COMP2611 COMPUTER ORGANIZATION
TOPIC 1 INTRODUCTION

Common Number System

- **A number system defines how a number can be represented using distinct symbols.**
- **A number can be represented differently in different systems**
  - For example, the two numbers $(2A)_{16}$ and $(101010)_2$ both refer to the same quantity, $(42)_{10}$.

Positional Notation

- Each digit position has an associated weight
- Numeric values are determined by the implicit positional values of the digits.

$$642_{(10)} = 6 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$

Base or Radix | Hundreds | Tens | Units | Position

- Positional notation as a formula

$$d_{n-1} \times R^{n-1} + d_{n-2} \times R^{n-2} + \ldots + d_1 \times R^1 + d_0 \times R^0$$

  - d_i: digit at i^{th} position
  - R: base or radix

Common Number Systems

System	Base	Symbols	Remark
Decimal	10	0, 1, … 9	used by people
Binary	2	0, 1	used by digital computer
Hexadecimal	16	0, 1, … 9, A, B, … F	great ways to concisely represent a binary sequence

Binary Number System (base 2)

- Used to model the series of electrical signals computers use to represent information
- **Base 2**, two symbols: 0, 1 (binary digits, or just **bits**)
  - 0 : no voltage or an off state
  - 1 : presence of voltage or an on state
- A sequence of bits (a.k.a. **bit sequence**) usually work together

Binary <-> Decimal Conversion

- **Binary -> Decimal**
  - Expand using positional notation

$$
\begin{aligned}
100101_{(2)} &= 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
&= 32 + 0 + 0 + 4 + 0 + 1 \\
&= 37_{(10)}
\end{aligned}
$$

- **Decimal -> Binary**
  - Do the reverse
  - Determine largest power of 2 ≤ number, write template and solve it

$$
\begin{aligned}
37_{(10)} &= ? \times 2^5 + ? \times 2^4 + ? \times 2^3 + ? \times 2^2 + ? \times 2^1 + ? \times 2^0 \\
&= 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0
\end{aligned}
$$

Size vs. Rate/Frequency

- **When dealing with a size (e.g., Memory or file)**
  - Kilo – 2^{10} or 1024
  - Mega – 2^{20} or 1024 Kilo
  - Giga – 2^{30} or 1024 Mega
  - Tera – 2^{40} or 1024 Giga
  - Peta - 2^{50} or 1024 Tera
  - …

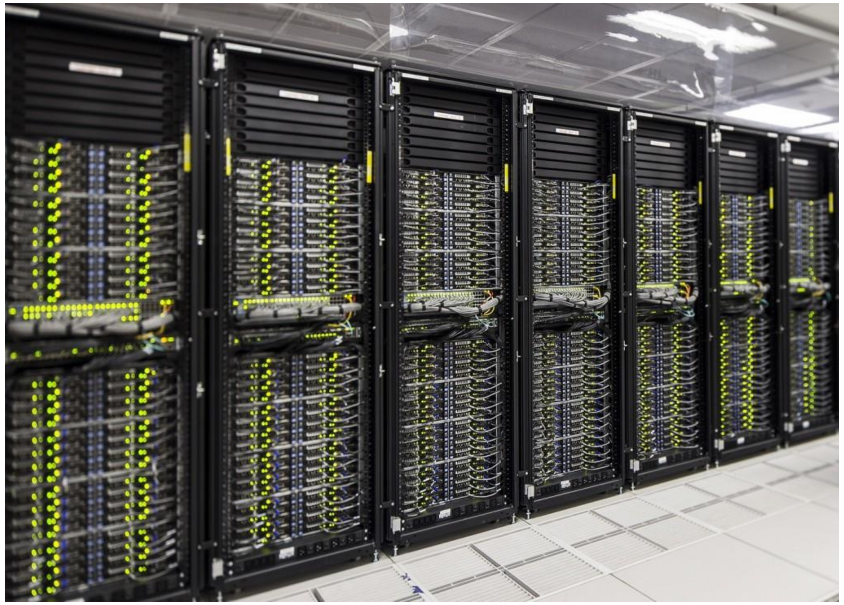
  Example:
  - The memory in my computer is 4 Gigabytes
  - The PPT file for this lecture is 2.5 Megabytes

- **When dealing with a rate/frequency (e.g., # instructions per second, # clock ticks per second)**
  - Kilo – 10^{3} or 1000
  - Mega – 10^{6} or 1000 Kilo
  - Giga – 10^{9} or 1000 Mega
  - Tera – 10^{12} or 1000 Giga
  - Peta - 10^{15} or 1000 Tera
  - …

  Example:
  - The speed of my network card is 1 Gigabit per second
  - The speed of my Intel processor is 2.89 Gigahertz

Classes of Computers

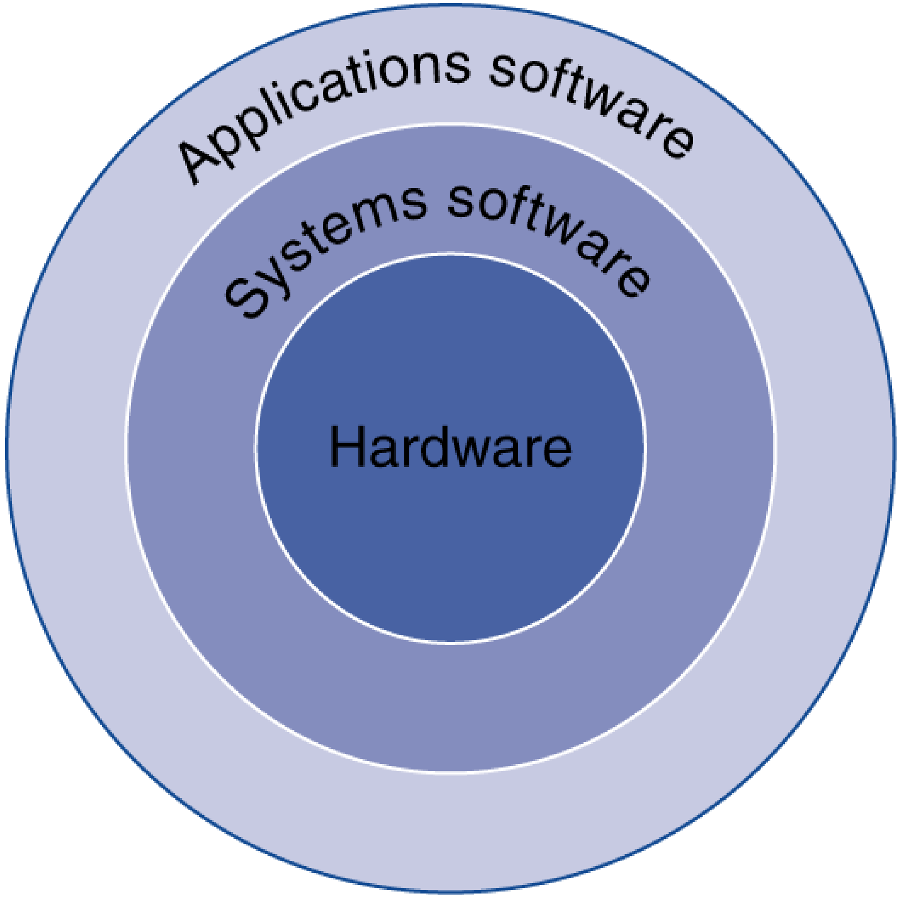
- **Personal computers**
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- **Server computers**
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- **Supercomputers**
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- **Embedded computers**
  - Hidden as components of systems
  - Stringent power/performance/cost constraints



