Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
18 June 2023
Real-World PIM Tutorial Opening Talk @ ISCA-50
The Problem

Computing is Bottlenecked by Data
Data is Key for AI, ML, Genomics, …

- Important workloads are all data intensive

- They require rapid and efficient processing of large amounts of data

- Data is increasing
  - We can generate more than we can process
  - We need to perform more sophisticated analyses on more data
Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks

Source: https://youtu.be/Bh13Idwcb0Q?t=283
Data is Key for Future Workloads

**In-memory Databases**
[Mao+, EuroSys’12; Clapp+ (Intel), IISWC’15]

**Graph/Tree Processing**
[Xu+, IISWC’12; Umuroglu+, FPL’15]

**In-Memory Data Analytics**
[Clapp+ (Intel), IISWC’15; Awan+, BDCloud’15]

**Datacenter Workloads**
[Kanev+ (Google), ISCA’15]
Data Overwhelms Modern Machines

In-memory Databases

Graph/Tree Processing

Data → performance & energy bottleneck

In-Memory Data Analytics
[Clapp+ (Intel), IISWC’15; Awan+, BDCloud’15]

Datacenter Workloads
[Kanev+ (Google), ISCA’15]
Data is Key for Future Workloads

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
Video Playback
Google’s video codec

VP9
Video Capture
Google’s video codec
Data Overwhelms Modern Machines

Chrome

TensorFlow Mobile

Data $\rightarrow$ performance & energy bottleneck

VP9 $\rightarrow$ YouTube

Video Playback

Google’s video codec

Video Capture

Google’s video codec
Data is Key for Future Workloads

Development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Genome Analysis

1 Sequencing

Data → performance & energy bottleneck

2 Read Mapping

3 Variant Calling

4 Scientific Discovery
We Need Faster & Scalable Genome Analysis

Understanding *genetic variations, species, evolution, ...*

Predicting the *presence and relative abundance of microbes* in a sample

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

SAFARI

And, many, many other applications ...
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

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

Published: 02 April 2018  Article history ▼

New Genome Sequencing Technologies

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

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

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

*Published: 02 April 2018  Article history ▼*

Oxford Nanopore MinION

Data → performance & energy bottleneck
Problems with (Genome) Analysis Today

Special-Purpose Machine for Data Generation

General-Purpose Machine for Data Analysis

FAST  SLOW

Slow and inefficient processing capability
Large amounts of data movement

SAFARI  This picture is similar for many “data generators & analyzers” today
Onur Mutlu and Can Firtina, "Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
[arXiv version]
Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]
Beginner Reading on Genome Analysis

Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu

“From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”
Computational and Structural Biotechnology Journal, 2022
[Source code]

Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

Mohammed Alser *, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu *

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

FPGA-based Near-Memory Analytics

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu,
  "FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"

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

Gagandeep Singh\(^\diamond\)  Mohammed Alser\(^\diamond\)  Damla Senol Cali\(^\times\)
Dionysios Diamantopoulos\(^\downarrow\)  Juan Gómez-Luna\(^\diamond\)
Henk Corporaal\(^*\)  Onur Mutlu\(^\diamond\,\times\)

\(^\diamond\)ETH Zürich  \(^\times\)Carnegie Mellon University
\(^*\)Eindhoven University of Technology  \(^\downarrow\)IBM Research Europe
Near-Memory Acceleration using FPGAs

IBM POWER9 CPU

HBM-based FPGA board

Near-HBM FPGA-based accelerator

Two communication technologies: CAPI2 and OCAPI
Two memory technologies: DDR4 and HBM
Two workloads: Weather Modeling and Genome Analysis
Performance & Energy Greatly Improve

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

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

HBM alleviates memory bandwidth contention vs. DDR4
GenASM Framework [MICRO 2020]


[Lighting Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]

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


†Carnegie Mellon University  ▼Processor Architecture Research Lab, Intel Labs  ▼Bilkent University  ▼ETH Zürich  ▼Facebook  ©King Mongkut’s University of Technology North Bangkok  ▼University of Illinois at Urbana–Champaign
Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs

Joël Lindegger, Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Nika Mansouri Ghiasi, and Onur Mutlu,
"Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs"
[Online link at Bioinformatics Journal]
[arXiv preprint]
[Scrooge Source Code]

Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs

Joël Lindegger$\  
Juan Gómez-Luna$\  
Damla Senol Cali†  
Nika Mansouri Ghiasi$  
Mohammed Alser$\  
Onur Mutlu$\  

$\  ETH Zürich  
†Bionano Genomics

In-Storage Genome 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"


[Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Talk Video (17 minutes)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi\textsuperscript{1} Jisung Park\textsuperscript{1} Harun Mustafa\textsuperscript{1} Jeremie Kim\textsuperscript{1} Ataberk Olgun\textsuperscript{1} Arvid Gollwitzer\textsuperscript{1} Damla Senol Cali\textsuperscript{2} Can Firtina\textsuperscript{1} Haiyu Mao\textsuperscript{1} Nour Almadhoun Alserr\textsuperscript{1} Rachata Ausavarungnirun\textsuperscript{3} Nandita Vijaykumar\textsuperscript{4} Mohammed Alser\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \textsuperscript{2}Bionano Genomics \textsuperscript{3}KMUTNB \textsuperscript{4}University of Toronto
Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
[arXiv version]

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

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Züalal Bingöl³ Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim² Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr² Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

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

Accelerating Basecalling + Read Mapping

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

- Slides (pptx) (pdf)
- Longer Lecture Slides (pptx) (pdf)
- Lecture Video (25 minutes)
- arXiv version

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

Haiyu Mao¹ Mohammed Alser¹ Mohammad Sadrosadati¹ Can Firtina¹ Akanksha Baranwal¹
Damla Senol Cali² Aditya Manglik¹ Nour Almadhoun Alserr¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics

A Framework for Designing Efficient Deep Learning-Based Genomic Basecallers

Gagandeep Singh\textsuperscript{a}  Mohammed Alser\textsuperscript{*a} Alireza Khodamoradi\textsuperscript{*b}
Kristof Denolf\textsuperscript{b} Can Firtina\textsuperscript{a} Meryem Banu Cavlak\textsuperscript{a}
Henk Corporaal\textsuperscript{c} Onur Mutlu\textsuperscript{a}

\textsuperscript{a}ETH Zürich \quad \textsuperscript{b}AMD \quad \textsuperscript{c}Eindhoven University of Technology

Nanopore sequencing is a widely-used high-throughput genome sequencing technology that can sequence long fragments of a genome. Nanopore sequencing generates noisy electrical signals that need to be converted into a standard string of DNA nucleotide bases (i.e., A, C, G, T) using a computational step called basecalling. The accuracy and speed of basecalling have critical implications for every subsequent step in genome analysis. Currently, basecallers are developed mainly based on deep learning techniques to provide high sequencing accuracy without considering the compute demands of such tools. We observe that state-of-the-art basecallers (i.e., Guppy, Bonito, Fast-Bonito) are slow, inefficient, and memory-hungry
Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
Invited Lecture at Technion, Virtual, 26 January 2021.
[Slides (pptx) (pdf)]
[Talk Video (1 hour 37 minutes, including Q&A)]
[Related Invited Paper (at IEEE Micro, 2020)]
More on Fast & Efficient Genome Analysis …
Detailed Lectures on Genome Analysis

- Computer Architecture, Fall 2020, Lecture 3a
  - Introduction to Genome Sequence Analysis (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5

- Computer Architecture, Fall 2020, Lecture 8
  - Intelligent Genome Analysis (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14

- Computer Architecture, Fall 2020, Lecture 9a
  - GenASM: Approx. String Matching Accelerator (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - Accelerating Genomics (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId

https://www.youtube.com/onurmutlulectures
Genomics Course (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3T1&list=PL5Q2soXY2Zi8NqPDqOR1yRU_Cxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3T1&list=PL5Q2soXY2Zi8NqPDqOR1yRU_Cxjw-u18)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
BIO-Arch Workshop at RECOMB 2023

- April 14, 2023

BIO-Arch: Workshop on Hardware Acceleration of Bioinformatics Workloads

About

BIO-Arch is a new forum for presenting and discussing new ideas in accelerating bioinformatics workloads with the co-design of hardware & software and the use of new computer architectures. Our goal is to discuss new system designs tailored for bioinformatics. BIO-Arch aims to bring together researchers in the bioinformatics, computational biology, and computer architecture communities to strengthen the progress in accelerating bioinformatics analysis (e.g., genome analysis) with efficient system designs that include hardware acceleration and software systems tailored for new hardware technologies.

Venue

BIO-Arch will be held in The Social Facilities of Istanbul Technical University on April 14. Detailed information about how to arrive at the venue location with various transportation options can be found on the RECOMB website.

Our panel discussion will be held in conjunction with the main RECOMB conference. The panel discussion will be held in Marriott Şişli on April 17 at 17:00. You can find

https://www.youtube.com/watch?v=2rCsb4-nLmg

https://safari.ethz.ch/recomb23-arch-workshop/
Data Overwhelms Modern Machines …

- Storage/memory capability
- Communication capability
- Computation capability

Greatly impacts robustness, energy, performance, cost
A Computing System

- Three key components
  - Computation
  - Communication
  - Storage/memory

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.

Perils of Processor-Centric Design

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

Yet, system is still bottlenecked by memory & storage.
Deeper and Larger Memory Hierarchies

Core Count:
8 cores/16 threads

L1 Caches:
32 KB per core

L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020

AMD’s 3D Last Level Cache (2021)

AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

Additional 64 MB L3 cache die stacked on top of the processor die
- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache
Deeper and Larger Memory Hierarchies

IBM POWER10, 2020

Cores:
15-16 cores, 8 threads/core

L2 Caches:
2 MB per core

L3 Cache:
120 MB shared

Deeper and Larger Memory Hierarchies

A lot of SRAM

Storage

DRAM

DRAM

Storage

Apple M1 Ultra System (2022)
Data Overwhelms Modern Machines

Google's web browser

Google's machine learning framework

Data → performance & energy bottleneck

Video Playback

Google's video codec

Video Capture

Google's video codec

VP9

YouTube

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand\textsuperscript{1} Rachata Ausavarungnirun\textsuperscript{1} Saugata Ghose\textsuperscript{1} Youngsok Kim\textsuperscript{2} Eric Shiu\textsuperscript{3} Rahul Thakur\textsuperscript{3} Daehyun Kim\textsuperscript{4,3} Parthasarathy Ranganathan\textsuperscript{3} Onur Mutlu\textsuperscript{5,1}

SAFARI
Data Movement Overwhelms Accelerators

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,

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

Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.

