

Memory-Centric Computing

Geraldo F. Oliveira

Onur Mutlu

omutlu@gmail.com

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

29 October 2023

Real-World PIM Tutorial Opening Talk @ MICRO-56

SAFARI

ETH zürich

Carnegie Mellon

Computing

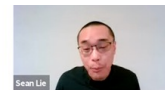
is Bottlenecked by Data

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

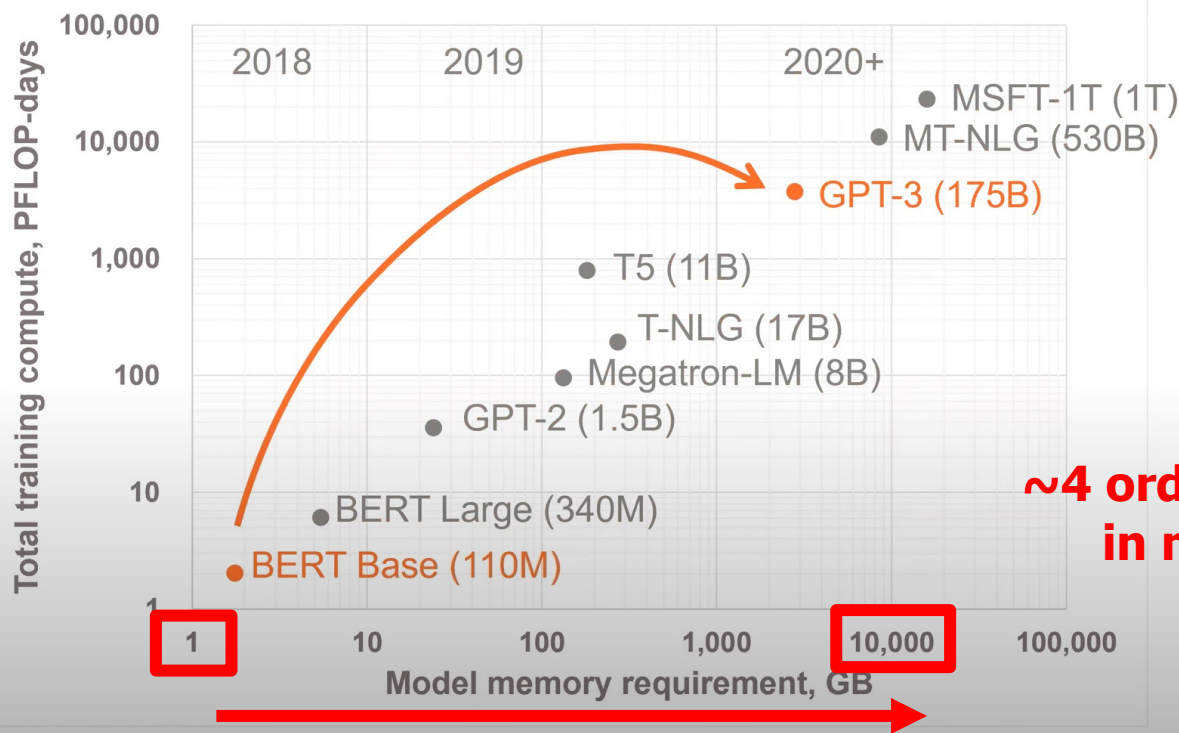
- Important workloads are all data intensive
- They require rapid and efficient processing of large amounts of data
- Data is increasing
 - We can generate more than we can process
 - We need to perform more sophisticated analyses on more data

Huge Demand for Performance & Efficiency

Exponential Growth of Neural Networks



Memory and compute requirements



1800x more compute
In just 2 years

Tomorrow, **multi-trillion** parameter models

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

Data is Key for Future Workloads



□□□□ □□□□ □□□□ **D** □□□□□□□□□□

□□□□□, E□□□□□□□□□□;

C□□□□□ (□□□□□□), □□□□ C□□□□□



G □□□□□□□□□□ □□□□□□□□□□□□□□□□

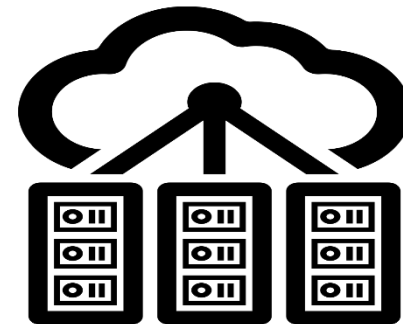
□□□□□, □□□□ C□□□□; □□□□□□□□□□□, F□□□□□□□



□□□□ □□□□ □□□□ **D** □□□□ **A** □□□□□□□□□□

□□□□□ (□□□□□□), □□□□ C□□□□□;

A□□□□□, BDC□□□□□□□□□□



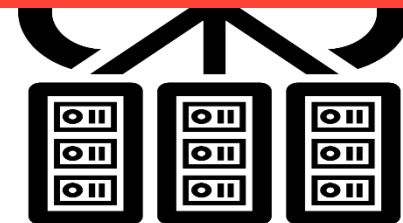
D □□□□□□□□□□ □□□□□□□□□□

□□□□□□□□ (G□□□□□□), □□□□□□□□□□

Data Overwhelms Modern Machines



Data → performance & energy bottleneck



⊕ (), || C;
A, BDC

⊕ (G), |CA

Data Overwhelms Modern Machines



Data → performance & energy bottleneck

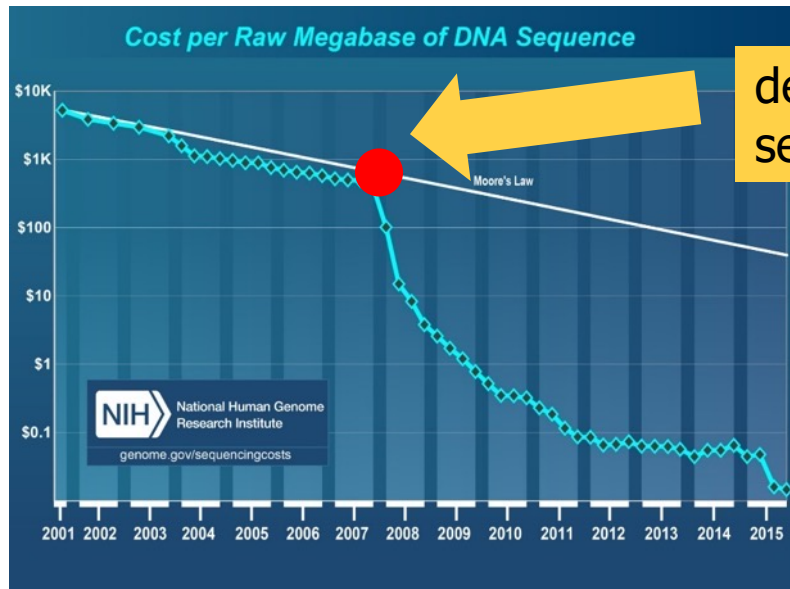
VP9



VP9

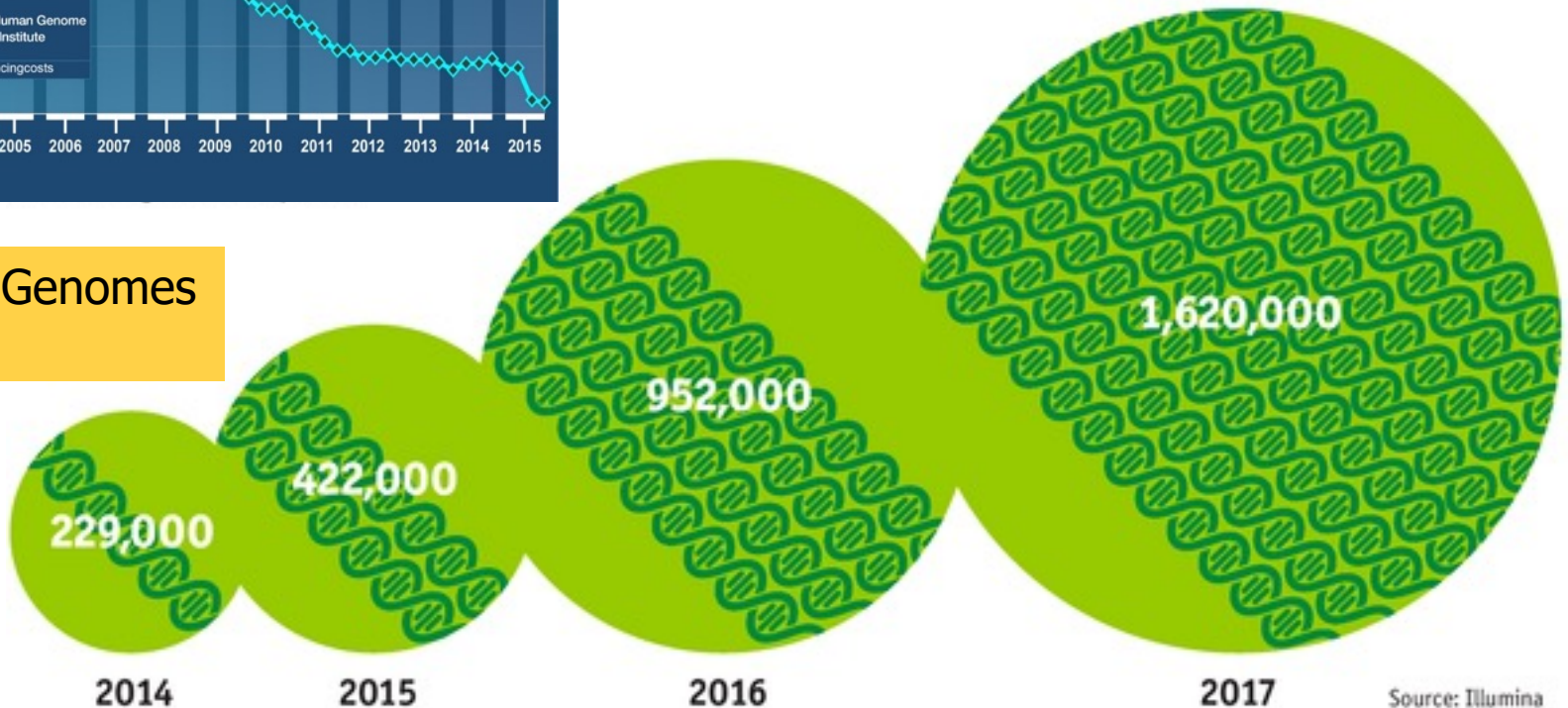


Data is Key for Future Workloads

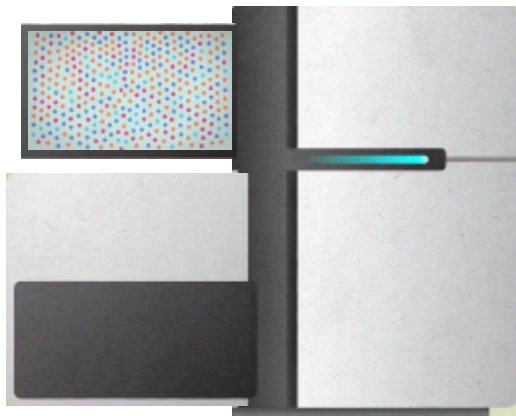


development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

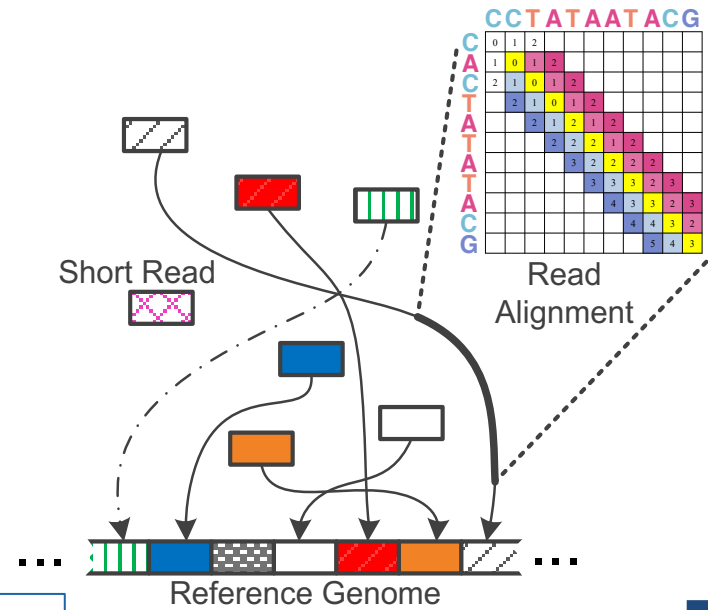


The Economist



Billions of Short Reads

ATATATACGTA
 TTTAGTACGTACGT
 ATACGTA
 CG CCCCTACGTA
 CGTACTAGTACGT
 TTAGTACGTACGT
 TACGTA
 TACGTA
 TTTAAACGTA
 CGTACTAGTACGT
 GGGAGTACGTACGT



1 Sequencing

Genome Analysis

Read Mapping 2

Data → performance & energy bottleneck

read4: CGCTTCCAT
 read5: CCATGACGC
 read6: TTCCATGAC



3 Variant Calling

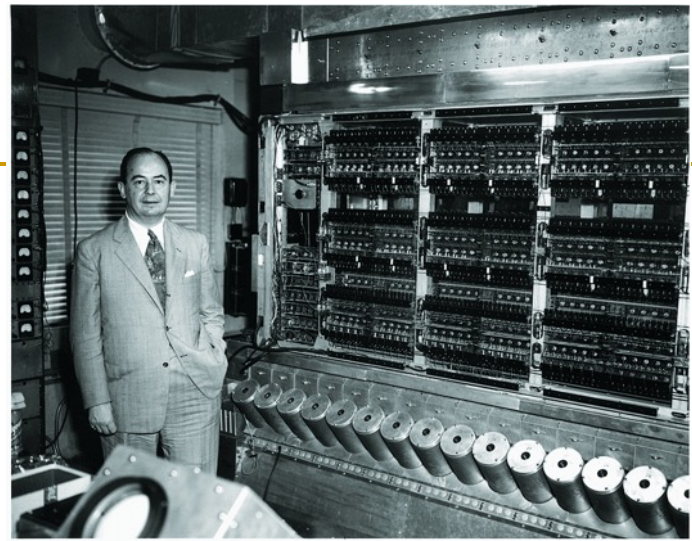
Scientific Discovery 4

Data Overwhelms Modern Machines ...

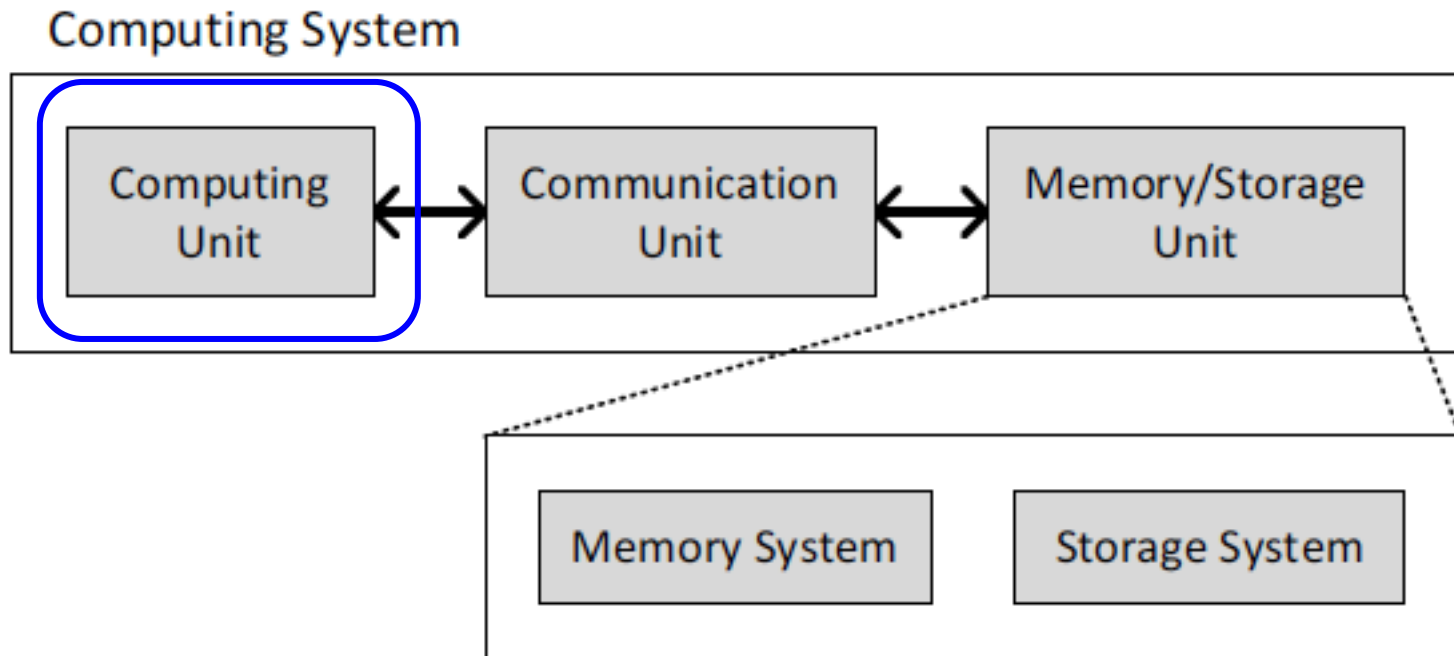
- Storage/memory capability
- Communication capability
- Computation capability
- Greatly impacts robustness, energy, performance, cost

A Computing System

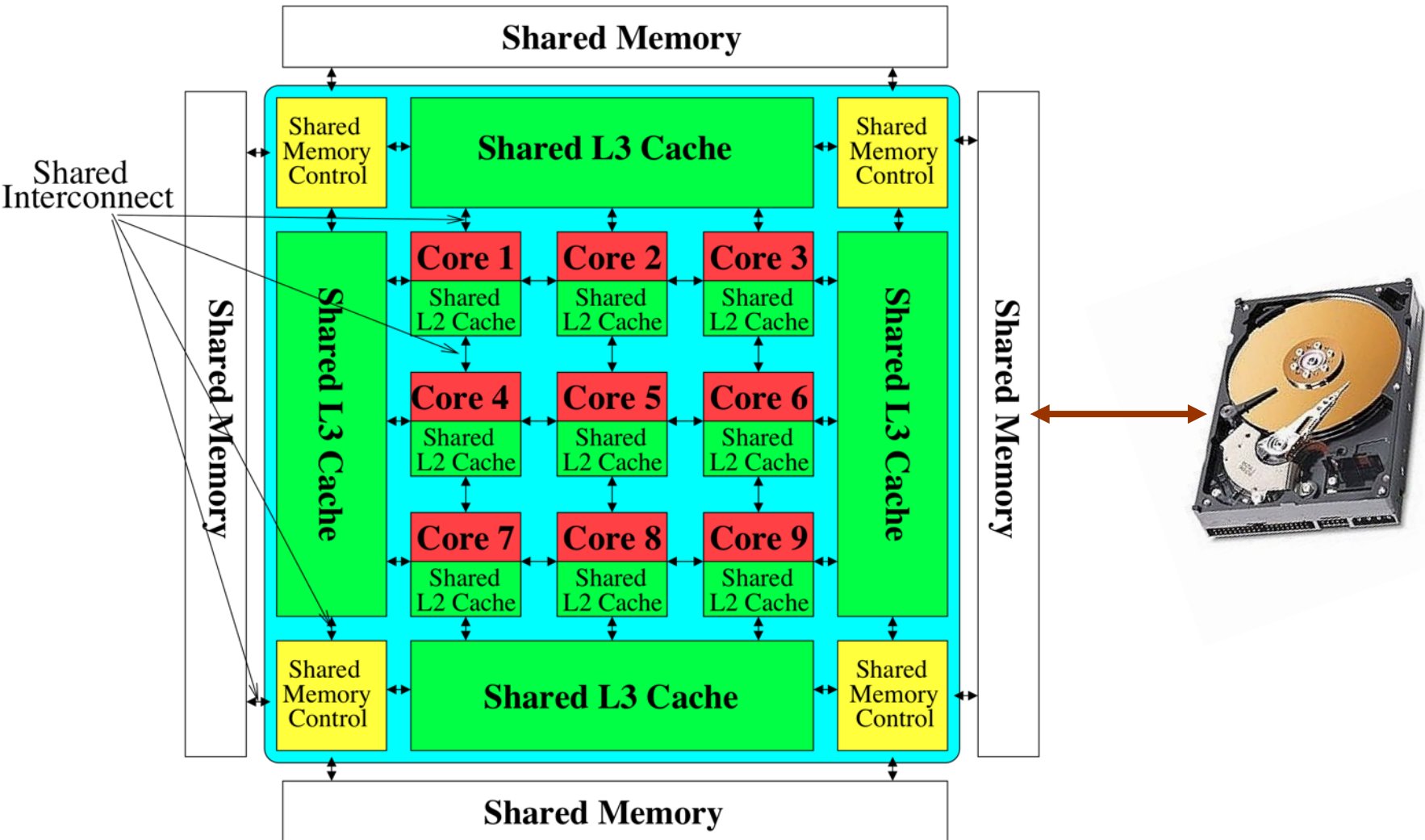
- Three key components
- Computation
- Communication
- Storage/memory



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

A Solution: Deeper and Larger Memory Hierarchies

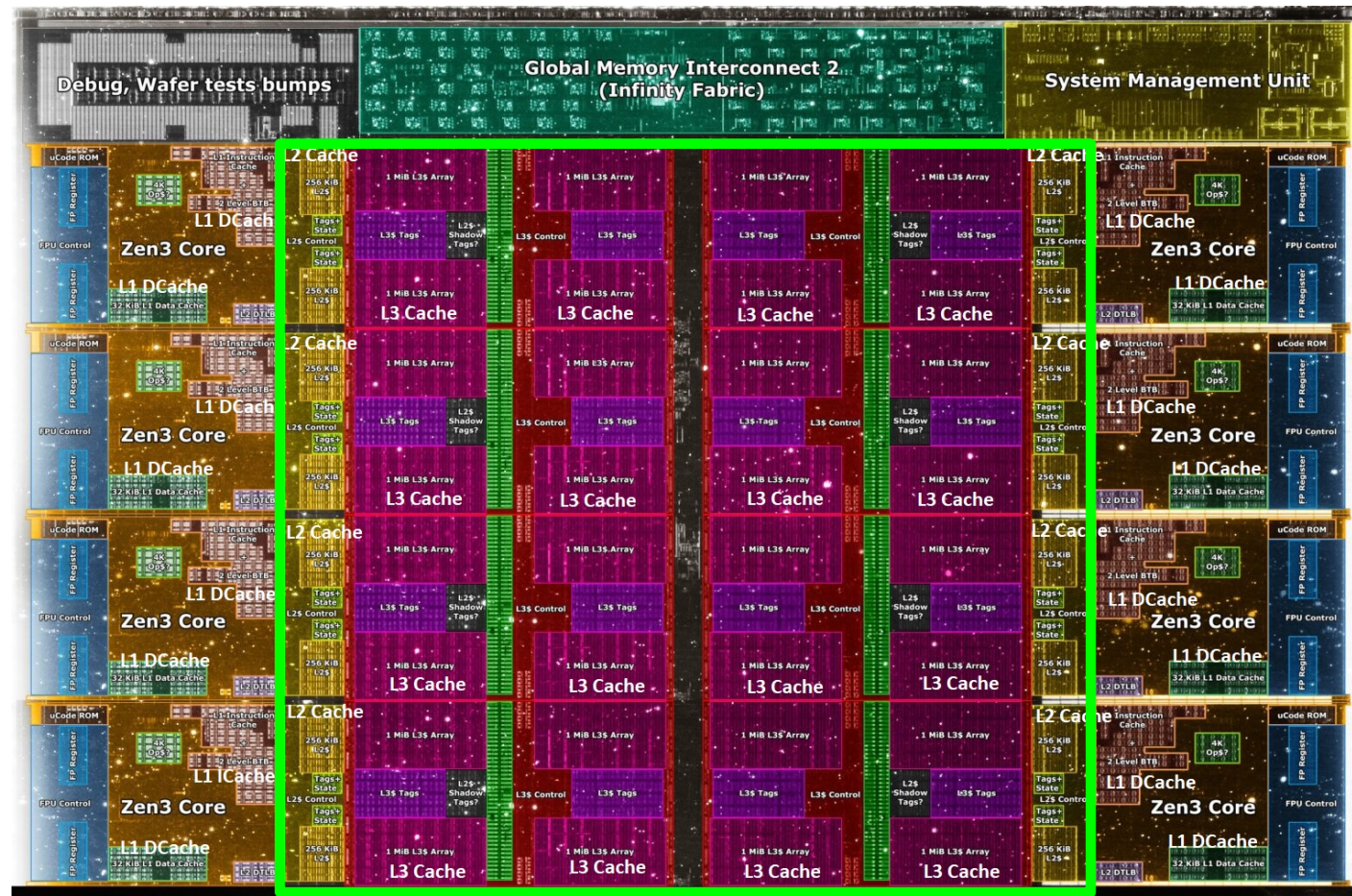
Core Count:
8 cores/16 threads

L1 Caches:
32 KB per core

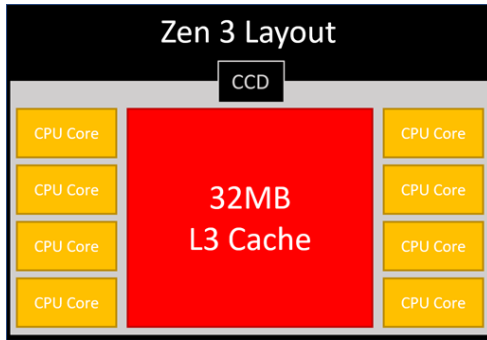
L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

AMD Ryzen 5000, 2020



AMD's 3D Last Level Cache (2021)

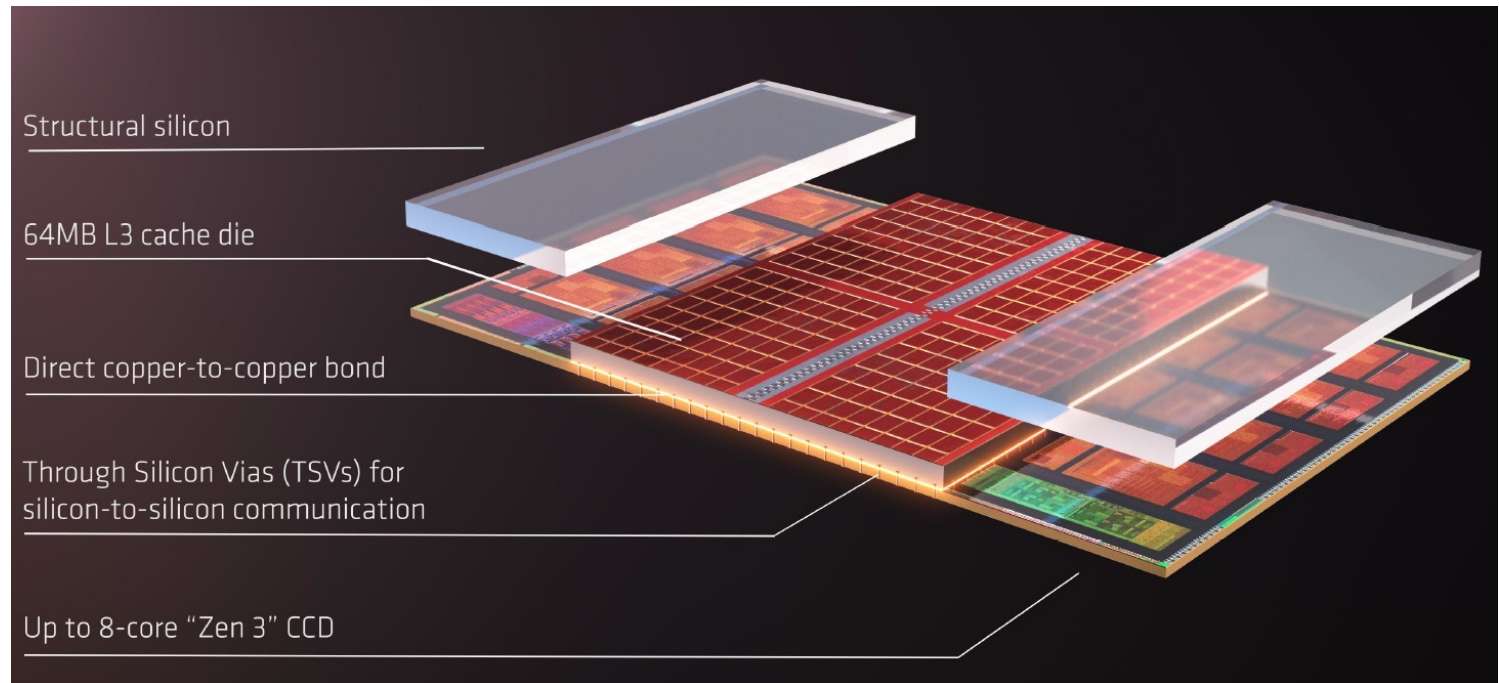


<https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2>

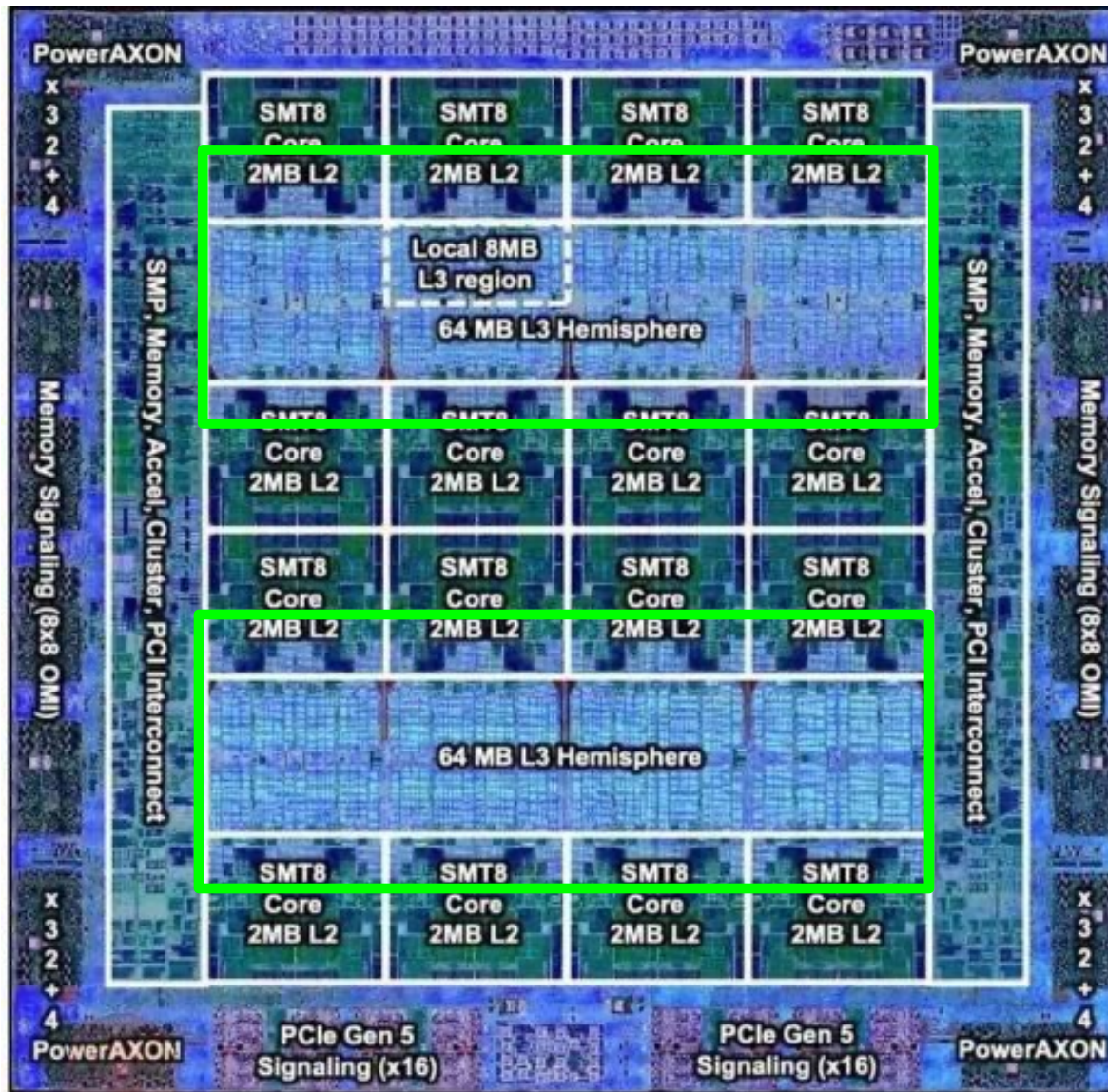
AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

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



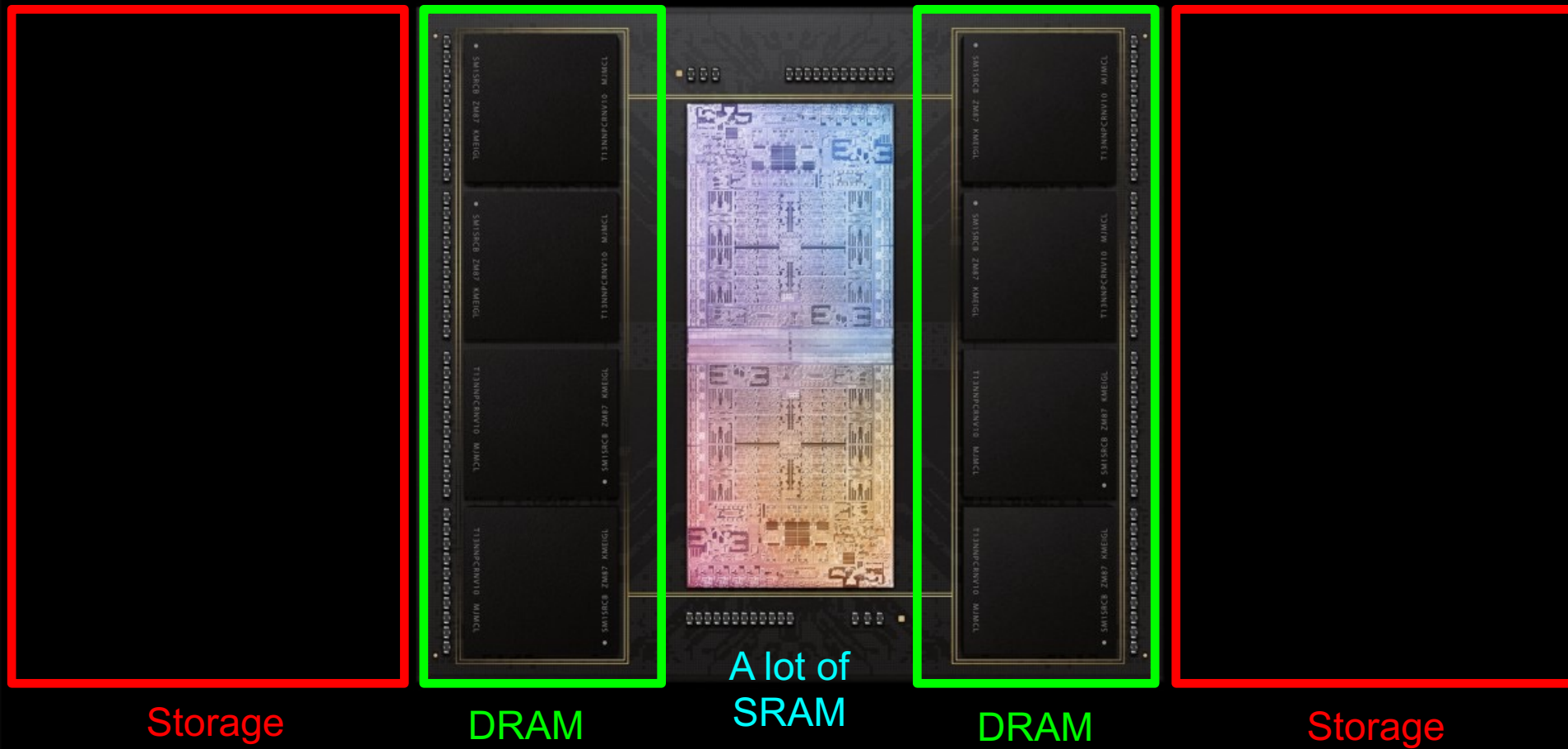
IBM POWER10,
2020

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)

Data Movement Overwhelms Modern Machines

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, "[Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks](#)" *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Williamsburg, VA, USA, March 2018.

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

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

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

The Problem

Data access is the major performance and energy bottleneck

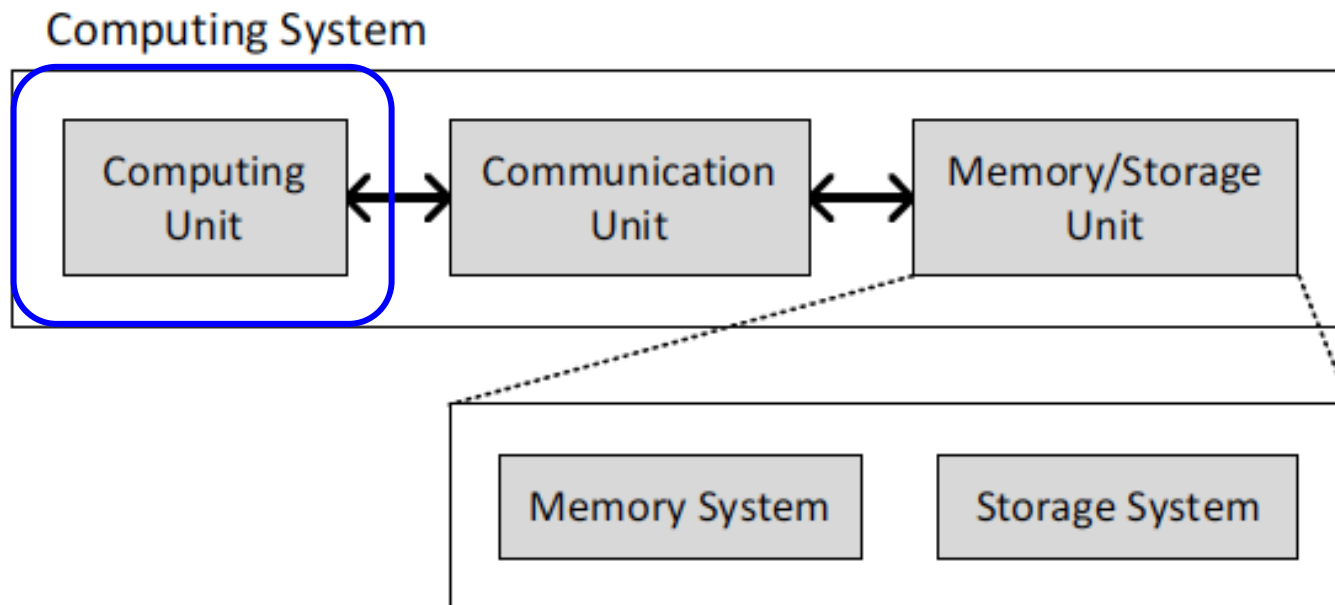
Our current
design principles
cause great energy waste
(and great performance loss)

The Problem

Processing of data
is performed
far away from the data

Today's Computing Systems

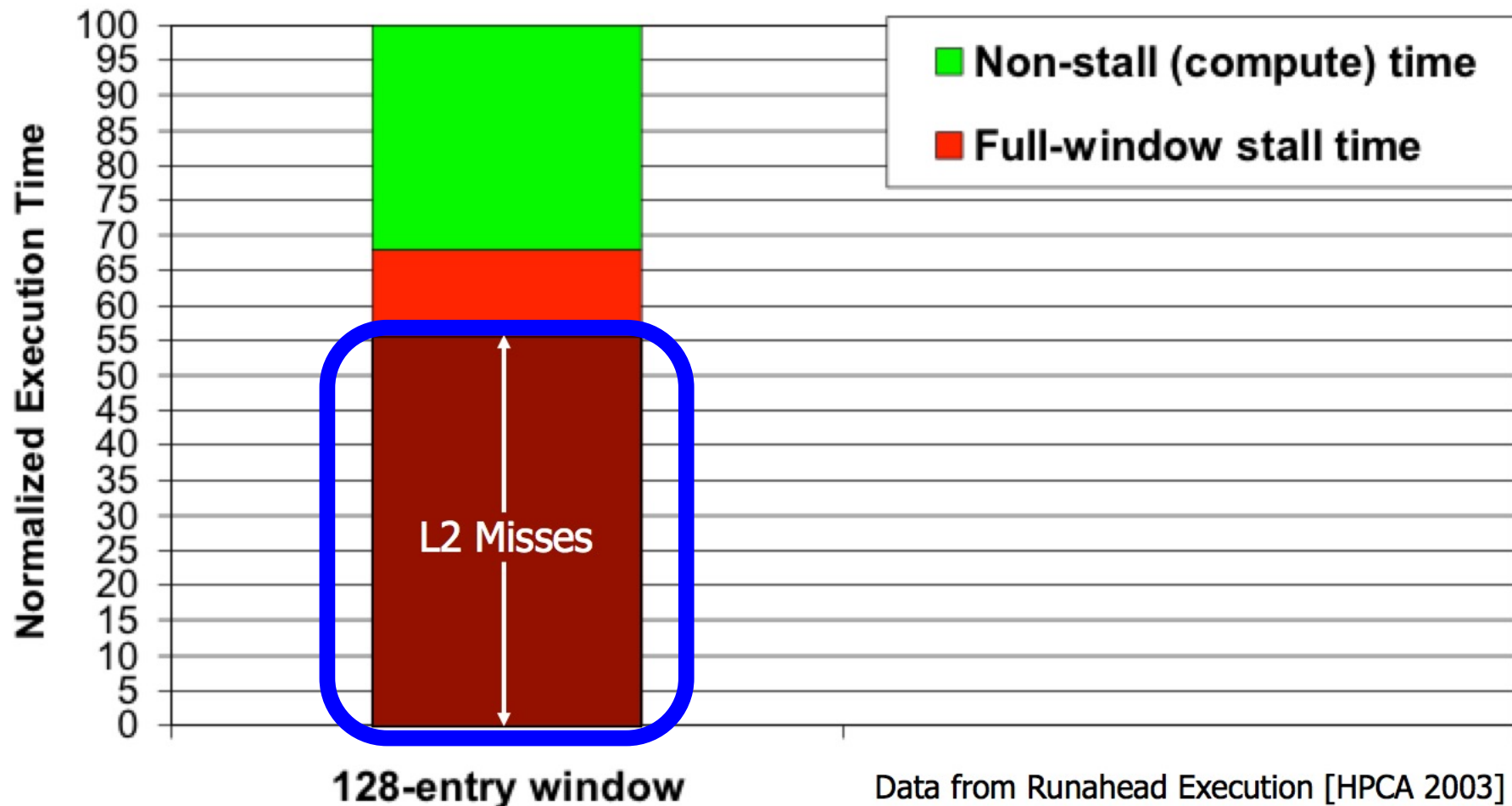
- Processor centric
- All data processed in the processor → at great system cost



Yet ...

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

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



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"
Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA), pages 129-140, Anaheim, CA, February 2003. [Slides \(pdf\)](#)
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 §

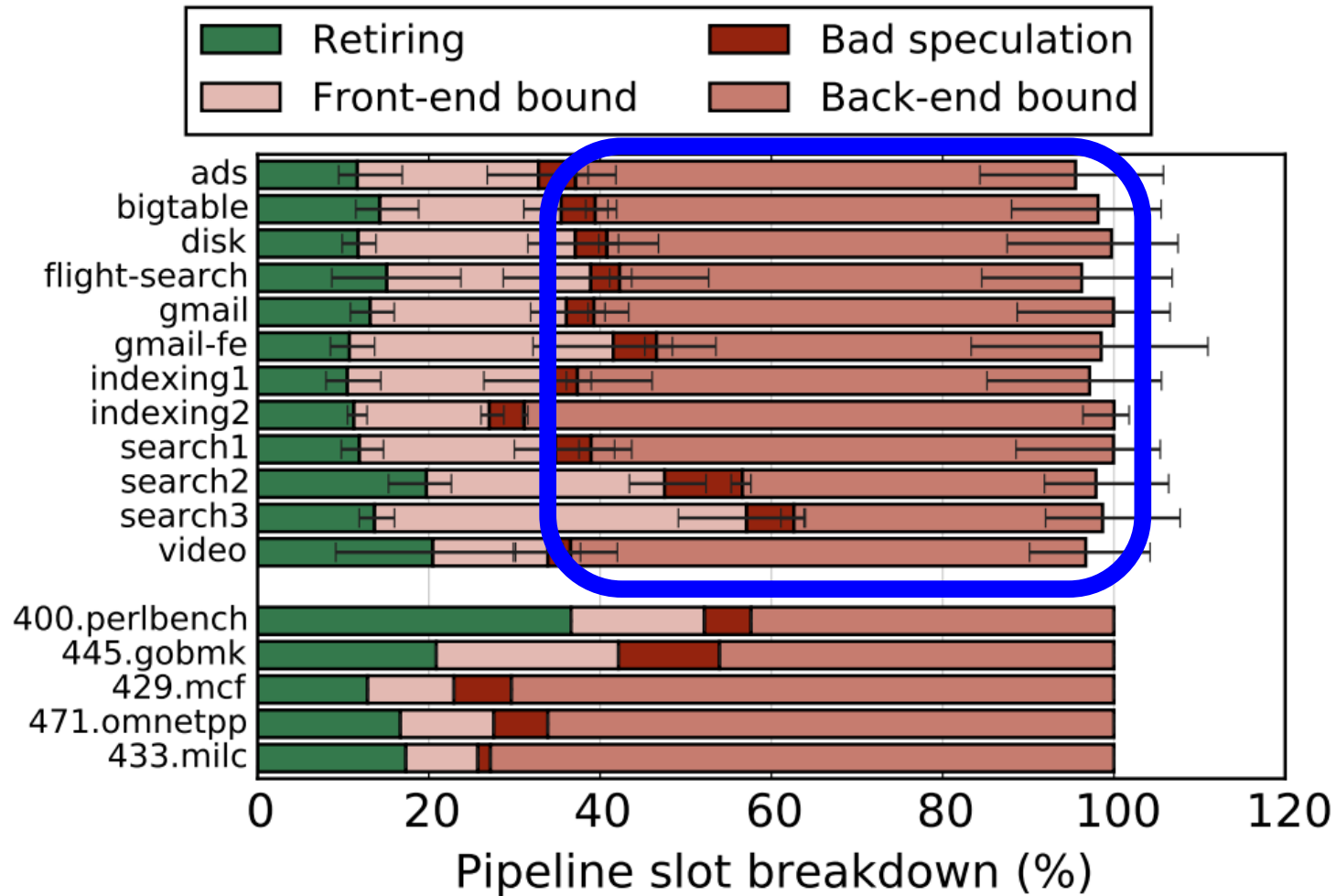
§ECE Department
The University of Texas at Austin
{onur,patt}@ece.utexas.edu

†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 Performance Perspective (Today)

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

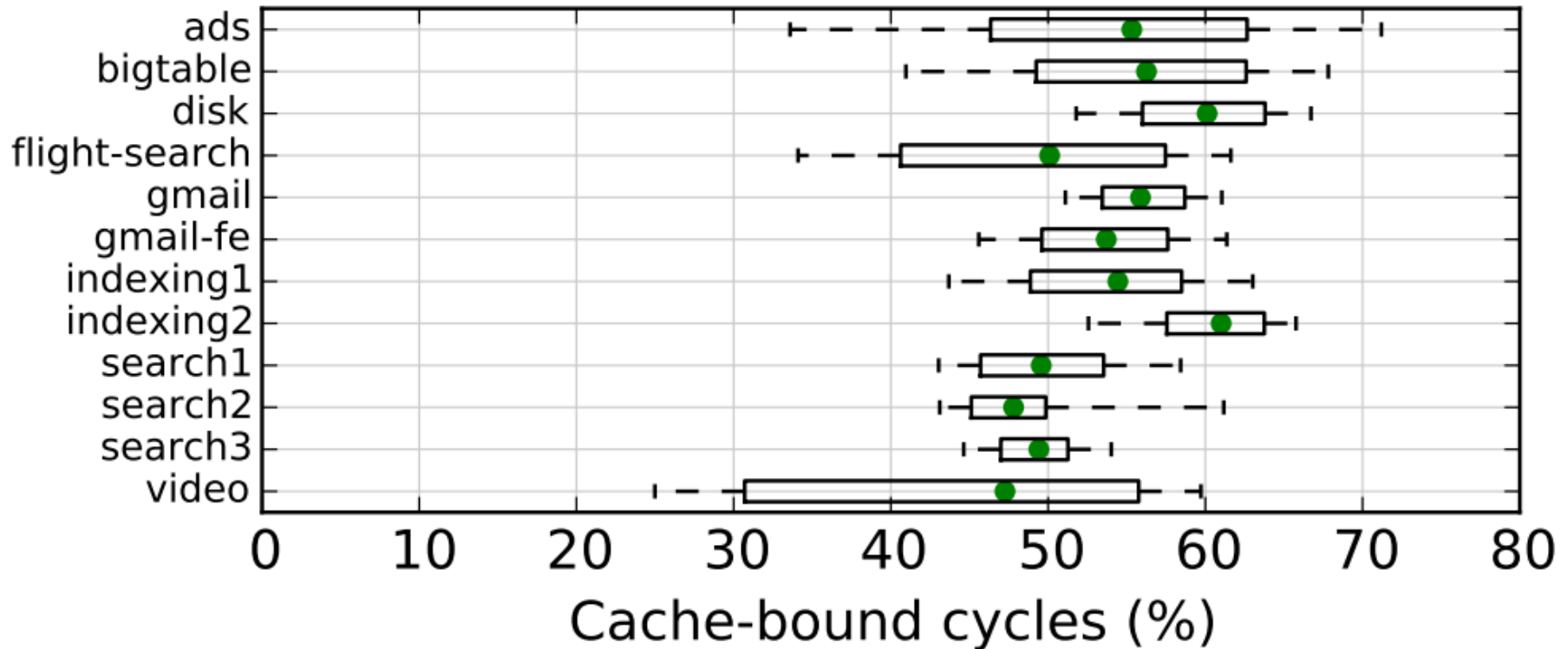


Figure 11: Half of cycles are spent stalled on caches.

