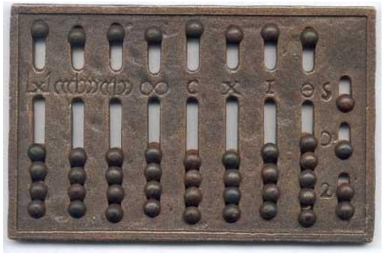
COE 530
Quantum Computer And Architecture

Lecture 1

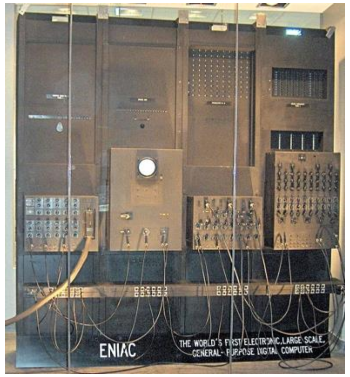
Classical Computer System I

References:
The Physics of Computing, Myrlin Wold
CMU15-213 Introdcution to Computer Systems
MIT 6-823 Computer System Architeture

History of computing - (From the Abacus to HPC)



Abacus
~ 1300AD



ENIAC
~1945

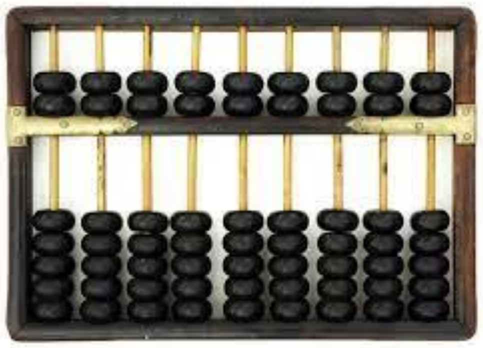
SHAHEEN III
~2025

Analytical Enging
~ 1830s

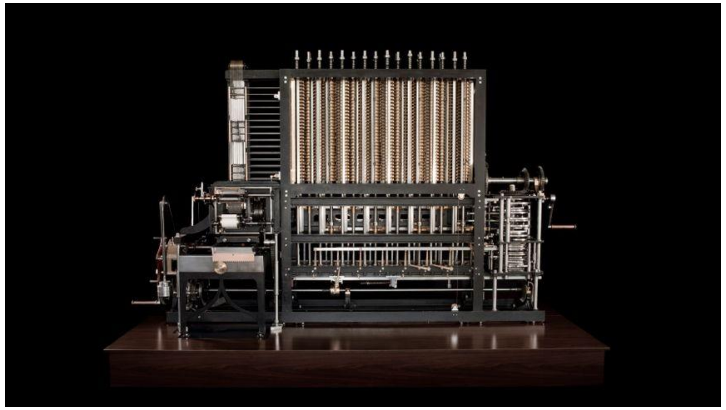
PCs and Laptops
~1990s

Mechanical Computers

- Early efforts focused on specific-purpose computation, i.e., calculation
- First known calculator is the *Abacus* in the 12th century
- *In the 19th century, the use of mechanical calculators was common*
  - *Hamann Manus R (link)*
- First concept of “general-purpose” mechanical computers dated back to 1810s by Charles Babbage and Ada Lovelace (The Analytical Engine)

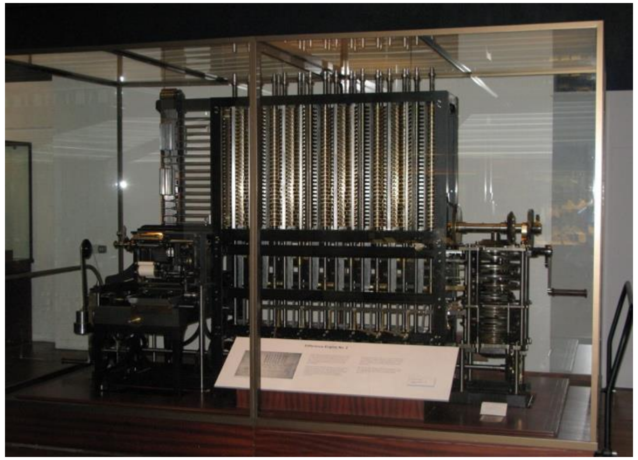


Hamann Manus R



Difference and Analytic Engine

- Charles Babbage designed the Difference Engine in 1823
- Idea:
  - Any continuous function can be approximated by a polynomial
  - Any Polynomial can be computed from *difference* tables
- Example:



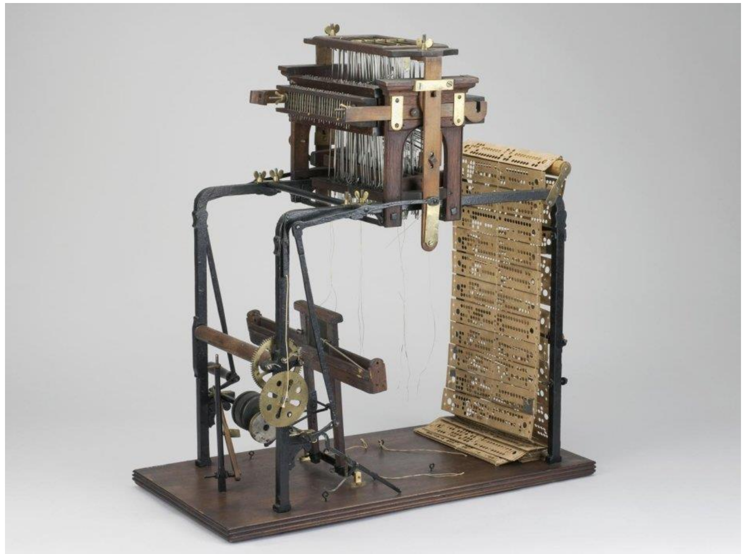
https://en.wikipedia.org/wiki/Difference_engine

n	**0**	**1**	**2**	**3**	**4**
$d_2(n)$			2	2	2
$d_1(n)$		2	4	6	8
$f(n)$	41	43	47	53	61

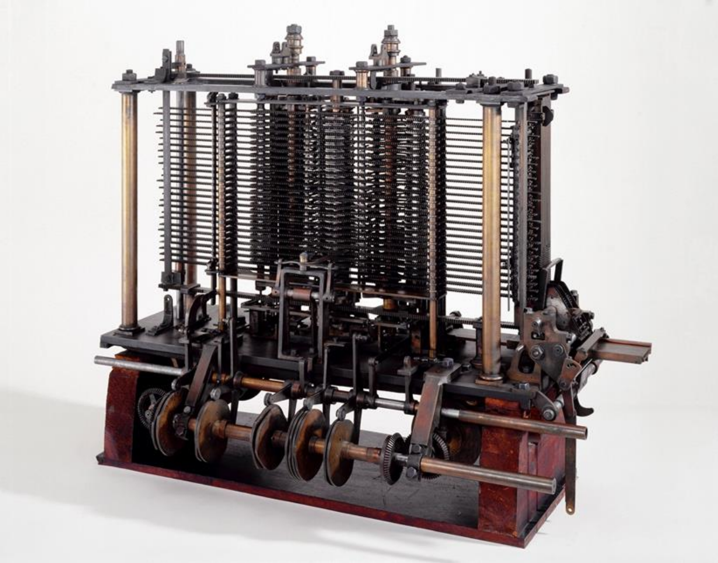
Any continous function can be implemented using an adder!

