No Required Readings
(for this lecture)
Recall: Major High-Level Goals of This Course

- In Digital Circuits & Computer Architecture
- Understand the basics
- Understand the principles (of design)
- Understand the precedents

Based on such understanding:
- learn how a modern computer works underneath
- evaluate tradeoffs of different designs and ideas
- implement a principled design (a simple microprocessor)
- learn to systematically debug increasingly complex systems
- Hopefully enable you to develop novel, out-of-the-box designs

The focus is on basics, principles, precedents, and how to use them to create/implement good designs
Recall: Why These Goals?

- Because you are here for a Computer Science degree

- Regardless of your future direction, learning the principles of digital design & computer architecture will be useful to
  - design better hardware
  - design better software
  - design better systems
  - make better tradeoffs in design
  - understand why computers behave the way they do
  - solve problems better
  - think “in parallel”
  - think critically
  - ...
We Have Come A Long Way

- Started from Transistor as the Building Block
- Logic Design (Combinational, Sequential, Timing)
- Execution Model, ISA and Microarchitecture
- Many Key Processing Paradigms
- The Memory System and the Memory Hierarchy
- Built up to Virtual Memory & System Software Mechanisms

Takeaway 1: **All we covered is real and used in real systems →** and, it is increasingly important

Takeaway 2: **Principles we covered apply broadly**

Takeaway 3: **Tradeoff analysis and critical thinking that you are exposed to apply even more broadly**
Recall: We Are Done With This…

- Single-cycle Microarchitectures
- Multi-cycle and Microprogrammed Microarchitectures
- Pipelining
- Issues in Pipelining: Control & Data Dependence Handling, State Maintenance and Recovery, ...
- Out-of-Order Execution
- Other Execution Paradigms
Approaches to (Instruction-Level) Concurrency

- Pipelining
- Fine-Grained Multithreading
- Out-of-order Execution
- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Systolic Arrays
- Decoupled Access Execute
- SIMD Processing (Vector and Array processors, GPUs)

Now you are very familiar with many processing paradigms
Approaches to (Instruction-Level) Concurrency

- Pipelining
- Fine-Grained Multithreading
- Out-of-order Execution
- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Systolic Arrays
- Decoupled Access Execute
- SIMD Processing (Vector and Array processors, GPUs)

Food for thought: tradeoffs of these different processing paradigms
We Are Also Done With This…

- Memory Organization & Technology
- Memory Hierarchy & Caches
- Advanced Caches
- Prefetching
- Virtual Memory

Now you are very familiar with memory systems
We Are Also Done With This…

- Memory Organization & Technology
- Memory Hierarchy & Caches
- Advanced Caches
- Prefetching
- Virtual Memory

Food for thought: tradeoffs of many different memory system designs & ideas
The Transformation Hierarchy

Food for thought:
how do tradeoffs span and affect the hierarchy
The Best Way to Approach This Course

- Take it as **a learning and growth experience**... all of it
- What we saw changed the world & endured the test of time...
- And, it will be more important...
- You may not all be future architects, but...
  - your development and thinking can greatly benefit from the concepts, tradeoffs, principles, critical thinking...
- **Focus on understanding, learning, critical analysis**
  - these are the agents for your growth
  - the course is designed to activate these agents (lifelong)
What We Did Not Cover
Computer Architecture is Very Rich

- Many ideas, much creativity, many tradeoffs and problems

- As scaling, performance, energy, reliability, security issues become worse in circuits and in software, computer architecture will be more and more important

- Already obvious in
  - AI/ML accelerators
  - Hardware security issues
  - Novel execution paradigms: Processing in memory
  - ...

- See lecture: Intelligent Architectures for Intelligent Machines
  - https://www.youtube.com/watch?v=5YKvtNM6XzY
Intelligent Architectures for Intelligent Machines

Onur Mutlu,
"Intelligent Architectures for Intelligent Machines"
Opening Talk at TU Vienna Mondays in Memory Webinar Series (MiM), Virtual, 3 May 2021.
[Slides (pptx) (pdf)] [Talk Video (57 minutes)]

https://www.youtube.com/watch?v=5YKvtNM6XzY
A Tutorial on Memory-Centric Systems

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

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

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

Data-centric

Data-driven

Data-aware
Recall: 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
State of the Art

- This is a great time to be a computer architect

- Circuits strained
- Applications ever more demanding
- Multiple possible emerging technologies
- Many requirements, many systems
- Many, many security, reliability issues
- Many big problems waiting to be solved
  - All across the hierarchy
- ...

