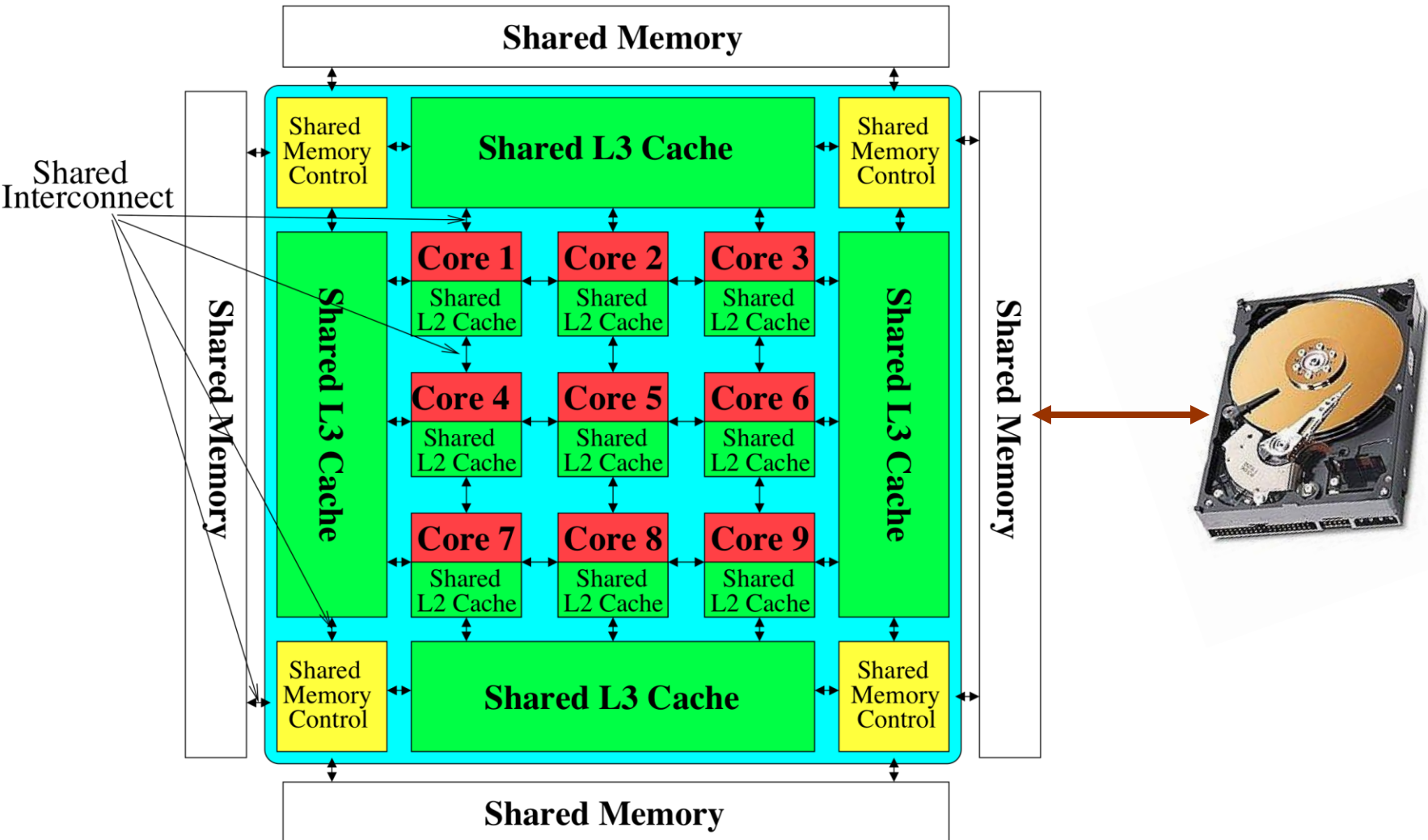
# Today's Computing Systems

---

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



# Perils of Processor-Centric Design



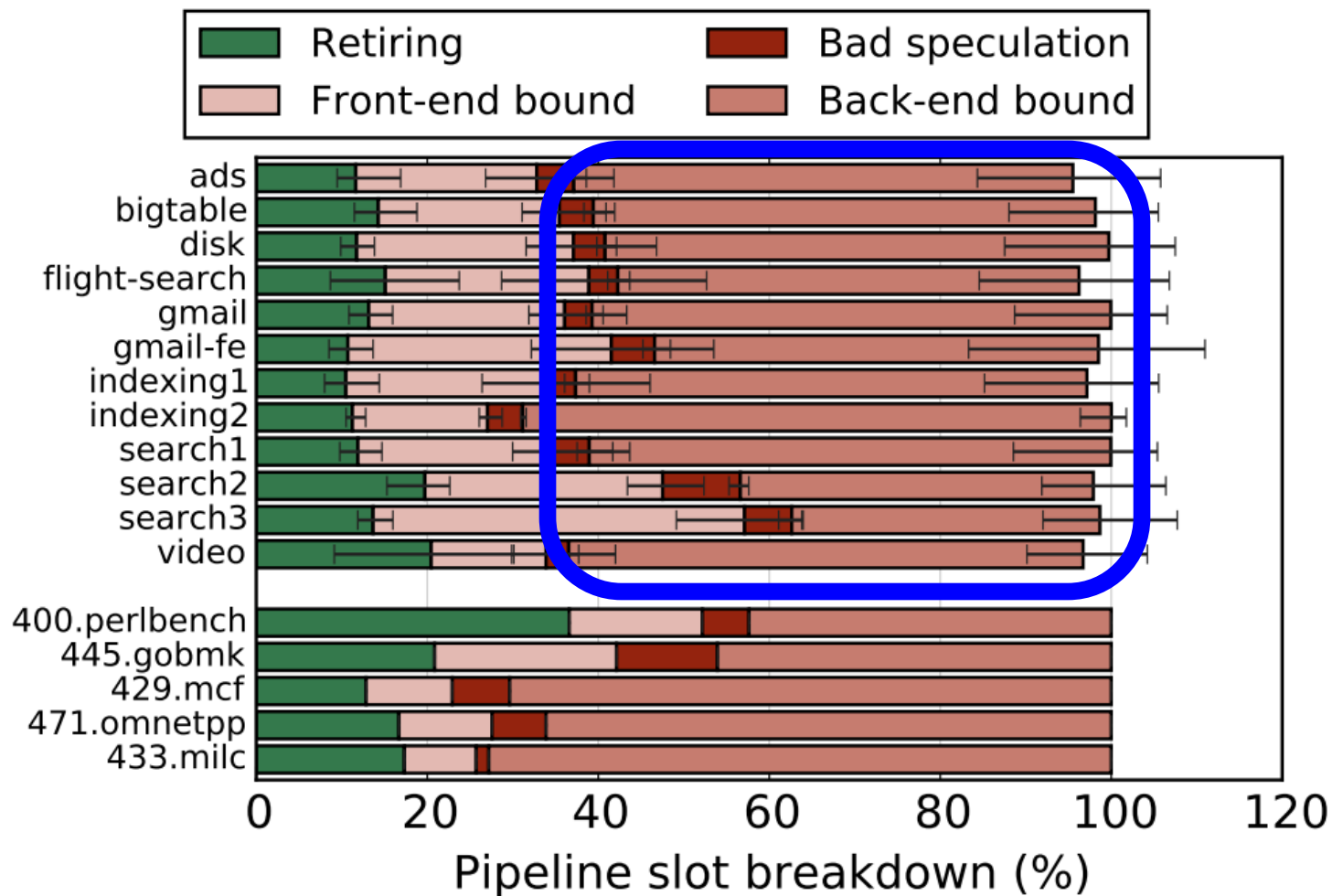
**Most of the system is dedicated to storing and moving data**

**Yet, system is still bottlenecked by memory**



# Processor-Centric System Performance

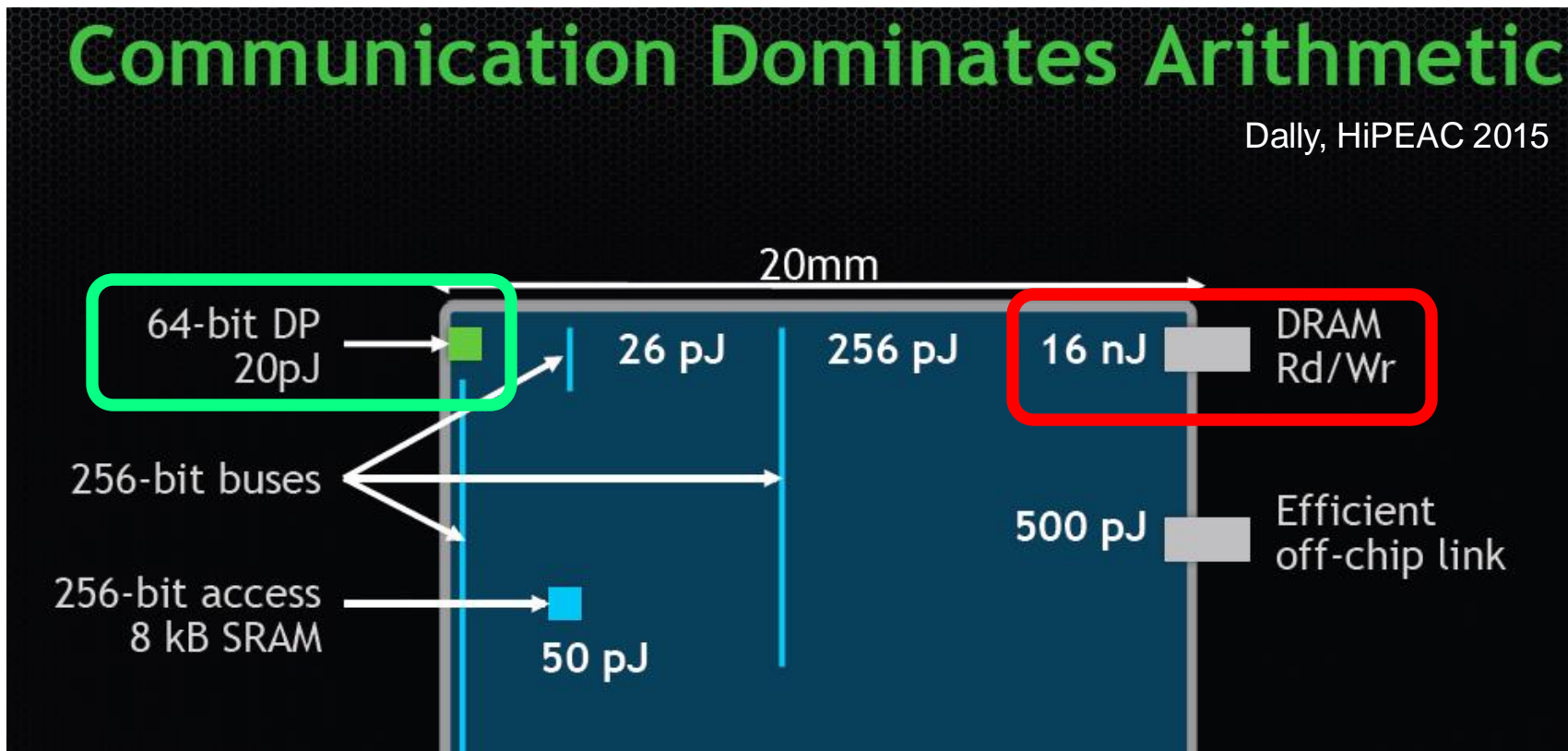
- All of Google's Data Center Workloads (2015):



# Data Movement vs. Computation Energy

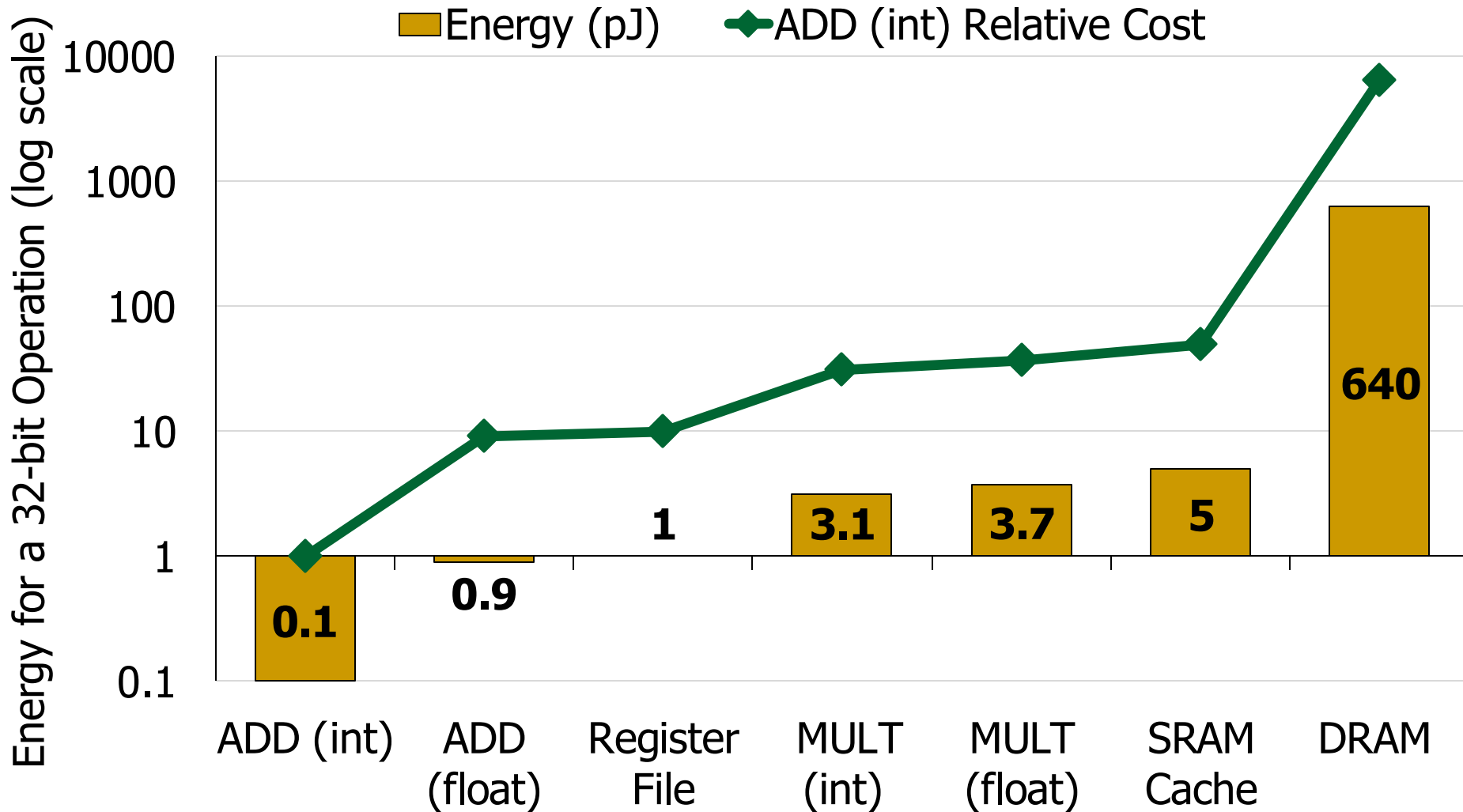
## Communication Dominates Arithmetic

Dally, HiPEAC 2015

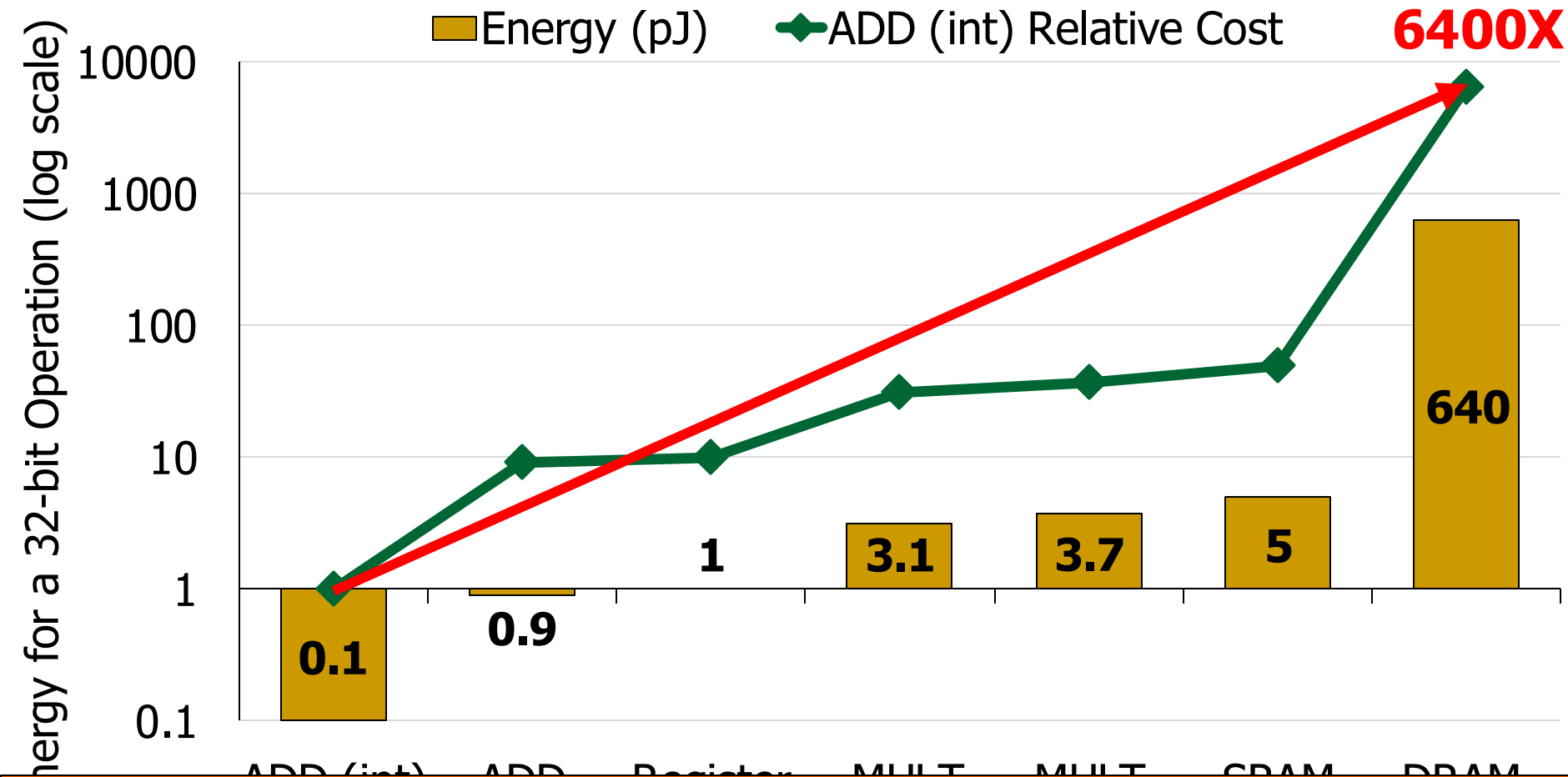


A memory access consumes  $\sim 100-1000X$  the energy of a complex addition

# Data Movement vs. Computation Energy



# 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<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

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

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

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

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

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

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

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

<sup>†</sup>Carnegie Mellon Univ.

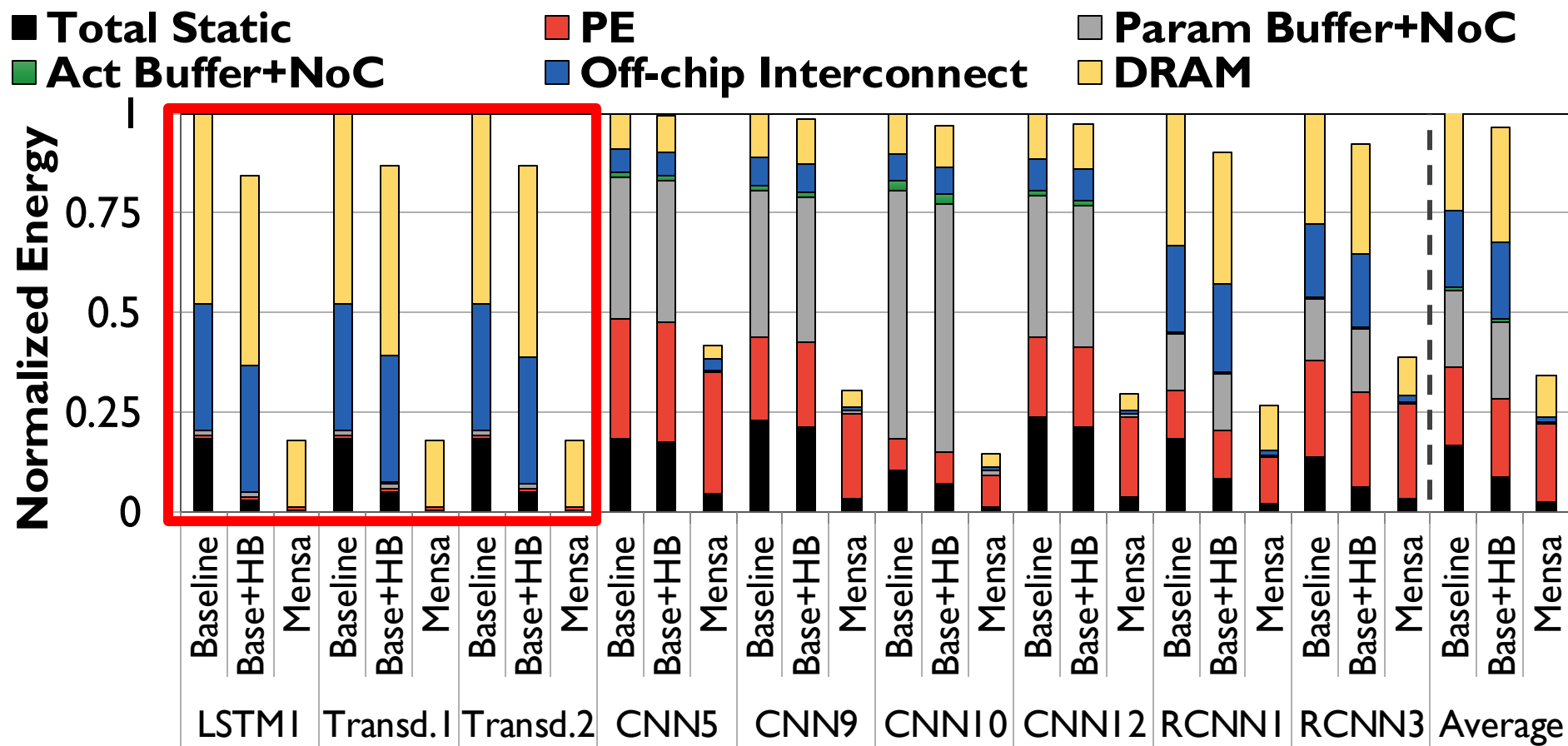
<sup>◇</sup>Stanford Univ.

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

<sup>§</sup>Google

<sup>\*</sup>ETH Zürich

# Energy Wasted on Data Movement



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

# Fundamental Problem

---

Processing of data  
is performed  
far away from the data



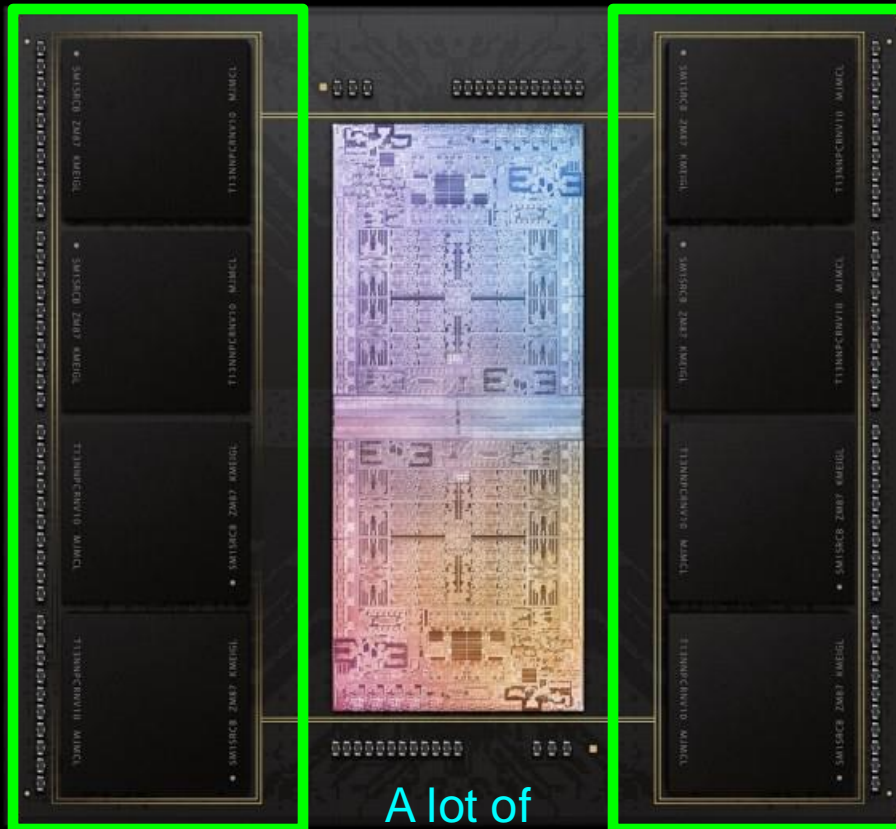
# We Need A Paradigm Shift To ...

