Tutorial on Memory-Centric Computing: Introduction

Geraldo F. Oliveira

Prof. Onur Mutlu

HEART 2024
21 June 2024
Brief Self Introduction

- **Geraldo F. Oliveira**
  - Researcher @ SAFARI Research Group since November 2017
  - Soon, I will defend my PhD thesis, advised by Onur Mutlu
  - [https://geraldofojunior.github.io/](https://geraldofojunior.github.io/)
  - [geraldofojunior@gmail.com](mailto:geraldofojunior@gmail.com) (Best way to reach me)
  - [https://safari.ethz.ch](https://safari.ethz.ch)

- **Research in:**
  - Computer architecture, computer systems, hardware security
  - Memory and storage systems
  - Hardware security, safety, predictability
  - Fault tolerance
  - Hardware/software cooperation
  - ...

---

SAFARI
Agenda

- Introduction to Memory-Centric Computing Systems
- Real-World Processing-Near-Memory Systems
- Processing-Using-Memory Architectures for Bulk Bitwise Operations
- Lunch Break
- PIM Programming & Infrastructure for PIM Research
- Tentatively: Hands-on Lab on Programming and Understanding a Real Processing-in-Memory Architecture
Agenda

- Introduction to Memory-Centric Computing Systems
- Real-World Processing-Near-Memory Systems
- Processing-Using-Memory Architectures for Bulk Bitwise Operations
- Lunch Break
- PIM Programming & Infrastructure for PIM Research
- Tentatively: Hands-on Lab on Programming and Understanding a Real Processing-in-Memory Architecture
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

1800x more compute
In just 2 years

Tomorrow, multi-trillion parameter models

~4 orders of magnitude increase in memory requirement in just two years!

https://www.youtube.com/watch?v=x2-qB0J7KHW
Data is Key for Future Workloads

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

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

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

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

In-memory Databases

Graph/Tree Processing

Data → performance & energy bottleneck

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

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

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
Video Playback
Google’s video codec

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

Chrome

TensorFlow Mobile

Data → performance & energy bottleneck

VP9 YouTube

Video Playback

Google’s video codec

VP9 YouTube

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

Data → performance & energy bottleneck
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
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

Apple M1 Ultra System (2022)

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

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

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

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

SAFARI
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
Yet ... 

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

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

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

The Performance Perspective (Today)

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

![Graph showing cache-bound cycles](image)

**Figure 11:** Half of cycles are spent stalled on caches.

Perils of Processor-Centric Design

- **Grossly-imbalanced systems**
  - Processing done only in **one place**
  - All else just stores and moves data: **data moves a lot**
    - Energy inefficient
    - Low performance
    - Complex

- **Overly complex and bloated processor (and accelerators)**
  - To tolerate data access from memory
  - Complex hierarchies and mechanisms
    - Energy inefficient
    - Low performance
    - Complex
The Energy Perspective

Communication Dominates Arithmetic

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

Dally, HiPEAC 2015
A memory access consumes ~100-1000X the energy of a complex addition.
Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)

- **Energy (pJ)**
- **ADD (int) Relative Cost**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
<th>Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD (int)</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>ADD (float)</td>
<td>0.9</td>
<td>3.1</td>
</tr>
<tr>
<td>Register File</td>
<td>3.7</td>
<td>3.1</td>
</tr>
<tr>
<td>MULT (int)</td>
<td>5</td>
<td>640</td>
</tr>
<tr>
<td>MULT (float)</td>
<td>640</td>
<td></td>
</tr>
<tr>
<td>SRAM Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRAM</td>
<td>640</td>
<td></td>
</tr>
</tbody>
</table>

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

We Do Not Want to Move Data!

Communication Dominates Arithmetic

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**
An Intelligent Architecture
Handles Data Well
How to Handle Data Well

- **Ensure data does not overwhelm** the components
  - via intelligent algorithms
  - via intelligent architectures
  - via whole 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
Architectures for Intelligent Machines

Data-centric

Data-driven

Data-aware
Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
We Need to Revisit the Entire Stack

We can get there step by step
Data-Centric (Memory-Centric) Architectures
Data-Centric Architectures: Properties

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

- **Low-latency and low-energy data access**
  - Low latency memory
  - Low energy memory

- **Low-cost data storage and 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

Apple M1 Ultra System (2022)

[Link to GSM Arena article]

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
We Need to Think Differently from the Past Approaches
Mindset: Memory as an Accelerator

Memory similar to a “conventional” accelerator
Processing in Memory: An Old Idea (I)


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

Cellular Logic-in-Memory Arrays

WILLIAM H. KAUTZ, MEMBER, IEEE

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

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

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

Fig. 1. Cellular sorting array I.

https://doi.org/10.1109/T-C.1969.222754
Processing in Memory: An Old Idea (II)


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.

Processing in Memory: An Old Idea (III)


A CASE FOR INTELLIGENT RAM

Dr. David Patterson
Thomas Anderson
Neal Cardwell
Richard Fromm
Kimberly Keeton
Christoforos Kozyrakis
Randi Thomas
Katherine Yelick
University of California, Berkeley

Two trends call into question the current practice of fabricating microprocessors and DRAMs as different chips on different fabrication lines. The gap between processor and DRAM speed is growing at 50% per year; and the size and organization of memory on a single DRAM chip is becoming awkward to use, yet size is growing at 60% per year.

