Introduction

#1

CS 0007
Introduction to
Computer Programming

Luís Oliveira

Summer 2020

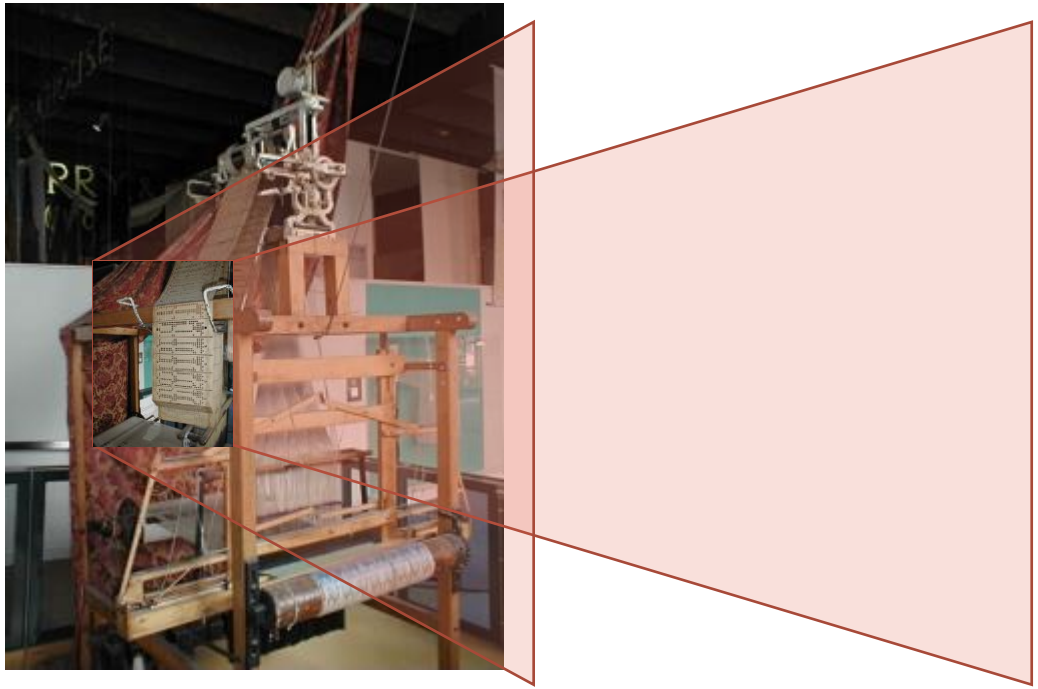
Computers

Where do they come from?

The Antikythera Mechanism

- **Thousands of years old**
  - Late second/early first century BC
  - That's like -100 ish
- **Used for astronomy**
  - Eclipses
  - Astronomical positions

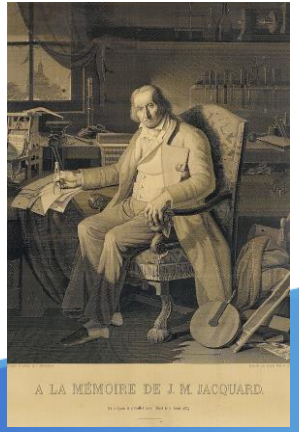
They are not (ALL) war machines

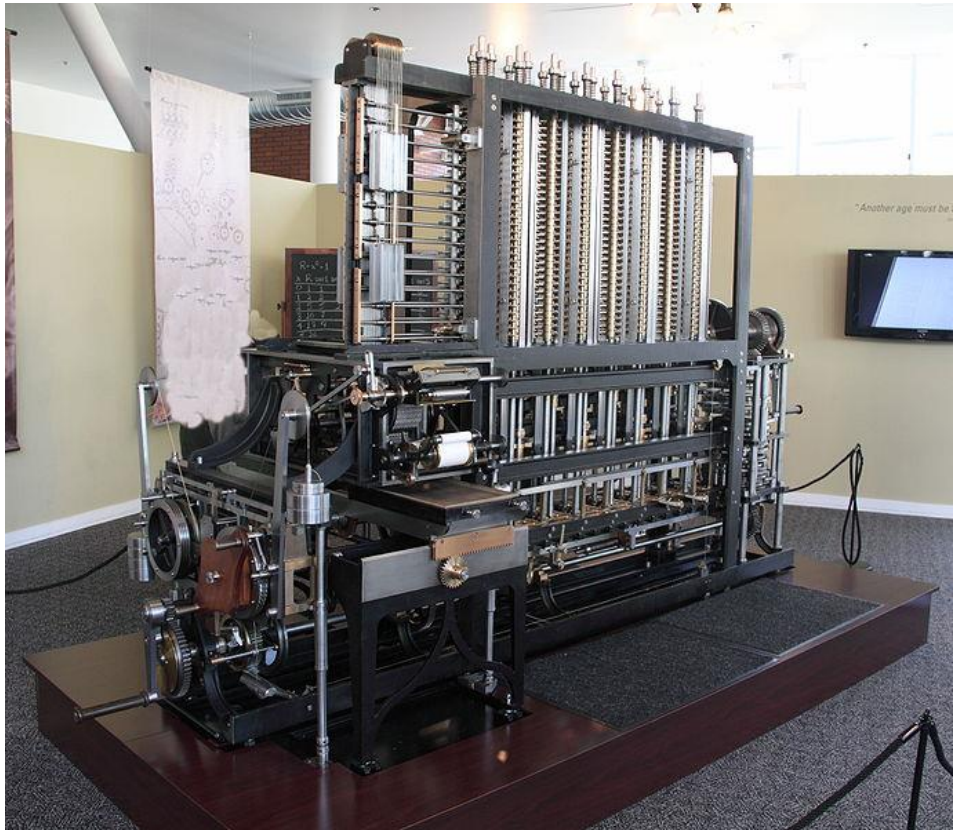


Jacquard machine

- **Mechanical loom (1804)**
  - Programmed using perforated cards
  - Used to produce complex patterns

Woven in silk using 24k punch cards!





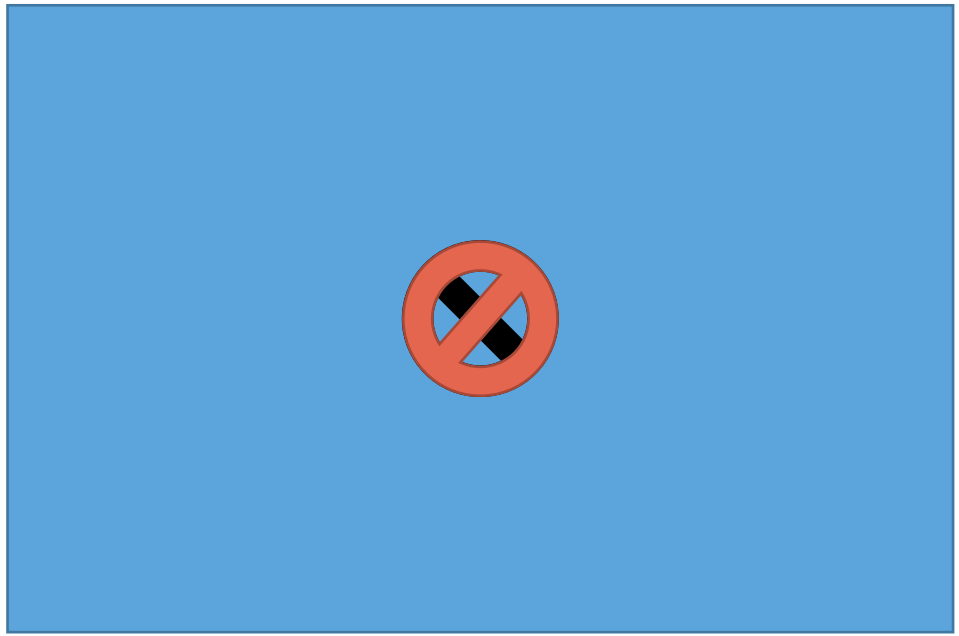
The Differential Engine

- **Designed by Charles Babbage**
  - 1792-1871
- ***Ermmm… Designed… right!***
  - It was intended as a programmable calculator
  - A **multipurpose calculator!**