Analytical Engine

- Charles Babbage (again) proposes the Analytical Engine
- Inspired by the Jacquard Looms
  - looms were controlled by punched cards
    - The set of cards with fixed punched holes dictated the pattern of weave ⇒ program
    - The same set of cards could be used with different colored threads ⇒ numbers
- The analytical engine demonstrates the first design for a general-purpose computer (a.k.a Turing complete?)
- It incorporated an <u>arithmetic logic unit</u>, <u>control flow </u>in the form of conditional branching and loops, and <u>integrated memory</u>



https://www.scienceandindustrymuseum.org.uk/objects-and-stories/jacquard-loom



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

Theory of Computing

- Computing started with Mathematician (Logician)
- Goerge Boole introduced logical expression in Boolean algebra
- Boolean algebra describes the basic logical operations such as AND, OR, and NOT
- The logical operations can be expressed by a Truth Table

A	Z
0	1
1	0

NOT
Truth Table

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

OR
Truth Table

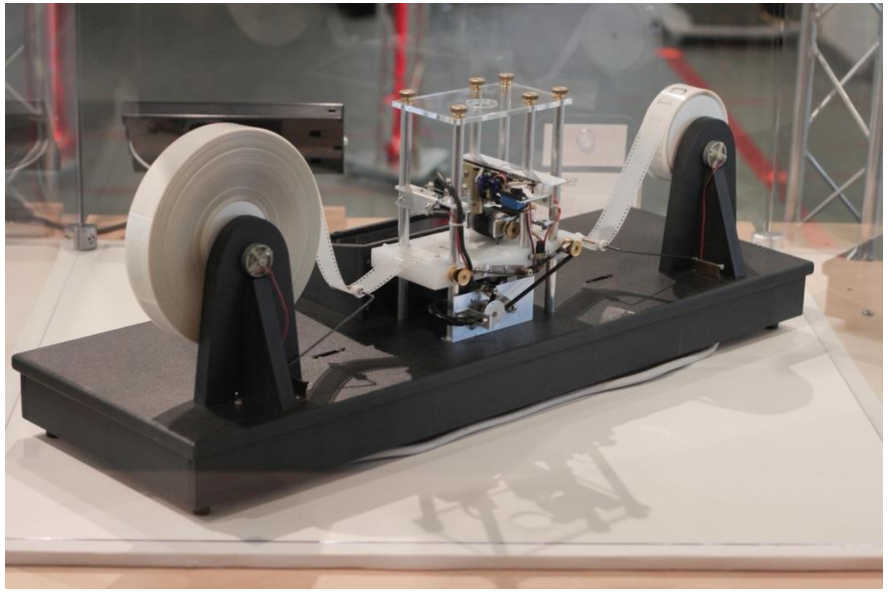
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

AND
Truth Table

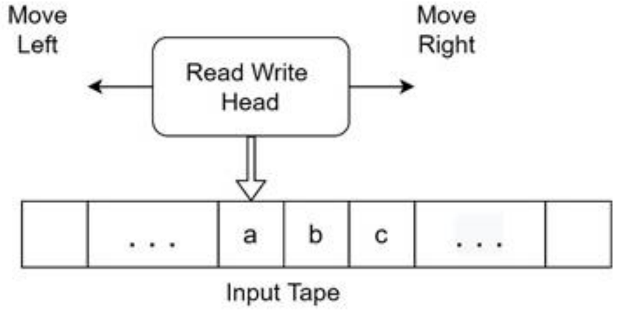
- The questions is – What functions could be calculated?

Turing Machine

- In 1930s, Alonzo Church developed a theory of computing using lambda terms to manipulate variables
- Later, this was turned into functional programming languages, such as LISP, Prolog, and Scheme
- Following Church theory, Alan Turing developed a theory of computing called **Turing machine**
- Turing machine is a hypothetical machine with the following elements
  - An infinite tape diviced into cells hat can hold discrete values
  - A head that can read and write cells as well as move the tape in both directions
  - A set ofruls in the head that tell it what to do based on the values of the cell currently being read



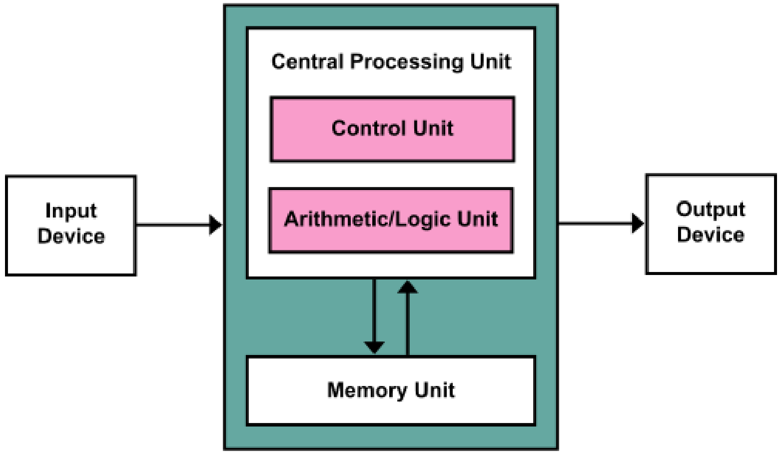
https://en.wikipedia.org/wiki/Turing_machine



https://www.tutorialspoint.com/automata_theory/representation_of_turing_machine.htm

From Theory to Machines

- Turing proved that his machine was equivalnet to Church’s lambda calculues, i.e., both can compute the same set of functions
- However, Turing machine introduced the modern concept of electronic computers
  - The tape corrospond to computer’s memory
  - The head corrospond to CPU
  - The directions corrosponds to the program
- Around the same time, John Von Neumann proposed a computer architecutre that includes
  - CPU
  - Memory
  - Input/output devices
- How is information stored?

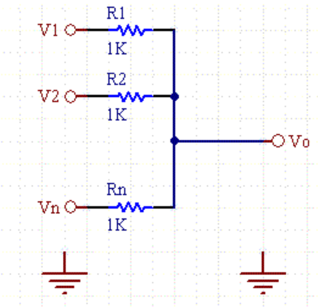


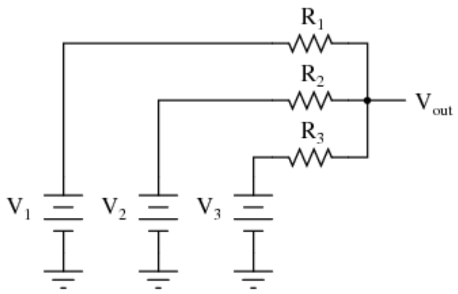
https://en.wikipedia.org/wiki/Von_Neumann_architecture

Analog Computers

- Before digital computers there were analog computers.

- Consider a couple of simple analog computers:
  - A simple circuit can allow one to adjust voltages using variable resistors and measure the output using a volt meter
  - A simple network of adjustable parallel resistors can allow one to find the average of the inputs





https://www.daycounter.com/Calculators/Voltage-Summer/Voltage-Summer-Calculator.phtml

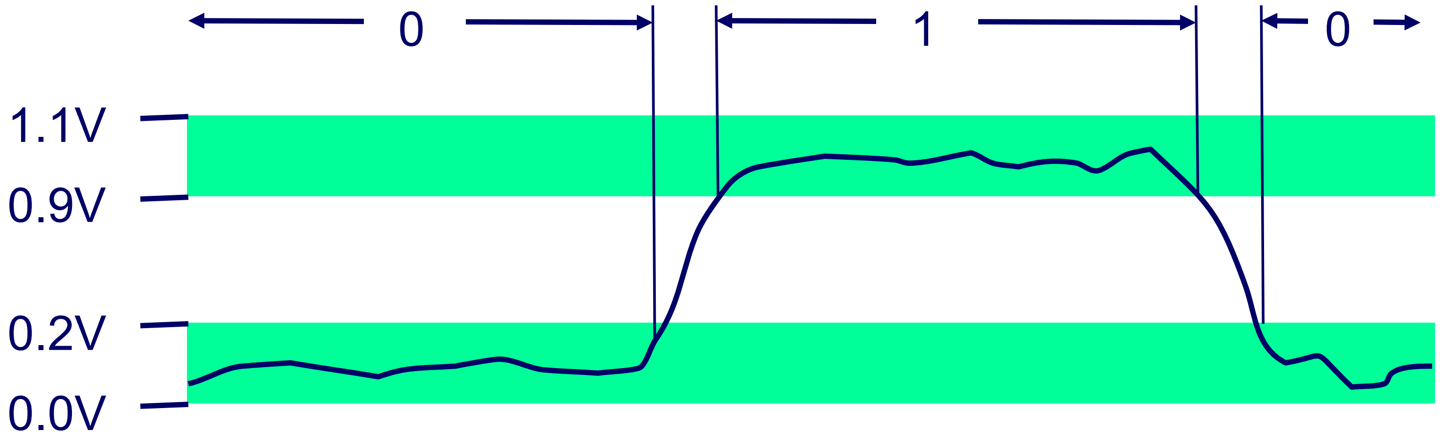
https://www.quora.com/What-is-the-most-basic-voltage-adder-circuit-without-a-transistor-op-amp-and-any-external-supply

From Analog to Digital

- We don’t try to measure exactly
  - We just ask, is it high enough to be “On”, or
  - Is it low enough to be “Off”.

