SAFARI Research Group
Introduction & Research

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
29 February 2024
Computer Architecture Seminar
Brief Self Introduction

- Onur Mutlu
  - Full Professor @ ETH Zurich ITET (INFK), since Sept 2015
  - Strecker Professor @ Carnegie Mellon University ECE (CS), 2009-2016, 2016-...
  - Started the Comp Arch Research Group @ Microsoft Research, 2006-2009
  - Worked @ Google, VMware, Microsoft Research, Intel, AMD
  - PhD in Computer Engineering from University of Texas at Austin in 2006
  - BS in Computer Engineering & Psychology from University of Michigan in 2000
  - https://people.inf.ethz.ch/omutlu/  omutlu@gmail.com

- Research and Teaching in:
  - **Computer architecture, systems, hardware security, bioinformatics**
  - Memory and storage systems
  - Robust & dependable hardware systems: security, safety, predictability, reliability
  - Hardware/software cooperation
  - New computing paradigms; architectures with emerging technologies/devices
  - Architectures for bioinformatics, genomics, health, medicine, AI/ML
  - ...
Art credit: Malti Redeker (https://www.instagram.com/malti.red/)
Current Mission

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

Build fundamentally better computers

Heterogeneous Processors and Accelerators

Hybrid Main Memory

Persistent Memory/Storage

Graphics and Vision Processing
Four Key Current Directions

- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Architectures for AI/ML, Genomics, Medicine, Health, ...
The Transformation Hierarchy

Computer Architecture (expanded view)

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

Computer Architecture (narrow view)
Axiom

To achieve the highest energy efficiency and performance:

we must take the expanded view of computer architecture

Co-design across the hierarchy: Algorithms to devices

Specialize as much as possible within the design goals
Current Research Mission & Major Topics

Build fundamentally better architectures

- Data-centric arch. for low energy & high perf.
  - Proc. in Mem/DRAM, NVM, unified mem/storage

- Low-latency & predictable architectures
  - Low-latency, low-energy yet low-cost memory
  - QoS-aware and predictable memory systems

- Fundamentally secure/reliable/safe arch.
  - Tolerating all bit flips; patchable HW; secure mem

- Architectures for ML/AI/Genomics/Health/Med
  - Algorithm/arch./logic co-design; full heterogeneity

- Data-driven and data-aware architectures
  - ML/AI-driven architectural controllers and design
  - Expressive memory and expressive systems

Broad research spanning apps, systems, logic with architecture at the center
Fundamentally Better Architectures

Data-centric

Data-driven

Data-aware
We Need to Exploit Good Principles

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

Open minds
A Blueprint for Fundamentally Better Architectures

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

Intelligent Architectures for Intelligent Computing Systems

Onur Mutlu
ETH Zurich
omutlu@gmail.com
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-april-2020/

Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch
Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 here). 2020 has
SAFARI Newsletter December 2021 Edition

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

Think Big, Aim High

ETH Zürich

View in your browser
December 2021
SAFARI PhD and Post-Doc Alumni

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

- **First Computer Architecture & Digital Design Course**
  - Digital Design and Computer Architecture
  - Spring 2021 Livestream Edition: [https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LiN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LiN)

- **Advanced Computer Architecture Course**
  - Computer Architecture
  - Fall 2021 Livestream Edition: [https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)

- **Seminar in Computer Architecture**
  - [https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4](https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4)
All Our Courses

Courses

Lecture Videos & Course Materials

Spring 2023:
Digital Design and Computer Architecture
Seminar in Computer Architecture
SAFARI Project & Seminars courses

Fall 2022:
Computer Architecture
Seminar in Computer Architecture
SAFARI Project & Seminars courses

Spring 2022:
Digital Design and Computer Architecture
Seminar in Computer Architecture
SAFARI Project & Seminars courses

https://safari.ethz.ch/courses
DDCA (Spring 2022)

**Spring 2022 Edition:**

**Spring 2021 Edition:**

**Youtube Livestream (Spring 2022):**
- [https://www.youtube.com/watch?v=cpXdeE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6](https://www.youtube.com/watch?v=cpXdeE3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6)

**Youtube Livestream (Spring 2021):**
- [https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN](https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN)

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Comp Arch (Fall 2021)

- **Fall 2021 Edition:**
  - https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule
- **Fall 2020 Edition:**

- **Youtube Livestream (2021):**
  - https://www.youtube.com/watch?v=4yfkM_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
Seminar in Comp Arch (Fall 2022)

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

- **Fall 2021 Edition:**
  - [https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule](https://safari.ethz.ch/architecture_seminar/fall2021/doku.php?id=schedule)

- **Youtube Livestream (2022):**
  - [https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4](https://www.youtube.com/watch?v=4TcP297mdsI&list=PL5Q2soXY2Zi_7UBNmC9B8Yr5JSwTG9yH4)

- **Youtube Livestream (2021):**
  - [https://www.youtube.com/playlist?list=PL5Q2soXY2Zi_JMBjHT5wHv-A2FSI54b2v](https://www.youtube.com/playlist?list=PL5Q2soXY2Zi_JMBjHT5wHv-A2FSI54b2v)

- **Critical analysis & presentation course**
  - Taken by Bachelor’s/Masters/PhD students
  - Cutting-edge research topics + talks + fundamentals in Computer Architecture
  - 20+ research papers, presentations, analyses, discussions, brainstorming
PIM Course (Spring 2022)

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

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

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
**Genomics (Spring 2022)**

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

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXYZi8NrPDgOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXYZi8NrPDgOR1yRU_Cxxjw-u18)

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

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

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=heterogeneous_systems](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](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](https://www.youtube.com/onurmutlulectures)
HW/SW Co-Design (Spring 2022)

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

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

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

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Solid-State Drives (Spring 2022)