Intelligent RAM, or IRAM, merges processing and memory into a single chip to lower memory latency, increase memory bandwidth, and improve energy efficiency. It also allows more flexible selection of memory size and organization, and promises savings in board area. This article reviews the state of microprocessors and DRAMs today, explores some of the opportunities and challenges for IRAMs, and finally estimates what computer designers can scale the number of memory chips independently of the number of processors. Most desktop systems have one processor and 4 to 32 DRAM chips, but most server systems have 2 to 16 processors and 32 to 256 DRAMs. Memory systems have standardized on single-in-line memory module (SIMM) or dual-in-line memory module (DIMM) packaging, which allow the end user to scale the amount of memory in a system.

Quantitative evidence of the industry’s success is its size: In 1995, DRAMs were a $37-billion industry, and microprocessors were a $20-billion industry. In addition to financial success, the technologies of these industries have improved at unparalleled rates. DRAM capacity has quadrupled on average every three years since 1976, while microprocessor speed has done the same.

https://doi.org/10.1109/40.592312
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/

A Curious Phenomenon [Kim et al., ISCA 2014]

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
RowHammer [ISCA 2014]

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
[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
¹ Carnegie Mellon University  ² Intel Labs
Memory Scaling Issues Are Real


[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 Yaglıkçı  
giray.yaglikci@safari.ethz.ch  
ETH Zürich  
Zürich, Switzerland

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
Latest RowHammer Lecture

Collapse of the “Galloping Gertie”

Securing the Memory System: The Story of RowHammer - Talk at NYU 23 June 2023 (Prof. Onur Mutlu)

https://www.youtube.com/watch?v=p1pjF8WvERQ
The Push from Circuits and Devices

Main Memory Needs
Intelligent Controllers
An Example Intelligent Controller

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

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


[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Intel Hardware Security Academic Awards Short Talk Slides (pptx) (pdf)]
[Talk Video (22 minutes)]
[Short Talk Video (7 minutes)]
[Intel Hardware Security Academic Awards Short Talk Video (2 minutes)]
[BlockHammer Source Code]

Intel Hardware Security Academic Award Finalist (one of 4 finalists out of 34 nominations)

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

A. Giray Yağlıkçı¹  Minesh Patel¹  Jeremie S. Kim¹  Roknoddin Azizi¹  Ataberk Olgun¹  Lois Orosa¹  Hasan Hassan¹  Jisung Park¹  Konstantinos Kanellopoulos¹  Taha Shahroodi¹  Saugata Ghose²  Onur Mutlu¹

¹ETH Zürich  ²University of Illinois at Urbana–Champaign
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, Noheun 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
Changsik Yoo    Sangjoon Hwang    Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

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

ETH Zürich
RowPress [ISCA 2023]

- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tuğrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu,
"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[RowPress Source Code and Datasets (Officially Artifact Evaluated with All Badges)]

Officially artifact evaluated as available, reusable and reproducible. Best artifact award at ISCA 2023.

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
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
Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

- **Refresh**
  - Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
  - Leakage current of cell access transistors increasing

- **tWR**
  - Contact resistance between the cell capacitor and access transistor increasing
  - On-current of the cell access transistor decreasing
  - Bit-line resistance increasing

- **VRT**
  - Occurring more frequently with cell capacitance decreasing
Call for Intelligent Memory Controllers

DRAM Process Scaling Challenges

- Refresh
  - Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, *Hongzhong Zheng, **John Halbert, **Kuljit Bains, SeongJin Jang, and Joo Sun Choi

Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel
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
Challenge and Opportunity for Future

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

Diagram:
- Processor Core
- Cache
- Memory
- Interconnect
- Database
- Graphs
- Media
- Query
- Results
- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
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


SAFARI

Processing in Memory:
Two Approaches

1. Processing near Memory
2. Processing using Memory
Two PIM Approaches

5.2. Two Approaches: Processing Using Memory (PUM) vs. Processing Near Memory (PNM)

Many recent works take advantage of the memory technology innovations that we discuss in Section 5.1 to enable and implement PIM. We find that these works generally take one of two approaches, which are categorized in Table 1: (1) processing using memory or (2) processing near memory. We briefly describe each approach here. Sections 6 and 7 will provide example approaches and more detail for both.

Table 1: Summary of enabling technologies for the two approaches to PIM used by recent works. Adapted from [341] and extended.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example Enabling Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Using Memory</td>
<td>SRAM \</td>
</tr>
<tr>
<td></td>
<td>Phase-change memory (PCM)\</td>
</tr>
<tr>
<td></td>
<td>Magnetic RAM (MRAM)\</td>
</tr>
<tr>
<td></td>
<td>Resistive RAM (RRAM)/memristors</td>
</tr>
<tr>
<td>Processing Near Memory</td>
<td>Logic layers in 3D-stacked memory\</td>
</tr>
<tr>
<td></td>
<td>Silicon interposers\</td>
</tr>
<tr>
<td></td>
<td>Logic in memory controllers\</td>
</tr>
<tr>
<td></td>
<td>Logic in memory chips (e.g., near bank)\</td>
</tr>
<tr>
<td></td>
<td>Logic in memory modules\</td>
</tr>
<tr>
<td></td>
<td>Logic near caches\</td>
</tr>
<tr>
<td></td>
<td>Logic near/in storage devices</td>
</tr>
</tbody>
</table>
Tutorial on Memory-Centric Computing: Introduction

Geraldo F. Oliveira
Prof. Onur Mutlu

HEART 2024
21 June 2024