- We have two states, so we have a binary, or 2-ary, system.
  - We represent these states as 0 and 1

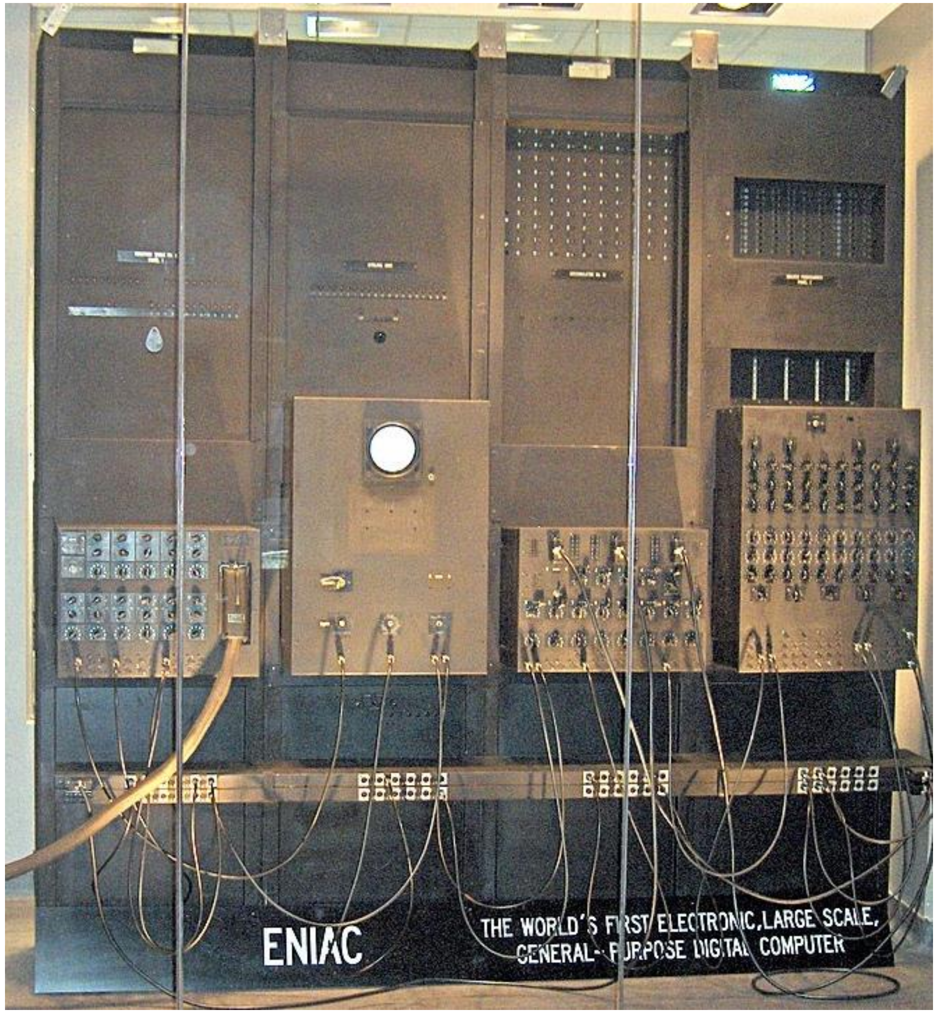
- Now we can easily interpret, communicate, and duplicate signals well enough to know what they mean.



This slide is from CMU course

Electronic Numerical Integrator and Computer (ENIAC)

- Inspired by Atanasoff and Berry, Eckert and Mauchly designed and built ENIAC (1943-45) at the University of Pennsylvania
- The first, completely electronic, operational,
  - general-purpose analytical calculator!
  - 30 tons, 72 square meters, 200KW
- Performance
  - Read in 120 cards per minute
  - Addition took 200 µs, Division 6 ms
- ENIAC's programming system was external
  - Sequences of instructions were executed independently of the results of the calculation
  - Human intervention required to take instructions "out of order"



This slide is from MIT course

Electronic Discrete Variable Automatic Computer (EDVAC)

- Eckert, Mauchly, John von Neumann and others designed EDVAC (1944) to solve this problem
  - Solution was the *stored program computer* ⇒ *"program can be manipulated as data"*

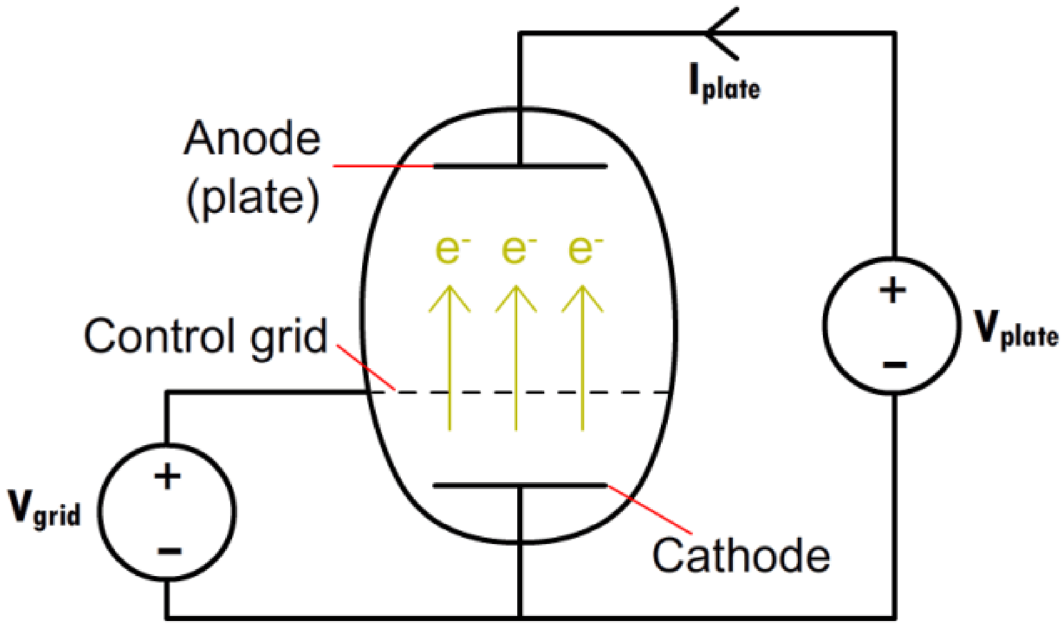
Program = A sequence of instructions

ENIAC	⇒	EDVAC
18,000 tubes		4,000 tubes
20 10-digit numbers		2000 word storage
		mercury delay lines

Let's pause here and talk about the hardware

Vacuum Tube

- Components
  - Cathode (Emitter): Heated to release electrons
  - Anode (Collector): Positively charged to attract the emitted electrons
  - Grid (Base): A mesh placed between cathode and anode
- How It Works
  - Electron Emission: The filament heats the cathode, causing it to emit a cloud of electrons
  - Electron Flow: These electrons are attracted to the positive anode (or plate).
  - Control (Amplification/Switching):
    - A small voltage applied to the grid creates an electric field that either attracts or repels the electrons.
    - A negative grid voltage pushes electrons back, reducing current (turn off)
    - A less negative or positive grid voltage allows more electrons through (turn on)
    - This allows a weak input signal on the grid to control a strong output current, enabling amplification or using the on/off states for logic.



