

Prof. Dr. Florian Künzner

Technical University of Applied Sciences Rosenheim, Computer Science

Herelich Willkommen.

Start: 8:01

> Bitte: Vorname + Nachmanne > Sichständig umbenenne in Technelinevliste

CA 1 – Intro

Slide 1 von 35

CAMPUS RosenheimComputer Science



Question

What is the second most important tool of a computer scientist?

CAMPUS Rosenheim Computer Science



Question

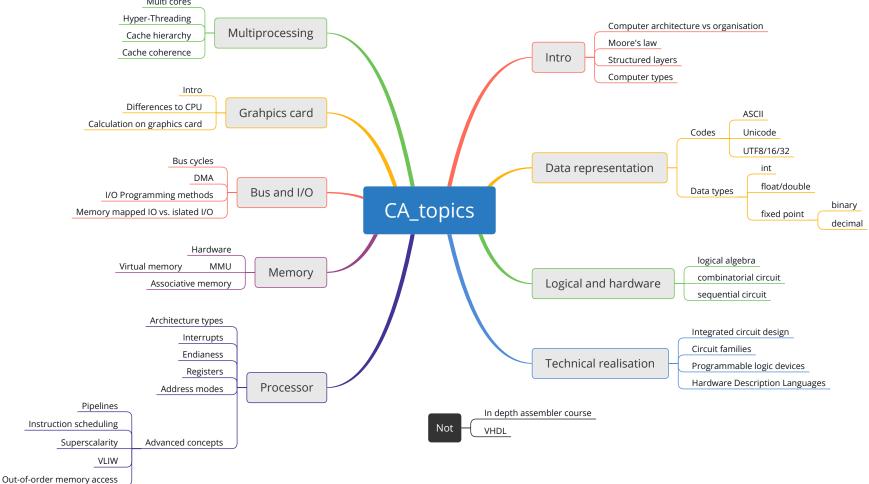
What is the most important tool of a computer scientist?

CAMPUS Rosenheim

Computer Science



Overview



CAMPUS Rosenheim **Computer Science**



Goal

CA::Intro

- Motivation: Know why it is worth learning CA
- Computer architecture vs organisation
- Moore's law
- Structured layers
- Computer types
- Analogue vs digital

CAMPUS Rosenheim

Computer Science



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming
- High performance computing programming
- Do understand computers now and in 5 (...) years

CAMPUS Rosenheim

Computer Science



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming
- High performance computing programming
- Do understand computers now and in 5 (...) years

CAMPUS Rosenheim

Computer Science



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming
- High performance computing programming
- Do understand computers now and in 5 (...) years

CAMPUS Rosenheim **Computer Science**



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)

CAMPUS Rosenheim **Computer Science**



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)

CAMPUS Rosenheim **Computer Science**



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster

CAMPUS Rosenheim **Computer Science**



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming

Analogue vs digital Summary Literature Moore's law Structured layers Assembler Computer types

CAMPUS Rosenheim **Computer Science**



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming

CAMPUS Rosenheim Computer Science



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming
- High performance computing programming
- Do understand computers now and in 5 (...) years

Analogue vs digital Summary Literature Moore's law Structured layers Assembler Computer types

CAMPUS Rosenheim

Computer Science



Motivation: Why should you learn it?

- You should know the second most important tool
- Able to buy/specify hardware
- Optimise for hardware (hardware instructions)
- Write better software (algorithms)
- Find bugs or bottlenecks faster
- Embedded systems design and programming
- Real-time systems design and programming
- High performance computing programming
- Do understand computers now and in 5 (...) years

CAMPUS Rosenheim Computer Science



Material

Material for lecture and exercises:

https://inf-git.fh-rosenheim.de/Lectures/RA_exercises.git

CAMPUS RosenheimComputer Science



Time and date

Event	Day	Time	Room
Lecture	Wednesday	08:00 - 09:30 o'clock	Online
Exercise 1	Thursday	09:45 - 11:15 o'clock	Online
Exercise 2	Thursday	11:45 - 13:15 o'clock	Online
Exercise 3	Thursday	13:45 - 15:15 o'clock	Online

