227-0085-56L Projekte & Seminare: Intelligent Architectures via Hardware/Software Cooperation

<table>
<thead>
<tr>
<th>Semester</th>
<th>Autumn Semester 2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecturers</td>
<td>O. Mutlu</td>
</tr>
<tr>
<td>Periodicity</td>
<td>every semester recurring course</td>
</tr>
<tr>
<td>Language of instruction</td>
<td>English</td>
</tr>
<tr>
<td>Comment</td>
<td>Only for Electrical Engineering and Information Technology BSc. Course can only be registered for once. A repeatedly registration in a later semester is not chargeable.</td>
</tr>
</tbody>
</table>

Abstract

The category of "Laboratory Courses, Projects, Seminars" includes courses and laboratories in various formats designed to impart practical knowledge and skills. Moreover, these classes encourage independent experimentation and design, allow for explorative learning and teach the methodology of project work.

Objective

Modern general-purpose processors are agnostic to an application’s high-level semantic information. Hence, they employ prediction-based techniques to enable computational and memory optimizations, such as prefetching, cache management policies, memory data placement, instruction scheduling, and many others. As such, the potential of such optimizations is limited due to the limited information the underlying hardware can discover on its own and such optimizations come with large area, power and complexity overheads required by the hardware for prediction purposes. Purely-hardware optimizations cannot achieve their performance potential and waste power, complexity and hardware area, since they are not aware of the application characteristics. On the other hand, purely-software optimizations are fundamentally tied up and limited by the underlying hardware.
The Role of This Course
P&S Hardware/Software Co-design: Contents

- We will introduce the need for hardware/software co-design in current computing systems, in order to achieve high performance and energy efficiency.

- You will get familiar with the current as well as futuristic hardware/software co-operative techniques.

- You will learn how to modify software to match the underlying hardware and vice-versa.

- You will work hands-on: analyzing workloads, creating new hardware/software interfaces, proposing new software and architectural solutions, etc.
Key Takeaways

- This P&S aims at improving your
  - **Knowledge** in Computer Architecture and Hardware/Software Co-Design
  - **Technical skills** in programming and developing architectural simulators
  - **Critical thinking and analysis**
  - **Interaction** with a big group of researchers
  - **Familiarity with key research directions**
  - **Technical presentation** of your project
Key Goal

(Learn how to) design efficient hardware/software co-operative techniques
Prerequisites of the Course

- Digital Design and Computer Architecture (or equivalent course)

- Familiarity with C/C++ programming

- Familiarity with Machine Learning frameworks
  - Only in specific projects

- Interest in
  - computer architectures and computing paradigms
  - discovering why things do or do not work and solving problems
  - making systems efficient and usable
Course Info: Who Are We? (I)

- 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
  - Hardware/software cooperation
  - Architectures for bioinformatics, health, medicine
  - ...

Course Info: Who Are We? (II)

- Konstantinos Kanellopoulos
  PhD Student
  Hardware/Software Interfaces | Hardware Security

- Rahul Bera
  PhD Student
  Memory systems | Prefetching
  Near memory computing

Get to know our research: [https://safari.ethz.ch/safari-group/](https://safari.ethz.ch/safari-group/)
Course Info: Who Are We? (III)

Juan Gómez Luna  
Senior Researcher and Lecturer  
Processing-In-Memory | Heterogeneous computing | Memory Systems | Bioinformatics | Medical imaging

Mohammad Sadrosadati  
Senior Researcher  
Heterogeneous computing | Processing-In-Memory | Memory Systems | Interconnection Networks

Nika Mansourighiasi  
PhD Student  
Processing-In-Memory | Emerging Memory & Processing Technologies

Get to know them and their research: [https://safari.ethz.ch/safari-group/](https://safari.ethz.ch/safari-group/)
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 December 2021 Edition

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

Think Big, Aim High

ETHzürich
SAFARI Live Seminars (1)

