

A Brief History of Computing

Before and After WWII



Last Time

- Course overview
- Course logistics
- Readings for week
 - Blown to Bits, Chapter 1
 - Begin “What is Code?”

Reminders

- Complete Academic Integrity
 - OLI Module & Quiz
- Academic Integrity form to be read, signed, and returned tomorrow.
Available on the course web page.

What should we review?

- Course Website?
- Course Schedule?
- Tools: Piazza, Gradescope, Autolapb
- Grading and other policies?
- Academic Integrity: OLI Module completed; form to be read, signed, and returned Thursday.
Available on the course web page.

Today's Lecture

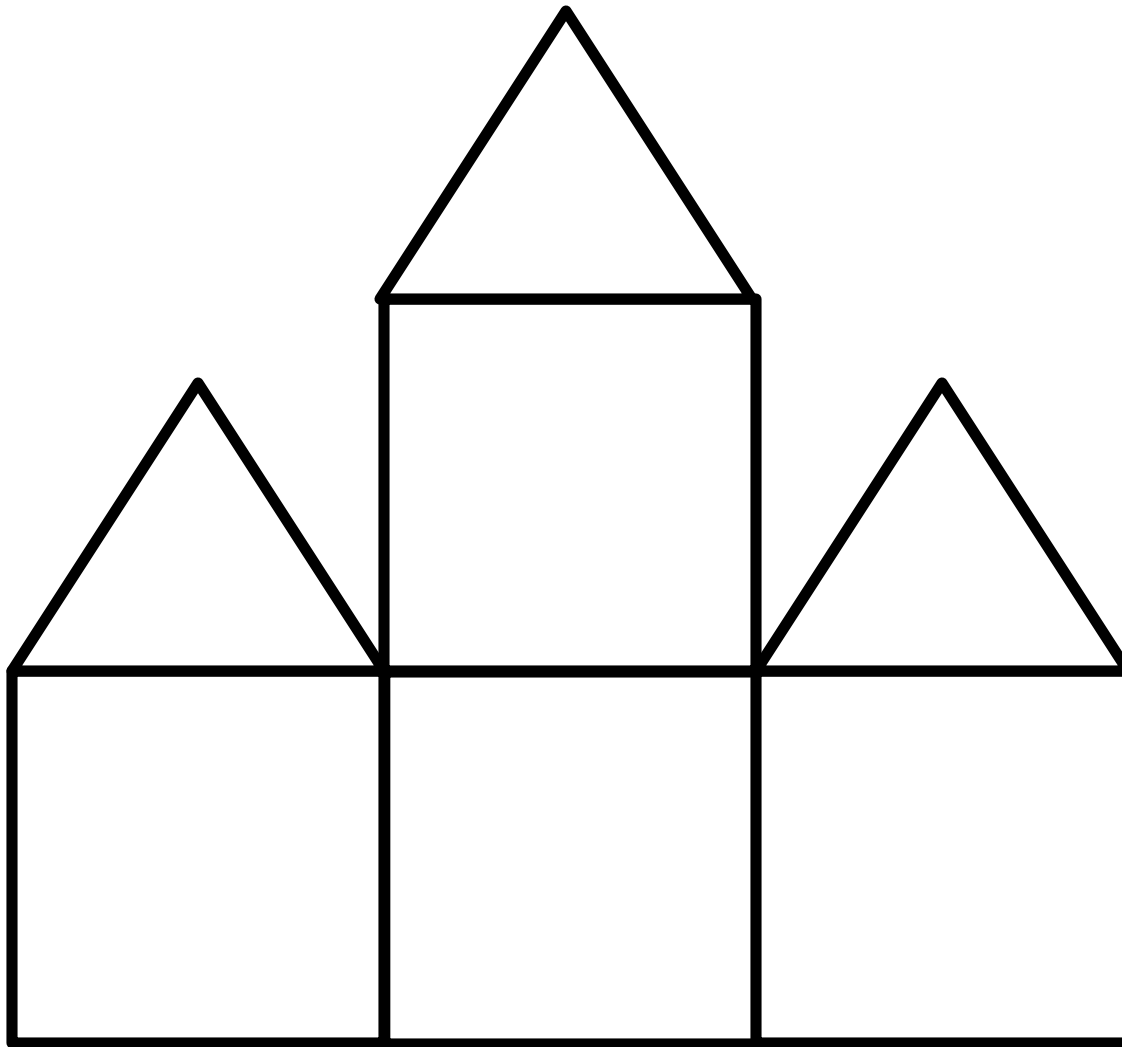
- ▣ Questions and Review
- ▣ A brief history of computing
 - ▣ Major milestones, big ideas
 - ▣ The interplay between technological developments and social, political factors.
- ▣ Very brief introduction to Python

Questions

- ▣ What is computation and computational thinking?
- ▣ How was Blockly?
- ▣ Lab submissions?
- ▣ Readings?

Questions

- ▣ Navigating Canvas, OLI, Autolab
- ▣ Using terminal, lab computers, etc
- ▣ <https://goo.gl/forms/kxL8n68Aqgu0qc1I2>
- ▣ OLI
 - ▣ Pretest
 - ▣ Module



We can even think in terms of houses. What is the advantage of using abstraction like this?

Managing complexity, reusability ...

General Purpose Computing

- We can design machines for specific computing tasks (averages, sums)
- Many earlier machines were fixed program computers
- Modern computers store sequences of instructions and execute them

Programs to Describe Mechanical Procedures

- What if we have millions of steps to specify for a computer?

We typically use higher-level programming languages to describe computations

Part 1: Computing Before World War 2

Early Devices For Arithmetic

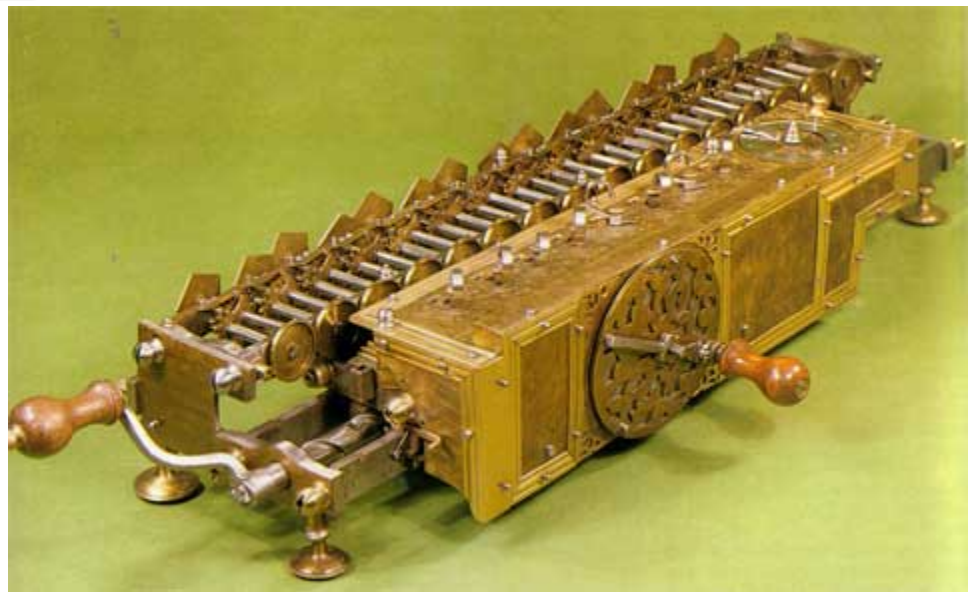
- Abacus: Humans do all the work
- Increase in scientific research in the seventeenth century motivated developments of tools to do arithmetic.
 - Napier's bones for multiplication
 - Pascaline to do addition and subtraction
 - Leibniz's wheel to do all of arithmetic operations

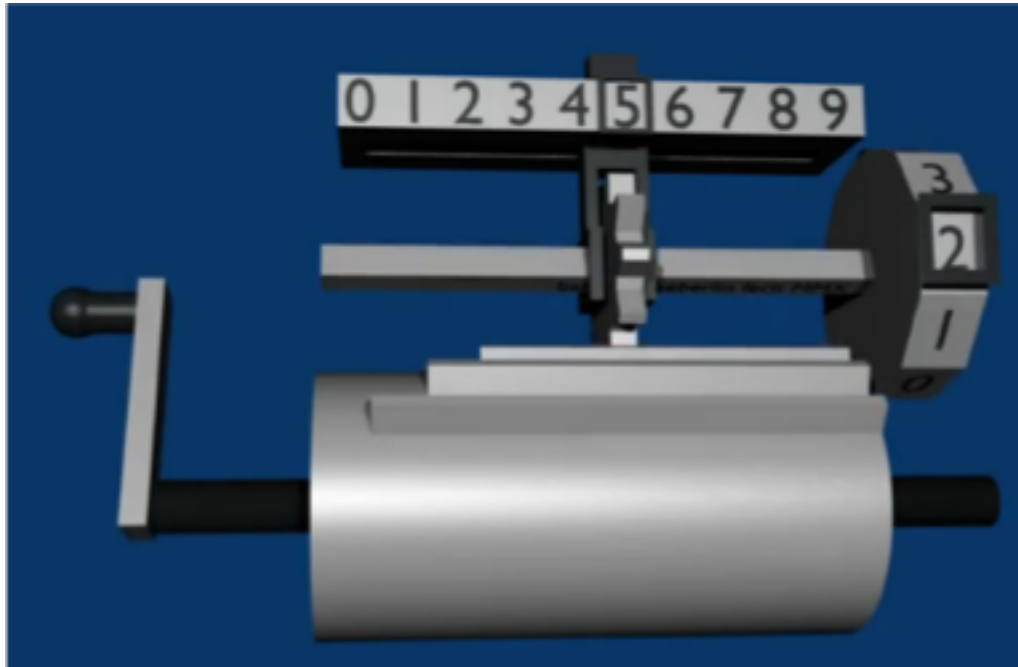
Mechanical Devices for Arithmetic



Blaise Pascal's
Pascaline (1643)

Leibniz's step reckoner
(designed in 1673,
completed in 1694)





Key problem:

How to make the carry work reliably in a purely mechanical system?

