System Architecture
Summer Semester 2023

Jan Reineke
Universität des Saarlandes

ORGANIZATIONAL ISSUES

Dates + Website

- Lectures:
  Wednesdays 8:30-10:00
  Friday 12:30-14:00
- Tutorials:
  Mondays 8-10, 10-12, and 12-14
- Weekly Office Hours

Course website:

https://cms.sic.saarland/sysarch23

Registration until Friday, April 14, 23:59.

Express your tutorial slot preferences!

Lectures

- In presence in Günter Hotz Hörsaal (E2 2)
- Recorded and later available on MS Stream

Tutorials

- Held in presence
- Weekly assignment sheets
  - solutions can be turned in voluntarily
  - first assignment sheet: April 12
  - first tutorials: on April 24
- Additional “tutorial assignments” are discussed during the tutorials

Quizzes

- 6 quizzes throughout the semester
  - during tutorials
  - first quiz on April 24
- You need 50% of the points to be admitted to the exam!

Exam modalities

- End-of-term exam: July 25, 2023, 10:00-12:00
- Re-Exam: September 29, 2023, 10:00-12:00

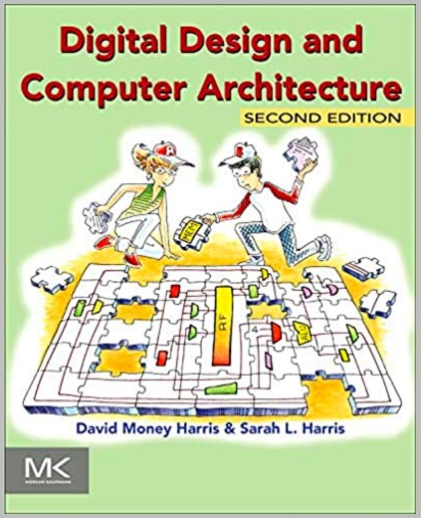
- Requirements to be admitted to the exams:
  - 50% of the points from the quizzes
    - about every two weeks
  - 50% of the points from the projects:
    - Computer architecture project
    - Operating systems project

- Grade: 80% Exam + 20% Project

Literature

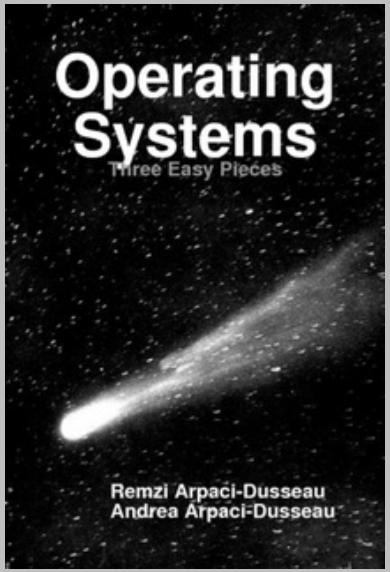
Computer Architecture

B. Becker, P. Molitor:
Technische Informatik:
Eine einführende
Darstellung 2008



D. Harris, S. Harris
Digital Design and
Computer Architecture,
2012

Operating Systems



R.+A. Arpaci-Dusseau:
Operating Systems:
Three Easy Pieces, 2015
www.ostep.org

Online access: https://cms.sic.saarland/sysarch23/2/Literature

Further literature

- A. Tanenbaum: *Structured Computer Organization* (4th Edition). Prentice Hall International 1999.
- D. Patterson, J. Hennessy: *Computer Organization & Design - The Hardware/Software Interface*, Elsevier, 2012.
- J. Hennessy, D. Patterson: *Rechnerarchitektur, Analyse, Entwurf, Bewertung*, Vieweg.
- J. Keller, W. Paul: *Hardware-Design*, Teubner, 1998.
- W. Stallings: *Betriebssysteme: Prinzipien und Umsetzung*. Pearson Studium, 2005.
- A. Tanenbaum: *Moderne Betriebssysteme*, Pearson Studium, 2009.
- A. Tanenbaum: *Modern Operating Systems*, Pearson, 2008 (englische Originalausgabe).

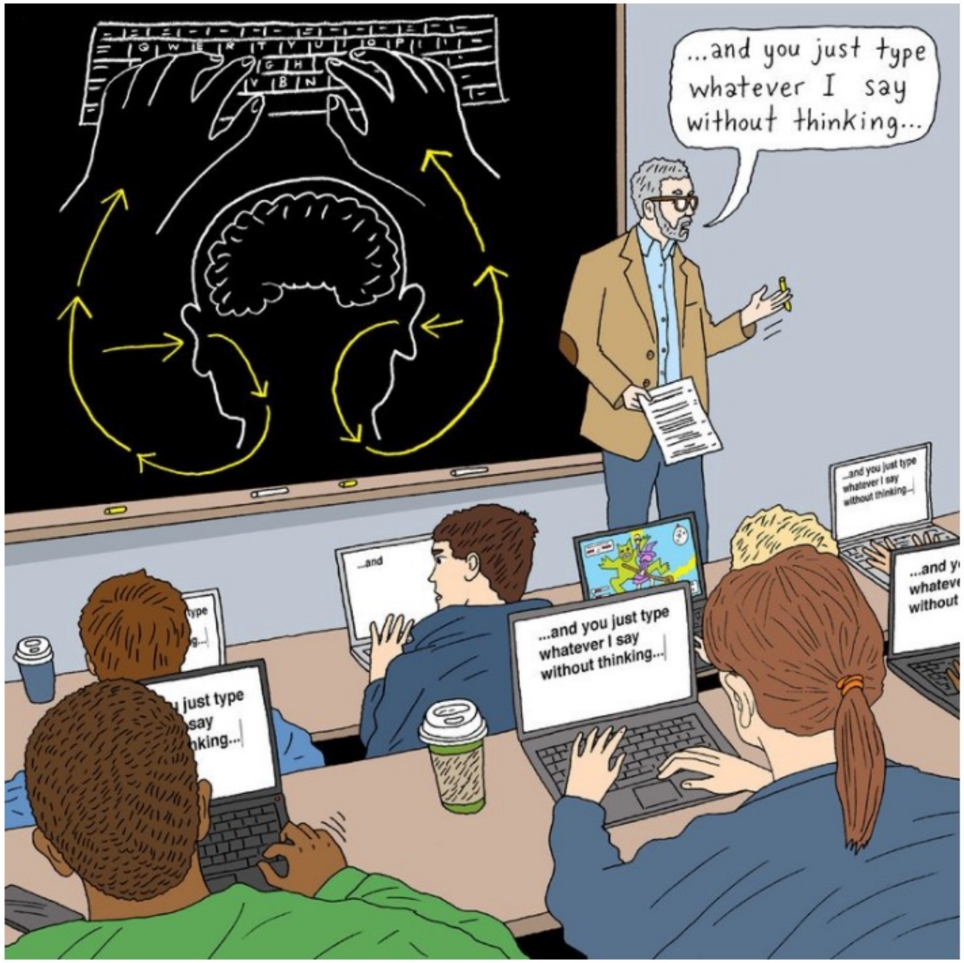
Laptops Are Great. But Not During a Lecture or a Meeting.

