Brief Self Introduction

Onur Mutlu

- Full Professor @ ETH Zurich ITET (INFK), since September 2015
- Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- https://people.inf.ethz.ch/omutlu/
- omutlu@gmail.com (Best way to reach me)
- https://people.inf.ethz.ch/omutlu/projects.htm

Research and Teaching in:

- Computer architecture, computer systems, hardware security, bioinformatics
- Memory and storage systems
- Hardware security, safety, predictability
- Fault tolerance
- Hardware/software cooperation
- Architectures for bioinformatics, health, medicine
- ...
Current Research Mission

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

Build fundamentally better architectures
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)
Computing System

Leiserson+, "There's plenty of room at the Top: What will drive computer performance after Moore's law?", Science, 2020

<table>
<thead>
<tr>
<th>Technology</th>
<th>The Top</th>
<th>Opportunity</th>
<th>The Bottom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>Technology</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>ISA (Architecture)</td>
<td>ISA</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Algorithm</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Problem</td>
<td>Problem</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Program/Language</td>
<td>Program/Language</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Hardware architecture</td>
<td>Hardware architecture</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Logic</td>
<td>Logic</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Devices</td>
<td>Devices</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
<tr>
<td>Electrons</td>
<td>Electrons</td>
<td>Opportunity</td>
<td>The Bottom</td>
</tr>
</tbody>
</table>

Richard Feynman, "There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics", a lecture given at Caltech, 1959.

Image source: https://science.sciencemag.org/content/368/6495/eaam9744
Software & Hardware Optimizations

Multiplying Two 4096-by-4096 Matrices

```python
for i in xrange(4096):
    for j in xrange(4096):
        for k in xrange(4096):
            C[i][j] += A[i][k] * B[k][j]
```

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Running time (s)</th>
<th>Absolute speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>25,552.48</td>
<td>1x</td>
</tr>
<tr>
<td>Java</td>
<td>2,372.68</td>
<td>11x</td>
</tr>
<tr>
<td>C</td>
<td>542.67</td>
<td>47x</td>
</tr>
<tr>
<td>Parallel loops</td>
<td>69.80</td>
<td>366x</td>
</tr>
<tr>
<td>Parallel divide and conquer</td>
<td>3.80</td>
<td>6,727x</td>
</tr>
<tr>
<td>plus vectorization</td>
<td>1.10</td>
<td>23,224x</td>
</tr>
<tr>
<td>plus AVX intrinsics</td>
<td>0.41</td>
<td>62,806x</td>
</tr>
</tbody>
</table>

Leiserson+, "There’s plenty of room at the Top: What will drive computer performance after Moore’s law?", Science, 2020
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/Graph/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

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

Broad research spanning apps, systems, logic with architecture at the center
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!

https://safari.ethz.ch
SAFARI Newsletter December 2021 Edition

https://safari.ethz.ch/safari-newsletter-december-2021/
SAFARI PhD and Post-Doc Alumni

- [https://safari.ethz.ch/safari-alumni/](https://safari.ethz.ch/safari-alumni/)
- Minesh Patel (ETH Zurich)
- Damla Senol Cali (Bionano Genomics)
- Nastaran Hajinazar (ETH Zurich)
- Gagandeep Singh (ETH Zurich)
- Amirali Boroumand (Stanford Univ)
- Jeremie Kim (ETH Zurich)
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor)
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook)
- Mohammed Alser (ETH Zurich)
- Yixin Luo (Google)
- Kevin Chang (Facebook)
- Rachata Ausavarungnirun (KMUNTB, Assistant Professor)
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor)
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher)
- Yoongu Kim (Google)
- Lavanya Subramanian (Intel Labs → Facebook)
- Samira Khan (Univ. of Virginia, Assistant Professor)
- Saugata Ghose (Univ. of Illinois, Assistant Professor)
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)
- Lois Orosa (Galicia Supercomputing Center)
- Jisung Park (POSTECH, Assistant Professor)
Onur Mutlu,
"SAFARI Research Group: Introduction & Research"
Talk at ETH Future Computing Laboratory Welcome Workshop (EFCL), Virtual, 6 July 2021.
[Slides (pptx) (pdf)]
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

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
Principle: Teaching and Research

Teaching drives Research
Research drives Teaching
Online 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_uej3aY39YB5pfW4SJ7L1N

- **Advanced Computer Architecture Course**
  - Computer Architecture
  - Fall 2020 Edition:
    https://www.youtube.com/watch?v=c3mPdZA-Fmc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN

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_5EFgo&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
DDCA (Spring 2022)

- **Spring 2022 Edition:**
  - https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule

- **Spring 2021 Edition:**

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=cpXde3HwyK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6

- **Youtube Livestream (Spring 2021):**
  - https://www.youtube.com/watch?v=LbCOEZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LIN

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

https://www.youtube.com/onurmutlulectures
Seminar (Spring’21)

- [Link to Schedule](https://safari.ethz.ch/architecture_seminar/spring2021/doku.php?id=schedule)

- [Link to YouTube Playlist](https://www.youtube.com/watch?v=t3m93ZpLOyw&list=PL5Q2soXY2Zi_awYdjmWVIUegsbY7TPGW4)

- **Critical analysis course**
  - Taken by Bachelor’s/Masters/PhD students
  - Cutting-edge research topics + fundamentals in Computer Architecture
  - 20+ research papers, presentations, analyses

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Livestream</th>
<th>Lecture</th>
<th>Readings</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>29.02  Thu.</td>
<td>Live</td>
<td>L1: Introduction and Basics</td>
<td>(PDF) (PPT)</td>
<td>Suggested</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Optional Lecture: Design Fundamentals</td>
<td>(PDF) (PPT)</td>
<td></td>
</tr>
<tr>
<td>W2</td>
<td>04.03  Thu.</td>
<td>Live</td>
<td>L2: Course Logistics</td>
<td>(PDF) (PPT)</td>
<td>Suggested</td>
</tr>
<tr>
<td>W3</td>
<td>11.03  Thu.</td>
<td>Live</td>
<td>L3: Example Review: RowClone Partitioning</td>
<td>(PDF) (PPT)</td>
<td>Suggested</td>
</tr>
<tr>
<td>W4</td>
<td>18.03  Thu.</td>
<td>Live</td>
<td>L4: Example Review: GateKeeper</td>
<td>(PDF) (PPT)</td>
<td>Suggested</td>
</tr>
<tr>
<td>W5</td>
<td>25.03  Thu.</td>
<td>Live</td>
<td>S1.1: Spectre Attacks, Exploring Sporadic Execution, SAP 2019</td>
<td>(PDF)</td>
<td>Mentioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S1.2: Blockhammer, Preventing RowHammer at Low Cost by Blocking Rapidly Accessed DRAM Rows, HPCA 2021</td>
<td>(PPT) (PDF)</td>
<td></td>
</tr>
<tr>
<td>W6</td>
<td>01.04  Thu.</td>
<td>Live</td>
<td>S2.1: D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput, HPCA 2019</td>
<td>(PPT) (PDF)</td>
<td>Mentioned</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S2.2: ComputeDRAM: In-Memory Compute Using Off-the-Shelf DRAM, MICRO 2019</td>
<td>(PDF)</td>
<td></td>
</tr>
<tr>
<td>W7</td>
<td>15.04  Thu.</td>
<td>Live</td>
<td>S3.1: PIM Enabled Instructions: A Line-Oriented, Locality-Aware Processing-in-Memory Architecture, MICRO 2021</td>
<td>(PDF)</td>
<td>Mentioned</td>
</tr>
</tbody>
</table>
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=9e4Chnwdo4o&list=PL5Q2soXY2Zi-841fUYUYUK9EsXKhQKRPyX](https://www.youtube.com/watch?v=9e4Chnwdo4o&list=PL5Q2soXY2Zi-841fUYUYUK9EsXKhQKRPyX)

- **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=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18](https://www.youtube.com/watch?v=DEL5A_Y3TI&list=PL5Q2soXY2Zi8NrPDgOR1yRU_Cxxjw-u18)

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

- [https://www.youtube.com/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

- **Youtube Livestream:**
  - https://www.youtube.com/watch?v=oF5fTrgFIY&list=PL5Q2soXYZi9XrgXR38lM_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
HW/SW Co-Design (Spring 2022)

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

- **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)
SSD Course (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)
Projects & Seminars: SoftMC
FPGA-Based Exploration of DRAM and RowHammer (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc](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](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](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](https://www.youtube.com/onurmutlulectures)
Projects & Seminars: Ramulator
Exploration of Emerging Memory Systems (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator](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](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=aMI1XROq3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV](https://www.youtube.com/watch?v=aMI1XROq3s&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
Hands-On Projects & Seminars Courses

- https://safari.ethz.ch/projects_and_seminars/doku.php

SAFARI Project & Seminars Courses (Spring 2022)

Welcome to the wiki for Project and Seminar courses SAFARI offers.

Courses we offer:

- Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments
- Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator
- Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
- Genome Sequencing on Mobile Devices
- Exploring the Processing-In-Memory Paradigm for Future Computing Systems
- Hands-on Acceleration on Heterogeneous Computing Systems
- Understanding and Designing Modern NAND Flash-Based Solid-State Drives (SSDs)
- Intelligent Architectures using Hardware/Software Cooperative Techniques
Focus on Insight
Encourage New Ideas
Principle: Learning and Scholarship

Focus on learning and scholarship
SAFARI Live Seminars (I)

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
Gerald F. Oliveira, ETH Zurich
DAMON: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

SAFARI Live Seminars in Computer Architecture
Gennady Palchikho, 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
Overview of a Modern SoC Architecture
• 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
• Several voltage sources exist, and some of them are shared between domains
• IO controllers and engines, IO interconnect, memory controller, and DDR typically each has an independent clock

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

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

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

Experimental Methodology
• We experimentally study three modern Intel processors
  • Haswell, Coffee-Lake, and Corinnes Lake
• We measure voltage and current using a Data Acquisition card (DataVUQ)

https://safari.ethz.ch/safari-seminar-series/
SAFARI Live Seminars (II)

SAFARI Live Seminar: Nastaran Hajinazar 27 Oct 2021
Posted on October 1, 2021 by event

SAFARI Live Seminar: Damla Senol Cali 07 Nov 2021
Posted on October 18, 2021 by event

SAFARI Live Seminar: Gennady Pekhimenko 08 Nov 2021
Posted on November 1, 2021 by event

SAFARI Live Seminar: Serghei Mangul 11 Nov 2021
Posted on November 6, 2021 by event

SAFARI Live Seminar: Rahul Bera 20 Dec 2021
Posted on January 18, 2022 by event

SAFARI Live Seminar: Loïs Orosa, 10 Feb 2022
Posted on January 18, 2022 by event

UPMEM PIM DRAM (1/2)

8 x 32-bit CPU added to a 4GB DRAM die:
- First Gen: 8 x CPU @ 660MHz, 8 x 64 MB banks (1 CPU for 1 bank)
- Second Gen: 8 x CPU @ 660MHz, 16 x 32 MB banks (1 CPU for 2 banks), secure Endi-ware

Multi-threaded CPU:
- In order execution at the thread level
- Out of order execution between threads when executing DMA instructions

Offering/Roadmap:
- 1st Gen: 24 hardware threads, scalar
- 2nd Gen: 16 hardware threads, scalar
- In production
- In design
- Planning

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zi_tOTAYm--dYbyNPL7JhwR9&index=1
SAFARI Live Seminar: Sudhanva Gurumurthi, Oct 25 2022
Posted on September 6, 2022 by ewent

We're excited to have Sudhanva Gurumurthi with us for our upcoming SAFARI Live Seminar!

Date: Tuesday, October 25 at 4:00 pm Zurich time (CEST)
Speaker: Sudhanva Gurumurthi, AMD Fellow
Link: Livestream on YouTube Link

Title: HBM3 RAS: The Journey to Enhancing Die-Stacked DRAM Resilience at Scale

https://safari.ethz.ch/safari-seminar-series/
SAFARI Live Seminar: Lana Josipović, Nov 7 2022

Posted on September 9, 2022 by ewent

Join us for our upcoming SAFARI Live Seminar

Date: Monday, November 7 at 4:00 pm Zurich time (CET)
Speaker: Lana Josipovic, DYNAMO Research Group, ETH Zurich
Link: Livestream on YouTube Link

Title: From C/C++ Code to High-Performance Dataflow Circuits

https://safari.ethz.ch/safari-seminar-series/
SAFARI Live Seminar, Christina Giannoula, Nov 9 2022
Posted on September 15, 2022 by ewen

Join us for our upcoming SAFARI Live Seminar

Date: Wednesday, November 9 at 4:00 pm Zurich time (CET)
Speaker: Christina Giannoula, School of Electrical and Computer Engineering, NTUA
Link: Livestream on YouTube Link

Title: Accelerating Irregular Applications via Efficient Synchronization and Data Access Techniques

https://safari.ethz.ch/safari-seminar-series/
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.