CAMPUS Rosenheim

Computer Science



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim

Computer Science



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim

Computer Science



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim

Computer Science



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim **Computer Science**



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex

CAMPUS Rosenheim **Computer Science**



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly

CAMPUS Rosenheim **Computer Science**



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim Computer Science



Lecture

- Presentation of concepts
- Discussion of concepts
- Mostly an introduction into concepts
- Reality is very complex
- Hardware evolves very quickly
- There is a large variety of different hardware for different purposes

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim

Computer Science



Exercise

Exercise content:

- Theoretical tasks
- C, C++, Java, and assembler coding
- Some microcontroller programming
- Homework may also be necessary!!!

- Updated repository with new exercise sheets.
- You should have a PC (notebook) with a Linux and/or the virtual machine for CA (or virtual machine from OS)
- There are some notebooks to borrow (up to 8)

CAMPUS Rosenheim Computer Science



Question

What are the components of a computer?

CAMPUS Rosenheim Computer Science



What is computer architecture? DEFINITION OF ARCHITECTURE USED BY IBM

In IBM 370 Principles of Operations, the **architecture** of a computer is defined as "**its attributes as seen by the programmer**; that is, the **conceptual structure and functional behavior** as distinct from the **organization** of the **data flow**, the **logical design**, the **physical design**, and the **performance** of any particular implementation.

Several dissimilar machine implementations may conform to a single architecture. When programs running on different machine implementations produce the results that are defined by a single architecture, the implementations are considered to be compatible."

[source: Prasad: IBM Mainframes. McGraw-Hill 1989]

CAMPUS Rosenheim Computer Science



What is computer architecture?

COMPUTER ARCHITECTURE vs COMPUTER ORGANIZATION

Computer **architecture** is a description (definition) of the **attributes** of a computing system as seen by a **machine language programmer** or a **compiler writer**. Writable control stores for modifying microcode during computer operation are not considered available to the normal machine language programmer.

Computer **organization** pertains to the various methods that can be used to **implement a specific computer architecture**.

[source: Hintz/Tabak: Microcontrollers. McGraw-Hill 1992.]

CAMPUS Rosenheim Computer Science



Computer architecture vs organisation

Computer architecture

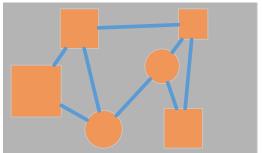
ADD R1,R2,R3 MOV R1, (R1) TAS

logical (interface)

Machine language programmer, compiler writer:

- Conceptual structure
- Functional behaviour

Computer organisation



physical (implementation)

- Data flow
- Logical design
- Physical design
- Performance

CAMPUS Rosenheim Computer Science



In this lecture

We consider both:

- Computer architecture
- Computer organisation

But the focus is more on: Computer architecture.

CAMPUS Rosenheim Computer Science



In this lecture

We consider both:

- Computer architecture
- Computer organisation

But the focus is more on: **Computer architecture**.

CAMPUS Rosenheim **Computer Science**



Questions?

All right? \Rightarrow



Question? \Rightarrow



and use chat

speak after | ask you to

Moore's law Structured layers Assembler Computer types Analogue vs digital Summary Literature

CAMPUS Rosenheim

Computer Science



Literature (1)

Rechnerarchitektur Von der digitalen Logik zum Parallelrechner

Author(s) Andrew S. Tanenbaum, Todd Austin

1. March 2014 Date

Edition 6. edition

Language German

ISBN 978-3868942385

Reference [1]



[source: https://www.pearson-studium.de]

Moore's law Structured layers Assembler Computer types Analogue vs digital Summary Literature

CAMPUS Rosenheim

Computer Science



Literature (2)

