Technische Hochschule Rosenheim
Technical University of Applied Sciences

Prof. Dr. Florian Künzner

Technical University of Applied Sciences Rosenheim, Computer Science

CA 1 – Intro

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Question

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

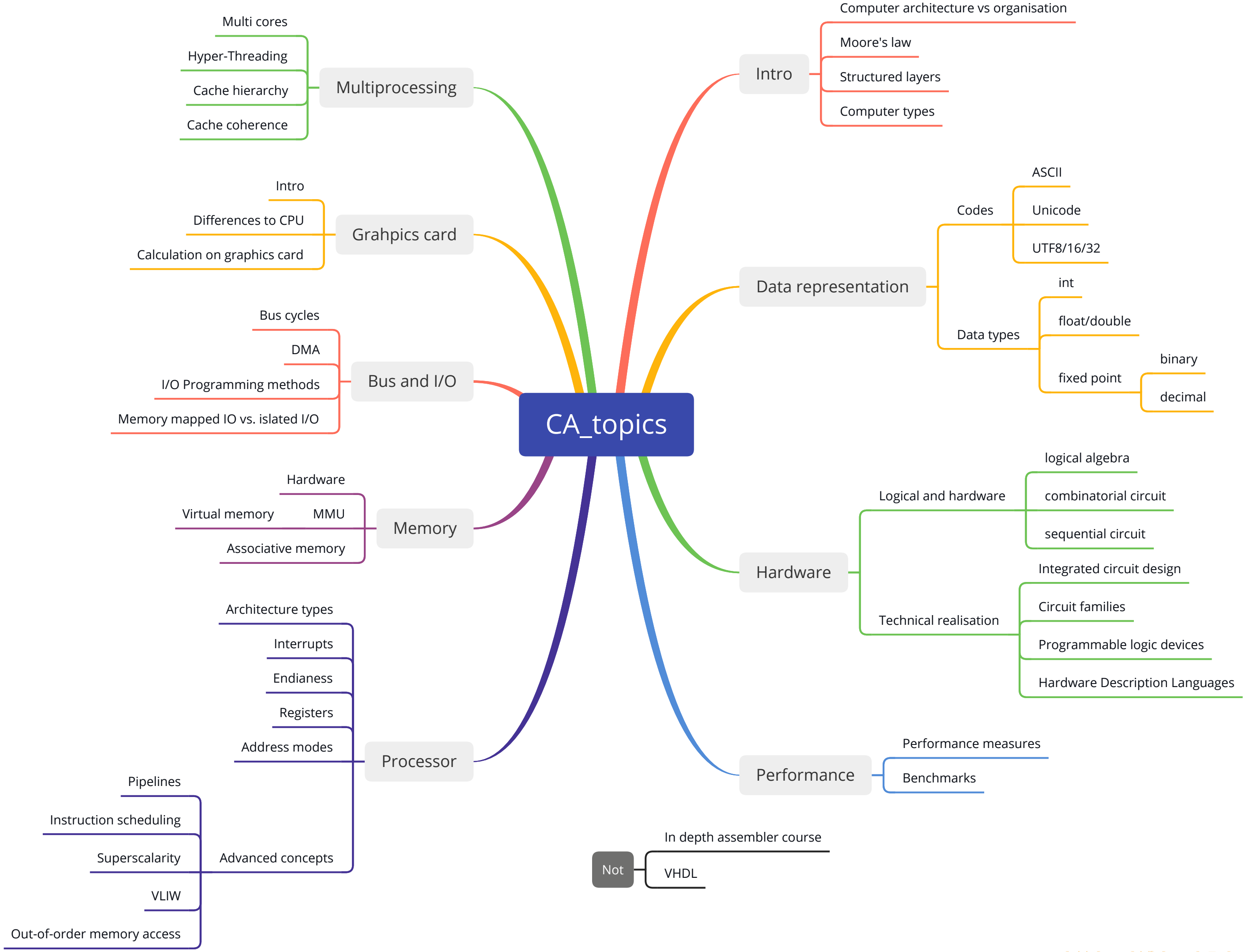
Question

What is the most important tool of a computer scientist?