The pre-history of computers

- The Differential Engine
  - Devised by J.H. Müller in the Hessian army (1784)
  - Designed by Charles Babbage (1819-ish)
  - Built at Science Museum library in London (1980s)
  - Outputs to a table that can be used for printing
    - Copying was a source of error
    - It still is nowadays
    - So never copy results manually if you can avoid it

The Analytical Engine



- **Designed by Charles Babbage**
  - *YES! Designed… again!*
- **Mechanical general-purpose computer**
  - Which had many modern characteristics

No actual picture because… it was never built

The pre-history of computers

- The Analytical Engine
  - It already included the essential ideas of modern computers
    - *Inputs and outputs*
    - *Execution of operations*
    - *Automatic control of operation*
  - However, due to its complexity (lack of funding) it was never built
    - And the fact that new features were constantly being added!
    - And old features were never completed
      - Does this sound familiar? It will, it will! ☺

“The Enchantress of Numbers” - the first programmer

- Augusta Ada King, Countess of Lovelace
  - Wrote **algorithms** for this computer →
    - Yeah, that one!
    - But they probably would have worked
  - Translating a paper, she added notes
  - A LOT of notes
    - More than the actual paper
  - Including instructions on how to calculate a number series
    - Note G
  - Studied the relation between maths and music

The Analytical Engine **has no pretensions (…) to *originate* anything**. It **can do whatever we *know how to order it* to perform**.

It **can *follow* analysis**; but it has **no power of *anticipating*** any analytical relations or truths.

(replica)

Hollerith Electric Tabulating System

- **Census happen every 10 years**
  - *Hey, they just did!*
- **It took people 8 years to count responses (in 1880)**
  - It would soon take more than 10!
  - 7,000 cards a day using this system
- **Company would become IBM**
  - After a merge with others

https://www.computerhistory.org/revolution/punched-cards/2/2/5

Bubbles

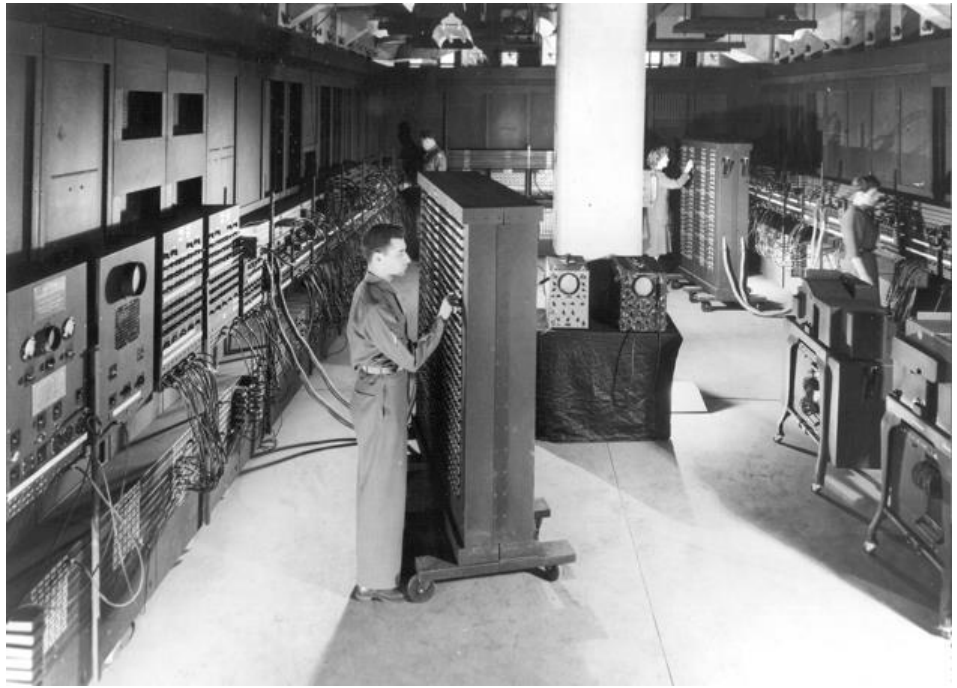
- Check what Bubbles has to say about it ☺
  - https://www.youtube.com/watch?v=L7jAOcc9kBU

Driven by the need for complex calculations

https://computerhistory.org/blog/first-steps-lectures-from-the-dawn-of-computing/

- George Stibitz (Bell Labs)
  - **Day-job**: Electrical engineer
  - Model K – binary addition **with relays** (Boolean algebra)
  - Complex Number Computer – used remotely via telegraph lines!!
  - Art with Amiga (1990s) - http://stibitz.denison.edu/art.html
- Konrad Zuse (Germany)
  - **Day-job**: Aircraft designer (civil engineer)
  - World's first programmable computer
  - Several computers used for military calculations
- John Atanasoff (Iowa State)
  - **Day-job**: Physics professor
  - Built the ABC (Atanasoff-Berry Computer)
    - solved 30 equations in 30 unknowns

ENIAC (Electronic Numerical Integrator and Computer)

