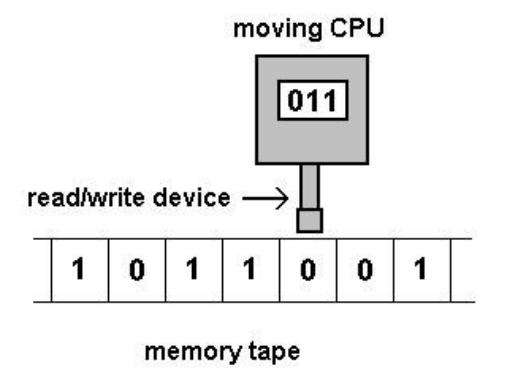
COMP1003 Computer Organization

Lecture 1 What is a Computer



United International College

Computation & Algorithm

- Computation is any type of arithmetic or non-arithmetic calculation that follows a well-defined model (e.g., an algorithm)
- Example of Algorithm: Euclid's Algorithm for computing the greatest common divisor (GCD) of two integers

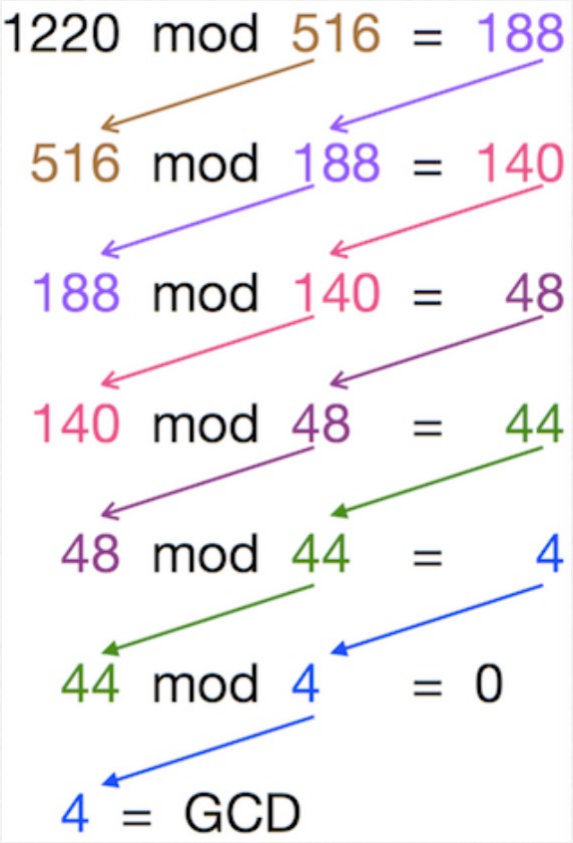
```
function gcd(a, b)
    while b ≠ 0
        t := b
        b := a mod b
        a := t
    return a
```

Recursive version

```
function gcd(a, b)
    if b = 0
        return a
    else
        return gcd(b, a mod b)
```

GCD(1220, 516) = ?

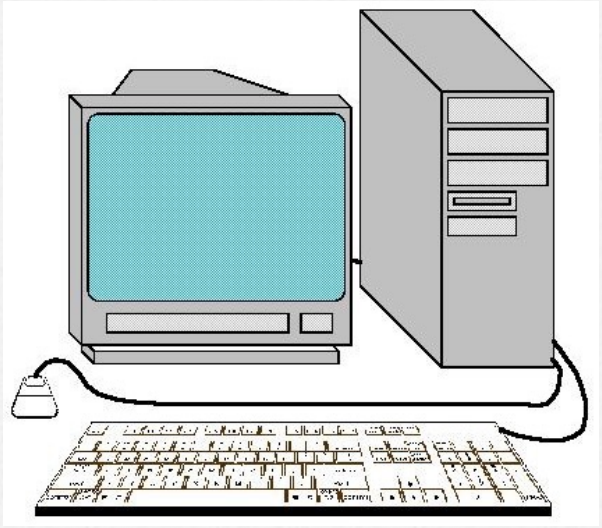
```
function gcd(a, b)
    while b ≠ 0
        t := b
        b := a mod b
        a := t
    return a
```



Human Computers

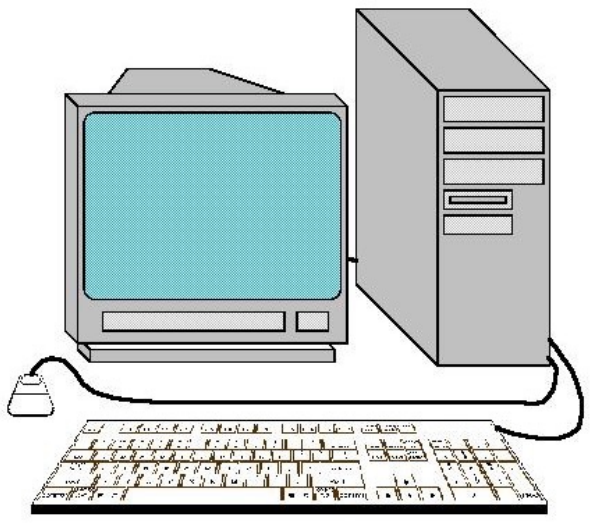
- Before electronic computers became commercially available, a computer meant one who computes: a person performing mathematical calculations.

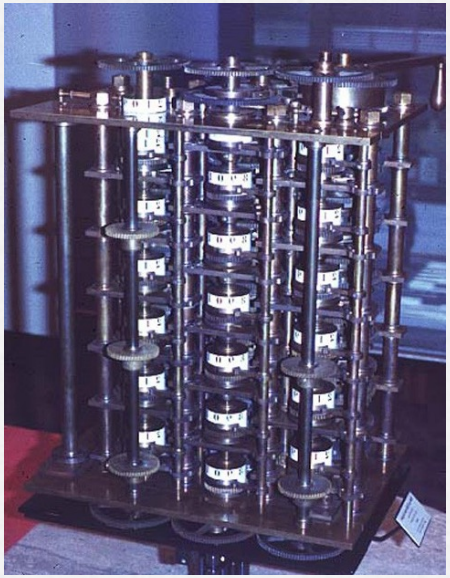
Modern computer vs Abacus



- how is a modern computer different from an abacus?

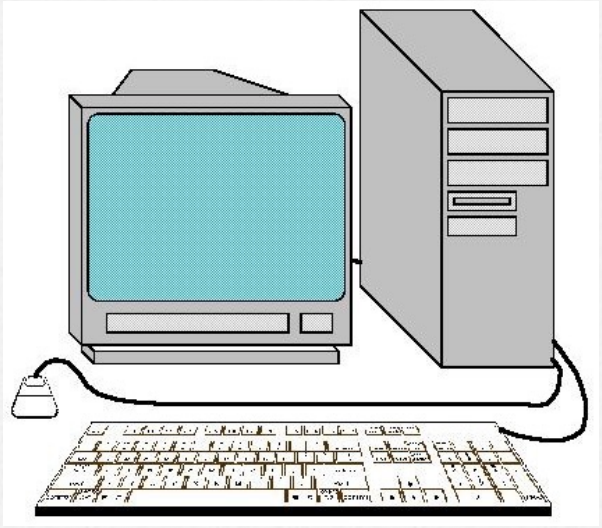
Modern Computer
vs
Mechanical Computer





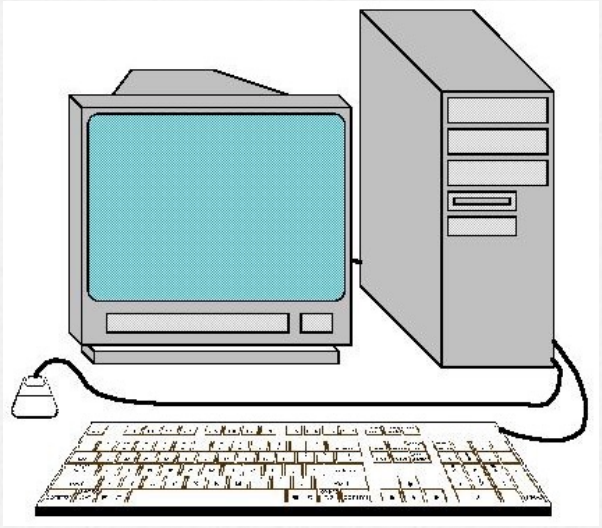
- how is a modern computer different from a mechanical computer?