IBM
Z

Raspberry Pi 4 Model B
© Raspberry Pi 2018
GPIO
DISPLAY
ETHERNET
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
Made in the UK
HDMI
CAMERA
POWER IN

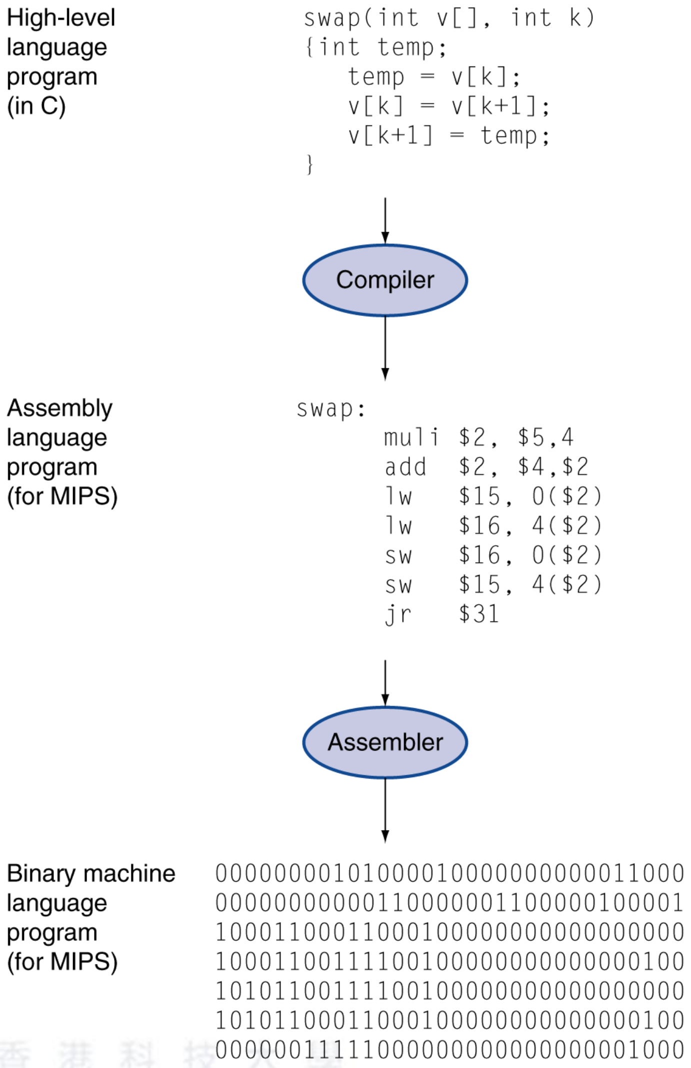
Below Your Program



- **Application software**
  - Written in high-level language
- **System software**
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- **Hardware**
  - Processor, memory, I/O controllers

