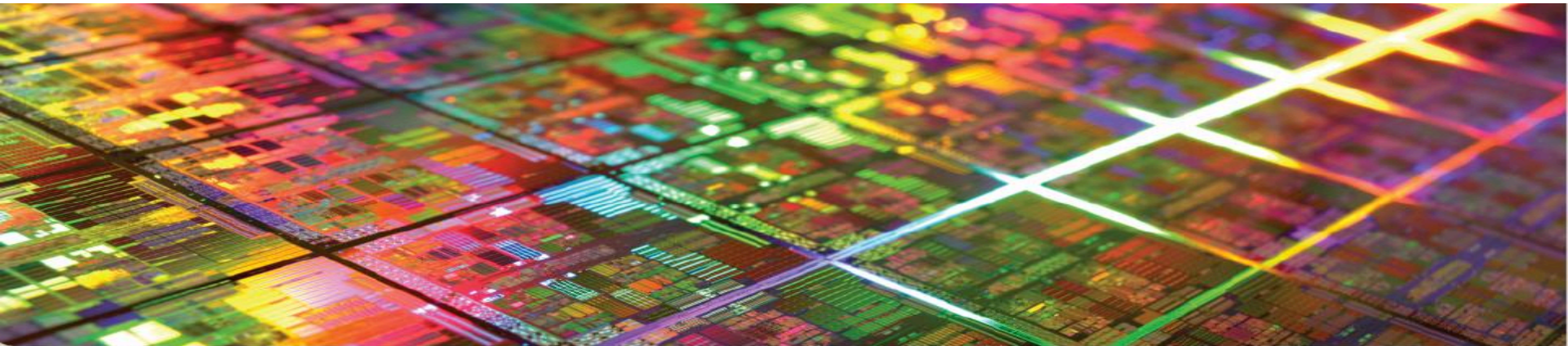


Rechnerorganisation

Prof. Dr. Wolfgang Karl

Vorlesung im Wintersemester 2025/2026 – Foliensatz: RO25-FS01



Kapitel 1

Motivation und Überblick über die Vorlesung Rechnerorganisation

Modul Technische Informatik

- **Vorlesung Digitaltechnik und Entwurfsverfahren (DT, TI-1)**
- **Vorlesung Rechnerorganisation (RO, TI-2)**

Modul Technische Informatik

- **Informatiker sollten den Aufbau, die Organisation und das Operationsprinzip von daten- bzw. informationsverarbeitenden Systemen verstehen.**
- **Problemlösungen erfordern ein sinnvolles Zusammenwirken von Hardware und Software: Hardware/Software Co-Design.**
- **Informatiker sollen kompetent in Software und in Hardware sein.**

VL Digitaltechnik und Entwurfsverfahren

Grundlagen der Informationsdarstellung im Rechner

■ Zahlendarstellung im Rechner

Stellenwertsystem

$$x_b = z_n b^n + z_{n-1} b^{n-1} + \dots + z_1 b^1 + z_0 + z_{-1} b^{-1} + \dots + z_{-m} b^{-m} = \sum_{i=-m}^n z_i b^i$$

Euklidischer Algorithmus
Horner Schema

Komplementdarstellung

$$z_{ek} = (2^{n+1} - 1) - z$$

$$z_{zk} = 2^{n+1} - z$$

Darstellung negativer
Zahlen

Festkommadarstellung

Gleitkommadarstellung

$$x = (-1^s * \text{Mantisse} * b^{\text{Exponent}}$$

Darstellung reeller
Zahlen

VL Digitaltechnik und Entwurfsverfahren

Grundlagen der Informationsdarstellung im Rechner

■ Zeichendarstellung im Rechner

	000	001	010	011	100	101	110	111
0000	NUL	DLE	SPACE	0	@	P	'	p
0001	SOH	DC 1	!	1	A	Q	a	q
0010	STX	DC 2	"	2	B	R	b	r
0011	ETX	DC 3	#	3	C	S	c	s
0100	EOT	DC 4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	
1111	SI	US	/	?	O		o	DEL