- ETH Zurich and Carnegie Mellon University
- https://safari.ethz.ch/
- omutlu@gmail.com

Overview

Repositories 71
Projects
Packages
Teams 1
People 44
Settings

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++ 311 161

- **prim-benchmarks** Public
  - PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...
  - C 53 21

- **DAMOV** Public
  - 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 processin...
  - C++ 26 4

- **SneakySnake** Public
  - SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude...
  - VHDL 41 8

- **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++ 146 93

- **rowhammer** Public
  - C 189 41

https://github.com/CMU-SAFARI/
Some Open Source Tools (I)

- Rowhammer – Program to Induce RowHammer Errors
  - [https://github.com/CMU-SAFA/rowhammer](https://github.com/CMU-SAFA/rowhammer)
- Ramulator – Fast and Extensible DRAM Simulator
  - [https://github.com/CMU-SAFA/ramulator](https://github.com/CMU-SAFA/ramulator)
- MemSim – Simple Memory Simulator
  - [https://github.com/CMU-SAFA/memsim](https://github.com/CMU-SAFA/memsim)
- NOCulator – Flexible Network-on-Chip Simulator
  - [https://github.com/CMU-SAFA/NOCulator](https://github.com/CMU-SAFA/NOCulator)
- SoftMC – FPGA-Based DRAM Testing Infrastructure
  - [https://github.com/CMU-SAFA/SoftMC](https://github.com/CMU-SAFA/SoftMC)

- Other open-source software from my group
  - [https://github.com/CMU-SAFA/](https://github.com/CMU-SAFA/)
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/
More Open Source Tools (III)

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.

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++ ✮ 311 ✭ 161

---

**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 ✮ 53 ✭ 21

---

**DAMOV** Public

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 processin...

- C++ ✮ 26 ✭ 4

---

**SneakySnake** Public

SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude...

- VHDL ✮ 41 ✭ 8

---

**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++ ✮ 146 ✭ 93

---

**rowhammer** Public


- C ✮ 189 ✭ 41

---

https://github.com/CMU-SAFARI/
**Pythia**
A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning.
- machine-learning
- reinforcement-learning
- prefetcher
- cache-replacement
- branch-predictor
- chamsim-simulator
- chamsim-tracer
- C++
  - MIT
  - 1
  - 90
  - 0
  - Updated yesterday

**BurstLink**

**MIG-7-PHY-DDR3-Controller**
A DDR3 Controller that uses the Xilinx MIG-7 PHY to interface with DDR3 devices.
- Verilog
  - MIT
  - 1
  - 1
  - 0
  - 0
  - Updated on Aug 22

**Pythia-HDL**
Implementation of Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning in Chisel HDL.
- machine-learning
- scala
- reinforcement-learning
- chisel
- chisel3
- ferrt
- hdl
- Scala
  - MIT
  - 0
  - 0
  - 0
  - 0
  - Updated on Jul 31

**HARP**
Private

**EINSim**
- simulator
- reliability
- statistical-inference
- dram
- error-correcting-codes
  - map-estimation
  - error-correction
  - C++
    - MIT
    - 0
    - 7
    - 0
    - 0
    - Updated on Jul 29

**DAMOV**
DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at https://arxiv.org/pdf/2105.03726.pdf)
- C++
  - MIT
  - 1
  - 10
  - 0
  - 0
  - Updated on Jul 13

**MetaSys**
MetaSys is the first open-source FPGA-based infrastructure with a prototype in a RISC-V core, to enable the rapid implementation and evaluation of a wide range of cross-layer software/hardware cooperative techniques techniques in real hardware. Described in our pre-print: https://arxiv.org/abs/2105.08123
- C++
  - MIT
  - 0
  - 0
  - 0
  - 0
  - Updated on Jul 9

**NATSA**
NATSA is the first near-data-processing accelerator for time series analysis based on the Matrix Profile (ScriMP) algorithm. NATSA exploits modern 3D-stacked High Bandwidth Memory (HBM) to enable efficient and fast matrix profile computation near memory. Described in ICCD 2020 by Fernandez et al. https://people.inf.ethz.ch/omutlu/pub/NATSA_time-
  - accelerator
  - hbm
  - time-series-analysis
  - matrix-profile
  - near-data-processing
  - scrimp
  - C++
    - MIT
    - 4
    - 0
    - 0
    - 0
    - Updated on Jun 28

**COVIDHunter**
COVIDHunter is an accurate and flexible COVID-19 outbreak simulation model that forecasts the strength of future mitigation measures and the numbers of cases, hospitalizations, and deaths for a given day, while considering the potential effect of environmental conditions. Described by Alser et al. (preliminary version at https://arxiv.org/abs/22...
  - simulation
  - epidemiology
  - covid-19
  - covid-19-data
  - covid-19-tracker
  - reproduction-number
  - covidhunter
  - Swift
    - MIT
    - 1
    - 5
    - 0
    - 0
    - Updated on Jun 27

**prim-benchmarks**
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 publicly-available real-world PIM architecture, the UPMEM PIM architecture. Described by Gómez-Luna et al. (preliminary version at https://arxiv.org/abs/22...
  - C
    - MIT
    - 8
    - 18
    - 0
    - 0
    - Updated on Jun 16

**SNP-Selective-Hiding**
An optimization-based mechanism to selectively hide the minimum number of overlapping SNPs among the family members who participated in the genomic studies (i.e., GWAS). Our goal is to distort the dependencies among the family members in the original database for achieving better privacy without significantly degrading the data utility.
  - gwas
  - genomics
  - data-privacy
  - differential-privacy
  - genomic-data-analysis
  - laplace-distribution
  - genomic-privacy
  - MATLAB
    - 0
    - 0
    - 0
    - 0
    - Updated on Jun 16

**SneakySnake**
SneakySnake is the first and the only pre-alignment filtering algorithm that works efficiently and fast on modern CPU, FPGA, and GPU architectures. It greatly (by more than two orders of magnitude) expedites sequence alignment calculation for both short and long reads. Described in the Bioinformatics (2020) by Alser et al. https://arxiv.org/abs/...
  - fp
  - gnu
  - smith-waterman
  - needleman-wunsch
  - sequence-alignment
  - long-reads
  - minimap2
  - VHDL
    - GPL-3.0
    - 6
    - 35
    - 0
    - 0
    - Updated on May 12

**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
    - 130
    - 250
    - 49
    - 4
    - Updated on May 11

**GenASM**
  - approximate-string-matching
  - read-mapping
  - hw-sw-co-design
  - read-alignment
  - bitap-algorithm
  - pre-alignment-filtering
  - genome-sequence-analysis
  - C
    - GPL-3.0
    - 3
    - 20
    - 0
    - 0
    - Updated on Mar 22

**AirLift**
AirLift is a tool that updates mapped reads from one reference genome to another. Unlike traditional mapping tools, AirLift maps to the nearest reference. The approach is based on locality sensitive hashing (LSH) and distributed (massively parallel) dynamic programming (D-PDP).

- C++
  - GPL-3.0
  - 1
  - 20
  - 0
  - 0
  - Updated on Jul 9

- C
  - GPL-3.0
  - 6
  - 35
  - 0
  - 0
  - Updated on May 12

- VHDL
  - GPL-3.0
  - 6
  - 35
  - 0
  - 0
  - Updated on May 12
Papers, Talks, Videos, Artifacts

- All are openly available at

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

  http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/
Principle: Environment of Freedom

Create an environment that values free exploration, openness, collaboration, hard work, creativity
My Suggestions to You
Follow Your Passion
(Do not get derailed by naysayers)
Principle: Build Infrastructure

Build Infrastructure to Enable Your Passion
Principle: Work Hard

Work Hard to Enable Your Passion
Suggestion to Researchers: Principle: Resilience

Be Resilient
Focus on learning and scholarship
Principle: Learning and Scholarship

The quality of your work defines your impact
Principle: Good Mindset, Goals & Focus

You can make a good impact on the world
Research & Teaching: Some Overview Talks

https://www.youtube.com/onurmutlulectures

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kgiZISOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=1

- Enabling In-Memory Computation
  - https://www.youtube.com/watch?v=njX_14584Jw&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=16

- Accelerating Genome Analysis
  - https://www.youtube.com/watch?v=r7sn41lH-4A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=41

- Rethinking Memory System Design
  - https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3

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

- The Story of RowHammer
  - https://www.youtube.com/watch?v=sgd7PHQQ1AI&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=39
An Interview on Research and Education

- **Computing Research and Education (@ ISCA 2019)**
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz](https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz)

- **Maurice Wilkes Award Speech (10 minutes)**
  - [https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=15)
More Thoughts and Suggestions

- Onur Mutlu,
  "Some Reflections (on DRAM)"
  Award Speech for ACM SIGARCH Maurice Wilkes Award, at the ISCA Awards Ceremony, Phoenix, AZ, USA, 25 June 2019.
  [Slides (pptx) (pdf)]
  [Video of Award Acceptance Speech (Youtube; 10 minutes) (Youku; 13 minutes)]
  [Video of Interview after Award Acceptance (Youtube; 1 hour 6 minutes) (Youku; 1 hour 6 minutes)]
  [News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"]

- Onur Mutlu,
  "How to Build an Impactful Research Group"
  57th Design Automation Conference Early Career Workshop (DAC), Virtual, 19 July 2020.
  [Slides (pptx) (pdf)]
More Thoughts and Suggestions (II)

- Onur Mutlu, "Computer Architecture: Why Is It So Important and Exciting Today?"
  Invited Lecture at Izmir Institute of Technology (IYTE), Virtual, 16 October 2020.
  [Slides (pptx) (pdf)]
  [Talk Video (2 hours 12 minutes)]

- 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

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
Richard Hamming

``You and Your Research''

Transcription of the
Bell Communications Research Colloquium Seminar
7 March 1986

How to Approach This Course

“Formative Experience”
How to Approach This Course

“High investment, high return”
How to Approach This Course

“Recorded lectures allowed me to go over the lectures when necessary”
How to Approach This Course

“YouTube allows me to watch the lectures on my TV”
How to Approach This Course

“The lecturer is very responsive to questions and remarks from students”
“Perhaps even better than in-person classes as questions can be asked asynchronously”
“Easy to understand course format with homework, labs, and lectures”
How to Approach This Course

“Paper reviews + assignments + labs, a really great plan to learn in a comprehensive way”
How to Approach This Course

“the course was fantastic and I would do it again at any time”
How to Approach This Course

Learning experience

Long-term tradeoff analysis

Critical thinking & decision making
How to Approach This Course

Concepts & Ideas

Fundamentals

Cutting-edge

Hands-on learning
How to Approach This Course

Your mindset will determine what you get out of the course
Richard Hamming

``You and Your Research''

Transcription of the
Bell Communications Research Colloquium Seminar
7 March 1986

If you really want to be a first-class scientist you need to know yourself, your weaknesses, your strengths, and your bad faults, like my egotism. How can you convert a fault to an asset? How can you convert a situation where you haven't got enough manpower to move into a direction when that's exactly what you need to do? I say again that I have seen, as I studied the history, the successful scientist changed the viewpoint and what was a defect became an asset.

In summary, I claim that some of the reasons why so many people who have greatness within their grasp don't succeed are: they don't work on important problems, they don't become emotionally involved, they don't try and change what is difficult to some other situation which is easily done but is still important, and they keep giving themselves alibis why they don't. They keep saying that it is a matter of luck. I've told you how easy it is; furthermore I've told you how to reform. Therefore, go forth and become great scientists!

Why Study Computer Architecture?
Computer Architecture

- is the **science** and **art** of designing **computing platforms** (hardware, interface, system SW, and programming model)

- to achieve a set of **design goals**
  - E.g., highest performance on earth on workloads X, Y, Z
  - E.g., longest battery life at a form factor that fits in your pocket with cost < $$$ CHF
  - E.g., best average performance across all known workloads at the best performance/cost ratio
  - ...