Modern Computer vs Calculator



- how is a modern computer different from a calculator?

Modern computer vs Human Computer



- how is a modern computer different from a human computer?

Definition: Computer

- A modern computer is an electronic, digital, general purpose computing machine that automatically follows a step-by-step list of instructions for solving a problem.

- This step-by-step list of instructions that a computer follows is also called a computer program.

What does a human need to compute by hand?

- Think & answer
- For example:
  - solving the problem GCD(a,b) = ?

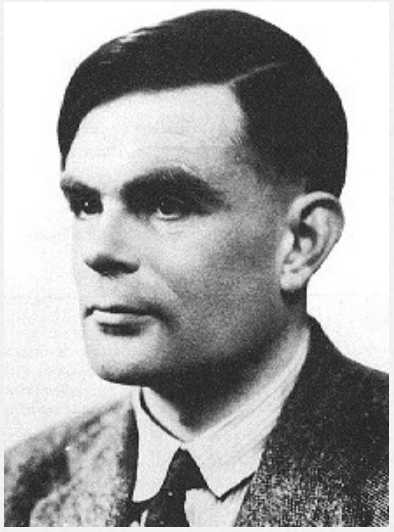
Hardware: Computer vs Human

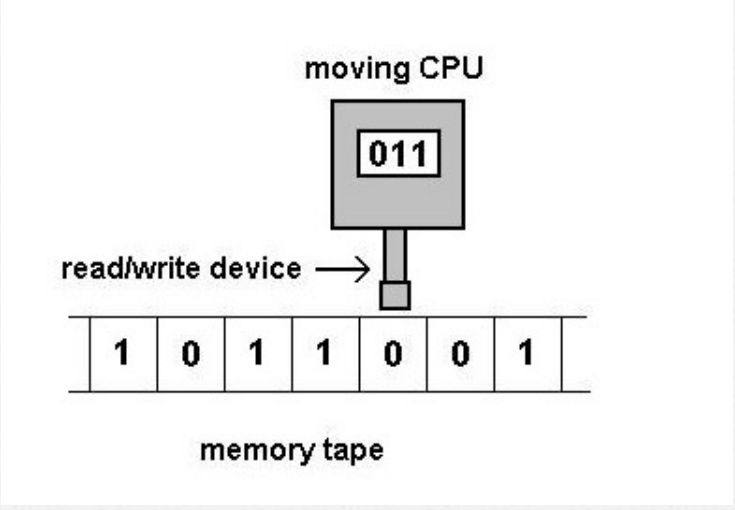
- Human brain: - CPU (Central Processing Unit)
  computer memory
- Paper: storage (hard disk)
- Eyes: input devices
- Hand & Pen: output devices

Computer Parts

An Abstract Computer: the Turing Machine

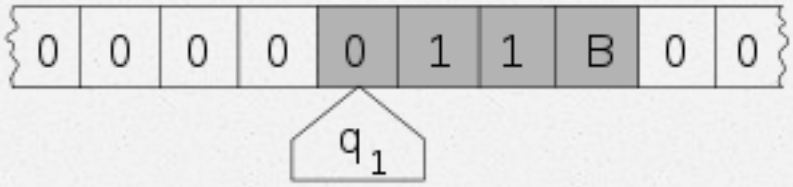
- In 1936, British mathematician Alan Turing developed a hypothetical device, the Turing machine, which is the abstract model of all computers



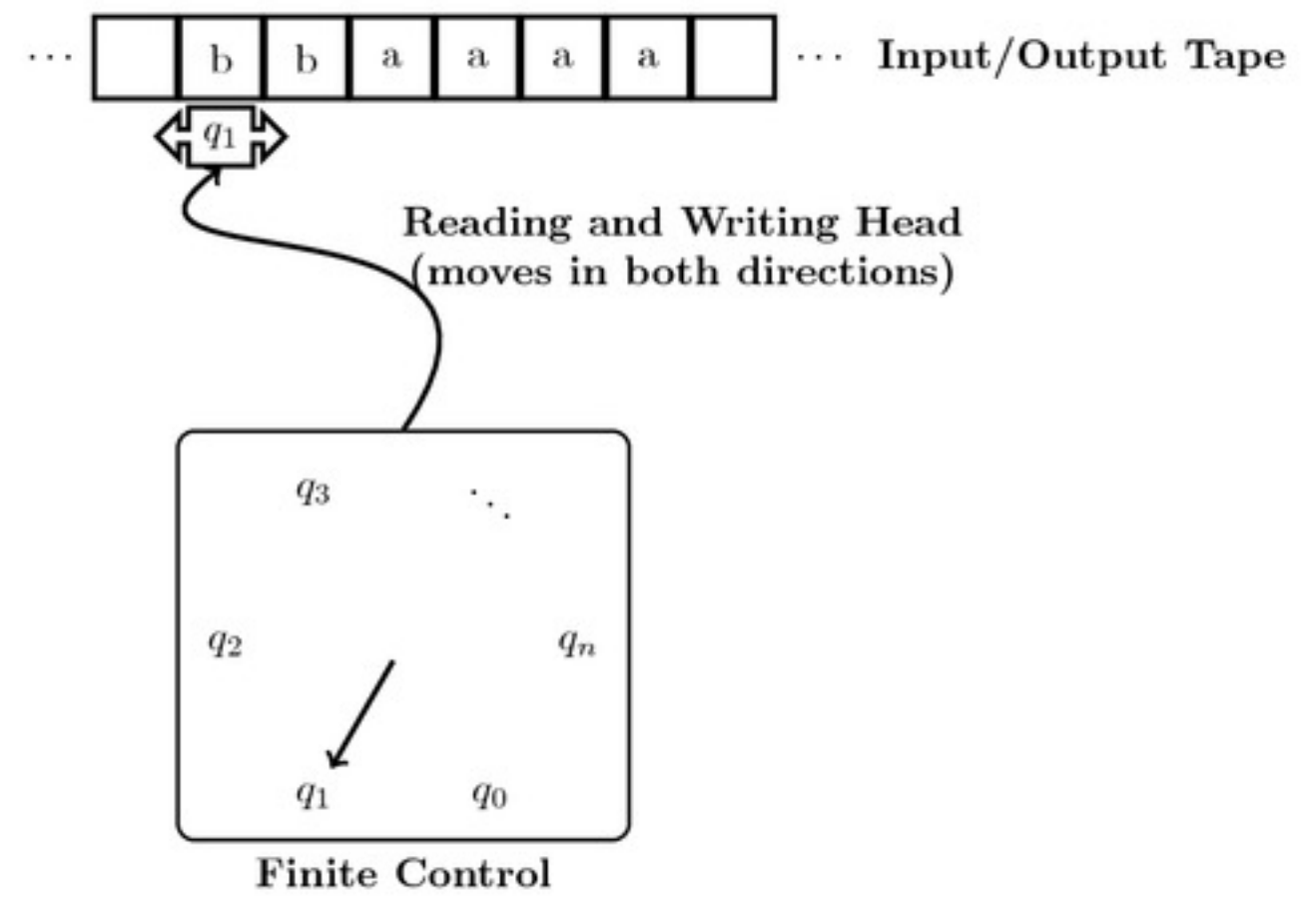


Turing Machine

- A Turing machine consists of
  - a tape divided into cells
  - a moving read/write head
  - a state register storing the state of the Turing machine
  - a finite table of instruction specifying what the machine does when reading the content of the current cell: move right/left; erase/write a symbol; change the state