It's harder than you think!

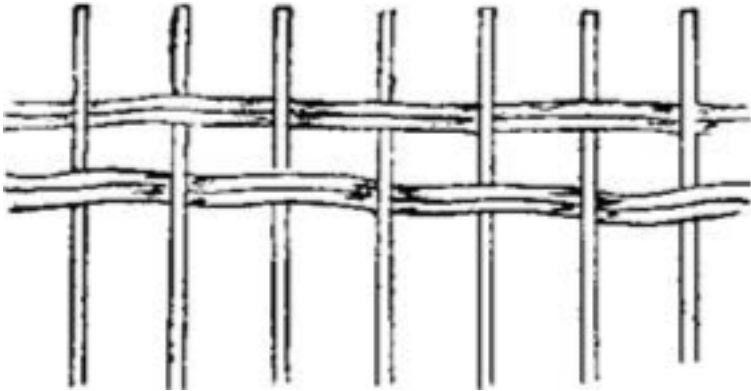
Leibniz's Stepping Drum

$$5 + 2 + 3 = 10$$

Shortcomings of Early Devices

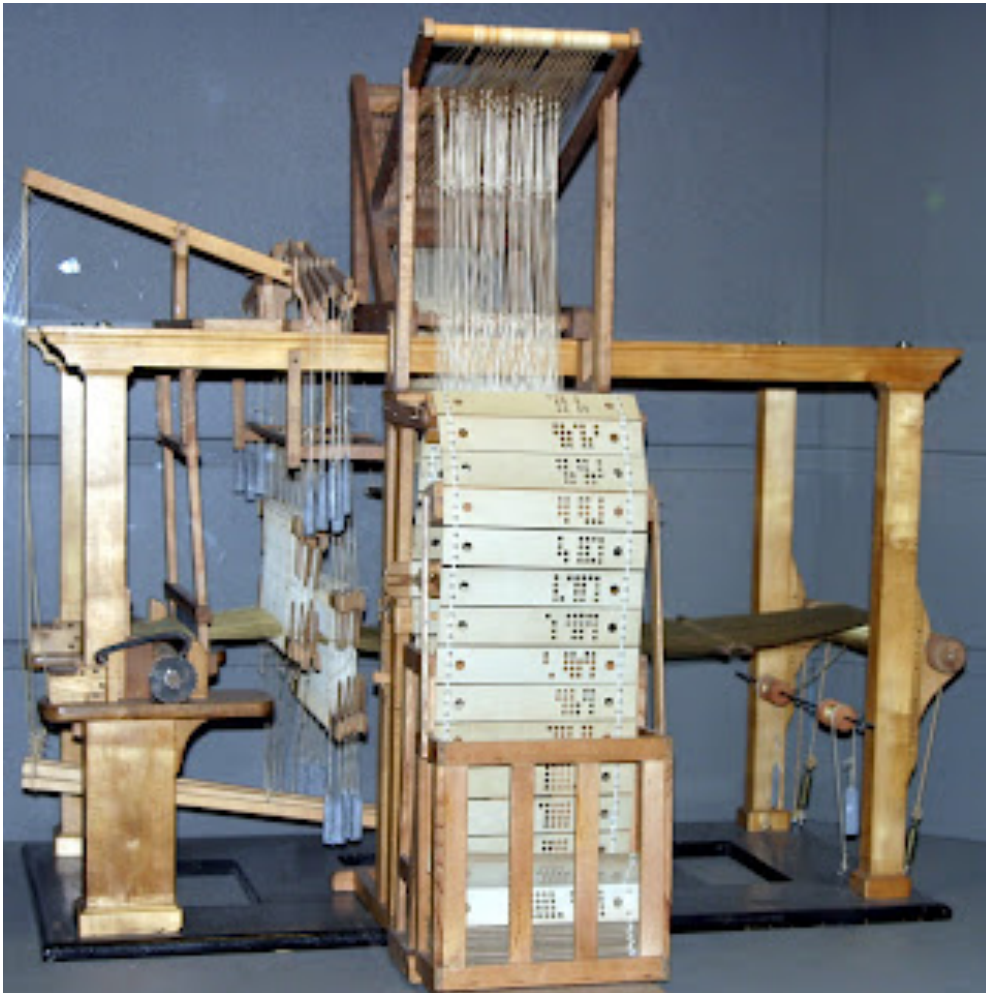
- They demonstrated how mechanization could simplify and speed up numerical computation.
- But
 - They did not have memory (storage for data)
 - They were not programmable

Weaving



- Simplest form: interweaving to make fabric
- How do you make patterns and pictures?

Jacquard's Loom: A Programmable Machine and Binary Storage



Developed by Joseph-Marie Jacquard (1801). The loom was controlled by a loop of punched cards. Holes in the punched cards determined how the knitting proceeded, yielding very complex weaves at a much faster rate.

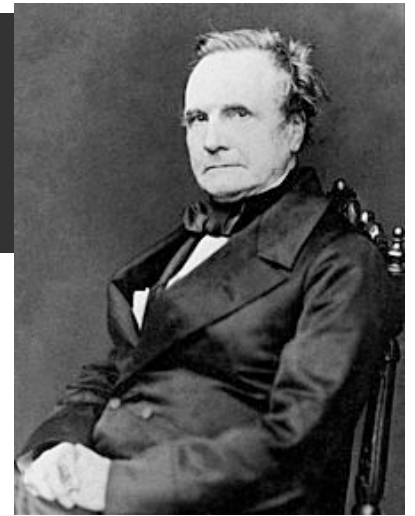
Complexity from simple primitives



Technophobia

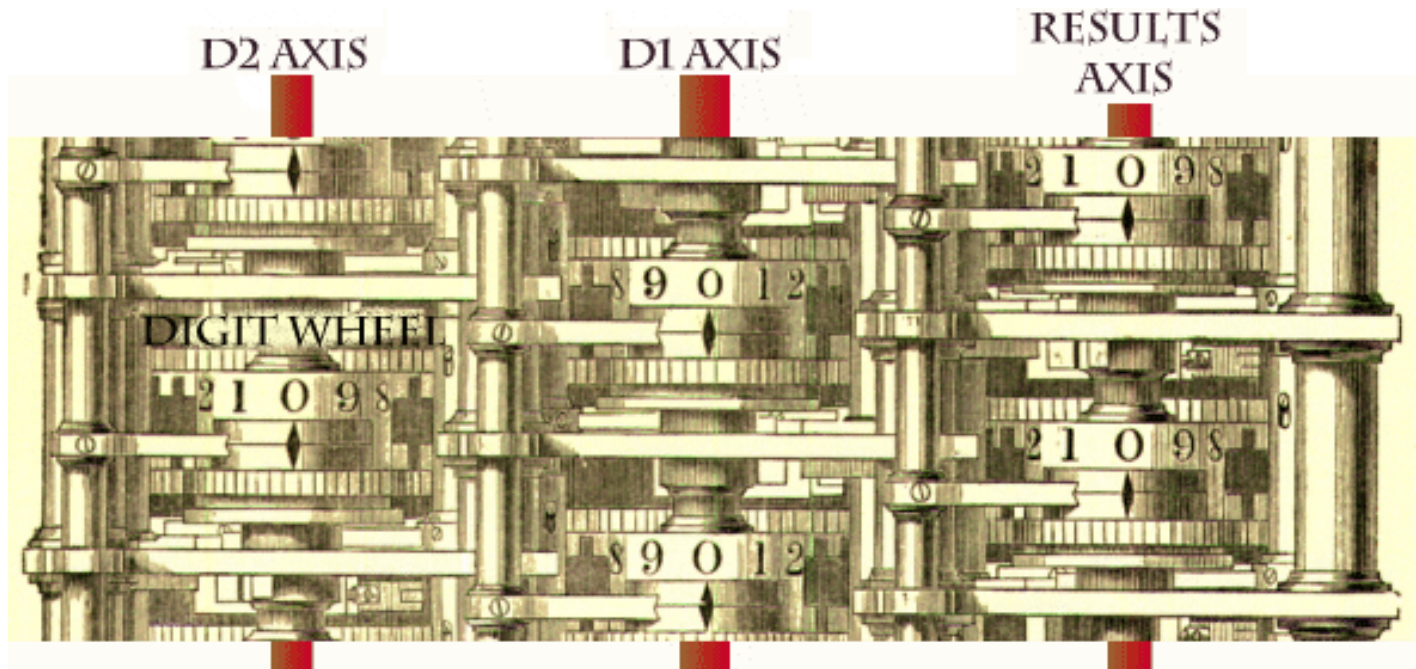
- Jacquard's loom showed how the knowledge of a human expert could be captured in machine-readable form.
- Mobilized craft guilds against this manufacturing technology.