- Designing a supercomputer is different from designing a smartphone → But, many fundamental principles are similar
Different Platforms, Different Goals

Source: http://www.sia-online.org (semiconductor industry association)
Different Platforms, Different Goals

Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/
Different Platforms, Different Goals
Different Platforms, Different Goals

Source: http://sm.pcmag.com/pcmag_uk/photo/g/google-self-driving-car-the-guts/google-self-driving-car-the-guts_dwx8.jpg
Different Platforms, Different Goals
Different Platforms, Different Goals

Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg
Different Platforms, Different Goals

Source: https://www.itmagazine.ch/artikel/72401/Fugaku_Der_schnellste_Supercomputer_der_Welt.html
Different Platforms, Different Goals

Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Different Platforms, Different Goals

New ML applications (vs. TPU3):
- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

250 TFLOPS per chip in 2021 vs 90 TFLOPS in TPU3
1 ExaFLOPS per board

https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests
Different Platforms, Different Goals

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

[YouTube Video](https://www.youtube.com/watch?v=j0z4FweCy4M)
Different Platforms, Different Goals

- **Tesla Dojo Chip & System**

Different Platforms, Different Goals

- Tesla Dojo Chip & System

V1 Dojo Interface Processor

32GB High-Bandwidth Memory
- 800 GB/s Total Memory Bandwidth

900 GB/s TTP Interface
- Tesla Transport Protocol (TTP) - Full custom protocol
- Provides full DRAM bandwidth to Training Tile

50 GB/s TTP over Ethernet (TTPoE)
- Enables extending communication over standard Ethernet
- Native hardware support

32 GB/s Gen4 PCIe Interface

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s
Different Platforms, Different Goals

- **Tesla Dojo Chip & System**

  **D1 Chip**

  - 362 TFLOPs BF16/CFP8
  - 22.6 TFLOPs FP32
  - 10TBps/dir. On-Chip Bandwidth
  - 4TBps/edge. Off-Chip Bandwidth
  - 400W TDP

  ![Image of D1 Chip](https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s)

  - 645mm²
  - 7nm Technology
  - 50 Billion Transistors
  - 11+ Miles Of Wires

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s
Different Platforms, Different Goals

- Tesla Dojo Chip & System

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s
Different Platforms, Different Goals

- Tesla Dojo Chip & System

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s
Different Platforms, Different Goals

- The largest ML accelerator chip (2021)
- 850,000 cores

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

Largest GPU
54.2 Billion transistors
826 mm²

NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/
Different Platforms, Different Goals

Mohammed Alser, Züal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
Different Platforms, Different Goals

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 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 changing 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 architecture, motivated by the emergence of new 3-D 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 PIM (processing-in-memory) benchmarks, a benchmark suite of 10 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 PIM 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 140 and 2,000 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and guidelines and hints for hardware and architecture designers of future PIM systems.

What is Computer Architecture?

- The science and art of designing, selecting, and interconnecting hardware components and designing the hardware/software interface to create a computing system that meets functional, performance, energy consumption, cost, and other specific goals.
The Transformation Hierarchy

Computer Architecture (expanded view)

Computer Architecture (narrow view)

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Logic
Devices
Electrons
Why Study Computer Architecture?

- **Enable better systems**: make computers faster, cheaper, smaller, more reliable, ...
  - By exploiting advances and changes in underlying technology/circuits

- **Enable new applications**
  - Life-like 3D visualization 20 years ago? Virtual reality?
  - Self-driving cars?
  - Personalized genomics? Personalized medicine?

- **Enable better solutions** to problems
  - Software innovation is built on trends and changes in computer architecture
    - > 50% performance improvement per year has enabled this innovation

- **Understand why computers work the way they do**
Today is a very exciting time to study computer architecture. Industry is in a large paradigm shift (to novel architectures) – many different potential system designs possible.

Many difficult problems motivating and caused by the shift:
- Huge hunger for data and new data-intensive applications
- Power/energy/thermal constraints
- Complexity of design
- Difficulties in technology scaling
- Memory bottleneck
- Reliability problems
- Programmability problems
- Security and privacy issues

No clear, definitive answers to these problems.
Computer Architecture Today (II)

- These problems affect all parts of the computing stack – if we do not change the way we design systems

- Many new demands from the top (Look Up)

- Fast changing demands and personalities of users (Look Up)

- Many new issues at the bottom (Look Down)

- No clear, definitive answers to these problems
Computing landscape is very different from 10-20 years ago.

Both UP (software and humanity trends) and DOWN (technologies and their issues), FORWARD and BACKWARD, and the resulting requirements and constraints.

Every component and its interfaces, as well as entire system designs are being re-examined.
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
"There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics" was a lecture given by physicist Richard Feynman at the annual American Physical Society meeting at Caltech on December 29, 1959.[1] Feynman considered the possibility of direct manipulation of individual atoms as a more powerful form of synthetic chemistry than those used at the time. Although versions of the talk were reprinted in a few popular magazines, it went largely unnoticed and did not inspire the conceptual beginnings of the field. Beginning in the 1980s, nanotechnology advocates cited it to establish the scientific credibility of their work.
Historical: Opportunities at the Bottom (II)

There's Plenty of Room at the Bottom

From Wikipedia, the free encyclopedia

Feynman considered some ramifications of a general ability to manipulate matter on an atomic scale. He was particularly interested in the possibilities of denser computer circuitry, and microscopes that could see things much smaller than is possible with scanning electron microscopes. These ideas were later realized by the use of the scanning tunneling microscope, the atomic force microscope and other examples of scanning probe microscopy and storage systems such as Millipede, created by researchers at IBM.

Feynman also suggested that it should be possible, in principle, to make nanoscale machines that "arrange the atoms the way we want", and do chemical synthesis by mechanical manipulation.

He also presented the possibility of "swallowing the doctor", an idea that he credited in the essay to his friend and graduate student Albert Hibbs. This concept involved building a tiny, swallowable surgical robot.

https://en.wikipedia.org/wiki/There%27s_Plenty_of_Room_at_the_Bottom
There’s plenty of room at the Top: What will drive computer performance after Moore’s law?

Much of the improvement in computer performance comes from decades of miniaturization of computer components, a trend that was foreseen by the Nobel Prize–winning physicist Richard Feynman in his 1959 address, “There’s Plenty of Room at the Bottom,” to the American Physical Society. In 1975, Intel founder Gordon Moore predicted the regularity of this miniaturization trend, now called Moore’s law, which, until recently, doubled the number of transistors on computer chips every 2 years.

Unfortunately, semiconductor miniaturization is running out of steam as a viable way to grow computer performance—there isn’t much more room at the “Bottom.” If growth in computing power stalls, practically all industries will face challenges to their productivity. Nevertheless, opportunities for growth in computing performance will still be available, especially at the “Top” of the computing-technology stack: software, algorithms, and hardware architecture.
Axiom, Revisited

There is plenty of room both at the top and at the bottom

but much more so

when you

communicate well between and optimize across

the top and the bottom
Hence the Expanded View

Computer Architecture (expanded view)

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
Some Cross-Layer Design Examples (Foreshadowing)
EDEN: Data-Aware Efficient DNN Inference

- Skanda Koppula, Lois Orosa, A. Giray Yaglikci, Roknoddin Azizi, Taha Shahroodi, Konstantinos Kanellopoulos, and Onur Mutlu,

"EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM"

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (38 minutes)]

EDEN: Enabling Energy-Efficient, High-Performance Deep Neural Network Inference Using Approximate DRAM

Skanda Koppula  Lois Orosa  A. Giray Yağlıkçı
Roknoddin Azizi  Taha Shahroodi  Konstantinos Kanellopoulos  Onur Mutlu

ETH Zürich
SMASH: SW/HW Indexing Acceleration

Konstantinos Kanellopoulos, Nandita Vijaykumar, Christina Giannoula, Roknoddin Azizi, Skanda Koppula, Nika Mansouri Ghiasi, Taha Shahroodi, Juan Gomez-Luna, and Onur Mutlu,

"SMASH: Co-designing Software Compression and Hardware-Accelerated Indexing for Efficient Sparse Matrix Operations"

Proceedings of the 52nd International Symposium on Microarchitecture (MICRO), Columbus, OH, USA, October 2019.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Full Talk Lecture (30 minutes)]
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)]
SW/HW Climate Modeling Accelerator

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

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

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

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (23 minutes)]

Nominated for the Stamatis Vassiliadis Memorial Award.

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

Gagandeep Singh$^{a,b,c}$, Dionysios Diamantopoulos$^{c}$, Christoph Hagleitner$^{c}$, Sander Stuijk$^{a}$, Onur Mutlu$^{b}$, Henk Corporaal$^{a}$, Juan Gómez-Luna$^{b}$

$^{a}$Eindhoven University of Technology
$^{b}$ETH Zürich
$^{c}$IBM Research Europe, Zurich
HW/SW Time Series Analysis Accelerator

- Ivan Fernandez, Ricardo Quislant, Christina Giannoula, Mohammed Alser, Juan Gómez-Luna, Eladio Gutiérrez, Oscar Plata, and Onur Mutlu,

"NATSA: A Near-Data Processing Accelerator for Time Series Analysis"
[Slides (pptx) (pdf)]
[Talk Video (10 minutes)]
[Source Code]

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

Ivan Fernandez§ Ricardo Quislant§ Christina Giannoula† Mohammed Alser‡
Juan Gómez-Luna‡ Eladio Gutiérrez§ Oscar Plata§ Onur Mutlu‡

§University of Malaga †National Technical University of Athens ‡ETH Zürich
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
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)]
Graph Processing Accelerator w/ PIM

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

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


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

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
Processing in Memory for Mobile Workloads

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"

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

- Kevin Hsieh, Samira Khan, Nandita Vijaykumar, Kevin K. Chang, Amirali Boroumand, Saugata Ghose, and Onur Mutlu,

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

Proceedings of the 34th IEEE International Conference on Computer Design (ICCD), Phoenix, AZ, USA, October 2016.

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

Kevin Hsieh† Samira Khan‡ Nandita Vijaykumar†
Kevin K. Chang† Amirali Boroumand† Saugata Ghose† Onur Mutlu§†
†Carnegie Mellon University ‡University of Virginia §ETH Zürich
Expressive (Memory) Interfaces

Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons and Onur Mutlu,
"A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory"
[Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video]

A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar†§ Abhilasha Jain† Diptesh Majumdar† Kevin Hsieh† Gennady Pekhimenko‡
Eiman Ebrahimiκ Nastaran Hajinazar‡ Phillip B. Gibbons† Onur Mutlu§†

†Carnegie Mellon University ‡University of Toronto
‡Simon Fraser University §ETH Zürich
κNVIDIA
One Problem: Limited SW/HW Communication

Higher-level information is not visible to HW

Software

Hardware

100011111...
101010011...

Instructions
Memory Addresses

Data Structures
Code Optimizations
Access Patterns

Integer
Float
Data Type
Char
A Solution: More Expressive Interfaces

[Diagram showing Performance, Software, Virtual Memory, ISA, Higher-level Program Semantics, Expressive Memory “XMem,” Functionality, and Hardware]
# Table 1: Summary of the example memory optimizations that XMem aids.