Turing Machine



Example of TM Computation

Turing Machine to Add Two Integers

$2 + 2 = 4$

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Input Tape = B0110110B

Transition Functions

$(q_0, 0) = \{q_1, 0, R\}$

$(q_1, 1) = \{q_1, 1, R\}$

$(q_1, 0) = \{q_2, 0, R\}$

$(q_2, 1) = \{q_3, 0, L\}$

$(q_2, 0) = \{q_5, B, R\} = \{F\}$

$(q_3, 0) = \{q_4, 1, R\}$

$(q_4, 1) = \{q_1, 0, L\}$

$(q_4, 0) = \{q_2 0, R\}$

Computation Trace

$q_0\,0110110 \vdash 0q_1 110110 \vdash 01q_1 10110 \vdash 011q_1 0110$

$\vdash 0110q_2 110 \vdash 011q_3 0010 \vdash 0111q_4 010 \vdash 01110q_2 10$

$\vdash 0111q_3 000 \vdash 01111q_4 00 \vdash 011110q_2 0 \vdash 011110Bq_5 \vdash \{F\}$

An Implementation of the Turing Machine

http://www.aturingmachine.com

The Church-Turing Thesis

- All things that can be computed can be computed by a Turing machine

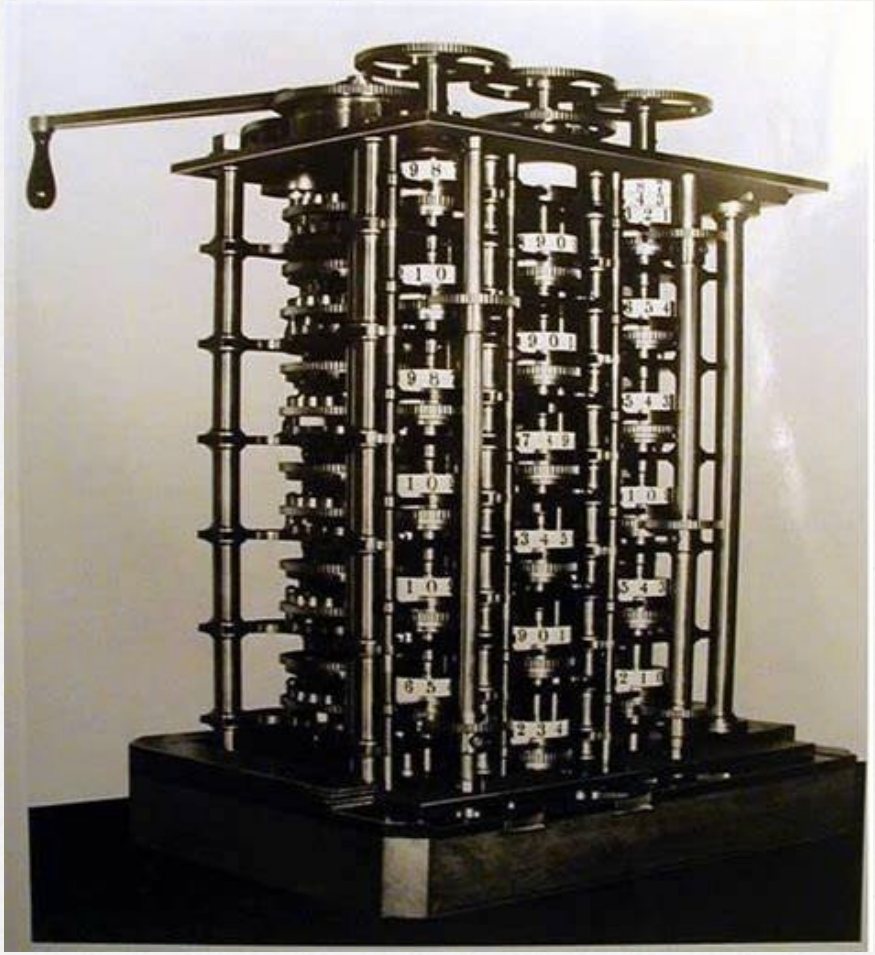
Universal Turing Machine

- Turing described a Turing machine that could simulate all other Turing machines.
  - inputs: data + a description of computation (Turing machine)
  - It is programmable – so it is a computer!– instructions are part of the input data–
  - A computer is a Universal Turing Machine!

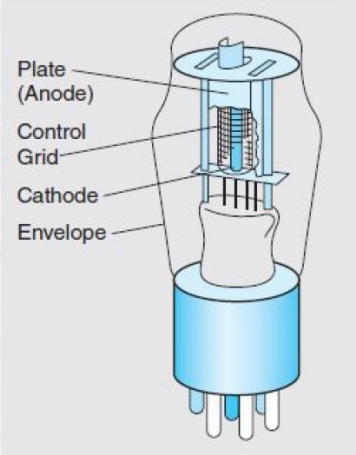
Historical Development

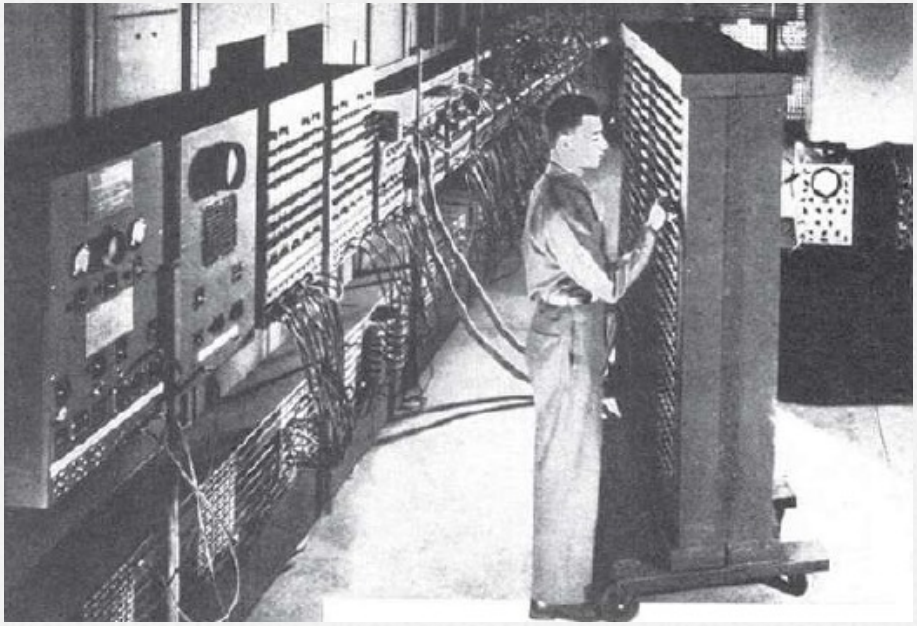
- Generation Zero: Mechanical Calculating Machines (1642-1945)
- The First Generation: Vacuum Tube Computers (1945-1953)
- The Second Generation: Transistor Computers (1954-1965)
- The Third Generation: Integrated Circuit (IC) Computers (1965-1980)
- The Fourth Generation: VLSI Computers (1980-)
- What's next?

