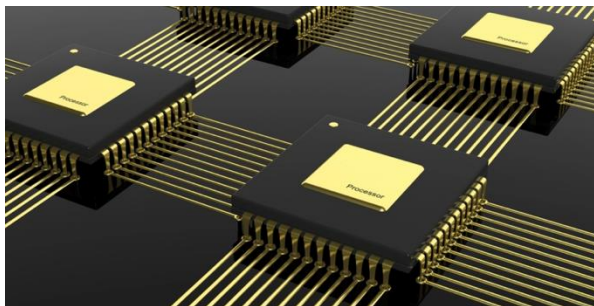# Design and Architecture of Computer Systems

Introduction and Logistics

CS 161

Instructor: Elaheh Sadredini

elaheh@cs.ucr.edu

University of California, Riverside

About Instructor

- Instructor: Elaheh Sadredini
  - Assistant Professor in Computer Science Department
  - Website: https://www.cs.ucr.edu/~elaheh/
  - Email: elaheh@cs.ucr.edu

  - PhD from the University of Virginia in 2019

  - Research Interests
    - Hardware acceleration for big-data applications (NLP, bioinformatics, AI, etc.)
    - Energy efficient cryptographic computation at the edge
    - Secure computing / privacy-preserving computing
    - Algorithm-architecture co-design

Two Undergraduate research position available! And It is paid ☺

Teaching Assistant ------ Sahar Ghoflsaz

- **Sahar Ghoflsaz**
- Ph.D. student in the CS Department
- Research area: secure computing, computer architecture
- Email: sghof001@ucr.edu
- Office hours: TBA (on the Calendar section of the course website)

Course Grader(s)

- **Contact graders and CC me for grading related issues within one week of when the grades are returned**
- **Find grader contact on the course website**

Course website

- https://www.cs.ucr.edu/~elaheh/teaching/cs161.html

CS 161 Design and Architecture of Computer Systems

Instructor: Elaheh Sadredini
Email: elaheh@cs.ucr.edu
Instructor office hours: By appointment

TA: Sahar Ghoflsaz, **Email:** sghof001@ucr.edu
TA office hours: See course calendar below

Graders:
- Eric Via, **Email:** eric.via@email.ucr.edu
- Marlon Lopez, **Email:** marlon.lopez@email.ucr.edu

Discussion sessions: See course calendar below

Location of class and discussion sessions can be found here.

Discussion Forum: Link can be found on Canvas

Course Calendar

No Class Days

- Tues Oct 14 (attending conference)
- Tues Nov 11 (Veterans day)
- Thu Nov 27 (Thanksgiving day)
- Total 18 Session

Fall 2025

- Home
- **Syllabus**
- Announcements
- Modules
- Assignments
- People
- Collaborations
- Item Banks
- Yuja Media/Storage
- LockDown Browser

Course Syllabus

Please see the course webpage for the course syllabus and other information:

https://www.cs.ucr.edu/~elaheh/teaching/cs161.html

Zoom link:

https://ucr.zoom.us/j/92323736551?jst=2

TA Zoom link:

