CS 250: Foundations of Computer Systems
[Introduction]

Computer Science
Colorado State University

** Lecture slides created by: Shrideep Pallickara
** Lecture slides edited by: Ariana Mims

Topics covered in this lecture

- Introduction
- Course overview and expectations
- Communications


BANG

BANG

- It's argued that 21st century folks ought to familiarize themselves with the key ideas underlying **BANG**
  - Bits, Atoms, Neurons, and Genes (BANG)
- Science has been remarkably successful in uncovering their core ideas
  - Quite possible we may never fully grasp how atoms, neurons, and genes actually work
- A consoling exception?
  - Bits and computing systems at large

Gaining a gestalt understanding of how the machine works

- At least one field in this BANG quartet can be fully laid bare to human comprehension
- The interactions between hardware and software were simple and transparent enough to produce a *coherent* picture
- Alas, as digital technologies have become increasingly more complex, this clarity is all but lost
  - Hidden under many layers of obscure interfaces and proprietary implementations

Inevitable consequence of this complexity?

- **Specialization**
  - Pursuit of many *niche* courses, each covering a single aspect of the field
- Many computer science students are missing the forest for the trees
  - Marshaled through a series of courses in programming, theory, and engineering
    - Without pausing to appreciate the beauty of the picture at large
- We will strive to do this
  - Allowing you to view problems (and devise solutions) from multiple vantage points

About Me

About me

- Masters in Computer Science and Cyber Security
  - Currently focused on Privacy and the implications of AI and Cybersecurity in "smart cities"
  - Specialized in ethical hacking of OT networks
- Bachelors in CS from CSU
  - Focus in Networks and Security
- Worked in academia, government, and industry

Contacting me

- Direct messages on Teams
  - Will get a response within 24 hours during the week
- Email - preferred for “emergencies” that require longer explanations / details or other people accross the university to be involved
  - Still send the direct message on teams directing me to the email
  - Could have a significantly longer wait time
- Stopping by office hours in CSB 450 or when my door is open

Course Logistics, Expectations, and such

Course textbook

- This class has three **optional** textbooks
  - The Elements of Computing Systems, second edition: Building a Modern Computer from First Principles. 2nd Edition. Noam Nisan and Shimon Schocken. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press.
  - How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine. Matthew Justice. ISBN-10/ISBN-13 : 1718500661 / 978-1718500662. No Starch Press.
  - The Secret Life of Programs: Understand Computers - Craft Better Code. Jonathan E. Steinhart. ISBN-10/ ISBN-13 : 1593279701 / 978-1593279707. No Starch Press.

The course slides …

- These slides refer to, and are built, from several texts
  - And technical papers and articles (with URLs)
- The references are listed at the end of every slide set

Textbooks that are reffered to during the course include … (1/2)

- The Elements of Computing Systems, second edition: Building a Modern Computer from First Principles. 2nd Edition. Noam Nisan and Shimon Schocken. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press.
- How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine. Matthew Justice. ISBN-10/ISBN-13 : 1718500661 / 978-1718500662. No Starch Press.
- The Secret Life of Programs: Understand Computers -- Craft Better Code. Jonathan E. Steinhart. ISBN-10/ ISBN-13 : 1593279701 / 978-1593279707. No Starch Press.
- Crafting Interpreters. Robert Nystrom. ISBN-10/ ISBN-13 : 0990582930 / 978-0990582939. Genever Benning.

