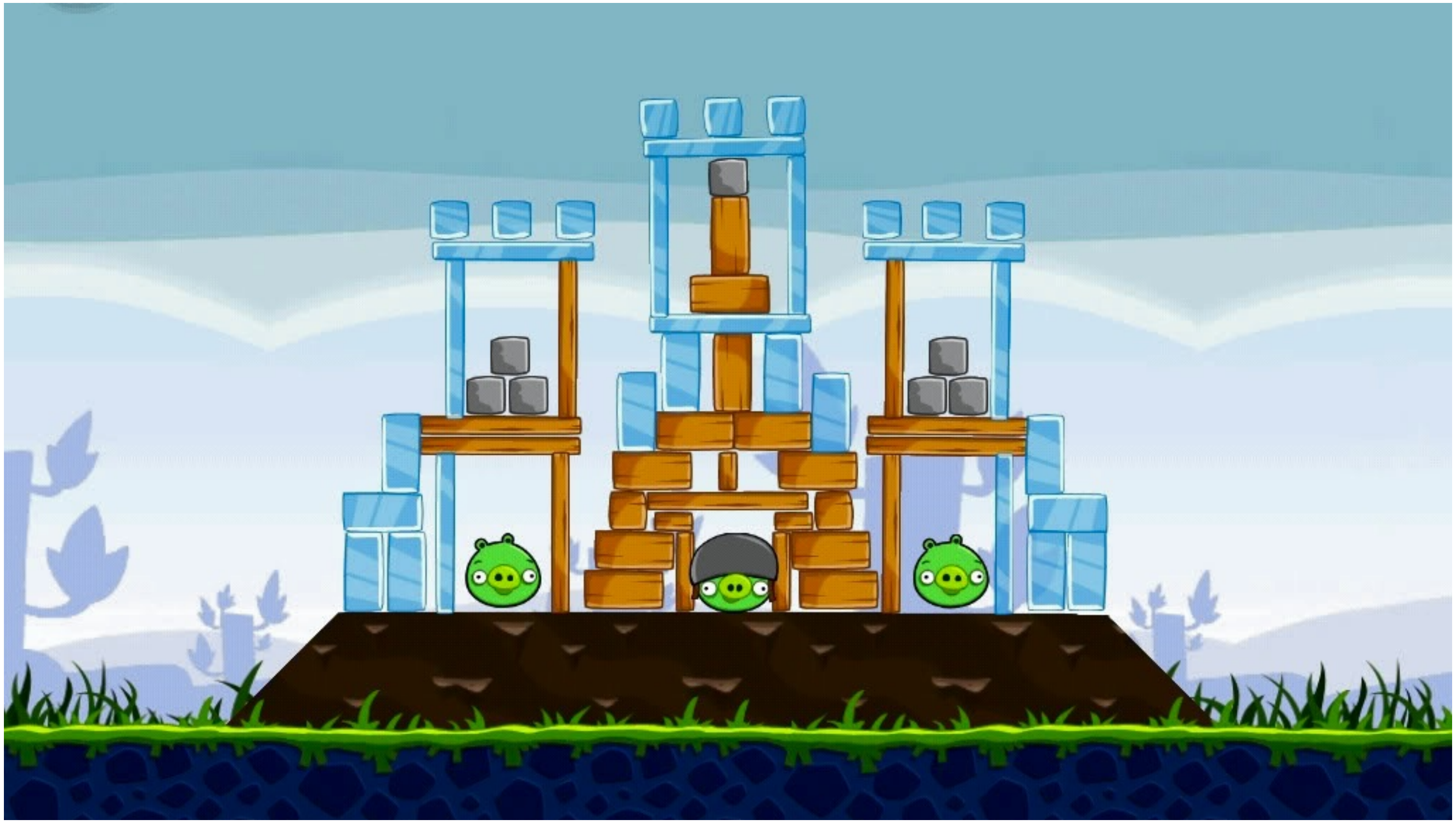
Computer Systems and Networks

ECPE 170

Introduction

A Modern Computer

Applications



Application – Angry Birds

- Written in a high level language (Objective C)
- What **resources** does *Angry Birds* need to run? (i.e. what does the *Angry Birds* executable file need to execute?)
  - Hardware
    - Processor(s) – Run program, display graphics, …
    - Memory – Store programs, store data
    - I/O – Touch screen, storage, network, 3-axis gyro, …
  - Software - Operating system

Software - Operating System

- Apple iOS – Used in iPads, iPhones, iPods, Apple TV
  - Variant of Mac OS X operating system used on traditional Macs
- **What are some jobs of this operating system?**
  - Manage hardware
  - Manage applications (multitasking)
- Written in high-level languages
  - C, C++, Objective C (varies by component)
  - **Can we run this code directly on the processor?**

Software - Compilers / Interpreters

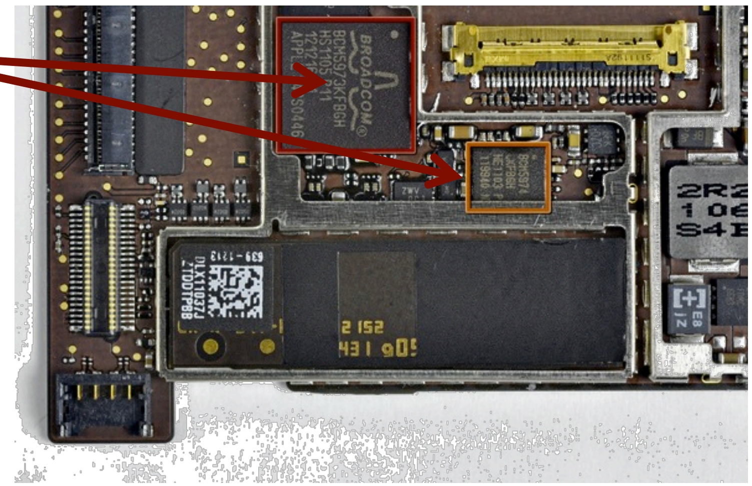
- These are programs that **build** other programs!
- Goal: Convert high-level languages into machine code that can be directly executed by hardware
- Examples
  - Apple Xcode
  - Microsoft Visual Studio
- **What's the difference between a compiler and interpreter?**

Hardware

http://www.ifixit.com/Teardown/iPad-2-Wi-Fi-Teardown/5071/1

Hardware

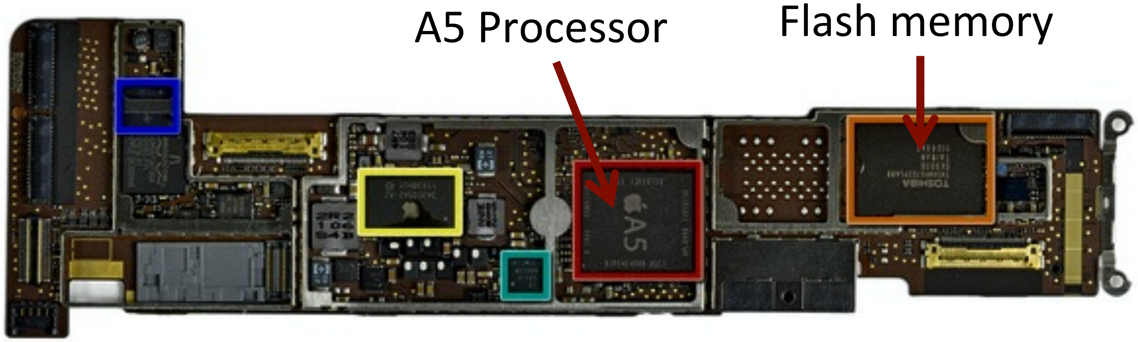
Touchscreen controllers



A5 Processor

Flash memory

Wi-Fi / Bluetooth Chip



iPad 2 Processor

- Apple A5 Processor
  - Clock speed – 1GHz
  - Dual core
  - 200MHz bus
  - 512 MB RAM

What do these mean?

- **What does a processor do?**
  - Executes assembly language instructions
    - **How???**

Microarchitecture

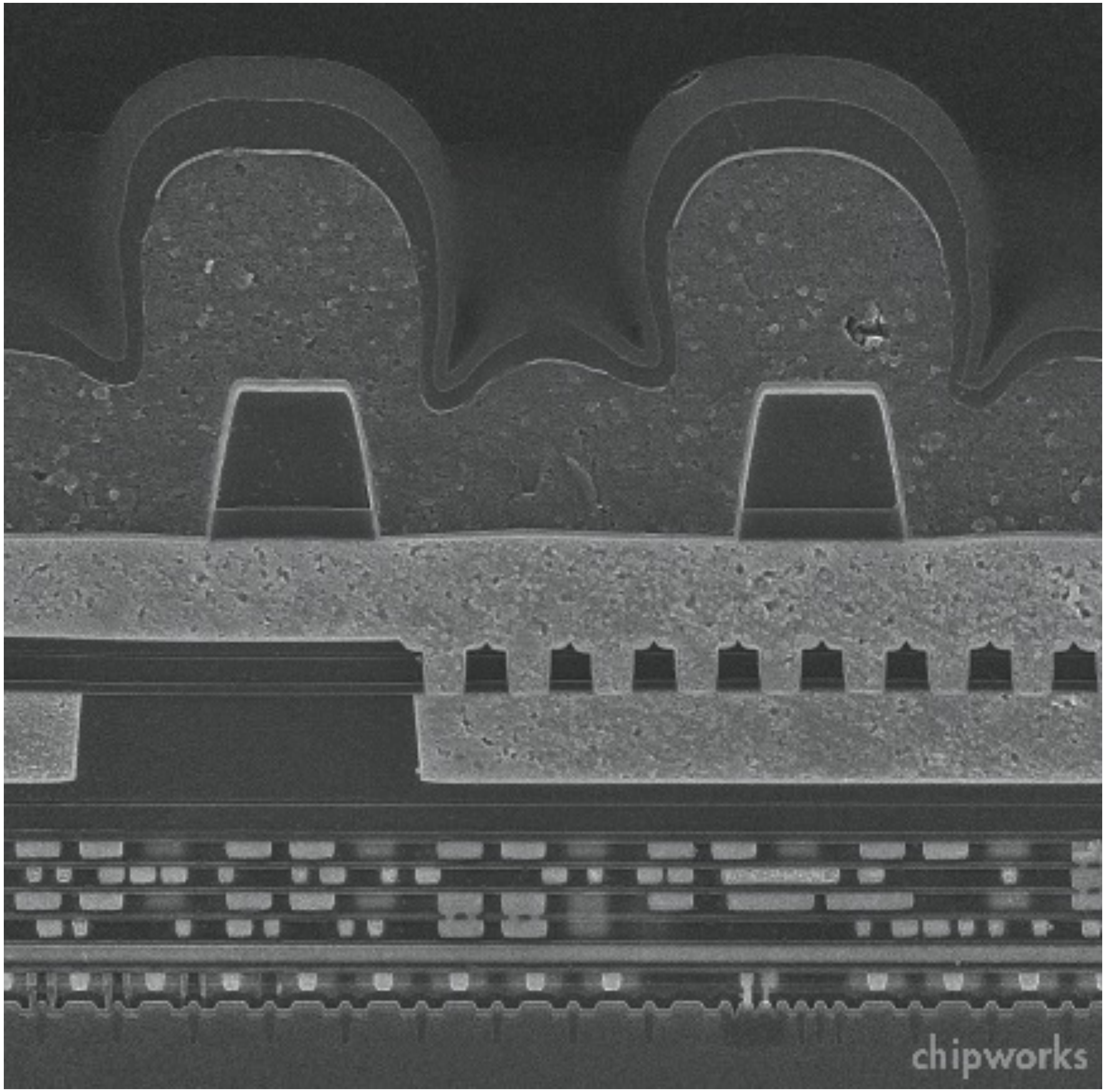
How Does It Work?

