CO20-320241

Computer Architecture and Programming Languages

CAPL

Lecture 1 & 2

Dr. Kinga Lipskoch

Fall 2019

Who am I?

- PhD in Computer Science at the Carl von Ossietzky University of Oldenburg
- University lecturer at the Department of Computer Science and Electrical Engineering
- Joined Jacobs University in January 2013
- Office: Research I, Room 94
- Telephone: +49 421 200-3148
- E-Mail: k.lipskoch@jacobs-university.de
- Office hours: Mondays 10:00 - 12:00

Course Resources

- Grader:
  https://grader.eecs.jacobs-university.de/courses/320241/2019_2/
- Slides and homework sheets will be posted there
- Offline questions: Office hours on Mondays 10:00 - 12:00
- You can make other appointments if necessary

Literature

- Ronald J. Tocci, Neal Widmer, Greg Moss: Digital Systems: Principles and Applications, 12th edition, Pearson Publishing, 2017
- David A. Patterson and John L. Hennessy: Computer Organization and Design, 5th edition, Morgan Kaufmann Publishers, 2013
- John Hennessy, David Patterson: Computer Architecture: A Quantitative Approach, 6th edition, Morgan Kaufmann Publishers, 2017
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman: Compilers: Principles, Techniques, and Tools, 2nd edition, Pearson Education Publishing, 2007
- Allen Holub: Compiler Design in C, 2nd edition, Prentice Hall, 1990

Course Goals

Understand the basic concepts of

- computer architectures
- data representations and logic circuits
- instruction set architectures
- datapath and control
- programming languages characteristics
- phases of compilation

Course Grading

- 30% - Homework
- 30% - Midterm exam
- 40% - Final exam

Homework

- To be solved individually
  - Discussion between students is encouraged, cheating is not
  - Each student needs to submit her/his own solution to the homework problems
- Homework will be graded by the TAs
  - Percentages
  - Template for submitting homework: https://grader.eecs.jacobs-university.de/courses/320241/2019_2/lectures/template_hw.tex
  - Update parts highlighted with blue
  - Submit one single PDF file
- Submit to Grader https://grader.eecs.jacobs-university.de/

Grader not Publicly Visible

- You can access Grader from campus without any additional connection or software
- To access Grader from outside of campus you need to use a VPN (Virtual Private Network) connection
- Tutorials from the Jacobs IRC IT team on how to install a VPN client: https://teamwork.jacobs-university.de/display/ircit/VPN+Access

Grading Criteria for Homework

- 10% deduction
  if not using the provided template for submitting the homework
  https://grader.eecs.jacobs-university.de/courses/320241/2019_2/lectures/template_hw.tex
- 50% – 70% deduction
  if providing only final result without showing the steps leading to the result
- Short tutorial for installing and working with LaTeX
  https://www.latex-tutorial.com
- The homework will be graded by the TAs
- If issues or questions then please first address to the TAs
- Weekly tutorials on Sundays in West Hall 4 at 17:00

Missing Homework, Midterm, Exams according to AP

- https://www.jacobs-university.de/sites/default/files/bachelor_policies_v2.1.pdf (page 9)
- Illness must be documented with a sick certificate
- Sick certificates and documentation for personal emergencies must be submitted to the Student Records Office by the third calendar day
- Predated or backdated sick certificates will be accepted only when the visit to the physician precedes or follows the period of illness by no more than one calendar day
- Students must inform the Instructor of Record before the beginning of the examination or class/lab session that they will not be able to attend
- The day after the excuse ends, students must contact the Instructor of Record in order to clarify the make-up procedure
- Make-up examinations have to be taken and incomplete coursework has to be submitted by no later than the deadline for submitting incomplete coursework as published in the Academic Calendar

Structure of the Lectures

- Thursday 17:15 - 18:30 Lecture Lecture Hall, Research I
- Friday 11:15 - 12:30 Lecture Lecture Hall, Research I
- Homework is due one week after releasing on Monday at 23:00 sharp

Planned Syllabus

- Part I: Data representations and logic circuits
- Part II: Instruction set architectures
- Part III: Datapath and control
- Part IV: Programming languages and compilation