Perils of Processor-Centric Design

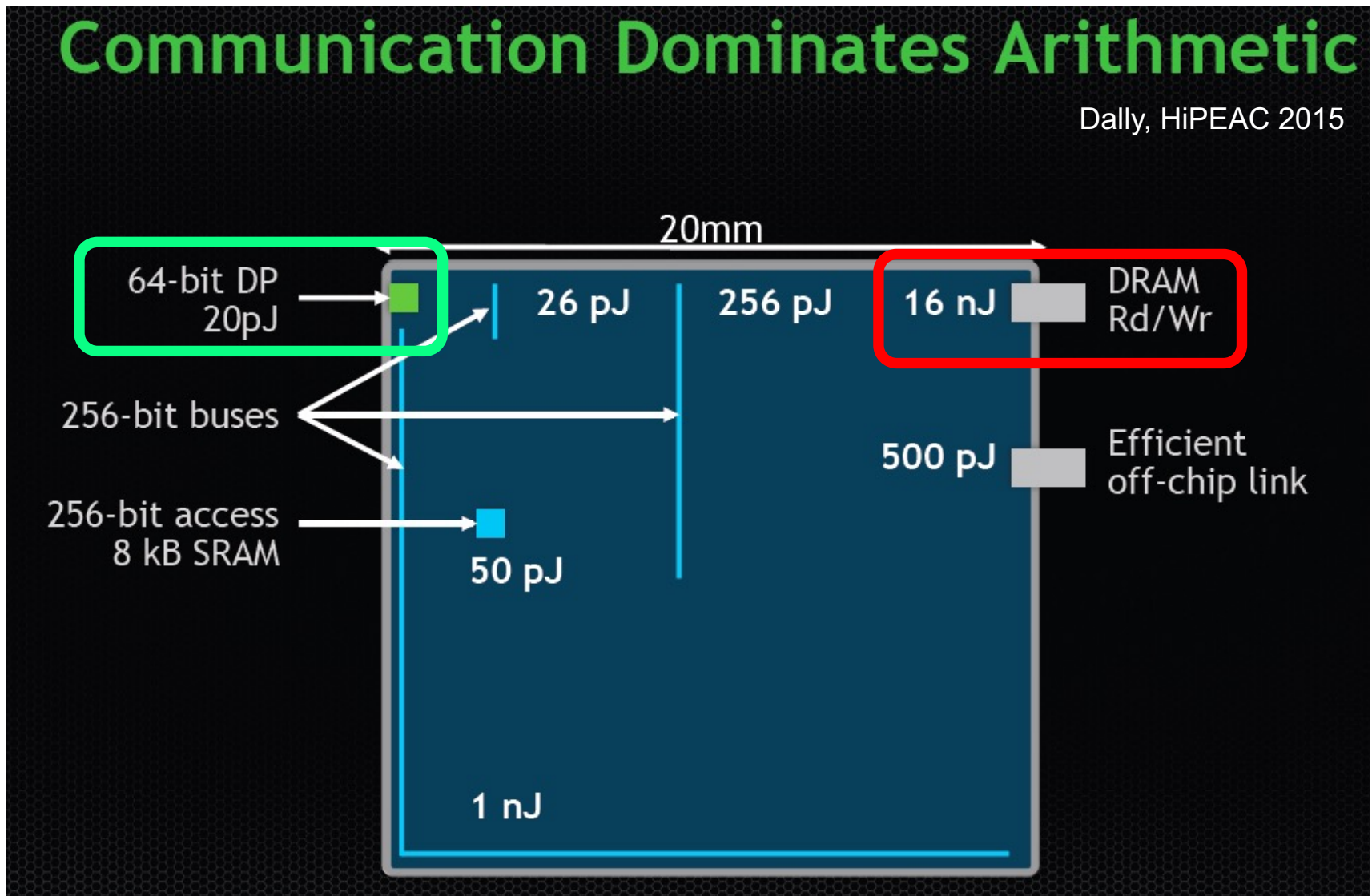
- **Grossly-imbalanced systems**
 - ❑ Processing done only in **one place**
 - ❑ All else just stores and moves data: **data moves a lot**
 - Energy inefficient
 - Low performance
 - Complex

- **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

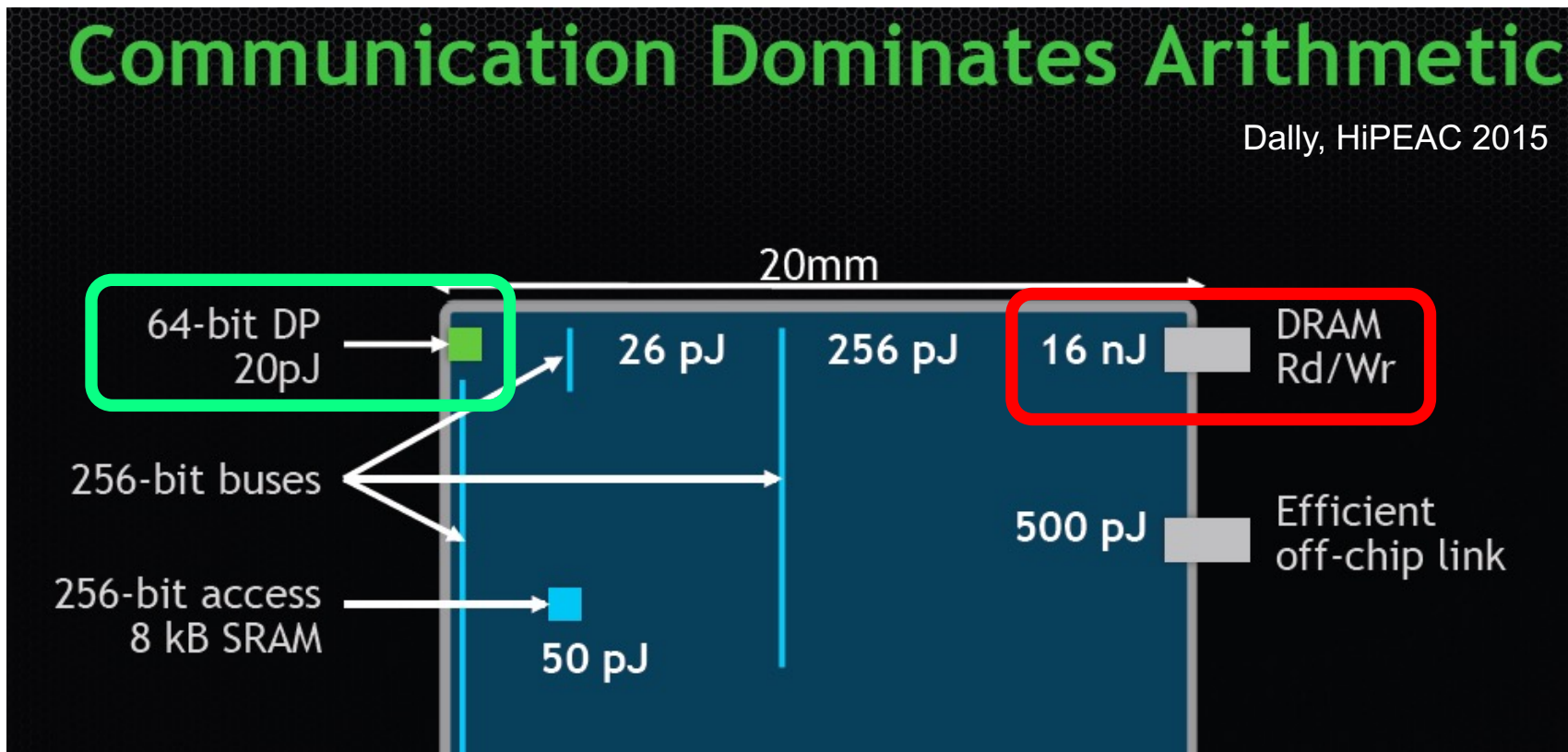
Dally, HiPEAC 2015



Data Movement vs. Computation Energy

Communication Dominates Arithmetic

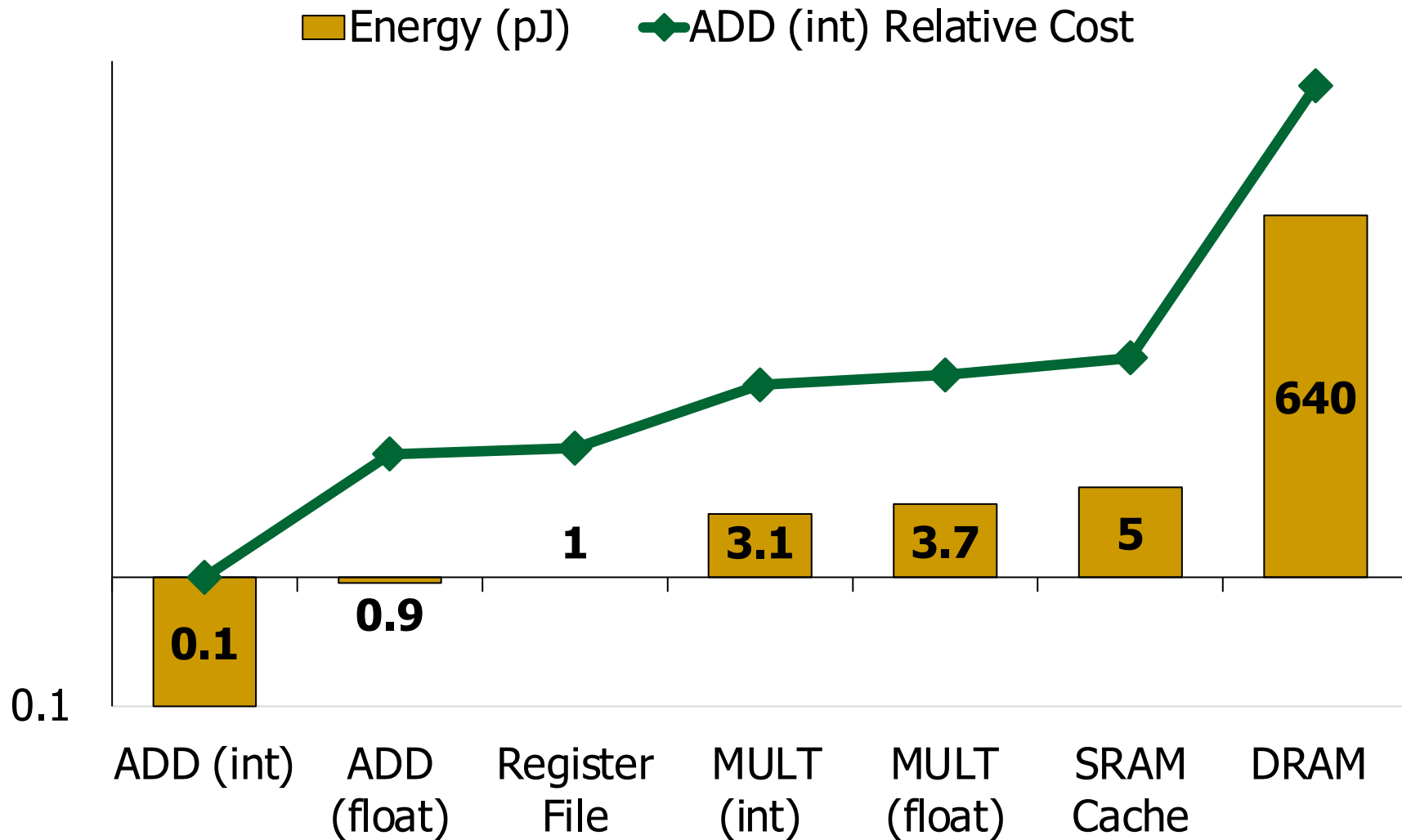
Dally, HiPEAC 2015



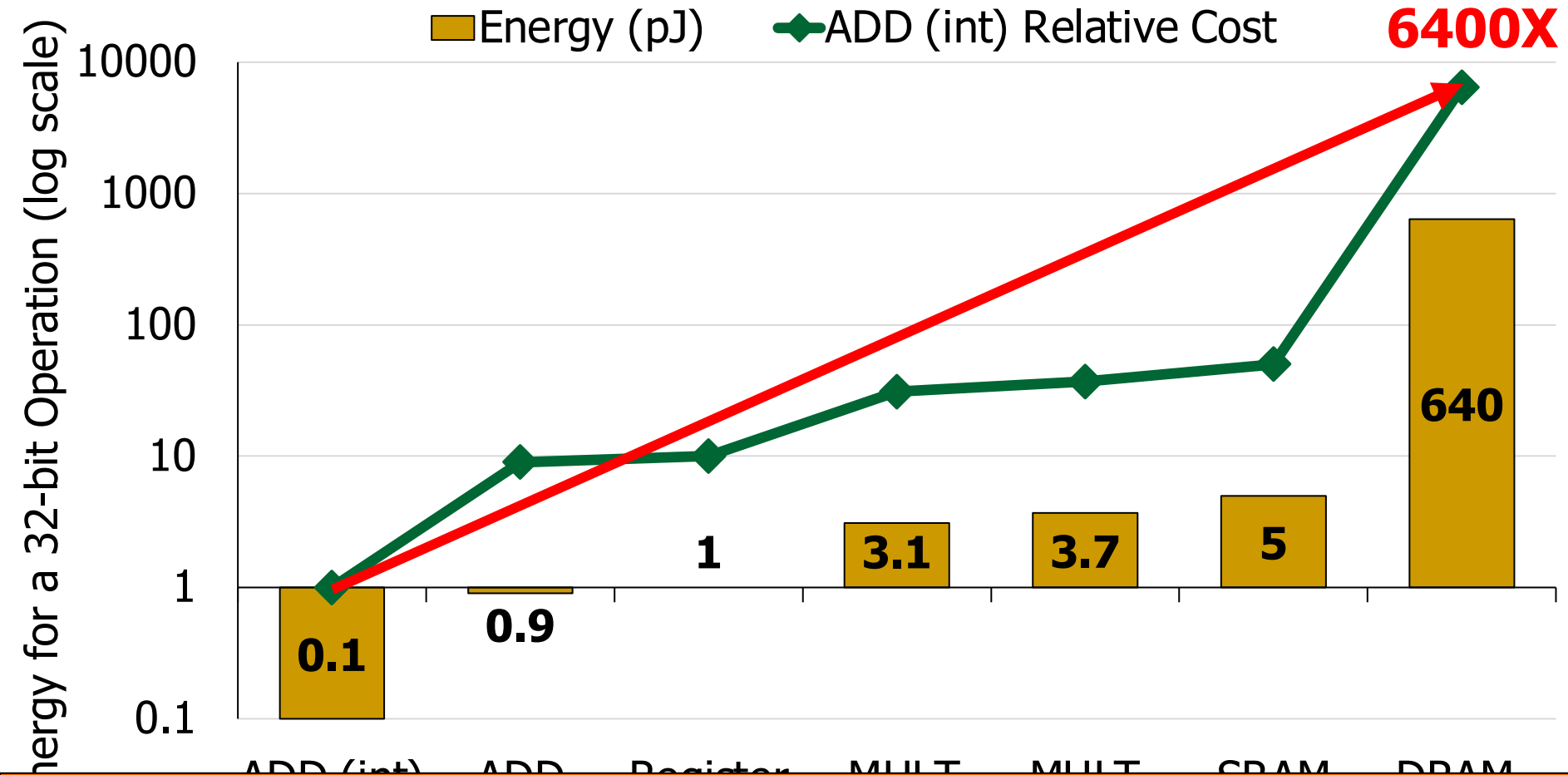
A memory access consumes $\sim 100-1000X$ the energy of a complex addition

Data Movement vs. Computation Energy

Energy for a 32-bit Operation (log scale)



Data Movement vs. Computation Energy

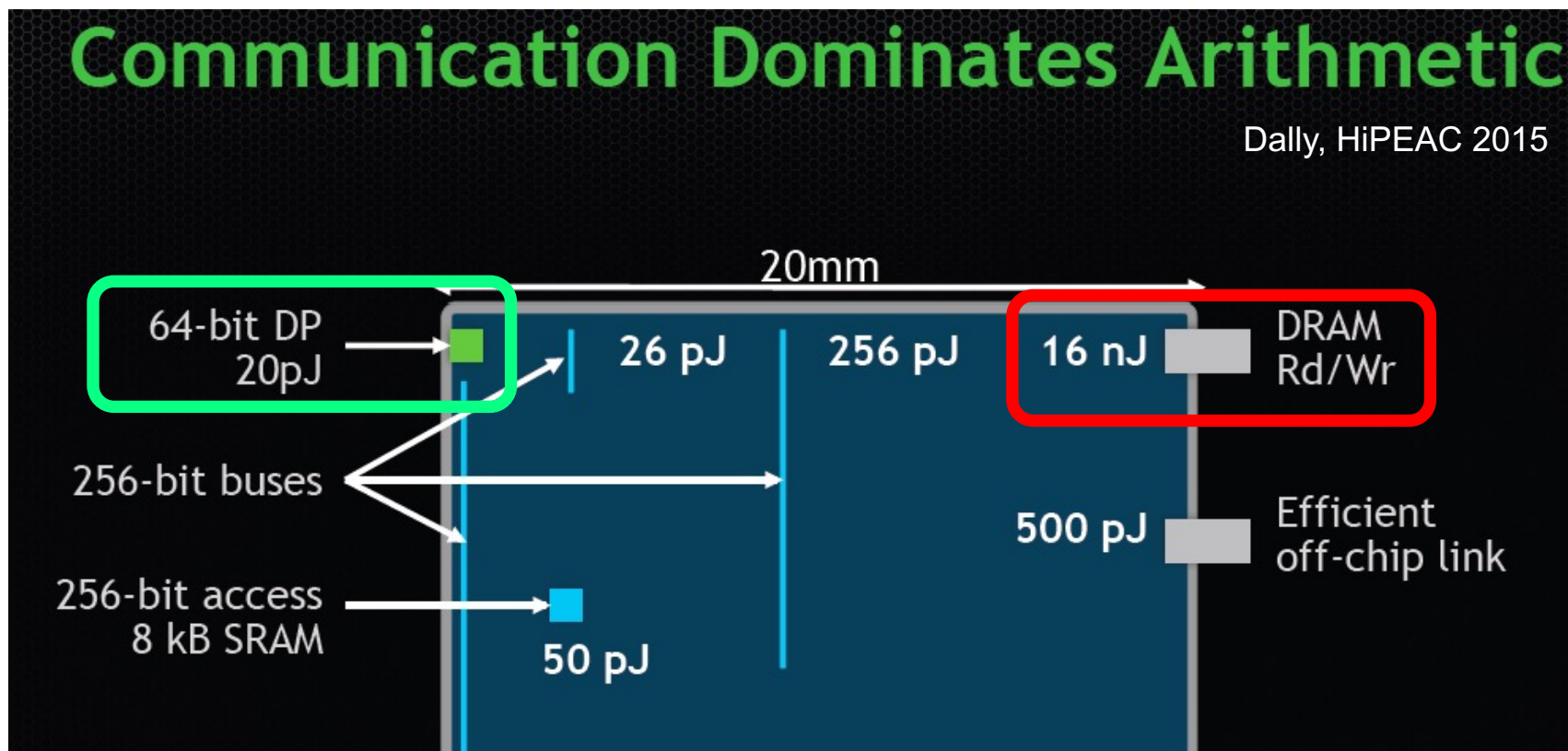


A memory access consumes 6400X the energy of a simple integer addition

We Do Not Want to Move Data!

Communication Dominates Arithmetic

Dally, HiPEAC 2015



A memory access consumes $\sim 100-1000X$
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

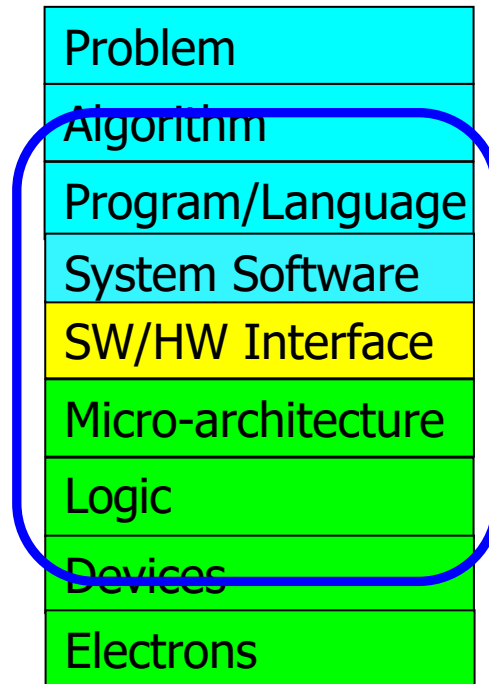
A Blueprint for Fundamentally Better Architectures

- Onur Mutlu,
"Intelligent Architectures for Intelligent Computing Systems"
Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (DATE), Virtual, February 2021.
[[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

We Need to Revisit the Entire Stack



We can get there step by step

Data-Centric (Memory-Centric) Architectures

Data-Centric Architectures: Properties

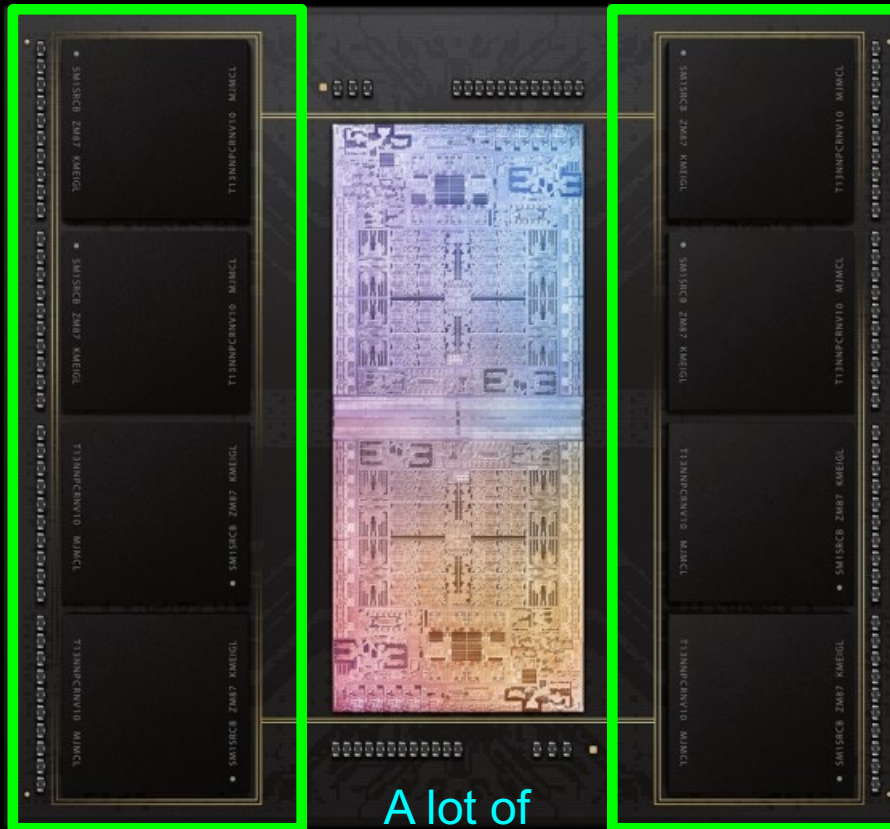
- **Process data where it resides** (where it makes sense)
 - Processing in and near memory structures
- **Low-latency and low-energy data access**
 - Low latency memory
 - Low energy memory
- **Low-cost data storage and processing**
 - High capacity memory at low cost: hybrid memory, compression
- **Intelligent data management**
 - Intelligent controllers handling robustness, security, cost, perf.

Processing Data

Where It Makes Sense

Process Data Where It Makes Sense

Sensors



A lot of
SRAM

Storage

DRAM

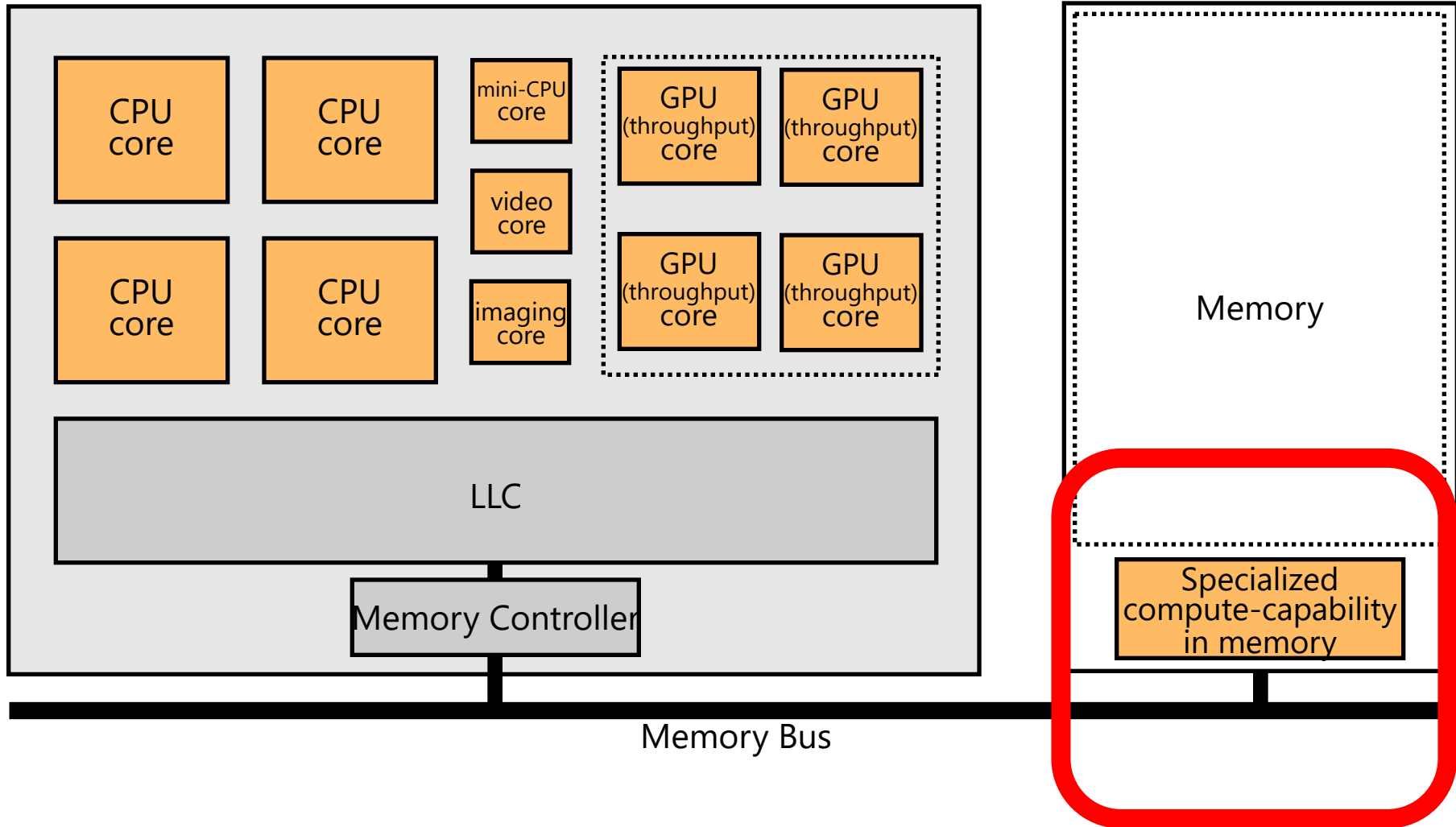
DRAM

Storage

Apple M1 Ultra System (2022)

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)

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

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

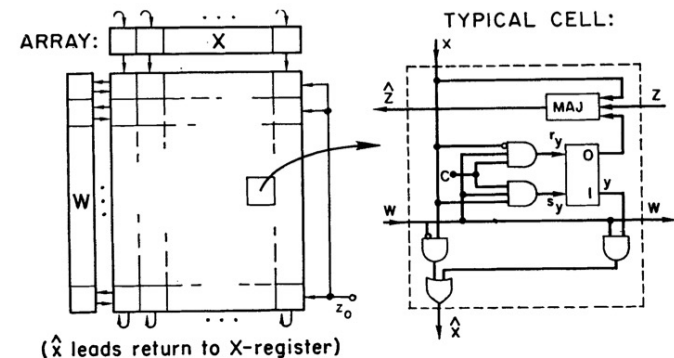
Cellular Logic-in-Memory Arrays

WILLIAM H. KAUTZ, MEMBER, IEEE

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

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

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



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

Fig. 1. Cellular sorting array I.

Processing in Memory: An Old Idea (II)

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

A Logic-in-Memory Computer

HAROLD S. STONE

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

Processing in Memory: An Old Idea (III)

- Patterson et al., “A Case for Intelligent RAM,” IEEE Micro 1997.

A CASE FOR INTELLIGENT RAM

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 esti-

puter 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

Why In-Memory Computation Today?

- **Huge problems with Memory Technology**

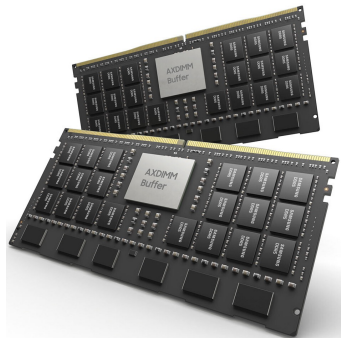
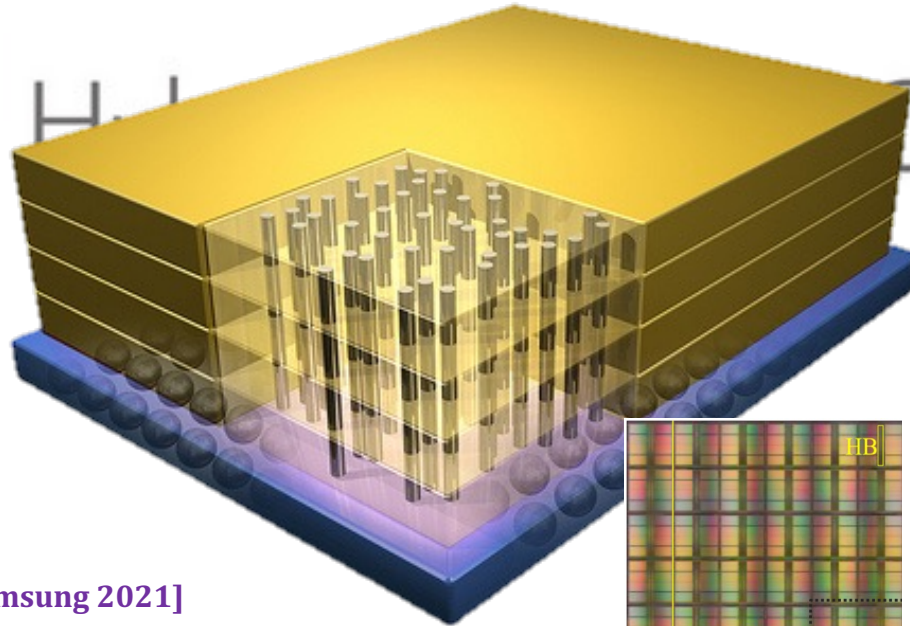
- Memory technology scaling is not going well (e.g., RowHammer)
- Many scaling issues demand intelligence in memory

- **Huge demand from Applications & Systems**

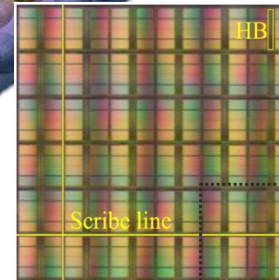
- Data access bottleneck
- Energy & power bottlenecks
- Data movement energy dominates computation energy
- Need all at the same time: performance, energy, sustainability
- We can improve all metrics by minimizing data movement

- **Designs are squeezed in the middle**

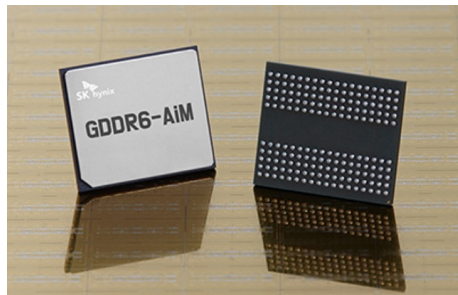
Processing-in-Memory Landscape Today



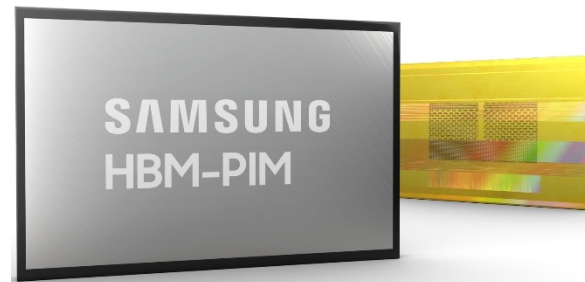
[Samsung 2021]



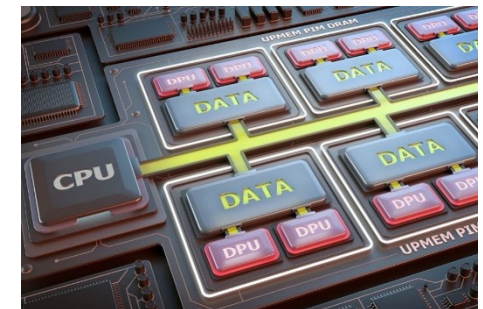
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]



[UPMEM 2019]

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]

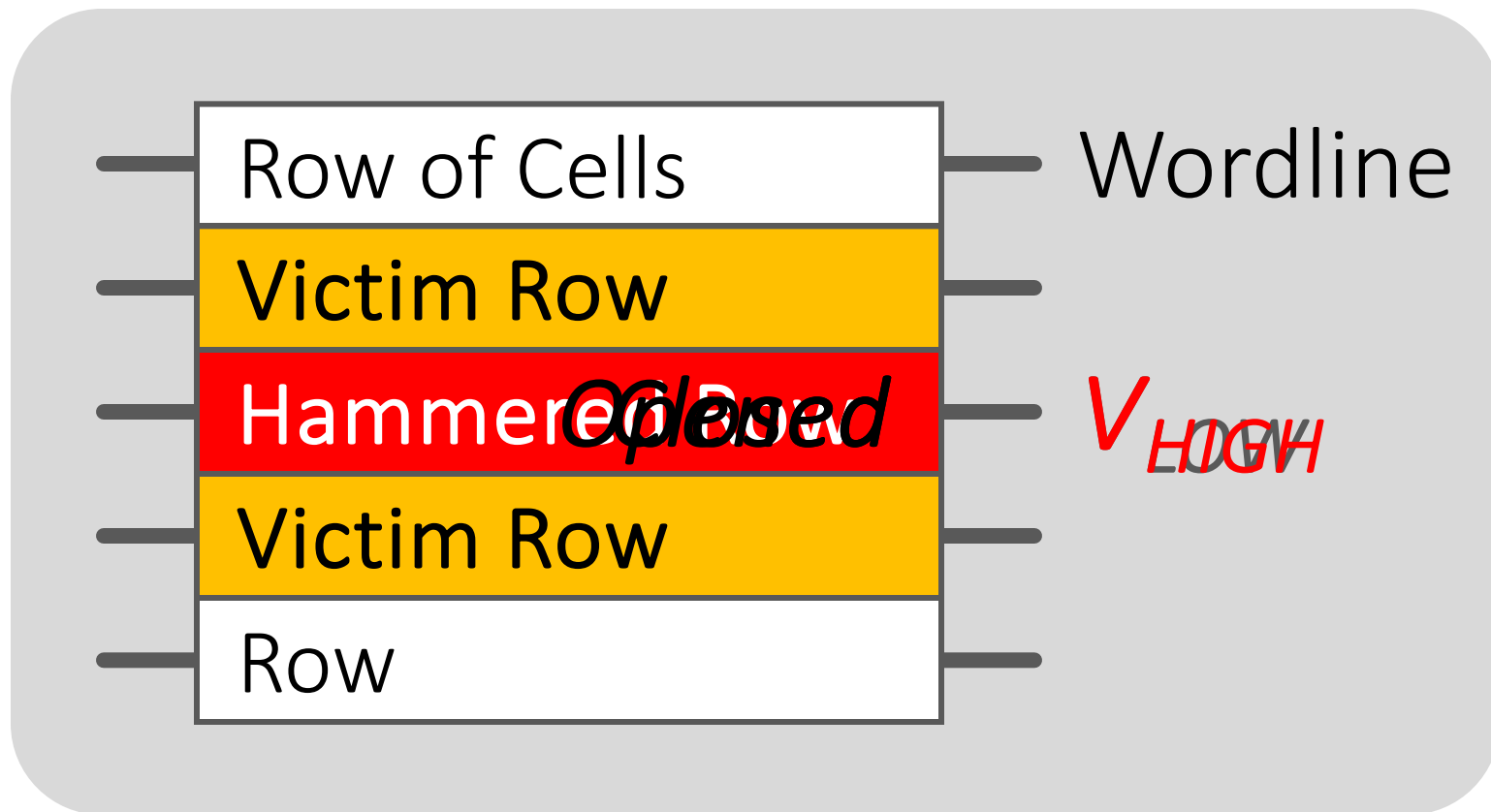
One can
predictably induce errors
in most DRAM memory chips

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



Rowhammer

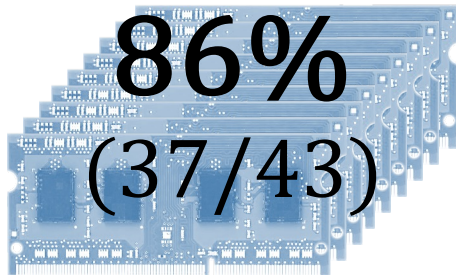
Modern Memory is Prone to Disturbance Errors



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**

Most DRAM Modules Are Vulnerable

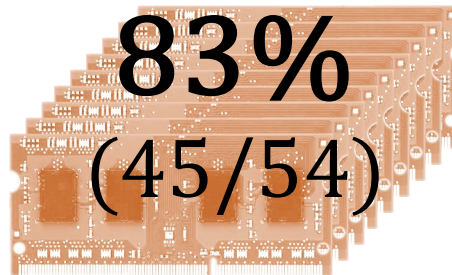
A company



Up to
 1.0×10^7

errors

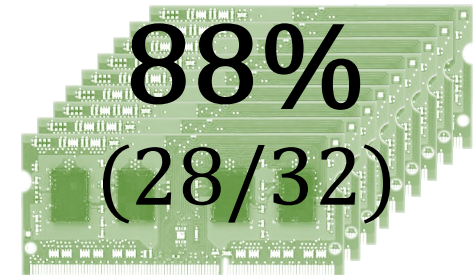
B company



Up to
 2.7×10^6

errors

C company



Up to
 3.3×10^5

errors

The RowHammer Vulnerability

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

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE
18276



TWEET

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"

Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.

[[Slides \(pptx\) \(pdf\)](#)] [[Lightning Session Slides \(pptx\) \(pdf\)](#)] [[Source Code and Data](#)] [[Lecture Video](#) (1 hr 49 mins), 25 September 2020]

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

Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 ([Retrospective \(pdf\) Full Issue](#)).

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 and Jeremie Kim,
["RowHammer: A Retrospective"](#)
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.
[[Preliminary arXiv version](#)]
[[Slides from COSADE 2019 \(pptx\)](#)]
[[Slides from VLSI-SOC 2020 \(pptx\) \(pdf\)](#)]
[[Talk Video](#) (1 hr 15 minutes, with Q&A)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

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"

Invited Tutorial at 14th IEEE Electron Devices Society International Memory Workshop (IMW), Dresden, Germany, May 2022.

[Slides (pptx)(pdf)]

[Tutorial Video (57 minutes)]



Security Aspects of DRAM
The Story of RowHammer

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
15 May 2022
IMW Tutorial

SAFARI ETH zürich Carnegie Mellon

Recent Premieres

The Story of RowHammer – Invited Tutorial at IMW 2022 (Intl. Memory Workshop) - Onur Mutlu

598 views • Premiered Jul 27, 2022

👍 19 🗑 DISLIKE ➦ SHARE ⬇ DOWNLOAD 🗑 CLIP 📌 SAVE ...



Onur Mutlu Lectures
27.6K subscribers

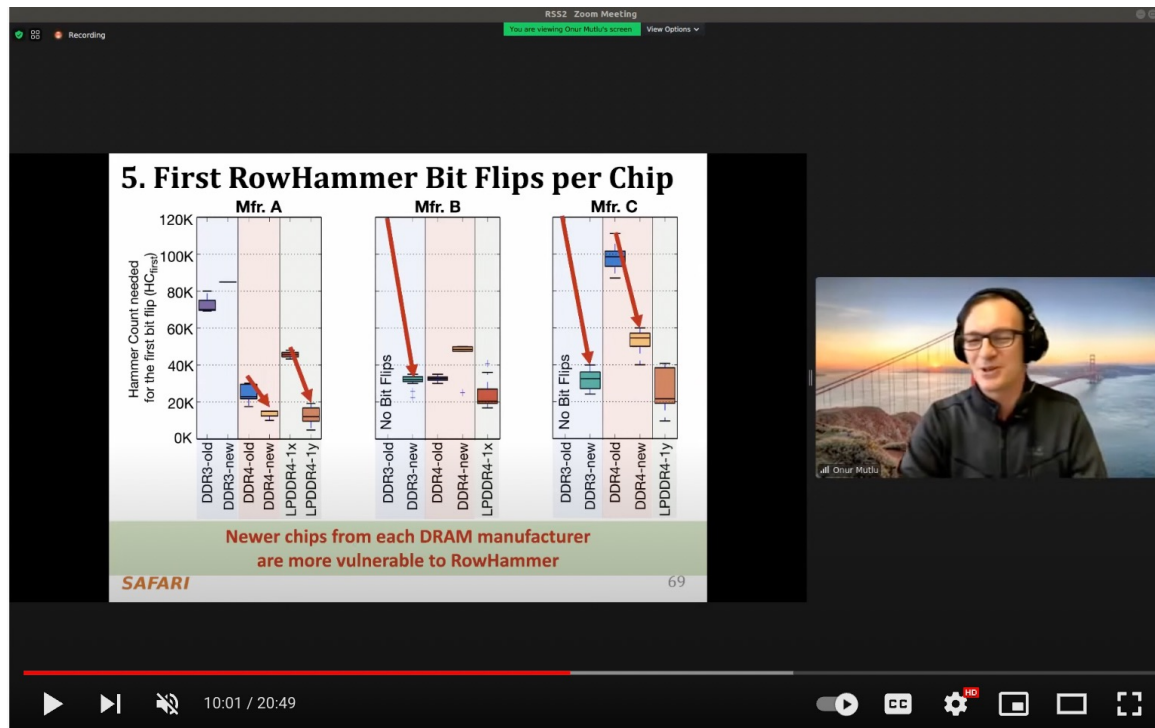
<https://www.youtube.com/watch?v=37hWgIkQRGO>

ANALYTICS

EDIT VIDEO

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 \(pptx\)\]](#) [\[pdf\]](#)



The Story of RowHammer - Invited Talk in Robust & Safe Software Workshop (ASPLOS 2022) - Onur Mutlu

402 views • Premiered Apr 27, 2022

👍 17 🗨 DISLIKE ➦ SHARE ⬇ DOWNLOAD ⚙ CLIP ⚙ SAVE ...



Onur Mutlu Lectures
24.5K subscribers

<https://www.youtube.com/watch?v=ctKTRYi96Bk>

SUBSCRIBED



Latest RowHammer Lecture

Collapse of the “Galloping Gertie”



2:18 / 1:24:50 • Secure, Reliable and Safe > SAFARI <http://www.wsdot.wa.gov/tbhistory/connections/connections3.htm> [Play] [CC] 3 [Settings] [Full Screen] [Exit Full Screen]

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



Onur Mutlu Lectures
35.2K subscribers



Subscribed ▾

👍 14



🔗 Share

⬇️ Download

✂️ Clip



454 views 1 month ago

Title: Securing the Memory System: The Story of RowHammer

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"

Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.

[[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

Industry's Intelligent DRAM Controllers (I)

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES

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

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

SK hynix Semiconductor, Icheon, Korea

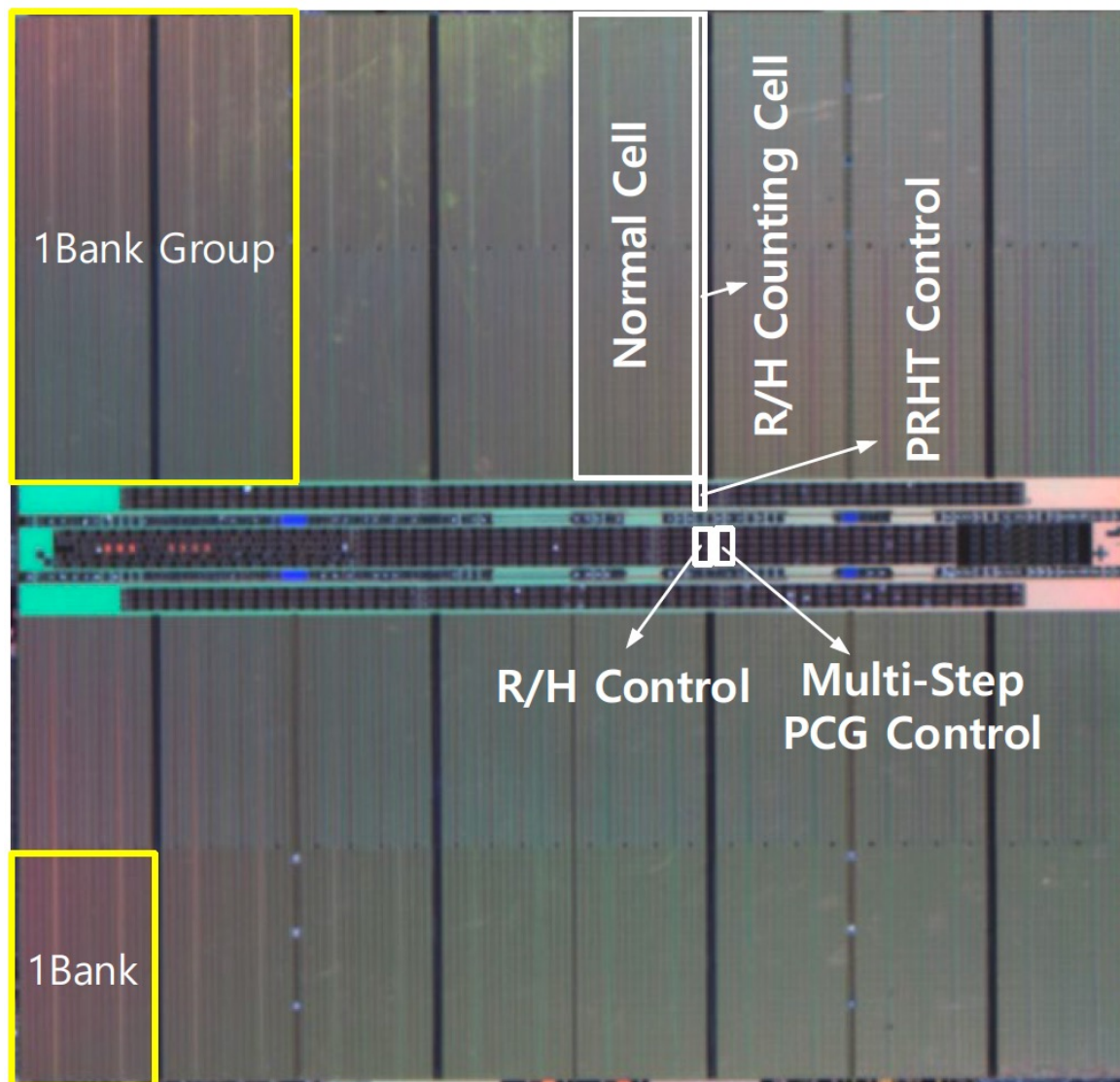


Industry's Intelligent DRAM Controllers (II)

SK hynix Semiconductor, Icheon, Korea

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

Industry's Intelligent DRAM Controllers (III)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES

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

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Dhyong 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

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

<https://arxiv.org/pdf/2302.03591v1.pdf>

Are We Now BitFlip Free?

- **Appears 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



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

Proceedings of the 50th International Symposium on Computer Architecture (ISCA), Orlando, FL, USA, June 2023.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Video](#) (3 minutes)]

[[RowPress Source Code and Datasets \(Officially Artifact Evaluated with All Badges\)](#)]

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

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"** *Proceedings of the 36th International Symposium on Computer Architecture (ISCA)*, pages 2-13, Austin, TX, June 2009. [Slides \(pdf\)](#)
One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

Architecting Phase Change Memory as a Scalable DRAM Alternative

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

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

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

Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

❖ Refresh

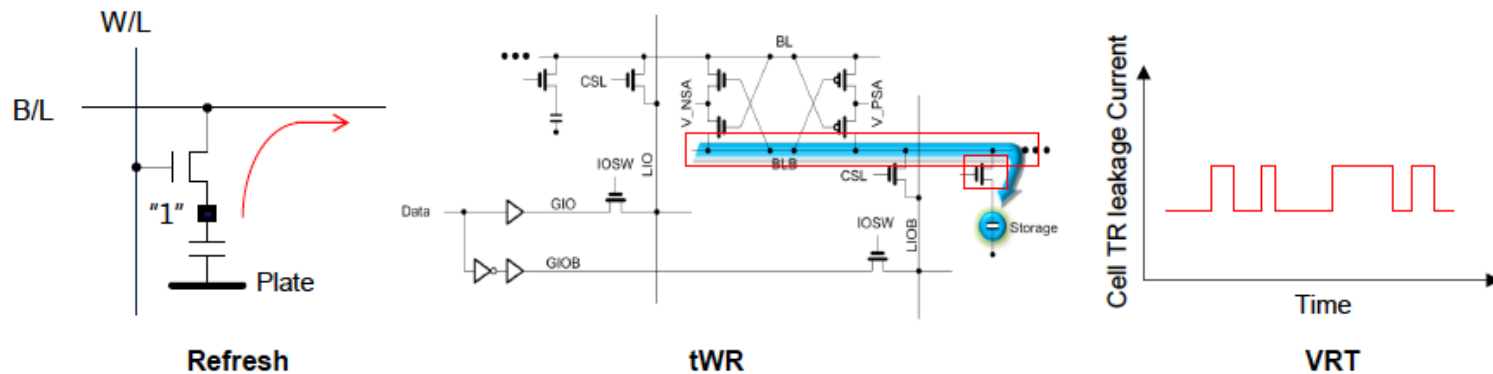
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

❖ 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

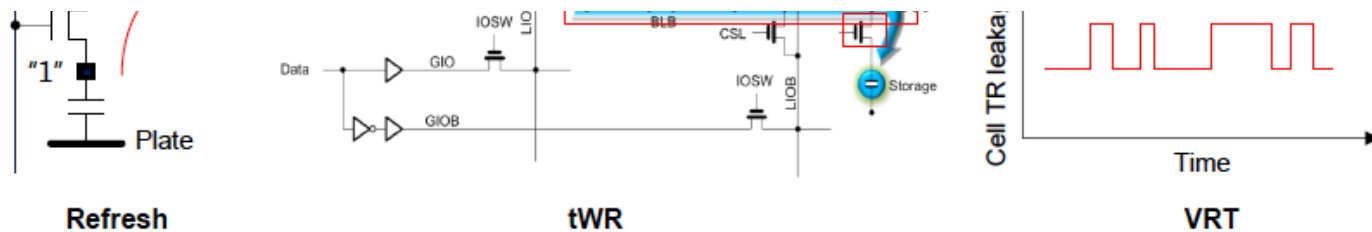
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

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

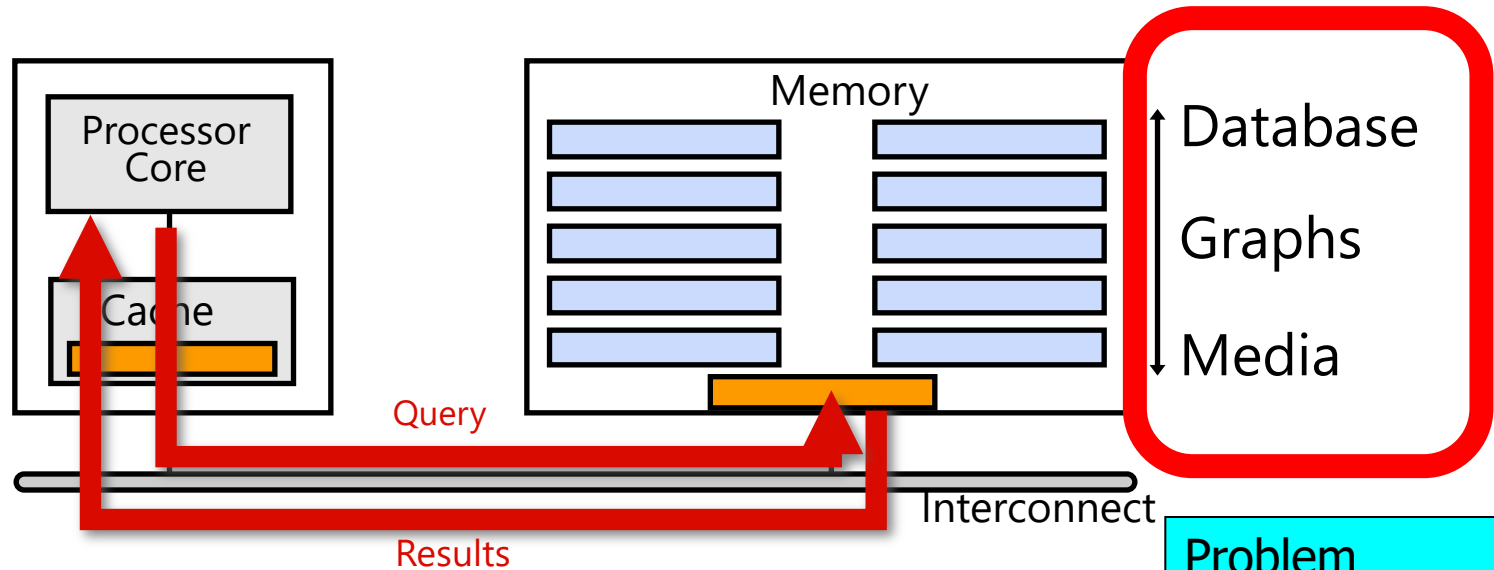
High Performance,

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?

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

PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^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"

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

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.