<table>
<thead>
<tr>
<th>Memory optimization</th>
<th>Example semantics provided by XMem (described in §3.3)</th>
<th>Example Benefits of XMem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache management</td>
<td>(i) Distinguishing between data structures or pools of similar data; (ii) Working set size; (iii) Data reuse</td>
<td>Enables: (i) applying different caching policies to different data structures or pools of data; (ii) avoiding cache thrashing by knowing the active working set size; (iii) bypassing/prioritizing data that has no/high reuse. (§5)</td>
</tr>
<tr>
<td>Page placement in DRAM e.g., [23, 24]</td>
<td>(i) Distinguishing between data structures; (ii) Access pattern; (iii) Access intensity</td>
<td>Enables page placement at the data structure granularity to (i) isolate data structures that have high row buffer locality and (ii) spread out concurrently-accessed irregular data structures across banks and channels to improve parallelism. (§6)</td>
</tr>
<tr>
<td>Cache/mem compression e.g., [25–32]</td>
<td>(i) Data type: integer, float, char; (ii) Data properties: sparse, pointer, data index</td>
<td>Enables using a different compression algorithm for each data structure based on data type and data properties, e.g., sparse data encodings, FP-specific compression, delta-based compression for pointers [27].</td>
</tr>
<tr>
<td>Data prefetching e.g., [33–36]</td>
<td>(i) Access pattern: strided, irregular, irregular but repeated (e.g., graphs), access stride; (ii) Data type: index, pointer</td>
<td>Enables (i) highly accurate software-driven prefetching while leveraging the benefits of hardware prefetching (e.g., by being memory bandwidth-aware, avoiding cache thrashing); (ii) using different prefetcher types for different data structures: e.g., stride [33], tile-based [20], pattern-based [34–37], data-based for indices/pointers [38, 39], etc.</td>
</tr>
<tr>
<td>DRAM cache management e.g., [40–46]</td>
<td>(i) Access intensity; (ii) Data reuse; (iii) Working set size</td>
<td>(i) Helps avoid cache thrashing by knowing working set size [44]; (ii) Better DRAM cache management via reuse behavior and access intensity information.</td>
</tr>
<tr>
<td>Approximation in memory e.g., [47–53]</td>
<td>(i) Distinguishing between pools of similar data; (ii) Data properties: tolerance towards approximation</td>
<td>Enables (i) each memory component to track how approximable data is (at a fine granularity) to inform approximation techniques; (ii) data placement in heterogeneous reliability memories [54].</td>
</tr>
<tr>
<td>Data placement NUMA systems e.g., [55, 56]</td>
<td>(i) Data partitioning across threads (i.e., relating data to threads that access it); (ii) Read-Write properties</td>
<td>Reduces the need for profiling or data migration (i) to co-locate data with threads that access it and (ii) to identify Read-Only data, thereby enabling techniques such as replication.</td>
</tr>
<tr>
<td>Data placement hybrid memories e.g., [16, 57, 58]</td>
<td>(i) Read-Write properties (Read-Only/Read-Write); (ii) Access intensity; (iii) Data structure size; (iv) Access pattern</td>
<td>Avoids the need for profiling/migration of data in hybrid memories to (i) effectively manage the asymmetric read-write properties in NVM (e.g., placing Read-Only data in the NVM) [16, 57]; (ii) make tradeoffs between data structure &quot;hotness&quot; and size to allocate fast/high bandwidth memory [14]; and (iii) leverage row-buffer locality in placement based on access pattern [45].</td>
</tr>
<tr>
<td>Managing NUCA systems e.g., [15, 59]</td>
<td>(i) Distinguishing pools of similar data; (ii) Access intensity; (iii) Read-Write or Private-Shared properties</td>
<td>(i) Enables using different cache policies for different data pools (similar to [15]); (ii) Reduces the need for reactive mechanisms that detect sharing and read-write characteristics to inform cache policies.</td>
</tr>
</tbody>
</table>
Expressive (Memory) Interfaces for GPUs

- Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons and Onur Mutlu, "The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs"
  [Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video]

The Locality Descriptor:
A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar†§ Eiman Ebrahimi‡§ Kevin Hsieh†
Phillip B. Gibbons† Onur Mutlu§†
†Carnegie Mellon University ‡NVIDIA §ETH Zürich
Heterogeneous-Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu, "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory". Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [Summary] [Slides (pptx) (pdf)] [Coverage on ZDNet]
Exploiting Memory Error Tolerance with Hybrid Memory Systems

Vulnerable data

Tolerant data

Reliable memory

Low-cost memory

On Microsoft’s Web Search workload
Reduces server hardware cost by 4.7 %
Achieves single server availability target of 99.90 %

Heterogeneous-Reliability Memory [DSN 2014]
Heterogeneous-Reliability Memory

Step 1: Characterize and classify application memory error tolerance

Step 2: Map application data to the HRM system enabled by SW/HW cooperative solutions

Reliable memory + software recovery (Par+R)

Unreliable memory

Reliable

Parity memory

Low-cost memory
Rethinking Virtual Memory

- Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,

"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"


[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[ARM Research Summit Poster (pptx) (pdf)]
[Talk Video (26 minutes)]
[Lightning Talk Video (3 minutes)]

The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

Nastaran Hajinazar*† Pratyush Patel* Minesh Patel* Konstantinos Kanellopoulos* Saugata Ghose† Rachata Ausavarungnirun© Geraldo F. Oliveira* Jonathan Appavoo◊ Vivek Seshadri▼ Onur Mutlu*†

*ETH Zürich †Simon Fraser University *University of Washington †Carnegie Mellon University ©King Mongkut’s University of Technology North Bangkok ◊Boston University ▼Microsoft Research India
Many Interesting Things Are Happening Today in Computer Architecture
Many Interesting Things Are Happening Today in Computer Architecture

Performance and Energy Efficiency
Intel Optane Persistent Memory (2019)

- Non-volatile main memory
- Based on 3D-XPoint Technology
PCM as Main Memory: Idea in 2009


One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight. 2022 Persistent Impact Prize.

Architecting Phase Change Memory as a Scalable DRAM Alternative

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

†Computer Architecture Group Microsoft Research Redmond, WA {blee, ipek, dburger}@microsoft.com
‡Computer Architecture Laboratory Carnegie Mellon University Pittsburgh, PA onur@cmu.edu
PCM as Main Memory: Idea in 2009


Phase-Change Technology and the Future of Main Memory
Cerebras’s Wafer Scale Engine (2019)

- The largest ML accelerator chip
- 400,000 cores

Cerebras WSE
1.2 Trillion transistors
46,225 mm²

Largest GPU
21.1 Billion transistors
815 mm²

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning
Cerebras’s Wafer Scale Engine-2 (2021)

- The largest ML accelerator chip (2021)
- 850,000 cores

Cerebras WSE-2
- 2.6 Trillion transistors
- 46,225 mm²

Largest GPU
- 54.2 Billion transistors
- 826 mm²

NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning

https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/
UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.

- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth

---

UPMEM Memory Modules

• E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
• P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz
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 GIANNIOLA, 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 architecture, 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 UPHTEM company has designed and manufactured the first publicly available real-world PIM architecture. The UPHTEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same die.

This paper provides the first comprehensive analysis of the first publicly available real-world PIM architecture. We tackle two key contributions. First, we conduct an empirical characterization of the UPHTEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PIM (DPU) microbenchmarks, a benchmark suite of 16 workloads from different application domains (e.g., dense/matrix operations, database, 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 UPHTEM 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 UPHTEM-based PIM systems with 140 and 2,500 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.

Dr. Juan Gomez-Luna

- Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware
- Based on two major works
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)]
# PrIM Benchmarks: Application Domains

<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.
Understanding a Modern PIM Architecture

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

JUAN GÓMEZ-LUNA¹, IZZAT EL HAJJ², IVAN FERNANDEZ¹,³, CHRISTINA GIANNELLA¹,⁴, GERALDO F. OLIVEIRA¹, AND ONUR MUTLU¹

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

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

https://github.com/CMU-SAFARI/prim-benchmarks
Understanding a Modern PIM Architecture

Juan Gómez Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, Onur Mutlu

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

SAFARI Live Seminar: Understanding a Modern Processing-in-Memory Architecture
2,579 views • Streamed live on Jul 12, 2021

https://www.youtube.com/watch?v=D8Hjy2iU9l4&list=PL5Q2soXY2Zj_tOTAYm--dYByNPL7JhwR9
More on Analysis of the UPMEM PIM Engine

Inter-DPU Communication

- There is no direct communication channel between DPUs

- Inter-DPU communication takes places 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=D8Hjy2iU9I4&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

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

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

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

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

Samsung Function-in-Memory DRAM (2021)

- FIMDRAM based on HBM2

[3D Chip Structure of HBM with FIMDRAM]

**Chip Specification**
- 128DQ / 8CH / 16 banks / BL4
- 32 PCU blocks (1 FIM block/2 banks)
- 1.2 TFLOPS (4H)
- FP16 ADD /
  Multiply (MUL) /
  Multiply-Accumulate (MAC) /
  Multiply-and- Add (MAD)

**ISSCC 2021 / SESSION 25 / DRAM / 25.4**

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

Young-Cheon Kwon¹, Suk Han Lee¹, Jaehoon Lee¹, Sang-Hysik Kwon¹, Je Min Ryu¹, Jong-Pil Son², Seongil O³, Hak-Soo Yu⁴, Haesuk Lee⁴, Soo Young Kim⁴, Youngmin Cho⁴, Jin Suk Kim⁴, Jongyoon Choi⁴, Hyun-Sung Shin⁴, Jin Kim⁴, BengSeng Phuah⁴, HyungMin Kim⁴, Myeong Jun Song⁴, Ahn Choi⁴, Daeho Kim⁴, SooYoung Kim⁴, Eun-Bong Kim⁴, David Wang⁴, Shinhaeng Kang⁴, Yuhwan Ro⁵, Seungwoo Seo⁵, JoonHo Song⁶, Juyeon Youn⁶, Kyomin Sohn⁶, Nam Sung Kim⁶

¹Samsung Electronics, Hwaseong, Korea
²Samsung Electronics, San Jose, CA
³Samsung Electronics, Suwon, Korea
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]
### Available instruction list for FIM operation

<table>
<thead>
<tr>
<th>Type</th>
<th>CMD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating Point</td>
<td>ADD</td>
<td>FP16 addition</td>
</tr>
<tr>
<td></td>
<td>MUL</td>
<td>FP16 multiplication</td>
</tr>
<tr>
<td></td>
<td>MAC</td>
<td>FP16 multiply-accumulate</td>
</tr>
<tr>
<td></td>
<td>MAD</td>
<td>FP16 multiply and add</td>
</tr>
<tr>
<td>Data Path</td>
<td>MOVE</td>
<td>Load or store data</td>
</tr>
<tr>
<td></td>
<td>FILL</td>
<td>Copy data from bank to GRFs</td>
</tr>
<tr>
<td>Control Path</td>
<td>NOP</td>
<td>Do nothing</td>
</tr>
<tr>
<td></td>
<td>JUMP</td>
<td>Jump instruction</td>
</tr>
<tr>
<td></td>
<td>EXIT</td>
<td>Exit instruction</td>
</tr>
</tbody>
</table>
Chip Implementation

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

[Digital RTL design for PCU block]
Samsung AxDIMM (2021)

- DIMM-based PIM
  - DLRM recommendation system

SK Hynix AiM: Chip Implementation (2022)

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

### AiM Die Photograph

![AiM Die Photograph](image)

### 1 Process Unit (PU) Area

<table>
<thead>
<tr>
<th></th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>0.19mm²</td>
</tr>
<tr>
<td>MAC</td>
<td>0.11mm²</td>
</tr>
<tr>
<td>Activation Function (AF)</td>
<td>0.02mm²</td>
</tr>
<tr>
<td>Reservoir Cap.</td>
<td>0.05mm²</td>
</tr>
<tr>
<td>Etc.</td>
<td>0.01mm²</td>
</tr>
</tbody>
</table>

- MAC 58%
- AF 11%
- Reservoir Cap. 26%
- Etc. 5%
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

256 b

2KB

Supplementary SRAM buffer

Multiplier x 16

Adder Tree

Accumulator & AF

RDMAC RDAF
Alibaba HB-PNM: Overall Architecture (2022)

- **3D-stacked logic die and DRAM die vertically bonded by hybrid bonding (HB)**

Figure 29.1.1: Motivations and comparison of state-of-the-art PNM/CIM architectures.

Figure 29.1.2: Illustration of 3D-stacked chip, cross-illustration of package, DRAM array layout and design blocks on logic die.

Figure 29.1.3: Overall architecture of PNM logic. Detailed flow of typical recommendation system.

Figure 29.1.4: Detailed design of Match Engine (ME), showing internal data-path micro-architecture of AddGen, distance calculator, and top-K engine.

Figure 29.1.5: Detailed design of Neural Engine (NE), showing internal datapath, interface modules, micro architecture of VPU and GEMM, FSM of control modules and lock-step debug module.