Many big innovations require computer architecture
Many Interesting Things Are Happening Today in Computer Architecture

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

250 TFLOPS per chip 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: TPU3

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
An Example Modern Systolic Array: TPU2


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
An Example Modern Systolic Array: TPU (I)

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.

Cerebras’s Wafer Scale Engine-2 (2021)

- The largest ML accelerator chip
- 850,000 cores
- 40 GB of on-chip memory
- 20 PB/s memory bandwidth

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

Largest GPU
54.2 Billion transistors
826 mm²

https://cerebras.net/product/#overview

NVIDIA Ampere GA100
Cerebras’s Wafer Scale Engine (2019)

- The largest ML accelerator chip
- 400,000 cores
- 18 GB of on-chip memory
- 9 PB/s memory bandwidth

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
Many Other Ideas and Topics

- Advanced memory systems
- Self-optimizing architectures (using machine learning principles)
- Processing-in-memory systems
- Emerging memory technologies
- Solid state disks and storage, I/O
- Interconnection networks, communication networks
- Many issues in multiprocessing and multithreading
- Distributed architectures
- Heterogeneous systems: Heterogeneous CPU-GPU-FPGA-HWAcc architectures
- QoS and predictable performance
- Reliable and secure architectures
- Programmability, portability
- Better HW/SW interfaces
- Reconfigurable computing
- Specialized and domain-specific architectures – genomics, medicine, health, AI/ML
- Unconventional architectures – nature-inspired, quantum, molecular, ...

... Covered in advanced computer architecture course(s)
Oculus, New York City

Recall:
Runahead Execution
**Perfect Caches:**

Load 1 Hit  
Load 2 Hit

**Small OoO Instruction Window:**

Load 1 Miss  
Load 2 Miss

**Runahead:**

Load 1 Miss  
Load 2 Miss  
Load 1 Hit  
Load 2 Hit

Saved Cycles
Effective prefetching can both improve performance and reduce hardware cost.
More on Runahead Execution

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"  

One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).

[Lecture Slides (pptx) (pdf)]
[Lecture Video (1 hr 54 mins)]
[Retrospective HPCA Test of Time Award Talk Slides (pptx) (pdf)]
[Retrospective HPCA Test of Time Award Talk Video (14 minutes)]

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu §  Jared Stark †  Chris Wilkerson ‡  Yale N. Patt §

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

†Microprocessor Research  
Intel Labs  
jared.w.stark@intel.com

‡Desktop Platforms Group  
Intel Corporation  
chris.wilkerson@intel.com
HIGH-PERFORMANCE THROUGHPUT COMPUTING

Throughput computing, achieved through multithreading and multicore technology, can lead to performance improvements that are 10 to 30× those of conventional processors and systems. However, such systems should also offer good single-thread performance. Here, the authors show that hardware scouting increases the performance of an already robust core by up to 40 percent for commercial benchmarks.

More on Runahead in Sun ROCK

Simultaneous Speculative Threading: A Novel Pipeline Architecture Implemented in Sun’s ROCK Processor

Shailender Chaudhry, Robert Cypher, Magnus Ekman, Martin Karlsson, Anders Landin, Sherman Yip, Håkan Zeffer, and Marc Tremblay
Sun Microsystems, Inc.
4180 Network Circle, Mailstop SCA18-211
Santa Clara, CA 95054, USA
{shailender.chaudhry, robert.cypher, magnus.ekman, martin.karlsson, anders.landin, sherman.yip, haakan.zeffer, marc.tremblay}@sun.com

Runahead Execution vs. Conventional Data Prefetching in the IBM POWER6 Microprocessor

Harold W. Cain        Priya Nagpurkar

IBM T.J. Watson Research Center
Yorktown Heights, NY
{tcain, pnagpurkar}@us.ibm.com