Generation Zero

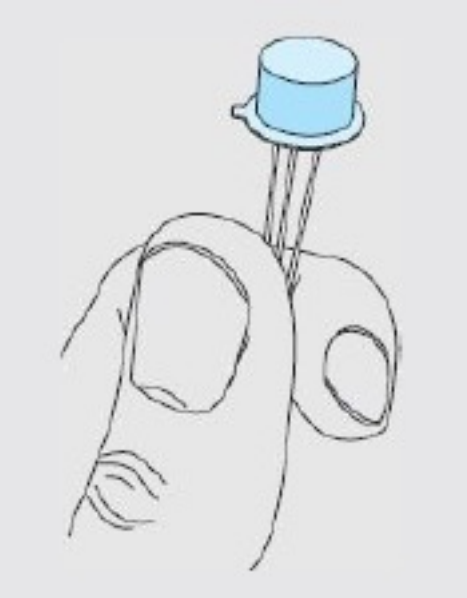


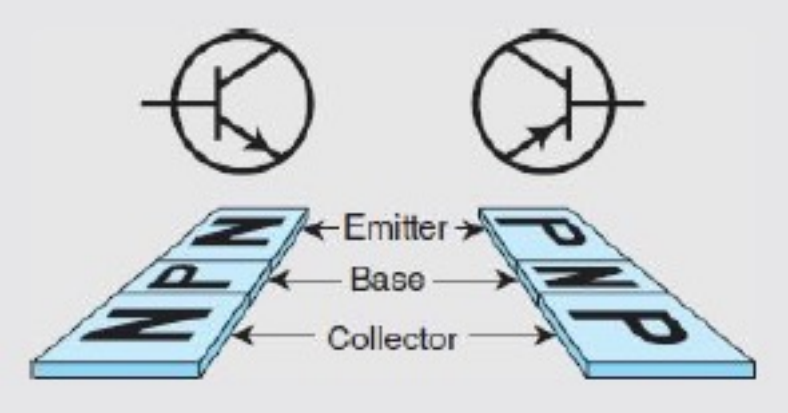
Generation One: Vacuum Tube (Valve)



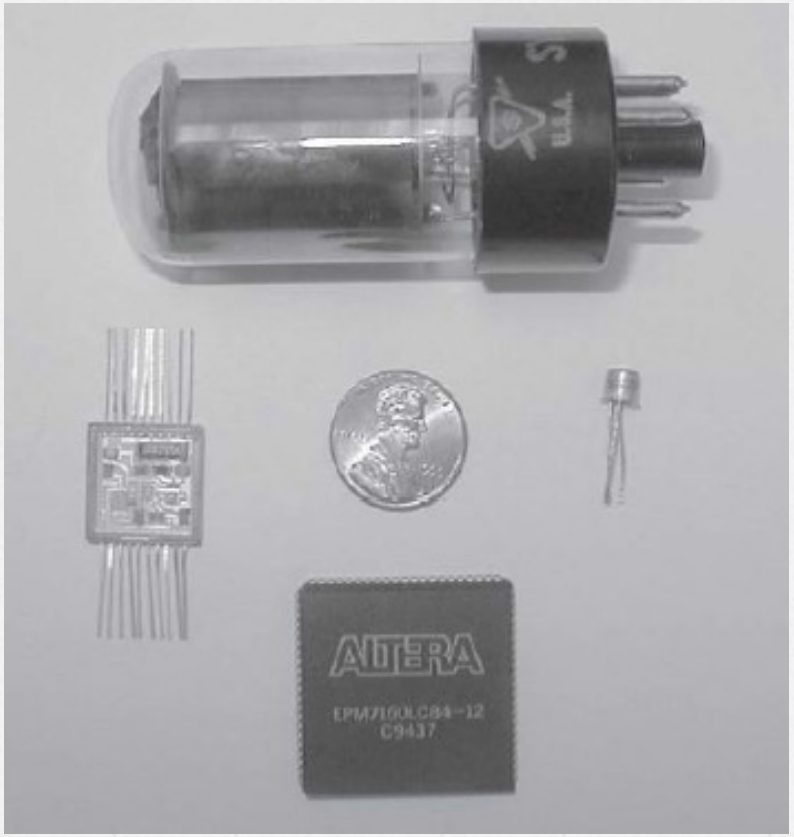


Generation Two: Transistors



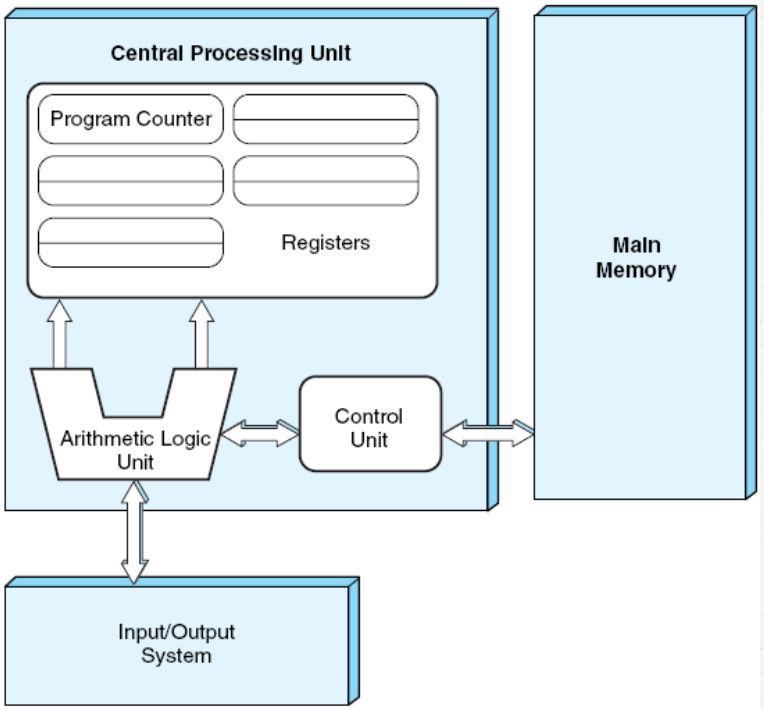


Generation 3 & 4: IC & VLSI



The von Neumann Architecture

- Also called stored-program architecture
- Both data and program are stored in the memory



The stored-program architecture

- A Central Processing Unit (CPU)
  - control unit
  - Arithmetic Logic Unit (ALU)
  - Registers
- Main memory
- I/O system
- a single path between the main memory and CPU, called the von Neumann bottleneck

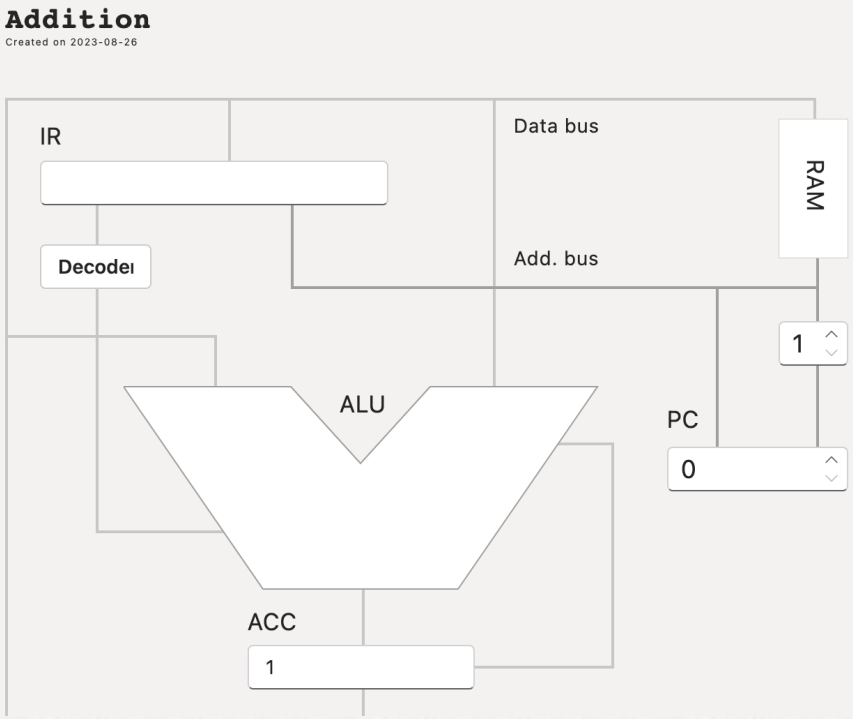
Von Neumann Execution Cycle

- also called the fetch-decode-execute cycle
  - the control unit fetch the next instruction from the memory
  - the instruction is decoded into a language that the ALU understands
  - data operands are fetched from the memory into the registers inside CPU
  - the ALU executes the instruction and places the result into the registers or memory