Figure 29.1.6: Illustration of FPGA-based evaluation platform, comparison with prior near-memory processing designs, and end-to-end performance evaluation of our HB chip and CPU-DRAM system on recommendation application.
Processing in Memory:
Two Approaches

1. Processing near Memory
2. Processing using Memory
Specialized Processing in Memory (2015)

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
  "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
  [Slides (pdf)] [Lightning Session Slides (pdf)]

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
Simple Processing in Memory (2015)

- Junwhan Ahn, Sungjoo Yoo, Onur Mutlu, and Kiyoungh Choi, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture"


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

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

Junwhan Ahn  Sungjoo Yoo  Onur Mutlu†  Kiyoungh Choi
junwhan@snu.ac.kr, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr

Seoul National University  †Carnegie Mellon University
Processing in Memory on 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"

Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

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

Christina Giannoula, Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas, Ivan Fernandez, Juan Gómez-Luna, Lois Orosa, Nectarios Koziris, Georgios Goumas, and Onur Mutlu,
"SynCron: Efficient Synchronization Support for Near-Data-Processing Architectures"

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

Christina Giannoula‡‡ Nandita Vijaykumar*‡ Nikela Papadopoulou† Vasileios Karakostas† Ivan Fernandez§‡
Juan Gómez-Luna‡ Lois Orosa‡ Nectarios Koziris† Georgios Goumas† Onur Mutlu‡
†National Technical University of Athens ‡ETH Zürich *University of Toronto §University of Malaga
Accelerating GPU Execution with PIM (I)

- Kevin Hsieh, Eiman Ebrahimi, Gwangsun Kim, Niladrish Chatterjee, Mike O'Connor, Nandita Vijaykumar, Onur Mutlu, and Stephen W. Keckler,

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


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

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

Kevin Hsieh† Eiman Ebrahimi† Gwangsun Kim* Niladrish Chatterjee† Mike O'Connor†
Nandita Vijaykumar† Onur Mutlu§† Stephen W. Keckler†
†Carnegie Mellon University †NVIDIA *KAIST §ETH Zürich
Accelerating Linked Data Structures

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

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

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

More on DAMOV Analysis Methodology & Workloads

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

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

https://www.youtube.com/watch?v=GWideVyo0nM&list=PL5Q2soXYZi_tOTAYm--dYByNPL7JhwR9&index=3
DAMOV is Open-Source

• We open-source our benchmark suite and our toolchain

DAMOV-SIM

DAMOV Benchmarks

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, HPG, Ligra, PARSEC, Parboil, PolyBench, Phoenix, Rodinia, SPLASH-2, STREAM.
[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
Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory
In-DRAM Processing (2013)


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

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

\textsuperscript{1}Microsoft Research India \textsuperscript{2}NVIDIA Research \textsuperscript{3}Intel \textsuperscript{4}ETH Zürich \textsuperscript{5}Carnegie Mellon University
In-DRAM Bulk Bitwise Execution (2017)

- Vivek Seshadri and Onur Mutlu, "In-DRAM Bulk Bitwise Execution Engine" 
  [Preliminary arXiv version]

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri  
Microsoft Research India  
visesha@microsoft.com

Onur Mutlu  
ETH Zürich  
onur.mutlu@inf.ethz.ch
**SIMDRAM Framework (2021)**

- Nastaran Hajinazar, Geraldo F. Oliveira, Sven Gregorio, Joao Dinis Ferreira, Nika Mansouri Ghiasi, Minesh Patel, Mohammed Alser, Saugata Ghose, Juan Gomez-Luna, and Onur Mutlu,

"**SIMDRAM: An End-to-End Framework for Bit-Serial SIMD Computing in DRAM**"


[2-page Extended Abstract](#)
[Short Talk Slides (pptx) (pdf)](#)
[Talk Slides (pptx) (pdf)](#)
[Short Talk Video (5 mins)](#)
[Full Talk Video (27 mins)](#)

---

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

*Nastaran Hajinazar*\(^1,2\)  
Nika Mansouri Ghiasi\(^1\)  
Minesh Patel\(^1\)  
Juan Gómez-Luna\(^1\)

*Geraldo F. Oliveira*\(^1\)  
Sven Gregorio\(^1\)  
Mohammed Alser\(^1\)  
Onur Mutlu\(^1\)

João Dinis Ferreira\(^1\)  
Saugata Ghose\(^3\)

\(^1\)ETH Zürich  
\(^2\)Simon Fraser University  
\(^3\)University of Illinois at Urbana–Champaign
Bulk Data Copy and Initialization in DRAM

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry,

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

Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013. [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]

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

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

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

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

Carnegie Mellon University †Intel Pittsburgh
LISA: Increasing Connectivity in DRAM

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

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


[Slides (pptx) (pdf)]
[Source Code]

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

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

†Carnegie Mellon University †Georgia Institute of Technology
FIGARO: Fine-Grained In-DRAM Copy

- Yaohua Wang, Lois Orosa, Xiangjun Peng, Yang Guo, Saugata Ghose, Minesh Patel, Jeremie S. Kim, Juan Gómez Luna, Mohammad Sadrosadati, Nika Mansouri Ghiasi, and Onur Mutlu,

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


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

Yaohua Wang*, Lois Orosa†, Xiangjun Peng○*, Yang Guo*, Saugata Ghose○†, Minesh Patel†, Jeremie S. Kim†, Juan Gómez Luna†, Mohammad Sadrosadati§, Nika Mansouri Ghiasi†, Onur Mutlu††

*National University of Defense Technology  †ETH Zürich  ○Chinese University of Hong Kong
○University of Illinois at Urbana–Champaign  ‡Carnegie Mellon University  §Institute of Research in Fundamental Sciences
Network-On-Memory: Fast Inter-Bank Copy

- Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Modarressi, Rachata Ausavarungnirun, Mohammad Sadrosadati, Onur Mutlu, and Masoud Daneshtalab,

"NoM: Network-on-Memory for Inter-Bank Data Transfer in Highly-Banked Memories"

In-DRAM Physical Unclonable Functions

Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
"The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
[Lightning Talk Video]
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
[Full Talk Lecture Video (28 minutes)]

The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim†§ Minesh Patel§ Hasan Hassan§ Onur Mutlu§†
†Carnegie Mellon University §ETH Zürich
In-DRAM True Random Number Generation

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


[Slides (pptx) (pdf)]
[Full Talk Video (21 minutes)]
[Full Talk Lecture Video (27 minutes)]

Top Picks Honorable Mention by IEEE Micro.

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

Jeremie S. Kim‡$, Minesh Patel§, Hasan Hassan§, Lois Orosa§, Onur Mutlu§‡

‡Carnegie Mellon University §ETH Zürich
In-DRAM True Random Number Generation

- Ataberk Olgun, Minesh Patel, A. Giray Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu,

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


[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Talk Video (25 minutes)]
[SAFARI Live Seminar Video (1 hr 26 mins)]

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

Ataberk Olgun$\dagger$ Minesh Patel$\dagger$ A. Giray Yağlıkçı$\dagger$ Haocong Luo$\dagger$
Jeremie S. Kim$\dagger$ F. Nisa Bostancı$\dagger$ Nandita Vijaykumar$\dagger$ Oğuz Ergin$\dagger$ Onur Mutlu$\dagger$

$\dagger$ETH Zürich $\dagger$TOBB University of Economics and Technology $\dagger$University of Toronto

SAFARI
In-DRAM True Random Number Generation

  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]

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

F. Nisa Bostancı†§
Ataberk Olgun†§
Lois Orosa§
A. Giray Yağlıkçı§
Jeremie S. Kim§
Hasan Hassan§
Oğuz Ergin†
Onur Mutlu§

†TOBB University of Economics and Technology
§ETH Zürich

In-DRAM Lookup-Table Based Execution

- To appear at MICRO 2022

pLUTo: Enabling Massively Parallel Computation in DRAM via Lookup Tables

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

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

In-Flash Bulk Bitwise Execution

- To appear at MICRO 2022

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

Jisung Park$\textsuperscript{§, V}  \quad $ Roknoddin Azizi$\textsuperscript{§}  \quad $ Geraldo F. Oliveira$\textsuperscript{§}  \quad $ Mohammad Sadrosadati$\textsuperscript{§}  \\
Rakesh Nadig$\textsuperscript{§}  \quad $ David Novo$\textdagger$  \quad $ Juan Gómez-Luna$\textsuperscript{§}  \quad $ Myungskuk Kim$\ddagger$  \quad $ Onur Mutlu$\textsuperscript{§}$

$\textsuperscript{§}$ETH Zürich  \quad $\textsuperscript{V}$POSTECH  \quad $\textsuperscript{†}$LIRMM, Univ. Montpellier, CNRS  \quad $\textsuperscript{‡}$Kyungpook National University
Processing in Memory: Two Approaches

1. Processing near Memory
2. Processing using Memory
PIM Review and Open Problems

A Modern Primer on Processing in Memory

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

SAFARI Research Group

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

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

Invited Book Chapter in Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann, Springer, to be published in 2023

SAFARI

A Modern Primer on Processing in Memory

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

SAFARI Research Group

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

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 energy 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 [1–26]. 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, 27–59], 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 [7–9, 60–64]. These costs are often exacerbated by the fact that much of the data brought into the caches is not reused by the CPU [62, 63, 65, 66], 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 applications, the processor-centric design paradigm leads to great inefficiency in performance, energy, and cost. For example, most of the real estate within a single compute
A Workload and Programming Ease Driven Perspective of Processing-in-Memory

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

†Carnegie Mellon University §ETH Zürich

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

A Tutorial on PIM

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
Detailed Lectures on PIM (I)

- Computer Architecture, Fall 2020, Lecture 6
  - Computation in Memory (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=oGcZAGwfEUE&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=12

- Computer Architecture, Fall 2020, Lecture 7
  - Near-Data Processing (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=j2GIigqn1Qw&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=13

- Computer Architecture, Fall 2020, Lecture 11a
  - Memory Controllers (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=TeG773OgiMQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=20

- Computer Architecture, Fall 2020, Lecture 12d
  - Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25

https://www.youtube.com/onurmutlulectures
Detailed Lectures on PIM (II)

- Computer Architecture, Fall 2020, Lecture 15
  - **Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=AIE1rD9G_YU&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=28](https://www.youtube.com/watch?v=AIE1rD9G_YU&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=28)

- Computer Architecture, Fall 2020, Lecture 16a
  - **Opportunities & Challenges of Emerging Memory Technologies** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=29](https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=29)

- Computer Architecture, Fall 2020, Guest Lecture
  - **In-Memory Computing: Memory Devices & Applications** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=wNmQHiEZNk&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=41](https://www.youtube.com/watch?v=wNmQHiEZNk&list=PL5Q2soXY2Zi9idyIgBxUz7xRPS-wisBN&index=41)
Many Interesting Things Are Happening Today in Computer Architecture

Performance and Energy Efficiency
TESLA Full Self-Driving Computer (2019)

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

https://youtu.be/Ucp0TTmvqOE?t=4236
Google TPU Generation I (~2016)

**Figure 3.** TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

**Figure 4.** Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Google TPU Generation II (2017)

4 TPU chips
vs 1 chip in TPU1

High Bandwidth Memory
vs DDR3

Floating point operations
vs FP16

45 TFLOPS per chip
vs 23 TOPS

Designed for training
and inference
vs only inference

Google TPU Generation III (2019)

32GB HBM per chip vs 16GB HBM in TPU2

4 Matrix Units per chip vs 2 Matrix Units in TPU2

90 TFLOPS per chip vs 45 TFLOPS in TPU2

https://cloud.google.com/tpu/docs/system-architecture
Google TPU Generation IV (2021)

New ML applications (vs. TPU3):
- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

250 TFLOPS per chip in 2021 vs 90 TFLOPS in TPU3
1 ExaFLOPS per board

https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests
An Example Modern Systolic Array: TPU (II)

As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the Unified Buffer [Kun80][Ram91][Ovt15b]. Figure 4 shows that data flows in from the left, and the weights are loaded from the top. A given 256-element multiply-accumulate operation moves through the matrix as a diagonal wavefront. The weights are preloaded, and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give the illusion that the 256 inputs are read at once, and that they instantly update one location of each of 256 accumulators. From a correctness perspective, software is unaware of the systolic nature of the matrix unit, but for performance, it does worry about the latency of the unit.
Figure 1. TPU Block Diagram. The main computation part is the yellow Matrix Multiply unit in the upper right hand corner. Its inputs are the blue Weight FIFO and the blue Unified Buffer (UB) and its output is the blue Accumulators (Acc). The yellow Activation Unit performs the nonlinear functions on the Acc, which go to the UB.
Many (Other) AI/ML Chips

- Alibaba
- Amazon
- Facebook
- Google
- Huawei
- Intel
- Microsoft
- NVIDIA
- Tesla
- Many Others and Many Startups...