Abstract

After many years of prefetching research, most commercially available systems support only two types of prefetching: software-directed prefetching and hardware-based prefetchers using simple sequential or stride-based prefetching algorithms. More sophisticated prefetching proposals, despite promises of improved performance, have not been adopted by industry. In this paper, we explore the efficacy of both hardware and software prefetching in the context of an IBM POWER6 commercial server. Using a variety of applications that have been compiled with an aggressively optimizing compiler to use software prefetching when appropriate, we perform the first study of a new runahead prefetching feature adopted by the POWER6 design, evaluating it in isolation and in conjunction with a conventional hardware-based sequential stream prefetcher and compiler-inserted software prefetching.

We find that the POWER6 implementation of runahead prefetching is quite effective on many of the memory intensive applications studied; in isolation it improves performance as much as 36% and on average 10%. However, it outperforms the hardware-based stream prefetcher on only two of the benchmarks studied, and in those by a small margin. When used in conjunction with the conventional prefetching mechanisms, the runahead feature adds an additional 6% on average, and 39% in the best case (GemsFDTD).
Runahead Execution in NVIDIA Denver

DENVER: NVIDIA’S FIRST 64-BIT ARM PROCESSOR

NVIDIA’s first 64-bit ARM processor, code-named Denver, leverages a host of new technologies, such as dynamic code optimization, to enable high-performance mobile computing. Implemented in a 28-nm process, the Denver CPU can attain clock speeds of up to 2.5 GHz. This article outlines the Denver architecture, describes its technological innovations, and provides relevant comparisons against competing mobile processors.

Runahead Execution in NVIDIA Denver

Reducing the effects of long cache-miss penalties has been a major focus of the microarchitecture, using techniques like prefetching and run-ahead. An aggressive hardware prefetcher implementation detects L2 cache requests and tracks up to 32 streams, each with complex stride patterns.

Run-ahead uses the idle time that a CPU spends waiting on a long latency operation to discover cache and DT LB misses further down the instruction stream and generates prefetch requests for these misses. These prefetch requests warm up the data cache and DT LB well before the actual execution of the instructions that require the data. Run-ahead complements the hardware prefetcher because it's better at prefetching nonstrided streams, and it trains the hardware prefetcher faster than normal execution to yield a combined benefit of 13 percent on SPECint2000 and up to 60 percent on SPECfp2000.

The core includes a hardware prefetch unit that Boggs describes as “aggressive” in preloading the data cache but less aggressive in preloading the instruction cache. It also implements a “run-ahead” feature that continues to execute microcode speculatively after a data-cache miss; this execution can trigger additional cache misses that resolve in the shadow of the first miss. Once the data from the original miss returns, the results of this speculative execution are discarded and execution restarts with the bundle containing the original miss, but run-ahead can preload subsequent data into the cache, thus avoiding a string of time-wasting cache misses. These and other features help Denver out-score Cortex-A15 by more than 2.6x on a memory-read test even when both use the same SoC framework (Tegra K1).


Gwennap, “NVIDIA’s First CPU is a Winner,” MPR 2014.
Runahead Readings

- **Required**

- **Recommended**
More on Runahead Execution

- Lecture video from Fall 2020, Computer Architecture:
  - https://www.youtube.com/watch?v=zPewo6IaJ_8

- Lecture video from Fall 2017, Computer Architecture:
  - https://www.youtube.com/watch?v=Kj3relihGF4

- Onur Mutlu,
  "Efficient Runahead Execution Processors"
  Nominated for the ACM Doctoral Dissertation Award by the University of Texas at Austin.

https://www.youtube.com/onurmutlulectures
More on Runahead Execution (I)

Review: Runahead Execution (Mutlu et al., HPCA 2003)

Small Window:
- Load 1 Miss
- Compute
- Stall
- Load 2 Miss
- Compute
- Stall
- Miss 1
- Miss 2

Runahead:
- Load 1 Miss
- Load 2 Miss
- Load 1 Hit
- Load 2 Hit
- Compute
- Runahead
- Compute
- Saved Cycles
- 18

Computer Architecture - Lecture 19a: Execution-Based Prefetching (ETH Zürich, Fall 2020)
395 views • Nov 29, 2020

https://www.youtube.com/watch?v=zPewo6laJ_8&list=PL5Q2soXY2Zi9xidylgBxUz7xRPS-wisBN&index=34
More on Runahead Execution (II)

Runahead Execution in NVIDIA Denver

Reducing the effects of long cache-miss penalties has been a major focus of the microarchitecture, using techniques like prefetching and run-ahead. An aggressive hardware prefetcher implementation detects L2 cache requests and tracks up to 32 streams, each with complex stride patterns.