Textbooks that are reffered to during the course include … (2/2)

- Computer Networks: A Systems Approach. Larry Peterson and Bruce Davie. 4th edition. Morgan Kaufmann. ISBN: 978-0-12-370548-8.
- Alex Petrov. Database Internals. ISBN-10/13: 1492040347/978-1492040347 O'Reilly Media.
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Martin Kleppmann. ISBN-10/ ISBN-13 : 1449373321 / 978-1449373320. O'Reilly Media. 2019.
- Shane Cook. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing). ISBN-10/ISBN-13: 0124159338/978-0124159334. 1st Edition. Morgan Kaufmann.
- Jeremy Kubica. Data Structures the Fun Way: An Amusing Adventure with Coffee-Filled Examples. No Starch Press. ISBN-10. /13: 1718502605/978-1718502604.
- Computer Organization and Design MIPS Edition: The Hardware/Software Interface. 5th Edition. David A. Patterson and John L. Hennessy. ISBN-10/ ISBN-13 0124077269/ 978-0124077263. Morgan Kaufmann.

On the Course schedule page

http://www.cs.colostate.edu/~cs250/schedule.html

- You will see the **topics** that will be covered and the **order** in which they will be covered
- The readings section will list the books (and the chapters therein) that form the basis for the materials
- You will also see the complete schedule for when the **assignments** are posted and when they are due

Canvas

- Where you will submit assignments
- Where you will be graded

Infospaces (**https://infospaces.cs.colostate.edu**)

- A **knowledge repository** that Shrideeps lab is building to enhance learning
- All videos are designed to be less than 2 minutes
- Improving Infospaces
  - Let us know what you would like to see
  - If you'd like to contribute to this repository let us know!

Office Hours

- Instructor: **Ariana Mims**
  - Office Hours: Thursdays from 9:00-10:00 am in CSB 450 and via teams or by Appointment
- TA: Office Hours will be in CSB120 and via MS Teams
  - **Dennis Kim** {Graduate Teaching Assistant}
  - **Yunik Tamraka**r {Graduate Teaching Assistant}
  - **Phil Hopkins** {Graduate Teaching Assistant}
  - **Santoshkumar Tongli** {Graduate Teaching Assistant}
  - **Emily Cosgriff** {Undergraduate Teaching Assistant}
  - **Sanjar Yuldashev** {Undergraduate Teaching Assistant}
  - **Kushal Alimineti** {Undergraduate Teaching Assistant}
  - **Omar Soliman** {Undergraduate Teaching Assistant}

Communications

- Do NOT use Canvas Messages for communications
- Send an email to the course email compsci_cs250@colostate.edu
  - An account that's checked by the entire team and allows us to respond to communications in a timely fashion
  - Send emails from accounts that match your name
    - No pseudonyms, please
- Make a post in the teams
  - DO NOT POST CODE IN TEAMS

Grading

Grading breakdown

- Assignments: 30%
  - HW1: 7.5%; HW2: 7.5%; HW3: 7.5%, and HW4: 7.5%
- Recitations {attendance + completion}: 10%
- Quizzes (10 best) : 10%
- Mid Term I: 15%
- Mid Term II: 15%
- Comprehensive final exam: 20%

Grading Policy [1/4]

- Letter grades will be based on the following standard breakpoints:
  - >= 90 is an A, >= 88 is an A-,
  - >= 86 is a B+, >=80 is a B, >=78 is a B-,
  - >= 76 is a C+, >=70 is a C,
  - >= 60 is a D, and <60 is an F
- I will not cut higher than this, but I *may* cut lower

Quizzes

- There will be 12-13 quizzes
  - We will take your 10 highest scores
  - If you miss class for some reason other than emergencies and university sanctioned reasons, you do not need to let me know and there will be NO makeup for quizzes (please don't ask to do this!)

Quizzes, midterms, and final

- Will only contain content that was taught in class
  - If it was not taught, it would not be on a quiz or exam
  - If they were covered in the slides, you should be able to answer the questions

Assignments

Assignments will target the following elements…

- Number representations and computing
- Memory
- Networking
- Storage systems

Assignments: Logistics

- Assignments will be due **at 8:00 pm MT on Wednesdays**
- You are allowed to submit up to 2 days late
  - There is a **7.5%** deduction for each day that you are late
  - Submissions after the late period will receive a ZERO
  - Submitting the wrong files? 30% deduction
- All assignments are **individual** assignments
- Assignments should be submitted via **Canvas**

