310-2202 โครงสร้างของระบบคอมพิวเตอร์
(Computer Organization)

Topic 1: Introduction to Computing Systems

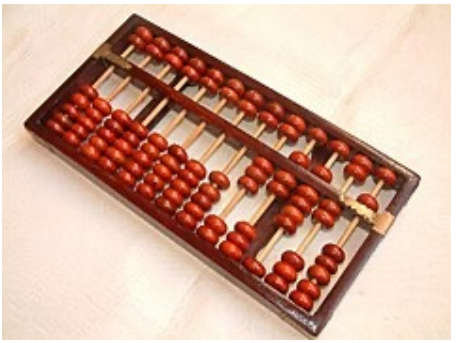
Damrongrit Setsirichok

310-2202 โครงสร้างของระบบคอมพิวเตอร์ (Computer Organization)

- การใช้งานระบบคอมพิวเตอร์ **ส่วนประกอบ**ของระบบคอมพิวเตอร์ **ประสิทธิภาพ**ของระบบคอมพิวเตอร์และตัวประมวลผล การใช้พลังงานของคอมพิวเตอร์ **สถาปัตยกรรมและชุดของภาษา**แอสแซมบลี การเขียนภาษาแอสแซมบลี ความสัมพันธ์ของภาษาแอสแซมบลี และภาษาระดับสูง การแสดงระบบเลข**จำนวนเต็มและเลขทศนิยม** พื้นฐานของ**ดิจิทัล**สำหรับการคำนวณ การเกิดโอเวอร์โฟล์ **โครงสร้างตัวประมวลผล**อย่างง่าย โครงสร้างตัวประมวลผลแบบไพพ์ไลน์ และการจัดการการทำงานของไพพ์ไลน์ **ระบบหน่วยความจำ**แบบลำดับชั้นหน่วยความจำแคช หน่วยความจำหลัก และหน่วยความจำเสมือน ระบบ**อินพุตเอาต์พุต**ต่อเชื่อมภายนอก **ตัวประมวลผลแบบขนาน** การรองรับ**เทคโนโลยีเสมือน** โครงสร้างและคุณลักษณะของตัวประมวลผลสำหรับระบบสมองกลฝังตัว

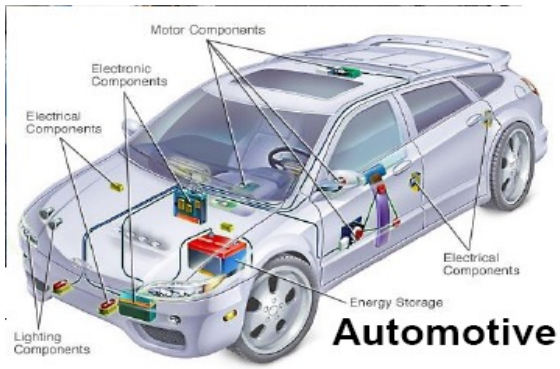
Questions

- How powerful are today's computers? Why are they so powerful?
  - What can computers do?
- Is an abacus a computer? How to understand Turing's contribution to computers?
  - How are they done?
- What are the serious flaws in today's computers?
  - What can't computers do?



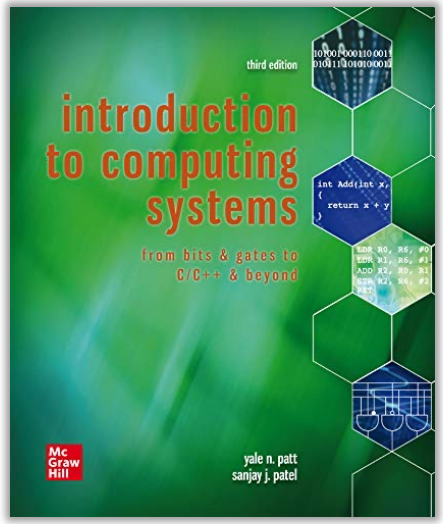
What is a computer?

- Everything



Book

- Part 1: Hardware (Chapter 1-4)
  - Representing data, transistors, gates, digital logic structures
  - von Neumann machine model
- Part 2: Software: Assembly language (Chapter 5-10)
  - Instructions, (structured) programming, input/output, *relationship to hardware*
- Part 3: Software: C programming (Chapter11-19)
  - Syntax, operators, control structures, functions, *pointers*, recursion, data structures, *relationship to assembly language*
  - Assume already familiar with programming (C)



Introduction to Computing Systems : from bits and gates to C and beyond; Yale N. Patt and Sanjay J. Patel;

Mc-Graw Hill, 2020, 3rd ed.

Focus on

- Chapter 2 Bits and Bytes
  - How do we represent information using electrical signals? (Low/High)
- Chapter 3 Digital Logic
  - How do we build circuits to process information?
- Chapter 4, 5, 6 Computer Machine Model, Processor and Instruction Set
  - How do we build a processor out of logic elements?
  - What operations (instructions) will we implement?
- Chapter 7 Assembly Language Programming
  - How do we use processor instructions to implement algorithms?
  - How do we write modular, reusable code? (subroutines)
- Chapter 8, 9, 10 I/O, Traps, and Interrupts
  - How does processor communicate with outside world?
- Chapter 11, C Implemention related to Hardware

Background / Prerequisites

- Requirement
  - Background in programming(C)
  - Assume you can program/debug in C
  - Digital Logic (Co-requisite)
  - Intention

LAB

- Little Computer 3 (LC-3) ISA Simulator/Assembler
  - Simplified Design: small enough to be described in a few pages
  - Educational Focus
- RISC-V
  - Research and Industry Use: both educational and practical purposes
  - Scalable and Extensible: suitable for a wide range of applications from small embedded systems to large supercomputers
  - Open-source

** RISC-V (Reduced Instruction Set Computer V)

**ARM (Advanced RISC Machines)

Two Very Important Ideas

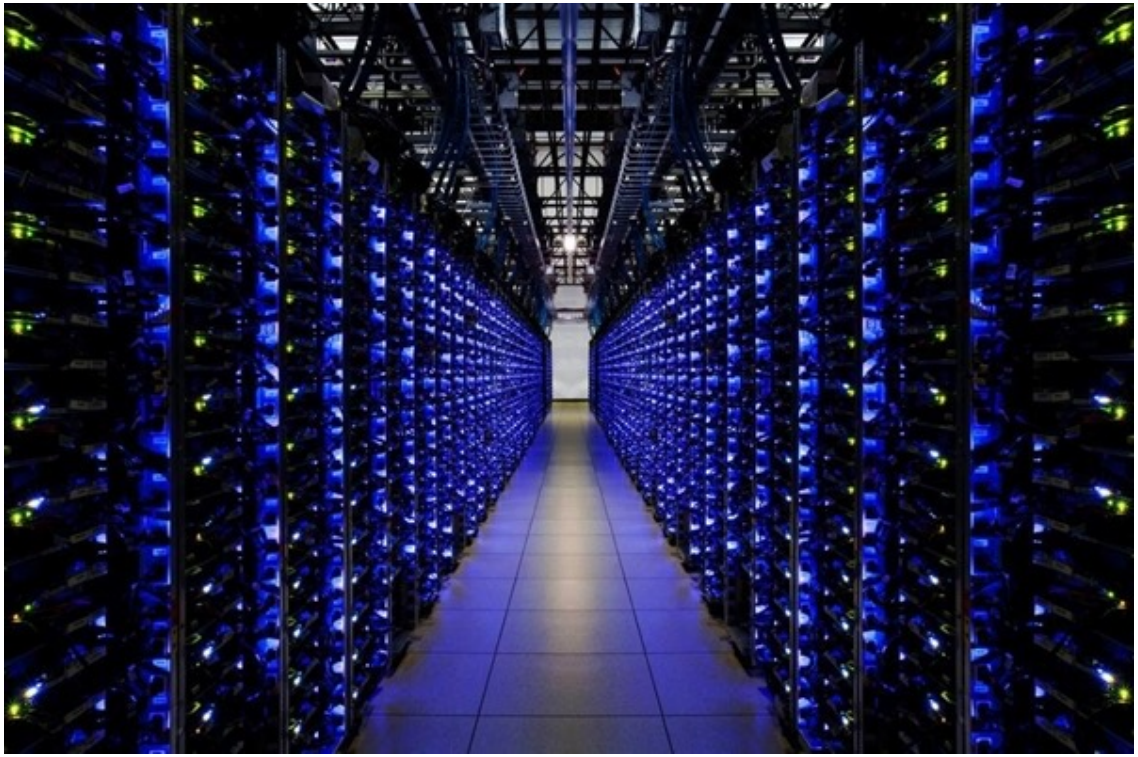
- **Idea 1:** All computers (the biggest and the smallest, the fastest and the slowest, the most expensive and the cheapest)
  - All computers can do exactly the same things.
  - Some computers can do things faster, but none can do more than any other.
- **Idea 2:** We describe our problems in English, or some other language spoken by people.
  - Very complicated tasks to transform our problem from the language of **humans to the voltages**

Turning a screw 35 degrees counterclockwise: $ 0.75
Knowing which screw to turn and by how much: 999.25

What's the difference between “Big” and “Small”?

