Introduction to Computer Architecture
About the course and logistics

Introduction to Computer Architecture
David Black-Schaffer

Contents

• About the course

• What is computer architecture?
  (And why should you care?)

• Course logistics
  – Feedback from last year
  – Changes for this year
  – Grading and assignments

About the course
# About this course

- **Introductory course to computer architecture**
  (Not for IT/DV students; they take it in the Spring)
- **Requirement** basic programming experience in an imperative language: C/Java
- **Same course for distance and local**
  - 7.5hp (200h) for distance/5.0hp (133h) for local
  - Local students have in-class practice exercises
  - Distance students have at-home practice exercises

---

## Contents

- **How a computer is built**
  - Logic → circuits → datapath
- **How a computer is controlled**
  - Instructions → microarchitecture → ISA → assembly
- **Performance implications**

- **Contents**
  - MIPS assembly
  - Logic design
  - Processor design
  - I/O and Memory
  - Caches
  - Virtual memory
  - Parallelism
  - Roofline

---

# Questions you should be asking…

**Why should I care about this stuff?**
I'm not designing computers...

**Architecture is essential for performance.**

**Why are we using MIPS? I don’t own a MIPS computer...**

**MIPS is simple. x86 is not.**
Learning Objectives

- Understand the functionality and operation of the basic elements of a computer system including processor, memory and input/output
- Reason about first-order performance
- Understand the hardware/software interface
- Understand and be able to write programs in assembly language

What is computer architecture?

(Why should you care?)
ARM introduced the “big.LITTLE” processor...

Backup: Who knows what ARM is?

What is this “big.LITTLE” thing?

- Big processor cores for high performance
- Little processor cores for low power
Why is ARM doing this?

Answer: So your cell phone battery lasts longer.

Power Efficiency (calculations/energy)

- **BIG**: Fast/high-power
- **LITTLE**: Slow/low-power

The details...

**LITTLE**
- Simple (few functional units)
- Slower clock (short pipeline)

**BIG**
- Complex
- More functional units
- Out of order execution
- Long pipeline
- Faster clock
- Bigger branch penalty

This is Computer Architecture!

How can they do this?

- 12/2005
- 2012:
  - Intel Phi 64 Pentium CPUs
- The amount of logic in a processor keeps growing as transistors keep shrinking.
Why should they do this?

- The Power Wall has expensive to cool anything hotter than 100°C.
- Result: Multicore needed to meet transistors’ temperature constraints.

What are others doing?

- GPUs: Lots of very small cores
- GPUs (small cores) + CPUs (big cores)

It's all about power efficiency:

The right architecture for the right job is more efficient.

This is Computer Architecture

- ARM LITTLE
- ARM big
- Nvidia Kepler
- AMD Fusion

- Understanding performance and efficiency
- Design tradeoffs for executing code
- Building the hardware
- Making it programmable
So...why should you care?

- Computers are evolving very fast
- Need to understand how they work to understand why they are changing
- Architecture is critical for performance and efficiency
- Not just about designing hardware:
  - How does big.LITTLE affect your software?
  - How easy is it to use a CPU/GPU for your software?
- Plus: it’s fun!

Feedback from Last year
(And what we're going to do about it)
**Student course evaluations**

Conclusion: Students loved the course!

- General feeling about the course: Poor (1), Good (5), Excellent (7)
- Lecture quality: Poor (1), Good (5), Excellent (7)
- How motivating was the course? Poor (1), Good (5), Excellent (7)
- Are you likely to take follow up courses? Poor (1), Good (5), Excellent (7)

**Exam performance**

- 40% of students scored < 60%
- Time did not help
  - Half the students failed the exam.

**Analysis**

- Students loved the course...
  ...but half failed.

- What’s going on?
  Lectures are a too passive way to learn.

- How can we fix it?
  Focus on active learning by increasing participation.
Course Logistics

Increasing active learning

- Change the course focus away from lectures:
  - Focus on solving problems
  - Focus on more student interaction

- How to do this?
  - "Flip" the classroom: lectures at home, problems in class
  - Use frequent self-assessment: identify unclear topics
  - Lots of student feedback

Course structure

- Lectures
  - Online
  - Required (yes/no)
  - Self-assessment quizzes

- Readings
  - Required (if you want to pass the exam)

- Practice Problems
  - In class (at home for distance course)
  - Required (pass/fail)

- Labs
  - In pairs
  - Required (graded)
Grading

• 60% written final exam
• 40% participation
  – Average of 5 lab grades (<3.0 = U)
  – U if lectures and quizzes not completed on time*
  – U if insufficient participation in practice sessions*
  – U if labs not turned in on time*

*You get 8 late days for any of these. Tell us if you want to use them.
*You get 8 late days for any of these. Tell us if you want to use them.

See the syllabus on the website for details.

How to learn the most in this class

• Do all of the following:
  – Read the book (on time)
  – Watch the lectures (on time)
  – Do the practice problems (in class or on time)
  – Do the labs (on time)

How to pass this class with the least effort

• Do all of the following:
  – Read the book (on time)
  – Watch the lectures (on time)
  – Do the practice problems (in class or on time)
  – Do the labs (on time)
We want you to learn

- Do all of the following:
  - Read the book (on time)
  - Watch the lectures (on time)
  - Do the practice problems (in class or on time)
  - Do the labs (on time)

All of these activities are designed to help you learn

- The exam will cover the book reading and the lectures: **neither one covers all the material**
- The practice problems and labs are designed to help you learn the material: **they make the exam easier**

How to get help

- Talk with the TAs or instructors during office hours
- Ask questions in-class or online
  - Course discussions or at the end of lectures
  - Answered by other students and the staff
  - Staff will answer emails by posted office hours at the latest

- Review on your own
  - Lectures online
  - Book isn’t going anywhere

- Work with other students
  - But don’t cheat (see the syllabus for details)
  - Lots of collaborative work in this course
  - We encourage you to work together

Schedule

See the course website for dates

- 5 Labs
- 14 Lectures
  - Must be watched before the class meeting
  - Self-assessment quizzes not graded, but used for practice
- 14 Readings
  - Will be on the exam
  - It’s a good book: read it
- 13 Practice Problems
  - Required (in class or online for distance)
  - Work with other students and get lots of help from us

- Labs and lectures have due dates. Don’t miss them.
- You have 8 late days for flexibility. Do use them.
Schedule

Lecture
• 14 lectures
• Watch the lectures online before
• Class sessions are for practice problems
• (don’t do the reading)

Labs
• 5 Labs
• 2 Labs each
• Lab 3 is 2 weeks
• Start on Monday, due at midnight on Sunday
• Submit on the website
• Do the labs in pairs

Tutorials
• 2 Tutorials for the labs
• No due dates, but watch them before starting the labs

Course website

• We will be using Canvas for the course
• You can see what’s due with the Assignments link
• You can watch lectures with the Modules link
• Use the Discussions to ask questions to the whole class and teaching staff

Online lectures

• Lectures are divided up into short (<10min) segments
• In-between segments you get a self-evaluation quiz (ungraded)
Late days

- You have 8 late days
- You can use them for any assignment, but you need to tell us
- Remember that using a late day does not postpone later assignments

For the next class meeting

- Make sure you register on Canvas
- If you don’t get an email (sent via studentportalen) email me immediately!
- (Optional) do the self-assessment quizzes for today’s lecture (lecture 1)
- Watch lecture 2
- Do the readings for lectures 1 (today) and 2 (next time)