Run-ahead uses the idle time that a CPU spends waiting on a long latency operation to discover cache and DTLB misses further down the instruction stream and generates prefetch requests for these misses. These prefetch requests warm up the data cache and DTLB well before the actual execution of the instructions that require the data. Run-ahead complements the hardware prefetcher because it’s better at prefetching non-strided streams, and it trains the hardware prefetcher faster than normal execution to yield a combined benefit of 13 percent on SPECint2000 and up to 60 percent on SPECfp2000.


Gwennap, “NVIDIA’s First CPU is a Winner,” MPR 2014.

https://www.youtube.com/watch?v=KFCOecRQTlc
My Suggestions to You
Suggestion to Researchers: Principle: Passion

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
You can make a good impact on the world
Recall: This
Recall: That

Source: http://cookiemagik.deviantart.com/art/Train-station-207266944 - Göttingen, DE
Recall: A Key Question

How was Calatrava able to design especially his key buildings?

Can have many guesses

- (Very) 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
Processing in Memory
The Problem

Data access is the major performance and energy bottleneck

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

Processing of data is performed far away from the data.
A Computing System

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

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.

A Computing System

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

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.
Today’s Computing Systems

- Are overwhelmingly processor centric
- All data processed in the processor → at great system cost
- Processor is heavily optimized and is considered the master
- Data storage units are dumb and are largely unoptimized (except for some that are on the processor die)
Yet ...

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

---

Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors".


One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).

Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu §  Jared Stark †  Chris Wilkerson ‡  Yale N. Patt §

§ECE Department  †Microprocessor Research  ‡Desktop Platforms Group
The University of Texas at Austin  Intel Labs  Intel Corporation
{onur,patt}@ece.utexas.edu  jared.w.stark@intel.com  chris.wilkerson@intel.com
The Performance Perspective (Today)

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

The Performance Perspective (Today)

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

![Diagram showing cache-bound cycles for various workloads.](image)

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

Perils of Processor-Centric Design

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

- **Overly complex and bloated processor (and accelerators)**
  - To tolerate data access from memory
  - Complex hierarchies and mechanisms
    - Energy inefficient
    - Low performance
    - Complex
Perils of Processor-Centric Design

Most of the system is dedicated to storing and moving data
A memory access consumes $\sim 100-1000X$ the energy of a complex addition.
Energy Waste in Mobile Devices


62.7% of the total system energy is spent on data movement
We Do Not Want to Move Data!

A memory access consumes \(~\text{100-1000X}\) the energy of a complex addition.
We Need A Paradigm Shift To …

- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric
Goal: Processing Inside Memory

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

1. Processing using Memory
2. Processing near Memory
Memory as an Active Accelerator

CPU core
CPU core
CPU core
mini-CPU core
video core
imaging core
GPU (throughput) core
GPU (throughput) core
GPU (throughput) core
GPU (throughput) core
LLC
Memory Controller
Memory Bus
Memory

Memory similar to a “conventional” accelerator

Specialized compute-capability in memory
In-Memory Bulk Bitwise Operations

- We can support **in-DRAM COPY, ZERO, AND, OR, NOT, MAJ**
- At low cost
- Using **analog computation capability** of DRAM
  - Idea: activating multiple rows performs computation

- **30-74X performance and energy improvement**

- New memory technologies enable even more opportunities
Ambit [MICRO’17]

In-DRAM Bulk Bitwise Execution


In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
SIMDRAM Framework


[2-page Extended Abstract]
[Short Talk Slides (pptx) (pdf)]
[Talk Slides (pptx) (pdf)]
[Short Talk Video (5 mins)]
[Full Talk Video (27 mins)]
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)]
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
Opportunity: 3D-Stacked Logic+Memory