https://www.engineering.com/vacuum-tubes-the-world-before-transistors/

Limitations:

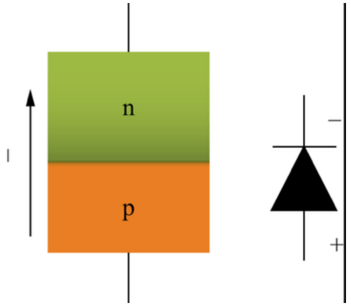
Size

Consume large amount of power

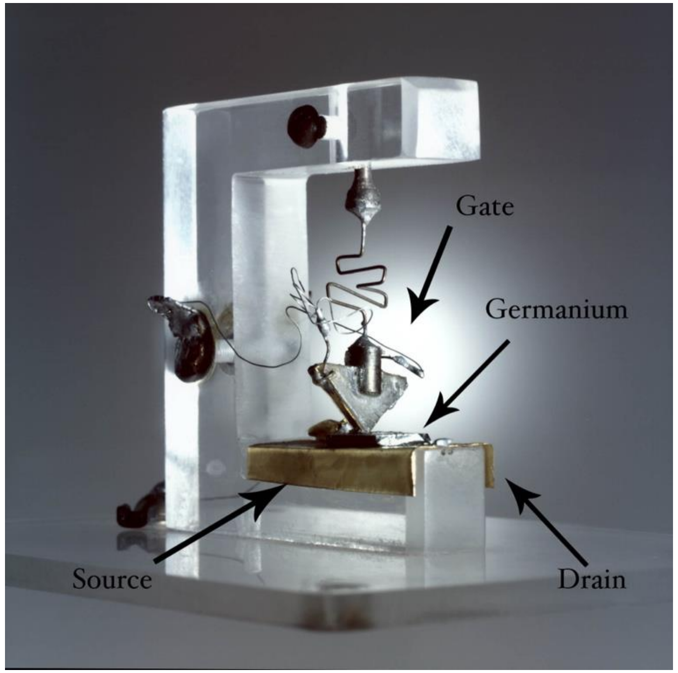
Not reliabe

Semiconductor Devices

- A group of scientists at Bell Laboratories decided to look into semiconductors as an alternative to vacuum tubes
- A semiconductor diode is made is built by putting together two differently doped pieces of silicon, one n-type and one p-type

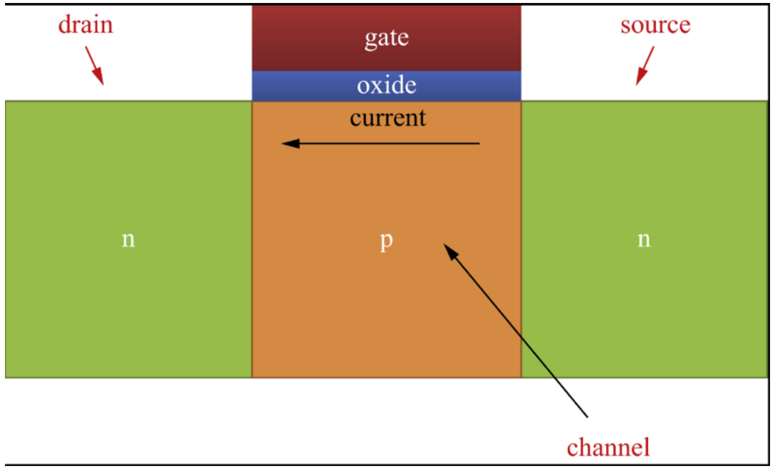


- The type of transistors that now dominates computer design is called Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET)

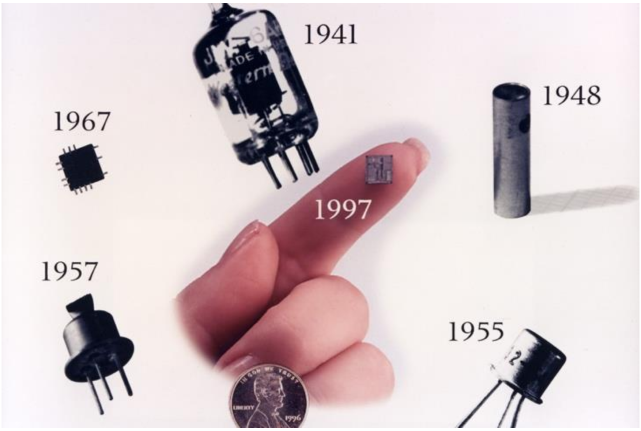


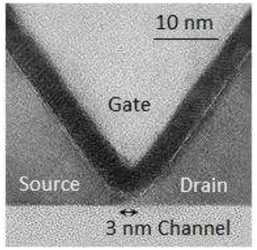
https://jessalapulapu.wordpress.com/wp-content/uploads/2015/07/first-transistor.jp

MOSFET

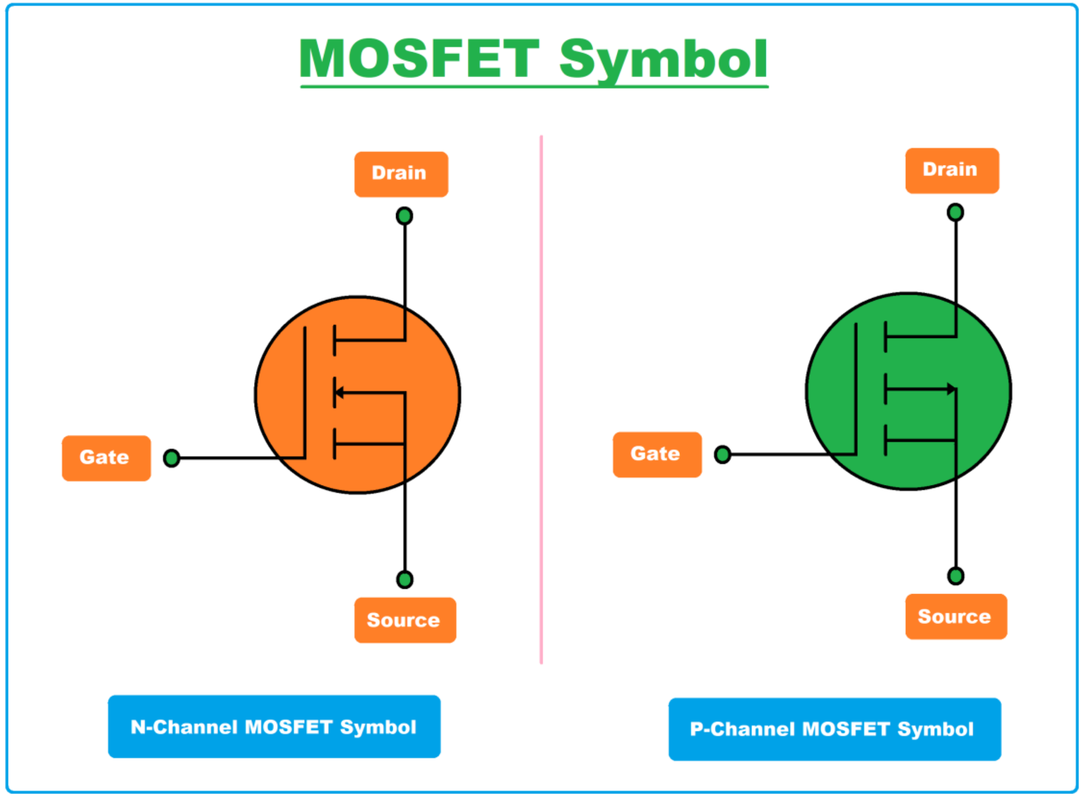


N-type MOS transistor





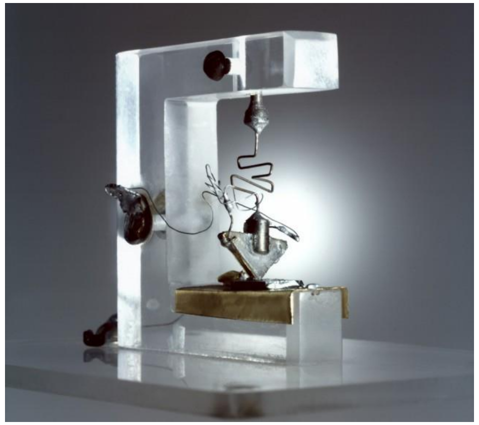
https://phys.org/news/2013-02-success-transistor-channel-length-nm.html

