1st Workshop on Memory-Centric Computing: Processing-Near-Memory

Geraldo F. Oliveira
https://geraldofojunior.github.io

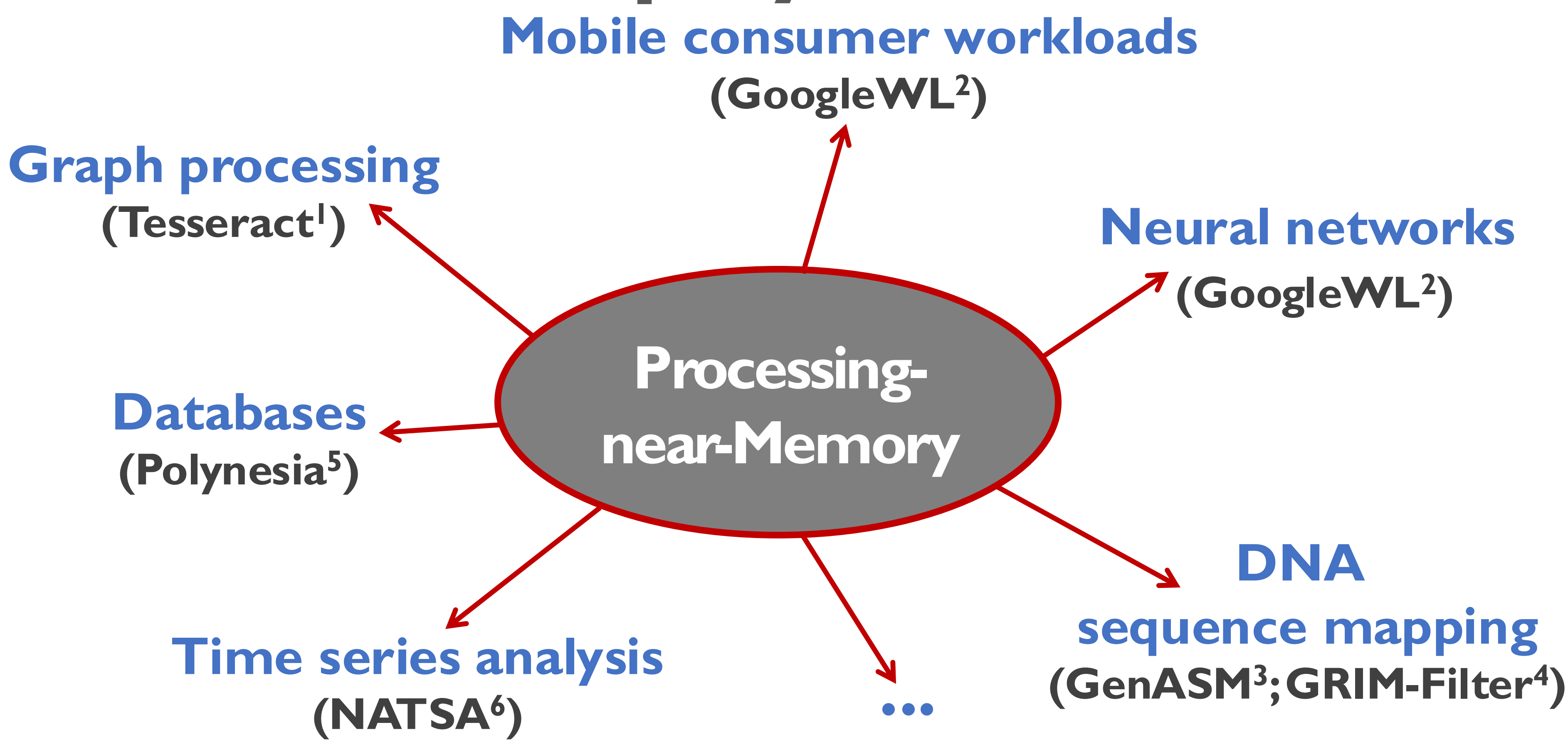
ASPLOS 2025
30 March 2025

SAFARI
ETH zürich

Processing in Memory: Two Approaches

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

When to Employ PNM



[1] Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA, 2015

[2] Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS, 2018

[3] Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis," MICRO, 2020

[4] Kim+, "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics, 2018

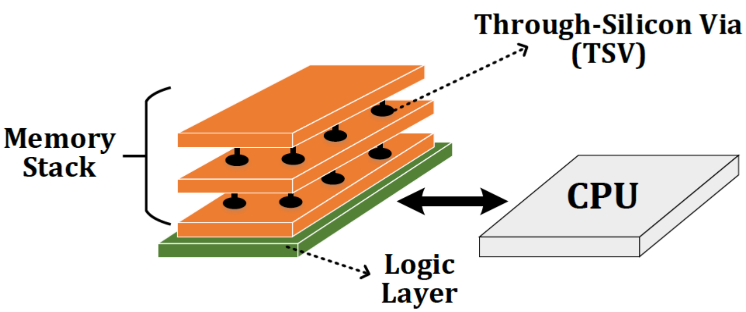
[5] Boroumand+, "Polynesia: Enabling High-Performance and Energy-Efficient Hybrid Transactional/Analytical Databases with Hardware/Software Co-Design," ICDE, 2022

[6] Fernandez+, "NATSA: A Near-Data Processing Accelerator for Time Series Analysis," ICCD, 2020

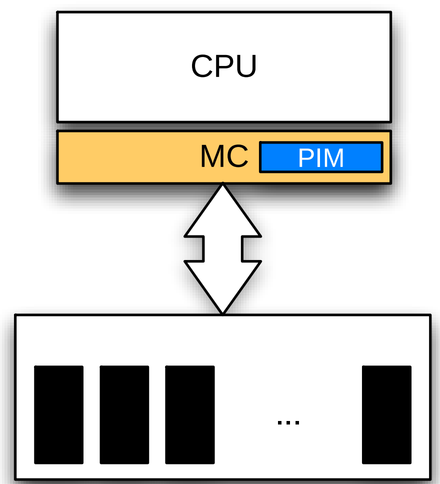
Processing Near-Memory (PNM)

- Processing Near-Memory (PNM)
  - Move computation closer to where the data resides

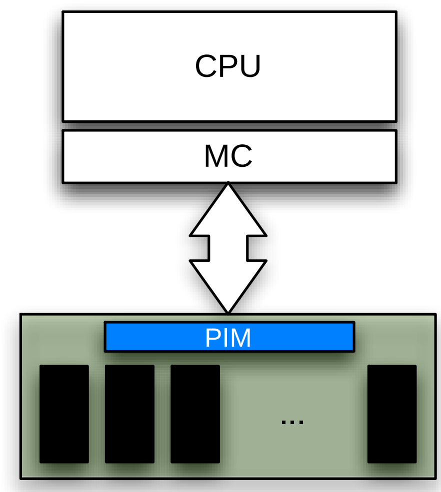
Logic layer
3D stacked DRAM



Memory controller



Memory module
(DIMM)



PNM: Design Challenges

- Limited power & area budget with 3D-stacked memories
  - e.g., area and power budget of the vault's underlying logic layer is just 4.4mm^2 and 312mW (circa HMC 2.0)

- Strict thermal constraints
  - It requires cooling solutions to remove heat throughout a 3D stack (i.e., volume-wise) instead of a 2D surface

- Challenging manufacturing of logic+DRAM
  - Logic process has been developed for speed performance, DRAM process for density and memory reliability
  - e.g., Logic gates implemented with memory process slowdowns by ~21.5% [Kim+, Integration'99]

Tesseract System for Graph Processing

- 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

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[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

PIM for Mobile Devices

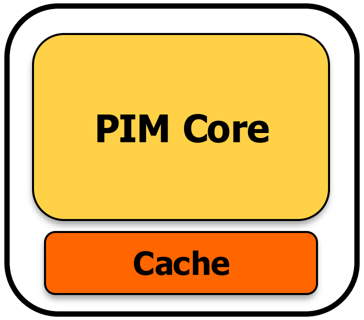
- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu,
  "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"
  *Proceedings of the 23rd International 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

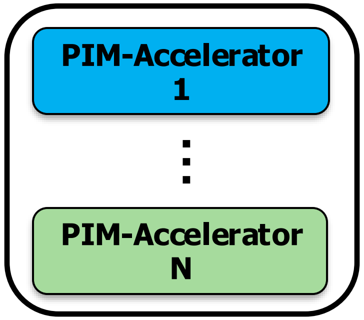
Amirali Boroumand[1] Saugata Ghose[1] Youngsok Kim[2]
Rachata Ausavarungnirun[1] Eric Shiu[3] Rahul Thakur[3] Daehyun Kim[4,3]
Aki Kuusela[3] Allan Knies[3] Parthasarathy Ranganathan[3] Onur Mutlu[5,1]

Possible PNM Designs

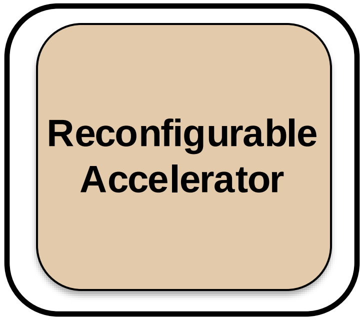
- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM



- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM

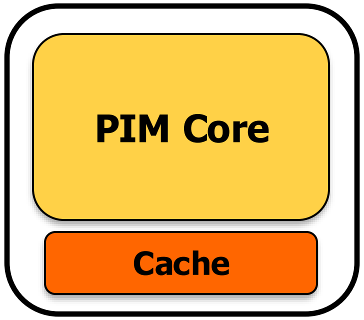


- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM



Possible PNM Designs

- **General-purpose programmable cores**
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM



- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM

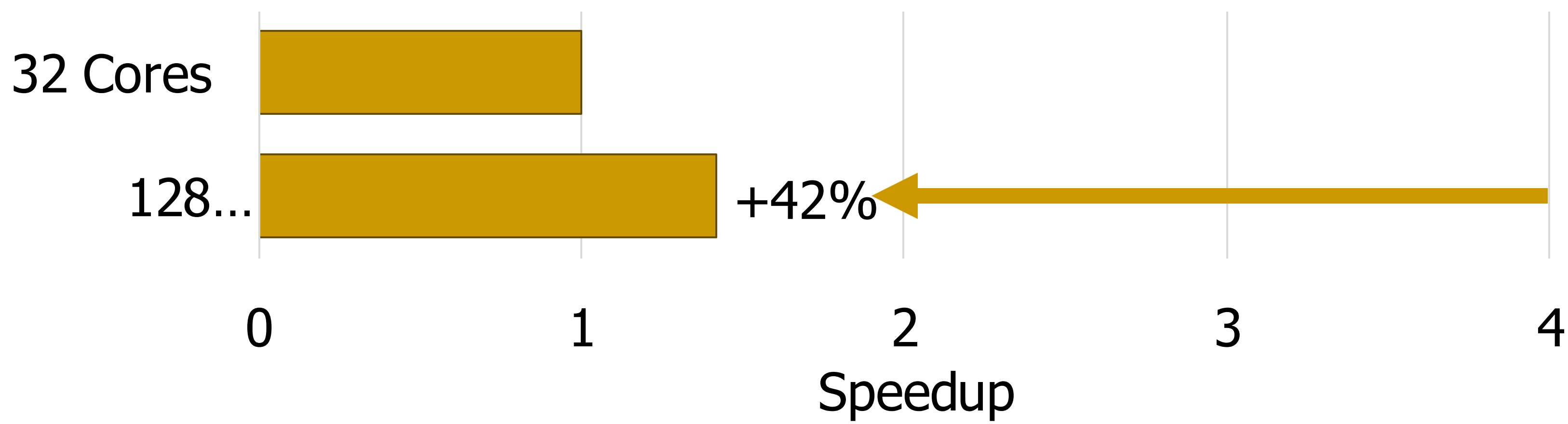
- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

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

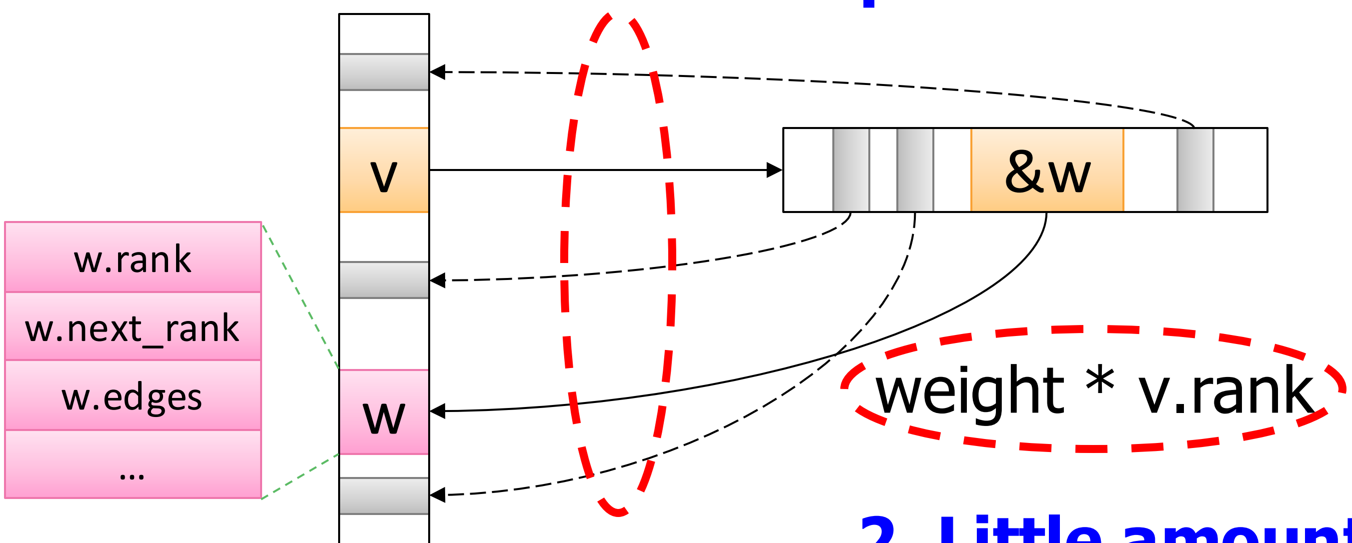
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

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

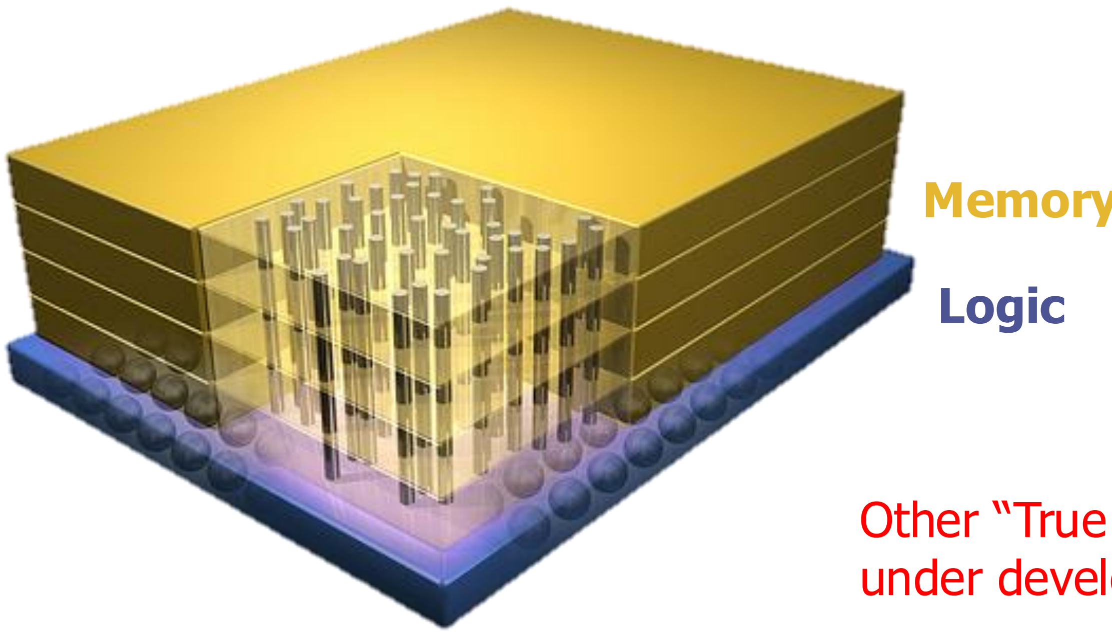
1. Frequent random memory accesses



2. Little amount of computation

Opportunity: 3D-Stacked Logic+Memory

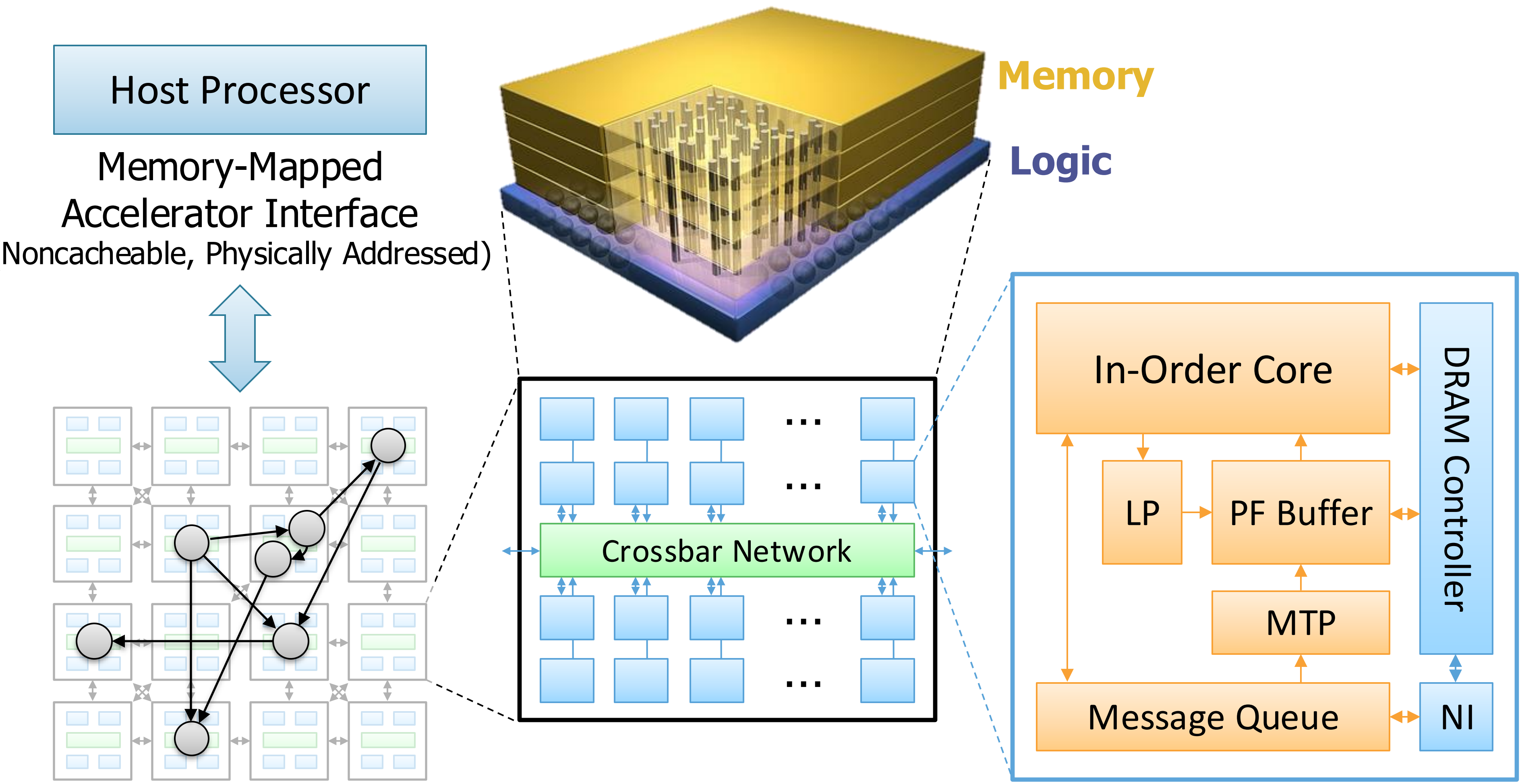
Hybrid Memory Cube
CONSORTIUM



Other "True 3D" technologies
under development

Tesseract System for Graph Processing

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



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

More on Tesseract

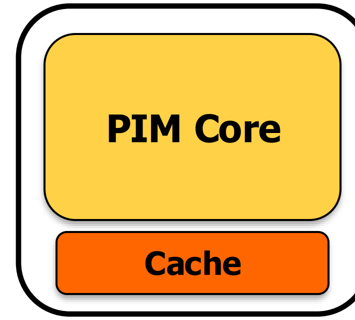
- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
"A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
*Proceedings of the 42nd International Symposium on Computer Architecture (**ISCA**)*, Portland, OR, June 2015.
[Slides (pptx) (pdf)] [Lightning Session 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

Possible PNM Designs

- **General-purpose programmable cores**
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM

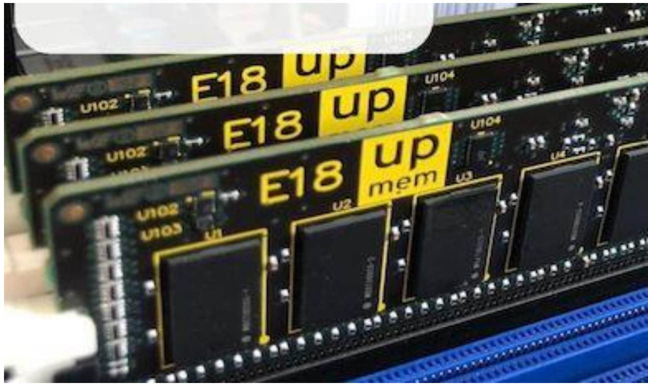


- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM

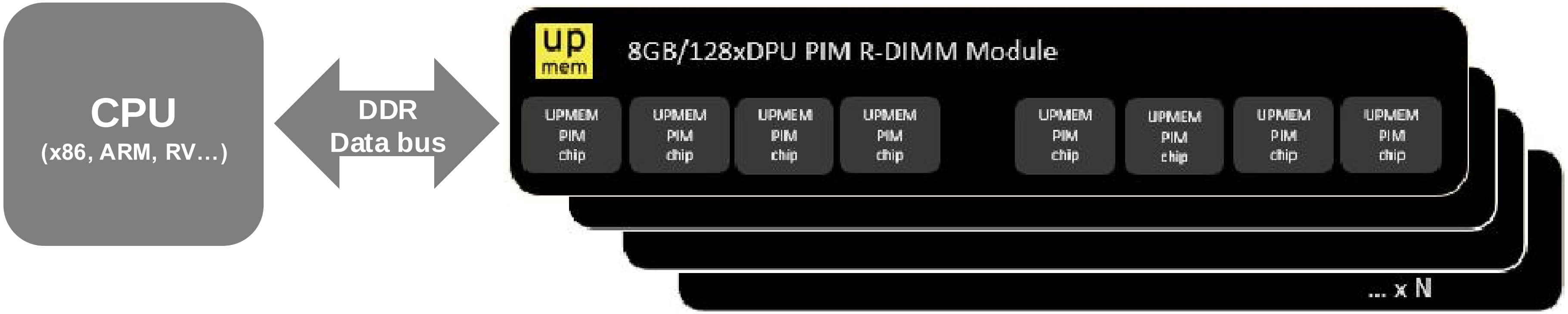
- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

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



https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

Accelerator Model (I)

- UPMEM DIMMs coexist with conventional DIMMs

- Integration of UPMEM DIMMs in a system follows an accelerator model

- UPMEM DIMMs can be seen as a loosely coupled accelerator
  - Explicit data movement between the main processor (host CPU) and the accelerator (UPMEM)
  - Explicit kernel launch onto the UPMEM processors

- This resembles GPU computing

System Organization (I)

- *FIG. 1 schematically illustrates a computing system comprising DRAM circuits having integrated processors according to an example embodiment*