Leer en español

Economic View

By SUSAN DYNARSKI NOV. 22, 2017

470



Peter Arkle

> "But what is really interesting is that the learning of students seated near the laptop users was also negatively affected."

RECENT COMMENTS

Garz November 29, 2017
What is true - people are just dumber these days.

Peter C. Herman November 27, 2017
I'm a university English professor, and I've banned all electronics from my classroom for the last five or so years. My usual line is "if...

Tom November 27, 2017
I wonder where these amazing students are who use their laptops in class to augment the learning that is supposed to be going on in class. ...

SEE ALL COMMENTS

https://www.nytimes.com/2017/11/22/business/laptops-not-during-lecture-or-meeting.html

Laptop policy

- Laptop use is only permitted in the **first** and the **last** row.
- In the first row only to create a transcript of the lecture. Send your transcript to the lecturer at the end of each lecture.

INTRODUCTION AND OVERVIEW

Systems – Course contents

- First of all: What is a system?
  - "A system is a group of interacting or interrelated entities that form a unified whole." [1]
- *Here*: particular systems, in which at least one of the components is a computer.
  - Computer systems
    - General-purpose computer, Laptop, Desktop, Server, Tablet, Smartphone, …
    - Embedded computer in cars, planes, washing machines, TVs, …
  - Operating systems

[1] https://en.wikipedia.org/wiki/System

Structure of computers (1)

Applications

Operating System

Hardware

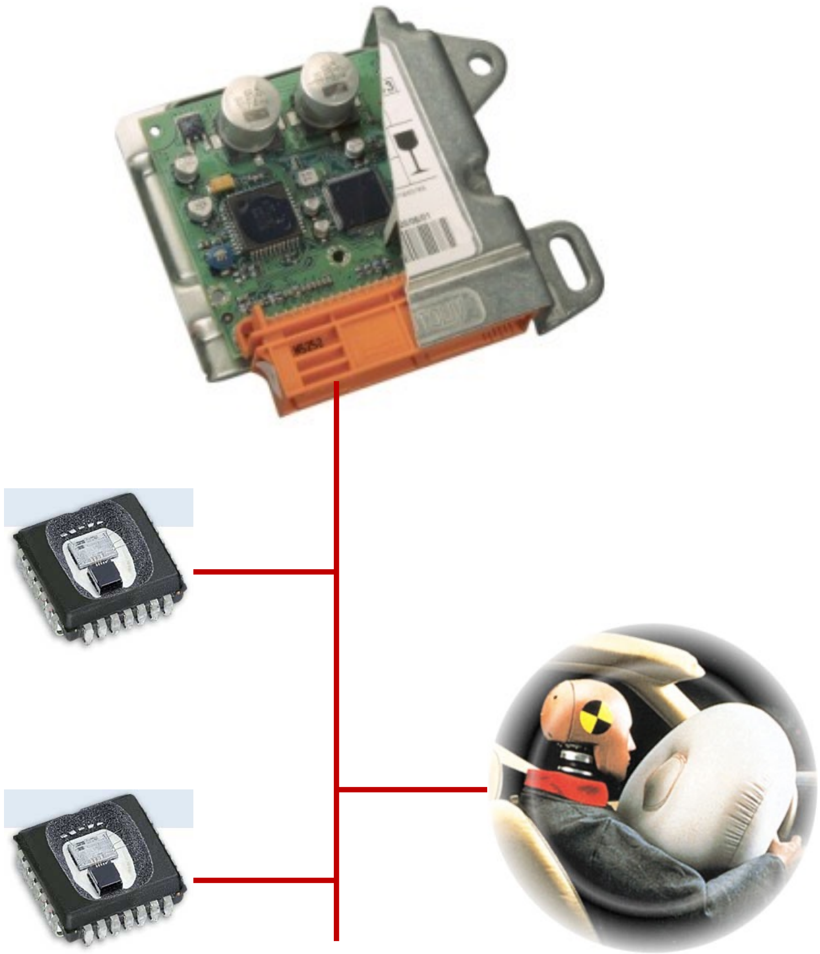
Structure of computers (2)

Apps

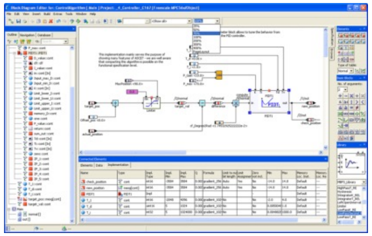
Operating System

Hardware

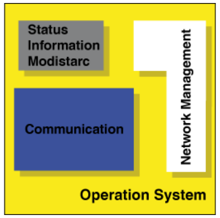
Structure of computers (3)



Application



Operating System



OSEK VDX

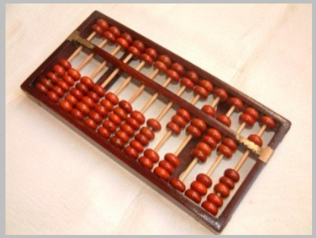
Hardware

Market shares

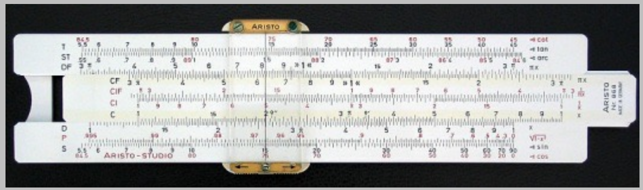
General purpose computers ~2%
Embedded computers ~98%

Historical developments (1)

- **ca. 1100 BC**
– Abacus: ancient calculating tool
  – exact origins unknown
  – different versions
  – still in use in Asia



- **1629**
– Slide rule (dt. Rechenschieber) (William Oughtred)
  "mechanical analog computer"



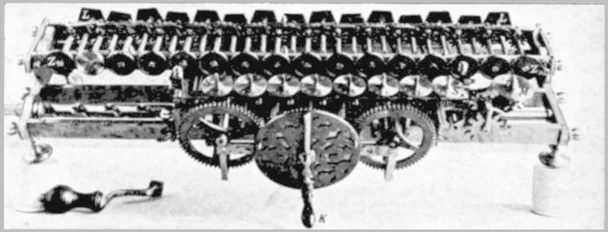
- **1642**
– Pascaline (Blaise Pascal)
  – mechanical calculator
  – limited to **addition**

[en.wikipedia.org]

Historical developments (2)

- **1666**
- Foundations of Logic (Gottfried Wilhelm Leibniz)

- **1673**
- Mechanical calculator, *"Stepped Reckoner"* (Leibniz)
  - **addition**, **subtraction**, **multiplication**, division



[en.wikipedia.org]