https://ucr.zoom.us/j/3499443474

Where to find Course Information

Home

Syllabus

Announcements 🚫

Modules

Assignments

People

Collaborations

Item Banks

Yuja Media/Storage

LockDown Browser

Course Analytics

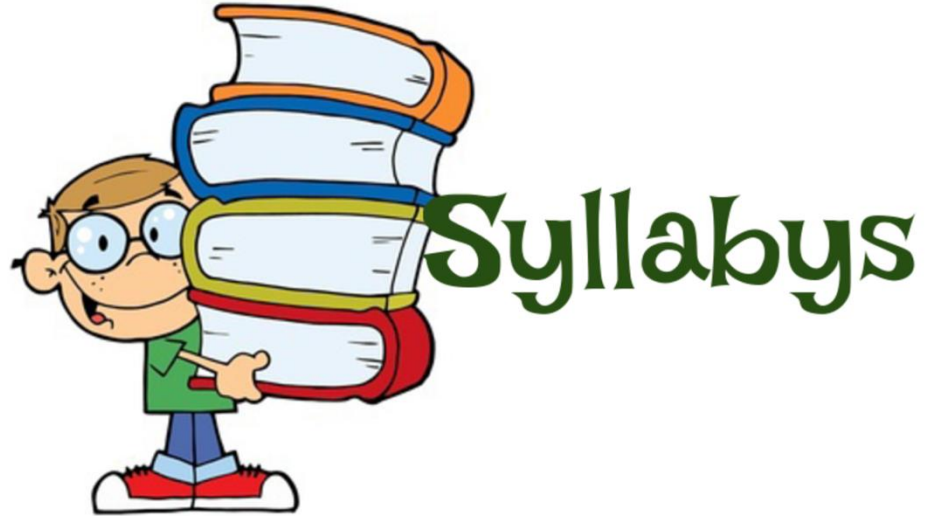Slack

Grades 🚫

⠿ ▸ **Logistics and Introduction**

⠿ ▸ **ISA 1**

⠿ ▸ **ISA 2**

⠿ ▸ **Arithmetic & Logic**

Contents

- **What is computer architecture?**

- **Why computer architecture is important**

- **Why should you care?**

- **Logistics, and requirements**
  - What will you learn?
  - What will I expect?
  - Policies
  - Resources

What is Computer Architecture?

Application

Gap too large to bridge in one step

Physics
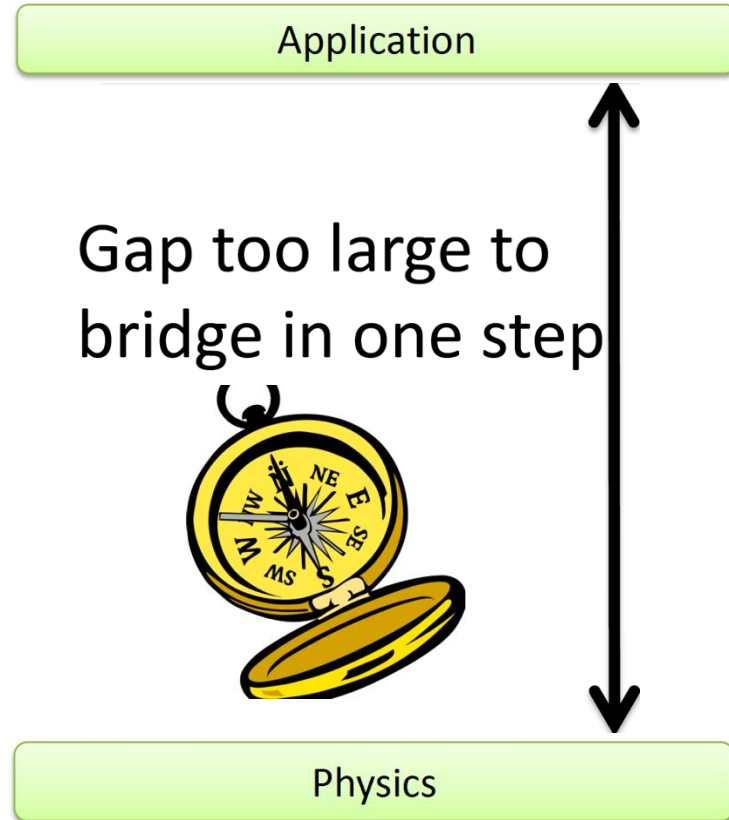
In its broadest definition, computer architecture is the design of the abstraction/implementation layers that allow us to execute information processing applications efficiently using manufacturing technologies

Abstraction in Modern Computing Systems

Application

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Microarchitecture

Design and Architecture of Computer Systems (CS 161)

Register-Transfer Level

CS 161L (Lab)

