CSC 411

Computer Organization (Fall 2024)
Lecture 1: Logistics

Prof. Marco Alvarez, University of Rhode Island

What is this course about?

- A comprehensive and exploration of fundamental principles in computer organization
  - dive into the fascinating interplay of hardware and software that underpins modern computing systems
    - from transistors to high-level programming
- Topics covered include:
  - instruction sets, assembly language programming, processor design, the memory hierarchy, and performance optimization, with a particular focus on the **RISC-V architecture**

How computers work?

A computing system

	Problem
	Algorithm
	Program / Language
	System Software (OS, VM)
ISA (Instruction Set Architecture)	SW / HW Interface
Implementation of the ISA	Microarchitecture
Building blocks (gates)	Logic
	Devices
	Electrons

What is computer architecture?

- Science of designing and implementing computing systems
  - focus on HW/SW interface and the layers below (instruction set architecture, processor design, memory systems)
  - expanding to upper layers
- Key design goals
  - maximize performance (execute tasks as quickly as possible)
  - optimize energy efficiency (while still meeting performance)
  - achieve best performance/cost ratio
- Design priorities vary by system type
  - e.g., supercomputers emphasize throughput, smartphones prioritize energy efficiency
  - fundamental principles apply across systems, with varying optimization priorities

Why study computer architecture?

- Understand current/future capabilities of computing systems
  - why computers work the way they do (principles and design decisions)
- Develop better software
  - best system programmers understand all abstraction levels and the underlying hardware (more efficient code, debugging complex issues, exploit hardware characteristics)
- Optimizing performance across the entire stack
  - writing well tuned software based on hardware knowledge
- Establish foundation for further work
  - prepare for advanced study and career opportunities

Why learning Assembly?

- Critical for understanding machine-level execution
  - modern compilers are highly optimized, one may not need to write code in Assembly
- Key advantages
  - debug complex issues
  - optimize program performance and identify sources of program inefficiency
  - implement system software
  - enhance security against malware

To illustrate the potential gains from performance engineering, consider multiplying two 4096-by-4096 matrices. Here is the four-line kernel of Python code for matrix-multiplication:

```
for i in xrange(4096):
    for j in xrange(4096):
        for k in xrange(4096):
            C[i][j] += A[i][k] * B[k][j]
```

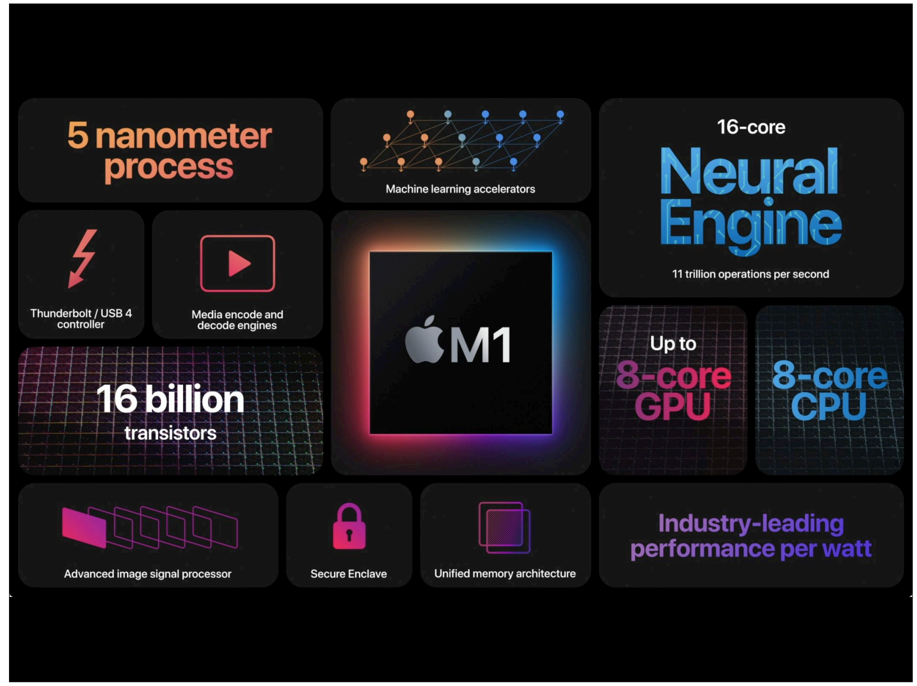
Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	—	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

