Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
13 July 2023
Lightning Talk @ DAC
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

Source: https://youtu.be/Bh13ldwcb0Q?t=283
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

**Cost per Raw Megabase of DNA Sequence**

- Development of high-throughput sequencing (HTS) technologies

**Number of Genomes Sequenced**

- 229,000 (2014)
- 422,000 (2015)
- 952,000 (2016)
- 1,620,000 (2017)

Source: Illumina

*The Economist*

Genome Analysis

1 Sequencing

2 Read Mapping

Data → performance & energy bottleneck

3 Variant Calling

4 Scientific Discovery
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 ▼

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 Data 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]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

Onur Mutlu    Can Firtina
ETH Zürich

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.
Today’s Computing Systems

- Processor centric

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

Data access is the major performance and energy bottleneck

Our current design principles cause great energy waste (and great performance loss)
Most of the system is dedicated to storing and moving data.

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

AMD Ryzen 5000, 2020

Core Count: 8 cores/16 threads

L1 Caches: 32 KB per core

L2 Caches: 512 KB per core

L3 Cache: 32 MB shared

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

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Data Overwhelms Modern Machines

Chrome

TensorFlow Mobile

Data $\rightarrow$ performance & energy bottleneck

VP9

Video Playback

YouTube

Google’s video codec

VP9

Video Capture

YouTube

Google’s video codec
Data Movement Overwhelms Modern Machines


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}  
Aki Kuusela\textsuperscript{3}  
Allan Knies\textsuperscript{3}

Saugata Ghose\textsuperscript{1}  
Eric Shiu\textsuperscript{3}  
Rahul Thakur\textsuperscript{3}  
Parthasarathy Ranganathan\textsuperscript{3}

Youngsok Kim\textsuperscript{2}  
Daehyun Kim\textsuperscript{4,3}  
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

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

Han+, "EIE: Efficient Inference Engine on Compressed Deep Neural Network," ISCA 2016.
A memory access consumes 6400X the energy of a simple integer addition.
Powerful Processors Mostly Wait for Data

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

The Problem

Processing of data is performed far away from the data.
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**
Processing Data
Where It Makes Sense
Process Data Where It Makes Sense

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php
Goal: Processing in Memory/Storage

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

Diagram:
- Processor Core
- Cache
- Memory/Storage
- Database
- Graphs
- Media
- Interconnect
- Query
- Results
Mindset: Memory as an Accelerator

Memory similar to a “conventional” accelerator
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.

Fig. 1. Cellular sorting array I.

https://doi.org/10.1109/T-C.1969.222754
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 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**
Intelligent Memory Controllers Can Avoid Many Failures & Enable Better Scaling
The Pull From the Top (Systems & Apps)

High Performance, Energy Efficient, Sustainable (All at the Same Time)
Processing-in-Memory Landscape Today

And, many other experimental chips and startups
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"

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=PL5Q2soXY2Zi8KzG2CQYRNQV0D0GOBrnKy

- **Youtube Livestream (Spring 2022):**
  - 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
Real PIM Tutorials [ISCA’23, ASPLOS’23, HPCA’23]

- June, March, Feb: 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 hands-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/
Real PIM Tutorial [ISCA 2023]

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

**ISCA 2023 Real-World PIM Tutorial**  
Sunday, June 18, Orlando, Florida

Organizers: Juan Gómez Luna, Onur Mutlu, Ataberk Olgun
Program: https://events.safari.ethz.ch/isca-pim-tutorial/

---

**Tutorial Materials**

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

---

International Symposium on Computer Architecture (ISCA)

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

https://www.youtube.com/live/GIb5EgSrWko?feature=share

https://events.safari.ethz.ch/isca-pim-tutorial/
## March 26: Lectures + Hands-on labs + Invited talks

### 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></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></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></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><a href="#">PDF</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></td>
</tr>
<tr>
<td>4:10pm-4:50pm</td>
<td>Dr. Yongkoo Kwon &amp; Eddy (Chunwook) Park (SK Hynix)</td>
<td>System Architecture and Software Stack for GDDR6-AIM</td>
<td><a href="#">PDF</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></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/)
Real PIM Tutorial [HPCA 2023]

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

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

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

1. Processing using Memory
2. Processing near Memory
A PIM Taxonomy

- **Nature** (of computation)
  - **Using**: Use operational properties of memory structures
  - **Near**: Add logic close to memory structures

- **Technology**
  - Flash, DRAM, SRAM, RRAM, MRAM, FeRAM, PCM, 3D, ...