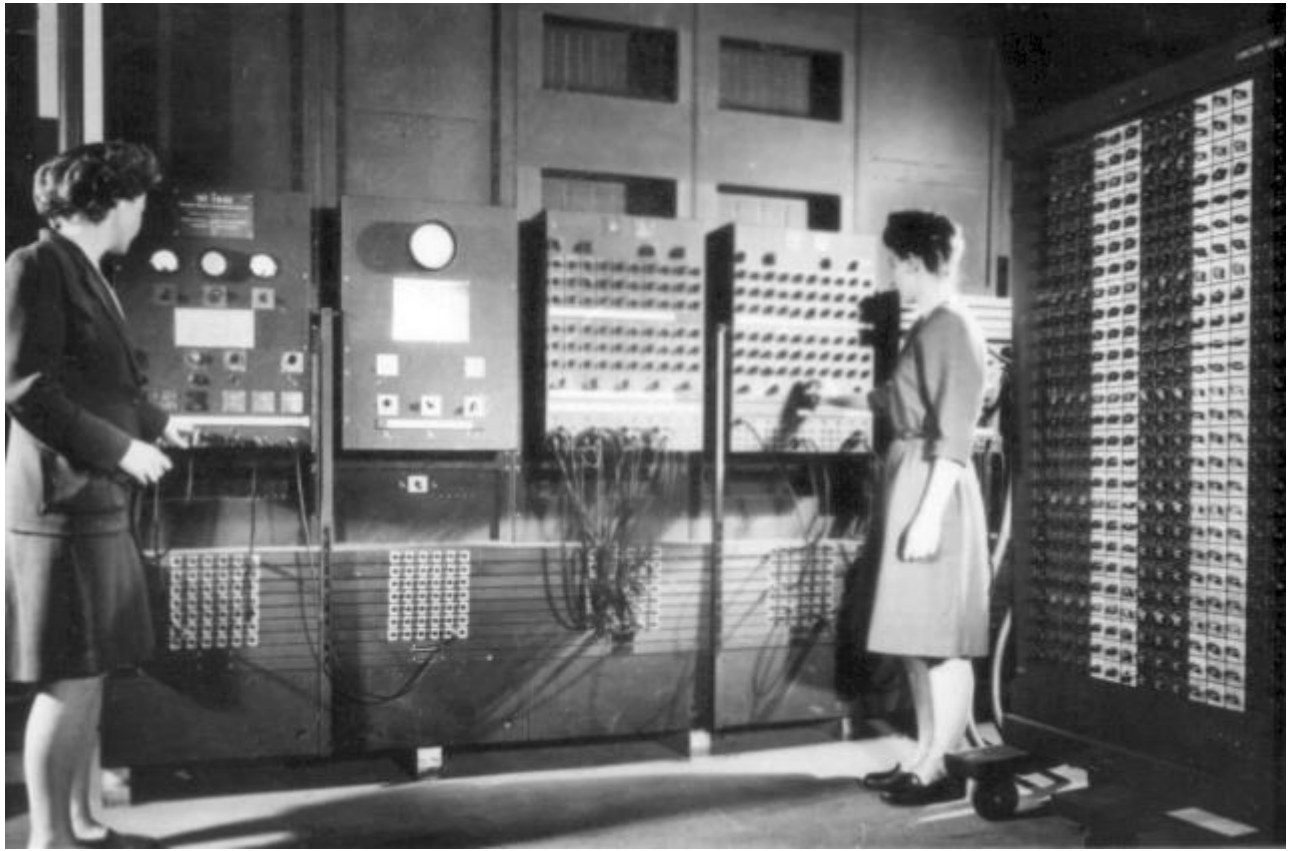


U. S. Army Photo

- 1946 – ENIAC
  - *University of Pennsylvania*
  - *Developed during WWII to calculate balistic missile trajectories*
  - *Designed by :*
    - *John Mauchly*
    - *J. Presper Eckert*
  - *Joined by a huge team!*
  - *Modular and reconfigurable*
    - *Flipping switches and connecting cables*

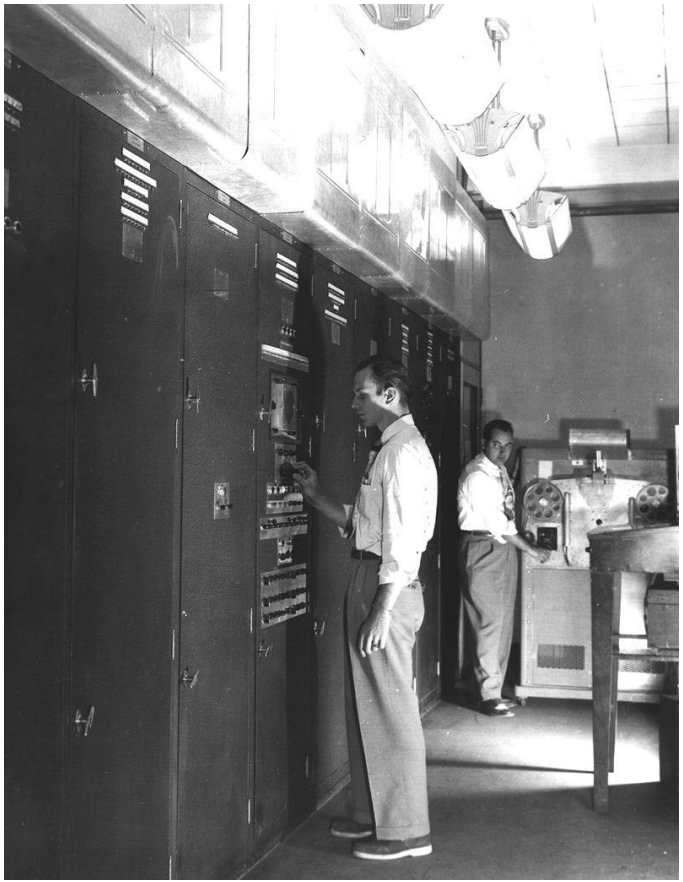
ENIAC (Electronic Numerical Integrator and Computer)

- Some numbers:
  - 18000 valves (tubes)
  - 1500 relays
  - 30 tons
  - 175 kW
  - 5000 additions / s
  - 357 multiplications / s
  - 40 divisions / s
  - Programs "hardwired"



U. S. Army Photo

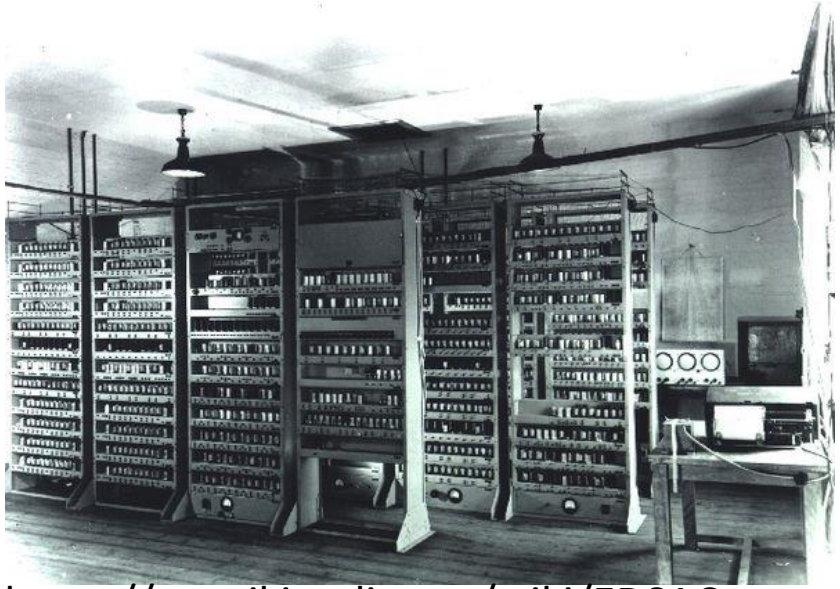
EDVAC (Electronic Discrete Variable Automatic Computer)



U. S. Army Photo

- 1947 – EDVAC
  - *University of Pennsylvania*
  - *The ENIAC team joined by John Von Neumann*
  - *A computer with a new concept:*
    - *"Memory Stored Program" – same as data*
  - *Became operational in 1951*

EDSAC (Electronic Delay Storage Automatic Calculator)



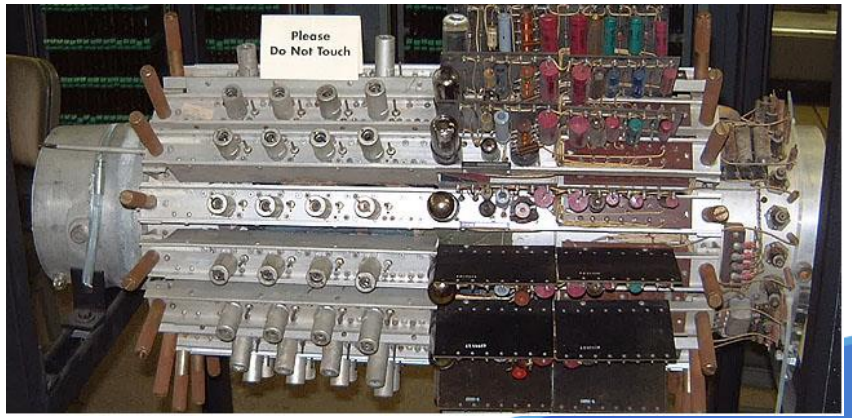
https://en.wikipedia.org/wiki/EDSAC

- 1949 – EDSAC
  - *Cambridge University*
  - *Designed by Maurice Wilkes*
  - *Based on the first EDVAC draft*
    - Not to be better, but to be used!
      - accessible and practical vs. push technology
    - Was completed before the EDVAC!
  - Used for scientific research
    - Chemistry, Medicine, Physics

https://www.tnmoc.org/edsac

UNIVAC (Universal Automatic Computer)

- 1951 – UNIVAC
  - *First commercial computer!*
    - *Sold 46! Units*
    - *Used to predict the 1952 presidential election*
  - *Used MERCURY!! memory (as did the EDSAC)*