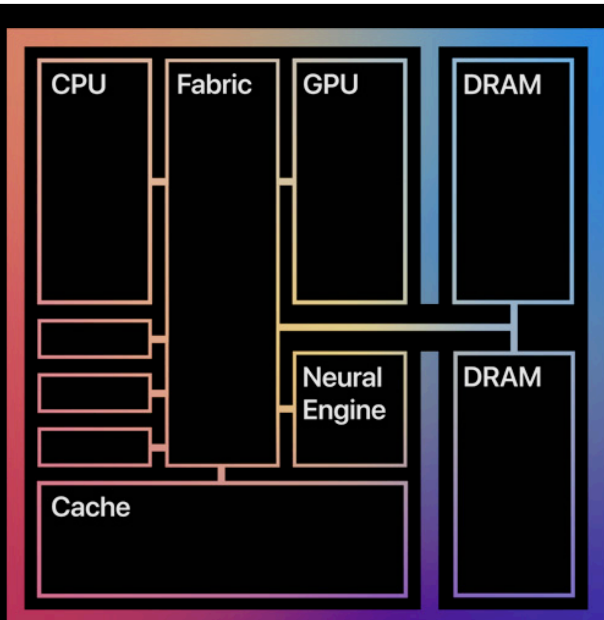
From: “There’s plenty of room at the Top: What will drive computer performance after Moore’s law? “

Modern computer architecture

- Key strategies for higher performance and efficiency
  - **co-design** across the hierarchy (bring algorithms to devices)
  - **specialize** as much as possible for targeted efficiency
- Outlook
  - fundamental building blocks and principles persist
  - emergence of novel architectures
  - evolving and powerful computing landscape
  - exciting era for computer architecture

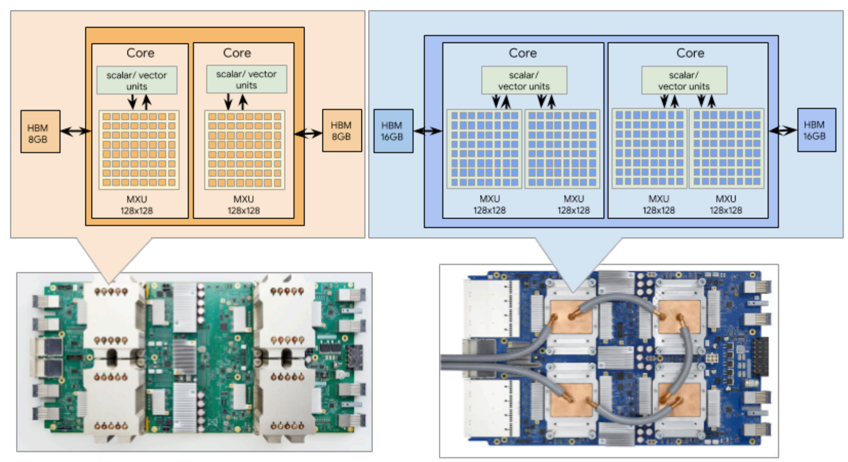
Modern computer architecture





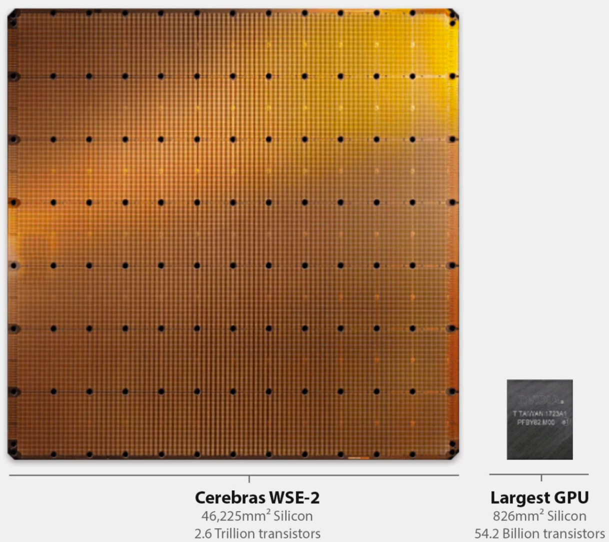
https://www.apple.com/mac/m1/

TPU (Tensor Processing Unit)