- Apple won’t tell us – trade secret!
- Experts can dissolve (with acid), burn, or grind off outer protective layers of chip and then peer inside:
  - Need a *really good* microscope!
  - *Reverse Engineering in the Semiconductor Industry:* http://www.scribd.com/doc/53742174/Reverse-Engineering

Can see this level of detail with your own eyes…

Divided into logic blocks with different functions:

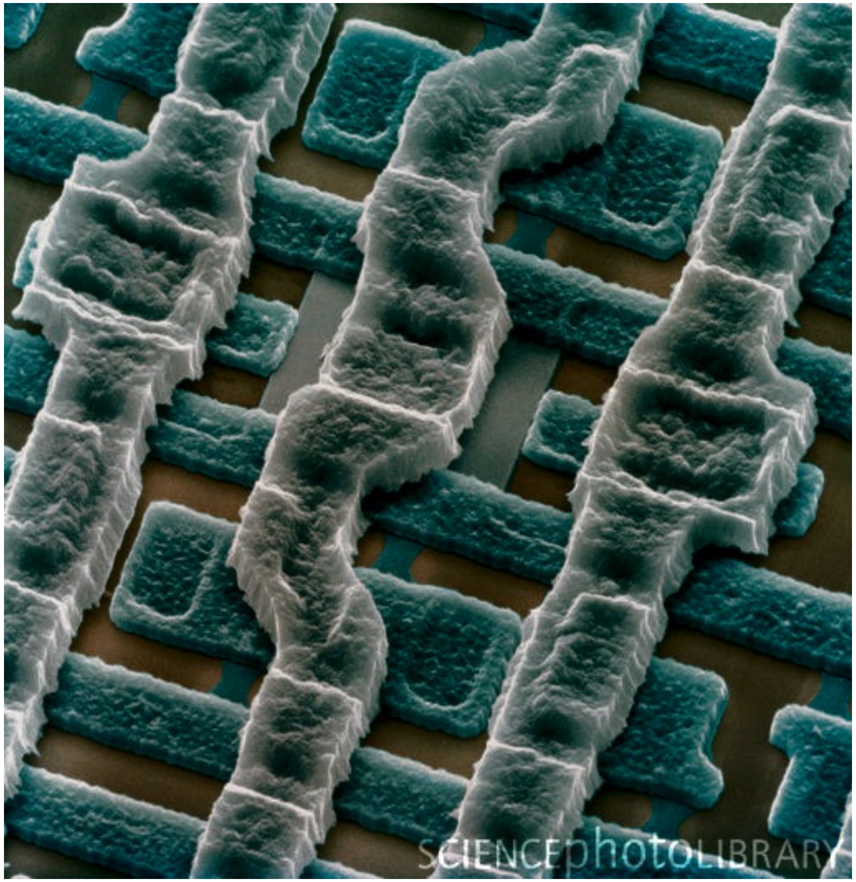
- Processor
- Cache memory
- Memory Controller
- WiFi
- Video
- Audio
- USB



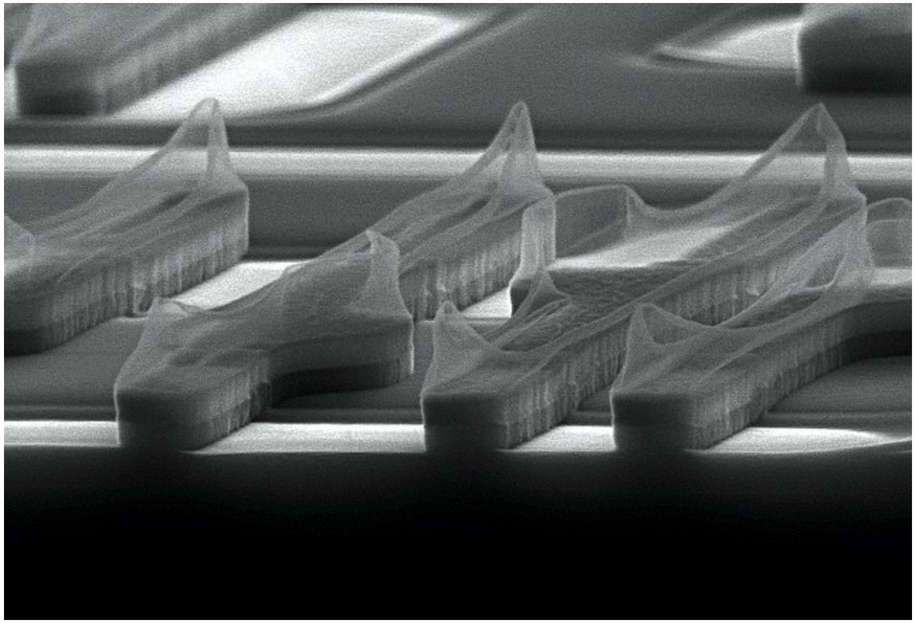
SEM Cross-Section of Apple A5

Digital Logic

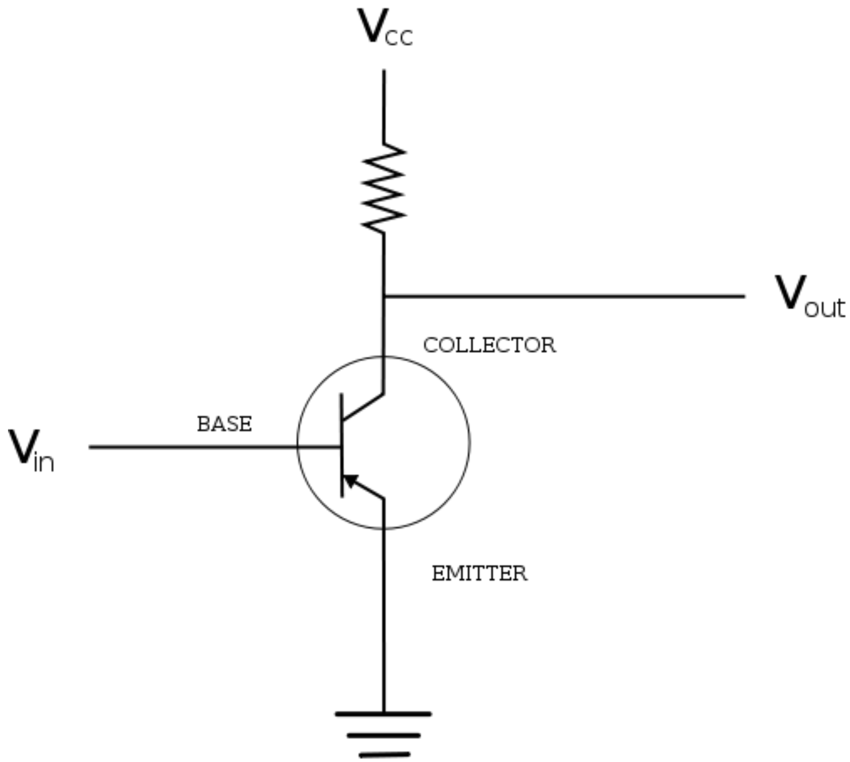
Memory cell



Transistor

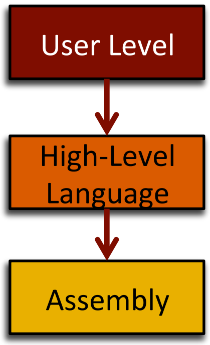


Transistors



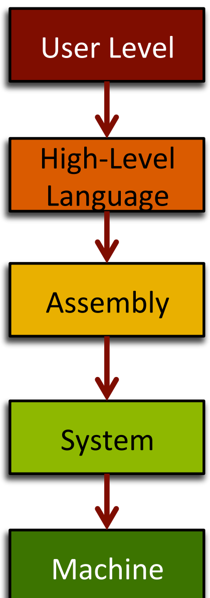
- You can still make assumptions at this level that the transistor is either "on" (1) or "off" (0)
- But below this is **analog circuits**

The Computer Level Hierarchy



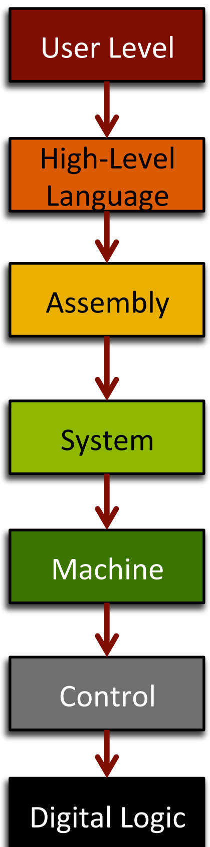
- Level 6: The **User Level** – "Angry Birds"
  - Program execution and **user interface** level
- Level 5: **High-Level Language Level** – "Objective C"
  - Programming languages like C++, Java, Python, …
- Level 4: **Assembly Language Level** – "ARM Assembly"
  - Program directly at this level, or …
  - **Use a compiler/interpreter** to process/convert high-level code

The Computer Level Hierarchy



- Level 3: **System Software Level** - “iOS”
  - Controls active programs and manages system resources
  - Assembly language instructions often pass through Level 3 without modification
- Level 2: **Machine Level**
  - Instruction Set Architecture (ISA) Level
  - Instructions are particular to the architecture of the specific machine (i.e. Intel processors, ARM processors, IBM processors…)

The Computer Level Hierarchy



- Level 1: **Control Level**
  - Decodes and executes instructions and moves data through the system
- Level 0: **Digital Logic Level**
  - Digital circuits, gates and wires implement the mathematical logic of all other levels

Course Overview

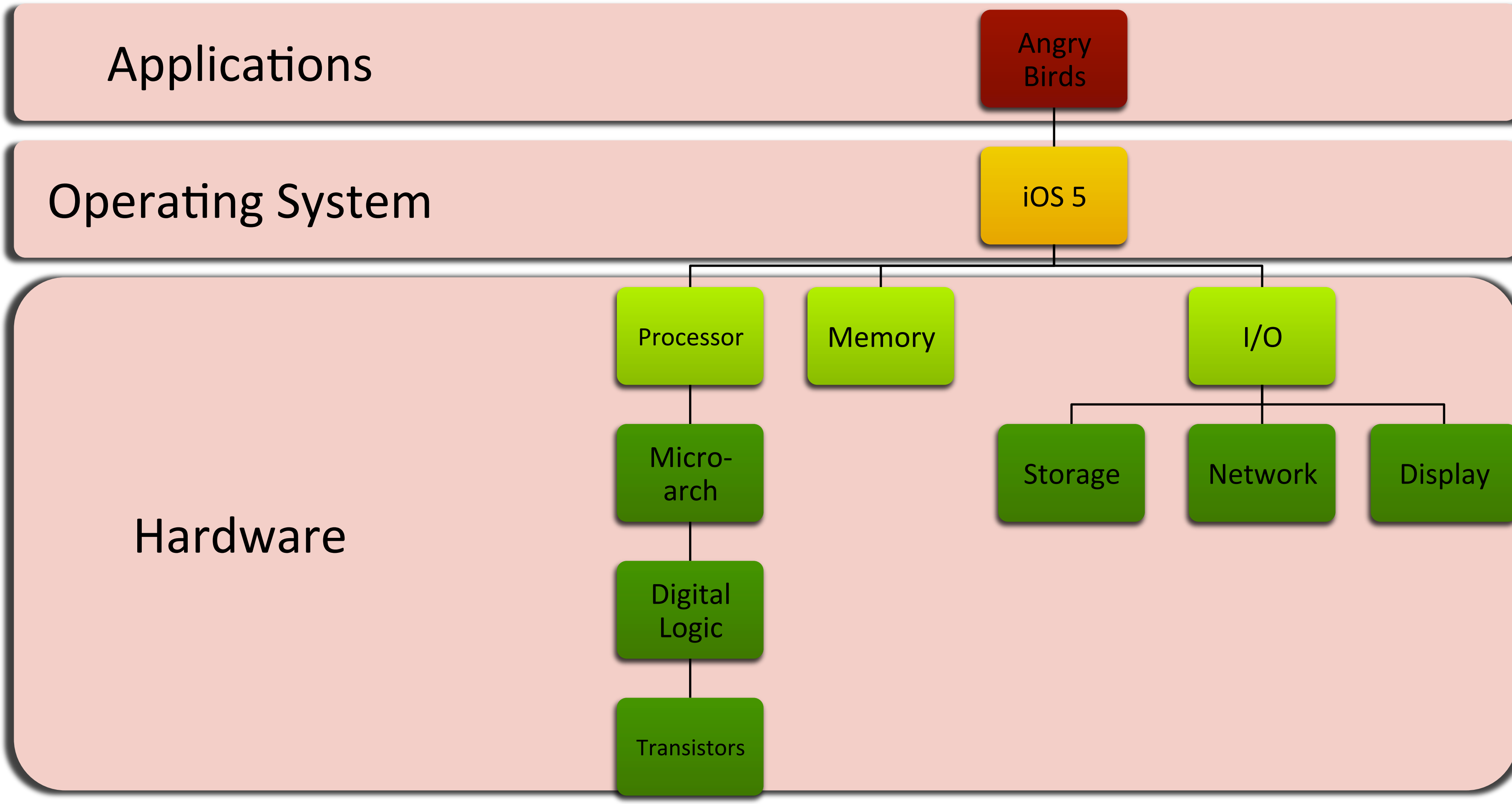
Overview

- Why study computer organization and architecture?
  - Design better programs and optimize their performance
    - Applications
    - Compilers
    - Operating Systems
    - Device Drivers
  - Evaluate (benchmark) computer system performance
  - Understand time, space, and price tradeoffs