Charles Babbage (1791-1871)



- Mathematician, industrialist, philosopher, politician.
- Frustrated by the many errors in printed mathematical tables (sines, cosines, logs, etc.) used in navigation and engineering.
- Observed that many long computations consist of operations that were regularly repeated.

Babbage's Difference Engine Computed 7th Degree Polynomials to 31 Digits



http://www.culture.com.au/brain_proj/CONTENT/BABBAGE.HTM

Building the Difference Engine



Photo of the
1832 Fragment
of a Difference Engine

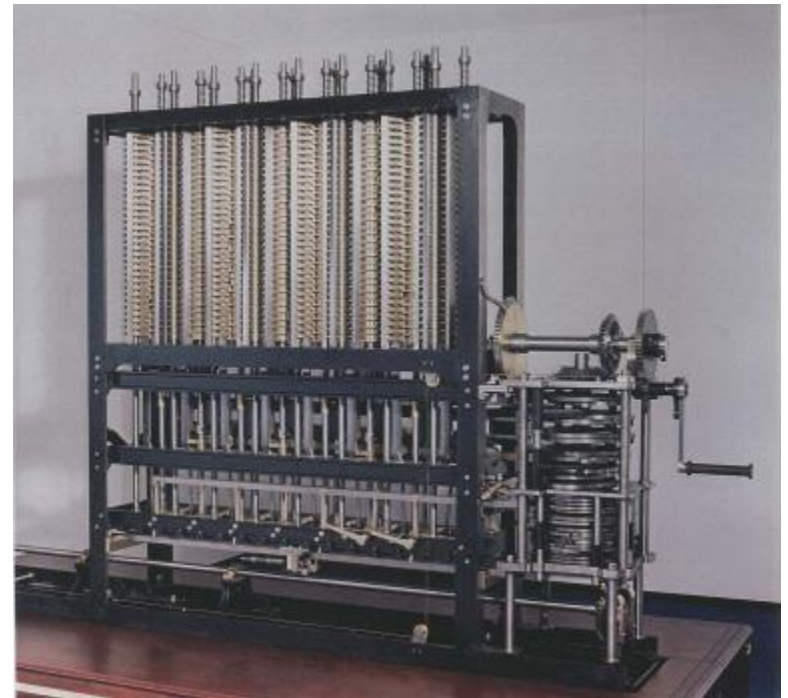


Photo of Babbage Difference Engine No. 2
constructed in 1991

(See video)

Charles Babbage's Analytical Engine

- Difference Engine (1822)
 - Never built (he ran out of money)

- Analytical Engine (1834-1836)
 - Babbage's more general "computer"
 - Never built, but its design is considered to be the foundation of modern computing
 - Had all the crucial features:
 - Arithmetic and logical operations
 - Digital data storage
 - Programs stored in memory

Ada Lovelace



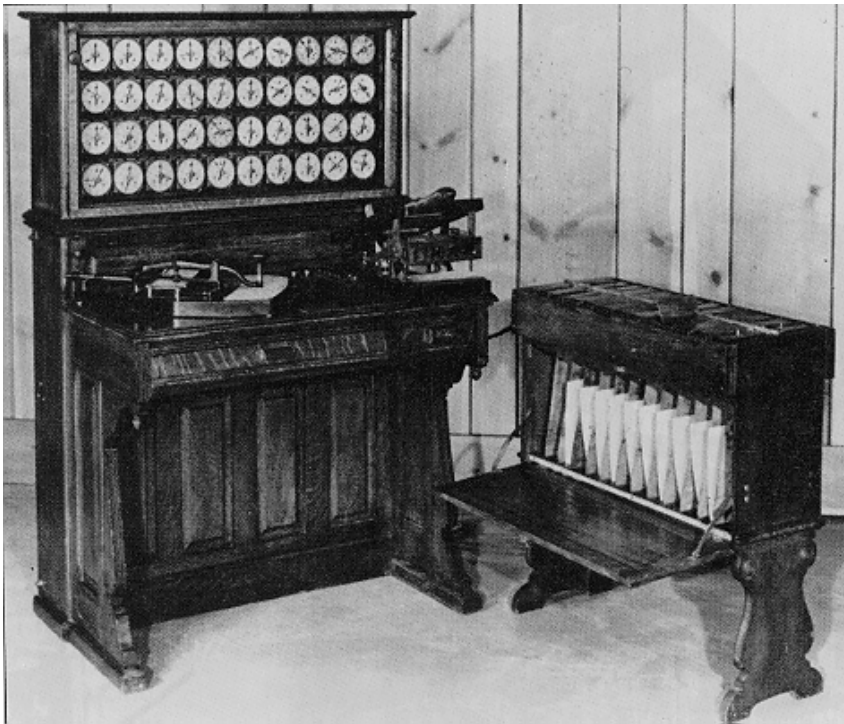
- 1815-1852
- Daughter of poet Lord Byron
- Translated Menabrea's *Sketch of the Analytical Engine* to English
 - Quadrupled its length by adding lengthy notes and detailed mathematical explanations
- Referred to as the world's first programmer
 - Described how the machine might be configured (programmed) to solve a variety of problems.

Herman Hollerith & The Hollerith Census Machine



- 1880 U.S. Census
 - The amount of data that needed to be analyzed was growing so quickly due to immigration
 - Required almost a decade to compute 1880 Census
- In 1882, Hollerith investigated a suggestion by Dr. John Shaw Billings, head of the division of Vital Statistics for the Census Bureau
 - “There ought to be some mechanical way of [tabulating Census data], something on the principle of the Jacquard loom, whereby holes in a card regulate the pattern to be woven.”

Hollerith's Census Machine



Demonstrated the advantages of automated information processing to solve large real-world problems