Gates

Circuits

Devices

Physics
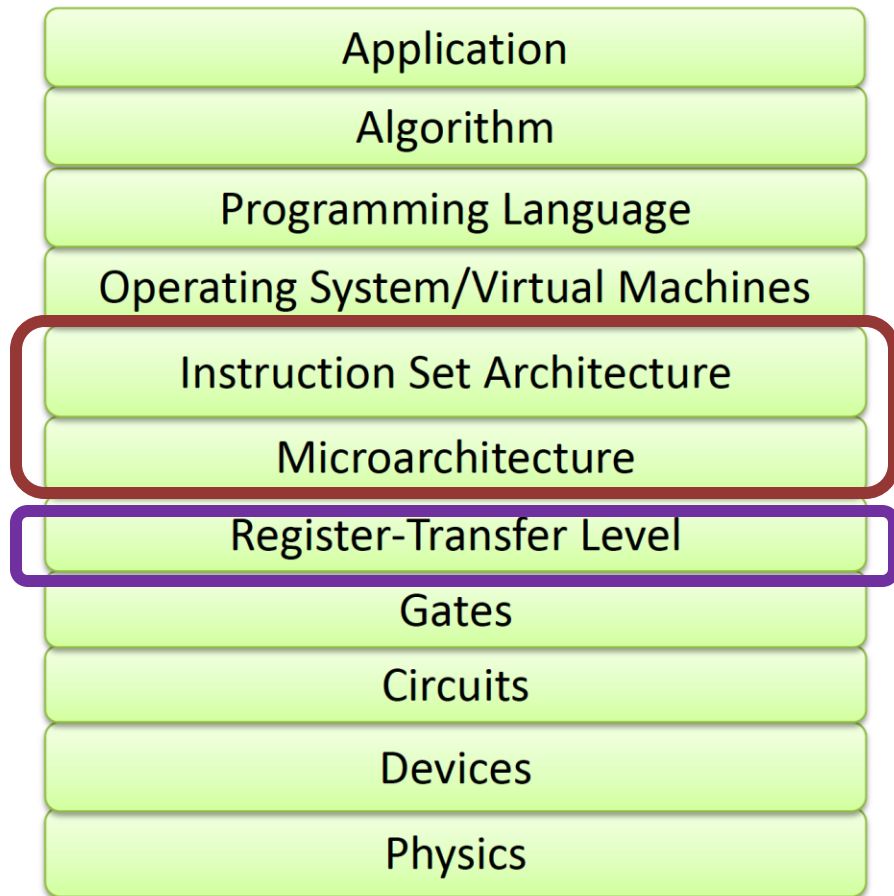
Why Study Computer Architecture and Hardware

Decline of
Moore's Law

Proliferation of
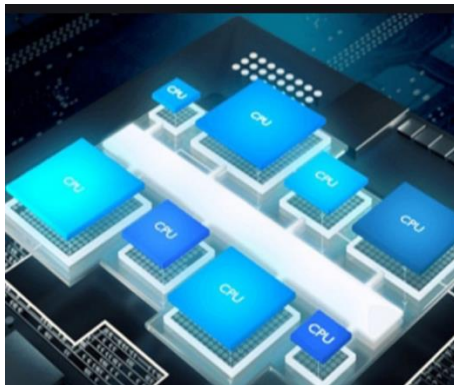Multi-core processors

Emergence of new
platforms

- Hardware knowledge helps programmer write better code
- This is a first step for chip/OS/compiler developers

Computing Trend

A: Can't increase power Need to improve power efficiency. Great way to use all those "free" transistors.

Transistors (thousands)
Relative performance
Clock Speed (MHz)
Power (W)
Cores per Chip

The Power Wall
Too expensive to cool anything hotter than this.

