CSC 411

Computer Organization (Spring 2024)
Lecture 1: Logistics

Prof. Marco Alvarez, University of Rhode Island

Welcome !

- Lectures
  - TTh 5:30-6:45p @ PHMC 240
- Lab
  - W 5-5:50p @ Zoom
- Office Hours
  - TBA
- Team
  - Marco Alvarez, Instructor
  - Ethan Carlson and Carl Stoker, TAs
- Course Website
  - https://homepage.cs.uri.edu/~malvarez/teaching/csc-411/

What is this course about?

- A comprehensive and rigorous exploration of fundamental principles in computer organization
  - delving into the fascinating interplay of hardware and software that underpins modern computing systems
- 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
  - HW/SW interface and below
  - expanding to upper layers
- Design goals
  - highest performance
  - optimizing for energy efficiency
  - best performance/cost ratio
- Think about design goals for a supercomputer vs an smartphone
  - fundamental principles are similar

Why study computer architecture?

- Understanding current/future capabilities of computing systems
  - why computers work the way they do
- Developing better software
  - best system programmers understand all abstraction levels and the underlying hardware
- Understanding computer performance
  - writing well tuned software requires knowing what's under the hood
- Setting up the fundamentals for further work on hardware design

To illustrate the potential gains from performance engineering, consider multiplying two 4096-by-4096 matrices. Here is the four-line kernel of the Python 2 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

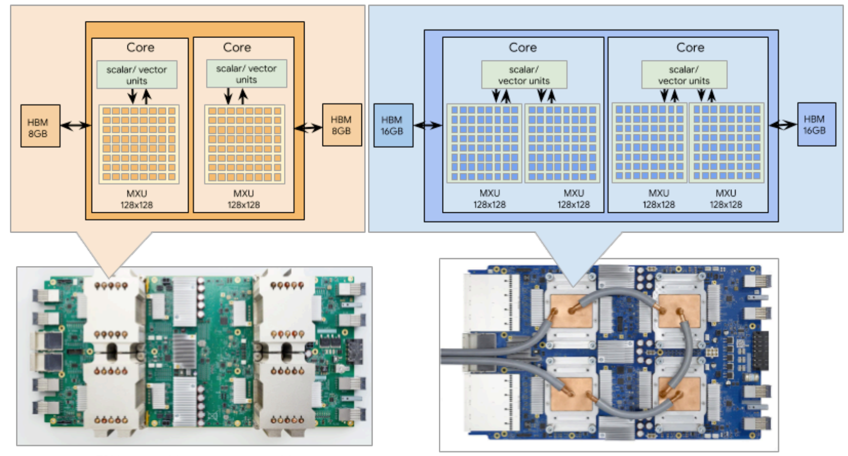
From: "There's plenty of room at the Top: What will drive computer performance after Moore's law? "

Modern computer architecture

- Achieving higher performance and efficiency
  - **co-design** across the hierarchy (bring algorithms to devices)
  - **specialize** as much as possible
- Looking forward …
  - same basic building blocks and design principles
- Exciting times in computer architecture
  - novel architectures
  - renovated and powerful computing landscape