---

- 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

Sensors



A lot of  
SRAM

Storage

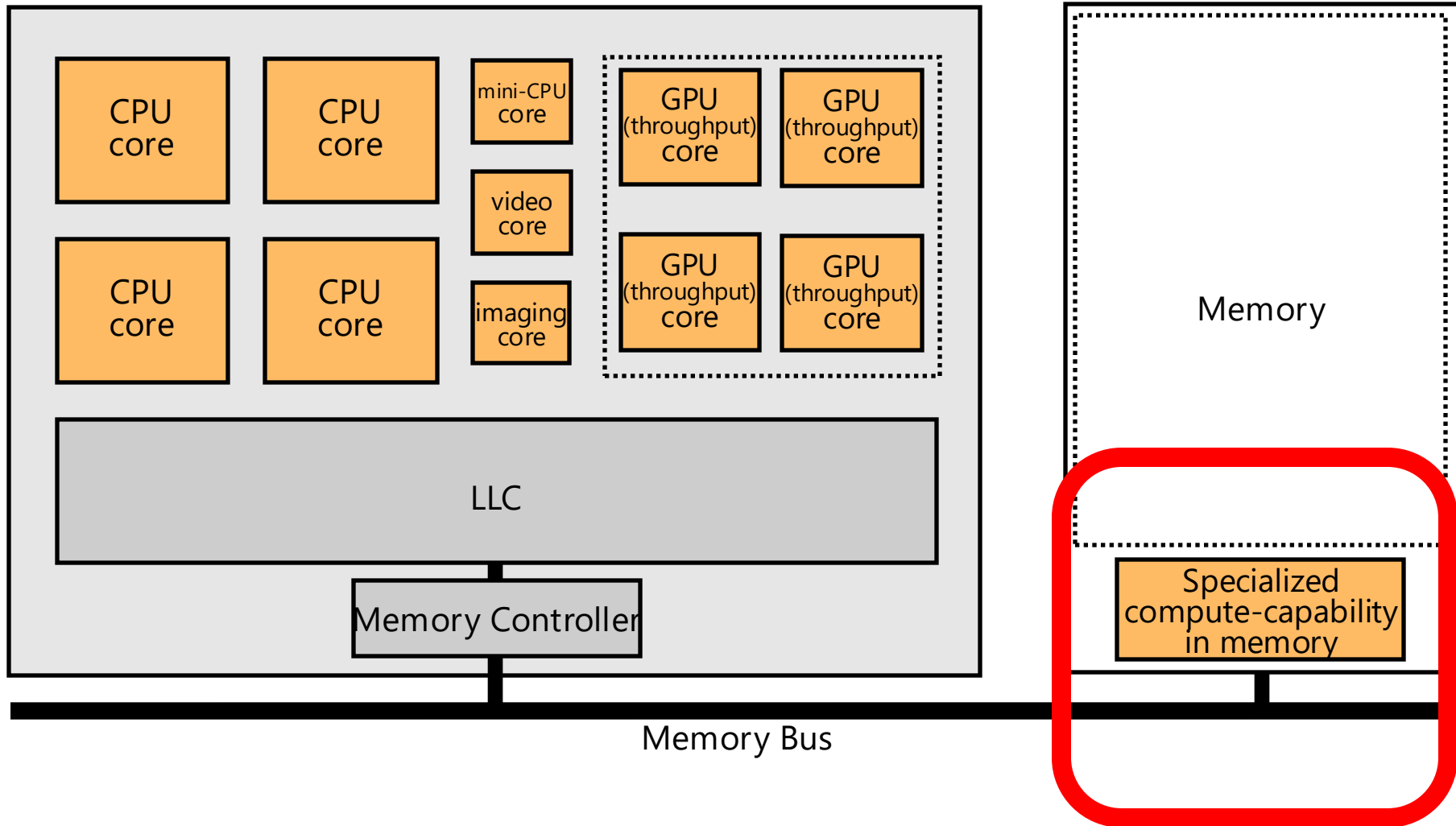
DRAM

DRAM

Storage

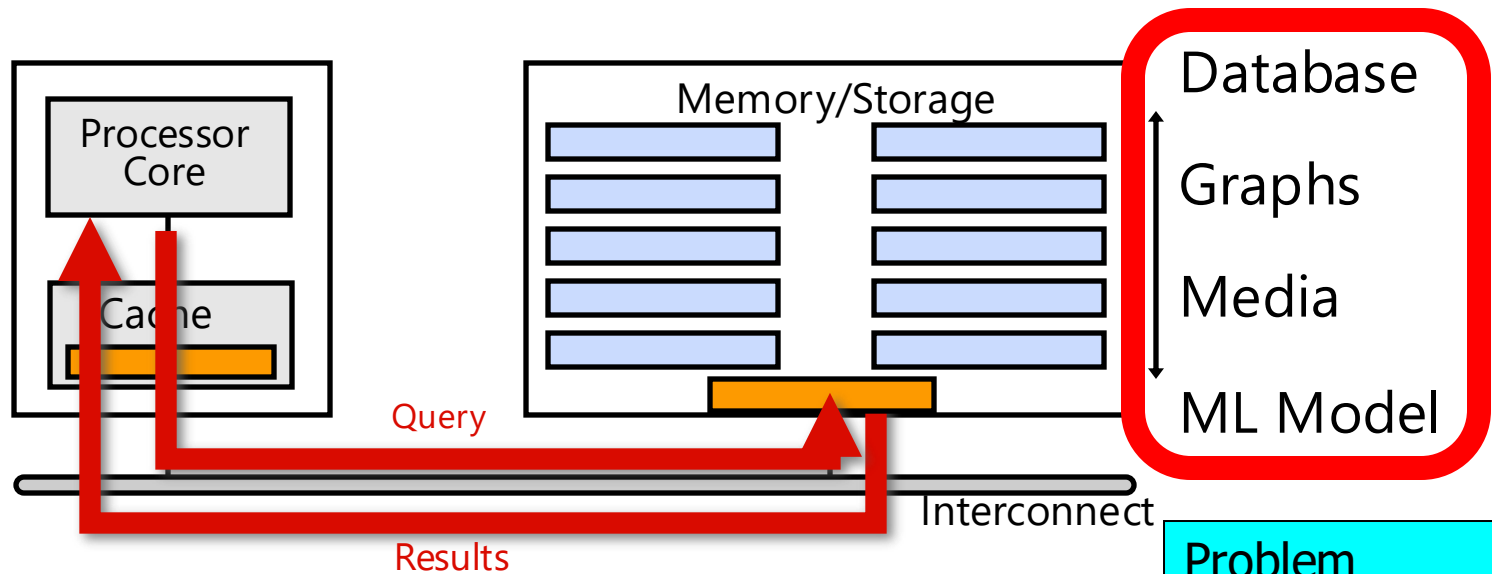
Apple M1 Ultra System (2022)

# 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?

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Logic
Devices
Electrons

# 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

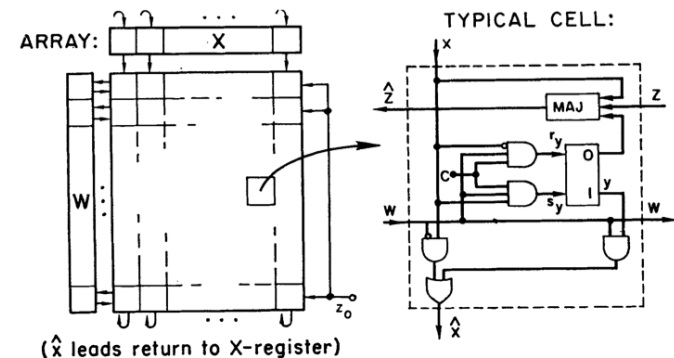
## Cellular Logic-in-Memory Arrays

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.



$$\begin{aligned}\hat{x} &= \bar{w}x + wy \\ s_y &= wcx, r_y = wc\bar{x} \\ \hat{z} &= M(x, \bar{y}, z) = x\bar{y} + z(x + \bar{y})\end{aligned}$$

Fig. 1. Cellular sorting array I.

# 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**

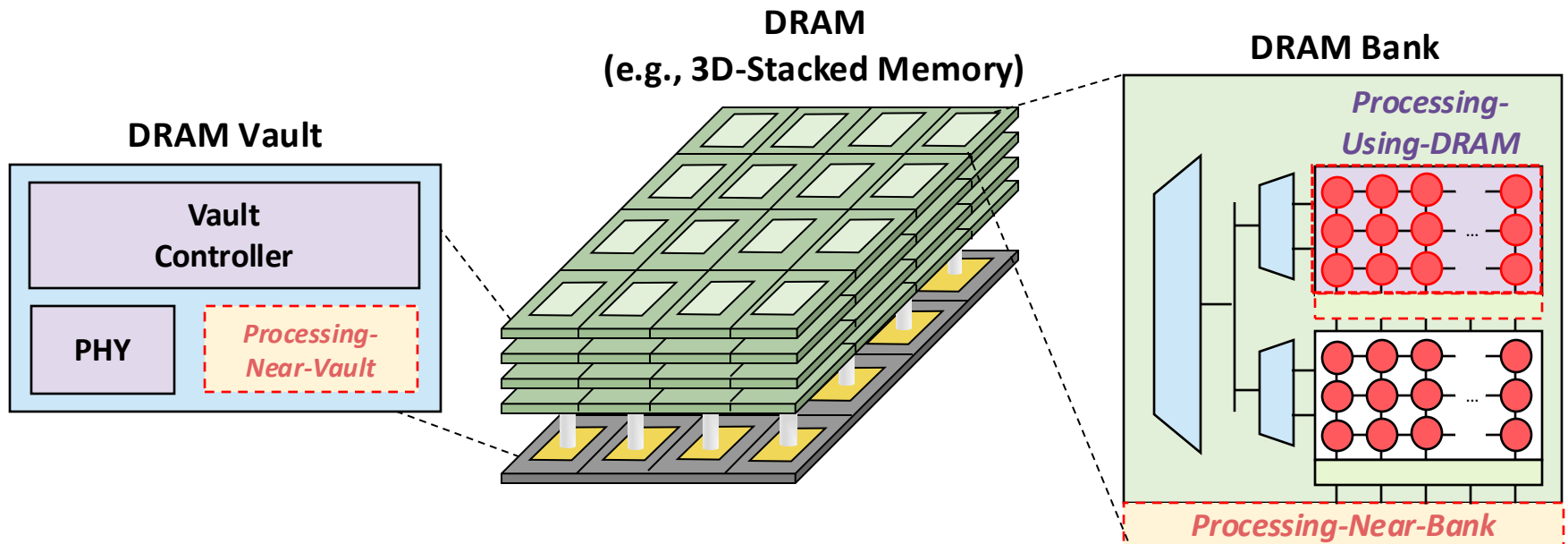
# Processing in Memory: Two Types

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

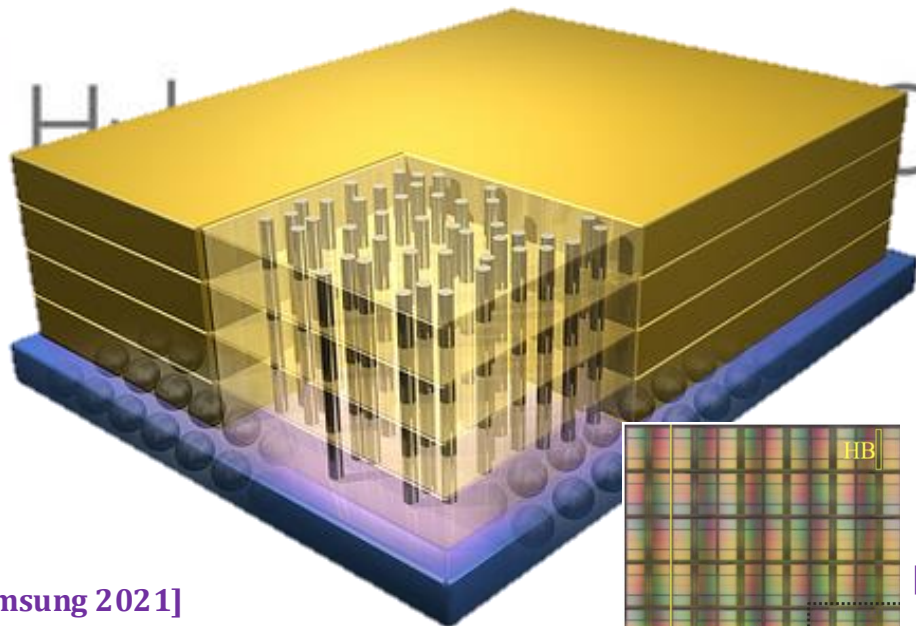


# Processing-in-Memory: Two Types

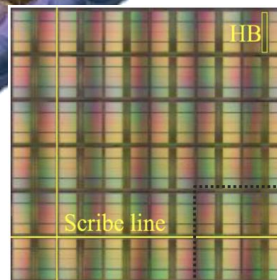
- 1 Processing-Near-Memory:** Computation logic is added to the same die as memory or to the logic layer of 3D-stacked memory
- 2 Processing-Using-Memory:** uses the operational principles of memory cells & circuitry to perform computation



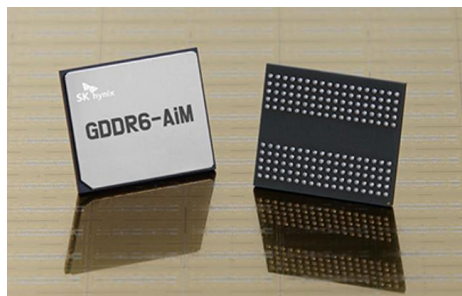
# Processing-in-Memory Landscape Today



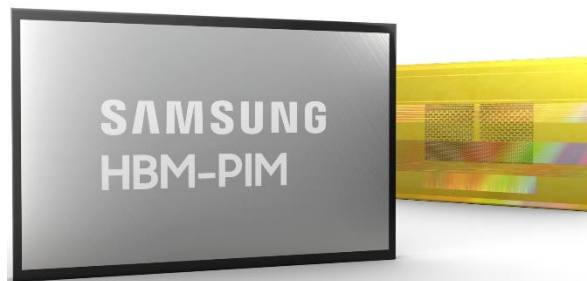
[Samsung 2021]



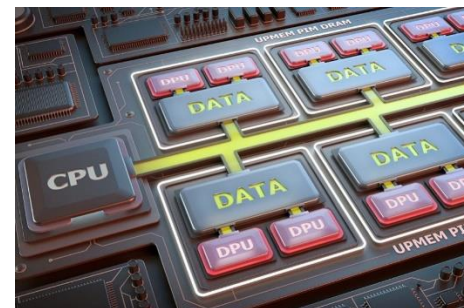
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

# 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 <sup>ID</sup>, Soohong Ahn <sup>ID</sup>, Taeyoung Ahn <sup>ID</sup>,  
Seungyong Lee <sup>ID</sup>, Myunghyun Rhee, Jooyoung Kim <sup>ID</sup>,  
Kwangsik Shin, Donguk Moon <sup>ID</sup>,  
Euseok Kim, and Kyoung Park <sup>ID</sup>

**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  $1.9\times$  better performance/power efficiency than the existing CPU system.

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

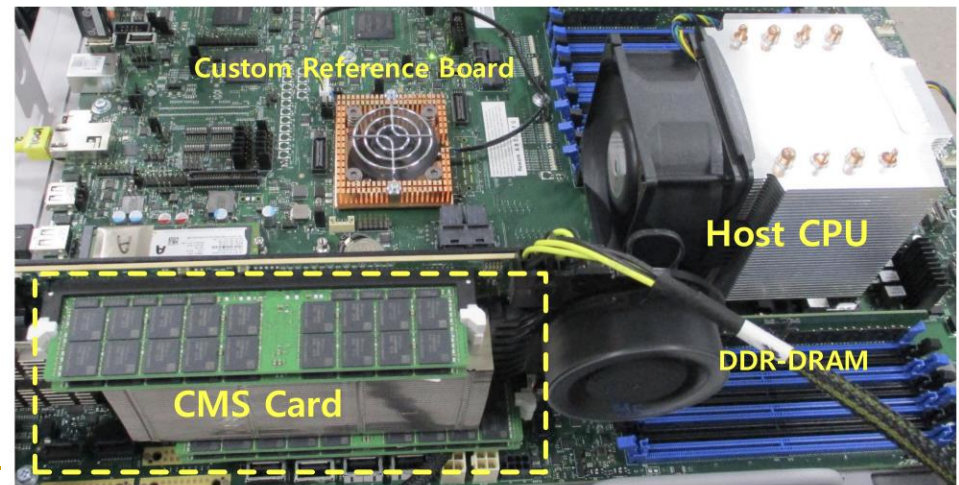


Fig. 6. FPGA prototype of proposed CMS card.

# Processing-in-Memory Landscape Today

## Samsung Processing in Memory Technology at Hot Chips 2023

By Patrick Kennedy - August 28, 2023



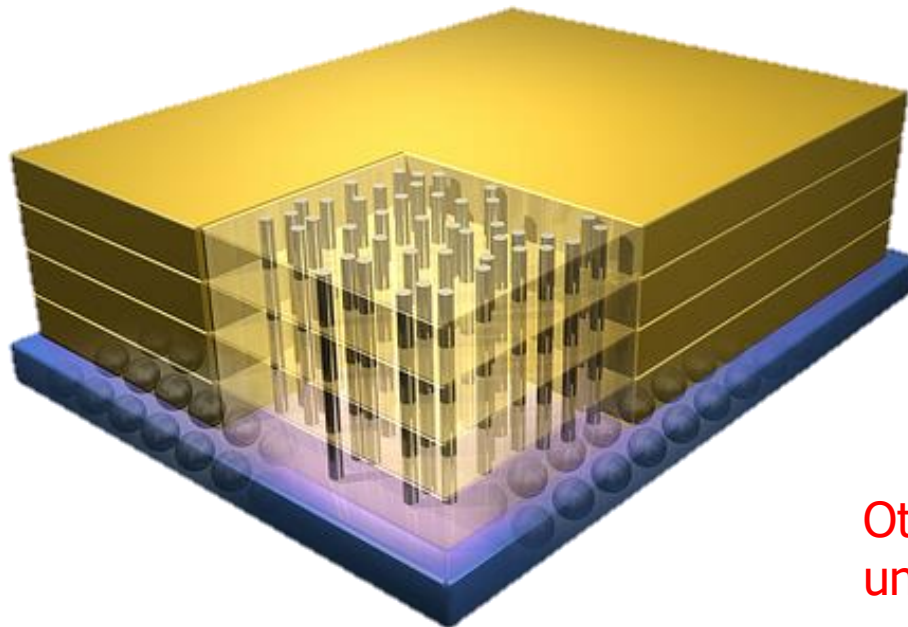
Samsung PIM PNM For Transformer Based AI HC35\_Page\_24

# Opportunity: 3D-Stacked Logic+Memory

---



Hybrid Memory Cube  
C O N S O R T I U M



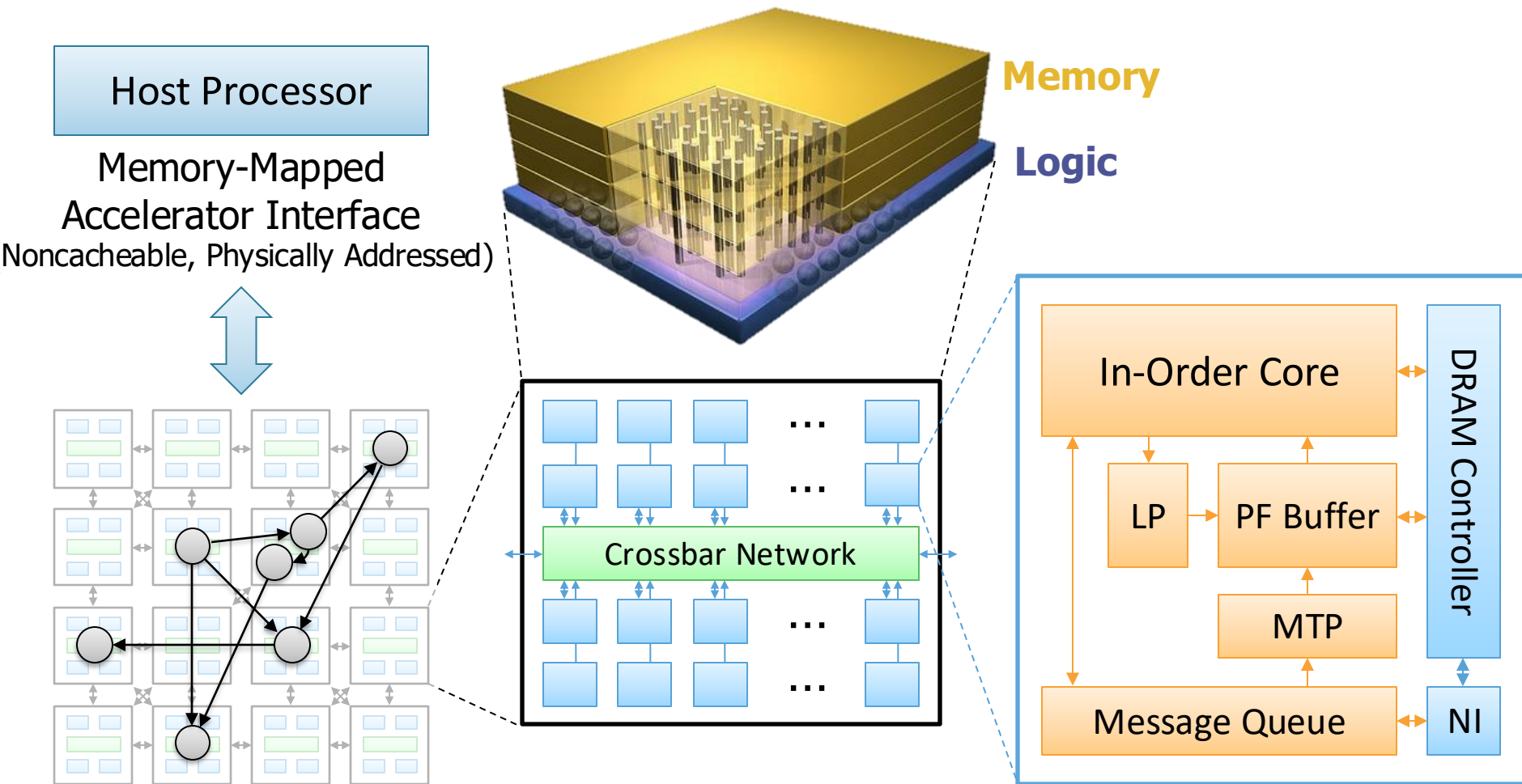
**Memory**

**Logic**

Other "True 3D" technologies  
under development

# Tesseract System for Graph Processing

Interconnected set of 3D-stacked memory+logic chips with simple cores



# 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<sup>§</sup>   Sungjoo Yoo   Onur Mutlu<sup>†</sup>   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

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

<sup>1</sup>ETH Zurich, Switzerland  
Thailand

<sup>2</sup>IIT Tirupati, India

<sup>3</sup>King Mongkut's University of Technology North Bangkok,

<sup>4</sup>Technical University of Ostrava, Czech Republic

<sup>5</sup>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<sup>†◇</sup>

Geraldo F. Oliveira<sup>\*</sup>

Saugata Ghose<sup>‡</sup>

Xiaoyu Ma<sup>§</sup>

Berkin Akin<sup>§</sup>

Eric Shiu<sup>§</sup>

Ravi Narayanaswami<sup>§</sup>

Onur Mutlu<sup>\*†</sup>

<sup>†</sup>*Carnegie Mellon Univ.*

<sup>◇</sup>*Stanford Univ.*

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

<sup>§</sup>*Google*

<sup>\*</sup>*ETH Zürich*

# 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)]

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

Amirali Boroumand<sup>1</sup>

Saugata Ghose<sup>1</sup>

Youngsok Kim<sup>2</sup>

Rachata Ausavarungnirun<sup>1</sup>

Eric Shiu<sup>3</sup>

Rahul Thakur<sup>3</sup>

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

Aki Kuusela<sup>3</sup>

Allan Knies<sup>3</sup>

Parthasarathy Ranganathan<sup>3</sup>

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

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

# 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<sup>1</sup> Jisung Park<sup>1</sup> Harun Mustafa<sup>1</sup> Jeremie Kim<sup>1</sup> Ataberk Olgun<sup>1</sup>  
Arvid Gollwitzer<sup>1</sup> Damla Senol Cali<sup>2</sup> Can Firtina<sup>1</sup> Haiyu Mao<sup>1</sup> Nour Almadhoun Alserr<sup>1</sup>  
Rachata Ausavarungnirun<sup>3</sup> Nandita Vijaykumar<sup>4</sup> Mohammed Alser<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich <sup>2</sup>Bionano Genomics <sup>3</sup>KMUTNB <sup>4</sup>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<sup>1</sup> Mohammad Sadrosadati<sup>1</sup> Harun Mustafa<sup>1</sup> Arvid Gollwitzer<sup>1</sup>  
Can Firtina<sup>1</sup> Julien Eudine<sup>1</sup> Haiyu Mao<sup>1</sup> Joël Lindegger<sup>1</sup> Meryem Banu Cavlak<sup>1</sup>  
Mohammed Alser<sup>1</sup> Jisung Park<sup>2</sup> Onur Mutlu<sup>1</sup>  
<sup>1</sup>ETH Zürich <sup>2</sup>POSTECH

# Many More Examples ...

---

## A Modern Primer on Processing-In-Memory

Onur Mutlu<sup>a</sup>, Saugata Ghose<sup>b</sup>, Juan Gómez-Luna<sup>c</sup>, Rachata Ausavarungnirun<sup>d</sup>,  
Mohammad Sadrosadati<sup>a</sup>, Geraldo F. Oliveira<sup>a</sup>

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

<sup>b</sup>*University of Illinois Urbana-Champaign*

<sup>c</sup>*NVIDIA Research*

<sup>d</sup>*MangoBoost Inc.*

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

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

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann**, Springer, 2022.*

# 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.

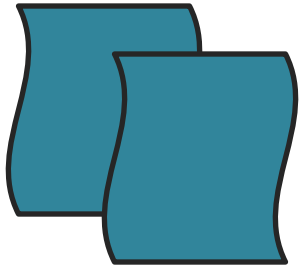
Hajinazar+, "SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM," ASPLOS 2021.