Approach	Example Enabling Technologies
Processing Using Memory	SRAM DRAM Phase-change memory (PCM) Magnetic RAM (MRAM) Resistive RAM (RRAM)/memristors
Processing Near Memory	Logic layers in 3D-stacked memory Silicon interposers Logic in memory controllers Logic in memory chips (e.g., near bank) Logic in memory modules Logic near caches Logic near/in storage devices

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun, ["A Modern Primer on Processing in Memory"](#)

*Invited Book Chapter in **Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann***, Springer, to be published in 2021. [[Tutorial Video on "Memory-Centric Computing Systems"](#) (1 hour 51 minutes)]

Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory

Accelerating In-Memory Graph Processing

- Large graphs are everywhere (circa 2015)



36 Million
Wikipedia Pages



1.4 Billion
Facebook Users

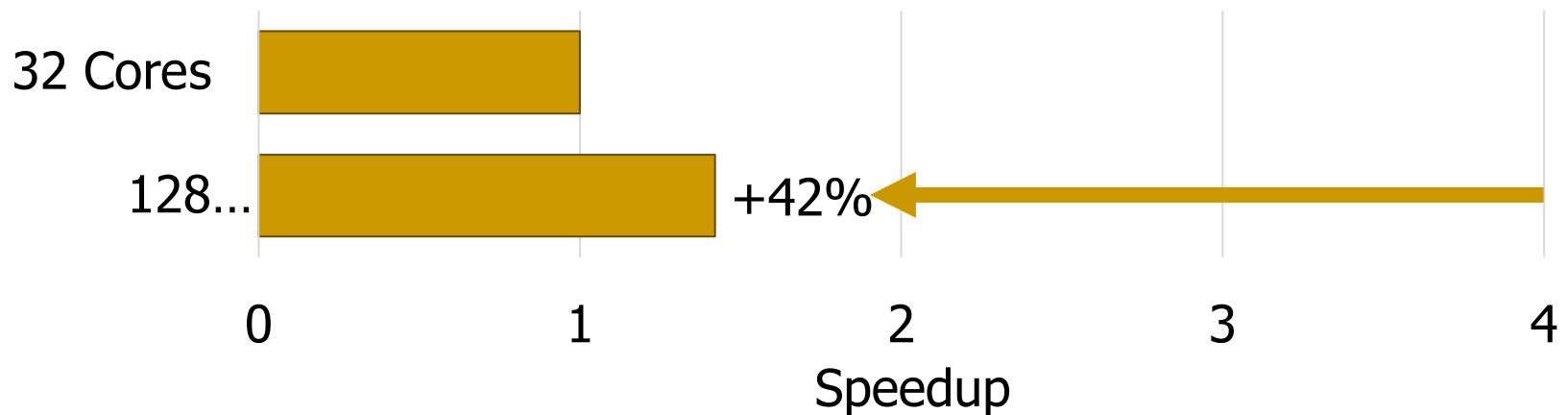


300 Million
Twitter Users



30 Billion
Instagram Photos

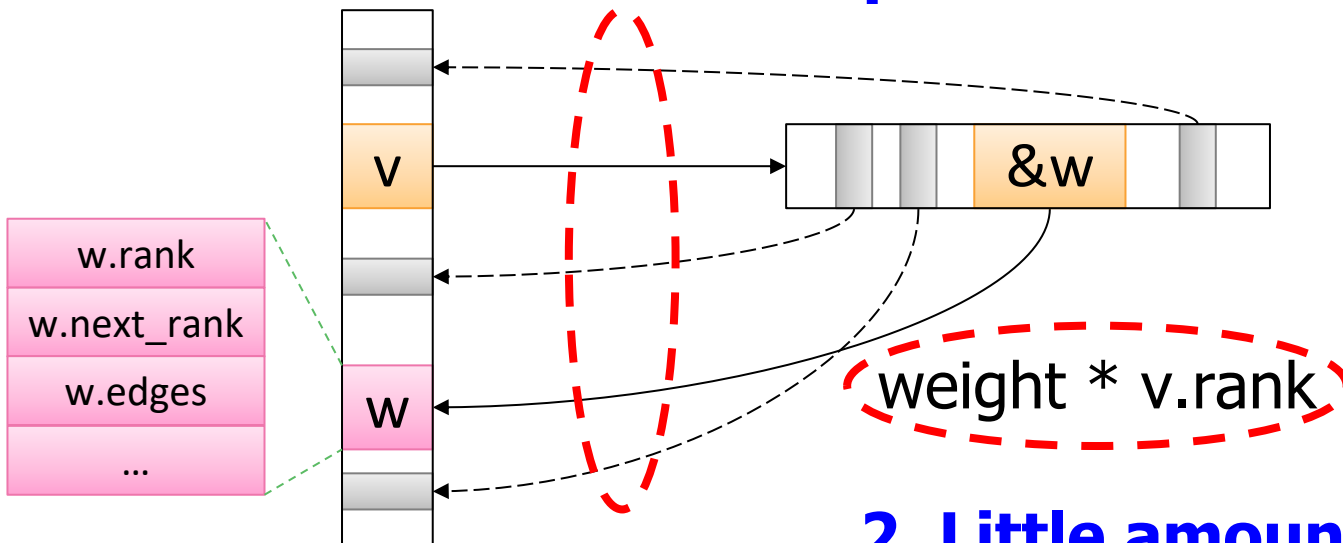
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

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

1. Frequent random memory accesses

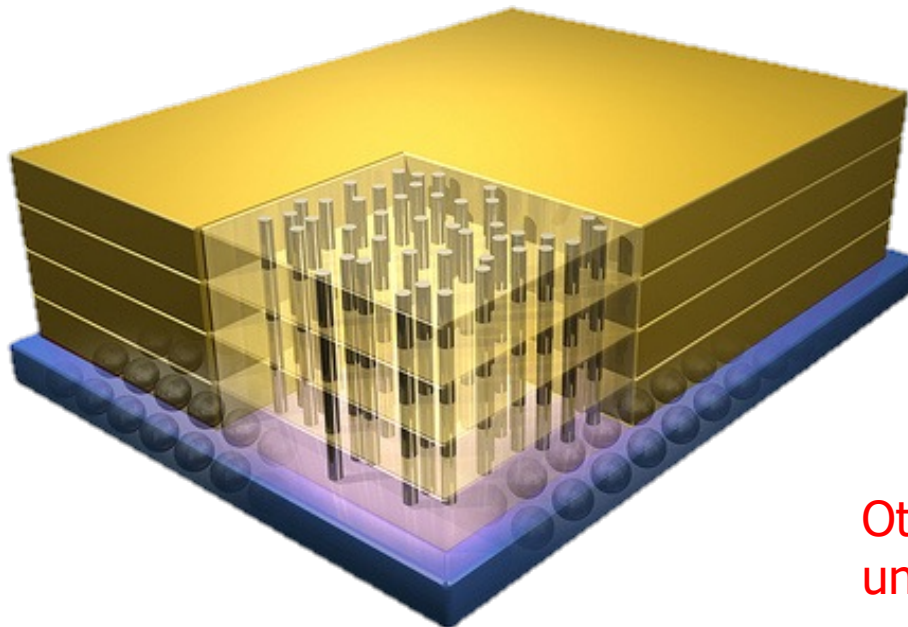


2. Little amount of computation

Opportunity: 3D-Stacked Logic+Memory



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



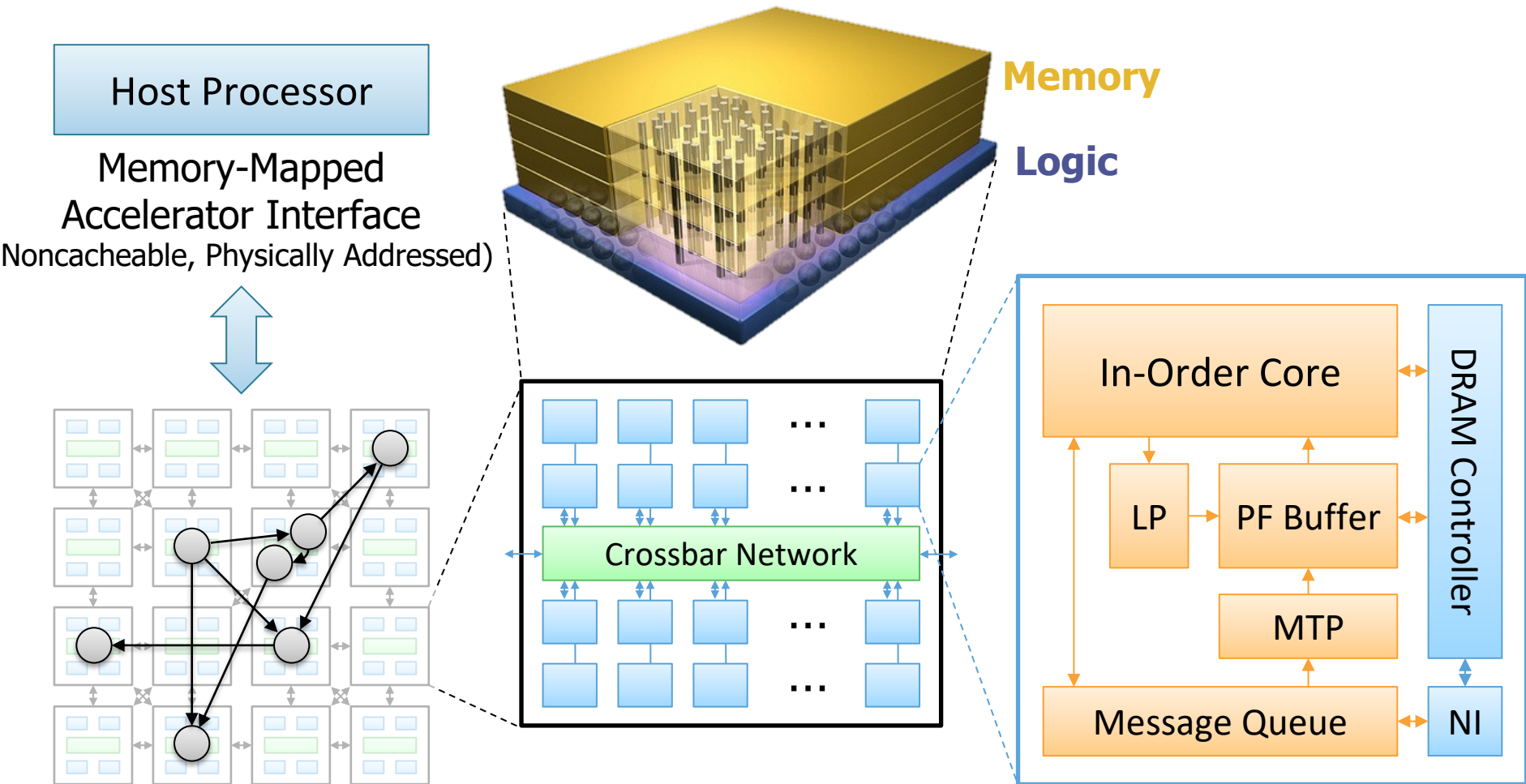
Memory

Logic

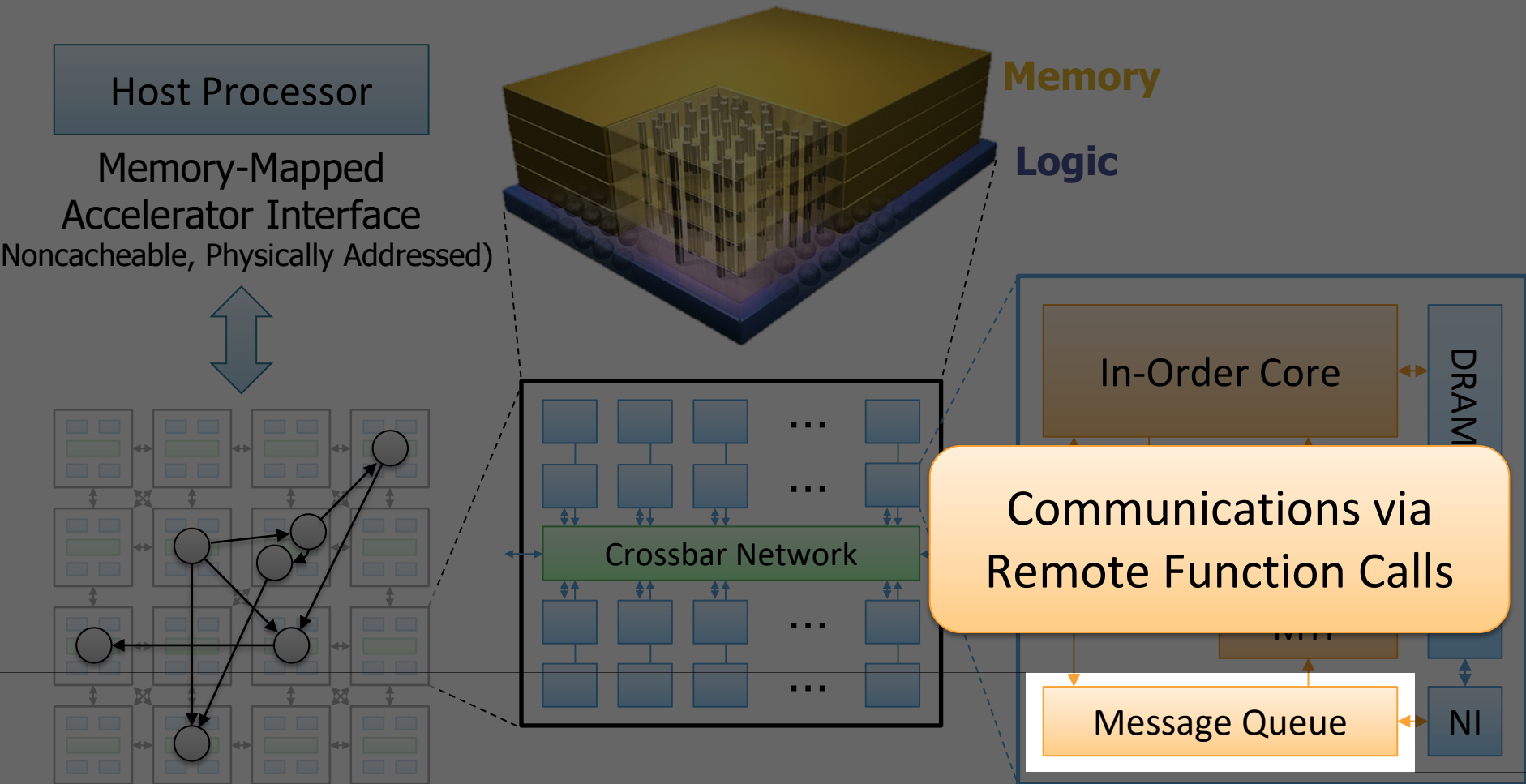
Other "True 3D" technologies
under development

Tesseract System for Graph Processing

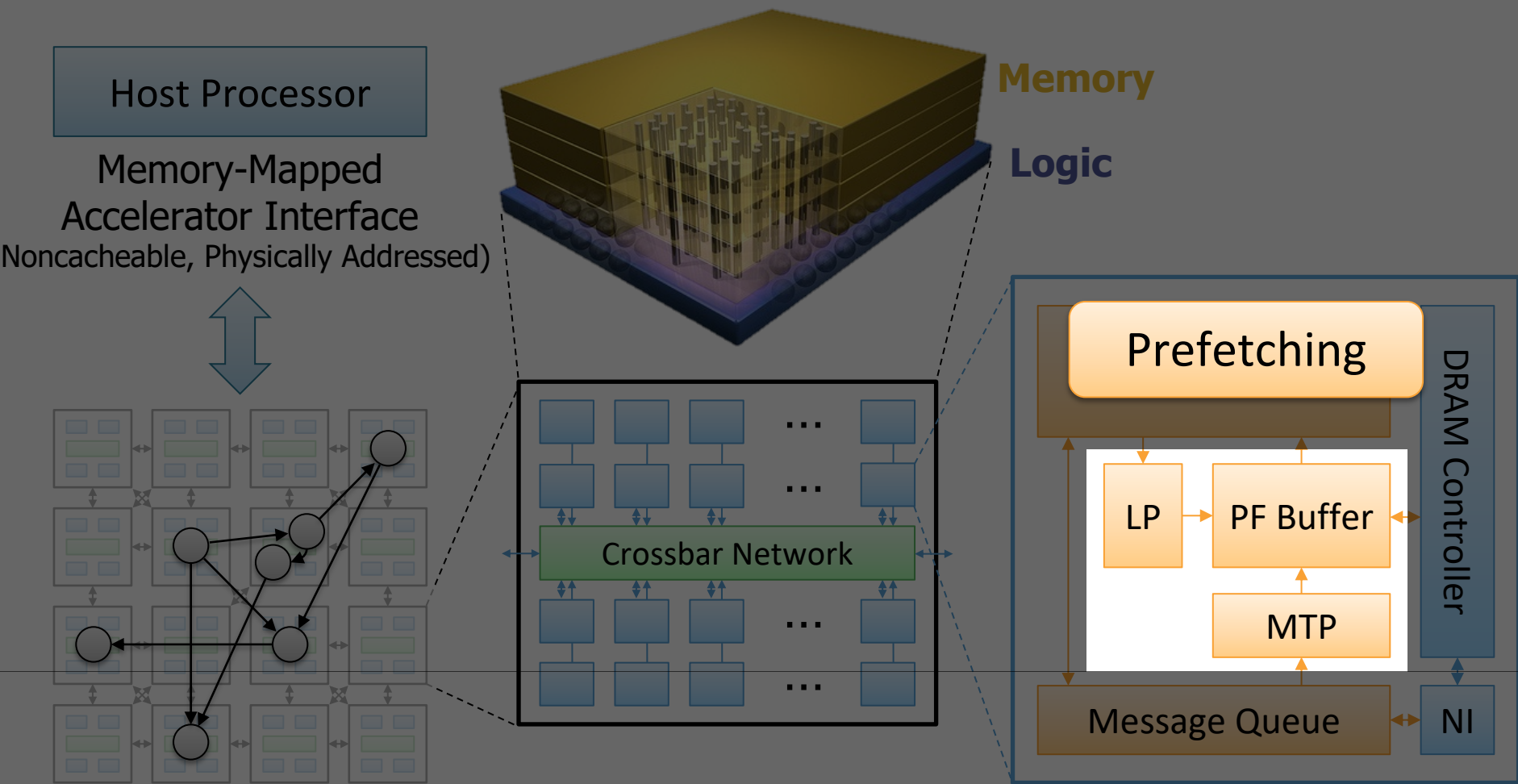
Interconnected set of 3D-stacked memory+logic chips with simple cores



Tesseract System for Graph Processing

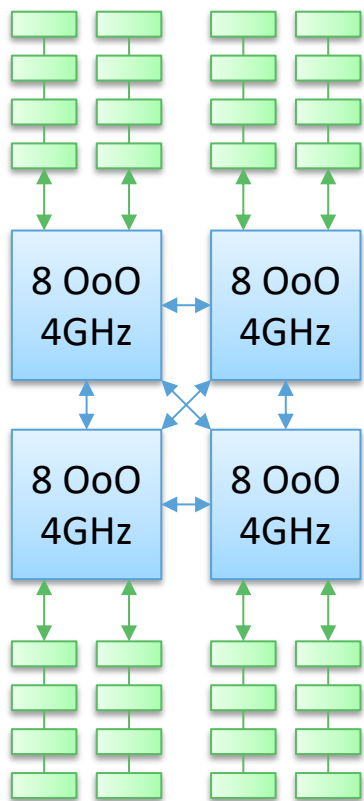


Tesseract System for Graph Processing



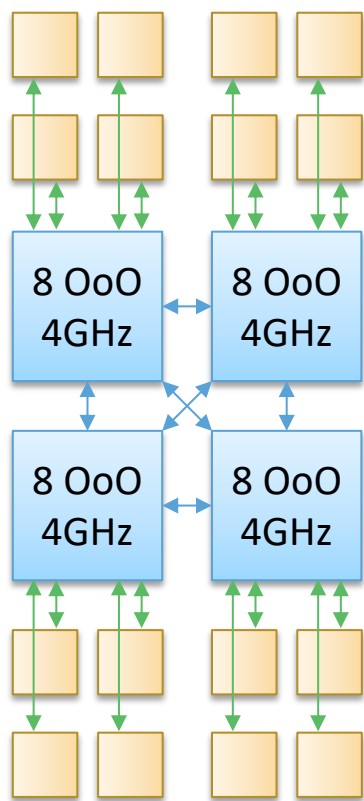
Evaluated Systems

DDR3-OoO



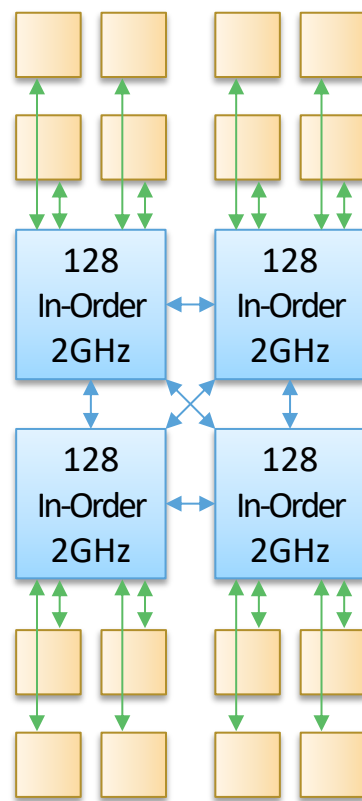
102.4GB/s

HMC-OoO



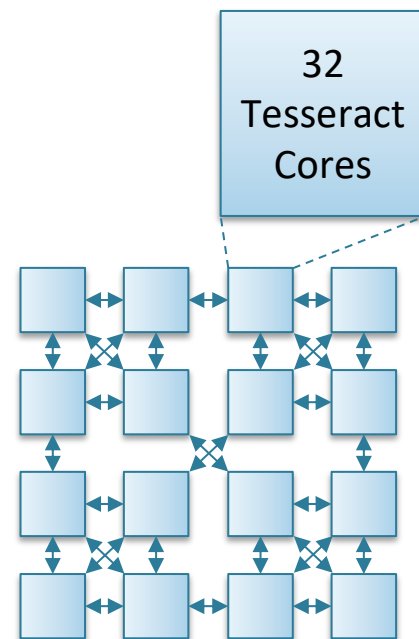
640GB/s

HMC-MC



640GB/s

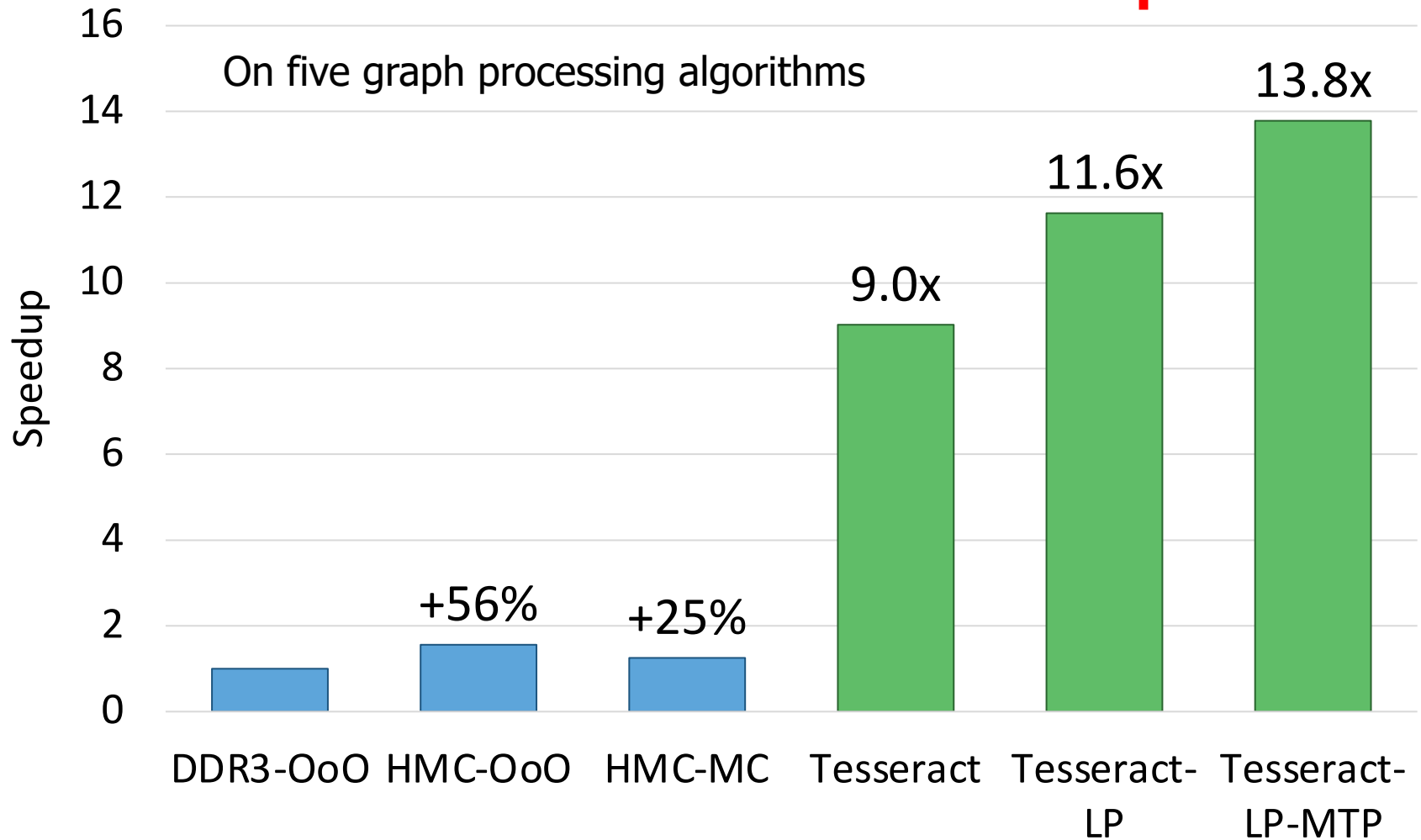
Tesseract



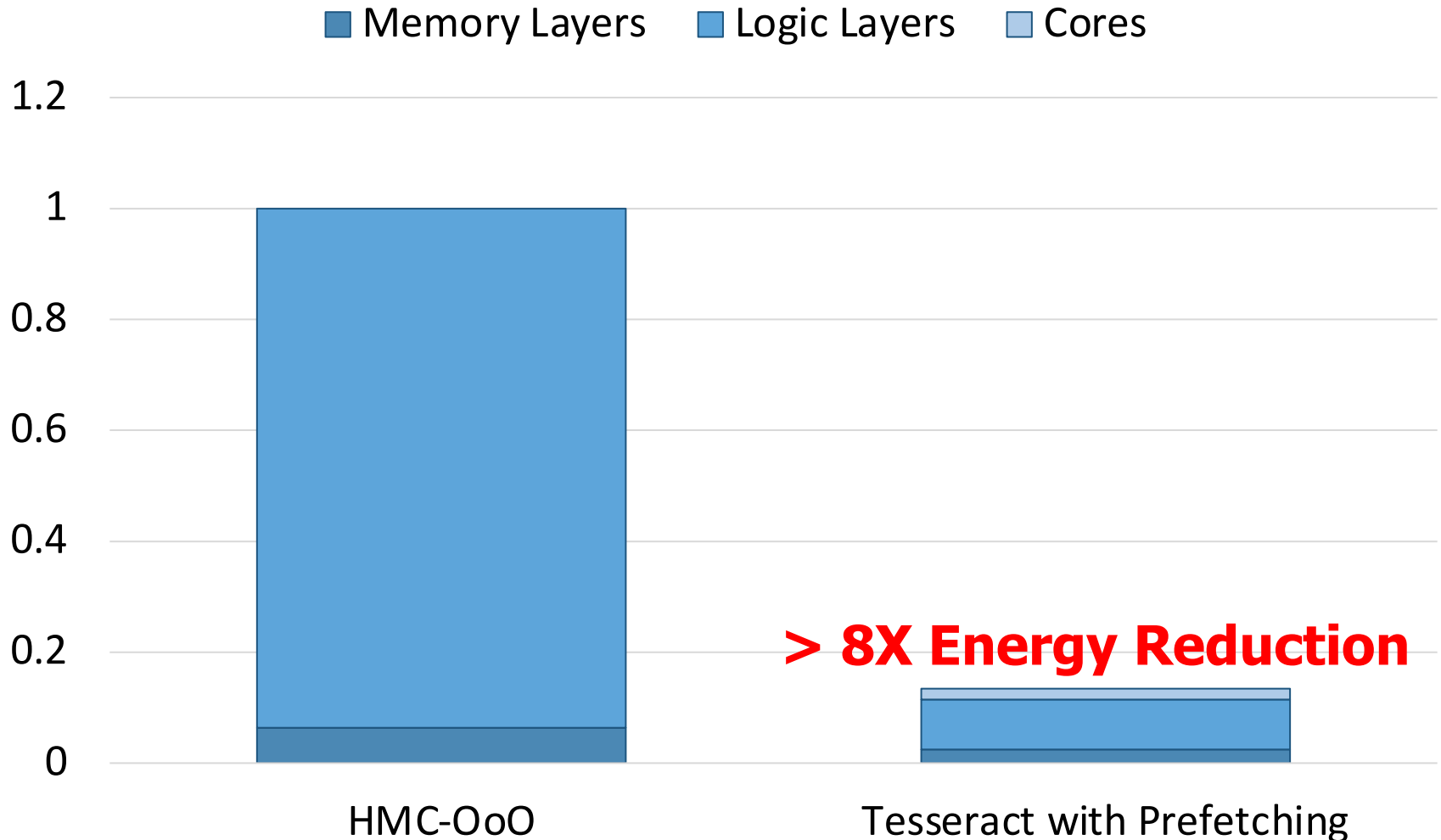
8TB/s

Tesseract Graph Processing Performance

>13X Performance Improvement



Tesseract Graph Processing System Energy



More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,

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

Proceedings of the 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.

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

***Top Picks Honorable Mention by IEEE Micro.
Selected to the ISCA-50 25-Year Retrospective Issue
covering 1996-2020 in 2023
(Retrospective (pdf) Full Issue).***

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

Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi

junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[§]Oracle Labs

[†]Carnegie Mellon University

In-Storage Genomic Data Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu, **["GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"](#)**
Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
[[Lightning Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Lightning Talk Video](#) (90 seconds)] [[Talk Video](#) (17 minutes)]

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

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹
Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

Genome Sequence Analysis

Data Movement from Storage



Storage
System

Main
Memory

Cache

Alignment
Computation
Unit
(CPU or
Accelerator)

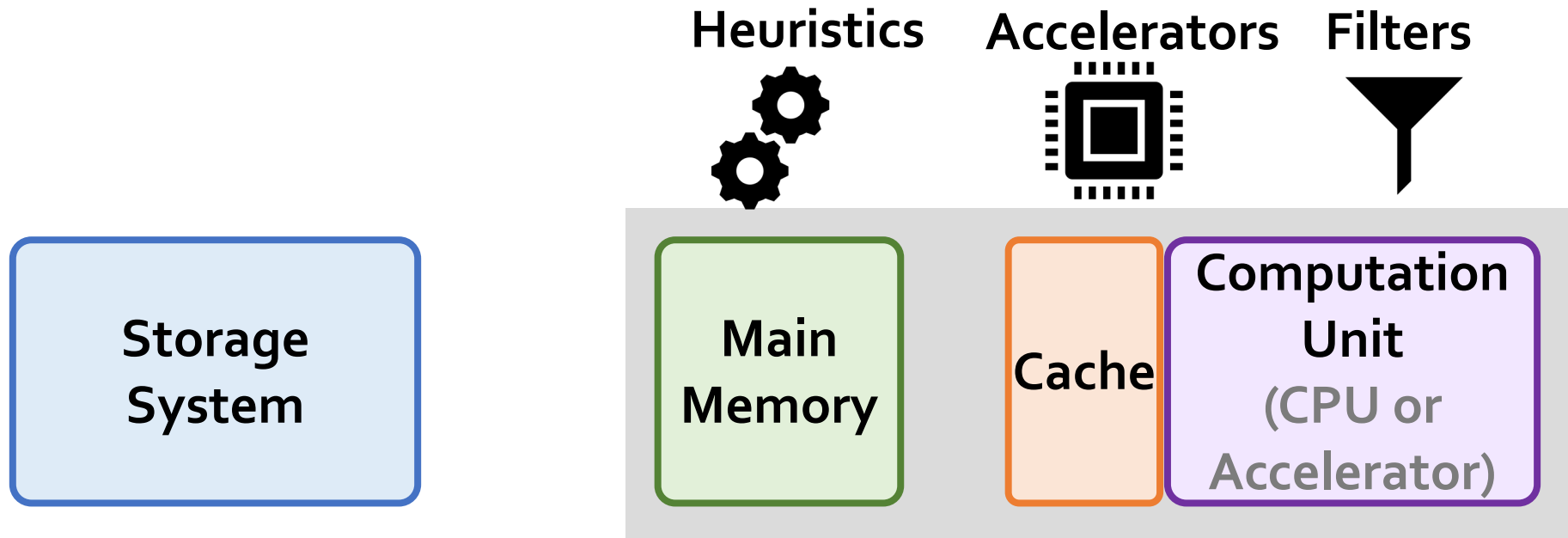


Computation overhead



Data movement overhead

Compute-Centric Accelerators



Computation overhead

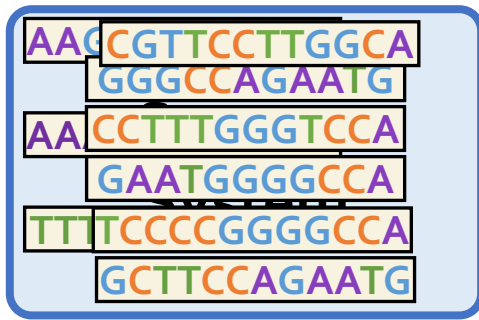


Data movement overhead

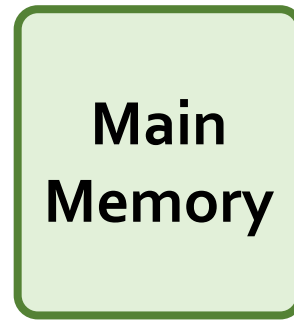
Key Idea: In-Storage Filtering



Filter reads that do not require alignment inside the storage system



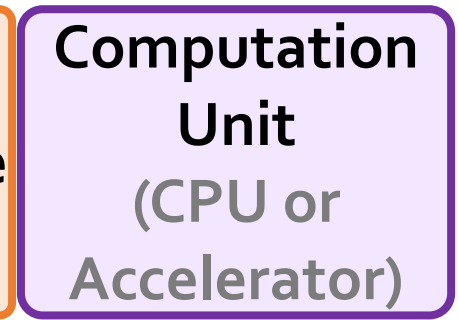
Filtered Reads



**Main
Memory**



Cache



**Computation
Unit
(CPU or
Accelerator)**

Exactly-matching reads

Do not need expensive approximate string matching during alignment

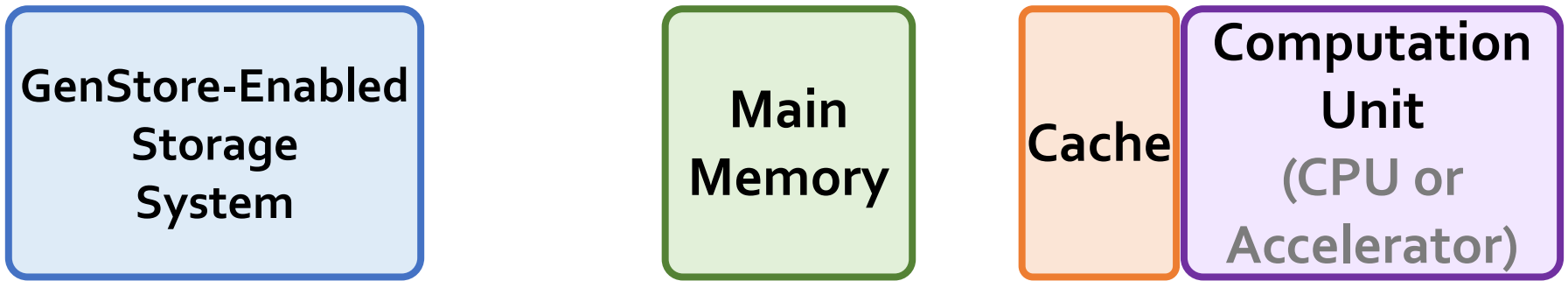
Non-matching reads

Do not have potential matching locations and can skip alignment

GenStore



Filter reads that do not require alignment inside the storage system



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"**](#)
Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Virtual, February-March 2022.
[[Lightning Talk Slides \(pptx\) \(pdf\)](#)]
[[Lightning Talk Video](#) (90 seconds)]

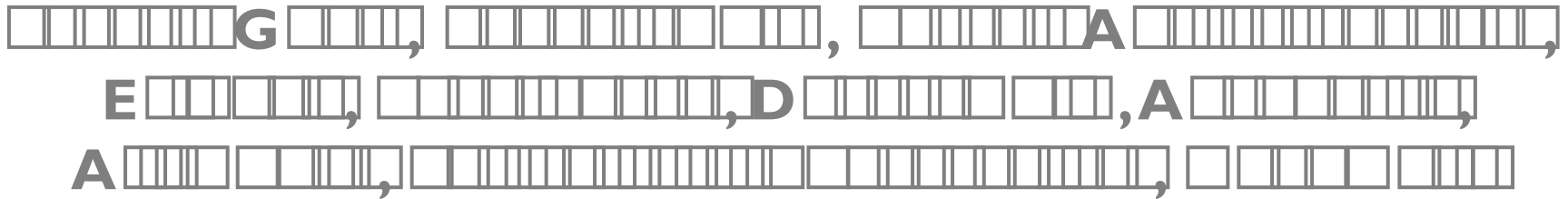
GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹
Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich ²Bionano Genomics ³KMUTNB ⁴University of Toronto

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

A  **B** 



SAFARI

Carnegie Mellon

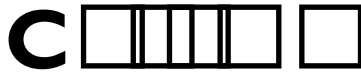
Google



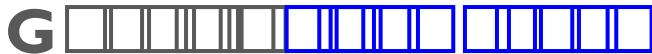
SEOUL
NATIONAL
UNIVERSITY

ETH zürich



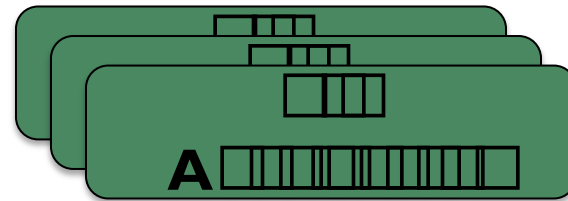
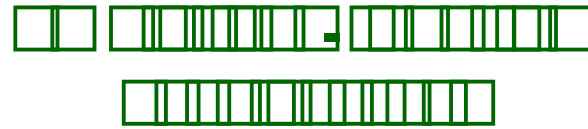
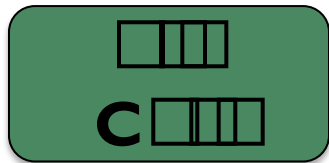
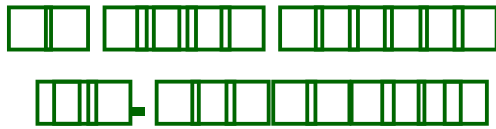
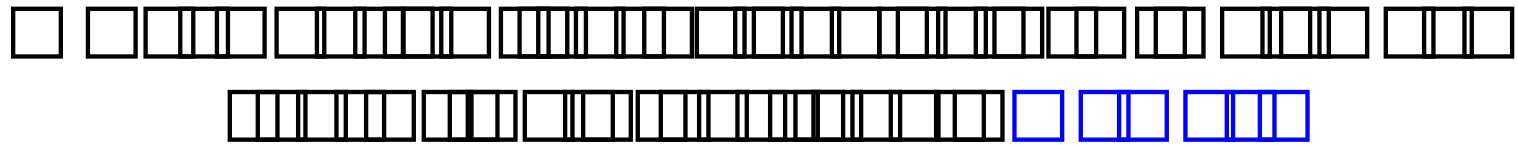
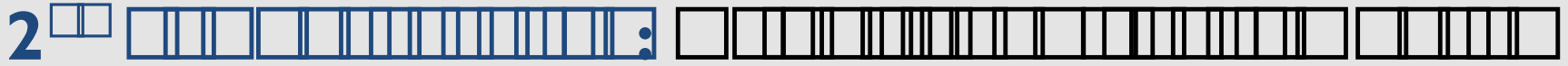
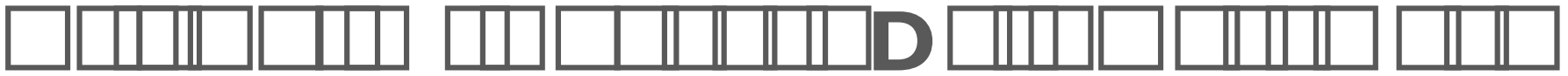


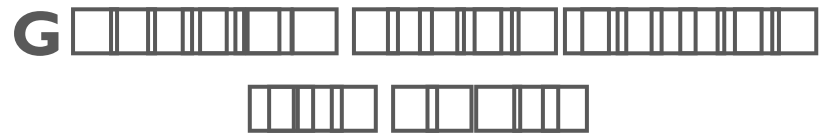
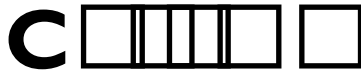
VP9



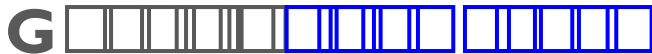
VP9



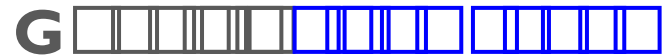


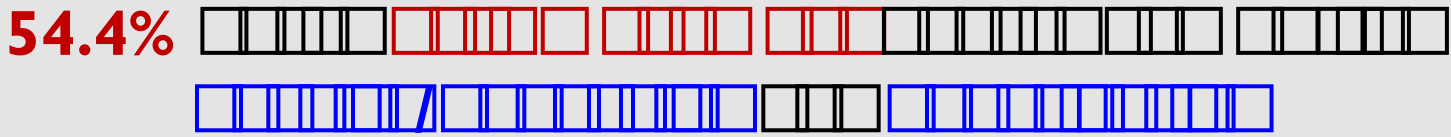
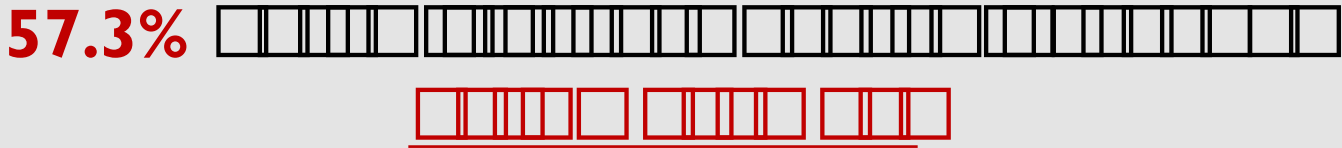
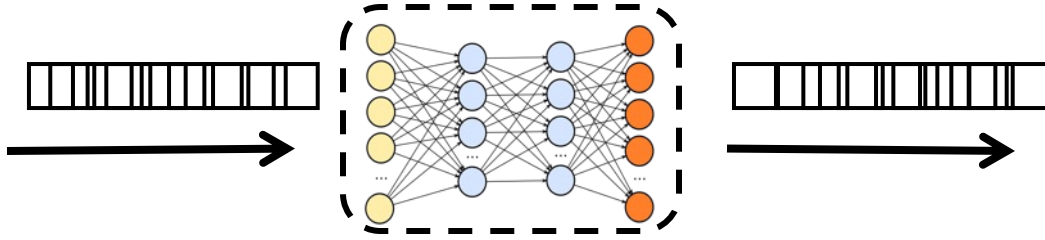


VP9



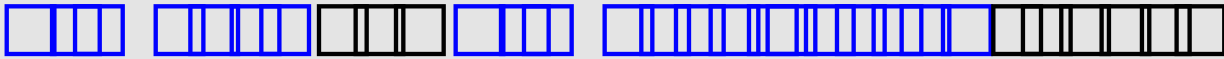
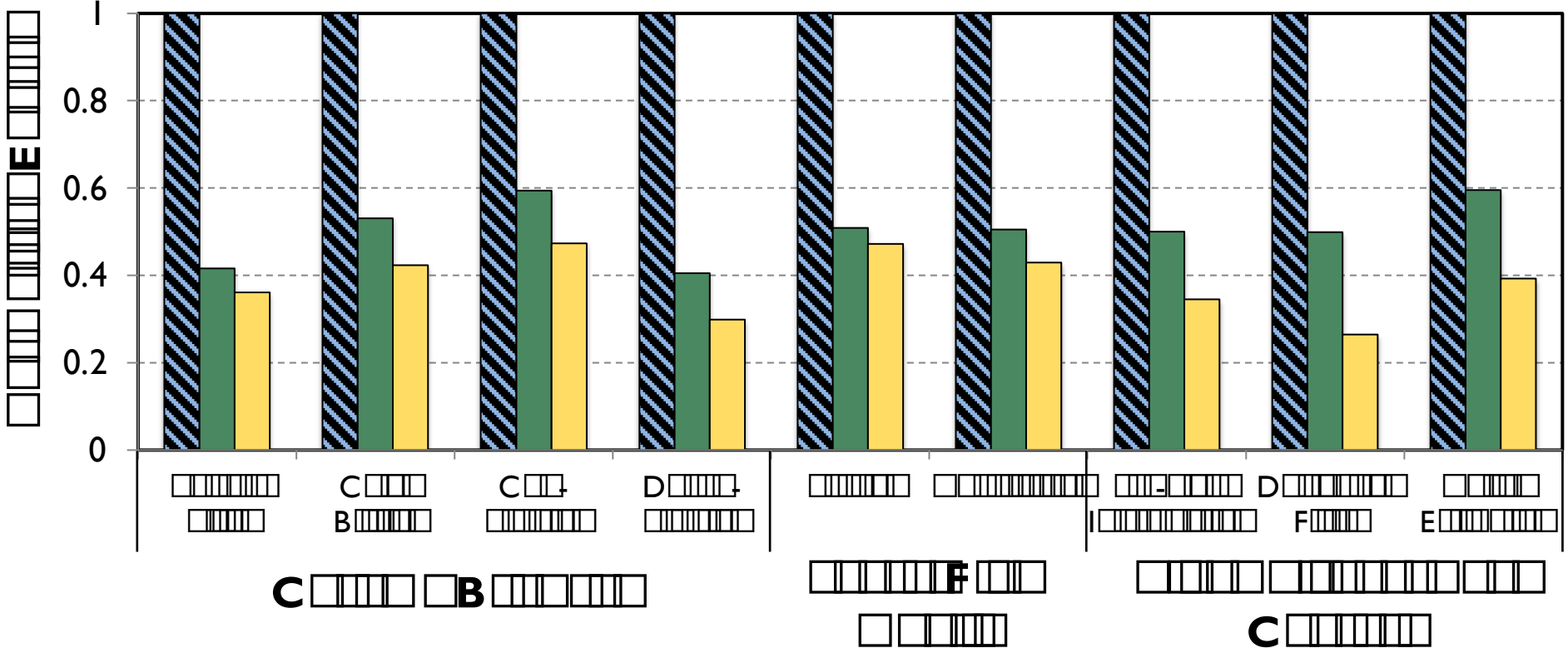
VP9







C



Empty box Empty box Empty box
Empty box Empty box Empty box Empty box
Empty box Empty box Empty box
49.1% Empty box Empty box 55.4%

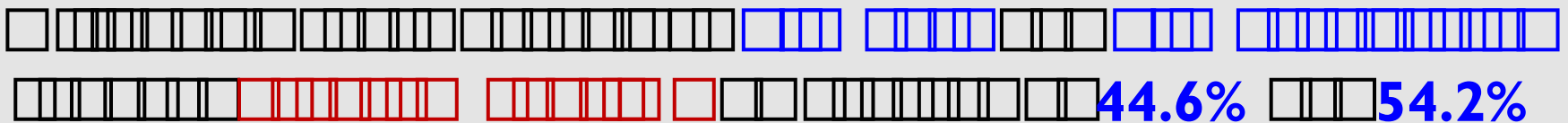
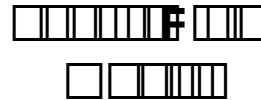
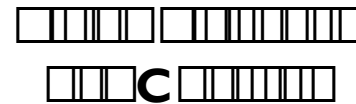
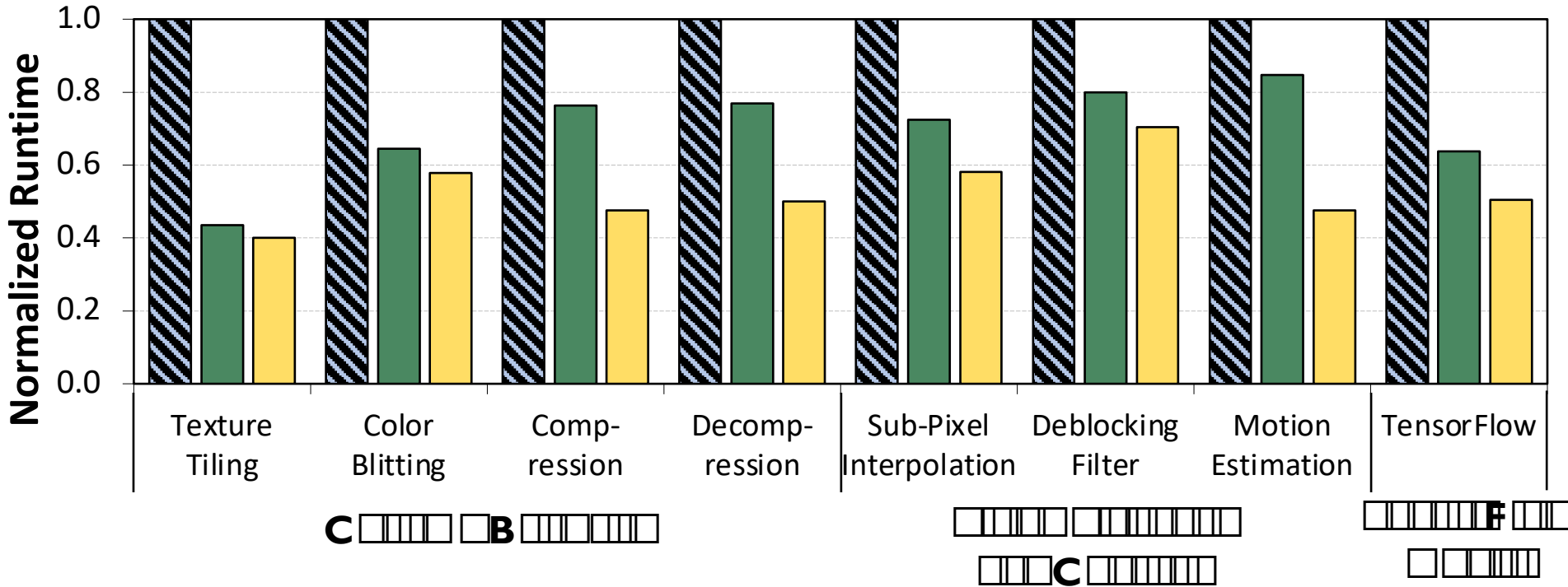
SAFARI



CPU-Only

PIM-Core

PIM-Acc



44.6%
54.2%

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"

Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Williamsburg, VA, USA, March 2018.

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

[[Lightning Talk Video](#) (2 minutes)]

[[Full Talk Video](#) (21 minutes)]

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

Onur Mutlu^{5,1}

Accelerating GPU Execution with PIM (I)

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

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

Transparent Offloading and Mapping (TOM):

Enabling Programmer-Transparent Near-Data Processing in GPU Systems

Kevin Hsieh[‡] Eiman Ebrahimi[†] Gwangsun Kim* Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu^{§‡} Stephen W. Keckler[†]

[‡]Carnegie Mellon University [†]NVIDIA *KAIST [§]ETH Zürich

Accelerating GPU Execution with PIM (II)

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

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

Ashutosh Pattnaik¹ Xulong Tang¹ Adwait Jog² Onur Kayiran³
Asit K. Mishra⁴ Mahmut T. Kandemir¹ Onur Mutlu^{5,6} Chita R. Das¹

¹Pennsylvania State University ²College of William and Mary
³Advanced Micro Devices, Inc. ⁴Intel Labs ⁵ETH Zürich ⁶Carnegie Mellon University

Accelerating Linked Data Structures

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,
"Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"
Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

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

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

Accelerating Dependent Cache Misses

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

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

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

Accelerating Dependent Cache Misses with an Enhanced Memory Controller

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

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

Accelerating Runahead Execution

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

Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads

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

**The University of Texas at Austin* [§]*ETH Zürich*

Accelerating Climate Modeling

- Gagandeep Singh, Dionysios Diamantopoulos, Christoph Hagleitner, Juan Gómez-Luna, Sander Stuijk, Onur Mutlu, and Henk Corporaal,

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

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

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (23 minutes)]

Nominated for the Stamatis Vassiliadis Memorial Award.

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

Gagandeep Singh^{a,b,c}

Dionysios Diamantopoulos^c

Christoph Hagleitner^c

Juan Gómez-Luna^b

Sander Stuijk^a

Onur Mutlu^b

Henk Corporaal^a

^aEindhoven University of Technology

^bETH Zürich

^cIBM Research Europe, Zurich

Accelerating Approximate String Matching

- Damla Senol Cali, Gurpreet S. Kalsi, Zülal Bingöl, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, "[GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis](#)"
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.
[[Lighting Talk Video](#) (1.5 minutes)]
[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]
[[Talk Video](#) (18 minutes)]
[[Slides \(pptx\)](#) ([pdf](#))]

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

Damla Senol Cali[†][✕] Gurpreet S. Kalsi[✕] Zülal Bingöl[∇] Can Firtina[◇] Lavanya Subramanian[‡] Jeremie S. Kim[◇][†]
Rachata Ausavarungnirun[○] Mohammed Alser[◇] Juan Gomez-Luna[◇] Amirali Boroumand[†] Anant Nori[✕]
Allison Scibisz[†] Sreenivas Subramoney[✕] Can Alkan[∇] Saugata Ghose^{*†} Onur Mutlu[◇][†][∇]
[†]Carnegie Mellon University [✕]Processor Architecture Research Lab, Intel Labs [∇]Bilkent University [◇]ETH Zürich
[‡]Facebook [○]King Mongkut's University of Technology North Bangkok ^{*}University of Illinois at Urbana-Champaign

Accelerating Sequence-to-Graph Mapping

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

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

Damla Senol 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 Mao¹ Mohammed Alser¹ Mohammad Sadrosadati¹ Can Firtina¹ Akanksha Baranwal¹
Damla Senol Cali² Aditya Manglik¹ Nour Almadhoun Alserr¹ Onur Mutlu¹
¹*ETH Zürich* ²*Bionano Genomics*

Accelerating Time Series Analysis

- Ivan Fernandez, Ricardo Quisiant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu, **"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"**
Proceedings of the 38th IEEE International Conference on Computer Design (ICCD), Virtual, October 2020.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (10 minutes)]
[[Source Code](#)]

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

Ivan Fernandez[§]

Ricardo Quisiant[§]

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 Ausavarungnirun, Jakub Beránek, Konstantinos Kanellopoulos, Kacper Janda, Zur Vonarburg-Shmaria, Lukas Gianinazzi, Ioana Stefan, Juan Gómez-Luna, Marcin Copik, Lukas Kapp-Schwoerer, Salvatore Di Girolamo, Nils Blach, Marek Konieczny, Onur Mutlu, and Torsten Hoefler,

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

Proceedings of the [54th International Symposium on Microarchitecture \(MICRO\)](#), Virtual, October 2021.