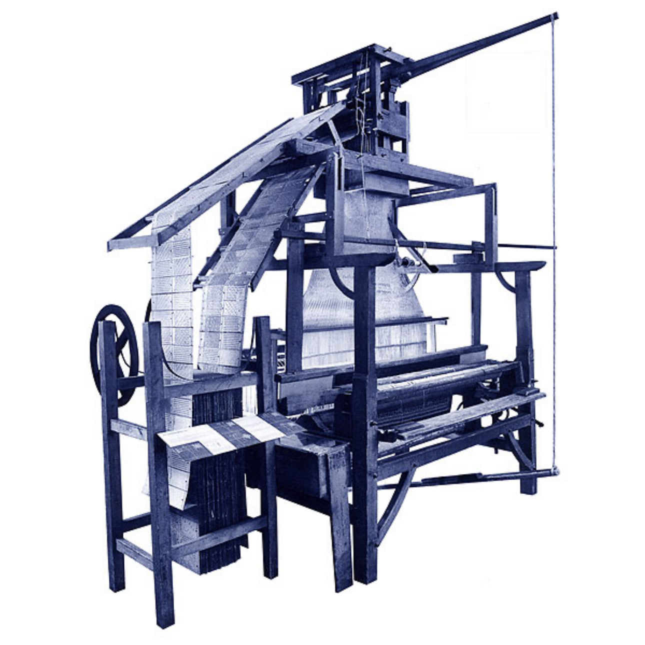
„Es ist unwürdig, die Zeit von hervorragenden Leuten mit knechtischen Rechenarbeiten zu verschwenden, weil bei Einsatz einer Maschine auch der Einfältigste die Ergebnisse sicher hinschreiben kann."

Gottfried Wilhelm Leibniz

"…It is beneath the dignity of excellent men to waste their time in calculation when any peasant could do the work just as accurately with the aid of a machine."

Historical developments (3)

- **1804**
  - Jacquard machine (Joseph Jacquard)
    (dt. Automatischer Webstuhl)
    - punched cards determine patterns
    - holes control raising and lowering of chaining threads
    - first ***Read-Only Memory, ROM***



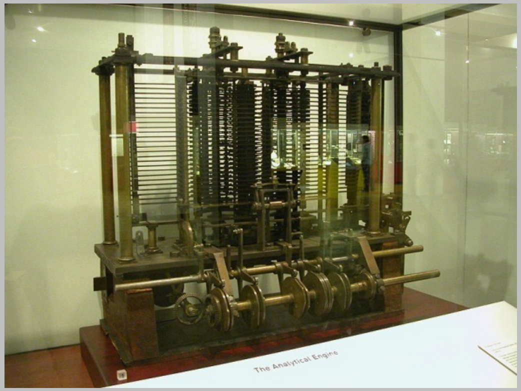
[Deutsches Museum München]

Historical developments (4)

- **1834**

– Analytical Engine (Charles Babbage)

  - theoretically programmable
  - first universal computer
  - **arithmetic unit** *"Mill"*
  - **memory** *"Store"*
  - based on punched cards
  - based on decimal system
  - unfortunately, never completed



[Bruno Barral, (ByB)]

Historical developments (5)

- **1847**
- **Boolean Algebra** (George Boole)
  - binary logic operations
  - basis for today's digital computers

- **1936**
- Turing machine (Alan Turing)
  - mathematical model of computation
  - notion of computability
  - Church-Turing thesis

Historical developments (6)

- **1940**

– Z3 (Konrad Zuse)

  – based on relays (2.200 relays)
  – 5,3 Hertz clock speed
  – 22-digit **binary numbers (floating-point format)**
  – decimal input and output
  – memory with 64 words
  – Overlapped execution of subsequent instructions: "**Pipelining**"
  – **addition** in 3 cycles, **multiplication** in 16 cycles
  – theoretically Turing-complete

Konrad Zuse (1910-1995)

[en.wikipedia.org]

Historical developments (7)

- **1945**

– von Neumann architecture (Presper Eckert, John Mauchly, John von Neumann)

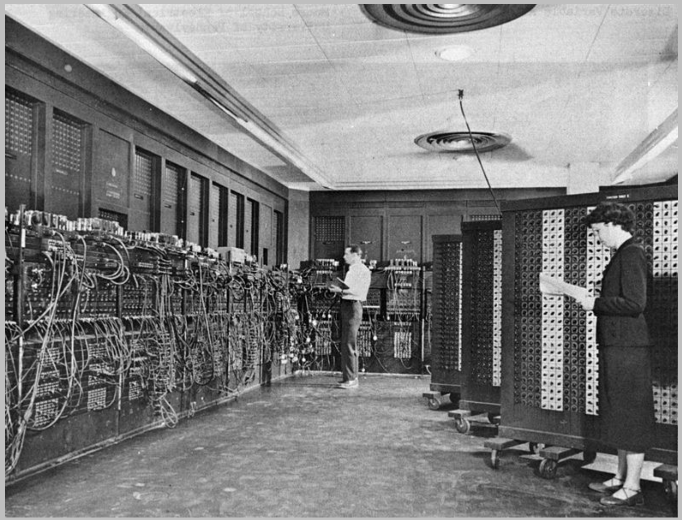
  - classical computer architecture
  - four main components:
    control unit, compute unit, memory, input/output mechanisms
  - programs and data *in the same* memory
  - compute unit with **arithmetic logic unit** (ALU) and **registers**
  - control unit with "*fetch-decode-execute*" instruction cycle
  - **binary coding**

Historical developments (8)

- **1946**

– ENIAC (John Mauchly, Presper Eckert): Electronic Numerical Integrator and Computer

  - 130 m^2, 30 tons, 140 kW
  - ca. 5.000 additions per second
  - programmed via cables
  - I/O via punched cards
  - still based on decimal system
  - built for ballistic computations

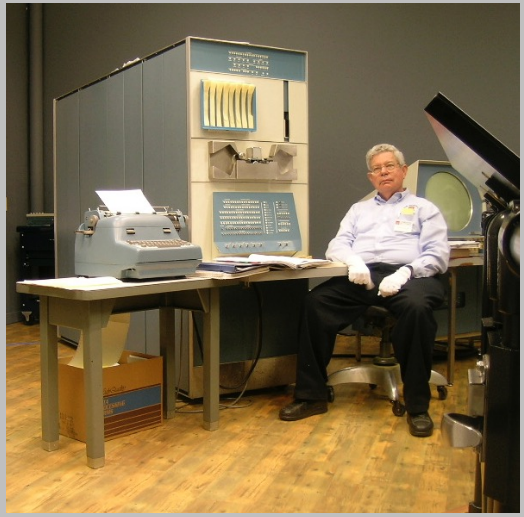


[en.wikipedia.org]

Historical developments (9)

- **1959**
- integrated circuit (Jack Kilby)

- **1961**
- PDP-1 (DEC): Programmed Data Processor 1
  - based on **transistors**
  - magnetic core memory for 4096 18-Bit words
  - 200 kHz clock rate
  - CRT, 512 x 512 pixel graphics



[en.wikipedia.org]

Historical developments (10)

- **1965**

– IBM 360

  – first family of computers with fixed **instruction set**
  – limited multitasking ability
  – 32-Bit words
  – 16 MB address space

[en.wikipedia.org]

Software / Hardware — Instruction set = interface between SW and HW

Historical developments (11)

