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
- ISA and Microarchitecture
- Key Execution Paradigms
- The Memory System
- Built up to System Software Mechanisms

- Takeaway 1: All we covered is real and used in real systems ➔ 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
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
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 my ETH inaugural lecture
  - https://www.youtube.com/watch?v=kgiZlSOcGFM
Recall from Lecture 2: Some Takeaways

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

- Many challenging and exciting problems in platform design
  - That noone has tackled (or thought about) before
  - That can have huge impact on the world’s future

- Driven by huge hunger for data and its analysis (“Big Data”), new applications, 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
  - Three walls: Energy, reliability, complexity
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 innovations require computer architecture
Many Other Ideas and Topics

- Prefetching
- Runahead execution: Efficient latency tolerance
- Emerging memory technologies
- Solid state disks and storage, I/O
- Interconnection networks
- Many issues in multiprocessing and multithreading
- Heterogeneous multiprocessors
- QoS and predictable performance
- Reliable architectures
- Secure architectures
- Programmability, portability
- Better HW/SW interfaces
- Reconfigurable computing
- Heterogeneous CPU-GPU-FPGA-HWAcc architectures
- Specialized and domain-specific architectures – genomics, medicine, health, AI/ML
- Unconventional architectures – nature-inspired, quantum, molecular, ...
- ...
Oculus, New York City

Prefetching
Prefetching

- **Idea:** Fetch the data before it is needed (i.e. pre-fetch) by the program

- **Why?**
  - Memory latency is high. If we can prefetch accurately and early enough we can reduce/eliminate that latency.
  - Can eliminate compulsory cache misses
  - Can it eliminate all cache misses? Capacity, conflict?

- Involves predicting which address will be needed in the future
  - Works if programs have predictable miss address patterns
Prefetching: The Four Questions

- What
  - What addresses to prefetch

- When
  - When to initiate a prefetch request

- Where
  - Where to place the prefetched data

- How
  - Software, hardware, execution-based, cooperative, ...
Outline of Prefetching Issues…

- Why prefetch? Why could/does it work?
- The four questions
  - What (to prefetch), when, where, how
- Software prefetching
- Hardware prefetching algorithms
- Execution-based prefetching
- Prefetching performance
  - Coverage, accuracy, timeliness
  - Bandwidth consumption, cache pollution
- Prefetcher throttling
- Issues in multi-core
- Prefetching in new execution paradigms
Runahead Execution
Perfect Caches:
Load 1 Hit
Load 2 Hit
Compute
Compute

Small OoO Instruction Window:
Load 1 Miss
Load 2 Miss
Compute
Stall
Compute
Stall
Miss 1
Miss 2

Runahead:
Load 1 Miss
Load 2 Miss
Load 1 Hit
Load 2 Hit
Compute
Runahead
Compute
Saved Cycles
Miss 1
Miss 2
Effect of Runahead in Sun ROCK

- Shailender Chaudhry talk, Aug 2008.
More on Runahead Execution

More on Runahead Execution (Short)

Runahead Readings

- Required

- Recommended
If You Want More of This...
Multiple Future Options

- Take the Bachelor’s Seminar in Comp Arch
  - Offered every Fall and Spring

- Take the Computer Architecture course
  - Offered every Fall

- Read the referenced papers methodically and critically
- Do research with us
  - [https://safari.ethz.ch/](https://safari.ethz.ch/)
SAFARI Research Group

Think BIG, Aim HIGH!

https://safari.ethz.ch
Bachelor’s Seminar in Computer Architecture

- Fall 2019
- 2 credit units

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

- Participation, presentation, synthesis report
- You can register for the course online
(Next) Computer Architecture Course

- Fall 2019
- 8 credit units

- **Introduces the basic components of a modern computing system (processors, memory, interconnects, storage).**
  - Covers the fundamental concepts of the different parts of modern computing systems.
  - Covers the latest trends by exploring the recent research in Industry and Academia.

- 2 exams (midterm & final), lab assignments, homeworks
- You can register for the course online
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, Mohammed, Lois, Hasan)
  - Take the seminar course and the “Computer Architecture” course
  - Do readings and assignments on your own

- There are many exciting projects and research positions available, spanning:
  - 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
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.
Design of Digital Circuits
Lecture 26: Epilogue

Prof. Onur Mutlu
ETH Zurich
Spring 2019
31 May 2019
Research & Teaching: Some Overview Talks

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

- Enabling In-Memory Computation
  - https://www.youtube.com/watch?v=oHqsNbxgdzM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=7

- Accelerating Genome Analysis
  - https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=9

- Rethinking Memory System Design
  - https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=3
Research in Computer Architecture
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 innovations require computer architecture
Example: Why In-Memory Computation Today?