Computer Organization and Design RISC-V Edition The Hardware Software Interface

Author(s) David A. Patterson, John L. Hennessy

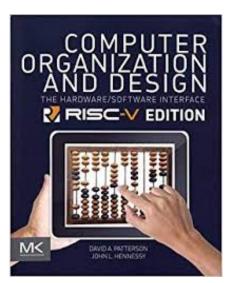
Date 22. May 2017

RISC-V ed Edition

Language English

ISBN 978-0128122754

Reference [2]



[source: https://www.amazon.de]

CAMPUS Rosenheim Computer Science

Technische Hochschule Rosenheim Technical University of Applied Sciences

Literature (3)

Grundlagen der Technischen Informatik

Author(s) Dirk W. Hoffmann

Date 5. September 2016

Edition 5. edition

Language German

ISBN 978-3446448674

Reference [3]



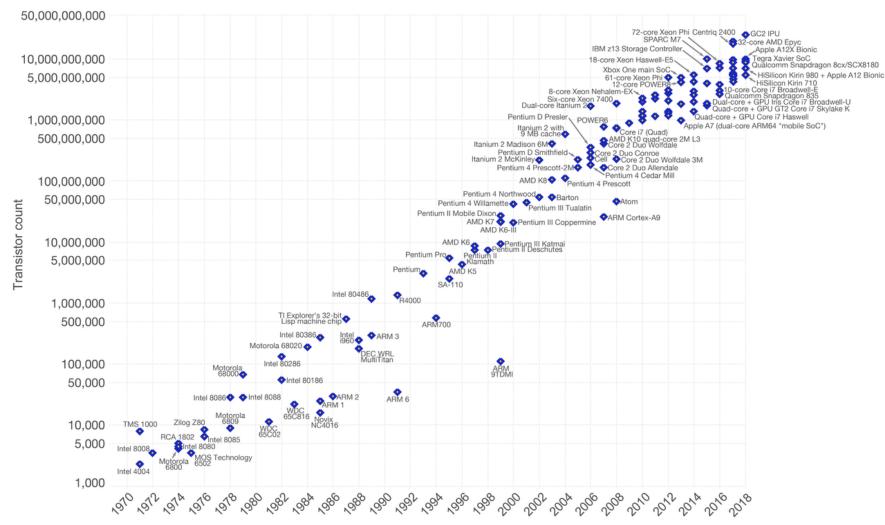
[source: https://www.hanser-fachbuch.de]

Literature

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



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.

Licensed under CC-BY-SA by the author Max Roser.

CAMPUS RosenheimComputer Science

Technische Hochschule Rosenheim Technical University of Applied Sciences

Moore's law

Observation

- Number of transistors in a dense integrated circuit doubles about every two years (18 month)
- Exponential growth rate
- Named after Gordon Moore
- Co-founder of Intel

Gordon Moore



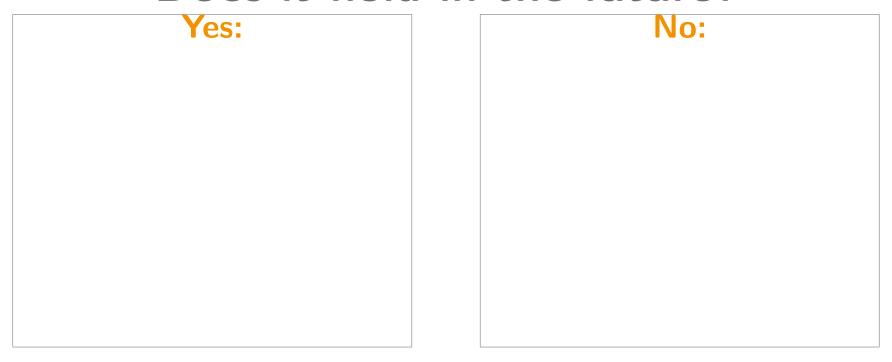
[source: forbes.com]