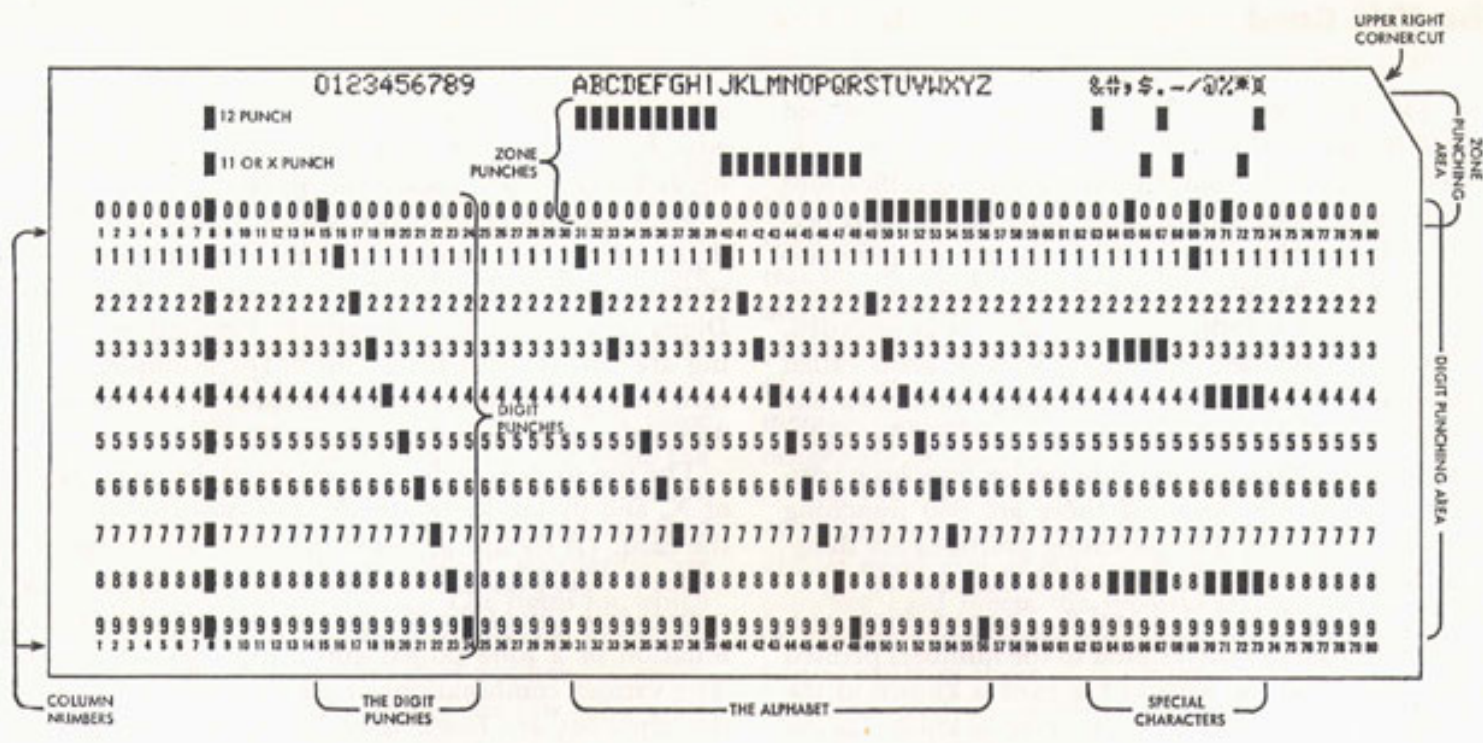
Photo: IBM

Hollerith's Census Machine

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|----|----|----|----|----|----|----|----|----|-----|----|----|----|---|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | CM | UM | Jp | Ch | Oc | In | 20 | 50 | 80 | Dv | Un | 3 | 4 | 3 | 4 | A | E | L | a | g |
| 5 | 6 | 7 | 8 | CL | UL | O | Mi | Qd | Mo | 25 | 55 | 85 | Wd | CY | 1 | 2 | 1 | 2 | B | F | M | b | h |
| 1 | 2 | 3 | 4 | CS | US | Mb | B | M | 0 | 30 | 60 | 0 | 2 | Mr | 0 | 15 | 0 | 15 | C | G | N | c | i |
| 5 | 6 | 7 | 8 | No | Hd | Wf | W | F | 5 | 35 | 65 | 1 | 3 | Sg | 5 | 10 | 5 | 10 | D | H | O | d | k |
| 1 | 2 | 3 | 4 | Fh | Ff | Fm | 7 | 1 | 10 | 40 | 70 | 90 | 4 | 0 | 1 | 3 | 0 | 2 | St | I | P | e | l |
| 5 | 6 | 7 | 8 | Hh | Hf | Hm | 8 | 2 | 15 | 45 | 75 | 95 | 100 | Un | 2 | 4 | 1 | 3 | 4 | K | Un | f | m |
| 1 | 2 | 3 | 4 | X | Un | Ft | 9 | 3 | i | c | X | R | L | E | A | 6 | 0 | US | Ir | Sc | US | Ir | Sc |
| 5 | 6 | 7 | 8 | Ot | En | Mt | 10 | 4 | k | d | Y | S | M | F | B | 10 | 1 | Gr | En | Wa | Gr | En | Wa |
| 1 | 2 | 3 | 4 | W | R | CK | 11 | 5 | l | e | Z | T | N | G | C | 15 | 2 | Sw | FC | EC | Sw | FC | EC |
| 5 | 6 | 7 | 8 | 7 | 4 | 1 | 12 | 6 | m | f | NG | U | O | H | D | Un | 3 | Nw | Bo | Hu | Nw | Bo | Hu |
| 1 | 2 | 3 | 4 | 8 | 5 | 2 | Oc | 0 | n | g | a | V | P | I | Al | Na | 4 | Dk | Fr | It | Dk | Fr | It |
| 5 | 6 | 7 | 8 | 9 | 6 | 3 | 0 | p | o | h | b | W | Q | K | Un | Pa | 5 | Ru | Ot | Un | Ru | Ot | Un |

Photo of a punch card for the Hollerith machine, from *John McPherson, Computer Engineer*, an oral history conducted in 1992 by William Aspray, IEEE History Center, Rutgers University, New Brunswick, NJ, USA.

- The entire 1890 census data was processed in 3 months and complete 1890 data was published in 1892.
- Total population of the U.S.: 62,622,250



An IBM punch card used from 1928 until the 1970s.

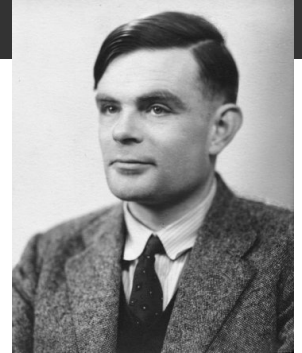
- Hollerith forms the Tabulating Machine Company in 1896 which eventually becomes IBM in 1924 through a merger and several name changes.

Computing from WWII

Effect of World War 2

- World War 2 gave rise to new information-based concerns
 - Ballistic tables, troop deployment data, secret codes
 - Military started funding projects to build automatic machines

Alan Turing



- Considered the “father” of modern computer science.
- Presented formalisms for the notions of computation and computability in the 1930’s.
- Worked at Bletchley Park in Great Britain during WW 2 to develop Colossus to help break the German Enigma Code.
- Developed the notion in 1950 of a test for machine intelligence now called the Turing Test.
- The Turing Award, the highest award in computing, is named in honor of Alan Turing.

Across the Atlantic in the US



Harvard Mark I
(IBM Archives)

Harvard Mark I (IBM Automatic Sequence Controlled Calculator)

Developed by Howard Aiken

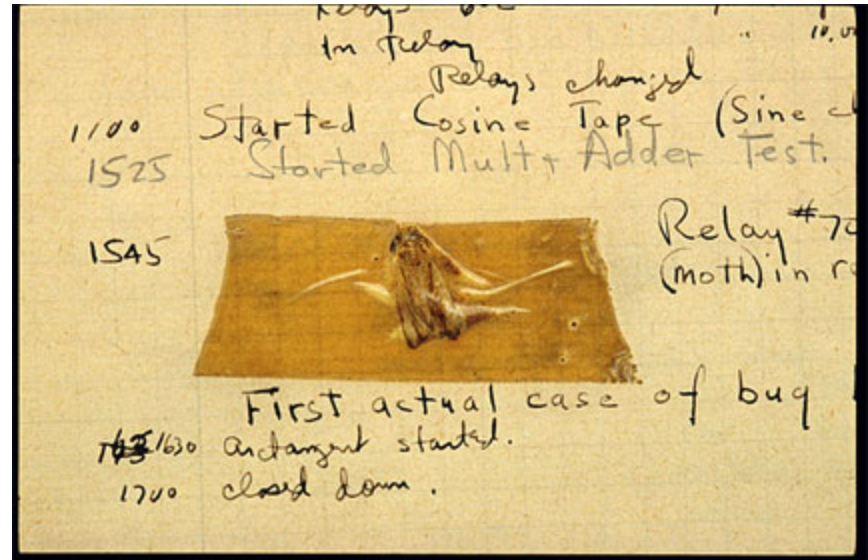
- Contained more than 750,000 components
 - over 50 feet long
 - 8 feet tall
 - weighed ~5 tons
- Sounded like a “roomful of ladies knitting”

Early Electronic Computers

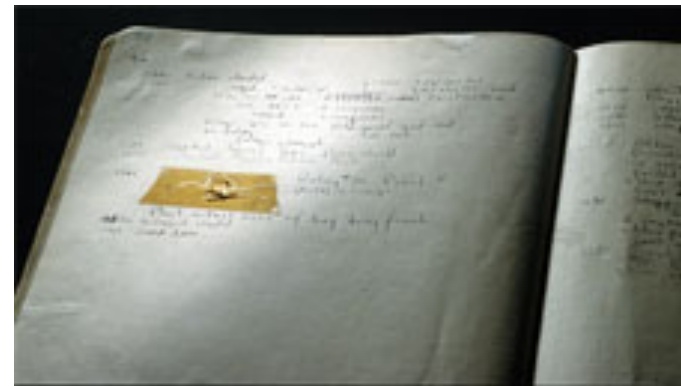
- Electromechanical – Harvard Mark I
- Purely electronic – ENIAC, Colossus, Atanasoff-Berry Computer (ABC), Z1 all had memory and were programmable.
- But they were not ***stored program computers***.
They were programmed externally.

The First Debugger

- Grace Hopper, working on the Harvard Mark II computer in 1947, found the first actual computer “bug” and coined the term “debugging”.



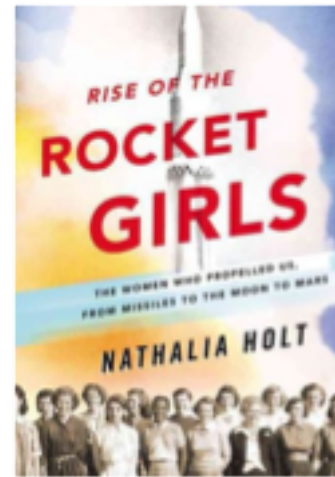
- The “Grace Hopper Celebration of Women in Computing” is an annual conference named in her honor.