Popular today:
SIMD =
Single
Instruction
Multiple
Data

- **1976**
  – Cray-1 – first vector processor
- **1980**
  – Hennessy und Patterson put forth **Reduced Instruction Set Computer** (RISC) [2018: Turing Award]
- **1985**
  – **MIPS** – first commercially successful RISC microprocessor
  – Altera and Xilinx: first Field-programmable Gate Arrays (FPGAs)
- **1987**
  – *Connection Machine*
    – first massively parallel computer with 65.536 processors

Historical developments (12)

- **1997**

– Pentium bug: buggy floating-point unit

- – loss of 475 million dollars
- – subsequently: increased use of formal methods for verification

- **2000**

– first microprocessor with a clock frequency of 1 GHz

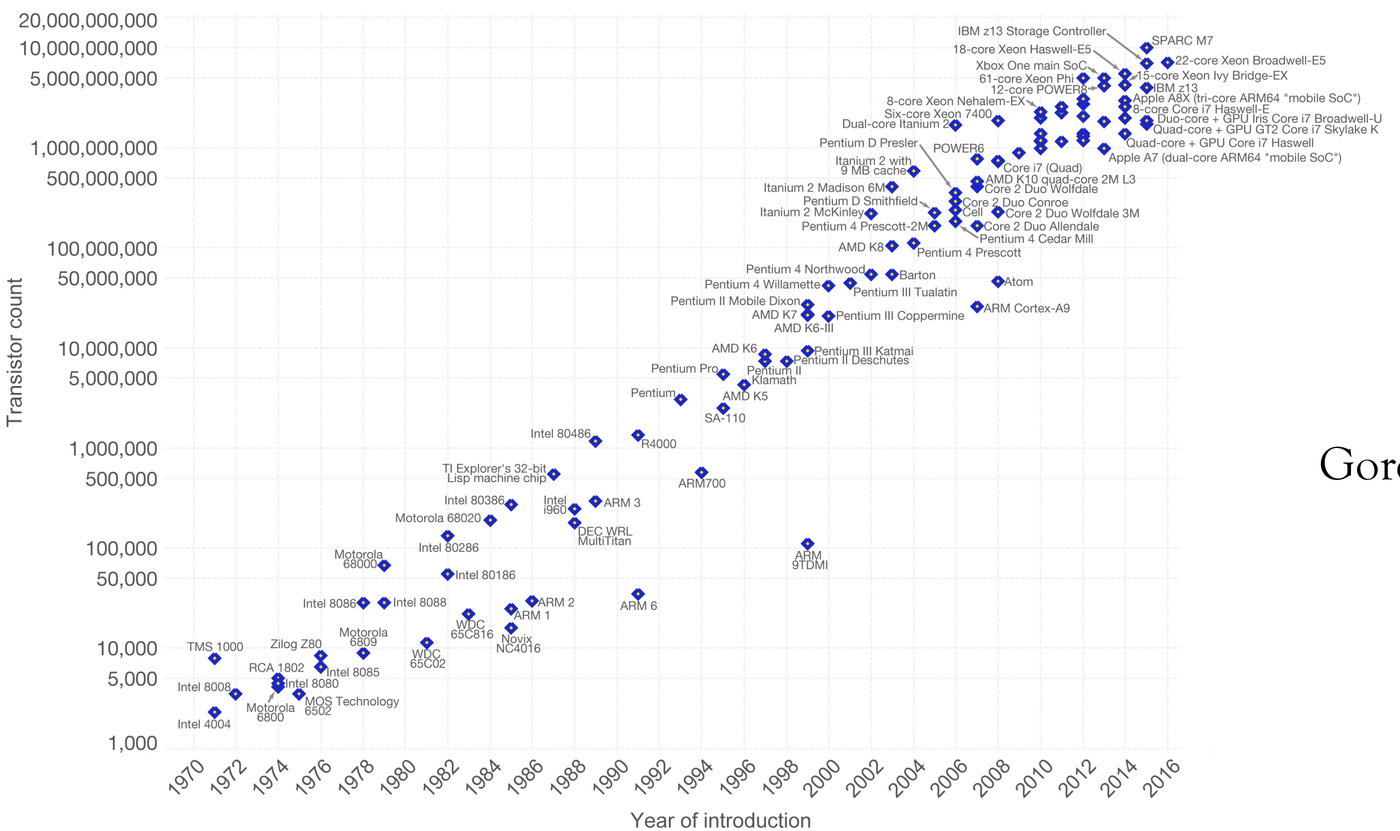
Growth in complexity

Moore's law: number of transistors doubles every 18 months

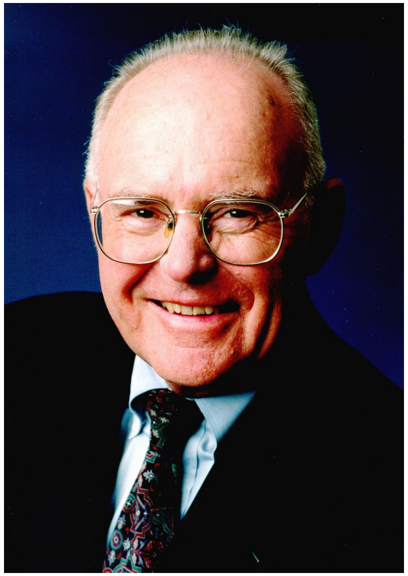
Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

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 as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