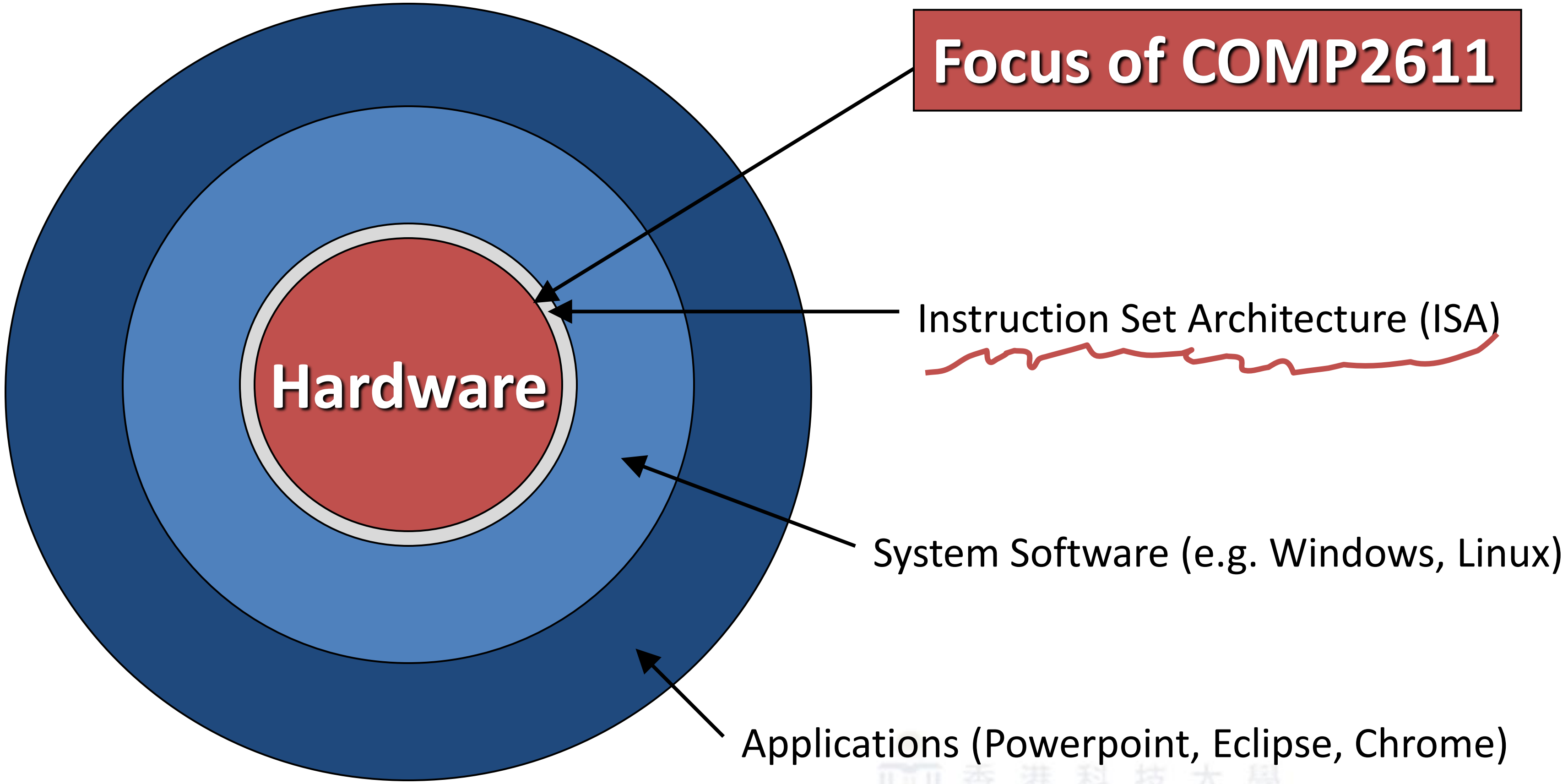
Levels of Program Code

- **High-level language**
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- **Assembly language**
  - Textual representation of instructions
  - Symbolic language
- **Hardware representation**
  - Binary digits (bits)
  - Encoded instructions and data



Levels of Abstraction

Impossible to understand computer components by looking at every single transistor. Instead, abstraction is needed.



Levels of Abstraction (cont'd)

- **Key ideas:**
  - Both hardware and software are organized into **hierarchical layers**.
  - Hierarchical organization helps to cope with system **complexity**.
  - Lower-level details are **hidden** to offer a simpler view at the higher levels.
  - Interaction between levels occurs only through well-defined **interface**.
    - Interface between hardware and software: Instruction set architecture (ISA)

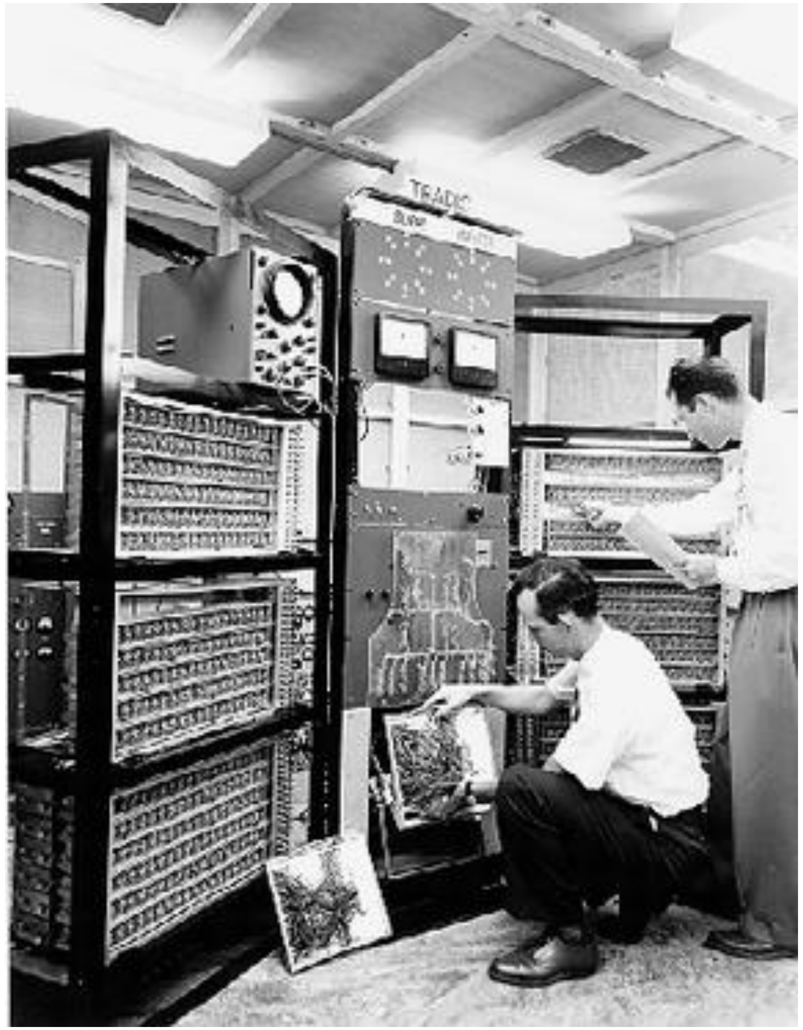
Instruction Set Architecture

An **instruction set architecture (ISA)** provides
an abstract interface between hardware and low-level software.

- **Advantage: allows different implementations of varying cost and performance to follow the same instruction set architecture (i.e., to run the same software).**
  - Example: 80x86, Pentium, Pentium II, Pentium III, Pentium 4 all implement the same ISA

- **Some instruction set architectures:**
  - Intel x86, ARM, MIPS, PowerPC, SPARC