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal



- CA_topics
  - Intro
    - Computer architecture vs organisation
    - Moore's law
    - Structured layers
    - Computer types
  - Data representation
    - Codes
      - ASCII
      - Unicode
      - UTF8/16/32
    - Data types
      - int
      - float/double
      - fixed point
        - binary
        - decimal
  - Hardware
    - Logical and hardware
      - logical algebra
      - combinatorial circuit
      - sequential circuit
    - Technical realisation
      - Integrated circuit design
      - Circuit families
      - Programmable logic devices
      - Hardware Description Languages
  - Performance
    - Performance measures
    - Benchmarks
  - Multiprocessing
    - Multi cores
    - Hyper-Threading
    - Cache hierarchy
    - Cache coherence
  - Grahpics card
    - Intro
    - Differences to CPU
    - Calculation on graphics card
  - Bus and I/O
    - Bus cycles
    - DMA
    - I/O Programming methods
    - Memory mapped IO vs. islated I/O
  - Memory
    - Hardware
    - MMU
      - Virtual memory
    - Associative memory
  - Processor
    - Architecture types
    - Interrupts
    - Endianess
    - Registers
    - Address modes
    - Advanced concepts
      - Pipelines
      - Instruction scheduling
      - Superscalarity
      - VLIW
      - Out-of-order memory access
- Not
  - In depth assembler course
  - VHDL

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Motivation: Why should you learn it?

Some reasons why it is worth studying:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Material

Material for lecture and exercises:

`https://inf-git.th-rosenheim.de/Lectures/CA_exercises.git`

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Time and date

Event	Day	Time	Room
Lecture	Wednesday	08:00 - 09:30 o'clock	A2.08
Exercise 1	Wednesday	11:45 - 13:15 o'clock	A2.08
Exercise 2	Wednesday	13:45 - 15:15 o'clock	A2.08
Exercise 3	Wednesday	15:30 - 17:00 o'clock	A2.08

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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

Best learning experience: discuss with me!

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

Best learning experience: discuss with me!

Exercise

Exercise content:

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

Equipment:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise

Exercise content:

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

Equipment:

- 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)

Exercise

Exercise content:

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

Equipment:

- 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)

Exercise

Exercise content:

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

Equipment:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise

Exercise content:

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

Equipment:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise

Exercise content:

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

Equipment:

- 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)

Exercise

Exercise content:

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

Equipment:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise

Exercise content:

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

Equipment:

- 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)

Exercise

Exercise content:

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

Equipment:

- 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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise

Exercise content:

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

Equipment:

- 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?

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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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

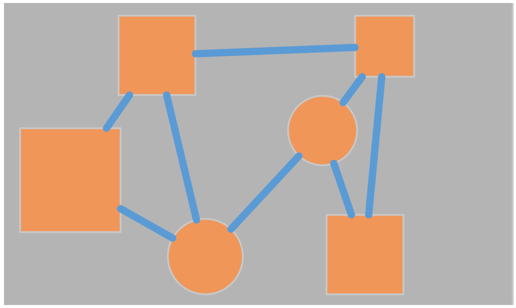
```
ADD R1,R2,R3
MOV R1, (R1)
TAS
```

logical (interface)

Machine language programmer, compiler writer:

- Conceptual structure
- Functional behaviour

Computer organisation



physical (implementation)

- Data flow
- Logical design
- Physical design
- Performance

In this lecture

We consider both:

- Computer architecture
- Computer organisation

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

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

In this lecture

We consider both:

- Computer architecture
- Computer organisation

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

CAMPUS Rosenheim
Computer Science

Literature (1)

Rechnerarchitektur
Von der digitalen Logik zum Parallelrechner

Author(s)	Andrew S. Tanenbaum, Todd Austin
Date	1. March 2014
Edition	6. edition
Language	German
ISBN	978-3868942385
Reference	[1]

Rechnerarchitektur
Von der digitalen Logik zum Parallelrechner
6., aktualisierte Auflage
Andrew S. Tanenbaum
Todd Austin
Pearson

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

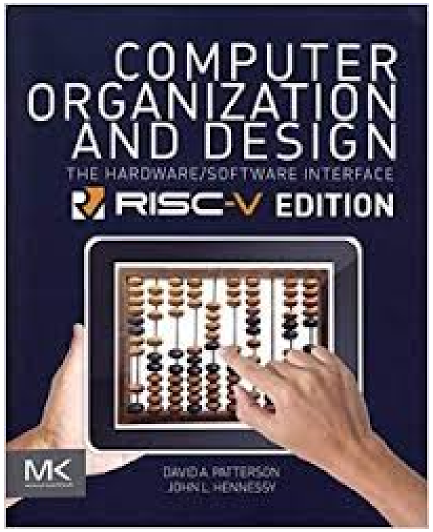
CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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
Edition	RISC-V ed
Language	English
ISBN	978-0128122754
Reference	[2]


COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
RISC-V EDITION
MK
DAVID A. PATTERSON
JOHN L. HENNESSY

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

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]

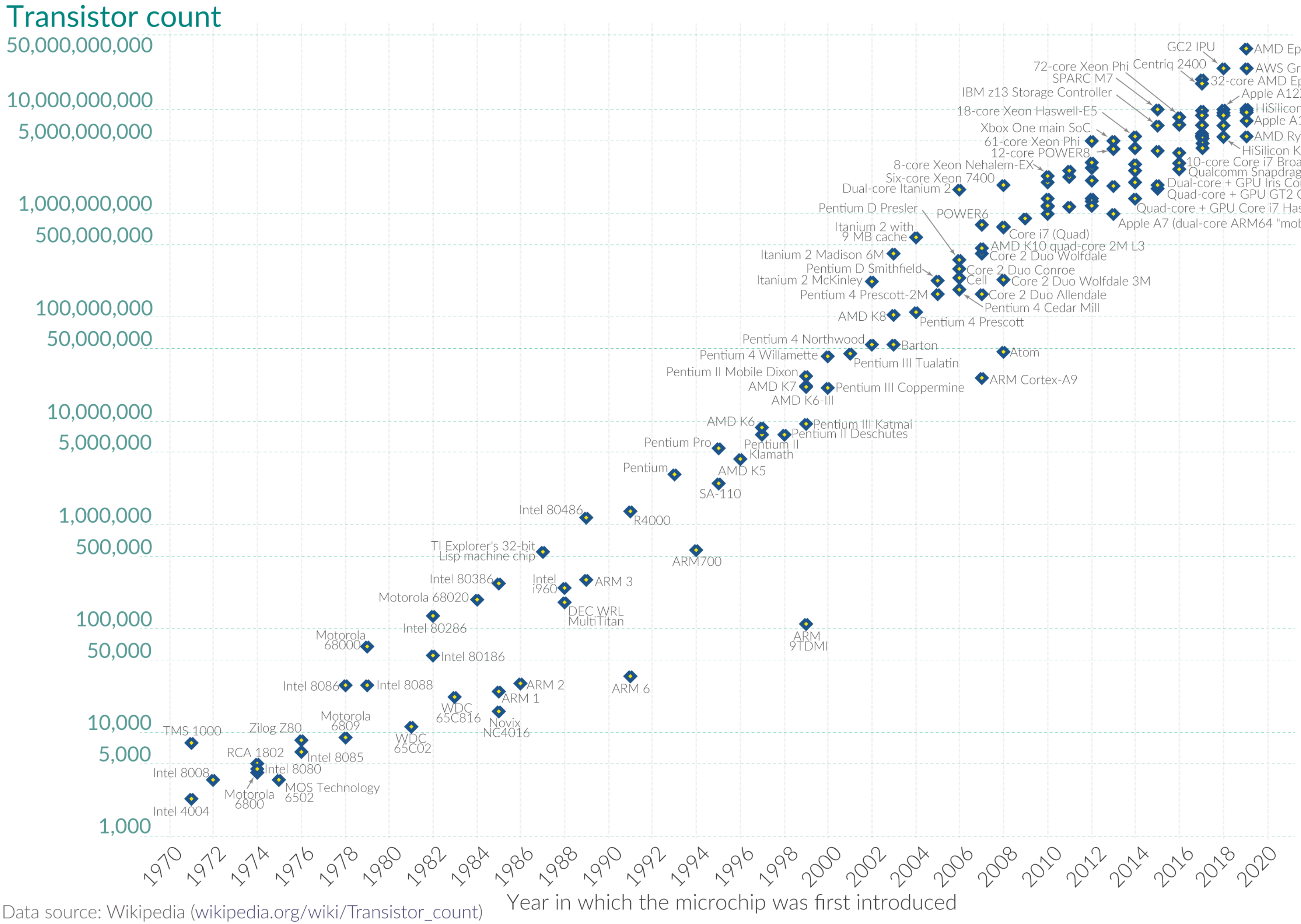
dirk w. HOFFMANN
GRUNDLAGEN DER TECHNISCHEN INFORMATIK
5., aktualisierte Auflage
Im Internet: Lösungen zu den Übungsaufgaben, Übungsblätter und weiteres Zusatzmaterial
HANSER

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

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)

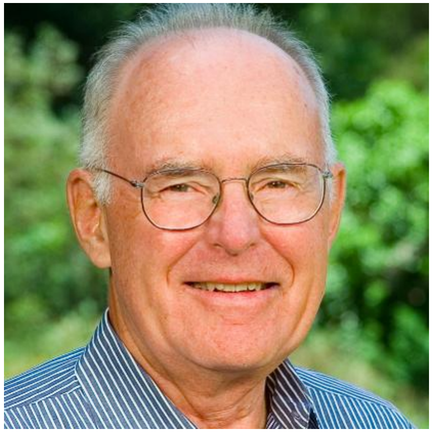
OurWorldinData.org – Research and data to make progress against the world's largest problems.