[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]

> 90% of the total system energy is spent on memory in large ML models
An Intelligent Architecture
Handles Data Well
How to Handle Data Well

- **Ensure data does not overwhelm the components**
  - via intelligent algorithms, architectures & system designs: algorithm-architecture-devices

- **Take advantage of vast amounts of data and metadata**
  - to improve architectural & system-level decisions

- **Understand and exploit properties of (different) data**
  - to improve algorithms & architectures in various metrics
Corollaries: Computing Systems Today …

- Are processor-centric vs. data-centric

- Make designer-dictated decisions vs. data-driven

- Make component-based myopic decisions vs. data-aware
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Revisit the Entire Stack

We can get there step by step
A Blueprint for Fundamentally Better Architectures

Onur Mutlu,
"Intelligent Architectures for Intelligent Computing Systems"
[Slides (pptx) (pdf)]
[IEDM Tutorial Slides (pptx) (pdf)]
[Short DATE Talk Video (11 minutes)]
[Longer IEDM Tutorial Video (1 hr 51 minutes)]

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
Data-Centric (Memory-Centric) Architectures
Data-Centric Architectures: Properties

- **Process data where it resides** *(where it makes sense)*
  - Processing in and near memory & sensor structures

- **Low-latency & low-energy data access**

- **Low-cost data storage & processing**
  - High capacity memory at low cost: hybrid memory, compression

- **Intelligent data management**
  - Intelligent controllers handling robustness, security, cost, perf.
Processing Data

Where It Makes Sense
Process Data Where It Makes Sense
Processing in/near Memory: An Old Idea


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.

CELL EQUATIONS: \[ \hat{x} = \overline{w}x + wy \]
[\[ s_y = wcx, r_y = wc\overline{x} \]
[\[ \hat{z} = M(x, \overline{y}, z) = x\overline{y} + z(x + \overline{y}) \]

Fig. 1. Cellular sorting array I.
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 problems with Memory Technology**
  - Memory technology scaling is not going well (e.g., RowHammer)
  - Many scaling issues demand intelligence in memory

- **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

- **Designs are squeezed in the middle**
Processing-in-Memory Landscape Today

And, many other experimental chips and startups
Memory Scaling Issues Are Real

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

Memory Scaling: A Systems Architecture Perspective

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

One can predictably induce errors in most DRAM memory chips.

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

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

A company
86% (37/43)
Up to $1.0 \times 10^7$ errors

B company
83% (45/54)
Up to $2.7 \times 10^6$ errors

C company
88% (28/32)
Up to $3.3 \times 10^5$ errors

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

A simple hardware failure mechanism can create a widespread system security vulnerability

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS
Memory Scaling Issues Are Real

  - [arXiv version]
  - [Slides (pptx) (pdf)]
  - [Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu  
onur.mutlu@safari.ethz.ch  
ETH Zürich  
Zürich, Switzerland

Ataberk Olgun  
ataberk.olgun@safari.ethz.ch  
ETH Zürich  
Zürich, Switzerland

A. Giray Yağlıkçı  
giray.yaglikci@safari.ethz.ch  
ETH Zürich  
Zürich, Switzerland

[SAFARI]  
Are We Now RowHammer Free?

- To Appear in ISCA 2023

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

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

ETH Zürich
Latest RowHammer Lecture

Collapse of the “Galloping Gertie” (1940)

Source: AP
http://www.wsdot.wa.gov/tnshistory/connections/connections3.htm

Seminar in Computer Architecture - Lecture 4: RowHammer (Spring 2023)

408 views 2 months ago 
Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2023)
Seminar in Computer Architecture, ETH Zürich, Spring 2023 (https://safari.ethz.ch/architecture_s...)

https://www.youtube.com/watch?v=e6G_Vbrqr_c
Onur Mutlu,
"Security Aspects of DRAM: The Story of RowHammer"


[Slides (pptx)(pdf)]
[Tutorial Video (57 minutes)]
10 Years of RowHammer in 20 Minutes

Onur Mutlu, "The Story of RowHammer"

Invited Talk at the Workshop on Robust and Safe Software 2.0 (RSS2), held with the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, 28 February 2022.

[Slides (pptx) (pdf)]

https://www.youtube.com/watch?v=ctKTRyi96Bk
The Push from Circuits and Devices

Main Memory Needs

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

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

SK hynix Semiconductor, Icheon, Korea
Industry’s Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

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

Seungki Hong  Dongha Kim  Jaehyung Lee  Reum Oh
Changskik Yoo  Sangjoon Hwang  Jooyoungh Lee

DRAM Design Team, Memory Division, Samsung Electronics

Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger,
  "Architecting Phase Change Memory as a Scalable DRAM Alternative"
  One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

Architecting Phase Change Memory as a Scalable DRAM Alternative

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

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

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

Intelligent Memory Controllers Can Avoid Many Failures & Enable Better Scaling
Three Key Systems & Application Trends

1. **Data access** is the major bottleneck
   - Applications are increasingly data hungry

2. **Energy** consumption is a key limiter

3. **Data movement energy** dominates compute
   - Especially true for off-chip to on-chip movement
Do We Want This?

Source: V. Milutinovic
Or This?

Source: V. Milutinovic
Challenge and Opportunity for Future

High Performance,
Energy Efficient,
Sustainable
(All at the Same Time)
The Problem

Data access is the major performance and energy bottleneck

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

Processing of data is performed far away from the data.
Today’s Computing Systems

- Processor centric

- All data processed in the processor \(\rightarrow\) at great system cost
It’s the Memory, Stupid!

“It’s the Memory, Stupid!” (Richard Sites, MPR, 1996)

Richard Sites

It’s the Memory, Stupid!
When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.

5, 1996 Microprocessor Report

I expect that over the coming decade memory subsystem design will be the only important design issue for microprocessors.

The Performance Perspective

The Performance Perspective

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"
  

  One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu §  Jared Stark †  Chris Wilkerson ‡  Yale N. Patt §

§ECE Department  †Microprocessor Research  ‡Desktop Platforms Group
The University of Texas at Austin  Intel Labs  Intel Corporation
{onur,patt}@ece.utexas.edu  jared.w.stark@intel.com  chris.wilkerson@intel.com
The Performance Perspective (Today)

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

\[
\text{Pipeline slot breakdown (\%)}
\]

Data Movement vs. Computation Energy

Communication Dominates Arithmetic

Dally, HiPEAC 2015

A memory access consumes $\sim 100\text{-}1000\times$ the energy of a complex addition.
Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)

- **ADD (int)**
- **ADD (float)**
- **Register File**
- **MULT (int)**
- **MULT (float)**
- **SRAM Cache**
- **DRAM**

Energy (pJ)

- ADD (int) Relative Cost

A memory access consumes 6400X the energy of a simple integer addition.
Energy Waste in Mobile Devices


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand\textsuperscript{1} \quad Saugata Ghose\textsuperscript{1} \quad Youngsok Kim\textsuperscript{2} \\
Rachata Ausavarungnirun\textsuperscript{1} \quad Eric Shiu\textsuperscript{3} \quad Rahul Thakur\textsuperscript{3} \\
Aki Kuusela\textsuperscript{3} \quad Allan Knies\textsuperscript{3} \quad Parthasarathy Ranganathan\textsuperscript{3} \\
Daehyun Kim\textsuperscript{4,3} \quad Onur Mutlu\textsuperscript{5,1}

SAFARI
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
We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015

A memory access consumes $\sim 100\text{-}1000X$ the energy of a complex addition
We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015

A memory access consumes ~100-1000X the energy of a complex addition
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*
Many questions ... How do we design the:

- compute-capable memory & controllers?
- processors & communication units?
- software & hardware interfaces?
- system software, compilers, languages?
- algorithms & theoretical foundations?
Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}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"

PIM Course (Fall 2022)

**Fall 2022 Edition:**
- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

**Spring 2022 Edition:**
- https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

**Youtube Livestream (Fall 2022):**
- https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0G0BrnK

**Youtube Livestream (Spring 2022):**
- https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYUK9EsXKhQKRPyX

**Project course**
- Taken by Bachelor’s/Master’s students
- Processing-in-Memory lectures
- Hands-on research exploration
- Many research readings

https://www.youtube.com/onurmutlulectures
SSD Course (Spring 2023)

Spring 2023 Edition:

Fall 2022 Edition:
- https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssds

Youtube Livestream (Spring 2023):
- https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB

Youtube Livestream (Fall 2022):
- https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4J15bwhAMpAp13&pp=iAQB

Project course
- Taken by Bachelor’s/Master’s students
- SSD Basics and Advanced Topics
- Hands-on research exploration
- Many research readings

https://www.youtube.com/onurmutlulectures
Genomics Course (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV](https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NpOr1yRUTxjw-u18](https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NpOr1yRUTxjw-u18)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Current Real PIM Tutorial (ISCA 2023)

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

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

https://events.safari.ethz.ch/isca-pim-tutorial/
We Need to Think Differently from the Past Approaches
Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory
Mindset: Memory as an Accelerator

Memory similar to a “conventional” accelerator
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

1) Low latency
2) Low bandwidth utilization
3) No cache pollution
4) No unwanted data movement

1046ns, 3.6μJ  →  90ns, 0.04μJ
RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates
Negligible HW cost

Step 1: Activate row A
Transfer row

Step 2: Activate row B
Transfer row

DRAM subarray

Row Buffer (4 Kbytes)

4 Kbytes

8 bits

Data Bus
RowClone: Latency and Energy Savings

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"

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]
RowClone in Off-the-Shelf DRAM Chips

- Idea: Violate DRAM timing parameters to mimic RowClone

ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAMs

Fei Gao  
feig@princeton.edu  
Department of Electrical Engineering  
Princeton University

Georgios Tziantzioulis  
georgios.tziantzioulis@princeton.edu  
Department of Electrical Engineering  
Princeton University

David Wentzlaff  
wentzlaf@princeton.edu  
Department of Electrical Engineering  
Princeton University

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$\dagger$ Juan Gómez Luna$\S$ Hasan Hassan$\S$ Konstantinos Kanellopoulos$\S$ Behzad Salami$\S\ast$

$\S$ETH Zürich $\S\dagger$TOBB ETÜ $\S$BSC

https://github.com/cmu-safari/pidram
https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s
Real Processing-using-Memory Prototype

https://github.com/cmu-safari/pidram
https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s
Real Processing-using-Memory Prototype

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-zyq` is a repository branched off of UCB-BAR's `fpga-zyq` repository. We use `fpga-zyq` 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-zyq` and read the README file to understand the overall workflow of the repository
   - Follow the readme in `fpga-zyq/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-zyq/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-zyq/README.md`

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

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx Phv 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://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
Lecture on RowClone & Processing using DRAM

Mindset: Memory as an Accelerator

- CPU core
- CPU core
- mini-CPU core
- video core
- imaging core
- GPU (throughput) core
- GPU (throughput) core
- GPU (throughput) core
- Memory

Specialized compute-capability in memory

Memory similar to a "conventional" accelerator

DEPARTMENT OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING (D-ITE)
Seminar in Computer Arch. - Meeting 3: RowClone: In-Memory Data Copy and Initialization (Fall 2021)

292 views • Streamed live on Oct 7, 2021

https://www.youtube.com/watch?v=n6Pwg1qax_E&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4&index=4
(Truly) In-Memory Computation

- We can support in-DRAM AND, OR, NOT, MAJ
- At low cost
- Using analog computation capability of DRAM
  - Idea: activating multiple rows performs computation
- 30-60X performance and energy improvement

- New memory technologies enable even more opportunities
  - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
  - Can operate on data with minimal movement
In-DRAM AND/OR: Triple Row Activation

Final State

\[ AB + BC + AC \]

\[ C(A + B) + \sim C(AB) \]
Bulk Bitwise Operations in Workloads

- Bitmap indices (database indexing)
- Set operations
- Encryption algorithms
- BitWeaving (database queries)
- BitFunnel (web search)
- DNA sequence mapping...