ECPE 170 Course Goals

- Present a complete view of how computer systems are constructed
  - From the lowest level of hardware to the user application level
- Understand the relationship between computer software and hardware
- Lay the foundation for future courses
  - Digital design / VLSI
  - Operating systems
  - Computer networking
  - Application development

ECPE 170 Course Topics



Course Mechanics

Websites

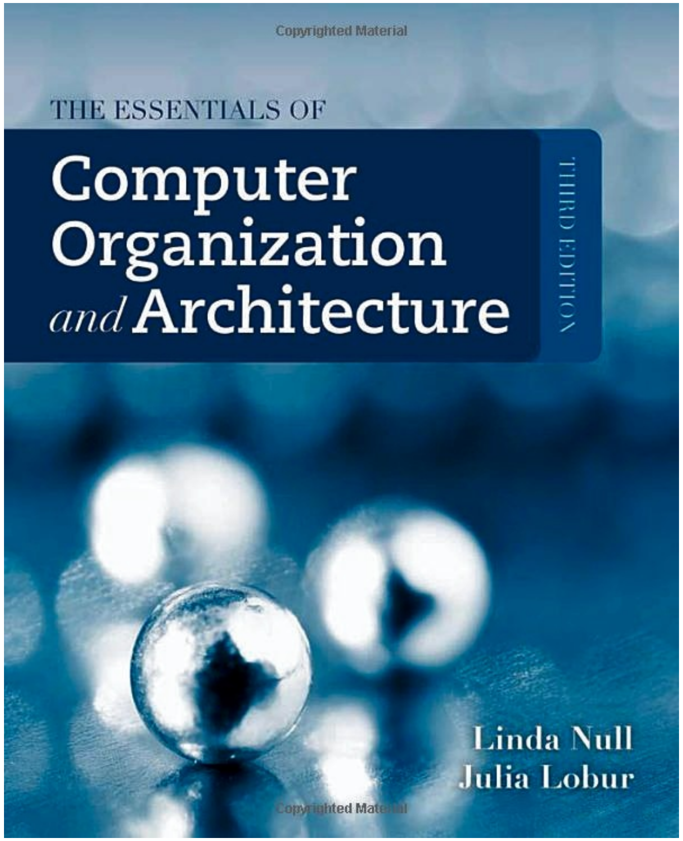
Main website (syllabus, schedule)

- http://ecs-network.serv.pacific.edu/ecpe-170

Sakai website (homework submission)

- http://pacific.rsmart.com/

Textbook



- *Computer Organization and Architecture* by Null/Lobur
- **Third** Edition
  - *If you buy a used copy:* Homework problems have been changed and reordered between the 2nd and 3rd edition – make sure you are doing the right problem!
- First homework set is assigned Thursday – get your book today!

Grading

- **Exams** – 60%
  - 4 exams (including the cumulative final)
  - Lowest grade is dropped
- **Quizzes** – 20%
  - 6 quizzes, drop the lowest grade
- **Homework** – 20%
  - Assigned almost every class

Homework

- Submit online via Sakai
- Due at the beginning of class
  - Late homework is not accepted
- Graded on correctness and an honest attempt to do the work
  - **Show work for partial credit!**
- "Make-up work" is not assigned, so be sure to turn in your homework

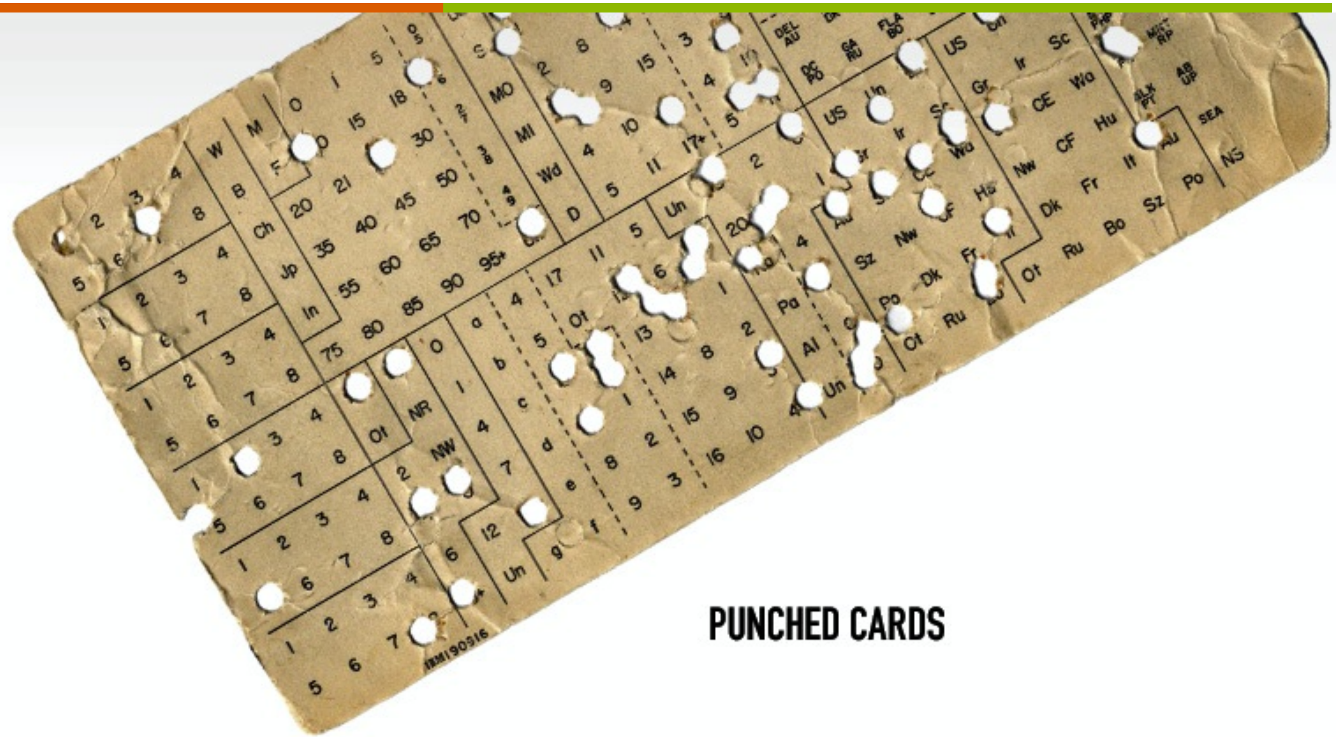
Computer History

Computer History

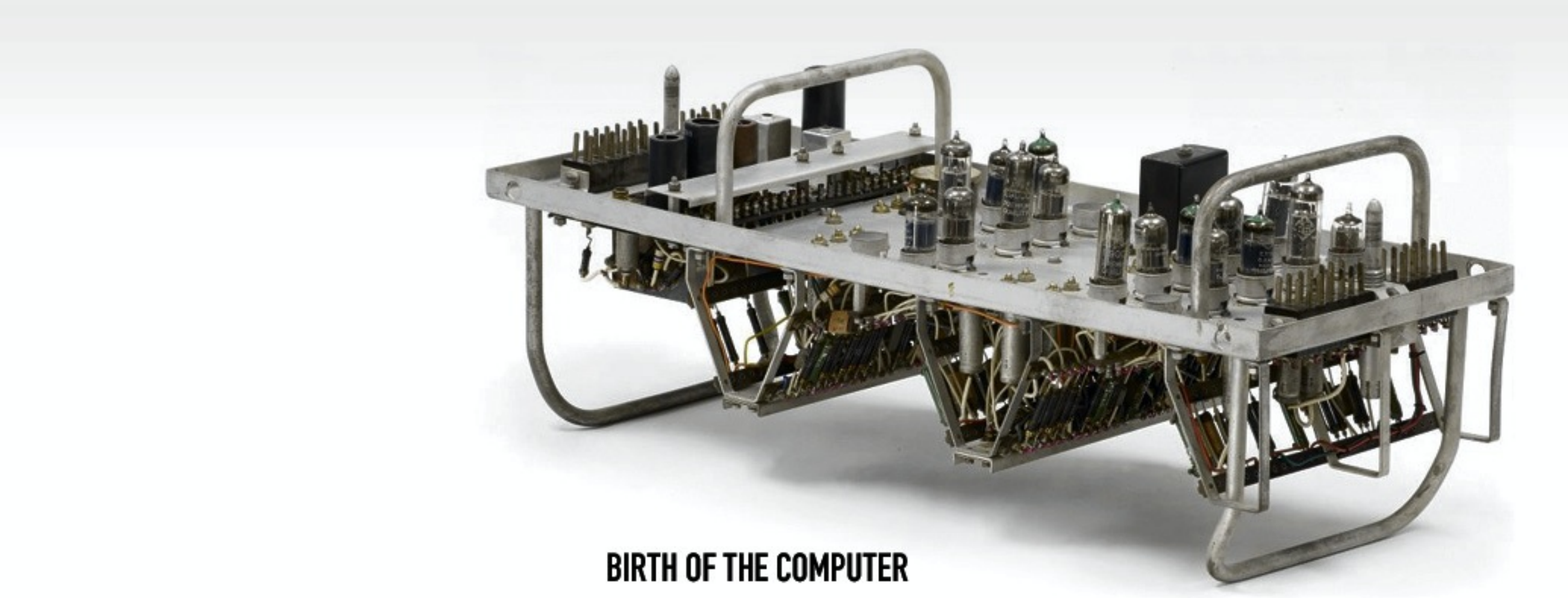
- **What is the first computer you remember using?**
  - **What did you use it for?**