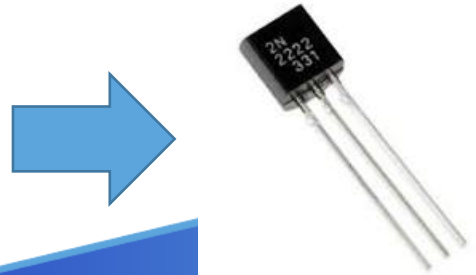
Delay Storage

https://en.wikipedia.org/wiki/Delay_line_memory

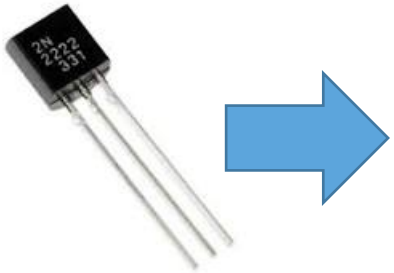
Then came the transistor

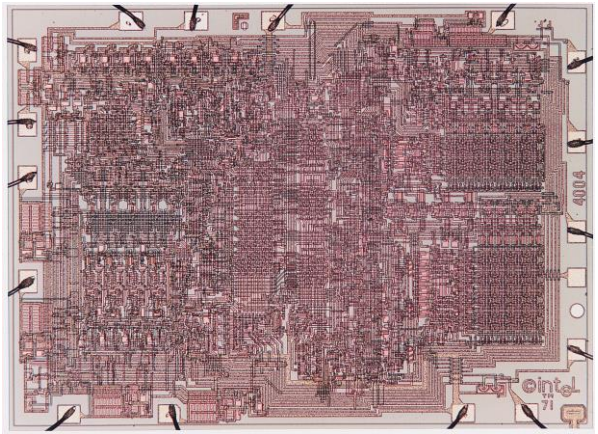
https://en.wikipedia.org/wiki/Transistor

- The symbol for a transistor
  - *Photo taken in the university where I did my masters*
- They were tiny
  - *Didn't get HOT!*
  - *Didn't break as often*



2300 of these in there

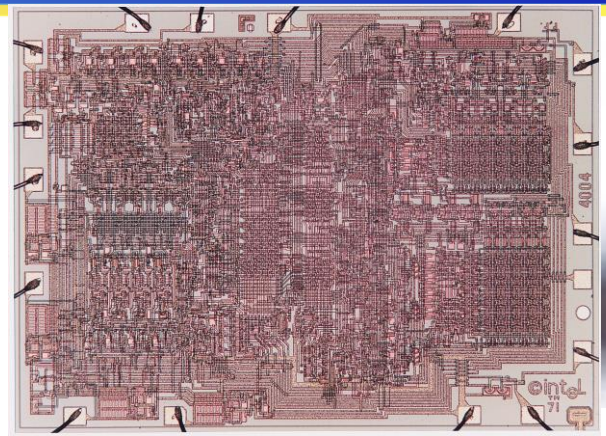


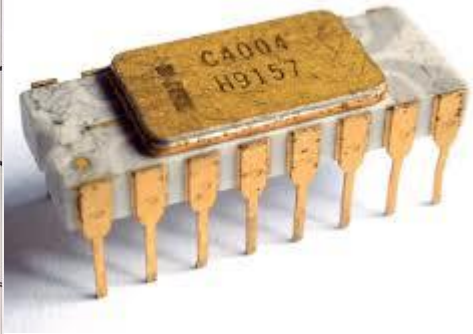


- **Things became** tiny
  - *More transistors could be fitted*
  - *Cheaper circuits*
  - *More affordable*

Extremely brief story of Intel CPUs

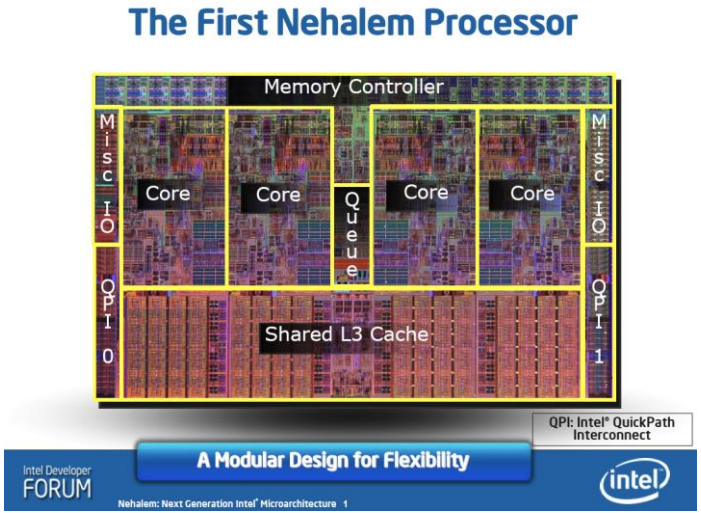
- 1971 – Intel 4004
  - 4-bit microprocessor
  - with 2300! Transistors



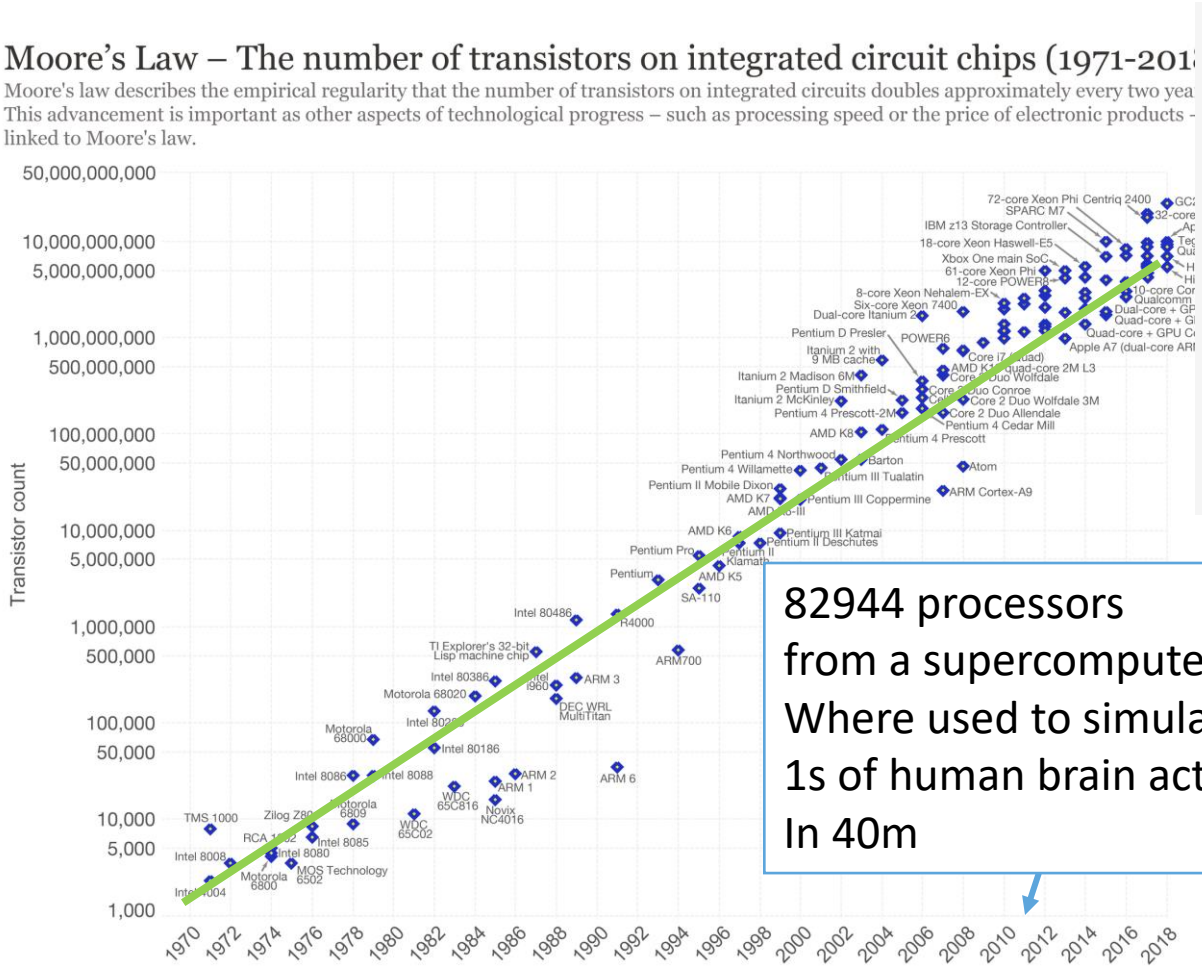


- 2004 – Pentium 4
  - x86 32-bit
  - 125 Million transistors

- 2017 – Kaby Lake
  - x86_64 64-bit
  - >1000 Million! (undisclosed?)