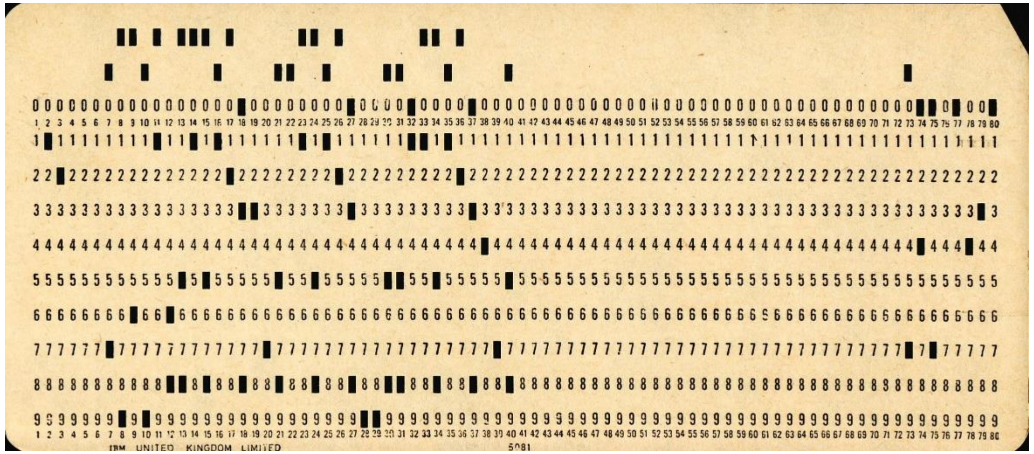
Computer in the Oooooooooold Days



IBM model 350 disk file in 1950’s

Programmer in the Old Days

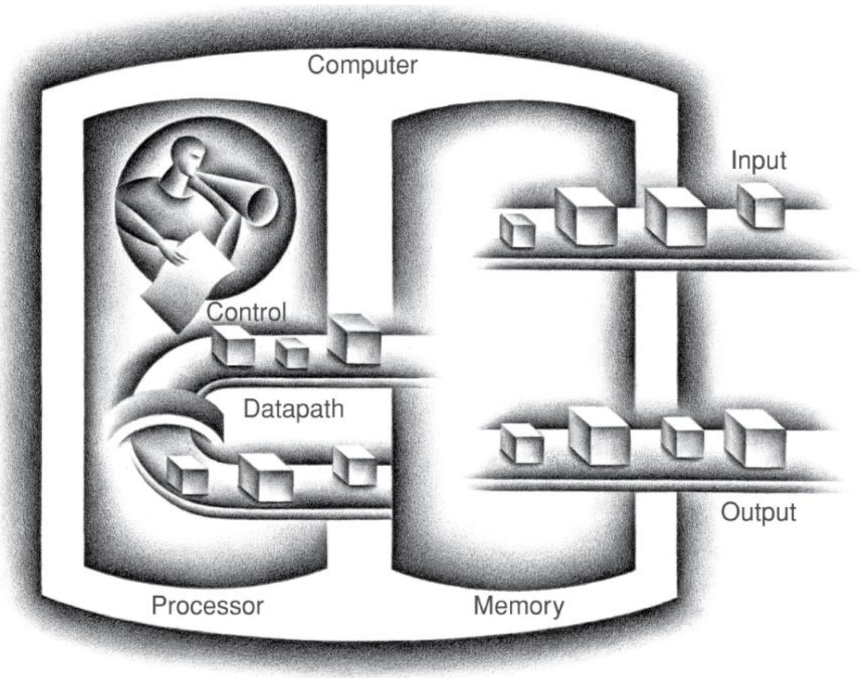
Programmers in 1970’s with punch card



Components of Computer

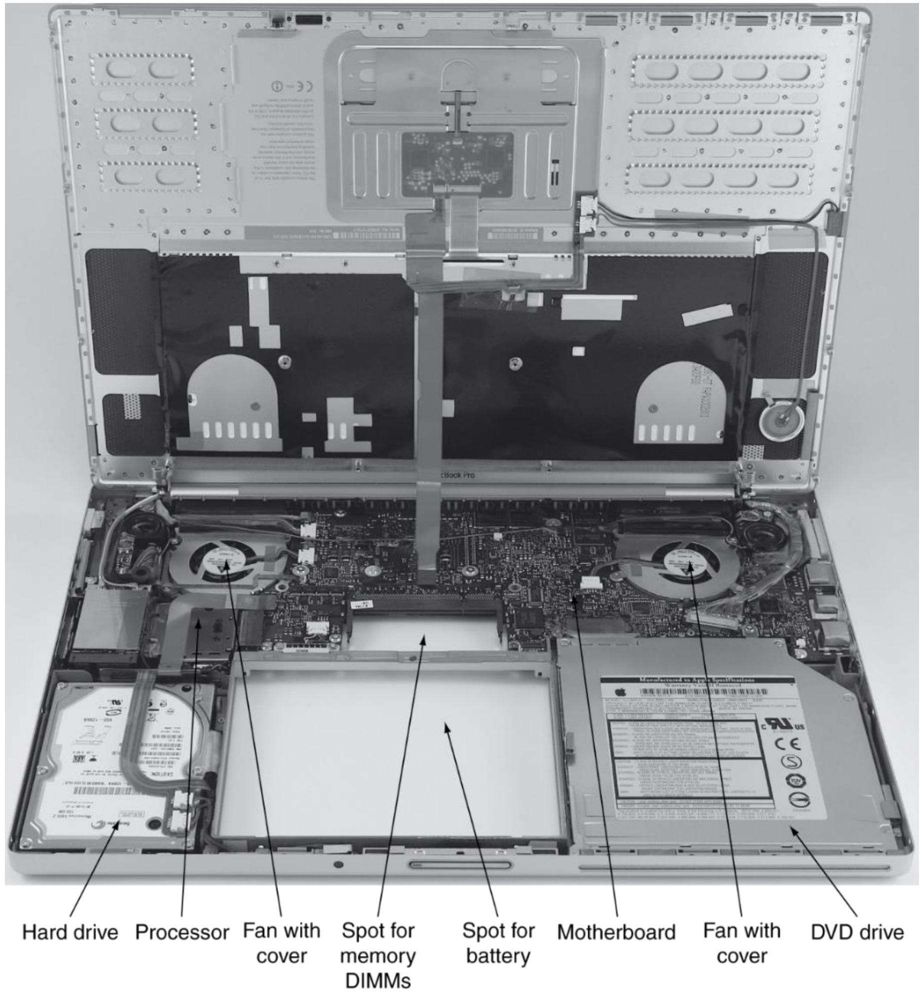
Five Basic Components (all kinds of computers)

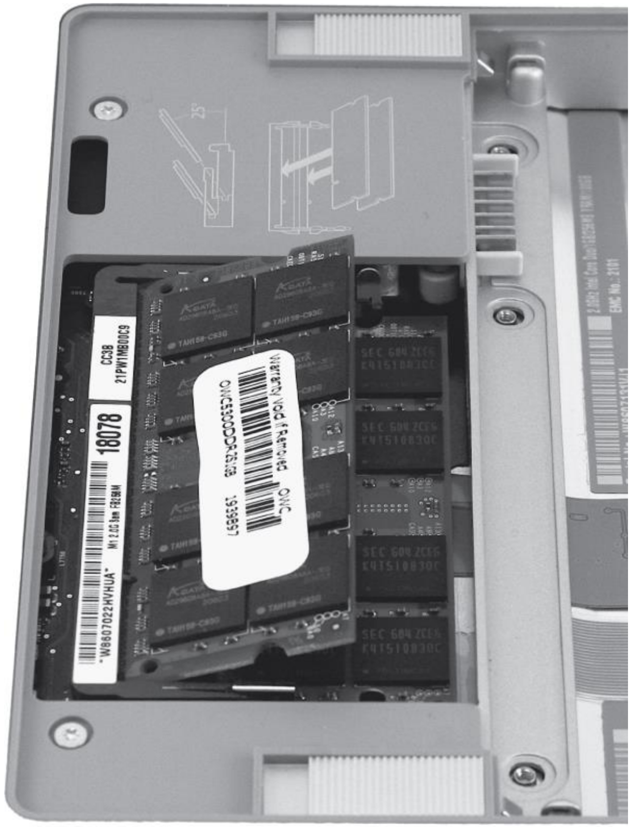
- **Input**:
  - To communicate with the computer
  - Data and instructions transferred to the memory
- **Output**:
  - To communicate with the user
  - Data is read from the memory
- **Memory**:
  - Large store to keep instructions and data
- **Processor, which consists of:**
  - **Datapath**: processes data according to instructions.
  - **Control**: commands the operations of input, output, memory, and datapath according to the instructions.