ASCII Kodierung

Unicode

■ Fehlerkorrigierende Codes

Hamming-Code

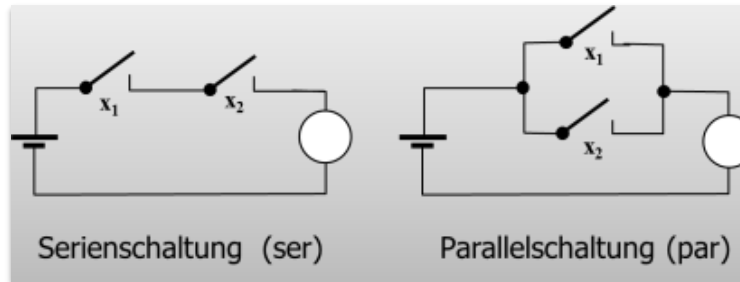
VL Digitaltechnik und Entwurfsverfahren

Schaltnetze

■ Formale Grundlagen

$$BA = [V, \oplus, \otimes]$$

$$SA = \{0,1\}, \wedge, \vee$$



Boole'sche Algebra

Schalteralgebra

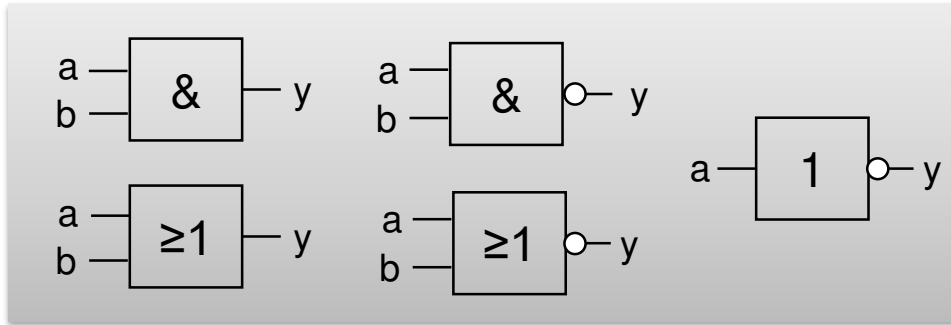
Schaltalgebra

VL Digitaltechnik und Entwurfsverfahren

Schaltnetze

■ Logikentwurf

- Abbildung einer abstrakten logischen Form auf realisierbare Bauelemente



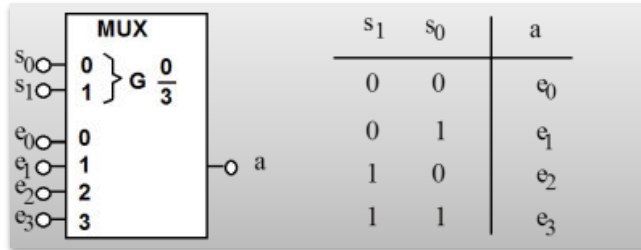
Logische Gatter

VL Digitaltechnik und Entwurfsverfahren

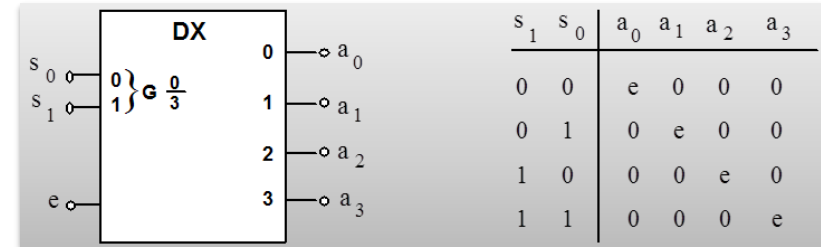
Schaltnetze

- Komplexere Bausteine
 - Beispiele

Multiplexer



Demultiplexer / Decoder

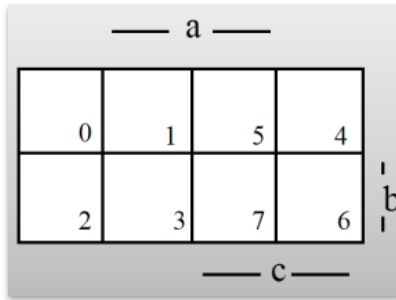


VL Digitaltechnik und Entwurfsverfahren

Schaltnetze

- Logikentwurf
 - Minimierungsverfahren

KV-Diagramm



Quine-McCluskey - Verfahren

Nr.	0. Ordnung	Nr.	1. Ordnung	Nr.	2. Ordnung
2	00010	2, 6	00-10	2, 6, 10, 14	0--10
4	00100	2, 10	0-010	2, 6, 18, 22	-0-10
5	00101	2, 18	-0010	2, 10, 18, 26	--010
6	00110	4, 5	0010-	4, 5, 12, 13	0-10- A
10	01010	4, 6	001-0	4, 6, 12, 14	0-1-0 B
12	01100	4, 12	0-100		
18	10010	5, 13	0-101	6, 14, 22, 30	--110
13	01101	6, 14	0-110	10, 14, 26, 30	-1-10
14	01110	6, 22	-0110	18, 22, 26, 30	1--10
22	10110	10, 14	01-10		
26	11010	10, 26	-1010		
		12, 13	0110-		
		12, 14	011-0		
		18, 22	10-10		
		18, 26	1-010		
30	11110	14, 30	-1110		
		22, 30	1-110		
		26, 30	11-10		
				Nr.	3. Ordnung
				2, 6, 10, 14,	---10 C
				18, 22, 26, 30	