[1] Li and Patel, BitWeaving, SIGMOD 2013
In-DRAM Acceleration of Database Queries


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

>4-12X Performance Improvement
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)]
In-DRAM Bulk Bitwise Execution


In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri  
Microsoft Research India  
visesha@microsoft.com

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


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

\textsuperscript{1}ETH Zürich \quad \textsuperscript{2}Simon Fraser University \quad \textsuperscript{3}University of Illinois at Urbana–Champaign
SIMDRAM Framework: Overview

Step 1: Generate MAJ logic

Desired operation

AND/OR/NOT logic

User Input

Step 2: Generate sequence of DRAM commands

MAJ

Step 3: Execution according to µProgram

SIMDRAM Output

New SIMDREAM µProgram

µProgram

Main memory

SIMDRAM-enabled application

foo () {
  bbop_new
}

User Input

SIMDRAM output

Instruction result in memory

Control Unit

µProgram

Memory Controller

SAFARI
SIMDRAM Key Results

Evaluated on:
- 16 complex in-DRAM operations
- 7 commonly-used real-world applications

SIMDRAM provides:

- 88× and 5.8× the throughput of a CPU and a high-end GPU, respectively, over 16 operations
- 257× and 31× the energy efficiency of a CPU and a high-end GPU, respectively, over 16 operations
- 21× and 2.1× the performance of a CPU and a high-end GPU, over seven real-world applications

SAFARI
More on SIMDRAM


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

*Nastaran Hajinazar¹,²
Nika Mansouri Ghiasi¹

*Geraldo F. Oliveira¹
Minesh Patel¹
Juan Gómez-Luna¹

Sven Gregorio¹
Mohammed Alser¹
Onur Mutlu¹
João Dinis Ferreira¹
Saugata Ghose³

¹ETH Zürich ²Simon Fraser University ³University of Illinois at Urbana–Champaign
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§
Lois Orosa$^

Gabriel Falcao†
Mohammad Sadrosadati§
Taha Shahroodi‡

Juan Gómez-Luna§
Jeremie S. Kim§
Anant Nori*
Onur Mutlu§

Mohammed Alser§
Geraldo F. Oliveira§

§ETH Zürich †IT, University of Coimbra
‡TU Delft

$^\text{ETH Zürich} †^{\text{IT, University of Coimbra}} \n
\text{Source Code (Officially Artifact Evaluated with All Badges)}

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"


[Lightning Talk Video]
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
[Full Talk Lecture Video (28 minutes)]

---

The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions
by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim†§, Minesh Patel§, Hasan Hassan§, Onur Mutlu§†
†Carnegie Mellon University §ETH Zürich

SAFARI
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"

[Slides (pptx) (pdf)]
[Full Talk Video (21 minutes)]
[Full Talk Lecture Video (27 minutes)]
Top Picks Honorable Mention by IEEE Micro.

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

Jeremie S. Kim‡$, Minesh Patel$, Hasan Hassan$, Lois Orosa$, Onur Mutlu$

‡Carnegie Mellon University $ETH Zürich
In-DRAM True Random Number Generation

- Ataberk Olgun, Minesh Patel, A. Giray Yağlıkçı, Haocong Luo, Jeremie S. Kim, F. Nisa Bostancı, Nandita Vijaykumar, Oğuz Ergin, and Onur Mutlu,

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

[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Talk Video (25 minutes)]
[SAFARI Live Seminar Video (1 hr 26 mins)]

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

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

§ETH Zürich †TOBB University of Economics and Technology ©University of Toronto

SAFARI
In-DRAM True Random Number Generation

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

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

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

†TOBB University of Economics and Technology  §ETH Zürich

In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungskuk Kim, and Onur Mutlu,

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

Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (44 minutes)]
[arXiv version]

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

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

§ETH Zürich ▼POSTECH †LIRMM, Univ. Montpellier, CNRS ‡Kyungpook National University

Summary: Flash-Cosmos

The first work that enables in-flash multi-operand bulk bitwise operations with a single sensing operation and high reliability.

Improves performance by 32x/25x/3.5x over OSP/ISP/ParaBit.

Improves energy efficiency by 95x/13.4x/3.3x over OSP/ISP/ParaBit.

Low-cost & requires no changes to flash cell arrays.
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

![Flash-Cosmos Diagram]

NAND Flash Chip

Operand $O_1$
Operand $O_2$
Operand $O_3$
...
Operand $O_{32}$

Bitlines (BLs)

$O_1 \& O_2 \& O_3 \& \ldots \& O_{32}$
Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories

Shuangchen Li\textsuperscript{1,*}, Cong Xu\textsuperscript{2}, Qiaosha Zou\textsuperscript{1,5}, Jishen Zhao\textsuperscript{3}, Yu Lu\textsuperscript{4}, and Yuan Xie\textsuperscript{1}

University of California, Santa Barbara\textsuperscript{1}, Hewlett Packard Labs\textsuperscript{2}
University of California, Santa Cruz\textsuperscript{3}, Qualcomm Inc.\textsuperscript{4}, Huawei Technologies Inc.\textsuperscript{5}
{shuangchenli, yuanxie}@ece.ucsb.edu\textsuperscript{1}
Other Readings on Processing using NVM


Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory
Mindset: Memory as an Accelerator

Memory similar to a “conventional” accelerator
Accelerating In-Memory Graph Analytics

- Large graphs are everywhere (circa 2015)
  - 36 Million Wikipedia Pages
  - 1.4 Billion Facebook Users
  - 300 Million Twitter Users
  - 30 Billion Instagram Photos

- Scalable large-scale graph processing is challenging

![Speedup Chart]

- 32 Cores
- 128...

Speedup: +42%
Key Bottlenecks in Graph Processing

```java
for (v: graph.vertices) {
    for (w: v.successors) {
        w.next_rank += weight * v.rank;
    }
}
```

1. Frequent random memory accesses
2. Little amount of computation
Opportunity: 3D-Stacked Logic+Memory

Other “True 3D” technologies under development
**Tesseract System for Graph Processing**

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

---

**Host Processor**

Memory-Mapped Accelerator Interface (Noncacheable, Physically Addressed)

---

**Memory**

**Logic**

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

**Crossbar Network**

**In-Order Core**

- LP
- PF Buffer
- MTP
- Message Queue

**DRAM Controller**

**NI**

---

**SAFARI** Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface
Noncacheable, Physically Addressed)

Logic
Memory

Crossbar Network

In-Order Core

Communications via Remote Function Calls
Message Queue

SAFARI
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface
(Noncacheable, Physically Addressed)

Crossbar Network

Prefetching
LP
PF Buffer
MTP
Message Queue

Memory
Logic

DRAM Controller
NI
Evaluated Systems

**DDR3-OoO**
- 8 OoO 4GHz

**HMC-OoO**
- 8 OoO 4GHz

**HMC-MC**
- 128 In-Order 2GHz

**Tesseract**
- 32 Tesseract Cores

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract Graph Processing Performance

On five graph processing algorithms

- DDR3-OoO
- HMC-OoO
- HMC-MC
- Tesseract
- Tesseract-LP
- Tesseract-LP-MTP

>13X Performance Improvement

Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.
Tesseract Graph Processing System Energy

Memory Layers | Logic Layers | Cores

HMC-OoO | Tesseract with Prefetching

> 8X Energy Reduction

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
  "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  Top Picks Honorable Mention by IEEE Micro.

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

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

SAFARI
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"


[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi\(^1\)  Jisung Park\(^1\)  Harun Mustafa\(^1\)  Jeremie Kim\(^1\)  Ataberk Olgun\(^1\)
Arvid Gollwitzer\(^1\)  Damla Senol Cali\(^2\)  Can Firtina\(^1\)  Haiyu Mao\(^1\)  Nour Almadhoun Alserr\(^1\)
Rachata Ausavarungnirun\(^3\)  Nandita Vijaykumar\(^4\)  Mohammed Alser\(^1\)  Onur Mutlu\(^1\)

\(^1\)ETH Zürich  \(^2\)Bionano Genomics  \(^3\)KMUTNB  \(^4\)University of Toronto
Genome Sequence Analysis

Data Movement from Storage

- Storage System
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)

Alignment

- Computation overhead
- Data movement overhead
Compute-Centric Accelerators

- Storage System
- Heuristics
- Accelerators
- Filters
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)

✓ Computation overhead
✗ Data movement overhead
Key Idea: In-Storage Filtering

Filter reads that do not require alignment inside the storage system.

- **Filtered Reads**
  - Exactly-matching reads: Do not need expensive approximate string matching during alignment.
  - Non-matching reads: Do not have potential matching locations and can skip alignment.
GenStore

Filter reads that do not require alignment inside the storage system

GenStore-Enabled Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✓ Data movement overhead

GenStore provides significant speedup (1.4x - 33.6x) and energy reduction (3.9x – 29.2x) at low cost
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"


[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\textsuperscript{1} Jisung Park\textsuperscript{1} Harun Mustafa\textsuperscript{1} Jeremie Kim\textsuperscript{1} Ataberk Olgun\textsuperscript{1} Arvid Gollwitzer\textsuperscript{1} Damla Senol Cali\textsuperscript{2} Can Firtina\textsuperscript{1} Haiyu Mao\textsuperscript{1} Nour Almadhoun Alserr\textsuperscript{1} Rachata Ausavarungnirun\textsuperscript{3} Nandita Vijaykumar\textsuperscript{4} Mohammed Alser\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \textsuperscript{2}Bionano Genomics \textsuperscript{3}KMUTNB \textsuperscript{4}University of Toronto

SAFARI
Consumer Devices

Consumer devices are everywhere!

Energy consumption is a first-class concern in consumer devices.
Popular Consumer Workloads

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
Video Playback
Google’s video codec

VP9
Video Capture
Google’s video codec
Energy Cost of Data Movement

1st key observation: 62.7% of the total system energy is spent on data movement

Potential solution: move computation close to data

Challenge: limited area and energy budget
Using PIM to Reduce Data Movement

2nd key observation: a significant fraction of the data movement often comes from simple functions.

We can design lightweight logic to implement these simple functions in memory.

Offloading to PIM logic reduces energy and improves performance, on average, by 2.3X and 2.2X.
Workload Analysis

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
YouTube Video Playback
Google’s video codec

VP9
YouTube Video Capture
Google’s video codec
TensorFlow Mobile

57.3% of the inference energy is spent on **data movement**

54.4% of the **data movement** energy comes from **packing/unpacking** and **quantization**
More on PIM for 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"


[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
Truly Distributed GPU Processing with PIM

```c
__global__
void applyScaleFactorsKernel( uint8_t * const out,
                            uint8_t const * const in, const double *factor,
                            size_t const numRows, size_t const numCols )
{
    // Work out which pixel we are working on.
    const int rowIdx = blockIdx.x * blockDim.x + threadIdx.x;
    const int colIdx = blockIdx.y;
    const int sliceIdx = threadIdx.z;

    // Check this thread isn't off the image
    if( rowIdx >= numRows ) return;

    // Compute the index of my element
    size_t linearIdx = rowIdx + colIdx*numRows +
                      sliceIdx*numRows*numCols;
}
```

Diagram:
- 3D-stacked memory (memory stack)
- SM (Streaming Multiprocessor)
- Logic layer
- Main GPU
- Crossbar switch
- Vault Ctrl
- Vault Ctrl
Accelerating GPU Execution with PIM (I)

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler,

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


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

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

Kevin Hsieh‡ Eiman Ebrahimi‡ Gwangsun Kim* Niladrish Chatterjee‡ Mike O'Connor‡
Nandita Vijaykumar‡ Onur Mutlu§‡ Stephen W. Keckler‡

‡Carnegie Mellon University †NVIDIA *KAIST §ETH Zürich
Accelerating GPU Execution with PIM (II)


Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques (PACT), Haifa, Israel, September 2016.