CAMPUS RosenheimComputer Science

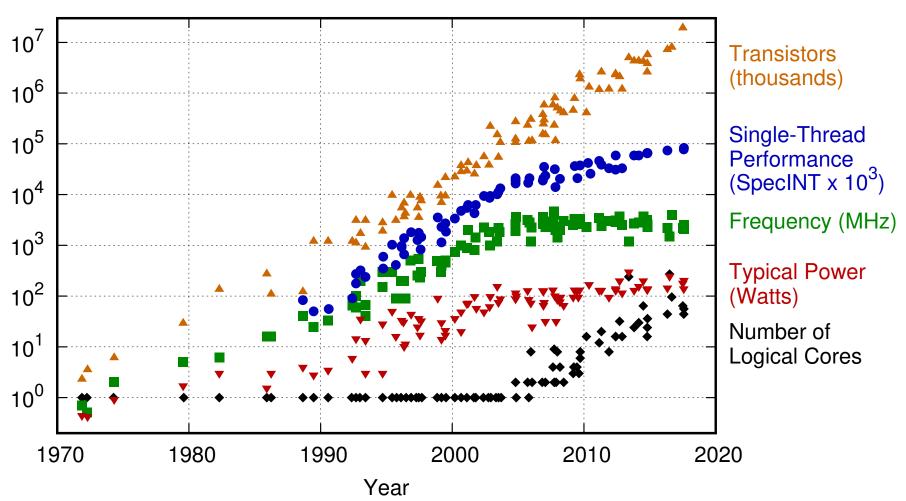


Moore's law

Does it hold in the future?



42 Years of Microprocessor Trend Data



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 2010-2017 by K. Rupp

[source: https://github.com/karlrupp/microprocessor-trend-data]

CAMPUS RosenheimComputer Science



Structured layers

From SOFTWARE to HARDWARE

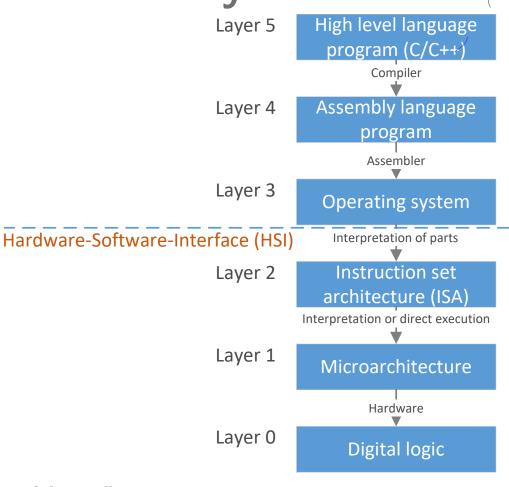
Analogue vs digital Literature Moore's law Structured layers Assembler Computer types Summary

CAMPUS Rosenheim

Computer Science



Structured layers



```
swap(int v[], int k){
 int temp;
 temp = v[k];
 v[k] = v[k+1];
 v[k+1] = temp;
 multi $2, $5,4
      $2, $4,$2
      $15, 0($2)
```

Hardware programming interface

Implementation of the ISA

Gates, integrated circuits

CAMPUS Rosenheim Computer Science



Assembler

- Knowledge from lecture "IT-Systeme" is assumed.
- You don't have to write a lot of assembler code.
- But: You have to interpret it and understand its basic operation mode.

CAMPUS Rosenheim Computer Science



Assembler

- Knowledge from lecture "IT-Systeme" is assumed.
- You don't have to write a lot of assembler code.
- But: You have to interpret it and understand its basic operation mode.

CAMPUS Rosenheim Computer Science



Assembler

- Knowledge from lecture "IT-Systeme" is assumed.
- You don't have to write a lot of assembler code.
- But: You have to interpret it and understand its basic operation mode.

CAMPUS Rosenheim Computer Science



Assembler

- Knowledge from lecture "IT-Systeme" is assumed.
- You don't have to write a lot of assembler code.
- But: You have to interpret it and understand its basic operation mode.