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.

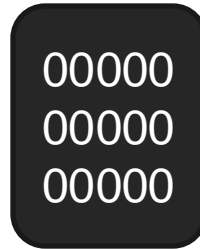


# Starting Simple: Data Copy and Initialization

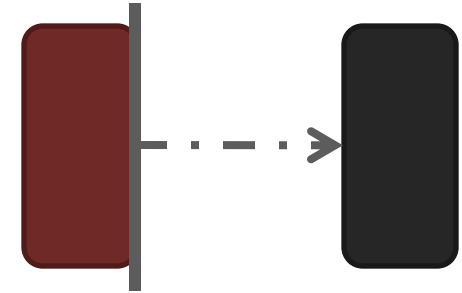
*memmove & memcpy: 5% cycles in Google's datacenter [Kanev+ ISCA'15]*



**Forking**



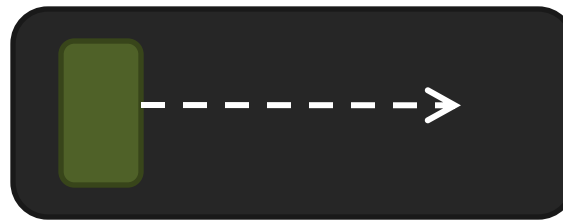
**Zero initialization  
(e.g., security)**



**Checkpointing**



**VM Cloning  
Deduplication**



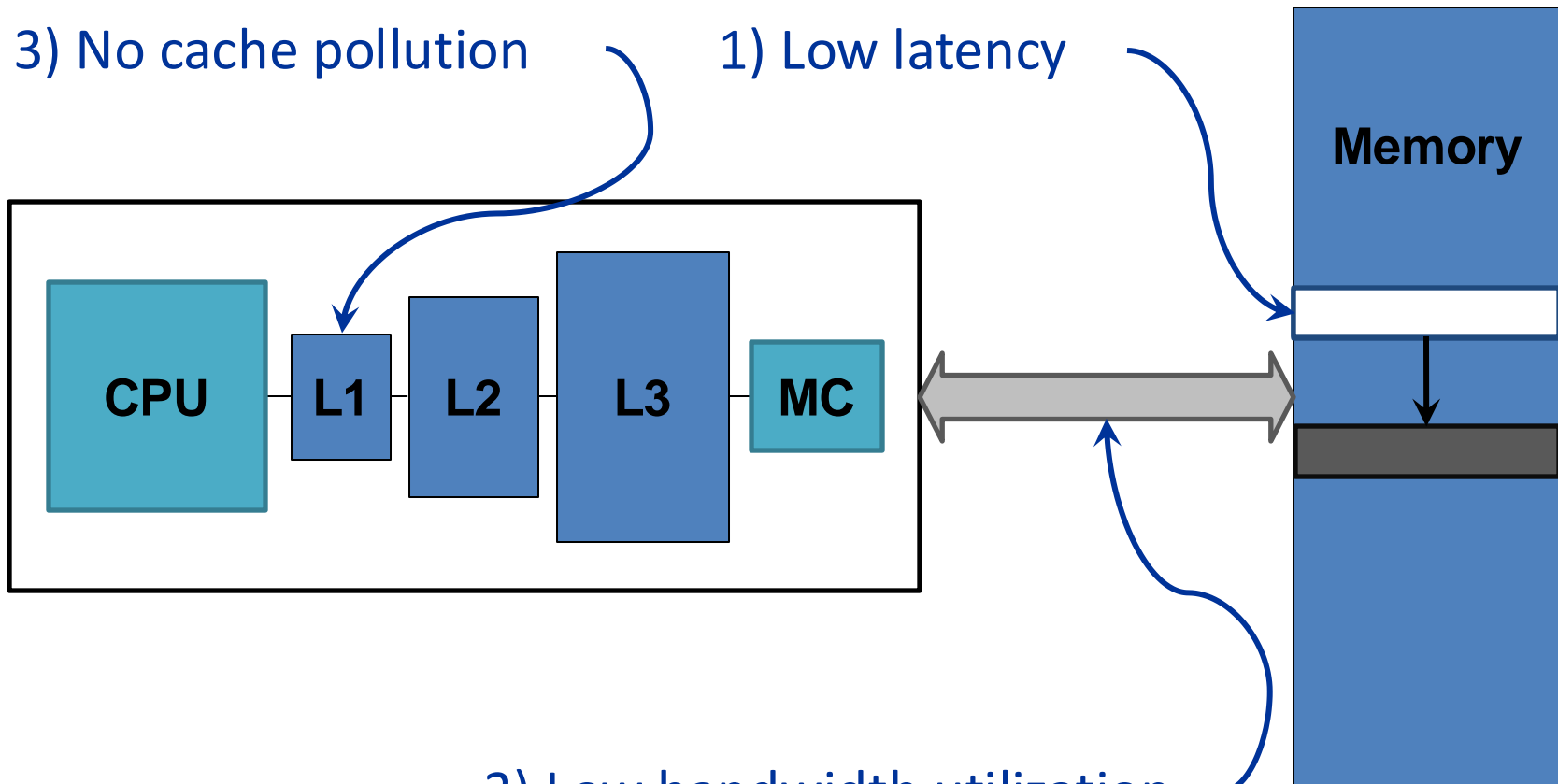
**Page Migration**

•••  
**Many more**

# Future Systems: In-Memory Copy

3) No cache pollution

1) Low latency



2) Low bandwidth utilization

4) No unwanted data movement

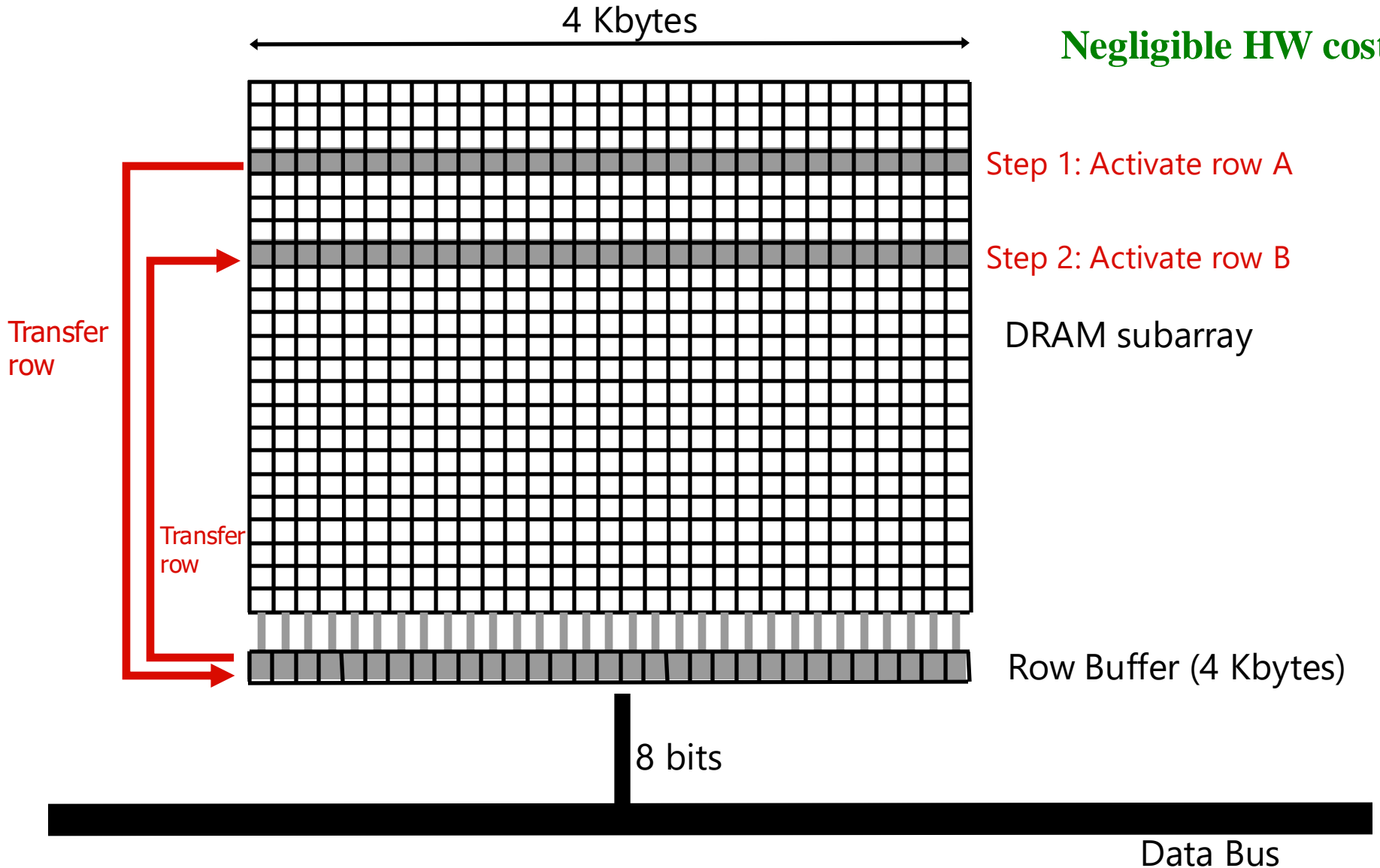
1046ns, 3.6uJ

→ 90ns, 0.04uJ

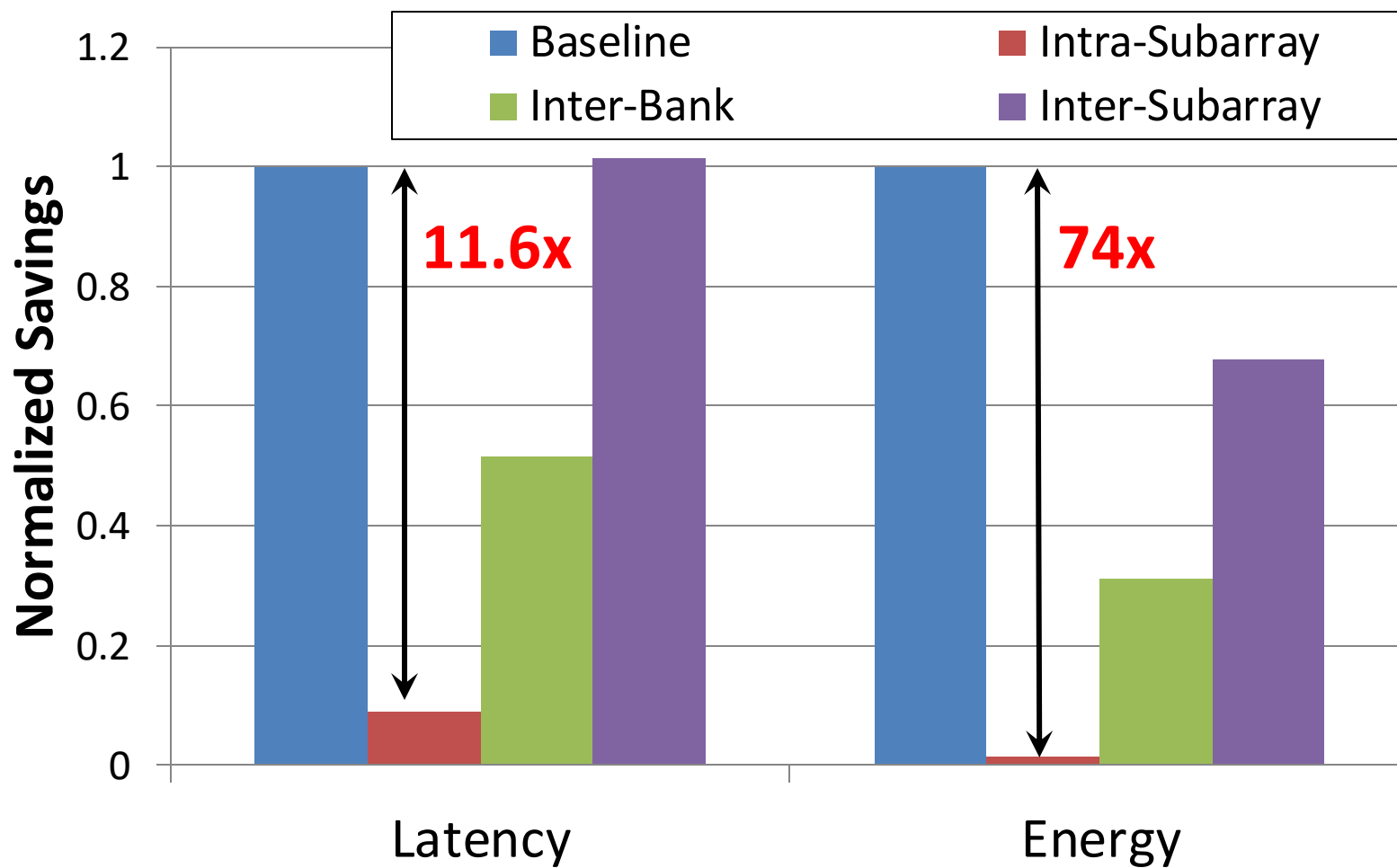
# RowClone: In-DRAM Row Copy

**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

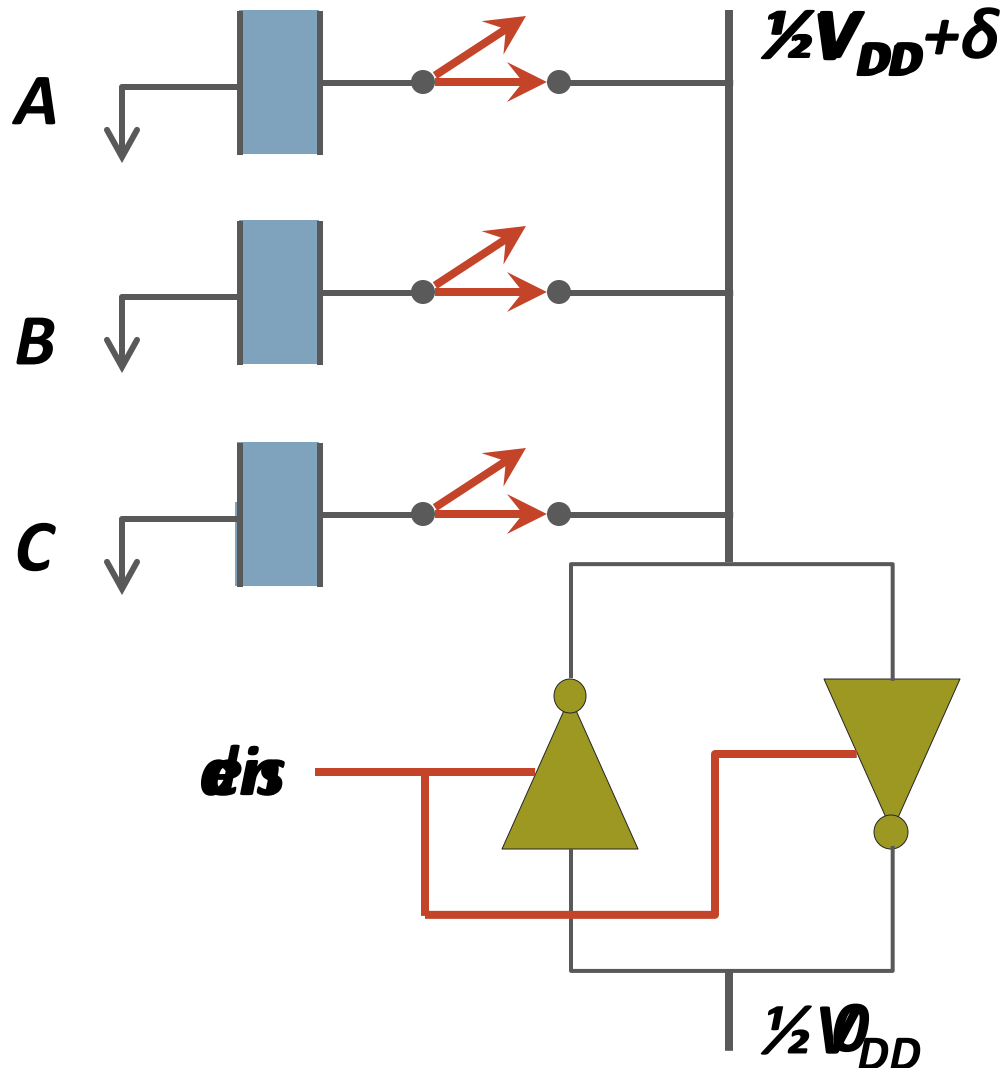
Vivek Seshadri      Yoongu Kim      Chris Fallin\*      Donghyuk Lee  
vseshadr@cs.cmu.edu    yoongukim@cmu.edu    cfallin@c1f.net    donghyuk1@cmu.edu