Part I: Data Representations and Logic Circuits

Computers Everywhere

- How many different computers have you already used today?
  - PC, Monitor
  - Smartphone
  - Washing machine
  - Digital camera
  - MP3 player
  - Airplane, car
- All of these computers use software, often especially written for specific purpose and architecture

Computer Architecture (1)

- Refers to overall organization of computer system
- Specifies
  - functionality of major components
  - interconnection among components
- You will learn architectural concepts, but in the beginning we will also look at some low-level technical details
- Later we will abstract away from details

Computer Architecture (2)

- Software
  - Variables in different formats: characters, integers, floating point numbers, pointers, etc.
  - Instructions: data transfer, arithmetic, comparison, control flow
- Hardware
  - Registers, register file, memory, disks, etc.
  - Adders, multipliers, divider, etc.
- Register-transfer level: machine instructions
- Logic level: AND, OR, etc.
- Transistor level: electronic components implementing logical operations
- Technology
  - CMOS (complementary metal-oxide-semiconductor), bipolar CMOS, etc.
  - Germanium, silicon, gallium, arsenic

Questions

- Instruction set
  - What does the machine understand?
  - What does the machine need to understand?
  - What is the interface between hardware and software?
- Machine organization
  - How does it work?
  - How do we design a computer?
  - How does the speed depend on the architecture?

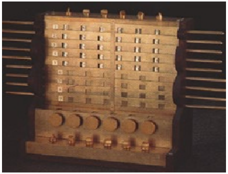
Brief History of Computers

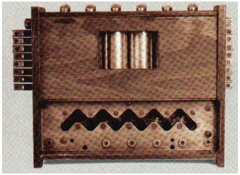
- First phase: Mechanical machines
- Second phase: Electronic machines
  - Relays
  - Tubes
  - Transistors
- Third phase: Digital computers
- Fourth phase: Networking

Wilhelm Schickard

- Lived in Tübingen
  1592 – 1635
- Built first automatic calculator, the "Calculating Clock"

First Automatic Calculator



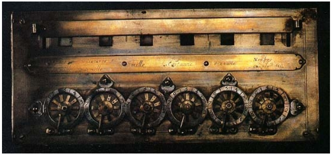


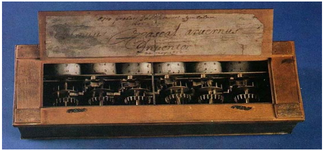
- Calculating Clock, 1623
- Machine could add, subtract, multiply and divide six-digit numbers, and indicated an overflow of this capacity by ringing a bell
- Destroyed in a fire while still incomplete

Blaise Pascal

- Born in Clermont-Ferrand, June 19th, 1623
- Died 1662 in Paris
- Famous mathematician, built Pascal's calculator (Pascaline), which could add and subtract

Pascaline



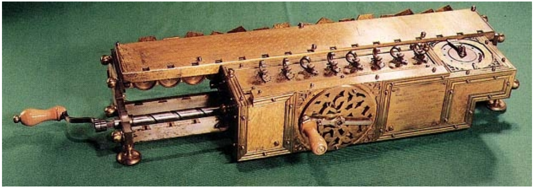


- First mechanical adding machine, around 1642
- Machine was used until 1799
- Had about 50 prototypes and later built around 20 more machines

Gottfried Wilhelm, Freiherr von Leibnitz (1)

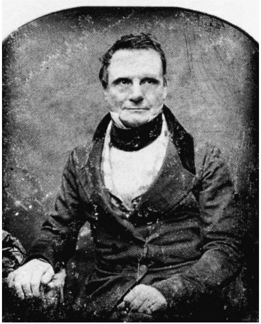
- Mathematician and Philosopher, born in 1646 in Leipzig
- Died in Hannover, November 14^{th}, 1716
- Invented the binary number system, which is basis of all modern computers

Gottfried Wilhelm, Freiherr von Leibnitz (2)



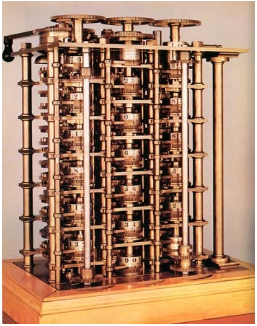
- First computing machine, 1668
- Could add, subtract, multiply and divide