- **Spring 2022 Edition:**

- **Youtube Livestream:**
  - [https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMI2RAq](https://www.youtube.com/watch?v=_q4rm71DsY4&list=PL5Q2soXY2Zi8vabcse1kL22DEcgMI2RAq)

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

[https://www.youtube.com/onurmutlulectures](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
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=aMllXRQd3s&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
An Interview on Computing Futures

https://www.youtube.com/watch?v=8ffSEKZhmvo
Principle: Teaching and Research

Teaching drives Research
Research drives Teaching
Research & Teaching: Some Overview Talks

- Future Computing Architectures
  - [https://www.youtube.com/watch?v=kgiZlSOcGFm&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=1](https://www.youtube.com/watch?v=kgiZlSOcGFm&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=1)

- Enabling In-Memory Computation
  - [https://www.youtube.com/watch?v=njX_1458Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=16](https://www.youtube.com/watch?v=njX_1458Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=16)

- Accelerating Genome Analysis
  - [https://www.youtube.com/watch?v=r7sn41lH4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=41](https://www.youtube.com/watch?v=r7sn41lH4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=41)

- Rethinking Memory System Design
  - [https://www.youtube.com/watch?v=F7xZLKMY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3](https://www.youtube.com/watch?v=F7xZLKMY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3)

- Intelligent Architectures for Intelligent Machines
  - [https://www.youtube.com/watch?v=c6_LqzuNdkw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=25](https://www.youtube.com/watch?v=c6_LqzuNdkw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=25)

- The Story of RowHammer
  - [https://www.youtube.com/watch?v=s57PHQO1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=39](https://www.youtube.com/watch?v=s57PHQO1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=39)
SAFARI Live Seminars

SAFARI Live Seminars in Computer Architecture
Dr. Juan Gómez Luna, ETH Zurich
Understanding a Modern Processing-in-Memory Architecture: Benchmarking and Experimental Characterization

SAFARI Live Seminars in Computer Architecture
Dr. Andrew Walker, Schiltron Corporation & Nexgen Power Systems
An Addiction to Low Cost Per Memory Bit – How to Recognize it and What to Do About It

SAFARI Live Seminars in Computer Architecture
Geraldo F. Oliveira, ETH Zurich
DAMON: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

SAFARI Live Seminars in Computer Architecture
Gennady Pakhmoun, University of Toronto
Efficient DNN Training at Scale: from Algorithms to Hardware

SAFARI Live Seminars in Computer Architecture
Jawad Haj-Yahya, Huawei Research Center Zurich
Power Management Mechanisms in Modern Microprocessors and Their Security Implications

SAFARI Live Seminars in Computer Architecture
Ataberk Olgun, TOBB & ETH Zurich
GuCAS-PTRNG: High-Throughput True Random Number Generation Using Quadruple Bus Activation in Commercial DRAM Chips

SAFARI Live Seminars in Computer Architecture
Minas Patel, ETH Zurich
Enabling Effective Error Mitigation in Memory Chips That Use On-Die ECCs

SAFARI Live Seminars in Computer Architecture
Christina Giannoula, National Technical University of Athens
Efficient Synchronization Support for Near-Data-Processing Architectures

SAFARI Live Seminars in Computer Architecture
Jawad Haj-Yahya, Huawei Research Center Zurich

https://safari.ethz.ch/safari-seminar-series/
Special Research Sessions & Courses

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

[Video Link: https://www.youtube.com/watch?v=queukNs5XI3g]
Open-Source Artifacts

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.

Overview
- 138 followers
- ETH Zurich and Carnegie Mellon U...
- https://safari.ethz.ch/
- omutlu@gmail.com

Pinned

- **ramulator** (Public)
  - A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...
  - C++
  - 371
  - 173

- **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++
  - 163
  - 97

- **prim-benchmarks** (Public)
  - PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publi...
  - C
  - 65
  - 25

- **rowhammer** (Public)
  - C
  - 196
  - 41

- **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
  - 40
  - 7

- **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
  - 91
  - 27

[https://github.com/CMU-SAFAIRI/](https://github.com/CMU-SAFAIRI/)
Papers, Talks, Videos, Artifacts

- All are available at

  https://people.inf.ethz.ch/omutlu/projects.htm

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/
Example Research Topics: Quick Overview
Many Interesting Things Are Happening Today in Computer Architecture

Performance
Energy Efficiency
Sustainability
Many Interesting Things Are Happening Today in Computer Architecture

Reliability
Safety
Security
Privacy
Many Interesting Things Are Happening Today in Computer Architecture

More Demanding Workloads
Many Interesting Things Are Happening Today in Computer Architecture

New (Device) Technologies
SAFARI Research & Brainstorming Meetings

- PIM (weekly) – **Thu 10:30am & Fri 4pm**
- AI/ML (weekly) – **Tue 4pm**
- Storage (weekly) – **Mon 10:30am**
- DRAM (weekly) – **Wed 2:30pm**
- Bioinformatics (weekly) – **Tue 1pm**
- Emerging Tech & Apps (weekly) – **Mon 2pm**
- HW/SW + Security (weekly) – **Wed 11am**
- Reading Group (weekly) – **Mon 5pm**
- General Group Meeting – **Fri 5pm**

- Also: Meetings with Industry (regular or on-demand)
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!
SAFARI Research Group
Introduction & Research

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

29 February 2024
Computer Architecture Seminar
Backup Slides
(A Bit Old)
Build fundamentally better architectures
Four Key Issues in Future Platforms

- Fundamentally **Secure/Reliable/Safe** Architectures

- Fundamentally **Energy-Efficient** Architectures
  - **Memory-centric** (Data-centric) Architectures

- Fundamentally **Low-Latency and Predictable** Architectures

- Architectures for **AI/ML, Genomics, Medicine, Health**
Architectures for Intelligent Machines