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, Schlüter 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
Gennady Pekhimenko, 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
Christina Giannoula, National Technical University of Athens
Efficient Synchronization: Support for Near-Data-Processing Architectures

SAFARI Live Seminars in Computer Architecture
Ataberk Olgun, TOBB & ETH Zurich
QUAC-FTRNG: High-Throughput True Random Number Generation Using Quadruple Base Activation in Conventional DRAM Chips

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

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

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 DDR4 typically each has an independent clock

SAFARI Research Group

https://safari.ethz.ch/safari-seminar-series/
Current Research Focus Areas

**Research Focus:** Computer architecture, HW/SW, bioinformatics

- Memory and storage (DRAM, flash, emerging), interconnects
- Heterogeneous & parallel systems, GPUs, systems for data analytics
- System/architecture interaction, new execution models, new interfaces
- Energy efficiency, fault tolerance, hardware security, performance
- Genome sequence analysis & assembly algorithms and architectures
- Biologically inspired systems & system design for bio/medicine

**Broad research spanning apps, systems, logic with architecture at the center**
Course Requirements and Expectations

- Study the learning materials

- Each student will carry out a hands-on project
  - Build, implement, code, and design with close engagement from the supervisors

- Participation
  - Ask questions, contribute thoughts/ideas
  - Read relevant papers

- Presentation & GitHub repository

We will help the projects with good progress to get published in good venues!
Your Responsibilities

- Several Lectures:
  - Monday 12:30-1:30 PM

- Working on your project for ~4-5 hours per week

- Meeting your mentors weekly is required to be able to track progress efficiently
Course Website

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

- Useful information for the course

- Check your email and Moodle frequently for announcements

- We will also have Moodle for Q&A, announcements, ..
Next Meetings

- We will give you a chance to select a project

- Then, we will have **1-1 meetings** to match your interests, skills, and background with a suitable project

- It is important that you **study the learning materials before** our next meeting!

- We will **assign the projects next week**
Next Meetings

- Presentation of research papers
- Background on computer architecture topics (e.g. Virtual Memory)
- Tutorial on microarchitectural simulator
Hardware/Software Co-design
**Axiom**

To achieve the highest energy efficiency and performance:

**we must take the expanded view**

of computer architecture

<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>

**Co-design across the hierarchy:**

Algorithms to devices

**Specialize as much as possible within the design goals**
Hardware/Software Co-design

- Design application-specific standalone accelerators
  + High performance/energy benefits
  - Low Flexibility

- Extend general-purpose processors with application- or task-specific hardware components
  + High performance/energy benefits
  - Abrupt changes in the processor

- Revisiting the existing hardware/software interfaces to enhance performance, security, portability
Apple M1 Max Processor

- **ProRes**: encode and decode
- **Thunderbolt 4**
- **57 billion Transistors**
- **16-core Neural Engine**: 11 trillion operations per second
- **Industry-leading performance per watt**
- **5 nm process**
- **Secure Enclave**
- **Support for four external displays**
- **Up to 64 GB Unified memory**
- **10-core CPU**
- **32-core GPU**
- **400 GB/s Memory bandwidth**
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**

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
SAFARI Live Seminar

https://www.youtube.com/watch?v=x2-qB0J7KHz

Thinking Outside the Die: Architecting the ML Accelerator of the Future

Sean Lie
Co-founder & Chief HW Architect, Cerebras
Cerebras’s Wafer Scale Engine (2019)

- The largest ML accelerator chip (2019)
- 400,000 cores

Cerebras WSE
1.2 Trillion transistors
46,225 mm²

Largest GPU
21.1 Billion transistors
815 mm²
NVIDIA TITAN V

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.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/](https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/)
Google TensorFlow + TPU
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 (2019)

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.
System Architecture Design Today

- Human-driven
  - Humans design the policies (how to do things)

- Many (too) simple, short-sighted policies all over the system

- No automatic data-driven policy learning

- (Almost) no learning: cannot take lessons from past actions

Can we design fundamentally intelligent architectures?
An Intelligent Architecture