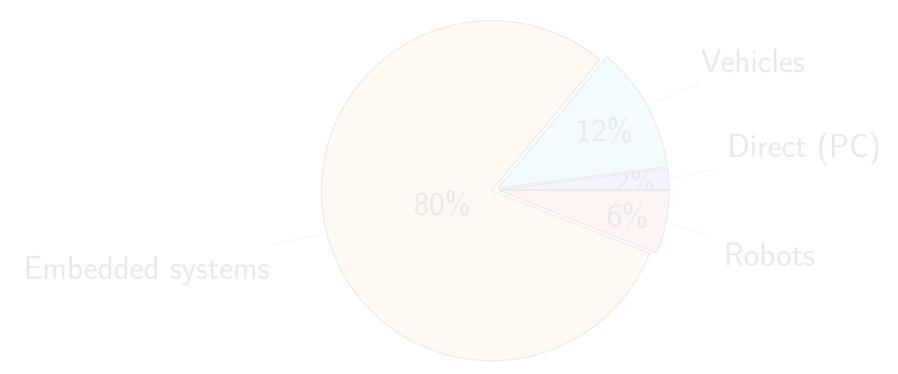
CAMPUS Rosenheim

Computer Science



Computer types

Where are the processors?: Estimated 98% of 8 billion CPUs produced in 2000 used for embedded applications.



[source: DARPA/Intel (Tennenhouse), year 2000]

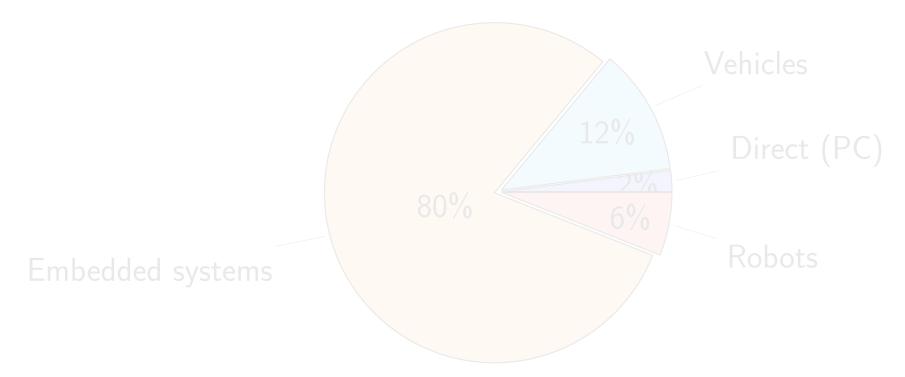
CAMPUS Rosenheim

Computer Science



Computer types

Where are the processors?: Estimated 98% of 8 billion CPUs produced in 2000 used for embedded applications.



[source: DARPA/Intel (Tennenhouse), year 2000]

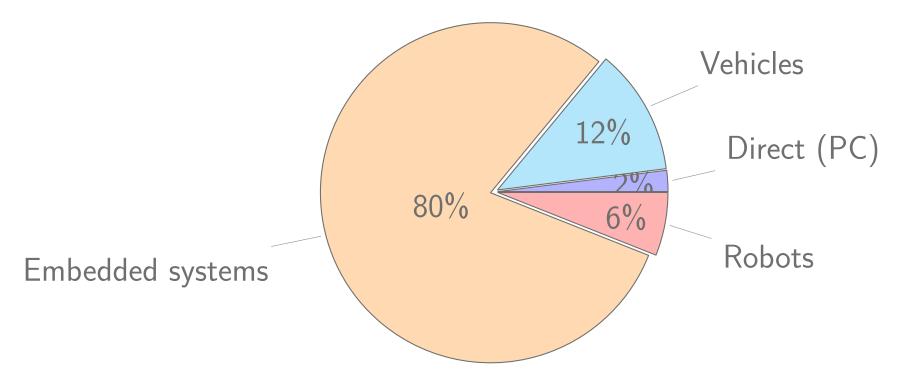
Moore's law Structured layers Assembler Analogue vs digital Summary Literature Computer types

CAMPUS Rosenheim **Computer Science**



Computer types

Where are the processors?: Estimated 98% of 8 billion CPUs produced in 2000 used for embedded applications.