Data-centric

Data-driven

Data-aware
The Problem

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

- Important workloads are all data intensive

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

- Data is increasing
  - We can generate more than we can process
Data is Key for Future Workloads

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

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

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

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

In-memory Databases

Graph/Tree Processing

Data → performance & energy bottleneck

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

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

Chrome
Google’s web browser

TensorFlow Mobile
Google’s machine learning framework

VP9
Video Playback
Google’s video codec

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

Chrome

TensorFlow Mobile

Data → performance & energy bottleneck

VP9

Video Playback

Google’s video codec

Video Capture

Google’s video codec
Data is Key for Future Workloads

Development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Genome Analysis

1. **Sequencing**
2. **Read Mapping**
3. **Variant Calling**
4. **Scientific Discovery**

Data → performance & energy bottleneck
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

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

**Published:** 02 April 2018  **Article history ▼**


[Open arxiv.org version]
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

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

*Briefings in Bioinformatics*, bby017, [https://doi.org/10.1093/bib/bby017](https://doi.org/10.1093/bib/bby017)

**Published:** 02 April 2018  **Article history** ▼
Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]
GenASMs Framework [MICRO 2020]

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

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

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

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

Damla Senol Cali⁴, Gurpreet S. Kalsi⁴, Zülal Bingöl⁴, Can Firtina⁴, Lavanya Subramanian⁶, Jeremie S. Kim⁶
Rachata Ausavarungnirun⁷, Mohammed Alser⁷, Juan Gomez-Luna⁷, Amirali Boroumand⁶, Anant Nori⁴
Allison Scibisz⁶, Sreenivas Subramoney⁴, Can Alkan⁷, Saugata Ghose⁶, Onur Mutlu⁶

⁴ Carnegie Mellon University ⁵ Processor Architecture Research Lab, Intel Labs ⁶ Bilkent University ⁷ ETH Zürich ⁸ Facebook ⁹ King Mongkut’s University of Technology North Bangkok  *University of Illinois at Urbana–Champaign

SAFARI
FPGA-based Processing Near Memory


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

Gagandeep Singh\* Mohammed Alser\* Damla Senol Cali\†
Dionysios Diamantopoulos\† Juan Gómez-Luna\‡
Henk Corporaal* Onur Mutlu\‡\*\†

\*ETH Zürich \*†Carnegie Mellon University
\*Eindhoven University of Technology \†IBM Research Europe

SAFARI
Future of Genome Sequencing & Analysis

MinION from ONT

SmidgION from ONT

More on Fast & Efficient Genome Analysis …

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

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

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

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

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - Accelerating Genomics (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqlgwiDRQDTyId

https://www.youtube.com/onurmutlulectures
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.
Data Overwhelms Modern Machines

Chrome

TensorFlow Mobile

Data → performance & energy bottleneck

VP9

Video Playback

Google’s video codec

Video Capture

Google’s video codec
62.7% of the total system energy is spent on data movement

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

\textsuperscript{1}SAFARI
A memory access consumes \(~100-1000\times\) the energy of a complex addition.
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: Architectures Today …

Architectures are terrible at dealing with data
- Designed to mainly store and move data vs. to compute
- They are processor-centric as opposed to data-centric

Architectures are terrible at taking advantage of vast amounts of data (and metadata) available to them
- Designed to make simple decisions, ignoring lots of data
- They make human-driven decisions vs. data-driven

Architectures are terrible at knowing and exploiting different properties of application data
- Designed to treat all data as the same
- They make component-aware decisions vs. data-aware
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
Processing Data
Where It Makes Sense
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.
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**
Challenge and Opportunity for Future Computing Architectures with Minimal Data Movement
Challenge and Opportunity for Future

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

Fundamentally

High-Performance

(Data-Centric)

Computing Architectures
Goal: Processing Inside Memory

- Many questions ... How do we design the:
  - compute-capable memory & controllers?
  - processor chip and in-memory units?
  - software and hardware interfaces?
  - system software, compilers, languages?
  - algorithms and theoretical foundations?
Mindset: Memory-Centric Computing

Memory similar to a “conventional” accelerator
A Modern Primer on Processing in Memory

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

\textit{SAFARI Research Group}

\textsuperscript{a}ETH Zürich
\textsuperscript{b}Carnegie Mellon University
\textsuperscript{c}University of Illinois at Urbana-Champaign
\textsuperscript{d}King Mongkut’s University of Technology North Bangkok


\textbf{SAFARI}

A Modern Primer on Processing in Memory

Onur Mutlu, Saugata Ghose, Juan Gómez-Luna, Rachata Ausavarungnirun

SAFARI Research Group

ETH Zürich
Carnegie Mellon University
University of Illinois at Urbana-Champaign
King Mongkut's University of Technology North Bangkok

Abstract

Modern computing systems are overwhelmingly designed to move data to computation. This design choice goes directly against at least three key trends in computing that cause performance, scalability and energy bottlenecks: (1) data access is a key bottleneck as many important applications are increasingly data-intensive, and memory bandwidth and power do not scale well, (2) energy consumption is a key limiter in almost all computing platforms, especially server and mobile systems, (3) data movement, especially off-chip to on-chip, is very expensive in terms of bandwidth, energy and latency, much more so than computation. These trends are especially severely-felt in the data-intensive server and energy-constrained mobile systems of today.

At the same time, conventional memory technology is facing many technology scaling challenges in terms of reliability, energy, and performance. As a result, memory system architects are open to organizing memory in different ways and making it more intelligent, at the expense of higher cost. The emergence of 3D-stacked memory plus logic, the adoption of error correcting codes inside the latest DRAM chips, proliferation of different main memory standards and chips, specialized for different purposes (e.g., graphics, low-power, high bandwidth, low latency), and the necessity of designing new solutions to serious reliability and security issues, such as the RowHammer phenomenon, are an evidence of this trend.