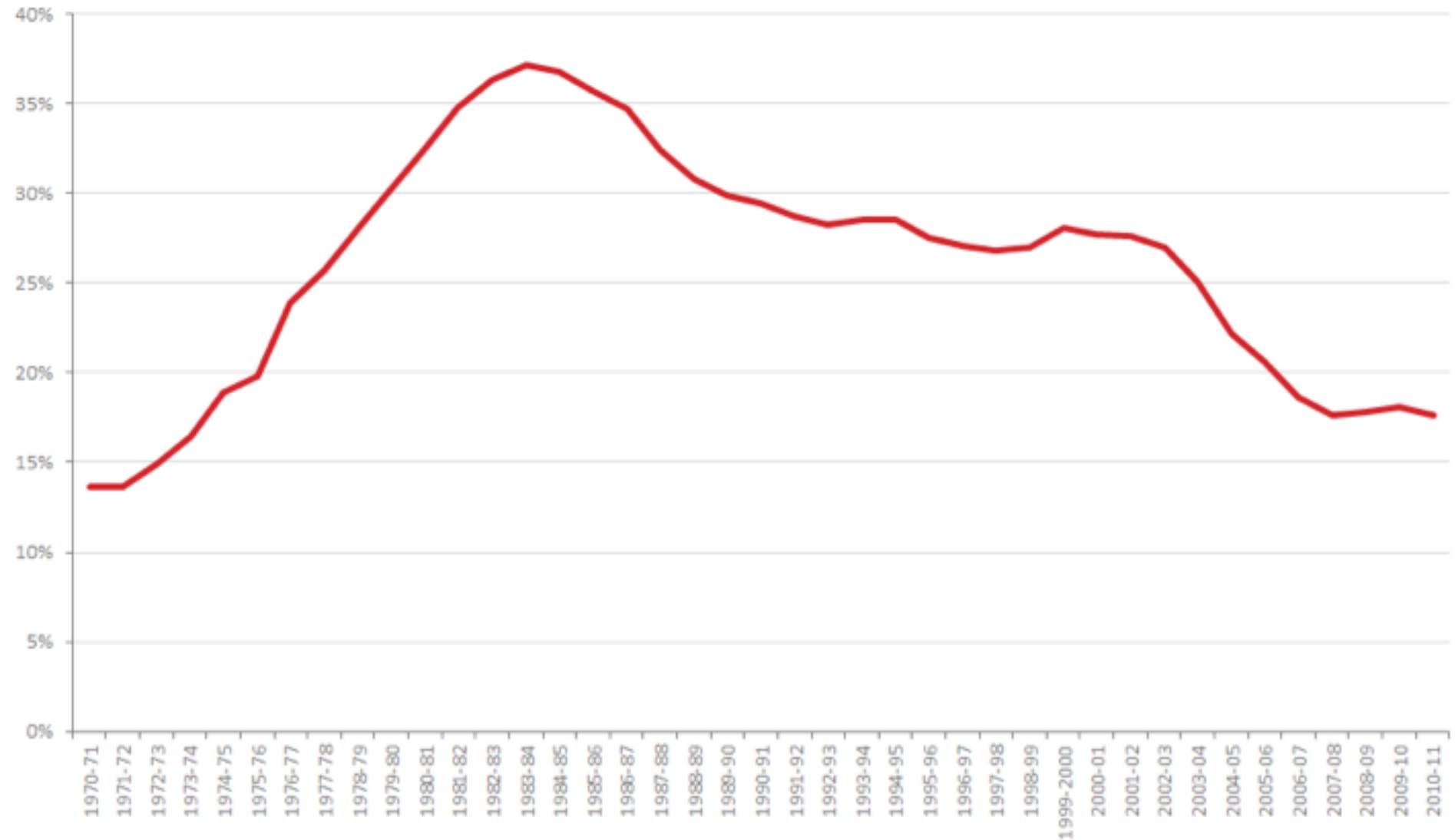
Grace Hopper



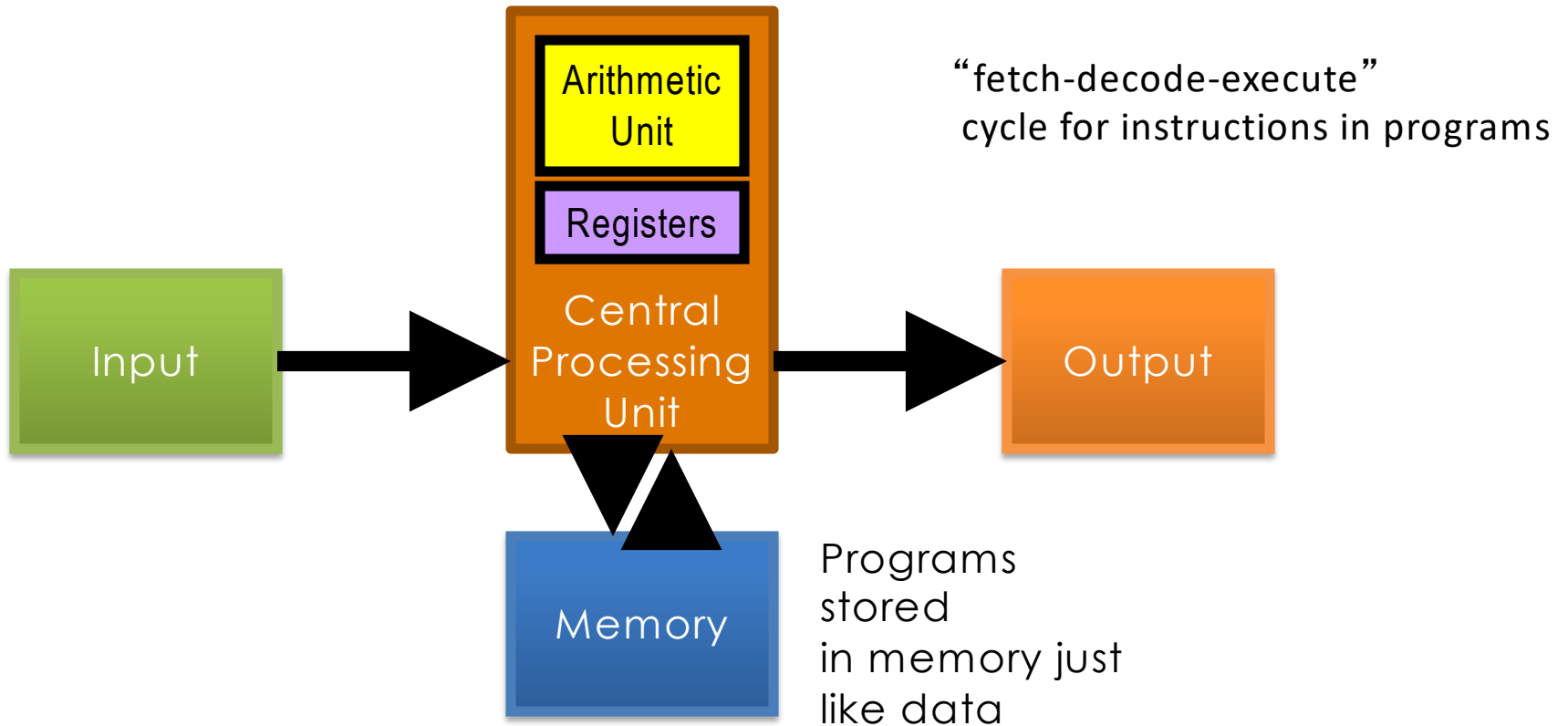
History of Computing is not the History of Great Men...



Percent of CS Bachelor's Degrees Awarded to Women, 1970-2011



Von Neumann Architecture



Mauchly, Eckert, von Neumann

- Mauchly and Eckert are generally credited with the idea of the stored-program
- BUT: John von Neumann publishes a draft report that describes the concept and earns the recognition as the inventor of the “stored-program” concept
 - “von Neumann architecture”
 - *A First Draft of a Report of the EDVAC* published in 1945
 - <http://www.wps.com/projects/EDVAC/>



von Neumann,
Member of the Navy
Bureau of Ordnance
1941-1955

UNIVAC and the First Compiled Programming Language

■ UNIVAC I

- Built by Remington Rand to compute 1950 U.S. census but completed in 1951
- Used to predict the winner of the 1952 U.S. Presidential Election based on ~3.4M votes



J. Presper Eckert and Walter Cronkite
next to the UNIVAC in 1952
(Center for the Study of Technology and Society)

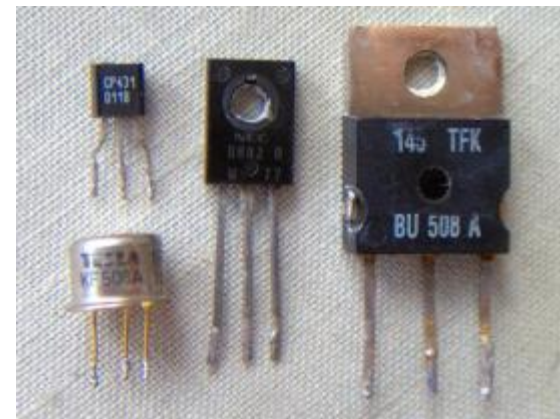
From Vacuum Tubes to Transistors



unreliable
bulky

Before the transistor was invented,
we used vacuum tubes: 2-3 inches tall;
equivalent to 1-3 transistors

reliable
smaller





Integrated Circuit

- A network of transistors and other electronic components incorporated on a single silicon chip
- Mass production capability
- Miniaturization of computers was desirable
 - Essential for the space race

Noyce and Kilby

- Robert Noyce and Jack Kilby are credited with the invention of the integrated circuit (IC) or microchip.
 - Robert Noyce co-founded Intel in 1968.
 - Kilby won the Nobel Prize in Physics in 2000.
- By the mid 1970s, ICs contained tens of thousands of transistors per chip.
 - In 1970, Intel created the 1103--the first generally available DRAM (memory) chip.
 - Today, you would need more than 65,000 of them to put 8 MB of memory into a PC.

Mainframes

- In the 1960s and 1970s, large computers called “mainframes” became widespread in businesses and universities. IBM was the largest computer maker.