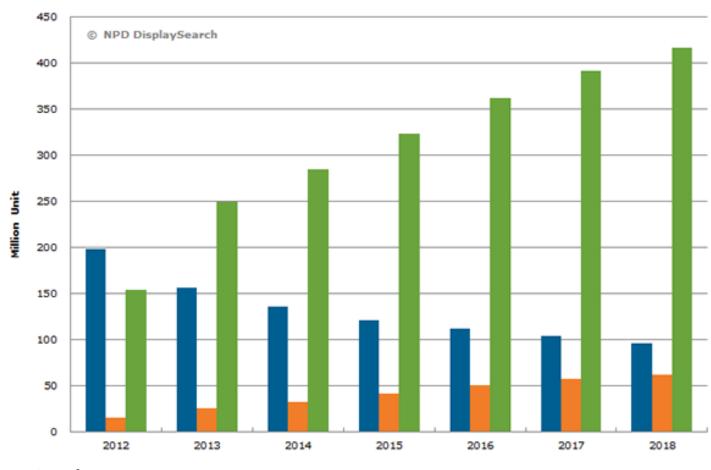
[source: DARPA/Intel (Tennenhouse), year 2000]

CAMPUS Rosenheim

Computer Science



Computer types



[source: thenextweb.com]
Prof. Dr. Florian Künzner, SoSe 2021

■ Standard Notebook PC ■ Ultra-Slim PC ■ Tablet PC

CA 1 — Intro

Slide 28 von 35

CAMPUS Rosenheim

Computer Science



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...
- PCs: workstations, notebooks
- Server: grid of workstations, cloud
- **Mainframes**: (high I/O throughput, e.g. e-commerce or banking transactions)
- Supercomputer: high performance computing systems

CAMPUS Rosenheim

Computer Science



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...

CAMPUS Rosenheim **Computer Science**



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...
- **PCs**: workstations, notebooks

CAMPUS Rosenheim **Computer Science**



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...
- **PCs**: workstations, notebooks
- **Server**: grid of workstations, cloud

CAMPUS Rosenheim Computer Science



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...
- **PCs**: workstations, notebooks
- Server: grid of workstations, cloud
- **Mainframes**: (high I/O throughput, e.g. e-commerce or banking transactions)
- Supercomputer: high performance computing systems

CAMPUS Rosenheim Computer Science



- Microcontroller: embedded systems, smartphones, vehicles, robots, ...
- **PCs**: workstations, notebooks
- **Server**: grid of workstations, cloud
- **Mainframes**: (high I/O throughput, e.g. e-commerce or banking transactions)
- **Supercomputer**: high performance computing systems

Analogue vs digital Summary Organisation Intro Literature Moore's law Structured layers Assembler Computer types

CAMPUS Rosenheim **Computer Science**



Analogue vs digital

General meaning

Analogue \approx Corresponding, similar, analogous, applicable An "analogue" is a similar or corresponding "thing".



CAMPUS Rosenheim Computer Science



Analogue vs digital

General meaning

Analogue \approx Corresponding, similar, analogous, applicable

An "analogue" is a similar or corresponding "thing".

Digital \approx With numbers (lat. digitus = "Finger (for counting)")

Extended meaning in IT (electronics)



CAMPUS Rosenheim

Computer Science



Analogue vs digital

General meaning

Analogue \approx Corresponding, similar, analogous, applicable

An "analogue" is a similar or corresponding "thing".