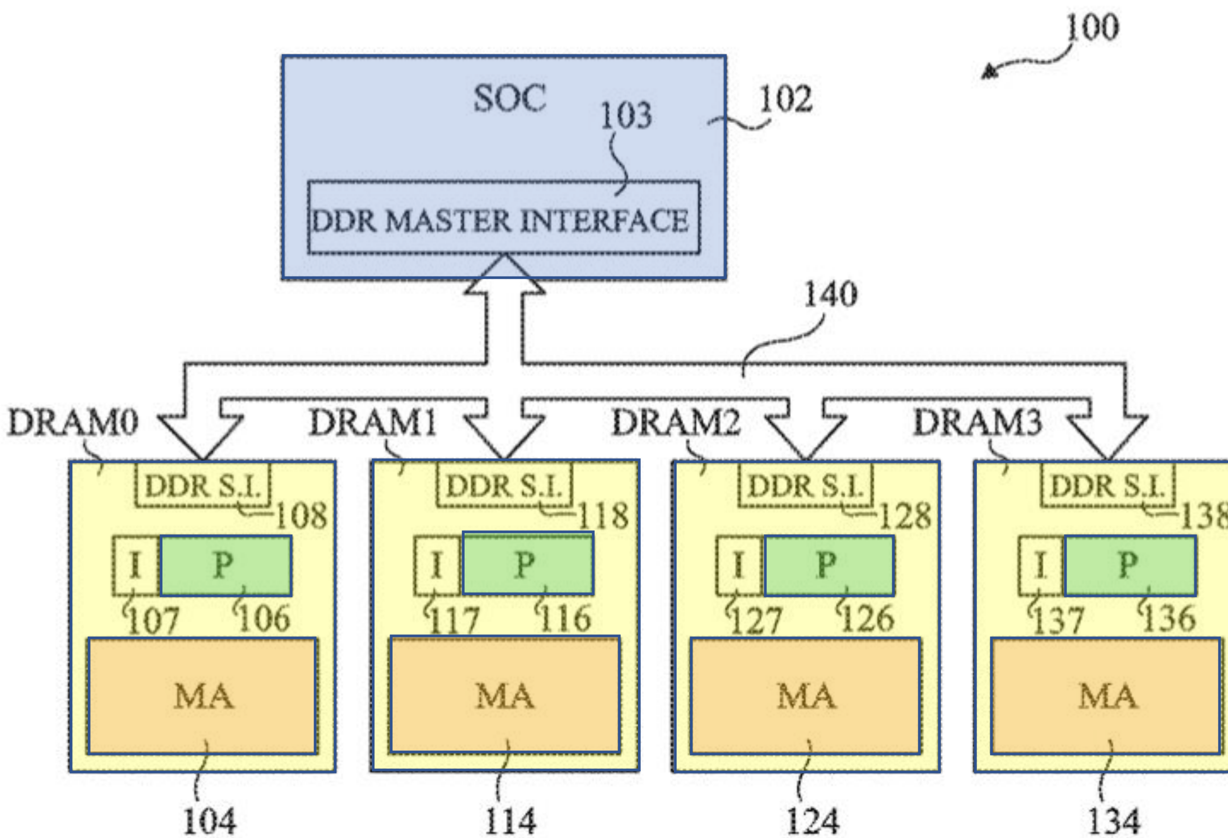
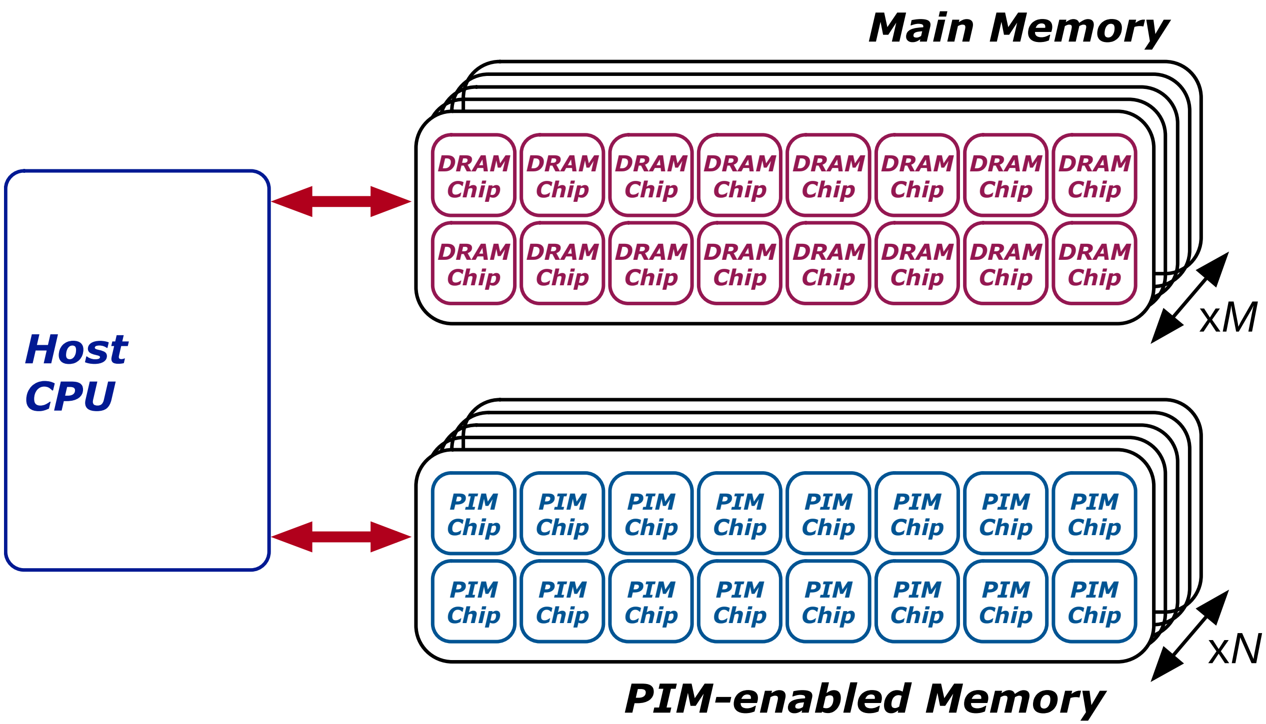


Fig 1

Fabrice Devaux, Jean-François Roy. "Memory circuit with integrated processor." US 10,324,870 B2.

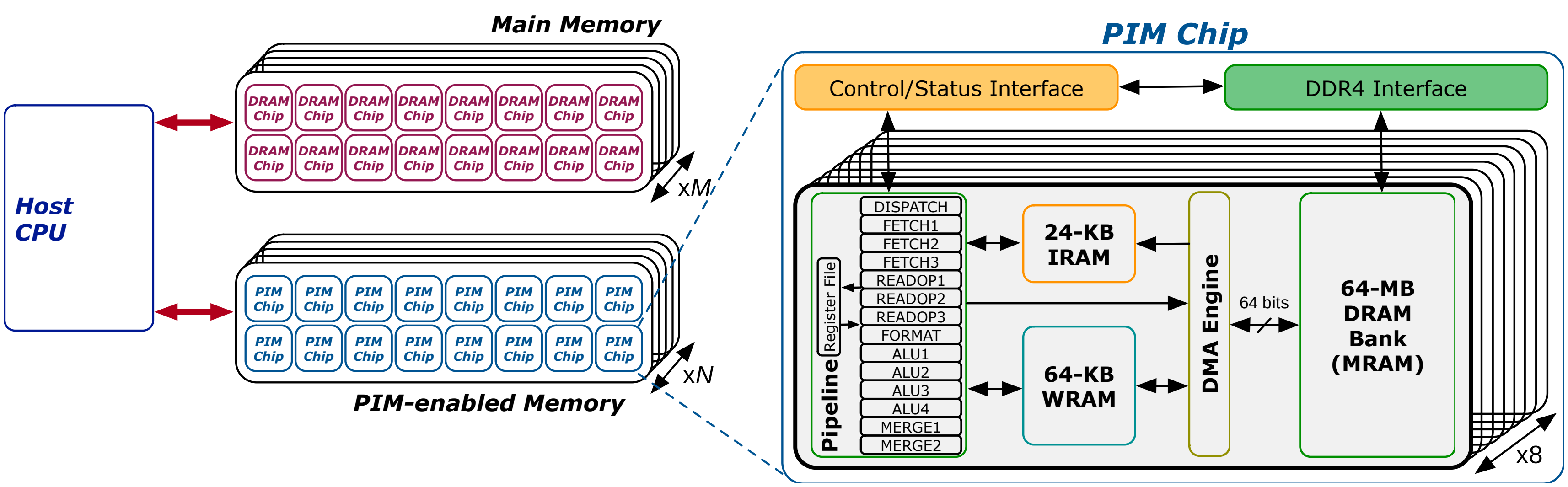
System Organization (II)

- In a UPMEM-based PIM system UPMEM DIMMs coexist with regular DDR4 DIMMs

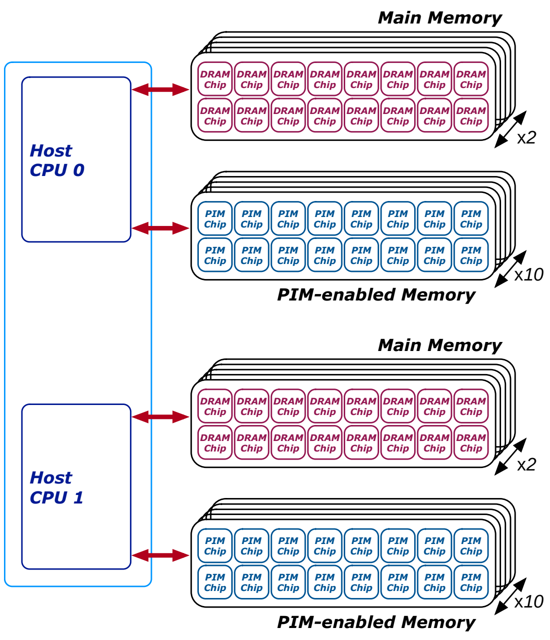


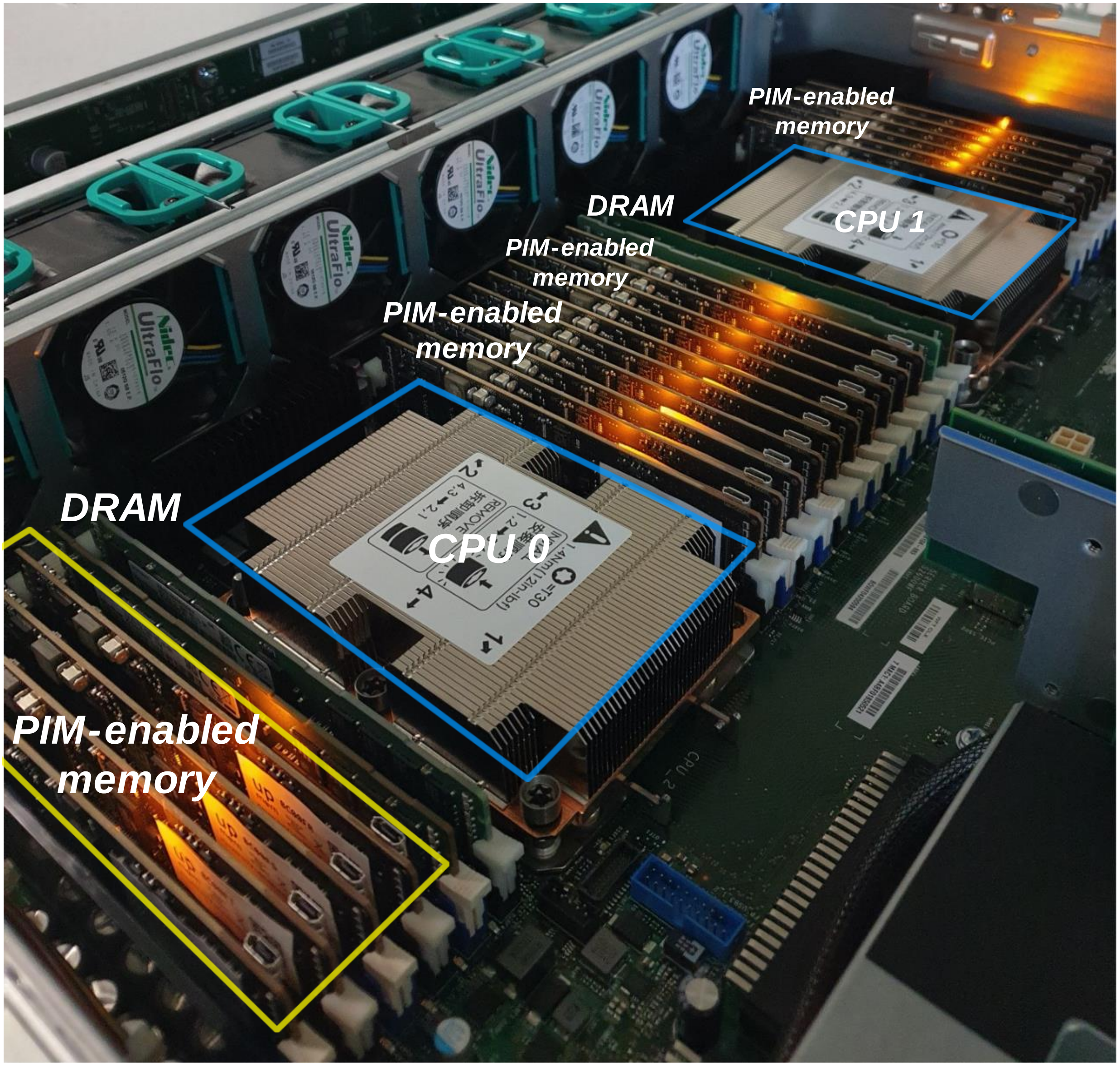
System Organization (III)

- A UPMEM DIMM contains 8 or 16 chips
  - Thus, 1 or 2 ranks of 8 chips each
- Inside each PIM chip there are:
  - 8 64MB banks per chip: Main RAM (MRAM) banks
  - 8 DRAM Processing Units (DPUs) in each chip, 64 DPUs per rank



2,560-DPU System (II)





DRAM Processing Unit (I)

- *FIG. 4 schematically illustrates part of the computing system of FIG. 1 in more detail according to an example embodiment*

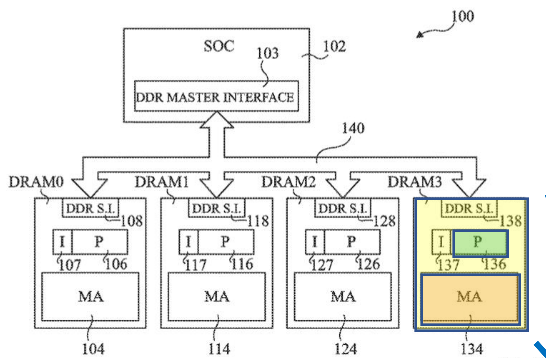


Fig 1

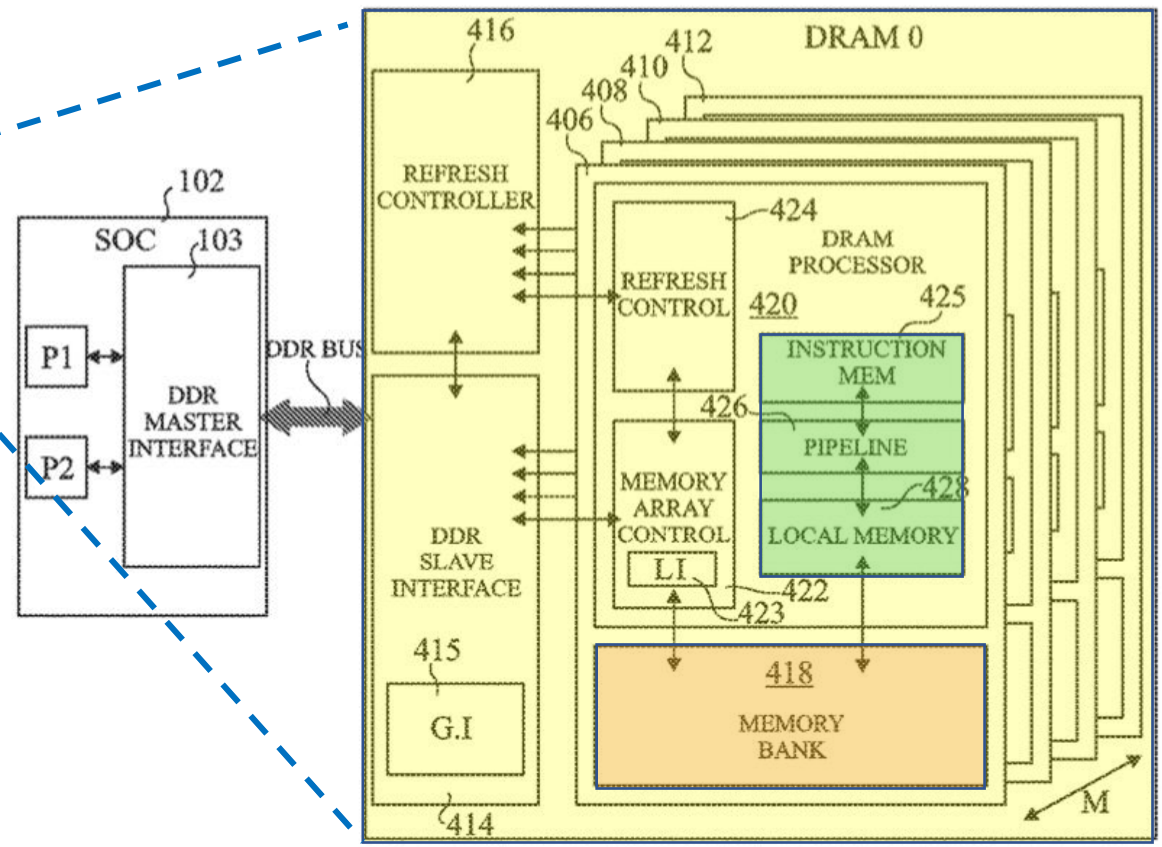
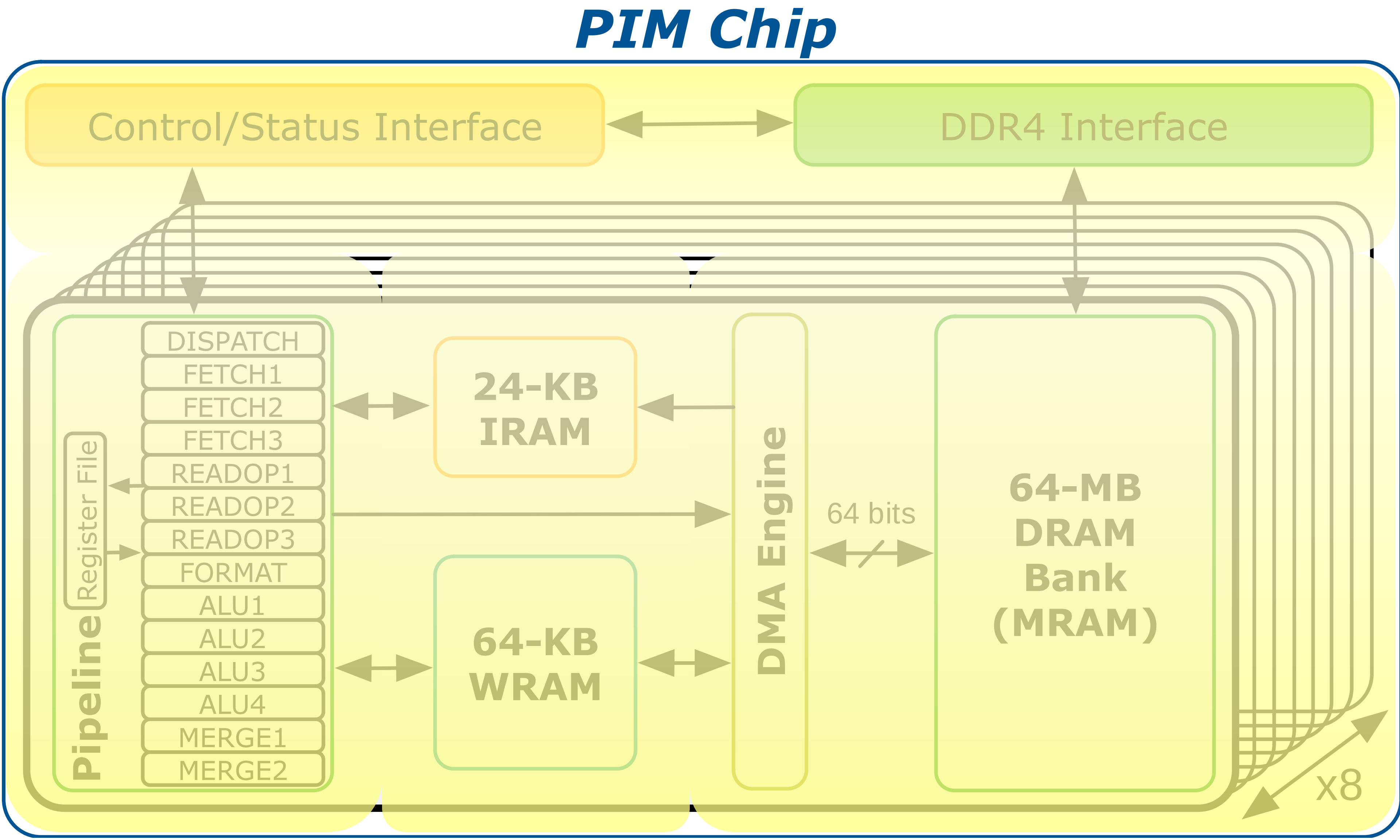


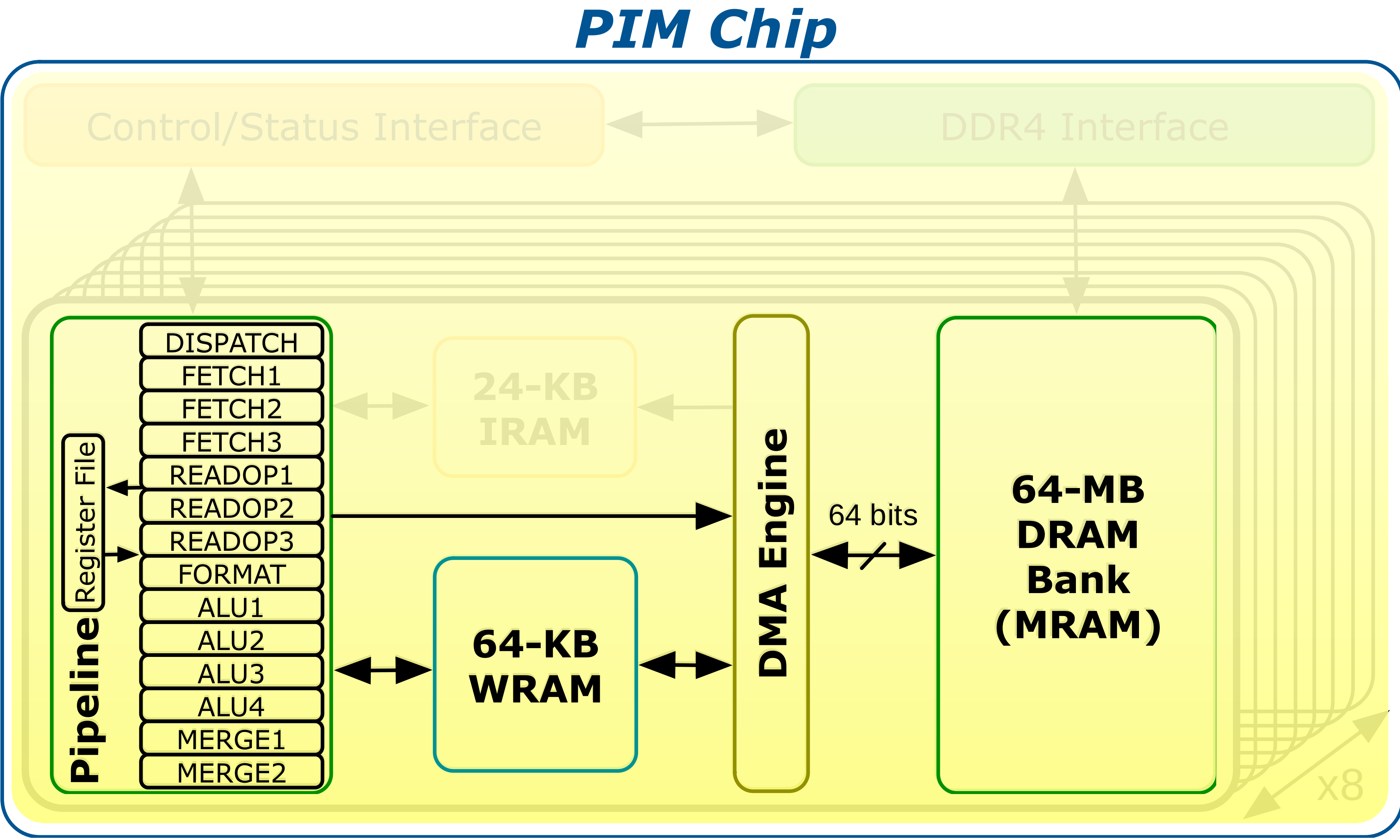
Fig 4

Fabrice Devaux, Jean-François Roy. "Memory circuit with integrated processor." US 10,324,870 B2.