algebraische Verfahren

graphische Verfahren

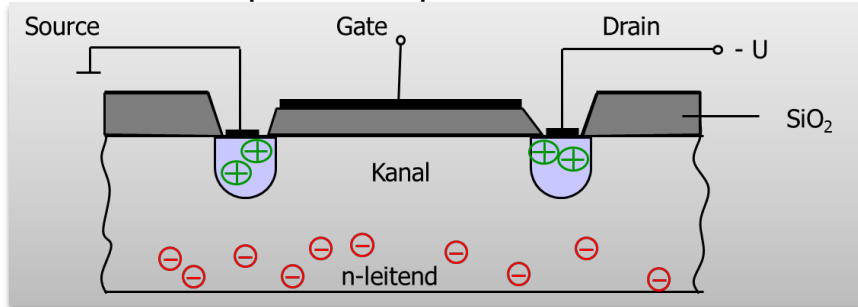
tabellarische Verfahren

VL Digitaltechnik und Entwurfsverfahren

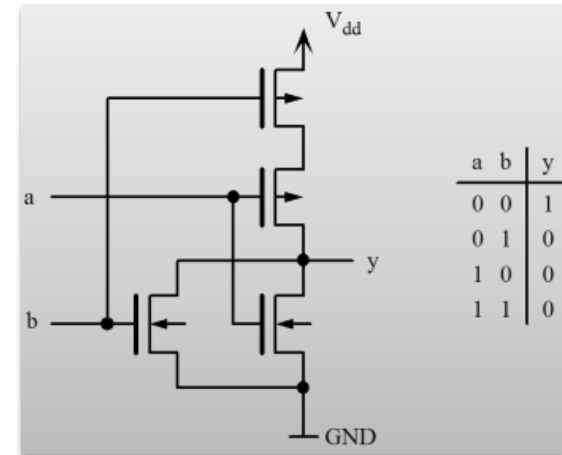
Schaltnetze

■ Realisierung von Schaltnetzen

Selbstsperrende p-Kanal-MOSFETs



CMOS NOR

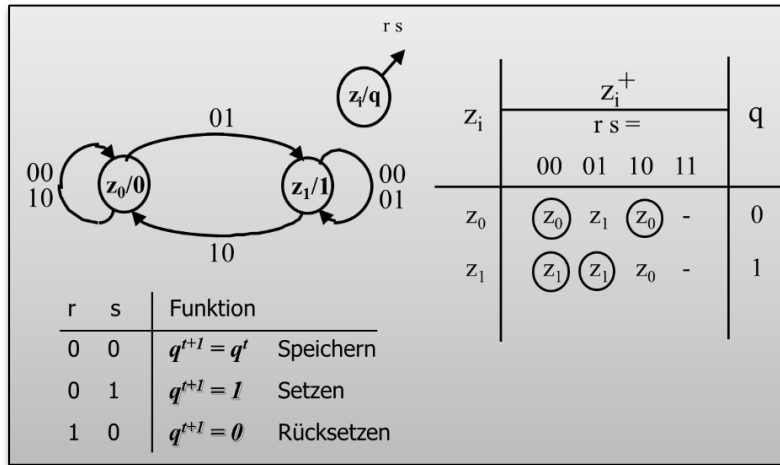


VL Digitaltechnik und Entwurfsverfahren

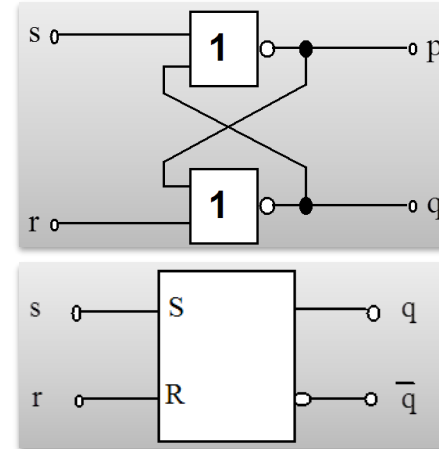
Schaltwerke

Entwurf von Schaltwerken

Entwurf eines einfachen Speichers



Asynchrones RS-Flipflop



D-Flipflops

JK-Flipflops

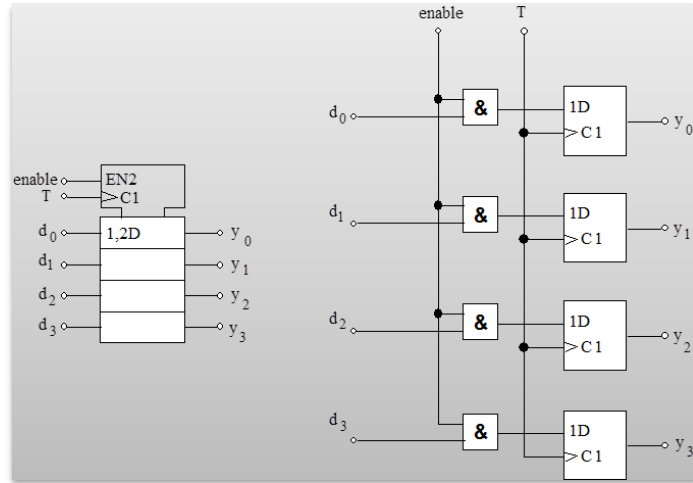
T-Flipflops

VL Digitaltechnik und Entwurfsverfahren

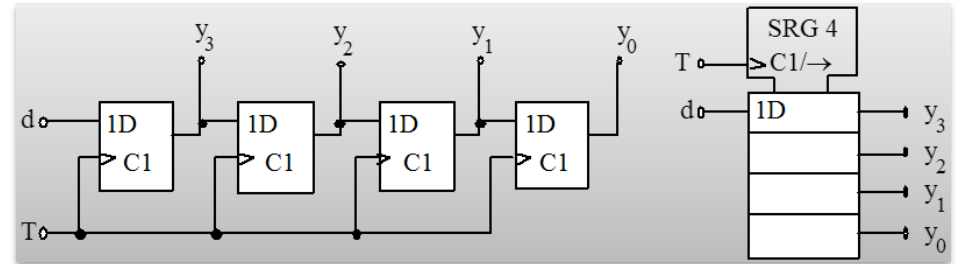
Schaltwerke

- Entwurf von Schaltwerken
 - Beispiele: Register

4-Bit-Register aus D-Flipflops



Schieberegister

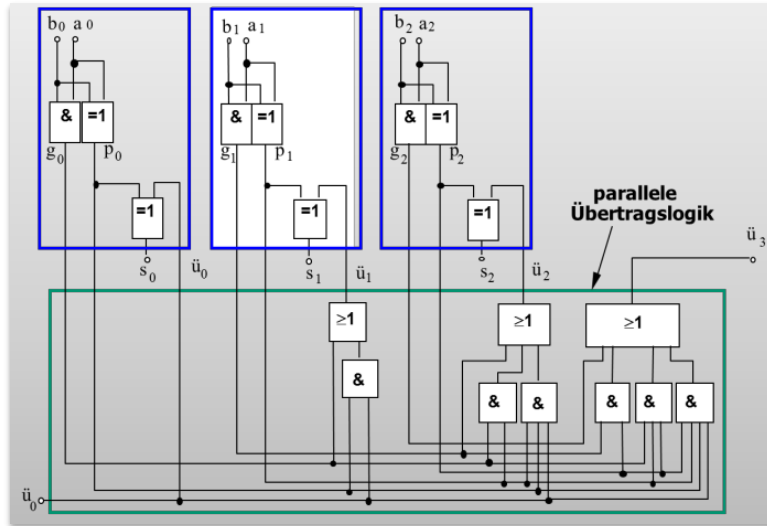


VL Digitaltechnik und Entwurfsverfahren

Schaltwerke

- Entwurf von Schaltwerken
 - Beispiele: Arithmetik-Komponenten

3-Bit-Carry-lookahead-Addierer

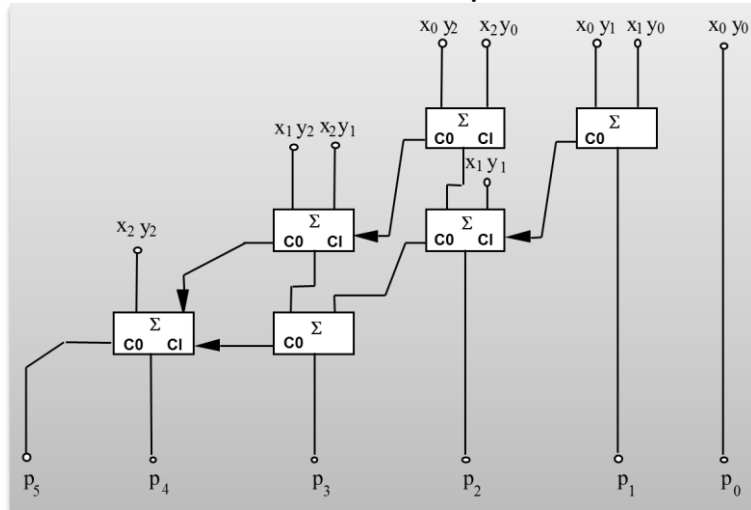


VL Digitaltechnik und Entwurfsverfahren

Schaltwerke

- Entwurf von Schaltwerken
 - Beispiele: Arithmetik-Komponenten

3-Bit-Parallelmultiplizierer



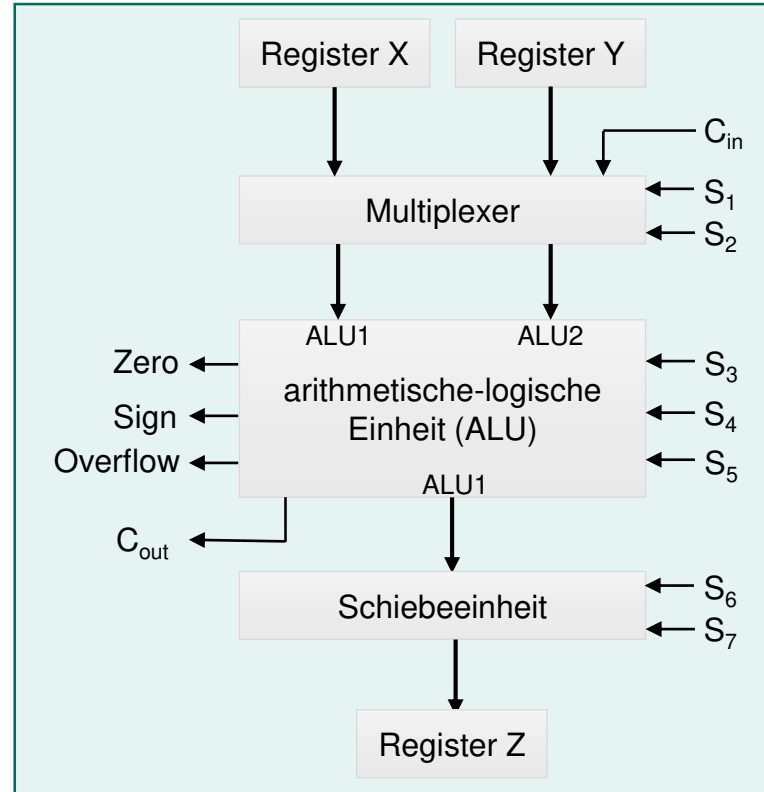
$$P = (x_0 \cdot 2^0 + x_1 \cdot 2^1 + x_2 \cdot 2^2) \cdot (y_0 \cdot 2^0 + y_1 \cdot 2^1 + y_2 \cdot 2^2)$$

$$P = 2^0 x_0 y_0 + 2^1 (x_0 y_1 + x_1 y_0) + 2^2 (x_0 y_2 + x_1 y_1 + x_2 y_0) + 2^3 (x_1 y_2 + x_2 y_1) + 2^4 x_2 y_2$$