Grading Policy

- If you have problems with the grading
  - Talk to the TAs first
  - Complaints must be lodged via an email to compsci_cs250@colostate.edu within **10** days of the grade being posted on Canvas

RECITATIONS

Recitations: Will be in the CS Building

- **Recitation Schedule**

Recitation	Day	Time	Location
R1	Wed	9:00 - 9:50 AM	CSB-315
R2	Wed	11:00 - 11:50 AM	CSB-315
R3	Wed	12:00 - 12:50 PM	CSB-315
R4	Wed	1:00 - 1:50 PM	CSB-315
R5	Wed	2:00 - 2:50 PM	CSB-315
R6	Wed	3:00 - 3:50 PM	CSB-315

  - Will be helpful to prepare you for the assignments
  - Recitation grading a based on **attendance** and **completion** scores

EXPECTATIONS

Expectations

- Attend all classes
- You will focus on the discussions and not on..
  - Other assignments
  - Your phone
  - Shopping on Amazon

What it takes to Succeed

What it takes to succeed

- You are expected to work at least 9-10 hours a week outside of class
- If you miss a lecture add 3 hours per missed lecture
- Work on assignments every day
  - There is no such thing as waiting for inspiration to strike!
- Reflect on your performance
- Work in bigger-sized chunks
- Document your code

How to fail this class

- Believing that you can learn just from showing up
- Missing lectures
- Procrastinating
- Not attacking the problem and working on the fringes

Please don't take away learning opportunities from other students

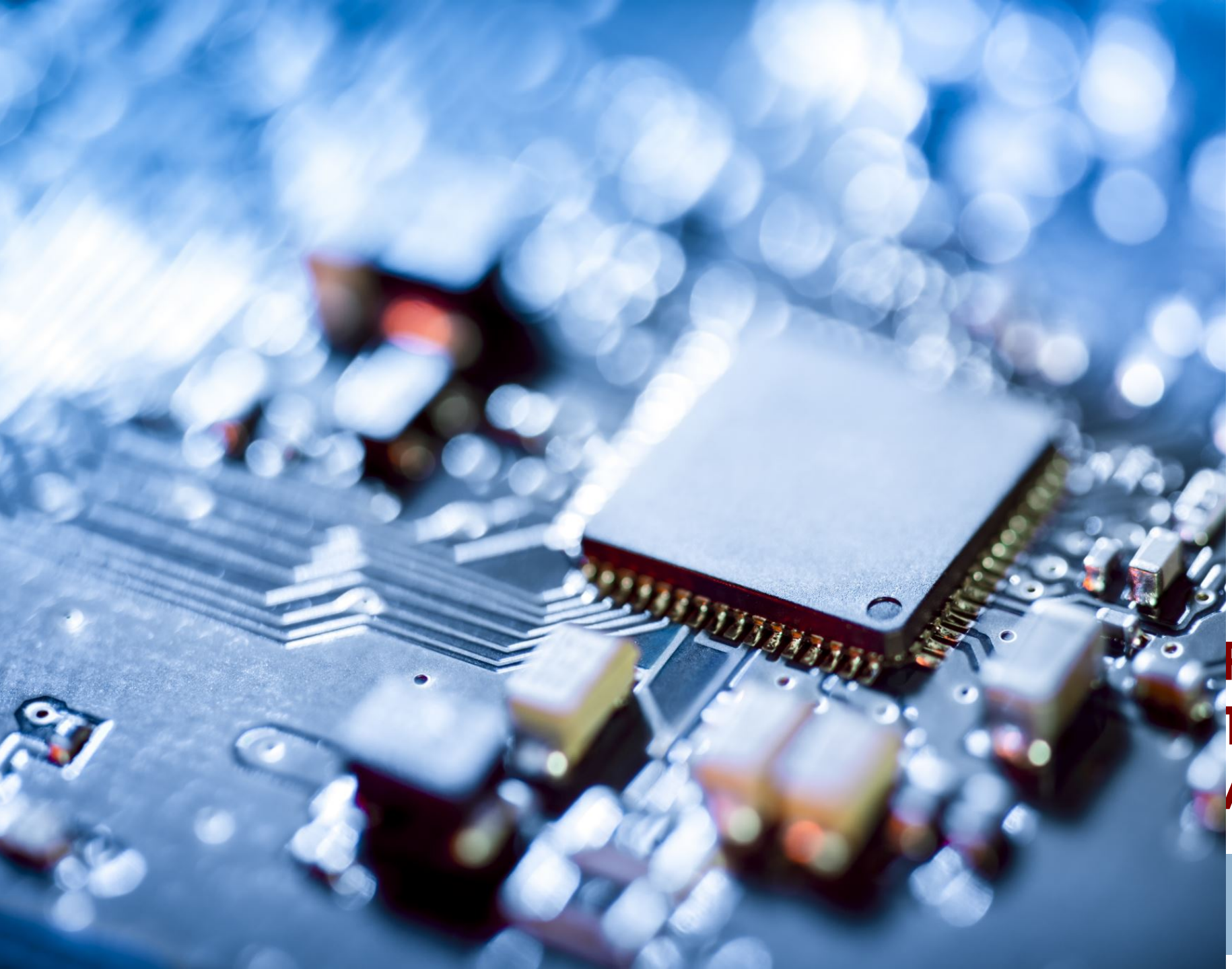
- If you need to use a laptop or tablet with an external keyboard
  - Sit in the last 2 rows
  - Turn off wireless
  - Use it for taking notes
- Tablet with a stylus or pencil is fine
- Put your phones away