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

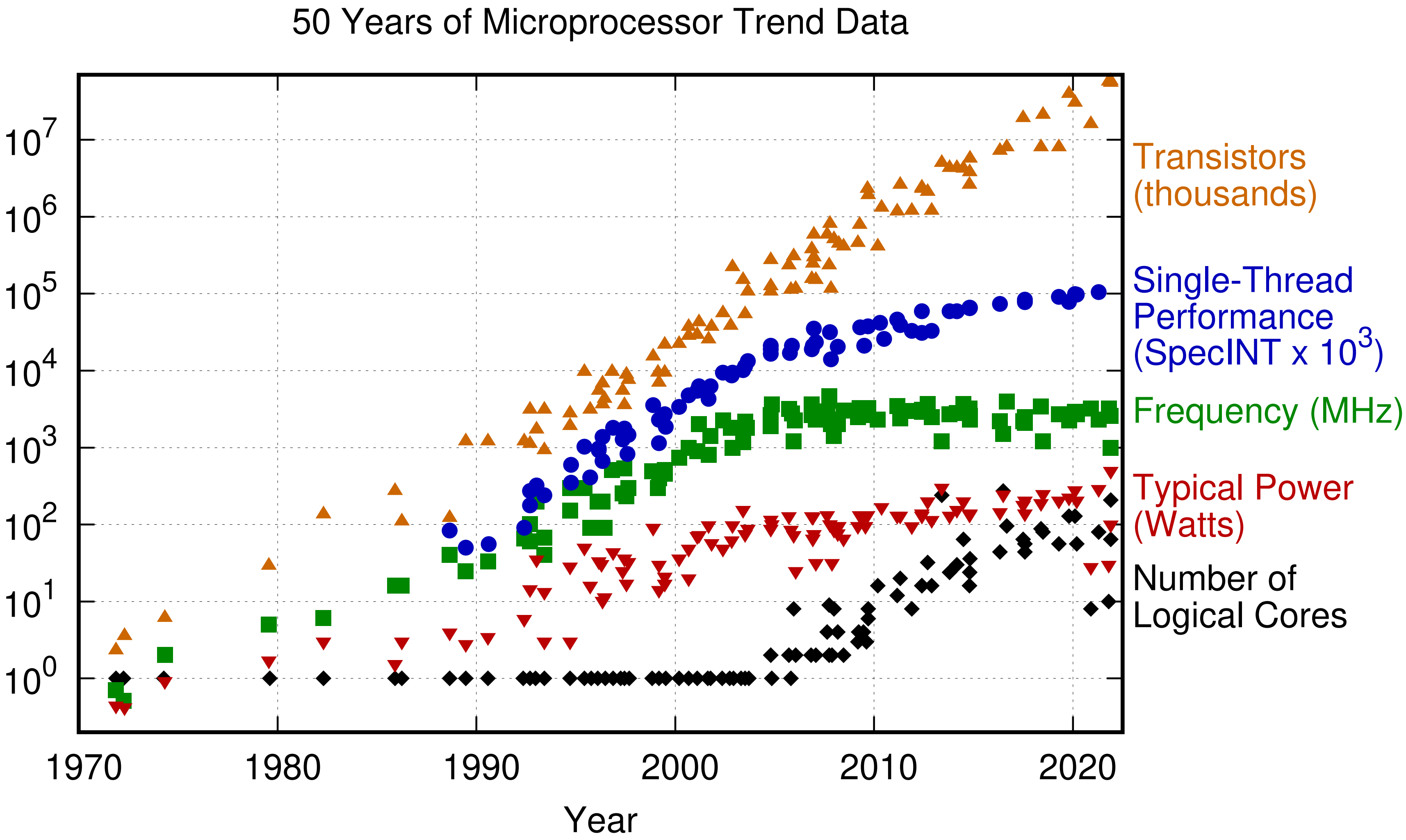


[source: forbes.com]

CAMPUS Rosenheim
Computer Science

Moore's law

Does it hold in the future?