Result: Multicore
Need to sell those extra transistors somehow...
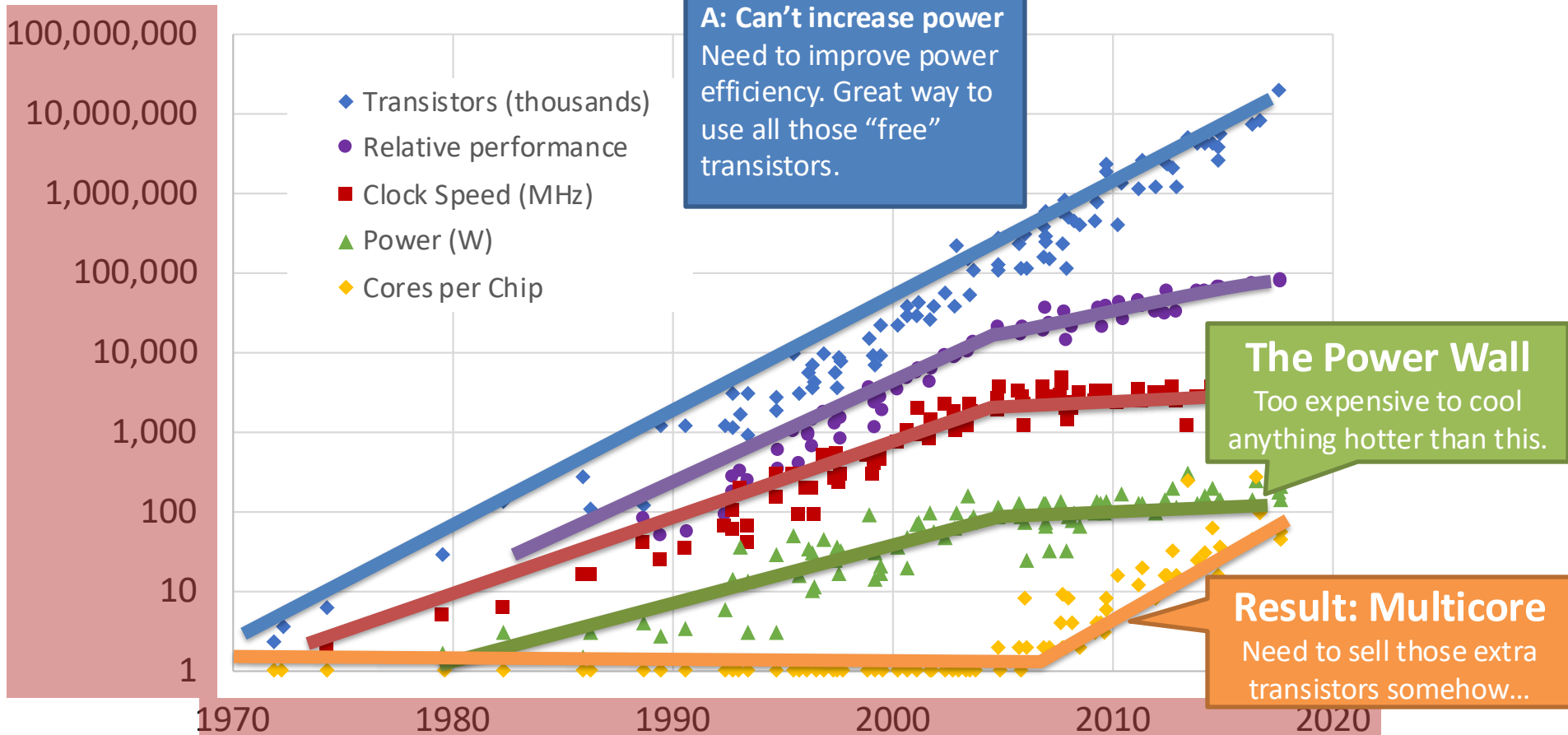