TOPICS COVERED IN CS250

Part I: Binary and Boolean Logic

Binary and Boolean Logic

- Number representations
  - Boolean Algebra, Boolean Logic
- Gates
- Signed numbers and floating-point representations
  - Two's and One's complement number representations
- Synthesize Boolean functions from Truth Tables
- Prove how all Boolean functions can be constructed using only NAND gates



Part II: The von Neumann Architecture

The von Neumann Architecture

- Processors, cores and hyperthreading
  - Mapping threads to execution pipelines
- Memory hierarchy and its impact on performance
  - Cache organization: L1, L2 and L3 caches
  - Associative memory and direct-mapped caches
  - Main Memory: CPU RAM (addressing and organization)
- Why miniaturization works?

CPU meet the GPU (TPU, etc.)

- GPU: Design, operations, and concurrency
- The right tool for the right job
  - How the GPU waxed while the CPU waned
  - GPU Limitations: general purpose computations and memory management

Part III
Networked Computer Systems

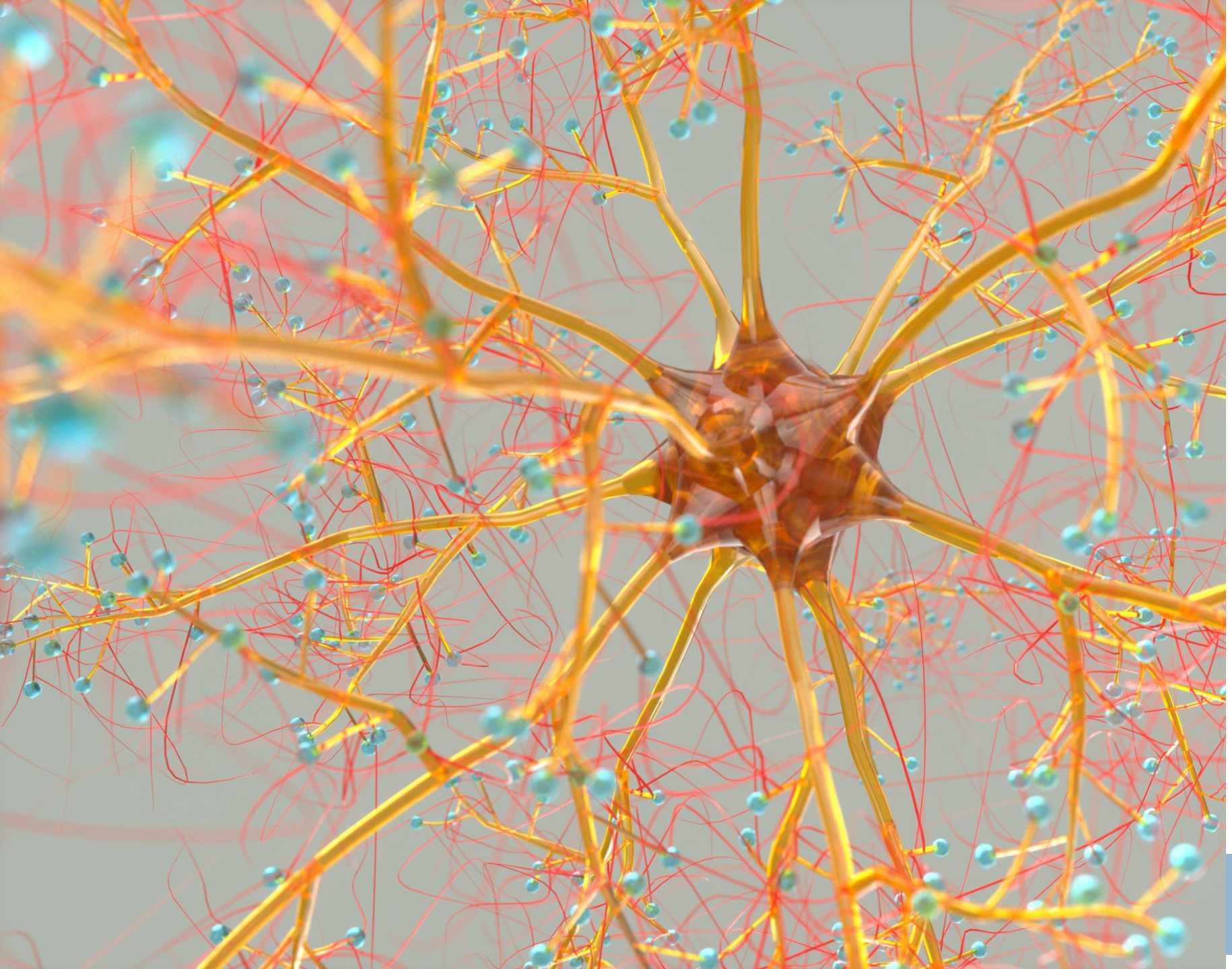
Networking

- IPv4 and IPv6
  - Encapsulation, packet header formats, fragmentation, and extension headers
- TCP
  - Sliding window, buffering, reliable and ordered delivery
- UDP
- DNS, private IP addresses, and NAT

Part IV: Storage Systems

Storage Systems

- File systems and databases
  - File control blocks and indexing schemes
- Pitfalls of using in-memory data structures for storage systems
- Binary search, dynamic data structures, BSTs
- B-Trees (and variants)



PART V: FUTURE COMPUTING SYSTEMS

Future Computing Systems

- The von Neumann bottleneck
- Where to from here?

Algorithms, Computations & Computers

Algorithm

- Sequence of steps to accomplish desired objectives
  - E.g., sorting numbers, factorizing numbers, processing graphs, etc.

- Humans can carry out algorithms, but we are much slower than machines
  - Modern computers are a trillion times faster!