Moore's Law



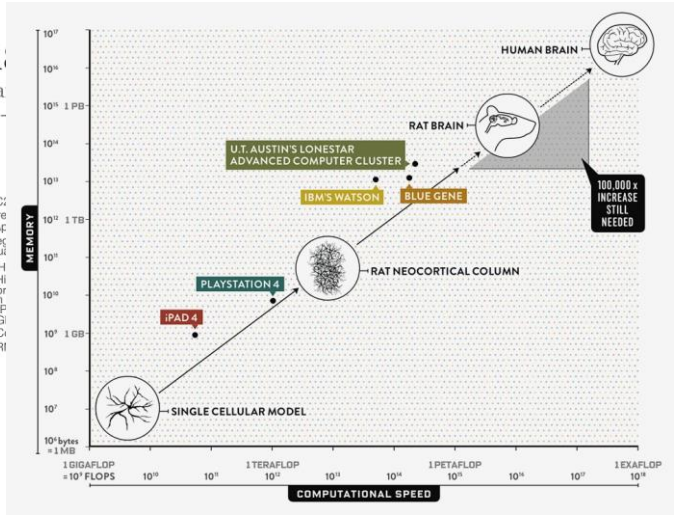


Illustration:
https://www.wired.com/2013/05/neurologist-markam-human-brain/

82944 processors
from a supercomputer
Where used to simulate
1s of human brain activity
In 40m

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

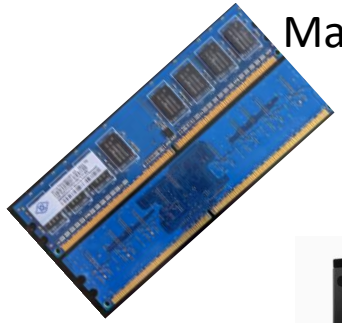
The Hardware

All different but all (mostly) the same

What hardware?

Central Processing Unit (CPU)

Main memory (RAM)



Secondary memory (Hard Drive, Solid State Drive, CD/DVD/BluRay)

Power Supply

Monitor

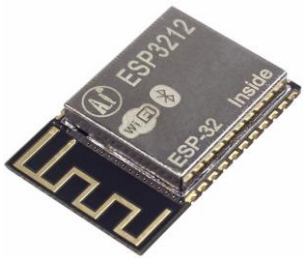
Graphics Card (Accelerators)

Motherboard

All connected

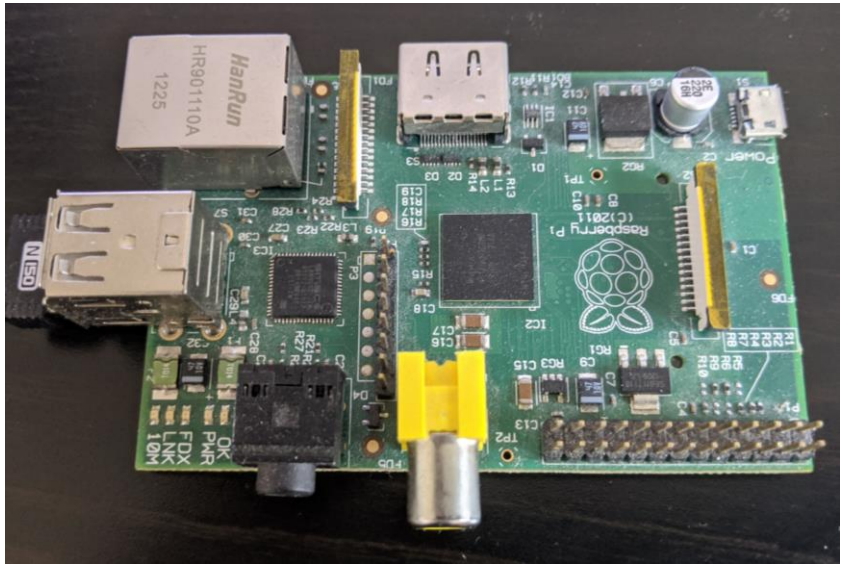
The Internet

Some are small and cheap



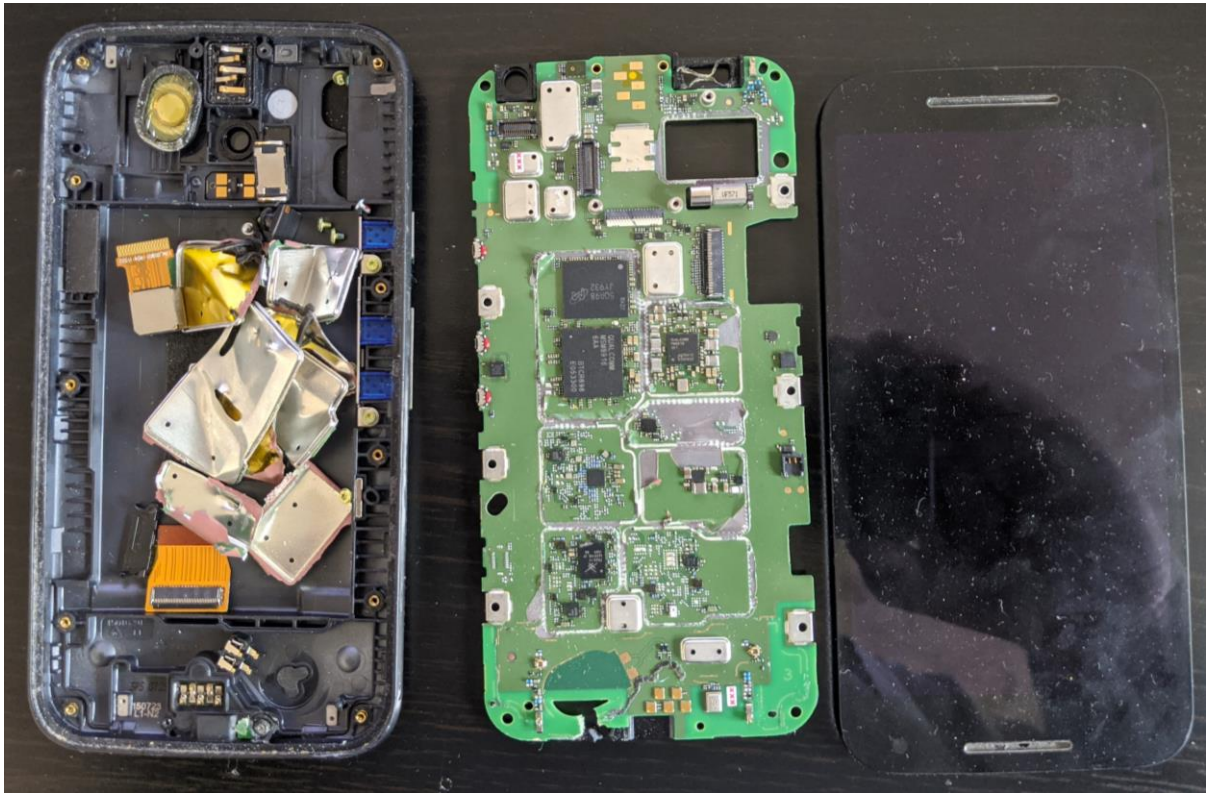
- They run a single program
  - *E.g your refrigerator*
- Are used by hobbyists
  - *For small projects*