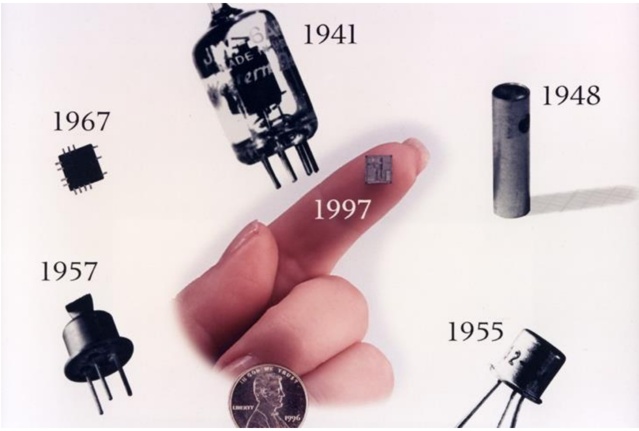


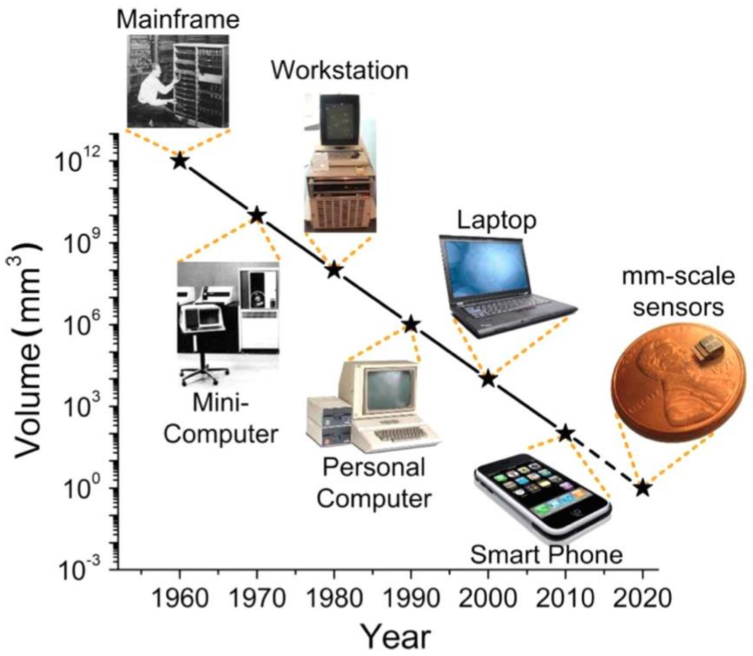
https://images.theengineeringprojects.com/image/main/2018/02/MOSFET-Symbol.png

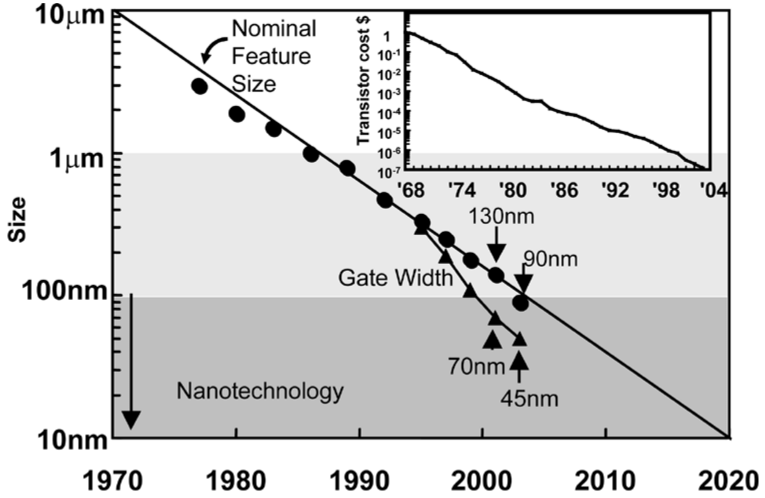
Transistors and Computers

- Since then, the main goals of computer architecture revolve around the tradeoff between
  - cost (size, components, $$), and
  - performance (speed, efficiency, accuracy)
- This was possible thank to advancement in manufacturing smaller electronics (particularly transistors)

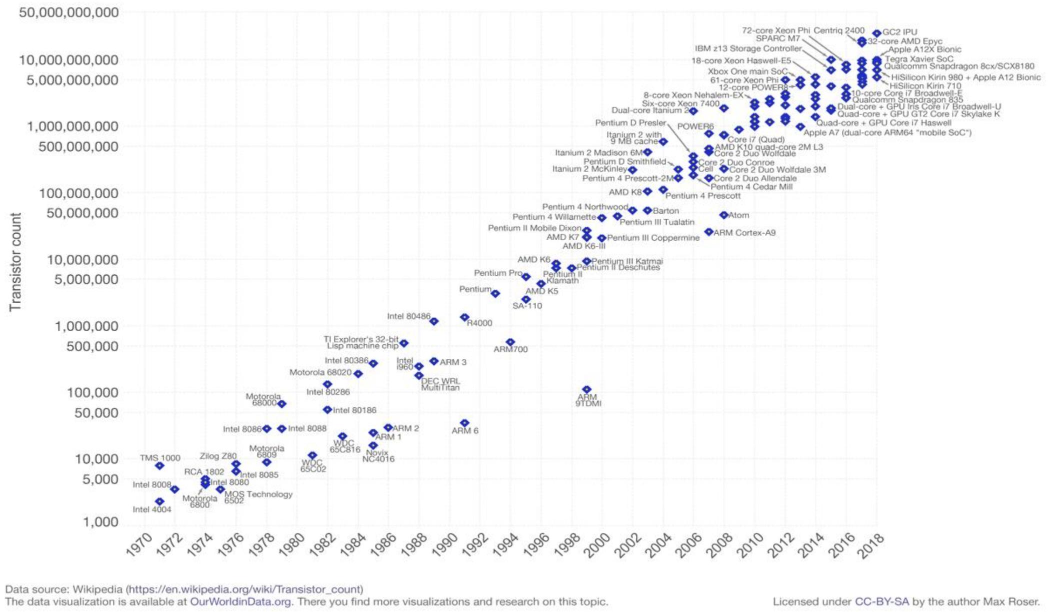








Moore’s Law and Future of Computing



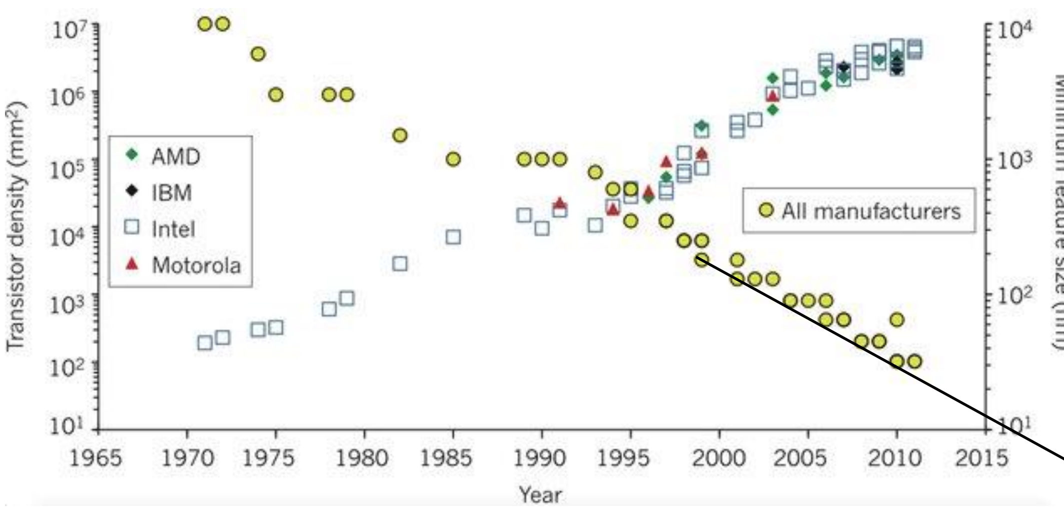
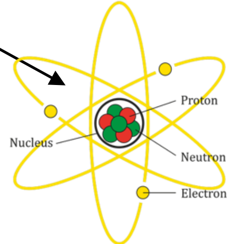