Computer History Museum


CALCULATORS


PUNCHED CARDS


ANALOG COMPUTERS

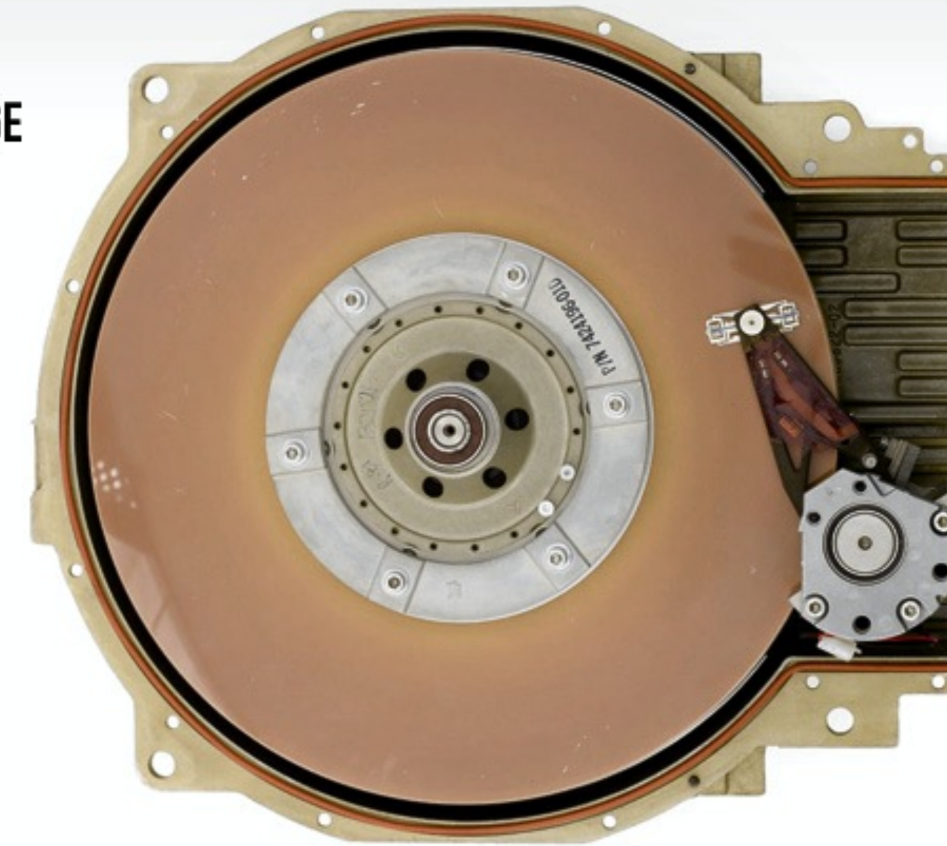

BIRTH OF THE COMPUTER

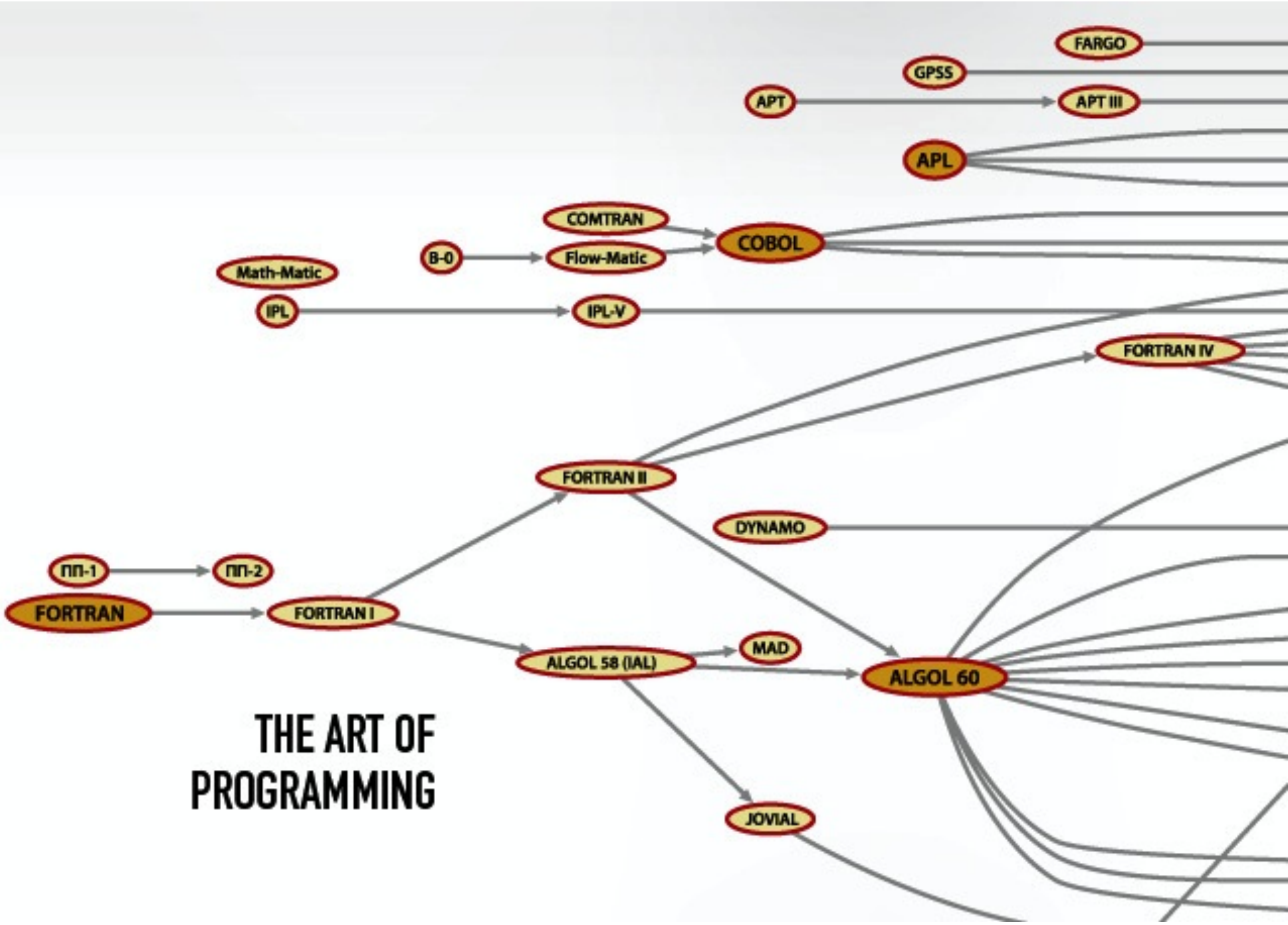
EARLY COMPUTER COMPANIES

REAL-TIME COMPUTING


MAINFRAMES

MEMORY AND STORAGE

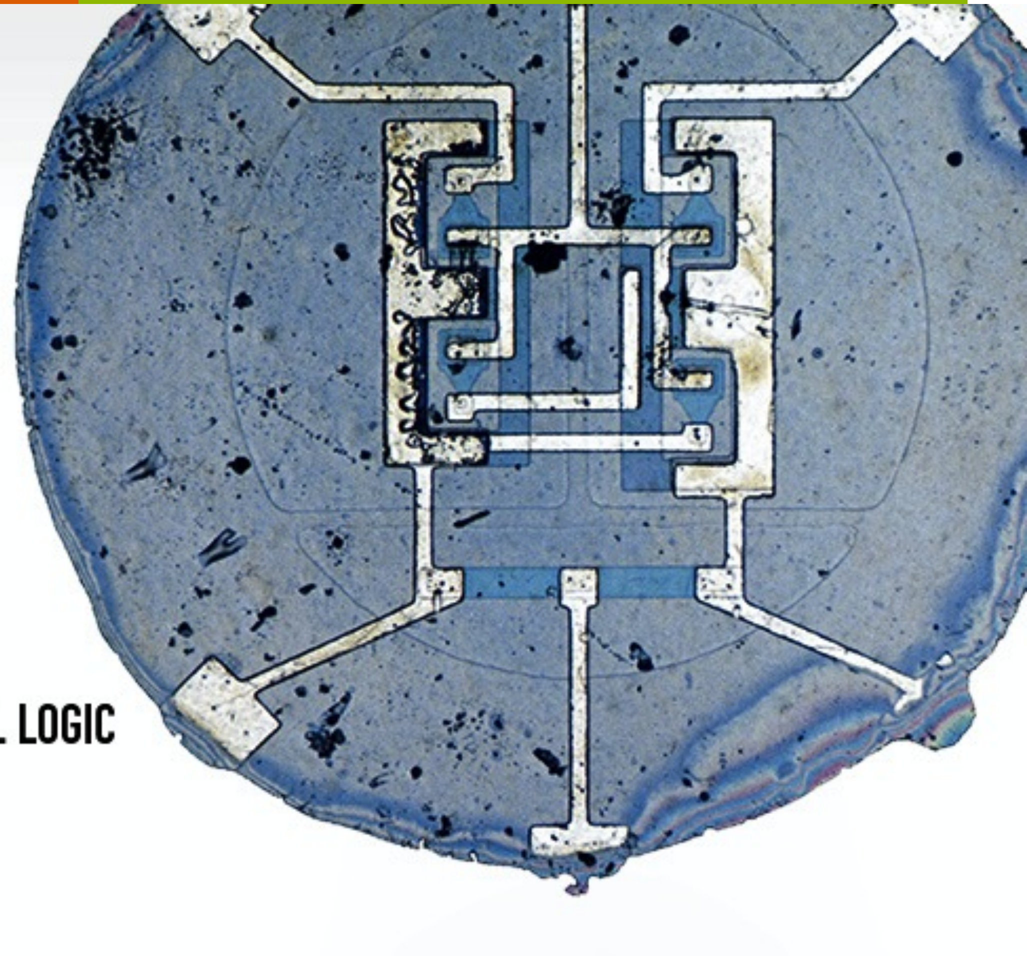


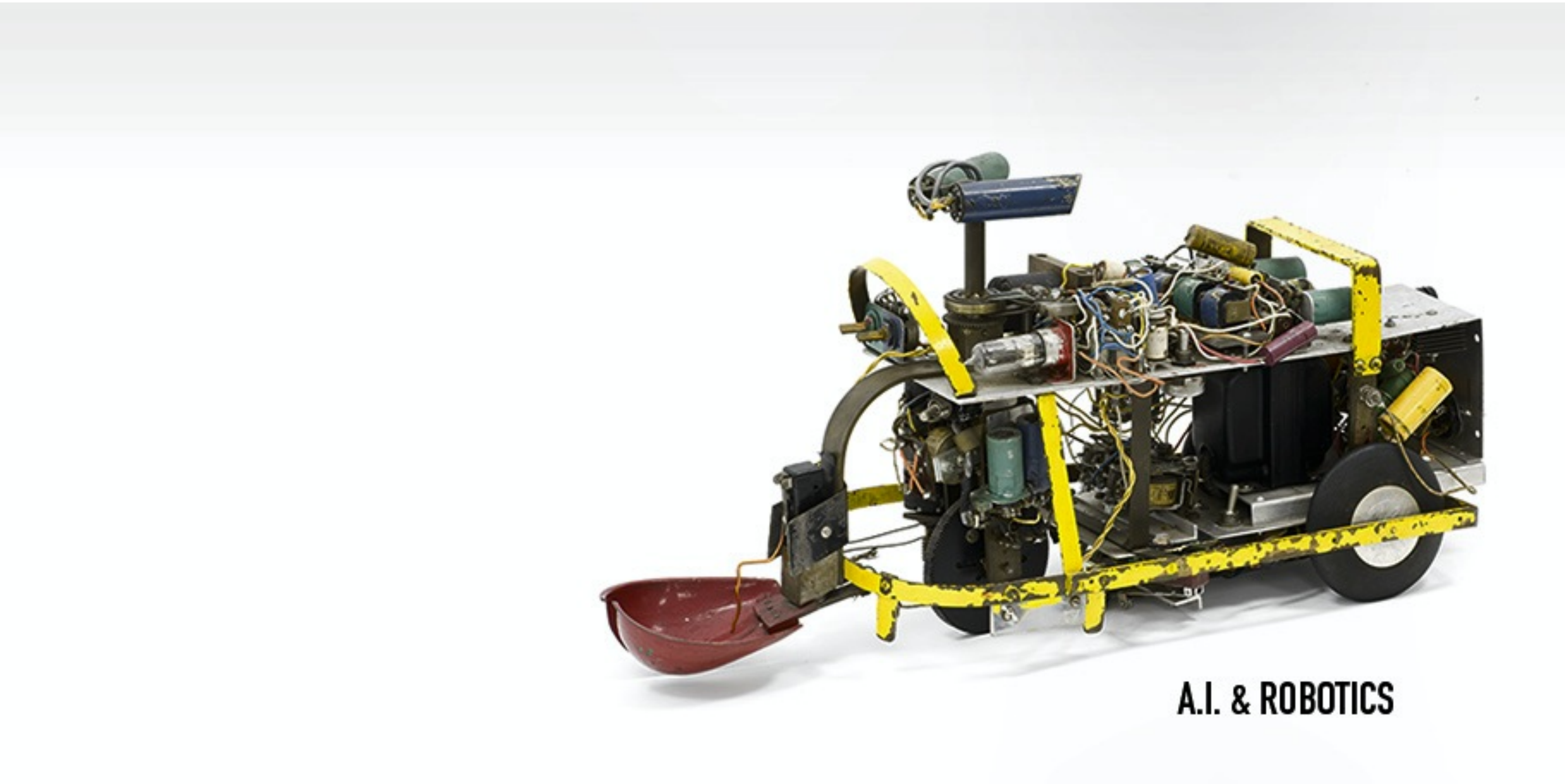

FARGO
GPSS
APT
APT III
APL
COMTRAN
COBOL
B-0
Flow-Matic
Math-Matic
IPL
IPL-V
FORTRAN IV
FORTRAN II
DYNAMO
ПП-1
ПП-2
FORTRAN
FORTRAN I
MAD
ALGOL 58 (IAL)
ALGOL 60
JOVIAL
THE ART OF
PROGRAMMING