$$P = 2^0 P_0 + 2^1 P_1 + 2^2 P_2 + 2^3 P_3 + 2^4 P_4 + 2^5 P_5$$

VL Digitaltechnik und Entwurfsverfahren

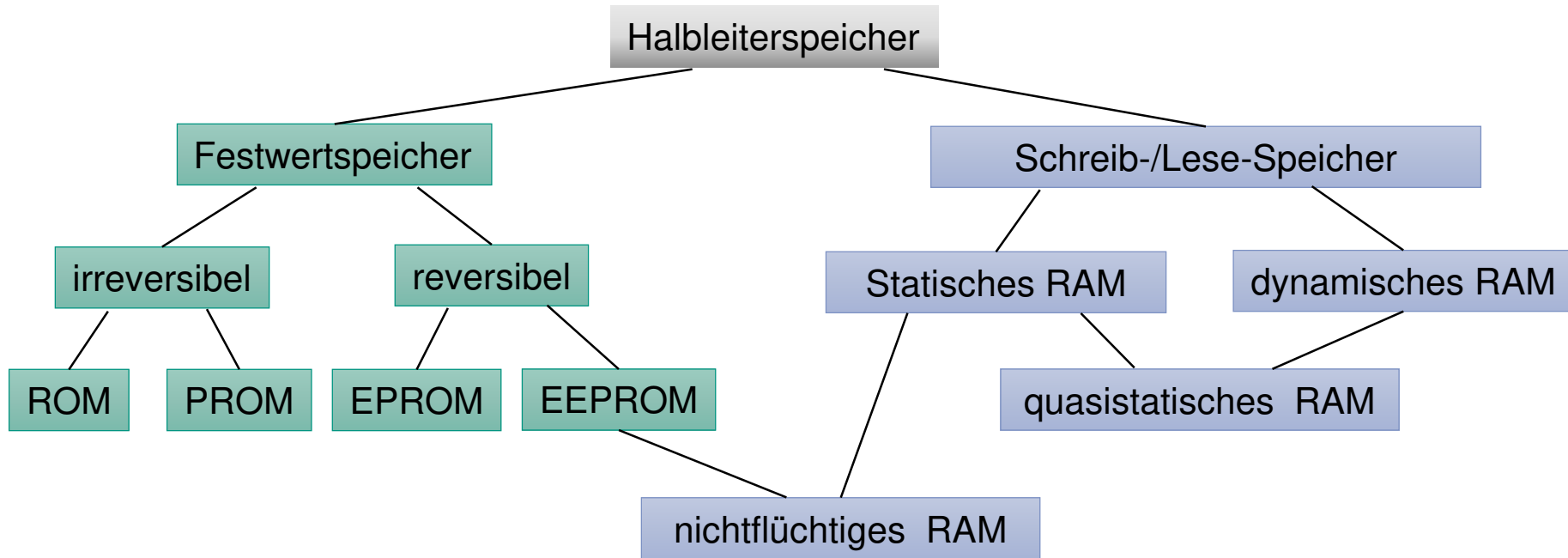
Arithmetische-logische Einheit



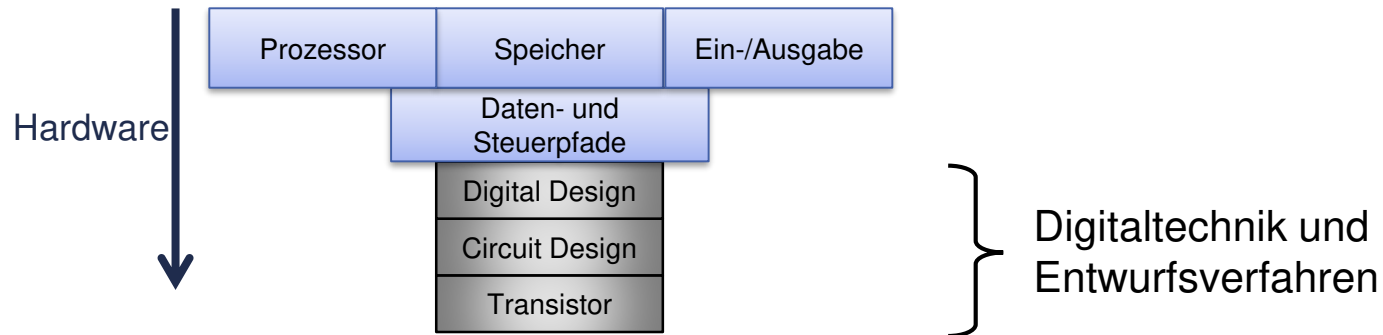
VL Digitaltechnik und Entwurfsverfahren

Speicher

■ Speichertypen

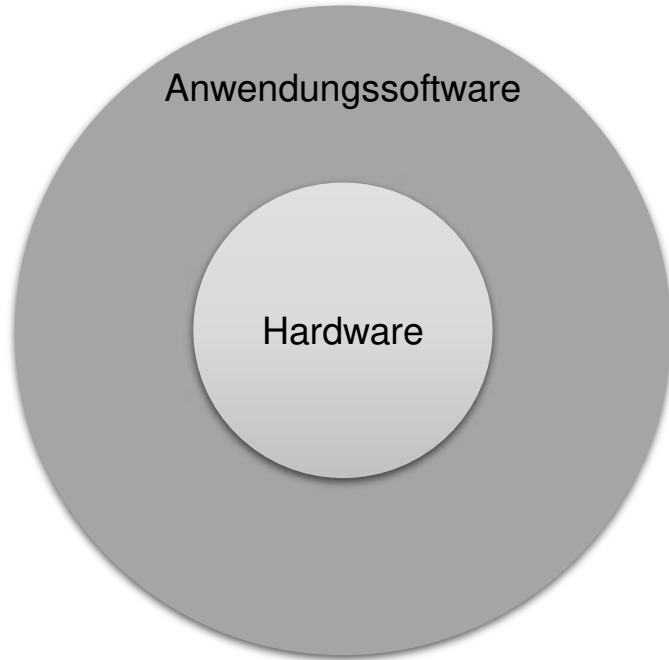


Modul „Technische Informatik“



Abstraktion

Vereinfachte Sicht von Hardware und Software als hierarchische Ebenen



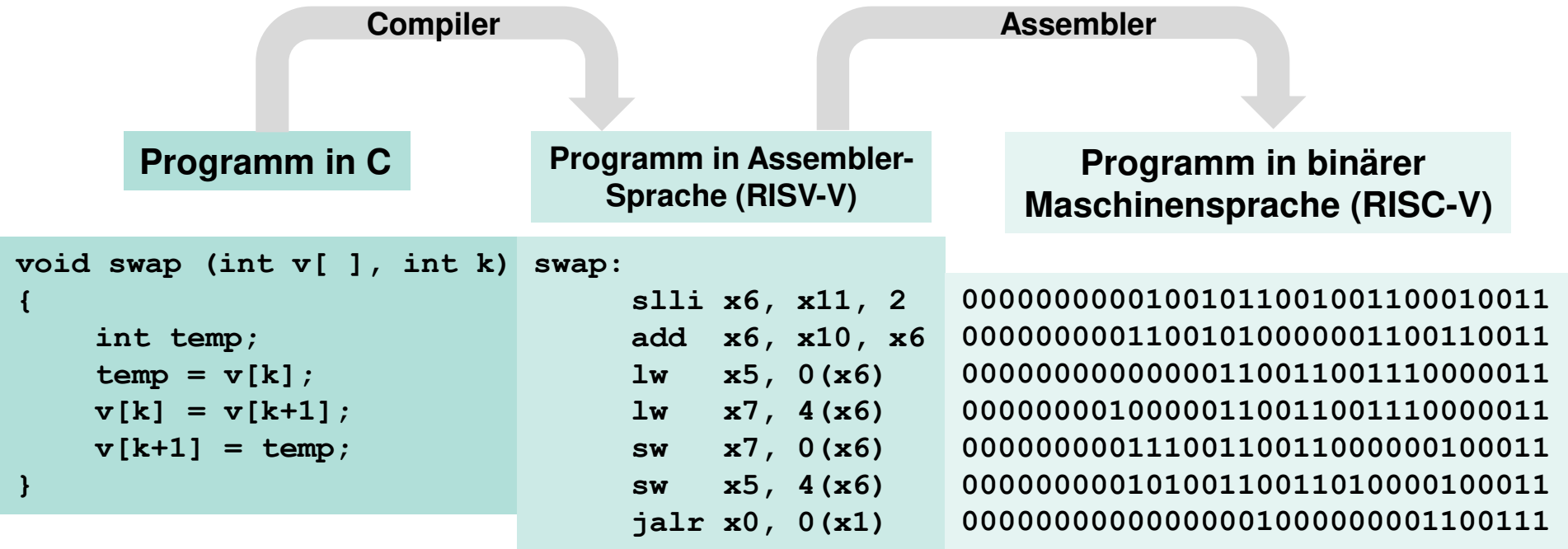
Problemstellung
(Textverarbeitung, Datenbanken,
Robotersteuerung, ...)

Algorithmen
(Suchen, Sortieren,
Bewegungsplanung, ...)

Höhere Programmiersprache
(C, C++, Java, ...)