This chapter discusses recent research that aims to practically enable computation close to data, an approach we call processing-in-memory (PIM). PIM places computation mechanisms in or near where the data is stored (i.e., inside the memory chips, in the logic layer of 3D-stacked memory, or in the memory controllers), so that data movement between the computation units and memory is reduced or eliminated. While the general idea of PIM is not new, we discuss motivating trends in applications as well as memory circuits/technology that greatly exacerbate the need for enabling it in modern computing systems. We examine at least two promising new approaches to designing PIM systems to accelerate important data-intensive applications: (1) processing using memory by exploiting analog operational properties of DRAM chips to perform massively-parallel operations in memory, with low-cost changes, (2) processing near memory by exploiting 3D-stacked memory technology design to provide high memory bandwidth and low memory latency to in-memory logic. In both approaches, we describe and tackle relevant cross-layer research, design, and adoption challenges in devices, architecture, systems, and programming models. Our focus is on the development of in-memory processing designs that can be adopted in real computing platforms at low cost. We conclude by discussing work on solving key challenges to the practical adoption of PIM.

Keywords: memory systems, data movement, main memory, processing-in-memory, near-data processing, computation-in-memory, processing using memory, processing near memory, 3D-stacked memory, non-volatile memory, energy efficiency, high-performance computing, computer architecture, computing paradigm, emerging technologies, memory scaling, technology scaling, dependable systems, robust systems, hardware security, system security, latency, low-latency computing
1. Introduction

Main memory, built using the Dynamic Random Access Memory (DRAM) technology, is a major component in nearly all computing systems, including servers, cloud platforms, mobile/embedded devices, and sensor systems. Across all of these systems, the data working set sizes of modern applications are rapidly growing, while the need for fast analysis of such data is increasing. Thus, main memory is becoming an increasingly significant bottleneck across a wide variety of computing systems and applications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Alleviating the main memory bottleneck requires the memory capacity, energy, cost, and performance to all scale in an efficient manner across technology generations. Unfortunately, it has become increasingly difficult in recent years, especially the past decade, to scale all of these dimensions [1, 2, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49], and thus the main memory bottleneck has been worsening.

A major reason for the main memory bottleneck is the high energy and latency cost associated with data movement. In modern computers, to perform any operation on data that resides in main memory, the processor must retrieve the data from main memory. This requires the memory controller to issue commands to a DRAM module across a relatively slow and power-hungry off-chip bus (known as the memory channel). The DRAM module sends the requested data across the memory channel, after which the data is placed in the caches and registers. The CPU can perform computation on the data once the data is in its registers. Data movement from the DRAM to the CPU incurs long latency and consumes a significant amount of energy [7, 50, 51, 52, 53, 54]. These costs are often exacerbated by the fact that much of the data brought into the caches is not reused by the CPU [52, 53, 55, 56], providing little benefit in return for the high latency and energy cost.

The cost of data movement is a fundamental issue with the processor-centric nature of contemporary computer systems. The CPU is considered to be the master in the system, and computation is performed only in the processor (and accelerators). In contrast, data storage and communication units, including the main memory, are treated as unintelligent workers that are incapable of computation. As a result of this processor-centric design paradigm, data moves a lot in the system between the computation units and communication/storage units so that computation can be done on it. With the increasingly data-centric nature of contemporary and emerging appli-
A Workload and Programming Ease Driven Perspective of Processing-in-Memory

Saugata Ghose† Amirali Boroumand† Jeremie S. Kim†§ Juan Gómez-Luna§ Onur Mutlu§†

†Carnegie Mellon University §ETH Zürich

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

SAFARI

Processing in Memory: Two Approaches

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

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

- New memory technologies enable even more opportunities
  - Memristors, resistive RAM, phase change mem, STT-MRAM, ...
  - Can operate on data with minimal movement
Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

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

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]
In-DRAM Bulk Bitwise Execution Paradigm


---

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**


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

*Nastaran Hajinazar¹,²  Nika Mansouri Ghiasi¹  Geraldo F. Oliveira¹  Minesh Patel¹  Sven Gregorio¹  Mohammed Alser¹  João Dinis Ferreira¹  Juan Gómez-Luna¹  Onur Mutlu¹  Saugata Ghose³

¹ETH Zürich  ²Simon Fraser University  ³University of Illinois at Urbana–Champaign
SIMDRAM Key Idea

• **SIMDRAM**: An end-to-end processing-using-DRAM framework that provides the **programming interface**, the **ISA**, and the **hardware support** for:
  
  - **Efficiently** computing **complex** operations in DRAM
  
  - Providing the ability to implement **arbitrary** operations as required
  
  - Using an **in-DRAM massively-parallel SIMD substrate** that requires **minimal** changes to DRAM architecture
SIMDRAM Framework: Overview

Step 1: Generate MAJ logic

Step 2: Generate sequence of DRAM commands

Step 3: Execution according to μProgram

SIMDRAM Output

New SIMDRAm μProgram

Main memory

μProgram

New SIMDRAm instruction

SIMDRAM Output

Instruction result in memory
SIMDRAM Key Results

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

SIMDRAM provides:

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

- **SIMDRAM:**
  - Enables *efficient* computation of a *flexible* set and wide range of operations in a PuM *massively parallel* SIMD substrate
  - Provides the hardware, programming, and ISA support, to:
    - Address key *system integration* challenges
    - Allow programmers to define and employ *new operations* without hardware changes

---

**SIMDRAM is a promising PuM framework**

- Can *ease the adoption* of processing-using-DRAM architectures
- Improves the *performance* and *efficiency* of processing-using-memory architectures
Processing in Memory: Two Approaches

1. Processing using Memory
2. Processing near Memory
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 Processing-in-DRAM Engine (2019)**


UPMEM Memory Modules

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

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

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJI, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOLIA, 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 data 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 UPNEM company has designed and manufactured the first publicly available real-world PIM architecture. The UPNEM-PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called Data-Driven Processing Units (DPU), integrated in the same die.