TPU v2 - 4 chips, 2 cores per chip

TPU v3 - 4 chips, 2 cores per chip



https://www.cerebras.net/product-chip/

Cerebras's Wafer Scale ML Engine

Course organization

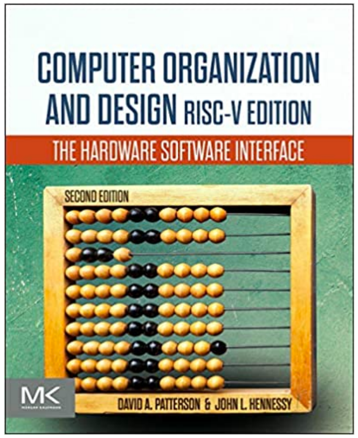
Course website

- URL
  - https://homepage.cs.uri.edu/~malvarez/teaching/csc-411
  - Syllabus
  - Schedule
  - Presentations
  - Resources

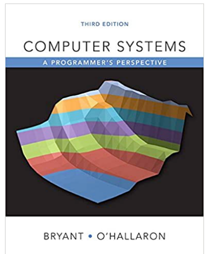
Course information

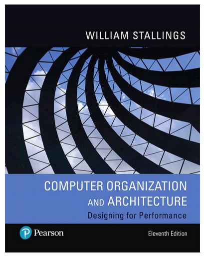
- Lectures
  - MW 4:30-5:45p
- Lab
  - T 5-5:50p (via Zoom)
- Team
  - Marco Alvarez, Instructor
  - Emily Light
  - Liam Cannon
- Office Hours
  - TBA

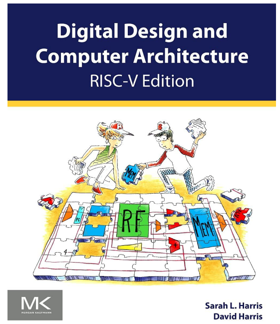
Recommended textbooks



Required







Grading

- Homework assignments (15%)
  - programming (primarily C and RISC-V)
  - problem sets
- Midterm exam (30%)
  - Oct 9th
- Technical presentation (20%)
  - teams of 3
  - specific advanced topic related to computer organization
- Final exam (35%)
  - Dec 16th

Coursework

- Homework assignments
  - combination of written problems and programming tasks
  - individual, however discussions and collaboration are allowed
    - you must write your own code and solutions
  - late submissions **NOT accepted**
    - ample time given to complete (6-9 days)
    - start and submit early, leaving plenty of time for updates
    - use office hours for guidance and feedback
- Exams
  - in-person, open-book (printed materials only)
  - no electronic devices allowed
  - mix of multiple-choice, short answer, and problem-solving questions
  - designed to test understanding, not just memorization

Coursework

- Technical presentation
  - team work (groups of 3) and dive deep into an advanced specialized topic
    - topics provided by instructor, covering cutting-edge areas in computer organization
    - 20-minute presentation to share your insights/findings
  - enhance your communication and collaboration skills
  - contribute to the class's collective knowledge
  - **outstanding presentations** will receive extra credit

Academic integrity

- Assignments
  - collaborate on ideas, not on solutions
  - sharing solutions, copying work, or using uncredited AI-generated content **prohibited**
- AI and LLMs
  - use AI tools (e.g., ChatGPT, Gemini, Claude, Copilot) to enhance understanding
    - brainstorming, concept exploration, and problem-solving approaches
    - students responsible for understanding and verifying AI-generated content
  - proper citation required in all submissions for AI-assisted work
  - AI complements, doesn't replace, your critical thinking
  - seek instructor guidance if uncertain about appropriate AI tool usage

Support tools

Ed Discussion: Academic discussions, polls, quizzes.

Gradescope: Assignment submission and grading.

Zoom: Virtual labs and office hours.

How to succeed?

- Attend all lectures/labs
  - lectures run synchronously and are not being recorded
  - attendance usually correlates with higher grades
- Participate and think critically
  - no laptops, no cellphones, unless taking notes
  - turn on your cameras during online labs and feel free to ask questions
  - use the online forum and office hours regularly
- Work on your assignments
  - focus on excellence rather than just “getting a good grade”
  - start assignments early