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

Ashutosh Pattnaik\textsuperscript{1} Xulong Tang\textsuperscript{1} Adwait Jog\textsuperscript{2} Onur Kayiran\textsuperscript{3} Asit K. Mishra\textsuperscript{4} Mahmut T. Kandemir\textsuperscript{1} Onur Mutlu\textsuperscript{5,6} Chita R. Das\textsuperscript{1}

\textsuperscript{1}Pennsylvania State University \textsuperscript{2}College of William and Mary \textsuperscript{3}Advanced Micro Devices, Inc. \textsuperscript{4}Intel Labs \textsuperscript{5}ETH Zürich \textsuperscript{6}Carnegie Mellon University
Accelerating Linked Data Structures

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,

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

Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

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

Kevin Hsieh† Samira Khan‡ Nandita Vijaykumar†
Kevin K. Chang† Amirali Boroumand† Saugata Ghose† Onur Mutlu§†

†Carnegie Mellon University ‡University of Virginia §ETH Zürich
Accelerating Dependent Cache Misses

- Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt, "Accelerating Dependent Cache Misses with an Enhanced Memory Controller"
  

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

Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

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

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

Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

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

*The University of Texas at Austin  §ETH Zürich
Accelerating Climate Modeling

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,
  "NERO: A Near High-Bandwidth Memory Stencil Accelerator for Weather Prediction Modeling"

Proceedings of the 30th International Conference on Field-Programmable Logic and Applications (FPL), Gothenburg, Sweden, September 2020.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (23 minutes)]

Nominated for the Stamatis Vassiliadis Memorial Award.

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

Gagandeep Singh\textsuperscript{a,b,c}, Dionysios Diamantopoulos\textsuperscript{c}, Christoph Hagleitner\textsuperscript{c}, Sander Stuijk\textsuperscript{a}, Onur Mutlu\textsuperscript{b}, Henk Corporaal\textsuperscript{a}, Juan Gómez-Luna\textsuperscript{b}

\textsuperscript{a}Eindhoven University of Technology
\textsuperscript{b}ETH Zürich
\textsuperscript{c}IBM Research Europe, Zurich
Accelerating Approximate String Matching

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

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


[Lighting Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]

GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis
Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

"SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping"
[arXiv version]

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

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³ Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim² Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr² Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

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

Accelerating Basecalling + Read Mapping

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

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

Haiyu Mao$^1$ Mohammed Alser$^1$ Mohammad Sadrosadati$^1$ Can Firtina$^1$ Akanksha Baranwal$^1$
Damla Senol Cali$^2$ Aditya Manglik$^1$ Nour Almadhoun Alserr$^1$ Onur Mutlu$^1$

$^1$ETH Zürich  $^2$Bionano Genomics

Accelerating Time Series Analysis

- Ivan Fernandez, Ricardo Quislant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu, "NATSA: A Near-Data Processing Accelerator for Time Series Analysis"


- [Slides (pptx) (pdf)]
- [Talk Video (10 minutes)]
- [Source Code]

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

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

§University of Malaga †National Technical University of Athens ‡ETH Zürich
Accelerating Graph Pattern Mining

Maciej Besta, Raghavendra Kanakagiri, Grzegorz Kwasniewski, Rachata Ausavarungrun, 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 Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungrun³, Jakub Beránek⁴, Konstantinos Kanellopoulos¹, Kacper Janda⁵, Zur Vonarburg-Shmaria¹, Lukas Gianinazzi¹, Ioana Stefan¹, Juan Gómez-Luna¹, Marcin Copik¹, Lukas Kapp-Schwoerer¹, Salvatore Di Girolamo¹, Nils Blach¹, Marek Konieczny⁵, Onur Mutlu¹, Torsten Hoefler¹

¹ETH Zurich, Switzerland  ²IIT Tirupati, India  ³King Mongkut’s University of Technology North Bangkok, Thailand  ⁴Technical University of Ostrava, Czech Republic  ⁵AGH-UST, Poland
Accelerating HTAP Database Systems

  [arXiv version]
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]

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

Amirali Boroumand‡
  ‡Google

Saugata Ghose◊
  ◊Univ. of Illinois Urbana-Champaign

Geraldo F. Oliveira‡

Onur Mutlu‡
  ‡ETH Zürich

Accelerating Neural Network 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†‡
Geraldo F. Oliveira*  Saugata Ghose‡  Xiaoyu Ma§  Berkin Akin§  Eric Shiu§  Ravi Narayanaswami§
Onur Mutlu*†

†Carnegie Mellon Univ.  °Stanford Univ.  ‡Univ. of Illinois Urbana-Champaign  §Google  *ETH Zürich
Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand  Saugata Ghose  Berkin Akin
Ravi Narayanaswami  Geraldo F. Oliveira  Xiaooyu Ma
Eric Shiu  Onur Mutlu

PACT 2021

SAFARI

Carnegie Mellon  University of Illinois Urbana-Champaign  Google  ETH Zürich
# Executive Summary

**Context:** We extensively analyze a state-of-the-art edge ML accelerator (Google Edge TPU) using 24 Google edge models

- Wide range of models (CNNs, LSTMs, Transducers, RCNNs)

**Problem:** The Edge TPU accelerator suffers from three challenges:

- It operates significantly below its peak throughput
- It operates significantly below its theoretical energy efficiency
- It inefficiently handles memory accesses

**Key Insight:** These shortcomings arise from the monolithic design of the Edge TPU accelerator

- The Edge TPU accelerator design does not account for layer heterogeneity

**Key Mechanism:** A new framework called Mensa

- Mensa consists of heterogeneous accelerators whose dataflow and hardware are specialized for specific families of layers

**Key Results:** We design a version of Mensa for Google edge ML models

- Mensa improves performance and energy by 3.0X and 3.1X
- Mensa reduces cost and improves area efficiency

SAFARI
Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models

13 CNN

6 RNN
Transducers

2 LSTMs

3 RCNN

Google Edge TPU

Speech Recognition

Face Detection

Language Translation

Image Captioning

SAFARI
Diversity Across the Models

**Insight 1:** There is significant variation in terms of layer characteristics across the models.
Diversity Within the Models

Insight 2: even within each model, layers exhibit significant variation in terms of layer characteristics.

For example, our analysis of edge CNN models shows:

Variation in MAC intensity: up to 200x across layers

Variation in FLOP/Byte: up to 244x across layers
Identifying Layer Families

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

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

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

**Mensa-G reduces energy consumption by 3.0X** compared to the baseline Edge TPU.
Mensa-G improves inference throughput by 3.1X compared to the baseline Edge TPU.
Mensa: Highly-Efficient ML Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,

"Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (PACT), Virtual, September 2021.
[Slides (pptx) (pdf)]
[Talk Video (14 minutes)]

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

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

†Carnegie Mellon Univ. ‡Stanford Univ. ‡Univ. of Illinois Urbana-Champaign §Google *ETH Zürich
FPGA-based Processing Near Memory


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

Gagandeep Singh♦ Mohammed Alser♦ Damla Senol Cali♦
Dionysios Diamantopoulos▽ Juan Gómez-Luna♦
Henk Corporaal* Onur Mutlu♦▽

♦ETH Zürich ▽Carnegie Mellon University
*Eindhoven University of Technology ▽IBM Research Europe
We Need to Revisit the Entire Stack

We can get there step by step
A Modern Primer on Processing in Memory

Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}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"

SAFARI

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

Saugata Ghose† Amirali Boroumand† Jeremie S. Kim‡§ Juan Gómez-Luna§ Onur Mutlu§†
†Carnegie Mellon University §ETH Zürich

Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu, "Processing-in-Memory: A Workload-Driven Perspective"
[Preliminary arXiv version]

Processing in Memory: Adoption Challenges

1. Processing using Memory
2. Processing near Memory
Eliminating the Adoption Barriers

How to Enable Adoption of Processing in Memory
Potential Barriers to Adoption of PIM

1. Applications & software for PIM

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

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

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

5. Infrastructures to assess benefits and feasibility

All can be solved with change of mindset
We Need to Revisit the Entire Stack

We can get there step by step
Adoption: How to Keep It Simple?

Adoption: How to Maintain Coherence? (I)

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

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

Amirali Boroumand†, Saugata Ghose†, Minesh Patel†, Hasan Hassan†§, Brandon Lucia†,
  Kevin Hsieh†, Krishna T. Malladi*, Hongzhong Zheng*, and Onur Mutlu††

†Carnegie Mellon University  *Samsung Semiconductor, Inc.  §TOBB ETÜ  ††ETH Zürich
Challenge: Coherence for Hybrid CPU-PIM Apps

The diagram illustrates the speedup of various applications with different coherence models. The x-axis represents different applications, and the y-axis shows the speedup. The bars represent different coherence models:

- **CPU-only**: Traditional coherence
- **FG**: Fine-grained coherence
- **CG**: Coarse-grained coherence
- **NC**: No coherence
- **LazyPIM**: Lazy coherence
- **Ideal-PIM**: Ideal PIM coherence

The graph shows that traditional coherence (CPU-only) has the highest performance, followed by the various coherence models. The graph also highlights the impact of coherence overhead across different applications.
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"


CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand†
Brandon Lucia†
Saugata Ghose†
Rachata Ausavarungnirun†‡
Nastaran Hajinazar◆†
Krishna T. Malladi§
.Hasan Hassan*
Kevin Hsieh†
Hongzhong Zheng§
Onur Mutlu*†

†Carnegie Mellon University
‡ETH Zürich
◆Simon Fraser University
§Samsung Semiconductor, Inc.

SAFARI
Adoption: How to Support Synchronization?

  - Slides (pptx) (pdf)
  - Short Talk Slides (pptx) (pdf)
  - Talk Video (21 minutes)
  - Short Talk Video (7 minutes)

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

Christina Giannoula†‡  Nandita Vijaykumar*‡  Nikela Papadopoulou†  Vasileios Karakostas†  Ivan Fernandez§‡  Juan Gómez-Luna†  Lois Orosa†  Nectarios Koziris†  Georgios Goumas†  Onur Mutlu‡

†National Technical University of Athens  ‡ETH Zürich  *University of Toronto  §University of Malaga
Adoption: How to Support Virtual Memory?