DRAM Processing Unit (II)

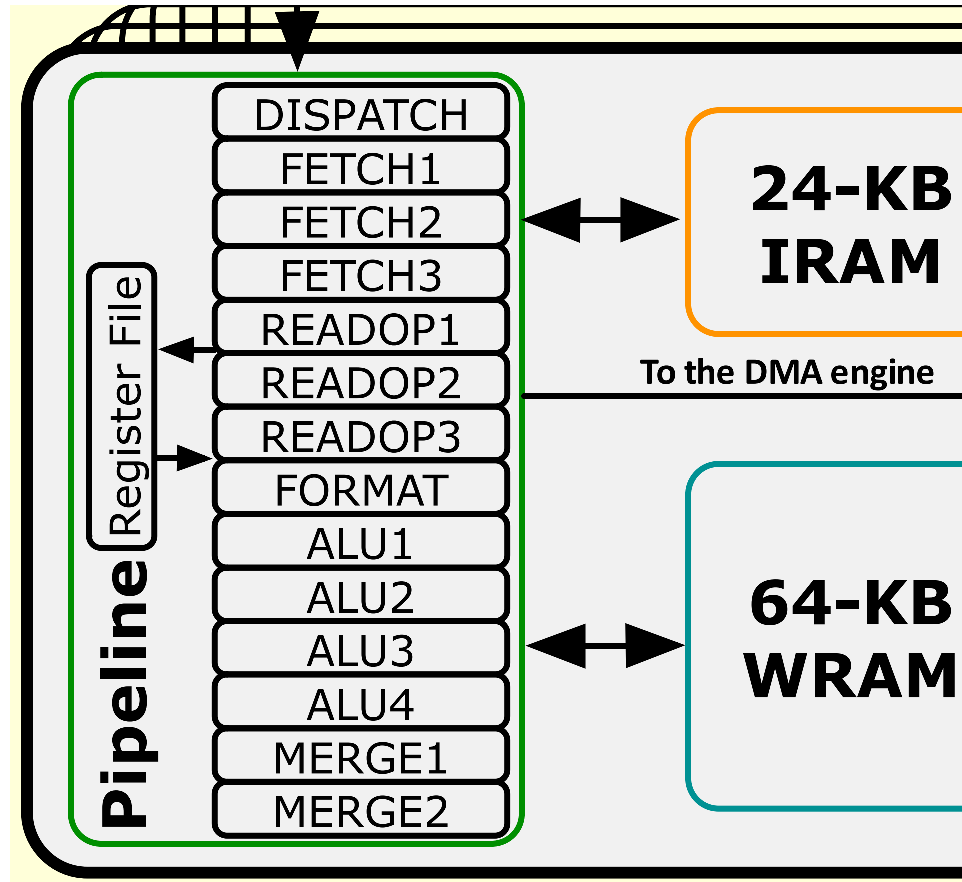


DPU: Arithmetic Throughput vs. Operational Intensity



DPU Pipeline

- In-order pipeline
  - Up to 425 MHz
- Fine-grain multithreaded
  - 24 hardware threads
- 14 pipeline stages
  - DISPATCH: Thread selection
  - FETCH: Instruction fetch
  - READOP: Register file
  - FORMAT: Operand formatting
  - ALU: Operation and WRAM
  - MERGE: Result formatting



DPU Instruction Set Architecture

- Specific 32-bit ISA
  - Aiming at scalar, in-order, and multithreaded implementation
  - Allowing compilation of 64-bit C code
  - LLVM/Clang compiler

Instruction Set Architecture — UPMEM DPU SDK 2021.2.0 Documentation

UPMEM development tools documentation

» Instruction Set Architecture   View page source

Instruction Set Architecture

This section covers the architecture concepts required to understand and use UPMEM DPU processor as a software developer. It is also providing an exhaustive list of the available processor instructions.

Software developers should use this section as a reference manual to develop or debug assembly code.

Resources overview

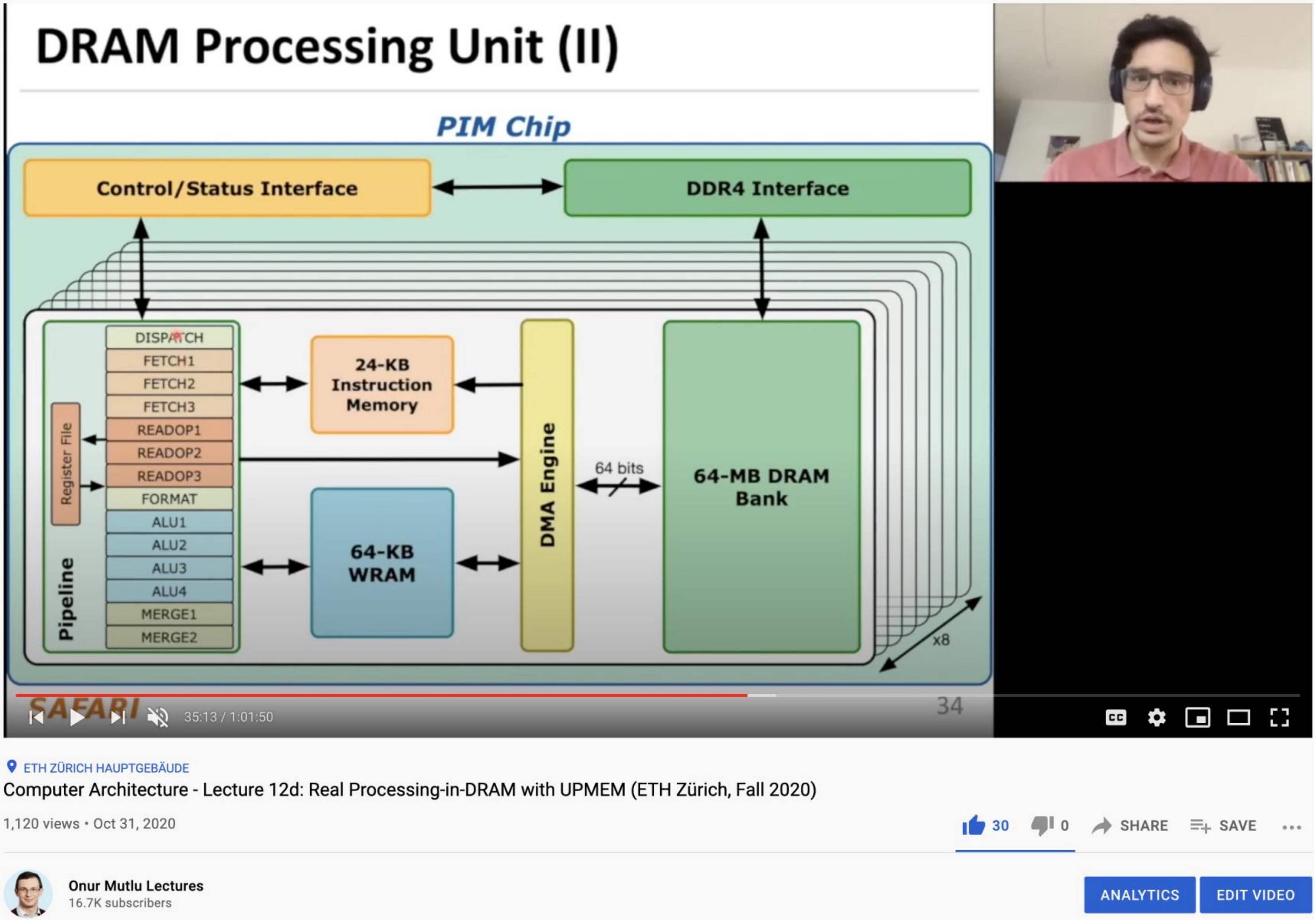
Thread registers

The system is composed of 24 hardware threads. Each of them owns a set of private resources:

- 24 general purpose 32-bits registers named `r0` through `r23`
- A 16-bits wide program counter, named PC. Notice that the PC value does not address an instruction in memory, but the index of such an instruction directly. For example, a PC equal to 1 represents the second instruction in the DPU's program memory.
- Two persistent flags, keeping information about the previous result of an arithmetic or logical instruction:
  - ZF: last result is equal to zero

https://sdk.upmem.com/2021.2.0/201_IS.html#

More on the UPMEM PIM System



https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26

Experimental Analysis of the UPMEM PIM Engine

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJJ, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA 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 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

Recent SRC TECHCON Presentation

- Dr. Juan Gomez-Luna
  - Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
  - Based on two major works
    - https://arxiv.org/pdf/2105.03814.pdf
    - https://arxiv.org/pdf/2207.07886.pdf

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

Year: 2021, Pages: 1-7

DOI Bookmark: 10.1109/IGSC54211.2021.9651614

Authors

Juan Gómez-Luna, ETH Zürich

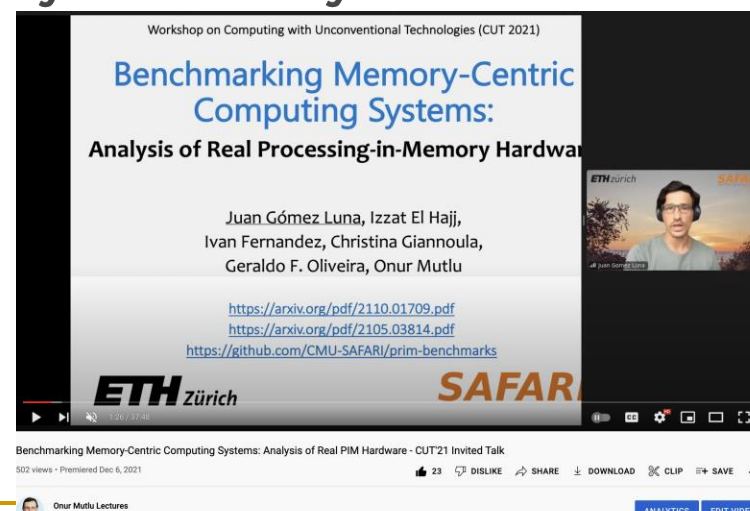
Izzat El Hajj, American University of Beirut

Ivan Fernandez, University of Malaga

Christina Giannoula, National Technical University of Athens

Geraldo F. Oliveira, ETH Zürich

Onur Mutlu, ETH Zürich



UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
  "Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware"
  *Invited Paper at Workshop on Computing with Unconventional Technologies (**CUT**)*, Virtual, October 2021.
  [arXiv version]
  [PrIM Benchmarks Source Code]
  [Slides (pptx) (pdf)]
  [Talk Video (37 minutes)]
  [Lightning Talk Video (3 minutes)]

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

Juan Gómez-Luna
ETH Zürich

Izzat El Hajj
American University of Beirut

Ivan Fernandez
University of Malaga

Christina Giannoula
National Technical University of Athens

Geraldo F. Oliveira
ETH Zürich

Onur Mutlu
ETH Zürich

Understanding a Modern PIM Architecture

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

JUAN GÓMEZ-LUNA[1], IZZAT EL HAJJ[2], IVAN FERNANDEZ[1,3], CHRISTINA GIANNOULA[1,4], GERALDO F. OLIVEIRA[1], AND ONUR MUTLU[1]

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

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

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

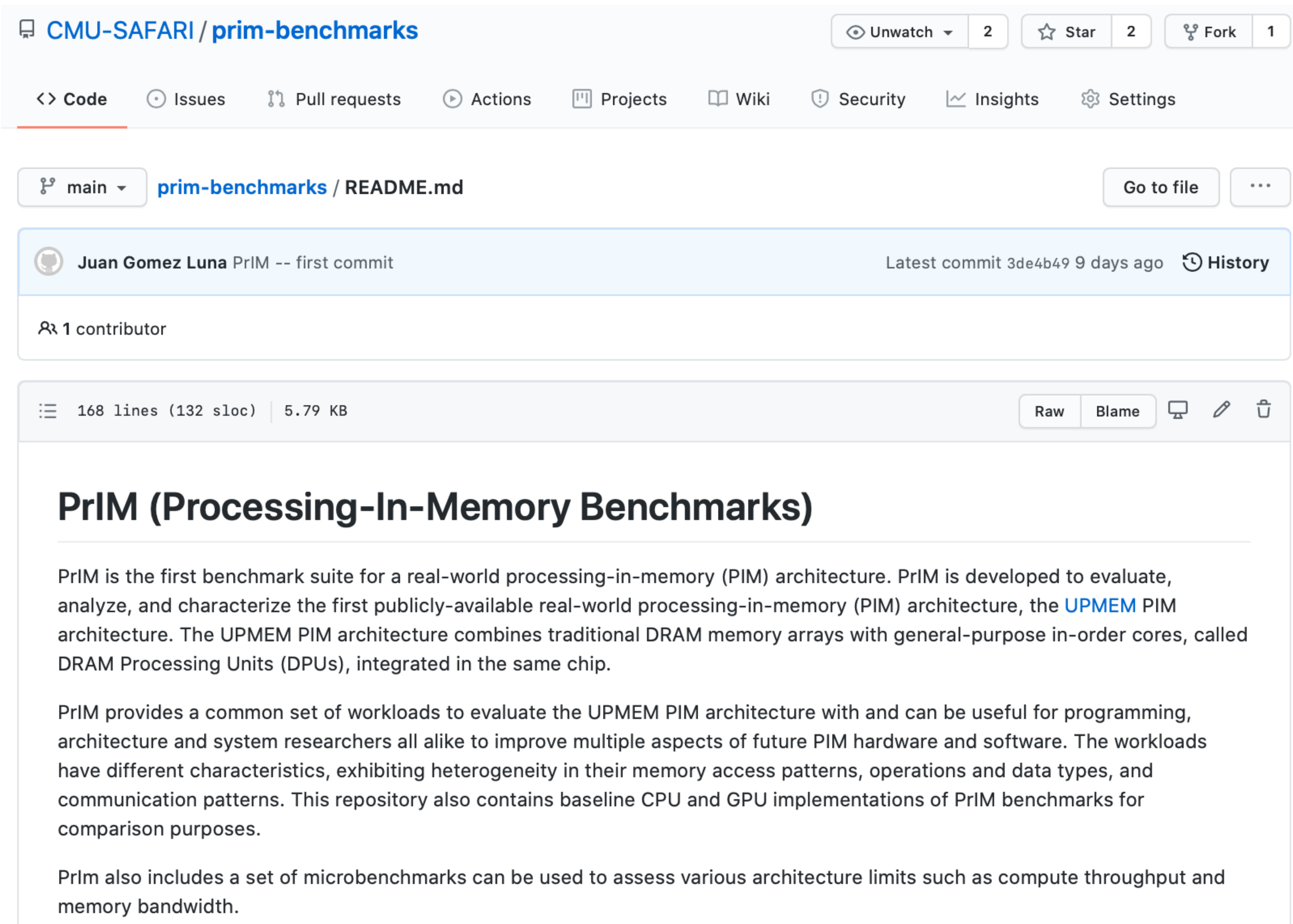
https://github.com/CMU-SAFARI/prim-benchmarks

PrIM Benchmarks: Application Domains

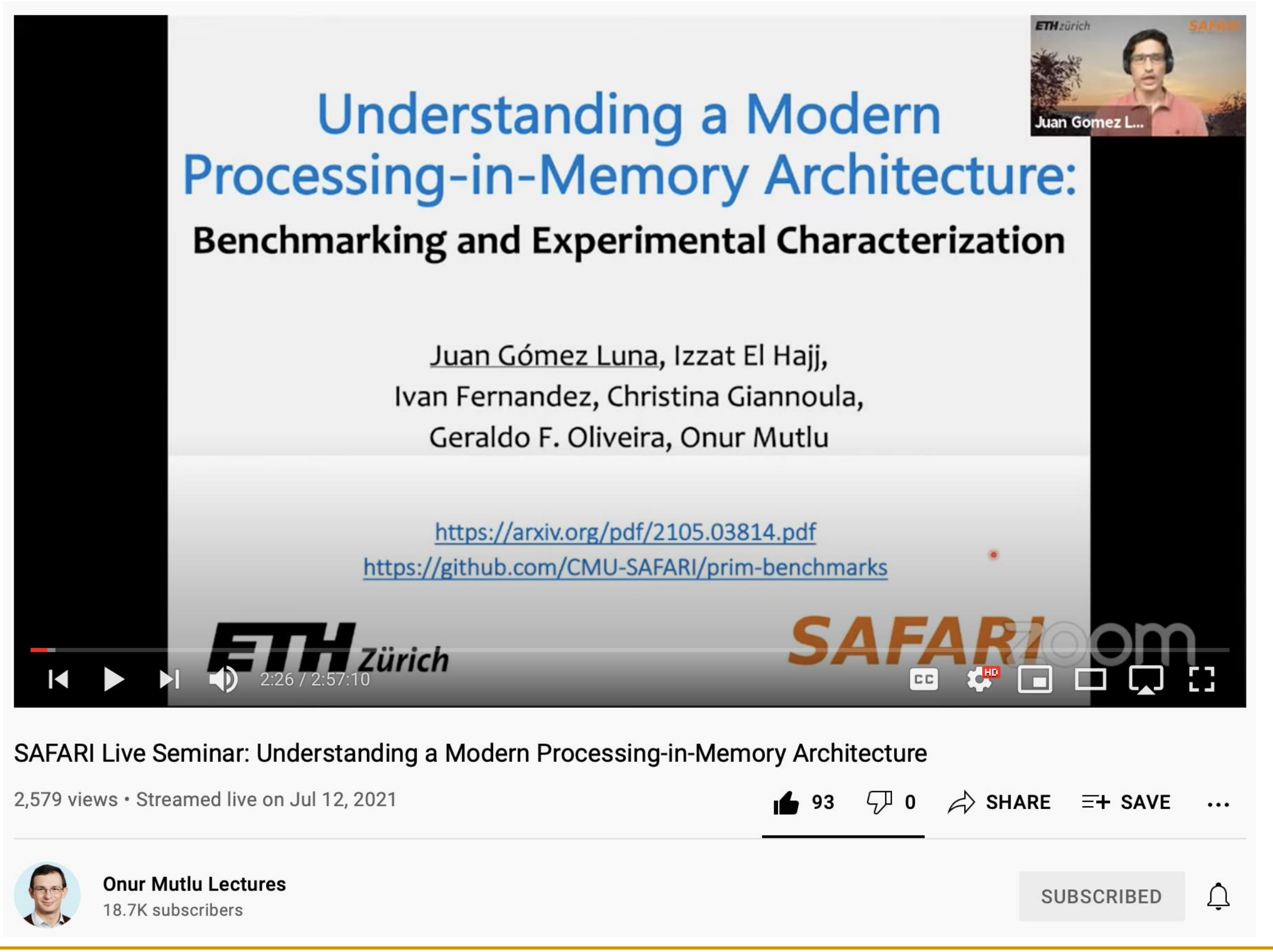
Domain	Benchmark	Short name
Dense linear algebra	Vector Addition	VA
	Matrix-Vector Multiply	GEMV
Sparse linear algebra	Sparse Matrix-Vector Multiply	SpMV
Databases	Select	SEL
	Unique	UNI
Data analytics	Binary Search	BS
	Time Series Analysis	TS
Graph processing	Breadth-First Search	BFS
Neural networks	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

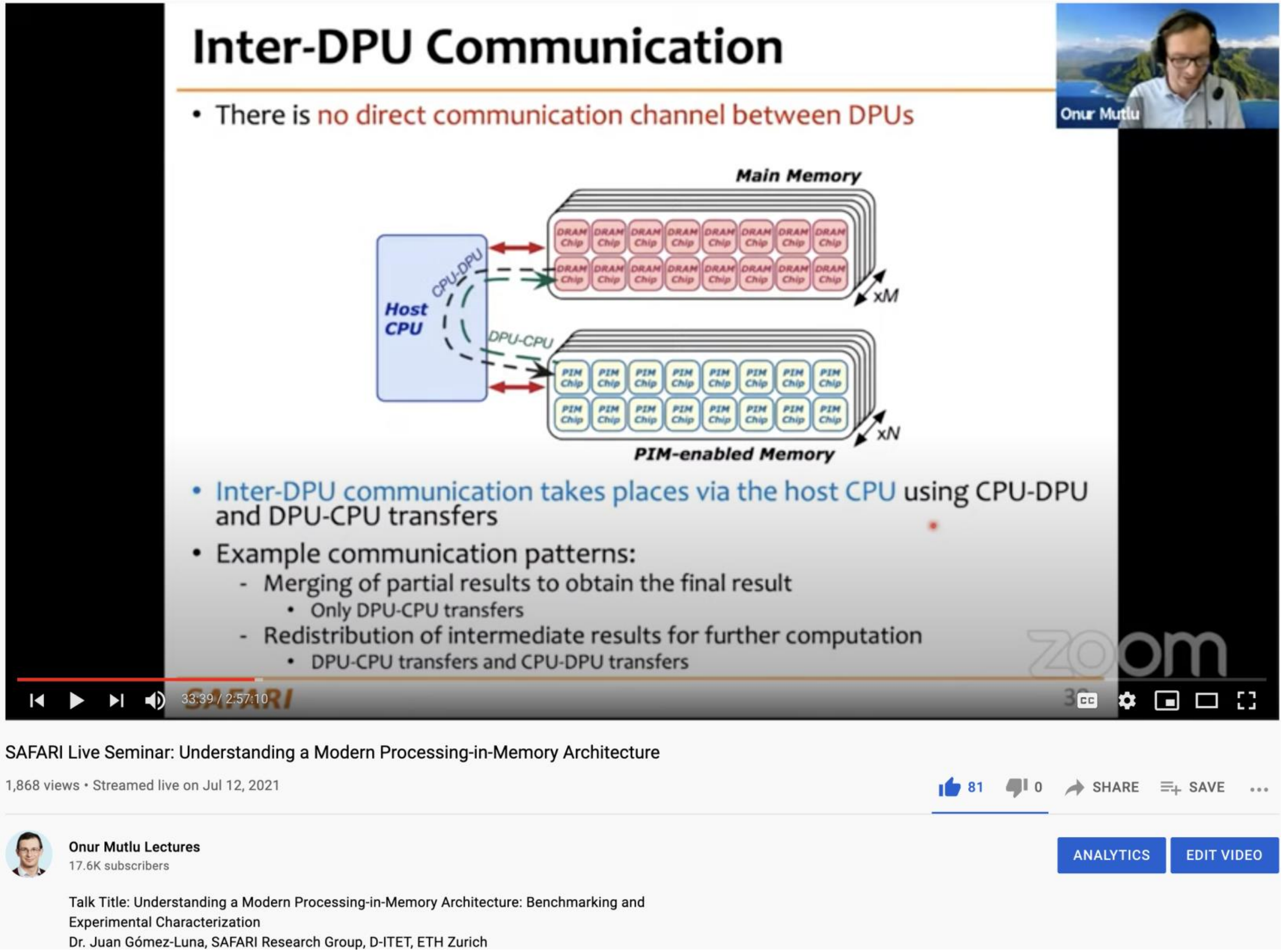


Understanding a Modern PIM Architecture

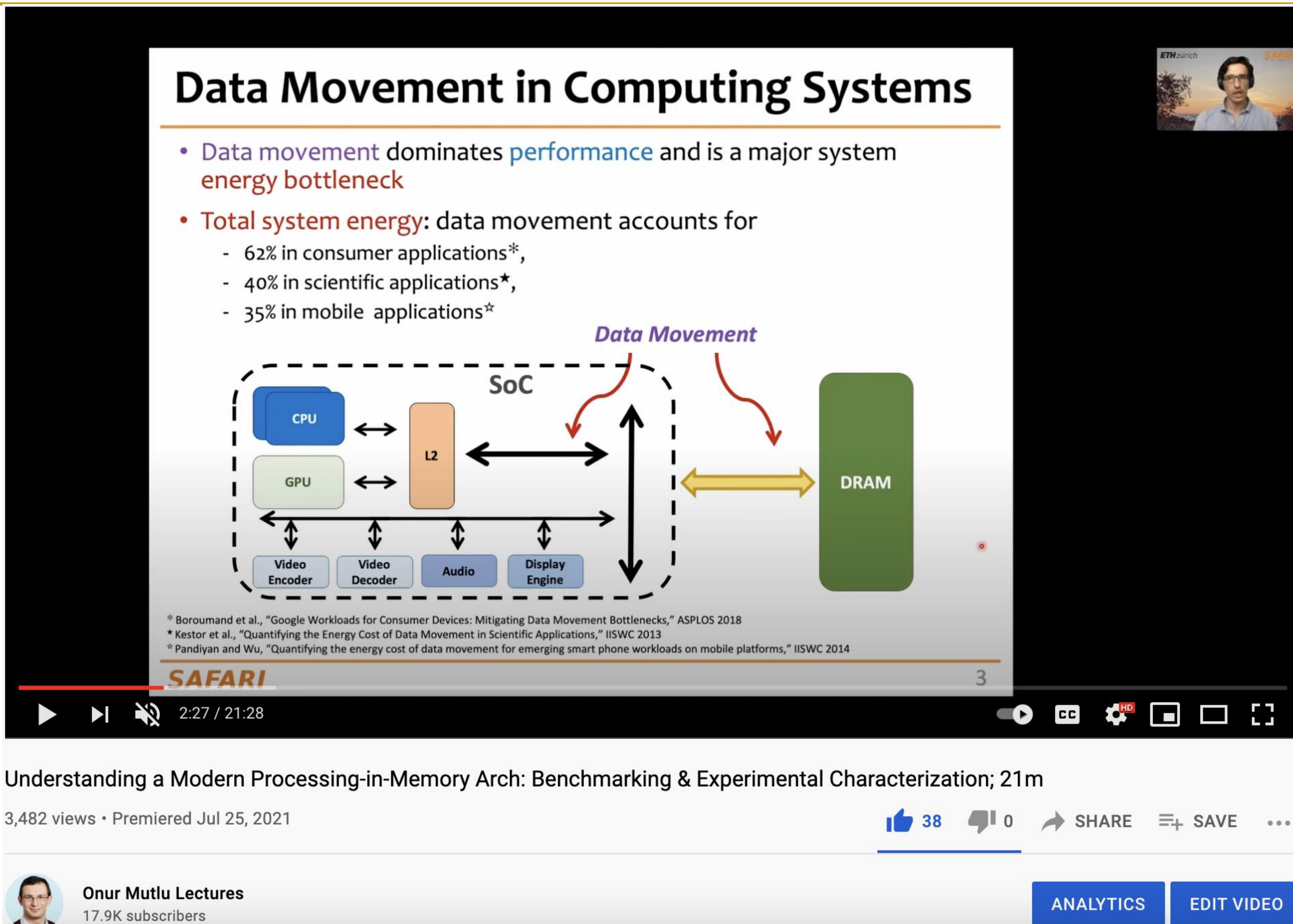


https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9

More on Analysis of the UPMEM PIM Engine



More on Analysis of the UPMEM PIM Engine



https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159

ML Training on Real PIM Systems

- Juan Gómez Luna, Yuxin Guo, Sylvan Brocard, Julien Legriel, Remy Cimadomo, Geraldo F. Oliveira, Gagandeep Singh, and Onur Mutlu,
"Evaluating Machine Learning Workloads on Memory-Centric Computing Systems"
*Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Raleigh, North Carolina, USA, April 2023.
[arXiv version, 16 July 2022.]
[PIM-ML Source Code]
Best paper session.