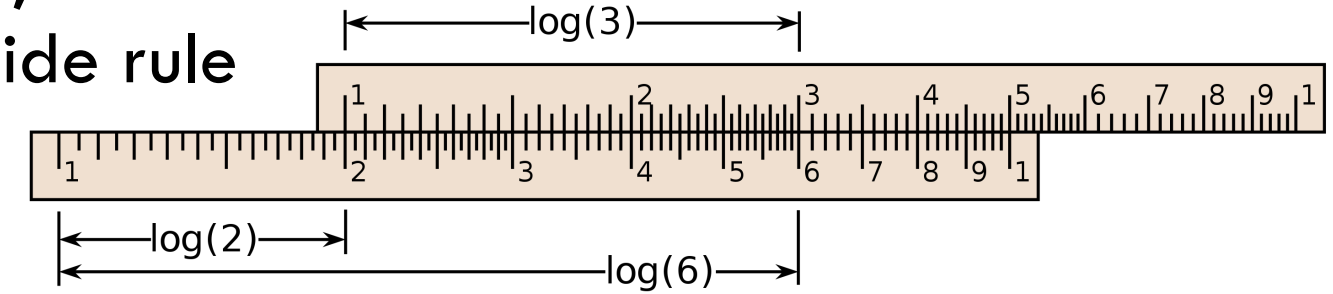
Computations and computers

- Computations are complex series of numerical calculations
- Computers are agents that carry out operations within a computation

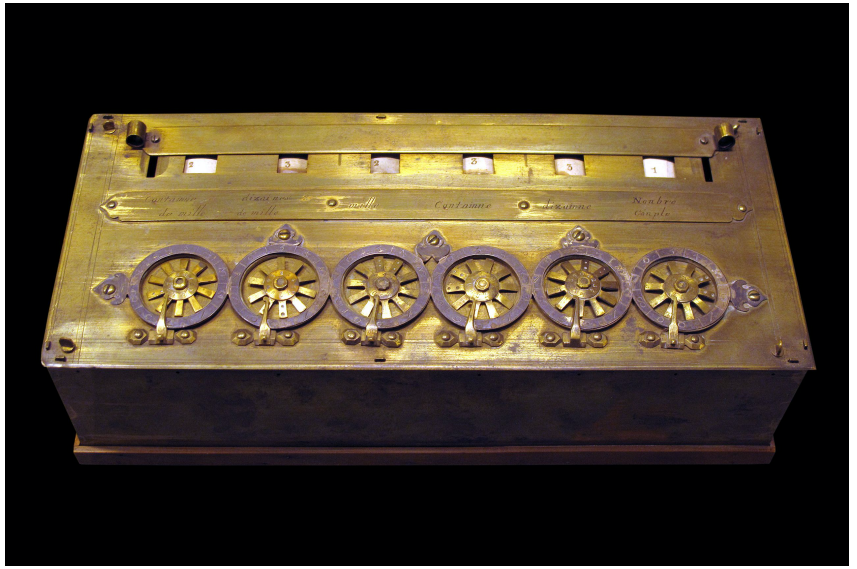
A (very) Brief History Of Computers

A Brief History of Computers: Part I

- Napier invented the logarithm, which became the principle of the slide rule (invented circa 1620)
  - Could not add or subtract