Eliminating the Adoption Barriers

Processing-in-Memory in the Real World
Processing-in-Memory Landscape Today

This does not include many experimental chips and startups
Real PIM Tutorial (HPCA 2023)

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

Goal: Processing Inside Memory

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

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

https://events.safari.ethz.ch/real-pim-tutorial/
Real PIM Tutorial (ASPLOS 2023)

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

---

**Real-world Processing-in-Memory Systems for Modern Workloads**

- **Tutorial Description**
  
  Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

  Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

  A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

- **2.560-DPU Processing-in-Memory System**

  PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecks our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) examine architectural alternatives for PIM kernels, and (3) translate those considerations and experiences into practical design decisions.

---

**Tutorial Materials**

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

---

[https://www.youtube.com/watch?v=oYCaLcT0Kmo](https://www.youtube.com/watch?v=oYCaLcT0Kmo)

[https://events.safari.ethz.ch/asplos-pim-tutorial/](https://events.safari.ethz.ch/asplos-pim-tutorial/)
Current Real PIM Tutorial (ISCA 2023)

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

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

https://events.safari.ethz.ch/isca-pim-tutorial/
UPMEM Processing-in-DRAM Engine (2019)

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

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

UPMEM Memory Modules

- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz
2,560-DPU Processing-in-Memory System

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

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

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

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

This paper provides the first comprehensive analysis of the first publicly available real-world PIM architecture. We evaluate two key microarchitectures. First, we conduct an experimental characterization of the UPNEM-based PIM system using microbenchmarks across various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PIM-Bench (Processing-in-Memory benchmark), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and energy consumption characteristics of PIM benchmarks on the UPNEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPNEM-based PIM systems with 110 and 2,560 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and guidelines and hints for hardware and architecture designers of future PIM systems.

More on the UPMEM PIM System

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidy1gBxBxUz7xRPS-wisBN&index=26
Experimental Analysis of the UPMEM PIM Engine

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

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

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

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

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PrIM (Processing-In-Memory benchmarks), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

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

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

https://github.com/CMU-SAFARI/prim-benchmarks
Recent SRC TECHCON Presentation

- Dr. Juan Gomez-Luna
  - Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
  - Based on two major works

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

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

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

https://www.youtube.com/watch?v=nphV36SrysA
**Key Takeaway 1**

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

The throughput saturation point is as low as \( \frac{1}{4} \) OP/B, i.e., 1 integer addition per every 32-bit element fetched.
**Key Takeaway 2**

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

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

<table>
<thead>
<tr>
<th>System</th>
<th>Process Node</th>
<th>Total Cores</th>
<th>Frequency</th>
<th>Peak Performance</th>
<th>Capacity</th>
<th>Total Bandwidth</th>
<th>TDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Xeon E5-1223 v6 CPU [241]</td>
<td>14 nm</td>
<td>8 (4 cores)</td>
<td>3.3 GHz</td>
<td>20.4 GFLOPS*</td>
<td>32 GB</td>
<td>37.1 GHz/s</td>
<td>73 W</td>
</tr>
<tr>
<td>NVIDIA Titan V GPU [277]</td>
<td>14 nm</td>
<td>5120 (112 SMs, 4480 Cores)</td>
<td>1.2 GHz</td>
<td>12,288.1 GFLOPS</td>
<td>12 GB</td>
<td>652.8 GHz/s</td>
<td>250 W</td>
</tr>
<tr>
<td>2,356-DPU PIM System</td>
<td>2x nm</td>
<td>2,356 *</td>
<td>350 MHz</td>
<td>894.6 GFLOPS</td>
<td>139.75 GB</td>
<td>1.7 TB/s</td>
<td>38 W</td>
</tr>
<tr>
<td>640-DPU PIM System</td>
<td>2x nm</td>
<td>640</td>
<td>267 MHz</td>
<td>179.9 GFLOPS</td>
<td>40 GB</td>
<td>333.75 GB/s</td>
<td>96 W</td>
</tr>
</tbody>
</table>

*Estimated GFLOPS = 3.3 GHz × 4 cores × 2 instructions per cycle.

*Estimated TDP = | 3.3 GHz | × 1.2 W/chip [199].

---

**Diagram:**

- **CPU**
- **GPU**
- **640 DPUs**
- **2556 DPUs**

**X-axis:** More PIM-suitable workloads (1)
**Y-axis:** Speedup over CPU (log scale)

**Graph:**

- **VA**
- **SEL**
- **UNI**
- **BS**
- **HST-S**
- **HST-L**
- **RED**
- **SCAN-SSA**
- **SCAN-RSS**
- **TRNS**

- **GEMV**
- **SpMV**
- **TS**
- **BFS**
- **MLP**
- **NW**
- **GMEAN (1)**
- **GMEAN (2)**
- **GMEAN**
Key Takeaway 3

The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).
Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

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

el1goluj@gmail.com

https://github.com/CMU-SAFARI/prim-benchmarks
UPMEM PIM System Summary & Analysis

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

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

*Invited Paper at Workshop on Computing with Unconventional Technologies (CUT), Virtual, October 2021.*

[arXiv version]
[PrIM Benchmarks Source Code]
[Slides (pptx) (pdf)]
[Talk Video (37 minutes)]
[Lightning Talk Video (3 minutes)]

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

Juan Gómez-Luna
ETH Zürich

Izzat El Hajj
American University of Beirut

Ivan Fernandez
University of Malaga

Christina Giannoula
National Technical University of Athens

Geraldo F. Oliveira
ETH Zürich

Onur Mutlu
ETH Zürich
## PrIM Benchmarks: Application Domains

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

- All microbenchmarks, benchmarks, and scripts
- [https://github.com/CMU-SAFAIR/prim-benchmarks](https://github.com/CMU-SAFAIR/prim-benchmarks)

PrIM (Processing-In-Memory Benchmarks)

PrIM is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publicly-available real-world processing-in-memory (PIM) architecture, the UPMEM PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same chip.

PrIM provides a common set of workloads to evaluate the UPMEM PIM architecture with and can be useful for programming, architecture and system researchers all alike to improve multiple aspects of future PIM hardware and software. The workloads have different characteristics, exhibiting heterogeneity in their memory access patterns, operations and data types, and communication patterns. This repository also contains baseline CPU and GPU implementations of PrIM benchmarks for comparison purposes.

PrIM also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.
Understanding a Modern PIM Architecture

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

JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ¹,³, CHRISTINA GIANNOULA¹,⁴, GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹

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

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

https://github.com/CMU-SAFARI/prim-benchmarks
Understanding a Modern PIM Architecture

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

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

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

https://www.youtube.com/watch?v=D8Hjy2lU9I4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is no direct communication channel between DPUs

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

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

https://www.youtube.com/watch?v=D8Hjy2lU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

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

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

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159
ML Training on a Real PIM System

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

Juan Gómez-Luna¹ Yuxin Guo¹ Sylvan Brocard² Julien Legriel²
Remy Cimadomo² Geraldo F. Oliveira¹ Gagandeep Singh¹ Onur Mutlu¹
¹ETH Zürich ²UPMEM

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

Juan Gómez-Luna¹ Yuxin Guo¹ Sylvan Brocard² Julien Legriel²
Remy Cimadomo² Geraldo F. Oliveira¹ Gagandeep Singh¹ Onur Mutlu¹
¹ETH Zürich ²UPMEM

https://www.youtube.com/watch?v=qeukNs5Xl3g&t=11226s
ML Training on a Real PIM System

• Need to optimize data representation
  (1) fixed-point
  (2) quantization
  (3) hybrid precision

• Use lookup tables (LUTs) to implement complex functions (e.g., sigmoid)

• Optimize data placement & layout for streaming

• Large speedups: 2.8X/27X vs. CPU, 1.3x/3.2x vs. GPU
ML Training on Real PIM Talk Video

Comparison to CPU and GPU (III)

- Decision tree and K-means with Criteo 1TB dataset

PIM version of DTR is **62x** faster than the CPU version and **4.5x** faster than the GPU version

PIM version of KME is **2.7x** faster than the CPU version and **3.2x** faster than the GPU version

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

https://www.youtube.com/watch?v=60pkaI5AeM4
ML Training on Real PIM Systems


PIM-ML Source Code

Best paper session.

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

Juan Gómez-Luna\textsuperscript{1} Yuxin Guo\textsuperscript{1} Sylvan Brocard\textsuperscript{2} Julien Legriel\textsuperscript{2} Remy Cimadomo\textsuperscript{2} Geraldo F. Oliveira\textsuperscript{1} Gagandeep Singh\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \quad \textsuperscript{2}UPMEM

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

SAFARI

SpMV Multiplication on Real PIM Systems

- Appears at SIGMETRICS 2022

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

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

[https://github.com/CMU-SAFARI/SparseP](https://github.com/CMU-SAFARI/SparseP)

**SAFARI**
[https://www.youtube.com/watch?v=5kaOsJKlGrE](https://www.youtube.com/watch?v=5kaOsJKlGrE)
SparseP
Towards Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures

Christina Giannoula
Ivan Fernandez, Juan Gomez-Luna,
Nectarios Koziris, Georgios Goumas, Onur Mutlu
SparseP: Key Contributions

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

   **SparseP is Open-Source**
   - SparseP: [https://github.com/CMU-SAFARI/SparseP](https://github.com/CMU-SAFARI/SparseP)

2. **Comprehensive analysis** of SpMV on the first commercially-available real PIM system
   - 26 sparse matrices
   - Comparisons to state-of-the-art CPU and GPU systems
   - Recommendations for software, system and hardware designers

   **Recommendations for Architects and Programmers**
SparseP Talk Video

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

Christina Giannoula
Ivan Fernandez, Juan Gomez-Luna, Nectarios Koziris, Georgios Goumas, Onur Mutlu

https://www.youtube.com/watch?v=5kaOsJKlGrE
Transcendental Functions on Real PIM Systems

- Maurus Item, Juan Gómez Luna, Yuxin Guo, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu,

"TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems"


[arXiv version]
[Slides (pptx) (pdf)]
[TransPimLib Source Code]
[Talk Video (17 minutes)]

TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems

Maurus Item          Juan Gómez-Luna          Yuxin Guo
Geraldo F. Oliveira  Mohammad Sadrosadati   Onur Mutlu
ETH Zürich

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

SAFARI

Sequence Alignment on Real PIM Systems

- Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, and Izzat El Hajj,

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


[Online link at Bioinformatics Journal]

[arXiv preprint]

[AiM Source Code]

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

Safaa Diab¹ Amir Nassereldine¹ Mohammed Alser² Juan Gómez Luna²

Onur Mutlu² Izzat El Hajj¹

¹American University of Beirut ²ETH Zürich


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

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

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

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

Samsung Function-in-Memory DRAM (2021)

- FIMDRAM based on HBM2

[3D Chip Structure of HBM with FIMDRAM]

**Chip Specification**

- 128DQ / 8CH / 16 banks / BL4
- 32 PCU blocks (1 FIM block/2 banks)
- 1.2 TFLOPS (4H)
- FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and-Add (MAD)

---

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


*Samsung Electronics, Hwasung, Korea
**Samsung Electronics, San Jose, CA
**Samsung Electronics, Suwon, Korea
Programmable Computing Unit

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

[Block diagram of PCU in FIMDRAM]
**Available instruction list for FIM operation**