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

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]

[1]ETH Zürich [2]UPMEM

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

https://arxiv.org/pdf/2207.07886.pdf

ML Training on a Real PIM System

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

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]

[1]ETH Zürich [2]UPMEM

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

Juan Gómez-Luna[1] Yuxin Guo[1] Sylvan Brocard[2] Julien Legriel[2]
Remy Cimadomo[2] Geraldo F. Oliveira[1] Gagandeep Singh[1] Onur Mutlu[1]

[1]ETH Zürich [2]UPMEM

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

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

SpMV Multiplication on Real PIM Systems

- Appears at 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

https://www.youtube.com/watch?v=5kaOsJKlGrE

Transcendental Functions on Real PIM Systems

- Maurus Item, Juan Gómez Luna, Yuxin Guo, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu,
  "TransPimLib: Efficient Transcendental Functions for Processing-in-Memory Systems"
  *Proceedings of the 2023 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Raleigh, North Carolina, USA, April 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [TransPimLib Source Code]
  [Talk Video (17 minutes)]

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

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

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

https://arxiv.org/pdf/2304.01951.pdf

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 Diab [1] Amir Nassereldine [1] Mohammed Alser [2] Juan Gómez Luna [2]
Onur Mutlu [2] Izzat El Hajj [1]

[1] American University of Beirut [2] ETH Zürich

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

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

Homomorphic Operations on Real PIM Systems

- Harshita Gupta, Mayank Kabra, Juan Gómez-Luna, Konstantinos Kanellopoulos, and Onur Mutlu,
  "Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System"
  *Proceedings of the 2023 IEEE International Symposium on Workload Characterization Poster Session (**IISWC**)*, Ghent, Belgium, October 2023.
  [arXiv version]
  [Lightning Talk Slides (pptx) (pdf)]
  [Poster (pptx) (pdf)]

Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra* Juan Gómez-Luna Konstantinos Kanellopoulos Onur Mutlu
ETH Zürich

Accelerating Reinforcement Learning

- Kailash Gogineni, Sai Santosh Dayapule, Juan Gomez-Luna, Karthikeya Gogineni, Peng Wei, Tian Lan, Mohammad Sadrosadati, Onur Mutlu, Guru Venkataramani,
  "SwiftRL: Towards Efficient Reinforcement Learning on Real Processing-In-Memory Systems"
  *Proceedings of the 2024 IEEE International Symposium on Performance Analysis of Systems and Software (**ISPASS**)*, Indianapolis, Indiana, May 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]

SwiftRL: Towards Efficient Reinforcement Learning on Real Processing-In-Memory Systems

Kailash Gogineni[1] Sai Santosh Dayapule[1] Juan Gómez-Luna[2] Karthikeya Gogineni[3]
Peng Wei[1] Tian Lan[1] Mohammad Sadrosadati[2] Onur Mutlu[2] Guru Venkataramani[1]

[1]George Washington University, USA [2]ETH Zürich, Switzerland [3]Independent

Accelerating ML Training on Real PIM Systems

- Steve Rhyner, Haocong Luo, Juan Gómez-Luna, Mohammad Sadrosadati, Jiawei Jiang, Ataberk Olgun, Harshita Gupta, Ce Zhang, and Onur Mutlu, **"PIM-Opt: Demystifying Distributed Optimization Algorithms on a Real-World Processing-In-Memory System"** *Proceedings of the 33rd International Conference on Parallel Architectures and Compilation Techniques (**PACT**)*, Long Beach, CA, USA, October 2024. [Preliminary arXiv version]

PIM-Opt: Demystifying Distributed Optimization Algorithms on a Real-World Processing-In-Memory System

Steve Rhyner[1] Haocong Luo[1] Juan Gómez-Luna[2] Mohammad Sadrosadati[1]
Jiawei Jiang[3] Ataberk Olgun[1] Harshita Gupta[1] Ce Zhang[4] Onur Mutlu[1]

[1]ETH Zurich [2]NVIDIA [3]Wuhan University [4]University of Chicago

https://arxiv.org/pdf/2404.07164

Accelerating GNNs on Real PIM Systems

- https://arxiv.org/pdf/2402.16731

PyGim : An Efficient Graph Neural Network Library for Real Processing-In-Memory Architectures

CHRISTINA GIANNOULA, University of Toronto, Canada, ETH Zürich, Switzerland, Vector Institute, Canada, and CentML, Canada
PEIMING YANG, University of Toronto, Canada
IVAN FERNANDEZ, Barcelona Supercomputing Center, Spain, Universitat Politècnica de Catalunya, Spain, and ETH Zürich, Switzerland
JIACHENG YANG, University of Toronto, Canada and Vector Institute, Canada
SANKEERTH DURVASULA, University of Toronto, Canada and Vector Institute, Canada
YU XIN LI, University of Toronto, Canada
MOHAMMAD SADROSADATI, ETH Zürich, Switzerland
JUAN GOMEZ LUNA, NVIDIA, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland
GENNADY PEKHIMENKO, University of Toronto, Canada, Vector Institute, Canada, and CentML, Canada

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 Diab [1] Amir Nassereldine [1] Mohammed Alser [2] Juan Gómez Luna [2]
Onur Mutlu [2] Izzat El Hajj [1]

[1]American University of Beirut [2]ETH Zürich

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

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

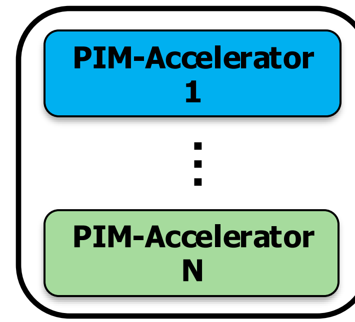
Summary

- 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

Possible PNM Designs

- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM

- **Fixed-function units**
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM



- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

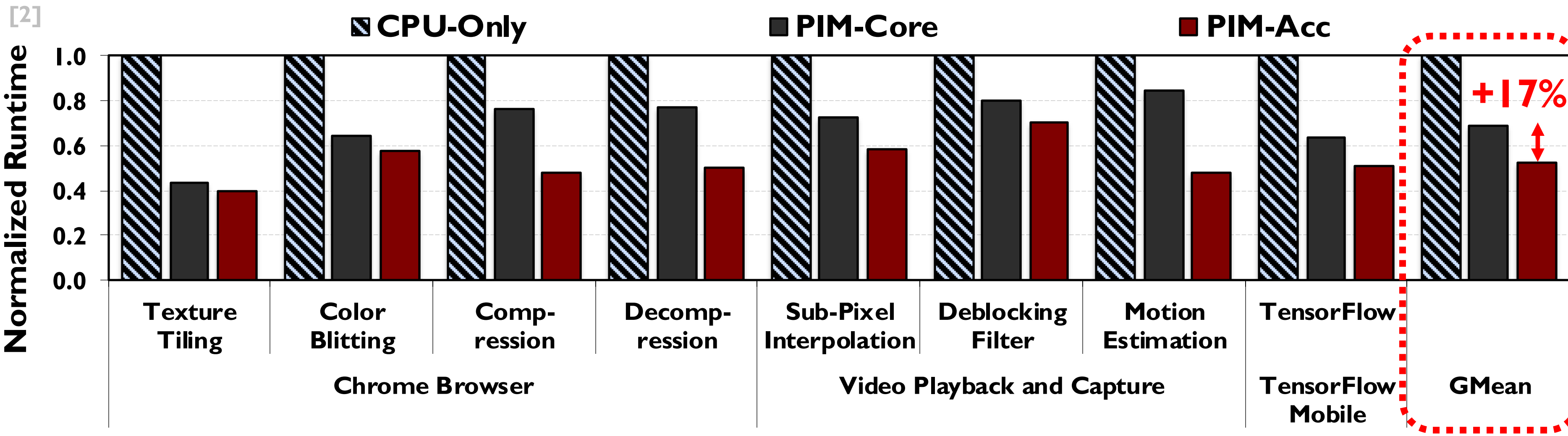
Drawbacks and Limitations of PIM

PIM designs are restricted by low area and power budgets, manufacturing challenges, and limited clock frequencies

↓

To avoid subpar performance, an efficient PIM architecture needs to take into consideration PIM constraints

[2]

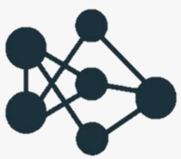


Co-designing hardware and software to take advantage of PIM properties while mitigating its shortcomings can lead to a better system design

HW/SW Co-Design for PIM

We follow a two-step approach to co-design software and hardware to efficiently take advantage of PIM paradigm

Target Application



Step 1:
Application Profiling

- **HW/SW requirements**
- **performance bottleneck**
- **energy bottleneck**

Step 2:
Co-design SW and HW

high-performance and energy-efficient PIM architecture

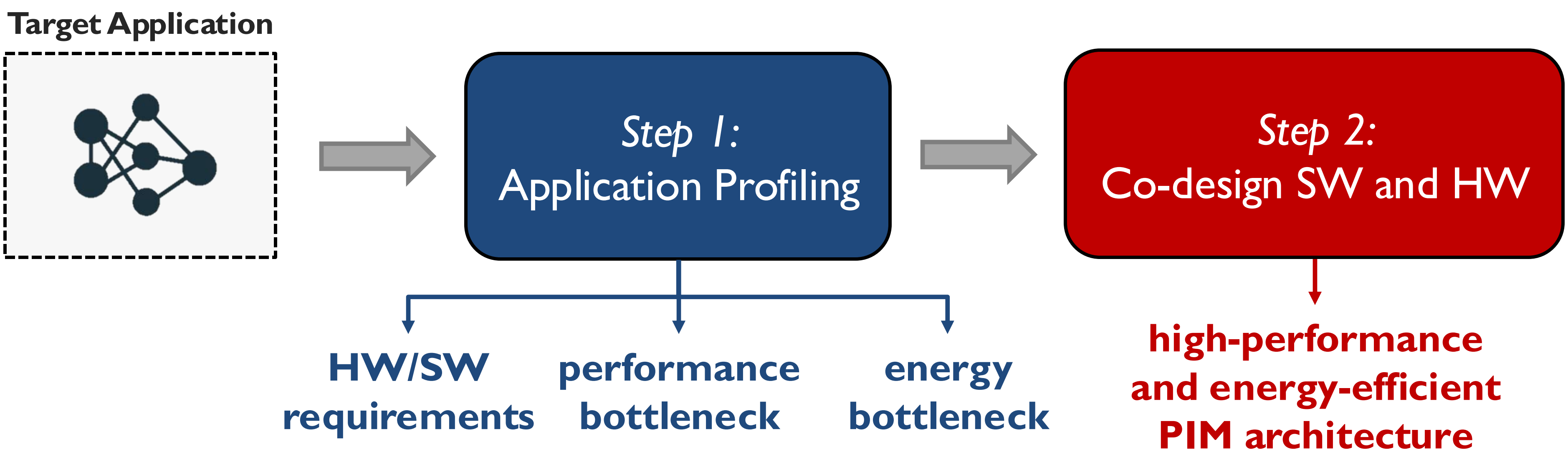
We showcase our two-step approach for several applications:

1 **Machine learning inference models for edge devices**

2 **Genome sequence alignment & filtering**

HW/SW Co-Design for PIM

We follow a two-step approach to co-design software and hardware to efficiently take advantage of PIM paradigm



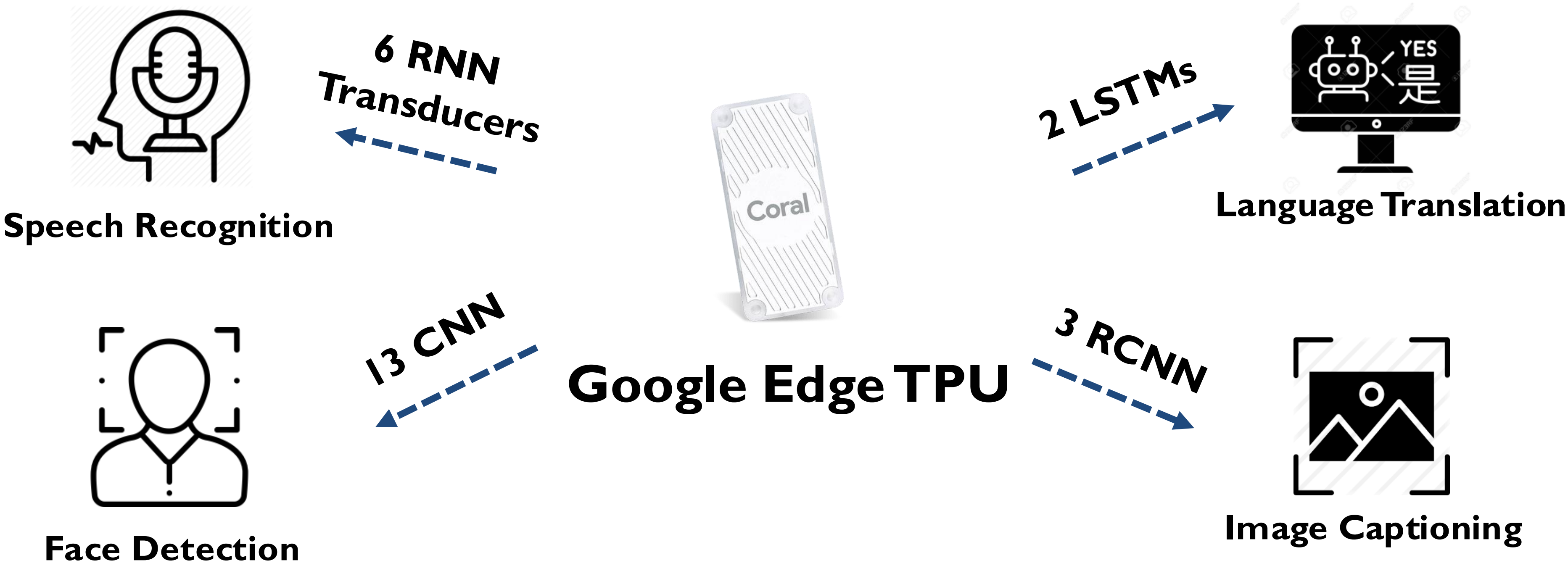
We showcase our two-step approach for several applications:

1 **Machine learning inference models for edge devices**

2 **Genome sequence alignment & filtering**

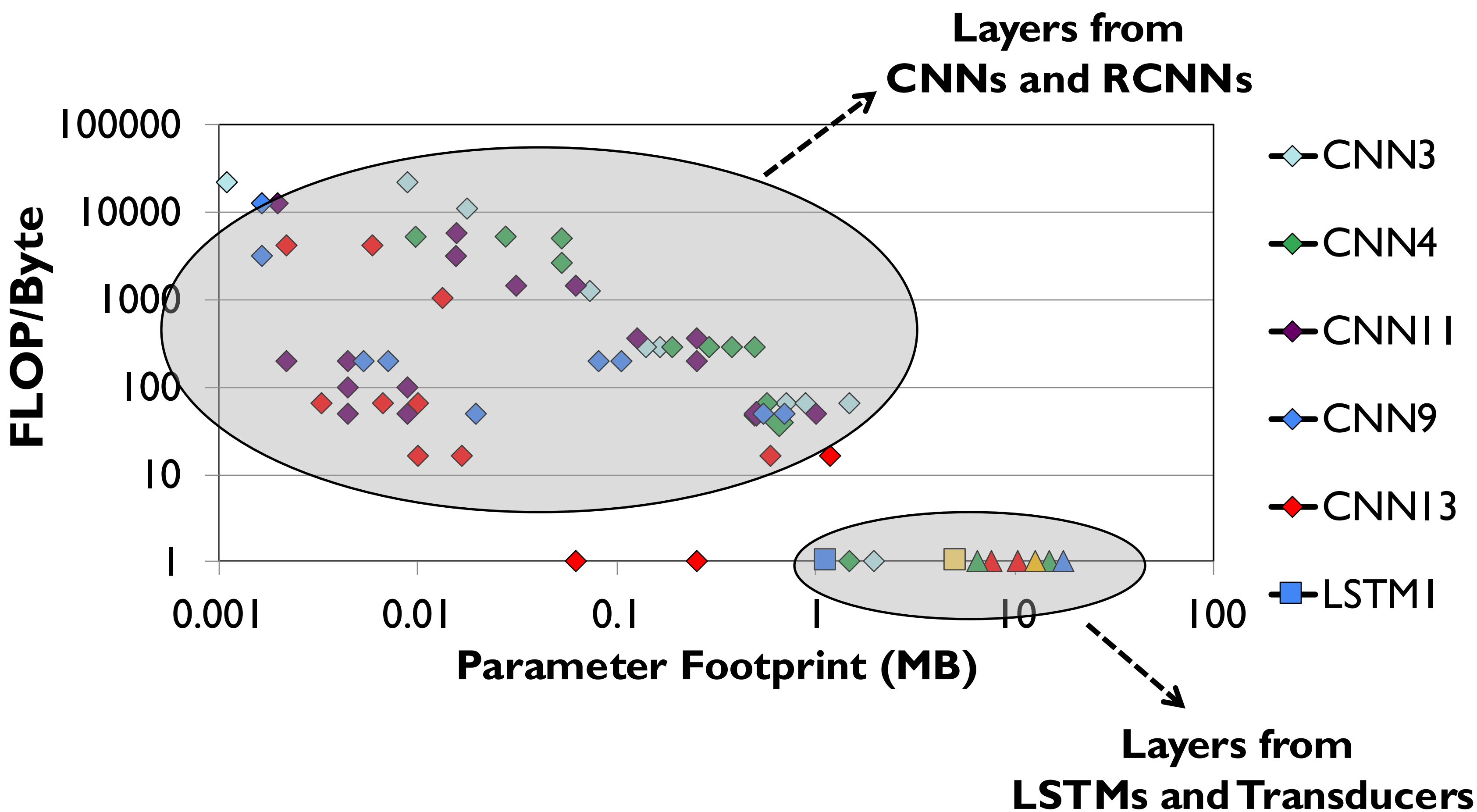
Google Edge Neural Network Models

We analyze inference execution using 24 edge NN models



Diversity Across the Models

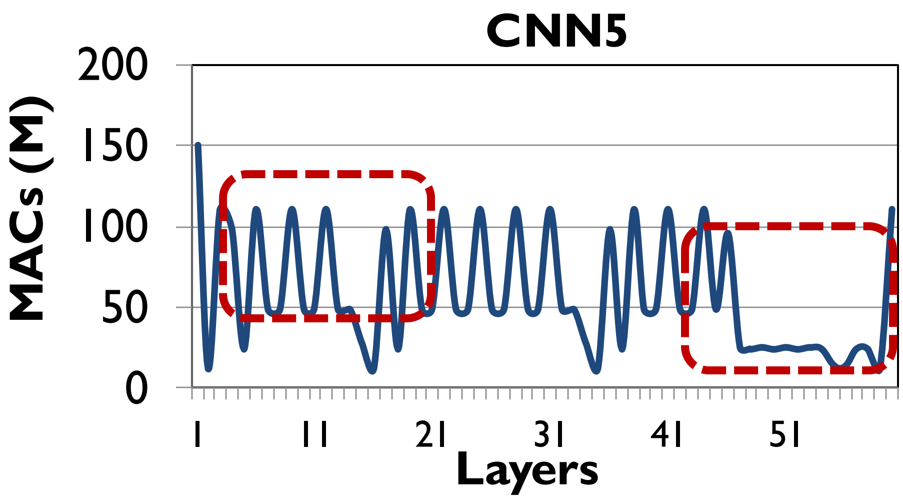
Insight 1: there is significant variation in terms of <u>layer characteristics</u> across the models

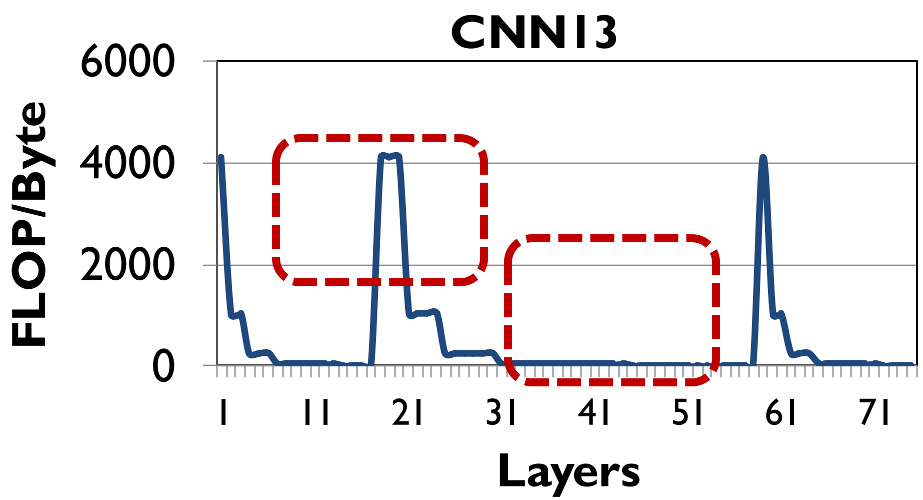


Diversity Within the Models

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

For example, our analysis of edge CNN models shows:

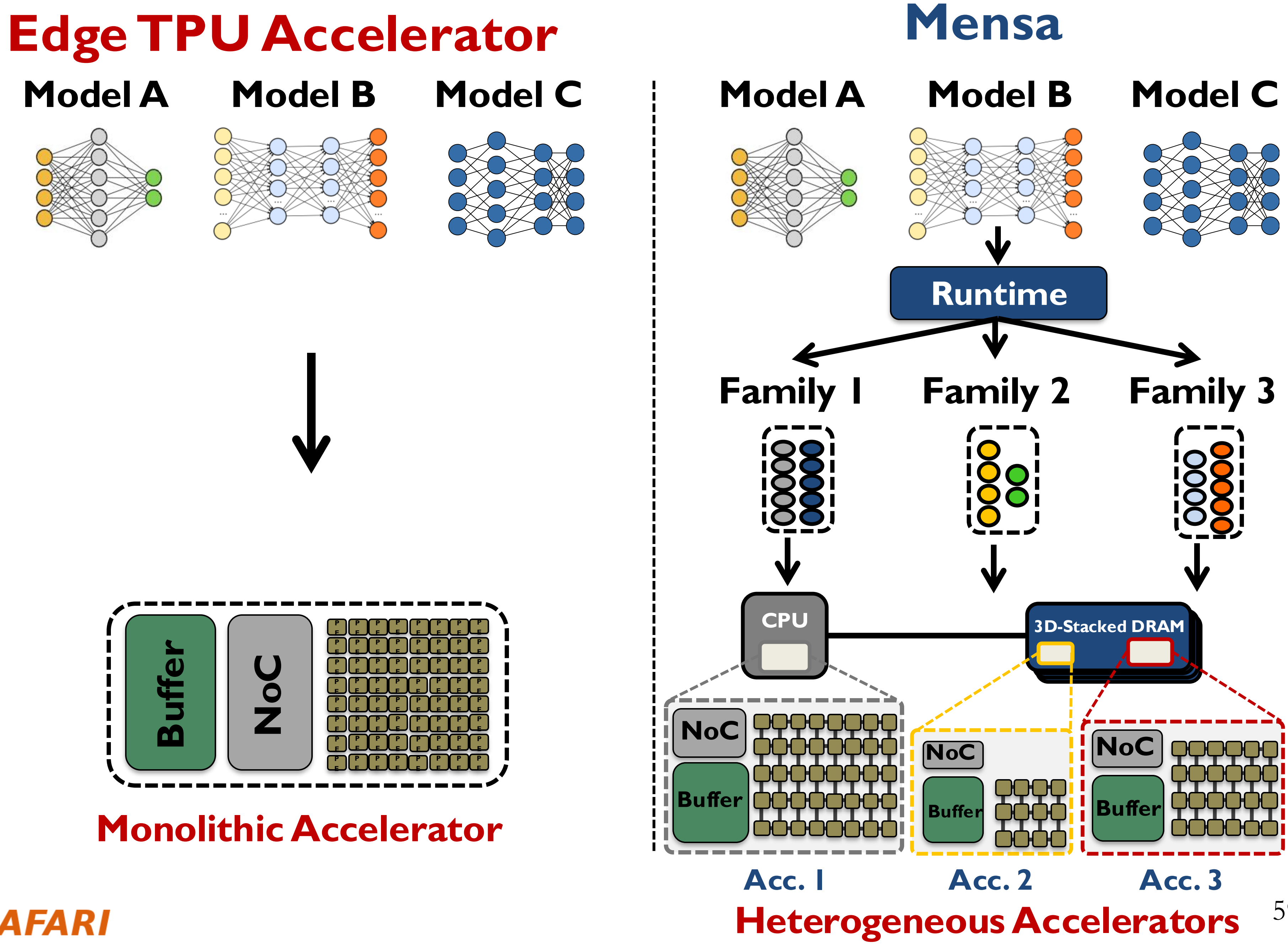




Variation in MAC intensity: up to 200x across layers

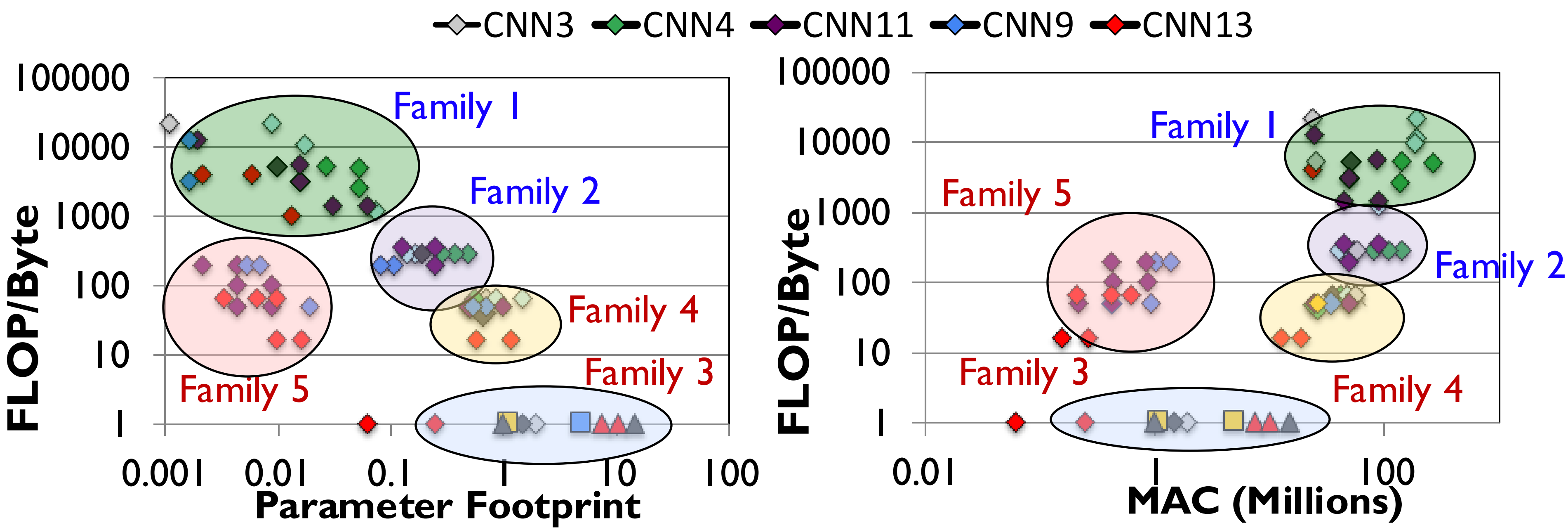
Variation in FLOP/Byte: up to 244x across layers

Mensa High-Level Overview



Identifying Layer Families

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



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

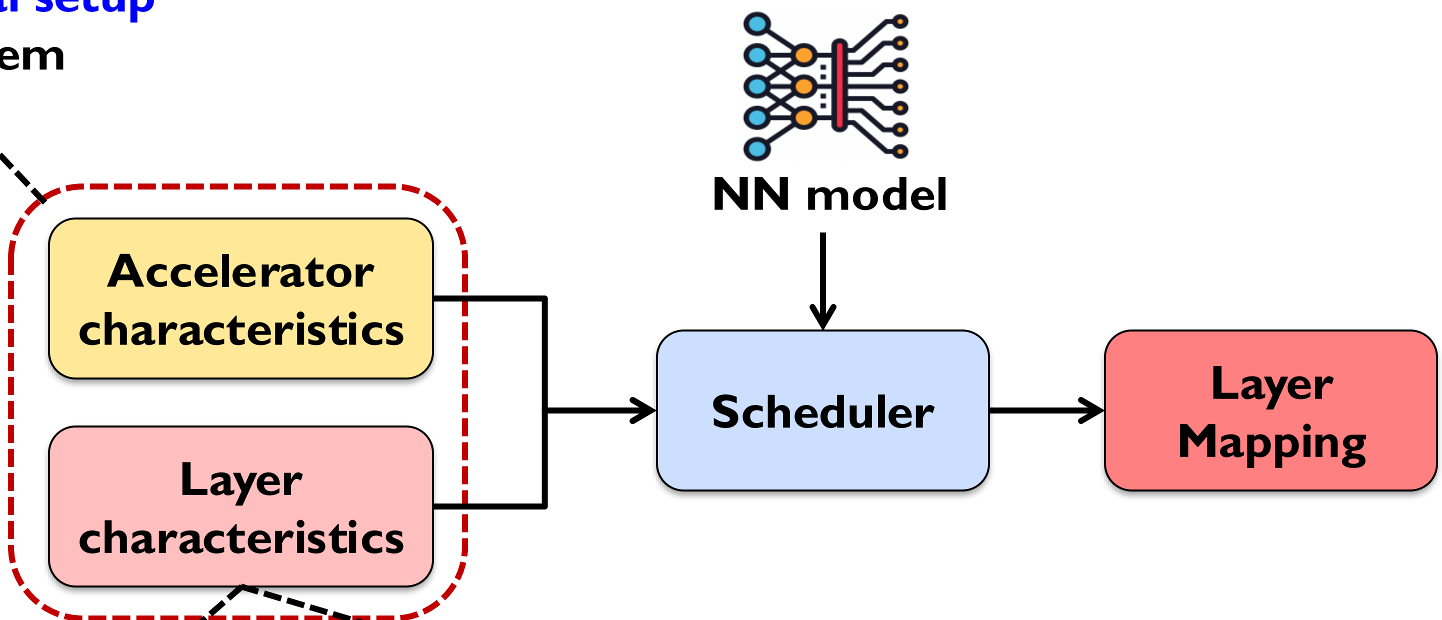
Families 3, 4 & 5: high parameter footprint, low data reuse and MAC intensity → data-centric layers

SAFARI

Mensa Runtime Scheduler

The goal of Mensa's software runtime scheduler is to identify which accelerator each layer in an NN model should run on

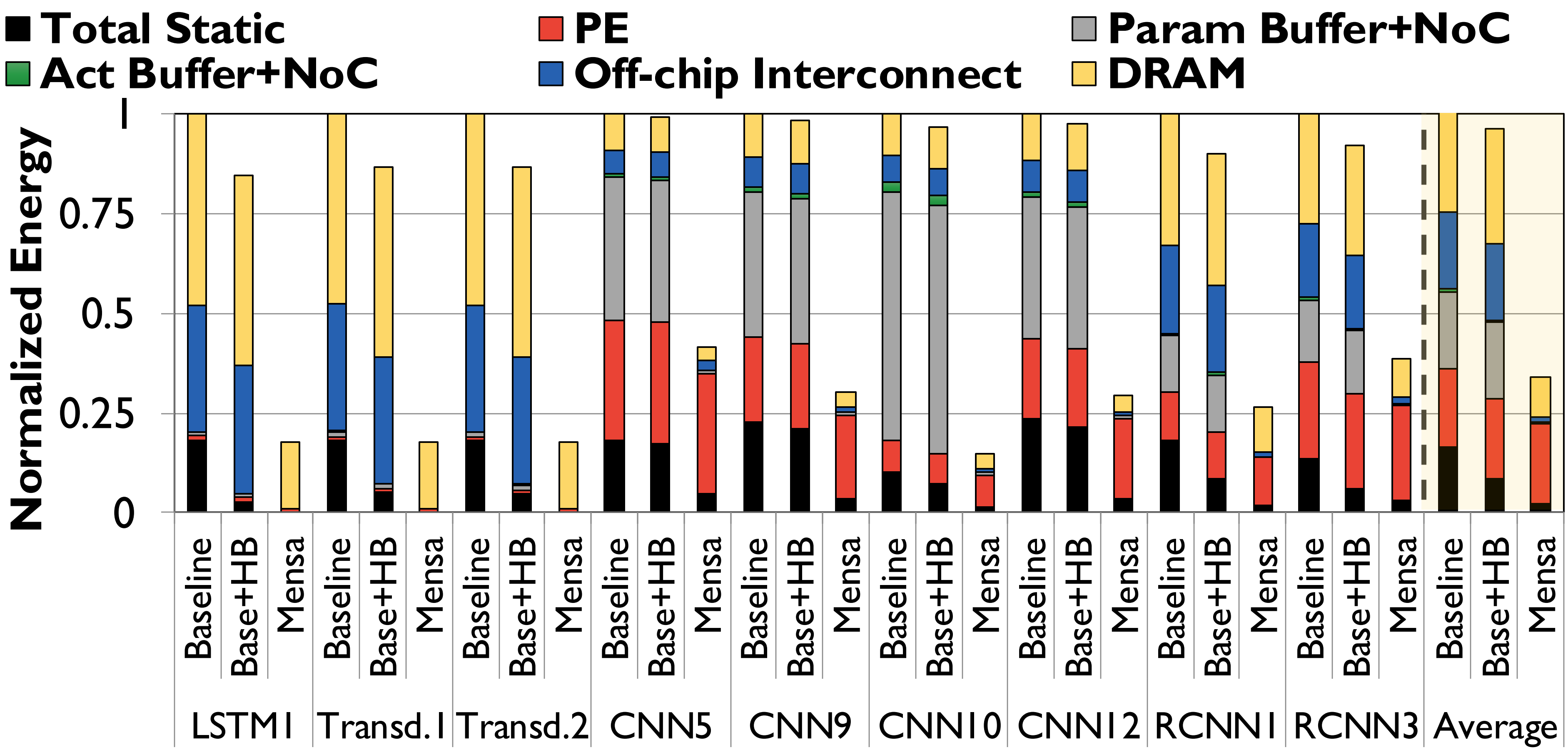
Generated once during initial setup of a system



Each of the accelerators caters to a specific family of layers

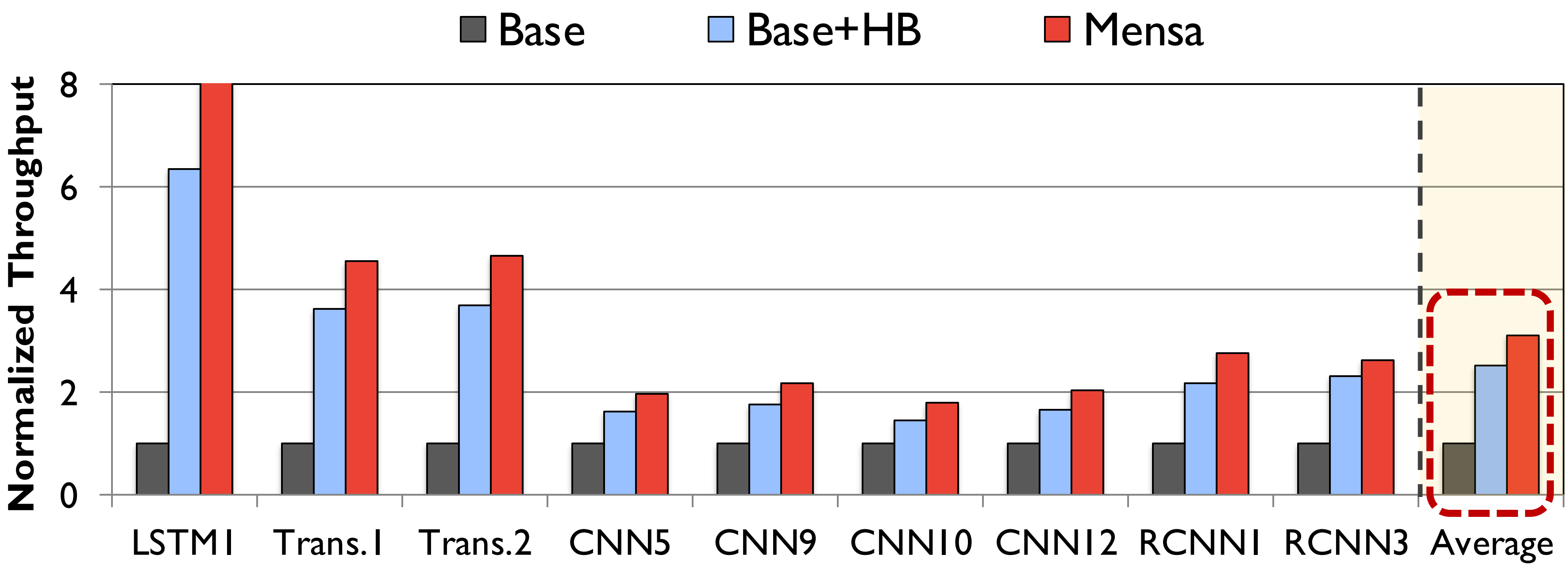
Layers tend to group together into a small number of families

Mensa: Energy Reduction



Mensa-G reduces energy consumption by 3.0X compared to the baseline Edge TPU

Mensa: Throughput Improvement



Mensa-G improves inference throughput by 3.1X compared to the baseline Edge TPU

Mensa: Highly-Efficient ML Inference

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
  "Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
  *Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (**PACT**)*, Virtual, September 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (14 minutes)]

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

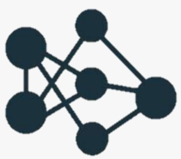
Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[*] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[*†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [*]*ETH Zürich*

HW/SW Co-Design for PIM

We follow a two-step approach to co-design software and hardware to efficiently take advantage of PIM paradigm

Target Application



Step 1:
Application Profiling

- **HW/SW requirements**
- **performance bottleneck**
- **energy bottleneck**

Step 2:
Co-design SW and HW

high-performance and energy-efficient PIM architecture

We showcase our two-step approach for several applications:

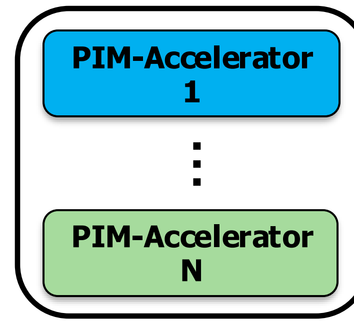
1 Machine learning inference models for edge devices

2 **Genome sequence alignment & filtering**

Possible PNM Designs

- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM

- **Fixed-function units**
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM



- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

Samsung Function-in-Memory DRAM (2021)

Samsung Newsroom

CORPORATE | PRODUCTS | PRESS RESOURCES | VIEWS | ABOUT US

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

Korea on February 17, 2021

Audio Share

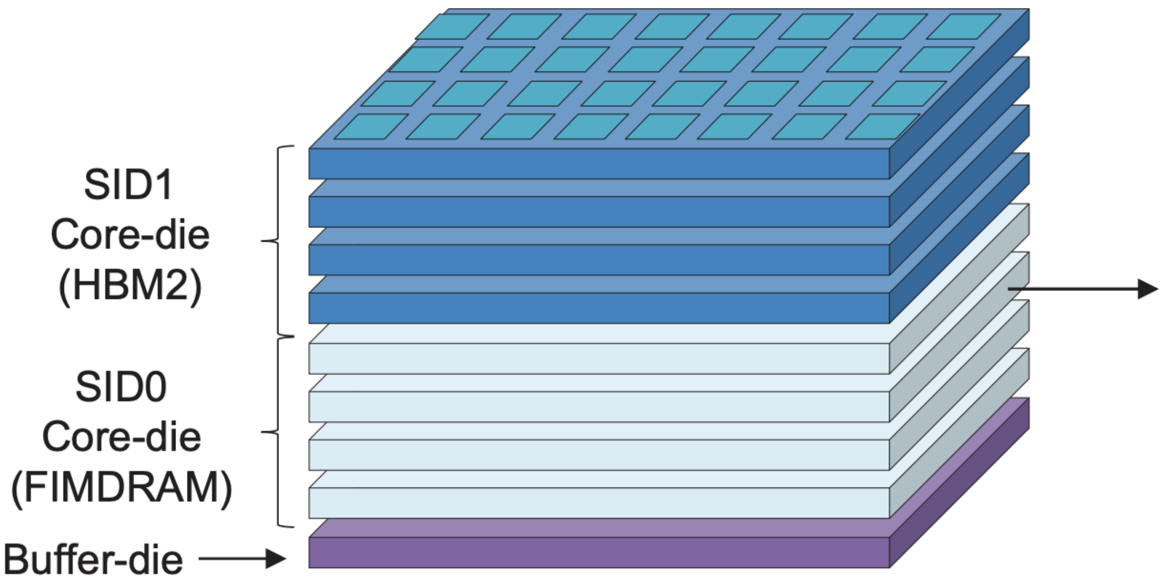
The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%

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

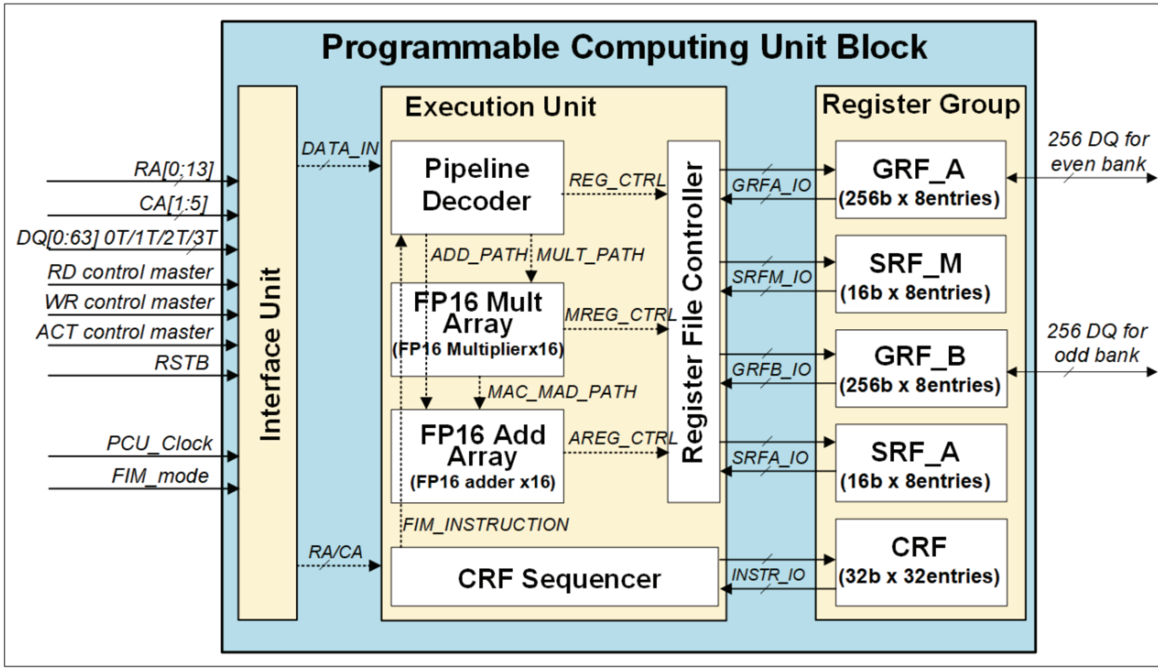
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung 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 Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung 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

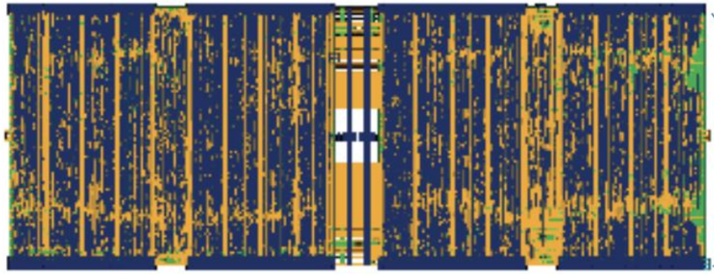
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

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

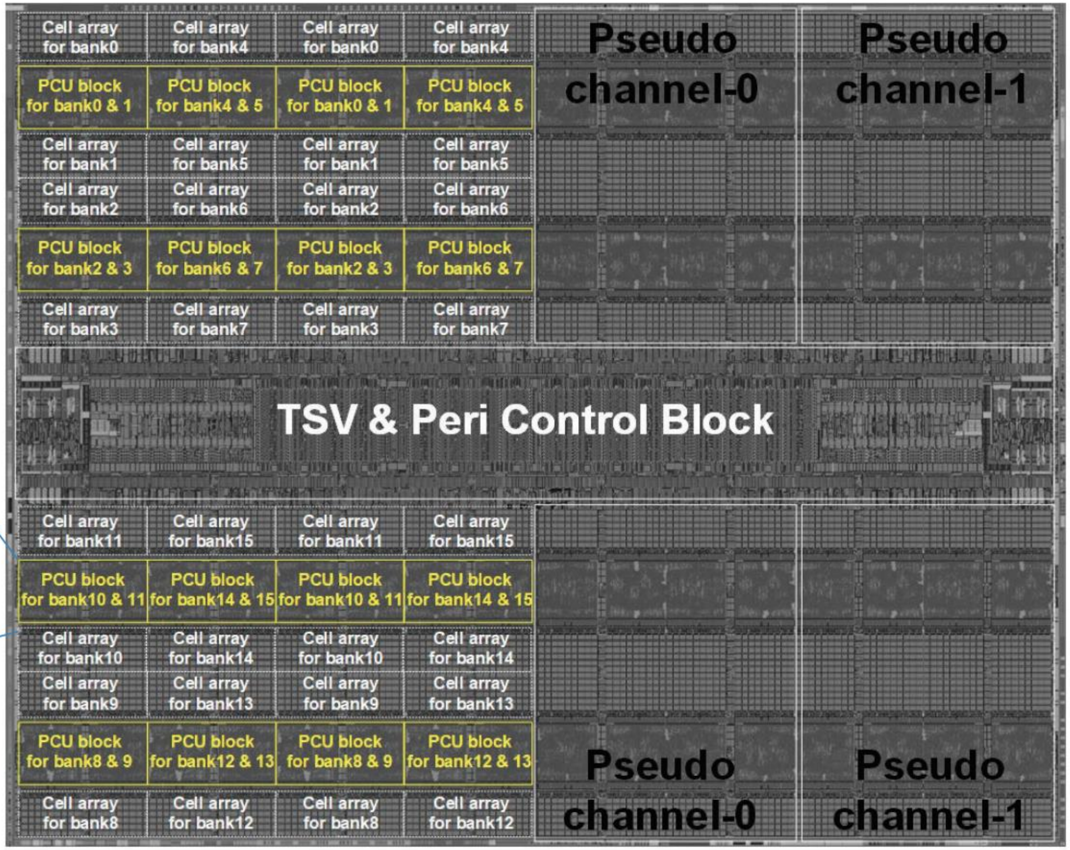
Samsung Function-in-Memory DRAM (2021)

Chip Implementation

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



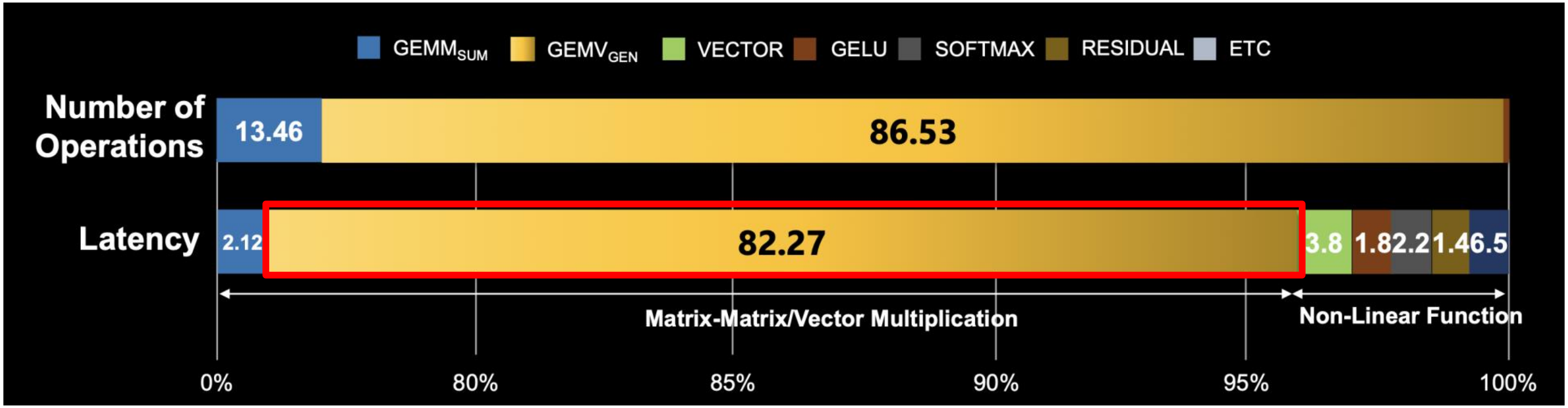
[Digital RTL design for PCU block]