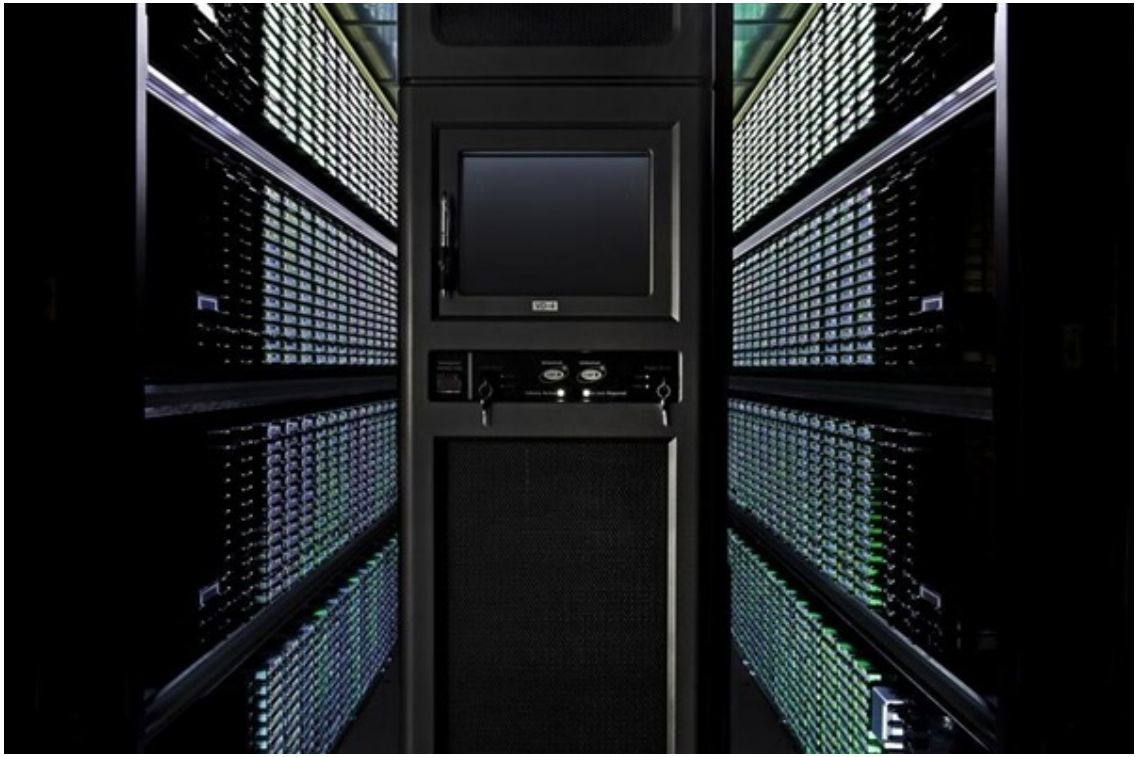
https://en.wikipedia.org/wiki/Frontier_(supercomputer)

LEDs indicator on the server to monitor the normal operation of the system



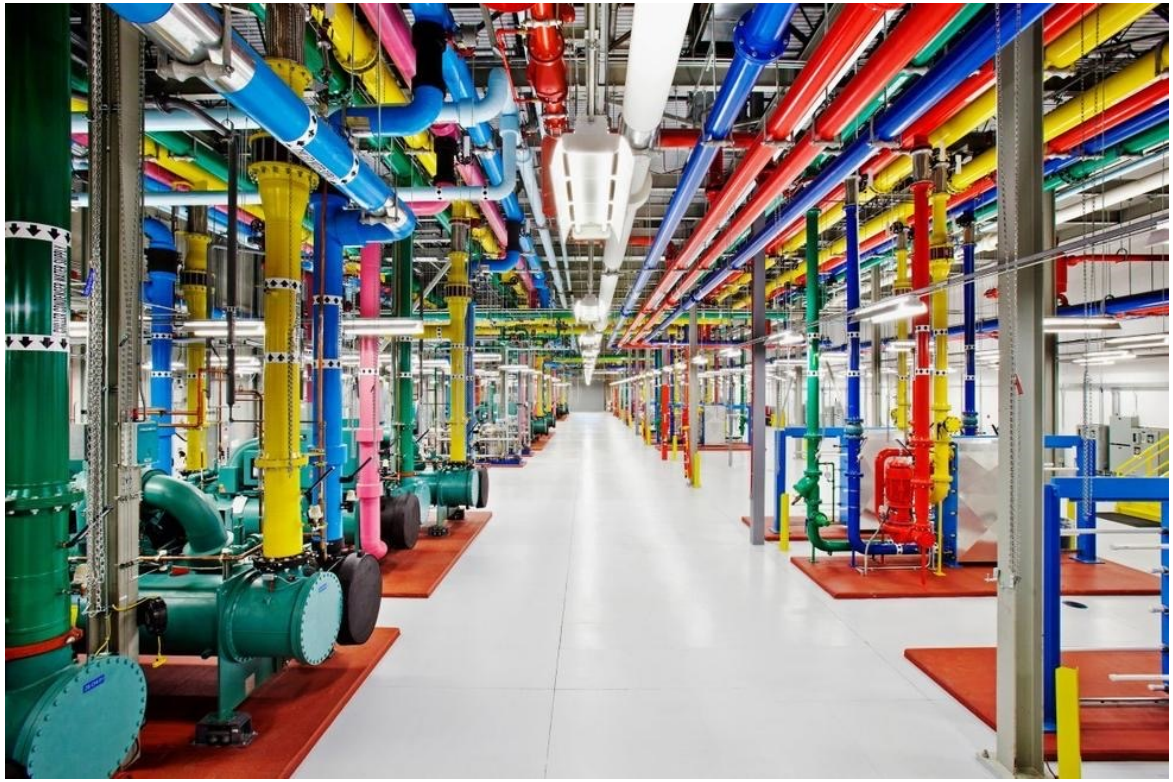
High-speed routers and switches built into the data center