Charles Babbage



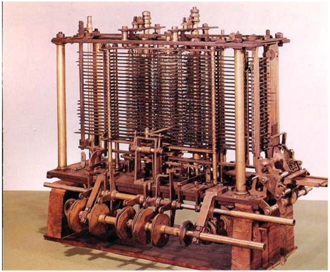
- Born December 26th, 1791 in London
- Died in 1871 in London
- In that time numerical tables were calculated by men called computers
- Originator of the idea of a programmable machine

Difference Engine



- Completed in 1822
- Could add and subtract
- Compute tables for naval navigation
- Output = punched results
- Could run only one algorithm

Analytical Engine



- A general-purpose computer designed by Babbage
- Analytical Engine has never been completed
- Was planned to be programmed using punch cards

Programmable Machine

- Joseph Marie Jacquard's loom called "Jacquard loom", invented in 1801
- Uses punch-cards to control the weaving of patterns (e.g., silk)
- Did not operate well, therefore unsuccessful

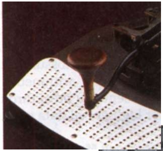
Augusta Ada Byron

- Born in 1815 in London
- Died in 1852 in London
- In a public article Ada describes an algorithm for the analytical engine to compute Bernoulli numbers
- The first programmer
- The computer language Ada was named after her

Herman Hollerith

- Born in 1860 in New York
- Died in 1929 in Washington, D.C.
- Developed a mechanical tabulator and paper punch card, 1884, Massachusetts Institute of Technology
- Application in summarizing information, later accounting
- Founder of the Tabulating Machine Company (1896) later merged to become IBM

Storage Medium: Punchcards

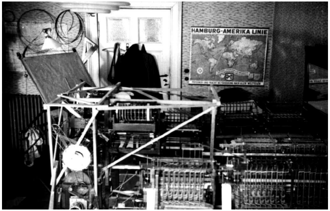


- Early punch cards
- Punch card workers (probably at the census in 1890) knowing every single bit of the storage medium

Konrad Zuse (1)

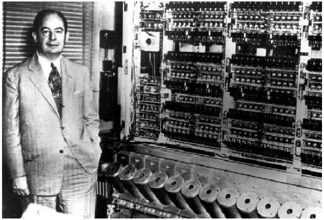
- Born in Berlin, June 22nd, 1910
- Died in Hünfeld, December 18th, 1995

Konrad Zuse (2)



- The Z1 computer in the living room of Konrad Zuse's parents in 1936
- Z1 - mechanical programmable digital computer (1938)
- Z2 - fully functional computer (1940)
- Z3 - first program control using a binary system
- Plankalkül, probably the first programming language
- Founder of Zuse KG, taken over by Siemens

John Louis von Neumann



- John von Neumann, next to the ENIAC (Electronic Numerical Integrator And Computer)
- First memory programmable computer: EDVAC (Electronic Discrete Variable Automatic Computer) (1944)
- Gave name to the concept of von-Neumann architecture of modern computers
- Single storage structure to hold both instructions and data
- Storage is separated from the processing unit

Electronic Numerical Integrator and Computer (ENIAC)

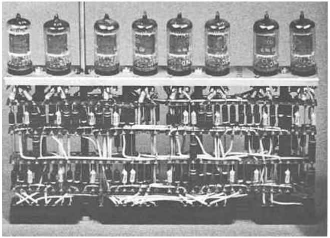
- 18000 vacuum tubes
- 1500 relays
- 30 tons weight
- 140 kW power usage
- 20 registers (10-digit decimal)

- 1946, School of Electrical Engineering at the University of Pennsylvania
- Used vacuum tubes (not mechanical devices) to do its calculations
- First electronic computer
- But it could not store its programs (its set of instructions)

Brief History of Computer Technology

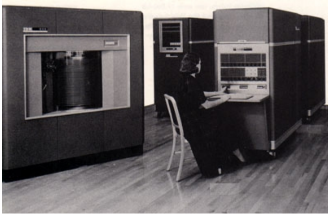
- 1951 – 1958 1st generation: Vacuum tubes
- 1959 – 1963 2nd generation: Transistors
- 1964 – 1979 3rd generation: ICs
- 1979 – 4th generation: VLSI (Very Large Scale Integration)
- Now – ULSI (Ultra Large Scale Integration)
  - Frequency increases, but the slope is not as steep as it used to be
  - Multi Cores