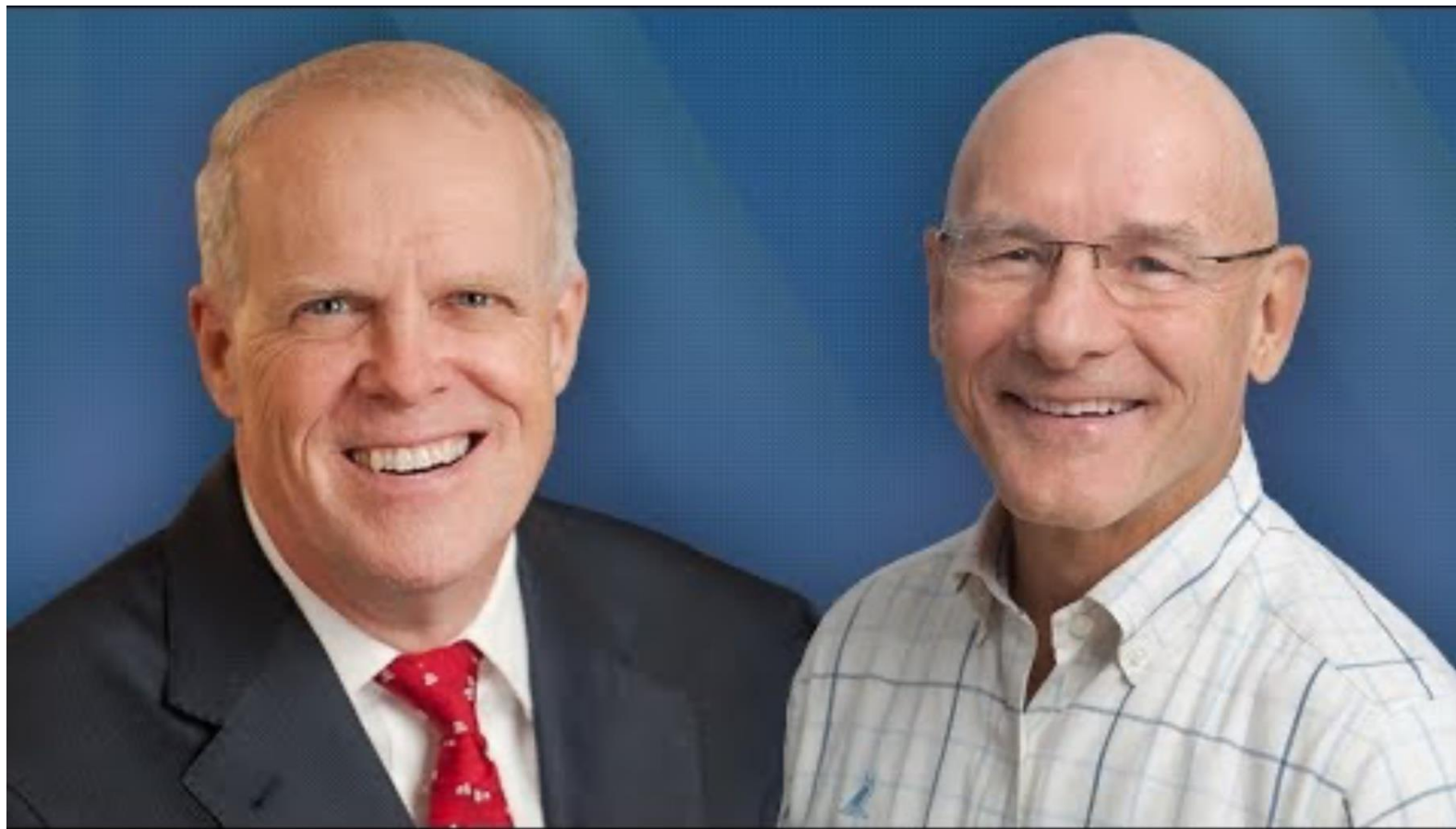
So…why should **you** care?