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 2010-2021 by K. Rupp

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

CAMPUS Rosenheim
Computer Science

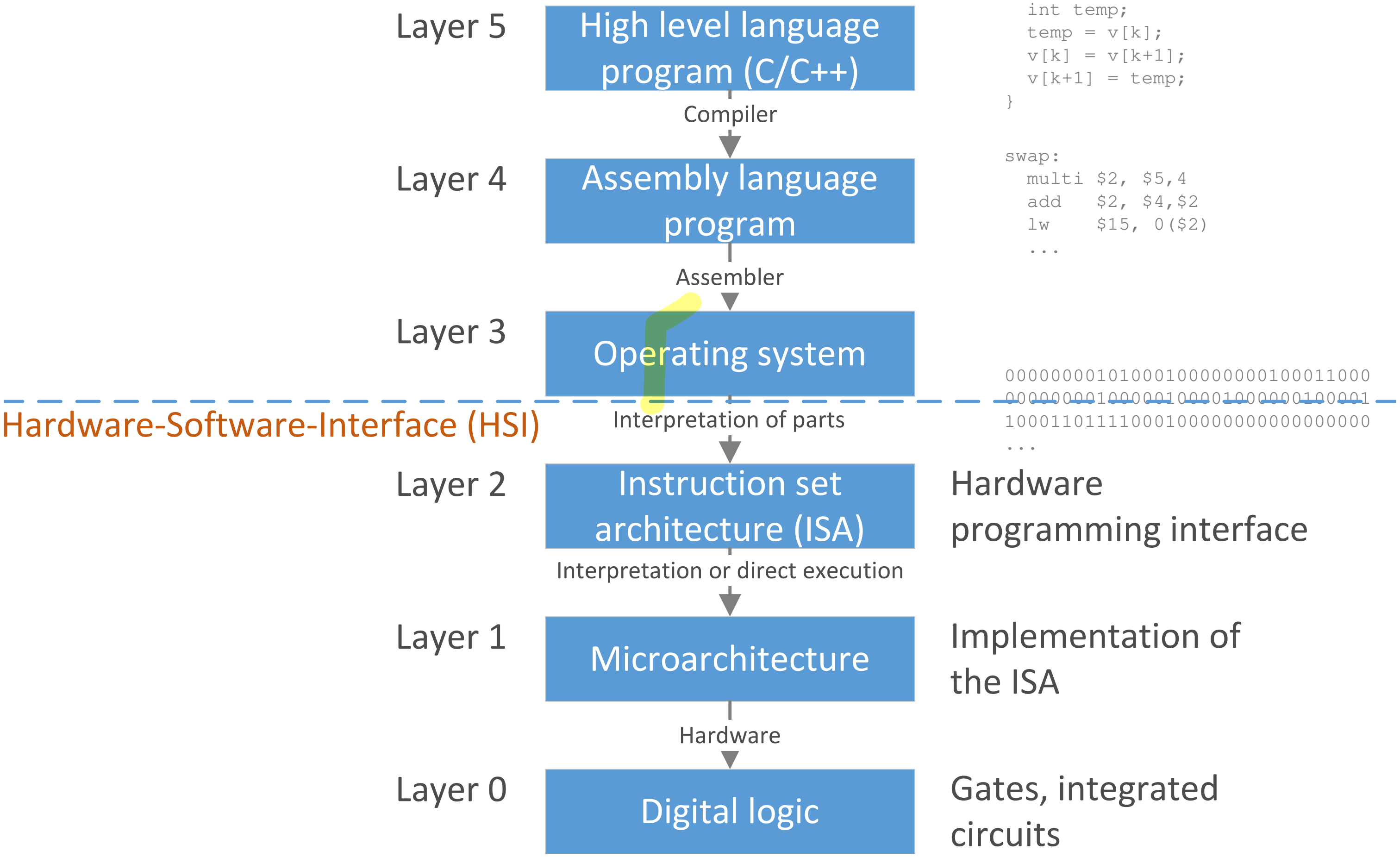
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Structured layers

From SOFTWARE to HARDWARE

CAMPUS Rosenheim
Computer Science

Structured layers



Layer 5 High level language program (C/C++)

Compiler

Layer 4 Assembly language program

Assembler

Layer 3 Operating system

Hardware-Software-Interface (HSI)

Interpretation of parts

Layer 2 Instruction set architecture (ISA) Hardware programming interface

Interpretation or direct execution

Layer 1 Microarchitecture Implementation of the ISA

Hardware

Layer 0 Digital logic Gates, integrated circuits

```
swap(int v[], int k){
  int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

```
swap:
  multi $2, $5,4
  add   $2, $4,$2
  lw    $15, 0($2)
  ...
```

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111100010000000000000000
...
```

[cmp: [1, P. 24], [2, P. 15]]

Assembler

The study of computer architecture is always a study of the instruction set architecture (ISA).