Example: X + Y = Z

```
0  // X + Y = Z
1  LOD X
2  ADD Y
3  STO Z
4  HLT
```

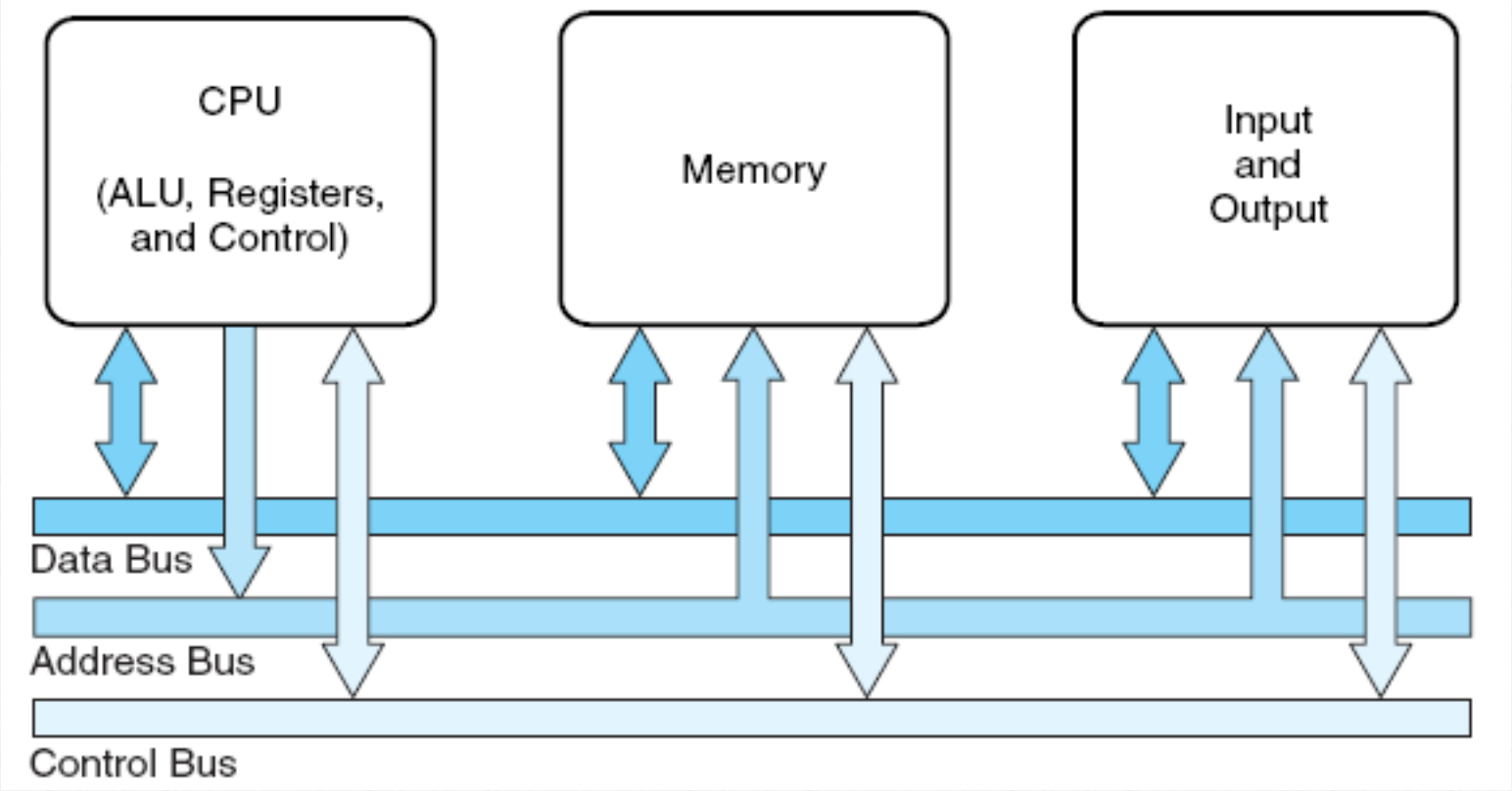
X	2
Y	2
Z	0
W	0
T1	0
T2	
T3	
T4	
T5	
T6	
T7	
T8	
T9	
T10	



The von Neumann Bottleneck

- CPU and memory are separate
- All data and code are in the memory
- CPU is usually faster than memory
- CPU is forced to wait for needed data to be transferred to or from memory

The system bus model of the von Neumann Architecture



Discussion

- What is abstraction?

The Notion of Abstraction

- The process of forming a concept by identifying common features among a group of individuals, or by ignoring spatial-temporal aspects of these individuals
- The essence of abstraction is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context.

From Abstract to Concrete

- (1) a publication
  - (2) a newspaper
    - (3) The San Francisco Chronicle
      - (4) the May 18 edition of the Chronicle
        - (5) my copy of the May 18 edition of the Chronicle

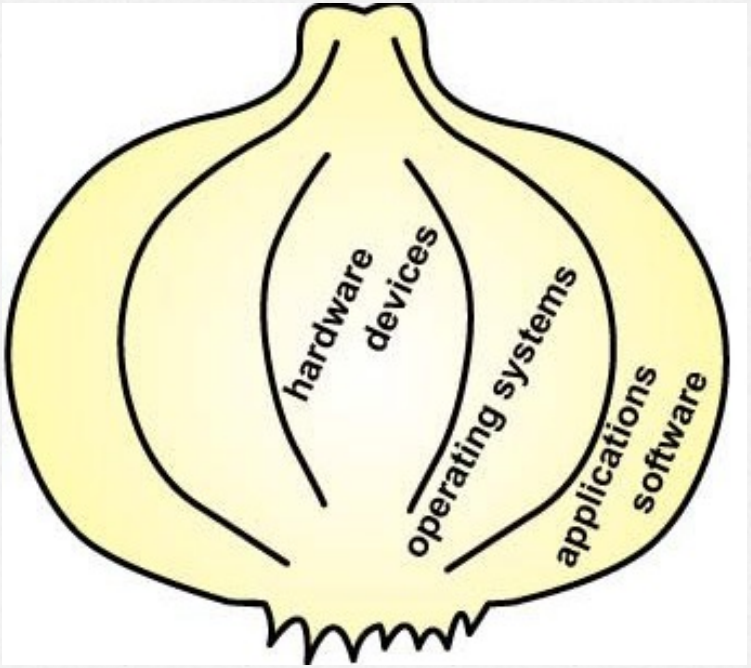
Abstraction

- Abstraction allows us to ignore the details and focus on a few concepts at a higher level