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 UPNEM-PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PIM-Bench (PIM benchmarking), a benchmark suite of 16 workloads from different application domains (e.g., dense-sparsity 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 PIM benchmarks on the UPNEM-PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPNEM-based PIM systems with 140 and 2,560 DPU provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware architecture designers of future PIM systems.

More on the UPMEM PIM System

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgxBxUz7xRPS-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 GIANNOLA, 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.

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

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

PrIM (Processing-In-Memory Benchmarks)

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

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

PrIM also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.
Key Takeaway 1

The UPMEM PIM architecture is fundamentally compute bound. As a result, the most suitable workloads are memory-bound.
KEY TAKEAWAY 2

The most well-suited workloads for the UPMEM PIM architecture use no arithmetic operations or use only simple operations (e.g., bitwise operations and integer addition/subtraction).
Key Takeaway 3

The most well-suited workloads for the UPMEM PIM architecture require little or no communication across DPUs (inter-DPU communication).
**Key Takeaway 4**

- UPMEM-based PIM systems **outperform state-of-the-art CPUs** in terms of performance and energy efficiency on most of PrIM benchmarks.

- UPMEM-based PIM systems **outperform state-of-the-art GPUs on a majority of PrIM benchmarks**, and the outlook is even more positive for future PIM systems.

- UPMEM-based PIM systems are **more energy-efficient than state-of-the-art CPUs and GPUs** on workloads that they provide **performance improvements** over the CPUs and the GPUs.
More on UPMEM System & Analysis

Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu, "Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture"
[arXiv preprint]
[PrIM Benchmarks Source Code]
[Slides (pptx) (pdf)]
[Long Talk Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[SAFARI Live Seminar Slides (pptx) (pdf)]
[SAFARI Live Seminar Video (2 hrs 57 mins)]
[Lightning Talk Video (3 minutes)]
[Short Talk Video (21 minutes)]
[1-hour Talk Video (58 minutes)]

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

Juan Gómez-Luna¹  Izzat El Hajj²  Ivan Fernandez¹,³  Christina Giannoula¹,⁴  Geraldo F. Oliveira¹  Onur Mutlu¹

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

SAFARI

https://github.com/CMU-SAFARI/prim-benchmarks
More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

• There is no direct communication channel between DPUs

• Inter-DPU communication takes place via the host CPU using CPU-DPU and DPU-CPU transfers

• Example communication patterns:
  - Merging of partial results to obtain the final result
  - Only DPU-CPU transfers
  - Redistribution of intermediate results for further computation
  - DPU-CPU transfers and CPU-DPU transfers

SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture
1,868 views • Streamed live on Jul 12, 2021

Onur Mutlu Lectures
17.6K subscribers

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

https://www.youtube.com/watch?v=D8Hiy2iU9j&list=PL5Q2soXY2Zi_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Data Movement in Computing Systems

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

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

https://www.youtube.com/watch?v=Pp9jSU2b9oM&list=PL5Q2soXY2Zi8_VVChACnON4sfh2bJ5IrD&index=159
FPGA-based Processing Near Memory


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

Gagandeep Singh◊ Mohammed Alser◊ Damla Senol Cali★
Dionysios Diamantopoulos▼ Juan Gómez-Luna◊
Henk Corporaal★ Onur Mutlu★

◊ETH Zürich ★Carnegie Mellon University
▼Eindhoven University of Technology ▼IBM Research Europe
DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

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

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

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

Mobile consumer workloads
(GoogleWL²)

Near-Data Processing

Graph processing
(Tesseract¹)

Databases
(Polynesia⁵)

Time series analysis
(NATSA⁶)

Neural networks
(GoogleWL²)

DNA sequence mapping
(GenASM³; GRIM-Filter⁴)

...
Key Approach

• New **workload characterization methodology** to analyze:
  - data movement bottlenecks
  - suitability of different data movement mitigation mechanisms

• Two main profiling strategies:

  **Architecture-independent profiling:**
  characterizes the memory behavior *independently* of the underlying *hardware*

  **Architecture-dependent profiling:**
  evaluates the *impact of the system configuration on the memory behavior*
Methodology Overview

**Step 1** Application Profiling
- **Profiler**
  - roi_begin
  - roi_end

**DAMOV-SIM Simulator**
- ld 0xFF
- st 0xAF
- ld 0xFF
- st 0xAF
- ld 0xFF

**Memory Traces**
- Scalability Analysis
- # Cores

**Methodology Output**
- Memory Bottleneck Classes
- SAFARI

**Step 2** Locality-based Clustering
- High
- Low

**Step 3** Memory Bottleneck Class.
- High
- Low

**User Input**
- Target Application
  - Source Code
  - User Input
Step 1: Application Profiling

- We analyze **345 applications** from distinct domains:
  - Graph Processing
  - Deep Neural Networks
  - Physics
  - High-Performance Computing
  - Genomics
  - Machine Learning
  - Databases
  - Data Reorganization
  - Image Processing
  - Map-Reduce
  - Benchmarking
  - Linear Algebra
  ...