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

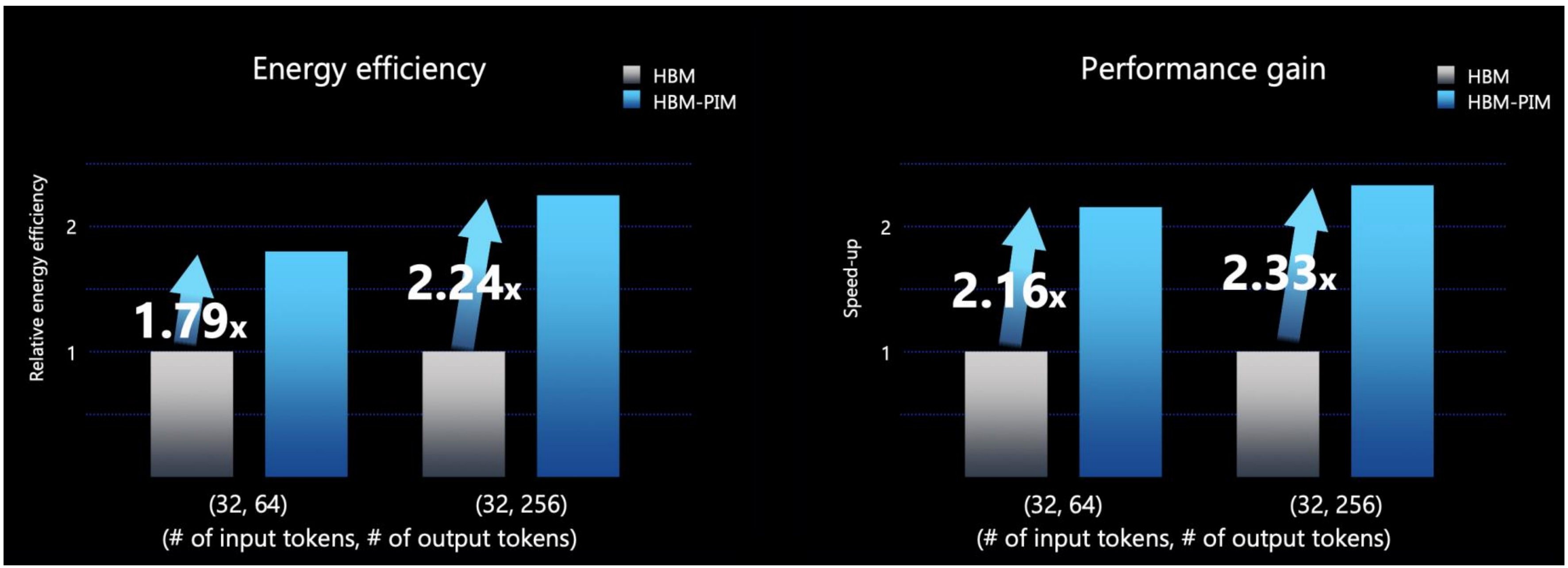


[J. H. Kim+ HC, 2023]

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

- AMD MI100 GPUs fabricated with HBM-PIM

- Experimental setup: GPT-J (6B, 32 input tokes), single AMD MI100-PIM GPU

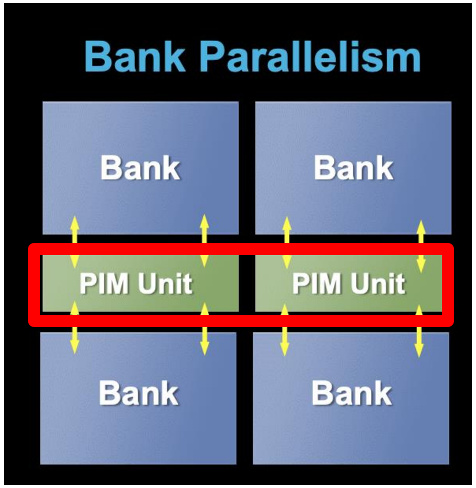


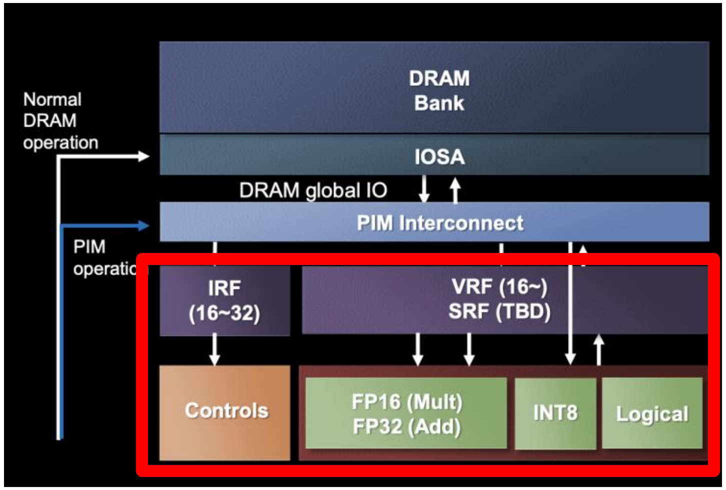
- GPT can be accelerated by more than 2x over baseline

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

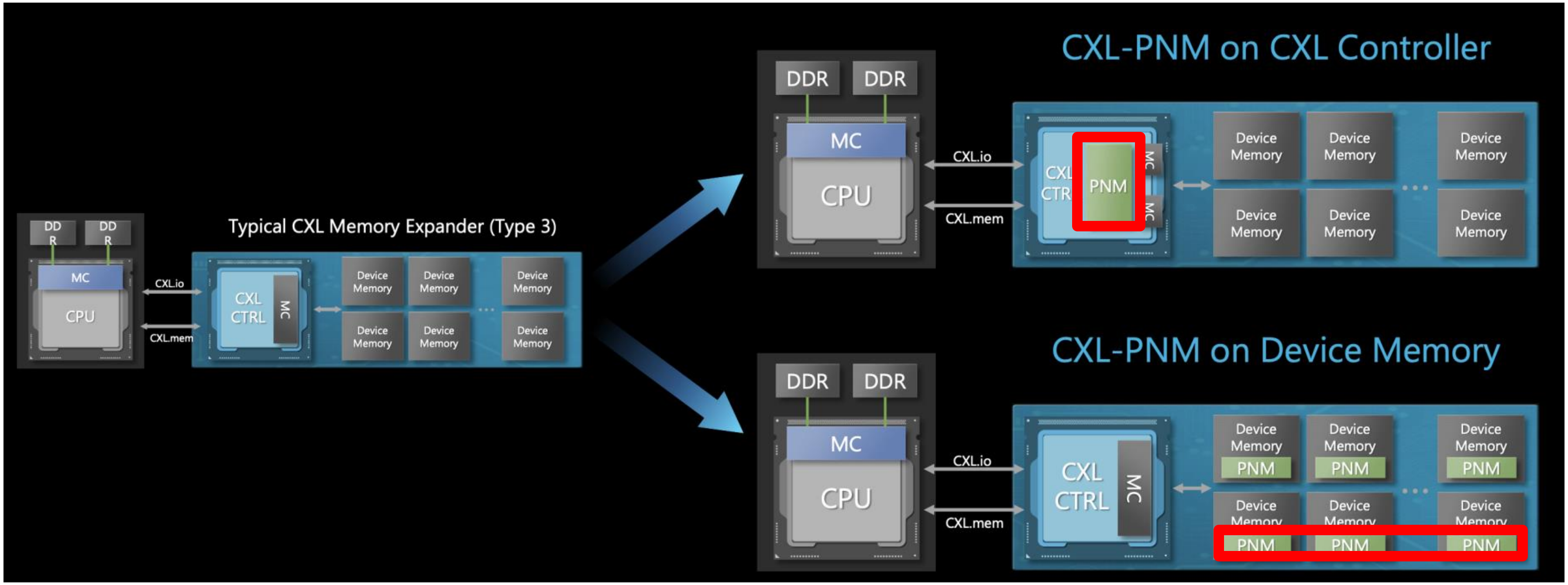




- 4.47x performance gains and 70.6% energy reduction in GPT-2

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

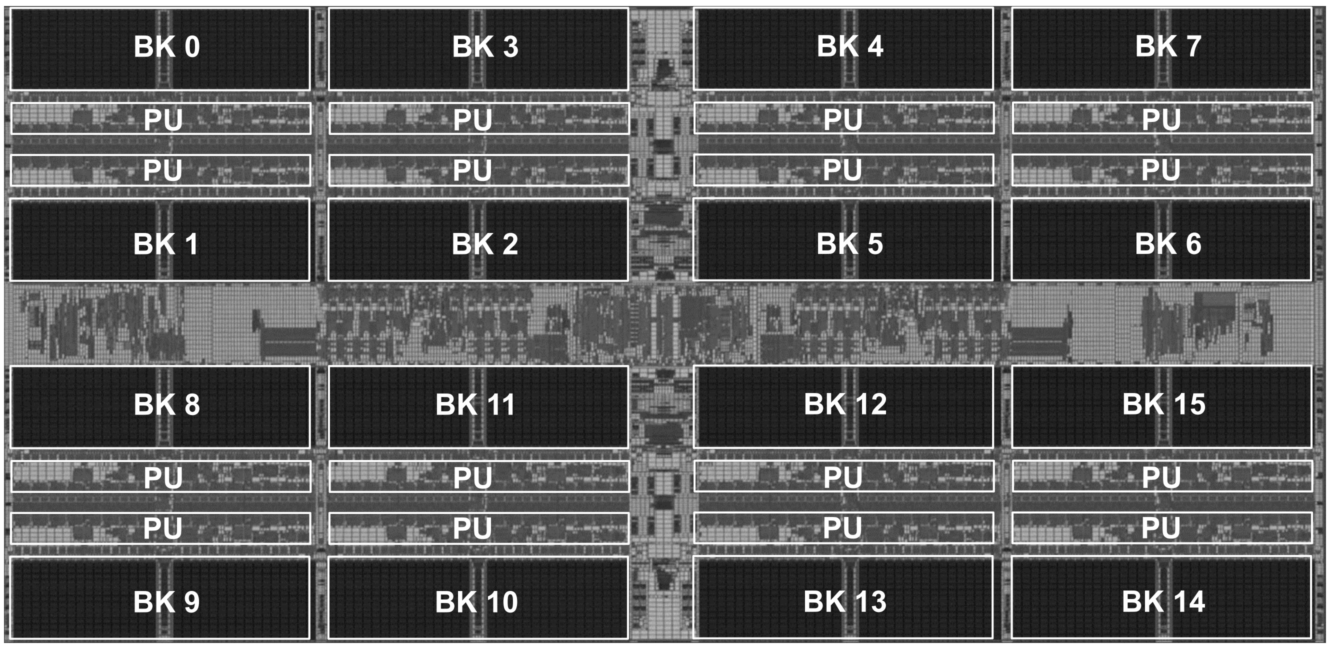


- Multiple CXL-PNM can offer 4.4x higher energy efficiency and 53% higher throughput than multiple GPUs

SK hynix AiM: Chip Implementation (2022)

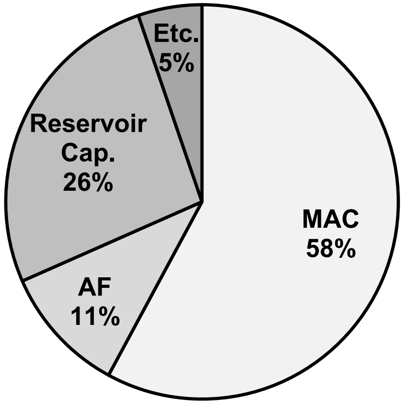
- 4 Gb AiM die with 16 processing units (PUs)

AiM Die Photograph



1 Process Unit (PU) Area

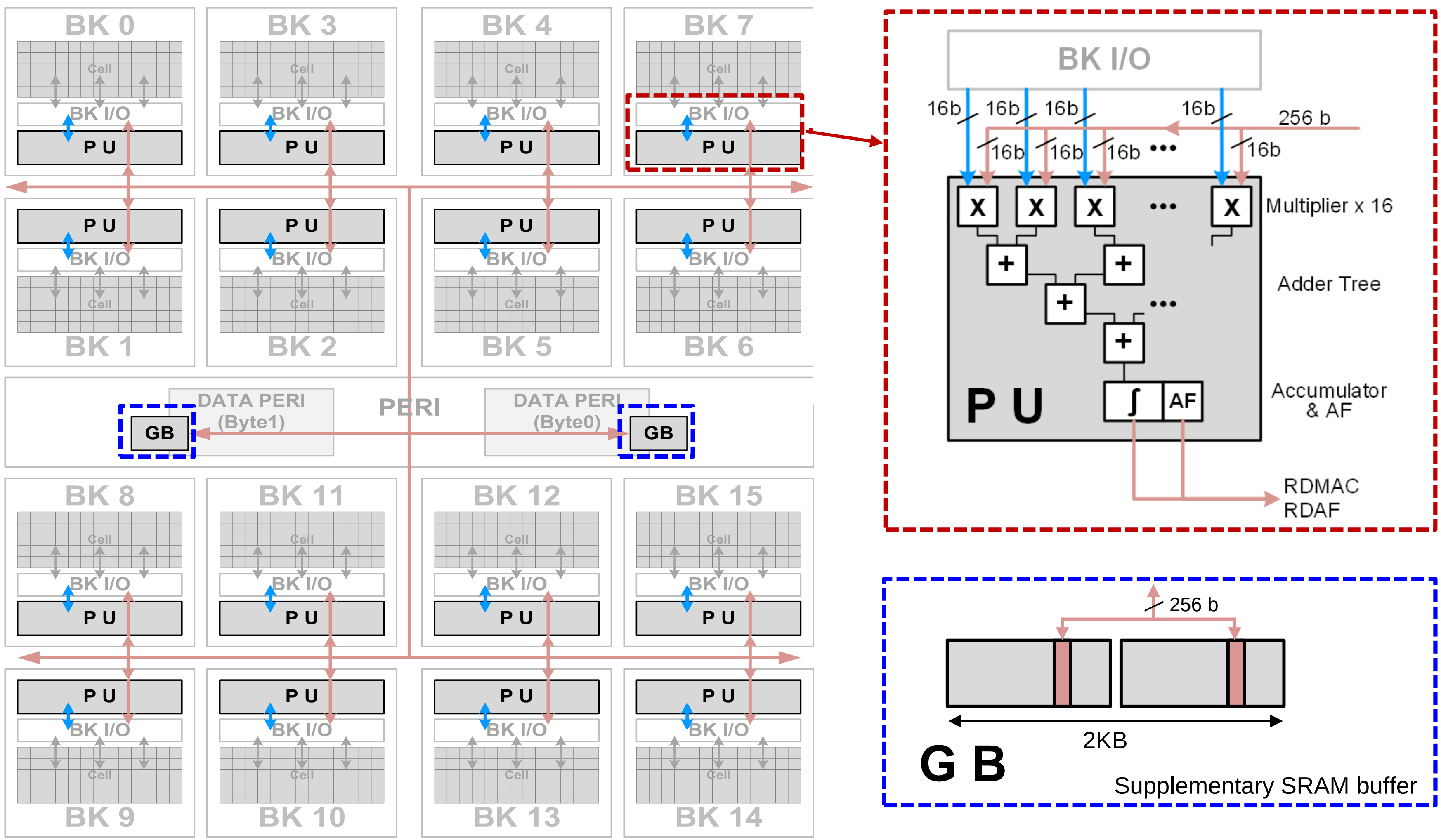
Total	0.19mm^2
MAC	0.11mm^2
Activation Function (AF)	0.02mm^2
Reservoir Cap.	0.05mm^2
Etc.	0.01mm^2



Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022

SK hynix AiM: System Organization (2022)

- GDDR6-based AiM architecture



Lee et al., A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications, ISSCC 2022

Possible PNM Designs

- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM

- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM

- **Reconfigurable architectures**
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

FPGA-based Processing Near Memory

- 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.

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]
[a]Eindhoven University of Technology [b]ETH Zürich [c]IBM Research Europe, Zurich

Heterogeneous System: CPU+FPGA

Source: IBM

CAPI2

Source: AlphaData

POWER9 AC922

DDR4-based AD9V3 board

We evaluate two POWER9+FPGA systems:

1. HBM-based board AD9H7 **AD9V3**

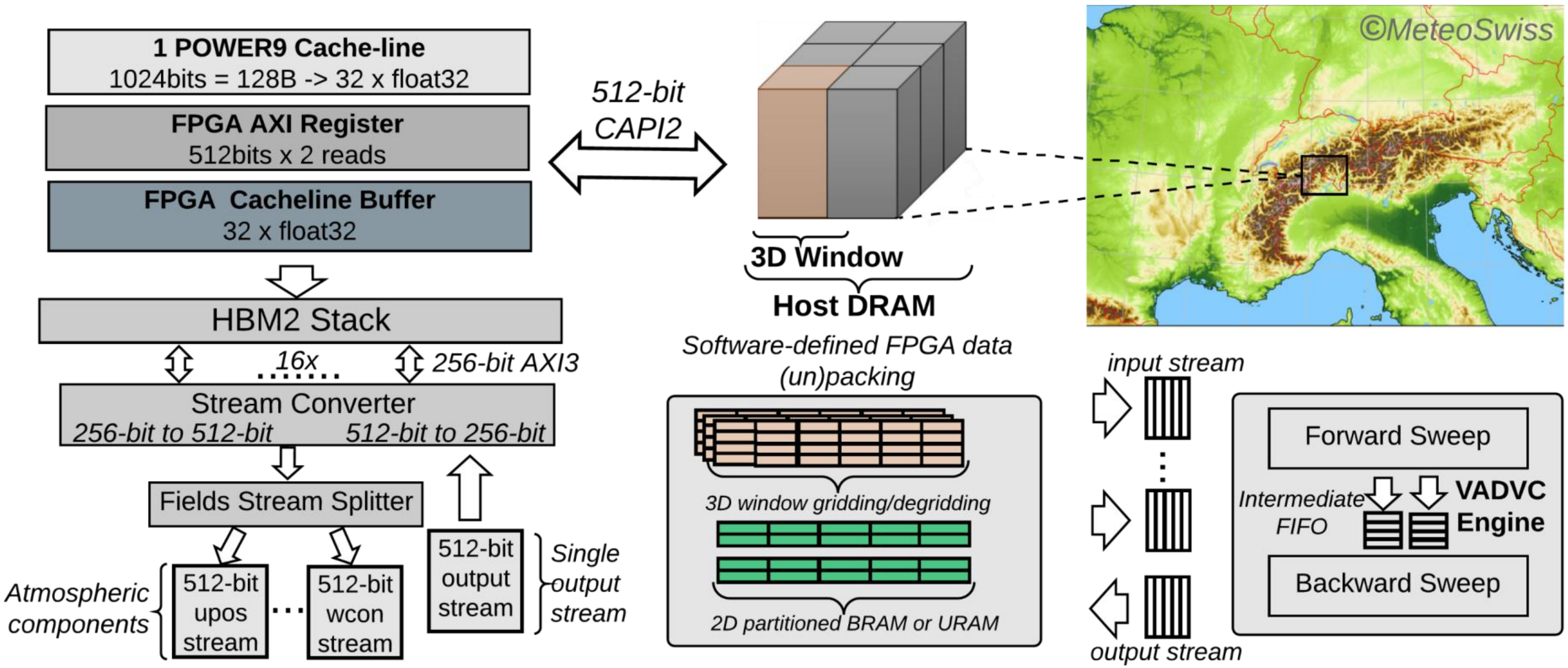
Xilinx Virtex Ultrascale+™ XCVU37P-2

2

2. DDR4-based board

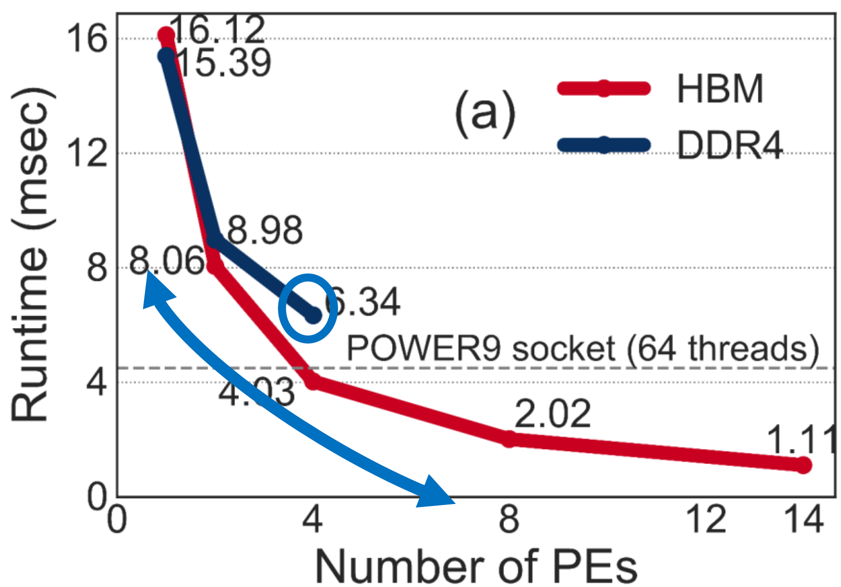
Xilinx Virtex Ultrascale+™ XCVU3P-

NERO Design Flow

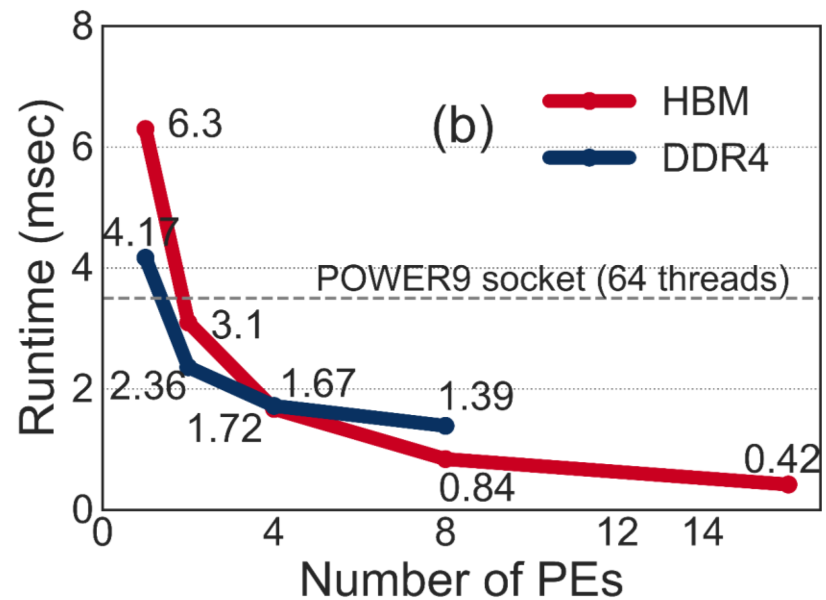


NERO Performance Analysis

Vertical Advection



Horizontal Diffusion



NERO is 4.2x and 8.3x faster than a complete POWER9 socket

Possible PNM Designs

- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM

- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM

- **Reconfigurable architectures**
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM

Samsung AxDIMM (2021)

Samsung Newsroom

CORPORATE | PRODUCTS | INSIGHTS | PRESS RESOURCES

Samsung Brings In-Memory Processing Power to Wider Range of Applications

Korea on August 24, 2021

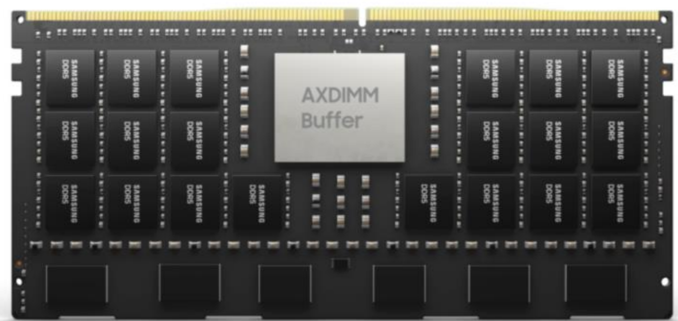
Audio Share

Integration of HBM-PIM with the Xilinx Alveo AI accelerator system will boost overall system performance by 2.5X while reducing energy consumption by more than 60%

PIM architecture will be broadly deployed beyond HBM, to include mainstream DRAM modules and mobile memory

Samsung Electronics, the world leader in advanced memory technology, today showcased its latest advancements with processing-in-memory (PIM) technology at Hot Chips 33—a leading semiconductor conference where the most notable microprocessor and IC innovations are unveiled each year. Samsung's revelations include the first successful integration of its PIM-enabled High Bandwidth Memory (HBM-PIM) into a commercialized accelerator system, and broadened PIM applications to embrace DRAM modules and mobile memory, in accelerating the move toward the convergence of memory and logic.