First Generation: Tubes (1951 – 1958)



- Vacuum tubes as their main logic elements
- Punch cards to input and externally store data
- Rotating magnetic drums for internal storage of data and programs
- Programs written in
  - Machine language
  - Assembly language - compiler needed

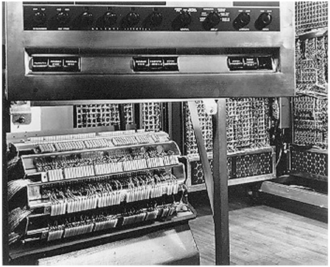
IBM 650 (1)



Best selling computer of the 1950's, with about 2000 units installed before it was discontinued in the early 1960's

The CPU was 5ft by 3ft by 6ft and weighed 1966 lbs, and rented for $3200 per month. The power unit was 5x3x6 ft and weighed 2972 pounds. The card reader/punch weighed 1295 pounds and rented for $550/month. The 650 could add or subtract in 1.63 milliseconds, multiply in 12.96 ms, and divide in 16.90 ms.

IBM 650 (2)



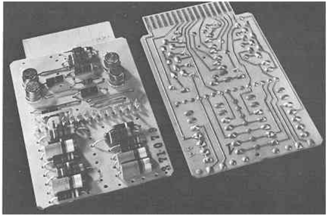
The memory was a rotating magnetic drum with 2000 word (10 digits and sign) capacity and random access time of 2.496 ms. For an additional $\$1,500/\text{month}$ you could add magnetic core memory of 60 words with access time of 0.096 ms.

Programming the 650

- FORTRAN – the first high-level, machine-independent programming language – marked a great leap forward in user friendliness, and was probably available for the 650 by this time
- First you punched your FORTRAN program on a key punch machine, along with any data and control cards. But since the 650 had no disk, the FORTRAN compiler was not resident.
- To compile your program, you fed the FORTRAN compiler deck into the card reader, followed by your FORTRAN source program as data. After some time, the machine would punch the resulting object deck.
- Your program would run and results would be punched onto yet another deck of cards.
- To see the results, you would feed the result deck into another machine, such as an IBM 407, to have it printed on paper (if the computer itself had no printer, as the original 650 did not).

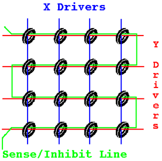
Source: Columbia University Computing History, 1958

Second Generation: Transistors (1959 – 1963)



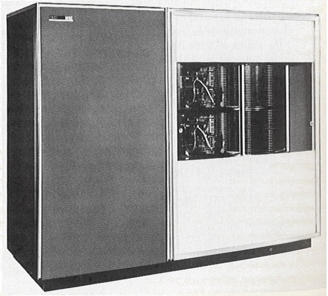
- Transistors replace tubes as main logic element
- Magnetic tape and disks begin to replace punched cards as external storage devices
- Magnetic cores strung on wire within the computer become the primary internal storage technology
- High-level programming languages
  - FORTRAN and COBOL

Magnetic Core



The unit was available with one capacity of 4,096 36-bit words. When the 737 replaced the IBM 706, the system's 701 Electronic Analytical Control Unit had to be altered. Customers could rent a 737 for a fee of $6,100 a month.

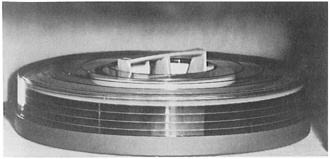
Disk Storage Unit 1961



- The IBM 1301 Disk Storage Unit (DSU) containing two storage modules in 1961
- Module capacity in the 7000 series computer context was about 28 million characters

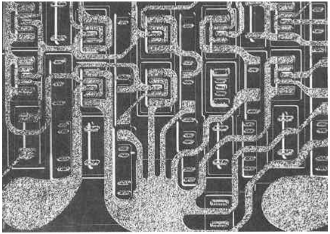
Hard Disk Storage

- An IBM 1316 disk pack, 1962
- Weight: about 10 pounds $\approx$ 4.5 kg
- Storage capacity: 2MB