- Many More to Come...
Many (Other) AI/ML Chips (2021)

AI Chip Landscape

Startup in China

MLU100/270X20
Journey
BM1680/7180
DeepEye1000
QuestCore
Nvidia
K210
AR9000
Vorittel61
T320/T210/S10
T4A8010
HaxHam
MembCorex1F1

IC Vendor/Fabless

ICP
NVIDIA
Qualcomm
AMD
Xilinx
Mediatek
UNISOC
Rockchip
Armartise
Harman
NXP
Texas Instruments
RENESAS
Toshiba

Automated Driving

AutoX
DriveONE
Audi
Toyota
Byton
AutoX
Yutian
Velodyne
X-by-Wire
Orin
Sea

Smart Voice

Caviar
Droex
Baidu
cT8
Baidu
Baidu
Baidu
Baidu

Compilers

TensorFlow
MLIR
GLOW
VTR
OctoML
NVIDIA TensorRT
apexML
nGraph
ONNC
Tiramisu Compiler
The Tensor Algebra Compiler (taco)

Startups Worldwide

FPGA/eFPGA
Achronix
Graphcore
GC2
SambaNova

Processing in Memory

MYTHIC
ARENA

Optical Computing

LUMINOUS
LIGHTMATTER

Neuromorphic

brainchip
ci

PEZY Computing
PEZY-S2
Ela Compute
ECG1531

Preferred Networks
MN-Core
INNOCRIT
Shaprio/Itami

Inferon
K1320

NeuroBlade
Novumind

Renium
eSilicon

GAP8
GREENWAVES

More at https://basicmi.github.io/AI-Chip/

IP/Design Service

arm
Synopsys
Imagination
Ceva
cadence
SiFive
ARTERIS

twente
design service with in-house IP

Western Digital
Nokia
LG

inspur
Fujitsu
Dell

Hewlett Packard Enterprise

Startup Results available

Tech Giants/System

Google
Microsoft
Facebook
AWS
IBM

Apple
Intel
Samsung
Alibaba Group

Nvidia
Qualcomm
AMD
Xilinx

Mediatek
Unisoc
Rockchip
Armartise
Harman
NXP
Texas Instruments
Renesas
Toshiba

AutoX
DriveONE
Audi
Toyota
Byton
AutoX
Yutian
Velodyne
X-by-Wire
Orin
Sea

Caviar
Droex
Baidu
cT8
Baidu
Baidu
Baidu

TensorFlow
MLIR
GLOW
VTR
OctoML
NVIDIA TensorRT
apexML
nGraph
ONNC

The Tensor Algebra Compiler (taco)

SAFARI
https://basicmi.github.io/AI-Chip/

AI Chip Landscape

V0.7 Dec., 2019

S.T.

IP/Design Service

arm
Synopsys
Imagination
Ceva
cadence
SiFive
ARTERIS

twente
design service with in-house IP

Western Digital
Nokia
LG

inspur
Fujitsu
Dell

Hewlett Packard Enterprise

Startup Results available

Tech Giants/System

Google
Microsoft
Facebook
AWS
IBM

Apple
Intel
Samsung
Alibaba Group

Nvidia
Qualcomm
AMD
Xilinx

Mediatek
Unisoc
Rockchip
Armartise
Harman
NXP
Texas Instruments
Renesas
Toshiba

AutoX
DriveONE
Audi
Toyota
Byton
AutoX
Yutian
Velodyne
X-by-Wire
Orin
Sea

Caviar
Droex
Baidu
cT8
Baidu
Baidu
Baidu

TensorFlow
MLIR
GLOW
VTR
OctoML
NVIDIA TensorRT
apexML
nGraph
ONNC

The Tensor Algebra Compiler (taco)

SAFARI
https://basicmi.github.io/AI-Chip/

197
Many Interesting Things Are Happening Today in Computer Architecture
Many Interesting Things
Are Happening Today
in Computer Architecture

Reliability
Security
Safety
The Story of RowHammer

- One can predictably induce bit flips in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable

- First example of how a simple hardware failure mechanism can create a widespread system security vulnerability
Modern DRAM is Prone to Disturbance Errors

Repeatedly reading a row enough times (before memory gets refreshed) induces disturbance errors in adjacent rows in most real DRAM chips you can buy today.

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

A company

86% (37/43)

Up to \(1.0 \times 10^7\) errors

B company

83% (45/54)

Up to \(2.7 \times 10^6\) errors

C company

88% (28/32)

Up to \(3.3 \times 10^5\) errors

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, (Kim et al., ISCA 2014)
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 advances, the traditional memory isolation assumptions are under threat due to the increasing occurrence of memory disturbances. In this paper, we present Flipping Bits in Memory Without Accessing Them (FBM): an experimental study of the DRAM rowhammer bug and its implications on the security of modern computing systems.

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

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn, 2015)

It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after.
“We can gain unrestricted access to systems of website visitors.”

Not there yet, but ...

ROOT privileges for web apps!

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA’16)

Source: https://lab.dsst.io/32c3-slides/7197.html
More Security Implications (II)

“Can gain control of a smart phone deterministically”

Hammer And Root

Millions of Androids

Source: https://fossbytes.com/drammer-rowhammer-attack-android-root-devices/

Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, CCS’16
More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface

"GRAND PWNING UNIT" — Drive-by Rowhammer attack uses GPU to compromise an Android phone

JavaScript based GLitch pwns browsers by flipping bits inside memory chips.

DAN GOODIN - 5/3/2018, 12:00 PM

Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

Pietro Frigo  
Vrije Universiteit Amsterdam  
p.frigo@vu.nl

Cristiano Giuffrida  
Vrije Universiteit Amsterdam  
giuffrida@cs.vu.nl

Herbert Bos  
Vrije Universiteit Amsterdam  
herbertb@cs.vu.nl

Kaveh Razavi  
Vrije Universiteit Amsterdam  
kaveh@cs.vu.nl
More Security Implications (IV)

- Rowhammer over RDMA (I)

Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM

Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar
VU Amsterdam

Radhesh Krishnan
VU Amsterdam

Elias Athanasopoulos
University of Cyprus

Cristiano Giuffrida
VU Amsterdam

Herbert Bos
VU Amsterdam

Kaveh Razavi
VU Amsterdam
More Security Implications (V)

- Rowhammer over RDMA (II)

Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests

Nethammer: Inducing Rowhammer Faults through Network Requests

Moritz Lipp  
Graz University of Technology

Misiker Tadesse Aga  
University of Michigan

Michael Schwarz  
Graz University of Technology

Daniel Gruss  
Graz University of Technology

Clémentine Maurice  
Univ Rennes, CNRS, IRISA

Lukas Raab  
Graz University of Technology

Lukas Lamster  
Graz University of Technology
More Security Implications (VI)

- IEEE S&P 2020

RAMBleed: Reading Bits in Memory Without Accessing Them

Andrew Kwong  
*University of Michigan*  
ankwong@umich.edu

Daniel Genkin  
*University of Michigan*  
genkin@umich.edu

Daniel Gruss  
*Graz University of Technology*  
daniel.gruss@iaik.tugraz.at

Yuval Yarom  
*University of Adelaide and Data61*  
yval@cs.adelaide.edu.au
More Security Implications (VII)

- USENIX Security 2019

Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks

Sanghyun Hong, Pietro Frigo†, Yiğitcan Kaya, Cristiano Giuffrida†, Tudor Dumitraș

University of Maryland, College Park
†Vrije Universiteit Amsterdam

A Single Bit-flip Can Cause Terminal Brain Damage to DNNs

One specific bit-flip in a DNN’s representation leads to accuracy drop over 90%

Our research found that a specific bit-flip in a DNN’s bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...
More Security Implications (VIII)

USENIX Security 2020

DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

Fan Yao
University of Central Florida
fan.yao@ucf.edu

Adnan Siraj Rakin
Arizona State University
asrakin@asu.edu

Deliang Fan
dfan@asu.edu

Degrade the inference accuracy to the level of Random Guess

Example: ResNet-20 for CIFAR-10, 10 output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**
Rowhammer RAM attack adapted to hit flash storage

Project Zero's two-year-old dog learns a new trick

By Richard Chirgwin 17 Aug 2017 at 04:27

From random block corruption to privilege escalation:
A filesystem attack vector for rowhammer-like attacks

Anil Kurmus      Nikolas Ioannou      Matthias Neugschwandtner      Nikolaos Papandreou
Thomas Parnell
IBM Research – Zurich
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]
RowHammer: 2019 and Beyond…

  - Preliminary arXiv version
  - Slides from COSADE 2019 (pptx)
  - Slides from VLSI-SOC 2020 (pptx) (pdf)
  - Talk Video (1 hr 15 minutes, with Q&A)

RowHammer: A Retrospective

Onur Mutlu\textsuperscript{§\$} \hspace{1cm} Jeremie S. Kim\textsuperscript{¶§}

\textsuperscript{§}ETH Zürich \hspace{1cm} \textsuperscript{¶}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\(^{\$\dagger}\), Minesh Patel\(^{\$}\), A. Giray Yaşlıkçı\(^{\$}\)
Hasan Hassan\(^{\$}\), Roknoddin Azizi\(^{\$}\), Lois Orosa\(^{\$}\), Onur Mutlu\(^{\$\dagger}\)

\(^{\$}\)ETH Zürich \quad \dagger\)Carnegie Mellon University
Key Takeaways from 1580 Chips

• Newer DRAM chips are more vulnerable to RowHammer

• There are chips today whose weakest cells fail after only 4800 hammers

• Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in more rows and 2) farther away from the victim row.

• Existing mitigation mechanisms are NOT effective
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*, and Kaveh Razavi*

*Vrije Universiteit Amsterdam 
§ETH Zürich 
¶Qualcomm Technologies Inc.
RowHammer in 2020 (III)

- 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)]

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§†
Microsoft Research, §ETH Zürich, †CMU, ‡MIT
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)]

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

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

¹ETH Zürich ²University of Illinois at Urbana–Champaign
Detailed Lectures on RowHammer

- Computer Architecture, Fall 2020, Lecture 4b
  - RowHammer (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=K Dy632z23UE&list=PL5Q2soXY2Zi9xidyIgBxU z7xRPS-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=PL5Q2soXY2Zi9xidyIgBxU z7xRPS-wisBN&index=9

- Computer Architecture, Fall 2020, Lecture 5b
  - RowHammer in 2020: Revisiting RowHammer (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=qR7XR-Eepcq&list=PL5Q2soXY2Zi9xidyIgBxU z7xRPS-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=PL5Q2soXY2Zi9xidyIgBxU z7xRPS-wisBN&index=11

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)]
Two RowHammer Papers at MICRO 2021

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu,

"A Deeper Look into RowHammer’s Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"

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

[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (21 minutes)]
[Lightning Talk Video (1.5 minutes)]
[arXiv version]

A Deeper Look into RowHammer’s Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses

Lois Orosa*
ETH Zürich

A. Giray Yaglıkçı*
ETH Zürich

Haocong Luo
ETH Zürich

Ataberk Olgun
ETH Zürich, TOBB ETÜ

Jisung Park
ETH Zürich

Hasan Hassan
ETH Zürich

Minesh Patel
ETH Zürich

Jeremie S. Kim
ETH Zürich

Onur Mutlu
ETH Zürich
Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, and Onur Mutlu,
"Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications"
Proceedings of the 54th International Symposium on Microarchitecture (MICRO), Virtual, October 2021.
[Slides (pptx) (pdf)]
[Short Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (25 minutes)]
[Lightning Talk Video (100 seconds)]
arXiv version

Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications

Hasan Hassan† Yahya Can Tuğrul†‡ Kaveh Razavi† Jeremie S. Kim† Onur Mutlu† Victor van der Veenσ
†ETH Zürich ‡TOBB University of Economics & Technology σQualcomm Technologies Inc.
A New RowHammer Paper at DSN 2022

- A. Giray Yağlıkçi, Haocong Luo, Geraldo F. de Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S. Kim, Lois Orosa, and Onur Mutlu, "Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices"

Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Baltimore, MD, USA, June 2022.