A robotic arm designed to help easily replacing the damaged hard disks



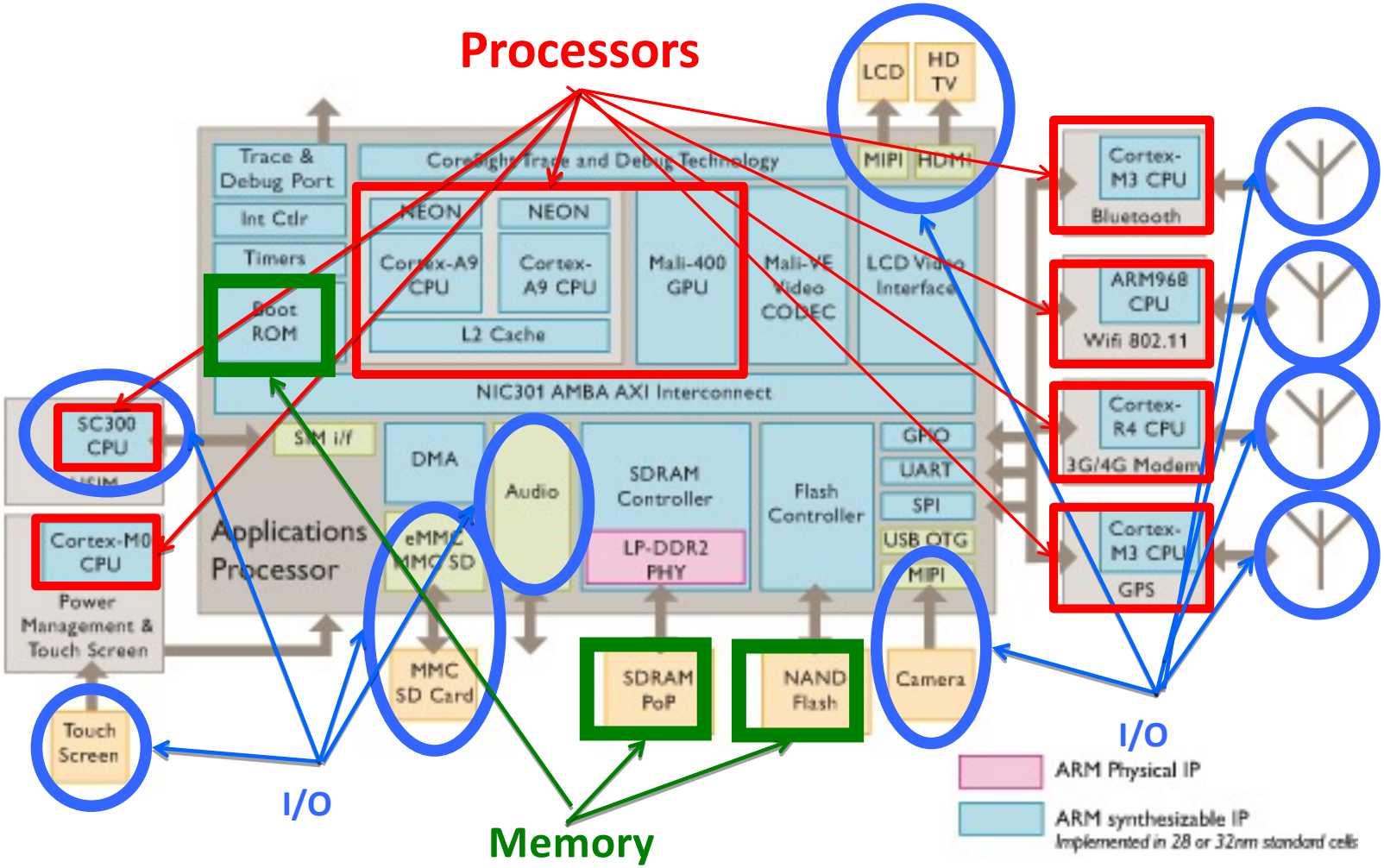
The hard disks replaced by a robotic arm

A piping system for dissipating heat in a data center



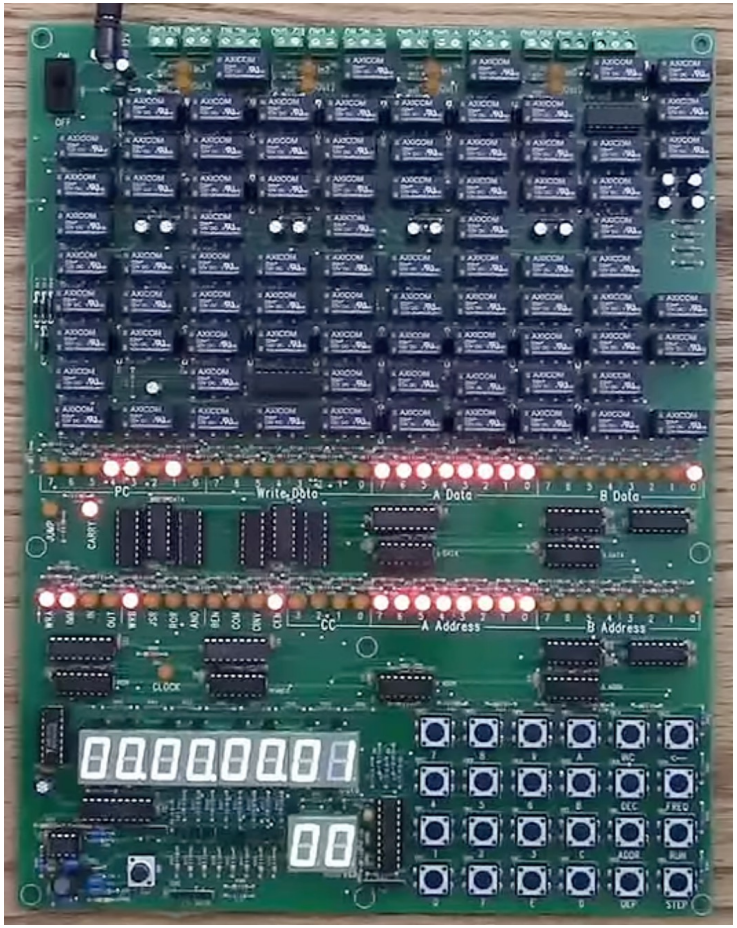
A large air duct used to dissipate heat in a data center

“Small” Computer Inside



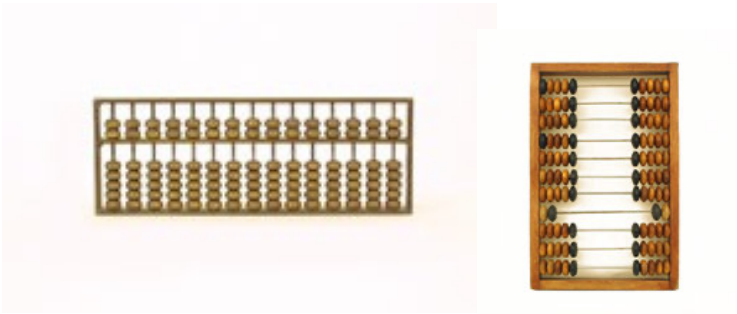
At the beginning

- https://www.youtube.com/watch?v=k1hJoalcK68&list=PLblu-CA4SK5KQ7MTzGwAtEF1Zq_NKXMex



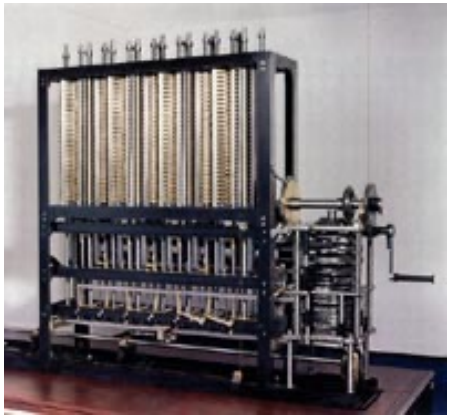
A computing tool that does not use electricity

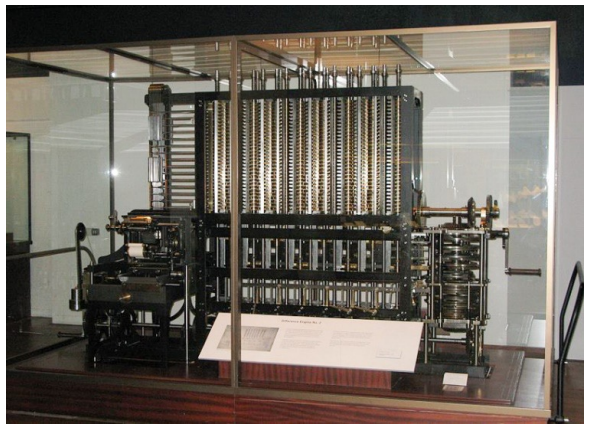
- Abacus



- Slide Rules

Babbage difference engine: the first mechanical computer, (1832)