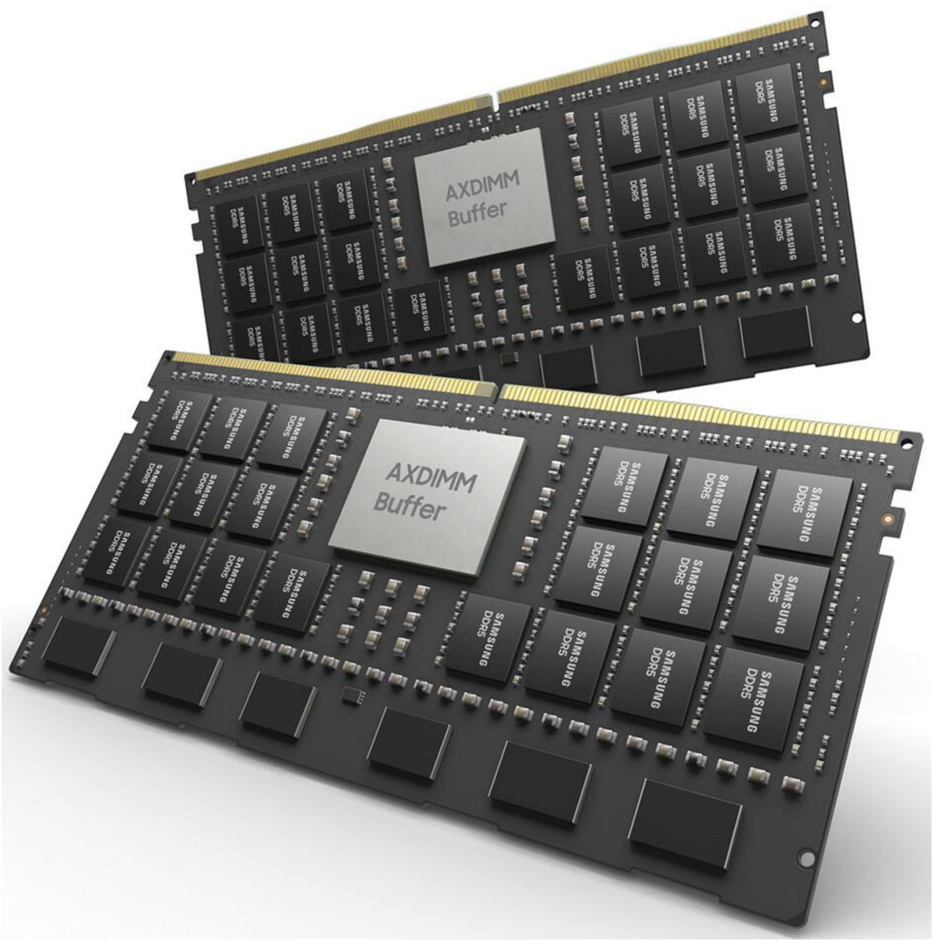
DRAM Modules Powered by PIM

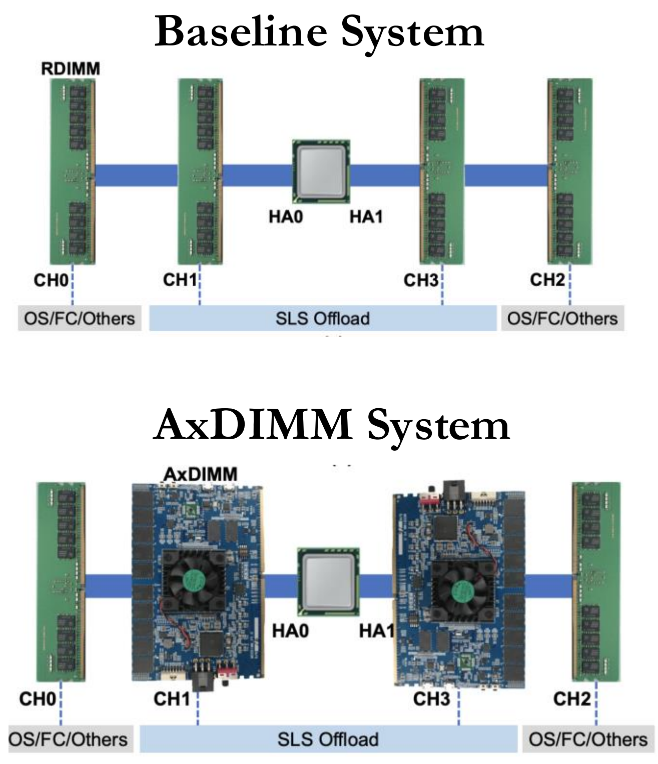


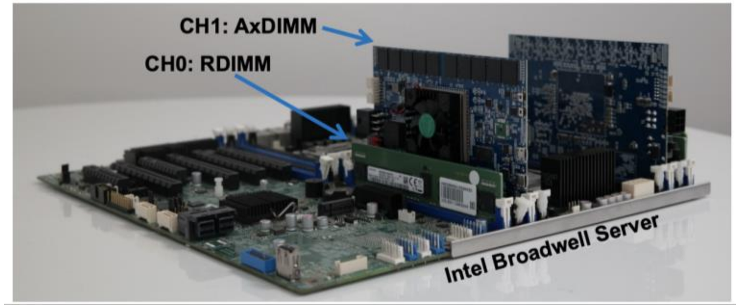
The Acceleration DIMM (AXDIMM) brings processing to the DRAM module itself, minimizing large data movement between the CPU and DRAM to boost the energy efficiency of AI accelerator systems. With an AI engine built inside the buffer chip, the AXDIMM can perform parallel processing of multiple memory ranks (sets of DRAM chips) instead of accessing just one rank at a time, greatly enhancing system performance and efficiency. Since the module can retain its traditional DIMM form factor, the AXDIMM facilitates drop-in replacement without requiring system modifications. Currently being tested on customer servers, the AXDIMM can offer approximately twice the performance in AI-based recommendation applications and a 40% decrease in system-wide energy usage.

Samsung AxDIMM (2021)

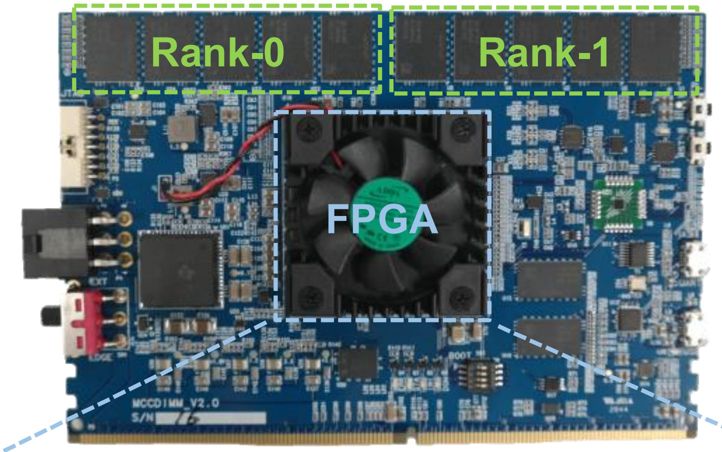
- DIMM-based PIM
  - DLRM recommendation system







AxDIMM Design: Hardware Architecture

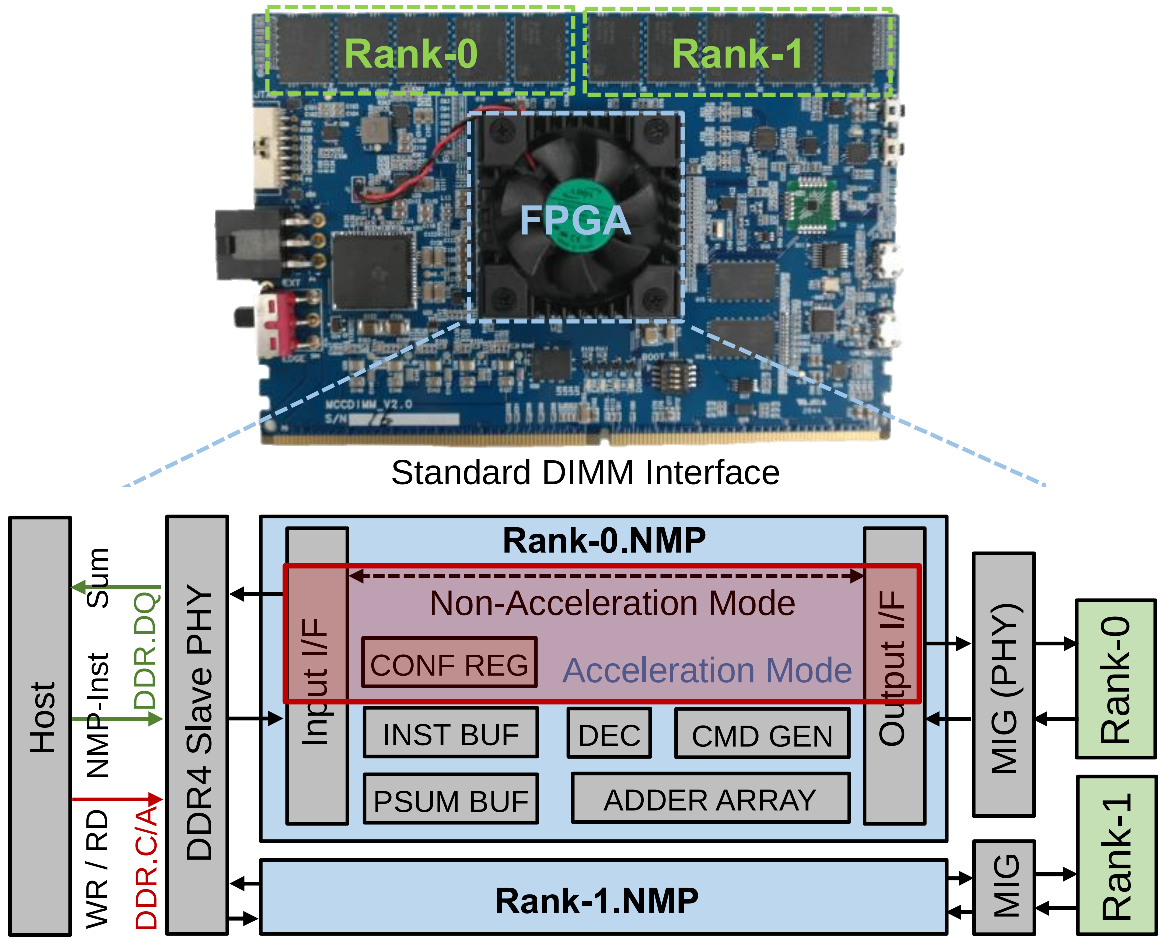


Standard DIMM Interface

FPGA board with standard DIMM interface:
It serves as a real-system
near-memory processing implementation

[Ke+, IEEE Micro'2021]

AxDIMM Design: Hardware Architecture



Two execution modes:
(1) non-acceleration mode
(2) acceleration mode (blocking)

Sparse Length Sum with AxDIMM (IEEE Micro 2021)

Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM

Liu Ke[*†], Xuan Zhang[†], Jinin So[‡], Jong-Geon Lee[‡], Shin-Haeng Kang[‡], Sukhan Lee[‡], Songyi Han[‡], YeonGon Cho[‡], JIN Hyun Kim[‡], Yongsuk Kwon[‡], KyungSoo Kim[‡], Jin Jung[‡], Ilkwon Yun[‡], Sung Joo Park[‡], Hyunsun Park[‡], Joonho Song[‡], Jeonghyeon Cho[‡], Kyomin Sohn[‡], Nam Sung Kim[‡], Hsien-Hsin S. Lee[*]

[*]Facebook, [†]Washington University in St. Louis, [‡]Samsung

https://doi.org/10.1109/MM.2021.3097700

An Architecture of Sparse Length Sum Accelerator in AxDIMM

Shin-haeng Kang
DRAM Design Team 1
Samsung Electronics
Hwasung, South Korea
s-h.kang@samsung.com

Byeongho Kim
DRAM Design Team 1
Samsung Electronics
Hwasung, South Korea
bh1122.kim@samsung.com

Sukhan Lee
DRAM Design Team 1
Samsung Electronics
Hwasung, South Korea
sh1026.lee@samsung.com

Kyomin Sohn
DRAM Design Team 1
Samsung Electronics
Hwasung, South Korea
kyomin.sohn@samsung.com

https://doi.org/10.1109/AICAS54282.2022.9869896

Database Operations with AxDIMM (DaMoN 2022)

Improving In-Memory Database Operations with Acceleration DIMM (AxDIMM)

Donghun Lee
Minseon Ahn
Jungmin Kim
dong.hun.lee@sap.com
minseon.ahn@sap.com
jungmin.kim@sap.com
SAP Labs Korea

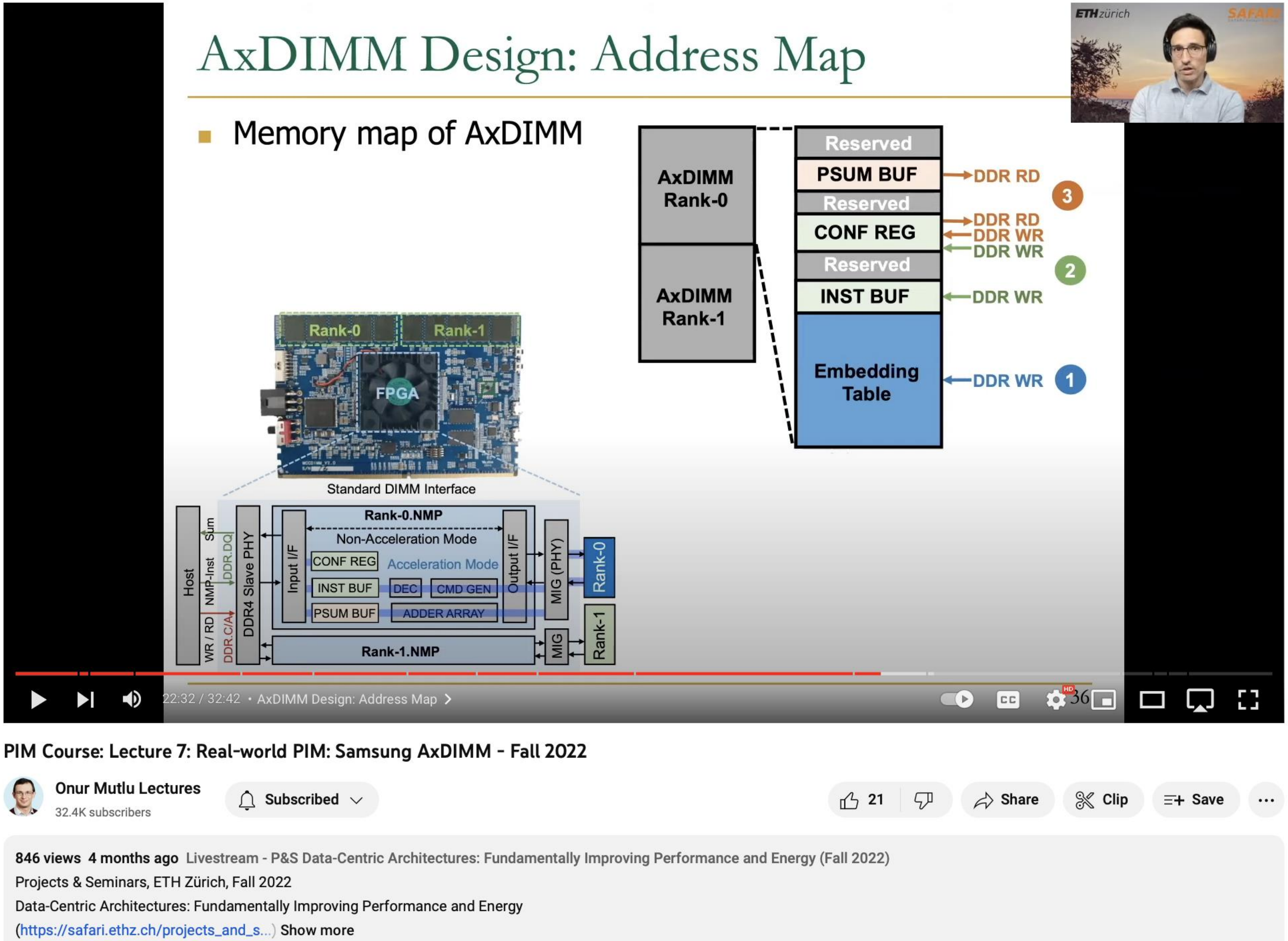
Oliver Rebholz
oliver.rebholz@sap.com
SAP SE

Jinin So
Jong-Geon Lee
Jeonghyeon Cho
Vishnu Charan Thummala
jinin.so@samsung.com
jg1021.lee@samsung.com
caleb1@samsung.com
vishnu.c.t@samsung.com
Samsung Electronics

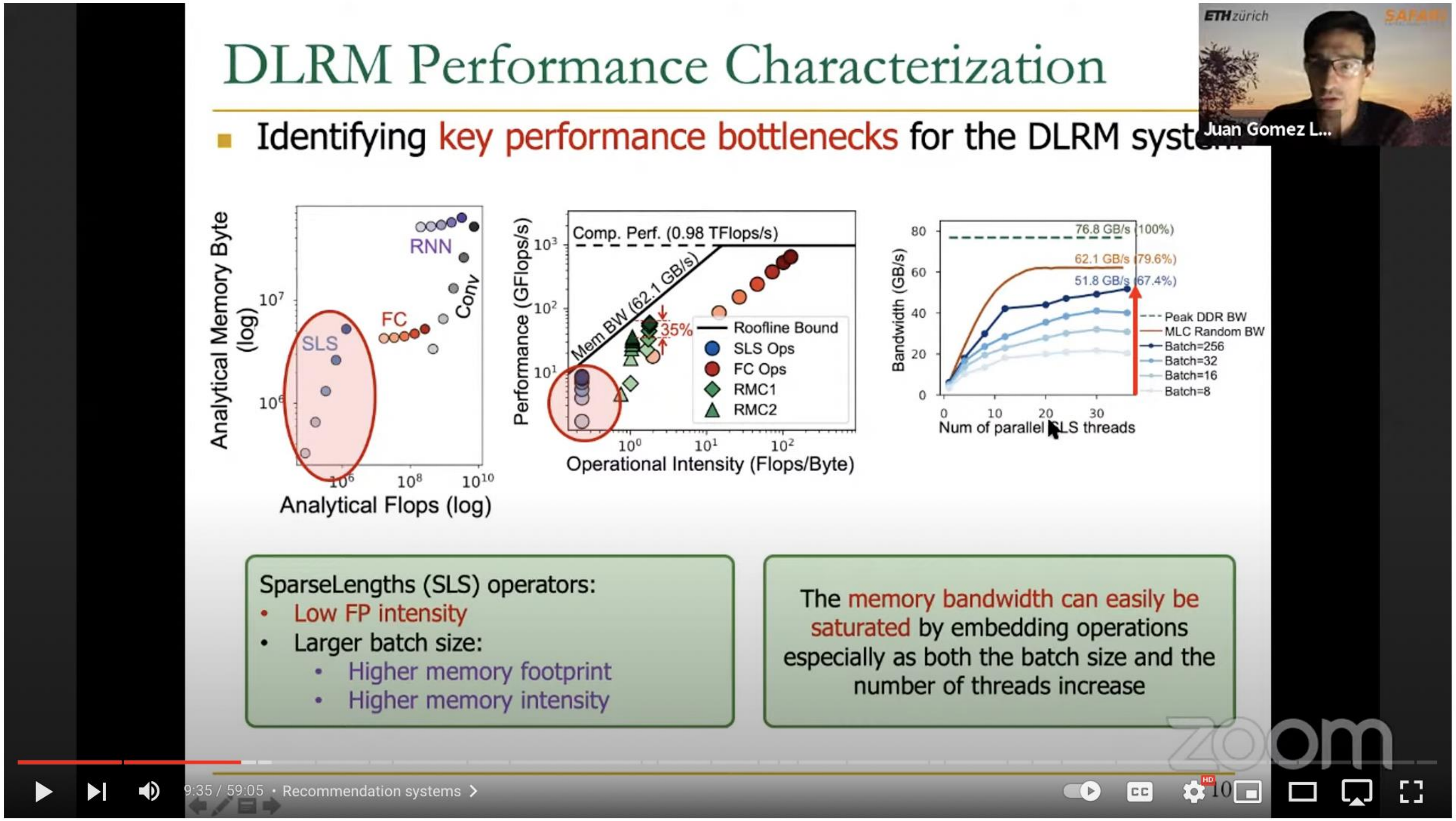
Ravi Shankar JV
Sachin Suresh Upadhya
Mohammed Ibrahim Khan
Jin Hyun Kim
venkata.ravi@samsung.com
sachin1.s@samsung.com
ibrahim.khan@samsung.com
kjh5555@samsung.com
Samsung Electronics

https://doi.org/10.1145/3533737.3535093

Longer Lecture on AxDIMM



Another Longer Lecture on AxDIMM

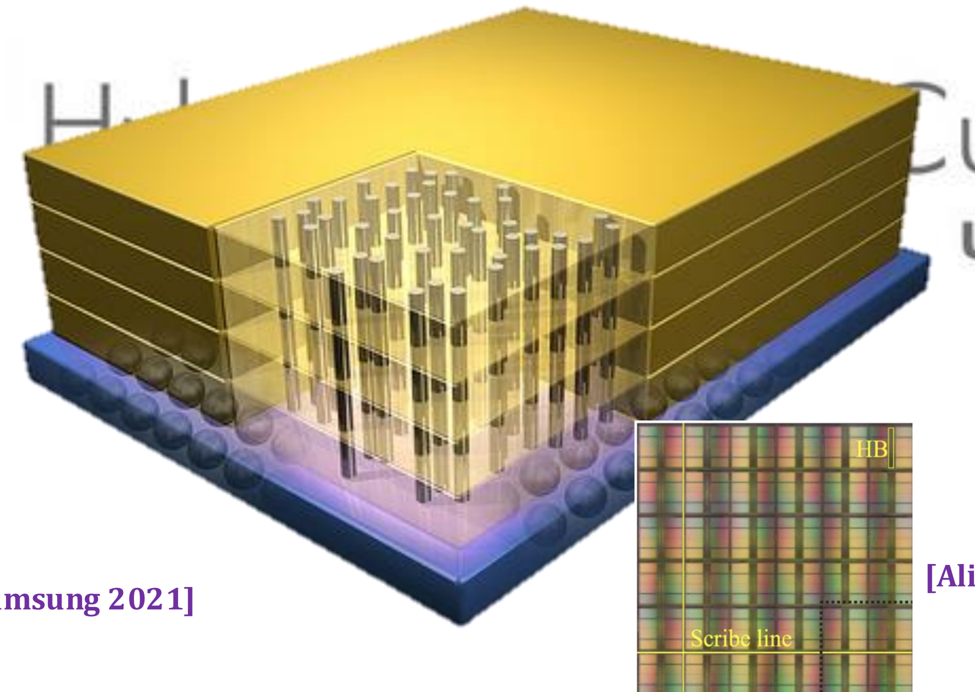


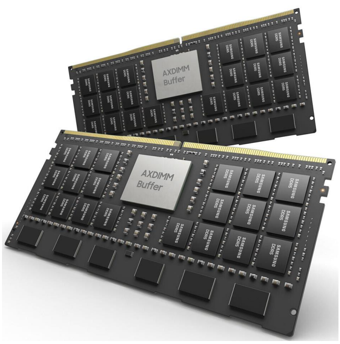
Processing in Memory Course: Meeting 5: Real-world PIM architectures IV - Fall'21

Onur Mutlu Lectures
32.4K subscribers
Subscribed | 18 | Share | Clip | Save

676 views Streamed 1 year ago Livestream - P&S Exploring the Processing-in-Memory Paradigm for Future Computing Systems (Fall 2021)
Project & Seminar, ETH Zürich, Fall 2021
Exploring the Processing-in-Memory Paradigm for Future Computing Systems (https://safari.ethz.ch/projects_and_s...)
Show more

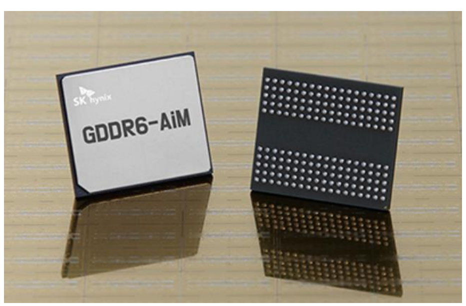
Processing-in-Memory Landscape Today



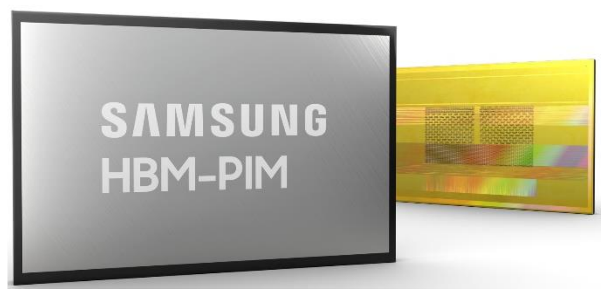


[Samsung 2021]