- 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

Assembler

The study of computer architecture is always a study of the instruction set architecture (ISA).

- 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

Assembler

The study of computer architecture is always a study of the instruction set architecture (ISA).

- 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

Assembler

The study of computer architecture is always a study of the instruction set architecture (ISA).

- 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

Computer types

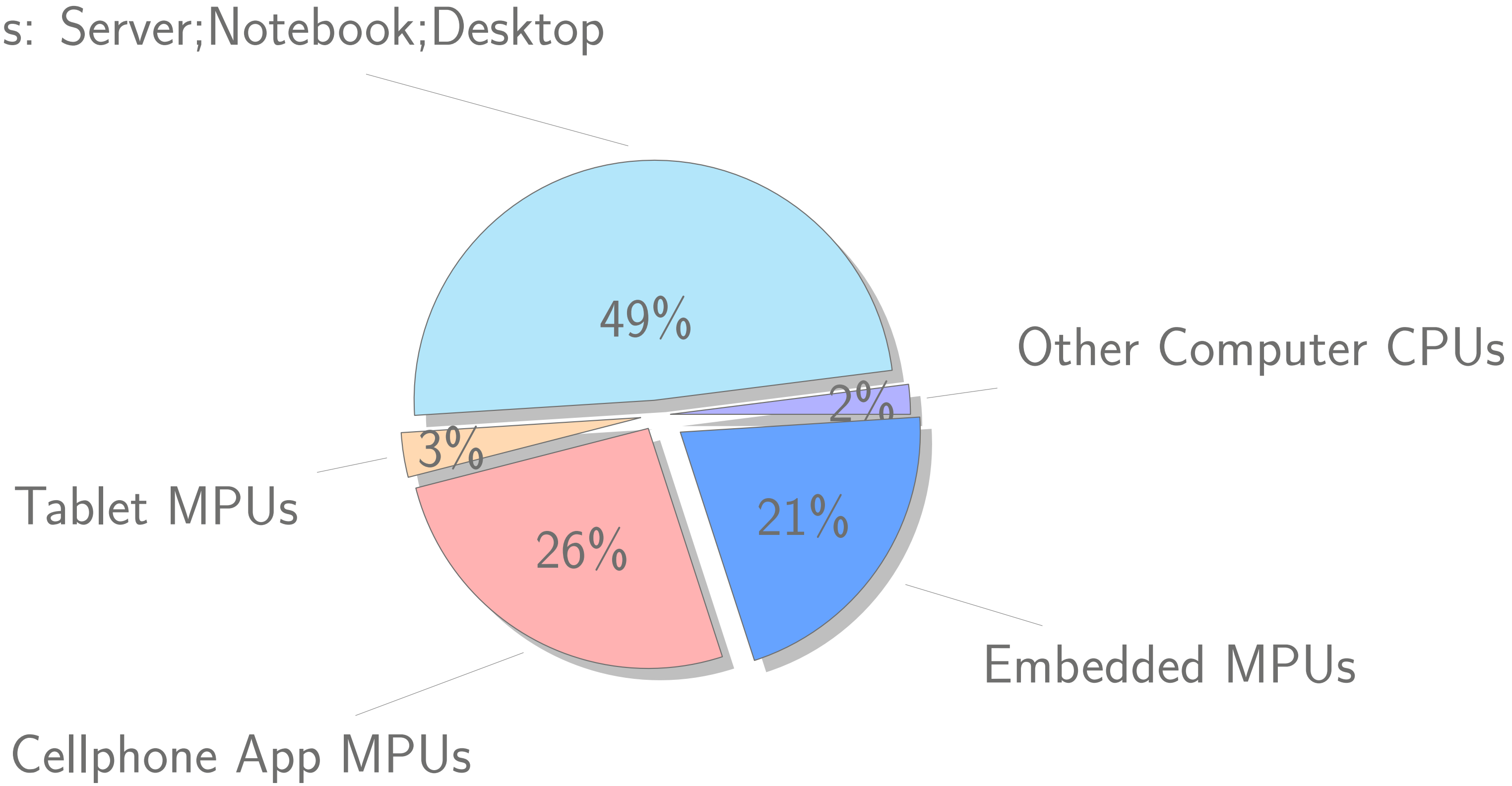
2020 MPU (microprocessor processor unit) sales by application:

[source: design-reuse.com]