SAFARI
Six classes of data movement bottlenecks:

- Each class ↔ data movement mitigation mechanism
DAMOV is Open Source

• We open-source our benchmark suite and our toolchain

DAMOV: 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

Get DAMOV at:

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

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

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

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

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

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

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

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXY2Zj_tOTAYm--dYByNPL7JhwR9&index=3
More on DAMOV

  - [arXiv preprint]
  - [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
Eliminating the Adoption Barriers

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

1. **Functionality** and **applications & software** for PIM

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

3. **System** support: coherence, synchronization, virtual memory

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

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Program/Language</td>
</tr>
<tr>
<td>System Software</td>
</tr>
<tr>
<td>SW/HW Interface</td>
</tr>
<tr>
<td>Micro-architecture</td>
</tr>
<tr>
<td>Logic</td>
</tr>
<tr>
<td>Devices</td>
</tr>
<tr>
<td>Electrons</td>
</tr>
</tbody>
</table>
Challenge and Opportunity for Future

Data-Driven (Self-Optimizing) Computing Architectures
Challenge and Opportunity for Future

Data-Aware
(Expressive)
Computing Architectures
Onur Mutlu,
"Memory-Centric Computing Systems"
[Slides (pptx) (pdf)]
[Executive Summary Slides (pptx) (pdf)]
[Tutorial Video (1 hour 51 minutes)]
[Executive Summary Video (2 minutes)]
[Abstract and Bio]
[Related Keynote Paper from VLSI-DAT 2020]
[Related Review Paper on Processing in Memory]

https://www.youtube.com/watch?v=H3sEaINPBOE

https://www.youtube.com/onurmutlulectures
Architectures for Intelligent Machines

Data-centric

Data-driven

Data-aware
More Detailed Research Overview
Slides from ISCA 2021
Mentoring Workshop Panel

Onur Mutlu,
"Applying to Graduate School & Doing Impactful Research"
Invited Panel Talk at the 3rd Undergraduate Mentoring Workshop, held with the 48th International Symposium on Computer Architecture (ISCA), Virtual, 18 June 2021.
[Slides (pptx) (pdf)]
[Talk Video (50 minutes)]
A Talk on Impactful Research & Teaching

Applying to Grad School & Doing Impactful Research

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
13 June 2020
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

SAFARI  ETH Zürich  Carnegie Mellon

Arch. Mentoring Workshop @ISCA’21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu
1,563 views • Premiered Jun 16, 2021

Onur Mutlu Lectures
17.2K subscribers

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021
(https://sites.google.com/wisc.edu/uar...)

https://www.youtube.com/watch?v=83tlorht7Mc&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=54
Example Research Topics:
Quick Overview
Challenge and Opportunity for Future

High Performance

(to solve the toughest & all problems)
Personalized and Private

(in every aspect of life: health, medicine, spaces, devices, robotics, ...)
Accelerating Genome Analysis

Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"

[Slides (pptx)(pdf)]
[Talk Video (1 hour 2 minutes)]

Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

Saugata Ghose
University of Illinois at Urbana-Champaign and Carnegie Mellon University

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and Bilkent University
GenASM Framework [MICRO 2020]


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

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

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  

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

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

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

Published: 02 April 2018   Article history ▼

Future of Genome Sequencing & Analysis

MinION from ONT

SmidgION from ONT

More on Fast & Efficient Genome Analysis

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

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

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

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

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - Accelerating Genomics (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqlgwIDRQDTyId

SAFARI  
https://www.youtube.com/onurmutlulectures
The Problem

Computing is Bottlenecked by Data
Modern Systems are Bottlenecked by Data Storage and Movement
Modern Systems are Bottlenecked by Memory
Memory Scaling: A Systems Architecture Perspective

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

Challenge and Opportunity for Future

Fundamentally Secure, Reliable, Safe Computing Architectures
Infrastructures to Understand Such Issues

An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)

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

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)
Infrastructures to Understand Such Issues


- Flexible
- Easy to Use (C++ API)
- Open-source

[https://github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)
SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

Hasan Hassan$^{1,2,3}$, Nandita Vijaykumar$^3$, Samira Khan$^{4,3}$, Saugata Ghose$^3$, Kevin Chang$^3$, Gennady Pekhimenko$^{5,3}$, Donghyuk Lee$^{6,3}$, Oguz Ergin$^2$, Onur Mutlu$^{1,3}$

$^1$ETH Zürich $^2$TOBB University of Economics & Technology $^3$Carnegie Mellon University
$^4$University of Virginia $^5$Microsoft Research $^6$NVIDIA Research
One can predictably induce errors in most DRAM memory chips.
A simple hardware failure mechanism can create a widespread system security vulnerability.

Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG  SECURITY  08.31.16  7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS
One Can Take Over an Otherwise-Secure System

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

Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology

Project Zero

News and updates from the Project Zero team at Google

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn+, 2015)
Rowhammer
First RowHammer Analysis

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]
Future of Memory Reliability/Security

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

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

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu

Onur Mutlu and Jeremie Kim, "RowHammer: A Retrospective" 
[Preliminary arXiv version]
[Slides from COSADE 2019 (pptx)]
[Slides from VLSI-SOC 2020 (pptx) (pdf)]
[Talk Video (30 minutes)]

---

**RowHammer: A Retrospective**

Onur Mutlu§‡ 
§ETH Zürich

Jeremie S. Kim‡§ 
‡Carnegie Mellon University
RowHammer in 2020
RowHammer in 2020 (I)

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,

"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (20 minutes)]
[Lightning Talk Video (3 minutes)]

Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

Jeremie S. Kim§† Minesh Patel§ A. Giray Yağlıkçı§
Hasan Hassan§ Roknoddin Azizi§ Lois Orosa§ Onur Mutlu§†

§ETH Zürich †Carnegie Mellon University
RowHammer in 2020 (II)

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,

"TRRespass: Exploiting the Many Sides of Target Row Refresh"


[Slides (pptx) (pdf)]
[Lecture Slides (pptx) (pdf)]
[Talk Video (17 minutes)]
[Lecture Video (59 minutes)]
[Source Code]
[Web Article]

Best paper award.
Pwnie Award 2020 for Most Innovative Research. Pwnie Awards 2020

TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo*† Emanuele Vannacci*† Hasan Hassan§ Victor van der Veen¶
Onur Mutlu§ Cristiano Giuffrida* Herbert Bos* Kaveh Razavi*