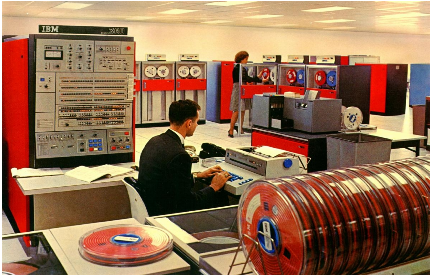
Source: Columbia University Computing History

Third Generation: ICs (1964 – 1979)



- Integrated circuits (ICs) replace individual transistors
- Magnetic tape and disks completely replace punch cards as external storage devices
- Metal oxide semiconductor (MOS) memory, which, like integrated circuits, use silicon-backed chips
- Operating systems
- Advanced programming
  - BASIC
  - Microsoft founded in 1975

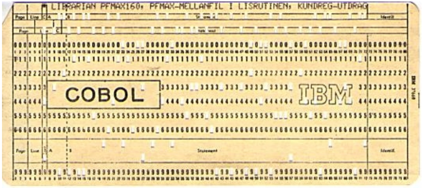
IBM System/360



- Model 40 in the figure
- 2.0 MHz
- 128 – 256Kb memory
- $> \$550,000$

- A computer family, rather than one specific model
- Commercially very successful (upgrade possibility)

IBM Punchcard



- Contained either commands for controlling automated machinery or data for data processing applications
- Both commands and data were represented by the presence or absence of holes in predefined positions

DEC PDP-8

- First commercial minicomputer
- Less than $\$20,000$
- Single bus connects several components of a computer
- A bus is collection of parallel wires

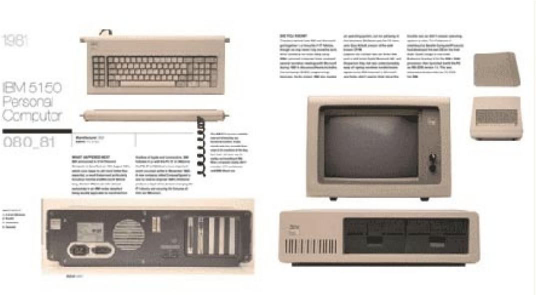
Fourth Generation: (1979−present)

- Large-scale and very large-scale integrated circuits (LSICs and VLSICs)
- Microprocessors that contained memory, logic, and control circuits (an entire CPU = Central Processing Unit) on a single chip
- Personal Computers
  - Apple II
  - IBM PC
- Graphical User Interfaces (GUI) for PCs arrive in the early 1980's

Apple II

- Apple II released to public in 1977, by Stephen Wozniak and Steven Jobs
- Price $\$1,298$
- First Apple Mac released in 1984

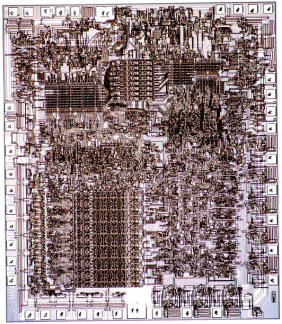
IBM Personal Computer



IBM PC introduced in 1981

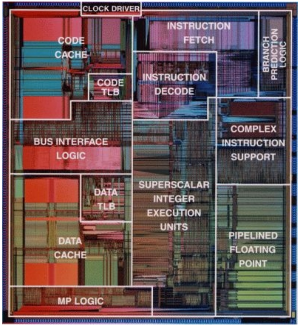
Pricing started at $1,565

Intel 8080



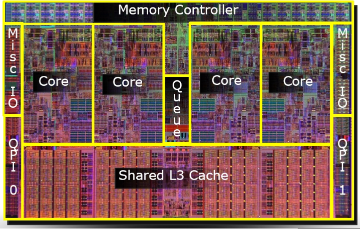
- Introduced in 1974
- Intel 4004 was the first IC, but Intel 8080 was the first commercially successful IC

Pentium I



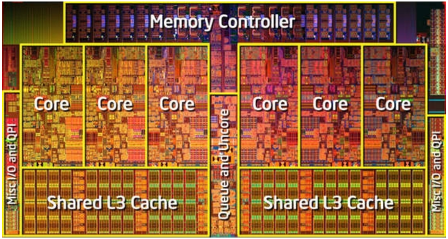
- Introduced in 1993
- The Pentium is a fifth-generation *x*86 architecture microprocessor by Intel
- Started with 60 MHz
- 800-nm process = half the distance between identical features of a memory cell