[[Slides \(pdf\)](#)]

[[Talk Video](#) (22 minutes)]

[[Lightning Talk Video](#) (1.5 minutes)]

[[Full arXiv version](#)]

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

Maciej Besta¹, Raghavendra Kanakagiri², Grzegorz Kwasniewski¹, Rachata Ausavarungnirun³, 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
Thailand

²IIT Tirupati, India

³King Mongkut's University of Technology North Bangkok,

⁴Technical University of Ostrava, Czech Republic

⁵AGH-UST, Poland

Accelerating HTAP Database Systems

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

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

Amirali Boroumand[†]
[†]*Google*

Saugata Ghose[◇]
[◇]*Univ. of Illinois Urbana-Champaign*

Geraldo F. Oliveira[‡]
[‡]*ETH Zürich*

Onur Mutlu[‡]

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*









G               **E**     **D**   **:**

A           

           **B**      

A           **B**   

      **G**   **F.**     

E        

 **AC**  **2021**

SAFARI

E

C: (G E) 24 G

- (C, , , C)

: E :

-
-
-

: :

E

-

: A

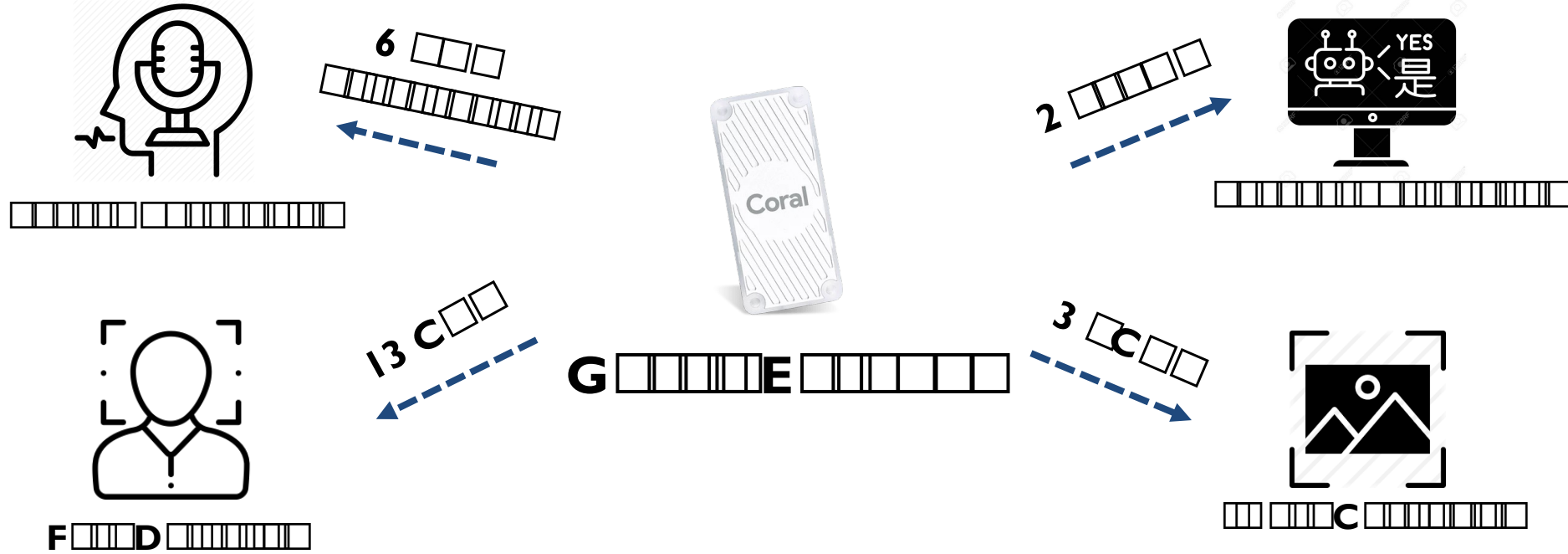
-

: G

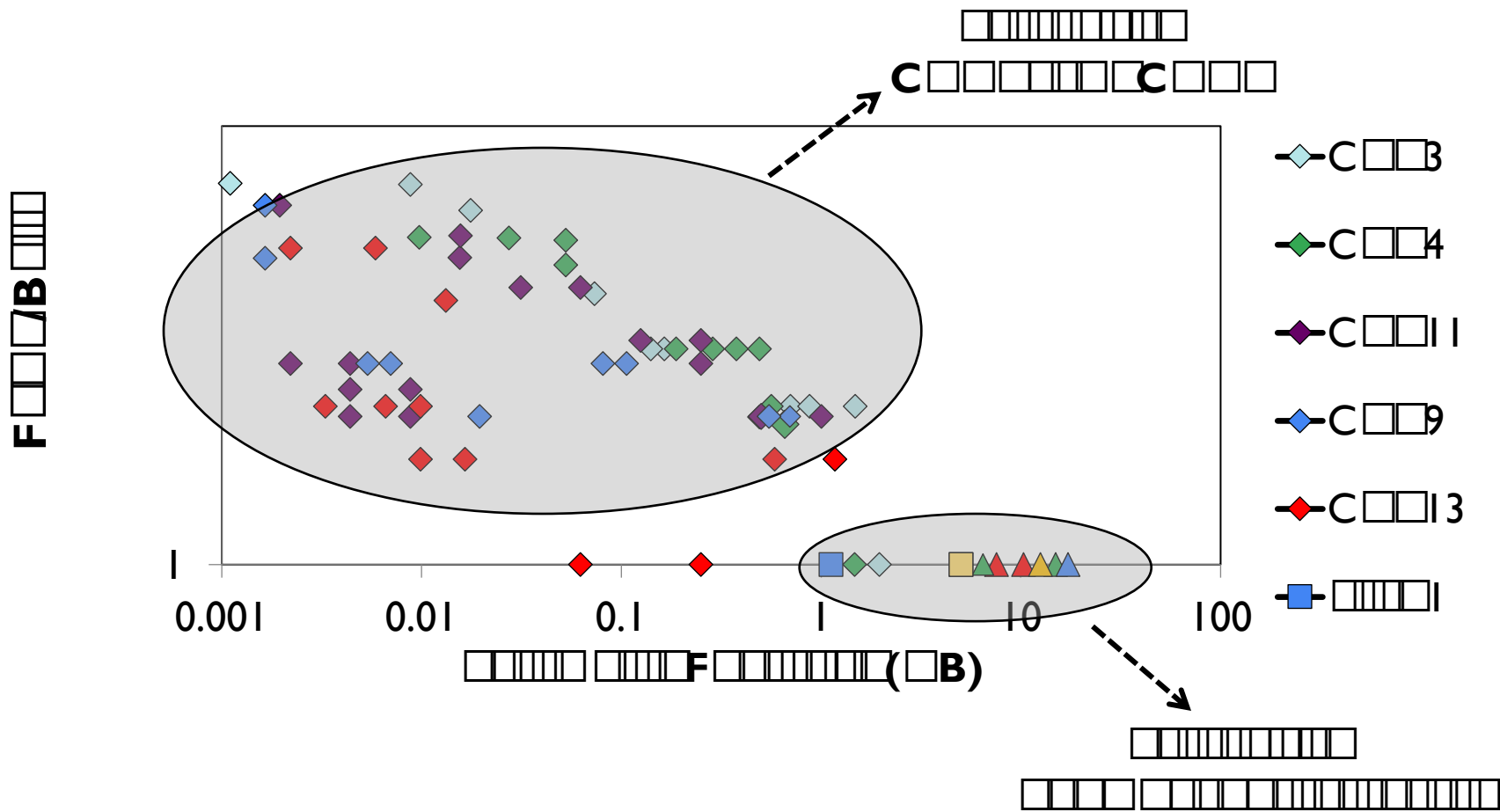
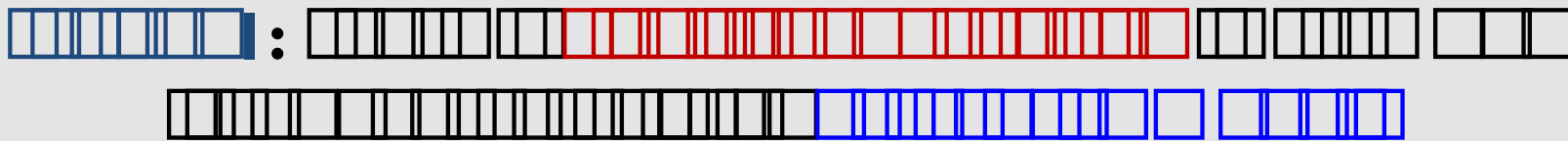
- 3.0 3.1
-

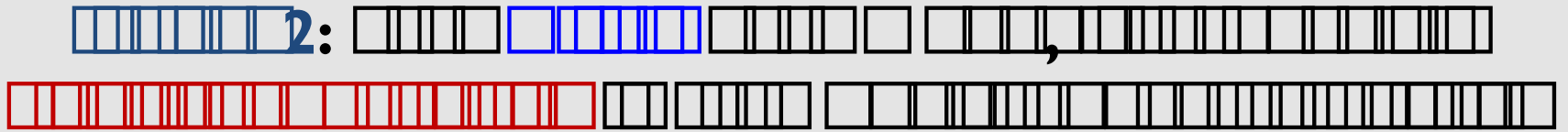
G  E    

  24   

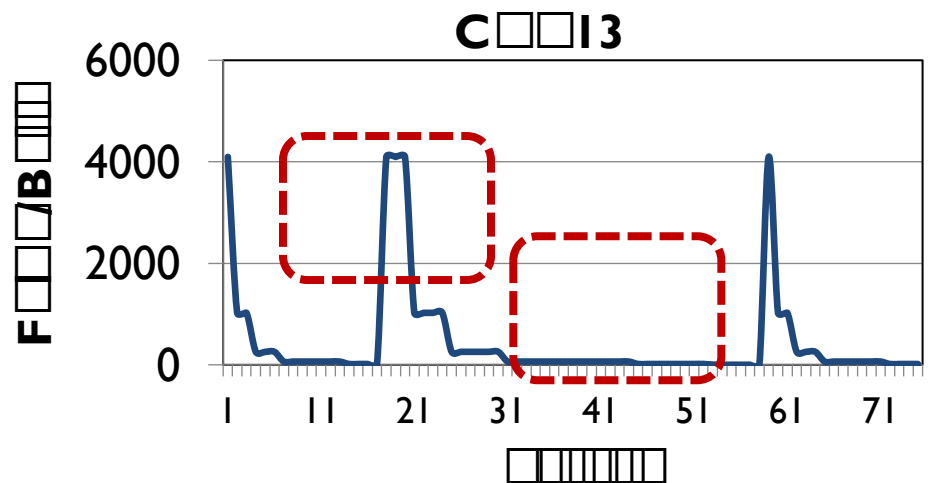
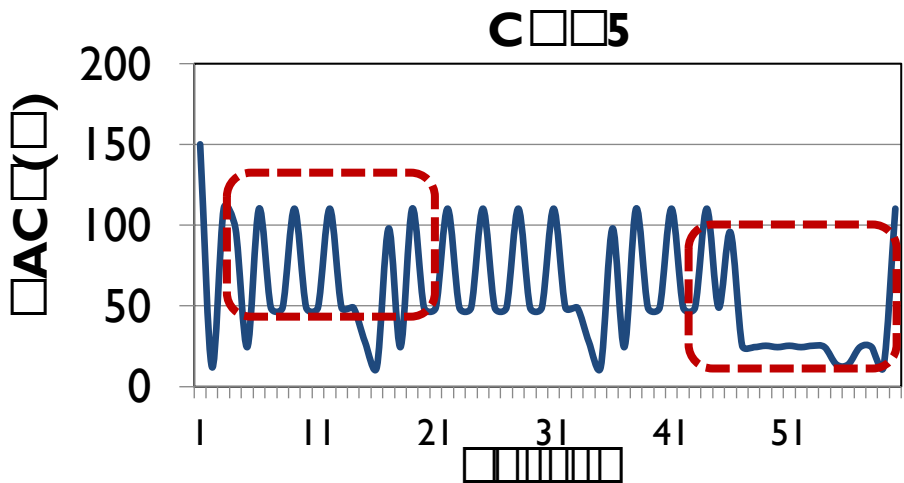


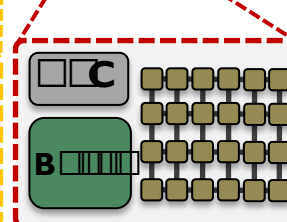
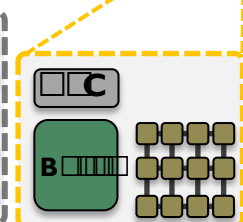
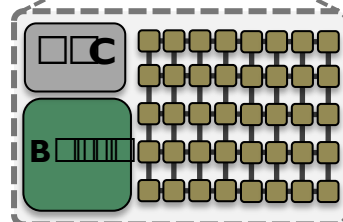
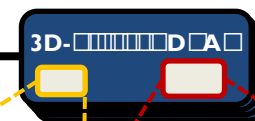
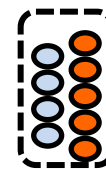
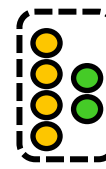
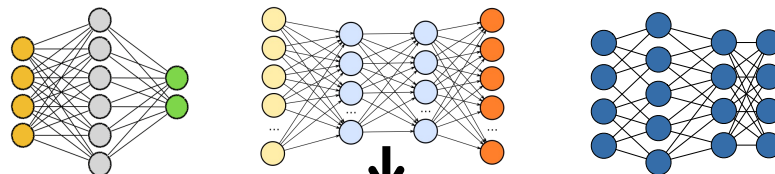
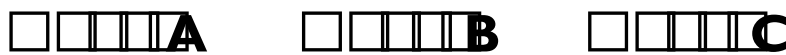
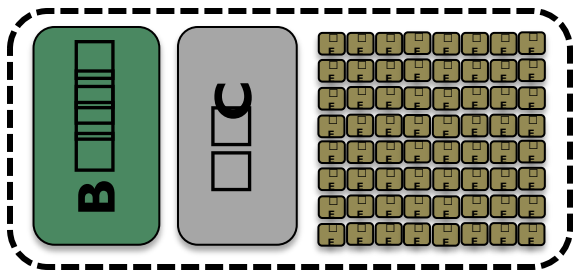
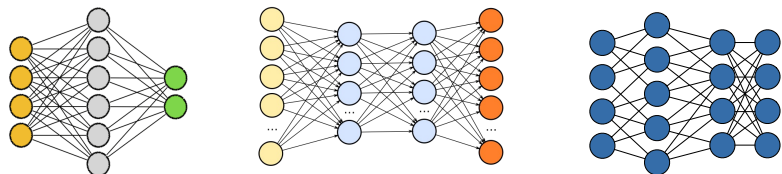
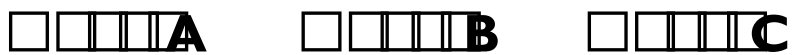
D  A   

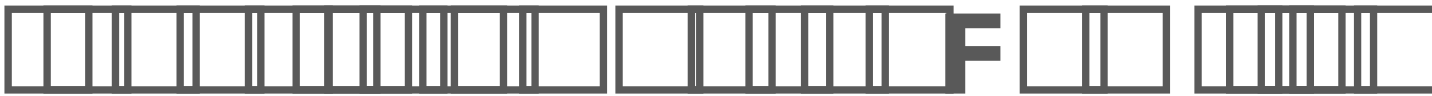




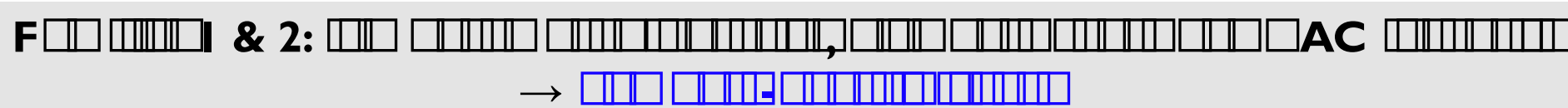
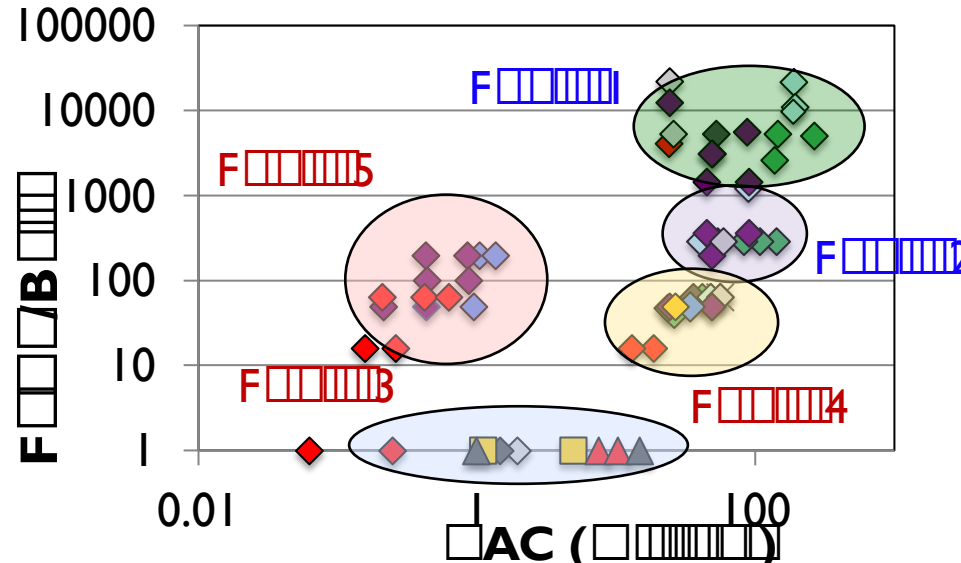
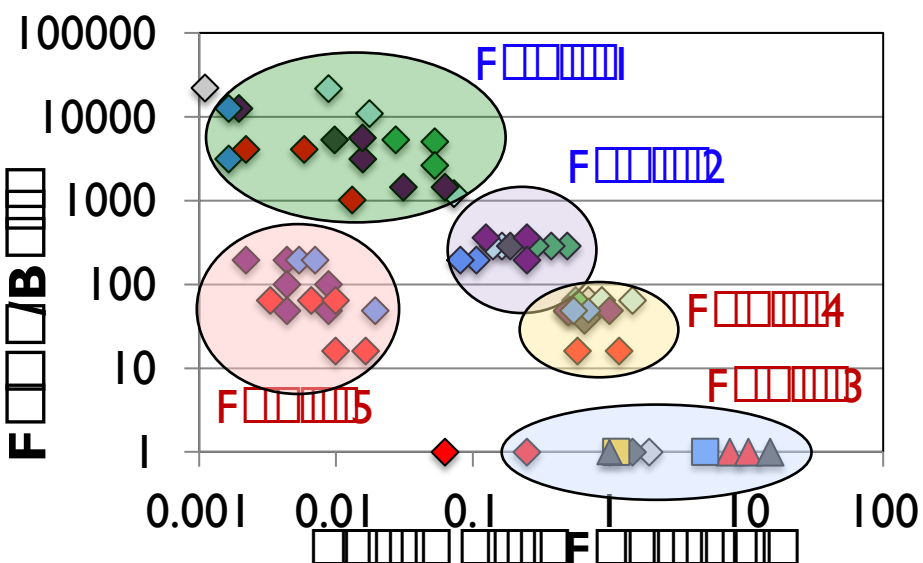
F





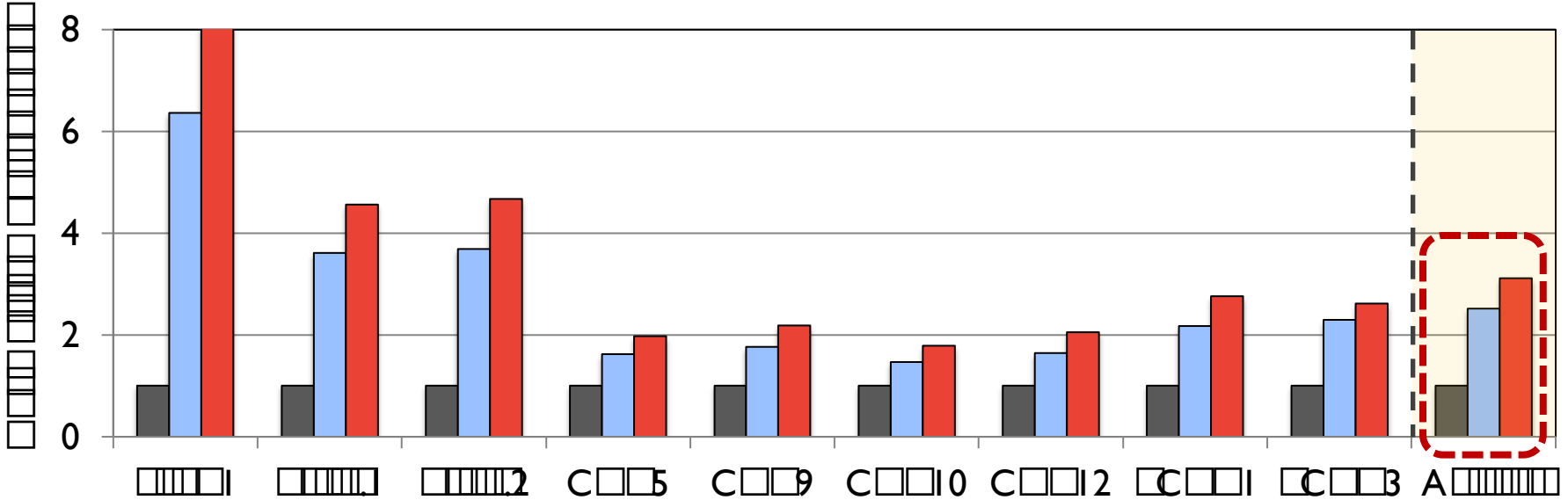


◆ CNN3
 ◆ CNN4
 ◆ CNN11
 ◆ CNN9
 ◆ CNN13





B B + HB



G

 3.1

E

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*

Accelerating Data-Intensive Workloads

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"**
Proceedings of the 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.
[[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

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

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

FPGA-based Processing Near Memory

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

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

Gagandeep Singh[◇] Mohammed Alser[◇] Damla Senol Cali[✕]

Dionysios Diamantopoulos[▽] Juan Gómez-Luna[◇]

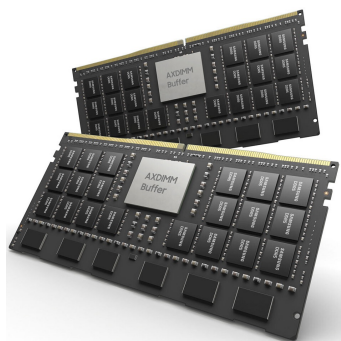
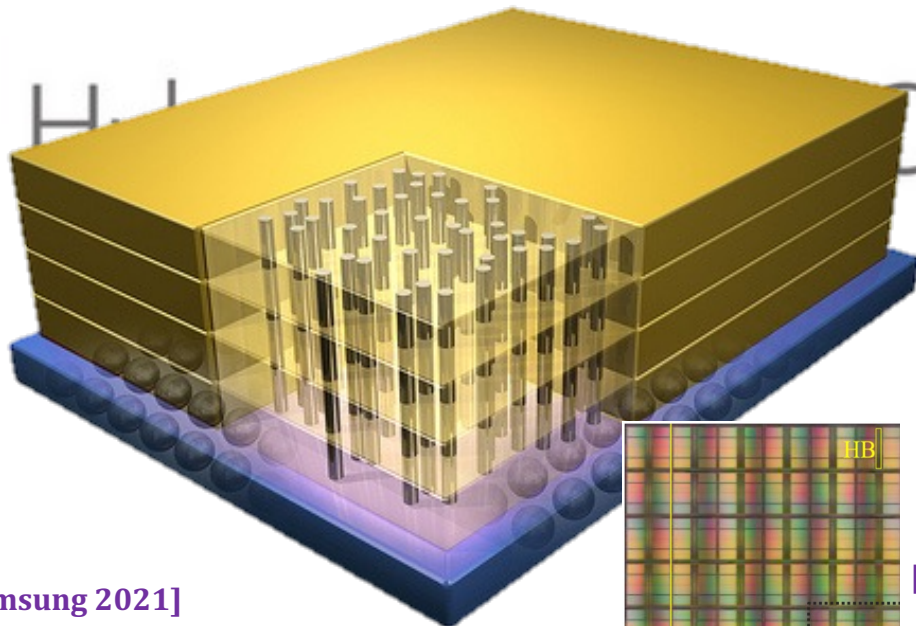
Henk Corporaal^{*} Onur Mutlu^{◇✕}

[◇]*ETH Zürich* [✕]*Carnegie Mellon University*

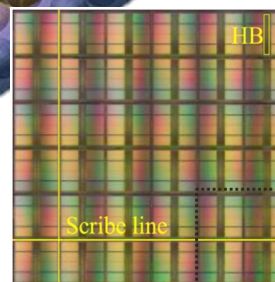
^{*}*Eindhoven University of Technology* [▽]*IBM Research Europe*

Processing-in-Memory in the Real World

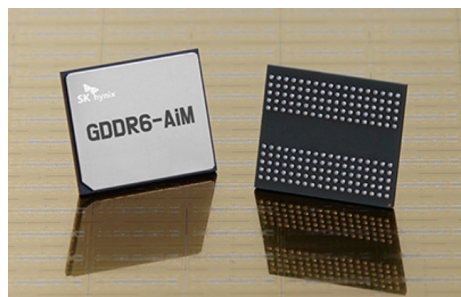
Processing-in-Memory Landscape Today



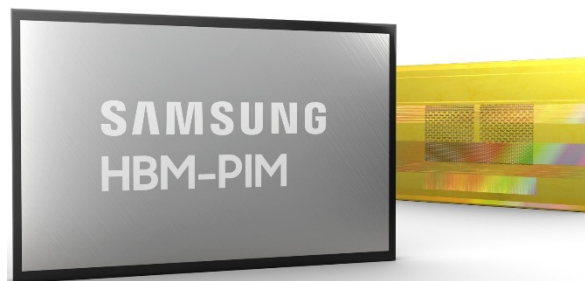
[Samsung 2021]



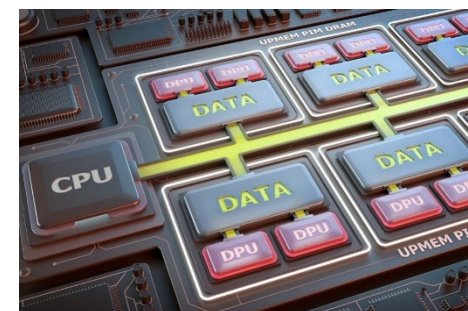
[Alibaba 2022]



[SK Hynix 2022]



[Samsung 2021]

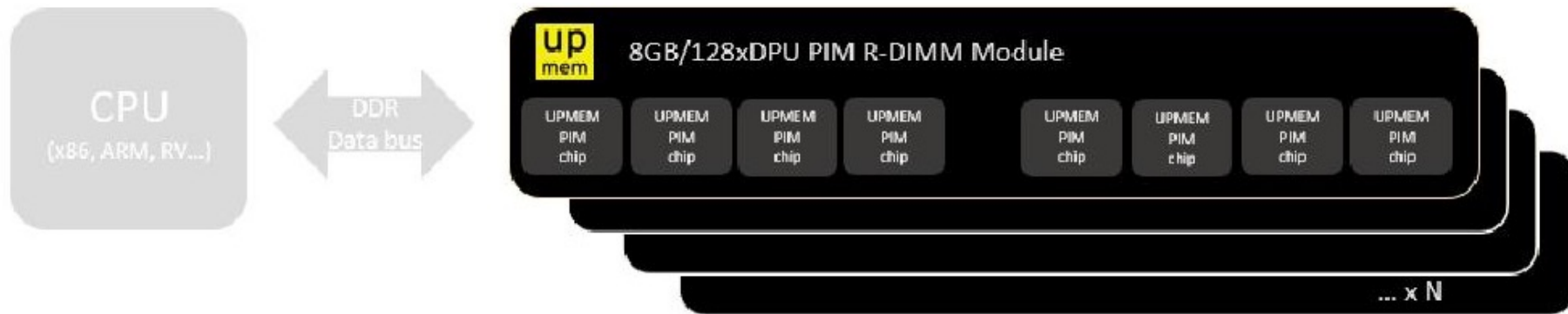


[UPMEM 2019]

This does not include many experimental chips and startups

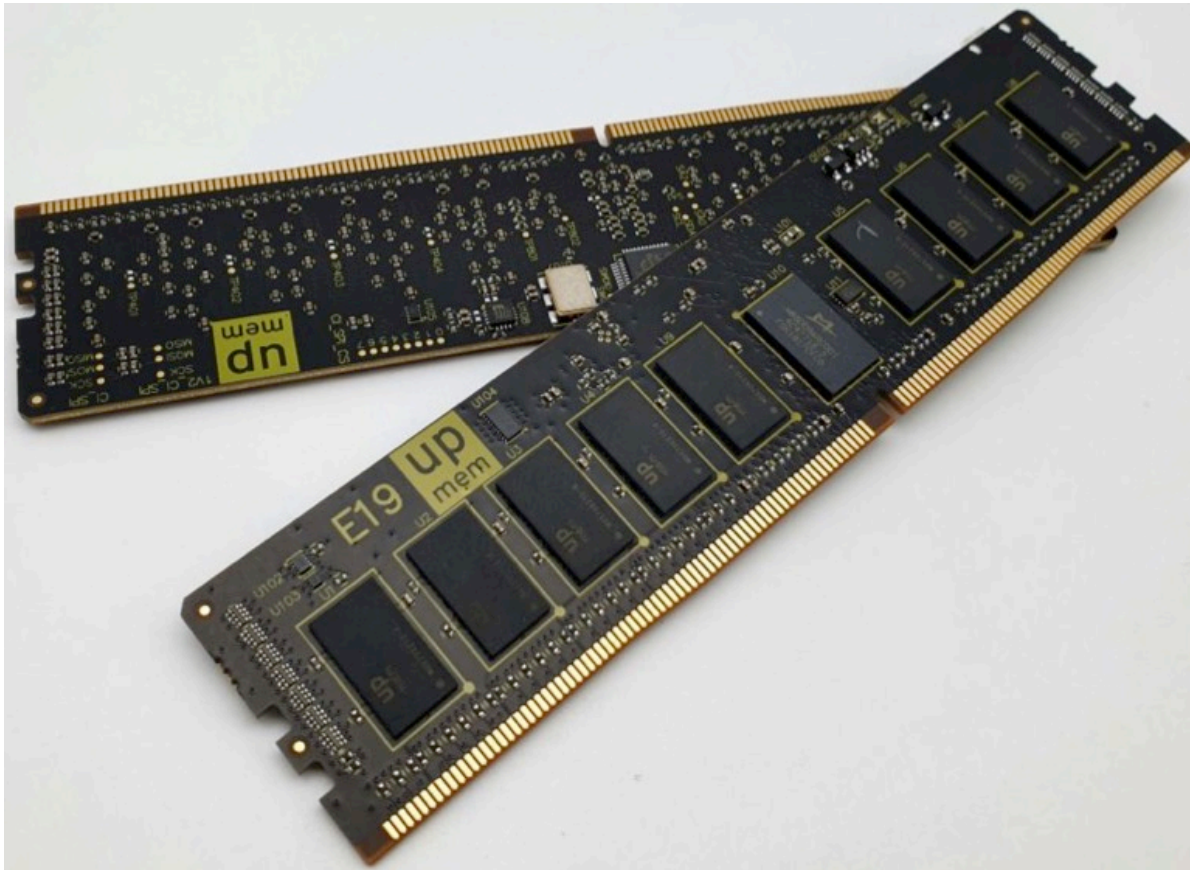
UPMEM Processing-in-DRAM Engine (2019)

- **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

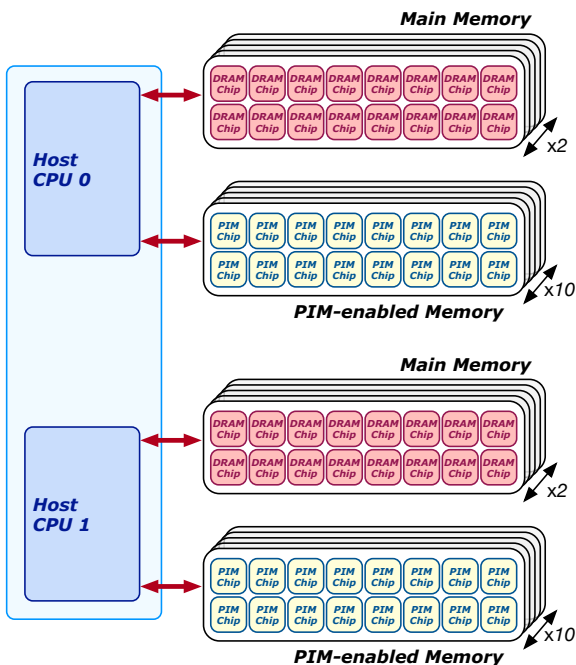


UPMEM Memory Modules

- 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



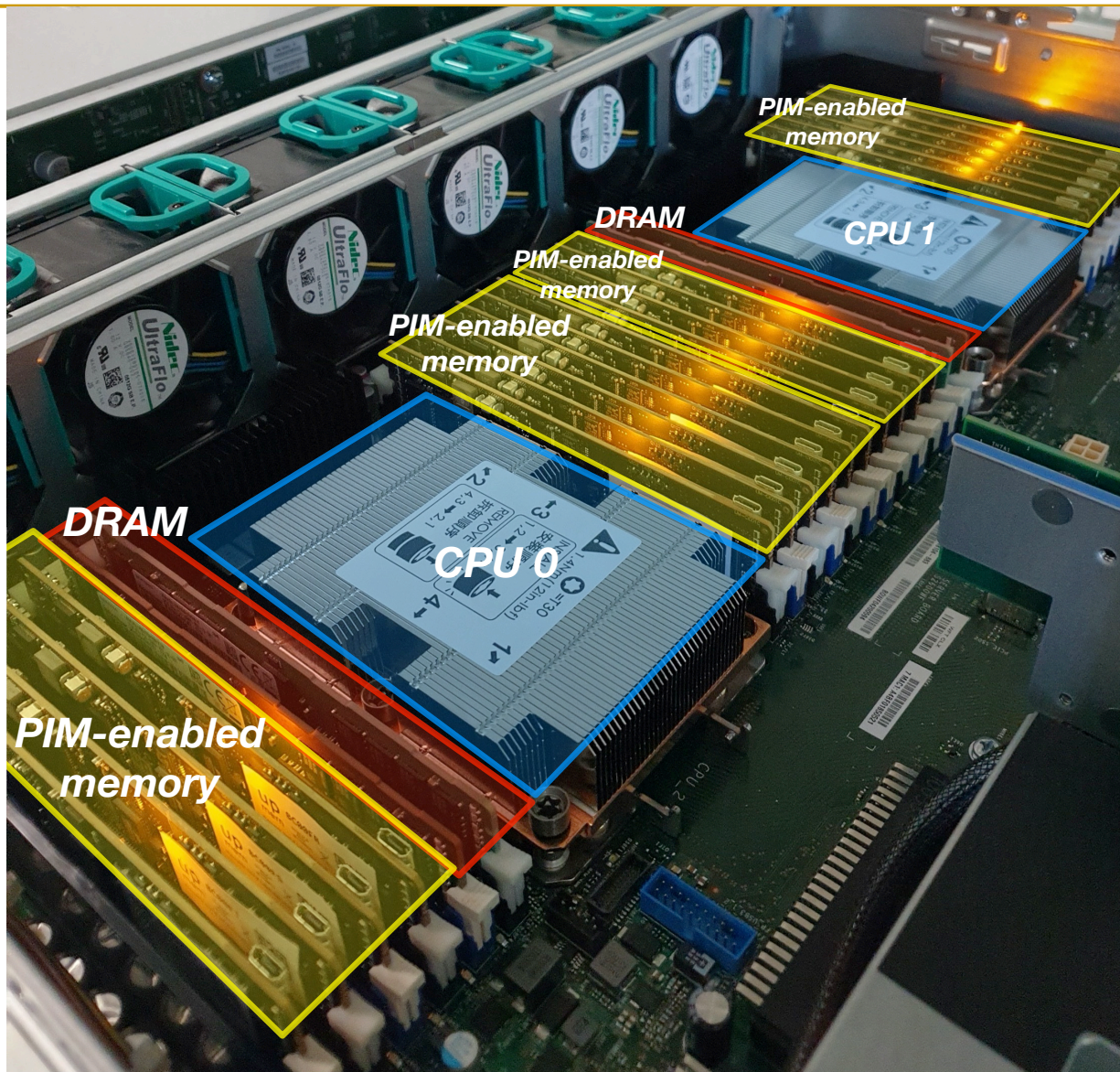
Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

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

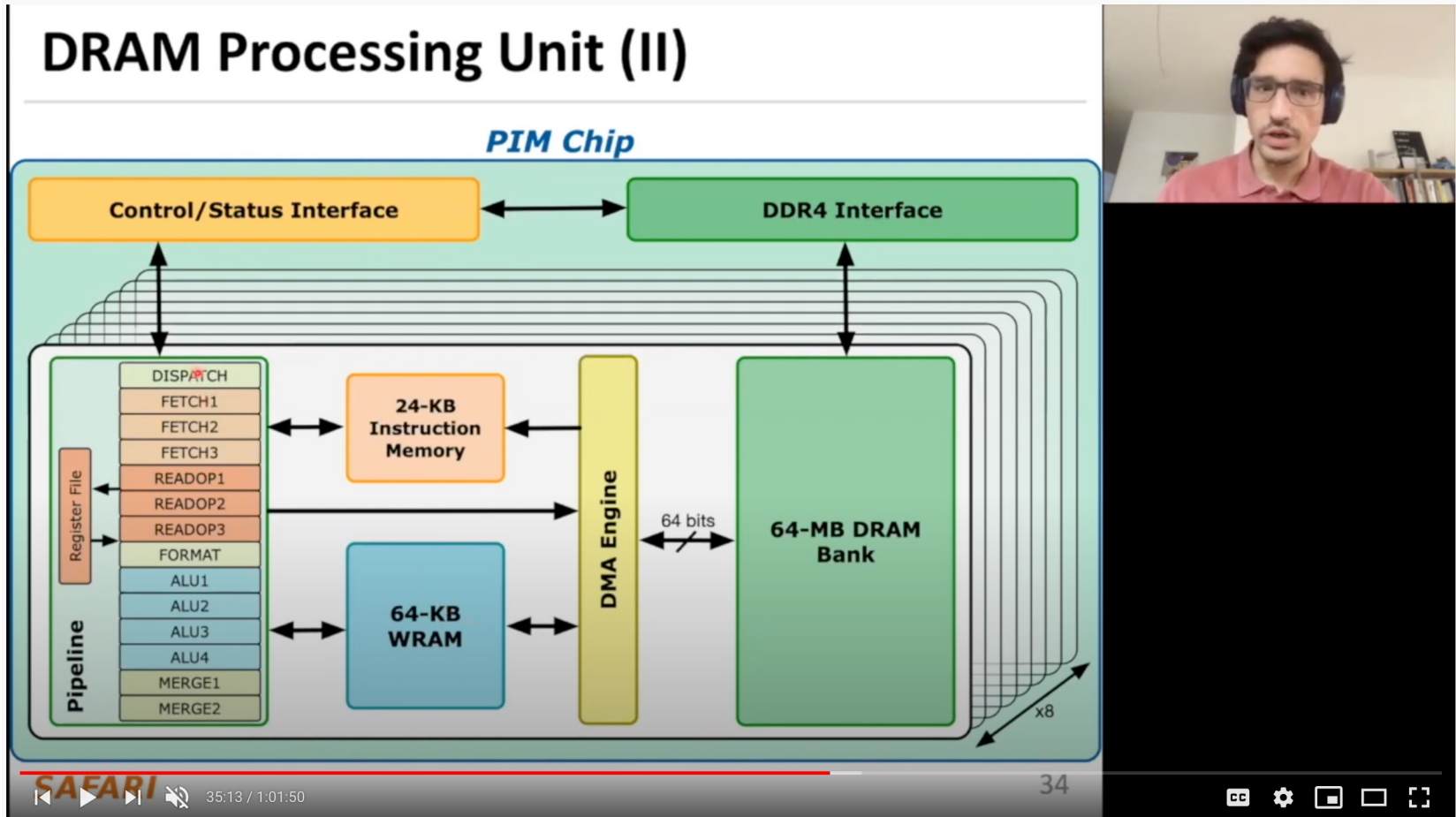
Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens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

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

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM (Processing-In-Memory benchmarks)*, a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 440 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.



More on the UPMEM PIM System



ETH ZÜRICH HAUPTGEBÄUDE

Computer Architecture - Lecture 12d: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)

1,120 views • Oct 31, 2020

30 0 SHARE SAVE ...



Onur Mutlu Lectures
16.7K subscribers

ANALYTICS

EDIT VIDEO

<https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIqBxUz7xRPS-wisBN&index=26>

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 FERNANDEZ^{1,3}, CHRISTINA GIANNOULA^{1,4},
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://arxiv.org/pdf/2105.03814.pdf>

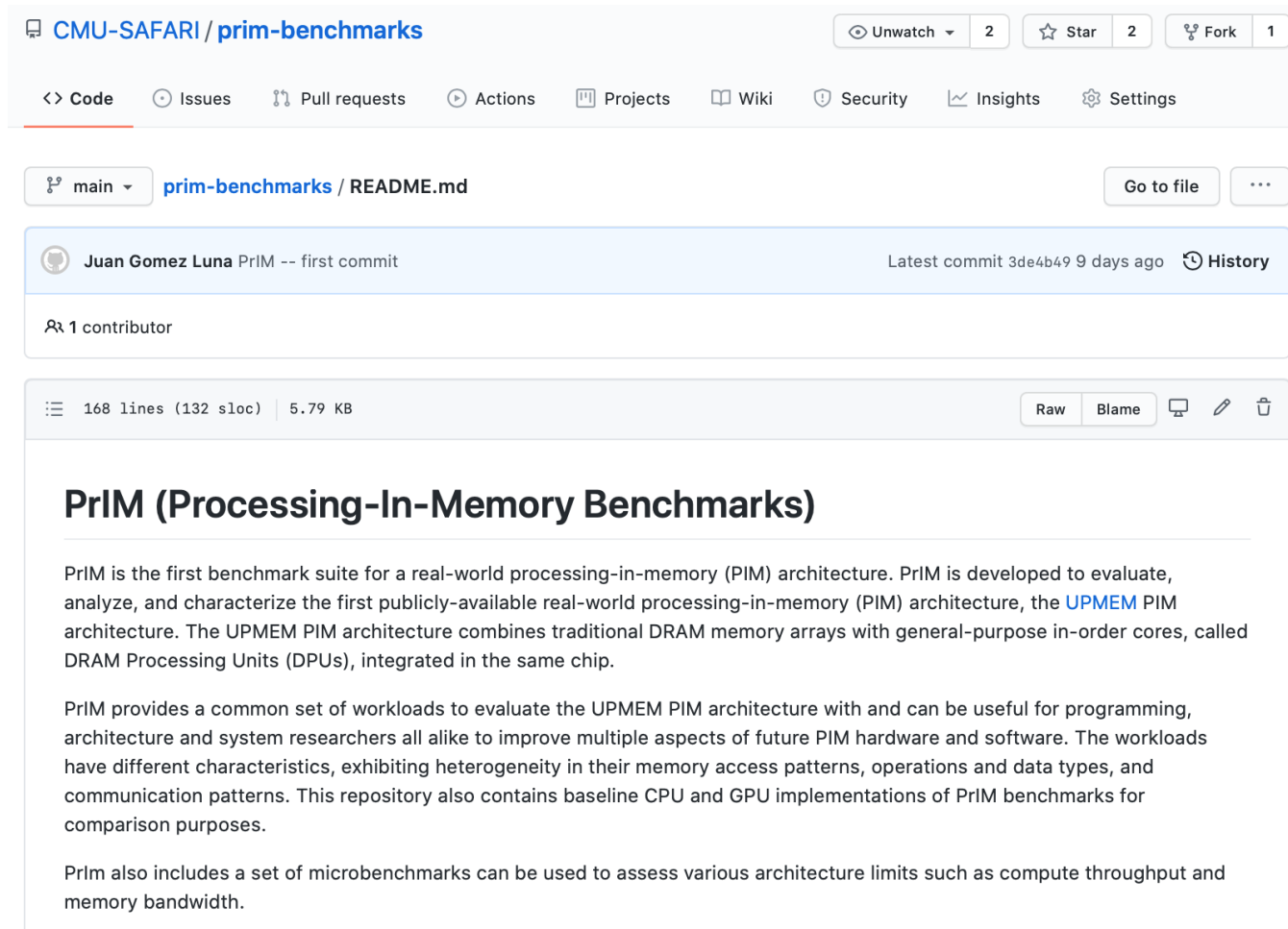
<https://github.com/CMU-SAFARI/prim-benchmarks>

PrIM Benchmarks: Application Domains

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

PrIM Benchmarks are Open Source

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



CMU-SAFARI / prim-benchmarks

Unwatch 2 Star 2 Fork 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main prim-benchmarks / README.md Go to file

Juan Gomez Luna PrIM -- first commit Latest commit 3de4b49 9 days ago History

1 contributor

168 lines (132 sloc) 5.79 KB Raw Blame

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.

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

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

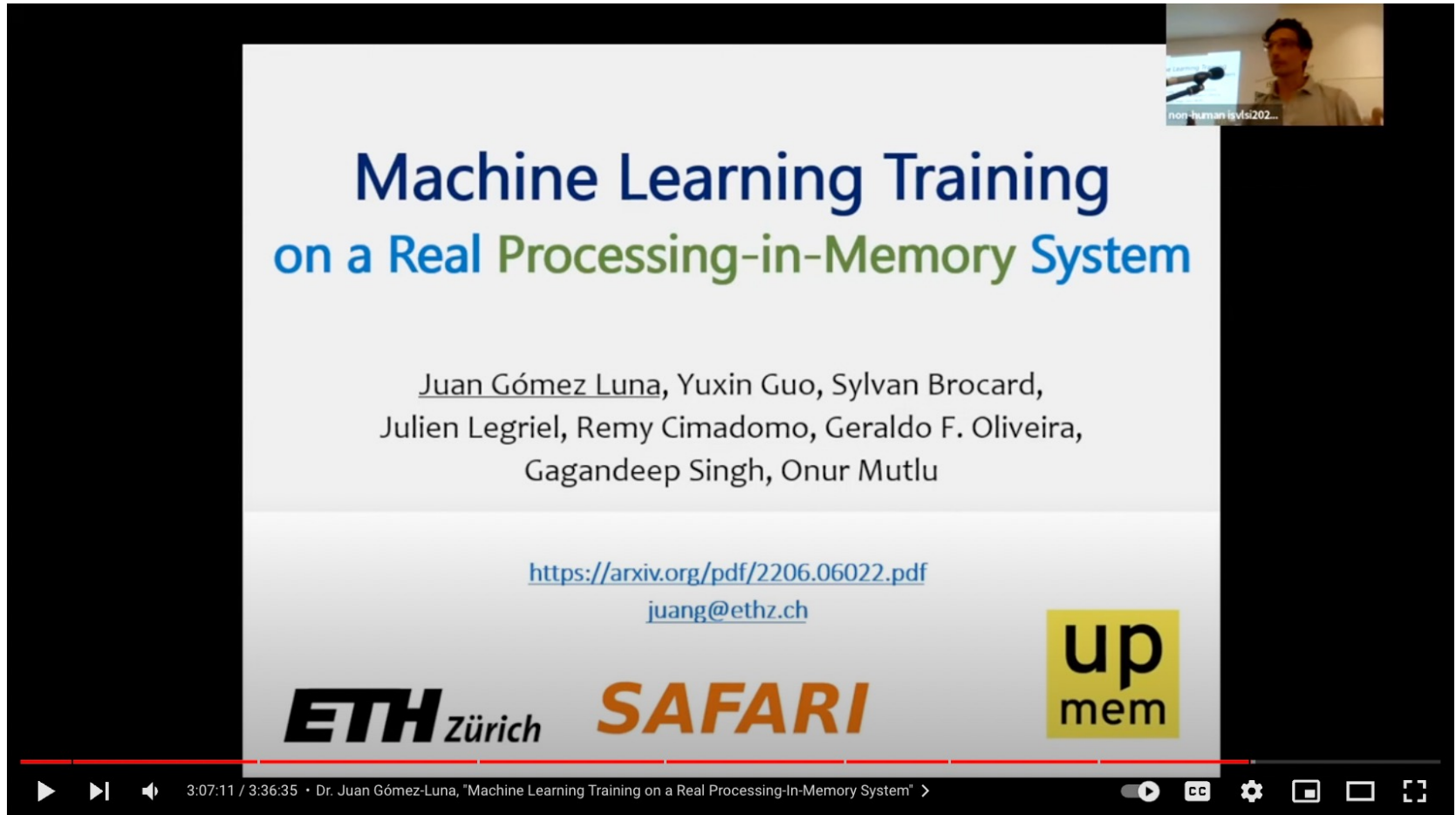
Long version: <https://arxiv.org/pdf/2207.07886.pdf>

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

ML Training on a Real PIM System

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

ML Training on Real PIM Talk Video



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

Juan Gómez Luna, Yuxin Guo, Sylvan Brocard,
Julien Legriel, Remy Cimadomo, Geraldo F. Oliveira,
Gagandeep Singh, Onur Mutlu

<https://arxiv.org/pdf/2206.06022.pdf>
juang@ethz.ch

ETH Zürich **SAFARI** up mem

3:07:11 / 3:36:35 • Dr. Juan Gómez-Luna, "Machine Learning Training on a Real Processing-In-Memory System" >

ISVLSI 2022 Special Session on Processing-in-Memory

1,345 views • Premiered Aug 9, 2022

61 DISLIKE SHARE DOWNLOAD CLIP SAVE ...

 **Onur Mutlu Lectures**
26.9K subscribers

ANALYTICS EDIT VIDEO

SpMV Multiplication on Real PIM Systems

- Appears in SIGMETRICS 2022

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

CHRISTINA GIANNOULA, ETH Zürich, Switzerland and National Technical University of Athens, Greece

IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland

NECTARIOS KOZIRIS, National Technical University of Athens, Greece

GEORGIOS GOUMAS, National Technical University of Athens, Greece

ONUR MUTLU, ETH Zürich, Switzerland

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

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

Sequence Alignment on Real PIM Systems

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

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

Safaa Diab¹ Amir Nassereldine¹ Mohammed Alser² Juan Gómez Luna²
Onur Mutlu² Izzat El Hajj¹

¹American University of Beirut ²ETH Zürich

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

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

Summary

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

Samsung Function-in-Memory DRAM (2021)



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

Korea on February 17, 2021

Audio



Share



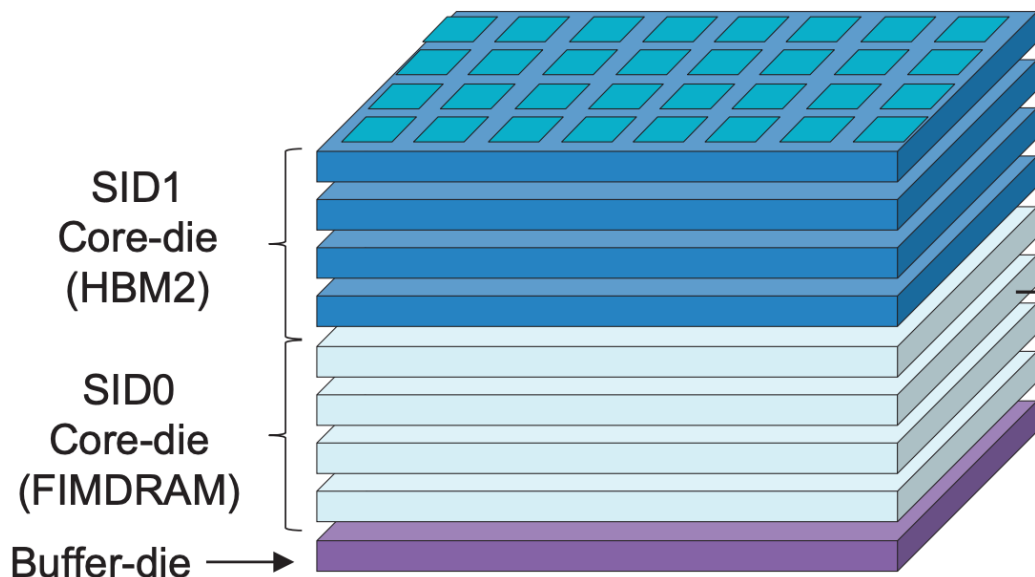
The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%

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

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

Samsung Function-in-Memory DRAM (2021)

■ FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

Chip Specification

128DQ / 8CH / 16 banks / BL4

32 PCU blocks (1 FIM block/2 banks)

1.2 TFLOPS (4H)

**FP16 ADD /
Multiply (MUL) /
Multiply-Accumulate (MAC) /
Multiply-and- Add (MAD)**

ISSCC 2021 / SESSION 25 / DRAM / 25.4

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

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang², Shinhaeng Kang¹, Yuhwan Ro³, Seungwoo Seo³, JoonHo Song³, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwaseong, Korea

²Samsung Electronics, San Jose, CA

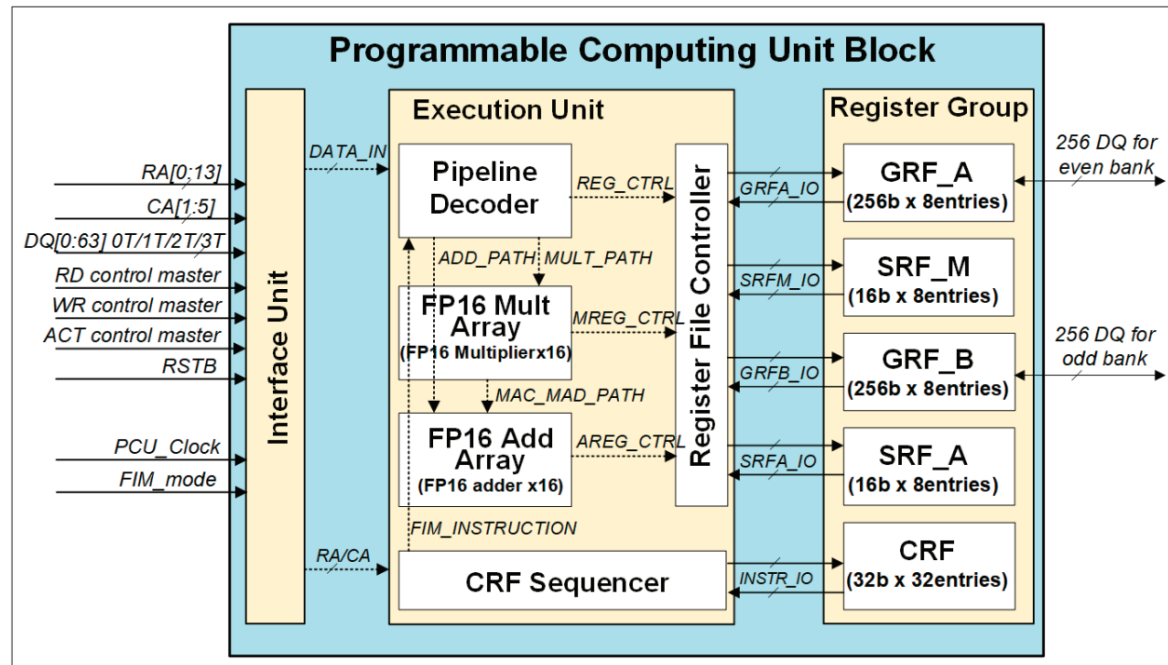
³Samsung Electronics, Suwon, Korea

Samsung Function-in-Memory DRAM (2021)