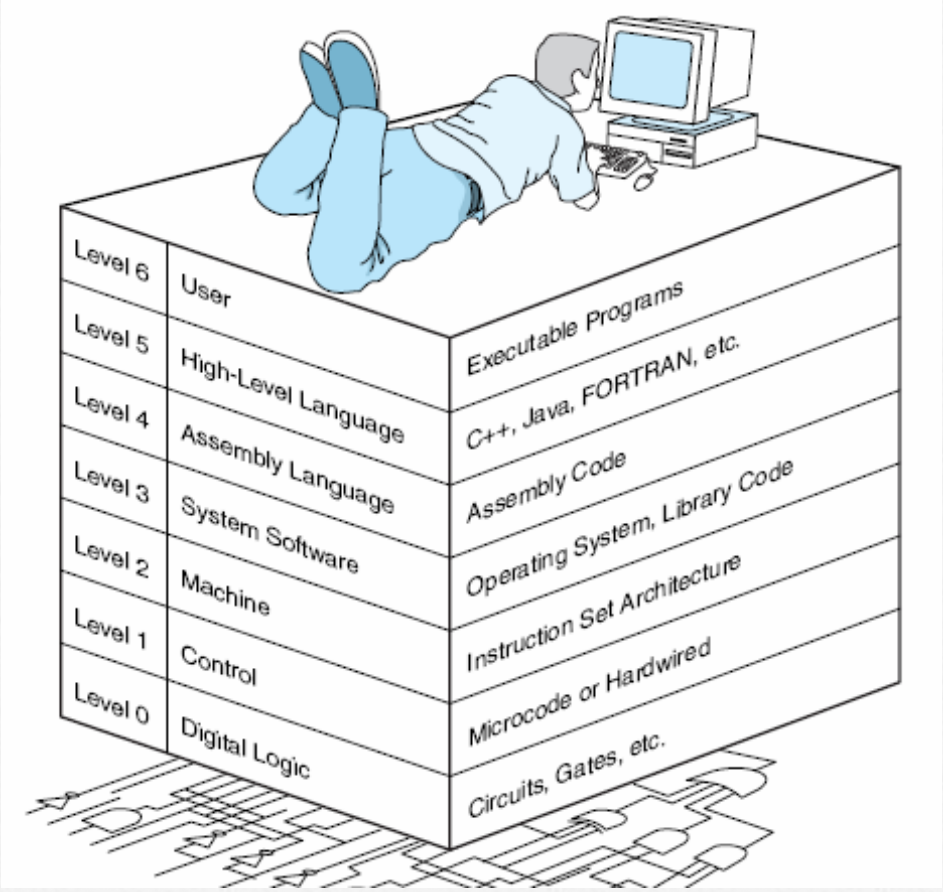
- Un-abstraction is the ability to go from the abstraction back to the underlying details

Abstract Model of a Computer

- Computer as an onion



The Abstract Levels of Modern Computing Systems



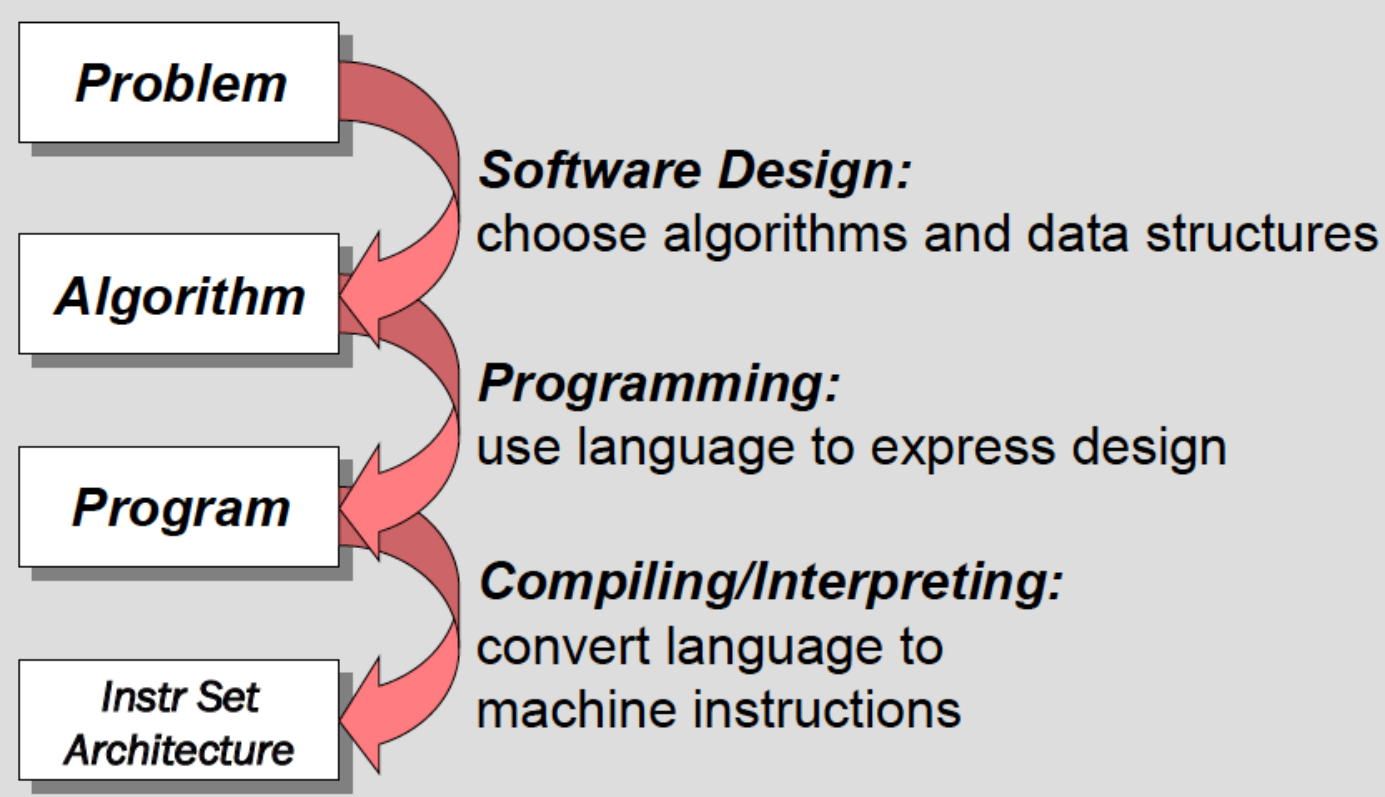
Levels of Abstraction

- User level: applications such as qq.exe
- High level language: C, Java, C++
- Assembly language
- Operating system
- Machine Language: Instruction Set A
- Control level: micro-code or hardwired
- Digital logic: circuits, gates

Gap between the User and the Gates

- How can you get the electrons to solve your problem such as 1+ 2 + …+100 = ?
- There is a gap between what the user wants and what the electrons can do
- Answer: you must go through levels of transformations