Some are small, cheap, and powerful



- **The Raspberry Pi**
  - *Affordable, yet powerful*
  - *~$35*
  - *Can be used for A LOT of projects*
    - *Home automation*
    - *Affordable PC*
    - *Great to learn how to program on a budget*

Some are small, expensive, and powerful

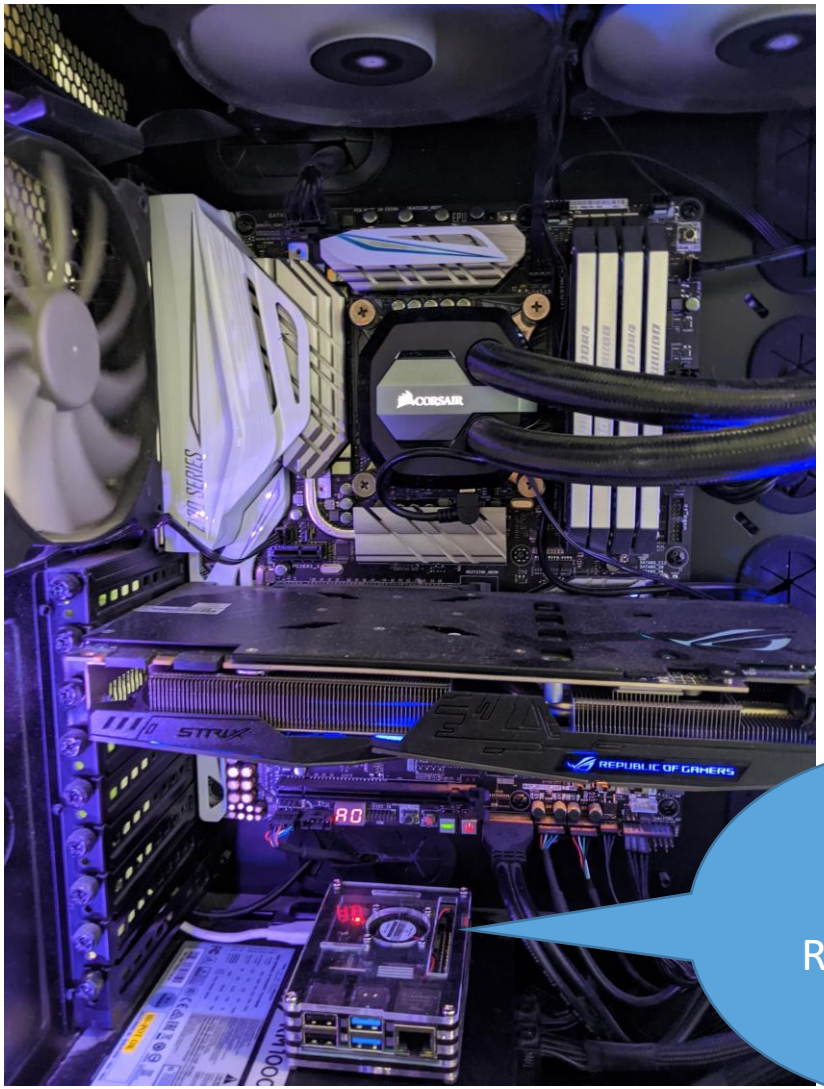


- My Moto G3 ☹
  - *I sacrificed it for you!*
    - *(battery was bloated, it had to go)*
  - *Had to rip some parts :D*
  - *Mobility is important (~1 day)*
  - *Portability is important*
  - *But it runs beefy apps!*

We hold them in our laps (does anyone do that often?)

- Power and mobility
  - Battery life is important
    - We want to fly with them ☹
  - Weight is important
  - Run demanding programs!

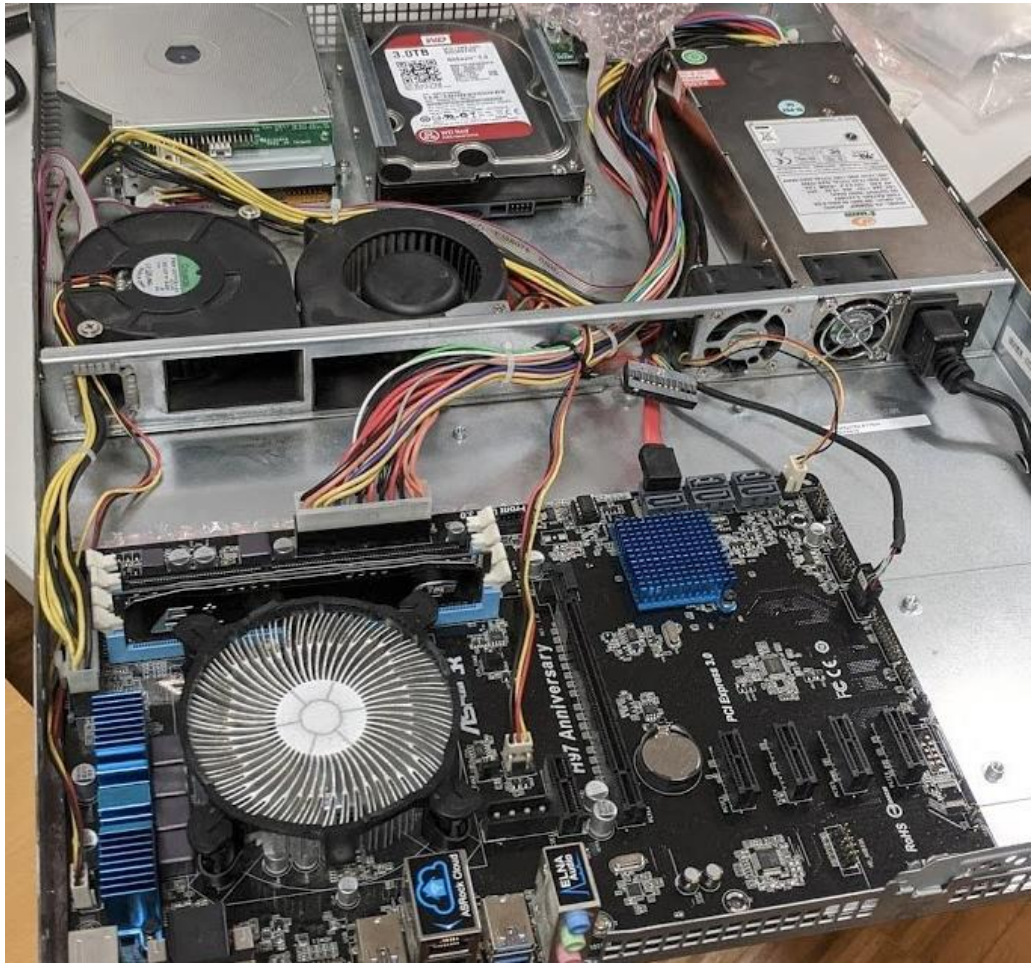
We have them at home



My computer

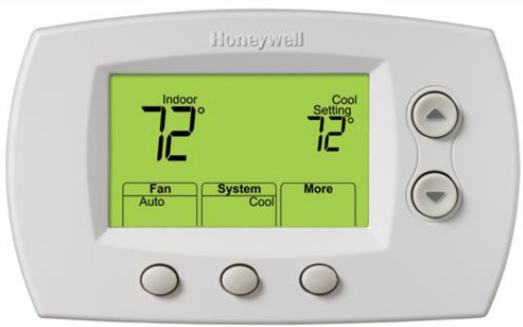
- **Desktop computers**
  - *Wide range of prices ($300 to +$5k)*
  - *Energy consumption not important*
    - *Beyond cost and heat generation*
  - *Performance*
    - *Games*
    - *Browsers!!*
    - *Word?*