<table>
<thead>
<tr>
<th>Type</th>
<th>CMD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating Point</td>
<td>ADD</td>
<td>FP16 addition</td>
</tr>
<tr>
<td></td>
<td>MUL</td>
<td>FP16 multiplication</td>
</tr>
<tr>
<td></td>
<td>MAC</td>
<td>FP16 multiply-accumulate</td>
</tr>
<tr>
<td></td>
<td>MAD</td>
<td>FP16 multiply and add</td>
</tr>
<tr>
<td>Data Path</td>
<td>MOVE</td>
<td>Load or store data</td>
</tr>
<tr>
<td></td>
<td>FILL</td>
<td>Copy data from bank to GRFs</td>
</tr>
<tr>
<td>Control Path</td>
<td>NOP</td>
<td>Do nothing</td>
</tr>
<tr>
<td></td>
<td>JUMP</td>
<td>Jump instruction</td>
</tr>
<tr>
<td></td>
<td>EXIT</td>
<td>Exit instruction</td>
</tr>
</tbody>
</table>
Chip Implementation

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

[TSV & Peri Control Block]

[Digital RTL design for PCU block]

ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 8GB Function-in-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Choon Keer, Suk Han Lee, Jaekwan Lee, Sang-Hyuk Kim, Seung Min Ry, Jun Pyo Song, Syoung D. Ha, Seung Gu Yu, Haeok Lee, Seo Young Kim, Yeongmin Cho, Jin Guk Kim, Jonghyun Choi, Hyun-Sung Kwon, Jin Kim, Bavgang Phyu, Hyesung Kim, Myeong Jun Song, Ahn Choi, Soo Chieh Koo, Sun Young Kim, Eun-Bong Kim, Caidi Wang, Bichaes Kang, Yulin N., Seungwoo Seo, Joohae Song, Jeeyoon Yoon, Kyomin Sohn, Nam Sung Kim

*Samsung Electronics, Hwasung, Korea
*Samsung Electronics, San Jose, CA
*Samsung Electronics, Suwon, Korea
Samsung AxDIMM (2021)

- DDRx-PIM
  - DLRM recommendation system

SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022

Seoul, February 16, 2022

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

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

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

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

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

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

System Architecture and Software Stack for GDDR6-AiM

Yongkee Kwon and Chanwook Park
SK hynix inc.
29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

Dimin Niu¹, Shuangchen Li¹, Yuhao Wang¹, Wei Han¹, Zhe Zhang², Yijin Guan², Tianchan Guan³, Fei Sun¹, Fei Xue¹, Lide Duan¹, Yuanwei Fang¹, Hongzhong Zheng¹, Xiping Jiang⁴, Song Wang⁴, Fengguo Zuo⁴, Yubing Wang⁴, Bing Yu⁴, Qiwei Ren⁴, Yuan Xie¹
Eliminating the Adoption Barriers

Processing-in-Memory in the Real World
DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
LOIS OROSA, ETH Zürich, Switzerland
SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA
NANDITA VIJAYKUMAR, University of Toronto, Canada
IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

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

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

When to Employ Near-Data Processing?

Mobile consumer workloads
(GoogleWL\(^2\))

Graph processing
(Tesseract\(^1\))

Databases
(Polynesia\(^5\))

Time series analysis
(NATSA\(^6\))

Near-Data Processing

Neural networks
(GoogleWL\(^2\))

DNA sequence mapping
(GenASM\(^3\); GRIM-Filter\(^4\))

...
Step 1: Application Profiling

- We analyze **345 applications** from distinct domains:

  - Graph Processing
  - Deep Neural Networks
  - Physics
  - High-Performance Computing
  - Genomics
  - Machine Learning
  - Databases
  - Data Reorganization
  - Image Processing
  - Map-Reduce
  - Benchmarking
  - Linear Algebra

...
Step 3: Memory Bottleneck Analysis

Six classes of data movement bottlenecks:

each class ↔ data movement mitigation mechanism
DAMOV is Open Source

• We open-source our benchmark suite and our toolchain

DAMOV-SIM

DAMOV Benchmarks

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

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

- We open-source our benchmark suite and our toolchain

Get DAMOV at:

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

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

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

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

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

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

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

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

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3
More on DAMOV Methods & Benchmarks

- 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 GÔMEZ-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
Challenge and Opportunity for Future

Fundamentally Energy-Efficient (Data-Centric) Computing Architectures
Challenge and Opportunity for Future

Fundamentally
High-Performance
(Data-Centric)
Computing Architectures
Challenge and Opportunity for Future Computing Architectures with Minimal Data Movement
Concluding Remarks
Concluding Remarks

- We must design systems to be **balanced, high-performance, energy-efficient** (all at the same time) → intelligent systems
  - Data-centric, data-driven, data-aware

- Enable computation capability inside and close to memory

- This can
  - Lead to **orders-of-magnitude** improvements
  - Enable new applications & computing platforms
  - Enable better understanding of nature
  - ...

- Future of **truly memory-centric computing** is bright
  - We need to do research & design across the computing stack
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Revisit the Entire Stack

We can get there step by step
We Need to Exploit Good Principles

- Data-centric system design
- All components intelligent
- Better (cross-layer) communication, better interfaces
- Better-than-worst-case design
- Heterogeneity
- Flexibility, adaptability

Open minds
A Blueprint for Fundamentally Better Architectures

- Onur Mutlu,


[Slides (pptx) (pdf)]
[IEDM Tutorial Slides (pptx) (pdf)]
[Short DATE Talk Video (11 minutes)]
[Longer IEDM Tutorial Video (1 hr 51 minutes)]

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
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

Thank you!
Acknowledgments

SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

https://safari.ethz.ch
Onur Mutlu’s SAFARI Research Group

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

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

Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch
SAFARI Newsletter December 2021 Edition

https://safari.ethz.ch/safari-newsletter-december-2021/

Think Big, Aim High

ETH Zürich

View in your browser
December 2021
SAFARI Introduction & Research

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

Think BIG, Aim HIGH!

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

- 241 followers
- ETH Zurich and Carnegie Mellon University
- https://safari.ethz.ch/
- omutlu@gmail.com

Pinned

```
- **ramulator** Public
  A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...
  - C++
  - 🌟 415
  - ⬇️ 187

- **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++
  - 🌟 185
  - ⬇️ 112

- **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 publi...
  - C
  - 🌟 82
  - ⬇️ 35

- **rowhammer** Public
  - C
  - 🌟 203
  - ⬇️ 40

- **SparseP** Public
  SparseP is the first open-source Sparse Matrix Vector Multiplication (SpMV) software package for real-world Processing-In-Memory (PIM) architectures. SparseP is developed to evaluate and characteri...
  - C
  - 🌟 55
  - ⬇️ 10

- **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
  - 🌟 99
  - ⬇️ 26
```

https://github.com/CMU-SAFARI/
Special Research Sessions & Courses

- Special Session at ISVLSI 2022: 9 cutting-edge talks

https://www.youtube.com/watch?v=qeukNs5XI3g
Special Research Sessions & Courses (II)

- Special Session at ISVLSI 2022: 9 cutting-edge talks

SAFARI
https://www.youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GObnK
Comp Arch (Fall 2021)

- **Fall 2021 Edition:**
  - https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule

- **Fall 2020 Edition:**

- **Youtube Livestream (2021):**
  - https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF

- **Youtube Livestream (2020):**
  - https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN

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

https://www.youtube.com/onurmutlulectures
DDCA (Spring 2022)

**Spring 2022 Edition:**

**Spring 2021 Edition:**

**Youtube Livestream (Spring 2022):**
- [https://www.youtube.com/watch?v=cpxdE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMqO2bbAoaG7c6](https://www.youtube.com/watch?v=cpxdE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMqO2bbAoaG7c6)

**Youtube Livestream (Spring 2021):**
- [https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3ay39YB5pfW4SJ7L1N](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3ay39YB5pfW4SJ7L1N)

**Bachelor’s course**
- 2nd semester at ETH Zurich
- Rigorous introduction into “How Computers Work”
- Digital Design/Logic
- Computer Architecture
- 10 FPGA Lab Assignments

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Processing-in-Memory Course (Fall 2022)

- Short weekly lectures
- Hands-on projects

Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)

Course Description

Data movement between the memory units and the compute units of current computing systems is a major performance and energy bottleneck. From large-scale servers to mobile devices, data movement costs dominate computation costs in terms of both performance and energy consumption. For example, data movement between the main memory and the processing cores accounts for 62% of the total system energy in consumer applications. As a result, the data movement bottleneck is a huge burden that greatly limits the energy efficiency and performance of modern computing systems. This phenomenon is an undesired effect of the dichotomy between memory and the processor, which leads to the data movement bottleneck.

Many modern and important workloads such as machine learning, computational biology, graph processing, databases, video analytics, and real-time data analytics suffer greatly from the data movement bottleneck. These workloads are exemplified by irregular memory accesses, relatively low data reuse, low cache line utilization, low arithmetic intensity (i.e., ratio of operations per accessed byte), and large datasets that greatly exceed the main memory size. The computation in these workloads cannot usually compensate for the data movement costs. In order to alleviate this data movement bottleneck, we need a paradigm shift from the traditional processor-centric design, where all computation takes place in the compute units, to a more data-centric design where processing elements are placed closer to or inside where the data resides. This paradigm of computing is known as Processing-in-Memory (PIM).

This is your perfect P&S if you want to become familiar with the main PIM technologies, which represent “the next big thing” in Computer Architecture. You will work hands-on with the first real-world PIM architecture, will explore different PIM architecture designs for important workloads, and will develop tools to enable research of future PIM systems. Projects in this course span software and hardware as well as the software/hardware interface. You can potentially work on developing and optimizing new workloads for the first real-world PIM hardware or explore new PIM designs in simulators, or do something else that can forward our understanding of the PIM paradigm.

https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory

https://youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNOVD0GOBrnKy
PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQVOD0GOBrnKy](https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQVOD0GOBrnKy)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX](https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPyX)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
February 26: Lectures + Hands-on labs + Invited Talks

Real-world Processing-in-Memory Architectures

Table of Contents
- Real-world Processing-in-Memory Architectures
- Tutorial Description
- Organizers
- Agenda (February 28, 2023)
- Lectures (tentative)
- Hands-on Labs (tentative)
- Learning Materials

Explained over several decades since the 1990s, PIM (Processing-in-Memory) systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neoverge, Mythic) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years.

2,560-DPU Processing-in-Memory System

Most of these architectures have in common that they place compute units near the memory arrays. But, there is more to come: Academia and Industry are actively exploring other types of PIM by, e.g., exploiting the analog operation of DRAM, SRAM, flash memory and emerging non-volatile memories.

PIM can provide large improvements in both performance and energy consumption, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to examine and research adoption issues of PIM using especially learnings from real PIM systems that are available today.

This tutorial focuses on the latest advances in PIM technology. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hands-on labs using real PIM systems, and (4) shed light on how to enable the adoption of PIM in future computing systems.