CAMPUS Rosenheim
Computer Science

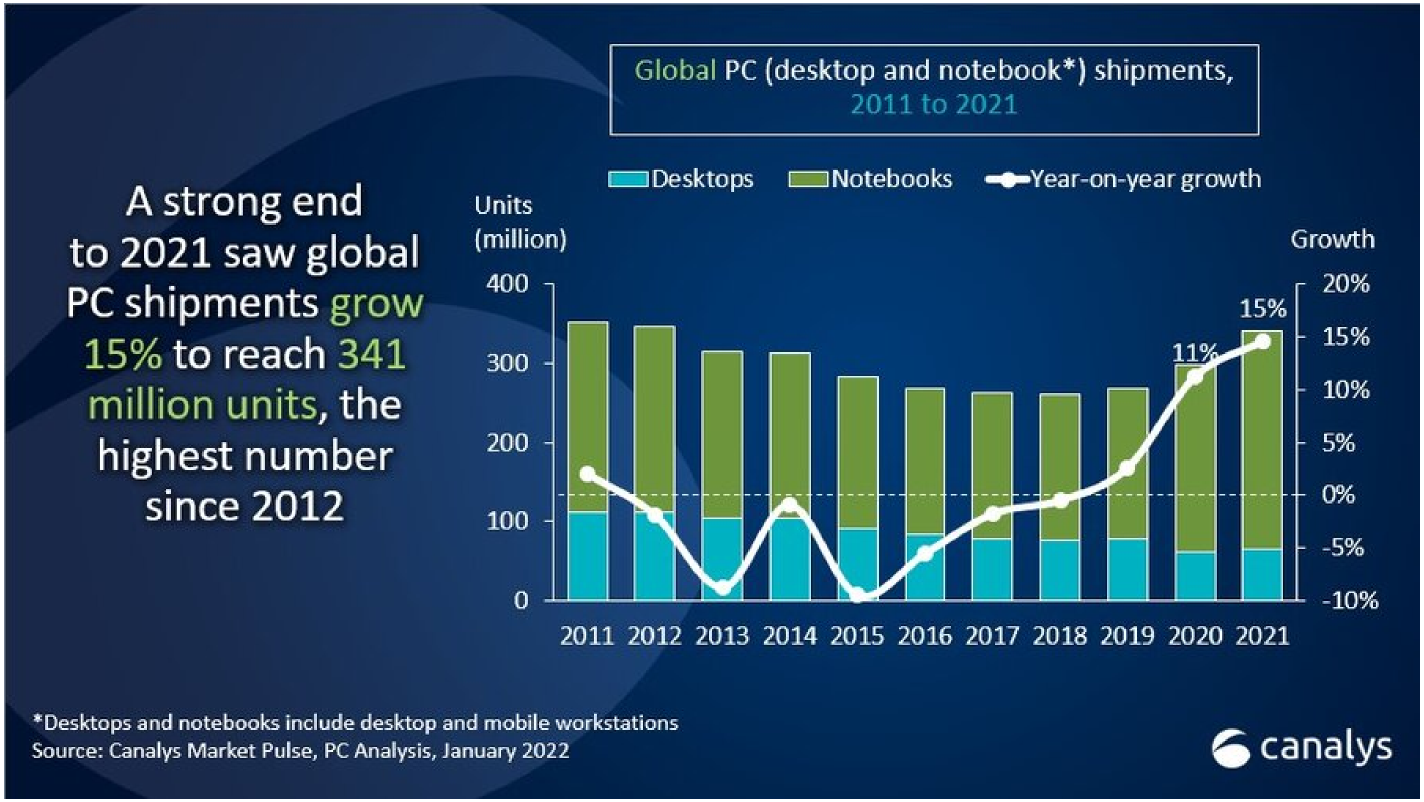
Computer types

2020 MPU (microprocessor processor unit) sales by application:



[source: design-reuse.com]

Computer types



[source: https://www.canalys.com/newsroom/global-pc-market-Q4-2021]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Computer types

- **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

Computer types

- **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

Computer types

- **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

Computer types

- **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

Computer types

- **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

Computer types

- **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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Analogue vs digital

General meaning

Analogue ≈ Corresponding, similar, analogous, applicable
An „analogue" is a similar or corresponding „thing".