Digital \approx With numbers (lat. digitus = "Finger (for counting)")

Extended meaning in IT (electronics)



CAMPUS Rosenheim

Computer Science



Analogue vs digital

General meaning

Analogue \approx Corresponding, similar, analogous, applicable

An "analogue" is a similar or corresponding "thing".

Digital \approx With numbers (lat. digitus = "Finger (for counting)")

Extended meaning in IT (electronics)

Analogue \approx continuous, steady, constantly

CAMPUS Rosenheim

Computer Science



Analogue vs digital

General meaning

Analogue \approx Corresponding, similar, analogous, applicable

An "analogue" is a similar or corresponding "thing".

Digital \approx With numbers (lat. digitus = "Finger (for counting)")

Extended meaning in IT (electronics)

Analogue \approx continuous, steady, constantly

Digital \approx stepwise, discrete

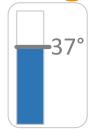
CAMPUS Rosenheim

Computer Science



Analogue vs digital

Analogue



analogue thermometer

Temperature

- Indirectly via a physical analogue
- Height of liquid
- Digitalisation through people (read)

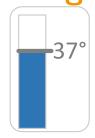
CAMPUS Rosenheim

Computer Science



Analogue vs digital

Analogue



Digital



analogue thermometer

Temperature

- Indirectly via a physical analogue
- Height of liquid
- Digitalisation through people (read)

digital (thermometer)

Temperature

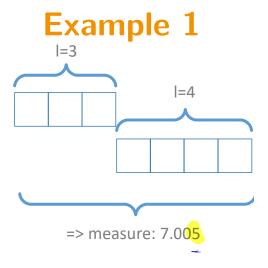
- Numerical display
- Internal: measure of physical analogue (resistor)
- Automatic digitalisation

CAMPUS Rosenheim

Computer Science



Analogue computer



analogue add

CAMPUS Rosenheim

Computer Science



Analogue computer

Example 1

=> measure: 7.005

=> read result: ~ 15.29

analogue add

analogue multiply

CAMPUS Rosenheim

Computer Science



Analogue computer: historical

Calculation by creation of a physical analogue.

Properties

- Very fast operations
- Not very accurate

Applications

- Solution of differential equations
- Real-time simulations

Keywords: Operational amplifier (Operationsverstärker)

Nowadays, analogue computers are not used very often. In the following, **only digital computers** will be discussed.

CAMPUS Rosenheim

Computer Science



Analogue computer: historical

Calculation by creation of a physical analogue.

Properties

- Very fast operations
- Not very accurate

Applications

- Solution of differential equations
- Real-time simulations

Keywords: Operational amplifier (Operationsverstärker)

Nowadays, analogue computers are not used very often. In the following, **only digital computers** will be discussed.

CAMPUS Rosenheim

Computer Science



Analogue computer: historical

Calculation by creation of a physical analogue.

Properties

- Very fast operations
- Not very accurate

Applications

- Solution of differential equations
- Real-time simulations

Keywords: Operational amplifier (Operationsverstärker)

CAMPUS Rosenheim Computer Science



Analogue computer: historical

Calculation by creation of a physical analogue.

Properties

- Very fast operations
- Not very accurate

Applications

- Solution of differential equations
- Real-time simulations

Keywords: Operational amplifier (Operationsverstärker)

Nowadays, analogue computers are not used very often. In the following, **only digital computers** will be discussed.

CAMPUS Rosenheim **Computer Science**



Questions?

All right? \Rightarrow

Question? \Rightarrow

and use chat

speak after | ask you to

CAMPUS Rosenheim

Computer Science



Summary and outlook

Summary

- Computer architecture vs organisation
- Moore's law
- Structured layers
- Computer types
- Analogue vs digital

Outlook

- Data representation
- Unicode and UTF
- Data types

CAMPUS Rosenheim

Computer Science



Summary and outlook

Summary

- Computer architecture vs organisation
- Moore's law
- Structured layers
- Computer types
- Analogue vs digital

Outlook

- Data representation
- Unicode and UTF
- Data types