- **Push from Technology**

- **Pull from Systems and Applications**
  - Data access is a major system and application bottleneck
  - Systems are energy limited
  - Data movement much more energy-hungry than computation
Current Research Focus Areas

Research Focus: Computer architecture, HW/SW, security, 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
Four Key Current Directions

- Fundamentally *Secure/Reliable/Safe* Architectures
- Fundamentally *Energy-Efficient* Architectures
  - *Memory-centric* (Data-centric) Architectures
- Fundamentally *Low-Latency* Architectures
- Architectures for *Genomics, Medicine, Health*
Research Across the Stack

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
Many questions ... How do we design the:
- compute-capable memory & controllers?
- processor chip?
- software and hardware interfaces?
- system software and languages?
- algorithms?
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"

to also appear in Proceedings of the *16th Asia Pacific Bioinformatics Conference (APBC)*, Yokohama, Japan, January 2018.
arxiv.org Version (pdf)

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 Alkan*⁴, and Onur Mutlu*⁶,¹

SAFARI
Nanopore Sequencing Technology and Tools: Computational Analysis of the Current State, Bottlenecks, and Future Directions

Damla Senol Cali\textsuperscript{1,*}, Jeremie Kim\textsuperscript{1,3}, Saugata Ghose\textsuperscript{1}, Can Alkan\textsuperscript{2,*} and Onur Mutlu\textsuperscript{3,1,*}

\textsuperscript{1}Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
\textsuperscript{2}Department of Computer Engineering, Bilkent University, Bilkent, Ankara, Turkey
\textsuperscript{3}Department of Computer Science, Systems Group, ETH Zürich, Zürich, Switzerland
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
Aim Here

5-10 years
The Research Formula

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

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


- 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 CS, since September 2015 (officially May 2016)
  - 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 (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
  - New computation, communication, storage paradigms
  - ...

- [Image of Onur Mutlu]
SAFARI Group Members @ ETH Zurich

4 Post-doctoral Researchers
8 PhD Students + 4 at CMU
5 Interns
15 Master’s and Bachelor’s Researchers
Teaching: Accelerated Memory Course (~6.5 hours)

- **ACACES 2018**
  - Memory Systems and Memory-Centric Computing Systems
  - Taught by Onur Mutlu July 9-13, 2018
  - ~6.5 hours of lectures

- **Website for the Course including Videos, Slides, Papers**
  - [https://www.youtube.com/playlist?list=PL5Q2soXY2Zi-HXxomthrpDpMJm05P6J9x](https://www.youtube.com/playlist?list=PL5Q2soXY2Zi-HXxomthrpDpMJm05P6J9x)

- **All Papers are at:**
  - [https://people.inf.ethz.ch/omutlu/projects.htm](https://people.inf.ethz.ch/omutlu/projects.htm)
  - Final lecture notes and readings (for all topics)
Teaching: Online Courses and Lectures

- **Parallel Computer Architecture Course Materials** (Lecture Videos)

- [https://people.inf.ethz.ch/omutlu/teaching.html](https://people.inf.ethz.ch/omutlu/teaching.html)
- [https://www.youtube.com/channel/UCIwQ8uOeRFgOEvBLYc3kc3g](https://www.youtube.com/channel/UCIwQ8uOeRFgOEvBLYc3kc3g)
- [https://www.youtube.com/user/cmu18447](https://www.youtube.com/user/cmu18447)
Research & Teaching: Some Overview Talks

https://www.youtube.com/watch?v=kgiZlSOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJhnV2YFBjl

- Future Computing Architectures
  - https://www.youtube.com/watch?v=kgiZlSOcGFM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJhnV2YFBjl&index=1

- Enabling In-Memory Computation
  - https://www.youtube.com/watch?v=oHqsNbxgdzM&list=PL5Q2soXY2Zi8D_5MGV6EnXEJhnV2YFBjl&index=7

- Accelerating Genome Analysis
  - https://www.youtube.com/watch?v=hPnSmfwu2-A&list=PL5Q2soXY2Zi8D_5MGV6EnXEJhnV2YFBjl&index=9

- Rethinking Memory System Design
  - https://www.youtube.com/watch?v=F7xZLNMIY1E&list=PL5Q2soXY2Zi8D_5MGV6EnXEJhnV2YFBjl&index=3