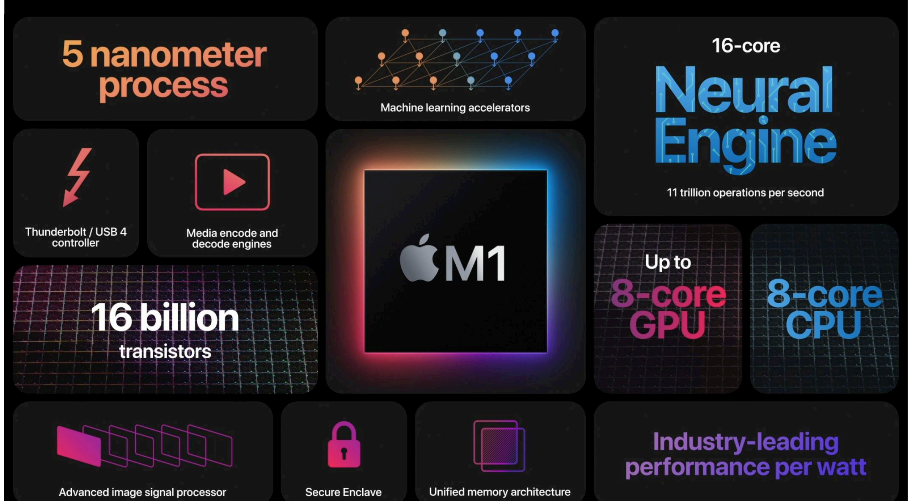
Examples of modern processors/systems

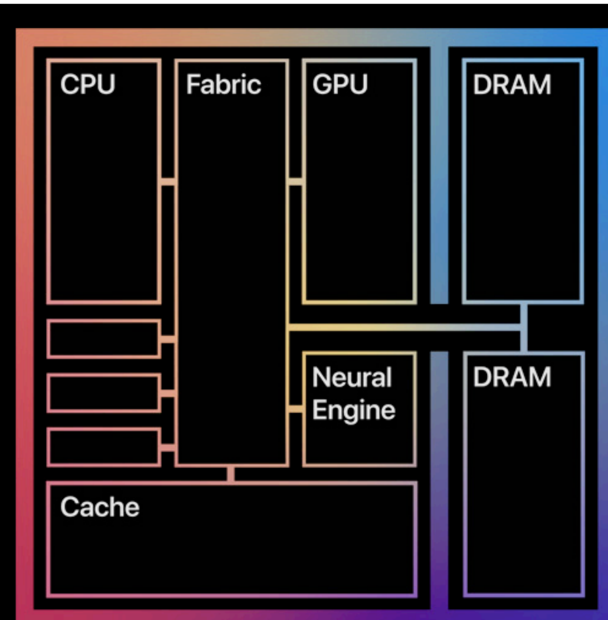
TPU (Tensor Processing Unit)



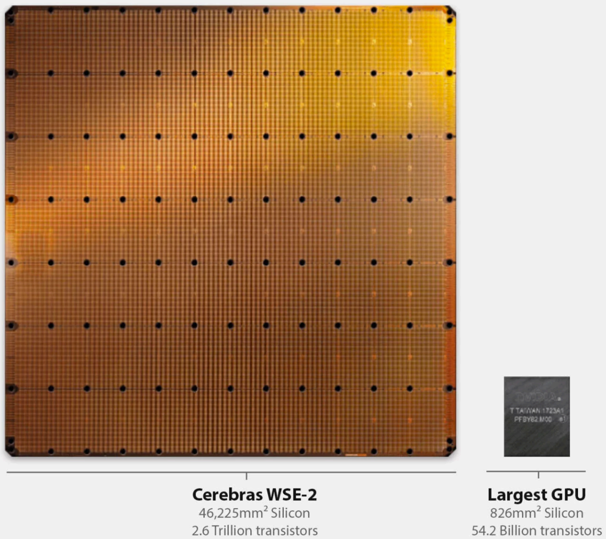
TPU v2 - 4 chips, 2 cores per chip

TPU v3 - 4 chips, 2 cores per chip





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



Cerebras's Wafer Scale ML Engine

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

Fugaku

Fugaku remains the No. 1 system. It has 7,630,848 cores which allowed it to achieve an HPL benchmark score of 442 Pflop/s. This puts it 3x ahead of the No. 2 system in the list. Fujitsu RIKEN Center for Computational Science, Japan.

https://www.top500.org/lists/top500/2021/11/

Course organization

Recommended textbooks

Required

Grading

- Attendance and participation (5%)
  - randomly recorded at select lectures and labs
- Assignments (20%)
  - programming (primarily C and RISC-V)
  - problem sets
- Technical presentation (20%)
  - teams of 2
  - delving deeper into a specific topic related to computer organization
- Midterm exam (25%)
  - Mar 7th
- Final exam (30%)
  - May 7th

gradescope

Assignments

- Discussions and collaboration are allowed
  - you must write your own code and solutions
- Late submissions **NOT** accepted
  - ample time given to complete (7-10 days)
  - start and submit early, leaving plenty of time for updates
- LLMs?
  - students are encouraged to use LLMs responsibly and ethically as aids to enhance their comprehension of the subject matter
  - it is imperative that students produce original work for their assignments
- Plagiarism?
  - just don't do it
  - reports are sent to the chair with copies to your dean, the student's dean, and the office of student life

Support tools

ed

Ed Discussion: hosts forums for discussions, polls, and quizzes, providing opportunities for students to interact with classmates and instructors, share insights, and test their understanding of key concepts.

gradescope

Gradescope: simplifies assignment submissions and grading, ensuring clear and consistent feedback from instructors. Students can review their graded work, visualize feedback points, and gain insights into their strengths and areas for improvement.

zoom

Zoom: the platform of choice for hosting virtual lab sessions and office hours. Students are strongly encouraged to connect using their University provided accounts on laptops or computers. Some lab sessions may involve the use of software beyond what is available on mobile devices.

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
  - turn on your cameras during online labs and feel free to ask questions
  - use the online forum (Ed Discussion)
  - use office hours regularly
- Start working on assignments early
  - avoid copying/pasting or google'ing answers