- **Location**
  - Sensor, Cold Storage, Hard Disk, SSD, Main Memory, Cache, Register File, Memory Controller, Interconnect, ...

- A tuple of the three determines “PIM type”
- One can combine multiple “PIM types” in a system
An Example PIM Type

- Nature: Using
- Technology: DRAM
- Location: Main Memory

Processing using DRAM in Main Memory

Processing using DRAM

- We can support
  - Bulk bitwise AND, OR, NOT, MAJ
  - Bulk bitwise COPY and INIT/ZERO
  - True Random Number Generation

- At low cost

- Using analog computation capability of DRAM
  - Idea: activating (multiple) rows performs computation

- 30-77X performance and energy improvement
  - Seshadri+“RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data,” MICRO 2013.
In-DRAM Bulk Bitwise MAJ/AND/OR + NOT

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

In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology


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

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,

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

*Proceedings of the 50th International Symposium on Microarchitecture (MICRO)*, Boston, MA, USA, October 2017.

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

---

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

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

\(^1\)Microsoft Research India  \(^2\)NVIDIA Research  \(^3\)Intel  \(^4\)ETH Zürich  \(^5\)Carnegie Mellon University
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: Programmability

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Miness Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu, "SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM"

[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}  \hspace{1cm}  *Geraldo F. Oliveira\textsuperscript{1}  \hspace{1cm}  Sven Gregorio\textsuperscript{1}  \hspace{1cm}  João Dinis Ferreira\textsuperscript{1}
Nika Mansouri Ghiasi\textsuperscript{1}  \hspace{1cm}  Minesh Patel\textsuperscript{1}  \hspace{1cm}  Mohammed Alser\textsuperscript{1}  \hspace{1cm}  Saugata Ghose\textsuperscript{3}
Juan Gómez-Luna\textsuperscript{1}  \hspace{1cm}  Onur Mutlu\textsuperscript{1}  \hspace{1cm}  \\
\textsuperscript{1}ETH Zürich  \hspace{1cm}  \textsuperscript{2}Simon Fraser University  \hspace{1cm}  \textsuperscript{3}University 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§ Gabriel Falcao†
Lois Orosa$▼ Mohammad Sadrosadati§
Mohammed Alser§
Juan Gómez-Luna§ Jeremie S. Kim§
Taha Shahroodi‡ Anant Nori*
Geraldo F. Oliveira§

§ETH Zürich †IT, University of Coimbra ▼Galicia Supercomputing Center ‡TU Delft *Intel

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 Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu,

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

[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$^\dag$, Minesh Patel$,^\dag$ A. Giray Yağılkçı$,^\dag$ Haocong Luo$,^\dag$

Jeremie S. Kim$,^\dag$ F. Nisa Bostanci$,^\dag$ Nandita Vijaykumar$,^\dag$ Oguz Ergin$^\dag$

$^\dag$ETH Zürich $^\dag$TOBB University of Economics and Technology $^\dag$University of Toronto

SAFARI
In-Flash Bulk Bitwise Execution

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

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

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

Jisung Park§∇ Roknoddin Azizi§ Geraldo F. Oliveira§ Mohammad Sadrosadati§
Rakesh Nadig§ David Novo† Juan Gómez-Luna§ Myungsuk Kim‡ Onur Mutlu§
§ETH Zürich  ∇ POSTECH  †LIRMM, Univ. Montpellier, CNRS  ‡Kyungpook National University

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§†, Juan Gómez Luna§, Hasan Hassan§, Konstantinos Kanellopoulos§, Oğuz Ergin†, Onur Mutlu§, Behzad Salami§*

§ETH Zürich, †TOBB ETÜ, *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
Building a PiDRAM Prototype

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

Rebuilding Steps

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

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

Generating DDR3 Controller IP sources

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

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

https://github.com/cmu-safari/pidram
https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s
More on PiDRAM

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oğuz Ergin, and Onur Mutlu,

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


[arXiv version]

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

[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (40 minutes)]
[PiDRAM Source Code]

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

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

§ETH Zürich  †TOBB University of Economics and Technology
Eliminating the Adoption Barriers

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

1. **Applications, systems & 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
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}

\textit{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](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=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0G0BrnK](https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0G0BrnK)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPy](https://www.youtube.com/watch?v=9e4Chnwdovo&list=PL5Q2soXY2Zi-841fUYYUK9EsXKhQKRPy)

- **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
Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
13 July 2023
Lightning Talk @ DAC
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]