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

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

Carnegie Mellon University    †Intel Pittsburgh

# In-DRAM AND/OR: Triple Row Activation

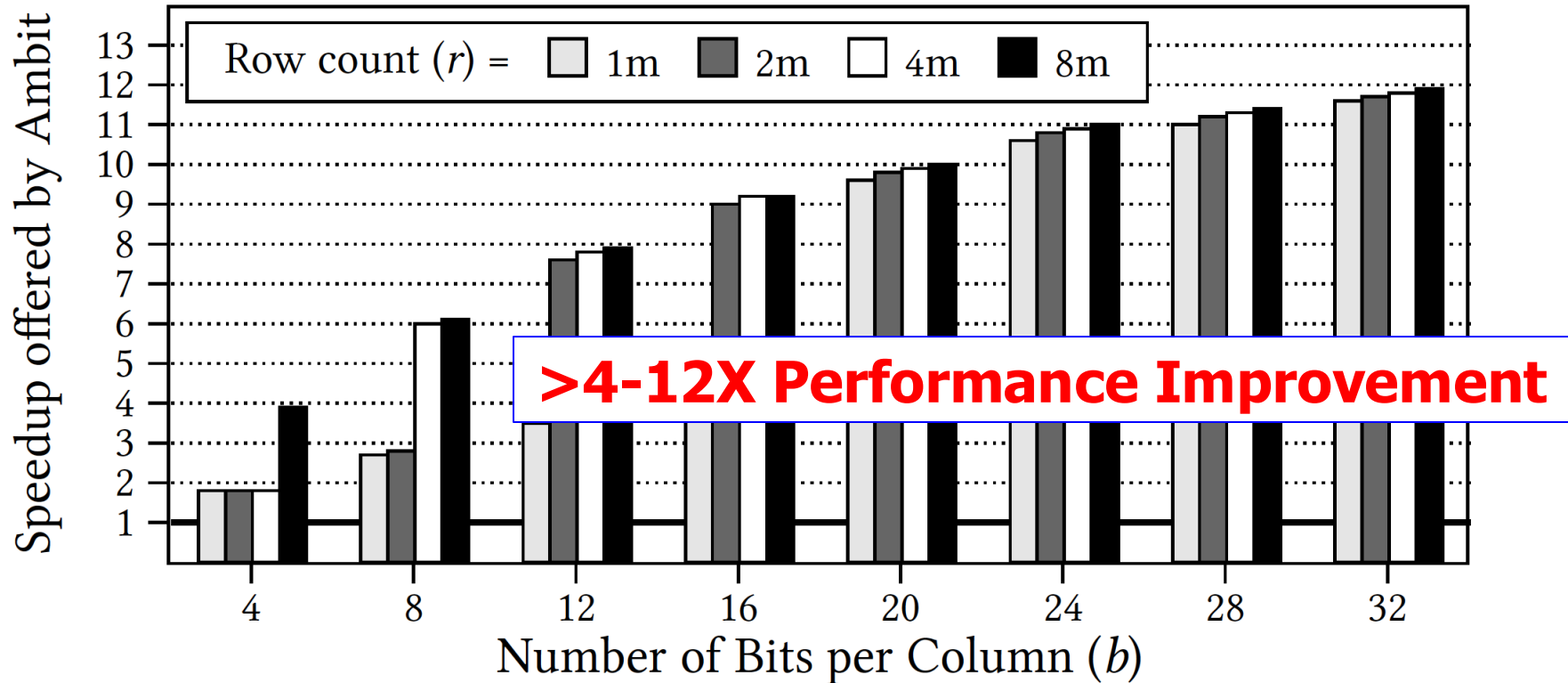


**Final State**  
 **$AB + BC + AC$**

**$C(A + B) +$**   
 **$\sim C(AB)$**

# In-DRAM Acceleration of Database Queries

`'select count(*) from T where c1 <= val <= c2'`

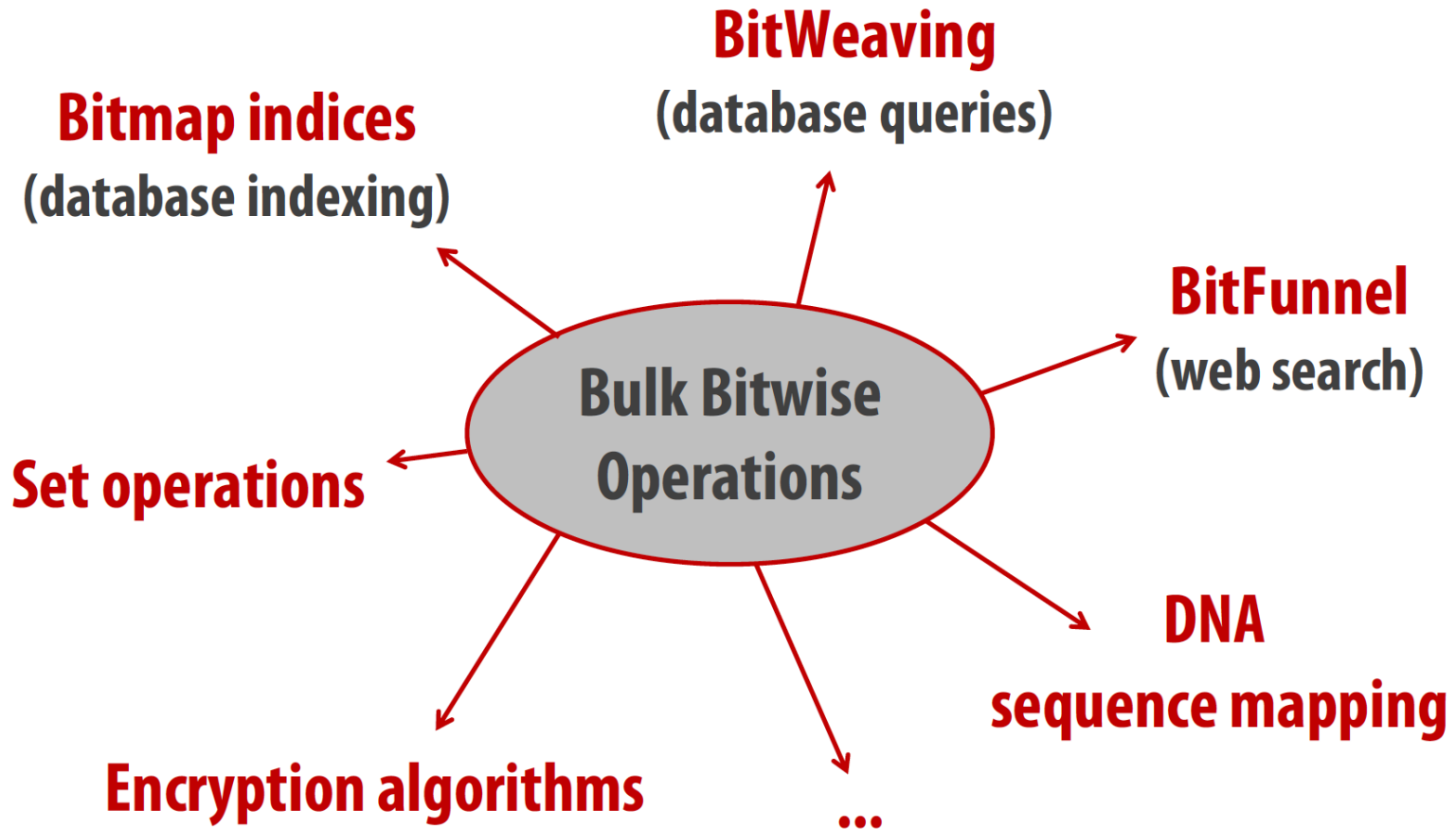


**Figure 11: Speedup offered by Ambit over baseline CPU with SIMD for BitWeaving**

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

# Bulk Bitwise Operations in Workloads

---





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

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

# SIMDRAM Framework

---

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

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

\*Nastaran Hajinazar<sup>1,2</sup>

Nika Mansouri Ghiasi<sup>1</sup>

\*Geraldo F. Oliveira<sup>1</sup>

Minesh Patel<sup>1</sup>

Juan Gómez-Luna<sup>1</sup>

Sven Gregorio<sup>1</sup>

Mohammed Alser<sup>1</sup>

Onur Mutlu<sup>1</sup>

João Dinis Ferreira<sup>1</sup>

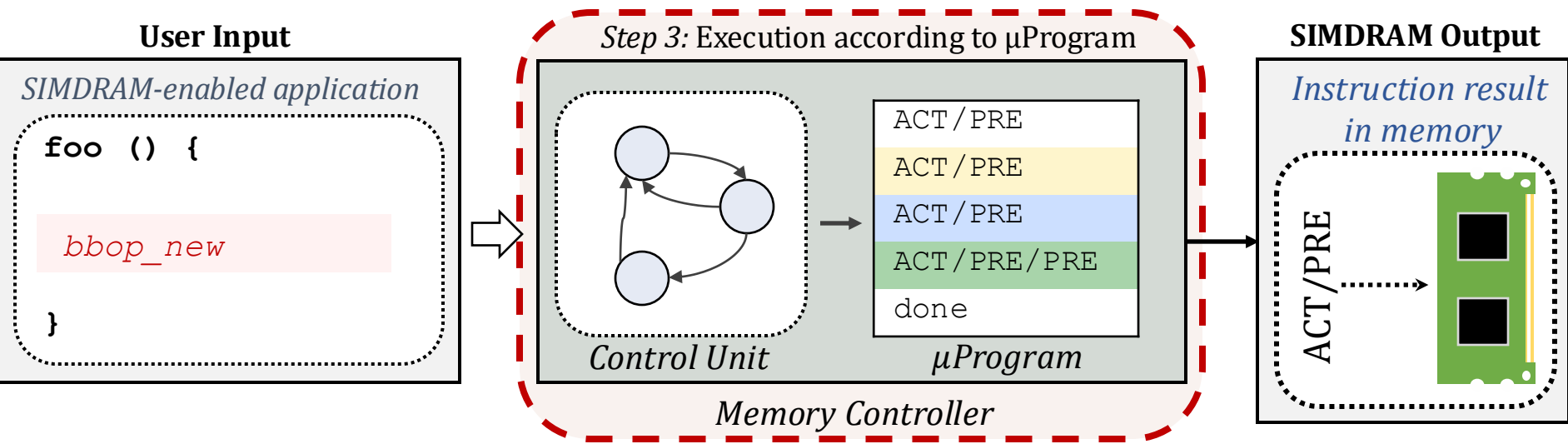
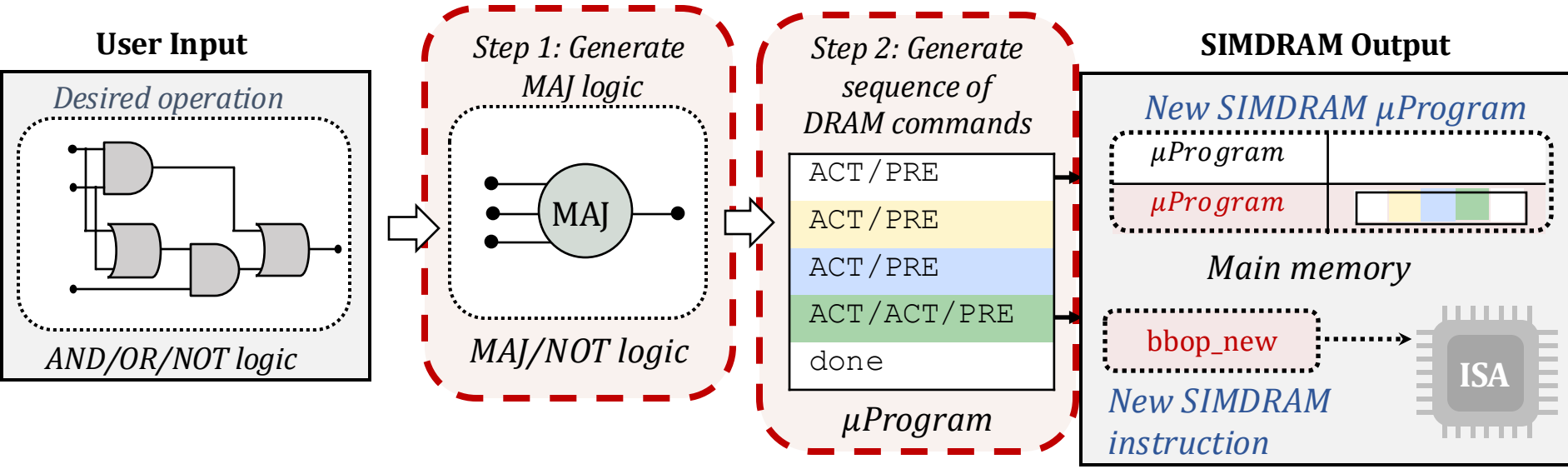
Saugata Ghose<sup>3</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

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

# SIMDRAM Framework: Overview



# SIMDRAM Key Results

Large improvements over **CPU** & **high-end GPU**:

**Throughput: 88× and 5.8×**

(16 complex operations)

**Energy: 257× and 31×**

(16 complex operations)

**Application Performance: 21× and 2.1×**

(seven common real-world applications)

# More on SIMD RAM

---

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

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

\*Nastaran Hajinazar<sup>1,2</sup>

Nika Mansouri Ghiasi<sup>1</sup>

\*Geraldo F. Oliveira<sup>1</sup>

Minesh Patel<sup>1</sup>

Juan Gómez-Luna<sup>1</sup>

Sven Gregorio<sup>1</sup>

Mohammed Alser<sup>1</sup>

Onur Mutlu<sup>1</sup>

João Dinis Ferreira<sup>1</sup>

Saugata Ghose<sup>3</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>Simon Fraser University

<sup>3</sup>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<sup>†</sup>     Ataberk Olgun<sup>†</sup>     Abdullah Giray Yağlıkçı<sup>†</sup>     F. Nisa Bostancı<sup>†</sup>  
Juan Gómez-Luna<sup>†</sup>     Saugata Ghose<sup>‡</sup>     Onur Mutlu<sup>†</sup>

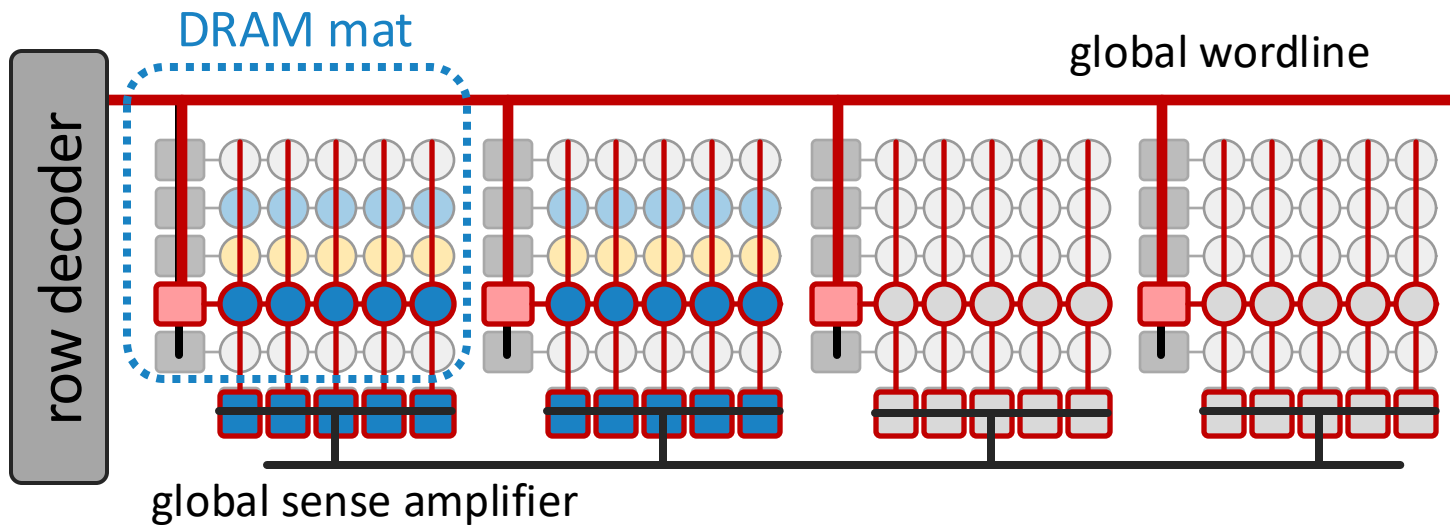
<sup>†</sup> *ETH Zürich*

<sup>‡</sup> *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 (I)

Enable narrower-width operations than a DRAM row



**Key Issue:**

on a DRAM access, the global wordline propagates across all DRAM mats



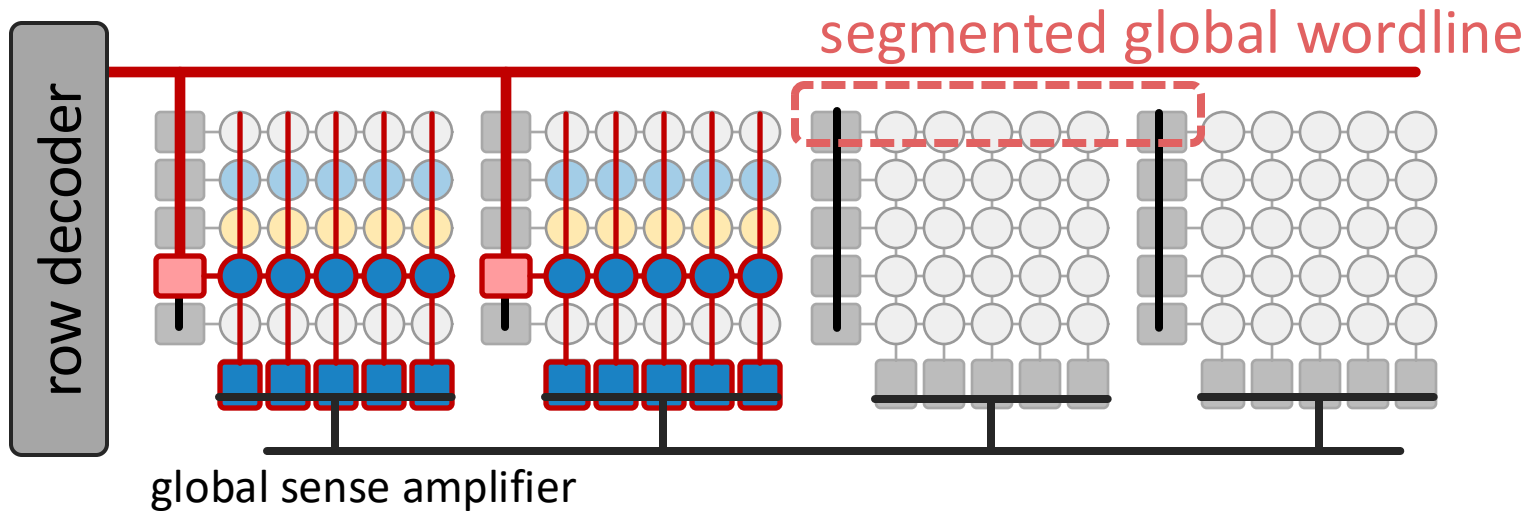
**Fine-Grained DRAM:**

**segment** the global wordline to access **individual** DRAM mats

# MIMDRAM: Key Idea (II)

## Fine-Grained DRAM:

segment the global wordline to access individual DRAM mats



## Fine-grained DRAM for energy-efficient DRAM access:

[Cooper-Balis+, 2010]: Fine-Grained Activation for Power Reduction in DRAM

[Udipi+, 2010]: Rethinking DRAM Design and Organization for Energy-Constrained Multi-Cores

[Zhang+, 2014]: Half-DRAM

[Ha+, 2016]: Improving Energy Efficiency of DRAM by Exploiting Half Page Row Access

[O'Connor+, 2017]: Fine-Grained DRAM

[Olgun+, 2024]: Sectored DRAM



# Sectored DRAM

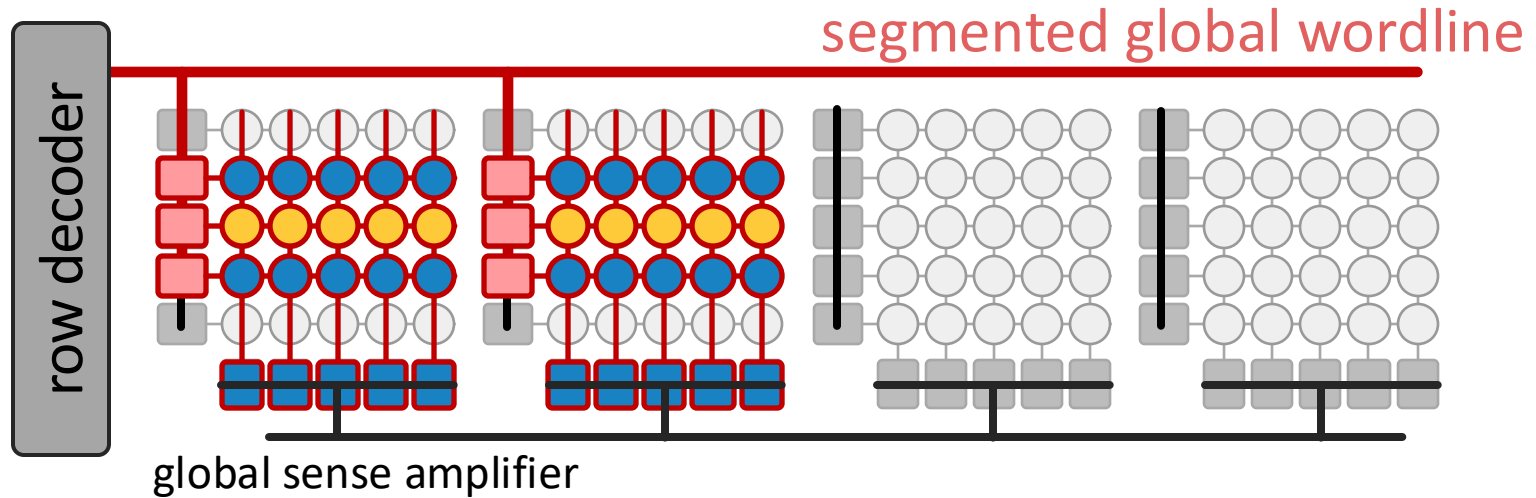
---

- 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<sup>§</sup>    F. Nisa Bostanci<sup>§†</sup>    Geraldo F. Oliveira<sup>§</sup>    Yahya Can Tuğrul<sup>§†</sup>    Rahul Bera<sup>§</sup>  
A. Giray Yağlıkcı<sup>§</sup>    Hasan Hassan<sup>§</sup>    Oğuz Ergin<sup>†</sup>    Onur Mutlu<sup>§</sup>

# MIMDRAM: Key Idea (III)

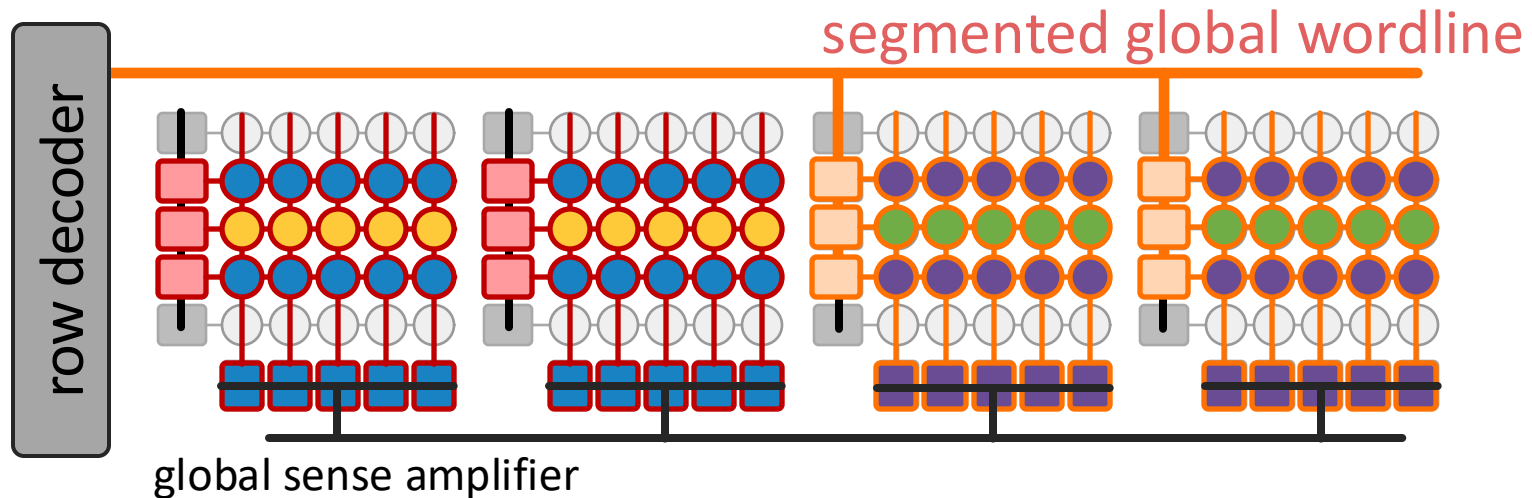


## Use fine-grained DRAM for processing-using-DRAM:

### 1 Improves SIMD utilization

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

# MIMDRAM: Key Idea (III)

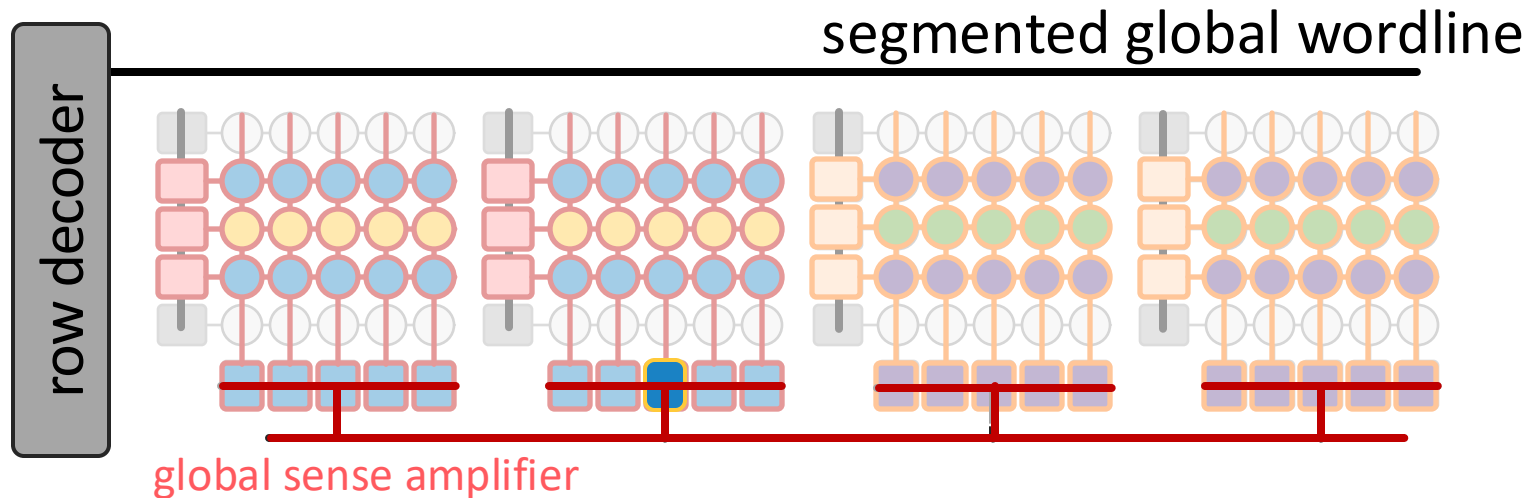


## Use fine-grained DRAM for processing-using-DRAM:

### 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**

# MIMDRAM: Key Idea (III)



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

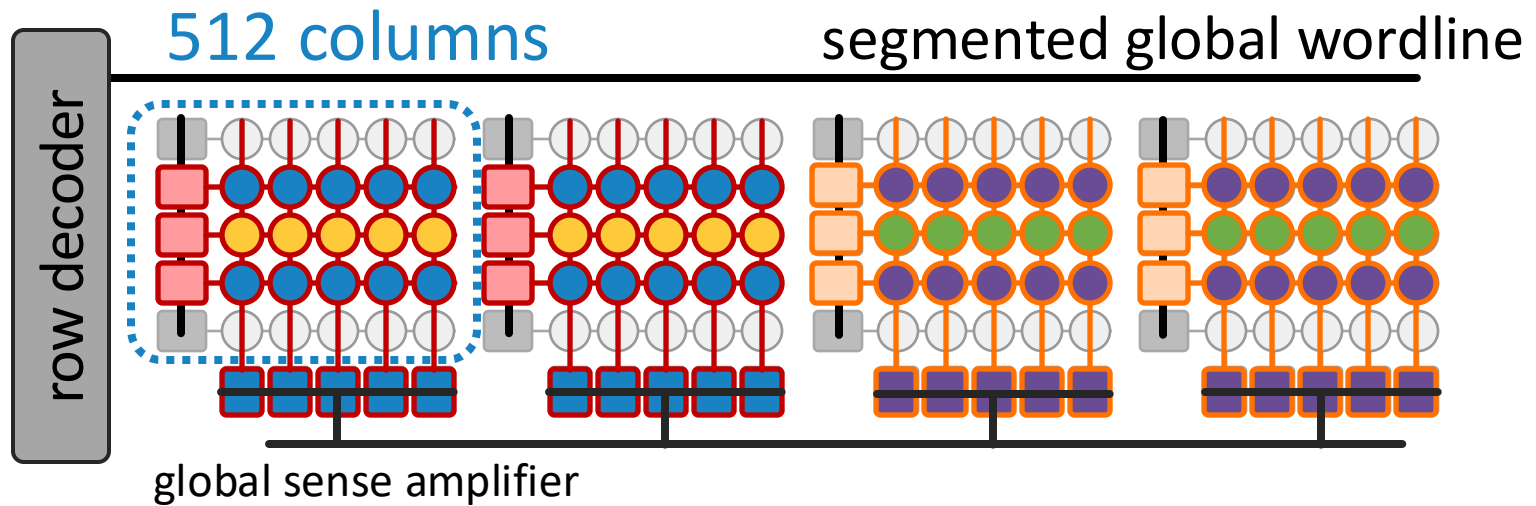
### 1 Improves SIMD utilization

- 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 (III)



## 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

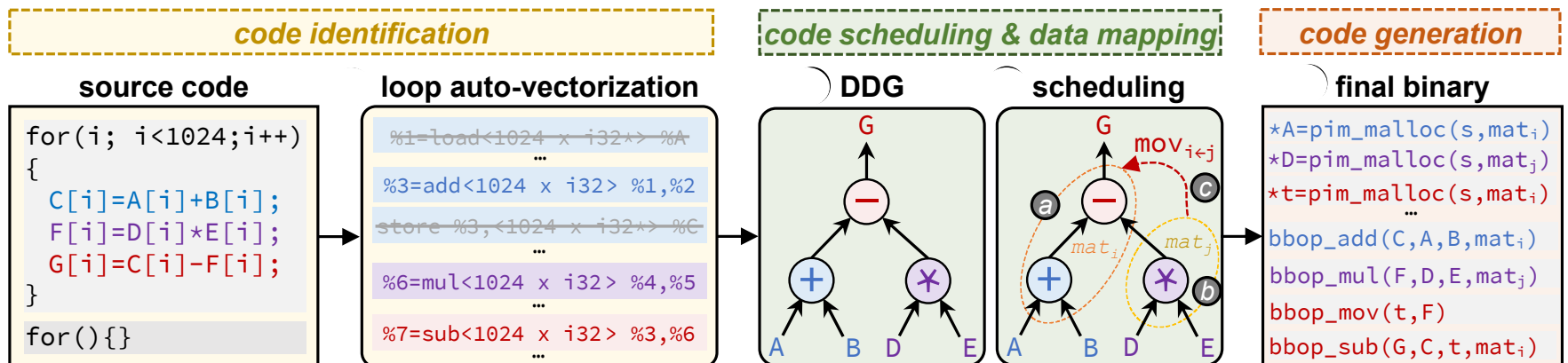
# MIMDRAM: Compiler Support (I)