Central
Processing
Unit



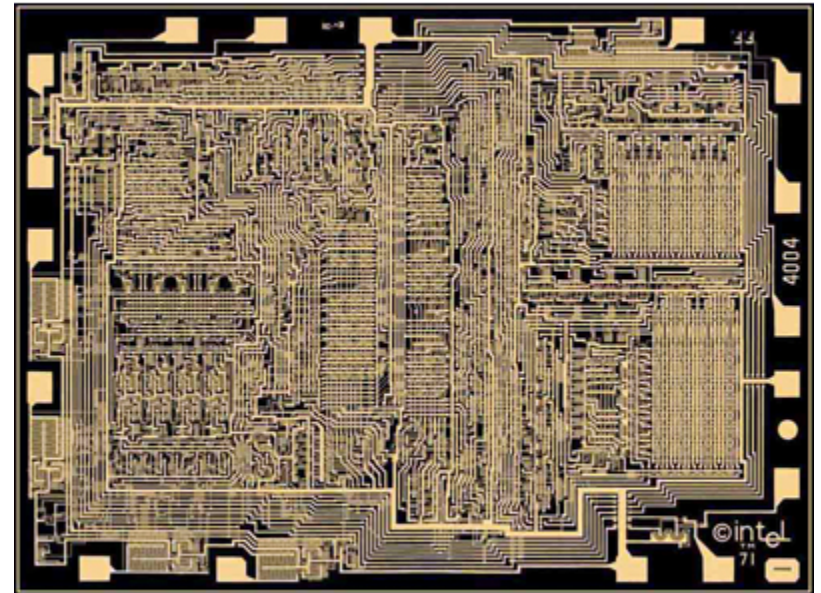
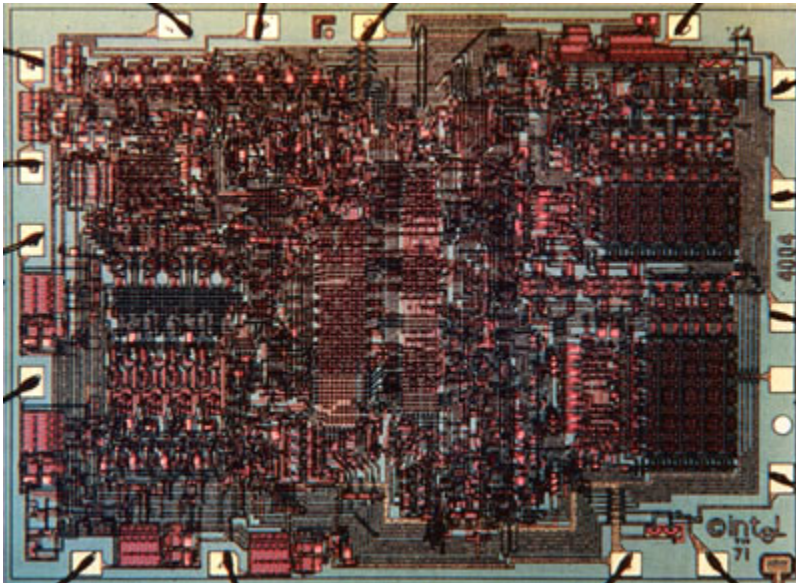
Card
Reader



IBM
System/360

The Microprocessor

- In 1971 Intel released the first microprocessor: the 4004, shown below.
- A microprocessor is an entire CPU on a chip.
- **The personal computer revolution began shortly thereafter.**



A Modern Inexpensive Personal Computer



1.4GHz dual-core Intel Core i5

4GB memory

500GB hard drive¹



Storage measured in terms of Bytes (1 B = 8 bits)

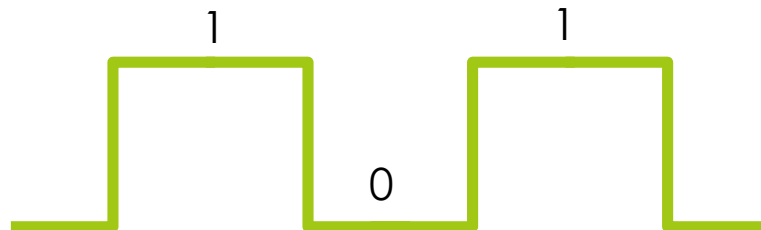
Intel HD Graphics 5000

OS X Yosemite

\$499.00

Data is Stored Using Bits

■ 1 – 0 ← Electric pulses (High - Low)



Say 'HI' to Digital World

| | | | | | | | |
|-----|----|----|----|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

72

| | | | | | | | |
|-----|----|----|----|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

73

HI

| | | |
|----------|----|-----|
| 01000000 | 64 | @ |
| 01000001 | 65 | A |
| 01000010 | 66 | B |
| 01000011 | 67 | C |
| 01000100 | 68 | D |
| 01000101 | 69 | E |
| 01000110 | 70 | F |
| 01000111 | 71 | G |
| 01001000 | 72 | H ← |
| 01001001 | 73 | I ← |
| 01001010 | 74 | J |

Units of Memory

| | | | | |
|------------|----|-------------|------------------|-------------------------|
| □ Byte | B | 8 bits (8b) | | |
| □ Kilobyte | KB | 1024 B | $= 2^{10}$ Bytes | $\approx 10^3$ Bytes |
| □ Megabyte | MB | 1024 KB | $= 2^{20}$ Bytes | $\approx 10^6$ Bytes |
| □ Gigabyte | GB | 1024 MB | $= 2^{30}$ Bytes | $\approx 10^9$ Bytes |
| □ Terabyte | TB | 1024 GB | $= 2^{40}$ Bytes | $\approx 10^{12}$ Bytes |
| □ Petabyte | PB | 1024 TB | $= 2^{50}$ Bytes | $\approx 10^{15}$ Bytes |

Examples

- How many bytes can be stored in a 4GB flash drive?

$$4 \times 2^{30} \text{ Bytes} \approx 4 \times 10^9 \text{ Bytes}$$

- How many bytes/second is a 16Mbps cable modem connection?

$$16 \times 2^{20} \text{ bps} = 16/8 \times 2^{20} \text{ Bps} \\ \approx 2 \times 10^6 \text{ bps}$$

Bits per second.
Note the lower case b.