Extended meaning in IT (electronics)

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Analogue vs digital

General meaning

Analogue	≈	Corresponding, similar, analogous, applicable An „analogue“ is a similar or corresponding „thing“.
Digital	≈	With numbers (lat. digitus = „Finger (for counting)“)

Extended meaning in IT (electronics)

Analogue vs digital

General meaning

Analogue	≈	Corresponding, similar, analogous, applicable An „analogue“ is a similar or corresponding „thing“.
Digital	≈	With numbers (lat. digitus = „Finger (for counting)“)

Extended meaning in IT (electronics)

Analogue vs digital

General meaning

Analogue	≈	Corresponding, similar, analogous, applicable An „analogue“ is a similar or corresponding „thing“.
Digital	≈	With numbers (lat. digitus = „Finger (for counting)“)

Extended meaning in IT (electronics)

Analogue	≈	continuous, steady, constantly

Analogue vs digital

General meaning

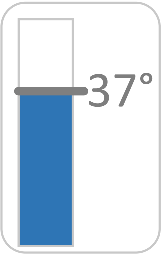
Analogue	≈	Corresponding, similar, analogous, applicable An „analogue“ is a similar or corresponding „thing“.
Digital	≈	With numbers (lat. digitus = „Finger (for counting)“)

Extended meaning in IT (electronics)

Analogue	≈	continuous, steady, constantly
Digital	≈	stepwise, discrete

Analogue vs digital

Analogue



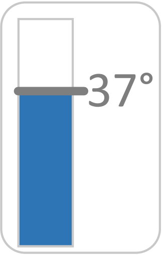
analogue thermometer

Temperature

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

Analogue vs digital

Analogue



Digital

analogue thermometer

Temperature

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

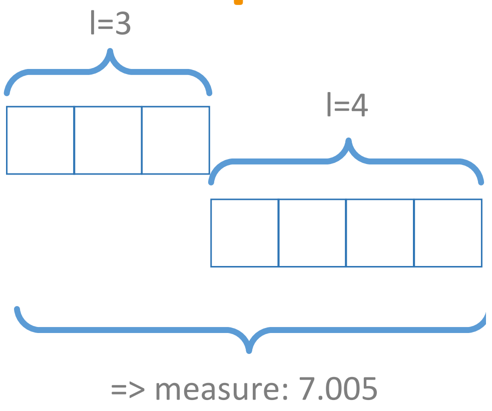
digital (thermometer)

Temperature

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

Analogue computer

Example 1

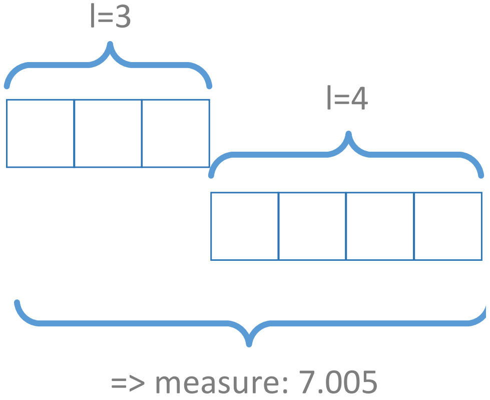


analogue add

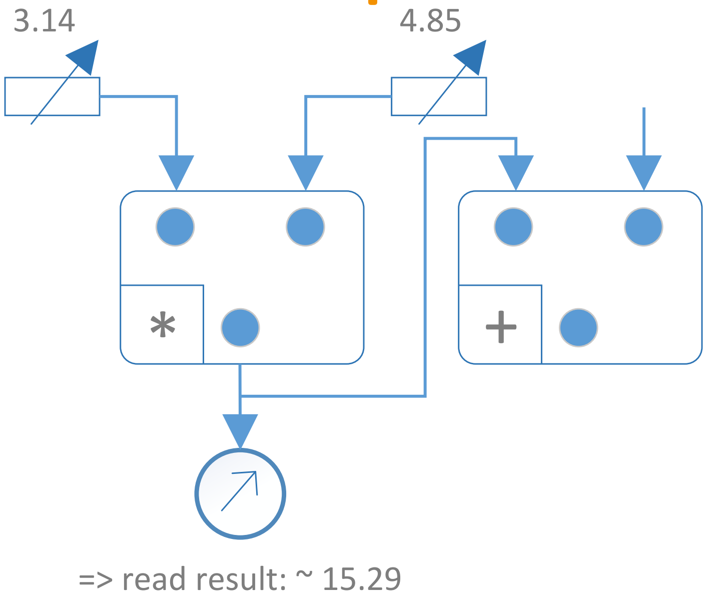
CAMPUS Rosenheim
Computer Science

Analogue computer

Example 1



Example 2



analogue add

analogue multiply

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.

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.

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

Technische Hochschule Rosenheim
Technical University of Applied Sciences

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