Goal

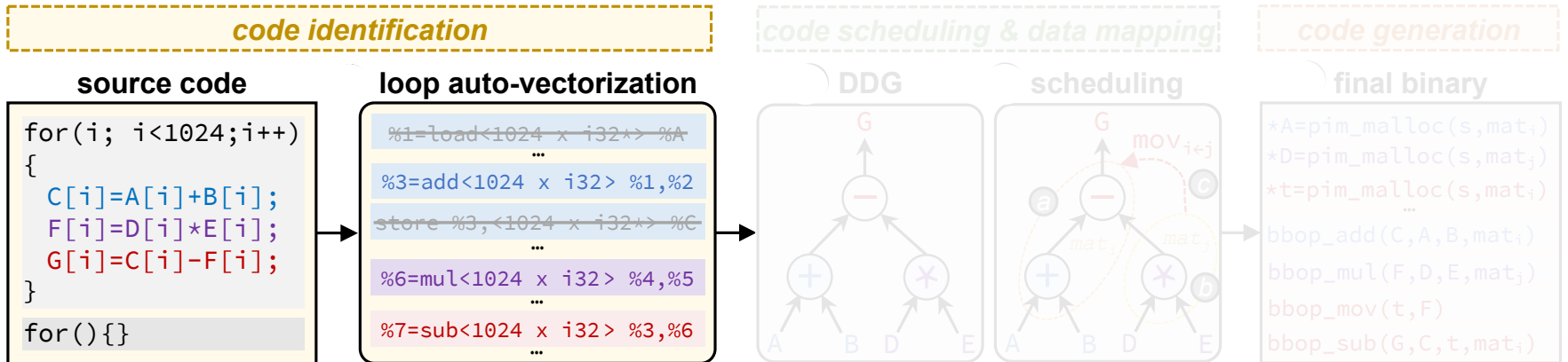
Transparently to programmer:  
extract SIMD parallelism from an application, and  
schedule PUD instructions while maximizing utilization



Three new LLVM-based passes targeting PUD execution

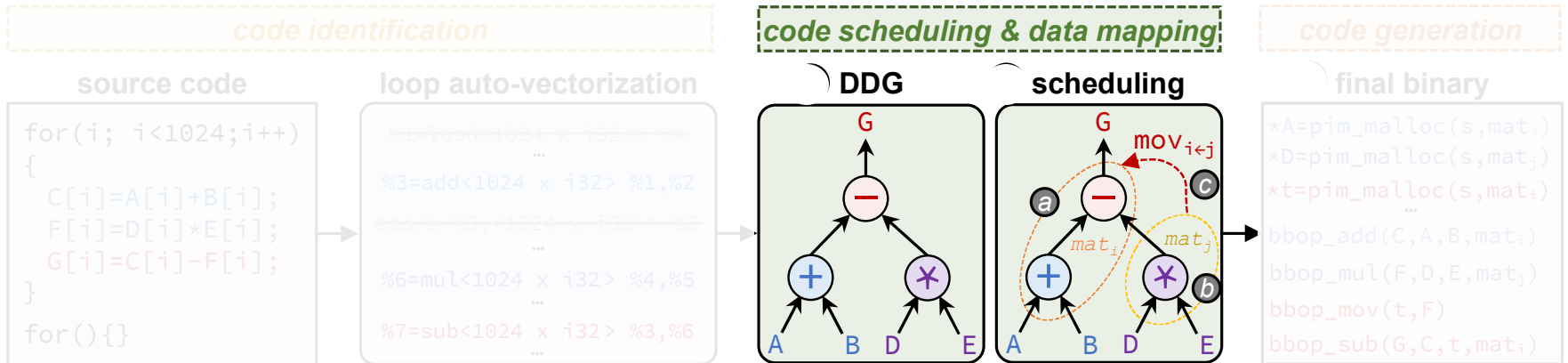


# MIMDRAM: Compiler Support (II)



Goal: Identify SIMD parallelism, generate PUD instructions, and set the appropriate vectorization factor

# MIMDRAM: Compiler Support (II)

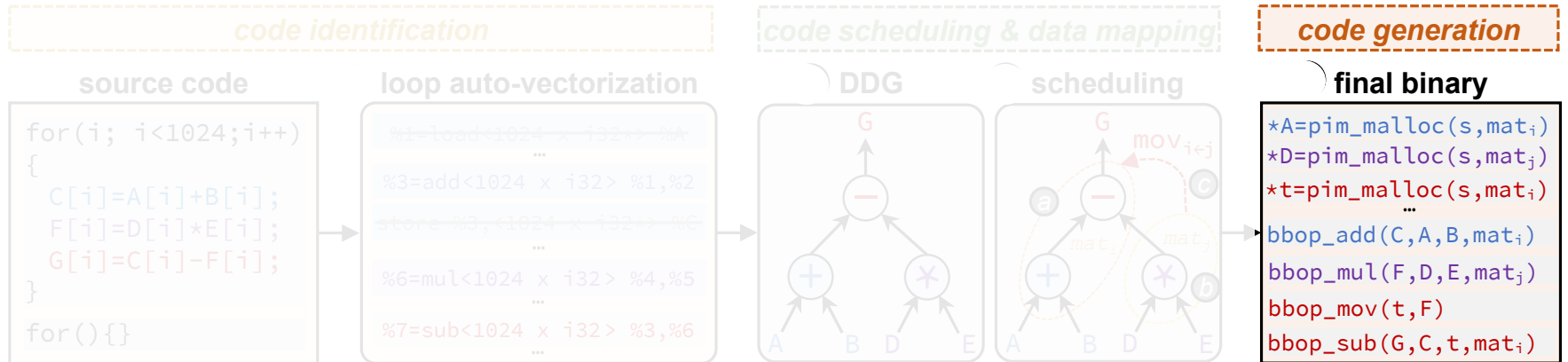


Goal: Identify SIMD parallelism, generate PUD instructions, and set the appropriate vectorization factor

Goal: Improve SIMD utilization by allowing the distribution of independent PUD instructions across DRAM mats



# MIMDRAM: Compiler Support (III)



Goal

Identify SIMD parallelism, generate PUD instructions, and set the appropriate vectorization factor

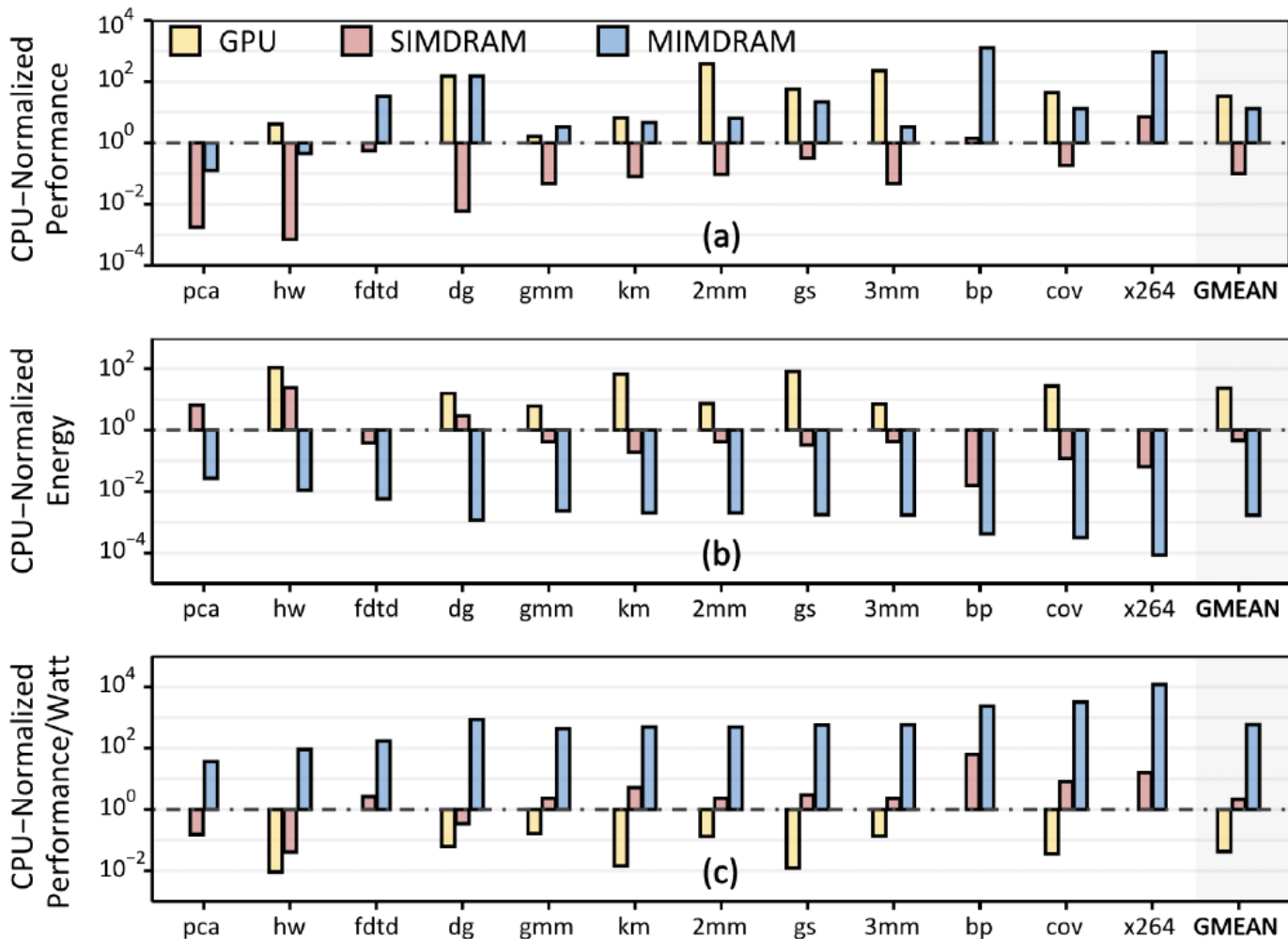
Goal

Improve SIMD utilization by allowing the distribution of independent PUD instructions across DRAM mats

Goal

Generate the appropriate binary for data allocation and PUD instructions

# MIMDRAM Perf, Energy, Perf/Watt



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

# 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 DRAM timing parameters to mimic RowClone

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

Ataberk Olgun<sup>§†</sup>

Juan Gómez Luna<sup>§</sup>

Konstantinos Kanellopoulos<sup>§</sup>

Behzad Salami<sup>§\*</sup>

Hasan Hassan<sup>§</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

§ETH Zürich

†TOBB ETÜ

\*BSC

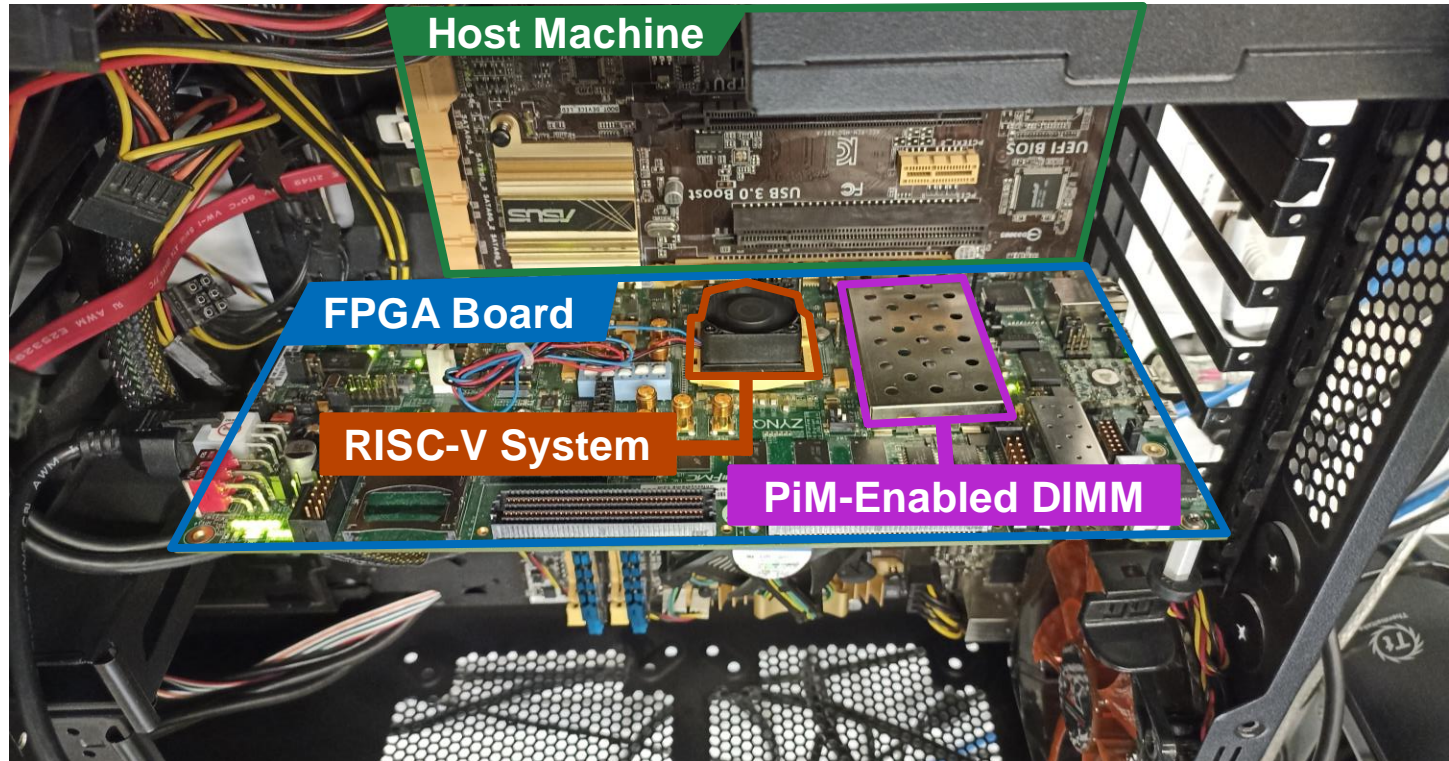
<https://arxiv.org/pdf/2111.00082.pdf>

<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:

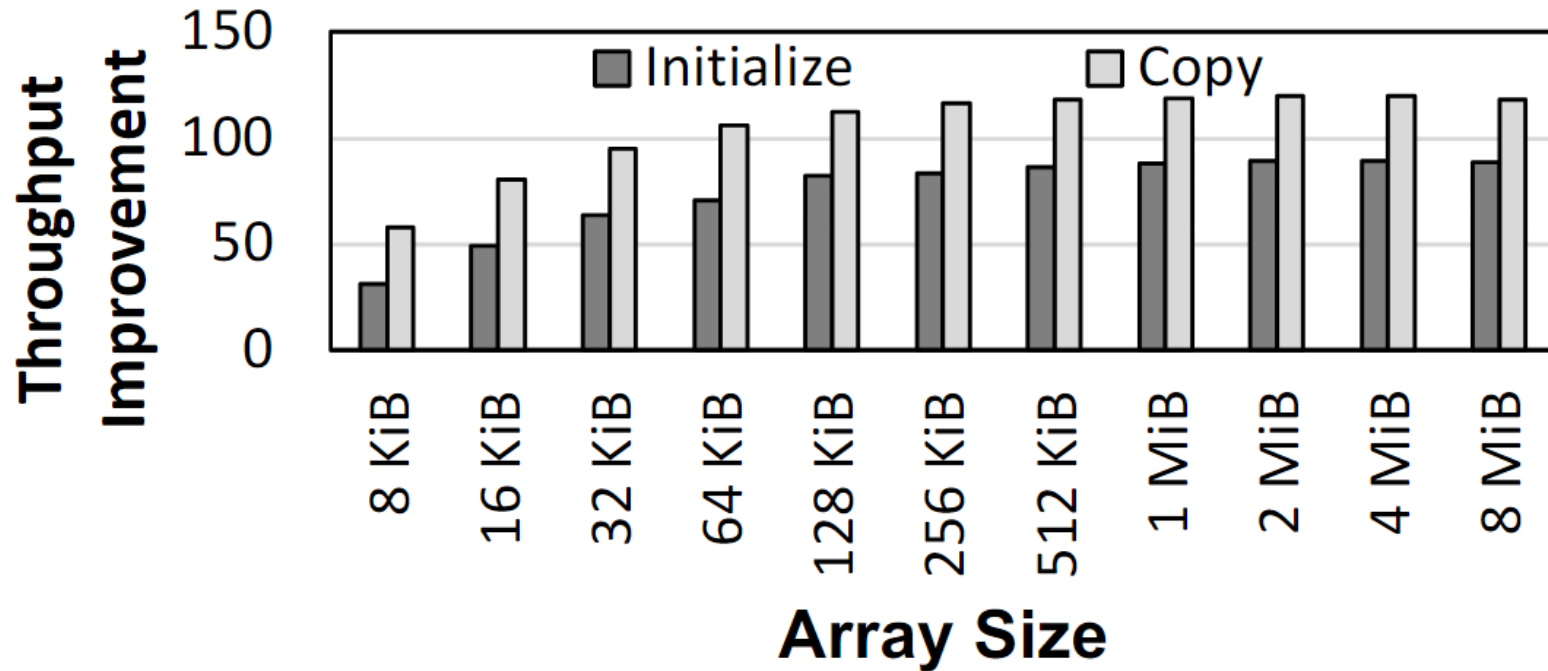
- 1- Open IP Catalog
- 2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

<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<sup>§</sup>

Juan Gómez Luna<sup>§</sup>

Konstantinos Kanellopoulos<sup>§</sup>

Behzad Salami<sup>§</sup>

Hasan Hassan<sup>§</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*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

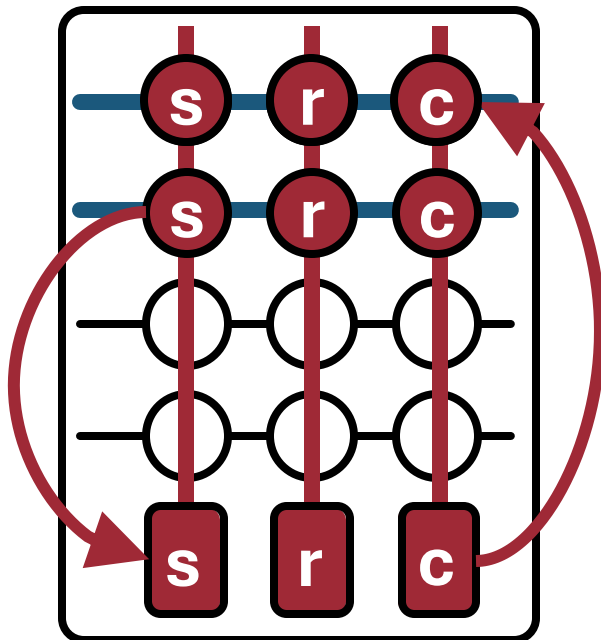
**2** Can perform **NOT operation** with up to **32 output operands**

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

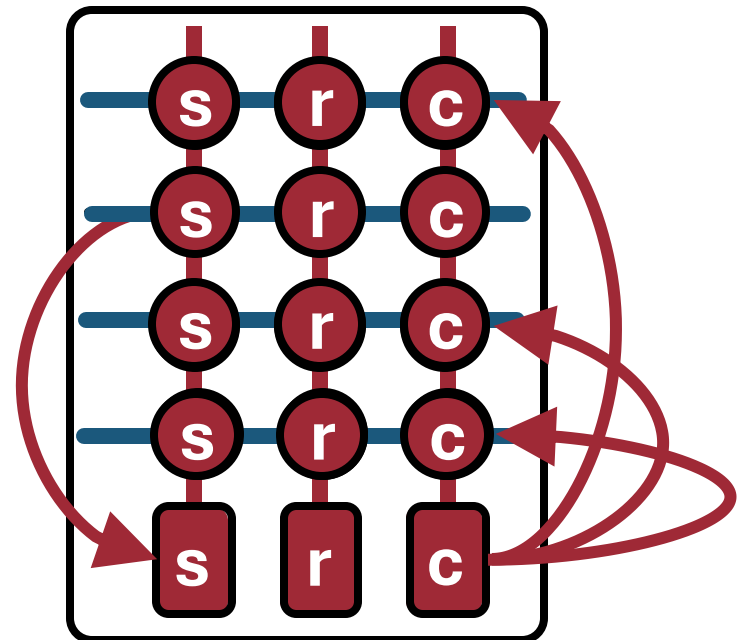
# In-DRAM Multiple Row Copy (Multi-RowCopy)