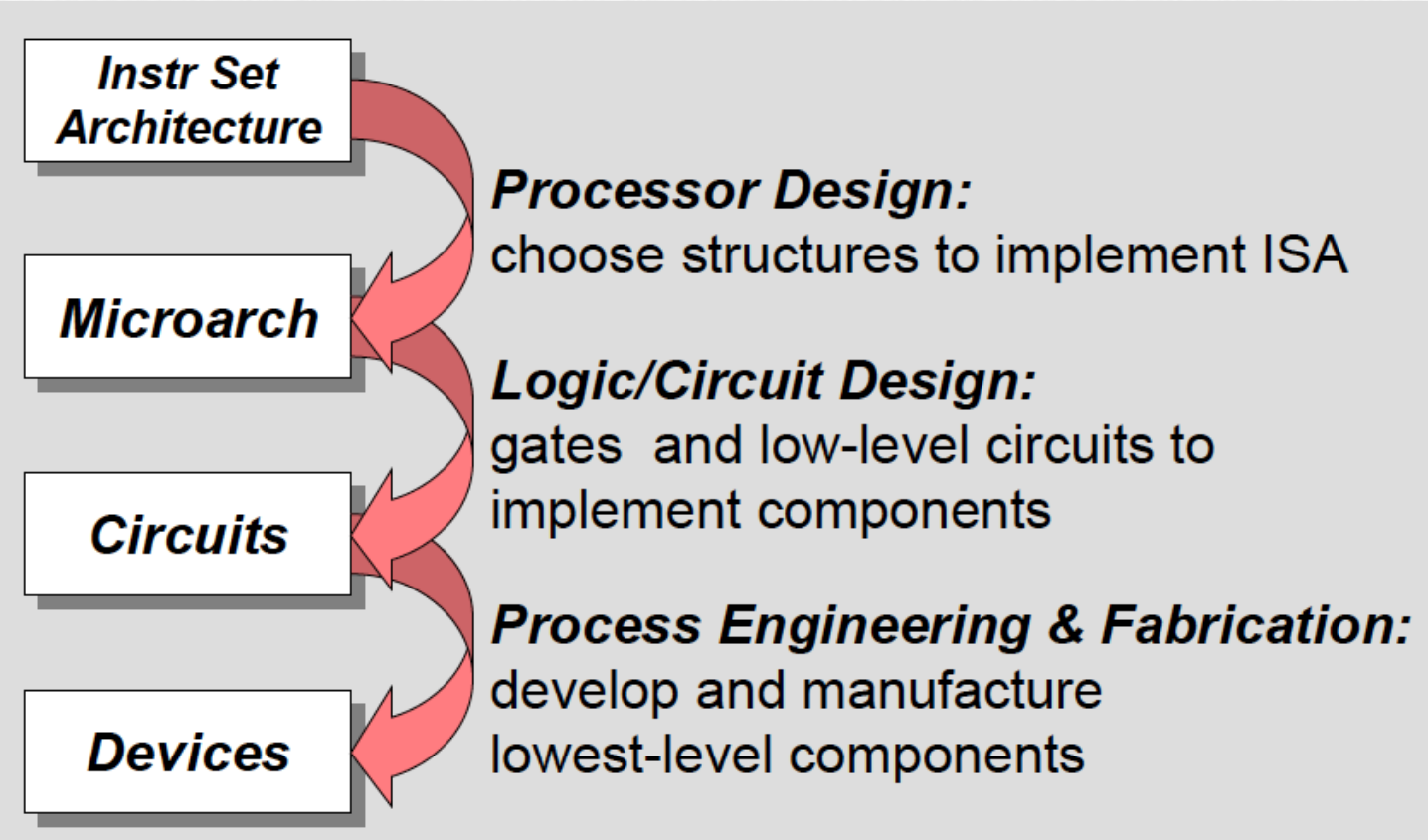
Levels of Transformations



The Program Levels

- Problem: What you want to do
- Algorithm: step-by-step procedure to solve the problem
- Program: a computer program that implements the algorithm using some programming languages