Programmable Computing Unit

■ Configuration of PCU block

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



[Block diagram of PCU in FIMDRAM]

ISSCC 2021 / SESSION 25 / DRAM / 25.4

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

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhaeng Kang¹, Yuhwan Ro¹, Seungwoo Seo¹, JoonHo Song¹, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwasong, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea

Samsung Function-in-Memory DRAM (2021)

[Available instruction list for FIM operation]

Type	CMD	Description
Floating Point	ADD	FP16 addition
	MUL	FP16 multiplication
	MAC	FP16 multiply-accumulate
	MAD	FP16 multiply and add
Data Path	MOVE	Load or store data
	FILL	Copy data from bank to GRFs
Control Path	NOP	Do nothing
	JUMP	Jump instruction
	EXIT	Exit instruction

ISSCC 2021 / SESSION 25 / DRAM / 25.4

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

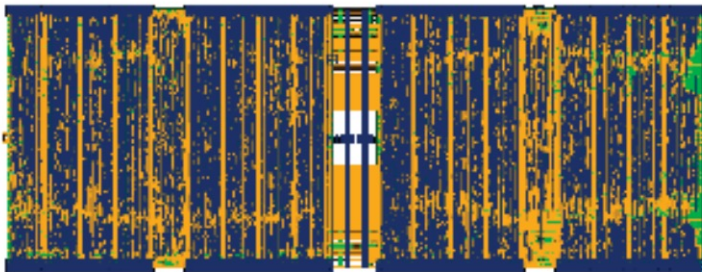
Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyoon Choi¹, Hyun-Sung Shin¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang¹, Shinhaeng Kang¹, Yuhwan Ro¹, Seungwoo Seo¹, JoonHo Song¹, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

¹Samsung Electronics, Hwasong, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea

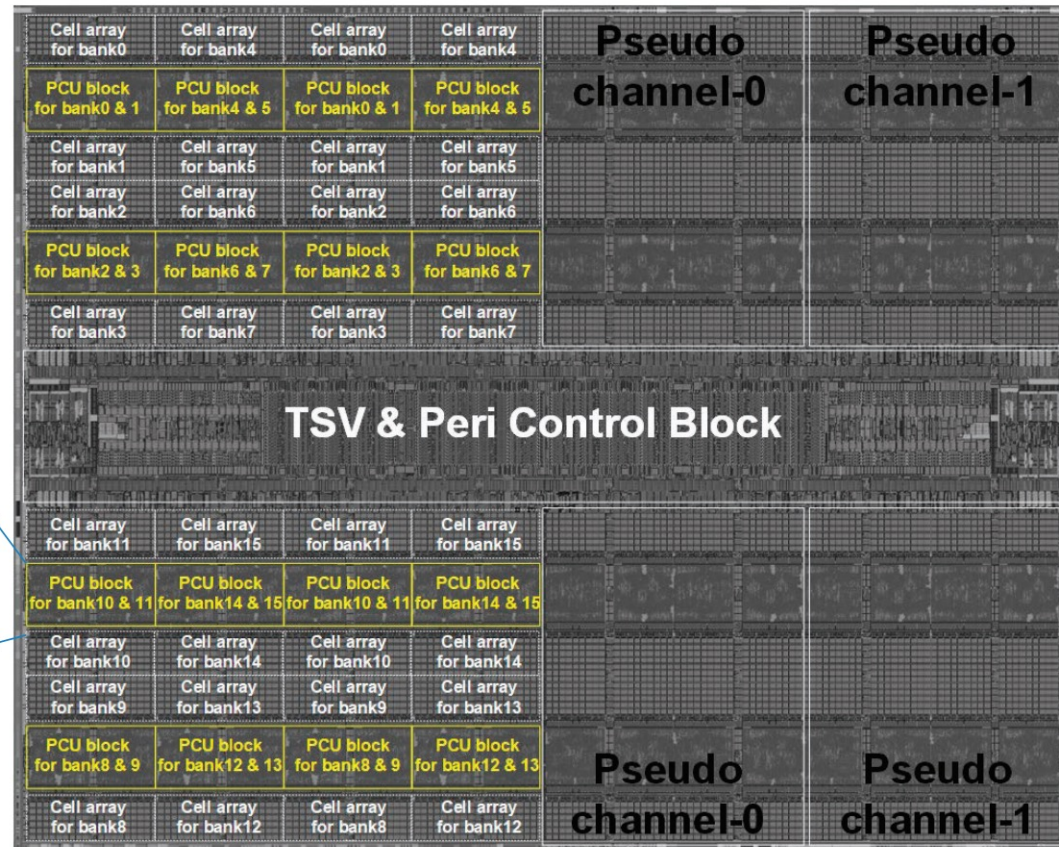
Samsung Function-in-Memory DRAM (2021)

Chip Implementation

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



[Digital RTL design for PCU block]



ISSCC 2021 / SESSION 25 / DRAM / 25.4

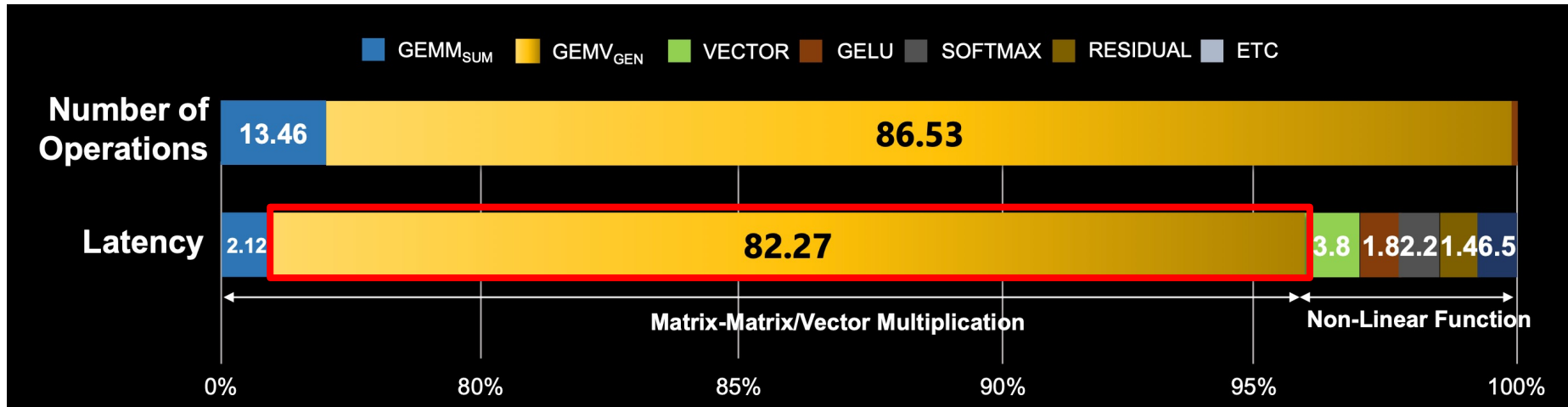
25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hyuk Kwon¹, Je Min Ryu¹, Jong-Pil Son¹, Seongil O¹, Hak-Soo Yu¹, Haesuk Lee¹, Soo Young Kim¹, Youngmin Cho¹, Jin Guk Kim¹, Jongyeon Choi¹, Hyun-Sung Shim¹, Jin Kim¹, BengSeng Phuah¹, HyoungMin Kim¹, Myeong Jun Song¹, Ahn Choi¹, Daeho Kim¹, SooYoung Kim¹, Eun-Bong Kim¹, David Wang², Shrinraeng Kang³, Yulwan Ro³, Seungwoo Seo³, JoonHo Song³, Jaeyoun Youn¹, Kyomin Sohn¹, Nam Sung Kim¹

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

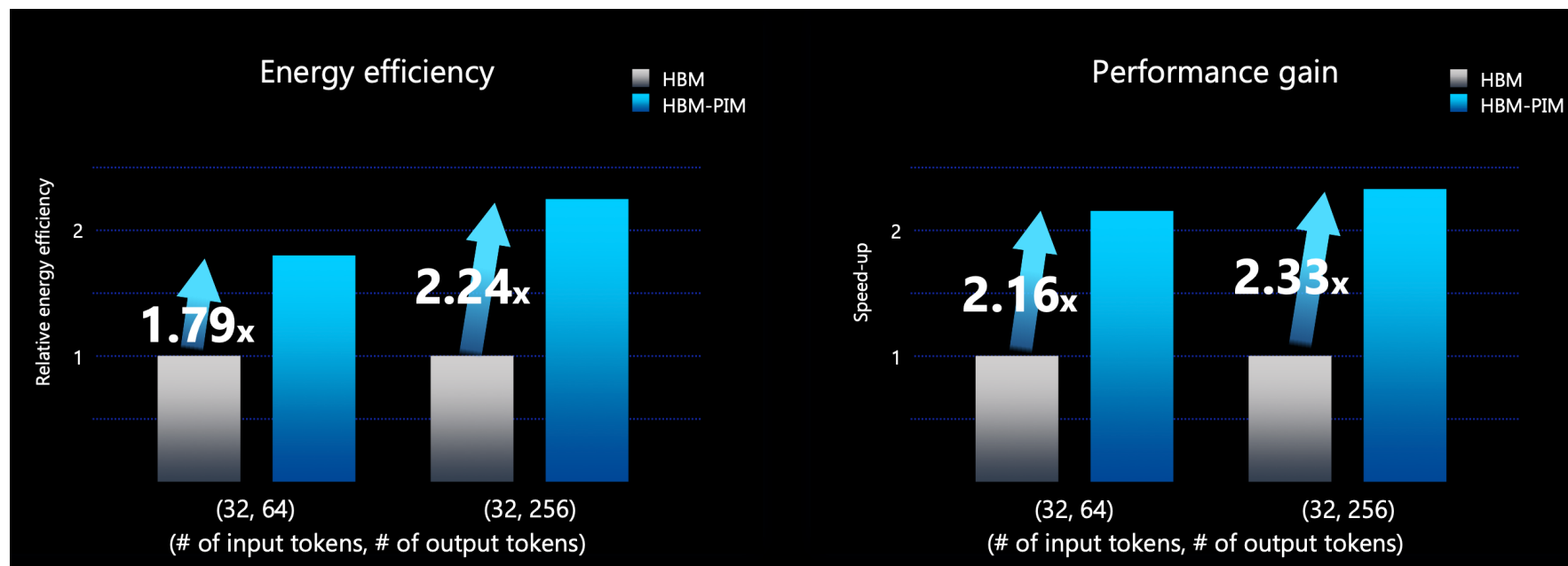
Samsung PNM Solutions for Generative AI (2023)

- Main target: **transformer** decoders used in **ChatGPT, GPT-3**
 - **Compute-bound step**: Summarization
 - **Memory-bound step**: Generation
 - Most of the execution time is spent on the **memory copy** from the **host CPU memory** to the **CPU memory**
- **GEMV** portion can be **60%-80%** of total generation latency, which is the target of PIM/PNM



Solution I: Samsung's HBM-PIM (2023)

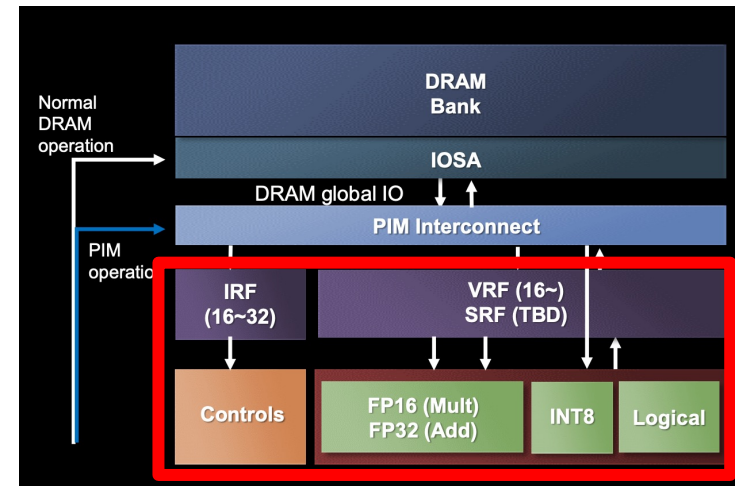
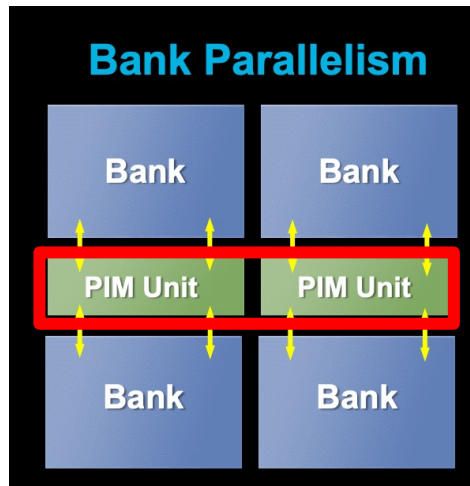
- AMD MI100 GPUs fabricated with HBM-PIM
- Experimental setup: GPT-J (6B, 32 input tokens), single AMD MI100-PIM GPU



- GPT can be accelerated by more than 2x over baseline

Solution II: Samsung's LPDDR-PIM (2023)

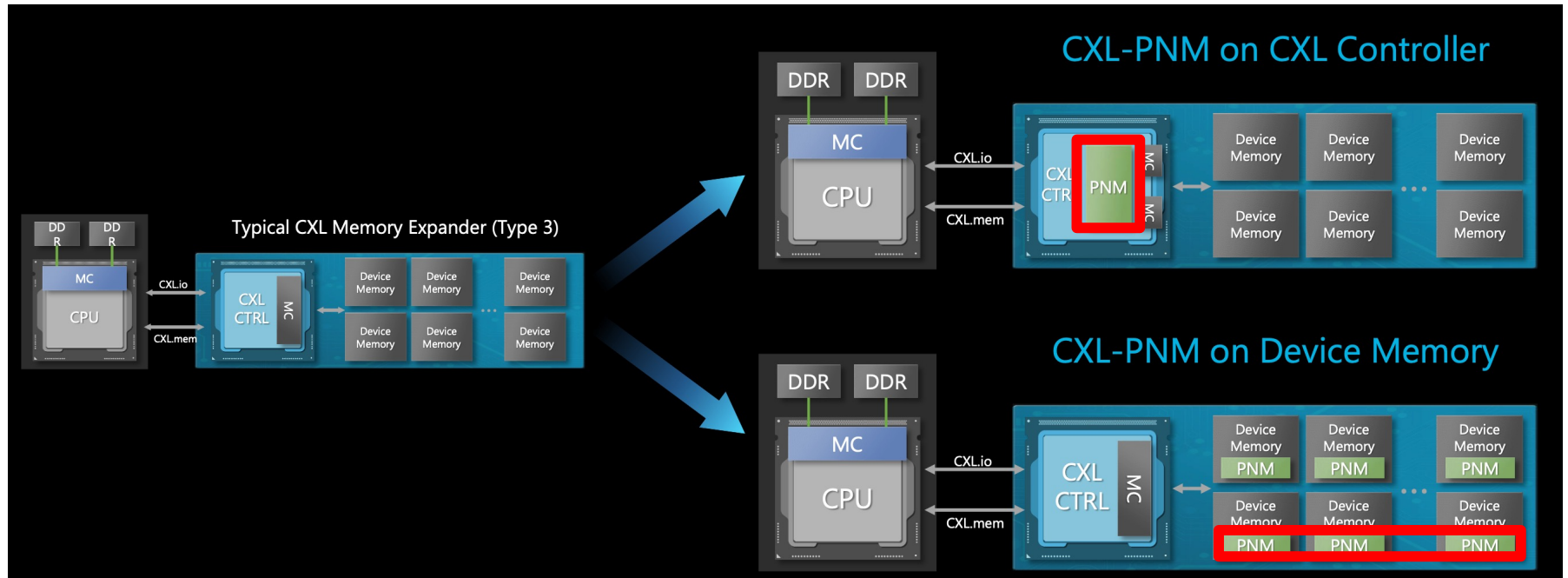
- PIM for on-device generative AI
 - Datacenter **costs** and **power consumption** are increasing due to the growing demand for cloud AI
- LPDDR-PIM improves **battery life** by preventing memory over-provisioning just for bandwidth



- 4.47x **performance gains** and 70.6% **energy reduction** in GPT-2

Solution III: Samsung's CXL-PNM (2023)

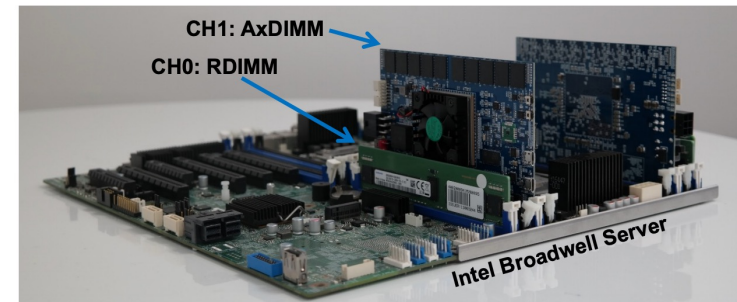
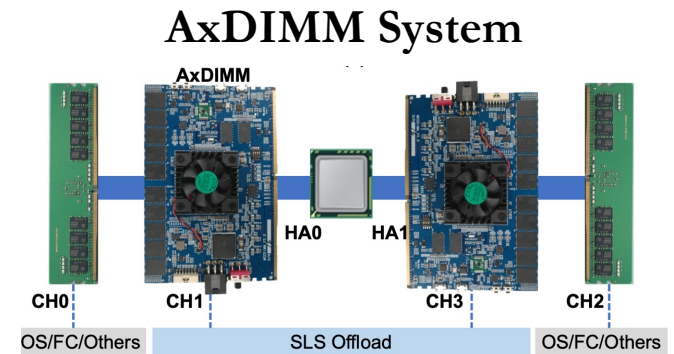
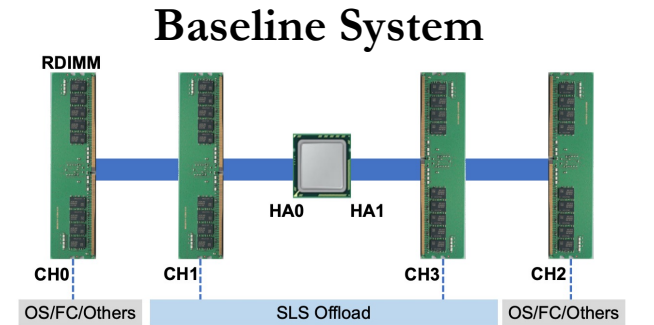
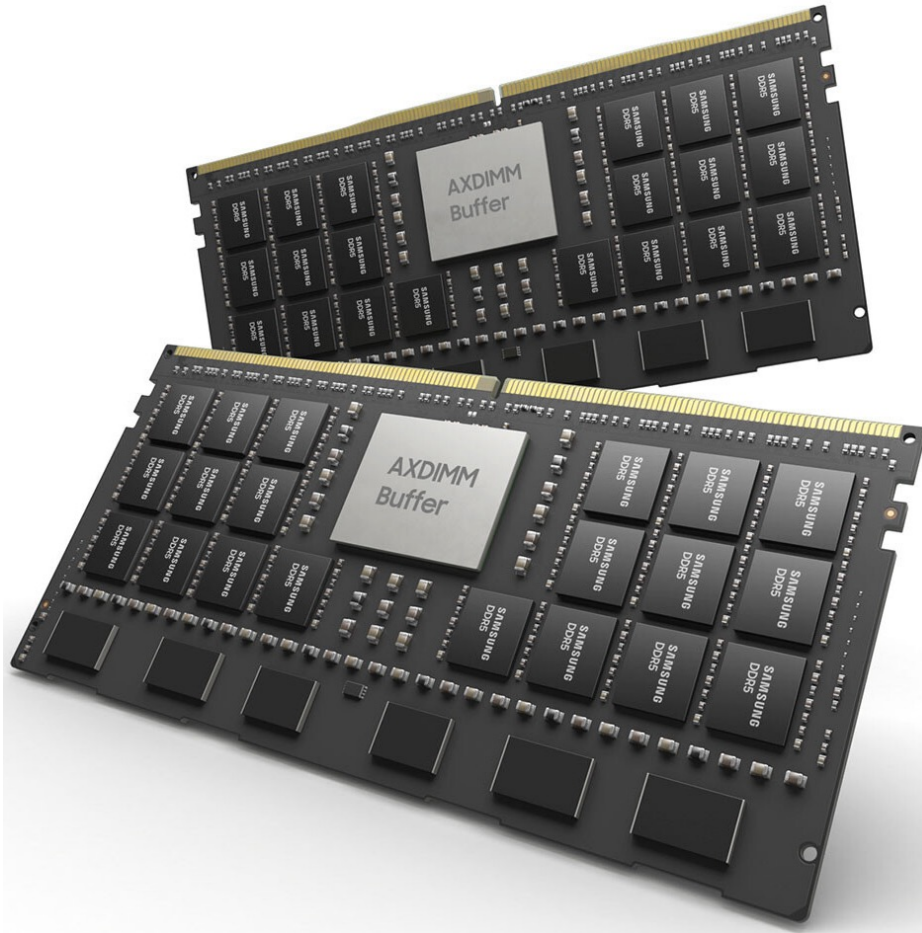
- A CXL-based processing-near-memory solution
 - Improves **capacity, bandwidth, and power**
 - Large-scale large-language models are often **capacity-bound**



- Multiple CXL-PNM can offer **4.4x higher energy efficiency** and **53% higher throughput** than multiple GPUs

Samsung AxDIMM (2021)

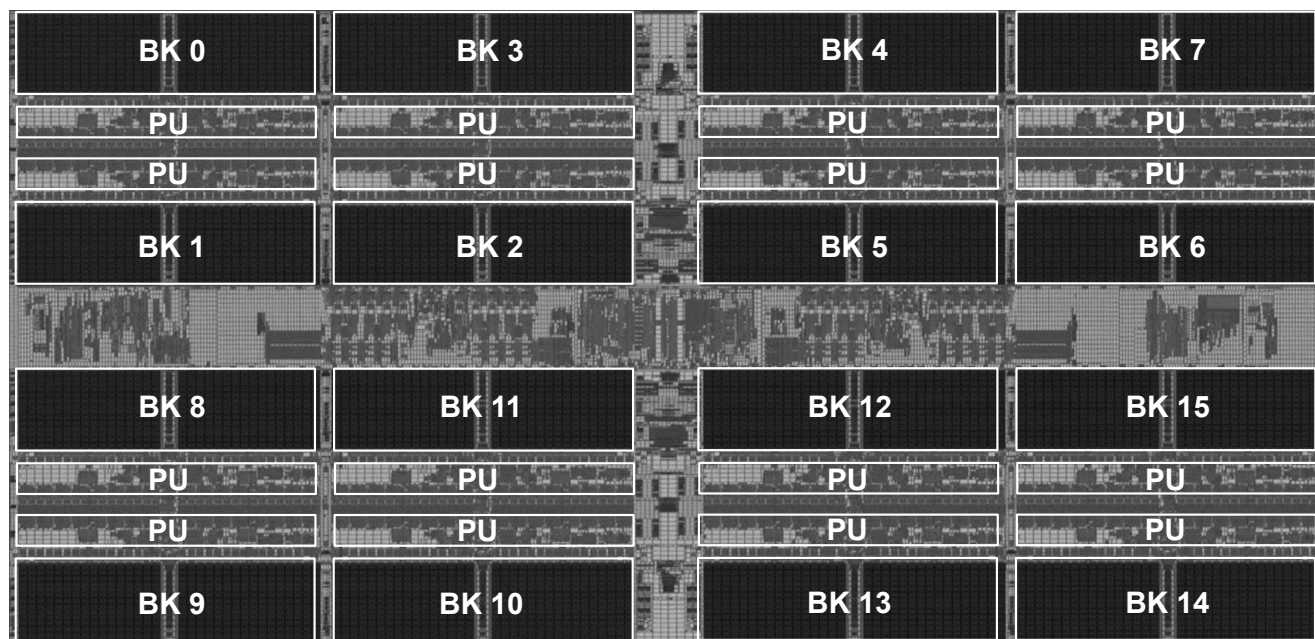
- DDRx-PIM
 - Deep learning recommendation system



SK Hynix AiM: Chip Implementation (2022)

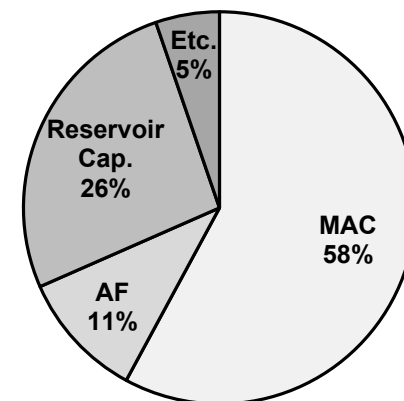
- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph



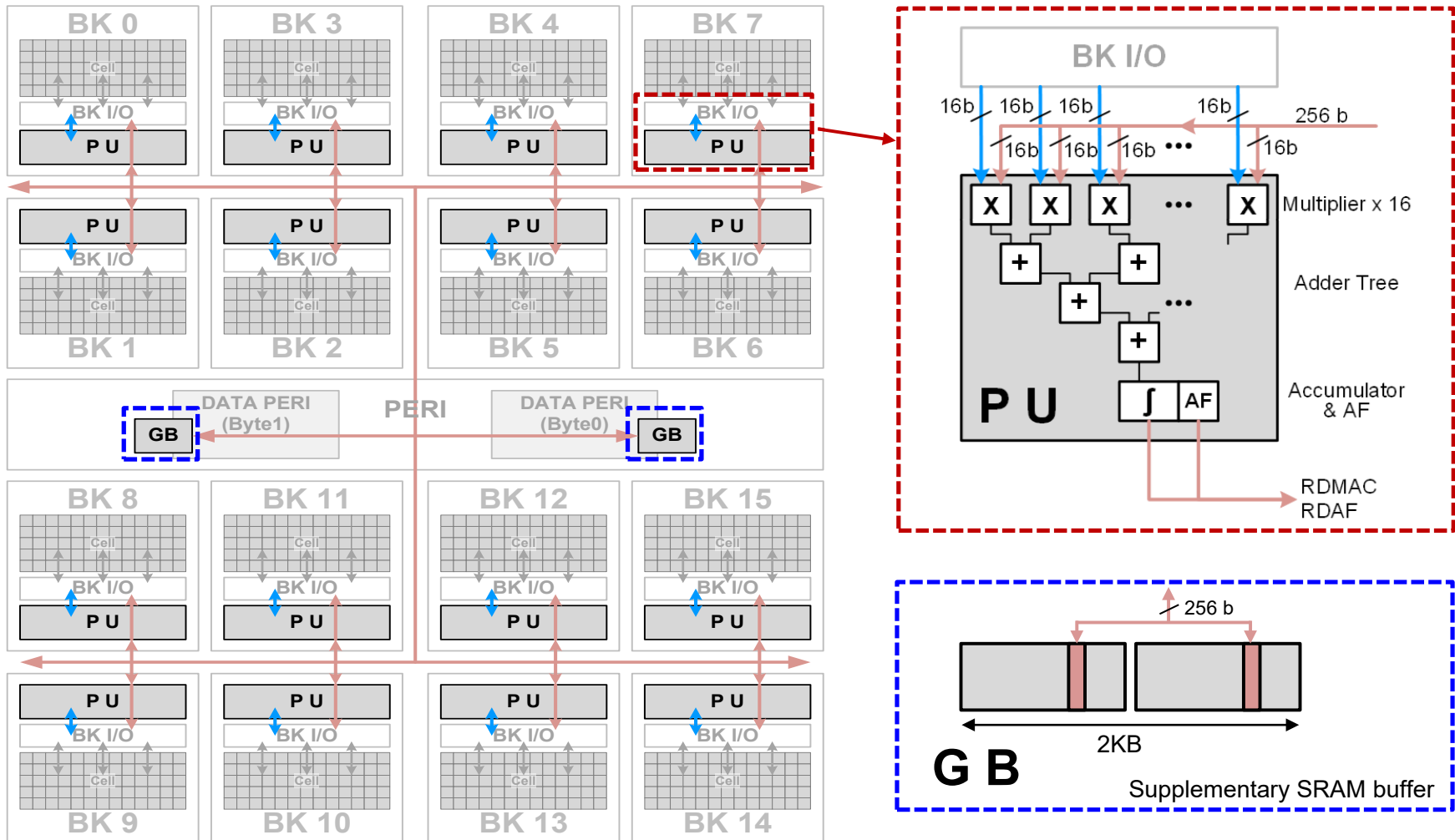
1 Process Unit (PU) Area

Total	0.19mm ²
MAC	0.11mm ²
Activation Function (AF)	0.02mm ²
Reservoir Cap.	0.05mm ²
Etc.	0.01mm ²



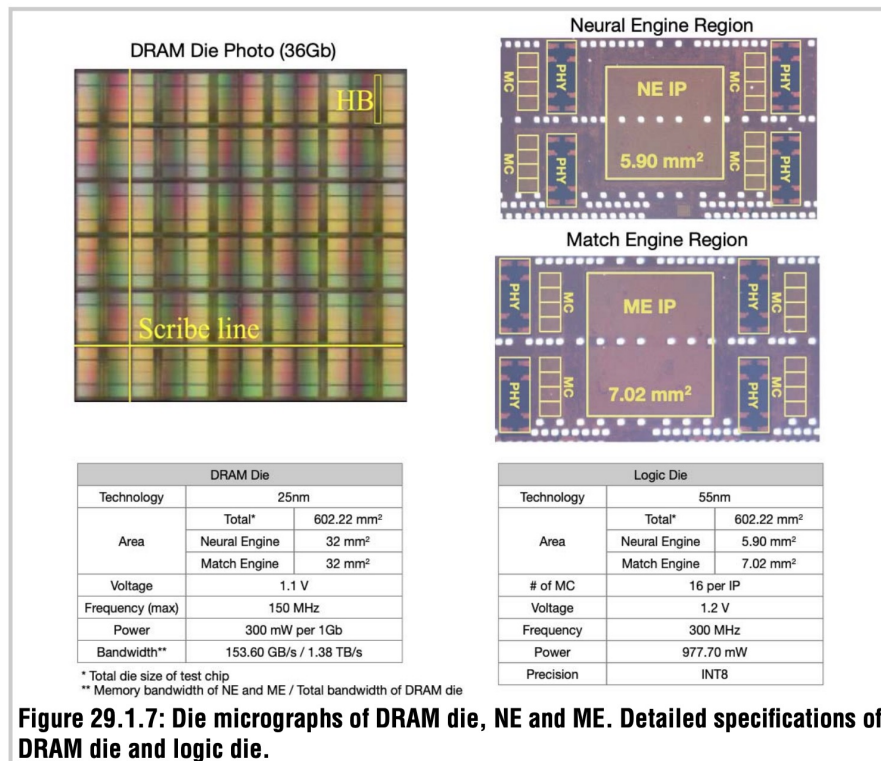
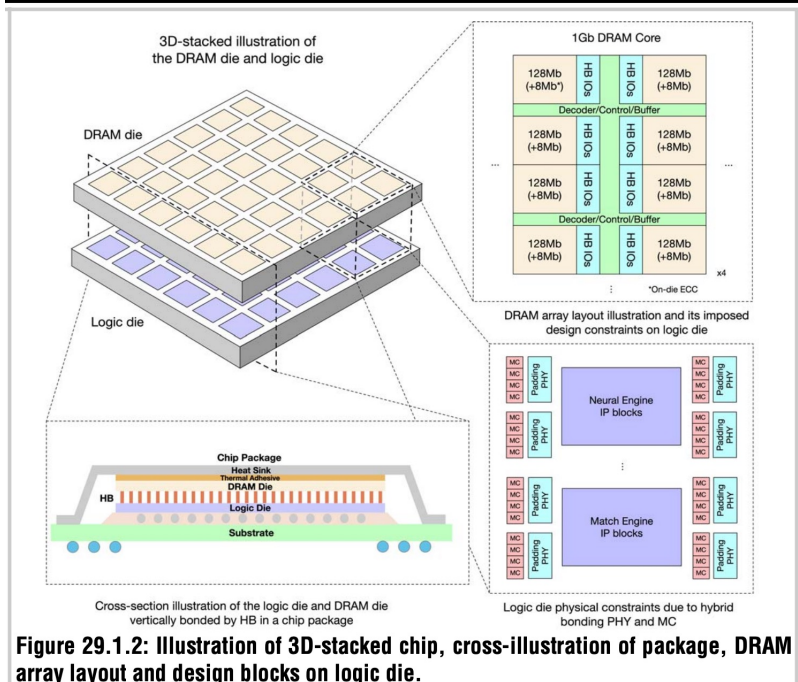
SK Hynix AiM: System Organization (2022)

■ GDDR6-based AiM architecture



AliBaba PIM Recommendation System (2022)

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



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

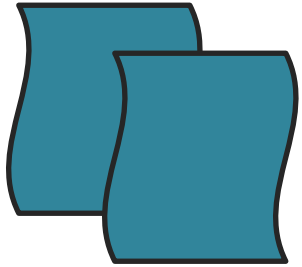
Dimin Niu¹, Shuangchen Li¹, Yuhao Wang¹, Wei Han¹, Zhe Zhang², Yijin Guan², Tianchan Guan³, Fei Sun¹, Fei Xue¹, Lide Duan¹, Yuanwei Fang¹, Hongzhong Zheng¹, Xiping Jiang⁴, Song Wang⁴, Fengguo Zuo⁴, Yubing Wang⁴, Bing Yu⁴, Qiwei Ren⁴, Yuan Xie¹

Processing in Memory: Two Approaches

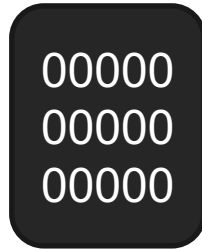
1. Processing near Memory
2. Processing using Memory

Starting Simple: Data Copy and Initialization

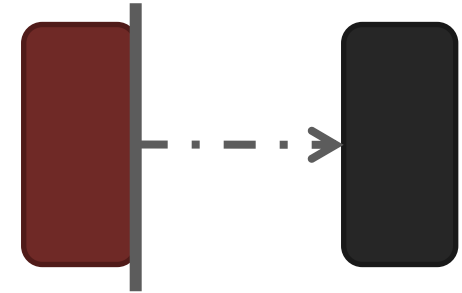
□□□□□□□□ & □□□□□□□□ 5% □□□□□□□□ G □□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ □□□□□□□□



Forking



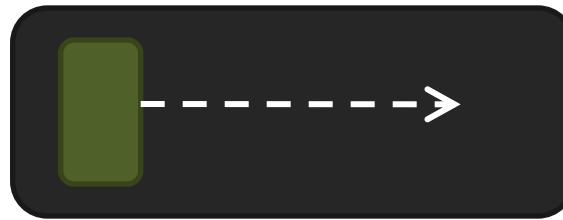
**Zero initialization
(e.g., security)**



Checkpointing



**VM Cloning
Deduplication**



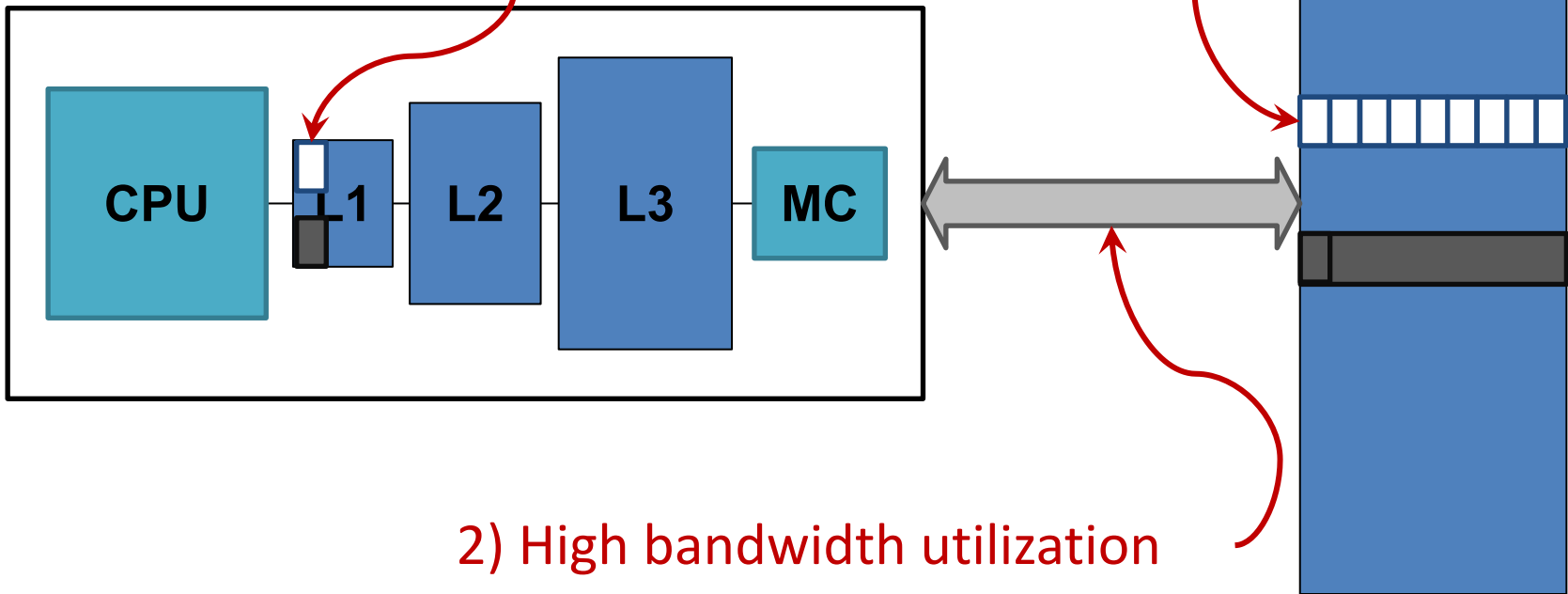
Page Migration

•••
Many more

Today's Systems: Bulk Data Copy

1) High latency

3) Cache pollution

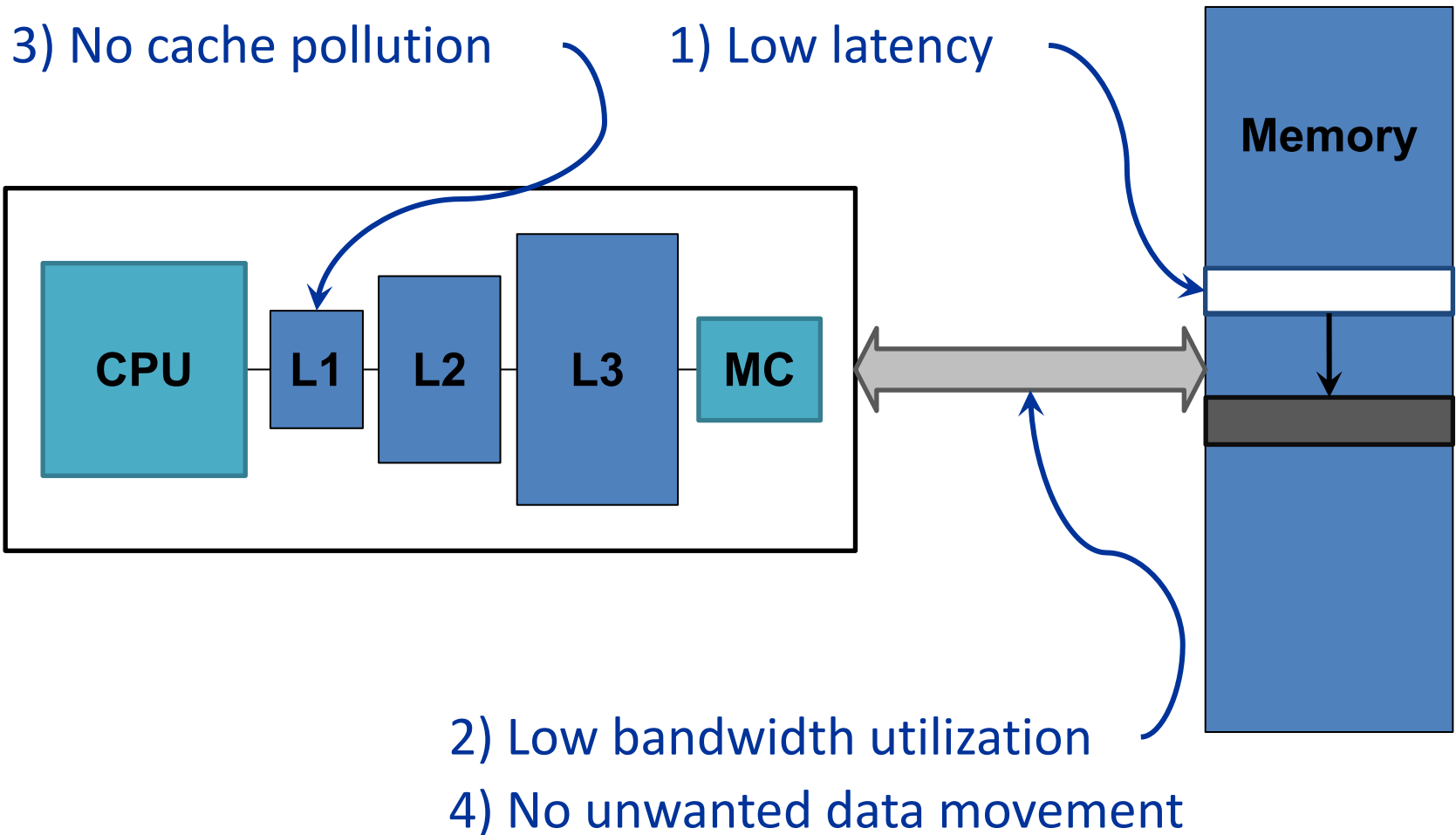


2) High bandwidth utilization

4) Unwanted data movement

1046ns, 3.6uJ (for 4KB page copy via DMA)

Future Systems: In-Memory Copy

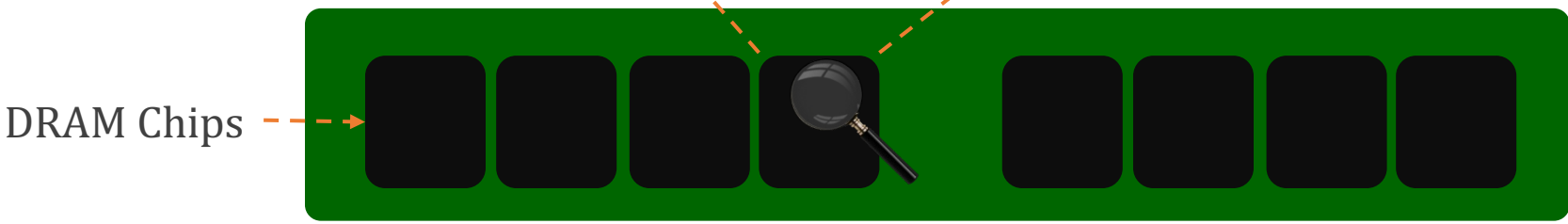
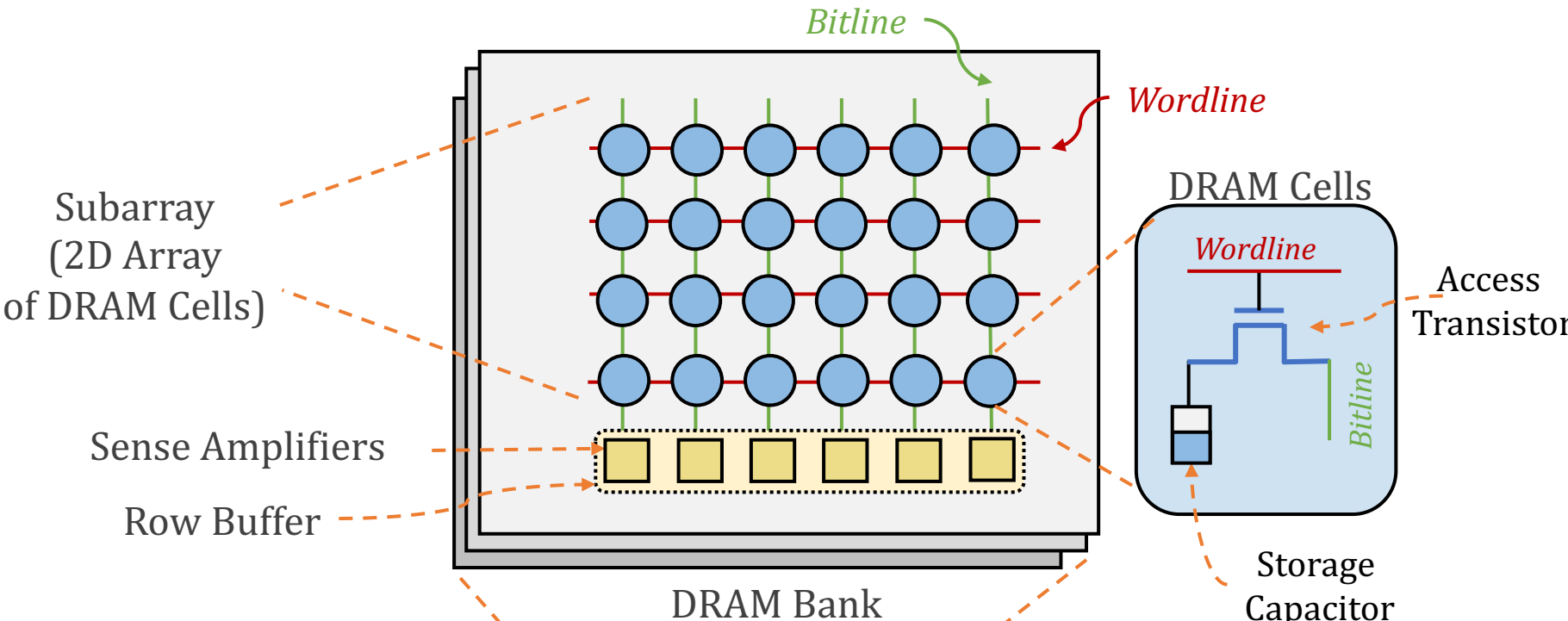


1046ns, 3.6uJ → 90ns, 0.04uJ

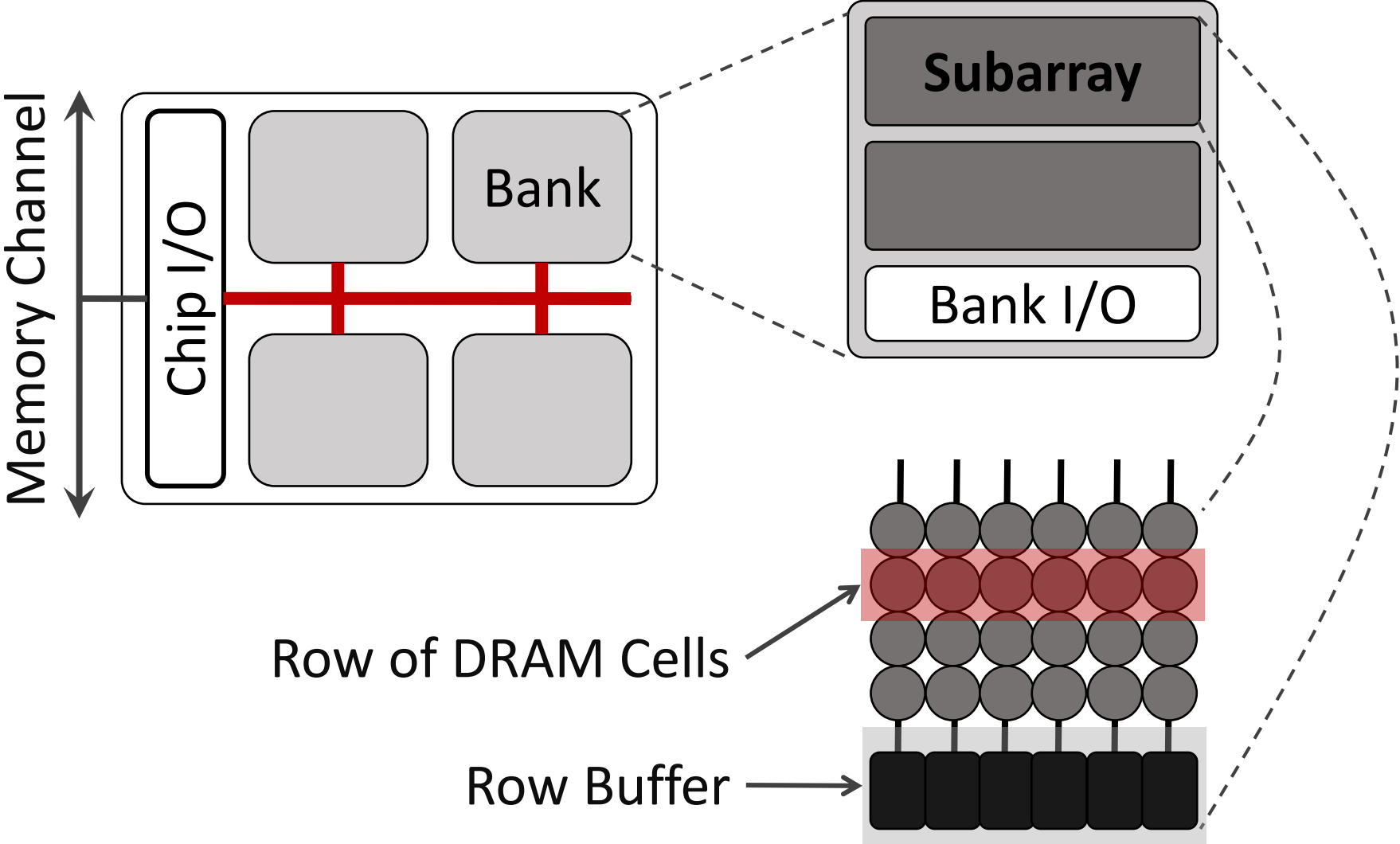
Brief Review:

Inside A DRAM Chip

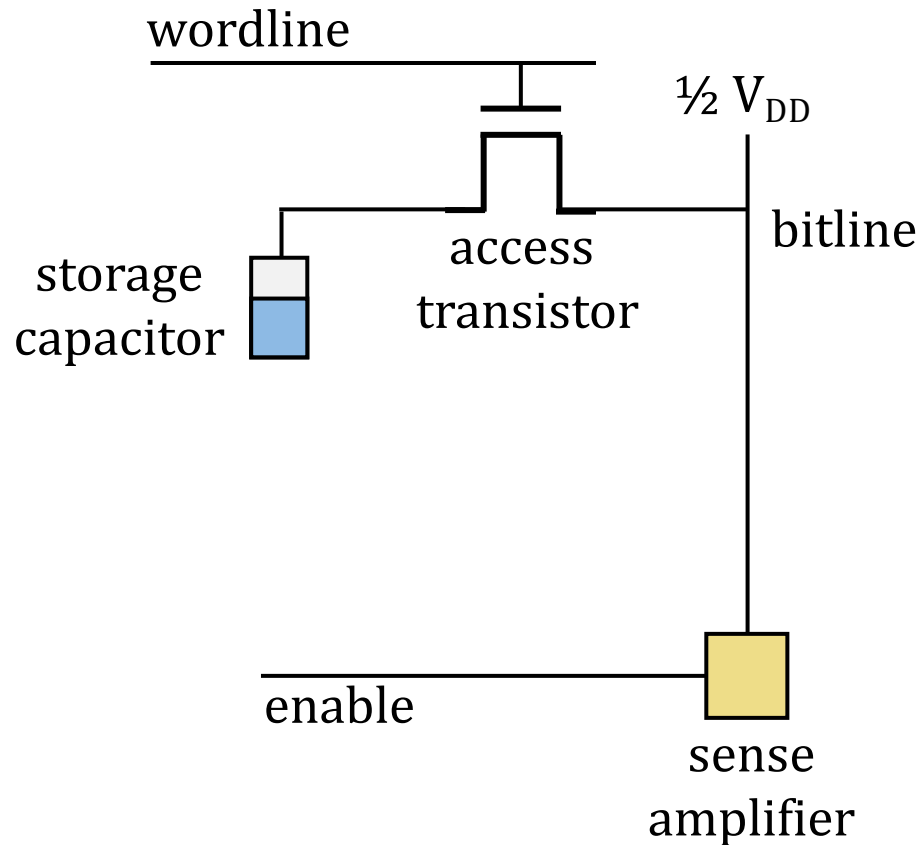
Inside a DRAM Chip



Inside a DRAM Chip: Another View



DRAM Cell Operation

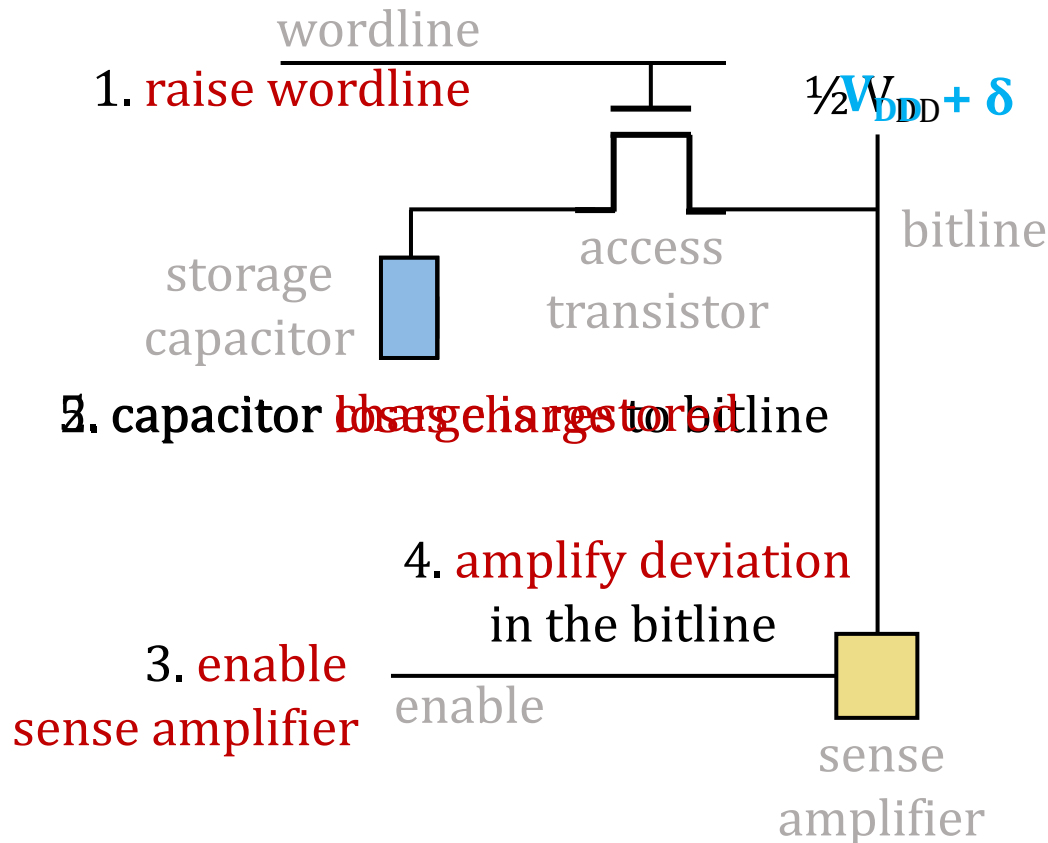


1. ACTIVATE (ACT)

2. READ/WRITE

3. PRECHARGE (PRE)

DRAM Cell Operation (1/3)

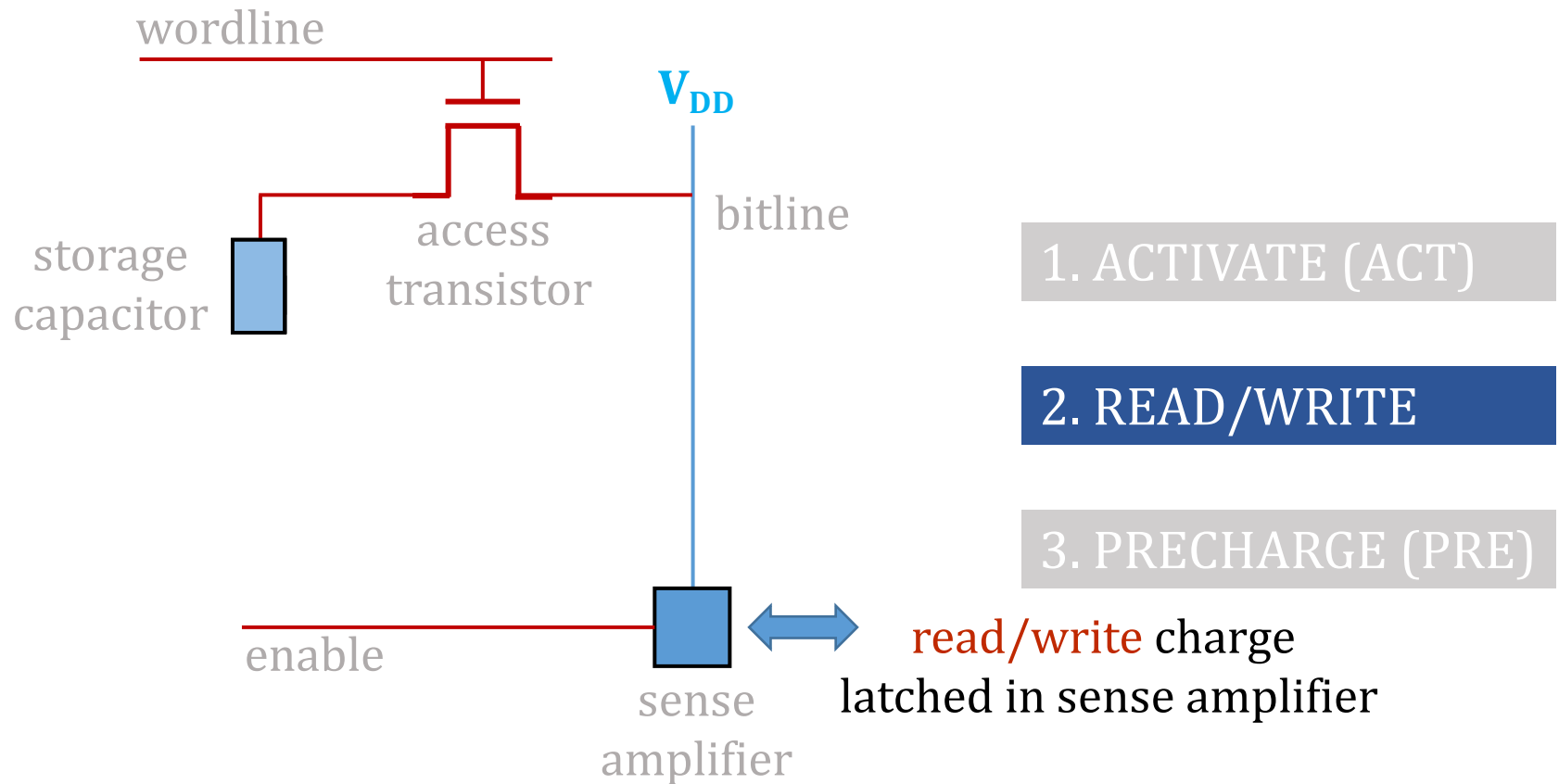


1. ACTIVATE (ACT)

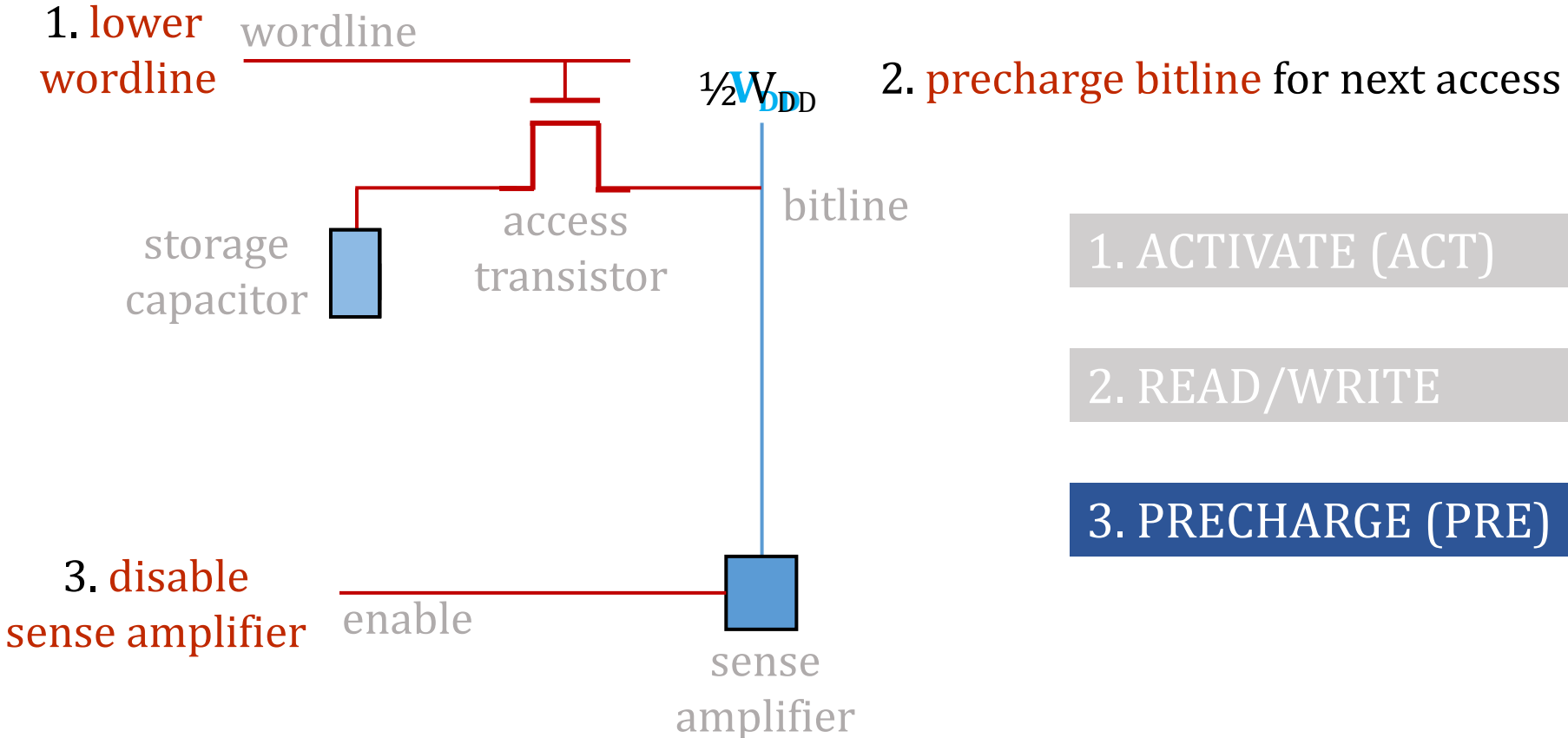
2. READ/WRITE

3. PRECHARGE (PRE)

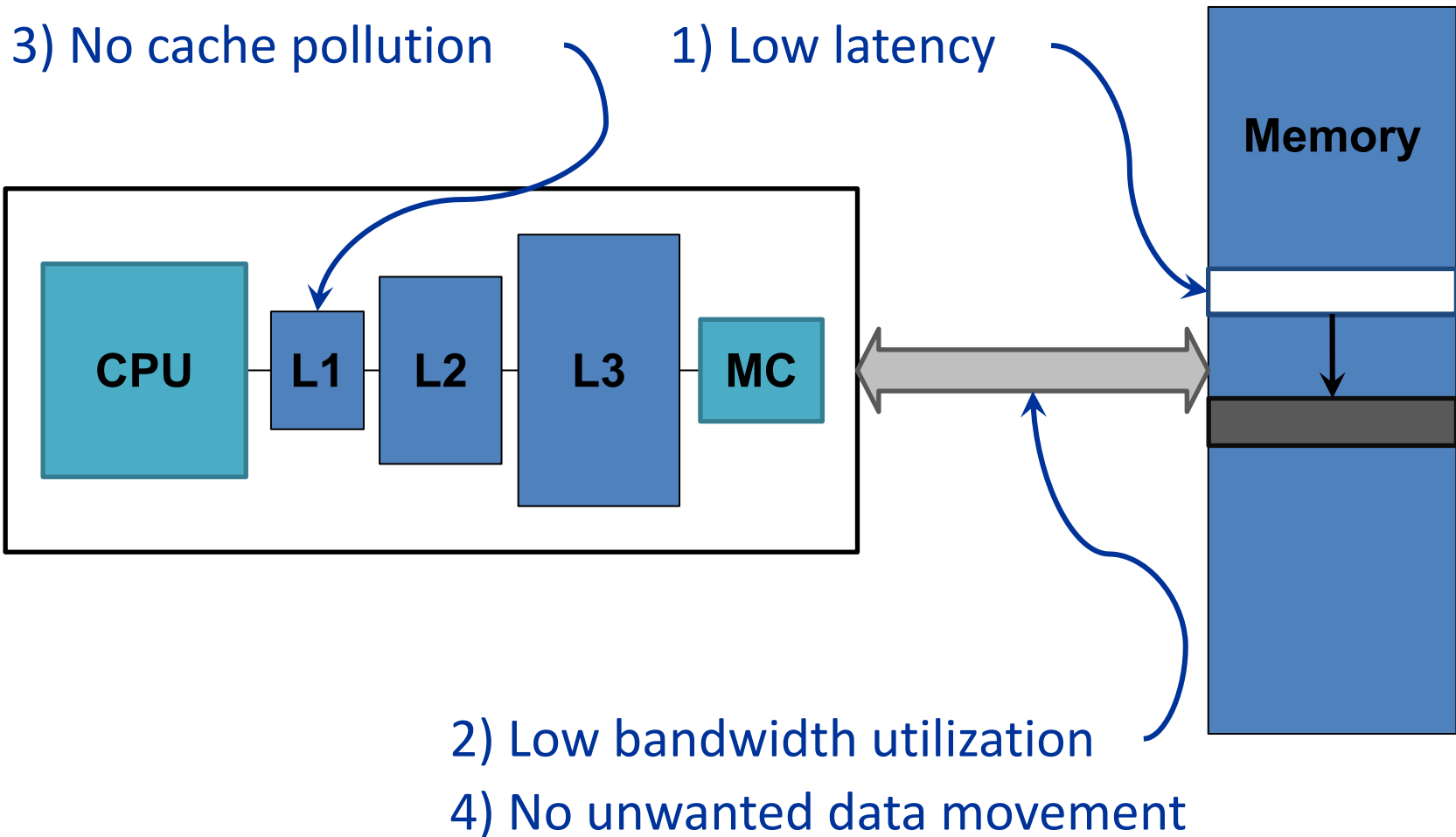
DRAM Cell Operation (2/3)



DRAM Cell Operation (3/3)



Future Systems: In-Memory Copy

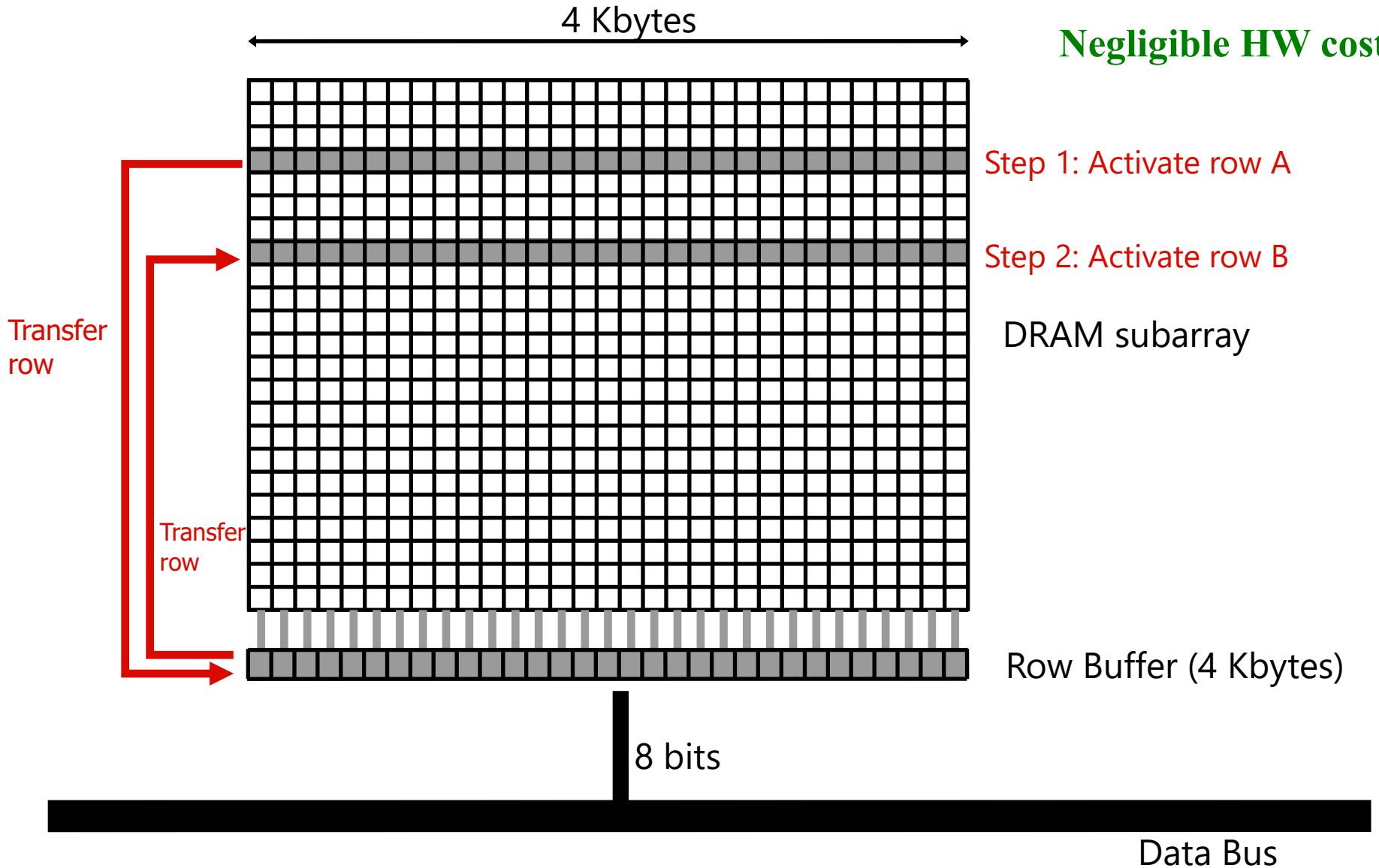


1046ns, 3.6uJ → 90ns, 0.04uJ

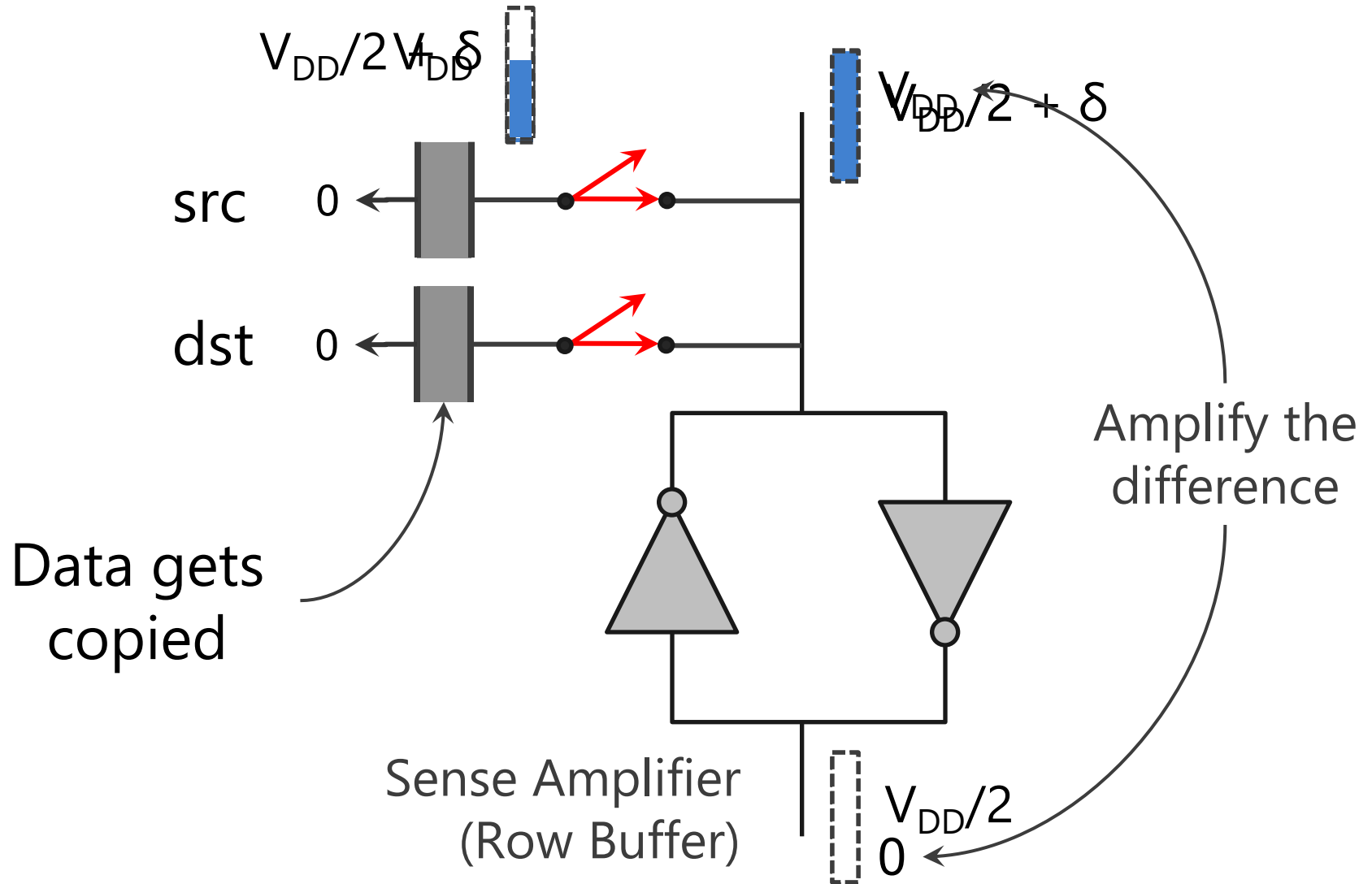
RowClone: In-DRAM Row Copy

Idea: Two consecutive ACTivates

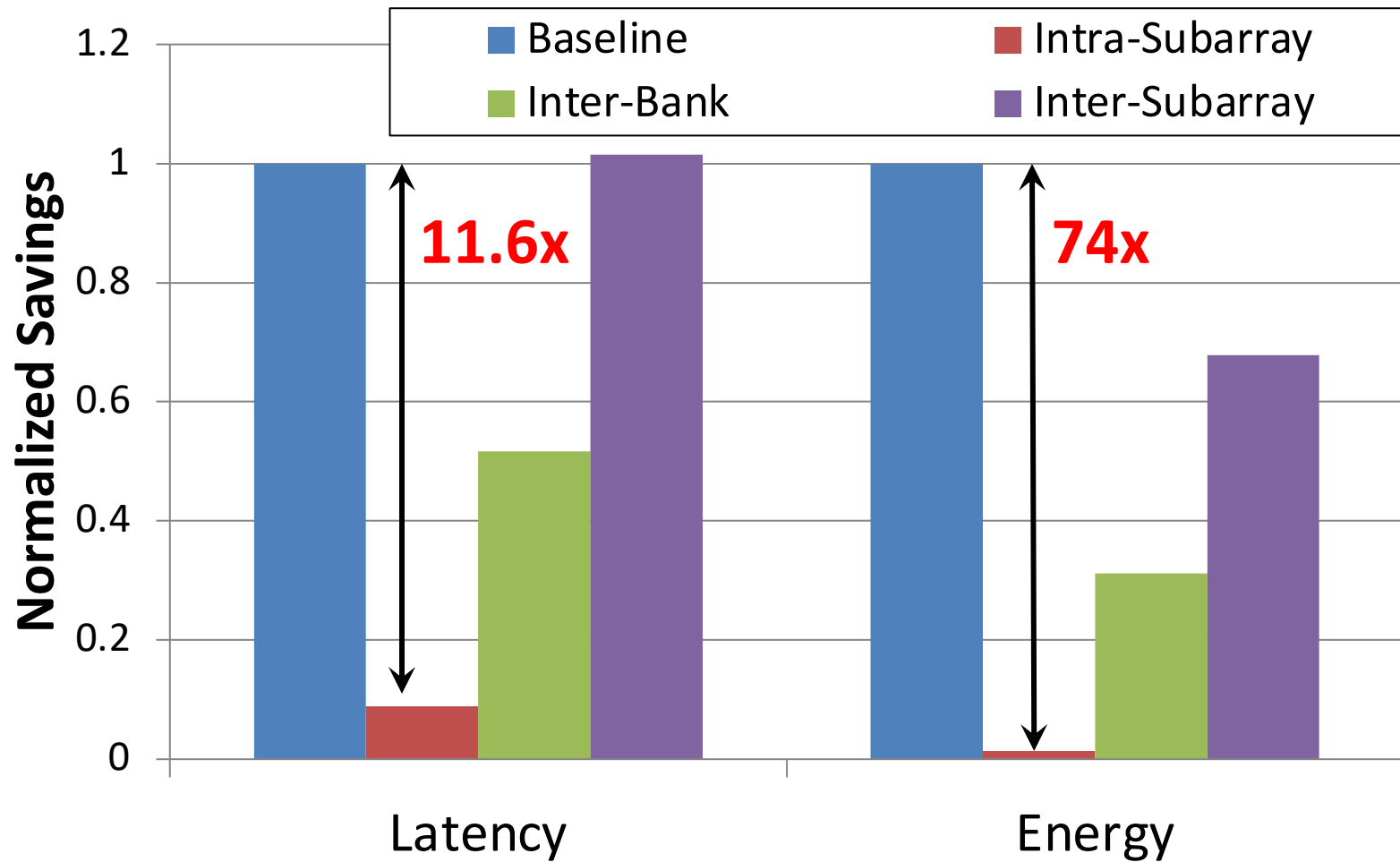
Negligible HW cost



RowClone: Intra-Subarray



RowClone: Latency and Energy Savings



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

More on RowClone

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry,
["RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization"](#)
Proceedings of the [46th International Symposium on Microarchitecture \(MICRO\)](#), Davis, CA, December 2013. [[Slides \(pptx\)](#)] [[pdf](#)] [[Lightning Session Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)]

RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization

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

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

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

Carnegie Mellon University †Intel Pittsburgh

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^{§†}

Juan Gómez Luna[§]

Konstantinos Kanellopoulos[§]

Behzad Salami^{§*}

Hasan Hassan[§]

Oğuz Ergin[†]

Onur Mutlu[§]

§ETH Zürich

†TOBB ETÜ

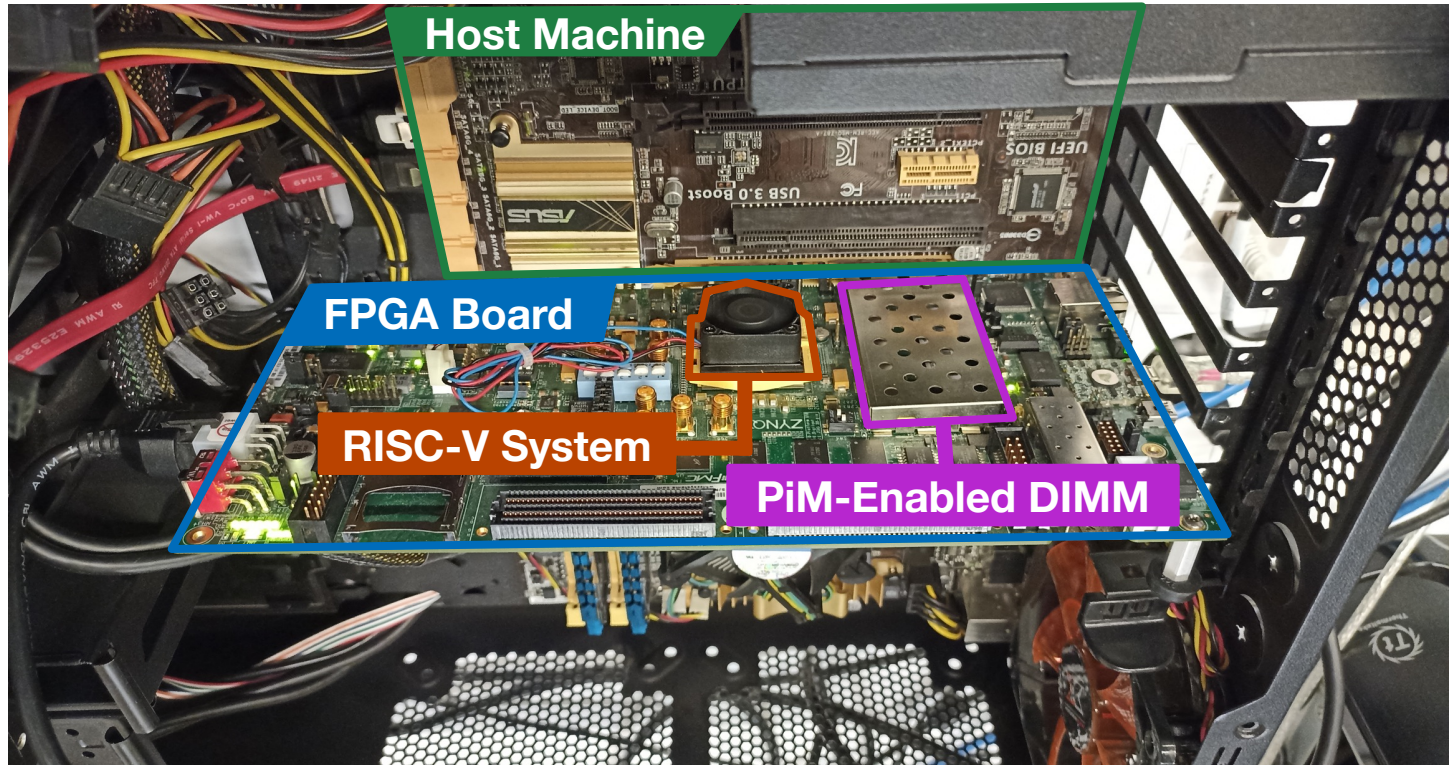
*BSC

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

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

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

Real Processing Using Memory Prototype



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

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

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

Real Processing Using Memory Prototype

☰ README.md ✎

Building a PiDRAM Prototype

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

Rebuilding Steps

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

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

Generating DDR3 Controller IP sources

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

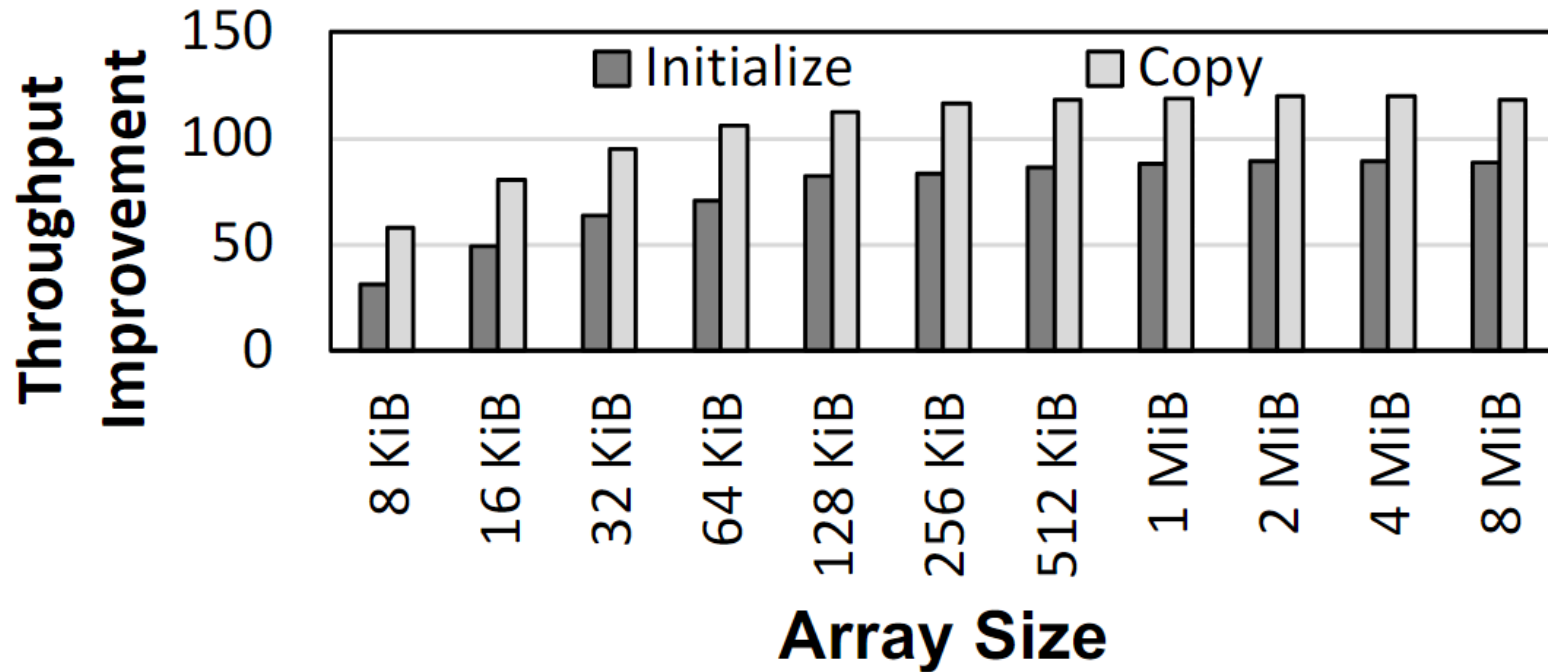
- 1- Open IP Catalog
- 2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

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

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

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

Microbenchmark Copy/Initialization Throughput



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

PiDRAM is Open Source

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

CMU-SAFARI / PiDRAM Public

Edit Pins

Watch 3

Fork 2

Star 21

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master

2 branches 0 tags

Go to file

Add file

Code

About



olgunataberk	Fix small mistake in README	46522cc on Dec 5, 2021	11 commits
controller-hardware	Add files via upload		7 months ago
fpga-zynq	Adds instructions to reproduce two key results		7 months ago
README.md	Fix small mistake in README		7 months ago

PiDRAM is the first flexible end-to-end framework that enables system integration studies and evaluation of real Processing-using-Memory techniques. Prototype on a RISC-V rocket chip system implemented on an FPGA. Described in our preprint: <https://arxiv.org/abs/2111.00082>

README.md



PiDRAM

PiDRAM is the first flexible end-to-end framework that enables system integration studies and evaluation of real Processing-using-Memory (PuM) techniques. PiDRAM, at a high level, comprises a RISC-V system and a custom memory controller that can perform PuM operations in real DDR3 chips. This repository contains all sources required to build PiDRAM and develop its prototype on the Xilinx ZC706 FPGA boards.

Readme

21 stars

3 watching

2 forks

Releases

No releases published

[Create a new release](#)

Extended Version on ArXiv

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

arXiv > cs > arXiv:2111.00082

Search...

All fields

Search

Help | Advanced Search

Computer Science > Hardware Architecture

[Submitted on 29 Oct 2021 (v1), last revised 19 Dec 2021 (this version, v3)]

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

Processing-using-memory (PuM) techniques leverage the analog operation of memory cells to perform computation. Several recent works have demonstrated PuM techniques in off-the-shelf DRAM devices. Since DRAM is the dominant memory technology as main memory in current computing systems, these PuM techniques represent an opportunity for alleviating the data movement bottleneck at very low cost. However, system integration of PuM techniques imposes non-trivial challenges that are yet to be solved. Design space exploration of potential solutions to the PuM integration challenges requires appropriate tools to develop necessary hardware and software components. Unfortunately, current specialized DRAM-testing platforms, or system simulators do not provide the flexibility and/or the holistic system view that is necessary to deal with PuM integration challenges.

We design and develop PiDRAM, the first flexible end-to-end framework that enables system integration studies and evaluation of real PuM techniques. PiDRAM provides software and hardware components to rapidly integrate PuM techniques across the whole system software and hardware stack (e.g., necessary modifications in the operating system, memory controller). We implement PiDRAM on an FPGA-based platform along with an open-source RISC-V system. Using PiDRAM, we implement and evaluate two state-of-the-art PuM techniques: in-DRAM (i) copy and initialization, (ii) true random number generation. Our results show that the in-memory copy and initialization techniques can improve the performance of bulk copy operations by 12.6x and bulk initialization operations by 14.6x on a real system. Implementing the true random number generator requires only 190 lines of Verilog and 74 lines of C code using PiDRAM's software and hardware components.

Comments: 15 pages, 12 figures

Subjects: **Hardware Architecture (cs.AR)**

Cite as: [arXiv:2111.00082](https://arxiv.org/abs/2111.00082) [cs.AR]

(or [arXiv:2111.00082v3](https://arxiv.org/abs/2111.00082v3) [cs.AR] for this version)

<https://doi.org/10.48550/arXiv.2111.00082>

Download:

- [PDF](#)
- [Other formats](#)



Current browse context:

cs.AR

[< prev](#) | [next >](#)

[new](#) | [recent](#) | [2111](#)

Change to browse by:

[cs](#)

References & Citations

- [NASA ADS](#)
- [Google Scholar](#)
- [Semantic Scholar](#)

DBLP - CS Bibliography

[listing](#) | [bibtex](#)

[Juan Gómez-Luna](#)
[Behzad Salami](#)
[Hasan Hassan](#)
[Oguz Ergin](#)
[Onur Mutlu](#)

Export Bibtex Citation

Bookmark



Long Talk + Tutorial on Youtube

https://youtu.be/s_z_S6FYpC8

The video frame shows a slide titled "Alloc_align Example". At the top, it displays two lines of C code: `A = alloc_align(16*1024, 0);` and `B = alloc_align(16*1024, 0);`. Below the code, a diagram illustrates the memory layout. Two horizontal arrows represent "Array A" and "Array B", each labeled "16 KBs". Array A is shown as a sequence of four light blue boxes, with the first box labeled "4 KB". Below the first three boxes, the virtual addresses "0x0000", "0x1000", and "0x2000" are listed. Array B is shown as a sequence of four yellow boxes, with the last box labeled "0x7000". Below the array diagrams, a grid shows "Row 1" and "Row 0" with columns labeled "Bank 0", "Bank 1", and "Bank 2". A "zoom" watermark is visible in the bottom right corner of the video frame.

Processing in Memory Course: Meeting 6: End-to-end Framework for Processing-using-Memory - Fall'21

615 views • Streamed live on 9 Nov 2021 • Project & Seminar, ETH Zürich, Fall 2021 Show more

👍 25 🗑 Dislike ➦ Share ⬇ Download ✂ Clip ⚙ Save ...

 Onur Mutlu Lectures
25.7K subscribers

SUBSCRIBED



Functionally-Complete Real PUM Prototype

- Ismail Emir Yuksel, Yahya Can Tuğrul, Ataberk Olgun, Nisa Bostanci, A. Giray Yaglikci, Geraldo F. Oliveira, Haocong Luo, Juan Gómez-Luna, Mohammad Sadrosadati, and Onur Mutlu, "[Functionally-Complete Boolean Logic in DRAM: An Experimental Characterization and Analysis of Real DRAM Chips,](#)" *Proceedings of the [30th International Symposium on High-Performance Computer Architecture](#) (**HPCA**), Edinburgh, Scotland, March 2024.*

2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)

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

Improvements on RowClone

RowClone Extensions and Follow-Up Work

- Can we do **faster inter-subarray copy**?
 - Yes, **see LISA [Chang et al., HPCA 2016]**
- Can we enable **data movement at smaller granularities within a bank**?
 - Yes, **see FIGARO [Wang et al., MICRO 2020]**
- Can we do **better inter-bank copy**?
 - Yes, **see Network-on-Memory [CAL 2020]**
- Can similar ideas and DRAM properties be used to perform **computation on data**?
 - Yes, **see Ambit [Seshadri et al., CAL 2015, MICRO 2017]**

LISA: Increasing Connectivity in DRAM

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,

"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"

Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Source Code](#)]

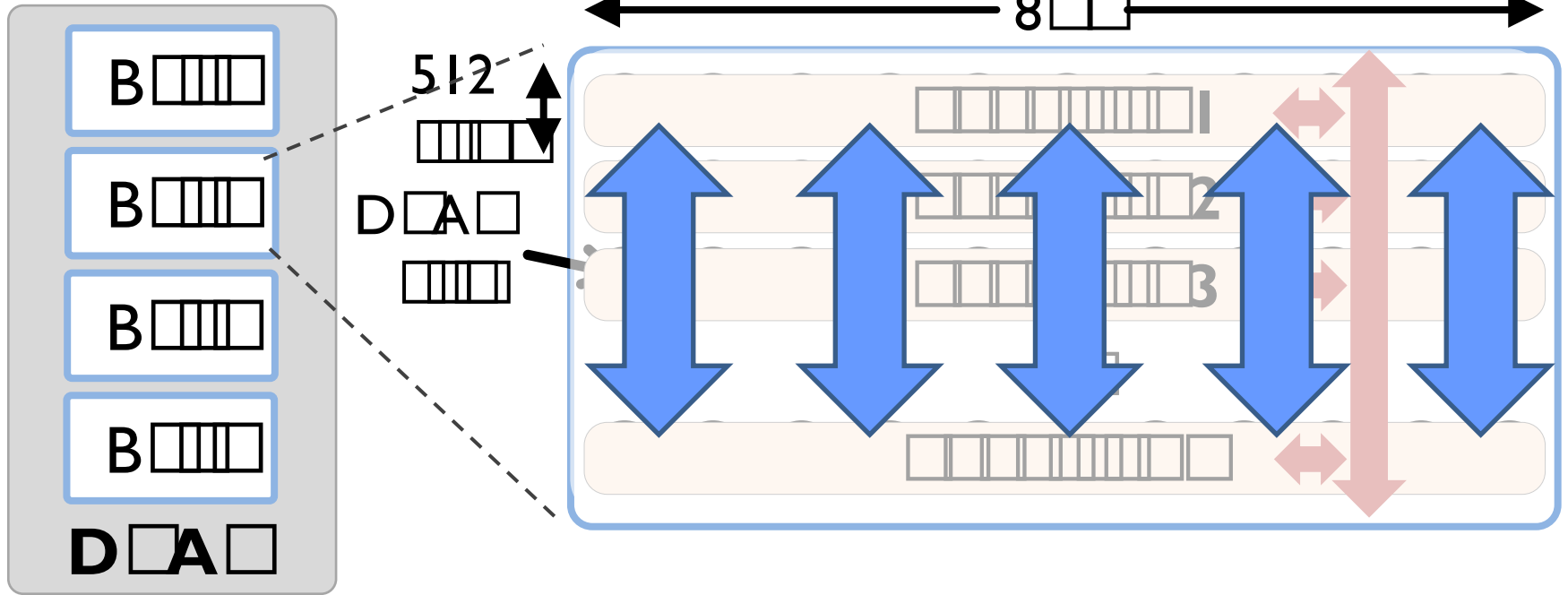
Low-Cost Inter-Linked Subarrays (LISA):

Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang[†], Prashant J. Nair^{*}, Donghyuk Lee[†], Saugata Ghose[†], Moinuddin K. Qureshi^{*}, and Onur Mutlu[†]

[†]*Carnegie Mellon University* ^{*}*Georgia Institute of Technology*

□ □ □ □ □ □ □ □ **D** □ □ □ □ □ □ □ □ □ □ □ □ □ □ **D** □ □ **A** □ ?



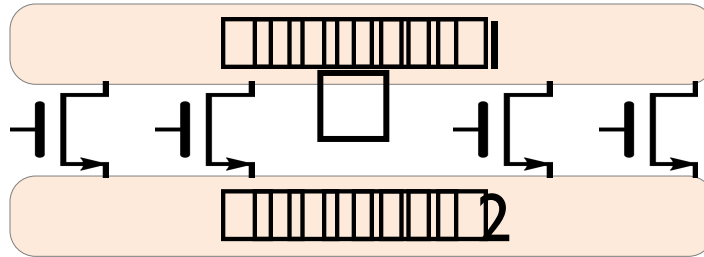
Goal: Provide a new substrate to enable wide connectivity between subarrays



- (A)

– F

– : 0.8% D A



- →

F : C 1.363ms → 0.148ms (9.2x)

→ 66% , -55% D A

I : H 48.7ns → 21.5ns (2.2x)

→ 5%

F : 13.1ns → 5.0ns (2.6x)

→ 8%

More on LISA

- Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,

"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"

Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Source Code](#)]

Low-Cost Inter-Linked Subarrays (LISA):

Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang[†], Prashant J. Nair^{*}, Donghyuk Lee[†], Saugata Ghose[†], Moinuddin K. Qureshi^{*}, and Onur Mutlu[†]

[†]*Carnegie Mellon University* ^{*}*Georgia Institute of Technology*

FIGARO: Fine-Grained In-DRAM Copy

- Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu,
"FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching"
Proceedings of the 53rd International Symposium on Microarchitecture (MICRO), Virtual, October 2020.

FIGARO: Improving System Performance via Fine-Grained In-DRAM Data Relocation and Caching

Yaohua Wang^{*} Lois Orosa[†] Xiangjun Peng^{⊙*} Yang Guo^{*} Saugata Ghose^{◇‡} Minesh Patel[†]
Jeremie S. Kim[†] Juan Gómez Luna[†] Mohammad Sadrosadati[§] Nika Mansouri Ghiasi[†] Onur Mutlu^{†‡}

^{*}National University of Defense Technology [†]ETH Zürich [⊙]Chinese University of Hong Kong

[◇]University of Illinois at Urbana–Champaign [‡]Carnegie Mellon University [§]Institute of Research in Fundamental Sciences

Network-On-Memory: Fast Inter-Bank Copy

- Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Modarressi, Rachata Ausavarungnirun, Mohammad Sadrosadati, Onur Mutlu, and Masoud Daneshtalab,
"NoM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories"
IEEE Computer Architecture Letters (**CAL**), to appear in 2020.

NOM: NETWORK-ON-MEMORY FOR INTER-BANK DATA TRANSFER IN HIGHLY-BANKED MEMORIES

Seyyed Hossein SeyyedAghaei Rezaei¹
Mohammad Sadrosadati³

Mehdi Modarressi^{1,3}
Onur Mutlu⁴

Rachata Ausavarungnirun²
Masoud Daneshtalab⁵

¹University of Tehran

²King Mongkut's University of Technology North Bangkok

³Institute for Research in Fundamental Sciences

⁴ETH Zürich

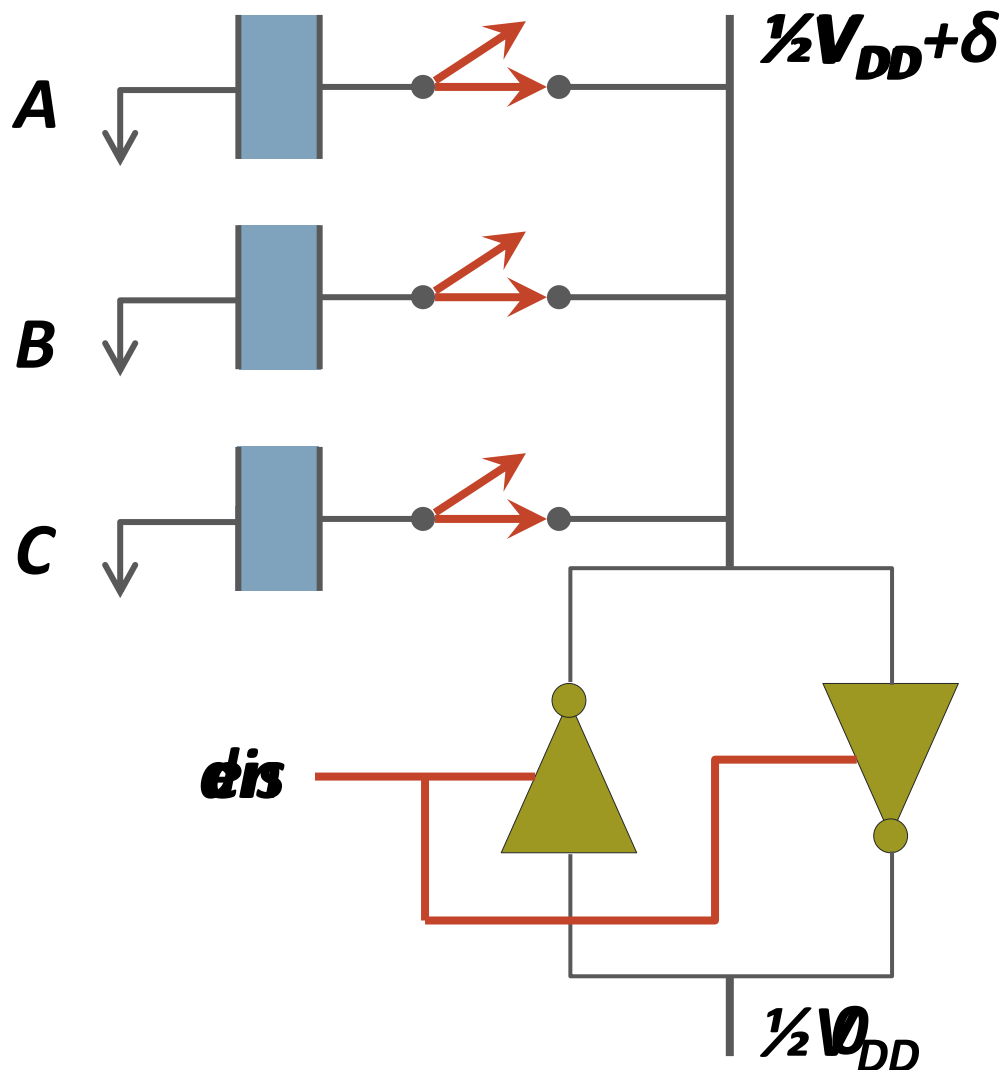
⁵Mälardalens University

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

- 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
 $AB + BC + AC$

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

More on In-DRAM Bulk AND/OR

- Vivek Seshadri, Kevin Hsieh, Amirali Boroumand, Donghyuk Lee, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
"Fast Bulk Bitwise AND and OR in DRAM"
IEEE Computer Architecture Letters (***CAL***), April 2015.