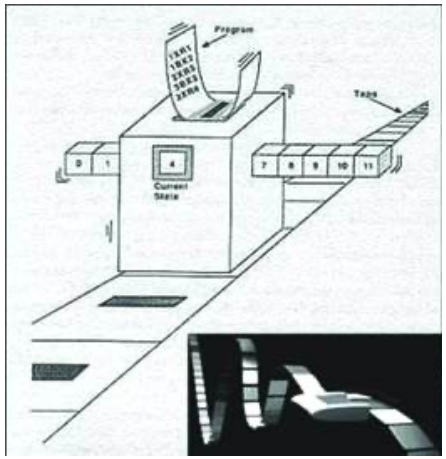


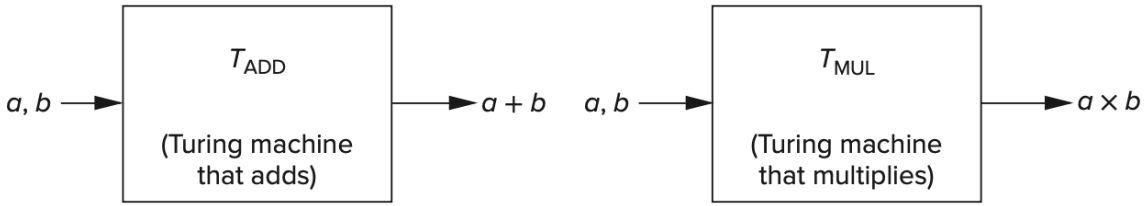
- Difference Engine No.2, 17 years later
- cost: £17,470

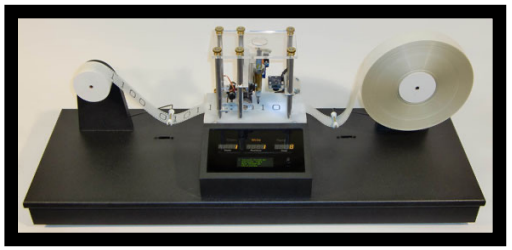
Turing machine (1937)

Alan Turing

- **Mathematical model of a device that can perform any computation** – Alan Turing
  - ability to read/write symbols on an infinite "tape"
  - state transitions, based on current state and symbol
- Every computation can be performed by some Turing machine. (Turing's thesis)

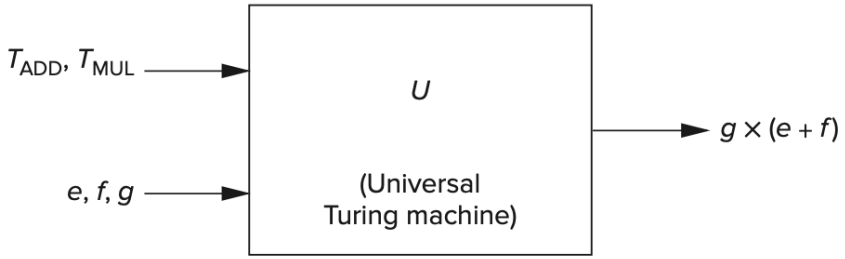






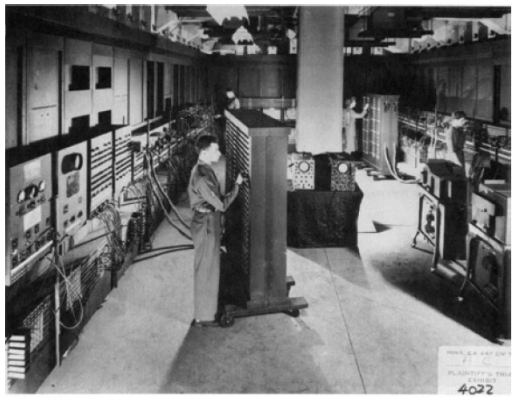
Universal Turing Machine

- Turing described a Turing machine that could implement all other Turing machines.
  - inputs: data, plus a description of computation (Turing machine)
- *U* is programmable – so is a computer!
  - instructions are part of the input data
  - a computer can emulate a Universal Turing Machine, and vice versa
- **Therefore**, a computer is a universal computing device

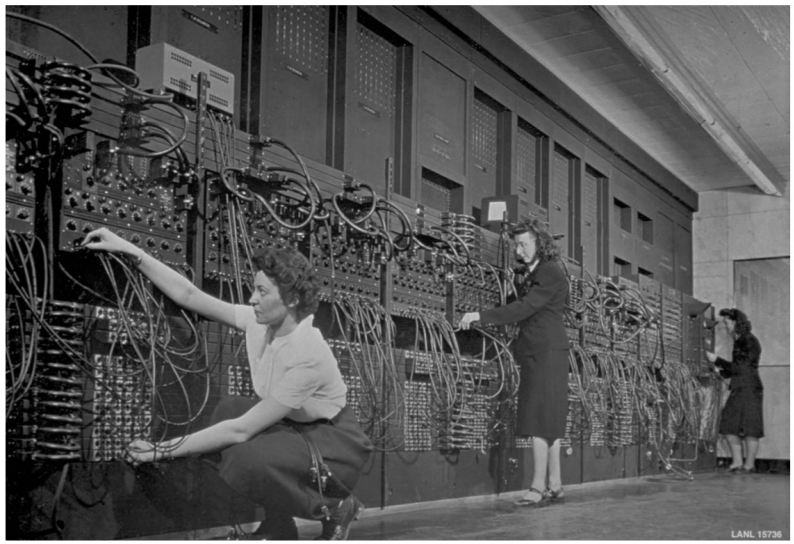


ENIAC - The first electronic computer ,1946

- ENIAC (Electrical Numerical Integrator And Calculator)
  - 17,468 vacuum tubes
  - Power 140kW
  - Weighed 30 tons
  - Occupied 1800 sq ft
  - 80 feet long
  - 8.5 feet high
  - Clock: 100kHz, About 5000 additions per second
  - RAM: ~230bytes, Could store 20 numbers Could store 20 numbers in main memory
  - IO: punched card
  - Cost about $500,000



1904， The world's first electron tube was born at the hands of the British physicist Fleming



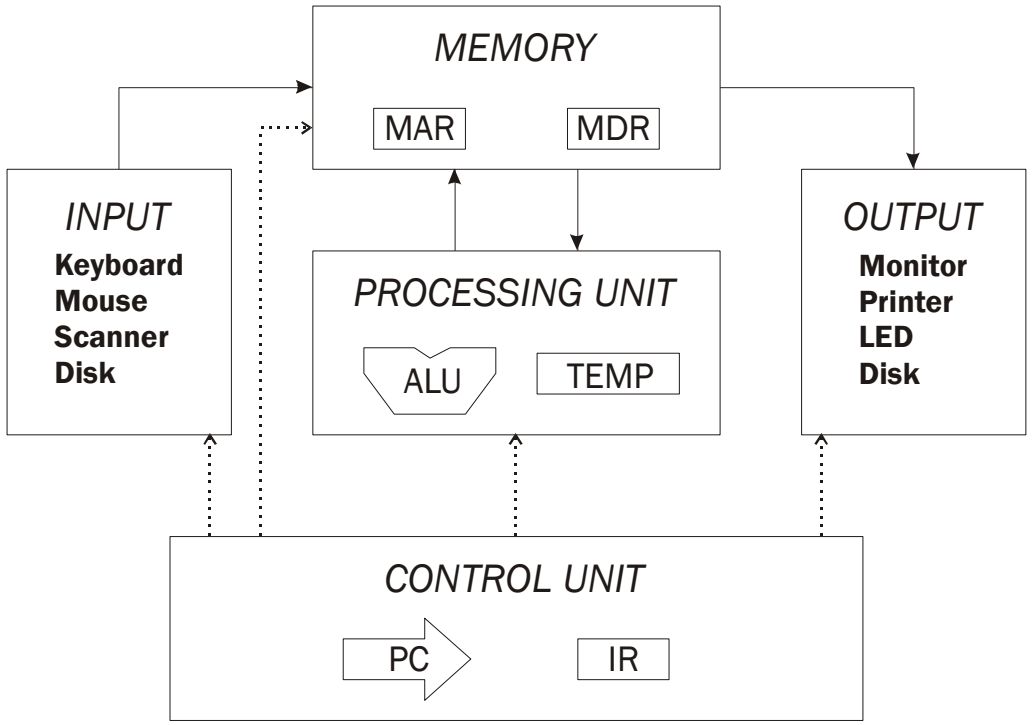
Changing the program could take days

ENIAC (1946)

- **First electronic general-purpose computer**
  - Construction started in secret at UPenn Moore School of Electrical Engineering during WWII to calculate firing tables for US Army, designed by Eckert and Mauchly
  - Twelve 10-decimal-digit accumulators
  - Had a conditional branch!