Abstraktion

Von der höheren Programmiersprache zur Sprache der Hardware



Abstraktion

Von der höheren Programmiersprache zur Sprache der Hardware

■ Höhere Programmiersprache

- Beispiele: C, C++, Java, ...

■ Assembler-Sprache

- Symbolische Darstellung von Maschinenbefehlen;

■ Maschinenbefehl

- Instruktion in Binärdarstellung, die die Computer Hardware versteht und ausführt;

■ Maschinensprache

- Menge der Maschinenbefehle

Abstraktion

Systemsoftware

■ Compiler

- Übersetzt ein Programm in einer höheren Programmiersprache in ein Programm in Assembler-Sprache

■ Assembler

- Übersetzt eine symbolische Darstellung von Maschinenbefehlen in deren jeweilige binäre Version

Abstraktion

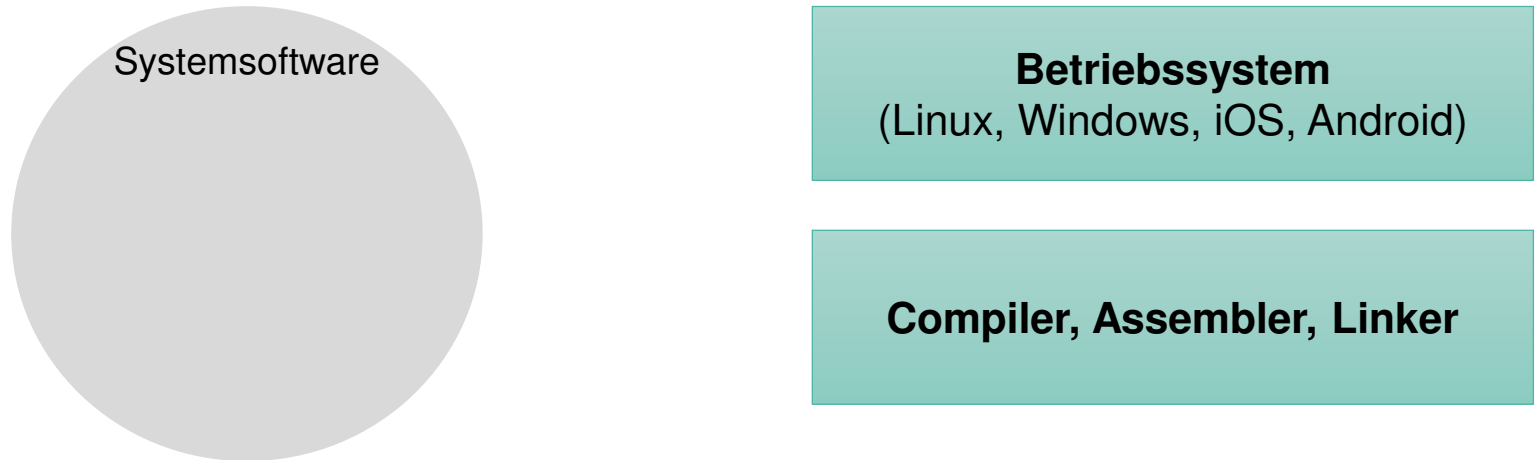
Systemsoftware

■ Betriebssystem

- Verwaltet die Ressourcen des Computers für die Anwendungsprogramme, die auf dem Computer ausgeführt werden
- Stellt eine Menge von Diensten und Überwachungsfunktionen bereit
 - Verwaltung von Prozessen und Threads (ausführbare Programme)
 - Speicherverwaltung
 - Bereitstellung grundlegender Schutzmechanismen
 - Bearbeitet grundlegende Ein- und Ausgabe-Operationen