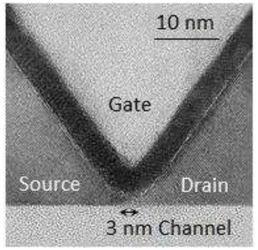
Figure 7. Transistor density (number of transistors/mm 2) and minimum feature size (mm) in microprocessor units.



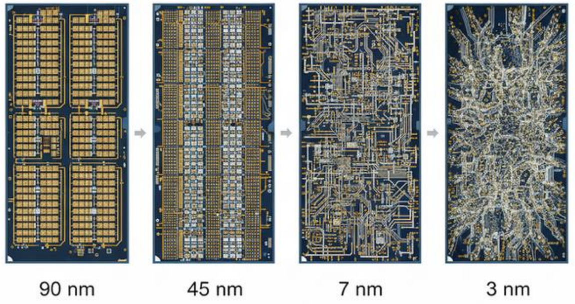
Atom size transistors

Moore's law is the observation that the number of transistors in a dense integrated circuit doubles about every two years.

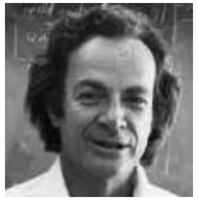
Moore’s law will soon run into major physical constraints!



https://phys.org/news/2013-02-success-transistor-channel-length-nm.html



https://medium.com/@usamahmoin/why-nanometers-matter-the-real-story-behind-chip-size-and-power-d5e3af8598d6



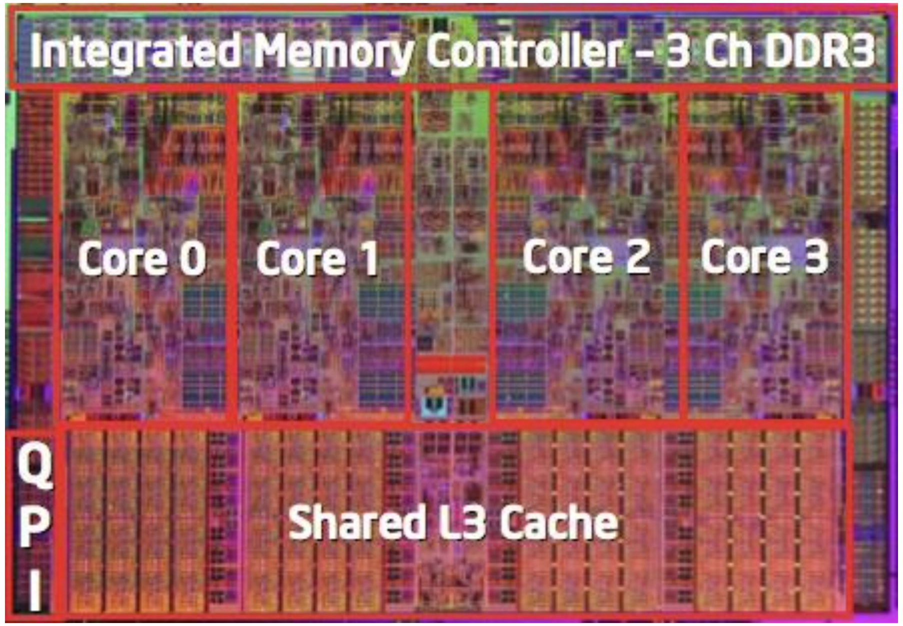
Richard Feynman

“There is plenty of room at the bottom”

Intel x86 Processors, cont.

- Machine Evolution

• 386	1985	0.3M
• Pentium	1993	3.1M
• Pentium/MMX	1997	4.5M
• PentiumPro	1995	6.5M
• Pentium III	1999	8.2M
• Pentium 4	2000	42M
• Core 2 Duo	2006	291M
• Core i7	2008	731M
• Core i7 Skylake	2015	1.9B



- Added Features
  - Instructions to support multimedia operations
  - Instructions to enable more efficient conditional operations
  - Transition from 32 bits to 64 bits
  - More cores

This slide is from CMU course

Intel x86 Processors, cont.

- Past Generations

		Process technology
1st Pentium Pro	1995	600 nm
1st Pentium III	1999	250 nm
1st Pentium 4	2000	180 nm
1st Core 2 Duo	2006	65 nm

- Recent & Upcoming Generations

Nehalem	2008	45 nm
Sandy Bridge	2011	32 nm
Ivy Bridge	2012	22 nm
Haswell	2013	22 nm
Broadwell	2014	14 nm
Skylake	2015	14 nm
Kaby Lake	2016	14 nm
Coffee Lake	2017	14 nm
Cannon Lake	2018	10 nm
Ice Lake	2019	10 nm
Tiger Lake	2020	10 nm
Alder Lake	2022	“intel 7” (10nm+++)

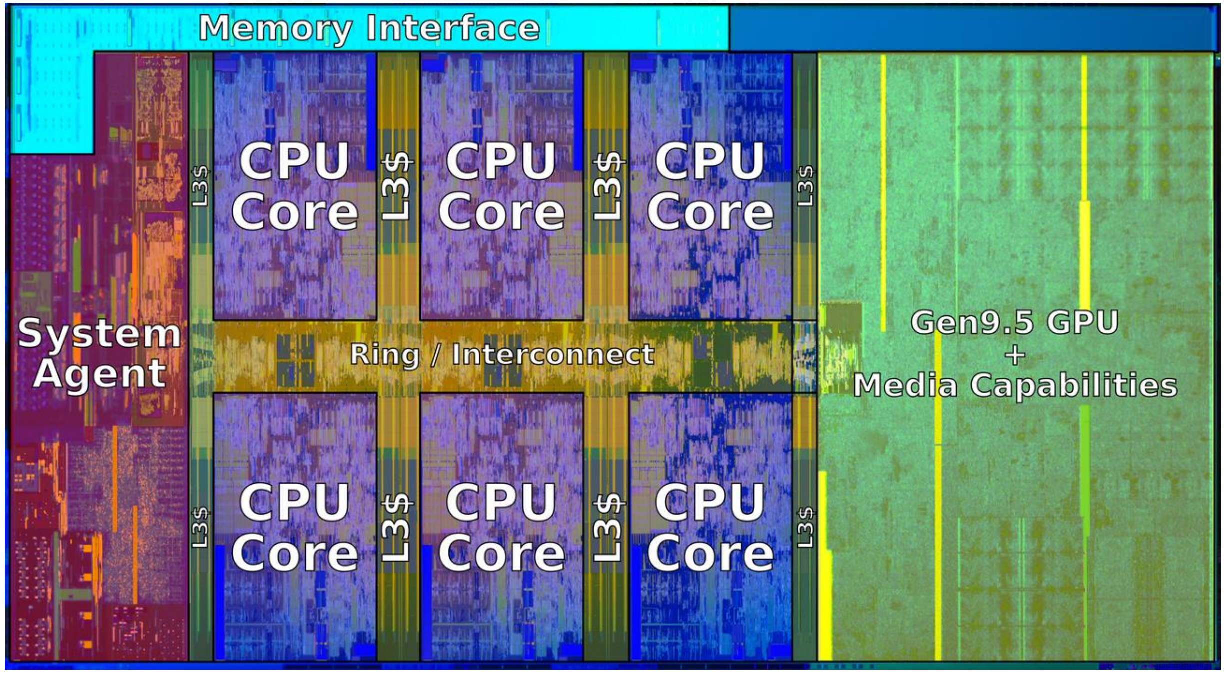
Process technology dimension
= width of narrowest wires
(10 nm ≈ 100 atoms wide)

(But this is changing now.)