- Computers are **evolving very fast**

- Need to understand how they work to understand why they are changing

- Architecture is critical for **performance and efficiency**

Hard problems + Smart people + Lots of money

- Plus: it's fun!

Critical Layer: Instruction Set Architecture (ISA)

ISA specifies:
- Data handling
- Arithmetic and logic operations
- Memory operations
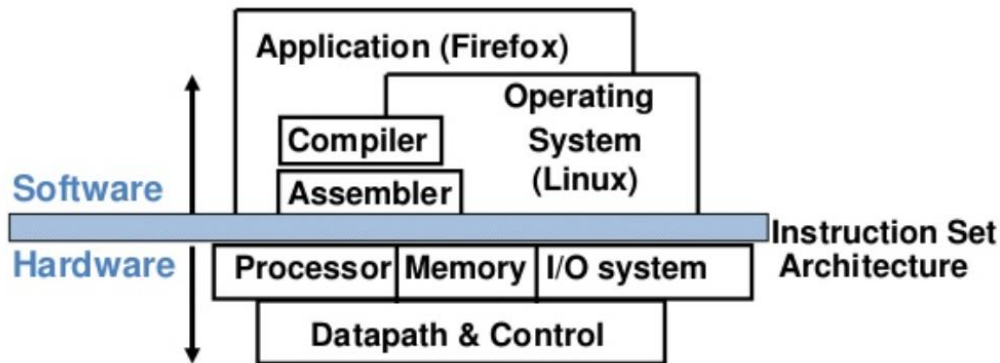- Control flow
- …

- Some well-known ISAs
  - x86 (Intel's and AMD's processors)
  - ARM (smartphones. Laptops, tablets, embedded systems)
  - PowerPC (IBM processors)
  - MIPS --> This is what we learn in this course

- Many different implementations can implement the s
  - Example: Pentium II, Pentium III, Pentium 4, Core i5, Core

Why are we using MIPS? I don't own a MIPS computer…

MIPS is simple. x86 is not.

MIPS (Microprocessor without Interlocking Pipeline Stages)

Silicon Graphics MIPS workstation

Used to be the main high-performance processor for graphics. Today it's used in printers.

Jurassic Park, 1993

What Do You Learn in This Course?

- **How a MIPS computer is controlled at the lowest level**
  - Assembly instructions and the instruction set architecture (ISA)

- **How a MIPS computer is built at the circuit level**
  - Logic design and processor datapath

- **How to get performance from a computer**
  - Pipelining, hazards, branch predictors, caches, and parallelism

- **How to make a practical computer**
  - Virtual memory, digital arithmetic, logic design, input/output

What Do You Learn in This Course?

- **How a MIPS computer is controlled at the lowest level**
  - Assembly instructions and the ISA

- **How a MIPS computer is built at the circuit level**
  - Logic design and processor datapath

- **How to get performance from a computer**
  - Pipelining, hazards, branch predictors, caches, and

- **How to make a practical computer**
  - Virtual memory, digital arithmetic, logic design, inp

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

Architecture is essential for performance.

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**

Where to Get Up-to-date Course Info?

- Website: https://www.cs.ucr.edu/~elaheh/teaching/cs161.html
  – Syllabus and logistic information
  – Tentative schedule
  – Policies
  – …

- Canvas:
  – Announcements (activate real-time notification)
  – Assignments, exams, solutions
  – Optional readings
  – Slides

- Discussion in Slack

- Attendance?

How Will You Be Evaluated?

- **30% homework assignments**
  - HW0: 2 point
  - HW1-HW8, each 4 points
  - One can be skipped or 1 minimum score will be dropped
- **40% lab assignments**
- **20% Mini videos to teach a concept in Computer Architecture**
- **10% in-class polling question**
  - It is graded based on participation not correctness.
  - You need to answer 90% of the question to receive full credit.

- **Total 100%**

You are responsible for having read the syllabus for details.