Nehalem



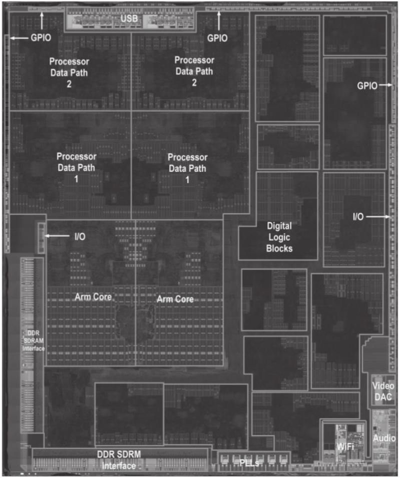
- Introduced in November 2008
- $731 * 10^6$ transistors
- 45-nm process

Gulftown



- Introduced in March 2010
- Smaller than Nehalem (32-nm process)
- $1.17 * 10^9$ transistors
- 6 cores, 12 MB shared L3 cache

A5 of Apple iPad 2 A1395

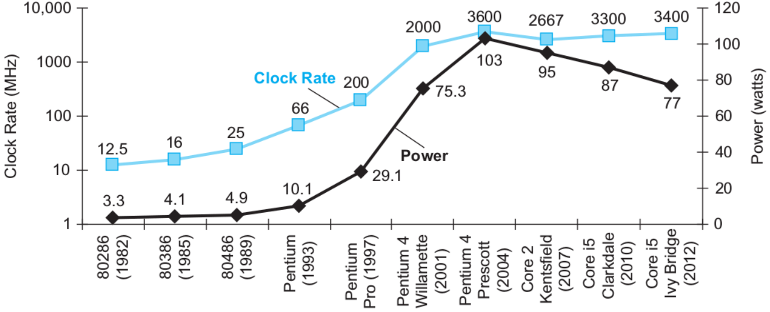


- Size of chip is 12.1 by 10.1 mm, and it was manufactured originally in a 45-nm process
- Two identical ARM processors or cores in the middle left of the chip and a PowerVR GPU with four datapaths in the upper left quadrant
- To the left and bottom side of the cores are interfaces to main memory (DRAM)

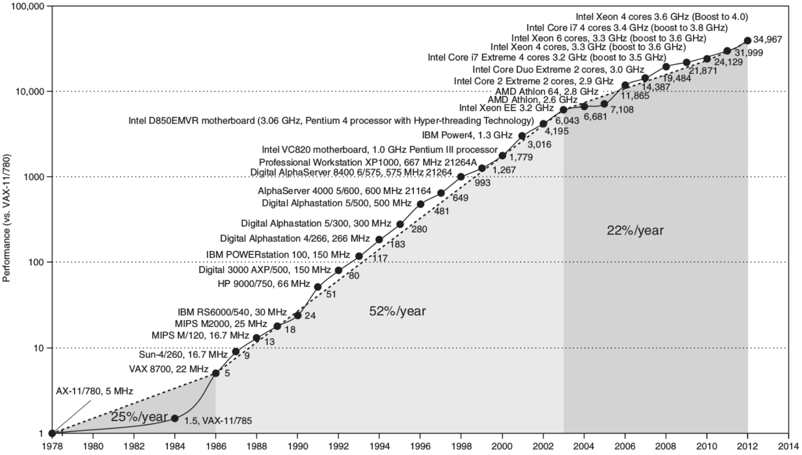
Moore's Law

- In 1965, Gordon Moore predicted that the number of transistors that can be integrated on a die would double every 18 to 24 months (i.e., grow exponentially with time)
- Amazingly visionary the million transistor/chip barrier was crossed in the 1980's
  - 2300 transistors, 1 MHz clock (Intel 4004) – 1971
  - 42 million transistors, 2 GHz clock (Intel Xeon) – 2000
  - 221 million transistors (Itanium II) 130-nm, 374 mm^2 – 2003
  - 592 million transistors (Itanium II) 9 MB Cache – 2004
  - 1.4 billion transistors (Ivy Bridge) – 2012
  - 1.75 billion transistors (Skylake), 14-nm, 122 mm^2 – 2015
  - 3 billion transistors (Snapdragon 835 by Qualcomm), 10-nm, 35% smaller than 14-nm series – 2017
  - 8.5 billion transistor (Snapdragon 8cx/SCX8180), 7-nm – 2018
  - 2019 Samsung and TSMC are producing 5-nm nodes and announced plans to produce 3-nm nodes