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

Goal: Processing Inside Memory

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

https://www.youtube.com/watch?v=f5-nT1tbz5w
https://events.safari.ethz.ch/real-pim-tutorial/
Real PIM Tutorial (ASPLOS 2023)

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

[Image of the tutorial schedule]

**Tutorial Materials**

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

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

https://events.safari.ethz.ch/asplos-pim-tutorial/
Current Real PIM Tutorial (ISCA 2023)

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

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

https://events.safari.ethz.ch/isca-pim-tutorial/
SSD Course (Spring 2023)

- **Spring 2023 Edition:**

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssds

- **Youtube Livestream (Spring 2023):**
  - https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4J15bwhAMpAp13&pp=iAQB

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - SSD Basics and Advanced Topics
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
Genomics Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=bioinformatics

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=bioinformatics

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=nA41964-9r8&list=PL5Q2soXY2Zi8tFlQvdxOdizD_EhVAMVQV

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDqOR1yRU_Cxxjw-u18

- Project course
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
Hetero. Systems (Spring’22)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=oF5fTrqFIY&list=PL5Q2soXY2Zi9XrgXR38IM_FTjmY6h7Gzm

- Project course
  - Taken by Bachelor’s/Master’s students
  - GPU and Parallelism lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
HW/SW Co-Design (Spring 2022)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=hw_sw_co_design)

- **Youtube Livestream:**
  - [https://youtube.com/playlist?list=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-NK](https://youtube.com/playlist?list=PL5Q2soXY2Zi8nH7un3ghD2nutKWWDk-NK)

- Project course
  - Taken by Bachelor’s/Master’s students
  - HW/SW co-design lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
RowHammer & DRAM Exploration (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO](https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO)

- **Bachelor’s course**
  - Elective at ETH Zurich
  - Introduction to DRAM organization & operation
  - Tutorial on using FPGA-based infrastructure
  - Verilog & C++
  - Potential research exploration

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Exploration of Emerging Memory Systems (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=aMllXROd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo29Z5ApqV](https://www.youtube.com/watch?v=aMllXROd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo29Z5ApqV)

- Bachelor’s course
  - Elective at ETH Zurich
  - Introduction to memory system simulation
  - Tutorial on using Ramulator
  - C++
  - Potential research exploration

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
18 June 2023
Real-World PIM Tutorial Opening Talk @ ISCA-50
Backup Slides
SAFARI PhD and Post-Doc Alumni

- Hasan Hassan (Rivos), EDAA Outstanding Dissertation Award 2023; S&P 2020 Best Paper Award, 2020 Pwinie Award, IEEE Micro TP HM 2020
- Christina Giannoula (Univ. of Toronto)
- Minesh Patel (ETH Zurich), DSN Carter Award for Best Thesis 2022; ETH Medal 2023; MICRO’20 & DSN’20 Best Paper Awards; ISCA HoF 2021
- Damla Senol Cali (Bionano Genomics), SRC TECHCON 2019 Best Student Presentation Award; RECOMB-Seq 2018 Best Poster Award
- Nastaran Hajinazar (Intel)
- Gagandeep Singh (AMD/Xilinx), FPL 2020 Best Paper Award Finalist
- Amirali Boroumand (Stanford Univ → Google), SRC TECHCON 2018 Best Presentation Award
- Jeremie Kim (Apple), EDAA Outstanding Dissertation Award 2020; IEEE Micro Top Picks 2019; ISCA/MICRO HoF 2021
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor), ISCA Hall of Fame 2021
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook), HiPEAC 2015 Best Student Presentation Award; ICCD 2012 Best Paper Award
- Mohammed Alser (ETH Zurich), IEEE Turkey Best PhD Thesis Award 2018
- Yixin Luo (Google), HPCA 2015 Best Paper Session
- Kevin Chang (Facebook), SRC TECHCON 2016 Best Student Presentation Award
- Rachata Ausavarungnirun (KMUNTB, Assistant Professor), NOCS 2015 and NOCS 2012 Best Paper Award Finalist
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor), ISCA Hall of Fame 2021; ASPLOS 2015 SRC Winner
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher), HPCA Hall of Fame 2018
- Yoongu Kim (Software Robotics → Google), TCAD’19 Top Pick Award; IEEE Micro Top Picks’10; HPCA’10 Best Paper Session
- Lavanya Subramanian (Intel Labs → Facebook)
- Samira Khan (Univ. of Virginia, Assistant Professor), HPCA 2014 Best Paper Session
- Saugata Ghose (Univ. of Illinois, Assistant Professor), DFRWS-EU 2017 Best Paper Award
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)
- Lois Orosa (Galicia Supercomputing Center, Director)
- Jisung Park (POSTECH, Assistant Professor)
- Gagandeep Singh (AMD/Xilinx, Researcher)
You Can Join Us!

- [https://safari.ethz.ch/apply/](https://safari.ethz.ch/apply/)

SAFARI Researcher Applications

Sign in

This is the application submission site to be considered for being a researcher in the SAFARI Research Group, directed by Professor Onur Mutlu (Publications and Teaching).

If you are interested in doing research in the SAFARI Research Group, please make sure you apply through this submissions site and supply as many of the requested documents and information as possible. Please read and follow the provided instructions and submit as complete an application as possible (given the position you are applying for).

We suggest studying the following materials before submission:
SAFARI Publications and Courses
Onur Mutlu's Online Lectures and Course Materials

We strongly recommend that you read and analyze critically as many recent papers from our group as possible. This is the best way to prepare for the application process. Our recommendation is that you use professor Mutlu's methodology for critically analyzing papers.

Guide On Reviewing Papers

Good luck!

Welcome to the SAFARI at ETH Zurich -- PhD, Postdoc, Internship, Visiting Researcher Applications (SAFARI Researcher Applications) submissions site.
Data-Driven (Self-Optimizing) Architectures
System Architecture Design Today

- Human-driven
  - Humans design the policies (how to do things)

- Many (too) simple, short-sighted policies all over the system

- No automatic data-driven policy learning

- (Almost) no learning: cannot take lessons from past actions

Can we design fundamentally intelligent architectures?
An Intelligent Architecture

- Data-driven
  - Machine learns the “best” policies (how to do things)

- Sophisticated, workload-driven, changing, far-sighted policies

- Automatic data-driven policy learning

- All controllers are intelligent data-driven agents

We need to rethink design (of all controllers)
Self-Optimizing Memory Controllers


Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek\(^1,2\), Onur Mutlu\(^2\), José F. Martínez\(^1\), Rich Caruana\(^1\)

\(^1\)Cornell University, Ithaca, NY 14850 USA
\(^2\)Microsoft Research, Redmond, WA 98052 USA
Self-Optimizing Memory Prefetchers


[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (20 minutes)]
[Lightning Talk Video (1.5 minutes)]
[Pythia Source Code (Officially Artifact Evaluated with All Badges)]
[arXiv version]

Officially artifact evaluated as available, reusable and reproducible.

Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera\textsuperscript{1}  Konstantinos Kanellopoulos\textsuperscript{1}  Anant V. Nori\textsuperscript{2}  Taha Shahroodi\textsuperscript{3,1}
Sreenivas Subramoney\textsuperscript{2}  Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich  \textsuperscript{2}Processor Architecture Research Labs, Intel Labs  \textsuperscript{3}TU Delft

Learning-Based Off-Chip Load Predictors

- Rahul Bera, Konstantinos Kanellopoulos, Shankar Balachandran, David Novo, Ataberk Olgun, Mohammad Sadrosadati, and Onur Mutlu,

"Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction"

Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Talk Video (12 minutes)]
[Lecture Video (25 minutes)]
[arXiv version]
[Source Code (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, reusable and reproducible.
Best paper award at MICRO 2022.

Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction

Rahul Bera¹ Konstantinos Kanellopoulos¹ Shankar Balachandran² David Novo³
Ataberk Olgun¹ Mohammad Sadrosadati¹ Onur Mutlu¹

¹ETH Zürich ²Intel Processor Architecture Research Lab ³LIRMM, Univ. Montpellier, CNRS

Self-Optimizing Hybrid SSD Controllers

Gagandeep Singh, Rakesh Nadig, Jisung Park, Rahul Bera, Nastaran Hajinazar, David Novo, Juan Gomez-Luna, Sander Stuijk, Henk Corporaal, and Onur Mutlu, "Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning"


[Slides (pptx) (pdf)]
[arXiv version]
[Sibyl Source Code]
[Talk Video (16 minutes)]

Sibyl: Adaptive and Extensible Data Placement in Hybrid Storage Systems Using Online Reinforcement Learning

Gagandeep Singh¹  Rakesh Nadig¹  Jisung Park¹  Rahul Bera¹  Nastaran Hajinazar¹
David Novo³  Juan Gómez-Luna¹  Sander Stuijk²  Henk Corporaal²  Onur Mutlu¹

¹ETH Zürich  ²Eindhoven University of Technology  ³LIRMM, Univ. Montpellier, CNRS

Data-Driven (Self-Optimizing) Computing Architectures
Data-Characteristic-Aware Architectures
Data-Aware Architectures

- A data-aware architecture understands what it can do with and to each piece of data.

- It makes use of different properties of data to improve performance, efficiency and other metrics:
  - Compressibility
  - Approximability
  - Locality
  - Sparsity
  - Criticality for Computation
  - Access Semantics
  - ...
One Problem: Limited Expressiveness

Higher-level information is not visible to HW

Software

Hardware

Data Structures

Code Optimizations

Access Patterns

100011111...
101010011...

Integer

Float

Data Type

Char

Instructions

Memory Addresses
A Solution: More Expressive Interfaces

- **Performance**
- **Functionality**

Software

- ISA
- Virtual Memory

Higher-level Program Semantics

Expressive Memory “XMem”

Hardware
Expressive (Memory) Interfaces

  [Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video]

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

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

†Carnegie Mellon University ‡University of Toronto †Simon Fraser University §ETH Zürich
Expressive (Memory) Interfaces for GPUs

Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu, "The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"
[Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video]

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

Nandita Vijaykumar†§ Eiman Ebrahimi‡ Kevin Hsieh†
Phillip B. Gibbons† Onur Mutlu§†
†Carnegie Mellon University ‡NVIDIA §ETH Zürich
Open-Source Frameworks for Data-Aware Systems

- Nandita Vijaykumar, Ataberk Olgun, Konstantinos Kanellopoulos, F. Nisa Bostanci, Hasan Hassan, Mehrshad Lotfi, Phillip B. Gibbons, and Onur Mutlu,

"MetaSys: A Practical Open-source Metadata Management System to Implement and Evaluate Cross-layer Optimizations"


[arXiv version]

Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.