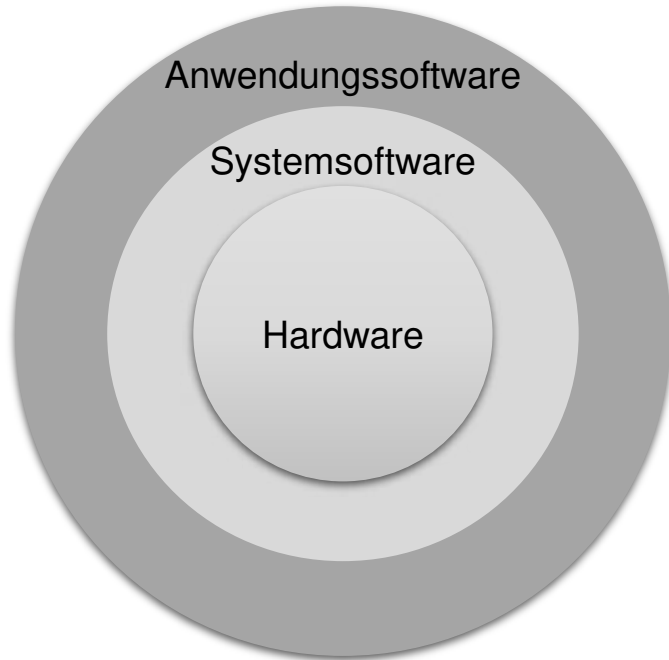
Abstraktion

Vereinfachte Sicht von Hardware und Software als hierarchische Ebenen



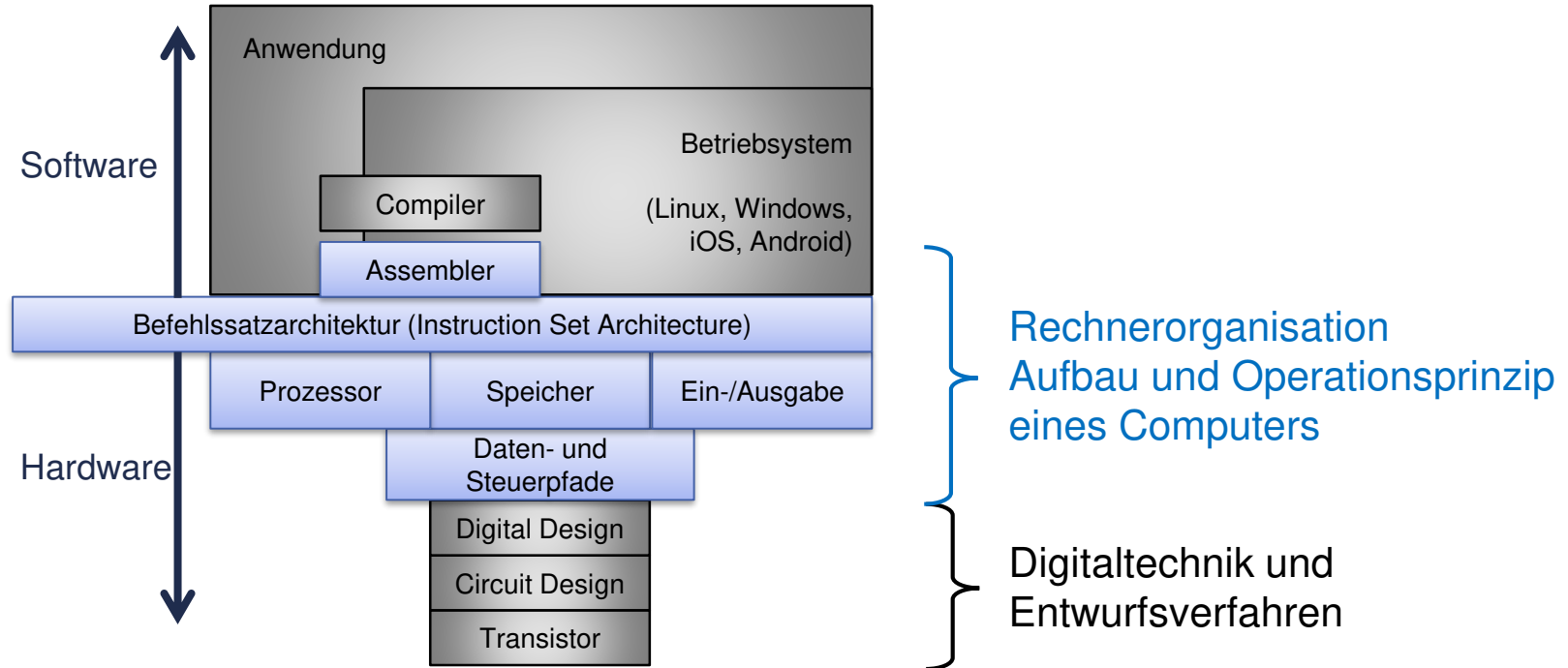
Abstraktion

Vereinfachte Sicht von Hardware und Software als hierarchische Ebenen



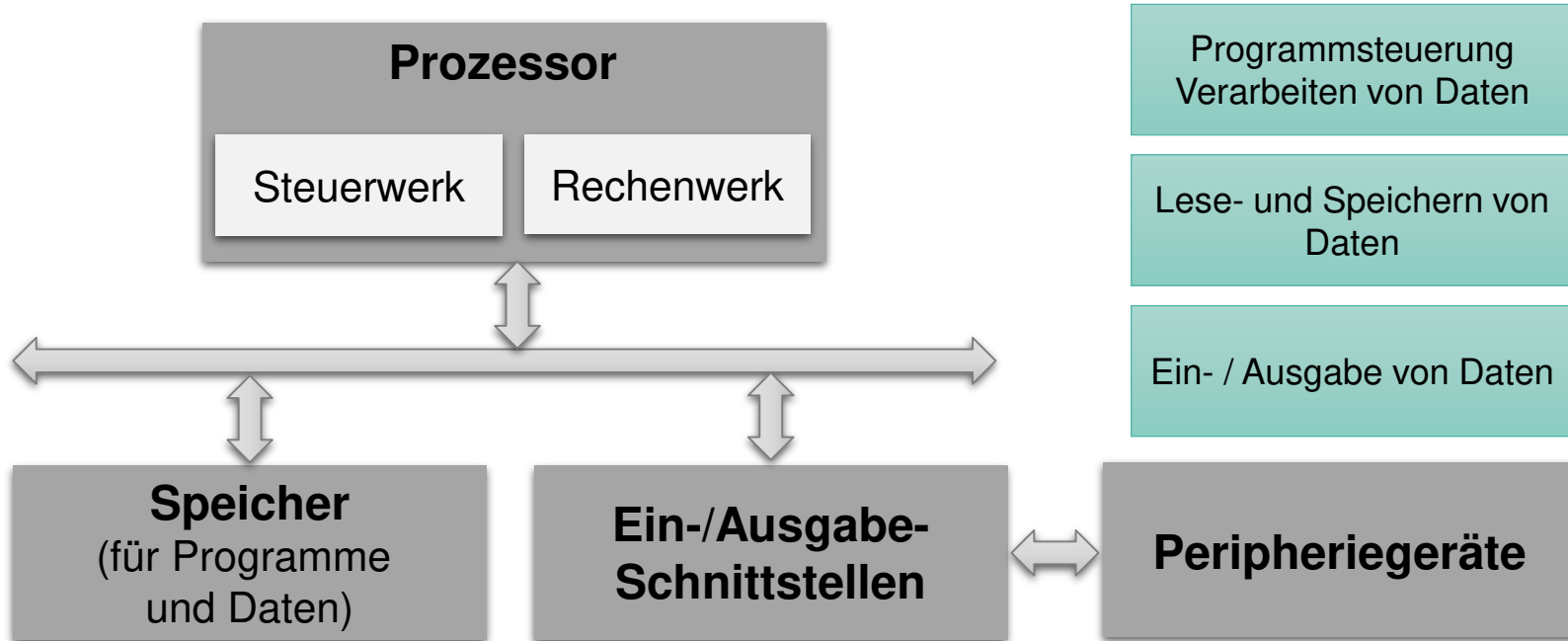
Modul Technische Informatik

Vereinfachte Sicht von Hardware und Software als hierarchische Ebenen



Rechnerorganisation

Einfaches Modell eines programmierbaren Universalrechners



Einsatzgebiete / Rechnermärkte

Desktop Computing

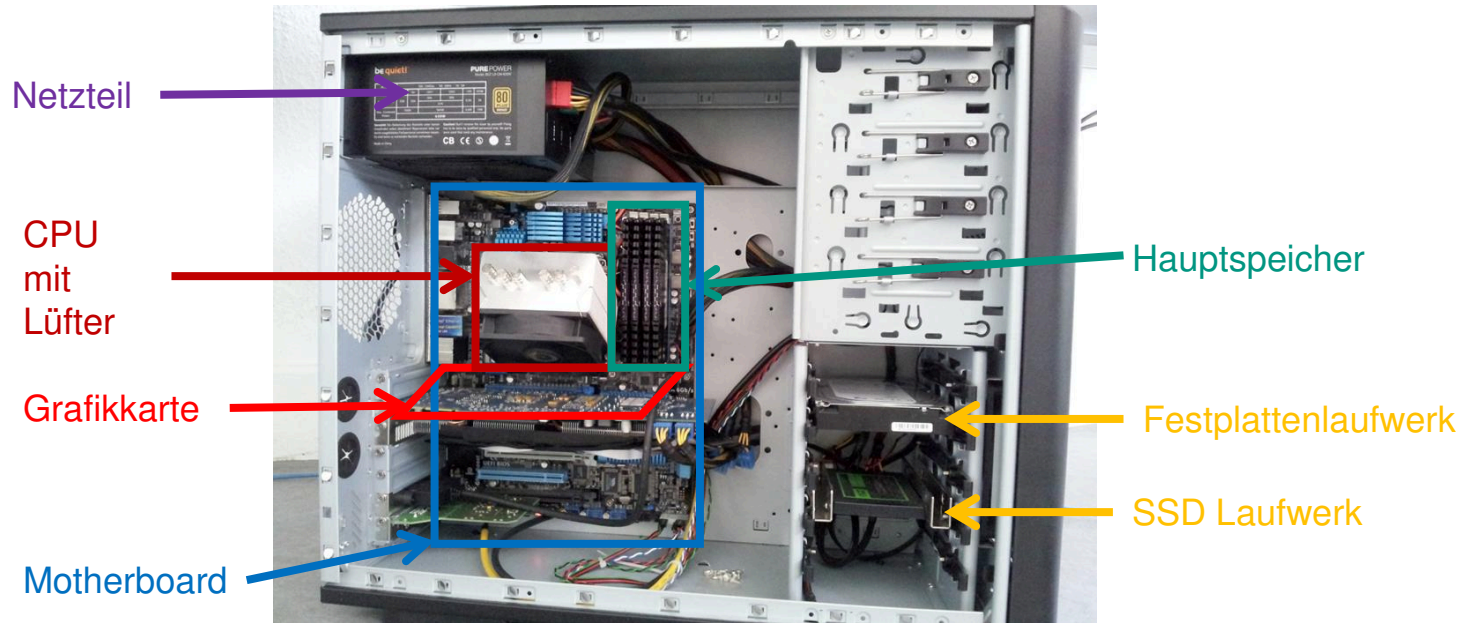
- PCs bis Workstations, Laptops, Notebooks
 - Günstiges Preis- / Leistungsverhältnis
 - Ausgewogene Rechenleistung für ein breites Spektrum von Anwendungen, einschließlich interaktiver Anwendungen (Grafik, Video, Audio) oder WEB-Anwendungen