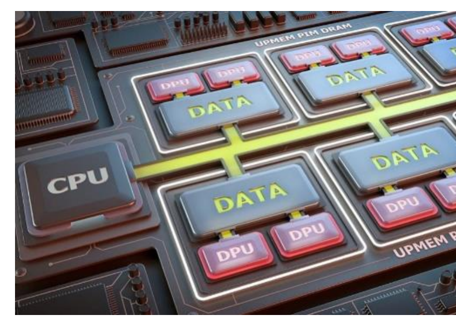
[Alibaba 2022]



[SK Hynix 2022]



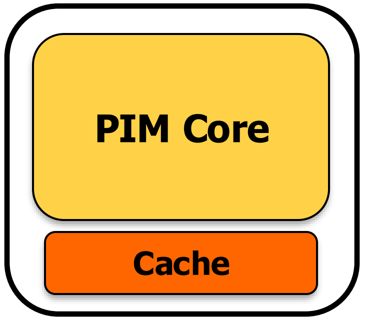
[Samsung 2021]



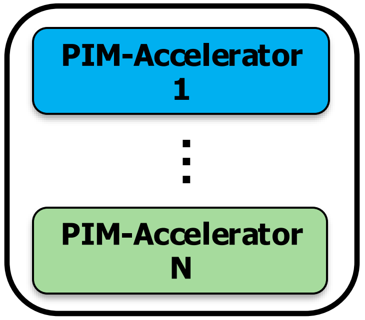
[UPMEM 2019]

Possible PNM Designs

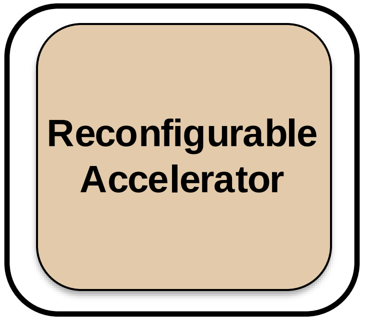
- General-purpose programmable cores
  - Wimpy cores (possibility of running any workload)
  - E.g. from academia: Tesseract PIM for Graph Processing
  - E.g. from industry: UPMEM PIM



- Fixed-function units
  - Hardware/software co-designed PIM for efficiency
  - E.g. from academia: Mensa for NN Edge Inference
  - E.g. from industry: Samsung HBM-PIM, SK hynix AiM



- Reconfigurable architectures
  - PNM cores coupled with FPGAs, CGRA
  - E.g. from academia: NERO for Weather Prediction
  - E.g. from industry: Samsung AxDIMM



Research Tools PNM: DAMOV-SIM

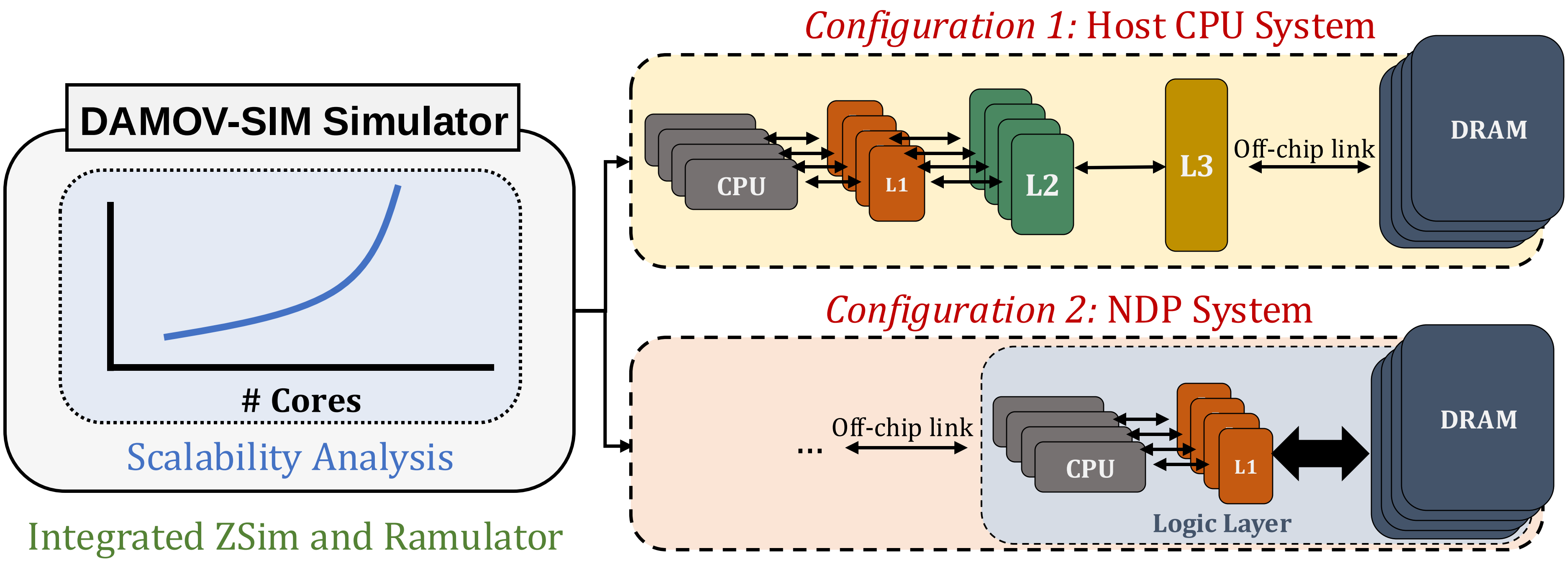
- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
  "DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
  IEEE Access, 8 September 2021.
  Preprint in ***arXiv***, 8 May 2021.
  [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

Step 3: Memory Bottleneck Classification (2/2)

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



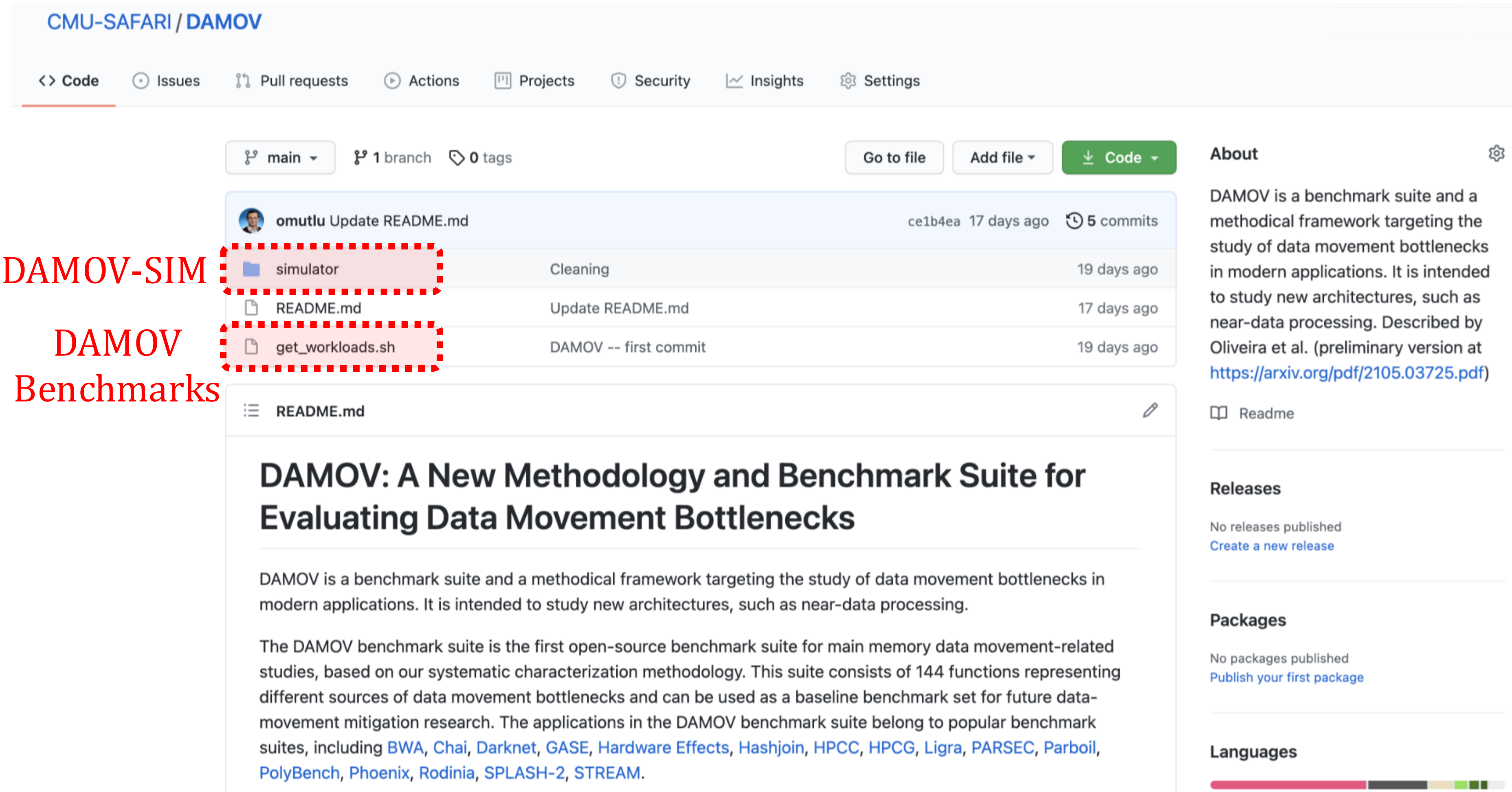
- Scalability Analysis:
  - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
  - 3D-stacked memory as main memory

DAMOV is Open Source

- We open-source our benchmark suite and our toolchain

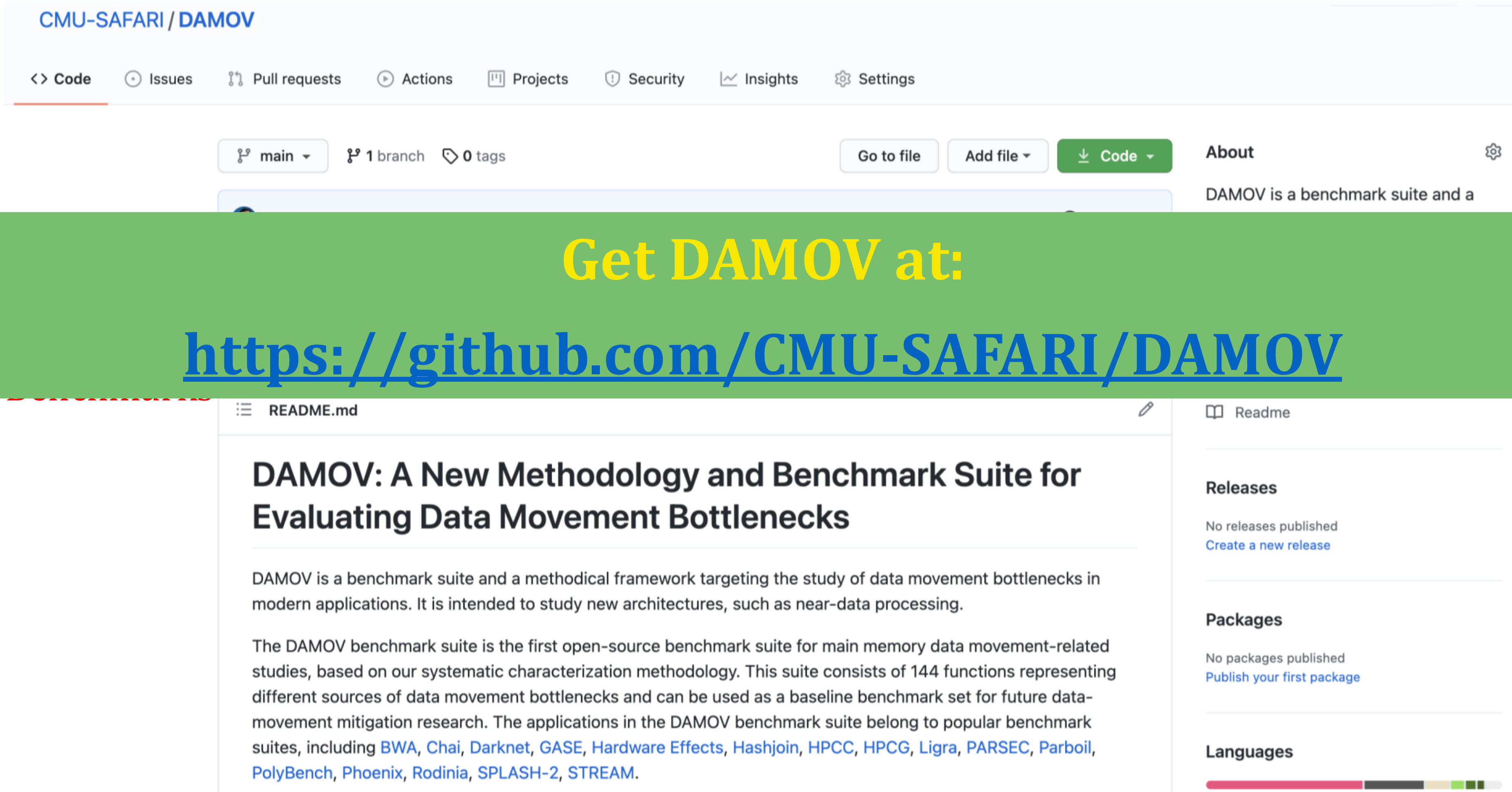
DAMOV-SIM

DAMOV Benchmarks



DAMOV is Open Source

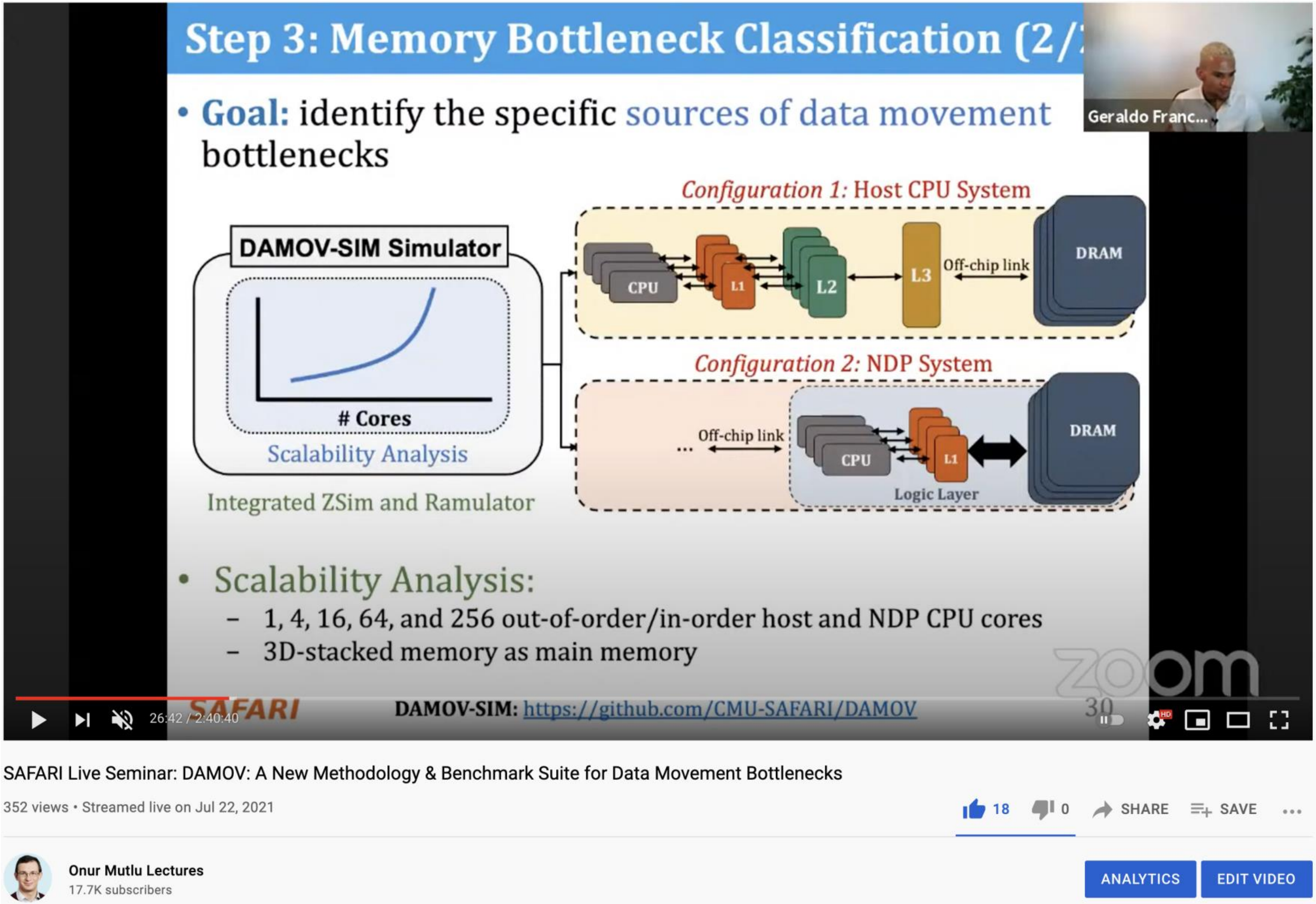
- We open-source our benchmark suite and our toolchain



Get DAMOV at:

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

More on DAMOV Analysis Methodology & Workloads



https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9&index=3

More on DAMOV Methods & Benchmarks

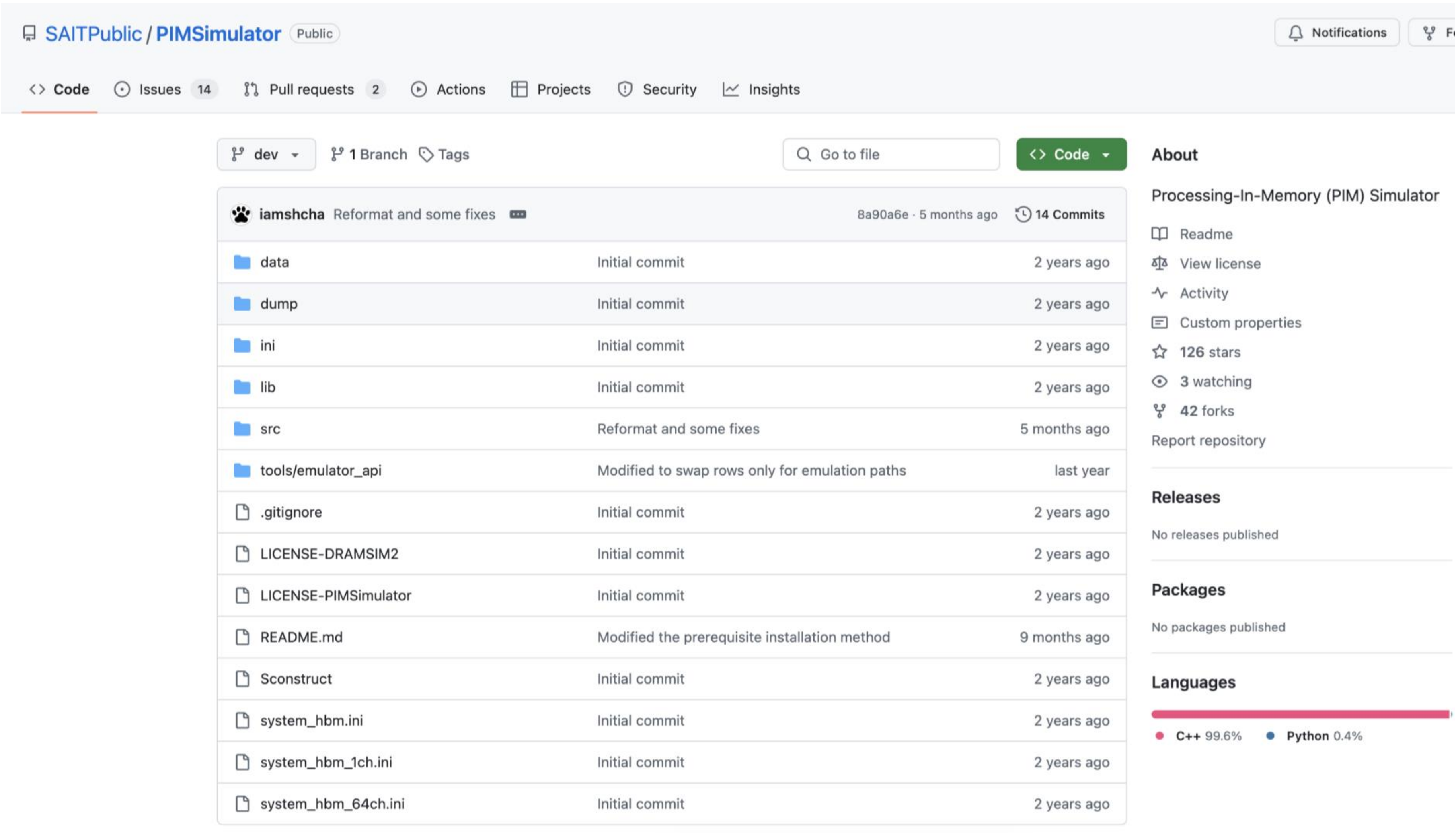
- Geraldo F. Oliveira, Juan Gomez-Luna, Lois Orosa, Saugata Ghose, Nandita Vijaykumar, Ivan fernandez, Mohammad Sadrosadati, and Onur Mutlu,
  "DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks"
  IEEE Access, 8 September 2021.
  Preprint in ***arXiv***, 8 May 2021.
  [arXiv preprint]
  [IEEE Access version]
  [DAMOV Suite and Simulator Source Code]
  [SAFARI Live Seminar Video (2 hrs 40 mins)]
  [Short Talk Video (21 minutes)]

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

GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
JUAN 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

Research Tools PNM: Samsung HBM-PIM

https://github.com/SAITPublic/PIMSimulator

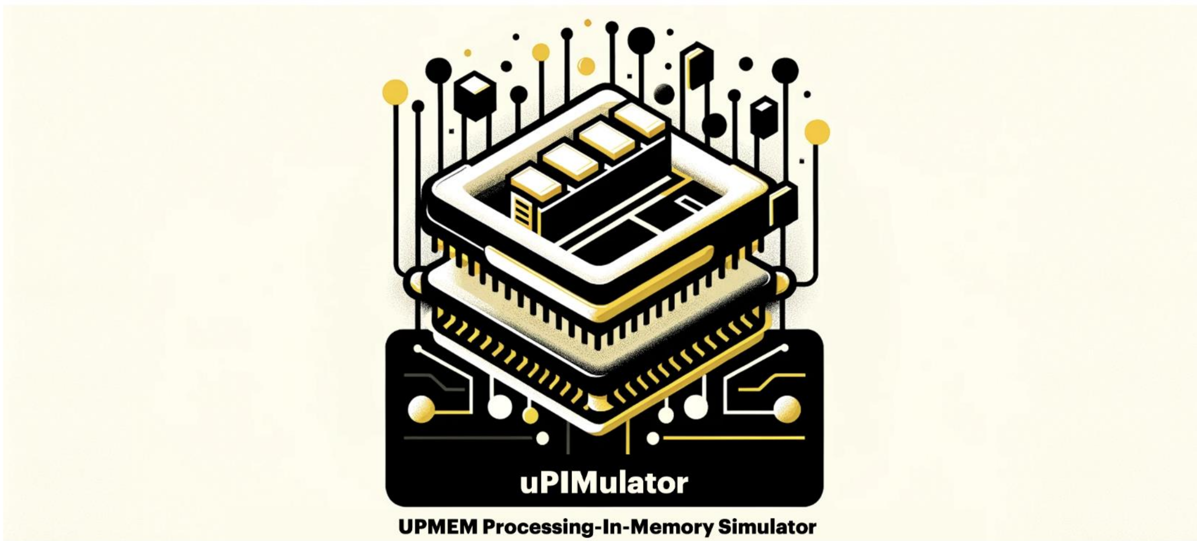


Research Tools PNM: UPMEM PIM (I)

https://github.com/VIA-Research/uPIMulator

README MIT license

📖 Introduction



Welcome to the uPIMulator Framework Documentation!

This documentation serves as your comprehensive guide to the uPIMulator framework, catering to both novice and experienced researchers. Here, you'll find the resources necessary to leverage uPIMulator effectively for your research projects.

We provide in-depth coverage of uPIMulator's features, from foundational concepts to advanced functionalities. Explore this documentation to unlock the full potential of uPIMulator and elevate your research endeavors.

Research Tools PNM: UPMEM PIM (II)

https://ieeexplore.ieee.org/document/10476411/

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

Pathfinding Future PIM Architectures by Demystifying a Commercial PIM Technology

Bongjoon Hyun Taehun Kim Dongjae Lee Minsoo Rhu
KAIST
{bongjoon.hyun, taehun.kim, dongjae.lee, mrhu}@kaist.ac.kr

1st Workshop on Memory-Centric Computing: Processing-Near-Memory

Geraldo F. Oliveira

https://geraldofojunior.github.io

ASPLOS 2025

30 March 2025

SAFARI

ETH zürich