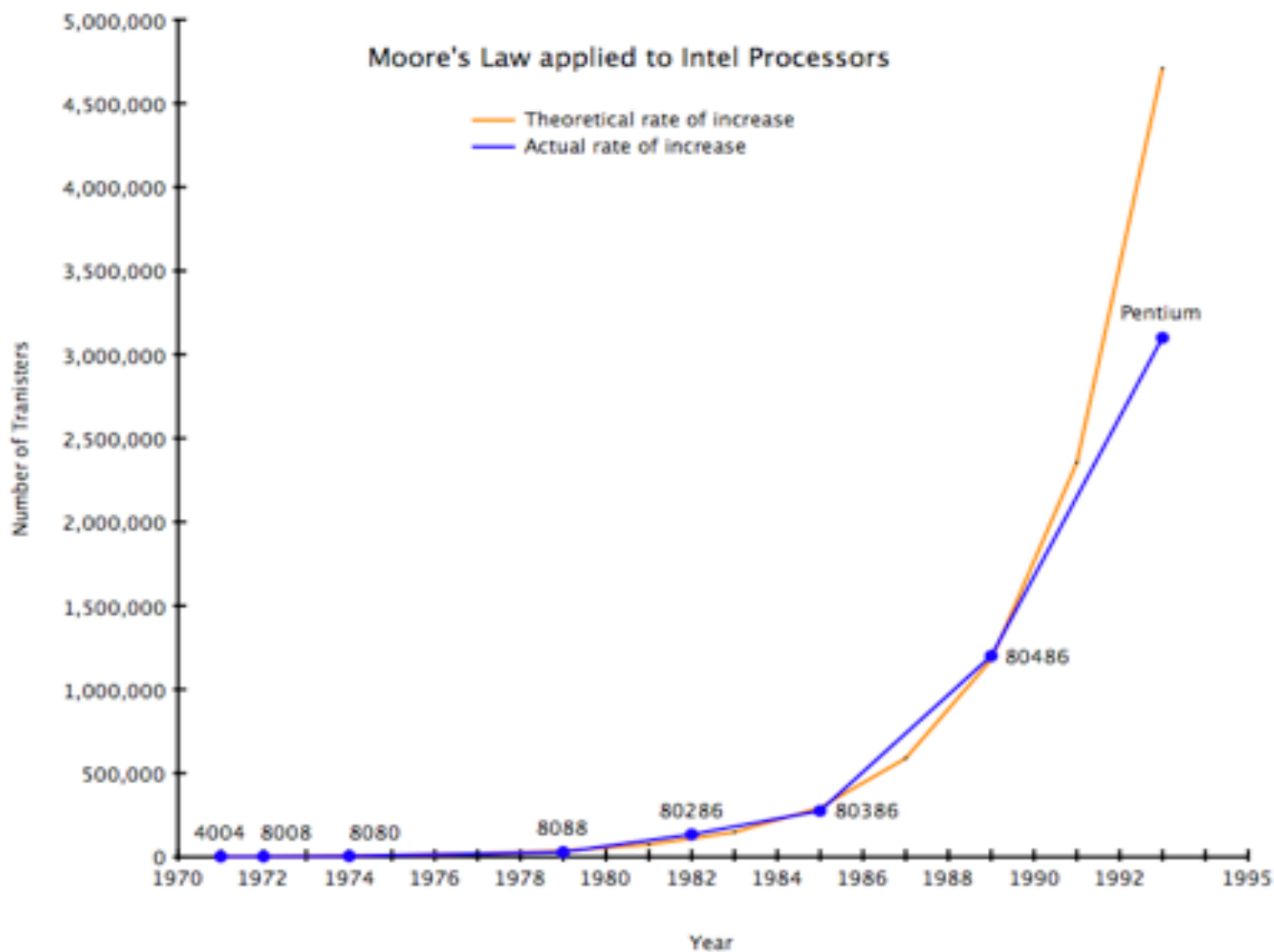
Bytes per second

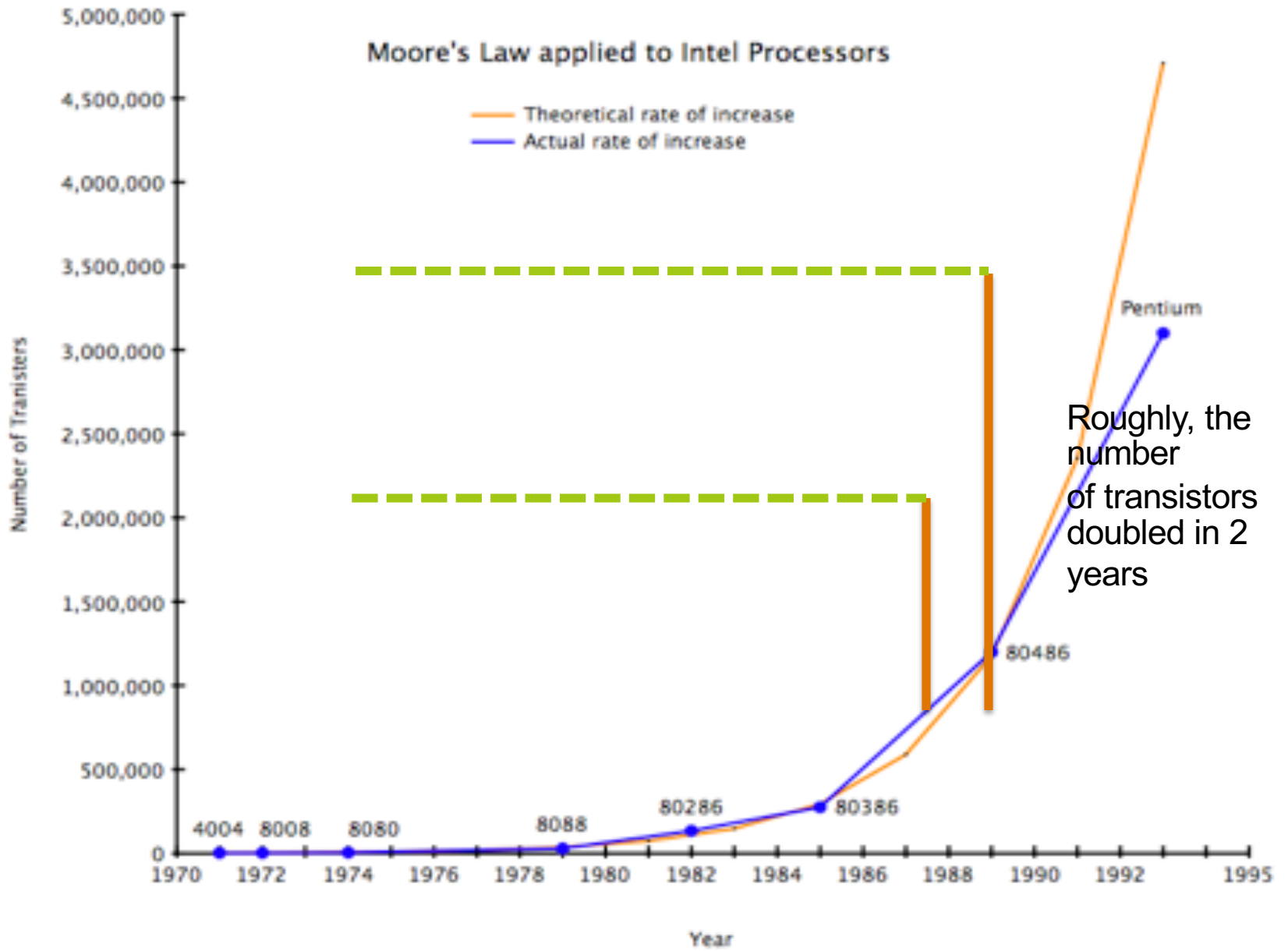
Moore's Law



- Gordon Moore co-founded Intel Corporation in 1968.

- Famous for his prediction on the growth of the semiconductor industry: Moore's Law
 - <ftp://download.intel.com/research/silicon/moorespaper.pdf>
 - An empirical observation stating in effect that the complexity of integrated circuits doubles every 18 months. ("complexity" generally means number of transistors on a chip)