Fast Bulk Bitwise AND and OR in DRAM

Vivek Seshadri*, Kevin Hsieh*, Amirali Boroumand*, Donghyuk Lee*,
Michael A. Kozuch†, Onur Mutlu*, Phillip B. Gibbons†, Todd C. Mowry*

*Carnegie Mellon University †Intel Pittsburgh

In-DRAM NOT: Dual Contact Cell

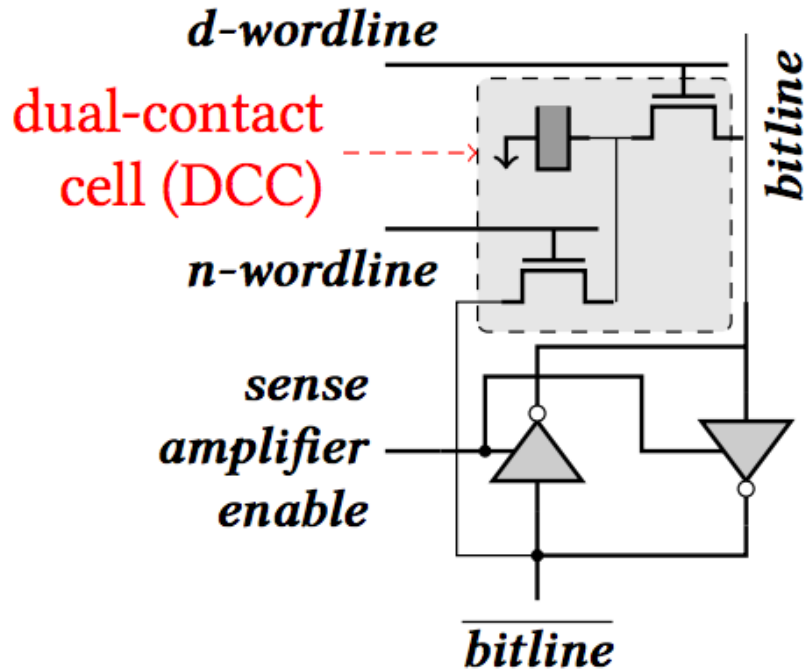


Figure 5: A dual-contact cell connected to both ends of a sense amplifier

Idea:
Feed the
negated value
in the sense amplifier
into a special row

Performance: In-DRAM Bitwise Operations

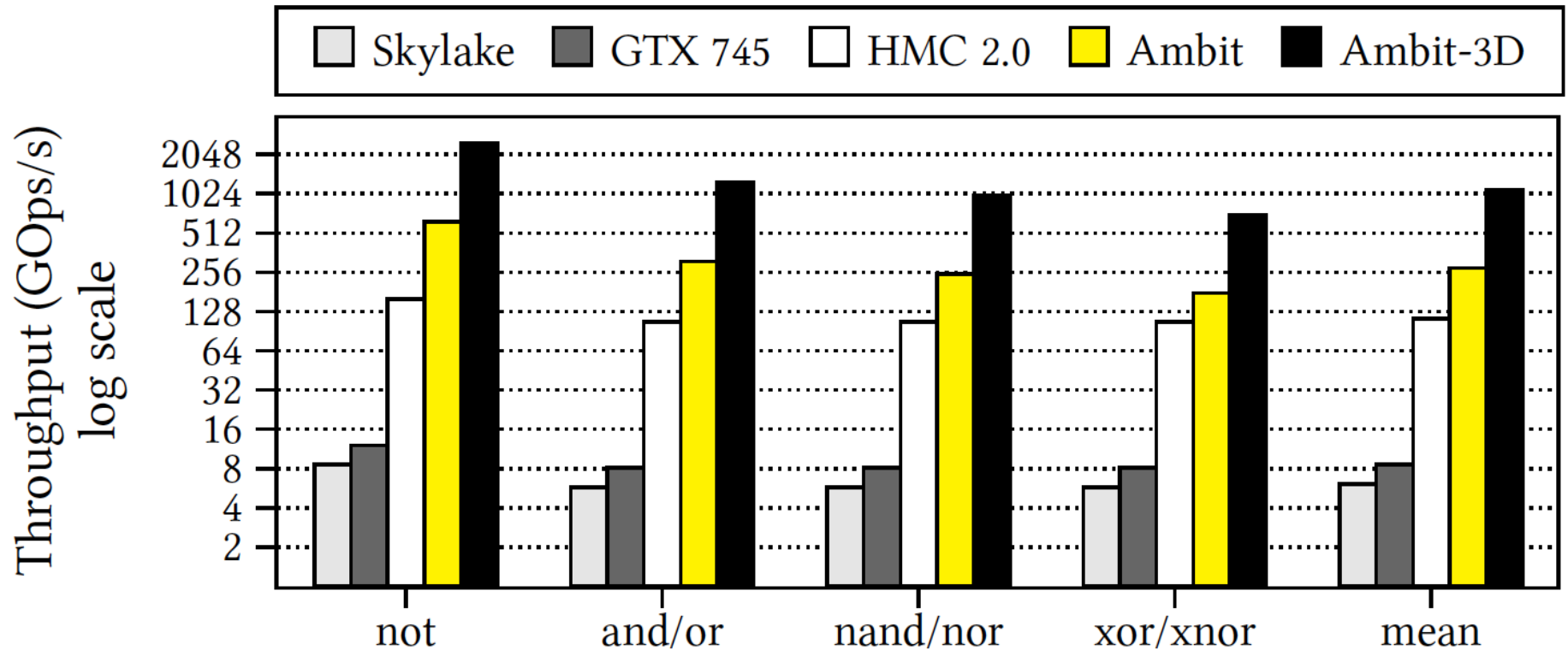


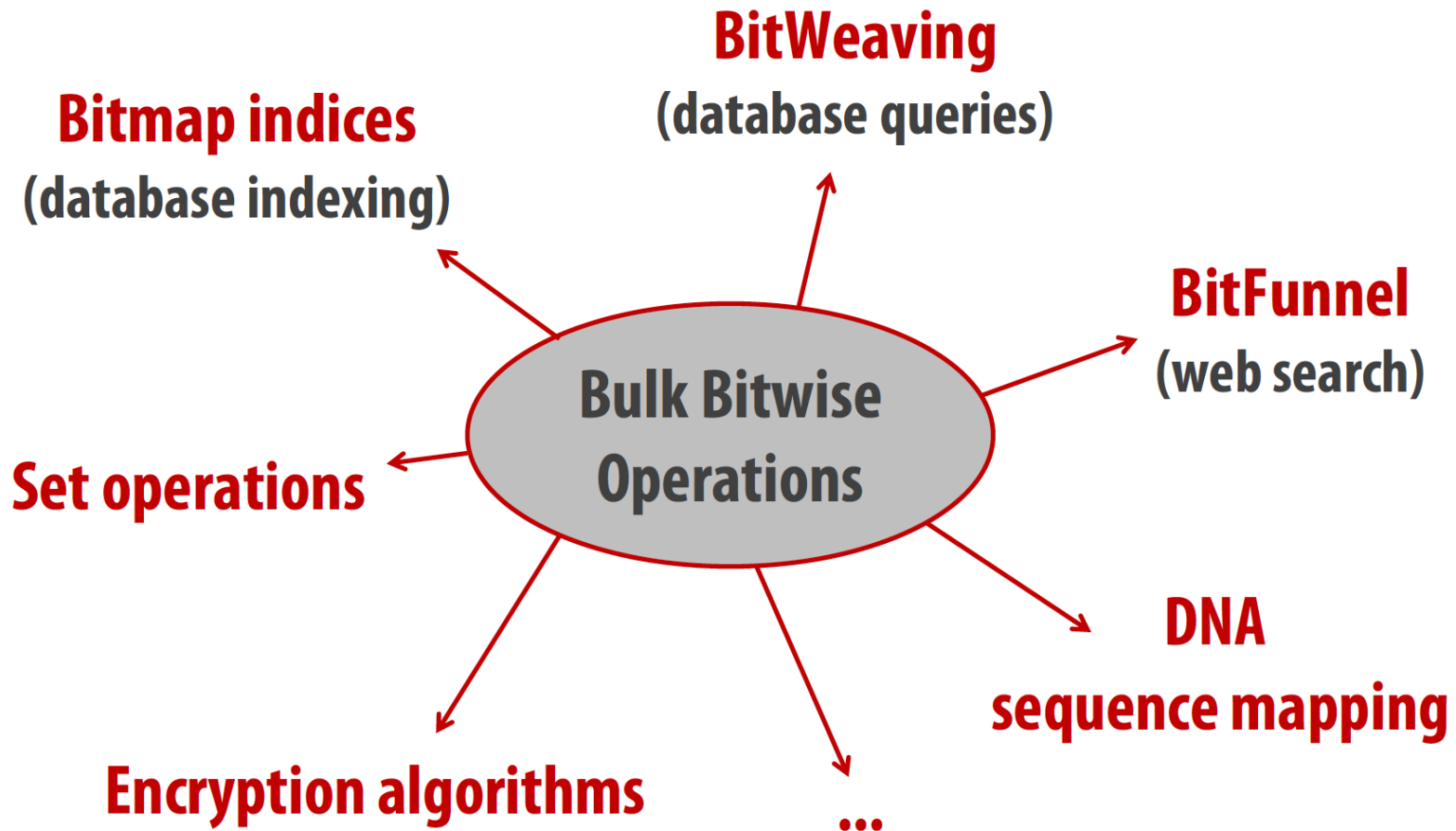
Figure 9: Throughput of bitwise operations on various systems.

Energy of In-DRAM Bitwise Operations

	Design	not	and/or	nand/nor	xor/xnor
DRAM &	DDR3	93.7	137.9	137.9	137.9
Channel Energy	Ambit	1.6	3.2	4.0	5.5
(nJ/KB)	(↓)	59.5X	43.9X	35.1X	25.1X

Table 3: Energy of bitwise operations. (↓) indicates energy reduction of Ambit over the traditional DDR3-based design.

Bulk Bitwise Operations in Workloads



In-DRAM Acceleration of Database Queries

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

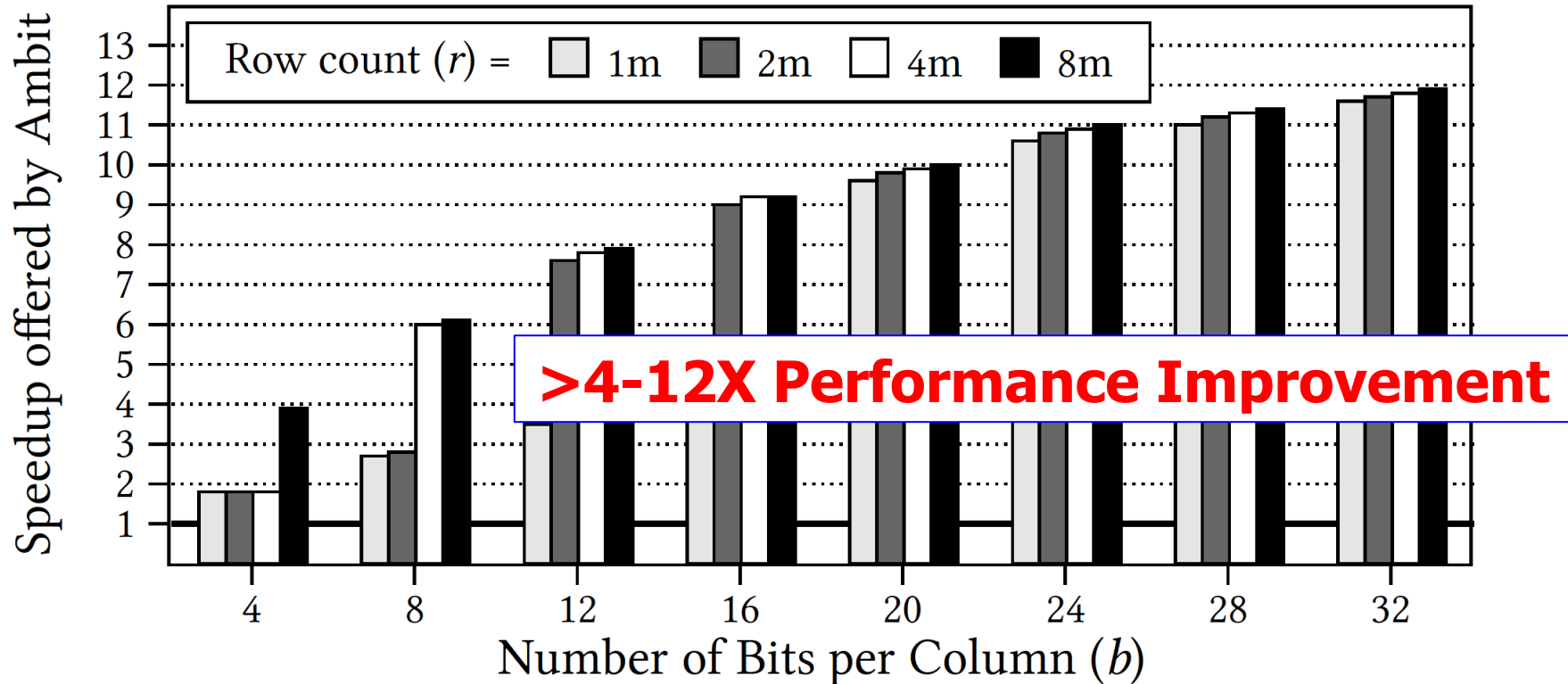


Figure 11: Speedup offered by Ambit over baseline CPU with SIMD for BitWeaving

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

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 Hassan⁴ Amirali Boroumand⁵
Jeremie Kim^{4,5} Michael A. Kozuch³ Onur Mutlu^{4,5} Phillip B. Gibbons⁵ Todd C. Mowry⁵

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

In-DRAM Bulk Bitwise Execution

- Vivek Seshadri and Onur Mutlu,
"In-DRAM Bulk Bitwise Execution Engine"
Invited Book Chapter in Advances in Computers, to appear
in 2020.
[[Preliminary arXiv version](#)]

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

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

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

*Nastaran Hajinazar^{1,2}

Nika Mansouri Ghiasi¹

*Geraldo F. Oliveira¹

Minesh Patel¹

Juan Gómez-Luna¹

Sven Gregorio¹

Mohammed Alser¹

Onur Mutlu¹

João Dinis Ferreira¹

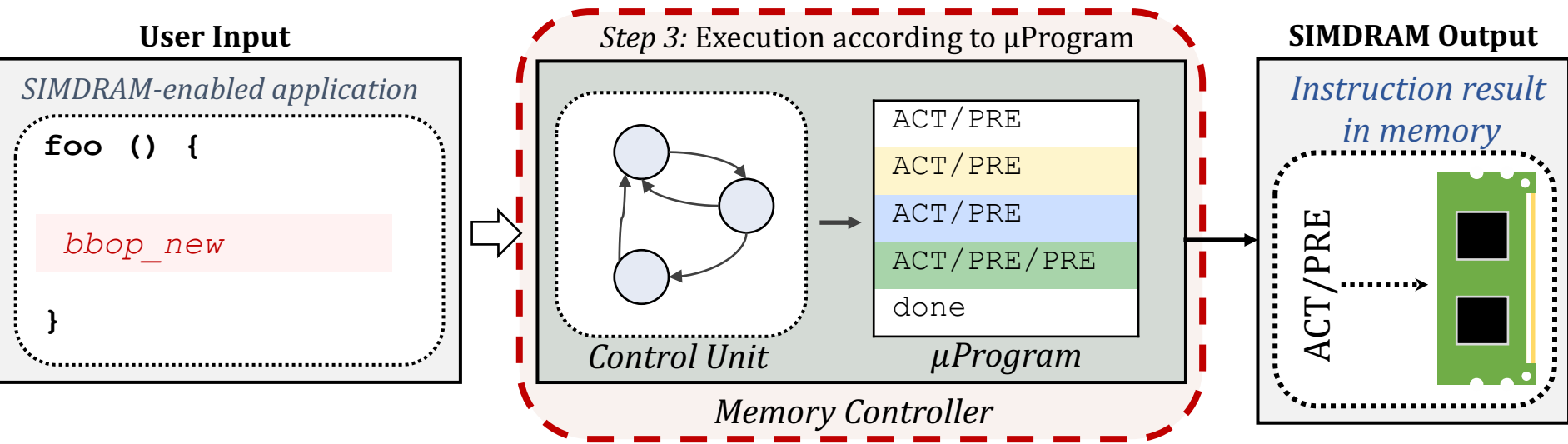
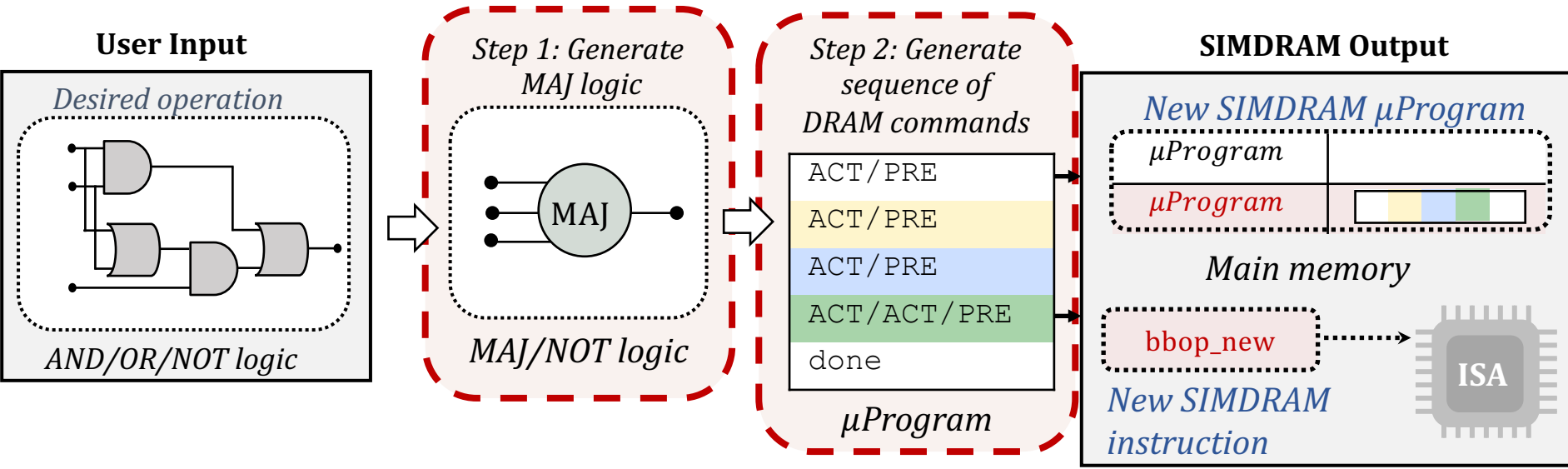
Saugata Ghose³

¹ETH Zürich

²Simon Fraser University

³University of Illinois at Urbana–Champaign

SIMDRAM Framework: Overview



More on SIMD RAM

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

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

*Nastaran Hajinazar^{1,2}

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 Physical Unclonable Functions

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

The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions

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

Jeremie S. Kim^{†§}

Minesh Patel[§]

Hasan Hassan[§]

Onur Mutlu^{§†}

[†]Carnegie Mellon University

[§]ETH Zürich

In-DRAM True Random Number Generation

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, "[D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput](#)"

Proceedings of the [25th International Symposium on High-Performance Computer Architecture \(HPCA\)](#), Washington, DC, USA, February 2019.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Full Talk Video](#) (21 minutes)]

[[Full Talk Lecture Video](#) (27 minutes)]

Top Picks Honorable Mention by IEEE Micro.

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

Jeremie S. Kim^{‡§}

Minesh Patel[§]

Hasan Hassan[§]

Lois Orosa[§]

Onur Mutlu^{§‡}

[‡]Carnegie Mellon University

[§]ETH Zürich

In-DRAM True Random Number Generation

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

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

Ataberk Olgun^{§†}

Minesh Patel[§]

A. Giray Yağlıkçı[§]

Haocong Luo[§]

Jeremie S. Kim[§]

F. Nisa Bostanci^{§†}

Nandita Vijaykumar^{§⊙}

Oğuz Ergin[†]

Onur Mutlu[§]

[§]*ETH Zürich*

[†]*TOBB University of Economics and Technology*

[⊙]*University of Toronto*

In-DRAM True Random Number Generation

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

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

F. Nisa Bostanci^{†§} 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-DRAM Lookup-Table Based Execution

João Dinis Ferreira, Gabriel Falcao, Juan Gómez-Luna, Mohammed Alser, Lois Orosa, Mohammad Sadrosadati, Jeremie S. Kim, Geraldo F. Oliveira, Taha Shahroodi, Anant Nori, and Onur Mutlu,

"pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables"

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

[[Slides \(pptx\)](#)] ([pdf](#))]

[[Longer Lecture Slides \(pptx\)](#)] ([pdf](#))]

[[Lecture Video](#) (26 minutes)]

[[arXiv version](#)]

[[Source Code \(Officially Artifact Evaluated with All Badges\)](#)]

Officially artifact evaluated as available, reusable and reproducible.



pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables

João Dinis Ferreira[§]

Gabriel Falcao[†]

Juan Gómez-Luna[§]

Mohammed Alser[§]

Lois Orosa^{§∇}

Mohammad Sadrosadati[§]

Jeremie S. Kim[§]

Geraldo F. Oliveira[§]

Taha Shahroodi[‡]

Anant Nori^{*}

Onur Mutlu[§]

[§]ETH Zürich

[†]IT, University of Coimbra

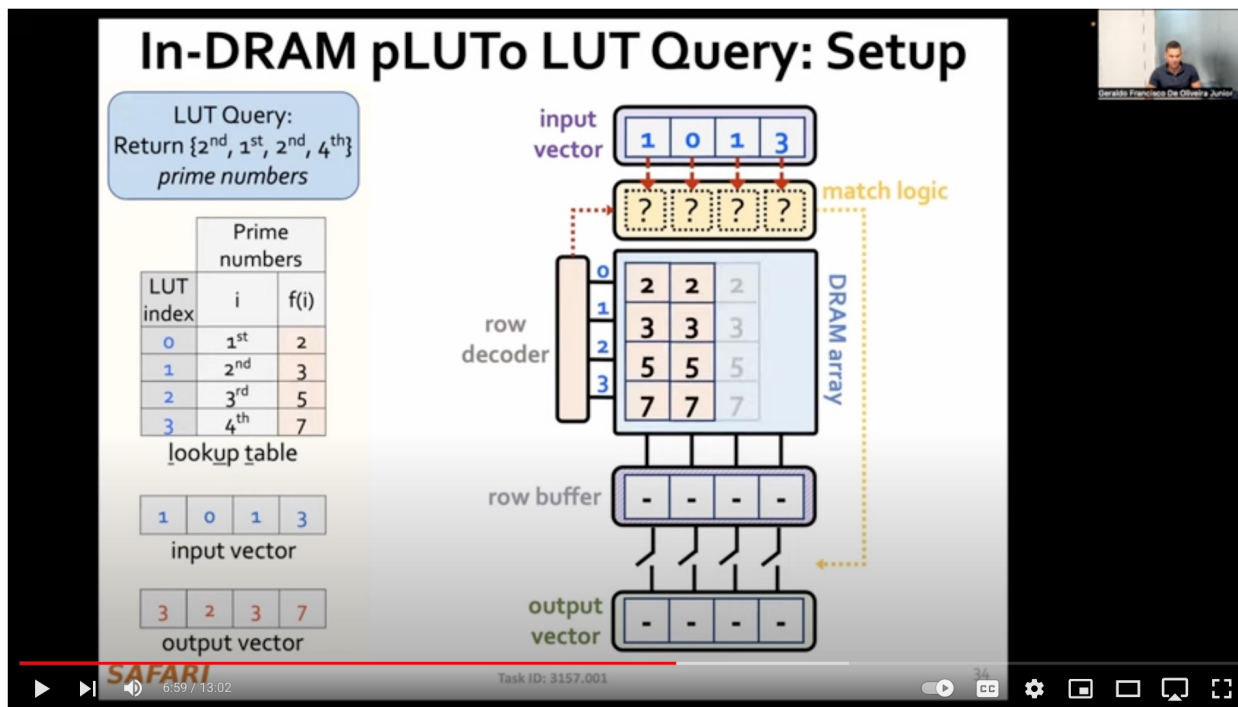
[∇]Galicia Supercomputing Center

[‡]TU Delft

^{*}Intel

SRC TECHCON Presentation

- Geraldo F. Oliveira
 - pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables
 - <https://arxiv.org/pdf/2104.07699.pdf>



pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables, SRC TECHCON 2023

Onur Mutlu Lectures
35.5K subscribers

Subscribed

17 | Share | Clip | Save

321 views 9 days ago
pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables
Speaker: Geraldo F. Oliveira ...more

In-Flash Bulk Bitwise Execution

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

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

Jisung Park^{§∇} Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsook 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

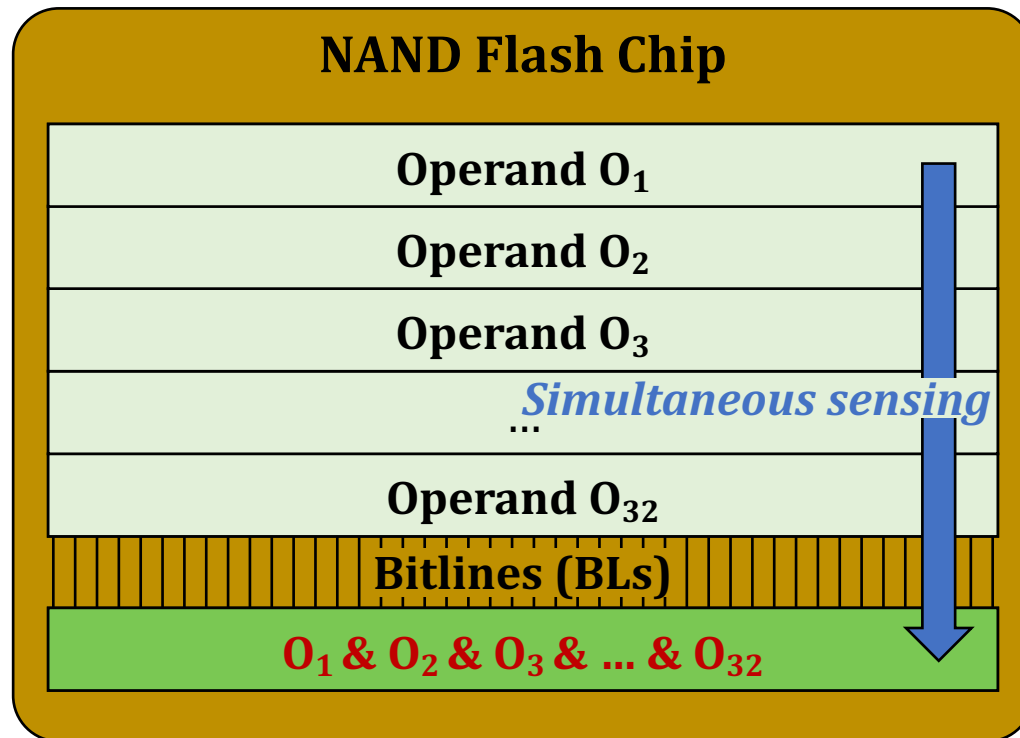


Low-cost & requires no changes to flash cell arrays

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



Pinatubo: RowClone and Bitwise Ops in PCM

Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories

Shuangchen Li^{1*}, Cong Xu², Qiaosha Zou^{1,5}, Jishen Zhao³, Yu Lu⁴, and Yuan Xie¹

University of California, Santa Barbara¹, Hewlett Packard Labs²

University of California, Santa Cruz³, Qualcomm Inc.⁴, Huawei Technologies Inc.⁵

{shuangchenli, yuanxie}@ece.ucsb.edu¹

Other Readings on Processing using NVM

- Shafiee+, “ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars”, ISCA 2016.
- Chi+, “PRIME: A Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory”, ISCA 2016.
- Prezioso+, “Training and Operation of an Integrated Neuromorphic Network based on Metal-Oxide Memristors”, Nature 2015
- Ambrogio+, “Equivalent-accuracy accelerated neural-network training using analogue memory”, Nature 2018.

Processing in Memory: Adoption Challenges

1. Processing near Memory
2. Processing using Memory

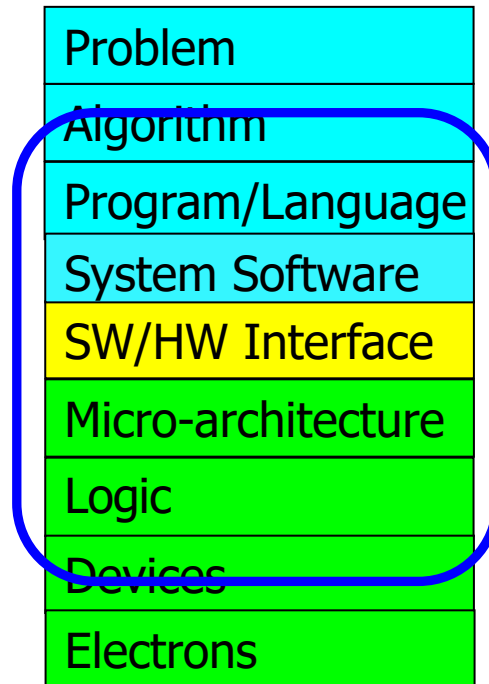
How to Enable Adoption of Processing in Memory

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?

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi, **"PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"** *Proceedings of the 42nd International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2015. [[Slides \(pdf\)](#)] [[Lightning Session Slides \(pdf\)](#)]

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

Junwhan Ahn Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi

junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University

[†]Carnegie Mellon University

Adoption: How to Ease Programmability? (I)

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

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

Geraldo F. Oliveira*

Alain Kohli*

David Novo‡

Juan Gómez-Luna*

Onur Mutlu*

**ETH Zürich*

‡*LIRMM, Univ. Montpellier, CNRS*

Adoption: How to Ease Programmability? (II)

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

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

Jinfan Chen¹ Juan Gómez-Luna¹ Izzat El Hajj² Yuxin Guo¹ Onur Mutlu¹
¹ETH Zürich ²American University of Beirut

Adoption: How to Ease Programmability? (III)

- Geraldo F. Oliveira, Ataberk Olgun, A. Giray Yaglicki, Nisa Bostanci, Juan Gómez-Luna, Saugata Ghose, and Onur Mutlu, **"MIMDRAM: An End-to-End Processing-using-DRAM System for Energy-Efficient and Programmer-Transparent MIMD Computing,"** *Proceedings of the 30th International Symposium on High-Performance Computer Architecture (HPCA)*, Edinburgh, Scotland, March 2024.

MIMDRAM: An End-to-End Processing-using-DRAM System
for Energy-Efficient and Programmer-Transparent MIMD Computing

Geraldo F. Oliveira[†] Ataberk Olgun[†] A. Giray Yaglicki[†] Nisa Bostanci[†]
Juan Gómez-Luna[†] Saugata Ghose[‡] Onur Mutlu[†]

[†] ETH Zürich

[‡] University of Illinois at Urbana-Champaign

Adoption: How to Maintain Coherence? (I)

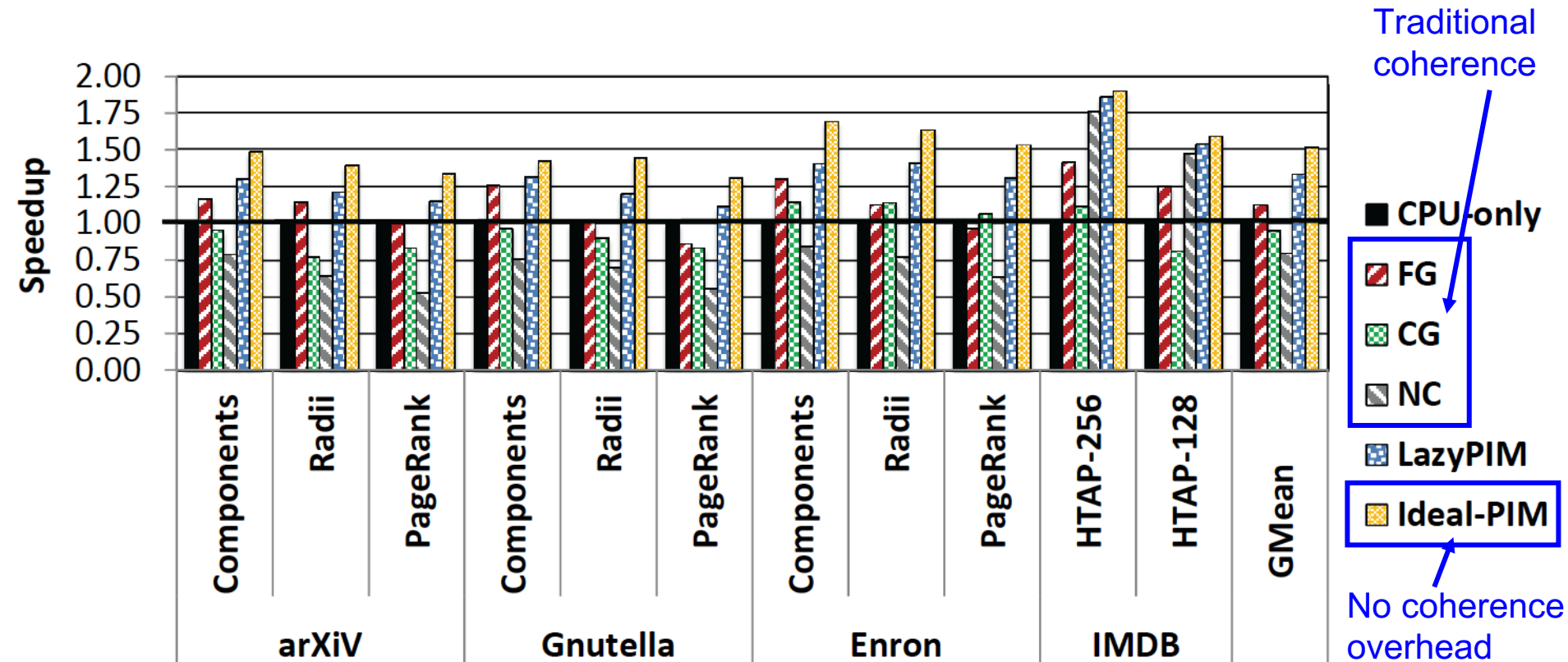
- Amirali Boroumand, Saugata Ghose, Minesh Patel, Hasan Hassan, Brandon Lucia, Kevin Hsieh, Krishna T. Malladi, Hongzhong Zheng, and Onur Mutlu,
"LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory"
IEEE Computer Architecture Letters (**CAL**), June 2016.

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



Adoption: How to Maintain Coherence? (II)

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

"CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators"

Proceedings of the 46th International Symposium on Computer Architecture (ISCA), Phoenix, AZ, USA, June 2019.

CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators

Amirali Boroumand[†]

Saugata Ghose[†]

Minesh Patel^{*}

Hasan Hassan^{*}

Brandon Lucia[†]

Rachata Ausavarungnirun^{†‡}

Kevin Hsieh[†]

Nastaran Hajinazar^{◇†}

Krishna T. Malladi[§]

Hongzhong Zheng[§]

Onur Mutlu^{*†}

[†]Carnegie Mellon University

^{*}ETH Zürich

[‡]KMUTNB

[◇]Simon Fraser University

[§]Samsung Semiconductor, Inc.

Adoption: How to Support Synchronization?

- Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, Onur Mutlu, [**"SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"**](#)
Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA), Virtual, February-March 2021.
[[Slides \(pptx\)](#)] [[pdf](#)]
[[Short Talk Slides \(pptx\)](#)] [[pdf](#)]
[[Talk Video](#) (21 minutes)]
[[Short Talk Video](#) (7 minutes)]

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

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

Adoption: How to Support Virtual Memory?

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,
["Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation"](#)
Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

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

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

Adoption: Code and Data Mapping

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler, **"Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems"**

Proceedings of the 43rd International Symposium on Computer Architecture (ISCA), Seoul, South Korea, June 2016.

[[Slides \(pptx\)](#) ([pdf](#))]

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

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

Kevin Hsieh[‡] Eiman Ebrahimi[†] Gwangsun Kim* Niladrish Chatterjee[†] Mike O'Connor[†]
Nandita Vijaykumar[‡] Onur Mutlu^{§‡} Stephen W. Keckler[†]

[‡]Carnegie Mellon University [†]NVIDIA ^{*}KAIST [§]ETH Zürich

DAMOV Analysis Methodology & Workloads

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

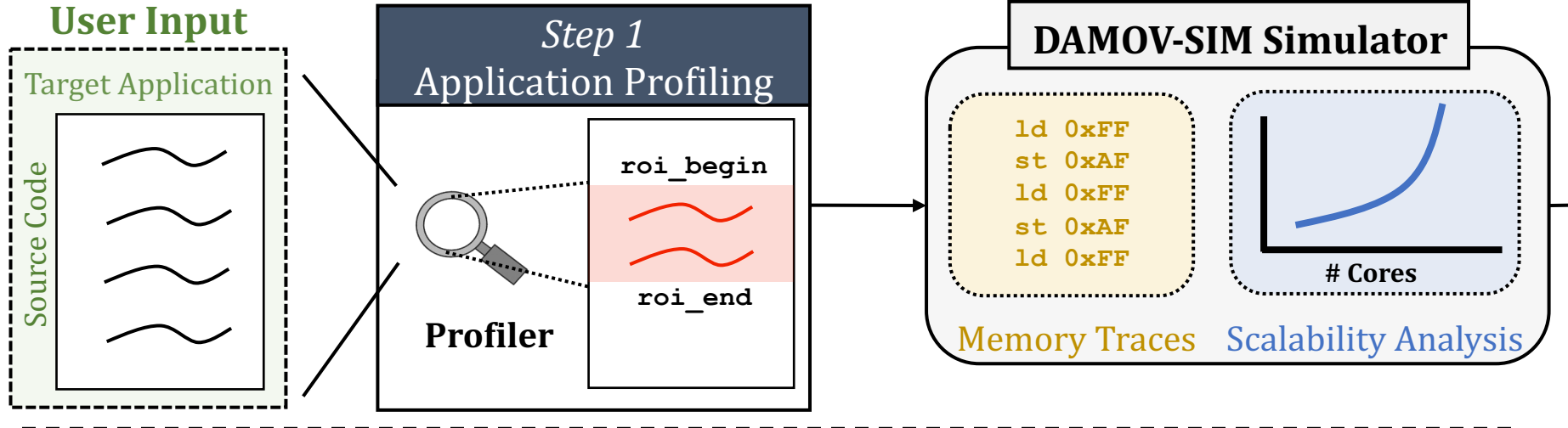
MOHAMMAD SADROSADATI, Institute for Research in Fundamental Sciences (IPM), Iran & ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

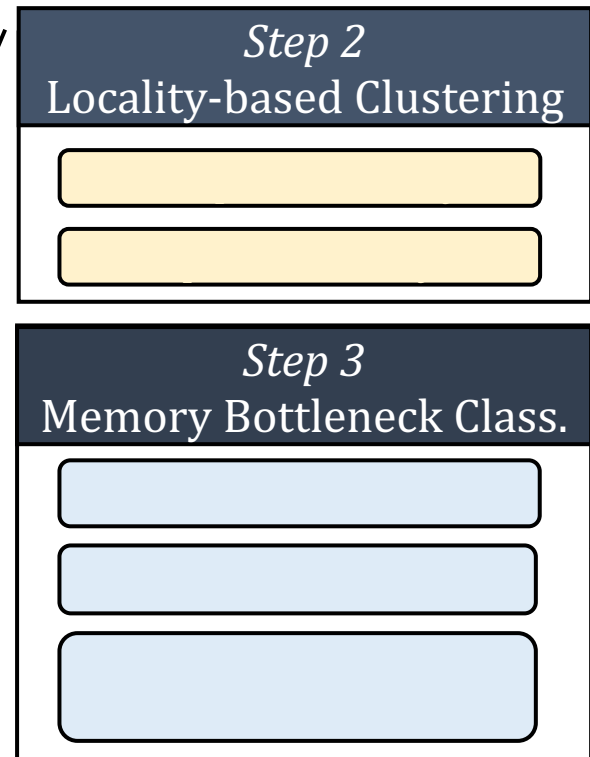
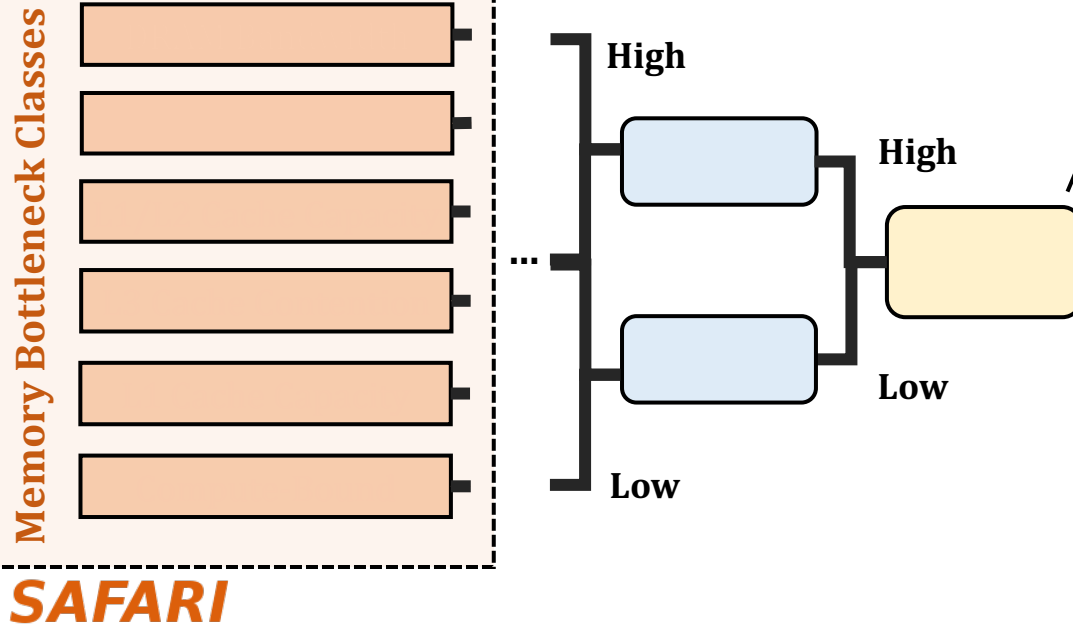
Data movement between the CPU and main memory is a first-order obstacle against improving performance, scalability, and energy efficiency in modern systems. Computer systems employ a range of techniques to reduce overheads tied to data movement, spanning from traditional mechanisms (e.g., deep multi-level 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>.

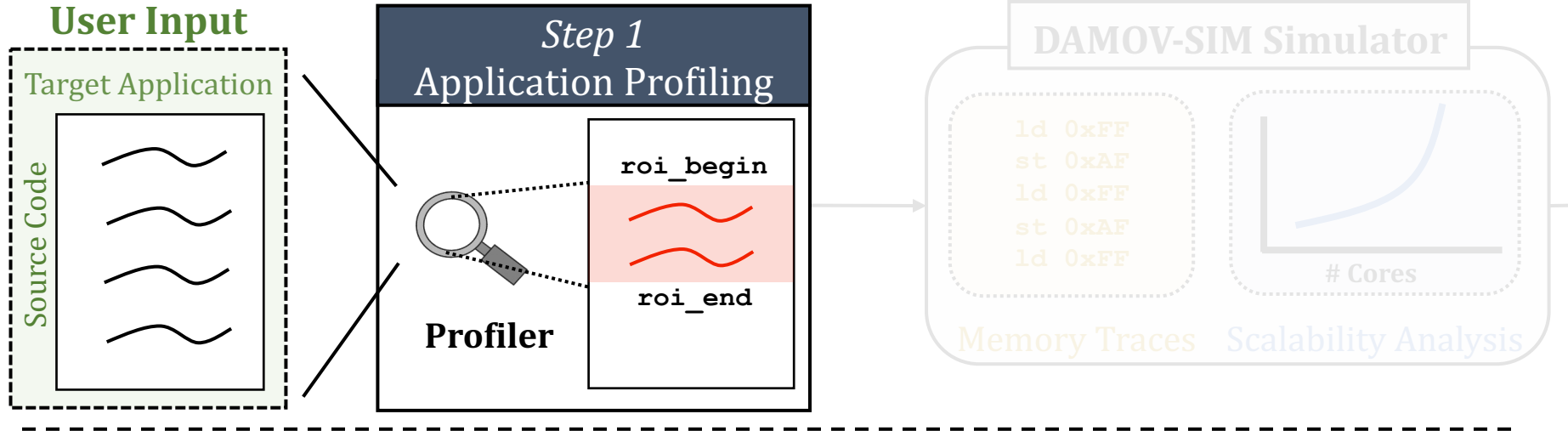
Methodology Overview



Methodology Output



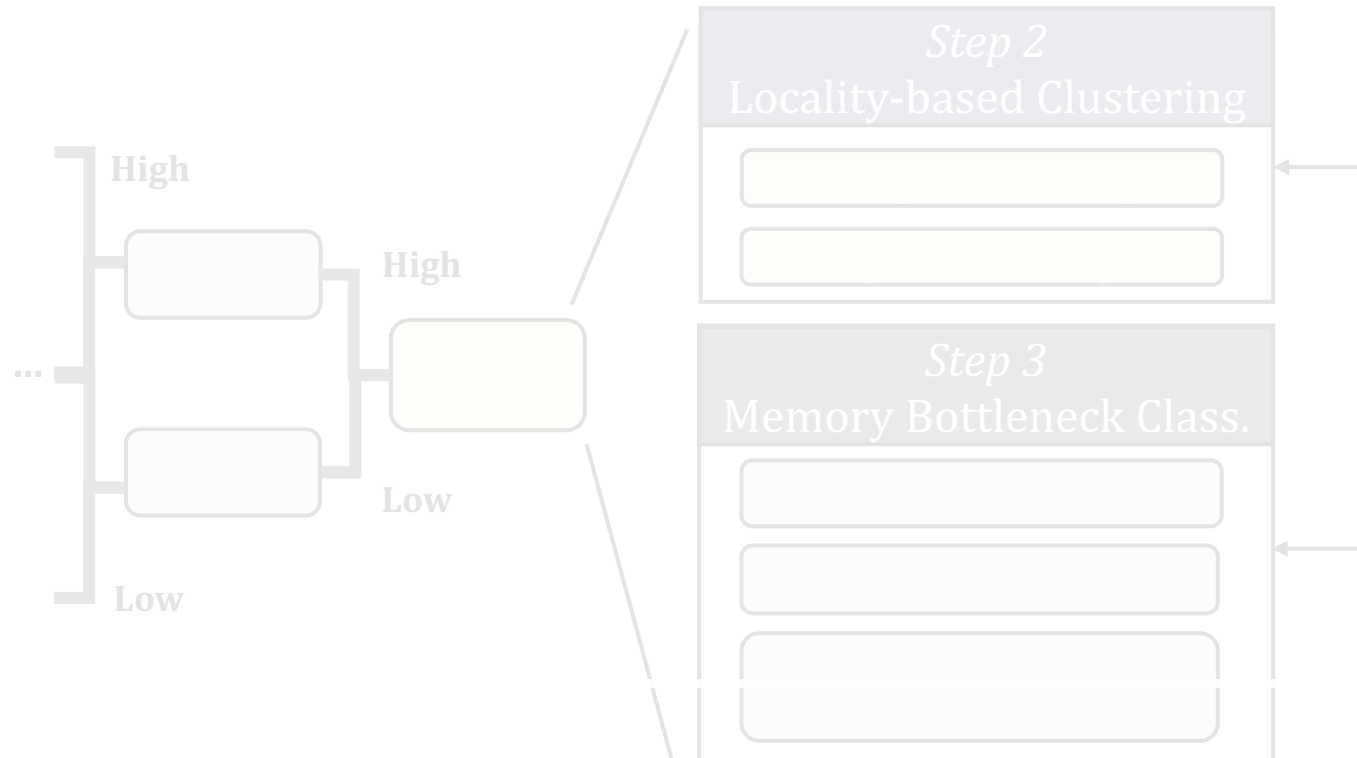
Methodology Overview



Methodology Output

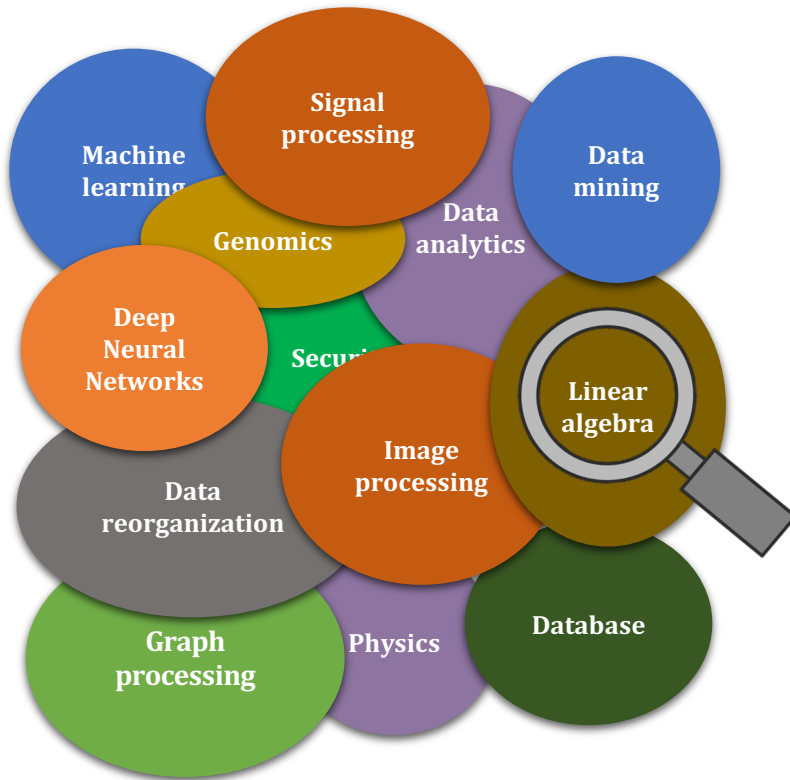


SAFARI

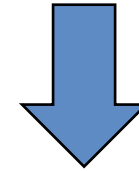


Step 1: Application Profiling

Goal: Identify **application functions** that suffer from **data movement bottlenecks**

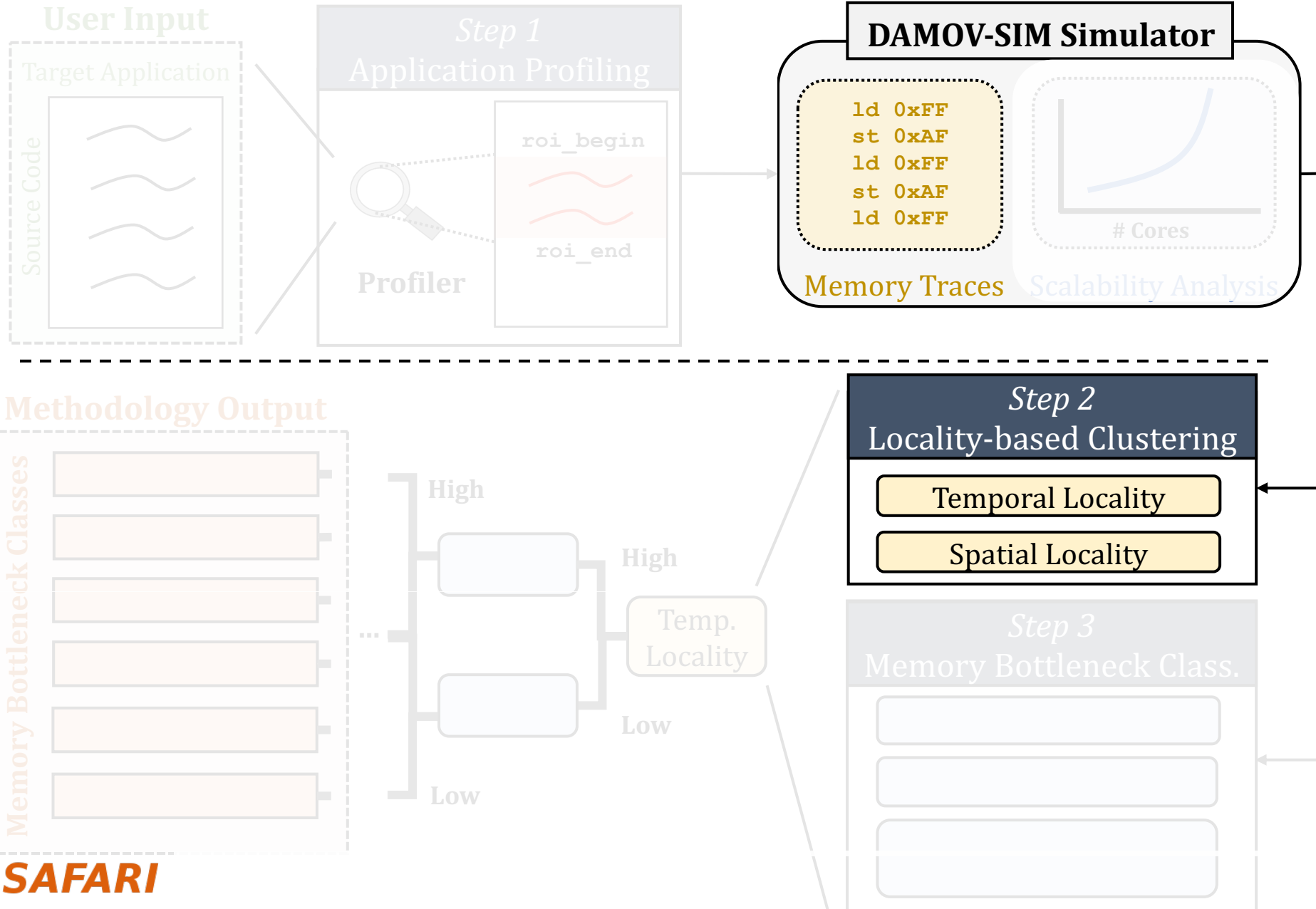


Hardware Profiling Tool:
Intel VTune



MemoryBound:
CPU is stalled due to load/store

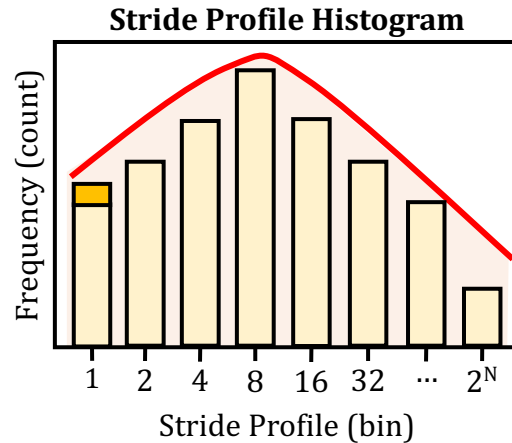
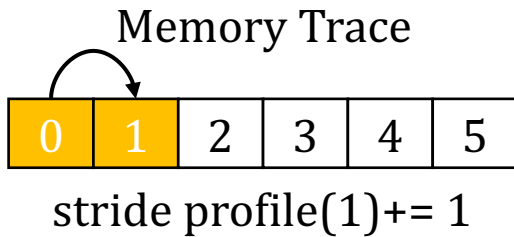
Methodology Overview



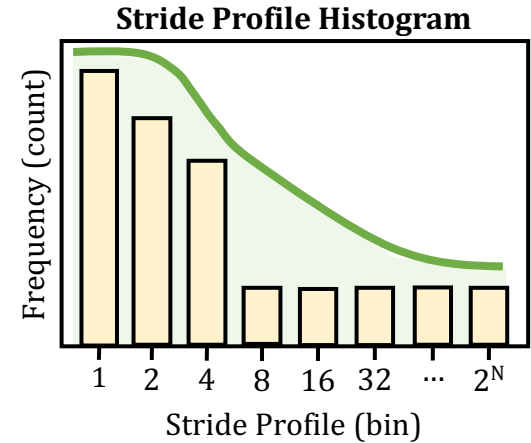
Step 2: Locality-Based Clustering

- **Goal:** analyze application's memory characteristics

Spatial Locality⁷



Low spatial locality

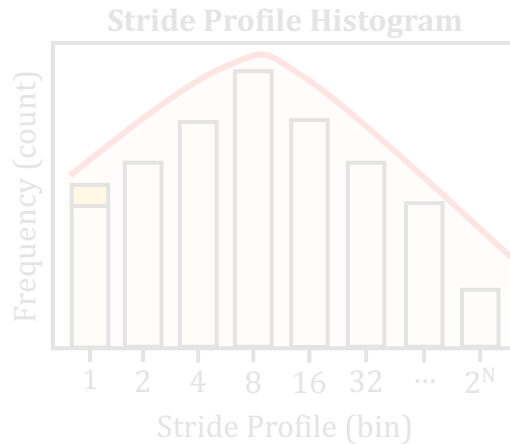
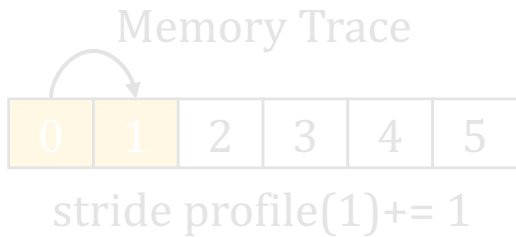