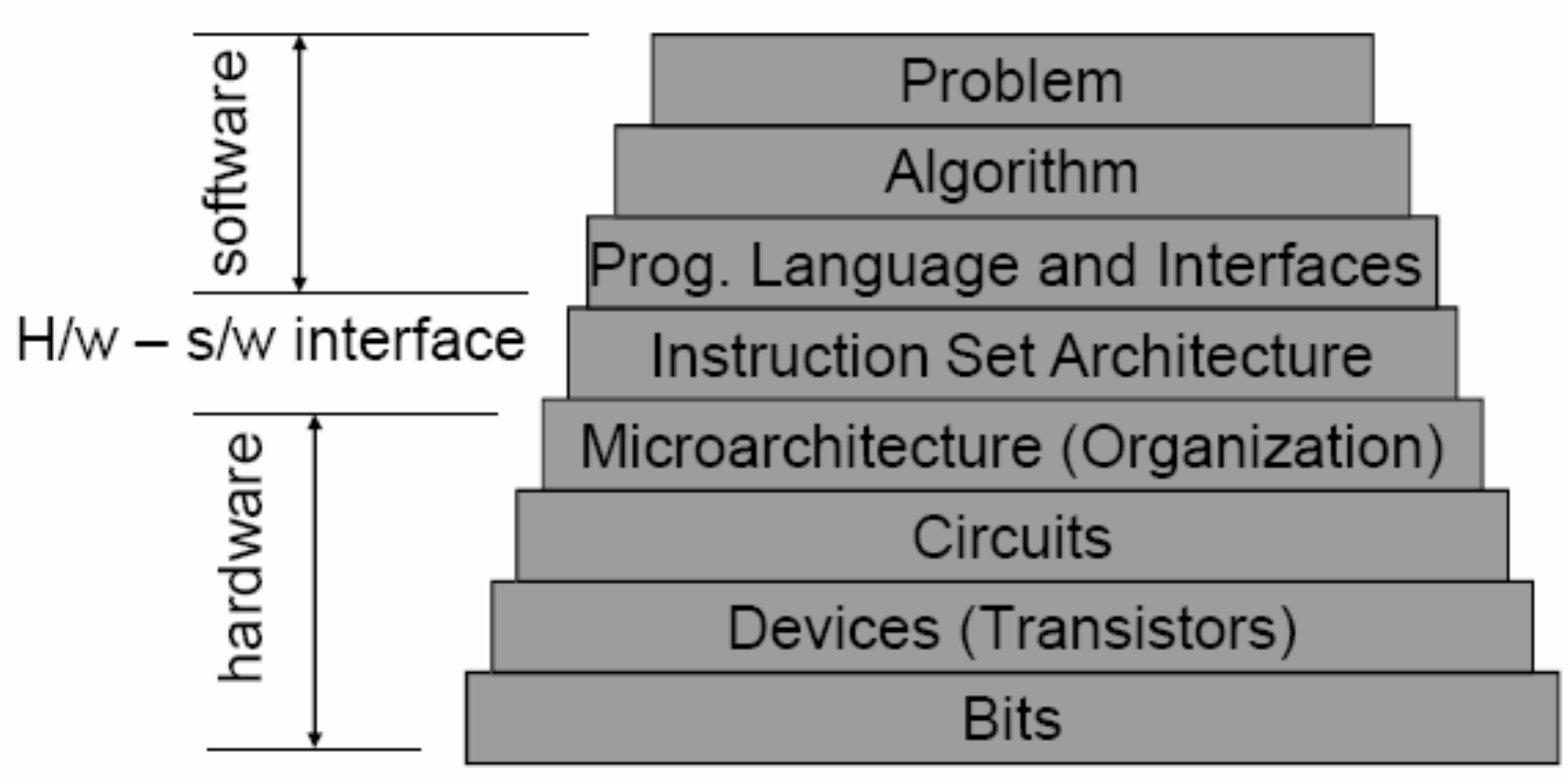
Levels of Transformations



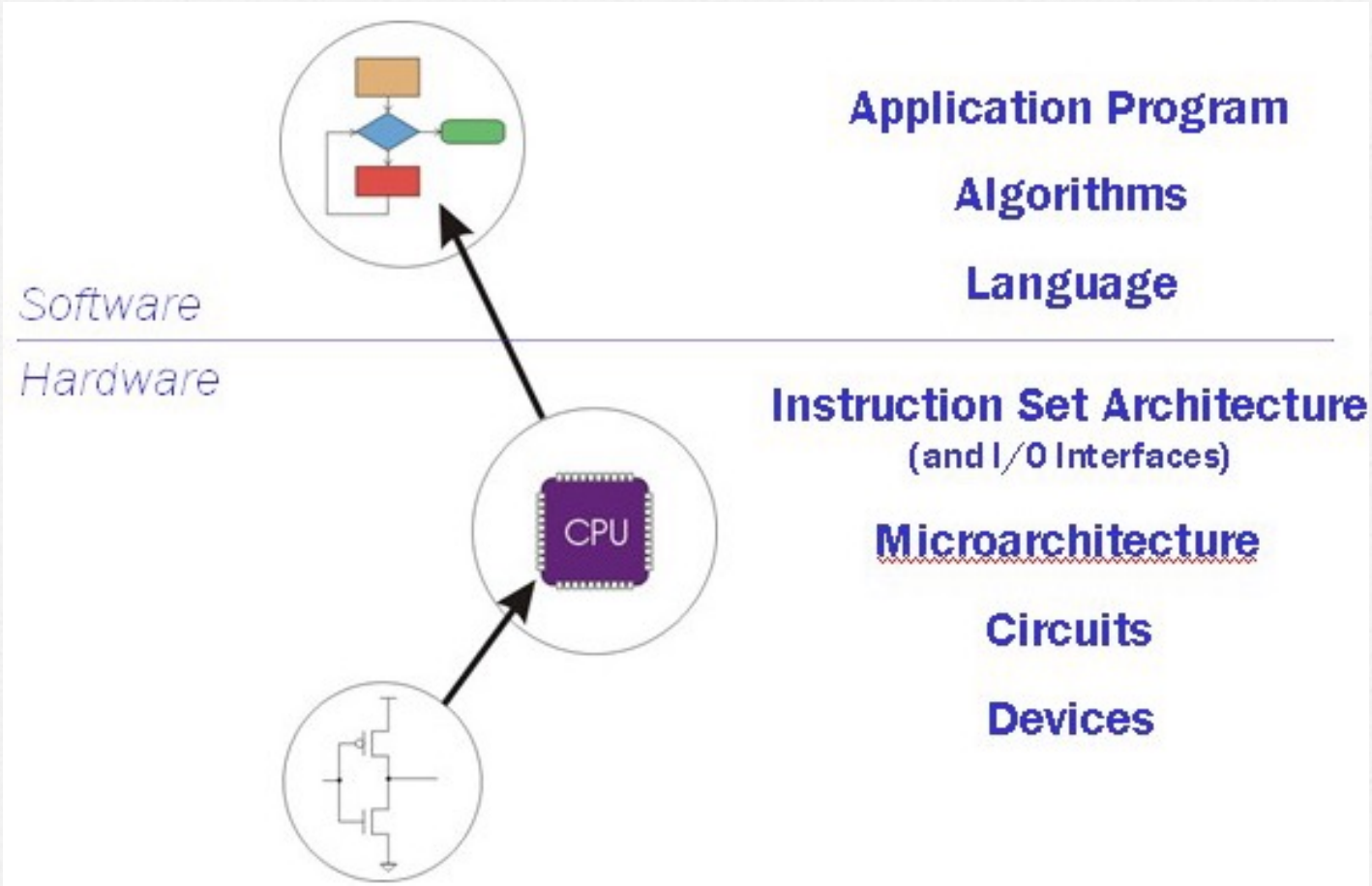
The Machine Levels

- Instruction Set Architecture (ISA): instructions that a CPU can execute
- Microarchitecture: implementation of ISA
- Circuits: Details of electrical circuits
- Devices (transistors): Circuits are built by interconnecting transistors
- Bits: Transistors operate on bits (“0” or “1”) that represent data and information

Layers of Transformations



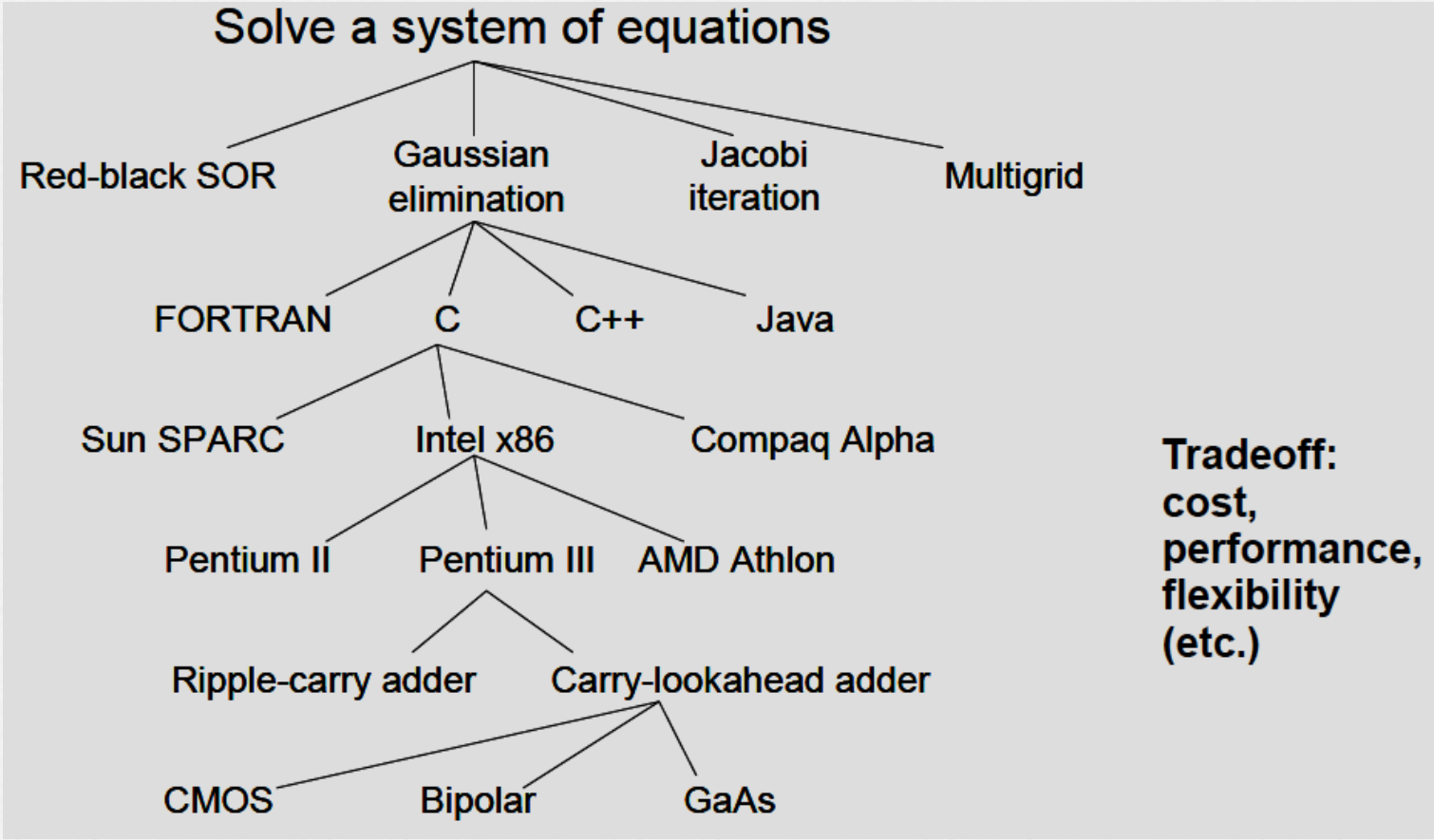
Hardware and Software



Hardware vs Software

- Whatever can be done by hardware can also be done by software, and vice versa
- Hardware implementations are faster but fixed
- Software implementations are more flexible but slower

Many Choices at Each Level



Tradeoff: cost, performance, flexibility (etc.)

What's Next

- We will start with the lowest level: bits