- Programmed by plugboard and switches, time consuming
- Purely electronic instruction fetch and execution, so fast
  - 10-digit x 10-digit multiply in 2.8ms (2000x faster than Mark-1)
- As a result of speed, it was almost entirely I/O bound
- As a result of large number of tubes, it was often broken (5 days was longest time between failures)

Von Neumann Model (stored program computer)



Two major inventions of the microprocessor chip

Stored program

+

Transistor technology

Change the program so that you can do all kinds of tasks on the same hardware

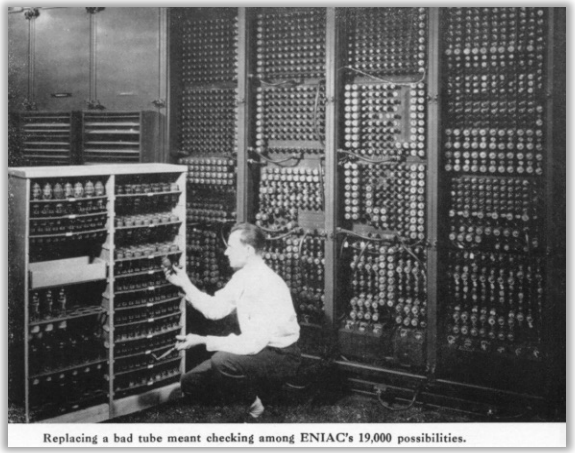
The device is smaller and faster than a vacuum tube

First computer vs. First microprocessor chip

1946, ENIAC (Electrical Numerical Integrator And Calculator)

- 18000 vacuum tubes
- 1500 relays
- 174 KW
- 30 tons
- 1800 sq. ft. footprint
- Clock: 100 kHz
- RAM: ~230bytes
- IO: punched card

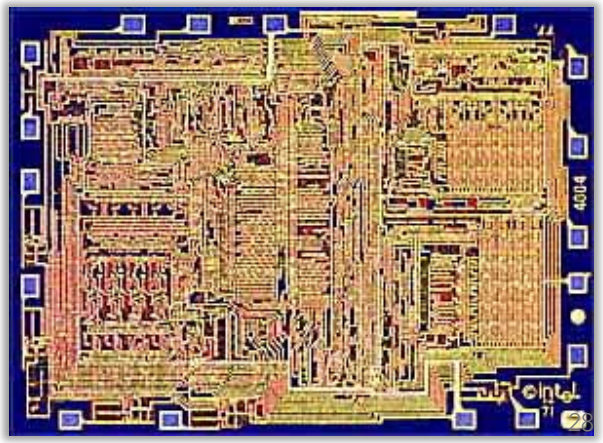
25 Years Later



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

1971, Intel 4004

- 10-micron process, NMOS-Only Logic
- 2,300 transistors
- 3x4 mm die
- 4-bit bus
- Performance < 0.1 MIPS
- 640 bytes of addressable Memory
- 750 kHz



1971, Intel 4004

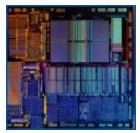
- 10-micron process, NMOS-Only Logic
- 2,300 transistors
- 3x4 mm die
- 4-bit bus
- Performance < 0.1 MIPS
- 640 bytes of addressable Memory
- 750 kHz

30 Years Later

2000, Intel Pentium IV

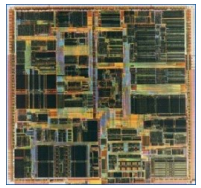
- Issues up to 5 uOPs per cycle
- 0.18-micron process
- 42 million transistors
- 217 mm die **
- 64-bit bus
- 8KB D-cache, 12KB op trace cache (I-cache), 256KB L2 cache
- 1.4 GHz

386
（275K）

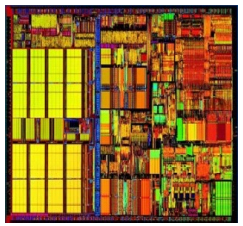


486
（1180K）

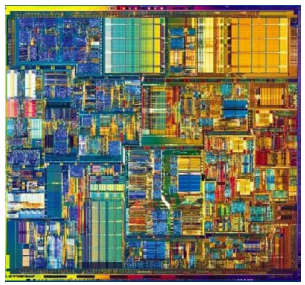
Pentium
（3100K）



Pentium II
（7500K）



Pentium III
（24000K）



Pentium IV
（42000K）

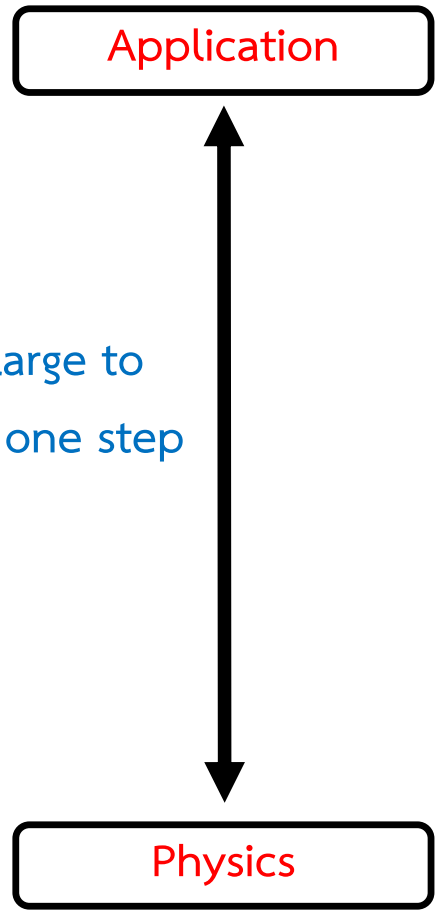
From Theory to Practice

- In theory, computer can compute anything
- that's possible to compute
  - given enough memory and time
- **In practice**, solving problems involves computing under constraints.
  - time
    - weather forecast, next frame of animation, ...
  - cost
    - cell phone, automotive engine controller, ...
  - power
    - cell phone, handheld video game, ...

How Do We Get the Electrons to Do the Work?

- Statement of the Problem
- Algorithm
- Program
- ISA
- Microarchitecture
- Logic Circuit
- Devices