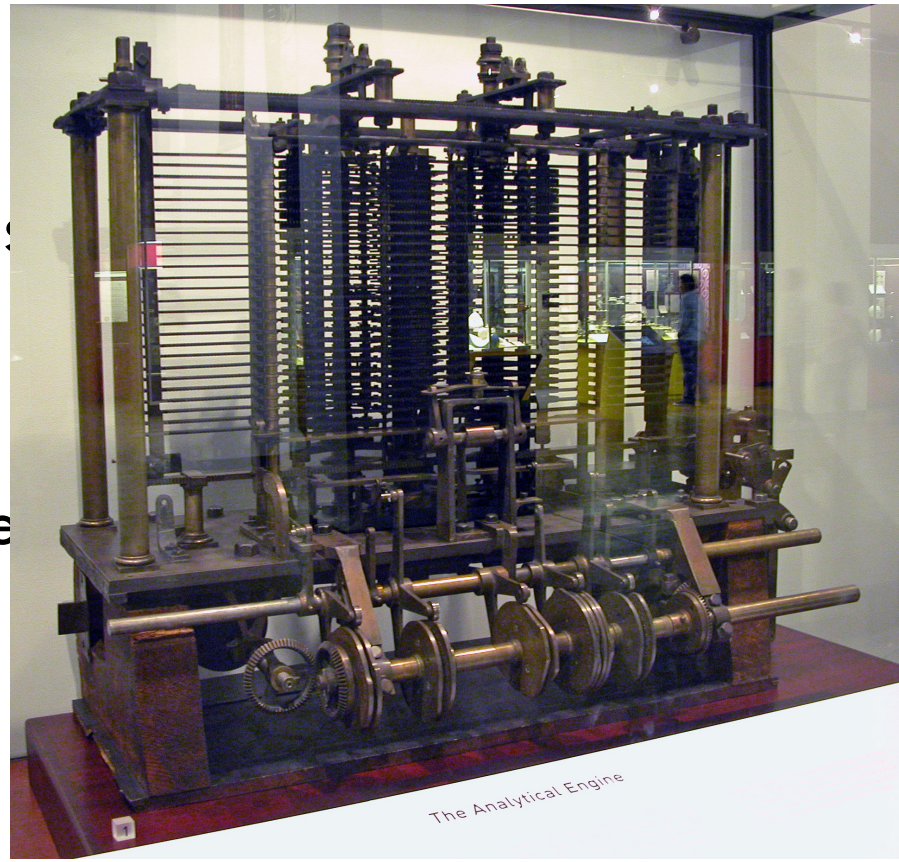


- Blaise Pascal designed and built an arithmetic machine in 1642 to add and subtract