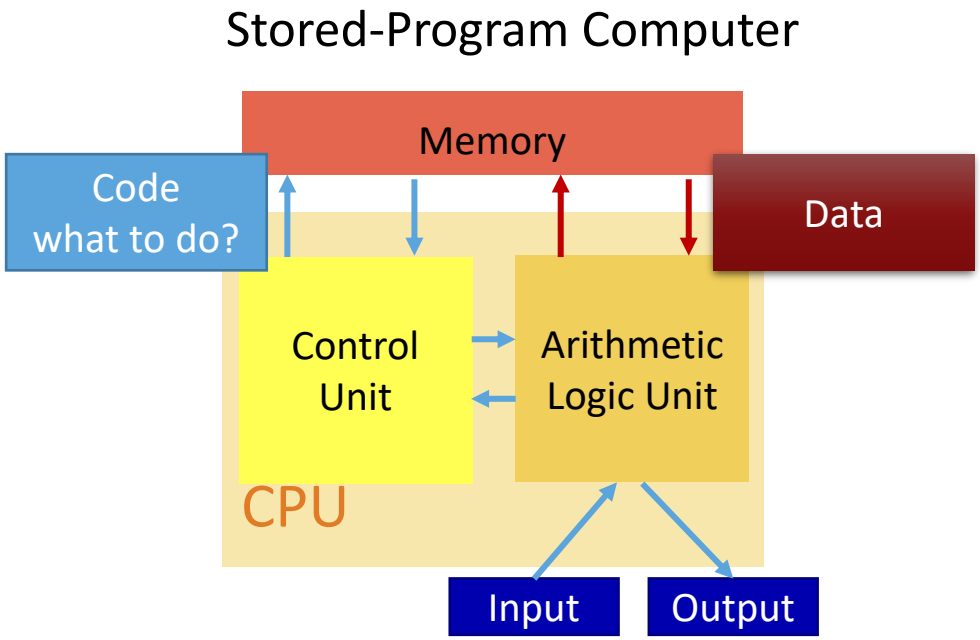
We have them in warehouses



- Server on a “drawer” (rack)
  - *Don’t have monitors*
    - *People don’t “use them” directly*
    - *Non-interactive*
  - *Crunch numbers and return results*
    - *Webpages*
    - *Remote storage (e.g., box)*

They come in all shapes and sizes





- The Von Neumann architecture
  - *Was developed for the EDVAC*
- CPU
  - *Control → Reads code & manages execution*
  - *ALU -> Performs calculations on data*
- Memory
  - *Contains information*
    - *The programs*
    - *The data*
- Inputs and outputs
  - *Connect the computer and the world*
    - *Keyboards, Disk drives, monitors, etc.*

The Hardware is hard (ah!) to change

- Once the circuits are made, there is not much you can change
  - It is still configurable and limited modifications are possible
  - Flipping switches and connecting cables (ENIAC) ☺
- But what if we want to use the computer for something new?
  - We need something flexible!
  - Something soft (ah!) and mouldable
- We need Software
  - Something the ENIAC programmers (the original computers) learned
    - Not “refrigerator ladies” → → → → → → → → → → → → → →
  - Leading to the development of programming languages
    - Famously: Grace Hopper and Betty Holberton (COBOL and Fortran)
    - *Top Secret Rosies: The Female "Computers" of WWII*

The Software

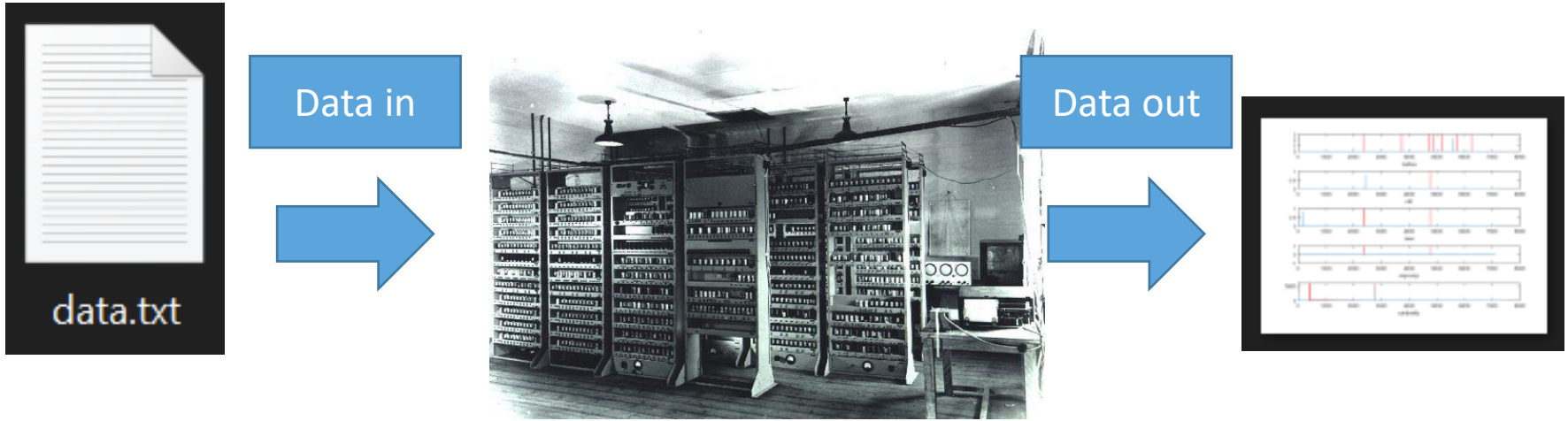
Why do we want to program?

How useful?

1 accessory

Many accessories

Machine with a single function



We want computers to do different things

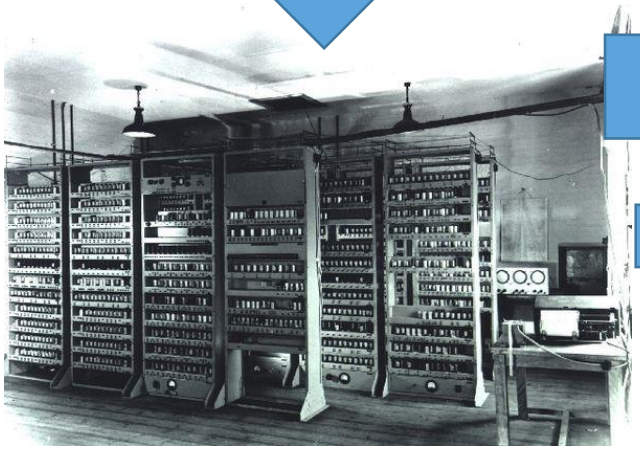
- **Computers are useful in many situations**
  - Because they are programable!