Einsatzgebiete / Rechnermärkte

Desktop Computing

- Personal Computer (PC): geöffnetes Gehäuse



Einsatzgebiete / Rechnermärkte

■ Personal Mobile Devices (PMD)

- Beispiele: Mobiltelefone, Smartphones, Tablets
 - Drahtlose Geräte mit Multimedia-Benutzerschnittstelle
- Breites Anwendungsspektrum, insbesondere Web-basierte und Multimedia-orientierte Anwendungen
- Leistungsfähige Unterstützung durch die Hardware



Quelle: www.apple.com

Einsatzgebiete / Rechnermärkte

Personal Mobile Devices (PMD)

■ Eigenschaften

- Hohe Rechenleistung
- Energieeffizienz
 - Möglichst lange Akku-Laufzeiten
- Low-Power Design
 - Wärmeabfuhr: Keine Lüfter zur Kühlung

■ Randbedingungen

- Begrenzte Größe für das Gerät;
- Hohe Stückzahlen → Herstellungskosten müssen berücksichtigt werden;

Einsatzgebiete / Rechnermärkte

Personal Mobile Devices (PMD)

■ Blick unter das Gehäuse: iPhone 13



Geschichtete Hauptplatine

Gehäuse
(Bildschirm, Rückwand)

Peripheriegeräte
(Kameras, Ohrlautsprecher, ...)

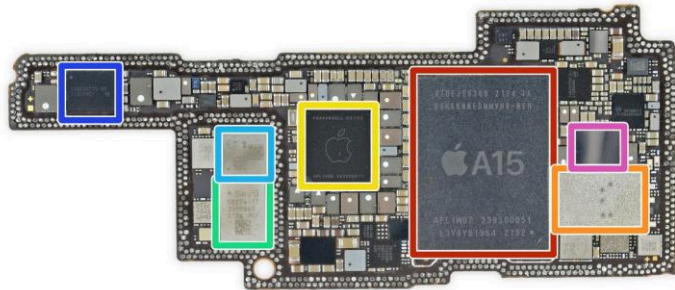
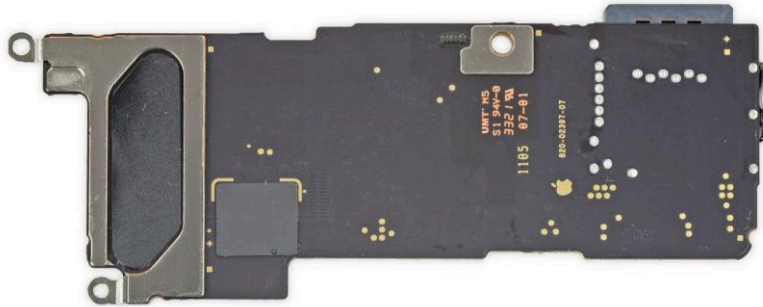
Akku

<https://guide-images.cdn.ifixit.com/igi/JtBACGbyZcYSdBfL.huge>

Einsatzgebiete / Rechnermärkte

Personal Mobile Devices (PMD)

- Blick unter das Gehäuse: iPhone 13 geschichtete Hauptplatine



<https://de.ifixit.com/Anleitung/Image/meta/EsJabFYWLkmQoCvV>

Prozessor und Speicher
(Apple APL1W07 A15 Bionic SoC,
geschichtet mit 6 GB SDRAM)

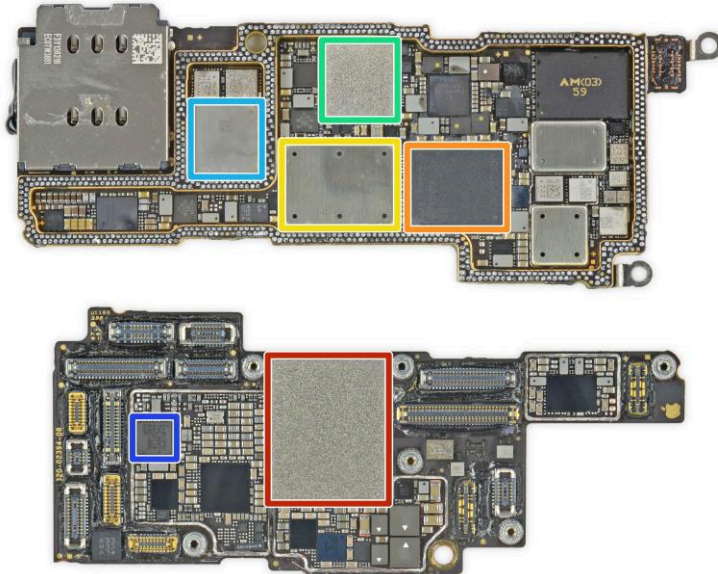
Ein-/Ausgabeschchnittstellen
(Kommunikation)

Leistungsmanagement

Einsatzgebiete / Rechnermärkte

Personal Mobile Devices (PMD)

- Blick unter das Gehäuse: iPhone 13 geschichtete Hauptplatine



128 GB Flash-Speicher

Ein-/Ausgabe-Schnittstellen
(Audio-, Video-Verarbeitung, Kommunikation,
z.B. 5G Modem, 5G RF-Transceiver,
WiFi/Bluetooth Modul, Display-Ansteuerung,
Beschleunigungsmesser, ...)

<https://de.ifixit.com/Anleitung/Image/meta/EsJabFYWLkmQoCvV>

Einsatzgebiete / Rechnermärkte

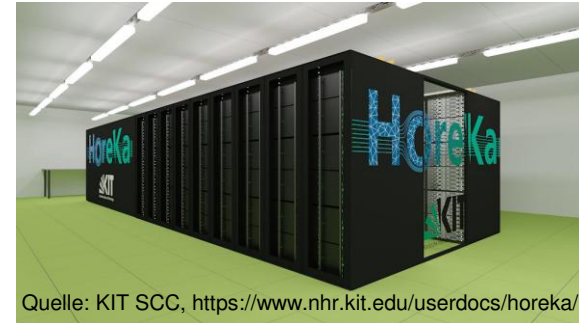
Server

- Rechen- und datenintensive Anwendungen, transaktionsorientierte Anwendungen;
- Hohe Anforderungen an die Verfügbarkeit und Zuverlässigkeit, Energieeffizienz;
- Skalierbarkeit;
- Große Datei-Systeme, Ein-/Ausgabesysteme;
- Abgeschlossene Räume, kostenintensiv;

Einsatzgebiete / Rechnermärkte

Server: Höchstleistungsrechner

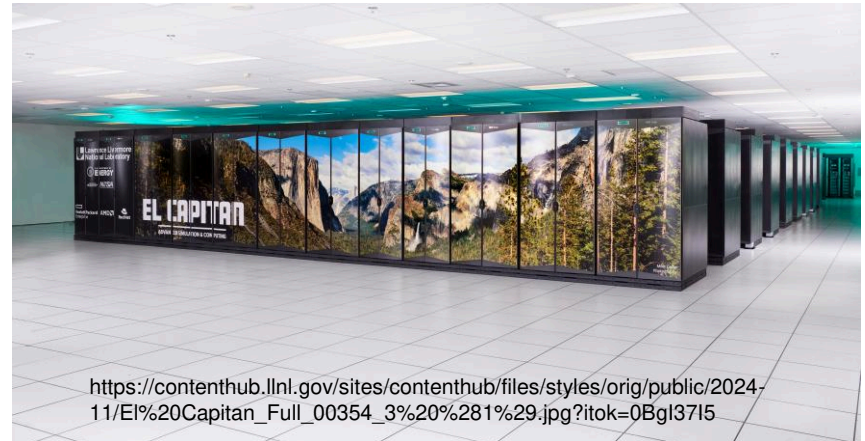
- Anwendungsbereich: technisch-wissenschaftliche Anwendungen
 - Rechnergestützte Simulation
 - Strömungsmechanik
 - Modellierung der globalen klimatischen Veränderungen
 - Struktur von Materialien
 - Medizintechnik
 - ...
- Massiv-parallele und kommunikationsintensive Anwendungen
 - Hohe Rechenleistung insbesondere bezüglich Gleitkommaverarbeitung
 - Data analytics
- Batch-Betrieb