Anatomy of a Desktop Computer

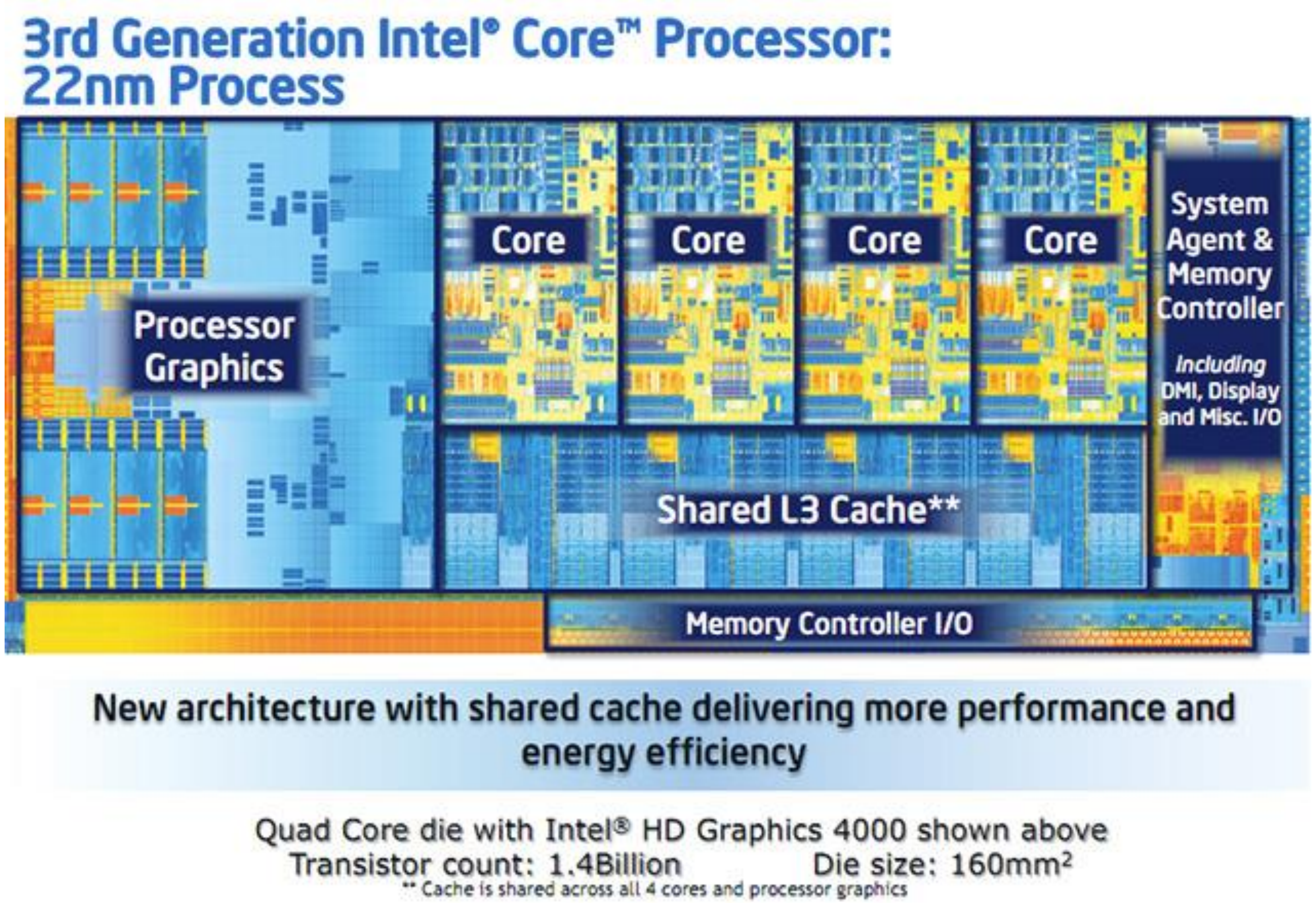
Anatomy of a Laptop Computer





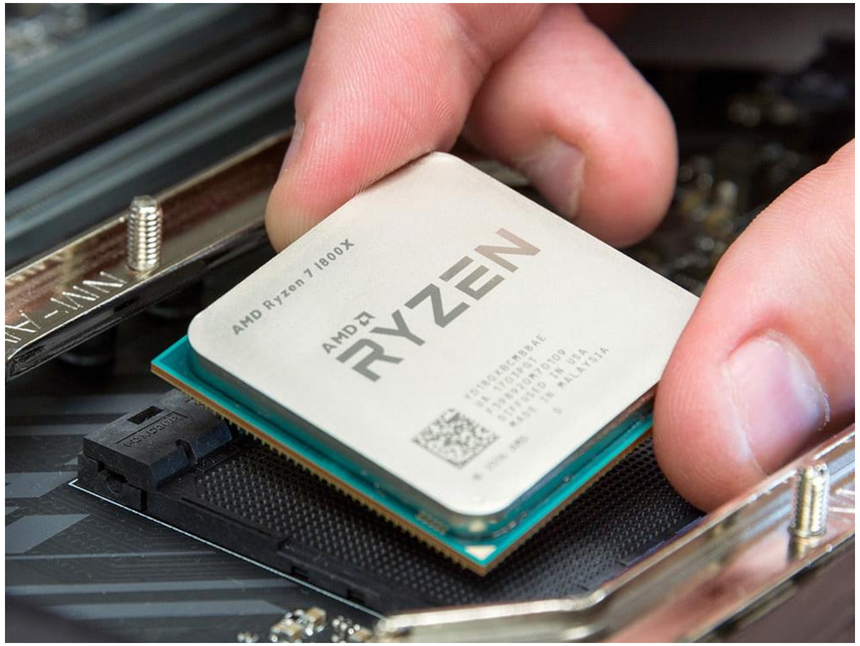
Anatomy of a Computer Processor

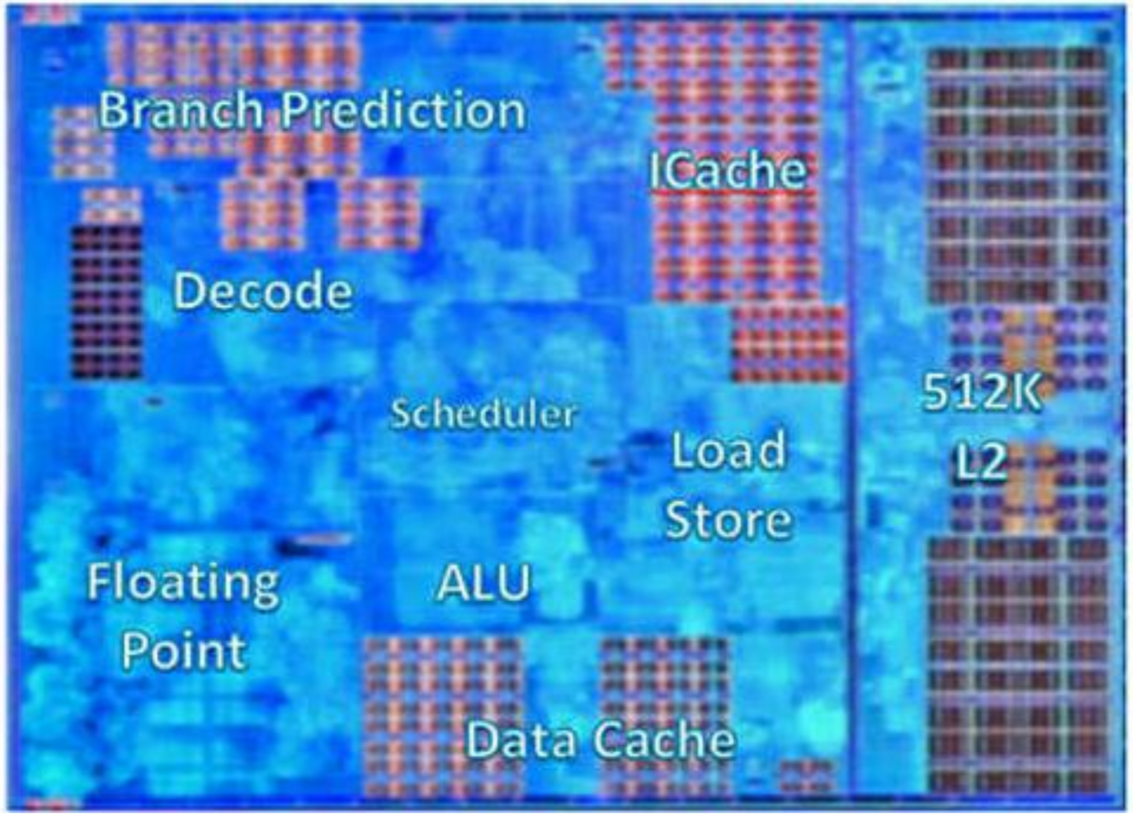
- Intel core i7



Anatomy of a Processor Core