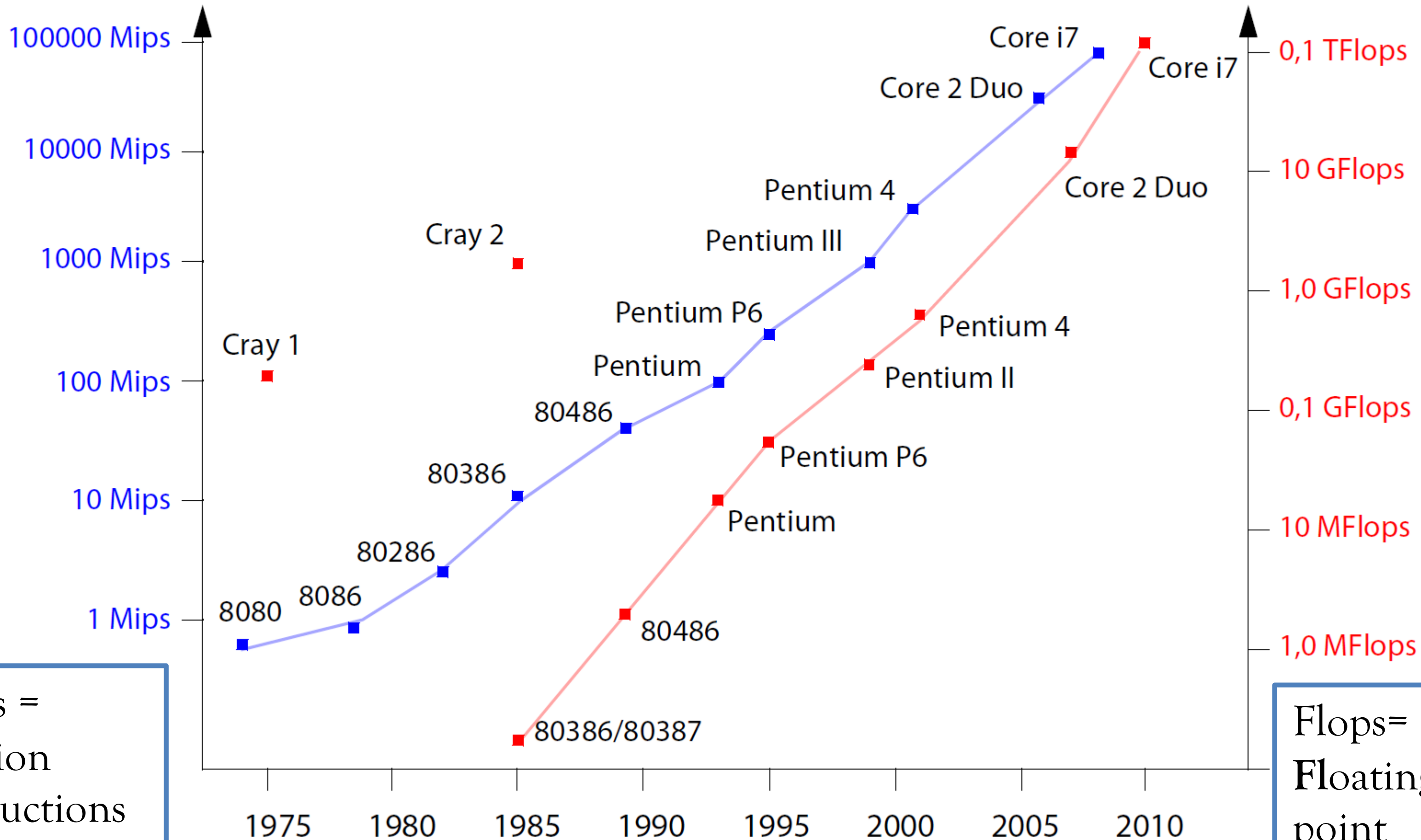


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.



Gordon Moore (1929-2023)
Intel Founder

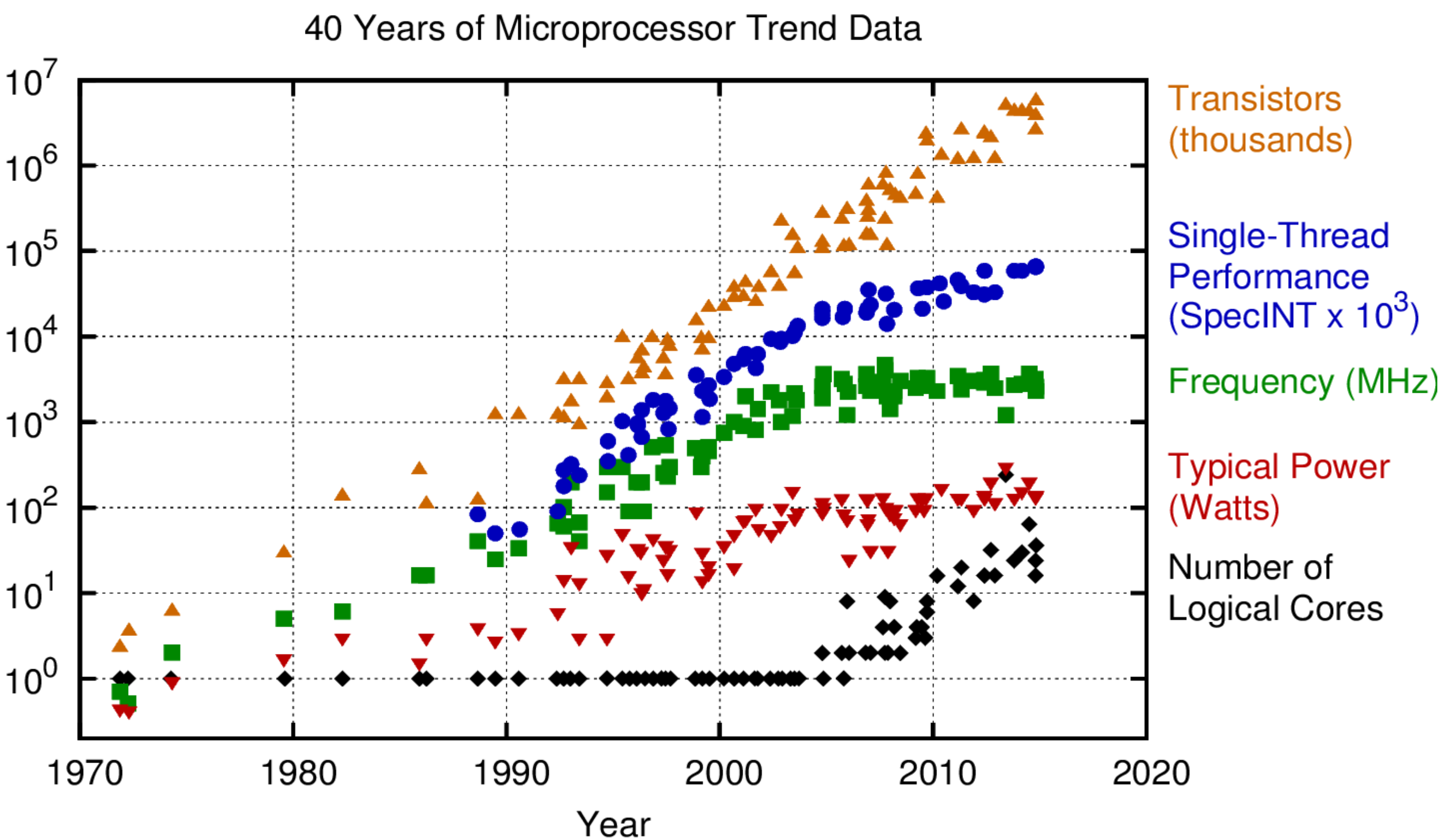
Performance improvements: The good old (?) days