*Vrije Universiteit Amsterdam §ETH Zürich ¶Qualcomm Technologies Inc.
RowHammer is still an open problem

Security by obscurity is likely not a good solution
Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers

Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,

"Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"
[Slides (pptx) (pdf)]
[Talk Video (17 minutes)]
BlockHammer Solution in 2021

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

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


[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Talk Video (22 minutes)]
[Short Talk Video (7 minutes)]
Detailed Lectures on RowHammer

- **Computer Architecture, Fall 2020, Lecture 4b**
  - RowHammer (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=KDY632z23UE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=8](https://www.youtube.com/watch?v=KDY632z23UE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=8)

- **Computer Architecture, Fall 2020, Lecture 5a**
  - RowHammer in 2020: TRRespass (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=pwRw7QqK_qA&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=9](https://www.youtube.com/watch?v=pwRw7QqK_qA&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=9)

- **Computer Architecture, Fall 2020, Lecture 5b**
  - RowHammer in 2020: Revisiting RowHammer (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=qR7XREepecq&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=10](https://www.youtube.com/watch?v=qR7XREepecq&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=10)

- **Computer Architecture, Fall 2020, Lecture 5c**
  - Secure and Reliable Memory (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=HvswnsfG3oQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=11](https://www.youtube.com/watch?v=HvswnsfG3oQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=11)

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Onur Mutlu,
"The Story of RowHammer"
Keynote Talk at Secure Hardware, Architectures, and Operating Systems Workshop (SeHAS), held with HiPEAC 2021 Conference, Virtual, 19 January 2021.
[Slides (pptx) (pdf)]
[Talk Video (1 hr 15 minutes, with Q&A)]
Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD’s reliability and lifetime.

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642
Understand and Model with Experiments (Flash)

One Important Takeaway

Main Memory Needs
Intelligent Controllers
Challenge and Opportunity for Future

Fundamentally
Low-Latency
Computing Architectures
Truly Reducing Memory Latency
Tiered-Latency DRAM

- Donghyuk Lee, Yoongu Kim, Vivek Seshadri, Jamie Liu, Lavanya Subramanian, and Onur Mutlu,

"Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture"

Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013. Slides (pptx)

Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture

Donghyuk Lee    Yoongu Kim    Vivek Seshadri    Jamie Liu    Lavanya Subramanian    Onur Mutlu
Carnegie Mellon University
Adaptive-Latency DRAM

Donghyuk Lee, Yoongu Kim, Gennady Pekhimenko, Samira Khan, Vivek Seshadri, Kevin Chang, and Onur Mutlu, "Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case"
[Slides (pptx) (pdf)] [Full data sets]
Analysis of Latency Variation in DRAM Chips

Kevin Chang, Abhijith Kashyap, Hasan Hassan, Samira Khan, Kevin Hsieh, Donghyuk Lee, Saugata Ghose, Gennady Pekhimenko, Tianshi Li, and Onur Mutlu,

"Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization"
[Slides (pptx) (pdf)]
[Source Code]
Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms

Donghyuk Lee, NVIDIA and Carnegie Mellon University
Samira Khan, University of Virginia
Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Carnegie Mellon University
Gennady Pekhimenko, Vivek Seshadri, Microsoft Research
Onur Mutlu, ETH Zürich and Carnegie Mellon University
Solar-DRAM: Putting It Together


Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines

Jeremie S. Kim‡§, Minesh Patel§, Hasan Hassan§, Onur Mutlu§‡
‡Carnegie Mellon University §ETH Zürich
CLR-DRAM: Capacity-Latency Reconfigurability

- Haocong Luo, Taha Shahroodi, Hasan Hassan, Minesh Patel, A. Giray Yaglikci, Lois Orosa, Jisung Park, and Onur Mutlu,

"CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off"


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (20 minutes)]
[Lightning Talk Video (3 minutes)]

CLR-DRAM: A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-Off

Haocong Luo§† Taha Shahroodi§ Hasan Hassan§ Minesh Patel§
A. Giray Yağlıkçı§ Lois Orosa§ Jisung Park§ Onur Mutlu§

§ETH Zürich †ShanghaiTech University
Low-Latency Solid-State Drives (SSDs)

- Jisung Park, Myungssuk Kim, Myoungjun Chun, Lois Orosa, Jihong Kim, and Onur Mutlu,

"Reducing Solid-State Drive Read Latency by Optimizing Read-Retry"


[2-page Extended Abstract]
[Short Talk Slides (pptx) (pdf)]
[Full Talk Slides (pptx) (pdf)]
[Short Talk Video (5 mins)]
[Full Talk Video (19 mins)]

Reducing Solid-State Drive Read Latency by Optimizing Read-Retry

Jisung Park¹  Myungsuk Kim²,³  Myoungjun Chun²  Lois Orosa¹  Jihong Kim²  Onur Mutlu¹

¹ETH Zürich  Switzerland  ²Seoul National University  Republic of Korea  ³Kyungpook National University  Republic of Korea
Lectures on Low-Latency Memory

- Computer Architecture, Fall 2020, Lecture 10
  - Low-Latency Memory (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=vQd1YgOH1Mw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=19

- Computer Architecture, Fall 2020, Lecture 12b
  - Capacity-Latency Reconfigurable DRAM (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=DUtPFW3jxq4&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=23

- Computer Architecture, Fall 2019, Lecture 11a
  - DRAM Latency PUF (ETH Zürich, Fall 2019)
  - https://www.youtube.com/watch?v=7gqnrTZpjxE&list=PL5Q2soXY2Zi-DyoI3HbqcdtUm9YWRR_z-&index=15

- Computer Architecture, Fall 2019, Lecture 11b
  - DRAM True Random Number Generator (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=Y3hPv1I5f8Y&list=PL5Q2soXY2Zi-DyoI3HbqcdtUm9YWRR_z-&index=16
A Tutorial on Low-Latency Memory