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

RowClone

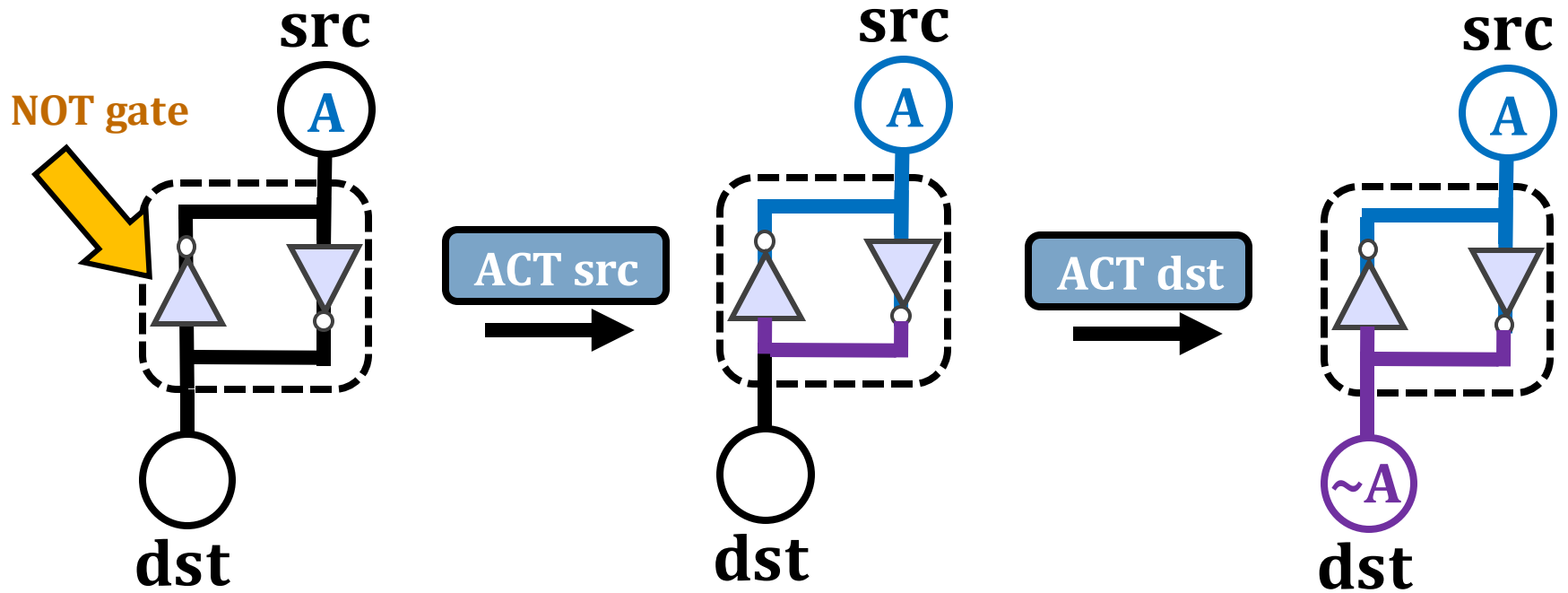


Multi-RowCopy



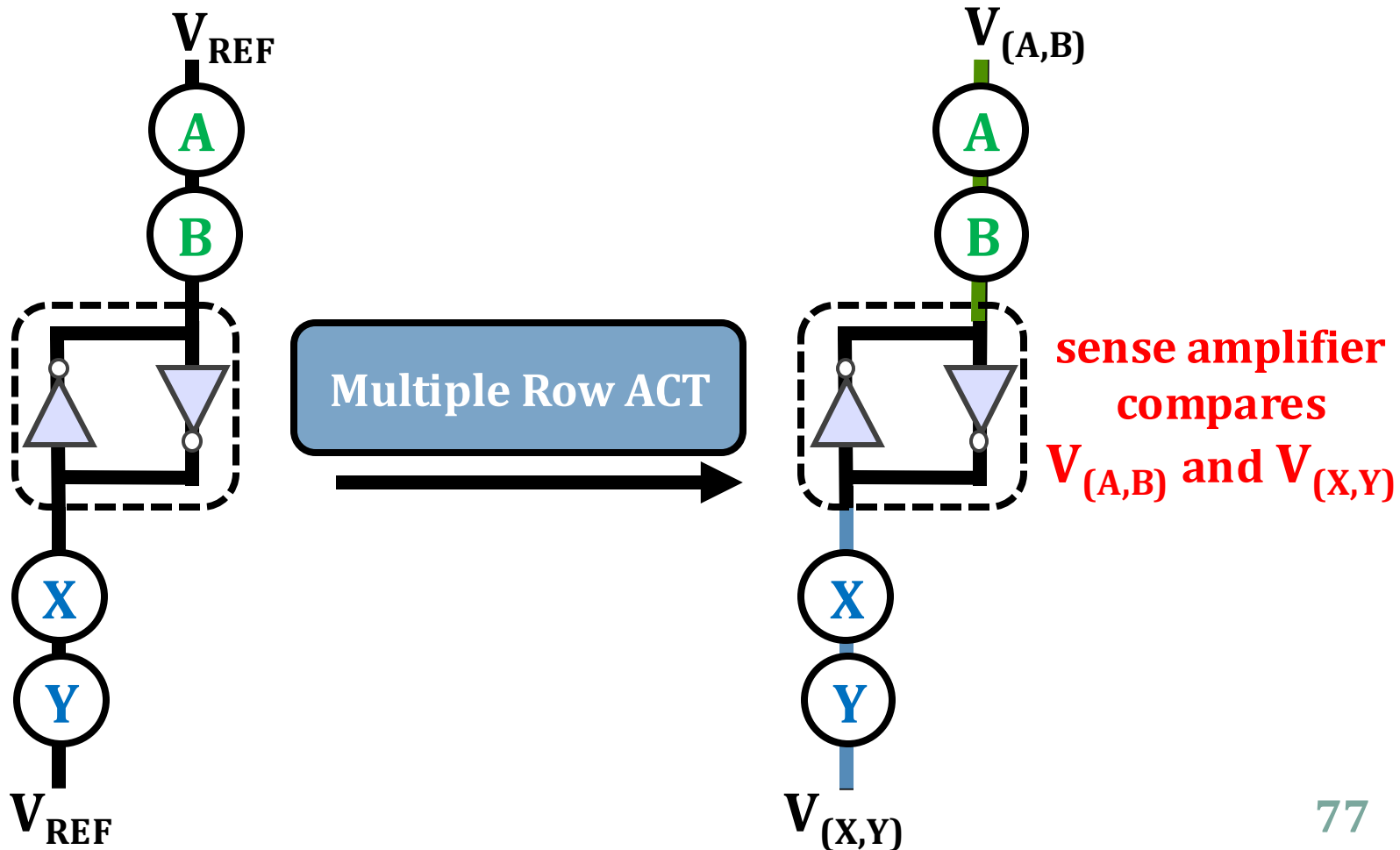
# Key Idea: NOT Operation

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

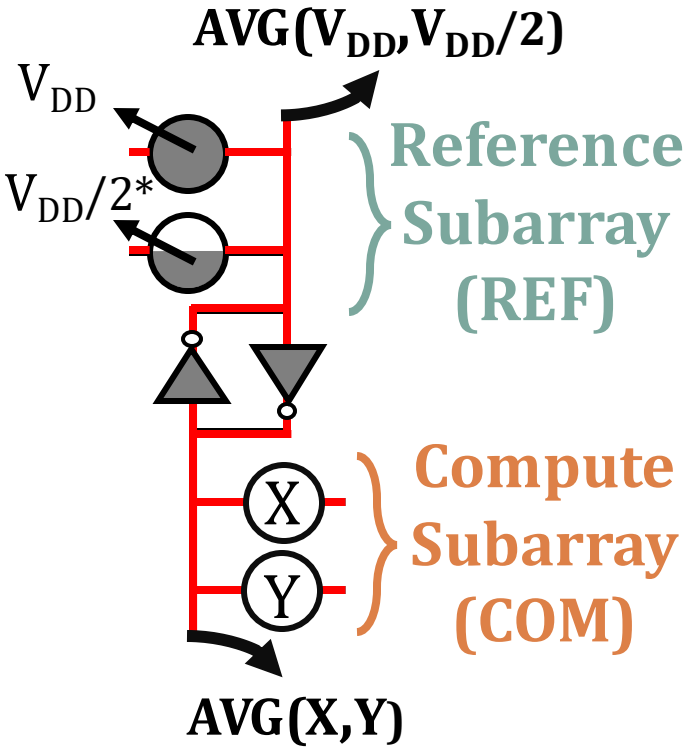


# Key Idea: NAND, NOR, AND, OR

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

# 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



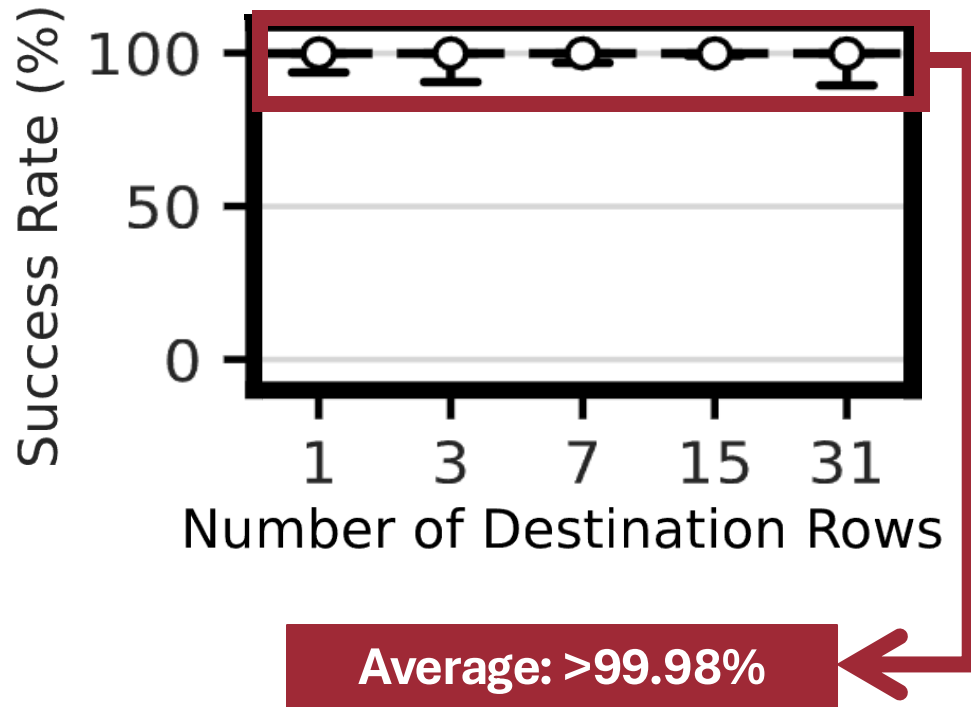


# DRAM Chips Tested

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

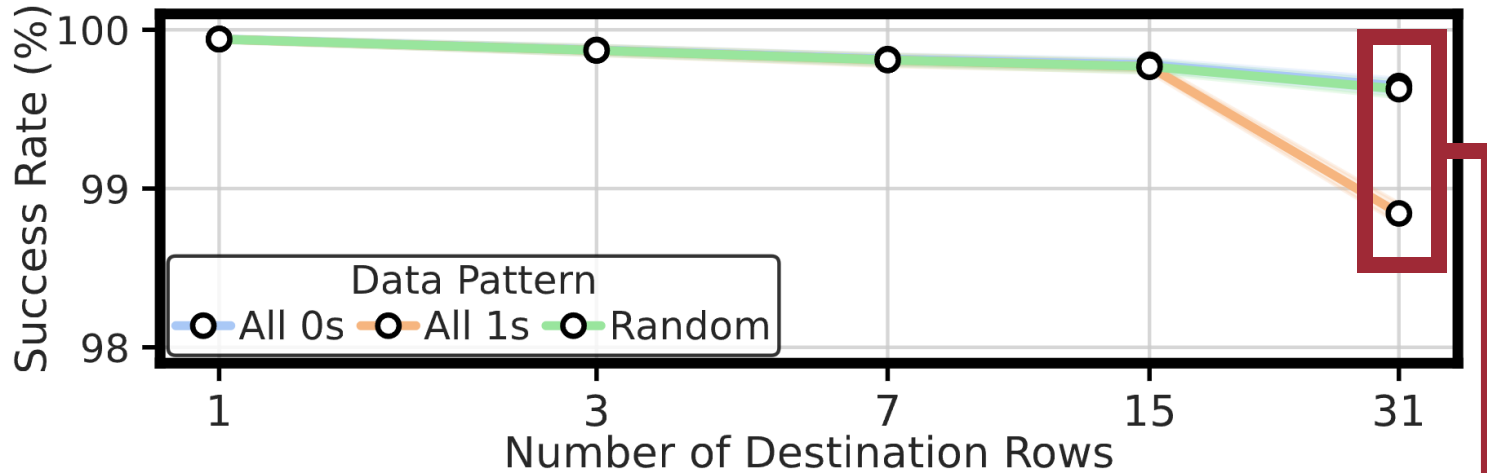
Chip Mfr.	#Modules (#Chips)	Die Rev.	Mfr. Date <sup>a</sup>	Chip Density	Chip Org.	Speed Rate
SK Hynix	9 (72)	M	N/A	4Gb	x8	2666MT/s
	5 (40)	A	N/A	4Gb	x8	2133MT/s
	1 (16)	A	N/A	8Gb	x8	2666MT/s
	1 (32)	A	18-14	4Gb	x4	2400MT/s
	1 (32)	A	16-49	8Gb	x4	2400MT/s
	1 (32)	M	16-22	8Gb	x4	2666MT/s
Samsung	1 (8)	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**

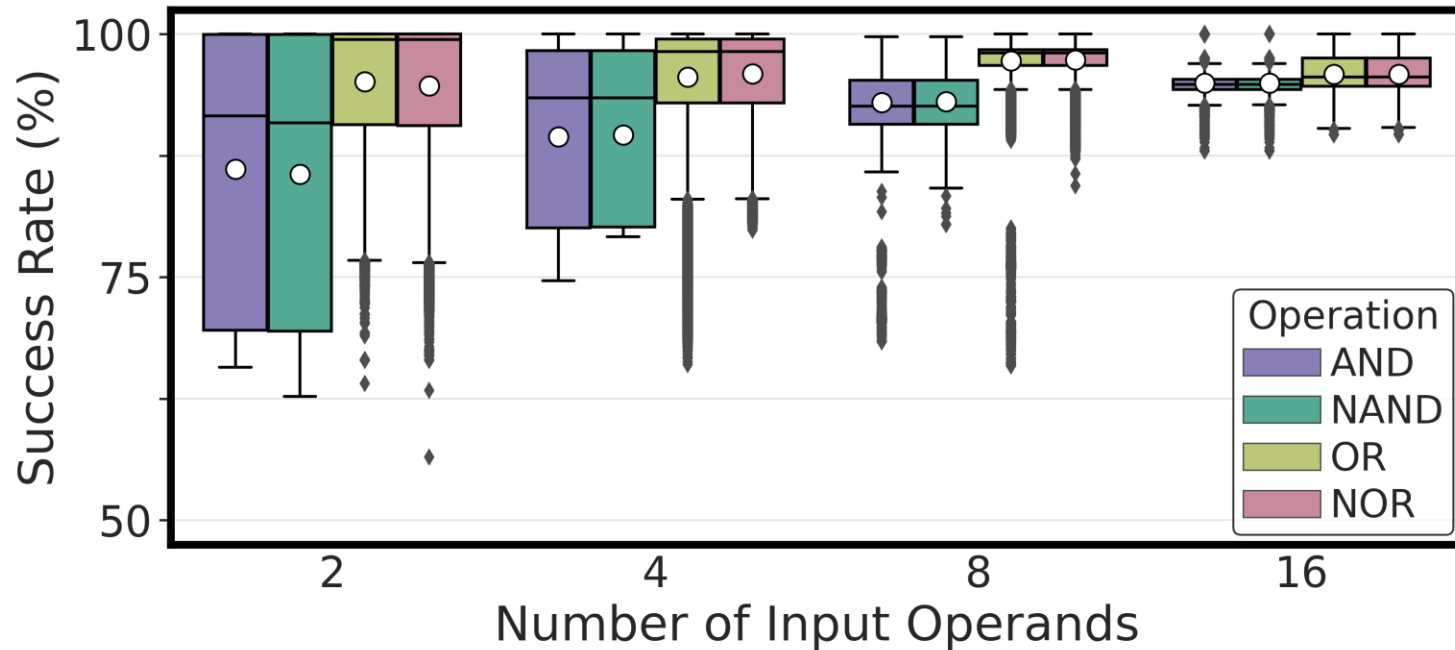
# Impact of Data Pattern



At most 0.79% decrease in average success rate

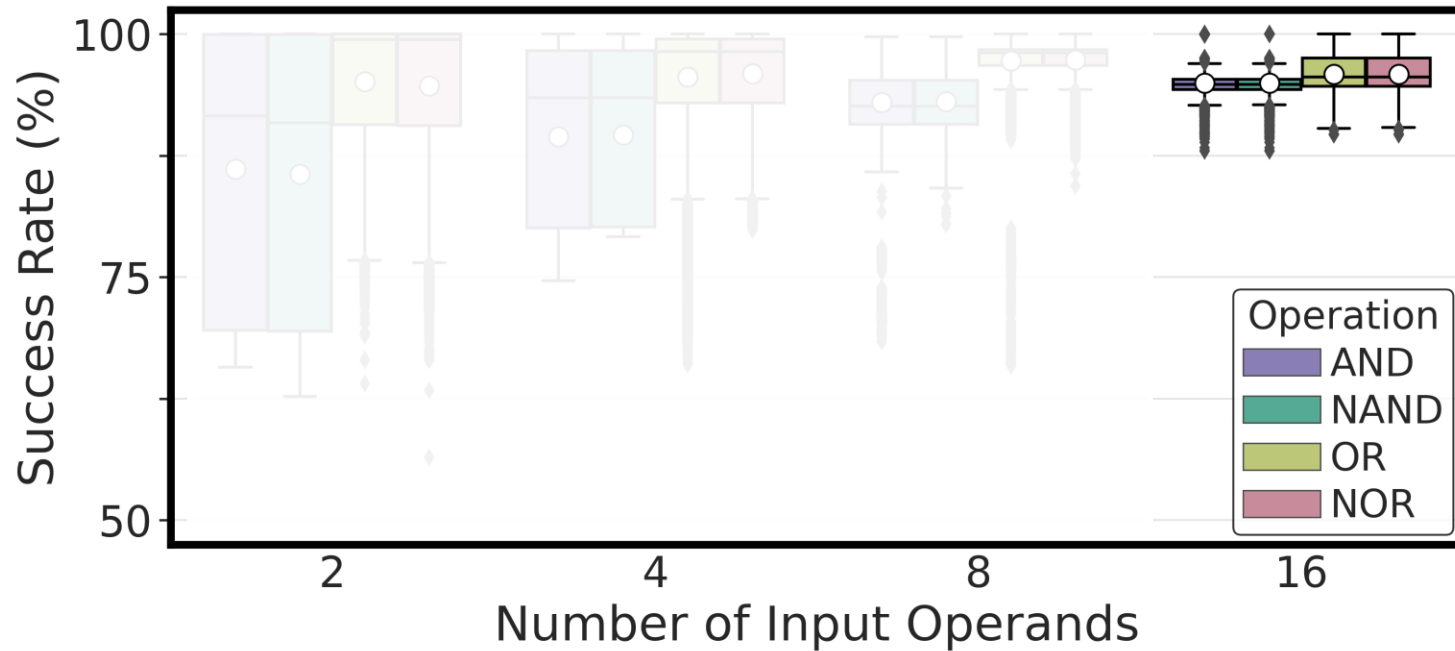
Data pattern has a small effect on the success rate of the Multi-RowCopy operation

# Performing AND, NAND, OR, and NOR



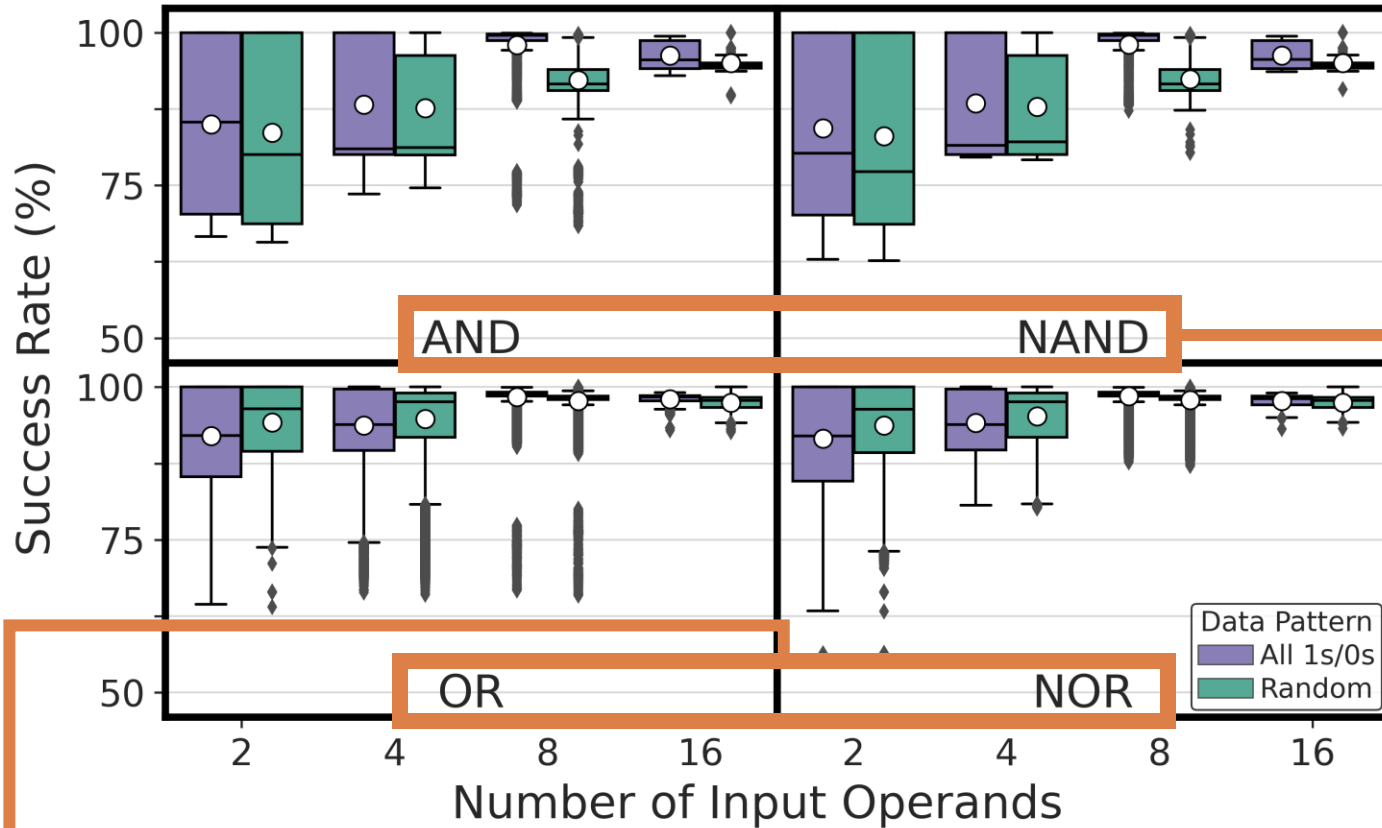
**COTS DRAM chips can perform {2, 4, 8, 16}-input AND, NAND, OR, and NOR operations**

# Performing AND, NAND, OR, and NOR



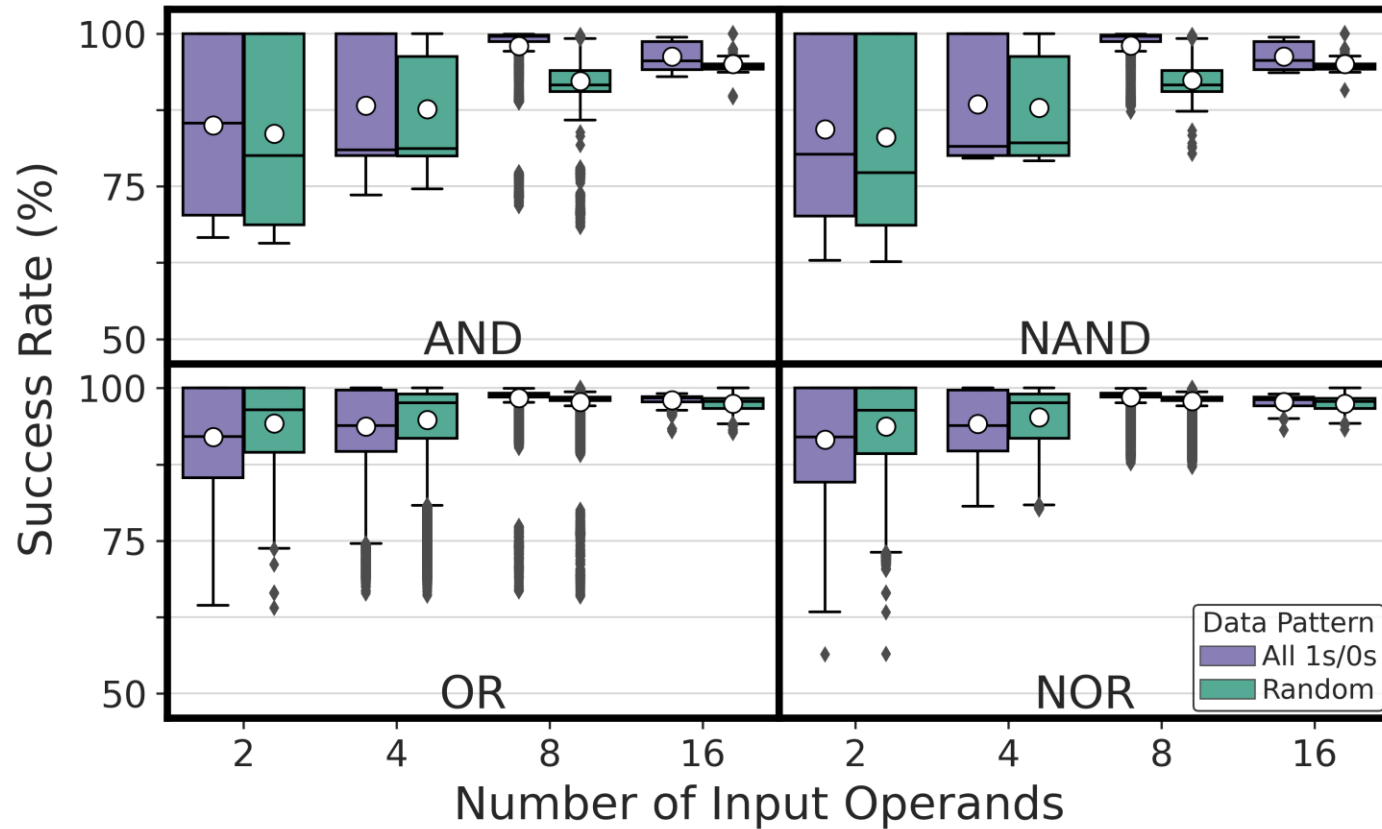
**COTS DRAM chips can perform  
16-input AND, NAND, OR, and NOR operations  
with very high success rate (>94%)**

# Impact of Data Pattern



Impact of data pattern is **consistent** across all tested operations

# Impact of Data Pattern



**Data pattern slightly affects the reliability of AND, NAND, OR, and NOR operations**

# More in the Paper

- Detailed hypotheses & key ideas to perform
  - NOT operation
  - Many-input AND, NAND, OR, and NOR operations
- How the reliability of bitwise operations are affected by
  - The location of activated rows
  - Temperature (for AND, NAND, OR, and NOR)
  - DRAM speed rate
  - Chip density and die revision
- Discussion on the limitations of COTS DRAM chips



# More on Functionally-Complete DRAM

---

- Ismail Emir Yüksel, Yahya Can Tuğrul, Ataberk Olgun, F. Nisa Bostancı, A. Giray Yağlıkçı, 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

Ismail 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 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<sup>1</sup> Yahya Can Tuğrul<sup>1,2</sup> F. Nisa Bostancı<sup>1</sup> Geraldo F. Oliveira<sup>1</sup>

A. Giray Yağlıkçı<sup>1</sup> Ataberk Olgun<sup>1</sup> Melina Soysal<sup>1</sup> Haocong Luo<sup>1</sup>

Juan Gómez-Luna<sup>1</sup> Mohammad Sadrosadati<sup>1</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*

<sup>2</sup>*TOBB University of Economics and Technology*

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<sup>‡§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§‡</sup>

<sup>‡</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich

# 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<sup>§†</sup>

Minesh Patel<sup>§</sup>

A. Giray Yağlıkçı<sup>§</sup>

Haocong Luo<sup>§</sup>

Jeremie S. Kim<sup>§</sup>

F. Nisa Bostanci<sup>§†</sup>

Nandita Vijaykumar<sup>§⊙</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*TOBB University of Economics and Technology*

<sup>⊙</sup>*University of Toronto*

# In-DRAM TRNG: Recent Results

## ■ N-row Activation

- initialize cell values to sample random values in sense amplifiers

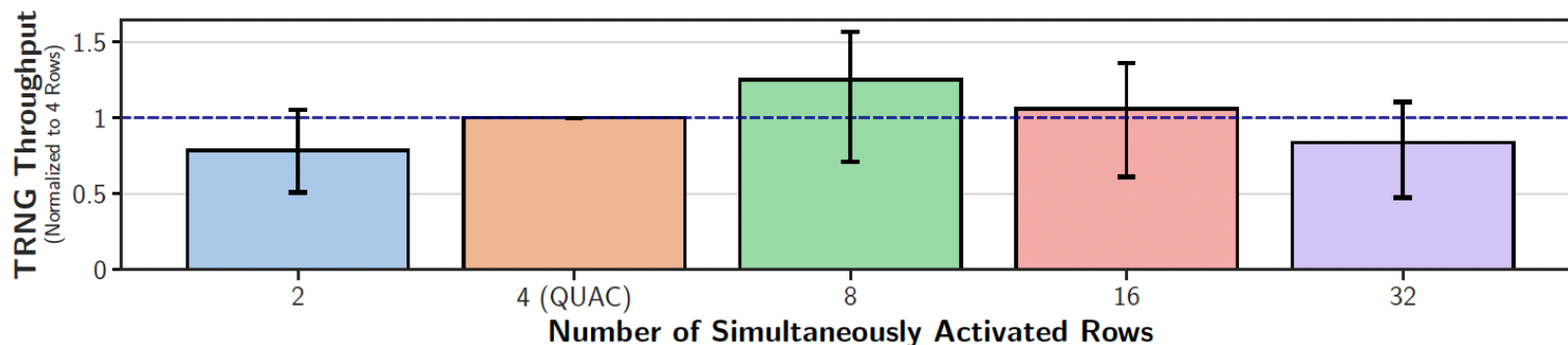


Fig. 11: **Throughput of generating true random numbers, as measured in 96 COTS DRAM chips using multiple-row activation, normalized to state-of-the-art DRAM-based TRNG, QUAC-TRNG (i.e., 4-row activation) [135].** Each error bar shows the range across all tested chips. We observe that random numbers 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  $1.25\times$  and  $1.06\times$ , respectively, while 2- and 32-row activation-based TRNG provides  $0.69\times$  and  $0.84\times$  the throughput of the state-of-the-art.