Gap too large to bridge in one step



Problems

Algorithms

Language

Machine (ISA) Architecture

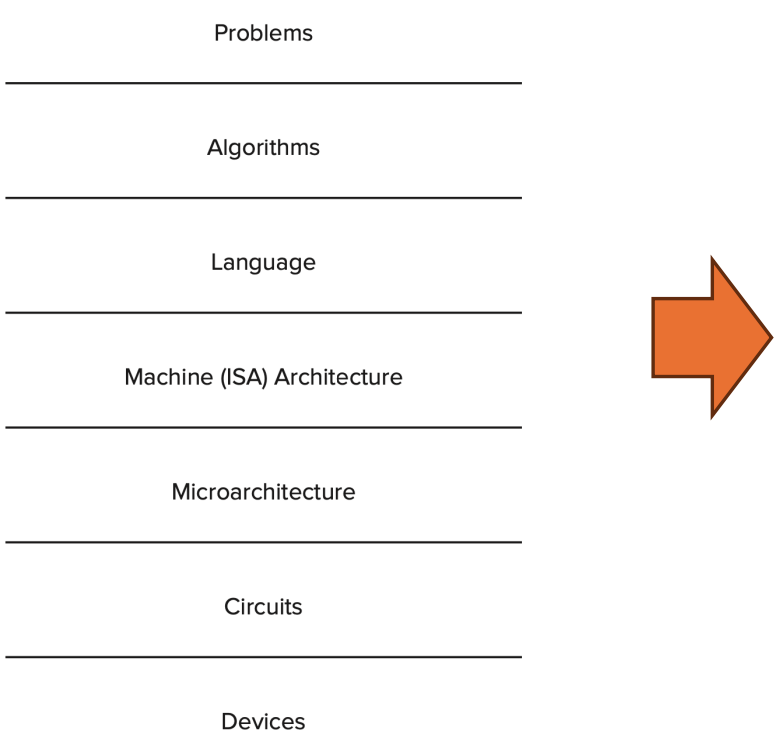
Microarchitecture

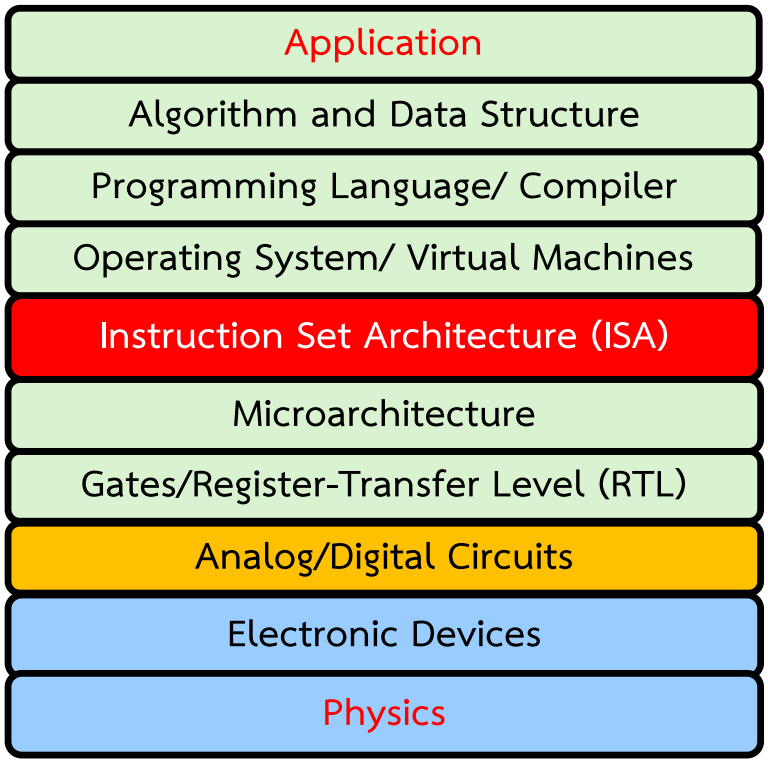
Circuits

Devices

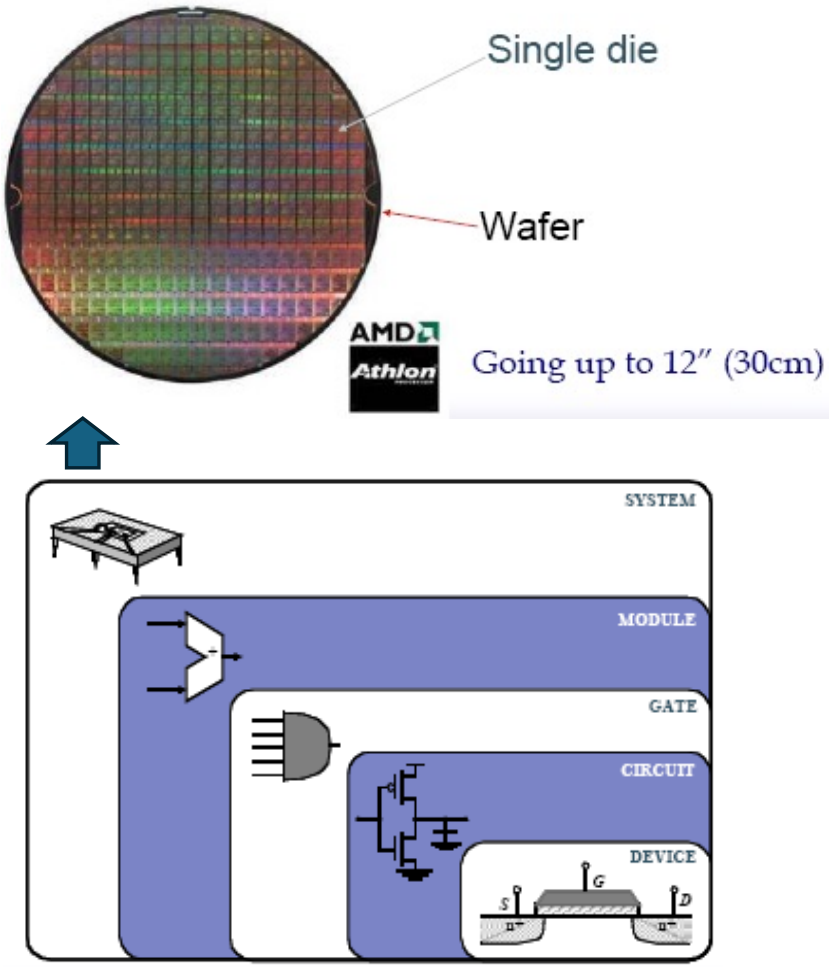
How Do We Get the Electrons to Do the Work?

- Abstraction helps us Manage Complexity

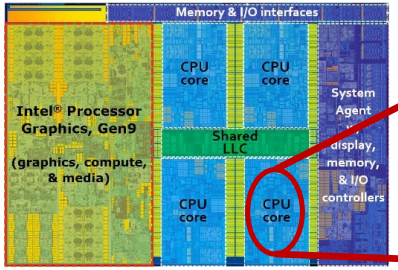


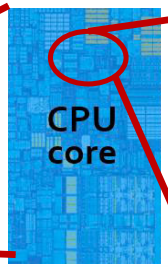


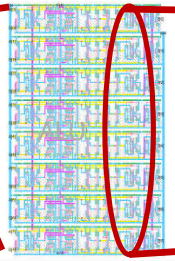
Abstraction to Simplify Hardware Design

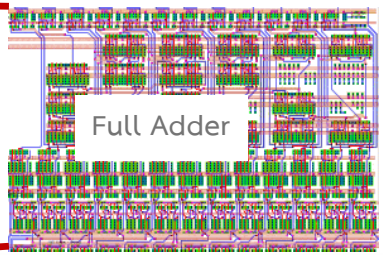


Abstraction to Simplify Hardware Design









Integrated Circuit Design

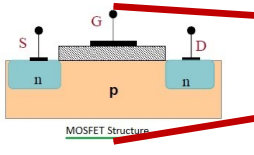
100 Modules/ IC

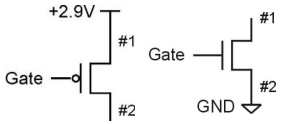
0.25M~20G Devices

Register Transfer Level (RTL) Design

1K~10K Cells/Module

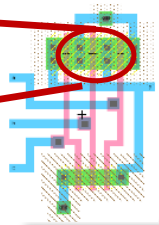
(100K Devices)

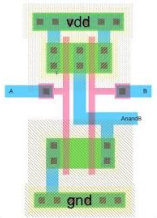


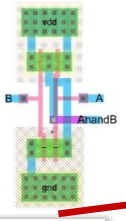


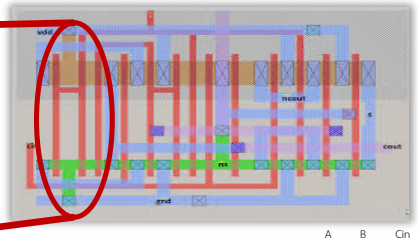
Transistor Physical Layout

Scheme for Representing Information

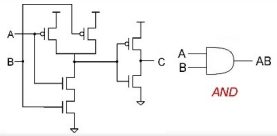








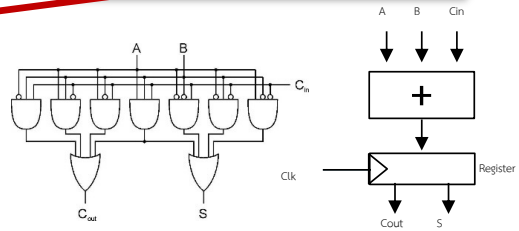
Gate Level Design



Circuit Level Design

(Transistor Level Design)

(2~8 Devices/Gate)