- **Computers can run different programs**
  - **Program:** A set of instructions that tell the computer what to do
  - Computers are not very smart actually, they do what programs tell them

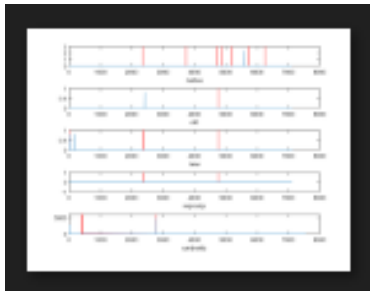
- **Examples of software:**
  - Windows, OSX, Linux – Operating Systems (OS) that manage your computer
  - Word, Firefox, Animal Crossing – Applications ran by the user

Programmable machine

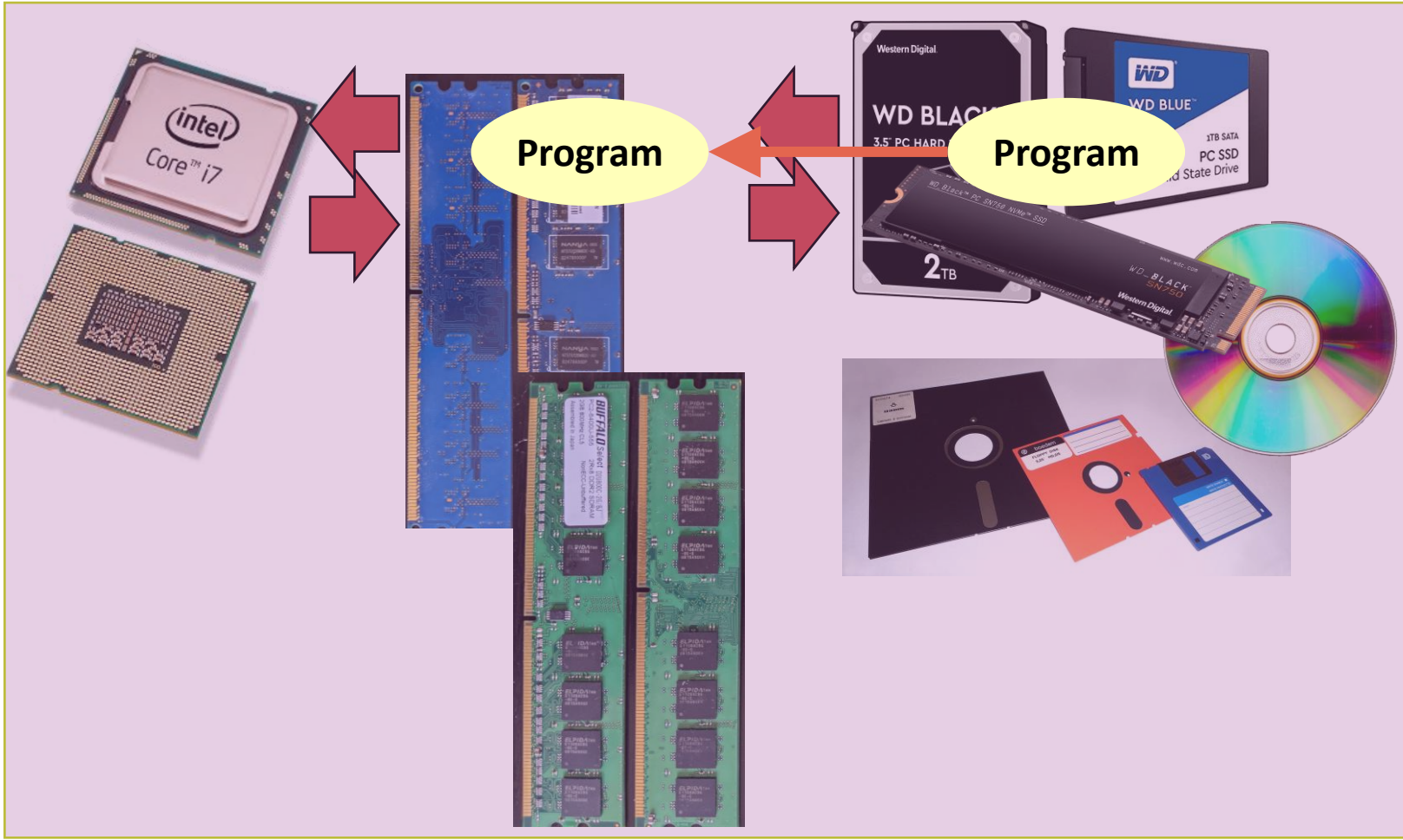
Data in



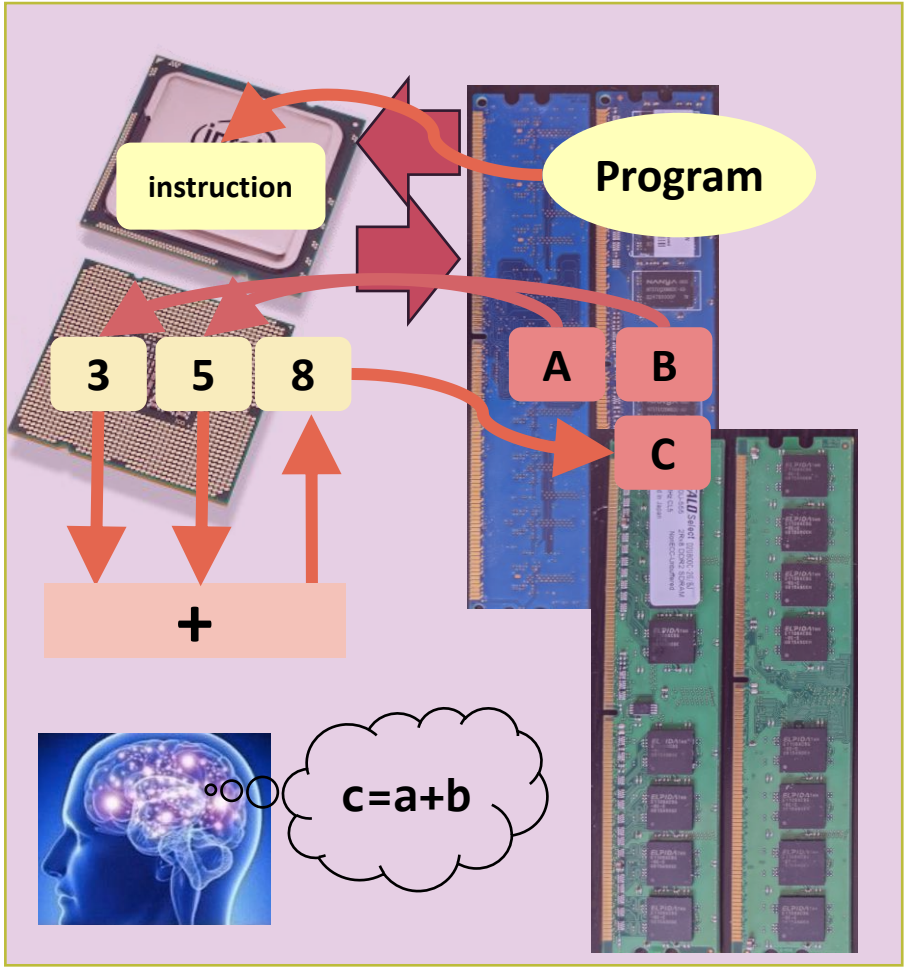
Data out



How a program runs



How a program runs



- Fetch
  - *Read from memory the next instruction (code)*
  - *I.e., what is next thing to do*
- Decode
  - *What does the instruction want me to do?*
  - *E.g., I want to add two numbers (A and B)*
- Execute
  - *Do the operation*
  - *E.g. add*
- Repeat

But how do we do it?

- To program a computer we must learn to speak (one of) its language(s)

- But importantly, we need to learn how to program.

What's the difference?

- Know words you do, right? ☺
  - But can you write a 1000 pages novel?
    - We'll aim a bit lower, maybe a couple of 100 pages?

Good news!!!

- The Java language uses patterns very similar to other languages
  - That's because those patterns serve programmers well.
- Other languages are not the same!
  - But they will be similar enough
- Once you learn those patterns, picking up a different language is easier
  - Why would you?
  - Would you use an atomic bomb to kill a fly?