[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[arXiv version]
[Talk Video (34 minutes, including Q&A)]
[Lightning Talk Video (2 minutes)]

Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

A. Giray Yağlıkçi\textsuperscript{1} Haocong Luo\textsuperscript{1} Geraldo F. de Oliviera\textsuperscript{1} Ataberk Olgun\textsuperscript{1} Minesh Patel\textsuperscript{1} Jisung Park\textsuperscript{1} Hasan Hassan\textsuperscript{1} Jeremie S. Kim\textsuperscript{1} Lois Orosa\textsuperscript{1,2} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \hspace{1cm} \textsuperscript{2}Galicia Supercomputing Center (CESGA)
RowHammer is still an open problem

Security by obscurity is likely not a good solution

Source: J. Masters, Redhat, FOSDEM 2018 keynote talk.
Meltdown and Spectre

- Someone can steal secret data from the system even though
  - your program and data are perfectly correct and
  - your hardware behaves according to the specification and
  - there are no software vulnerabilities/bugs

- Why?
  - Speculative execution leaves traces of secret data in the processor’s cache (internal storage)
    - It brings data that is not supposed to be brought/accessed if there was no speculative execution
  - A malicious program can inspect the contents of the cache to “infer” secret data that it is not supposed to access
  - A malicious program can actually force another program to speculatively execute code that leaves traces of secret data
More on Meltdown/Spectre Vulnerabilities

Project Zero

News and updates from the Project Zero team at Google

Wednesday, January 3, 2018

Reading privileged memory with a side-channel

Posted by Jann Horn, Project Zero

We have discovered that CPU data cache timing can be abused to efficiently leak information out of mis-speculated execution, leading to (at worst) arbitrary virtual memory read vulnerabilities across local security boundaries in various contexts.

Many Interesting Things Are Happening Today in Computer Architecture
Many Interesting Things Are Happening Today in Computer Architecture

More Demanding Workloads
As applications push boundaries, computing platforms will become increasingly strained.
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* ▼

Oxford Nanopore MinION

Data → performance & energy bottleneck
200 Oxford Nanopore sequencers have left UK for China, to support rapid, near-sample coronavirus sequencing for outbreak surveillance.

Fri 31st January 2020

Following extensive support of, and collaboration with, public health professionals in China, Oxford Nanopore has shipped an additional 200 MinION sequencers and related consumables to China. These will be used to support the ongoing surveillance of the current coronavirus outbreak, adding to a large number of the devices already installed in the country.

Each MinION sequencer is approximately the size of a stapler, and can provide rapid sequence information about the coronavirus.

700Kg of Oxford Nanopore sequencers and consumables are on their way for use by Chinese scientists in understanding the current coronavirus outbreak.

Source: https://nanoporetech.com/about-us/news/200-oxford-nanopore-sequencers-have-left-uk-china-support-rapid-near-sample
Population-Scale Microbiome Profiling

https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/
City-Scale Microbiome Profiling

Afshinneko+, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics", Cell Systems, 2015
Example: Rapid Surveillance of Ebola Outbreak

High-Throughput Genome Sequencers

... and more! All produce data with different properties.
High-Throughput Genome Sequencers

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


MinION from ONT

Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

SmidgION from ONT
The Genomic Era

development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

Genome Analysis

Read Mapping

Data → performance & energy bottleneck

Sequencing

Read Alignment

Read Mapping

Variant Calling

Scientific Discovery
Software Acceleration: Eliminate Useless Work

- Download the source code and try for yourself
  - Download link to FastHASH

Xin et al. BMC Genomics 2013, 14(Suppl 1):S13
http://www.biomedcentral.com/1471-2164/14/S1/S13

Accelerating read mapping with FastHASH

Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu¹*, Can Alkan³*

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013
Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

Hongyi Xin1,*, John Greth2, John Emmons2, Gennady Pekhimenko1, Carl Kingsford3, Can Alkan4,*, and Onur Mutlu2,*

GateKeeper: FPGA-Based Alignment Filtering

1st FPGA-based Alignment Filter.

- **x10^{12}** mappings
- **x10^3** mappings

1. High throughput DNA sequencing (HTS) technologies
2. Read Pre-Alignment Filtering
   - Fast & Low False Positive Rate
3. Read Alignment
   - Slow & Zero False Positives
GateKeeper: FPGA-Based Alignment Filtering

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan

"GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"

Bioinformatics, [published online, May 31], 2017.

[Source Code]
[Online link at Bioinformatics Journal]

GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu, Can Alkan

Bioinformatics, Volume 33, Issue 21, 1 November 2017, Pages 3355–3363,
https://doi.org/10.1093/bioinformatics/btx342

Published: 31 May 2017   Article history ▼
In-Memory DNA Sequence Analysis

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu,
  "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"
  *BMC Genomics*, 2018.
  Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
  [Slides (pptx) (pdf)]
  [Source Code]
  [arxiv.org Version (pdf)]
  [Talk Video at AACBB 2019]

GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies

Jeremie S. Kim¹,⁶*, Damla Senol Cali¹, Hongyi Xin², Donghyuk Lee³, Saugata Ghose¹, Mohammed Alser⁴, Hasan Hassan⁶, Oguz Ergin⁵, Can Alkan⁴* and Onur Mutlu⁶,¹*

*From* The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

SAFARI
Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,
"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"

*Source Code*
*Online link at Bioinformatics Journal*
Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu, "SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs". Bioinformatics, to appear in 2020.

[Source Code]
[Online link at Bioinformatics Journal]
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)]
In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,

"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]

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

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

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

SAFARI
New Applications: Graph Genomes [ISCA 2022]

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhias, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

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


[Slides (pptx) (pdf)]
[arXiv version]

**SeGraM: Accelerating Genomic Sequence-to-Graph Mapping via Algorithm/Hardware Co-Design**

<table>
<thead>
<tr>
<th>Damla Senol Cali</th>
<th>Konstantinos Kanellopoulos</th>
<th>Joel Lindegger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bionano Genomics</td>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
</tr>
<tr>
<td>USA</td>
<td>Switzerland</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Züal Bingöl</td>
<td>Gagandeep Singh</td>
<td>Ziyi Zuo</td>
</tr>
<tr>
<td>Bilken University</td>
<td>ETH Zurich</td>
<td>Carnegie Mellon University</td>
</tr>
<tr>
<td>Turkey</td>
<td>Switzerland</td>
<td>USA</td>
</tr>
<tr>
<td>Can Firtina</td>
<td>Meryem Banu Cavlak</td>
<td>Nika Mansouri Ghias</td>
</tr>
<tr>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Switzerland</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Jeremie Kim</td>
<td>Mohammed Alser</td>
<td>Juan Gómez Luna</td>
</tr>
<tr>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Switzerland</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Nour Almadhoun Alserr</td>
<td>ETH Zurich</td>
<td>Sreenivas Subramoney</td>
</tr>
<tr>
<td>ETH Zurich</td>
<td>ETH Zurich</td>
<td>Intel Labs</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Switzerland</td>
<td>India</td>
</tr>
<tr>
<td>Can Alkan</td>
<td>Saugata Ghose</td>
<td>Onur Mutlu</td>
</tr>
<tr>
<td>Bilken University</td>
<td>University of Illinois</td>
<td>ETH Zurich</td>
</tr>
<tr>
<td>Turkey</td>
<td>Urbana-Champaign</td>
<td>Switzerland</td>
</tr>
<tr>
<td></td>
<td>USA</td>
<td></td>
</tr>
</tbody>
</table>
New Applications: Ref Genome Updates

AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim¹, Can Firtina¹, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar¹,⁴, Mohammed Alser¹, Can Alkan² and Onur Mutlu¹,²,³*

Future of Genome Sequencing & Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
COVID-19 Nanopore Sequencing (I)

SARS-CoV-2 Whole genome sequencing

- RT Step: ~ 1 hr
- PCR: ~ 2.30 hr
- Add Barcodes: ~ 1 hr
- Add Adapter: ~ 30 m
- Sequence: ~ 1 hr
- Analyse: ~ 1 hr

7 hr

RNA to answer

Of which ~1 hr sequencing time

- From ONT (https://nanoporetech.com/covid-19/overview)
COVID-19 Nanopore Sequencing (II)

How are scientists using nanopore sequencing to research COVID-19?

Samples are collected → Validated SARS-CoV-2 RT-PCR test performed

- SARS-CoV-2 positive samples
- SARS-CoV-2 negative samples: used as negative controls

**How can this be used?**
- Genomic epidemiology: analyse variants & mutation rate, track spread of virus, identify clusters of transmission

**What are the results?**
- From RNA to full SARS-CoV-2 consensus sequence in ~7 hours

**How?**
- Targeted amplification of SARS-CoV-2 genome + multiplexed, rapid nanopore sequencing

**Targeted SARS-CoV-2 nanopore sequencing**

**Flowchart:**

- **Metagenomic nanopore sequencing**
  - SARS-CoV-2 Direct RNA whole genome sequencing: assess viral genome in its native RNA form and the effect of base modifications
  - Immune repertoire: assess response of the immune system to SARS-CoV-2 infection by sequencing of full-length immune cell receptor genes and transcripts
  - Whole human genome sequencing: investigate what might cause different responses to the virus in different people based on their genome

**What’s next?**

- Find out more at nanoporetech.com/covid19

[From ONT](https://nanoporetech.com/covid-19/overview)
Accelerating Genome Analysis: A Primer on an Ongoing Journey

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)]
More on Fast 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

https://www.youtube.com/onurmutlulectures
Many Interesting Things Are Happening Today in Computer Architecture

More Demanding Workloads
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**
[Mao+ (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

Data → performance & energy bottleneck

Chrome

TensorFlow Mobile

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\(^1\)  Rachata Ausavarungnirun\(^1\)  Saugata Ghose\(^1\)  Youngsok Kim\(^2\)
Aki Kuusela\(^3\)  Allan Knies\(^3\)  Eric Shiu\(^3\)  Rahul Thakur\(^3\)  Daehyun Kim\(^4,3\)
Parthasarathy Ranganathan\(^3\)  Onur Mutlu\(^5,1\)
A memory access consumes $\sim 100\text{-}1000\times$ the energy of a complex addition.
Many Interesting Things Are Happening Today in Computer Architecture
Many Novel Concepts Investigated Today

- **New Computing Paradigms (Rethinking the Full Stack)**
  - Processing in Memory, Processing Near Data
  - Neuromorphic Computing
  - Fundamentally Secure and Dependable Computers

- **New Accelerators (Algorithm-Hardware Co-Designs)**
  - Artificial Intelligence & Machine Learning
  - Graph Analytics
  - Genome Analysis

- **New Memories and Storage Systems**
  - Non-Volatile Main Memory
  - Intelligent Memory
As applications push boundaries, computing platforms will become increasingly strained.
Increasingly Diverging/Complex Tradeoffs

Communication Dominates Arithmetic

Dally, HiPEAC 2015
A memory access consumes $\sim 1000X$ the energy of a complex addition.
Increasingly Complex Systems

Past systems

Microprocessor  Main Memory  Storage (SSD/HDD)
Increasingly Complex Systems

- FPGAs
- Hybrid Main Memory
- Persistent Memory/Storage
- (General Purpose) GPUs
- Heterogeneous Processors and Accelerators

Modern systems
Computer Architecture Today

- Computing landscape is very different from 10-20 years ago
- Applications and technology both demand novel architectures

- Heterogeneous Processors and Accelerators
- Hybrid Main Memory
- General Purpose GPUs
- Persistent Memory/Storage

Every component and its interfaces, as well as entire system designs are being re-examined.
You can revolutionize the way computers are built, if you understand both the hardware and the software (and change each accordingly)

You can invent new paradigms for computation, communication, and storage

Recommended book: Thomas Kuhn, “The Structure of Scientific Revolutions” (1962)
- Pre-paradigm science: no clear consensus in the field
- Normal science: dominant theory used to explain/improve things (business as usual); exceptions considered anomalies
- Revolutionary science: underlying assumptions re-examined
You can revolutionize the way computers are built, if you understand both the hardware and the software (and change each accordingly)

You can invent new paradigms for computation, communication...

Recommended book: Thomas Kuhn, "The Structure of Scientific Revolutions" (1962)

- Pre-paradigm science: no clear consensus in the field
- Normal science: dominant theory used to explain/improve things (business as usual); exceptions considered anomalies
- Revolutionary science: underlying assumptions re-examined
Takeaways

- It is an exciting time to be understanding and designing computing architectures

- Many challenging and exciting problems in platform design
  - That no one has tackled (or thought about) before
  - That can have huge impact on the world’s future

- Driven by huge hunger for data (Big Data), new applications (ML/AI, graph analytics, genomics), ever-greater realism, ...
  - We can easily collect more data than we can analyze/understand

- Driven by significant difficulties in keeping up with that hunger at the technology layer
  - Five walls: Energy, reliability, complexity, security, scalability
Let’s Start with Some Fundamentals
Question: What Is This?
**Answer: The First Major Piece of a Famous Architect**

- **Bahnhof Stadelhofen:** “The train station has several of the features that became signatures of his work; straight lines and right angles are rare.”