Other “True 3D” technologies under development
Tesseract System for Graph Processing

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

Host Processor

Memory-Mapped Accelerator Interface (Noncacheable, Physically Addressed)

Crossbar Network

In-Order Core

LP

PF Buffer

MTP

Message Queue

DRAM Controller

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface (Noncacheable, Physically Addressed)

Logic

Memory

Crossbar Network

In-Order Core

Communications via Remote Function Calls

Message Queue

Host Processor

Memory

Logic

Crossbar Network

In-Order Core

Communications via Remote Function Calls

Message Queue
Tesseract System for Graph Processing

Host Processor
Memory-Mapped Accelerator Interface
Noncacheable, Physically Addressed

Logic

Memory

Crossbar Network

Prefetching

LP
PF Buffer
MTP
Message Queue

DRAM Controller
NI
Evaluated Systems

<table>
<thead>
<tr>
<th>DDR3-OoO</th>
<th>HMC-OoO</th>
<th>HMC-MC</th>
<th>Tesseract</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
</tr>
<tr>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
<td>8 OoO 4GHz</td>
</tr>
</tbody>
</table>

102.4GB/s | 640GB/s | 640GB/s | 8TB/s |

SAFARI Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
Tesseract Graph Processing Performance

>13X Performance Improvement

On five graph processing algorithms

- DDR3-OoO
- HMC-OoO
- HMC-MC
- Tesseract
- Tesseract-LP
- Tesseract-LP-MTP

>13X Performance Improvement

Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
**Tesseract Graph Processing Performance**

Memory Bandwidth Consumption

- DDR3-OoO: 80GB/s
- HMC-OoO: 190GB/s
- HMC-MC: 243GB/s
- Tesseract: 1.3TB/s
- Tesseract-LP: 2.2TB/s
- Tesseract-LP-MTP: 2.9TB/s
Tesseract Graph Processing System Energy

Ahn+, “A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing” ISCA 2015.
More on Tesseract

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

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

1. Functionality of and applications & software for PIM

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

3. System support: coherence & virtual memory

4. Runtime and compilation systems for adaptive scheduling, data mapping, access/sharing control

5. Infrastructures to assess benefits and feasibility

All can be solved with change of mindset
We Need to Revisit the Entire Stack

We can get there step by step
Challenge and Opportunity for Future Computing Architectures with Minimal Data Movement
Challenge and Opportunity for Future

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

Fundamentally High-Performance (Data-Centric) Computing Architectures
Saugata Ghose, Amirali Boroumand, Jeremie S. Kim, Juan Gomez-Luna, and Onur Mutlu,
"Processing-in-Memory: A Workload-Driven Perspective"
[Preliminary arXiv version]

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
System Organization

• UPMEM-based PIM system with 20 UPMEM memory modules of 16 chips each (40 ranks) → 2560 DPUs
More on the UPMEM PIM System

[YouTube Video: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)]

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2ZI9xyIqBxUz7xRPS-wisBN&index=26
Experimental Analysis of the UPMEM PIM Engine

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

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

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

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

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present PrIM (Processing-In-Memory benchmarks), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

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.
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

**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 28nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon1, Suk Han Lee1, Jaehoon Lee1, Sang-Hyuk Kwon1, Je Min Ryu1, Jong-Pil Son1, Seongil O1, Hak-Soo Yu1, Haesuk Lee1, Soo Young Kim1, Youngmin Cho1, Jin Guk Kim1, Jongyoon Choi1, Hyun-Sung Shin1, Jin Kim1, BengSeng Phua1, HyungMin Kim1, Myeong Jun Song1, Ahn Choi1, Daeho Kim1, SooYoung Kim1, Eun-Bong Kim1, David Wang1, ShinGyung Kang1, Yuhwan Ro1, Seungwoo Seo1, JoonHo Song1, Jaryoun Yoon1, Kyomin Sohn1, Nam Sung Kim1

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

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

[Digital RTL design for PCU block]
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](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](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](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/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=25)

