1 The Digital World

1.1 The Engine Behind: Electricity

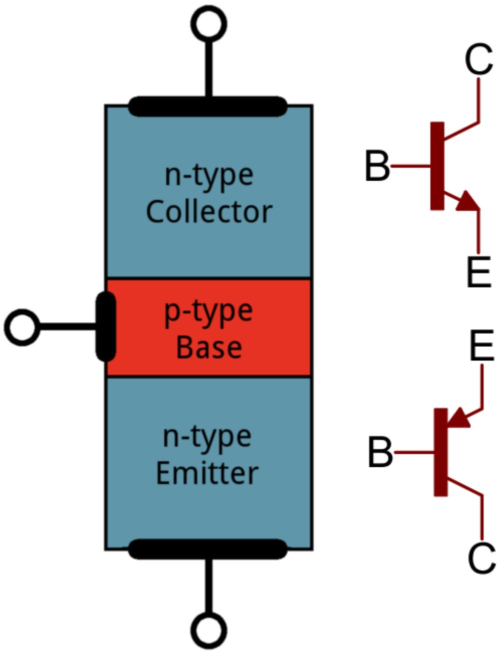
Main Inventions (Not discovery)

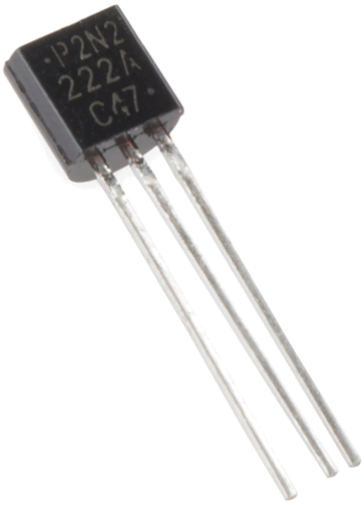
- **DC Current**
    - Benjamin Franklin

- **AC Current**
    - Nicolas Tesla

1.2 The Beginning: The Invention of Transistors

- This is the greatest invention of the 20th century
    - You will understand why it is so in a few





Inventors

- John Bardeen and Walter Brattain
- Invented Transistors in 1947

Why They Invented Transistors

- For the purpose of current amplifications

How Does It Work?

- Made up of semi-conductors (Mainly Silicon)
- Pure Silicon is not the best conductor
- Silicon has 4 electrons on its valence shell (14th element on the periodic table)
- Silicons form covalent bonds sharing the external electrons
- Using a technique called **doping**, they can be changed to a better one (N-Type or P-Type doping)
- When you dope a silicon in **N-P-N** manner, a **transistor** is born
- Electrons naturally migrate from the N side to the P side
- Once small number of electrons flow, then further flow is restricted because of the lack of required energy
- If we attach an external source, we can create a **forward biased diode**
- When two diodes are connected together, aka a **transistor**, no matter which side we turn the external source, one will be reverse biased, hence restricting the flow of electrons
- But, if we apply a second source, just enough to start the flow of electrons, it then creates more of the other available electrons to follow
- Which leads to **current amplification**
- Depending on the amount of voltage you apply to the transistor, it acts as an **ON or OFF switch**
- When expressed in binary, **ON is 1** and **OFF is 0**

Hence the Silicon Valley is named after this development

1.3 How Transistors Lead to the Creation of Computers

First Step: Logic Gates

- AND Gate
- OR Gate
- NOT Gate
- NAND Gate
- NOR Gate

Second Step: Flip-Flops

- Combination of logic gates, in a way that it remembers states
- Led to the ability of storing 0s and 1s
    - **Bit**
        - 1 bit of data in the very early days
        - Example
    - **Byte**
    - **Registers**
- **To Put Things in Perspective**
    - One byte = 8 bits
    - One byte requires eight flip-flops
    - One flip-flop requires 3 - 4 transistors
    - A modern day computer has around **250GB** storage capacity
        - Try to express this in terms of **transistors**

Third Step: Information Storage

- Numbers Storage
- Word Storage
- Image Storage
- Video Storage
- Sound Storage

Fourth Step: Data Processing

- **Basic Logic Circuits**
    - Combination of logic gates arranged in a certain way to perform certain tasks
        - Ex: Binary Adder
- A set of complex Logic Circuits is what we call a **processor**

- A **processor** is an electronic chip containing **millions and millions** of logic gates, and is capable of interpreting instructions

The machine that can store and process data is basically what a computer is

1.4 Increasing the Speed of Processors

- To perform computation, the processor needs **inputs**
- These inputs are stored in your **hard disk**
  - Has a huge number of registers
  - Takes time to find the requested data

RAMs

- Serve as facilitators

Caches

- Serve as facilitators

A Typical Computer Would Have

- 250GB Hard disk
- 4GB RAM
- 1GB Cache
- The higher the size of the **RAM** and **Cache**, the faster the computer — but more expensive

All these are interconnected on what we call a **"Mother Board"**

1.5 The Software Part

- Any software is just a bunch of **0s and 1s** arranged in a different way
  - Theoretically, you can write a software entirely using just 0s and 1s
  - Writing code just using 0s and 1s is called **"Machine Language"**

- Problem: Humans are not accustomed to using 0s and 1s
  - We use alphabets
  - 0s and 1s are for the **machines**

- **Solution:**
  - Rise of **Assembly Languages**
  - Then came **Programming Languages**

Programming Language Milestones

- **FORTRAN** – IBM, 1954
- **C** – Bell Laboratories (AT&T)
- **Java** – Sun Microsystems (early 1990s, later Oracle)
- **C#** – Microsoft, 2000
- **JavaScript** – Netscape Communications, 1995

Example: Add Two Numbers – Machine Language

1.6 Operating Systems

- As computers advanced, development, scheduling, and execution of jobs got more complex
- Led to the development of **Operating Systems (OS)**

Responsibilities of an OS

- Manage resources of the machine
  - Interacts with raw hardware
  - Scheduling and management
  - Manage user processes
  - Provides GUI (Graphical User Interface)

The OS controls what is sent to the processor, and it fetches and stores manipulated data.
The **processor** simply does the calculations ordered by the OS.

1.7 Computer and Mobile Applications

- An **Application** is a computer program designed to help perform an activity
    - Examples:
        - Calculator
        - Microsoft Office (Word, Excel, PowerPoint)
        - Chrome, Firefox
        - Sublime Text Editor
        - Over a **billion** applications exist

Typically Written Using:

- C
- C++
- Java
- C#
- Etc...

Our job: Choose the best software that helps us get the most from the computer

1.8 Best Applications for a Web Developer

- **Editor:**
    - Sublime Text
- **Browser:**
    - Chrome
    - Chrome Developers Tool
        - YouTube Demo
- **Source Control Repository:**
    - GitHub
- **Local Server:**
    - LAMP/WAMP
- **Photo Editing:**
    - Adobe Photoshop
- **FTP Client:**
    - Cyberduck

Conclusion

- Mastering your computer means mastering your **Operating System (OS)**
    - The faster your speed, the better
    - Try to understand everything about your OS

Master Your Operating System

Windows:

- 2hr Guide
- 40min Guide

Mac:

- 2hr Guide
- 45min Guide

Next Class

The Making of the Internet