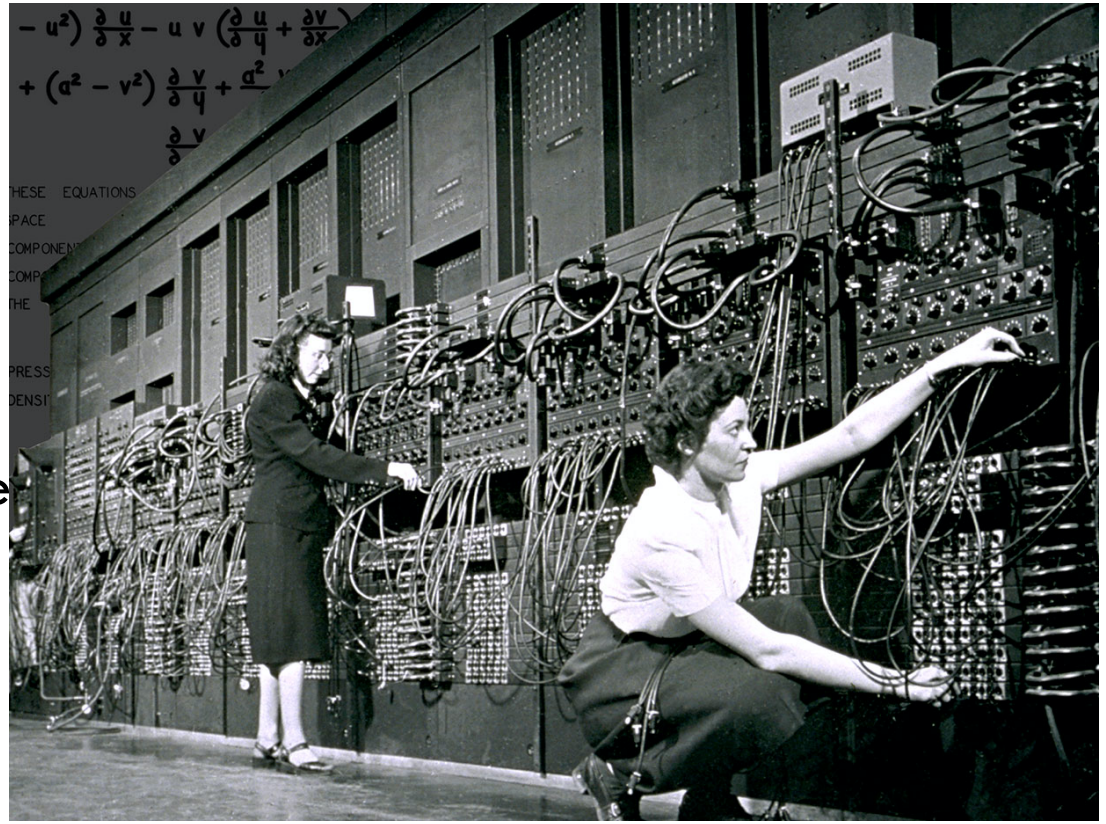
A Brief History of Computers: Part II

- In 1819, Babbage designed a machine of gears, shafts, and wheels that could calculate tables of arithmetic numbers such as logarithms
- 1890 US census, Hollerith's punched card machines tabulated large amounts of data
  - Jacquard's loom (1801) was the first place where punched cards were used



Modern computer systems design

- Has it origins dating back to1947 as part of work on ENIAC
- Breaks up a computing machine into three main subsystems:
  - The central processing unit (**CPU**) for performing arithmetic operations
  - The **memory** for storage of instructions and data
  - The input-output (**I/O**) unit for communicating with the external world
- This way of organizing a machine became known as the "**von Neumann architecture**"



Modern computers: The secret sauce?

- At its core modern computers harnesses the **movement of electrons** in circuits to carry out computations
- Computer circuits deal only with voltages, currents, switches, and malleable materials
  - Internally the computer does not process numbers and symbols

No information without representation

- To be processable, data must be represented as
  - Signals in the machine or
  - As measurable disturbances in the structure of storage media

COMPUTATIONS & COMPLEXITY

Computation complexity

- Measured in terms of time, space, and (increasingly energy)
- Impacts responsiveness and how you can scale
- Other implications
  - Cryptocurrency: Systems like Bitcoin, where the currency represents a solution to a problem and reflects the amount of computational work that needs to (and has been) performed

Perils of wishful thinking [Part I]

- Most common wish is that we can get computers to do **any** job we can conceive of
- Many jobs are impossible for computers