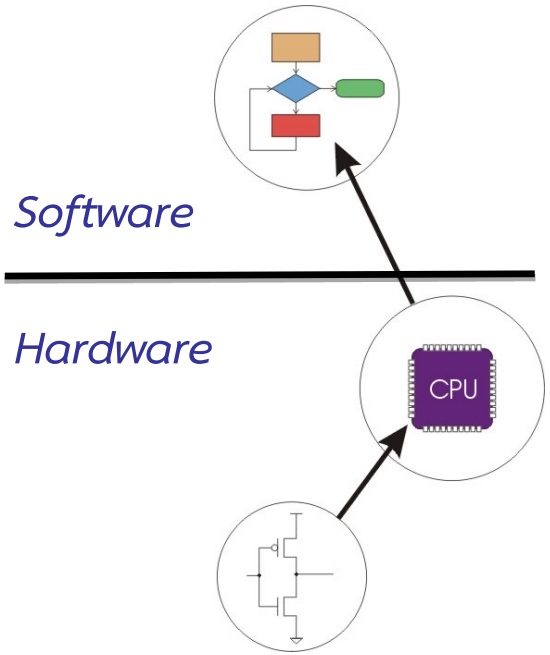


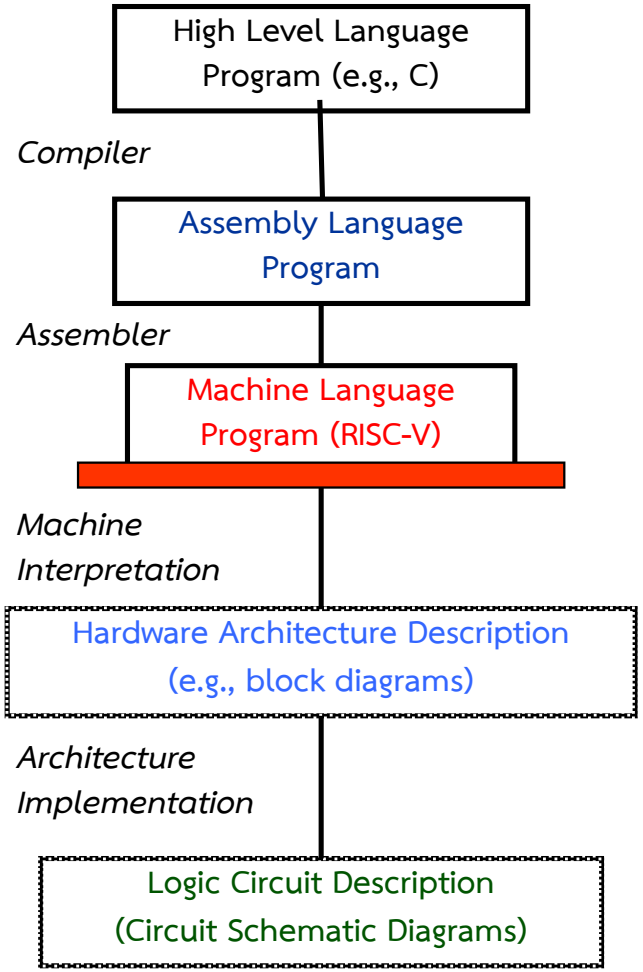
Register Transfer Level (RTL) Design

2~16 Gates/Cell

(16~64 Devices)

How do we get the electrons to do the work?

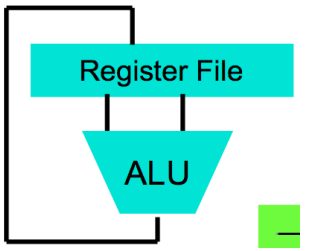


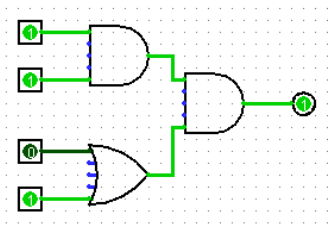



```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

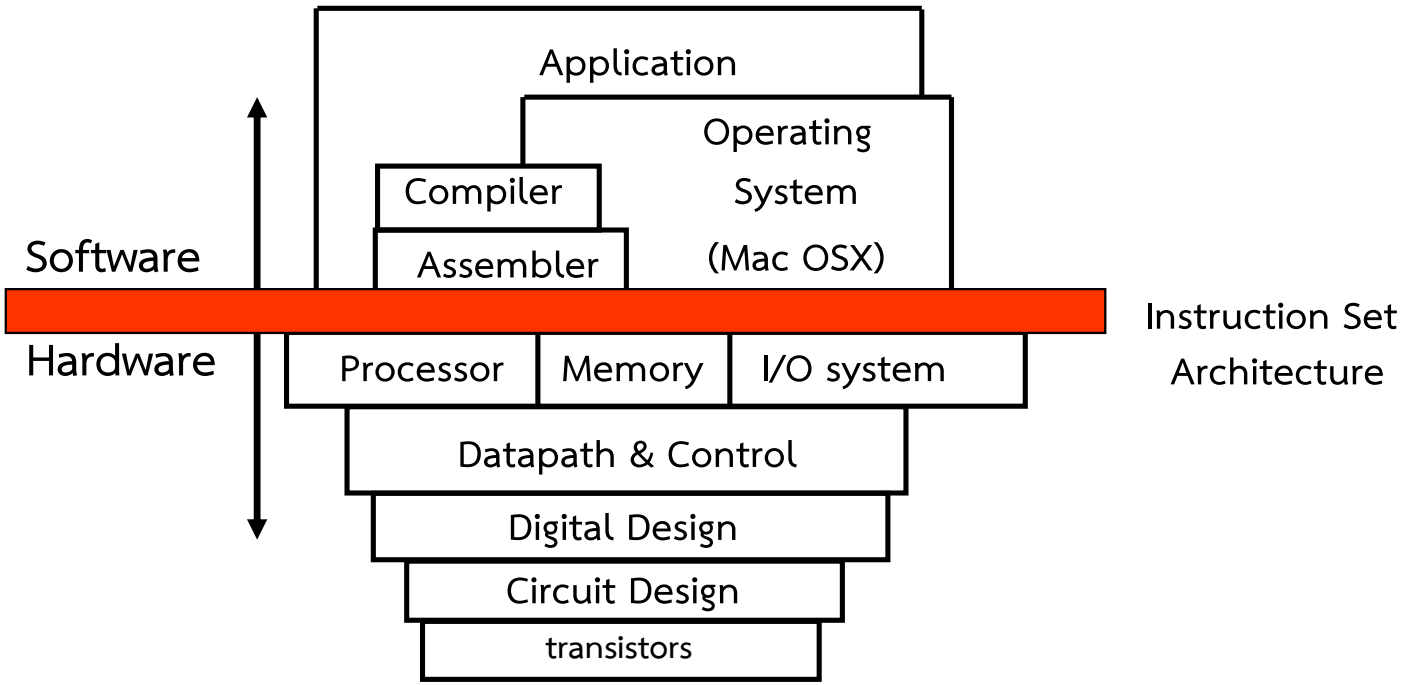
```
lw  $t0, 0(a0)
lw  $t1, 4(a0)
sw  $t1, 0(a0)
sw  $t0, 4(a0)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