High spatial locality

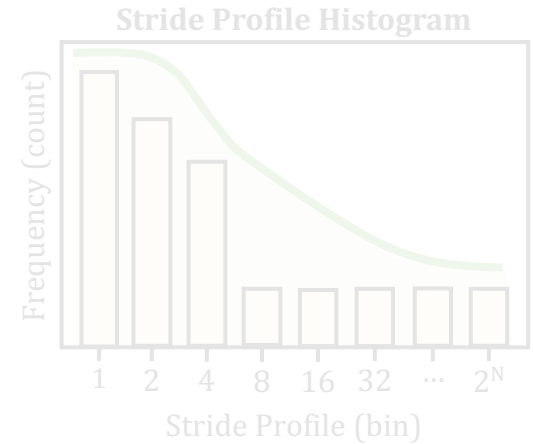
Step 2: Locality-Based Clustering

- **Goal:** analyze application's memory characteristics

Spatial Locality⁷

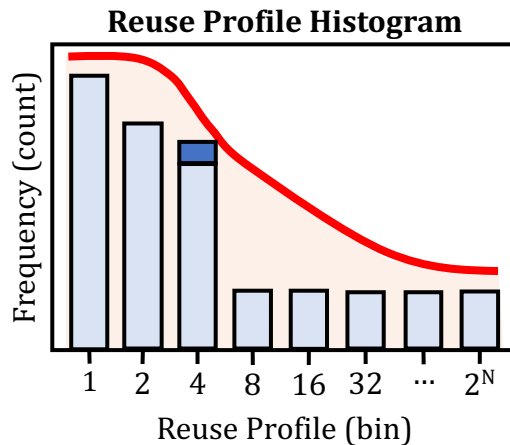
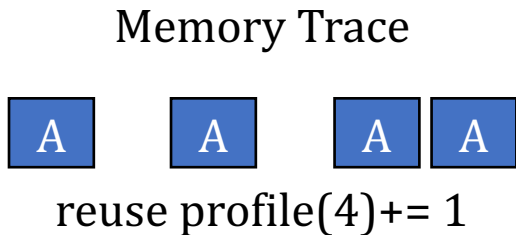


Low spatial locality

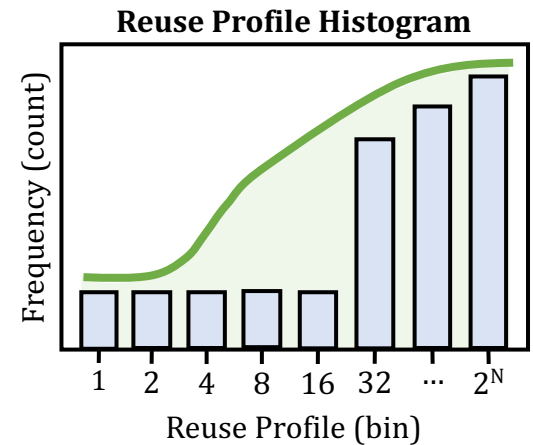


High spatial locality

Temporal Locality⁷

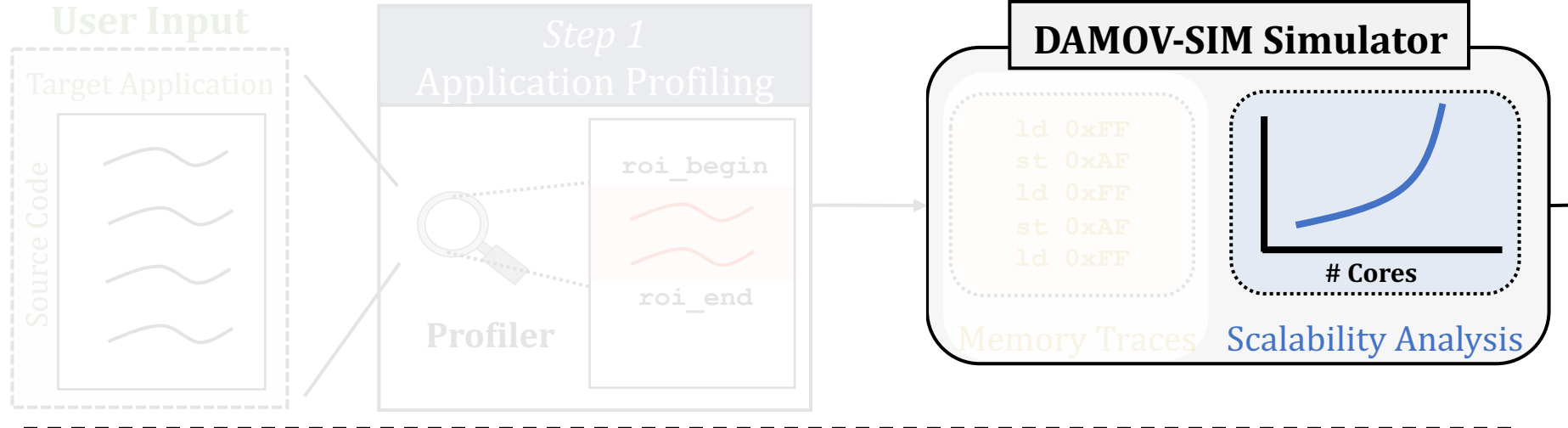


Low temporal locality



High temporal locality

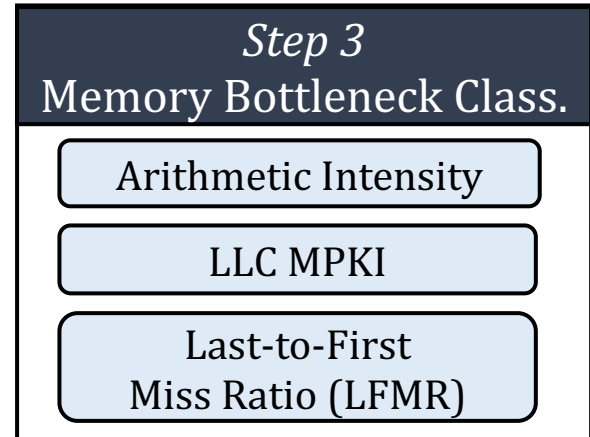
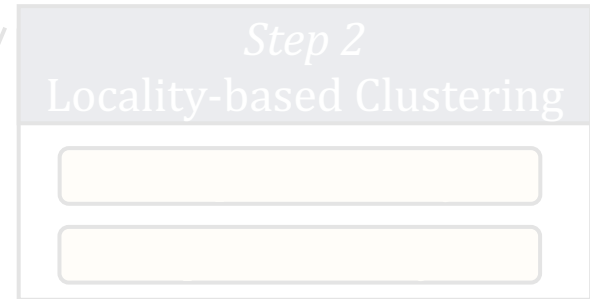
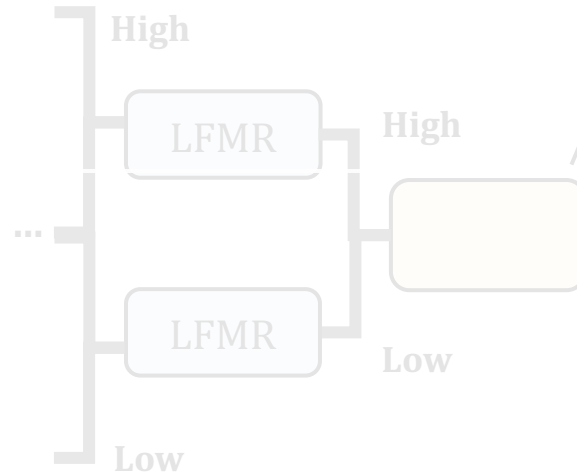
Methodology Overview



Methodology Output



SAFARI



Step 3: Memory Bottleneck Classification (1/2)

Arithmetic Intensity (AI)

- floating-point/arithmetic operations per L1 cache lines accessed
→ shows **computational intensity** per memory request

LLC Misses-per-Kilo-Instructions (MPKI)

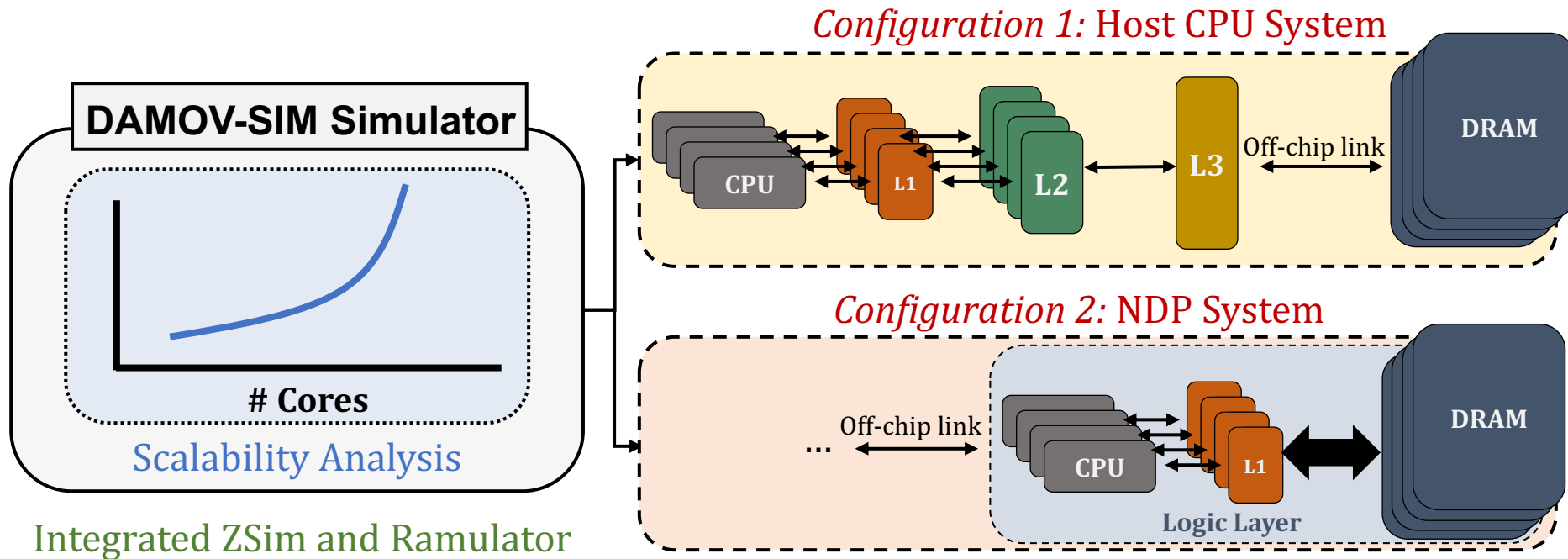
- LLC misses per one thousand instructions
→ shows **memory intensity**

Last-to-First Miss Ratio (LFMR)

- LLC misses per L1 misses
→ shows if an application **benefits from L2/L3 caches**

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

Step 3: Memory Bottleneck Analysis

Six classes of data movement bottlenecks:

each class \leftrightarrow data movement mitigation mechanism

Memory Bottleneck Class

1a: *DRAM Bandwidth*

1b: *DRAM Latency*

1c: *L1/L2 Cache Capacity*

2a: *L3 Cache Contention*

2b: *L1 Cache Capacity*

2c: *Compute-Bound*

DAMOV is Open Source

- We open-source our **benchmark suite** and our **toolchain**

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

omutlu Update README.md	ce1b4ea 17 days ago	5 commits
simulator	Cleaning	19 days ago
README.md	Update README.md	17 days ago
get_workloads.sh	DAMOV -- first commit	19 days ago

About

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>)

Readme

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages



DAMOV-SIM
DAMOV
Benchmarks

README.md

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](#).

DAMOV is Open Source

- We open-source our [benchmark suite](#) and our [toolchain](#)

CMU-SAFARI / DAMOV

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

Go to file

Add file

Code

About

DAMOV is a benchmark suite and a

Get DAMOV at:

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

README.md

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](#).

Readme

Releases

No releases published
[Create a new release](#)

Packages

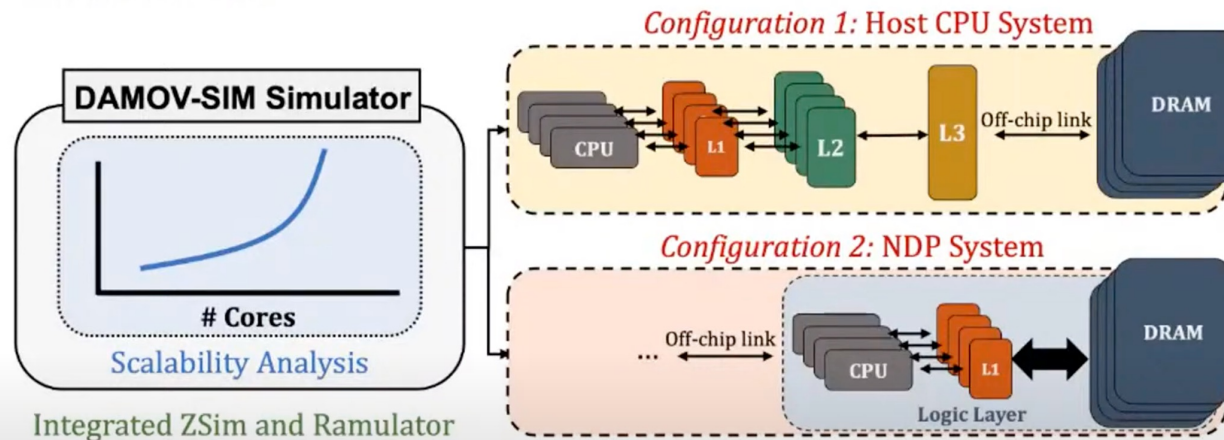
No packages published
[Publish your first package](#)

Languages

More on DAMOV Analysis Methodology & Workloads

Step 3: Memory Bottleneck Classification (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 DAMOV-SIM: <https://github.com/CMU-SAFARI/DAMOV> 30

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

352 views • Streamed live on Jul 22, 2021

18 0 SHARE SAVE ...



Onur Mutlu Lectures
17.7K subscribers

ANALYTICS

EDIT VIDEO

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 GÓMEZ-LUNA, ETH Zürich, Switzerland

LOIS OROSA, ETH Zürich, Switzerland

SAUGATA GHOSE, University of Illinois at Urbana–Champaign, USA

NANDITA VIJAYKUMAR, University of Toronto, Canada

IVAN FERNANDEZ, University of Malaga, Spain & ETH Zürich, Switzerland

MOHAMMAD SADROSADATI, ETH Zürich, Switzerland

ONUR MUTLU, ETH Zürich, Switzerland

Fundamentally
Energy-Efficient
(Data-Centric)
Computing Architectures

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

Computing Architectures with Minimal Data Movement

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

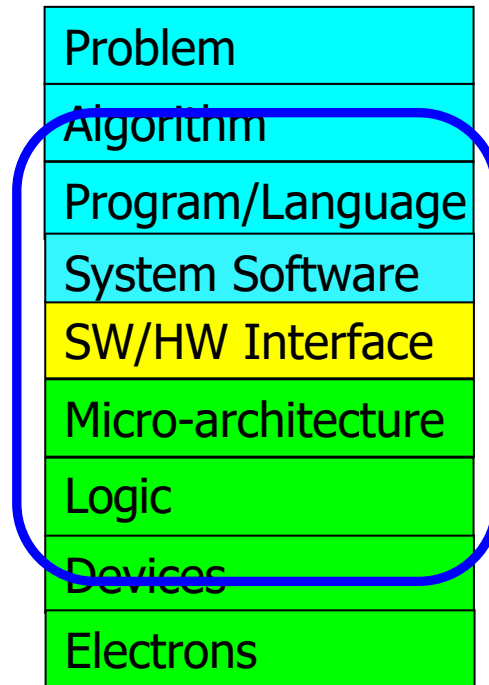
Fundamentally Better Architectures

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

PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*University of Illinois at Urbana-Champaign*

^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"](#)

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

A Blueprint for Fundamentally Better Architectures

- Onur Mutlu,
"Intelligent Architectures for Intelligent Computing Systems"
Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (DATE), Virtual, February 2021.
[[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/>



SAFARI
SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

SAFARI

<https://safari.ethz.ch>

SAFARI Newsletter December 2021 Edition

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

SAFARI
SAFARI Research Group

Think Big, Aim High



ETH zürich

View in your browser
December 2021



SAFARI Introduction & Research

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

SAFARI Research Group
Introduction & Research

Onur Mutlu
omutlu@gmail.com
<https://people.inf.ethz.ch/omutlu>
23 March 2023
Computer Architecture Seminar

SAFARI ETH zürich Carnegie Mellon

0:03 / 1:47:54 • Intro >

Seminar in Computer Architecture - Lecture 5: Potpourri of Research Topics (Spring 2023)



Onur Mutlu Lectures
32.6K subscribers

Subscribed

17



Share

Download

Clip



719 views Streamed 1 month ago Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2023)

SAFARI
SAFARI Research Group
safari.ethz.ch

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 U... 🔗 <https://safari.ethz.ch/> ✉ omutlu@gmail.com

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

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



prim-benchmarks

Public

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

● C ☆ 82 🍴 35




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




rowhammer

Public

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

● 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

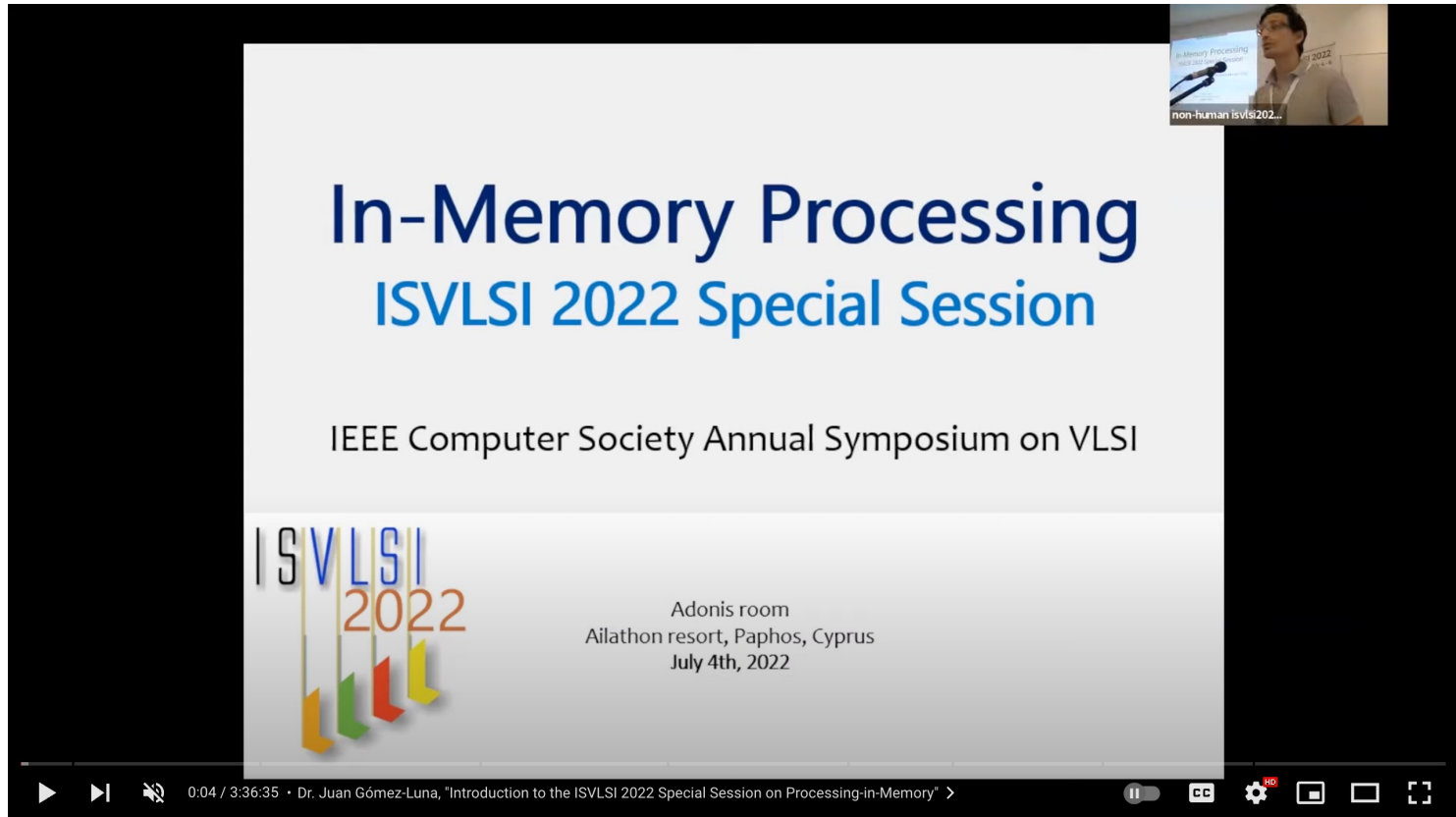
SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 99 🍴 26

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

Special Research Sessions & Courses

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



ISVLSI 2022 Special Session on Processing-in-Memory

1,286 views · Premiered Aug 9, 2022

61 DISLIKE SHARE DOWNLOAD CLIP SAVE ...



Onur Mutlu Lectures
26.9K subscribers

ANALYTICS EDIT VIDEO

Special Research Sessions & Courses (II)

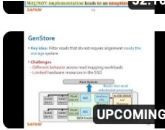


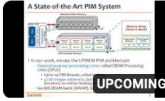
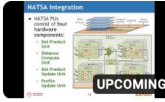


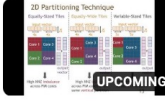

Special Session at ISVLSI 2022: 9 cutting-edge talks

Livestream - P&S Data-Centric Architectures: Fundamentally...
Onur Mutlu Lectures
27 videos 1,034 views Last updated on Feb 25, 2023

3.2. Their Approaches: Processing Using Memory (PIM) 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.

Onur Mutlu, Saugata Ghose, Juan Gomez-Liu, and Gokhan Arslanoglu
2. Hybrid PIM vs. Processing-in-Memory
3. Hybrid PIM vs. Processing-in-Memory
4. Hybrid PIM vs. Processing-in-Memory
5. Hybrid PIM vs. Processing-in-Memory
6. Hybrid PIM vs. Processing-in-Memory
7. Hybrid PIM vs. Processing-in-Memory

Approach	Enabling Subtechniques
Processing Using Memory	SRAM MRAM Phase-change memory (PCM) Resistive RAM (RRAM) Reconfigurable logic devices
Processing Near Memory	Logic built on DRAM-based memory Memory controller Logic in memory (e.g., sense amplifiers) Logic in memory (e.g., sense amplifiers) Logic in memory (e.g., sense amplifiers)

- 19  **GenStore: In-Storage Filtering for High-Performance and Energy-Efficient Genome Analysis**
Onur Mutlu Lectures • Premieres 3/12/23, 7:00 PM
- 20  **Introduction to the ISVLSI 2022 Special Session on Processing-in-Memory**
Onur Mutlu Lectures • 286 views • 2 days ago
- 21  **Heterogeneous Data-Centric Architectures for Data-Intensive Applications: Case Studies in ML and DB**
Onur Mutlu Lectures • 2 waiting • Premieres 3/10/23, 7:00 PM
- 22  **Machine Learning Training on a Real Processing-In-Memory System**
Onur Mutlu Lectures • Premieres 3/14/23, 7:00 PM
- 23  **Exploiting Near-Data Processing to Accelerate Time Series Analysis**
Onur Mutlu Lectures • Premieres 3/11/23, 7:00 PM
- 24  **PIDRAM: An FPGA-Based Framework for End-To-End Evaluation of Processing-In-DRAM Techniques**
Onur Mutlu Lectures • Premieres 3/9/23, 7:00 PM
- 25  **The Road to Widely Deploying Processing-In-Memory: Challenges and Opportunities**
Onur Mutlu Lectures • 399 views • 1 day ago
- 26  **SparseP: Efficient Sparse Matrix Vector Multiplication on Real Processing-In-Memory Architectures**
Onur Mutlu Lectures • 1 waiting • Premieres 3/13/23, 7:00 PM
- 27  **HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures**
Onur Mutlu Lectures • 1.6K views • Streamed 10 days ago

Comp Arch (Fall 2021)

Fall 2021 Edition:

- <https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule>

Fall 2020 Edition:

- <https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule>

Youtube Livestream (2021):

- https://www.youtube.com/watch?v=4yfkM_5EFg0&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>

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

- Computer Architecture FS20: Course Webpage
- Computer Architecture FS20: Lecture Videos
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HoICRP
- Verilog Practice Website (HDLBits)

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

2.560 DRAM Processing in a Memory System

Watch on YouTube

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

Recorded Lecture Playlist

Tesla Full Self-Driving Computer (2020)

- ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.

Watch on YouTube

Fall 2021 Lectures & Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	30.09 Thu.	YouTube Live	L1: Introduction and Basics 📄 (PDF) 📄 (PPT)	Required Mentioned	Lab 1 Out	HW 0 Out
	01.10 Fri.	YouTube Live	L2: Trends, Tradeoffs and Design Fundamentals 📄 (PDF) 📄 (PPT)	Required Mentioned		
W2	07.10 Thu.	YouTube Live	L3a: Memory Systems: Challenges and Opportunities 📄 (PDF) 📄 (PPT)	Described Suggested		HW 1 Out
			L3b: Course Info & Logistics 📄 (PDF) 📄 (PPT)			
			L3c: Memory Performance Attacks 📄 (PDF) 📄 (PPT)	Described Suggested		
	08.10 Fri.	YouTube Live	L4a: Memory Performance Attacks 📄 (PDF) 📄 (PPT)	Described Suggested	Lab 2 Out	
			L4b: Data Retention and Memory Refresh 📄 (PDF) 📄 (PPT)	Described Suggested		
			L4c: RowHammer 📄 (PDF) 📄 (PPT)	Described Suggested		



DDCA (Spring 2022)

Spring 2022 Edition:

https://safari.ethz.ch/digitaltechnik/spring2022/duku.php?id=schedule

Spring 2021 Edition:

https://safari.ethz.ch/digitaltechnik/spring2021/duku.php?id=schedule

Youtube Livestream (Spring 2022):

https://www.youtube.com/watch?v=cpXdE3HwvK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6

Youtube Livestream (Spring 2021):

https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

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>

Trace: schedule

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

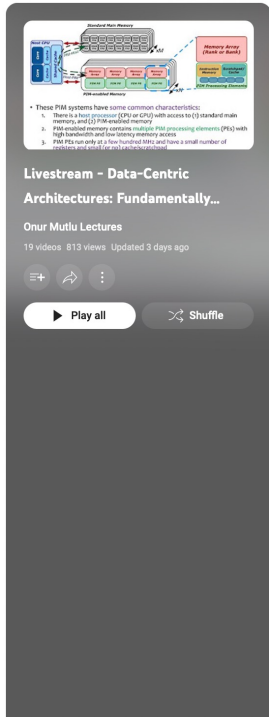
Recorded Lecture Playlist

Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics PDF (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset PDF (PPT) L2b: Mysteries in Computer Architecture PDF (PPT)	Required Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II PDF (PPT)	Required Suggested Mentioned		

Processing-in-Memory Course (Spring 2023)

- Short weekly lectures
- Hands-on projects



Livestream - Data-Centric Architectures: Fundamentally...


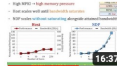
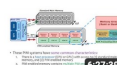

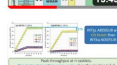




Onur Mutlu Lectures

19 videos · 813 views · Updated 3 days ago

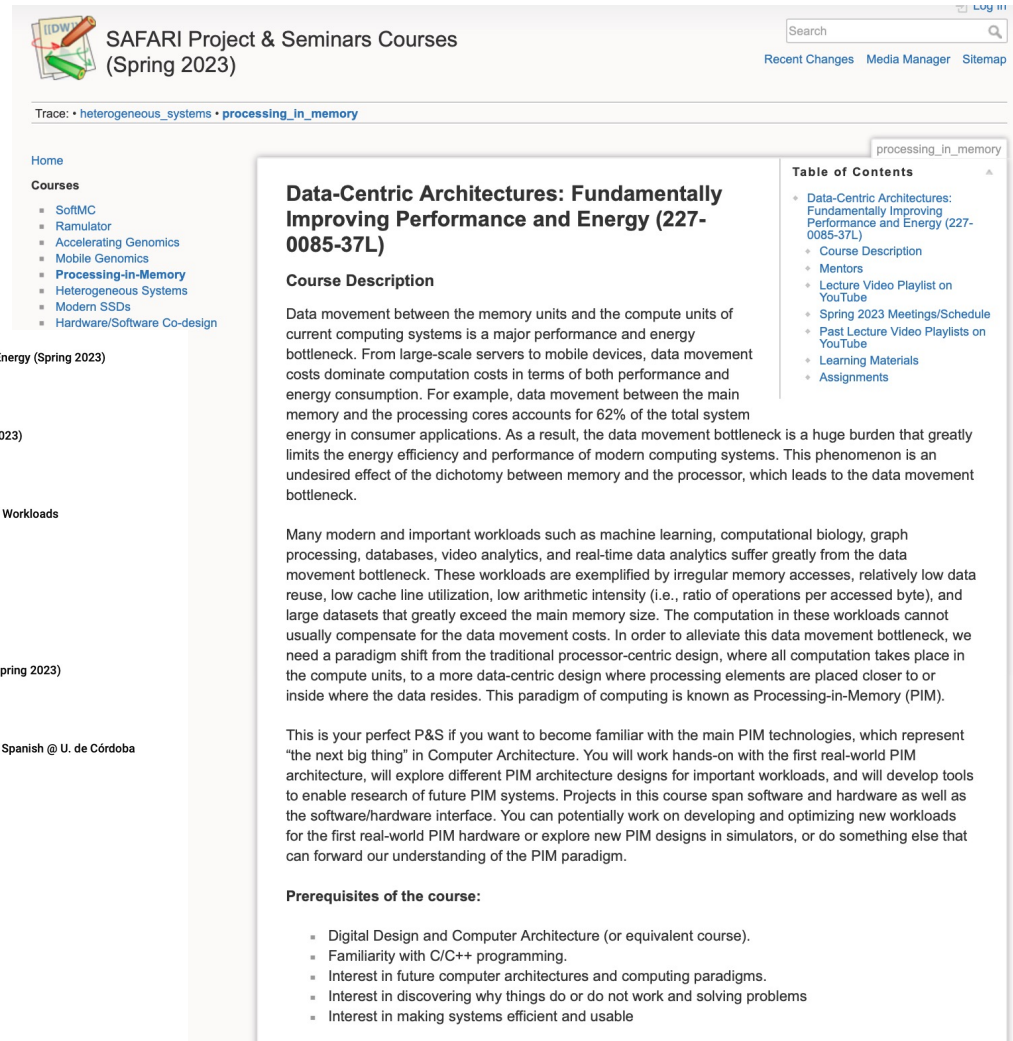
▶ Play all

⌂ Shuffle

• These PIM systems have some common characteristics:
1. There is a shared controller (CPU or GPU) with access to (1) external main memory, and (2) PIM enabled memory
2. The enabled memory contains random data processing elements (PEs) with high bandwidth and low latency memory access
3. PIM PE capability is a few hundred bits, and has a small number of context and state (1 or 2) on an external controller

-  **PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy (Spring 2023)**
Onur Mutlu Lectures · 1.1K views · Streamed 3 months ago
1:14:16
-  **PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks (Spring 2023)**
Onur Mutlu Lectures · 332 views · 2 months ago
16:37
-  **ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads**
Onur Mutlu Lectures · 1.5K views · Streamed 2 months ago
6:27:39
-  **PIM Course: Lecture 3: Real-world PIM: UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures · 411 views · 2 months ago
15:43
-  **PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM (Spring 2023)**
Onur Mutlu Lectures · 188 views · 2 months ago
24:10
-  **Análisis Experimental de una Arquitectura PIM - Juan Gómez Luna - Lecture in Spanish @ U. de Córdoba**
Onur Mutlu Lectures · 169 views · 2 months ago
2:27:12
-  **PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM (Spring 2023)**
Onur Mutlu Lectures · 483 views · 2 months ago
24:08
-  **PIM Course: Lecture 6: Real-world PIM: SK Hynix AIM (Spring 2023)**
Onur Mutlu Lectures · 573 views · 1 month ago
35:50
-  **PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM (Spring 2023)**
Onur Mutlu Lectures · 325 views · 1 month ago
21:32

https://www.youtube.com/playlist?list=PL5Q2soXY2zi_EObuoAZVSq_o6UySWQHvz



SAFARI Project & Seminars Courses (Spring 2023)

Trace: heterogeneous_systems · processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- Processing-in-Memory**
- Heterogeneous Systems
- Modern SSDs
- Hardware/Software Co-design

processing_in_memory

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.

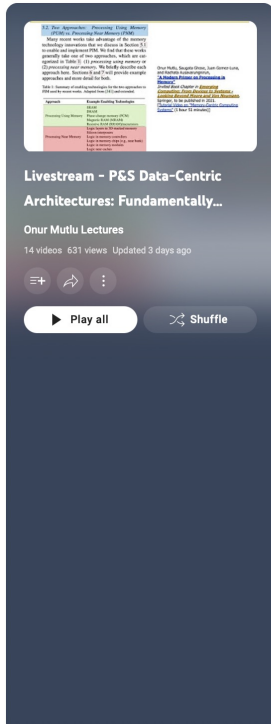
Prerequisites of the course:

- Digital Design and Computer Architecture (or equivalent course).
- Familiarity with C/C++ programming.
- Interest in future computer architectures and computing paradigms.
- Interest in discovering why things do or do not work and solving problems
- Interest in making systems efficient and usable

https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=processing_in_memory

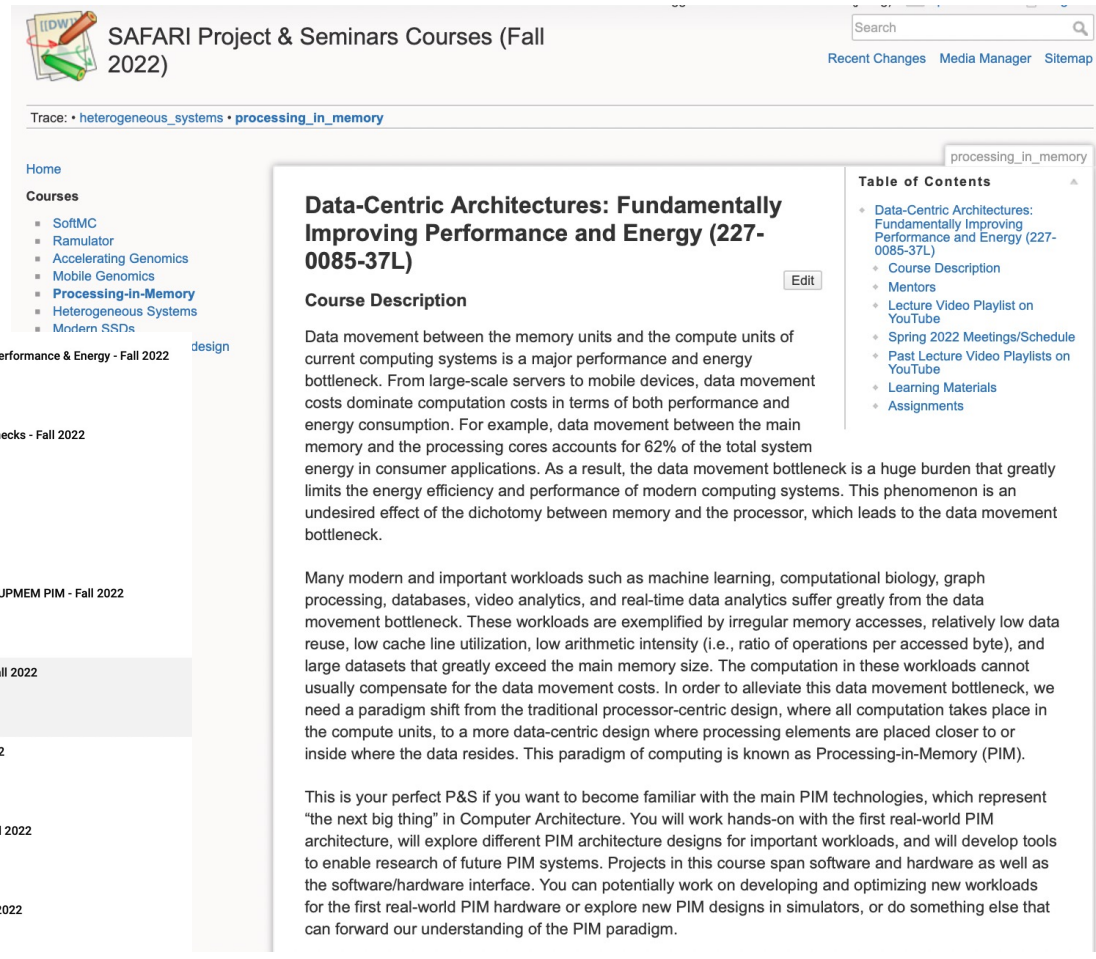
Processing-in-Memory Course (Fall 2022)

- Short weekly lectures
- Hands-on projects



Livestream - P&S Data-Centric Architectures: Fundamentally...
Onur Mutlu Lectures
14 videos 631 views Updated 3 days ago
Play all Shuffle

- 1 PIM Course: Lecture 1: Data-Centric Architectures: Improving Performance & Energy - Fall 2022
Onur Mutlu Lectures • 1K views • 3 months ago
1:34:40
- 2 PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks - Fall 2022
Onur Mutlu Lectures • 678 views • 2 months ago
59:51
- 3 PIM Course: Lecture 3: Real-world PIM: UPMEM PIM - Fall 2022
Onur Mutlu Lectures • 455 views • 2 months ago
31:46
- 4 PIM Course: Lecture 4: Real-world PIM: Microbenchmarking of UPMEM PIM - Fall 2022
Onur Mutlu Lectures • 275 views • 2 months ago
56:05
- 5 PIM Course: Lecture 5: Real-world PIM: Samsung HBM-PIM - Fall 2022
Onur Mutlu Lectures • 725 views • 2 months ago
1:06:09
- 6 PIM Course: Lecture 6: Real-world PIM: SK Hynix AiM - Fall 2022
Onur Mutlu Lectures • 1K views • 2 months ago
41:33
- 7 PIM Course: Lecture 7: Real-world PIM: Samsung AxDIMM - Fall 2022
Onur Mutlu Lectures • 767 views • 1 month ago
32:43
- 8 PIM Course: Lecture 8: Real-world PIM: Alibaba HB-PNM - Fall 2022
Onur Mutlu Lectures • 383 views • 1 month ago
41:44
- 9 PIM Course: Lecture 9: Programming PIM Architectures - Fall 2022
Onur Mutlu Lectures • 367 views • 1 month ago
46:43



SAFARI Project & Seminars Courses (Fall 2022)

Trace: heterogeneous_systems • processing_in_memory

Home

Courses

- SoftMC
- Ramulator
- Accelerating Genomics
- Mobile Genomics
- Processing-in-Memory
- Heterogeneous Systems
- Modern SSDs

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.

Table of Contents

- Data-Centric Architectures: Fundamentally Improving Performance and Energy (227-0085-37L)
- Course Description
- Mentors
- Lecture Video Playlist on YouTube
- Spring 2022 Meetings/Schedule
- Past Lecture Video Playlists on YouTube
- Learning Materials
- Assignments

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

<https://youtube.com/playlist?list=PL5Q2soXY2Zi8KzG2CQYRNQOVD0GOBrnKy>

Real PIM Tutorial (HPCA 2023)

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

HPCA 2023 Real-World PIM Tutorial

Real-world Processing-in-Memory Architectures

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, 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.

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 hand-on labs using real PIM systems, and (4) shed light on how to enable the adoption of PIM in future computing systems.

2,560-DPU Processing-in-Memory System

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

Goal: Processing Inside Memory

Processor Core

Memory

Interconnect

Query

Results

Database

Graphs

Media

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

HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures

Onur Mutlu Lectures

32.1K subscribers

1.8K views Streamed 1 month ago Livestream - P&S Data-Centric Architectures: Fundamentally Improving Performance and Energy (Fall 2022)

HPCA 2023 Tutorial: Real-World Processing-in-Memory Architectures

<https://events.safari.ethz.ch/real-pi...>

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

<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

ASPLOS 2023 Real-World PIM Tutorial

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

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) assess estimation strategies for PIM kernels, and (3)

Tutorial Materials

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

ASPLOS 2023 Tutorial

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

Accelerating Modern Workloads on a General-purpose PIM System

Dr. Juan Gómez Luna
Professor Onur Mutlu

ETH Zürich SAFARI

Sunday, March 26, 2023

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

Onur Mutlu Lectures

32.1K subscribers

Subscribed

33

Share

Clip

Save

Views Streamed 7 days ago Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

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

<https://events.safari.ethz.ch/asplos-2023/>

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

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

Real PIM Tutorial [ISCA 2023]

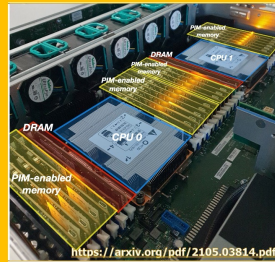
■ June 18: Lectures + Hands-on labs + Invited talks

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/>



Overview PIM | PNM | UPMEM PIM |
PNM for neural networks |
PNM for recommender systems |
PNM for ML workloads |
How to enable PIM? | PUM prototypes
Hands-on Labs: Benchmarking |
Accelerating real-world workloads



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

Onur Mutlu Lectures
33.9K subscribers

Subscribed

57

Share

Download

Clip

...

1,687 views · Streamed live on Jun 18, 2023 · Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)

[https://www.youtube.com/
live/GIb5EgSrWk0](https://www.youtube.com/live/GIb5EgSrWk0)

[https://events.safari.ethz.ch/
isca-pim-tutorial/](https://events.safari.ethz.ch/isca-pim-tutorial/)

Tutorial Materials

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

Current Real PIM Tutorial [MICRO 2023]

■ October 29: 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.

2,560-DPU Processing-in-Memory System

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

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

Onur Mutlu Lectures
34.6K subscribers

5 likes

Share

Save

<https://www.youtube.com/live/ohUooNSIxOI>

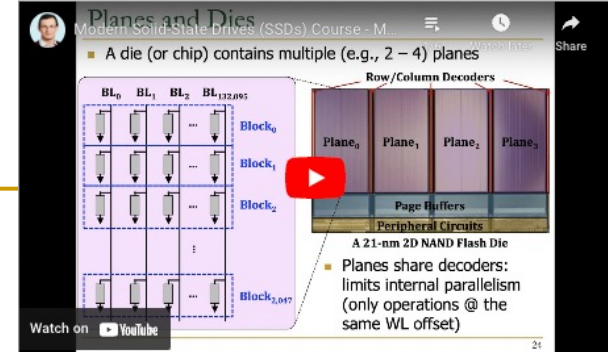
<https://events.safari.ethz.ch/micro-pim-tutorial>

Agenda (Tentative, October 29, 2023)

Lectures

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

SSD Course (Spring 2023)



Spring 2023 Edition:

- https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=modern_ssd

Fall 2022 Edition:

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

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=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB>

Project course

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

Fall 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	06.10		M1: P&S Course Presentation PDF PPT	Required Recommended	
W2	12.10	YouTube Live	M2: Basics of NAND Flash-Based SSDs PDF PPT	Required Recommended	
W3	19.10	YouTube Live	M3: NAND Flash Read/Write Operations PDF PPT	Required Recommended	
W4	26.10	YouTube Live	M4: Processing inside NAND Flash PDF PPT	Required Recommended	
W5	02.11	YouTube Live	M5: Advanced NAND Flash Commands & Mapping PDF PPT	Required Recommended	
W6	09.11	YouTube Live	M6: Processing inside Storage PDF PPT	Required Recommended	
W7	23.11	YouTube Live	M7: Address Mapping & Garbage Collection PDF PPT	Required Recommended	
W8	30.11	YouTube Live	M8: Introduction to MQSim PDF PPT	Required Recommended	
W9	14.12	YouTube Live	M9: Fine-Grained Mapping and Multi-Plane Operation-Aware Block Management PDF PPT	Required Recommended	
W10	04.01.2023	YouTube Premiere	M10a: NAND Flash Basics PDF PPT	Required Recommended	
			M10b: Reducing Solid-State Drive Read Latency by Optimizing Read-Retry PDF PPT Paper	Required Recommended	
			M10c: Evanesco: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems PDF PPT Paper	Required Recommended	
			M10d: DeepSketch: A New Machine Learning-Based Reference Search Technique for Post-Deduplication Delta Compression PDF PPT Paper	Required Recommended	
W11	11.01	YouTube Live	M11: FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives PDF PPT	Required	
W12	25.01	YouTube Premiere	M12: Flash Memory and Solid-State Drives PDF PPT	Recommended	

<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=PL5Q2soXY2Zi8tFIQvdxOdizD_EhVAMVQV

- **Youtube Livestream (Spring 2022):**

- https://www.youtube.com/watch?v=DEL_5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18

- Project course

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

<https://www.youtube.com/onurmutlectures>

SAFARI

Accelerating Genomics Course - Meeting 1: C...

Genomic Sample → Sequencing Machine → Reads → Read Mapping → Genomic Variants

1 Indexing: Reference Genome, k-mers, Index, k-mer content locations, Locating common k-mers

2 Pre-Alignment Filtering: Reference subsequences extracted at each common k-mer location

3 Sequence Alignment: Read, Reference subsequence, Dynamic Programming (DP) Matrix, SAM file (alignment score, edit distance, type and location of each edit)

Accelerating Indexing: Reducing the number of seeds, Reducing read movement during indexing

Accelerating Pre-Alignment Filtering: q-gram filtering, Pigeonhole principle, Base counting, Sparse DP

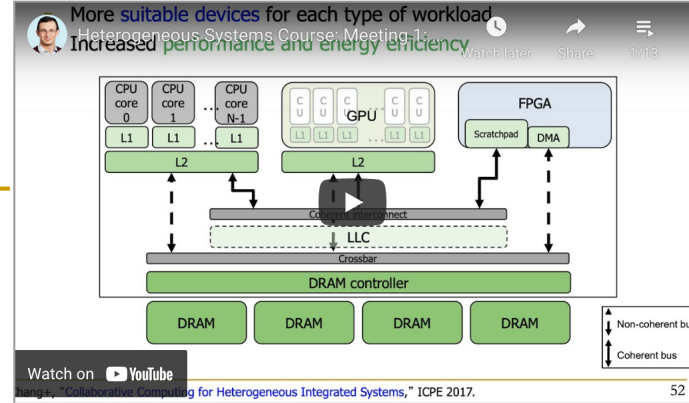
Accelerating Alignment: Accurate alignment accelerators, Heuristic-based alignment accelerators

Watch on YouTube

Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials
W1	11.3 Fri.	YouTube Live	M1: P&S Accelerating Genomics Course Introduction & Project Proposals PDF PPT	Required Materials Recommended Materials
W2	18.3 Fri.	YouTube Live	M2: Introduction to Sequencing PDF PPT	
W3	25.3 Fri.	YouTube Premiere	M3: Read Mapping PDF PPT	
W4	01.04 Fri.	YouTube Premiere	M4: GateKeeper PDF PPT	
W5	08.04 Fri.	YouTube Premiere	M5: MAGNET & Shouji PDF PPT	
W6	15.4 Fri.	YouTube Premiere	M6: SneakySnake PDF PPT	
W7	29.4 Fri.	YouTube Premiere	M7: GenStore PDF PPT	
W8	06.05 Fri.	YouTube Premiere	M8: GRIM-Filter PDF PPT	
W9	13.05 Fri.	YouTube Premiere	M9: Genome Assembly PDF PPT	
W10	20.05 Fri.	YouTube Live	M10: Genomic Data Sharing Under Differential Privacy PDF PPT	
W11	10.06 Fri.	YouTube Premiere	M11: Accelerating Genome Sequence Analysis PDF PPT	

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=oFO5fTrgFIY&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>

Spring 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	15.03 Tue.	YouTube Premiere	M1: P&S Course Presentation PDF PPT	Required Materials Recommended Materials	HW 0 Out
W2	22.03 Tue.	YouTube Premiere	M2: SIMD Processing and GPUs PDF PPT		
W3	29.03 Tue.	YouTube Premiere	M3: GPU Software Hierarchy PDF PPT		
W4	05.04 Tue.	YouTube Premiere	M4: GPU Memory Hierarchy PDF PPT		
W5	12.04 Tue.	YouTube Premiere	M5: GPU Performance Considerations PDF PPT		
W6	19.04 Tue.	YouTube Premiere	M6: Parallel Patterns: Reduction PDF PPT		
W7	26.04 Tue.	YouTube Premiere	M7: Parallel Patterns: Histogram PDF PPT		
W8	03.05 Tue.	YouTube Premiere	M8: Parallel Patterns: Convolution PDF PPT		
W9	10.05 Tue.	YouTube Premiere	M9: Parallel Patterns: Prefix Sum (Scan) PDF PPT		
W10	17.05 Tue.	YouTube Premiere	M10: Parallel Patterns: Sparse Matrices PDF PPT		
W11	24.05 Tue.	YouTube Premiere	M11: Parallel Patterns: Graph Search PDF PPT		
W12	01.06 Wed.	YouTube Premiere	M12: Parallel Patterns: Merge Sort PDF PPT		
W13	07.06 Tue.	YouTube Premiere	M13: Dynamic Parallelism PDF PPT		
W14	15.06 Wed.	YouTube Premiere	M14: Collaborative Computing PDF PPT		
W15	24.06 Fri.	YouTube Premiere	M15: GPU Acceleration of Genome Sequence Alignment PDF PPT		
W16	14.07 Thu.	YouTube Premiere	M16: Accelerating Agent-based Simulations PDF ODP		

HW/SW Co-Design (Spring 2022)

Spring 2022 Edition:

- https://safari.ethz.ch/projects and seminars/spring2022/doku.php?id=hw_sw_co_design

Youtube Livestream:

- 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

SMASH Hardware-Software Co-Design Mechanism: Sparse Matrix Compression
Enables **highly-efficient** sparse matrix compression and computation
General across a diverse set of sparse matrices and sparse matrix operations

Software
Efficient compression using a Hierarchy of Bitmaps

Hardware
Unit that scans bitmaps to accelerate indexing

Watch on YouTube

SMASH ISA

Hardware-Managed Memory
The Virtual Block Interface: A Flexible Alternative to the Conventional...

- Memory management is **delegated** to the **Memory Translation Layer (MTL)** in the memory controller
 - Address translation
 - Physical memory allocation
- Pros:** Many benefits, including
 - Physical memory is allocated only when the location needs to be written to memory

Diagram showing Processes (P1, P2, ..., Pn) connected to Virtual Blocks (VB1, VB2, VB3, VB4) in the VBI Address Space, which is managed by the Memory Translation Layer in the memory controller, leading to Physical Memory.

Watch on YouTube

33

2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Materials	Assignments
W0	16.03	YouTube Live	Intro to HW/SW Co-Design (PPTX) (PDF)	Required	HW 0 Out
W1	23.03		Project selection	Required	
W2	30.03	YouTube Live	Virtual Memory (I) (PPTX) (PDF)		
W3	13.04	YouTube Live	Virtual Memory (II) (PPTX) (PDF)		

<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

Spring 2022 Edition:

- 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

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>

Lecture Video Playlist on YouTube

Lecture Playlist

The screenshot shows a YouTube video player interface. At the top, there's a profile picture and the text 'SoftMC Course: Meeting 1: Logistics & Intro ...'. Below that, the video title 'P&S SoftMC' is displayed in a large, bold, red font. Underneath the title, the subtitle 'Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments' is shown. A large red play button is centered on the video area. Below the play button, the presenter's name 'Hasan Hassan' and 'Prof. Onur Mutlu' are listed, along with 'ETH Zürich'. At the bottom, there's a 'Watch on YouTube' button.

2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W0	23.02 Wed.	Video	P&S SoftMC Tutorial	SoftMC Tutorial Slides (PDF) (PPT)	
W1	08.03 Tue.	Video	M1: Logistics & Intro to DRAM and SoftMC (PDF) (PPT)	Required Materials Recommended Materials	HW0
W2	15.03 Tue.	Video	M2: Revisiting RowHammer (PDF) (PPT)	(Paper PDF)	
W3	22.03 Tue.	Video	M3: Uncovering in-DRAM TRR & TRRespass (PDF) (PPT)		
W4	29.03 Tue.	Video	M4: Deeper Look Into RowHammer's Sensitivities (PDF) (PPT)		
W5	05.04 Tue.	Video	M5: QUAC-TRNG (PDF) (PPT)		
W6	12.04 Tue.	Video	M6: PiDRAM (PDF) (PPT)		

Exploration of Emerging Memory Systems (Fall 2022)

Fall 2022 Edition:

- 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

Youtube Livestream (Spring 2022):

- https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlMLGw_Z8hBo2925ZApqV

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>

Lecture Video Playlist on YouTube

Lecture Playlist

The image shows a YouTube video player interface. At the top, it says 'Lecture Playlist' and 'Ramulator Course: Meeting 1: Logistics & Int...'. Below that, the video title is 'P&S Ramulator' in red, followed by the subtitle 'Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator'. The presenter information is 'Hasan Hassan' and 'Prof. Onur Mutlu' from 'ETH Zürich'. At the bottom, it says 'Watch on YouTube'.

2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator (PDF) (PPT)		HW0
W2	16.03 Fri.	Video	M2: Tutorial on Using Ramulator (PDF) (PPT)		
W3	25.02 Fri.	Video	M3: BlockHammer (PDF) (PPT)		
W4	01.04 Fri.	Video	M4: CLR-DRAM (PDF) (PPT)		
W5	08.04 Fri.	Video	M5: SIMDRAM (PDF) (PPT)		
W6	29.04 Fri.	Video	M6: DAMOV (PDF) (PPT)		
W7	06.05 Fri.	Video	M7: Synchron (PDF) (PPT)		

Memory-Centric Computing

Geraldo F. Oliveira

Onur Mutlu

omutlu@gmail.com

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

29 October 2023

Real-World PIM Tutorial Opening Talk @ MICRO-56

SAFARI

ETH zürich

Carnegie Mellon