This slide is from CMU course

2018 State of the Art: Coffee Lake

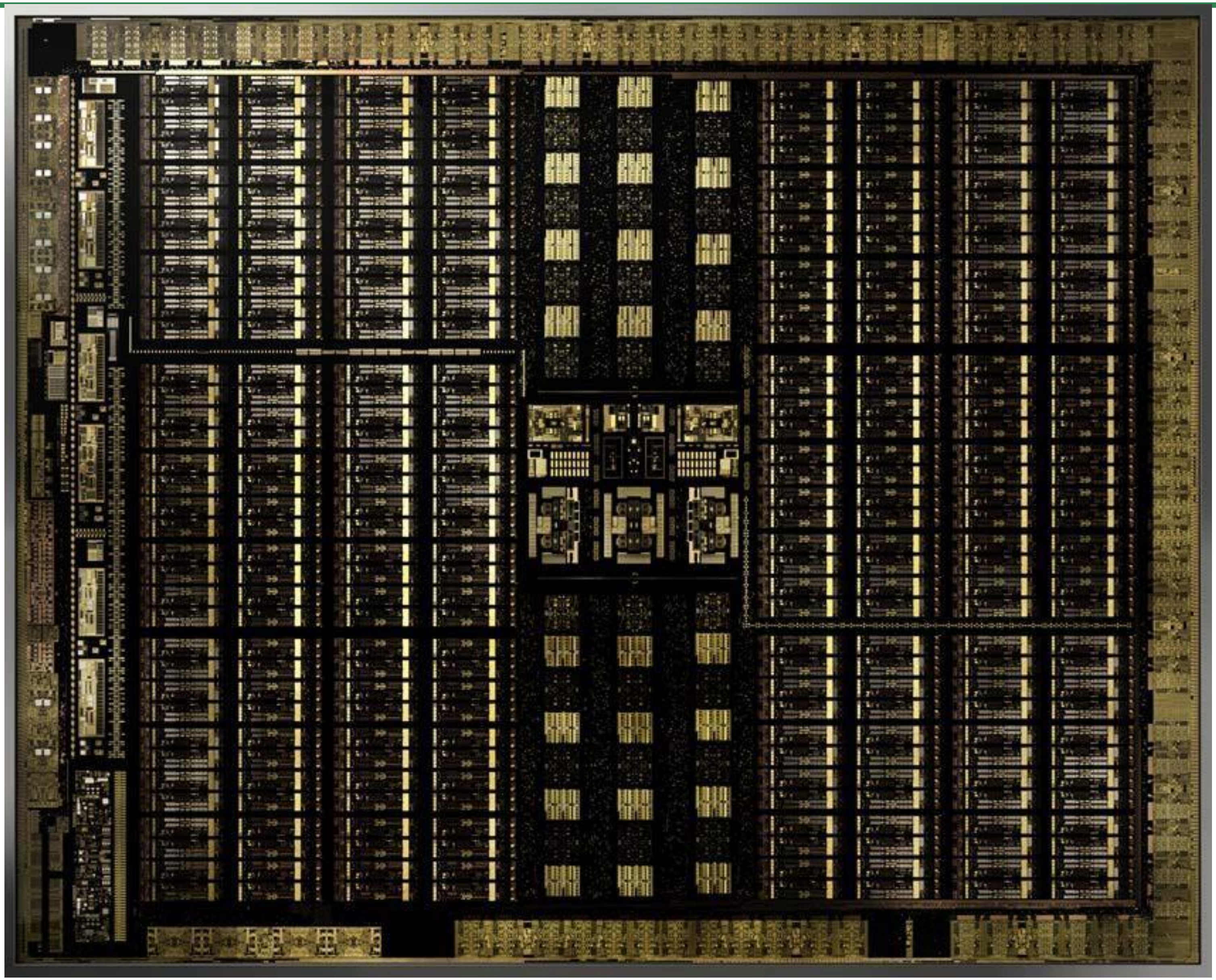
- Mobile Model: Core
  - 2.2-3.2 GHz
  - 45 W



- **Desktop Model: Core i7**
  - Integrated graphics
  - 2.4-4.0 GHz
  - 35-95 W

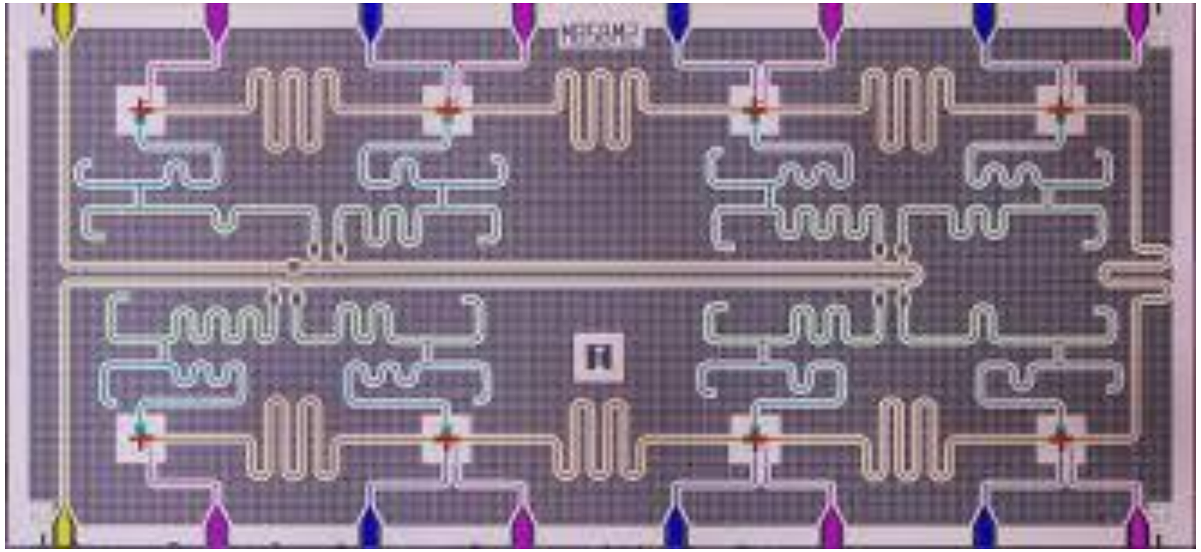
- **Server Model: Xeon E**
  - Integrated graphics
  - Multi-socket enabled
  - 3.3-3.8 GHz
  - 80-95 W