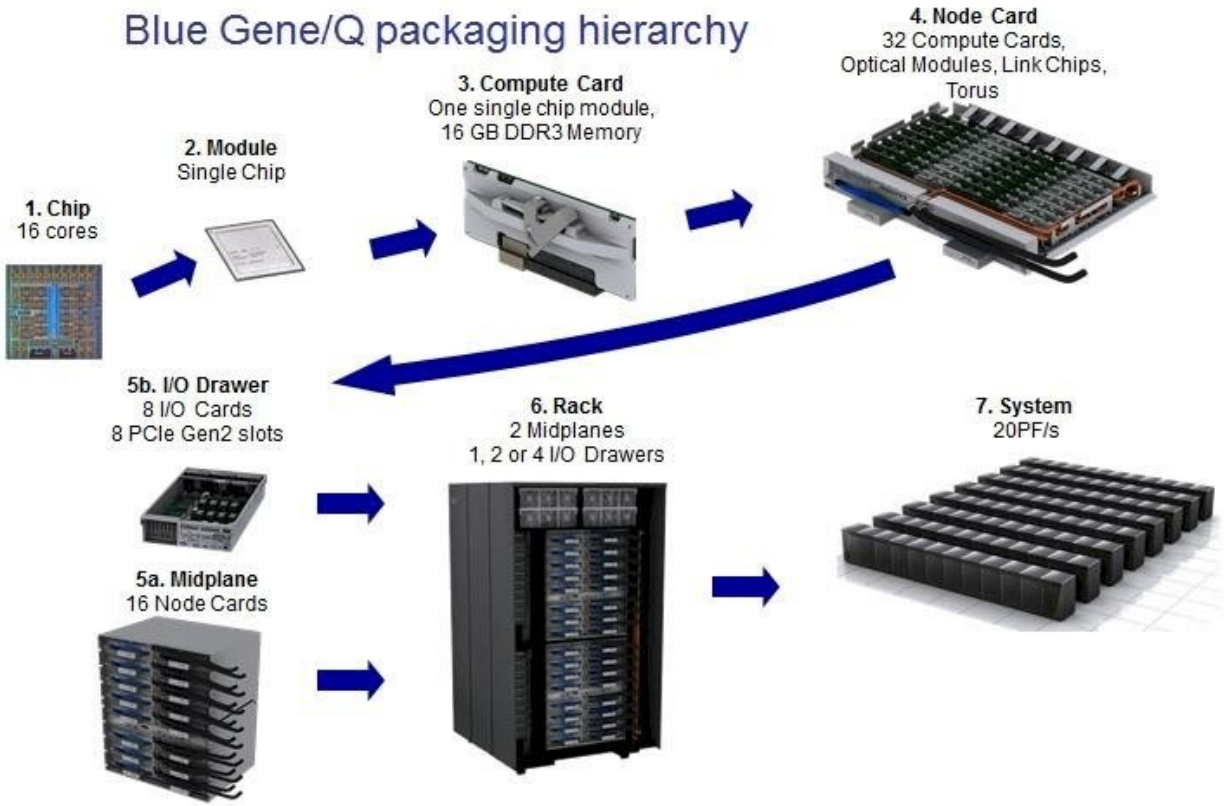




Old Machine Structures

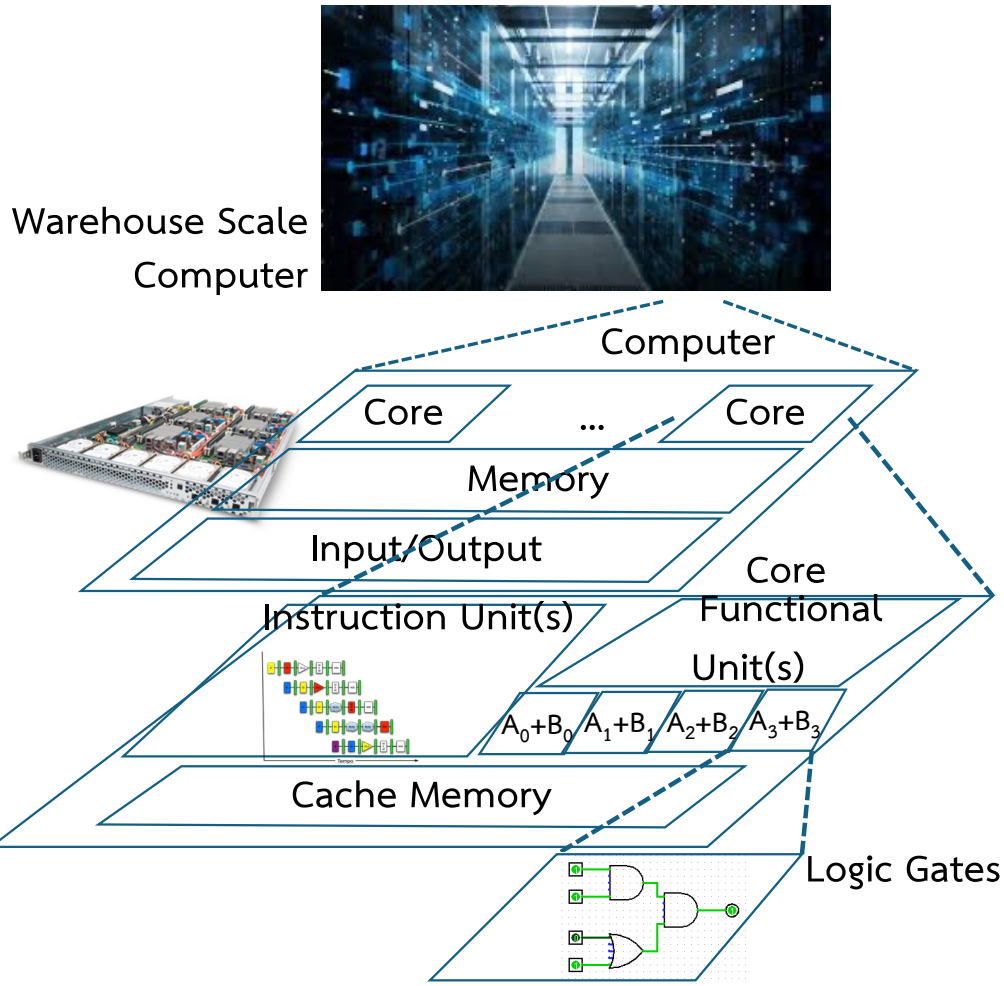


New Machine Structures



Logic gate to the Cloud

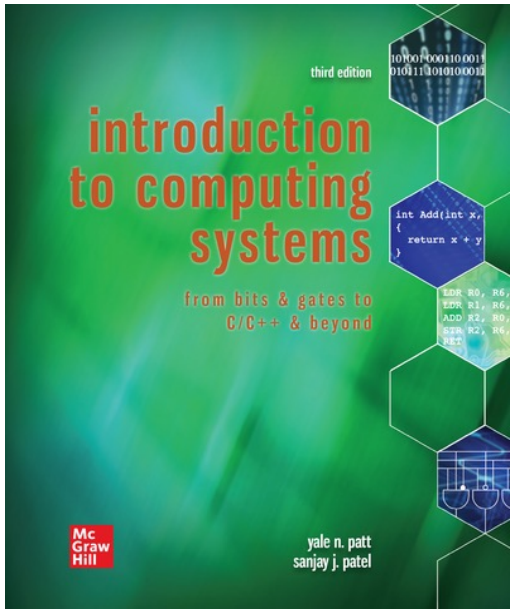
- Parallel Threads
  - Assigned to core e.g., Lookup, Ads
- Parallel Instructions
  - >1 instruction @ one time e.g., 5 pipelined instructions
- Parallel Data
  - >1 data item @ one time e.g., Add of 4 pairs of words
- Hardware Descriptions
  - All gates functioning in parallel at same time
- Programming Languages

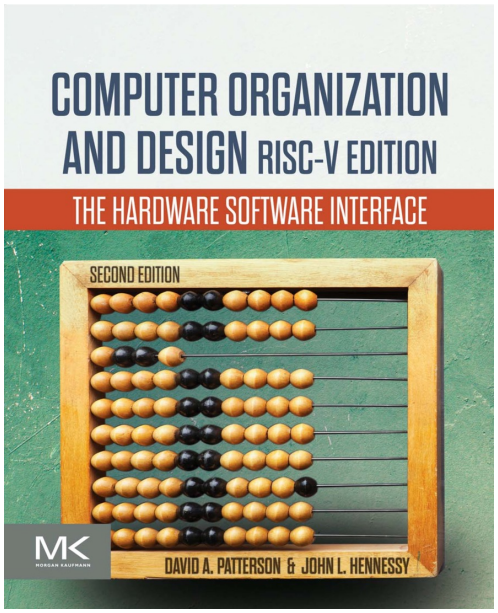


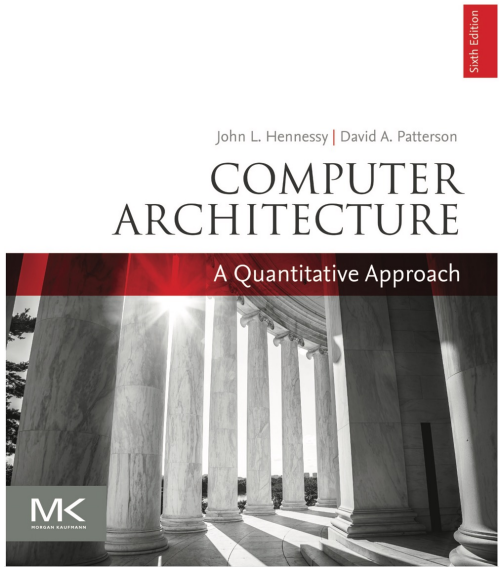
Homework

- 1.2
- 1.3
- 1.5
- 1.13
- 1.20
- 1.22
- 1.25

References







- Introduction to Computing Systems / Computer Architecture Courses
  - Coursera / Edx
  - Princeton University/ University of Science and Technology of China