[Slides (pptx) (pdf)]

[Preliminary Talk Video (14 minutes)]

[SAFARI Live Seminar Video (1 hour 26 minutes)]

[MetaSys Source Code]

*Best paper award at HiPEAC 2023.*

---

**MetaSys: A Practical Open-Source Metadata Management System to Implement and Evaluate Cross-Layer Optimizations**

Nandita Vijaykumar*  Ataberk Olgun§  Konstantinos Kanellopoulos§  Hasan Hassan§  Mehrshad Lotfi§  Phillip B. Gibbons†  Onur Mutlu§

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

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,

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

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (38 minutes)]
SMASH: SW/HW Indexing Acceleration

Konstantinos Kanellopoulos, Nandita Vijaykumar, Christina Giannoula, Roknoddin Azizi, Skanda Koppula, Nika Mansouri Ghiasi, Taha Shahroodi, Juan Gomez-Luna, and Onur Mutlu,
"SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations"

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (30 minutes)]

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

Konstantinos Kanellopoulos1 Nandita Vijaykumar2,1 Christina Giannoula1,3 Roknoddin Azizi1 Skanda Koppula1 Nika Mansouri Ghiasi1 Taha Shahroodi1 Juan Gomez Luna1 Onur Mutlu1,2

1ETH Zürich 2Carnegie Mellon University 3National Technical University of Athens
Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,

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


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[ARM Research Summit Poster (pptx) (pdf)]
[Talk Video (26 minutes)]
[Lightning Talk Video (3 minutes)]

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

Nastaran Hajinazar*† Pratyush Patel‡‡ Minesh Patel* Konstantinos Kanellopoulos* Saugata Ghose†† Rachata Ausavarungnirun○ Geraldo F. Oliveira* Jonathan Appavoo◊ Vivek Seshadri∇ Onur Mutlu*†

*ETH Zürich †Simon Fraser University ‡‡University of Washington ††Carnegie Mellon University ○King Mongkut’s University of Technology North Bangkok ◊Boston University ∇Microsoft Research India
Challenge and Opportunity for Future Data-Characteristic-Aware Computing Architectures
More Background Slides
Processing-in-Memory Landscape Today

And, many other experimental chips and startups
Memory Scaling Issues Are Real

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

Memory Scaling: A Systems Architecture Perspective

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

As Memory Scales, It Becomes Unreliable

- Data from all of Facebook’s servers worldwide
- Meza+, “Revisiting Memory Errors in Large-Scale Production Data Centers,” DSN’15.

Intuition: quadratic increase in capacity
Infrastructures to Understand Such Issues

Memory Testing Infrastructures

* SoftMC [Hassan+, HPCA’17] enhanced for DDR4
Updated Memory Testing Infrastructure

FPGA-based SoftMC (Xilinx Virtex UltraScale+ XCU200)

Fine-grained control over **DRAM commands**, timing \((\pm 1.5\text{ns})\), **temperature** \((\pm 0.1^\circ\text{C})\), and **voltage** \((\pm 1\text{mV})\)

One can predictably induce errors in most DRAM memory chips.

Modern Memory is Prone to Disturbance Errors

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

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

A company

86% (37/43)

Up to $1.0 \times 10^7$ errors

B company

83% (45/54)

Up to $2.7 \times 10^6$ errors

C company

88% (28/32)

Up to $3.3 \times 10^5$ errors

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

A simple hardware failure mechanism can create a widespread system security vulnerability.
Memory Scaling Issues Are Real

Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [Lecture Video (1 hr 49 mins), 25 September 2020]

One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link).

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

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹ Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs
Memory Scaling Issues Are Real

Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
[Slides (pptx) (pdf)]

The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu
Memory Scaling Issues Are Real

- Onur Mutlu and Jeremie Kim, "RowHammer: A Retrospective"
  [Preliminary arXiv version]
  [Slides from COSADE 2019 (pptx)]
  [Slides from VLSI-SOC 2020 (pptx) (pdf)]
  [Talk Video (1 hr 15 minutes, with Q&A)]

RowHammer: A Retrospective

Onur Mutlu§‡  Jeremie S. Kim‡§
§ETH Zürich  ‡Carnegie Mellon University
Memory Scaling Issues Are Real

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci, "Fundamentally Understanding and Solving RowHammer"
  Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkçı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

The Story of RowHammer Tutorial …

Onur Mutlu,
"Security Aspects of DRAM: The Story of RowHammer"
[Slides (pptx)(pdf)]
[Tutorial Video (57 minutes)]
Onur Mutlu, "The Story of RowHammer"
Invited Talk at the Workshop on Robust and Safe Software 2.0 (RSS2), held with the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, 28 February 2022.
[Slides (pptx) (pdf)]

https://www.youtube.com/watch?v=ctKTRyi96Bk
The Push from Circuits and Devices

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

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

SK hynix Semiconductor, Icheon, Korea
SK hynix Semiconductor, Icheon, Korea

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

Seungki Hong  Dongha Kim  Jaehyung Lee  Reum Oh  Changsik Yoo  Sangjoon Hwang  Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

Intel Optane Persistent Memory (2019)

- Non-volatile main memory
- Based on 3D-XPoint Technology
Emerging Memories Also Need Intelligent Controllers


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

Architecting Phase Change Memory as a Scalable DRAM Alternative

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

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

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

Intelligent Memory Controllers
Can Avoid Many Failures
& Enable Better Scaling
Three Key Systems & Application Trends

1. Data access is the major bottleneck
   - Applications are increasingly data hungry

2. Energy consumption is a key limiter

3. Data movement energy dominates compute
   - Especially true for off-chip to on-chip movement
Do We Want This?

Source: V. Milutinovic
Or This?

Source: V. Milutinovic
Challenge and Opportunity for Future

High Performance, Energy Efficient, Sustainable
(All at the Same Time)
Data access is the major performance and energy bottleneck

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

Processing of data is performed far away from the data
A Computing System

- Three key components
- Computation
- Communication
- Storage/memory

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.
A Computing System

- Three key components
- Computation
- Communication
- Storage/memory

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.

Today’s Computing Systems

- Processor centric

- All data processed in the processor → at great system cost
It’s the Memory, Stupid!

“It’s the Memory, Stupid!” (Richard Sites, MPR, 1996)

Richard Sites

It’s the Memory, Stupid!
When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.

I expect that over the coming decade memory subsystem design will be the only important design issue for microprocessors.

The Performance Perspective

The Performance Perspective

Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"


One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu § Jared Stark † Chris Wilkerson ‡ Yale N. Patt §

§ECE Department
The University of Texas at Austin
{onur,patt}@ece.utexas.edu

†Microprocessor Research
Intel Labs
jared.w.stark@intel.com

‡Desktop Platforms Group
Intel Corporation
chris.wilkerson@intel.com
The Performance Perspective (Today)

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

The Energy Perspective

Communication Dominates Arithmetic

Dally, HiPEAC 2015

- 64-bit DP 20pJ
- 256-bit buses
- 256-bit access 8 kB SRAM
- DRAM Rd/Wr 16 nJ
- Efficient off-chip link 500 pJ
- 50 pJ
- 26 pJ
- 256 pJ
- 1 nJ

SAFARI
A memory access consumes \(~100\text{-}1000\times\) the energy of a complex addition.
Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)

- ADD (int)
- ADD (float)
- Register File
- MULT (int)
- MULT (float)
- SRAM Cache
- DRAM

Energy (pJ)

ADD (int) Relative Cost

Data Movement vs. Computation Energy

A memory access consumes 6400X the energy of a simple integer addition.
Energy Waste in Mobile Devices


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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand\(^1\) \hspace{1cm} \text{Saugata Ghose}\(^1\) \hspace{1cm} \text{Youngsok Kim}\(^2\)
Rachata Ausavarungnirun\(^1\) \hspace{1cm} \text{Eric Shiu}\(^3\) \hspace{1cm} \text{Rahul Thakur}\(^3\)
Aki Kuusela\(^3\) \hspace{1cm} \text{Allan Knies}\(^3\) \hspace{1cm} \text{Parthasarathy Ranganathan}\(^3\)
\text{Daehyun Kim}\(^4,\(^3\) \hspace{1cm} \text{Onur Mutlu}\(^5,\(^1\)

SAFARI
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†‡
Geraldo F. Oliveira*
Saugata Ghose‡
Xiaoyu Ma§
Berkin Akin§
Eric Shiu§
Ravi Narayanaswami§
Onur Mutlu*†

†Carnegie Mellon Univ.  ○Stanford Univ.  ‡Univ. of Illinois Urbana-Champaign  §Google  *ETH Zürich
We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015

A memory access consumes \(~100-1000\times\) the energy of a complex addition
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**
Goal: Processing Inside Memory

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

A Modern Primer on Processing in Memory

Onur Mutlu\textsuperscript{a,b}, Saugata Ghose\textsuperscript{b,c}, Juan Gómez-Luna\textsuperscript{a}, Rachata Ausavarungnirun\textsuperscript{d}

SAFARI Research Group

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}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"

SAFARI

PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory
- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory

- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy
- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYUK9EsXKhQKRPyX

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

https://www.youtube.com/onurmutlulectures
March 26: Lectures + Hands-on labs + Invited talks

- **9:00am-10:20am**: Prof. Onur Mutlu - Memory-Centric Computing

- **10:40am-12:00pm**: Dr. Juan Gómez Luna - Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM

- **1:40pm-2:20pm**: Prof. Alexandra (Sasha) Fedorova (UBC) - Processing in Memory in the Wild

- **2:20pm-3:20pm**: Dr. Juan Gómez Luna & Alaberik Olgun - Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components

- **3:40pm-4:10pm**: Dr. Juan Gómez Luna - Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System

- **4:10pm-4:50pm**: Dr. Yongjie Kwon & Eddy (Chanwook) Park (SK Hynix) - System Architecture and Software Stack for GDDR6-AIM

- **4:50pm-5:00pm**: Dr. Juan Gómez Luna - Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture

---

**Real-world Processing-in-Memory Systems for Modern Workloads**

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) develop satisfactory algorithms for PIM kernels, and (3) design adequate hardware.

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

- **March 26: Lectures + Hands-on labs + Invited talks**
  - [Real PIM Tutorial (ASPLOS 2023)](https://events.safari.ethz.ch/asplos-pim-tutorial/)
  - [Real-world Processing-in-Memory Systems for Modern Workloads](https://www.youtube.com/watch?v=oYCaLcT0Kmo)
  - [Accelerating Modern Workloads on a General-purpose PIM System](https://events.safari.ethz.ch/asplos-2023-tutorial/)
  - [Tutorial Materials](#)

---

**Tutorial Materials**

<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker</th>
<th>Title</th>
<th>Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00am-10:20am</td>
<td>Prof. Onur Mutlu</td>
<td>Memory-Centric Computing</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>10:40am-12:00pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Processing-Near-Memory: Real PNM Architectures Programming General-purpose PIM</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>1:40pm-2:20pm</td>
<td>Prof. Alexandra (Sasha) Fedorova (UBC)</td>
<td>Processing in Memory in the Wild</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>2:20pm-3:20pm</td>
<td>Dr. Juan Gómez Luna &amp; Alaberik Olgun</td>
<td>Processing-Using-Memory: Exploiting the Analog Operational Properties of Memory Components</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>3:40pm-4:10pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Adoption issues: How to enable PIM? Accelerating Modern Workloads on a General-purpose PIM System</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>4:10pm-4:50pm</td>
<td>Dr. Yongjie Kwon &amp; Eddy (Chanwook) Park (SK Hynix)</td>
<td>System Architecture and Software Stack for GDDR6-AIM</td>
<td><a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
<tr>
<td>4:50pm-5:00pm</td>
<td>Dr. Juan Gómez Luna</td>
<td>Hands-on Lab: Programming and Understanding a Real Processing-in-Memory Architecture</td>
<td><a href="#">Handout</a> <a href="#">PDF</a> <a href="#">PPT</a></td>
</tr>
</tbody>
</table>
Current Real PIM Tutorial (ISCA 2023)

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

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

https://events.safari.ethz.ch/isca-pim-tutorial/
End of Backup Slides