- **ETH Alumnus, PhD in Civil Engineering**

  **Santiago Calatrava Valls** (born 28 July 1951) is a Spanish architect, structural engineer, sculptor and painter, particularly known for his bridges supported by single leaning pylons, and his railway stations, stadiums, and museums, whose sculptural forms often resemble living organisms.[1] His best-known works include the Milwaukee Art Museum, the Turning Torso tower in Malmö, Sweden, the Margaret Hunt Hill Bridge in Dallas, Texas, and the Museum of Tomorrow in Rio de Janeiro,
Compare To This

Source: http://cookiemagik.deviantart.com/art/Train-station-207266944
Question 2: What Is This?

Design

Calatrava said that the Oculus resembles a bird being released from a child's hand. The roof was originally designed to mechanically open to increase light and ventilation to the enclosed space. Herbert Muschamp, architecture critic of *The New York Times*, compared the design to the *Bethesda Terrace and Fountain* in Central Park, and wrote in 2004:
Strengths and Praise

“Santiago Calatrava's design for the World Trade Center PATH station should satisfy those who believe that buildings planned for ground zero must aspire to a spiritual dimension. Over the years, many people have discerned a metaphysical element in Mr. Calatrava's work. I hope New Yorkers will detect its presence, too. With deep appreciation, I congratulate the Port Authority for commissioning Mr. Calatrava, the great Spanish architect and engineer, to design a building with the power to shape the future of New York. It is a pleasure to report, for once, that public officials are not overstating the case when they describe a design as breathtaking.” [43]
However, Calatrava's original soaring spike design was scaled back because of security issues. The New York Times observed in 2005:

“In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass. [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural expressiveness. It may now evoke a slender stegosaurus more than it does a bird.” [45]
Stegosaurus

From Wikipedia, the free encyclopedia

For the pachycephalosaurid of a similar name, see Stegoceras.

Stegosaurus (/ˈstɛɡəsɔːrəs/)[1] is a genus of armored dinosaur. Fossils of this genus date to the Late Jurassic period, where they are found in Kimmeridgian to early Tithonian aged strata, between 155 and 150 million years ago, in the western United States and Portugal. Several

Source: https://en.wikipedia.org/wiki/Stegosaurus
Design Constraints: Noone is Immune

However, Calatrava's original soaring spike design was scaled back because of security issues. The *New York Times* observed in 2005:

"In the name of security, Santiago Calatrava's bird has grown a beak. Its ribs have doubled in number and its wings have lost their interstices of glass.... [T]he main transit hall, between Church and Greenwich Streets, will almost certainly lose some of its delicate quality, while gaining structural expressiveness. It may now evoke a slender *stegosaurus* more than it does a bird."[45]

The design was further modified in 2008 to eliminate the opening and closing roof mechanism because of budget and space constraints. [46]

The Transportation Hub has been dubbed "the world's most expensive transportation hub" for its massive cost for reconstruction—$3.74 billion dollars.[48][58] By contrast, the proposed two-mile PATH extension

Source: https://en.wikipedia.org/wiki/World_Trade_Center_station_(PATH)
Question: What Is This?
Fallingwater or Kaufmann Residence is a house designed by architect Frank Lloyd Wright in 1935 in rural southwestern Pennsylvania, 43 miles (69 km) southeast of Pittsburgh.[4] The home was built partly over a waterfall on Bear Run in the Mill Run section of Stewart Township, Fayette County, Pennsylvania, in the Laurel Highlands of the Allegheny Mountains.

*Time* cited it after its completion as Wright's "most beautiful job";[5] it is listed among *Smithsonian*’s Life List of 28 places "to visit before you die."[6] It was designated a National Historic Landmark in 1966.[3] In 1991, members of the American Institute of Architects named the house the "best all-time work of American architecture" and in 2007, it was ranked twenty-ninth on the list of America's Favorite Architecture according to the AIA.

Source: https://en.wikipedia.org/wiki/Fallingwater
Your First Comp Arch Assignment

- Go and visit Bahnhof Stadelhofen
  - Extra credit: Repeat for Oculus
  - Extra+ credit: Repeat for Fallingwater

- Appreciate the beauty & out-of-the-box and creative thinking

- Think about tradeoffs in the design of the Bahnhof
  - Strengths, weaknesses, goals of design

- Derive principles on your own for good design and innovation

- Due date: **Any time during this course**
  - Later during the course is better
  - Apply what you have learned in this course
  - Think out-of-the-box
But First, Today’s First Assignment

- Find The Differences Of This and That
Find The Differences of This and That
That

Source: http://cookiemagik.deviantart.com/art/Train-station-207266944
Many Tradeoffs Between Two Designs

- You can list them after you complete the first assignment...
Aside: Evaluation Criteria for the Designs

- Functionality (Does it meet the specification?)
- Reliability
- Space requirement
- Cost
- Expandability
- Comfort level of users
- Happiness level of users
- Aesthetics
- ...

How to evaluate goodness of design is always a critical question.
A Key Question

How was Calatrava able to design especially his key buildings?

Can have many guesses

- (Ultra) hard work, perseverance, dedication (over decades)
- Experience
- Creativity, Out-of-the-box thinking
- A good understanding of past designs
- Good judgment and intuition
- Strong skill combination (math, architecture, art, engineering, ...)
- Funding ($$$), luck, initiative, entrepreneurialism
- **Strong understanding of and commitment to fundamentals**
- Principled design
- ...

(You will be exposed to and hopefully develop/enhance many of these skills in this course)
Principled Design

“To me, there are two overriding principles to be found in nature which are most appropriate for building:

- one is the optimal use of material,
- the other the capacity of organisms to change shape, to grow, and to move.”

Santiago Calatrava

“Calatrava's constructions are inspired by natural forms like plants, bird wings, and the human body.”

Source: http://www.arcspace.com/exhibitions/unsorted/santiago-calatrava/
A Principled Design

Zoomorphic architecture

From Wikipedia, the free encyclopedia

**Zoomorphic architecture** is the practice of using animal forms as the inspirational basis and blueprint for architectural design. "While animal forms have always played a role adding some of the deepest layers of meaning in architecture, it is now becoming evident that a new strand of **biomorphism** is emerging where the meaning derives not from any specific representation but from a more general allusion to biological processes."[1]

Some well-known examples of Zoomorphic architecture can be found in the **TWA Flight Center** building in New York City, by Eero Saarinen, or the **Milwaukee Art Museum** by Santiago Calatrava, both inspired by the form of a bird’s wings.[3]
What Does This Remind You Of?

What About This?
Milwaukee Art Museum

Source: By Andrew C. from Flagstaff, USA - Flickr, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=379223
Athens Olympic Stadium

Source: By Spyrodrakopoulos - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=16172519
City of Arts and Sciences, Valencia

Florida Polytechnic University (I)

Source: http://www.architectmagazine.com/design/buildings/florida-polytechnic-university-designed-by-santiago-calatrava_o
Oculus, New York City

A Quote from The Other Famous Architect

“architecture [...] based upon principle, and not upon precedent” (Frank Lloyd Wright)

Source: http://www.fallingwater.org/
A Principled Design

Organic architecture

From Wikipedia, the free encyclopedia

**Organic architecture** is a philosophy of architecture which promotes harmony between human habitation and the natural world through design approaches so sympathetic and well integrated with its site, that buildings, furnishings, and surroundings become part of a unified, interrelated composition.

A well-known example of organic architecture is **Fallingwater**, the residence Frank Lloyd Wright designed for the Kaufmann family in rural Pennsylvania. Wright had many choices to locate a home on this large site, but chose to place the home directly over the waterfall and creek creating a close, yet noisy dialog with the rushing water and the steep site. The horizontal striations of stone masonry with daring **cantilevers** of colored beige concrete blend with native rock outcroppings and the wooded environment.
Another View

Source: https://roadtrippers.com/stories/falling-water
Yet Another View

Major High-Level Goals of This Course

- Understand the principles
- Understand the precedents

Based on such understanding:
- Enable you to evaluate tradeoffs of different designs and ideas
- Enable you to develop principled designs
- Enable you to develop novel, out-of-the-box designs

The focus is on:
- Principles, precedents, and how to use them for new designs

In Computer Architecture
Role of the (Computer) Architect

-- Look Backward (Examine old code)
-- Look forward (Listen to the dreamers)
-- Look Up (Nature of the problems)
-- Look Down (Predict the future of technology)

from Yale Patt’s lecture notes
Role of The (Computer) Architect

- **Look backward (to the past)**
  - Understand tradeoffs and designs, upsides/downsides, past workloads. Analyze and evaluate the past.

- **Look forward (to the future)**
  - Be the dreamer and create new designs. Listen to dreamers.
  - Push the state of the art. Evaluate new design choices.

- **Look up (towards problems in the computing stack)**
  - Understand important problems and their nature.
  - Develop architectures and ideas to solve important problems.

- **Look down (towards device/circuit technology)**
  - Understand the capabilities of the underlying technology.
  - Predict and adapt to the future of technology (you are designing for N years ahead). Enable the future technology.
Takeaways

- Being an architect is not easy
- You need to consider **many** things in designing a new system + have good intuition/insight into ideas/tradeoffs

- But, it is fun and can be very rewarding
- And, enables a great future
  - E.g., many scientific and everyday-life innovations would not have been possible without architectural innovation that enabled very high performance systems
  - E.g., your mobile phones
  - E.g., self-driving vehicles

- **This course will enable you to become a good computer architect**
So, I Hope You Are Here for This

Comp. Systems

- How does an assembly program end up executing as digital logic?
- What happens in-between?
- How is a computer designed using logic gates and wires to satisfy specific goals?

Digital Design

“C” as a model of computation
Programmer’s view of how a computer system works

Architect/microarchitect’s view: How to design a computer that meets system design goals.
Choices critically affect both the SW programmer and the HW designer

HW designer’s view of how a computer system works
Digital logic as a model of computation
Levels of Transformation

“The purpose of computing is [to gain] insight” (Richard Hamming)

We gain and generate insight by solving problems

How do we ensure problems are solved by electrons?

**Algorithm**

Step-by-step procedure that is **guaranteed to terminate** where each step is precisely stated and can be carried out by a computer

- Finiteness
- Definiteness
- Effective computability

Many algorithms for the same problem

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>ISA (Architecture)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Program/Language</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Runtime System (VM, OS, MM)</td>
<td></td>
</tr>
<tr>
<td>ISA (Architecture)</td>
<td>Microarchitecture</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Devices</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Electrons</td>
<td></td>
</tr>
</tbody>
</table>

**Microarchitecture**

An implementation of the ISA

**Digital logic circuits**

Building blocks of micro-arch (e.g., gates)

ISA (Instruction Set Architecture)

Interface/contract between SW and HW.

What the programmer assumes hardware will satisfy.
Aside: An Important Work By Hamming


-Introduced the concept of Hamming distance
  -number of locations in which the corresponding symbols of two equal-length strings is different

-Developed a theory of codes used for error detection and correction

- Also see:
A user-centric view: computer designed for users

- The entire stack should be optimized for user
The Power of Abstraction

- **Levels of transformation create abstractions**
  - Abstraction: A higher level only needs to know about the interface to the lower level, not how the lower level is implemented
  - E.g., high-level language programmer does not really need to know what the ISA is and how a computer executes instructions

- **Abstraction improves productivity**
  - No need to worry about decisions made in underlying levels
  - E.g., programming in Java vs. C vs. assembly vs. binary vs. by specifying control signals of each transistor every cycle

- Then, why would you want to know what goes on underneath or above?
Crossing the Abstraction Layers

- As long as everything goes well, not knowing what happens underneath (or above) is not a problem.

- What if
  - The program you wrote is running slow?
  - The program you wrote does not run correctly?
  - The program you wrote consumes too much energy?
  - Your system just shut down and you have no idea why?
  - Someone just compromised your system and you have no idea how?

- What if
  - The hardware you designed is too hard to program?
  - The hardware you designed is too slow because it does not provide the right primitives to the software?

- What if
  - You want to design a much more efficient and higher performance system?
Crossing the Abstraction Layers

Two key goals of this course are

- to understand how a processor works underneath the software layer and how decisions made in hardware affect the software/programmer

- to enable you to be comfortable in making design and optimization decisions that cross the boundaries of different layers and system components
An Example: Multi-Core Systems

*Die photo credit: AMD Barcelona
Another Example: Memory Refresh