Clock Rate and Power for Intel $x86$ Processors



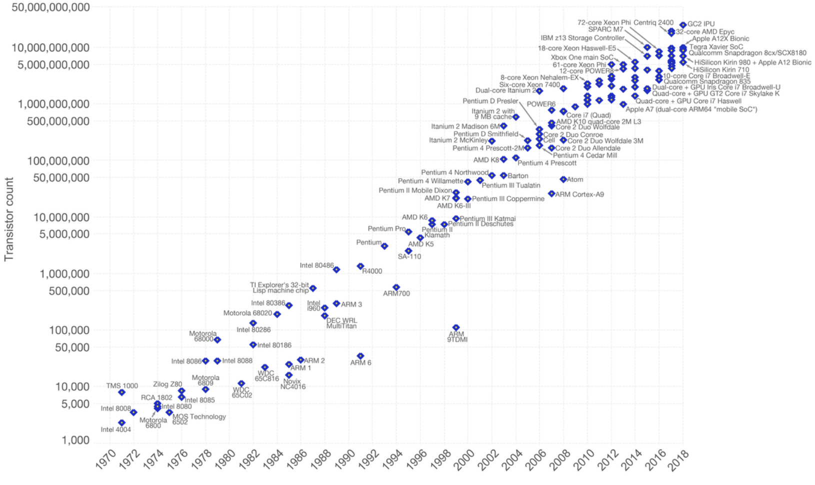
Processor Performance Increase



Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

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 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.

Numerical Representations (1)

- Physical systems use quantities which must be manipulated arithmetically
- Quantities may be represented numerically in either analog or digital form

Numerical Representations (2)

Analog Representation

- A continuously variable, proportional indicator
- Examples of analog representation:
  - Sound through a microphone causes voltage changes
  - Automobile speedometer changes with speed
  - Mercury thermometer varies over a range of values with temperature

Numerical Representations (3)

Digital Representation

- Varies in discrete (separate) steps
- Examples of digital representation:
  - Passing time is shown as a change in the display on a digital clock at one minute intervals
  - A change in temperature is shown on a digital display only when the temperature changes at least one degree

Digital and Analog Systems (1)

- Analog system
  - A combination of devices that manipulate values represented in analog form
- Digital system
  - A combination of devices that manipulate values represented in digital form

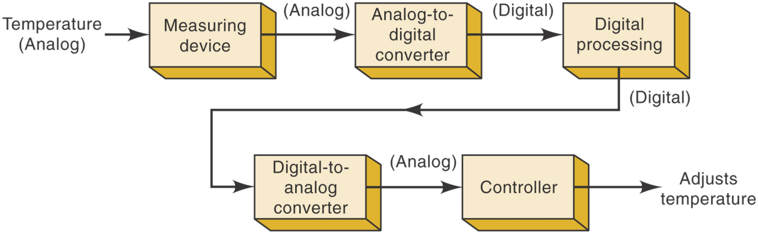
Digital and Analog Systems (2)

Advantages of digital systems:

- Ease of design
- Well suited for storing information
- Accuracy and precision are easier to maintain
- Programmable operation
- Less affected by noise (= random fluctuation in an electrical signal)
- Ease of fabrication on IC chips

Digital and Analog Systems (3)

Analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) are complicate circuitry



Digital and Analog Systems (4)

The audio CD is a typical hybrid (combination) system:

- Analog sound is converted into analog voltage
- Analog voltage is changed into digital through an ADC in the recorder
- Digital information is stored on the CD
- At playback the digital information is changed into analog by a DAC in the CD player
- The analog voltage is amplified and used to drive a speaker that produces the original analog sound

Digital and Analog Systems (5)

- There have been remarkable recent advances in digital technology
- Advances will continue as digital technology expands and improves
- We will look at concepts and tools that will prepare you to work with digital systems

Digital Number Systems (1)

Understanding digital systems requires an understanding of the:

- Decimal,
- Binary,
- Octal, and
- Hexadecimal number systems

Digital Number Systems (2)