- AMD Ryzen



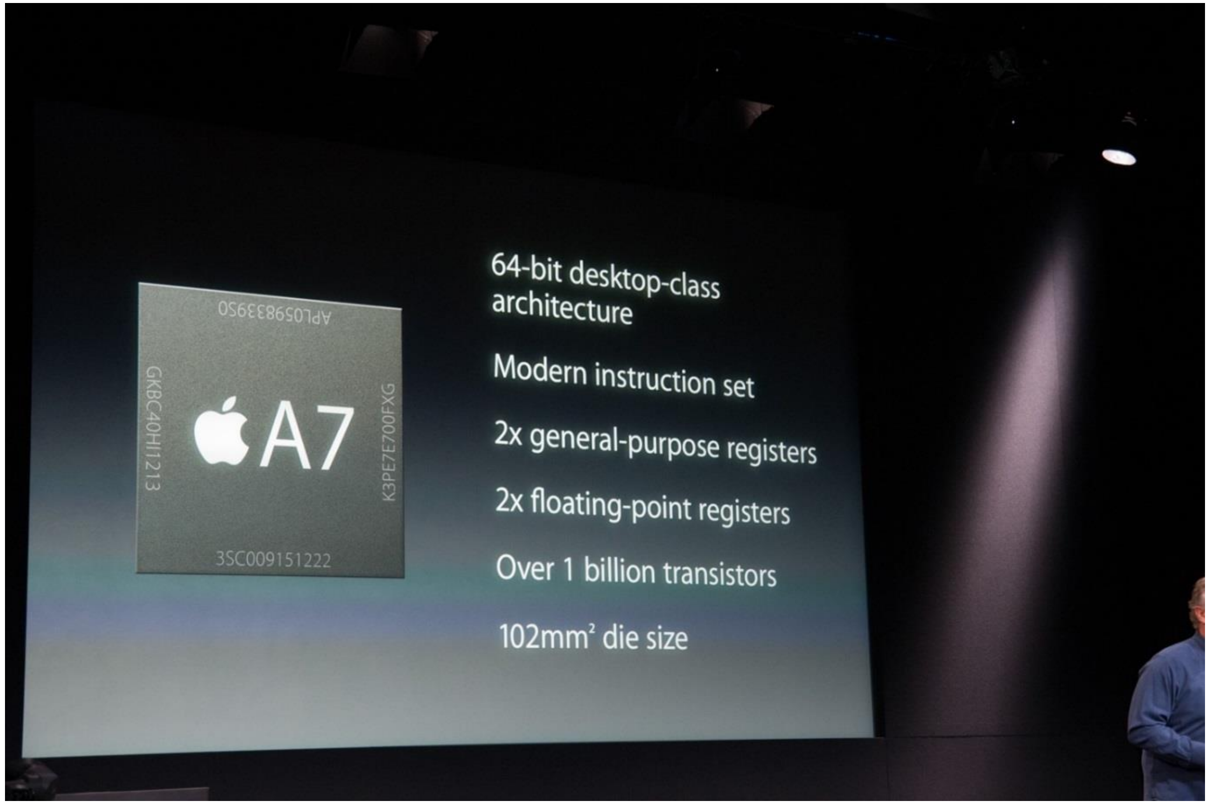


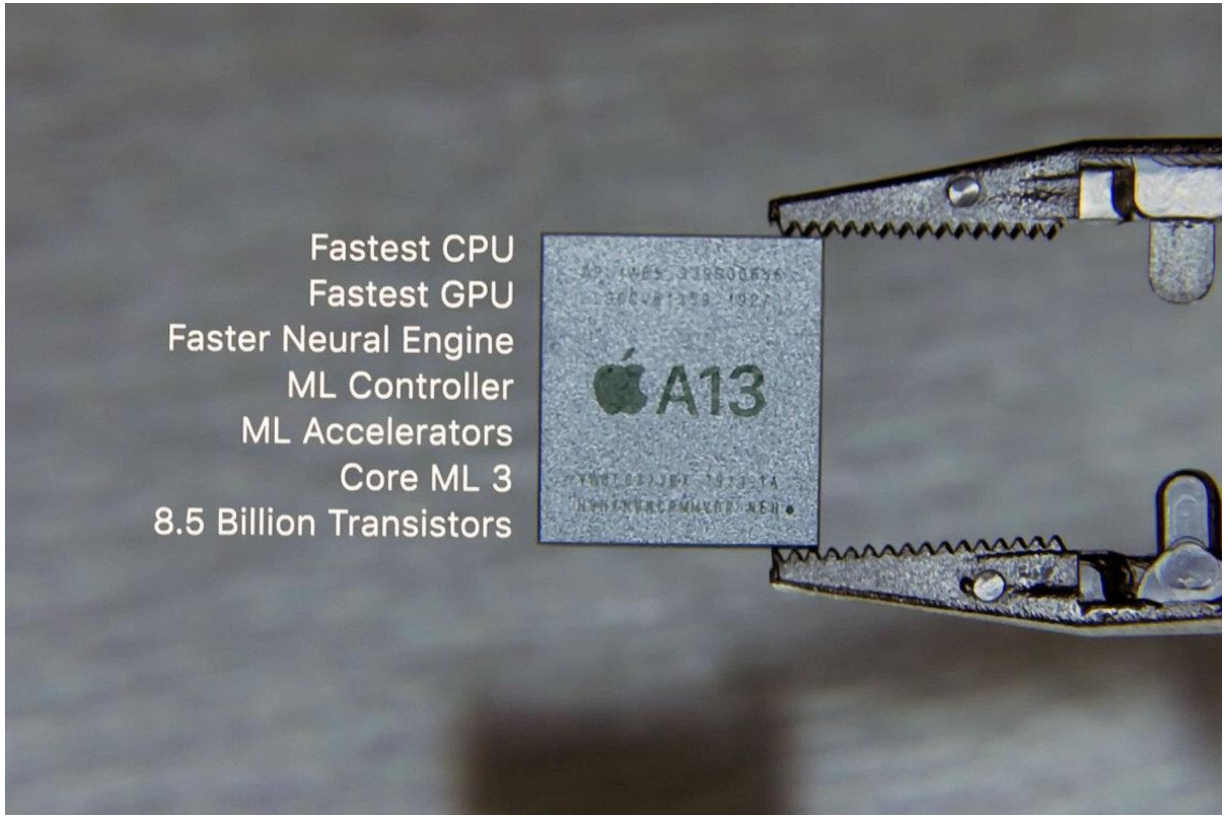
The Computer Revolution

Computers have led to a third revolution for civilization:
agricultural -> industrial -> information

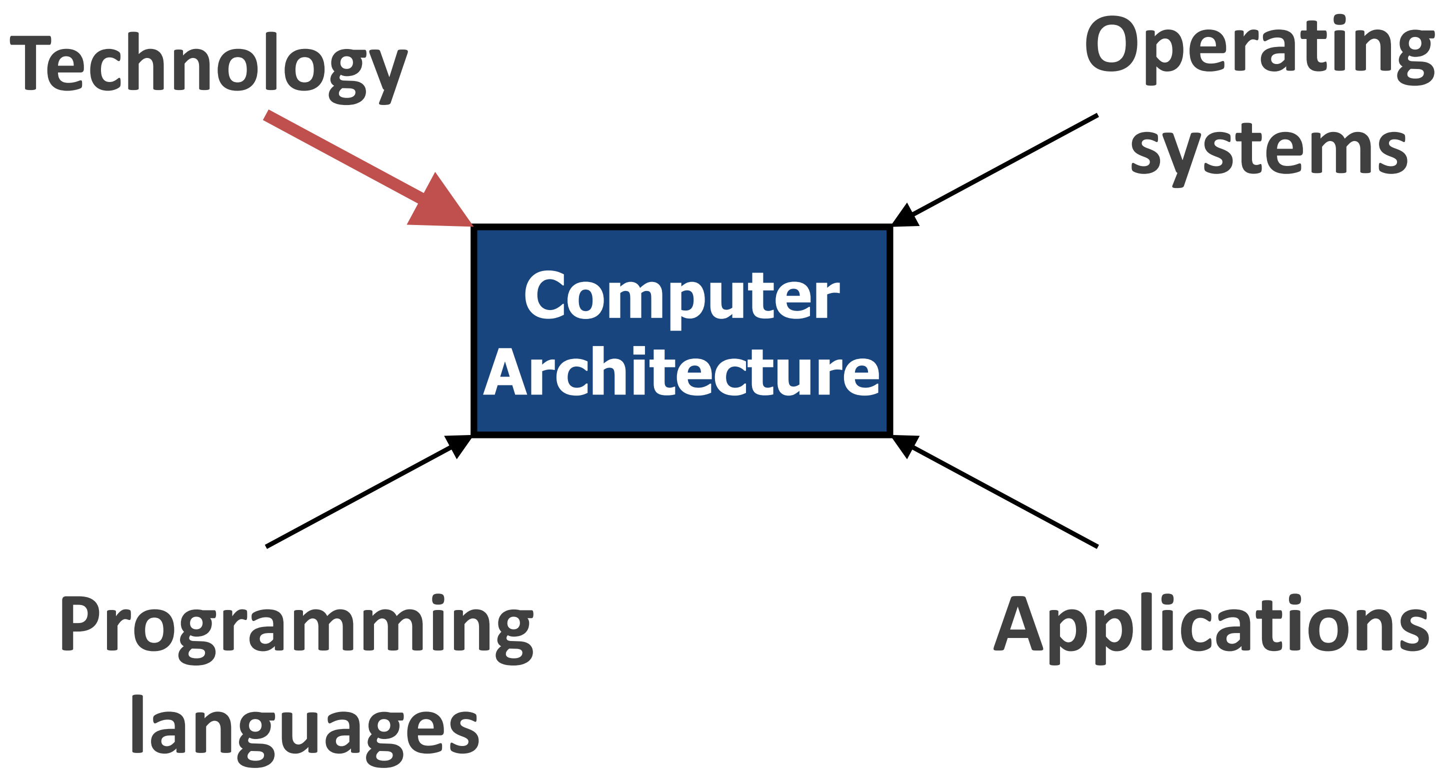
- **Progress in computer technology**
  - Underpinned by Moore's Law
- **Makes novel applications feasible**
  - Computers in automobiles
  - Artificial Intelligence
  - Human genome project
  - World Wide Web
  - Search Engines
- **Computers are pervasive**

What are these highlights?