Computer Architecture
Lecture 10: Low-Latency Memory

Prof. Onur Mutlu
ETH Zürich
Fall 2020
22 October 2020

https://www.youtube.com/onurmutlulectures
**We Need to Revisit the Entire Stack**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Program/Language</th>
<th>System Software</th>
<th>SW/HW Interface</th>
<th>Micro-architecture</th>
<th>Logic</th>
<th>Devices</th>
<th>Electrons</th>
</tr>
</thead>
</table>

**We can get there step by step**
Some Open Source Tools (I)

- **Rowhammer** – Program to Induce RowHammer Errors
  - [https://github.com/CMU-SAFARI/rowhammer](https://github.com/CMU-SAFARI/rowhammer)

- **Ramulator** – Fast and Extensible DRAM Simulator
  - [https://github.com/CMU-SAFARI/ramulator](https://github.com/CMU-SAFARI/ramulator)

- **MemSim** – Simple Memory Simulator
  - [https://github.com/CMU-SAFARI/memsim](https://github.com/CMU-SAFARI/memsim)

- **NOCulator** – Flexible Network-on-Chip Simulator
  - [https://github.com/CMU-SAFARI/NOCulator](https://github.com/CMU-SAFARI/NOCulator)

- **SoftMC** – FPGA-Based DRAM Testing Infrastructure
  - [https://github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)

- **Other open-source software from my group**
  - [https://github.com/CMU-SAFARI/](https://github.com/CMU-SAFARI/)
  - [http://www.ece.cmu.edu/~safari/tools.html](http://www.ece.cmu.edu/~safari/tools.html)
Some Open Source Tools (II)

- MQSim – A Fast Modern SSD Simulator
  - https://github.com/CMU-SAFARI/MQSim

- Mosaic – GPU Simulator Supporting Concurrent Applications
  - https://github.com/CMU-SAFARI/Mosaic

- IMPICA – Processing in 3D-Stacked Memory Simulator
  - https://github.com/CMU-SAFARI/IMPICA

- SMLA – Detailed 3D-Stacked Memory Simulator
  - https://github.com/CMU-SAFARI/SMLA

- HWASim – Simulator for Heterogeneous CPU-HWA Systems
  - https://github.com/CMU-SAFARI/HWASim

- Other open-source software from my group
  - https://github.com/CMU-SAFARI/
  - http://www.ece.cmu.edu/~safari/tools.html
More Open Source Tools (III)

- A lot more open-source software from my group
  - [https://github.com/CMU-SAFARI/](https://github.com/CMU-SAFARI/)
**Ramulator-PIM**
A fast and flexible simulation infrastructure for exploring general-purpose processing-in-memory (PIM) architectures. Ramulator-PIM combines a widely-used simulator for out-of-order and in-order processors (ZSim) with Ramulator, a DRAM simulator with memory models for DDRx, LPDDRx, GDDRx, WiOx, HBMx, and HMCx. Ramulator is described in the IEEE ...  

- **C++**  
  - MIT  
  - 11  
  - 29  
  - 0  
  - Updated 19 days ago

---

**SMASH**
SMASH is a hardware-software cooperative mechanism that enables highly-efficient indexing and storage of sparse matrices. The key idea of SMASH is to compress sparse matrices with a hierarchical bitmap compression format that can be accelerated from hardware. Described by Kanellopoulos et al. (MICRO '19)  

https://people.inf.ethz.ch/omutlu/pub/SMA...

- **C**  
  - 1  
  - 6  
  - 0  
  - 0  
  - Updated on May 17

---

**MQSim**
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 implementations, steady-state SSD conditions, and the full end-to-end latency of requests in modern SSDs. It is described in detail in the FAST 2018 paper by A...  

- **C++**  
  - MIT  
  - 54  
  - 62  
  - 10  
  - 1  
  - Updated on May 15

---

**Apollo**
Apollo is an assembly polishing algorithm that attempts to correct the errors in an assembly. It can take multiple set of reads in a single run and polish the assemblies of genomes of any size. Described in the Bioinformatics journal paper (2020) by Firtina et al. at https://people.inf.ethz.ch/omutlu/pub/apollo...

- **C++**  
  - GPL-3.0  
  - 1  
  - 12  
  - 0  
  - 0  
  - Updated on May 10

---

**Ramulator**
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 IEEE CAL 2015 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/ramulator_dram_simulator-ieee-cal15.pdf

- **C++**  
  - MIT  
  - 93  
  - 170  
  - 37  
  - 2  
  - Updated on Apr 13

---

**Shifted-Hamming-Distance**

- **C**  
  - GPL-2.0  
  - 5  
  - 20  
  - 0  
  - 1  
  - Updated on Mar 29

---

**SneakySnake**
The first and the only pre-alignment filtering algorithm that works on all modern high-performance computing architectures. It works efficiently and fast on CPU, FPGA, and GPU architectures and that greatly (by more than two orders of magnitude) expedites sequence alignment calculation. Described by Alser et al. (preliminary version at https://a...

- **VHDL**  
  - GPL-3.0  
  - 3  
  - 11  
  - 0  
  - 0  
  - Updated on Mar 10

---

**AirLift**
AirLift is a tool that updates mapped reads from one reference genome to another. Unlike existing tools, it accounts for regions not shared between the two reference genomes and enables remapping across all parts of the references. Described by Kim et al. (preliminary version at http://arxiv.org/abs/1912.08735)

- **C**  
  - 0  
  - 3  
  - 0  
  - 0  
  - Updated on Feb 19

---

**GPGPUSim-Ramulator**
The source code for GPGPUSim+Ramulator simulator. In this version, GPGPUSim uses Ramulator to simulate the DRAM. This simulator is used to produce some of the...
End of Slides on More Detailed Research Overview