Mips = **M**illion **i**nstructions **p**er **s**econd

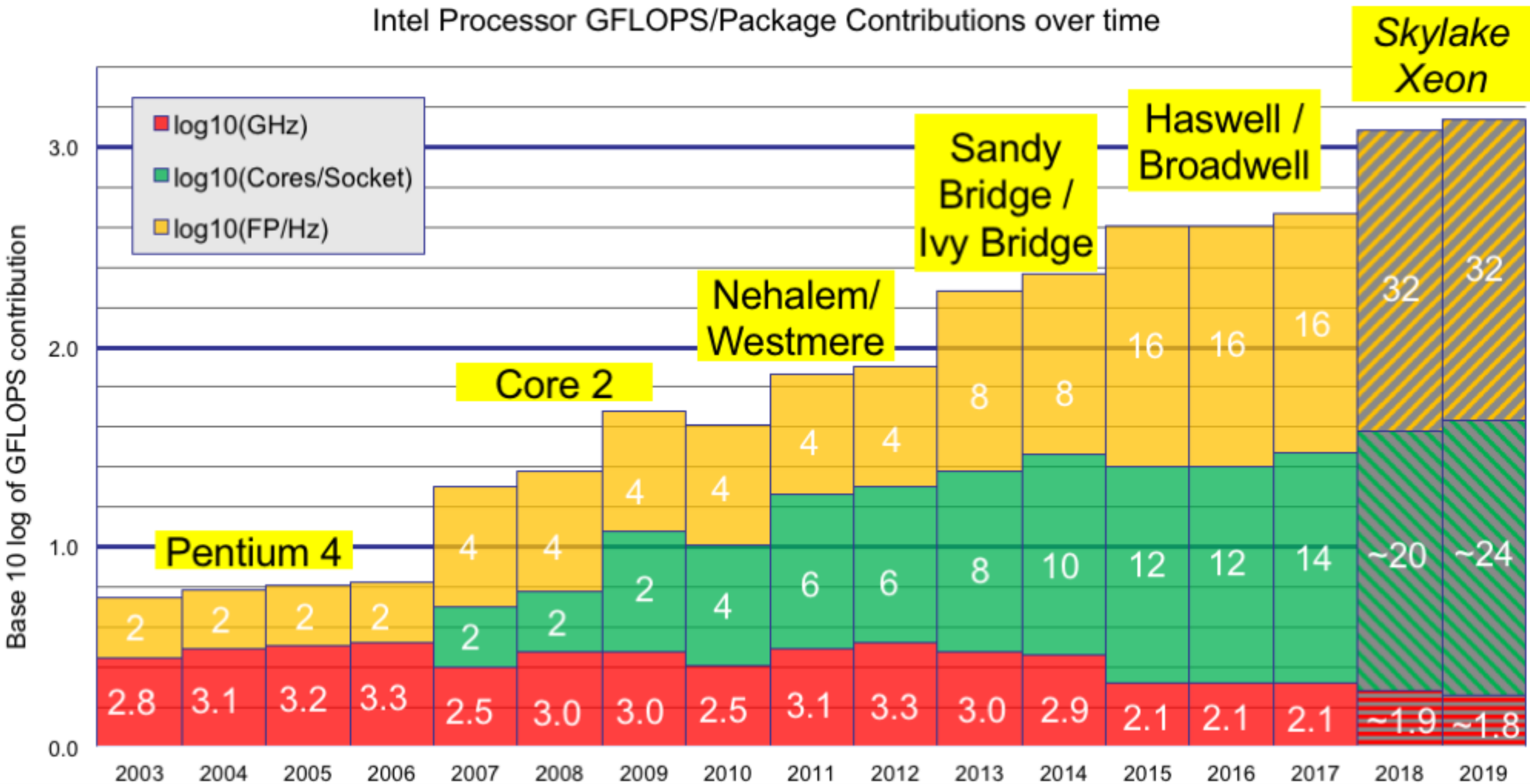
Flops= **Fl**oating-point **o**perations **p**er **s**econd

“Power wall”: no more frequency gains



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

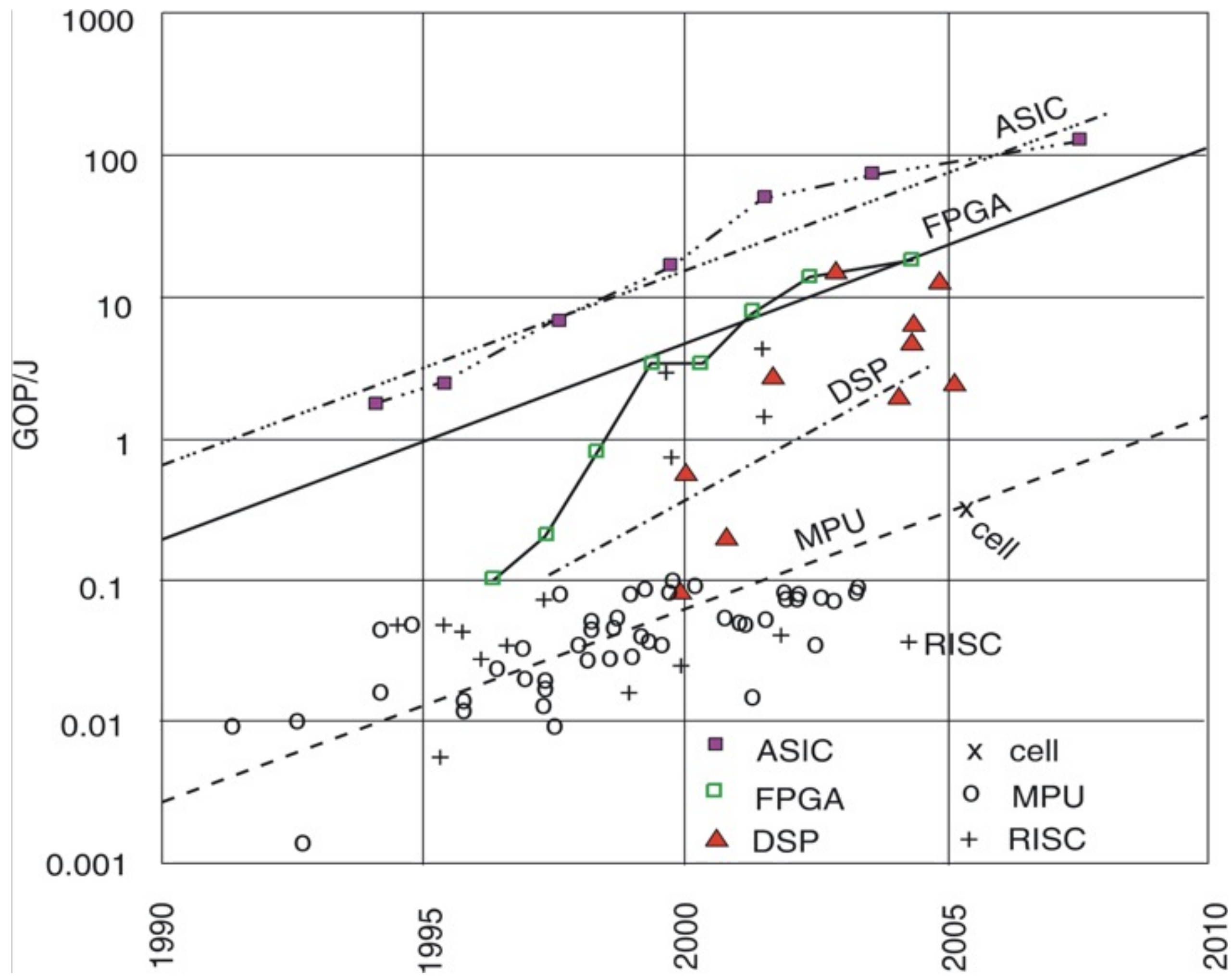
1. Consequence of "Power wall": More parallelism



2. Consequence of "Power wall": Trend towards specialized hardware

FPGA= **F**ield-**p**rogrammable **g**ate **a**rray

ASIC = **A**pplication-**s**pecific **i**ntegrated **c**ircuit



= Giga Operations per Joule

(logarithmic scale)

Historical developments (13)

- **2005**

– first dual-core micro processor

- **2015**

– RISC-V: an open standard instruction set architecture

- **2016**

– Google develops Tensor Processing Unit, an ASIC for machine-learning applications, 80x greater energy efficiency than general-purpose CPUs