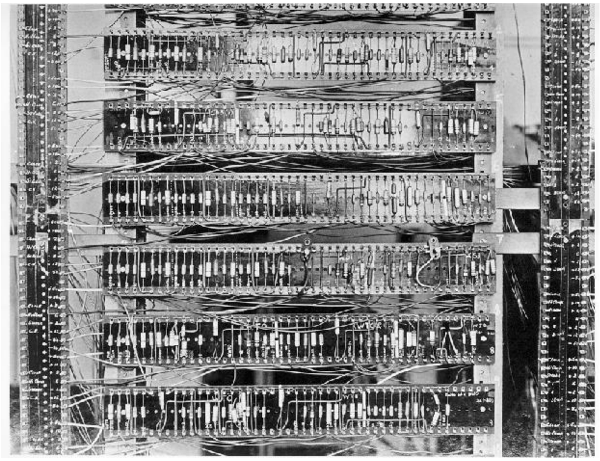
Rapidly Changing Forces on Computer Architecture



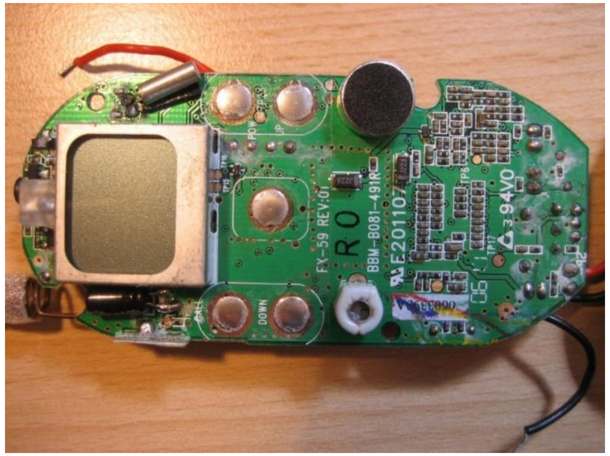
Technology Trend

- Electronics technology continues to evolve
- Increased capacity and performance
- Reduced cost

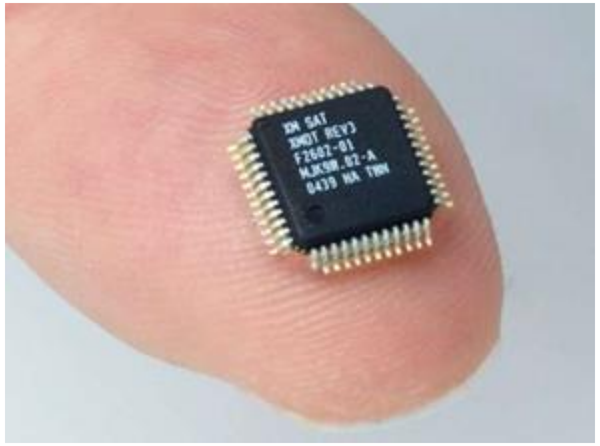
Vacuum Tubes (1950s)



Transistors (1950s and 1960s)



Integrated Circuits (1960s and 70s)

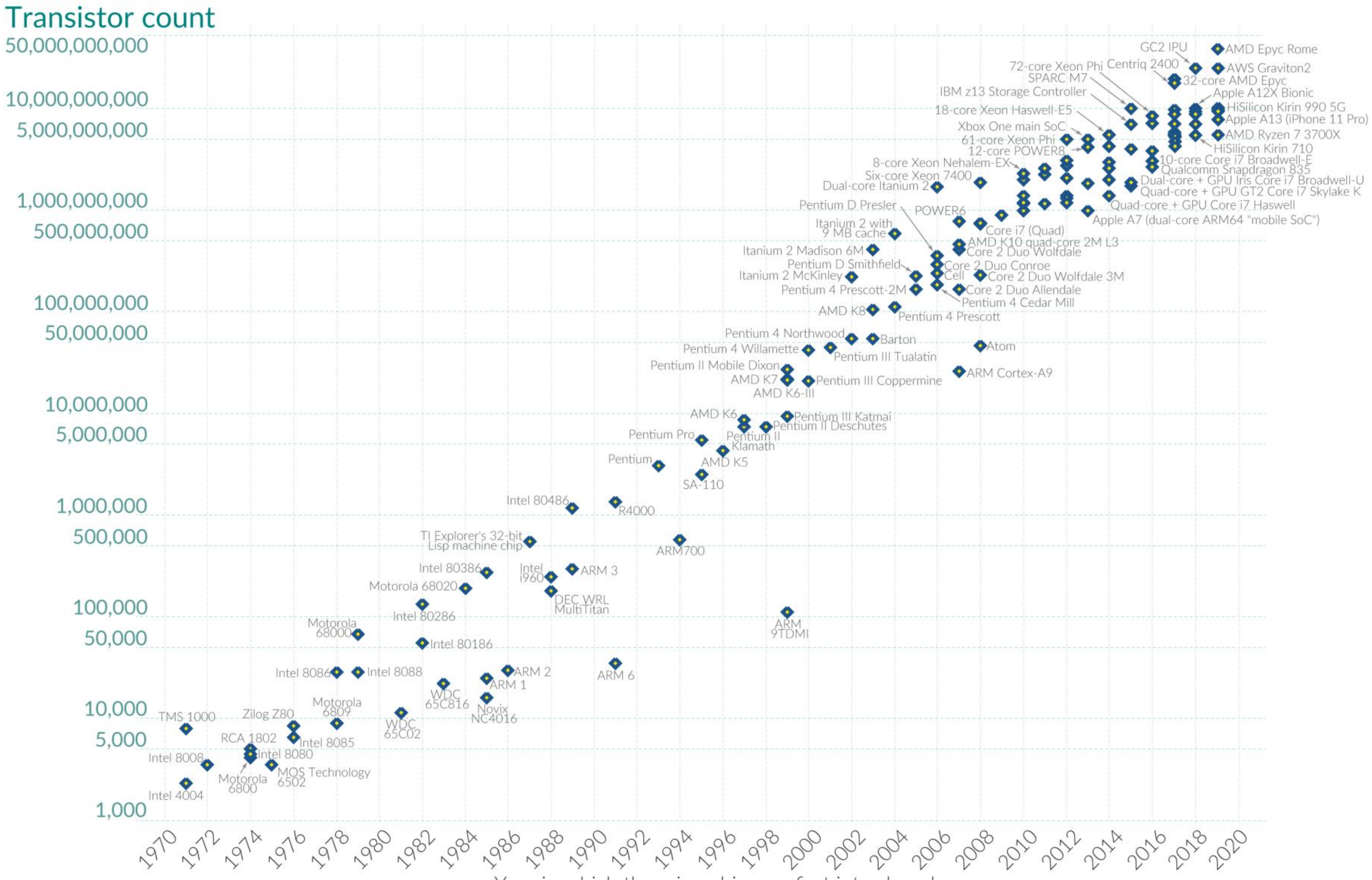


Very Large Scale Integrated (VLSI) Circuit (1980s and on)

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Concluding Remarks

- Recognize the five basic components of a computer
  - input, output, memory, processor (datapath + control)
- Understand the Principle of abstraction
  - Help cope with design complexity by hiding low level details
  - Levels of program code: high-level language, low-level language (e.g. assembly), machine code
- Instruction set architecture
  - The hardware/software interface
- Recognize the technology trend
  - Cost/performance is improving due to underlying technology development