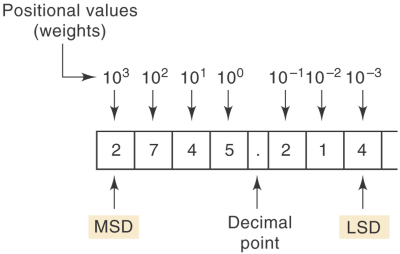
Number systems differ in the amount of symbols they use

- Decimal – 10 symbols (base 10)
- Hexadecimal – 16 symbols (base 16)
- Octal – 8 symbols (base 8)
- Binary – 2 symbols (base 2)

Digital Number Systems (3)

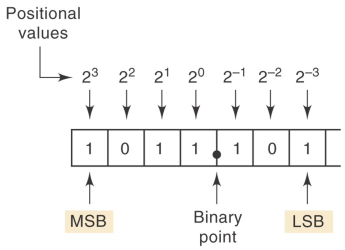
The decimal (base 10) system:

- 10 symbols: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
- Each number is a digit (from Latin for finger)
- Most significant digit (MSD) and least significant digit (LSD)
- Positional value may be stated as a digit multiplied by a power of 10



Binary System

- Lends itself to electronic circuit design since only two different voltage levels are required
- Binary quantities can be converted to other number systems
- Positional value may be stated as a digit multiplied by a power of 2



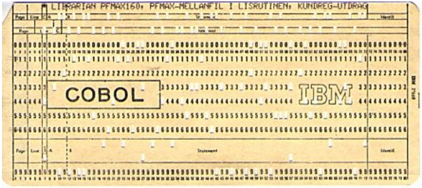
Binary Counting

Weights

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	Decimal equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

LSB

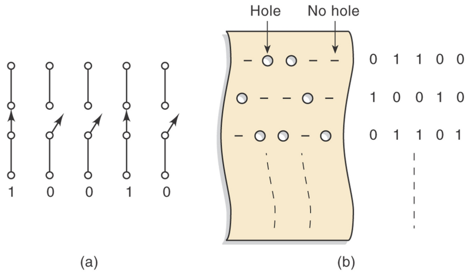
IBM Punchcard



- Contained either commands for controlling automated machinery or data for data processing applications
- Both commands and data were represented by the presence or absence of holes in predefined positions

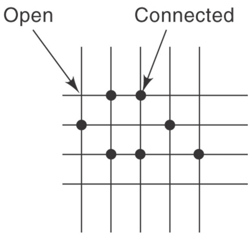
Representing Binary Quantities (1)

- Open and closed switches
- Paper Tape



Representing Binary Quantities (2)

- Wires and rows form a matrix
- This forms the foundation for programmable logic devices that will be studied in depth later



Representing Binary Quantities (3)

Other two state devices:

- Light bulb (off or on)
- Diode (conducting or not conducting)
- Relay (energized or not energized)
- Transistor (cutoff or saturation)
- Photocell (illuminated or dark)

Digital Circuits/Logic Circuits

- Digital circuits – produce and respond to predefined voltage ranges
- Logic circuits – used interchangeably with the term, digital circuits
- Digital integrated circuits (ICs) – provide logic operations in a small reliable package

Parallel and Serial Transmission

- Parallel transmission – all bits in a binary number are transmitted simultaneously. A separate line is required for each bit
- Serial transmission – each bit in a binary number is transmitted per some time interval
- Both methods have useful applications which will be seen in later chapters

Memory

- A circuit which retains a response to a momentary input is displaying memory
- Memory is important because it provides a way to store binary numbers temporarily or permanently
- Memory elements include:
  - Magnetic
  - Optical
  - Electronic latching circuits

Digital Computers (1)

- Computer – a system of hardware that performs arithmetic operations, manipulates data (usually in binary form), and makes decisions
- Computers perform operations based on instructions in the form of a program at high speed and with a high degree of accuracy

Digital Computers (2)

Major parts of a computer:

- Input unit – processes instructions and data into the memory
- Memory unit – stores data and instructions
- Control unit – interprets instructions and sends appropriate signals to other units as instructed
- Arithmetic/logic unit – arithmetic calculations and logical decisions are performed
- Output unit – presents information from the memory to the operator or process
- The control and arithmetic/logic units are often treated as one and called the central processing unit (CPU)

Five Classic Components of a Computer