SUPERCOMPUTERS

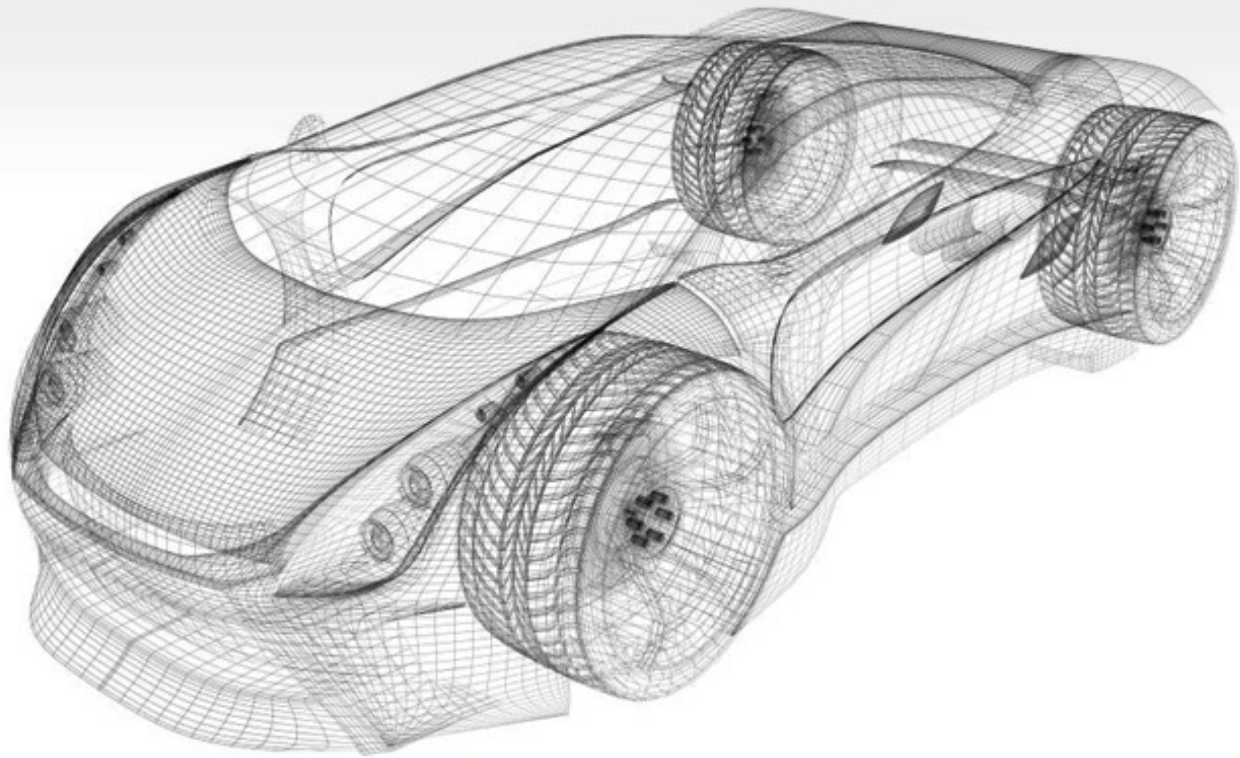
MINICOMPUTERS

DIGITAL LOGIC




A.I. & ROBOTICS

INPUT & OUTPUT



COMPUTER GRAPHICS, MUSIC, AND ART


COMPUTER GAMES


PERSONAL COMPUTERS

MOBILE COMPUTING


NETWORKING
HELLO
SEND
INT
GBYE

Computer History Museum

Mountain View, CA

Historical Development

- We can better understand modern computers by looking at how they developed through history
- Several centuries of computing "machinery"
- Classify computers into 4 generations based on key technological differences
  - Many dates are approximate – history is not black & white!

Historical Development

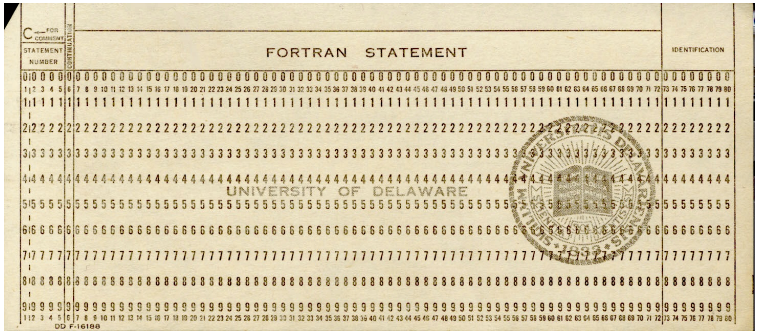
- **Generation Zero:** Mechanical Calculating Machines (1642 - 1945)
  - **Calculating Clock** – Early 1600's
    - Add/subtract numbers with 6 digits
    - Inventor died in a plague and his design was lost for centuries

Historical Development

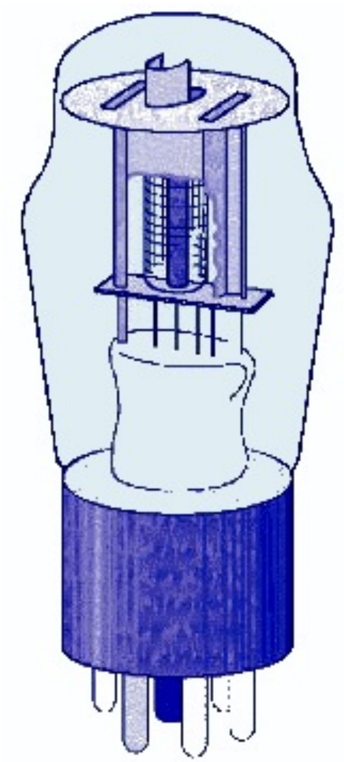
- **Generation Zero:** Mechanical Calculating Machines (1642 - 1945)
  - **Pascaline** – 1642
    - Add/subtract
    - Design used for hundreds of years!

Historical Development

- **Generation Zero:** Mechanical Calculating Machines (1642 - 1945)
  - **Difference Engine** – 1822
    - Solving polynomial equations
  - **Punched card tabulating machines**
    - First used in 1890 census
    - Punch cards were used for computer input up through the 1970's!



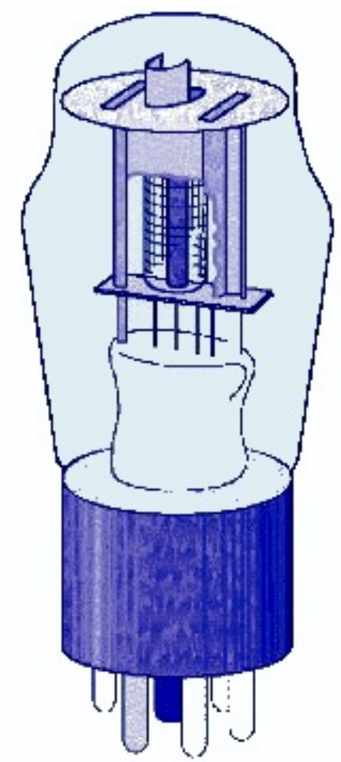
Historical Development

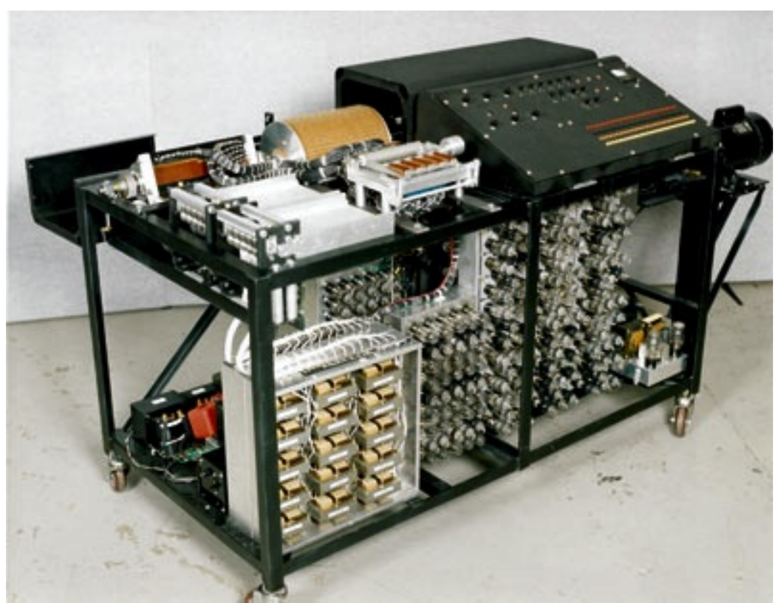


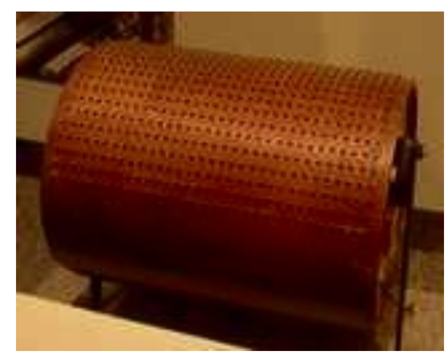
- **The First Generation**: Vacuum Tube Computers (1945 - 1953)
  - Vacuum tubes functioned as an *amplifier* and *switch*
    - Much faster than moving a mechanical switch!

Historical Development

- **The First Generation**: Vacuum Tube Computers (1945 - 1953)
- **Atanasoff Berry Computer** (1937 - 1938) solved systems of linear equations
  - Vacuum tubes for switches
  - Capacitors (on a physically rotating drum!) for memory

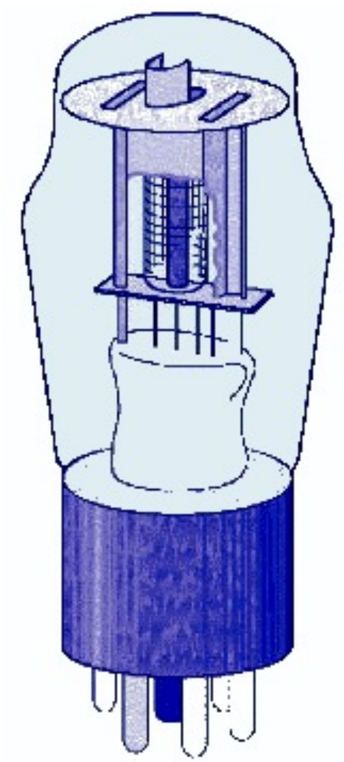


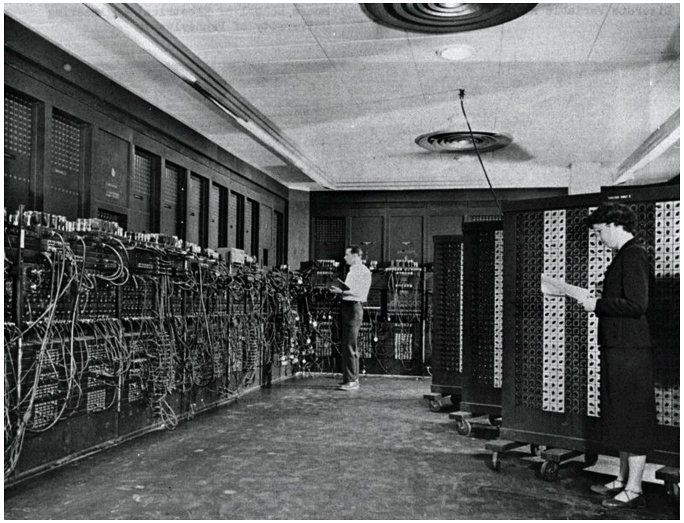


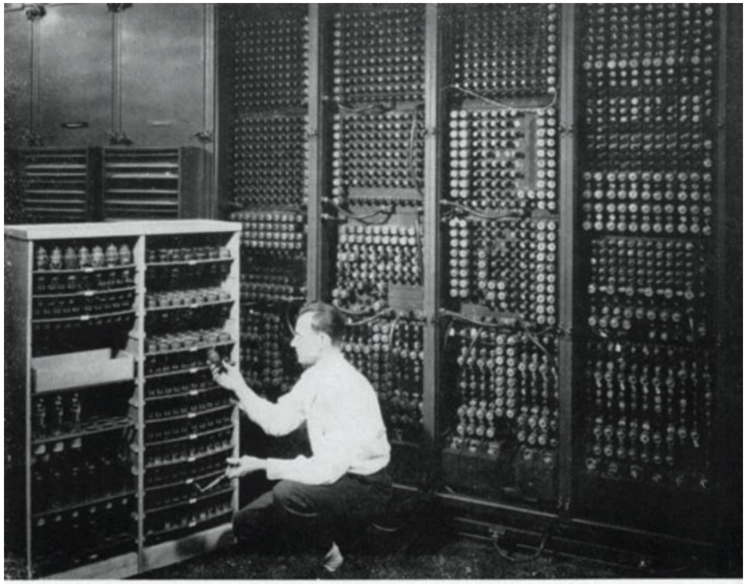


Historical Development

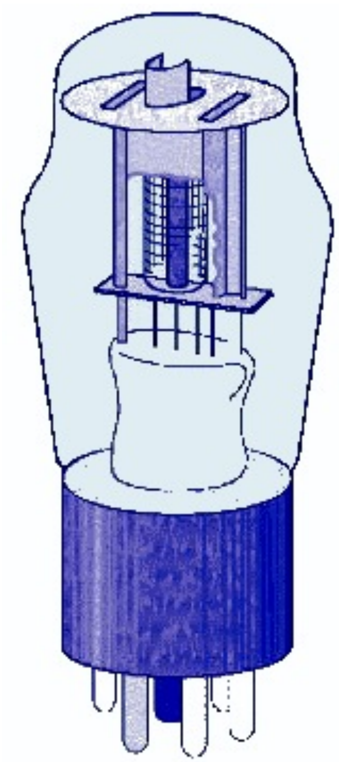
- **The First Generation:** Vacuum Tube Computers (1945 - 1953)
- **Electronic Numerical Integrator and Computer (ENIAC)** - 1946
  - First general-purpose computer!
  - 1000 bits of storage (~20 10-digit hex numbers)





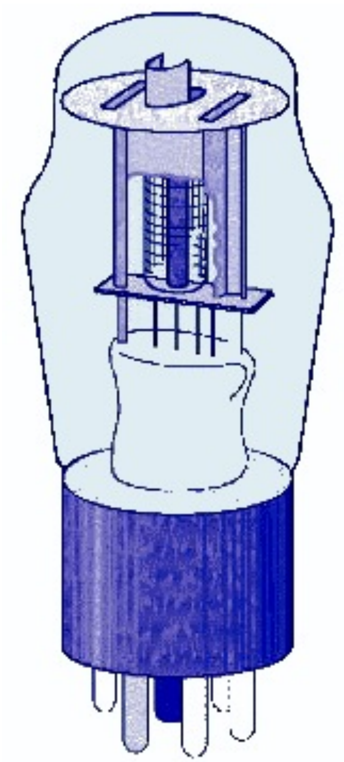


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.