# 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 Bostanci<sup>†§</sup>      Ataberk Olgun<sup>†§</sup>      Lois Orosa<sup>§</sup>      A. Giray Yağlıkçı<sup>§</sup>  
Jeremie S. Kim<sup>§</sup>      Hasan Hassan<sup>§</sup>      Oğuz Ergin<sup>†</sup>      Onur Mutlu<sup>§</sup>

<sup>†</sup>*TOBB University of Economics and Technology*      <sup>§</sup>*ETH Zürich*

# In-DRAM Physical Unclonable Functions

---

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

## The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions

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

Jeremie S. Kim<sup>†§</sup>

Minesh Patel<sup>§</sup>

Hasan Hassan<sup>§</sup>

Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>§</sup>ETH Zürich



# In-DRAM 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<sup>§</sup>

Gabriel Falcao<sup>†</sup>

Juan Gómez-Luna<sup>§</sup>

Mohammed Alser<sup>§</sup>

Lois Orosa<sup>§∇</sup>

Mohammad Sadrosadati<sup>§</sup>

Jeremie S. Kim<sup>§</sup>

Geraldo F. Oliveira<sup>§</sup>

Taha Shahroodi<sup>‡</sup>

Anant Nori<sup>\*</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>ETH Zürich

<sup>†</sup>IT, University of Coimbra

<sup>∇</sup>Galicia Supercomputing Center

<sup>‡</sup>TU Delft

<sup>\*</sup>Intel

# In-Flash Bulk Bitwise Execution

---

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myung Suk 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<sup>§∇</sup> Roknoddin Azizi<sup>§</sup> Geraldo F. Oliveira<sup>§</sup> Mohammad Sadrosadati<sup>§</sup>  
Rakesh Nadig<sup>§</sup> David Novo<sup>†</sup> Juan Gómez-Luna<sup>§</sup> Myung Suk Kim<sup>‡</sup> Onur Mutlu<sup>§</sup>

<sup>§</sup>ETH Zürich    <sup>∇</sup>POSTECH    <sup>†</sup>LIRMM, Univ. Montpellier, CNRS    <sup>‡</sup>Kyungpook National University

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

*SAFARI Research Group*

<sup>a</sup>*ETH Zürich*

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

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

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

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

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

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

## 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

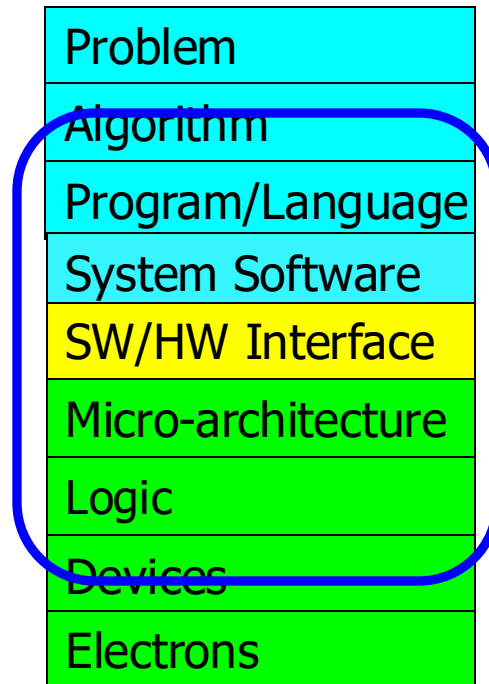
**All can be solved with change of mindset**

---

# We Need to Revisit the Entire Stack

---

- With a **memory-centric mindset**



**We can get there step by step**

# Concluding Remarks



Fundamentally  
Energy-Efficient  
**(Data-Centric)**

Computing Architectures

# Fundamentally High-Performance **(Data-Centric)** Computing Architectures

# Computing Architectures with Minimal Data Movement

# 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

# PIM Tutorial November 2024 Edition

## MICRO 2024 - Tutorial on Memory-Centric Computing Systems

Saturday, November 2<sup>nd</sup>, 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/>

# Upcoming PIM Tutorials/Workshops (I)

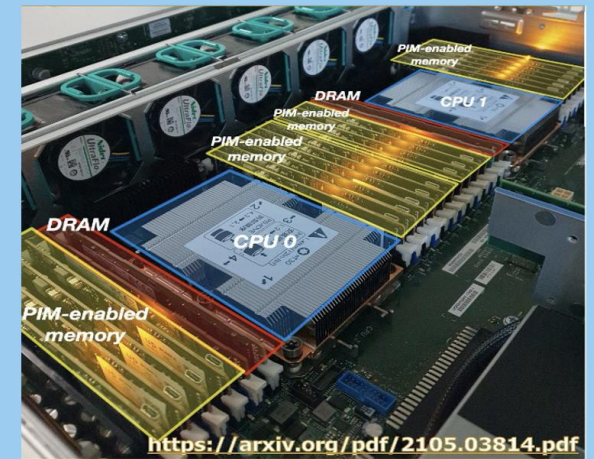
## ASPLOS 2025 - 1<sup>st</sup> Workshop on Memory-Centric Computing Systems

Sunday, March 30<sup>th</sup>, 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>

# Upcoming PIM Tutorials/Workshops (II)

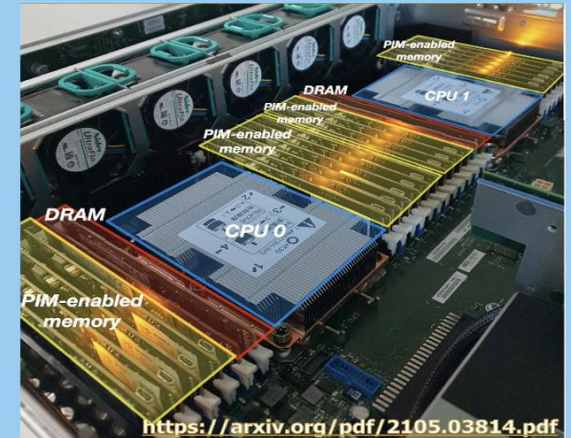
## ICS 2025 - 2<sup>nd</sup> Workshop on Memory-Centric Computing Systems

Sunday, June 8<sup>th</sup>, 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>

# Upcoming PIM Tutorials/Workshops (III)

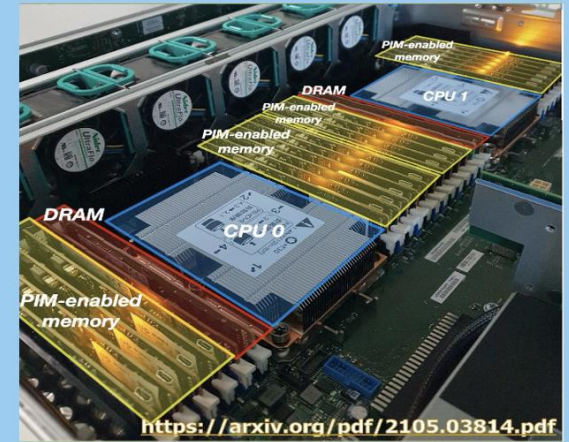
## ISCA 2025 - 3<sup>rd</sup> Workshop on Memory-Centric Computing Systems

Saturday, 21<sup>st</sup> 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>



# Open Source Tools: SAFARI GitHub



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

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

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

🏠 Overview 📁 Repositories 80 📁 Projects 📁 Packages 👤 People 13

### 📁 ramulator Public

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

● C++ ☆ 583 🍴 209

### 📁 prim-benchmarks Public

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

● C ☆ 137 🍴 50

### 📁 MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 277 🍴 149

### 📁 rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at [http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer\\_isca14.pdf](http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf).

● C ☆ 217 🍴 42

### 📁 SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 127 🍴 28

### 📁 Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (<https://arxiv.org/pdf/2109.12021.pdf>).

● C++ ☆ 117 🍴 36

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

# Memory-Centric Computing

## Recent Advances in Processing-in-DRAM

Onur Mutlu

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

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

1 March 2025

Memory-Centric Computing Tutorial @ PPOPP

**SAFARI**

**ETH** zürich



# Backup Slides

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

<sup>‡</sup>Carnegie Mellon University   <sup>†</sup>NVIDIA   \*KAIST   <sup>§</sup>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<sup>1</sup>   Juan Gómez-Luna<sup>1</sup>   Izzat El Hajj<sup>2</sup>   Yuxin Guo<sup>1</sup>   Onur Mutlu<sup>1</sup>  
<sup>1</sup>ETH Zürich   <sup>2</sup>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.

# 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<sup>†</sup>, Saugata Ghose<sup>†</sup>, Minesh Patel<sup>†</sup>, Hasan Hassan<sup>†§</sup>, Brandon Lucia<sup>†</sup>,  
Kevin Hsieh<sup>†</sup>, Krishna T. Malladi<sup>\*</sup>, Hongzhong Zheng<sup>\*</sup>, and Onur Mutlu<sup>‡†</sup>

<sup>†</sup> *Carnegie Mellon University*   <sup>\*</sup> *Samsung Semiconductor, Inc.*   <sup>§</sup> *TOBB ETÜ*   <sup>‡</sup> *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<sup>†</sup>

Saugata Ghose<sup>†</sup>

Minesh Patel<sup>\*</sup>

Hasan Hassan<sup>\*</sup>

Brandon Lucia<sup>†</sup>

Rachata Ausavarungnirun<sup>†‡</sup>

Kevin Hsieh<sup>†</sup>

Nastaran Hajinazar<sup>◇†</sup>

Krishna T. Malladi<sup>§</sup>

Hongzhong Zheng<sup>§</sup>

Onur Mutlu<sup>\*†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>ETH Zürich

<sup>‡</sup>KMUTNB

<sup>◇</sup>Simon Fraser University

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

# 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<sup>†</sup> Samira Khan<sup>‡</sup> Nandita Vijaykumar<sup>†</sup>

Kevin K. Chang<sup>†</sup> Amirali Boroumand<sup>†</sup> Saugata Ghose<sup>†</sup> Onur Mutlu<sup>§†</sup>

<sup>†</sup>*Carnegie Mellon University*   <sup>‡</sup>*University of Virginia*   <sup>§</sup>*ETH Zürich*

# 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<sup>§</sup>

Juan Gómez Luna<sup>§</sup>

Konstantinos Kanellopoulos<sup>§</sup>

Behzad Salami<sup>§</sup>

Hasan Hassan<sup>§</sup>

Oğuz Ergin<sup>†</sup>

Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*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.*  
[\[arXiv version\]](#)  
[\[Ramulator 2.0 Source Code\]](#)

## Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

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



# 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/>

# Open Source Tools: SAFARI GitHub



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

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

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

🏠 Overview 📁 Repositories 80 📁 Projects 📁 Packages 👤 People 13

### 📁 ramulator Public

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

● C++ ☆ 583 🍴 209

### 📁 prim-benchmarks Public

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

● C ☆ 137 🍴 50

### 📁 MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 277 🍴 149

### 📁 rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at [http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer\\_isca14.pdf](http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf).

● C ☆ 217 🍴 42

### 📁 SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 127 🍴 28

### 📁 Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (<https://arxiv.org/pdf/2109.12021.pdf>).

● C++ ☆ 117 🍴 36

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

# Funding Acknowledgments

---

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

Thank you!

# Acknowledgments

---

**SAFARI**  
*SAFARI Research Group*  
*safari.ethz.ch*

Think **BIG**, Aim **HIGH!**

<https://safari.ethz.ch>

---

# SAFARI Newsletter June 2023 Edition

- <https://safari.ethz.ch/safari-newsletter-june-2023/>

**SAFARI**  
SAFARI Research Group

Think Big, Aim High



**ETH** zürich

View in your browser

June 2023



# SAFARI Newsletter July 2024 Edition

- <https://safari.ethz.ch/safari-newsletter-july-2024/>



# Recall: DRAM Testing Infrastructure



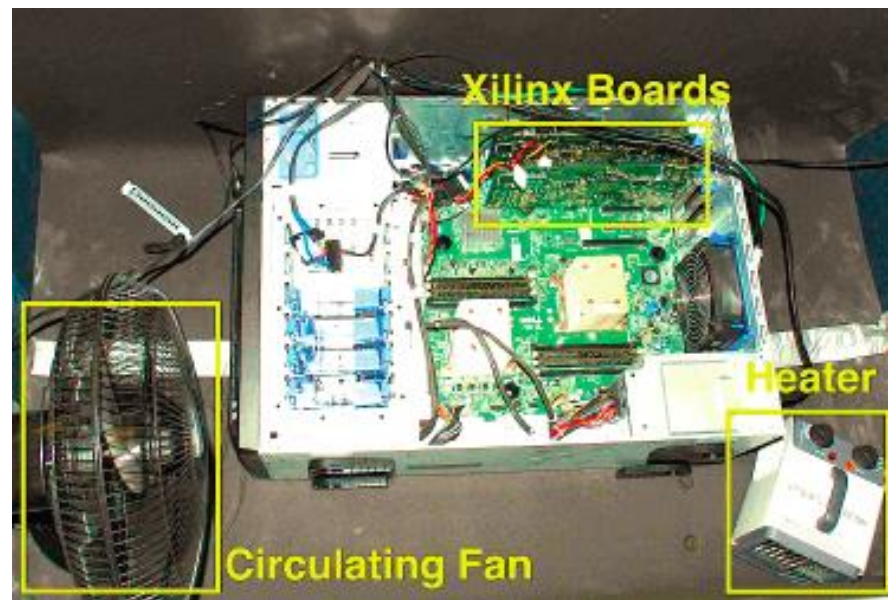
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

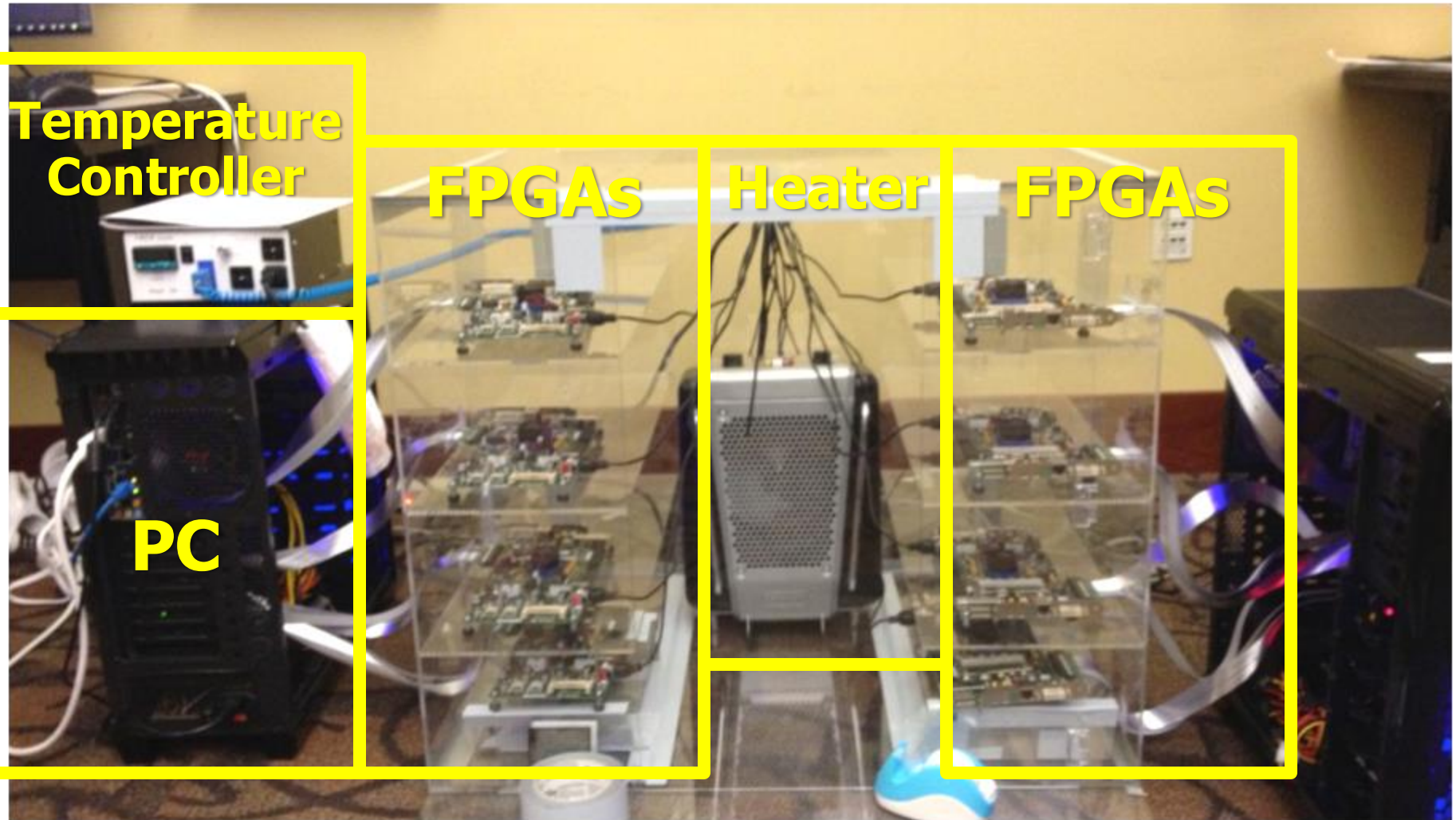
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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)



# Recall: DRAM Testing Infrastructure

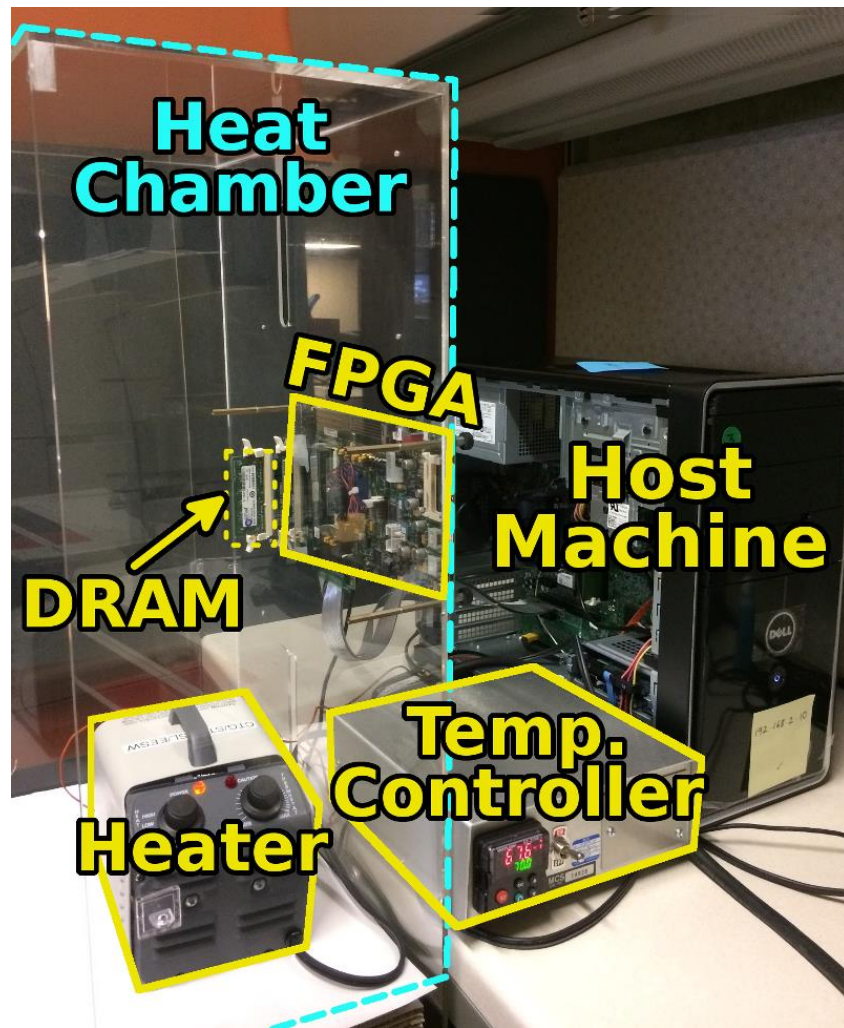




# SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., "[SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies](#)," HPCA 2017.

- Flexible
- Easy to Use (C++ API)
- Open-source  
[github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)



# 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<sup>1,2,3</sup> Nandita Vijaykumar<sup>3</sup> Samira Khan<sup>4,3</sup> Saugata Ghose<sup>3</sup> Kevin Chang<sup>3</sup>  
Gennady Pekhimenko<sup>5,3</sup> Donghyuk Lee<sup>6,3</sup> Oguz Ergin<sup>2</sup> Onur Mutlu<sup>1,3</sup>

<sup>1</sup>*ETH Zürich*   <sup>2</sup>*TOBB University of Economics & Technology*   <sup>3</sup>*Carnegie Mellon University*  
<sup>4</sup>*University of Virginia*   <sup>5</sup>*Microsoft Research*   <sup>6</sup>*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)]

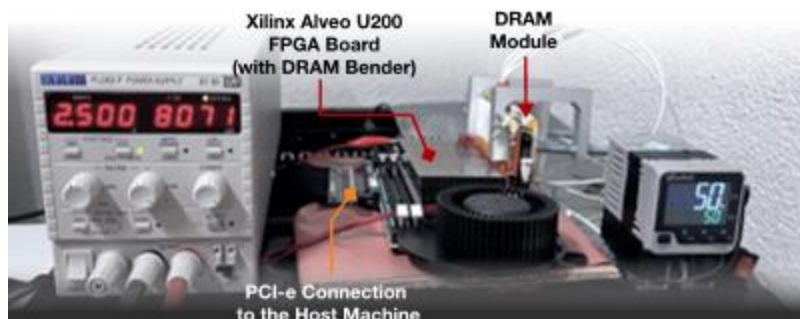
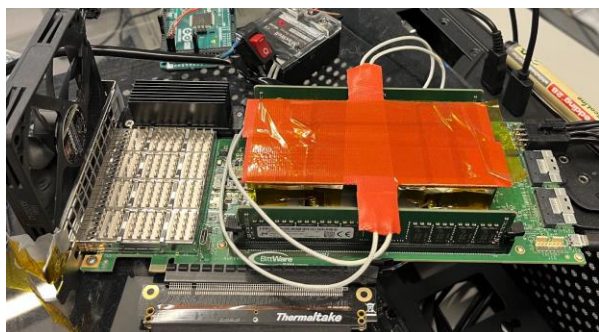
## DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

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

# DRAM Bender: Prototypes

Testing Infrastructure	Protocol Support	FPGA Support
SoftMC [134]	DDR3	One Prototype
LiteX RowHammer Tester (LRT) [17]	DDR3/4, LPDDR4	Two Prototypes
<b>DRAM Bender (this work)</b>	<b>DDR3/DDR4</b>	<b>Five Prototypes</b>

Five out of the box FPGA-based prototypes



# 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*

# DRAM Chips Are Already (Quite) Capable!

---

- <https://arxiv.org/pdf/2312.02880.pdf>

## **PULSAR: Simultaneous Many-Row Activation for Reliable and High-Performance Computing in Off-the-Shelf DRAM Chips**

Ismail Emir Yuksel   Yahya Can Tugrul   F. Nisa Bostanci   Abdullah Giray Yaglikci   Ataberk Olgun  
Geraldo F. Oliveira   Melina Soysal   Haocong Luo   Juan Gomez Luna   Mohammad Sadrosadati  
Onur Mutlu

ETH Zurich

We propose PULSAR, a new technique to enable high-success-rate and high-performance PuM operations in off-the-shelf DRAM chips. PULSAR leverages our new observation that a carefully-crafted sequence of DRAM commands simultaneously activates up to 32 DRAM rows. PULSAR overcomes the limitations of existing techniques by 1) replicating the input data to improve the success rate and 2) enabling new bulk bitwise operations (e.g., many-input majority, *Multi-RowInit*, and *Bulk-Write*) to improve the performance.