- **2018**

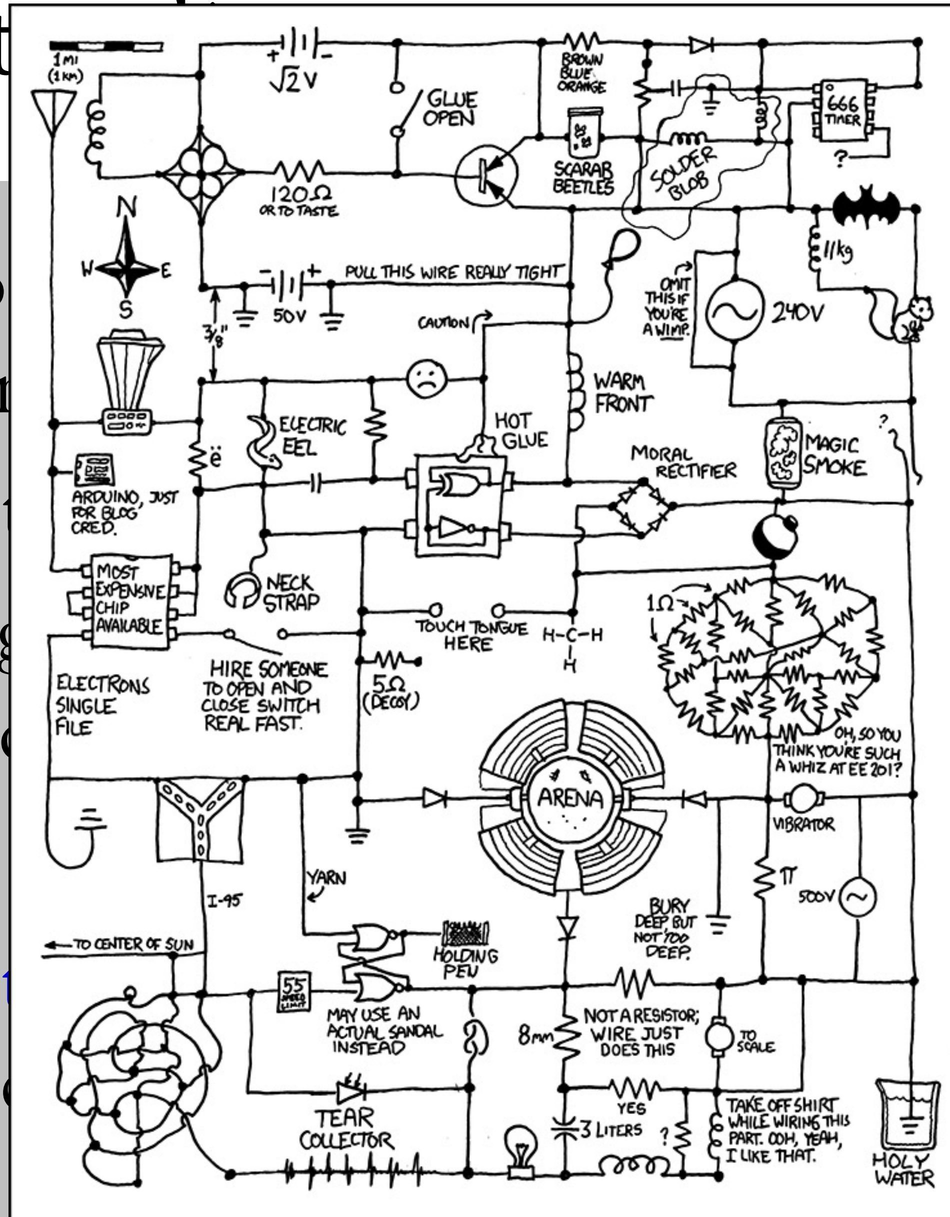
– Spectre and Meltdown:
security vulnerability in processors
with speculative execution and **caches**

Underst

How do yo … sting of
many co

- a comput … ransistors?
- operating … ses,
applicati … code?

→ Abstract … stems with
layers an



[XKCD 730]

Understanding systems

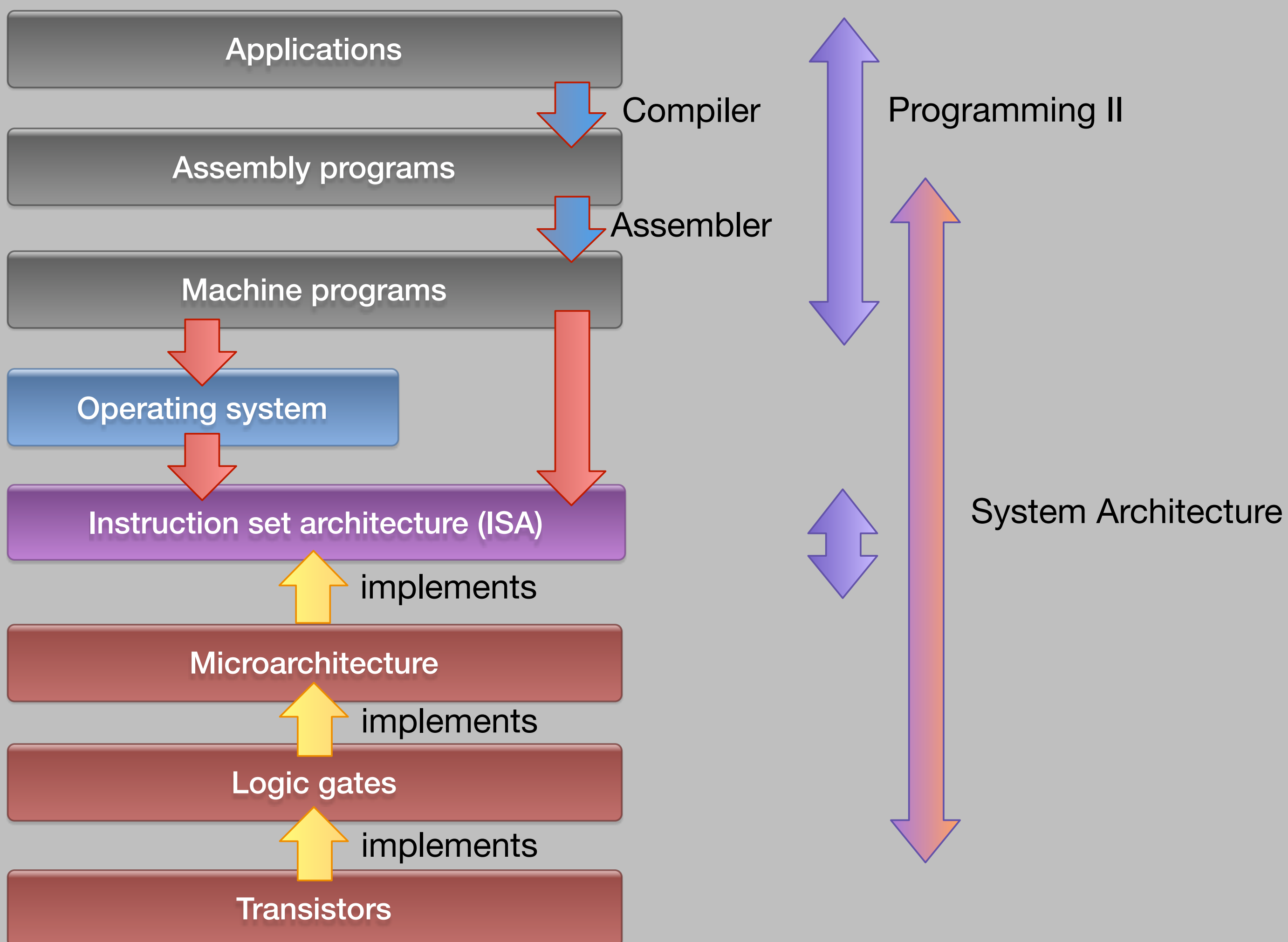
How do you **understand** a system consisting of many components:

- a computer, consisting of 10 billion transistors?
- operating systems, compilers, data bases, applications with millions of lines of code?

→ Abstraction, which often leads to systems with layers and hierarchies

[XKCD 730]

Abstraction layers in computer systems



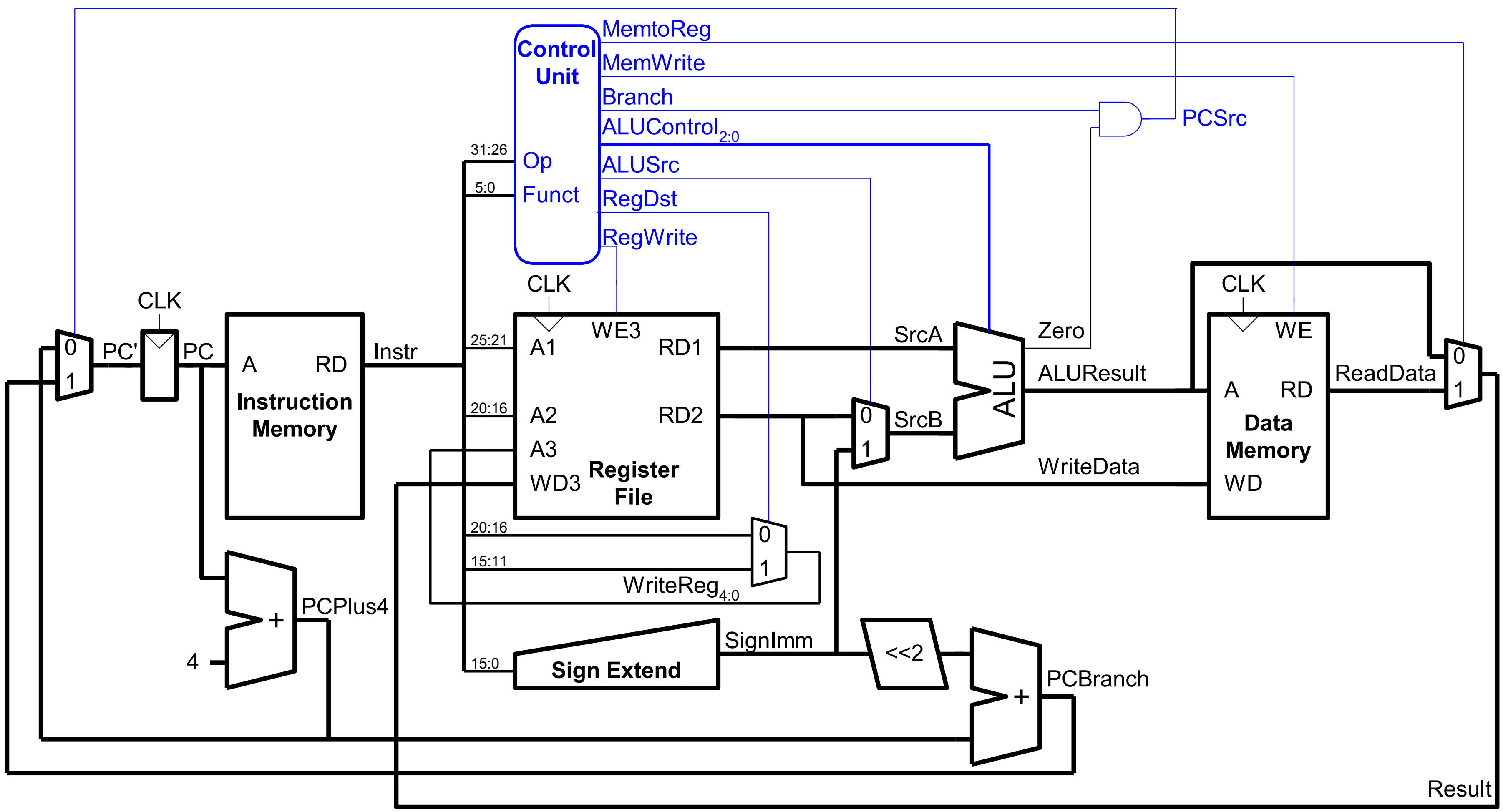
COMPUTER ARCHITECTURE

Mode of operation of processors

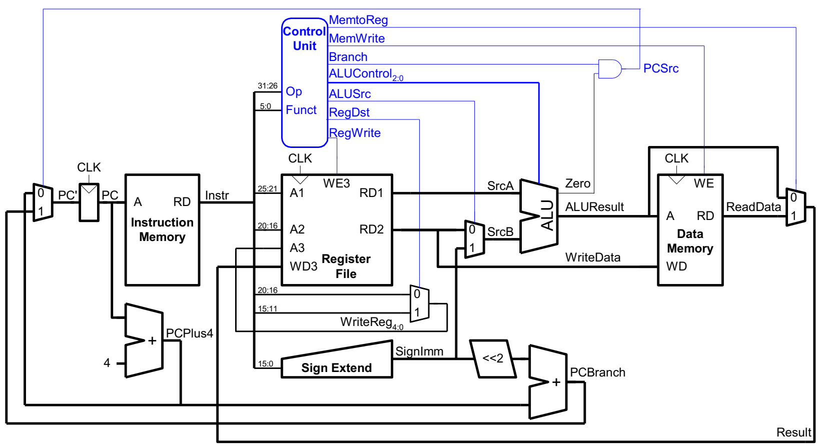
Fetch-decode-execute cycle:

Fetch	Fetch the next machine instruction from memory. Its address is stored in the program counter PC.
Decode	Analyze the instruction and load the necessary data; increment the PC.
Execute	Execute the instruction and store the result.

Structure of a simple processor (here: Harvard architecture)



Roadmap: Computer architecture



1. Combinatorial circuits: Boolean Algebra/Functions/Expressions/Synthesis
2. Number representations
3. Arithmetic Circuits: Addition, Multiplication, Division, ALU

4. Sequential circuits: Flip-Flops, Registers, SRAM, Moore and Mealy automata
5. Verilog

6. Instruction Set Architecture
7. Microarchitecture

8. Performance: RISC vs. CISC, Pipelining, Memory Hierarchy