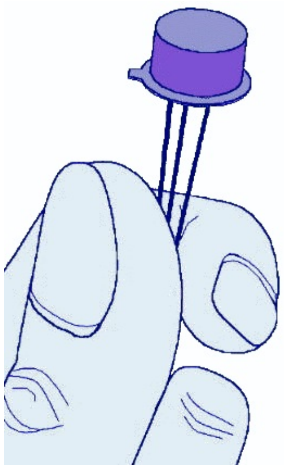

HOW MUCH IS $\sqrt[3]{2589^{16}}$?
The Army's ENIAC can give you the answer in a fraction of a second!
Think that's a stumper? You should see *some* of the ENIAC's problems! Brain twisters that if put to paper would run off this page and feet beyond . . . addition, subtraction, multiplication, division—square root, cube root, any root. Solved by an incredibly complex system of circuits operating 18,000 electronic tubes and tipping the scales at 30 tons!
The ENIAC is symbolic of many amazing Army devices with a brilliant future for you! The new Regular Army needs men with aptitude for scientific work, and as one of the first trained in the post-war era, you stand to get in on the ground floor of important jobs which have never before existed. You'll find that an Army career pays off.
The most attractive fields are filling quickly. Get into the swim while the getting's good! 1½, 2 and 3 year enlistments are open in the Regular Army to ambitious young men 18 to 34 (17 with parents' consent) who are otherwise qualified. If you enlist for 3 years, you may choose your own branch of the service, of those still open. Get full details at your nearest Army Recruiting Station.
YOUR REGULAR ARMY SERVES THE NATION AND MANKIND IN WAR AND PEACE
A GOOD JOB FOR YOU
U. S. Army
CHOOSE THIS FINE PROFESSION NOW!
33

Historical Development



- **The First Generation:** Vacuum Tube Computers (1945 - 1953)
- Significant drawbacks due to tube technology
  - Tubes are **fragile** and burn out within hundreds/ thousands of hours
  - Tubes are **hot** (need A/C)
  - Tubes are **power hungry** (ENIAC needed 174 kW)
  - Tubes are **large** (ENIAC took 1800 sq ft of space)
- Time for a better technology!

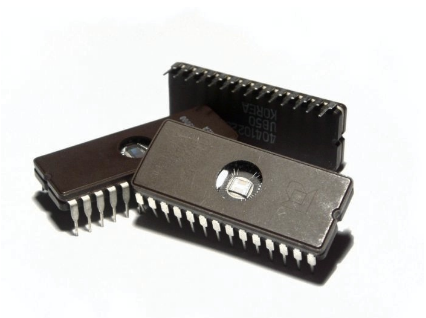
Historical Development



- **The Second Generation:** Transistorized Computers (1954 - 1965)
  - Transistors were much smaller, cooler and reliable
  - Systems still built in the same way as vacuum tube computers, but more compactly
- Examples
  - IBM 7094 (scientific) and 1401 (business)
  - Digital Equipment Corporation (DEC) PDP-1
  - Univac 1100
  - Control Data Corporation 1604
  - . . . and many others.

These systems had few architectural similarities

Historical Development

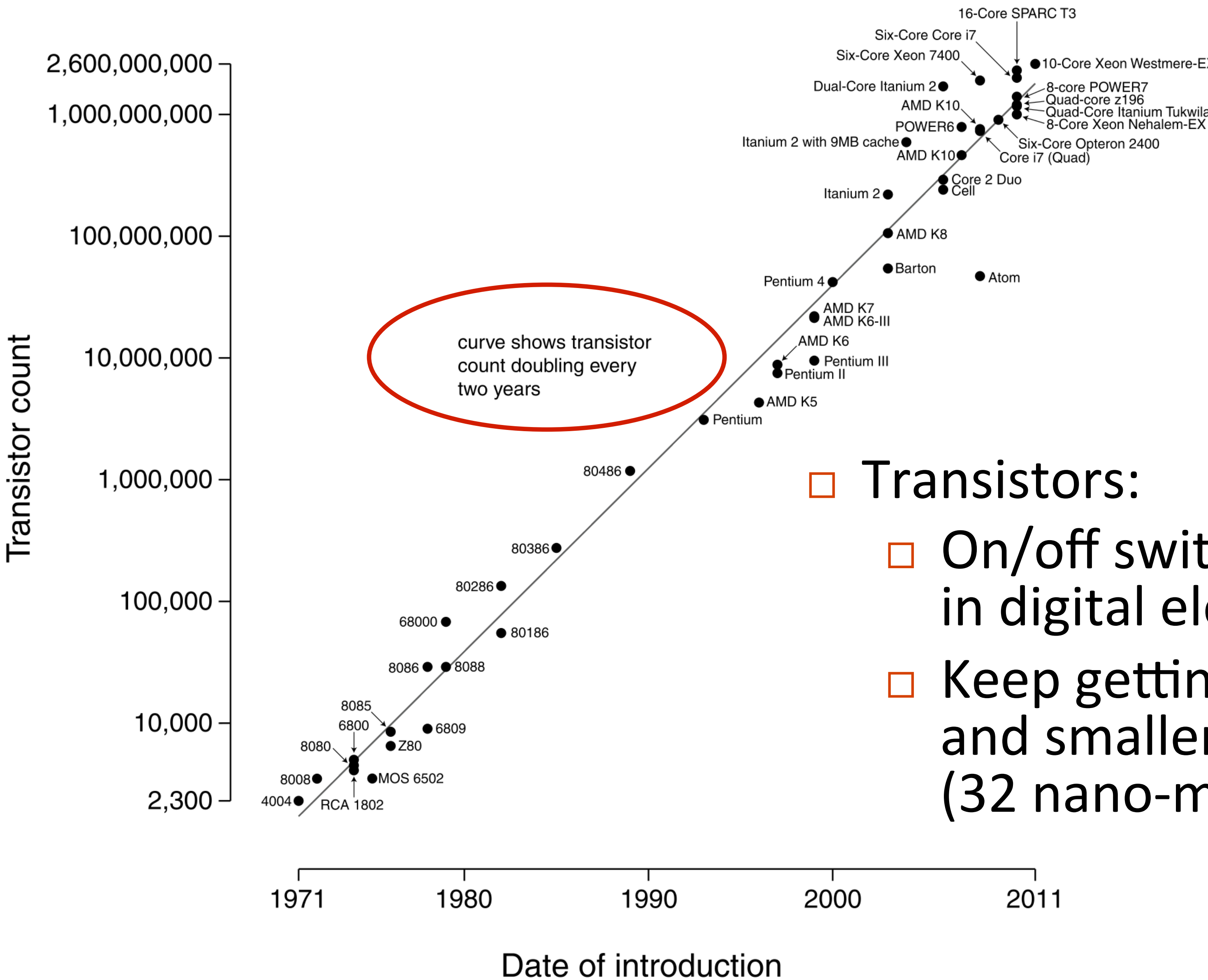


- The Third Generation: Integrated Circuit Computers (1965 – 1980)
  - Dozens to hundreds of transistors on a single chip
  - Build computers out of dozens to hundreds of chips
- Examples
  - IBM 360
    - Innovation – All computer models in this *family* used the same assembly language – thus, you could **re-use programs!**
    - IBM dominated the commercial marketplace
  - Cray-1 supercomputer
    - 8MB of memory for only $8.8 million!

Historical Development

- **The Fourth Generation:** VLSI Computers (1980 - ????)
  - Very large scale integrated circuits (VLSI) have more than 10,000 components per chip
  - Build microprocessors on a single chip
    - The first was the 4-bit Intel 4004
    - **4 bit?**
  - Later versions, such as the 8080, 8086, and 8088 spawned the idea of "personal computing"
- Transistors are getting smaller and smaller
  - **How far can this go?**

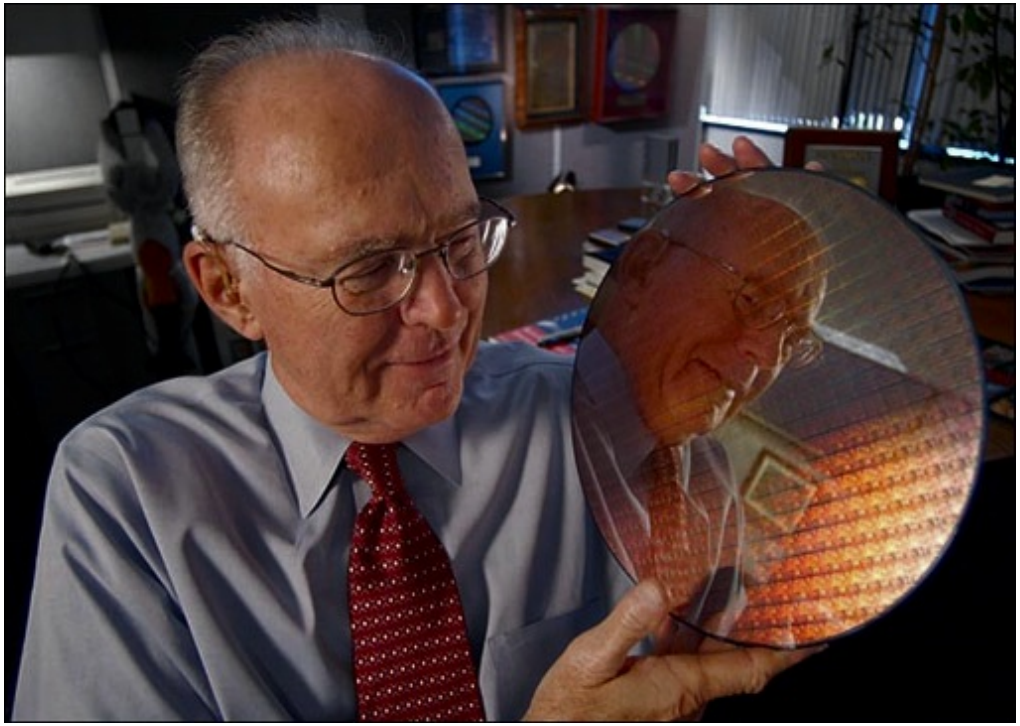
Microprocessor Transistor Counts 1971-2011 & Moore's Law



- Transistors:
  - On/off switches used in digital electronics
  - Keep getting smaller and smaller!
    (32 nano-meters)