HW Pipeline

Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11
HW1 out	HW2 out	HW3 out	HW4 out	HW5 out	HW6 out	HW7 out	HW8 out		
	HW1 due	HW2 due	HW3 due	HW4 due	HW5 due	HW6 due	HW7 due	HW8 due	
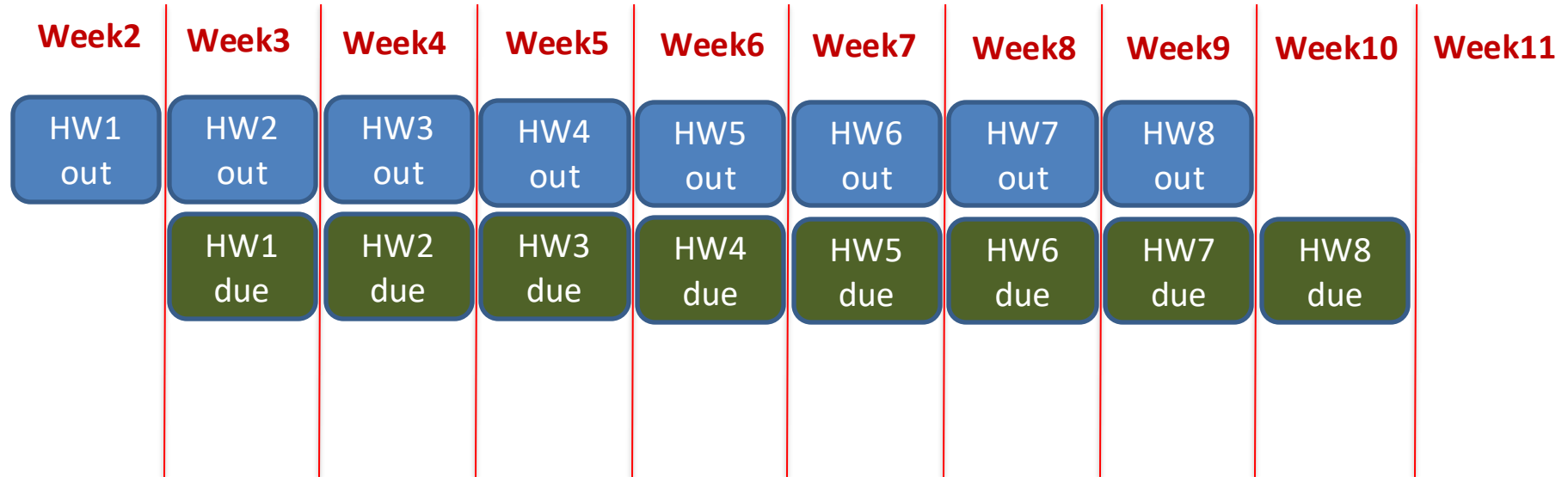
HW due dates will be released soon, so you can mark your calendar!

Tentative Schedule (See Course Website)

Week	Topic	Reading	Assignment Release	Assignment Due	Lab
1	Introduction & Fundamental Concepts	Ch 1	HW0 Release (Self Introduction)		
2	Instruction Set Architecture 1	Ch 2	HW1 Release (ISA1)	HW0 Due	
3	Instruction Set Architecture 2	Ch 2	HW2 Release (ISA2)	HW1 Due	
4	Arithmetic & Logic	Ch 3 & Ch 4.1-4.2	HW3 Release (Arithmetic/Logic)	HW2 Due	
5	Processor Control and Datapath	Ch 4.3-4.4	HW4 Release (Control/Datapath)	HW3 Due	
6	Processor Pipelining	Ch 4.5-4.6	HW5 Release (Pipelinig)	HW4 Due	
7	Pipeline Hazards	Ch 4.7-4.9	HW6 Release (Pipeline Hazards)	HW5 Due	
8	Caches	Ch 5.3-5.4	HW7 Release (Caches)	HW6 Due	
9	Virtual Memory	Ch 5.7	HW8 Release (Virtual Memory	HW7 Due	
10	Virtual Memory and Review	Ch 5.7		HW8 Due	
11					