- Data-driven
  - Machine learns the “best” policies (how to do things)

- Sophisticated, workload-driven, changing, far-sighted policies

- Automatic data-driven policy learning

- All controllers are intelligent data-driven agents

How do we start?
Self-Optimizing DRAM Controllers

- Engin Ipek, Onur Mutlu, José F. Martínez, and Rich Caruana,
  "Self Optimizing Memory Controllers: A Reinforcement Learning Approach"

Self-Optimizing Memory Controllers: A Reinforcement Learning Approach

Engin İpek\textsuperscript{1,2} Onur Mutlu\textsuperscript{2} José F. Martínez\textsuperscript{1} Rich Caruana\textsuperscript{1}

\textsuperscript{1}Cornell University, Ithaca, NY 14850 USA
\textsuperscript{2}Microsoft Research, Redmond, WA 98052 USA
Pythia: RL-Based Prefetching

- Rahul Bera, Konstantinos Kanellopoulos, Anant Nori, Taha Shahroodi, Sreenivas Subramoney, and Onur Mutlu,

"Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning"
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 (20 minutes)]
[Lightning Talk Video (1.5 minutes)]
[Pythia Source Code (Officially Artifact Evaluated with All Badges)]

Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera¹ Konstantinos Kanellopoulos¹ Anant V. Nori² Taha Shahroodi³,¹
Sreenivas Subramoney² Onur Mutlu¹

¹ETH Zürich  ²Processor Architecture Research Labs, Intel Labs  ³TU Delft
“Hermes: Accelerating Long-Latency Load Requests via Perceptron-Based Off-Chip Load Prediction”

Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago USA, October 2022.
Data-Aware Architectures

- A data-aware architecture understands what it can do with and to each piece of data.

- It makes use of different properties of data to improve performance, efficiency and other metrics:
  - Compressibility
  - Approximability
  - Locality
  - Sparsity
  - Criticality for Computation
  - Access Semantics
  - ...

One Problem: Limited Expressiveness

Higher-level information is not visible to HW

- Data Structures
- Code Optimizations
- Access Patterns
  - A → B → C
  - D → E → F

Software

Hardware

Instructions
Memory Addresses
A Solution: More Expressive Interfaces
Expressive (Memory) Interfaces

  [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 Ebrahimi‡ Nastaran Hajinazar† Phillip B. Gibbons† Onur Mutlu§†

†Carnegie Mellon University ‡University of Toronto §ETH Zürich
‡Simon Fraser University
Expressive (Memory) Interfaces for GPUs

  [Slides (pptx) (pdf)] [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video]
Another Example: EDEN for DNNs

- Deep Neural Network evaluation is very DRAM-intensive (especially for large networks)

1. Some data and layers in DNNs are very tolerant to errors

2. Reduce DRAM latency and voltage on such data and layers

3. While still achieving a user-specified DNN accuracy target by making training DRAM-error-aware

Data-aware management of DRAM latency and voltage for Deep Neural Network Inference
Example DNN Data Type to DRAM Mapping

Mapping example of ResNet-50:

Map more error-tolerant DNN layers to DRAM partitions with lower voltage/latency

4 DRAM partitions with different error rates
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.

[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
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)]

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

Konstantinos Kanellopoulos¹, Nandita Vijaykumar²,¹, Christina Giannoula¹,³, Roknoddin Azizi¹, Skanda Koppula¹, Nika Mansouri Ghiasi¹, Taha Shahroodi¹, Juan Gomez Luna¹, Onur Mutlu¹,²

¹ETH Zürich  ²Carnegie Mellon University  ³National Technical University of Athens
The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

Nastaran Hajinazar*† Pratyush Patel*‡ Minesh Patel* Konstantinos Kanellopoulos* Saugata Ghose† Rachata Ausavarungrunrun© 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
Thank you 😊
Questions?