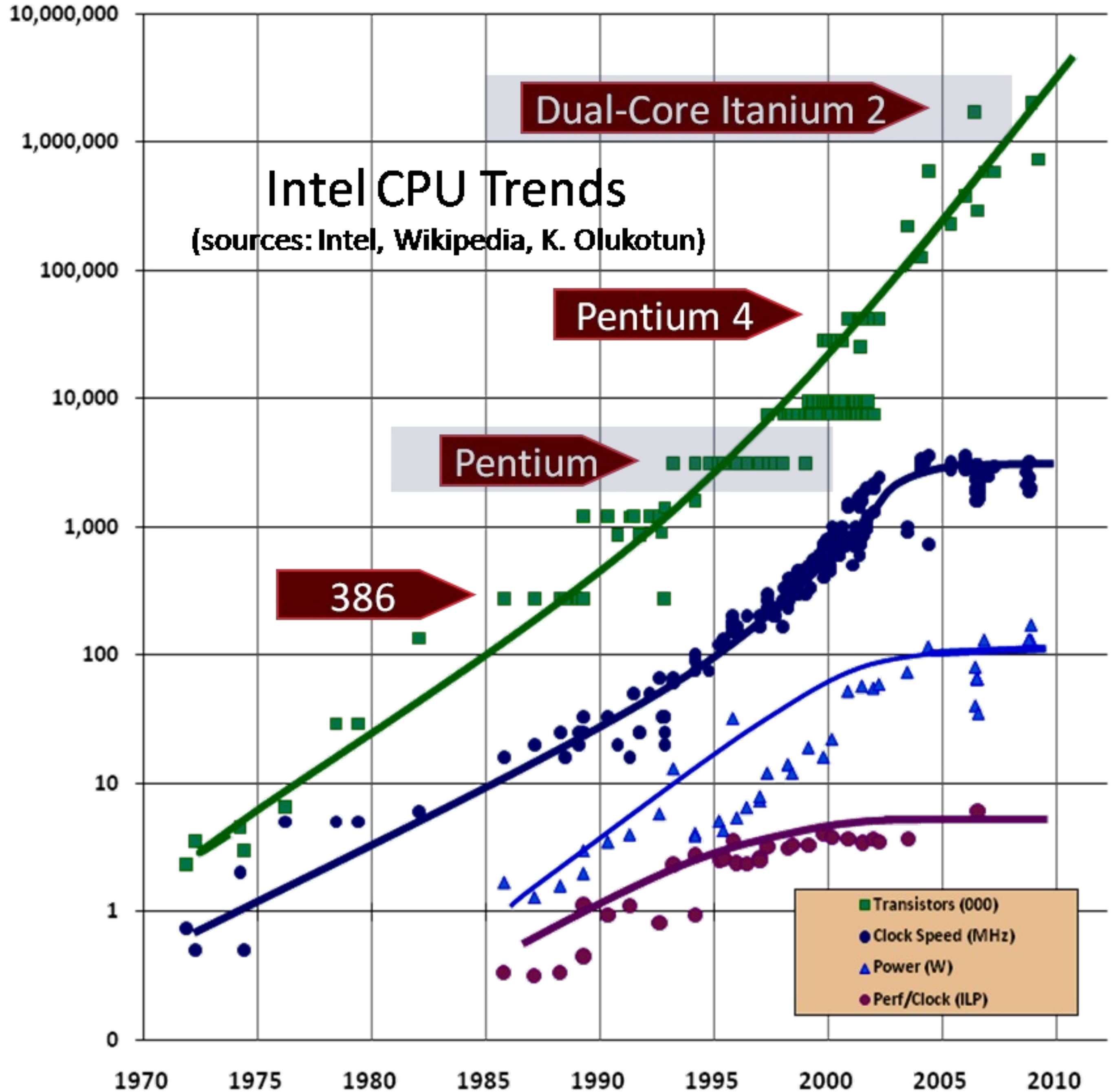
Moore's Law

- Gordon E. Moore
- Co-founder, Intel
- Proposed back in 1965
- Not a physical law!
  - An **observation of trends** in the semiconductor industry…



The "Law": The number of transistors available *at a given cost* doubles approximately every two years

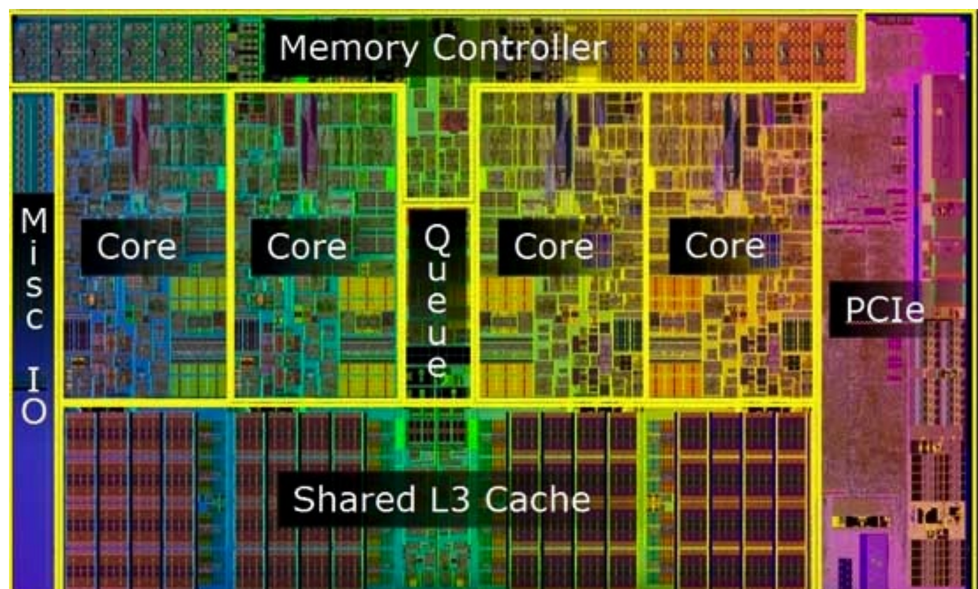
Where does Moore's Law end?



http://jai-on-asp.blogspot.com/2010/05/parallel-computing-and-net.html

What to do with a billion+ transistors?

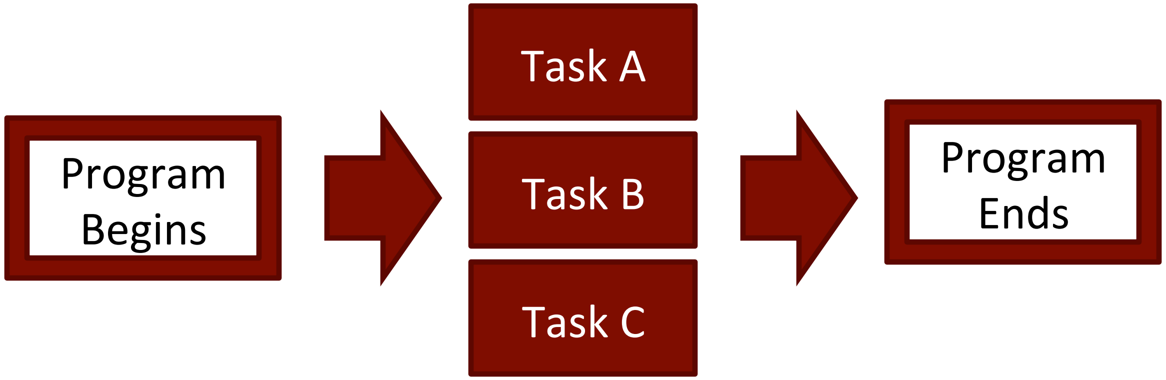
- Billions of transistors available today
  - More than we need to build a *single* fast and power-efficient (“cool running”) processor
- Let’s build **many processors** and put them on the same chip!
- How can we keep all the processors (“cores”) busy doing productive work?



What is Parallel Programming?

- Writing code with multiple "threads" of execution.
  - These threads can be assigned to different cores
- *Sequential execution* (what we have been doing so far) means that each task is executed one after the other

- *Parallel execution* means that tasks are done at the same time



Computer Operation

The von Neumann Model

- On the ENIAC, all programming was done at the digital logic level
- Programming the computer involved moving plugs and wires
- A different hardware configuration was needed to solve every unique problem type

Configuring the ENIAC to solve a "simple" problem required many days of work by skilled technicians

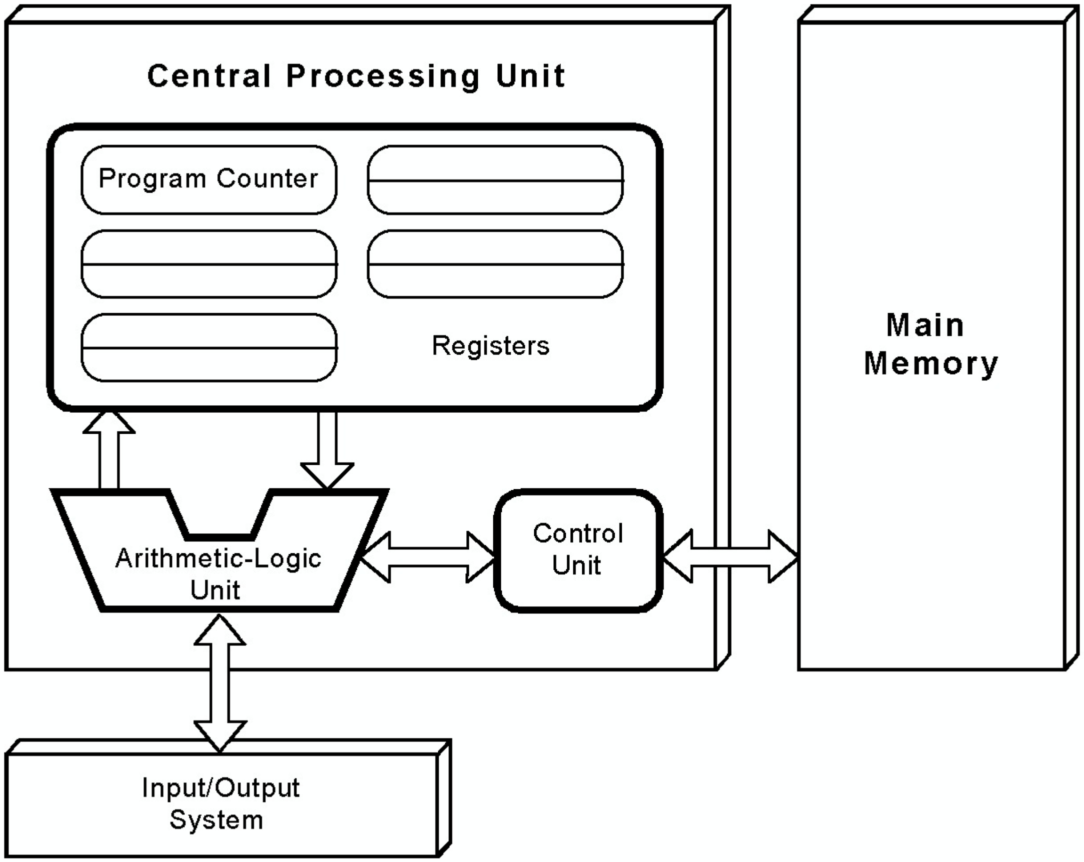
The von Neumann Model

- Inventors of the ENIAC (Mauchley and Eckert) conceived of a computer that could store instructions in memory
  - No need to re-wire the machine each time!
- First to publish this idea: John von Neumann
  - Contemporary of Mauchley and Eckert, who had to keep their ideas top secret (military)
- Stored-program computers have become known as *von Neumann Architecture* systems

The von Neumann Model

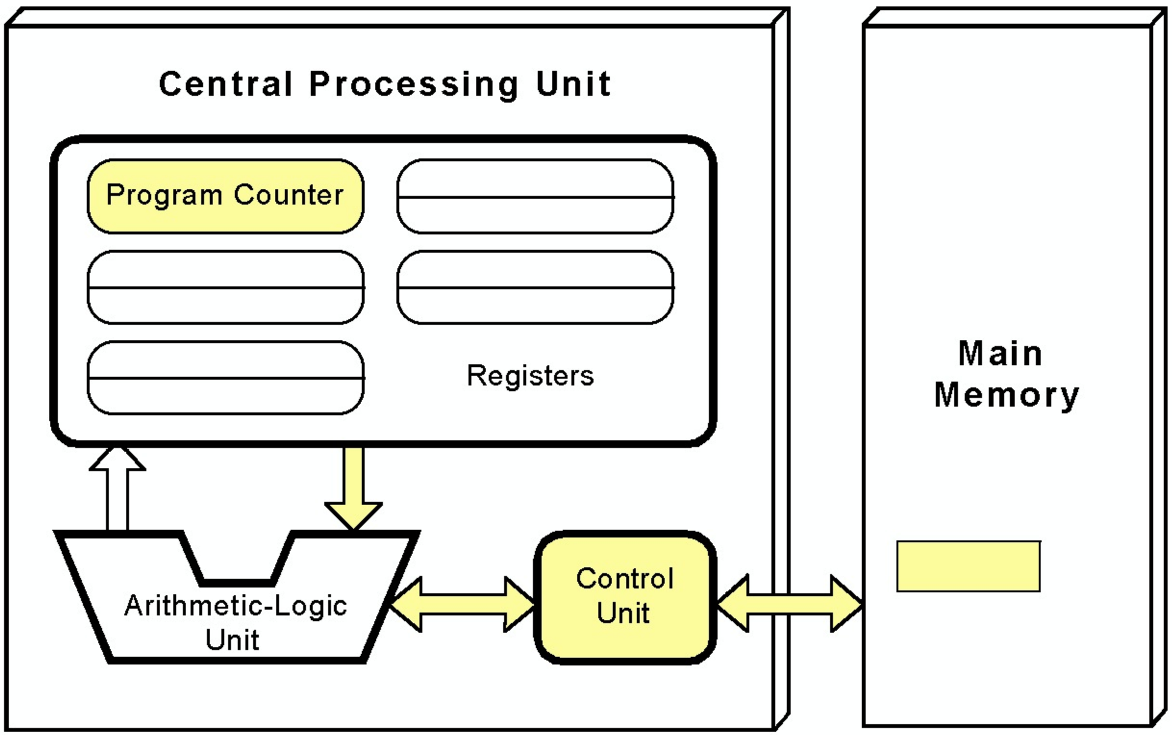
- Today's stored-program computers have the following characteristics:
  - Three hardware systems:
    - A **central processing unit** (CPU) to interpret programs
    - A **main memory** system to store programs & data
    - An **I/O** system to transfer data to/from the outside world
  - The capacity to carry out **sequential instruction** processing
  - A **single data path** between the CPU and main memory
    - This single path is known as the von Neumann bottleneck