GPU Chips

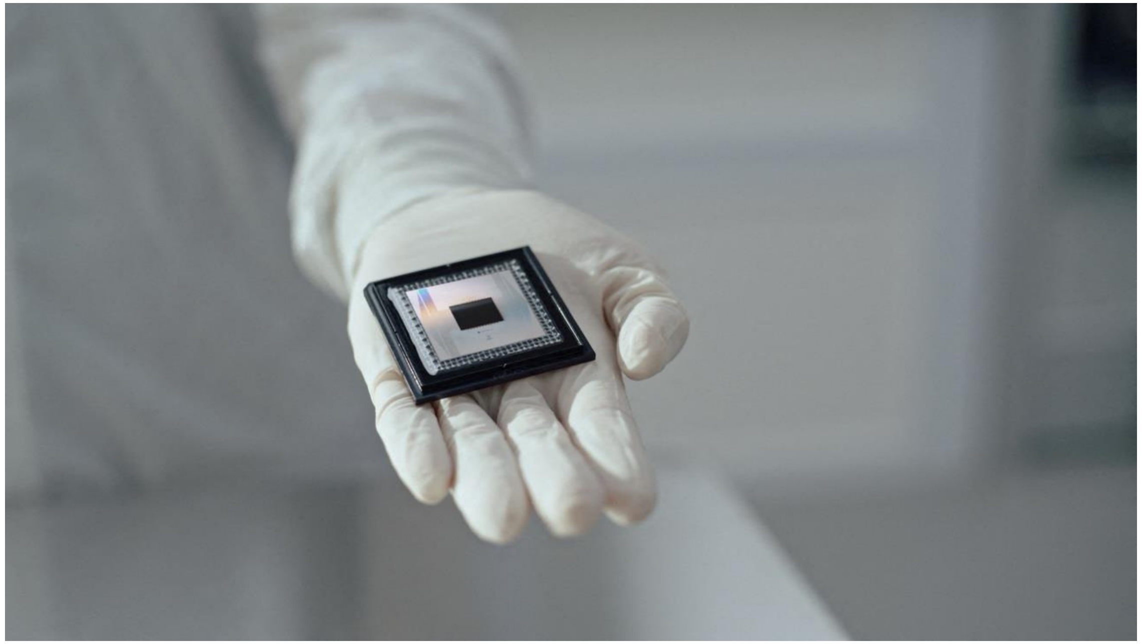


https://developer.nvidia.com/blog/nvidia-turing-architecture-in-depth/

How about Quantum Chips?



https://www.researchgate.net/figure/False-coloured-image-of-an-8-qubit-superconducting-quantum-processor-fabricated-at-ETH_fig3_327045512



https://www.npr.org/2024/12/11/nx-s1-5223486/google-new-chip-quantum-computing

Enough with Hardware, How About Software?

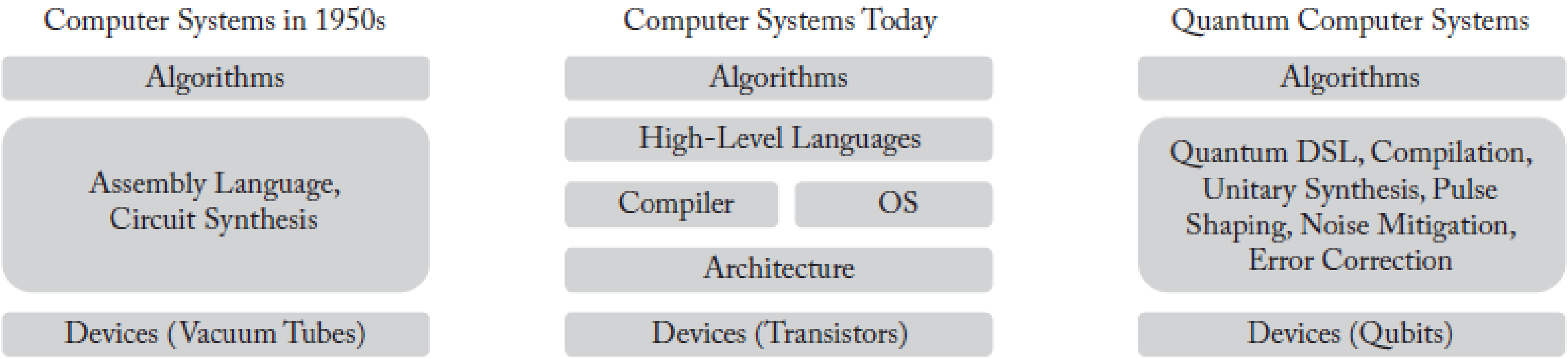


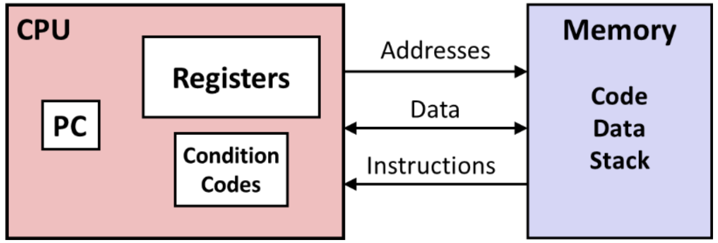
Figure 1.2: Architectural designs of classical vs. quantum computers. The abstraction layers for 1950s classical computing, today's classical computing, and quantum computing are compared.

Levels of Abstraction

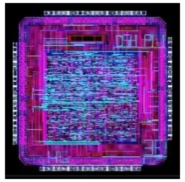
C programmer

```
#include <stdio.h>
int main(){
  int i, n = 10, t1 = 0, t2 = 1, nxt;
  for (i = 1; i <= n; ++i){
    printf("%d, ", t1);
    nxt = t1 + t2;
    t1 = t2;
    t2 = nxt; }
  return 0; }
```

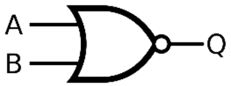
Assembly programmer

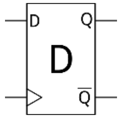


Computer Designer



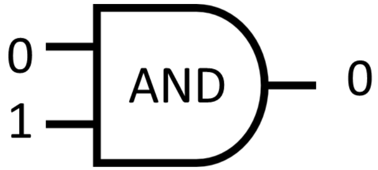
Gates, clocks, circuit layout, …

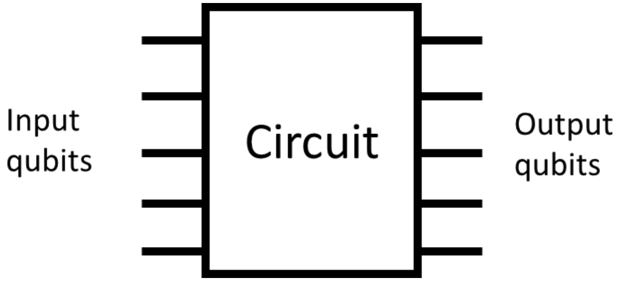


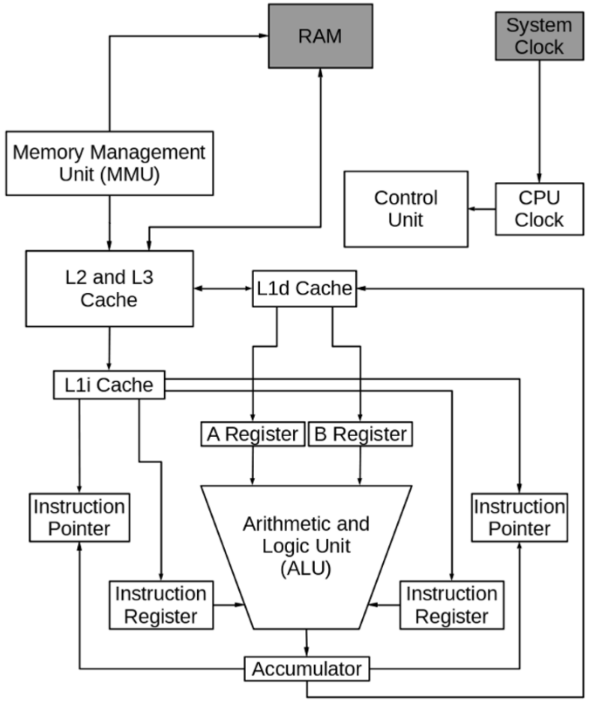


Next -- From Bits to Computers

- Conceptual models of computation
  - Turing machines
  - λ-calculus
  - Circuit model
- These models are equivalent
- Computer architecture follows the circuit model of computation
- Next, we will learn how to build a simple CPU starting from transistors!
- In PHYS514, you should learn how to build a simple qubit!







https://www.redhat.com/sysadmin/cpu-components-functionality