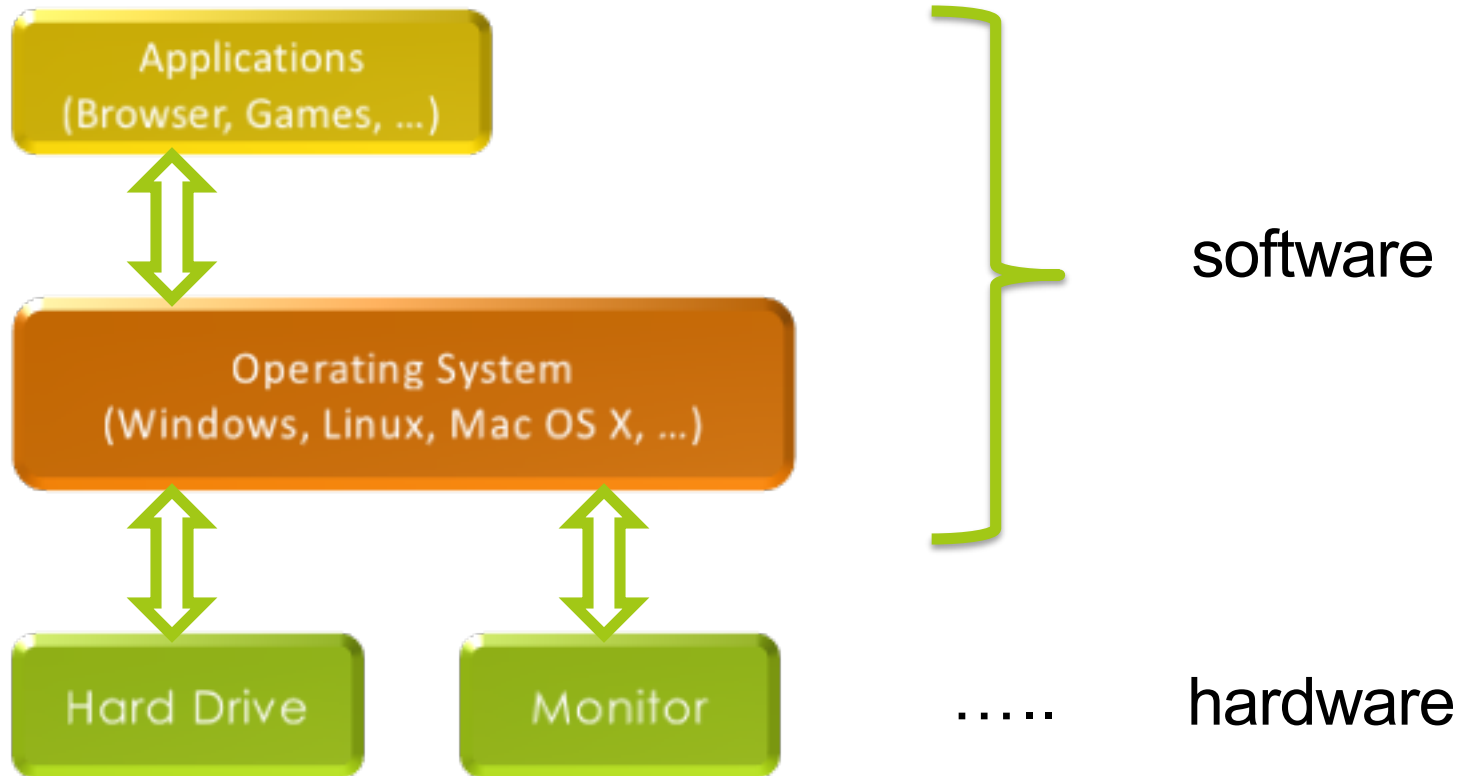
Better, Smaller, Cheaper



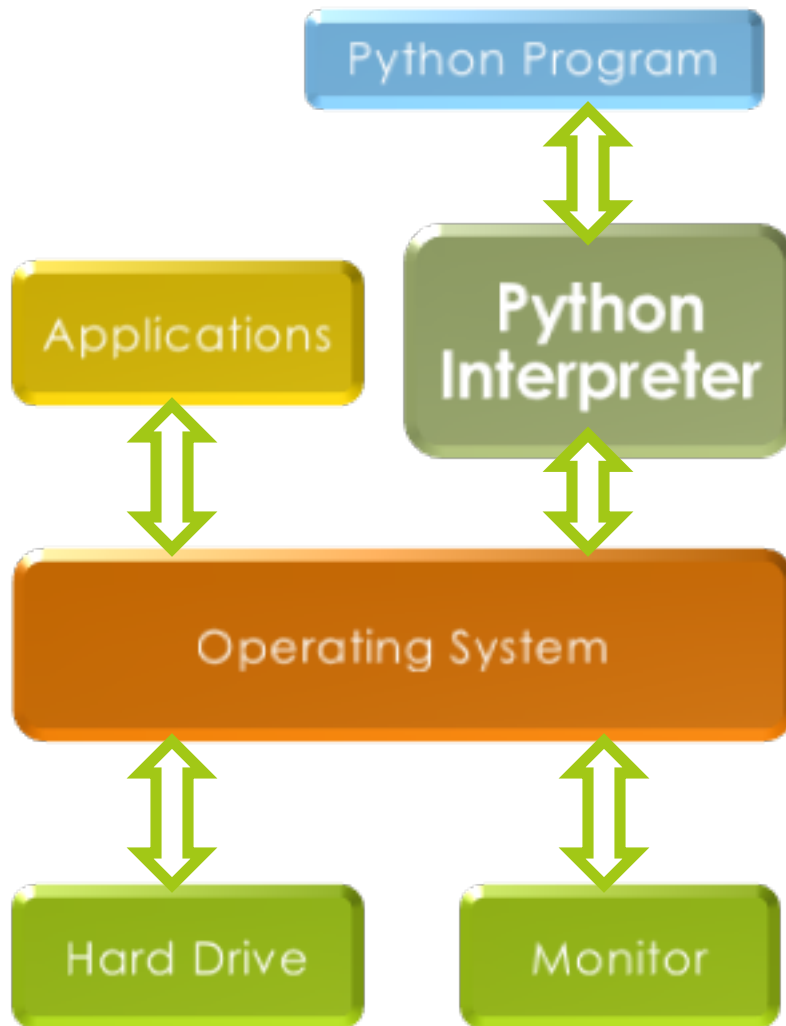
Computing power and related components of the digital revolution including memory, displays, sensors, digital cameras, software and communications bandwidth continue to get faster, cheaper, and smaller roughly at the pace Moore anticipated.

Towards Programming With Python

Hardware versus Software



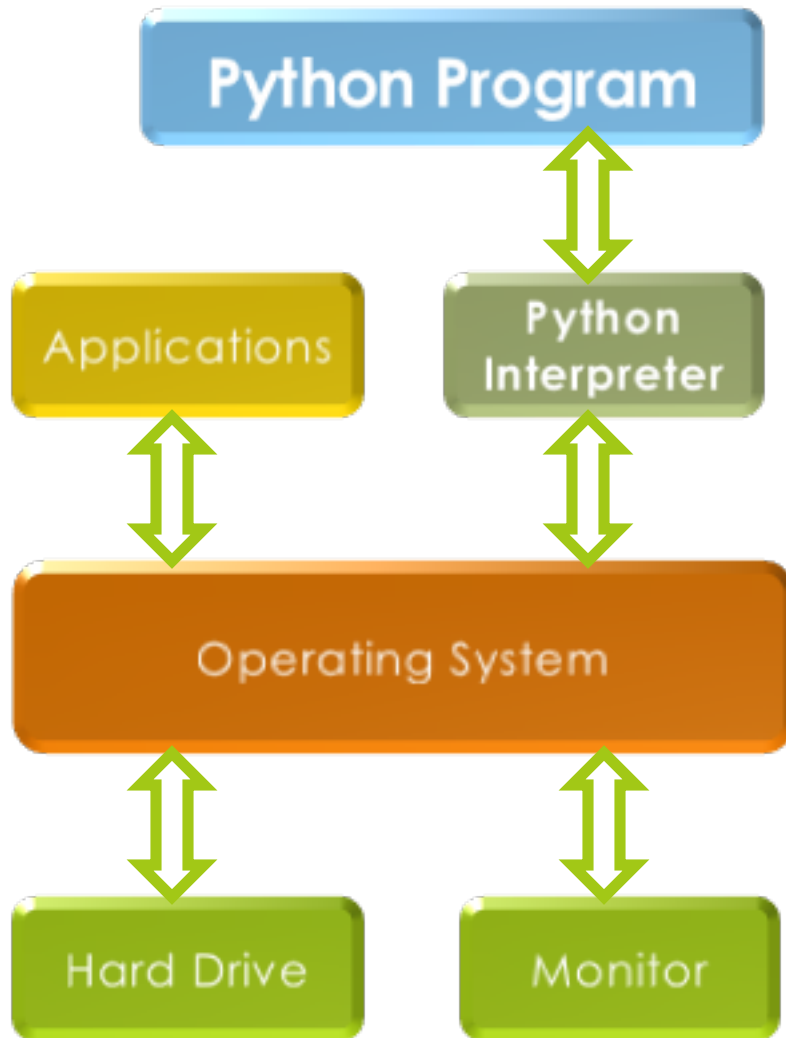
Execution of Python Programs



When you write a program in Python, Java etc. It does not run directly on the OS.

Another program called an interpreter or virtual machine takes it and runs it for you translating your commands into the language of the OS.

Execution of Python Programs



We will write Python programs that are executed by the Python Interpreter.

A Python Interpreter for your OS already exists. You can use the one on lab machines, install one for your laptop, or use one remotely

...

Using a Python Interpreter

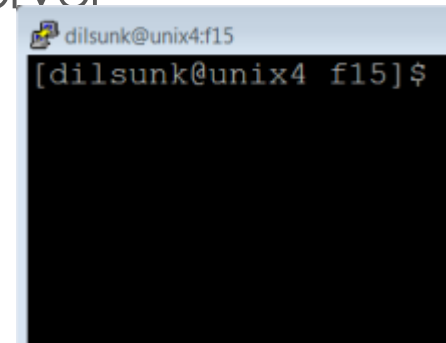
There are two ways to interact with a Python interpreter:

1. Tell it to execute a program that is saved in a file with a `.py` extension
2. Interact with it in a program called a shell

You will interact with Python in both ways.

A Short Introduction to Python

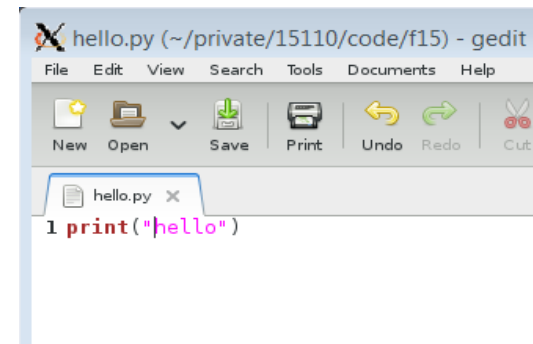
- Starting the Python interpreter either using a Unix Server at CMU or on your own computer
 - See the Resources page for specific instructions



```
dilsunk@unix4:f15  
[dilsunk@unix4 f15]$
```

- Creating .py files with a text editor

- Files with the .py extension can be created by any editor but needs a Python interpreter to be read.
- We have chosen editor editor for the course but you may use an editor of your own choice if you feel comfortable.
- Why IDE? Why Not?



```
hello.py (~/.private/15110/code/f15) - gedit  
File Edit View Search Tools Documents Help  
New Open Save Print Undo Redo Cut  
hello.py x  
1 print("hello")
```

Remember

- ❑ Next Lecture: Programming with Python
 - ❑ Note resources link and tutorials have extra info on getting running with python
- ❑ Tonight:
 - ❑ Pre-Test
 - ❑ Finish OLI
 - ❑ Programming Assignment 1
- ❑ For Thursday (9:00):
 - ❑ Academic Integrity Form
 - ❑ Problem Set 1

Useful Unix Commands (Part 1)

All commands must be typed in lower case.

`pwd` --> print working directory, prints where you currently are

`ls` --> list, lists all the files and folders in the directory

`cd` stands for 'change directory':

`cd lab1` --> change to the lab1 directory/folder

`cd ..` --> going up one directory/folder

`cd ../..` --> going up two directories

Useful Unix Commands (Part 2)

`mkdir lab1` --> make directory lab1 aka makes a folder called lab1

`rm -r lab1` --> removes the directory lab1
(-r stands for recursive, which deletes any possible folders in lab1 that might contain other files)

`cp lab1/file1.txt lab2` --> copies a file called file1.txt, which is inside of the folder lab1, to the folder lab2

`mv lab1/file1.txt lab2` --> moves a file called file1.txt, which is inside of the folder lab1, to the folder lab2

`zip zipfile.zip file1.txt file2.txt file3.txt` -->
zips files 1 to 3 into zipfile.zip

`zip -r zipfile.zip lab1/` --> zips up all files in the lab1 folder into zipfile.zip

Useful Unix Commands (Part 3)

`^c` --> ctrl + c, interrupts running program

`^d` --> ctrl + d, gets you out of python3

`"tab"` - autocompletes what you're typing based on the files in the current folder

`"up"` - cycles through the commands you've typed. Similarly for the opposite effect, press `"down"`

Useful Unix Commands (Part 4)

`python3 -i test.py` --> load test.py in python3, and
you can call the functions in test.py.

`gedit lb1.txt &` --> opens up lb1.txt on gedit and & allows you to
run your terminal at the same time
(else your terminal pauses until you close gedit)

And lastly, you can always do `man <command>` to find out more about a
particular command you're interested about (eg. `man cp`, `man ls`)