The von Neumann Model



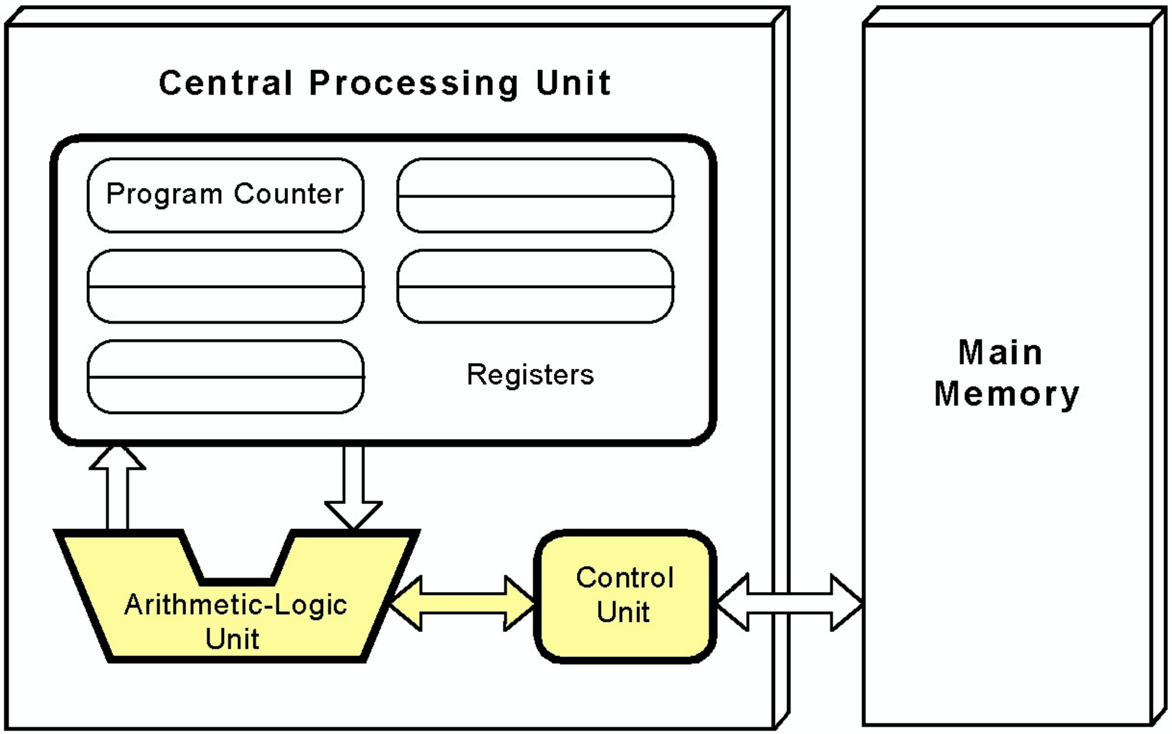
- This is a general depiction of a von Neumann system
- These computers employ a **fetch-decode-execute** cycle to run programs as follows . . .

The von Neumann Model



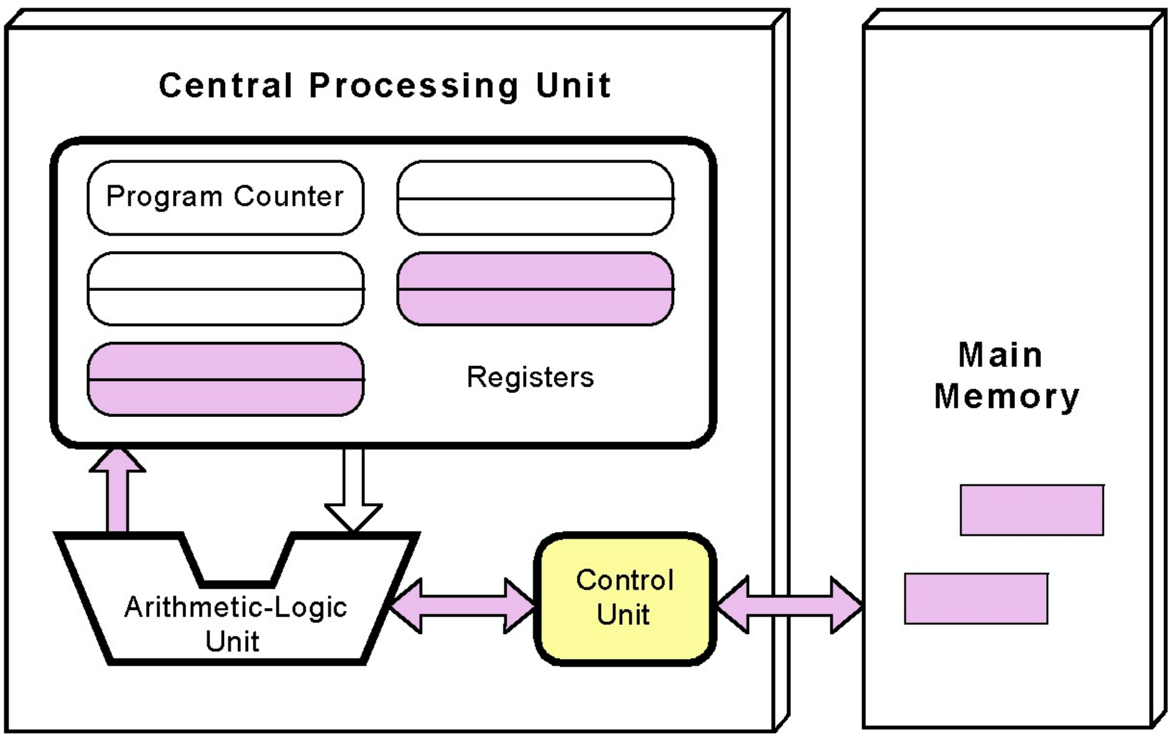
- The control unit **fetches** the next instruction from memory
- Which instruction?
  - Use the **program counter**

The von Neumann Model



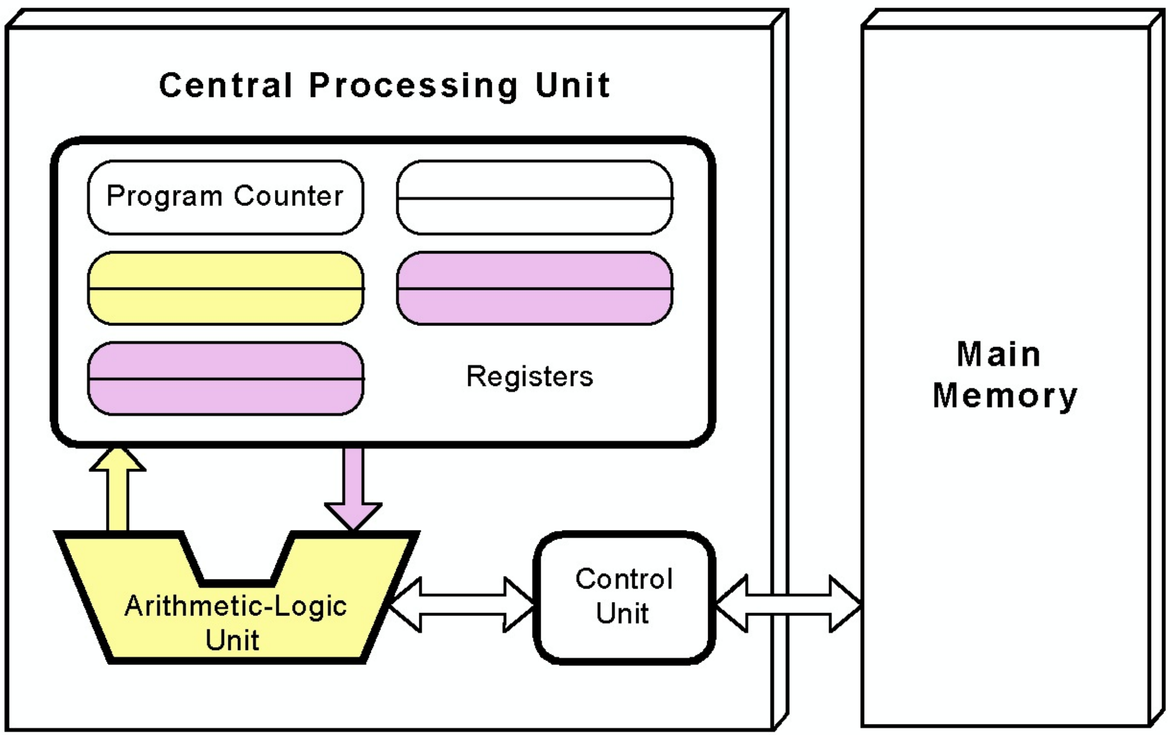
- The instruction is **decoded** into a language that the ALU can understand
  - Add?
  - Subtract?
  - Multiply?
  - Compare?
  - etc…

The von Neumann Model



- Any **data operands** required to execute the instruction are fetched from memory and placed into registers within the CPU
- Operands?
  - X = 3+5
  - *3 and 5 are operands…*

The von Neumann Model



- The ALU **executes** the instruction
- Results are placed back in memory (or temporary spots called **registers**)

Non-von Neumann Models

- Conventional stored-program computers have undergone many incremental improvements over the years
  - Specialized buses
  - Floating-point units
  - Cache memories
  - …
- Further improvements in computational power requires departure from the classic von Neumann architecture
  - One approach: Adding processors

Multi-Processor is an Old Idea!

- Late 1960s, dual processors used to increase computational throughput
- 1970s – Supercomputer systems introduced with 32 processors
- 1980s – Supercomputer systems built with 1,000 processors
- 1999 – IBM Blue Gene system with 1 million+ processors
- What is "new" is multiple processors in *your* PC

Next Class: Thursday

- Computer representations of numbers and letters
  - More than just conversion between decimal and binary
- First homework assigned
  - Get a copy of the textbook!

Units

Measures of Capacity and Speed

- Kilo- (K) = 1 thousand = 10^3 and 2^{10}
- Mega- (M) = 1 million = 10^6 and 2^{20}
- Giga- (G) = 1 billion = 10^9 and 2^{30}
- Tera- (T) = 1 trillion = 10^{12} and 2^{40}
- Peta- (P) = 1 quadrillion = 10^{15} and 2^{50}
- Exa- (E) = 1 quintillion = 10^{18} and 2^{60}
- Zetta- (Z) = 1 sextillion = 10^{21} and 2^{70}
- Yotta- (Y) = 1 septillion = 10^{24} and 2^{80}

Whether a metric refers to a power of ten or a power of two typically depends upon what is being measured.

Measures of Capacity and Speed

- Hertz = clock cycles per second (frequency)
  - 1MHz = 1,000,000Hz
  - Processor speeds are measured in MHz or GHz.
- Byte = a unit of storage
  - 1KB = 2^{10} = 1024 Bytes
  - 1MB = 2^{20} = 1,048,576 Bytes
  - 1GB = 2^{30} = 1,099,511,627,776 Bytes
  - Main memory (RAM) is measured in GB
  - Disk storage is measured in GB for small systems, TB (2^{40}) for large systems.

Measures of Time and Space

- Measures of time and space:
  - Milli- (m) = 1 thousandth = 10^{-3}
  - Micro- (μ) = 1 millionth = 10^{-6}
  - Nano- (n) = 1 billionth = 10^{-9}
  - Pico- (p) = 1 trillionth = 10^{-12}

Measures of Time and Space

- Millisecond = 1 thousandth of a second
  - Hard disk drive access times are often 10 to 20 milliseconds.
- Nanosecond = 1 billionth of a second
  - Main memory access times are often 50 to 70 nanoseconds.
- Micron (micrometer) = 1 millionth of a meter
  - Circuits on computer chips are measured in microns.