# DRAM Chips Are Already (Quite) Capable!

---

- **Appears at DSN 2024**



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

İsmail Emir Yüksel<sup>1</sup>   Yahya Can Tuğrul<sup>1,2</sup>   F. Nisa Bostancı<sup>1</sup>   Geraldo F. Oliveira<sup>1</sup>

A. Giray Yağlıkçı<sup>1</sup>   Ataberk Olgun<sup>1</sup>   Melina Soysal<sup>1</sup>   Haocong Luo<sup>1</sup>

Juan Gómez-Luna<sup>1</sup>   Mohammad Sadrosadati<sup>1</sup>   Onur Mutlu<sup>1</sup>

<sup>1</sup>*ETH Zürich*   <sup>2</sup>*TOBB University of Economics and Technology*

# *Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips*

## *Experimental Characterization and Analysis*



**İsmail Emir Yüksel**

Yahya C. Tuğrul F. Nisa Bostancı Geraldo F. Oliveira

A. Giray Yağlıkçı Ataberk Olgun Melina Soysal Haocong Luo

Juan Gómez-Luna Mohammad Sadr Onur Mutlu

**SAFARI**

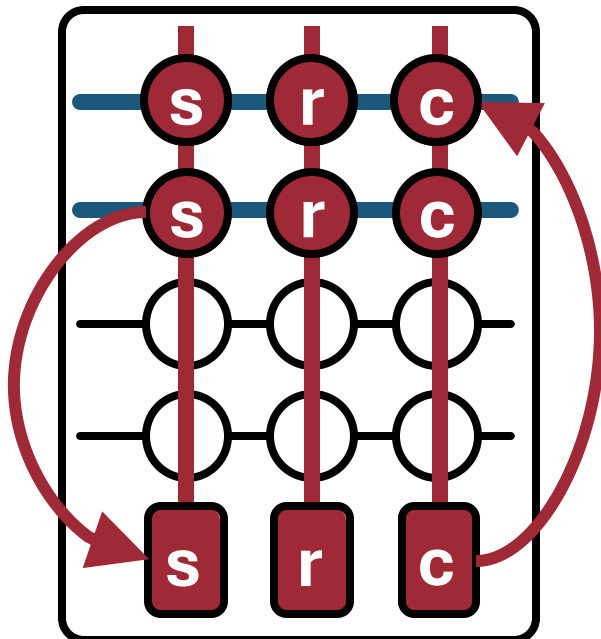
**ETH** zürich



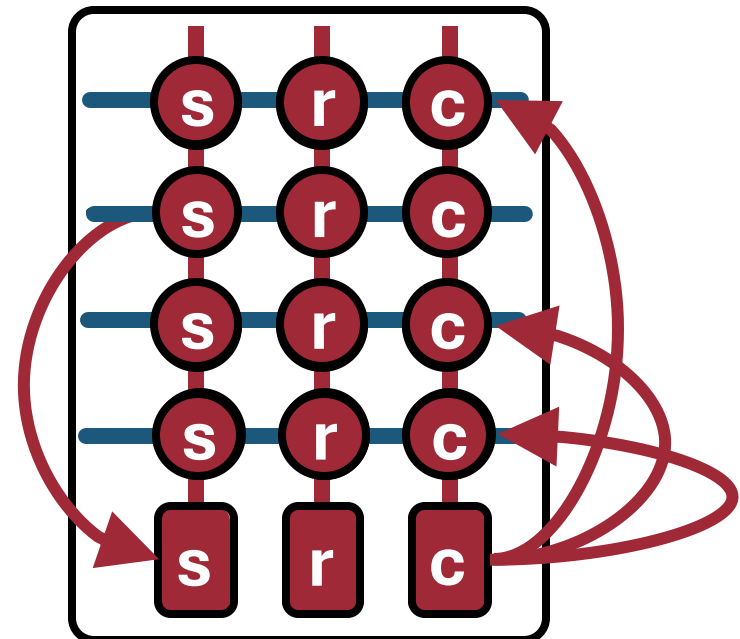
# In-DRAM Multiple Row Copy (Multi-RowCopy)

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

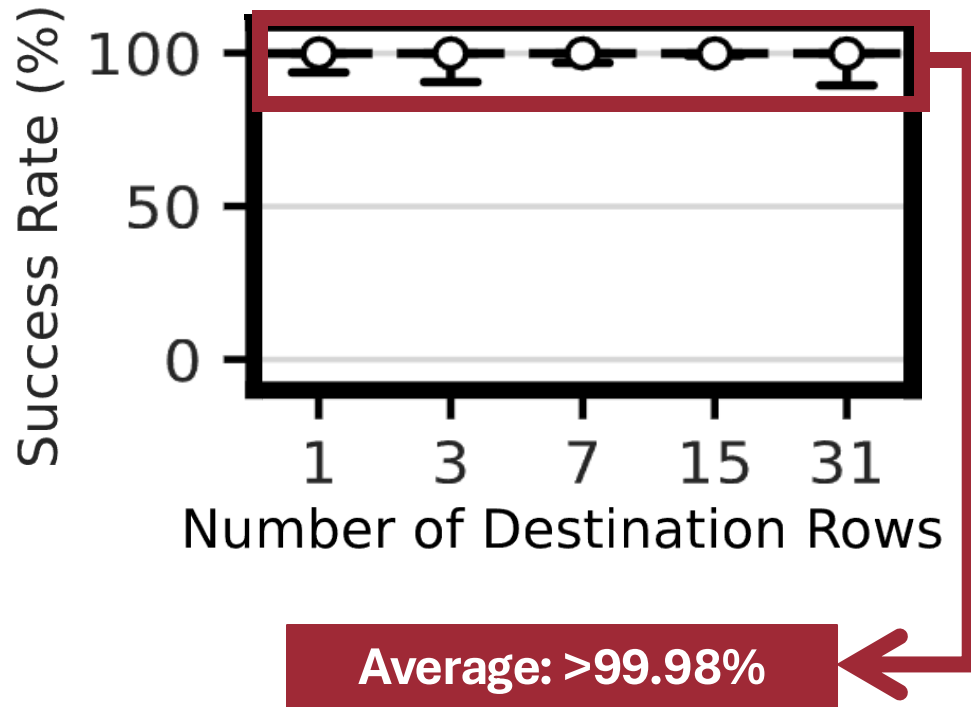
RowClone



Multi-RowCopy

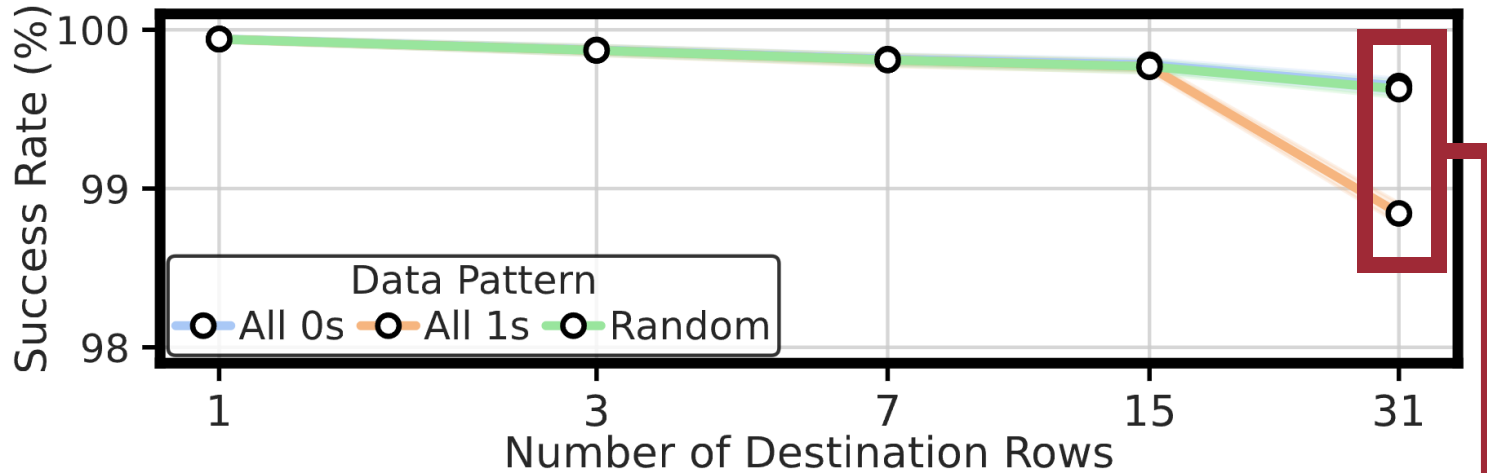


# Robustness of Multi-RowCopy



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

# Impact of Data Pattern



At most 0.79% decrease in average success rate

Data pattern has a small effect on the success rate of the Multi-RowCopy operation



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

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

*We experimentally analyze the computational capability of commercial off-the-shelf (COTS) DRAM chips and the robustness of these capabilities under various timing delays between DRAM commands, data patterns, temperature, and voltage levels. We extensively characterize 120 COTS DDR4 chips from two major manufacturers. We highlight four key results of our study. First, COTS DRAM chips are capable of 1) simultaneously activating up to 32 rows (i.e., simultaneous many-row activation), 2) executing a majority of  $X$  (MAJX) operation where  $X > 3$  (i.e., MAJ5, MAJ7, and MAJ9 operations), and 3) copying a DRAM row (concurrently) to up to 31 other DRAM rows, which we call **MuIti-RowCopy**. Second, storing multiple copies of MAJX's input operands on all simultaneously activated rows drastically increases the success rate (i.e., the percentage of DRAM cells that correctly perform the computation) of the MAJX operation. For example, MAJ3 with 32-row activation (i.e.,*

A subset of PIM proposals devise mechanisms that enable PUM using DRAM cells for computation, including data copy and initialization [67, 72, 77, 78, 89, 104, 127], Boolean logic [56, 64–66, 68, 70, 72, 76, 79, 122, 127–129], majority-based arithmetic [64, 66, 69, 72, 91, 127, 130, 131], and lookup table based operations [82, 106, 107, 132]. We refer to DRAM-based PUM as *Processing-Using-DRAM (PUD)* and the computation performed using DRAM cells as PUD operations.

PUD benefits from the bulk data parallelism in DRAM devices to perform bulk bitwise PUD operations. Prior works show that bulk bitwise operations are used in a wide variety of important applications, including databases and web search [64, 67, 79, 130, 133–140], data analytics [64, 141–144], graph processing [56, 80, 94, 130, 145], genome analysis [60, 99, 146–149], cryptography [150, 151], set operations [56, 64], and hyper-dimensional computing [152–154].

<https://arxiv.org/pdf/2405.06081>

# Our Work is Open Source and Artifact Evaluated

A screenshot of a GitHub repository page for "SiMRA-DRAM". The repository is public and has 4 watchers, 0 forks, and 6 stars. The main branch is "main". The repository contains a README, a LICENSE, and several folders: DRAM-Bender, analysis, and experimental\_data. The README is open, showing the title "Simultaneous Many-Row Activation in Off-the-Shelf DRAM Chips: Experimental Characterization and Analysis". The repository description states: "Source code & scripts for experimental characterization and demonstration of 1) simultaneous many-row activation, 2) up to nine-input majority operations and 3) copying one row's content to up 31 rows in real DDR4 DRAM chips. Described in our DSN'24 paper by Yuksel et al. at <https://arxiv.org/abs/2405.06081>".

File/Folder	Commit Message	Author	Time
DRAM-Bender	initial comit	a51abfa	last month
analysis	initial comit		last month
experimental_data	initial comit		last month
LICENSE	initial comit		last month
README.md	Update README.md		last month

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

# 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, Myungsook 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

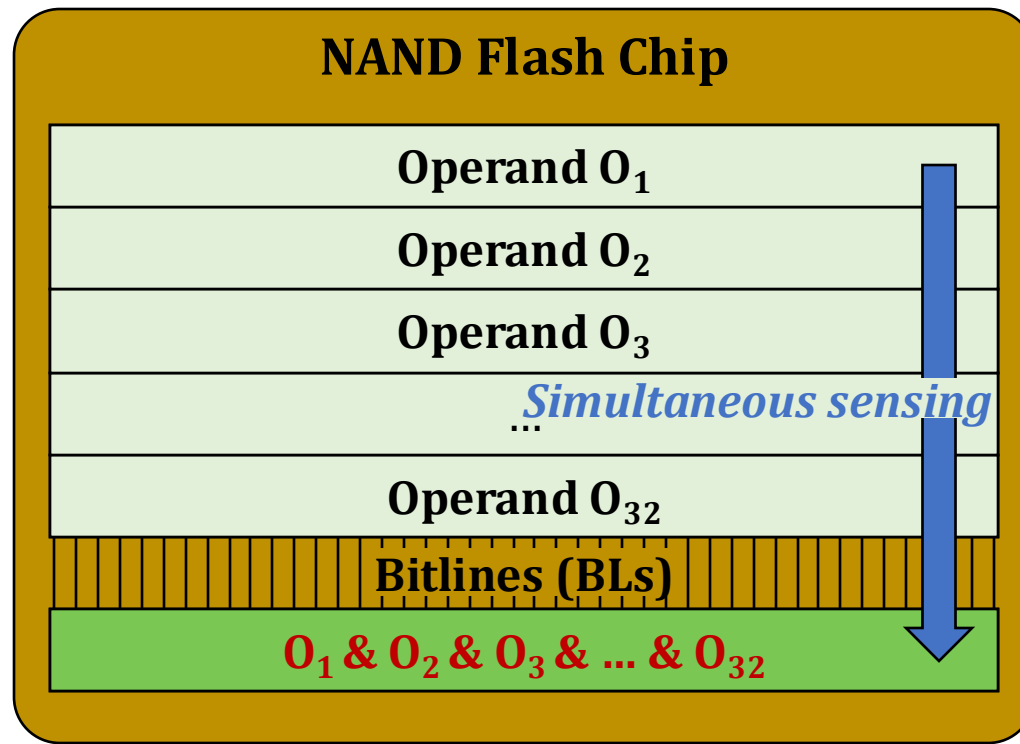
Jisung Park<sup>§∇</sup> Roknoddin Azizi<sup>§</sup> Geraldo F. Oliveira<sup>§</sup> Mohammad Sadrosadati<sup>§</sup>  
Rakesh Nadig<sup>§</sup> David Novo<sup>†</sup> Juan Gómez-Luna<sup>§</sup> Myungsook Kim<sup>‡</sup> Onur Mutlu<sup>§</sup>

<sup>§</sup>ETH Zürich    <sup>∇</sup>POSTECH    <sup>†</sup>LIRMM, Univ. Montpellier, CNRS    <sup>‡</sup>Kyungpook National University

# 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

*Operate  
as a resistance (1)  
or an open switch (0)*

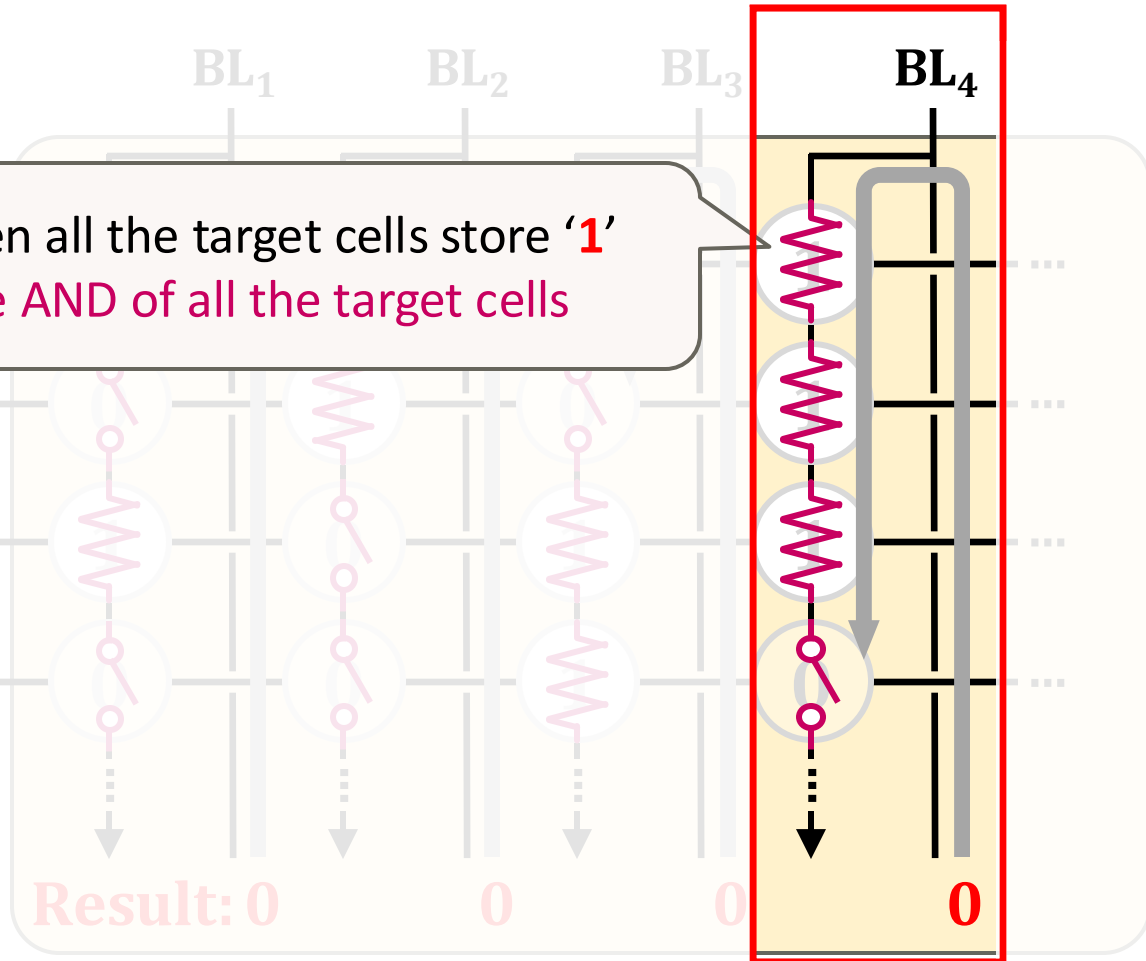
WL<sub>2</sub>  
WL<sub>3</sub>  
WL<sub>4</sub>

Result: 0

0

0

0



# Multi-Wordline Sensing (MWS): Bitwise AND

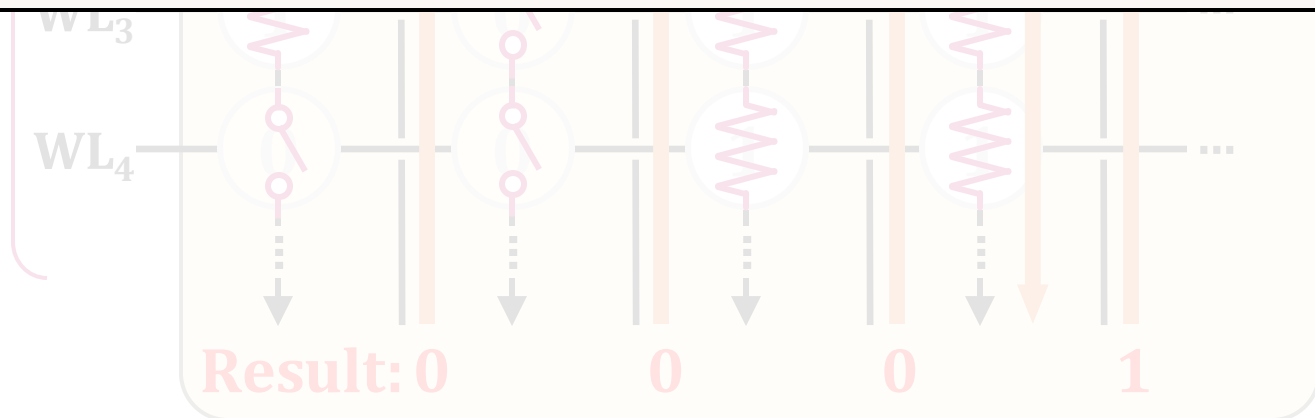
- Intra-Block MWS:

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<sup>§∇</sup> Roknoddin Azizi<sup>§</sup> Geraldo F. Oliveira<sup>§</sup> Mohammad Sadrosadati<sup>§</sup>  
Rakesh Nadig<sup>§</sup> David Novo<sup>†</sup> Juan Gómez-Luna<sup>§</sup> Myungsook Kim<sup>‡</sup> Onur Mutlu<sup>§</sup>

<sup>§</sup>*ETH Zürich*   <sup>∇</sup>*POSTECH*   <sup>†</sup>*LIRMM, Univ. Montpellier, CNRS*   <sup>‡</sup>*Kyungpook National University*



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

# Results: Real-Device Characterization

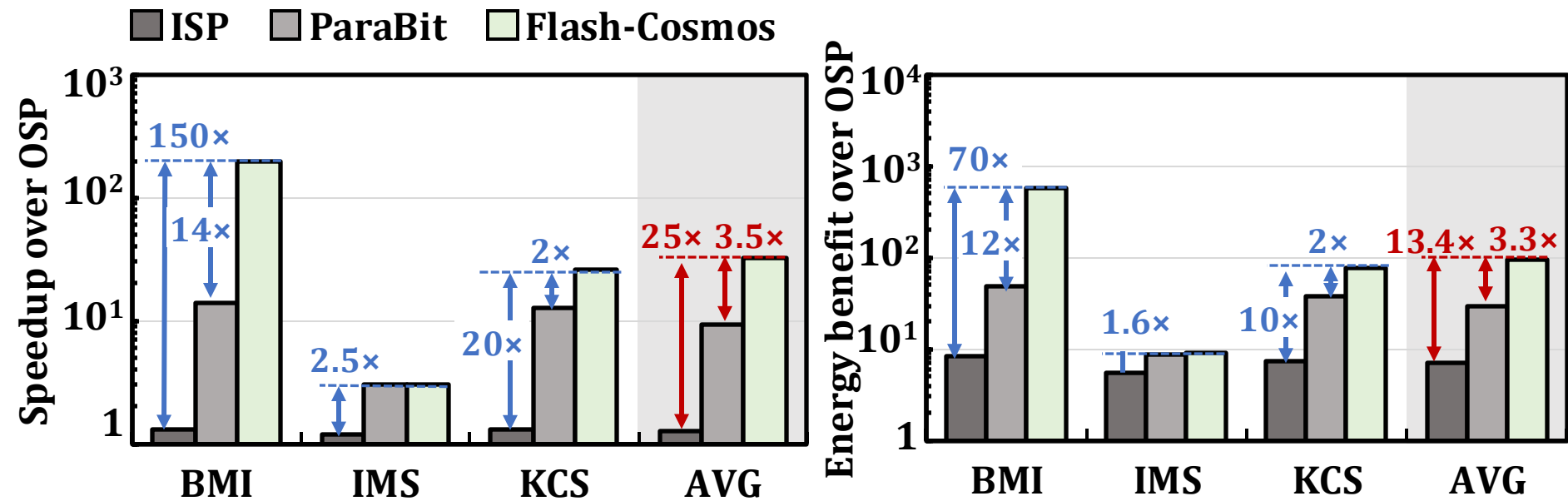
---

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, Myung Suk 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<sup>§∇</sup> Roknoddin Azizi<sup>§</sup> Geraldo F. Oliveira<sup>§</sup> Mohammad Sadrosadati<sup>§</sup>  
Rakesh Nadig<sup>§</sup> David Novo<sup>†</sup> Juan Gómez-Luna<sup>§</sup> Myung Suk Kim<sup>‡</sup> Onur Mutlu<sup>§</sup>

<sup>§</sup>ETH Zürich    <sup>∇</sup>POSTECH    <sup>†</sup>LIRMM, Univ. Montpellier, CNRS    <sup>‡</sup>Kyungpook National University