[https://www.youtube.com/onurmutlulectures](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=PL5Q2soXY2Zi9xydIgBxUz7xRPS-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=PL5Q2soXY2Zi9xydIgBxUz7xRPS-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=PL5Q2soXY2Zi9xydIgBxUz7xRPS-wisBN&index=41

https://www.youtube.com/onurmutlulectures
A Tutorial on Processing in Memory

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
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
PIM Can Enable New Medical Platforms

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

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

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017
Published: 02 April 2018 Article history ▼

Future of Genome Sequencing & Analysis

MinION from ONT

SmidgION from ONT
Accelerating Genome Analysis: Overview

- 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](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](https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14)

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

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - **Accelerating Genomics** (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=rgjI8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId](https://www.youtube.com/watch?v=rgjI8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId)

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
If You Want More of This...
Multiple Future Options

- Take the **Bachelor’s Seminar in Computer Architecture**
  - Offered **every Fall and Spring**
  - [https://safari.ethz.ch/architecture_seminar](https://safari.ethz.ch/architecture_seminar)

- Take the Master’s-level **Computer Architecture** course
  - Offered **every Fall**
  - [https://safari.ethz.ch/architecture](https://safari.ethz.ch/architecture)

- Do research with me and my **SAFARI research group**
  - Bachelor’s/Master’s theses, semester projects, internships
  - Opportunity to be a part of a vibrant, cutting-edge group
  - [https://safari.ethz.ch/](https://safari.ethz.ch/)
Think BIG, Aim HIGH!

https://safari.ethz.ch
Onur Mutlu’s SAFARI Research Group

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

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

Think BIG, Aim HIGH!

https://safari.ethz.ch
SAFARI Newsletter April 2020 Edition

https://safari.ethz.ch/safari-newsletter-april-2020/

Dear SAFARI friends,

2019 and the first three months of 2020 have been very positive and eventful times for SAFARI.
Dear SAFARI friends,

Happy New Year! We are excited to share our group highlights with you in this second edition of the SAFARI newsletter (You can find the first edition from April 2020 here). 2020 has
Bachelor’s Seminar in Computer Architecture

- Fall 2021
- 2 credit units

- Rigorous seminar on fundamental and cutting-edge topics in computer architecture
- Critical presentation, review, and discussion of major works in computer architecture
  - We will cover many ideas & issues, analyze their tradeoffs, perform critical thinking, discussion and brainstorming

- Participation, presentation, synthesis report
- You can register for the course online
  - https://safari.ethz.ch/architecture_seminar
Fall 2021
8 credit units

Covers many cutting-edge topics in memory systems, interconnects, multiprocessors, heterogeneous systems, PIM, and more...

Many topics we could not cover in this course

Exam, lab assignments, homeworks
You can register for the course online

https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule
Doing Research with Us

- If you are interested in learning more and doing research in Computer Architecture, three suggestions:
  - Email me with your interest (CC: Juan)
  - Take the seminar course or the “Computer Architecture” course
  - Do readings and assignments on your own

- There are many exciting projects and research positions, e.g.:
  - Memory systems
  - Hardware security
  - GPUs, FPGAs, heterogeneous systems, ...
  - New execution paradigms (e.g., in-memory computing)
  - Security-architecture-reliability-energy-performance interactions
  - Architectures for medical/health/genomics
  - A limited list is here: https://safari.ethz.ch/theses/
How To Do Projects with SAFARI?

- Be excited about topics we work on
- Get in touch with me & my students/postdocs
- Browse projects here (not all projects are listed)
  - https://safari.ethz.ch/theses/
- Take our courses and enjoy them 😊
- Read our papers and get in touch with authors
  - https://people.inf.ethz.ch/omutlu/projects.htm
- Email us & apply here: https://safari.ethz.ch/work-with-us/
Food for Thought: Two Quotes

The reasonable man adapts himself to the world; The unreasonable one persists in trying adapt the world to himself. Therefore all progress depends on the unreasonable man.

George Bernard Shaw

Progress is impossible without change, and those who cannot change their minds cannot change anything.

Digital Design & Computer Arch.