  - No algorithm that will inspect another algorithm and tell us whether it terminates or loops forever
    - It was logically impossible in 1936 when Alan Turing proved it so, and it is still impossible today

A problem for you to solve

- UPS truck driver must deliver packages to 10 cities
  - Wants to travel the **least** distance, but visit each city exactly once

Dept. Of Computer Science, Colorado State University

Perils of wishful thinking [Part II]

- Even if we stick to logically possible jobs, there are many that cannot be done in a reasonable time — they are **intractable**

- Example: Shortest tour of a set of cities that visits each city just once
  - The simplest way to find the shortest tour is to enumerate all possible tours and select the shortest

Intractability and Heuristics

- For a small set of 100 cities, this would take 10^{130} years on the world's fastest supercomputer
  - The age of the universe is 10^{10} years
  - Even the "simplest way" can be impossible!

- The picture gets even more confusing because in most cases there are fast algorithms to find an **approximate** answer
  - They are called **heuristics**

Wishful thinking is also when you believe the computer is smart

- If you are not precise in translating the algorithm into program steps, your computation will contain errors
- The computer amplifies your intelligence but has none of its own!

The contents of this slide-set are based on the following references

- Noam Nisan and Shimon Schocken. *The Elements of Computing Systems: Building a Modern Computer from First Principles*. 2nd Edition. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press. [Preface]
- Peter J. Denning and Matti Tedre. *Computational Thinking*. Essential Knowledge series. The MIT Press. ISBN-10:ISBN-13 0262536560/ 978-0262536561. 2019. [Chapter 2]