Course Policies (1)

- Late policy:
  - Late submissions for HWs (5% per hour, up to 10 hours)
- One HW assignment will be dropped or can be skipped
- Submissions
  - Please ensure your files are accessible and not corrupted. Submissions that cannot be opened or read will not be accepted.
  - Please make sure to upload the correct file. Wrong file submission, missing files, etc. requests will not be considered.
- Error in grading: notify the graders and the Instructor within a week after the grade is returned

Course Policies (2)

- You are responsible for getting Canvas and Slack announcements.
- Cheating → failing grade (No exceptions)
  - Everything must be your own work (unless otherwise specified)
  - Any misbehavior will be recorded and submitted to SCAIP
- AI tools such as ChatGPT can be used to support learning, but never to do the assignments for you
  - All submitted work must reflect your own thinking and understanding
  - Submitting AI-generated responses, even if edited, is not permitted.
  - Any use of AI that bypasses your own effort and reasoning will be treated as academic dishonesty and result in a failing grade in the course.

Grading Scale

A+: 97 and above

A: 94-96.9

A-: 90-93.9

B+: 87-89.9

B: 84-86.9

B-: 80-83.9

C+: 77-79.9

C: 74-76.9

C-: 70-73.9

D+: 67-69.9

D: 64-66.9

D-: 60-63.9

F: Below 60

Discussion Sessions

- TA will cover some basic topics, tools, and examples
- Asking clarification questions on the assignments and homework
- Asking clarification questions on the lab assignment
- After deadline: going over the solution on the assignments and exams
- Asking question about the course subjects
- Any other course-related questions

How to get help

- Talk with the TA during **office hours and discussion sessions**
  - See course website for times
- Ask questions **during class** or **during office hours**
  - Utilize the course discussion forum

- **Review** on your own
  - Lectures are all online
  - Solutions will be available online
  - Textbook (optional)
    - Patterson & Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th Edition

- Work with **other students**
  - But don't cheat (see the syllabus for details)

Recommended Textbook (Optional)

- Computer Organization and Design MIPS Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design)
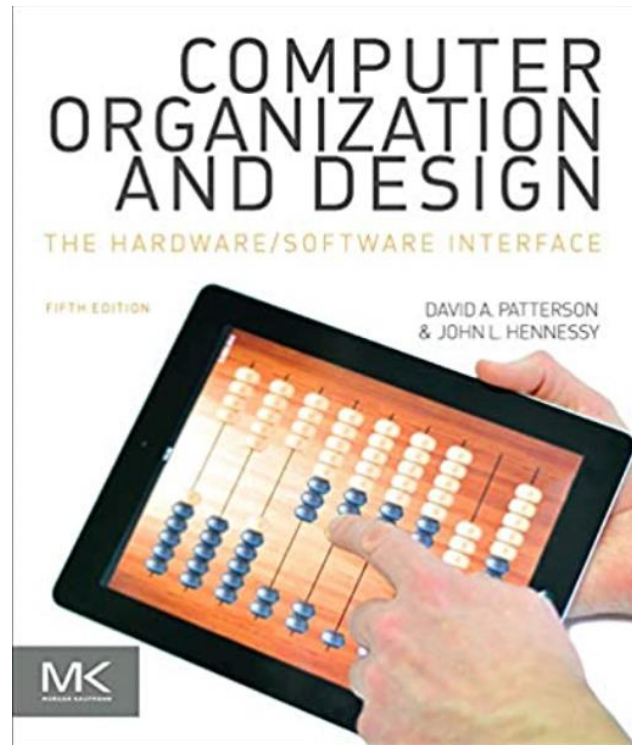
Feedback? Suggestions?

Weekly Surveys

What to do this week?

- Complete self introduction task
  - 2 points
- Refresh your mind on the logic design background

Next Lecture

- Overview of how a computer works