Lecture 26b: Epilogue

Prof. Onur Mutlu

ETH Zürich
Spring 2021
4 June 2021
Research in Computer Architecture
Current Research Mission

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

- Hybrid Main Memory
- Heterogeneous Processors and Accelerators
- Persistent Memory/Storage
- Graphics and Vision Processing

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
Current Research Mission & Major Topics

Build fundamentally better architectures

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

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

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

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

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

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

Broad research spanning apps, systems, logic with architecture at the center
Research & Teaching: Some Overview Talks

https://www.youtube.com/onurmutlulectures

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kqiZISOcGFM&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=r7sn41I-H-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_LgzuNdkw&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

- Maurice Wilkes Award Speech (10 minutes)
  - 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)]
Referenced Papers, Talks, Artifacts

- All are available at
  
  [https://people.inf.ethz.ch/omutlu/projects.htm](https://people.inf.ethz.ch/omutlu/projects.htm)

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

  [https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)

  [https://github.com/CMU-SAFARI/](https://github.com/CMU-SAFARI/)
Some Basics of Research
How To Do Research & Advanced Dev.

- We will talk a lot about this in this course

- Learning by example
  - Reading and evaluating strong and seminal papers & designs

- Learning by doing
  - Semester-long research/design projects, masters’ projects, PhD thesis

- Learning by open, critical discussions
  - Paper reading groups, frequent brainstorming and discussions
  - Design sessions
  - Collaborations
What Is The Goal of Research?

- **To generate new insight**
  - that can enable what previously did not exist

- Research is a hunt for insight that can eventually impact the world
Some Basic Advice for Good Research

- Choose great problems to solve: Have great taste
  - Difficult
  - Important
  - High impact

- Read heavily and critically

- Think big (out of the box)
  - Do not restrain yourself to tweaks or constraints of today
  - Yet, think about adoption issues

- Aim high

- Write and present extremely well
Looking here for lost keys
Lost keys here
Looking here
Current Architecture Practice
Enable this point

5-10 years
The Research Formula

$$\text{ROI} = \frac{\text{reward}}{\text{risk} \times \text{effort}}$$
Reward

If you are wildly successful, what difference will it make?

ROI = \frac{\text{reward}}{\text{risk} \times \text{effort}}
Effort

Learn as much as possible with as little work as possible

\[ ROI = \frac{\text{reward}}{\text{risk} \times \text{effort}} \]
Effort

Do the minimum analysis and experimentation necessary to make a point

\[ ROI = \frac{\text{reward}}{\text{risk} \times \text{effort}} \]
Research is a *hunt for insight*

Need to get off the beaten path to find new insights
Recommended Talk

- Bill Dally, Moving the needle: Effective Computer Architecture Research in Academy and Industry
  ISCA 2010 Keynote Talk.

- Acknowledgment: Past few slides are from this talk

What transfers is *insight*

Not academic design

Not performance numbers
“The purpose of computing is insight, not numbers”

Richard Hamming
Some Personal Examples
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/](https://people.inf.ethz.ch/omutlu/)
- [omutlu@gmail.com](mailto:omutlu@gmail.com) (Best way to reach me)
- [https://people.inf.ethz.ch/omutlu/projects.htm](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, robust systems
- Hardware/software cooperation
- Architectures for bioinformatics, health, medicine, intelligent decision making
- ...
Onur Mutlu’s SAFARI Research Group

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

https://safari.ethz.ch/safari-newsletter-april-2020/

38+ Researchers

Think BIG, Aim HIGH!

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

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

https://safari.ethz.ch/safari-newsletter-january-2021/
Principle: Teaching and Research

Teaching drives Research

Research drives Teaching

...
Principle: Insight and Ideas

Focus on Insight

Encourage New Ideas
Focus on learning and scholarship
Principle: Environment of Freedom

Create an environment that values free exploration, openness, collaboration, hard work, creativity
Principle: Learning and Scholarship

The quality of your work defines your impact
Research & Teaching: Some Overview Talks

https://www.youtube.com/onurmutlulectures

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kgiZISOcGFМ=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)]
A memory access consumes $\sim100\text{-}1000\times$ the energy of a complex addition.