Einsatzgebiete / Rechnermärkte

Server: Höchstleistungsrechner

- **Beispiel: El Capitan - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11,**
 - TOP 500 Liste - Nr. 1 (Juni 2025)
 - Zur Zeit der schnellste Höchstleistungsrechner der Welt!
 - Standort: DOE/NNSA/Lawrence Livermore National Lab, USA
 - Anzahl Prozessorkerne: 11.039.616
 - Rechenleistung:
 - **1,742 x 10¹⁸ Flops**
 - **3. ExaFlop-Computer!**
 - Power: 29,5 MW



Einsatzgebiete / Rechnermärkte

Server im kommerziellen Bereich

- **Mainframes: Großrechneranlagen, z. B. IBM z-Systems**
 - In Rechenzentren für Banken, Versicherungen, Verwaltungen
 - Transaktionsorientierte Anwendungen
 - Hohe Ausfallsicherheit



Quelle: <https://developer.ibm.com/components/ibmz/blogs/inside-the-new-ibm-z15-t02-and-linuxone-iii-it2/>

Einsatzgebiete / Rechnermärkte

Server im kommerziellen Bereich

■ Cluster

- Menge von Desktop-Rechnern oder Servern, die über ein lokales Netzwerk (LAN) oder schnelles dediziertes Verbindungsnetzwerk miteinander verbunden sind;
 - Erscheint als ein großer Rechner;
 - Jeder Knoten hat sein eigenes Betriebssystem
 - Kommunikation über Netzwerkprotokoll
-
- Interaktive Anwendungen
 - WEB-Suche, soziale Netzwerke, Video-Streaming, Online Handel ...
 - Software-as-Service (SaaS)

Einsatzgebiete / Rechnermärkte

Ware-House-Scale-Computer (WSC)

- Grundlage für Internetdienste, die täglich weltweit genutzt werden:
 - Internet-Suche
 - Soziale Netzwerke
 - E-Mail-Dienste
 - Video sharing
 - Online-shopping
 - ...
- **Cloud-Computing**

Einsatzgebiete / Rechnermärkte

Ware-House-Scale-Computer (WSC): Beispiel: Google

■ Aufbau

- 50.000 -100.000 Server mit Verbindungsnetzwerk



Quelle: J. Hennessy, D. Patterson: Computer Architecture – A Quantative Approach, Morgan Kaufmann Publishers, 6th Ed., 2019, Copyright © 2019, Elsevier Inc. All rights reserved.

Einsatzgebiete / Rechnermärkte

Ware-House-Scale-Computer (WSC): Beispiel: Google

■ Aufbau

- Infrastruktur für Kühlung, Stromversorgung



Quelle:<https://datacenters.google/locations/hanau-germany/>

Einsatzgebiete / Rechnermärkte

Eingebettete Systeme (Embedded Systems)

- Mikroprozessorsysteme, eingebettet in Geräten, daher nicht unbedingt sichtbar;
 - Beispiele: Automobil, Unterhaltungselektronik, Telekommunikation, Haushaltsgeräte, ...
- Vielfältige Aufgaben:
 - Steuerungsaufgaben
 - Echtzeitanforderungen
- Rechensysteme sind auf spezielle Aufgaben zugeschnitten
 - Hohe Leistungsfähigkeit für spezielle Anwendung
- Breites Preis-/Leistungsspektrum
 - Von einfachen 8-, 16-Bit Microcontrollern bis hin zu komplexen Spezialprozessoren

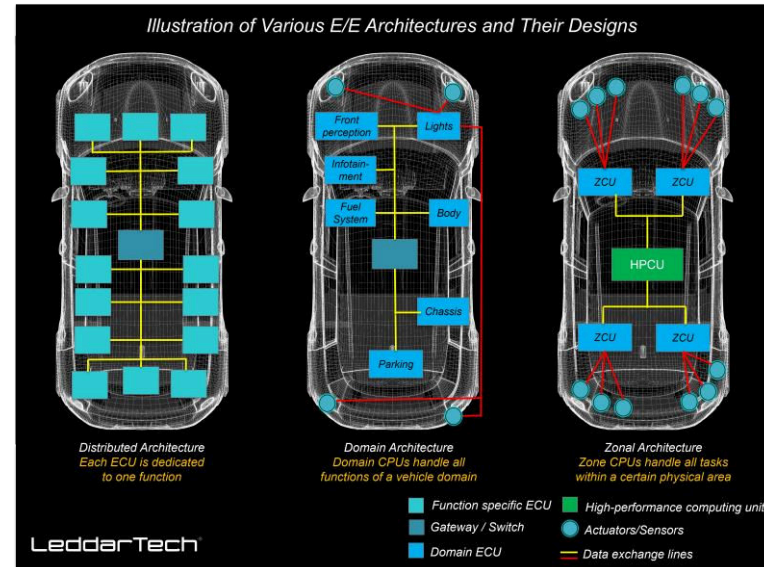
Einsatzgebiete / Rechnermärkte

Eingebettete Systeme (Embedded Systems)

- Beispiele: Automobil
 - E/E-Architektur: elektrische/elektronische Architektur



Quelle: <https://www.vdi-wissensforum.de/weiterbildung-automobil/ee-architektur/>



<https://leddartech.com/white-paper-exploring-the-relationship-between-ad-as-ee-architecture-and-perception-systems/>

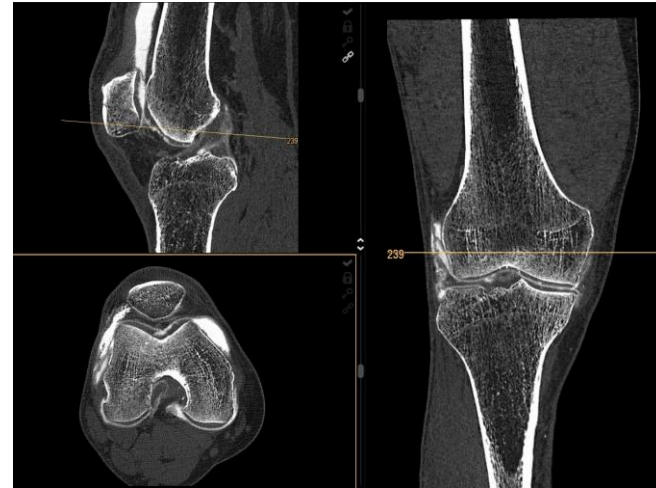
Einsatzgebiete / Rechnermärkte

Eingebettete Systeme (Embedded Systems)

- Beispiele: Medizintechnik
 - Computer Tomographie (CT)
 - Bildgebendes Verfahren in der Radiologie



[https://de.wikipedia.org/wiki/Computertomografie#/media/Datei:CT_Naeotom_Alpha_Pilsen_2022_\(cropped\).jpg](https://de.wikipedia.org/wiki/Computertomografie#/media/Datei:CT_Naeotom_Alpha_Pilsen_2022_(cropped).jpg)



https://de.wikipedia.org/wiki/Computertomografie#/media/Datei:Knee_ScannedWithPhotoncountingCt.jpg

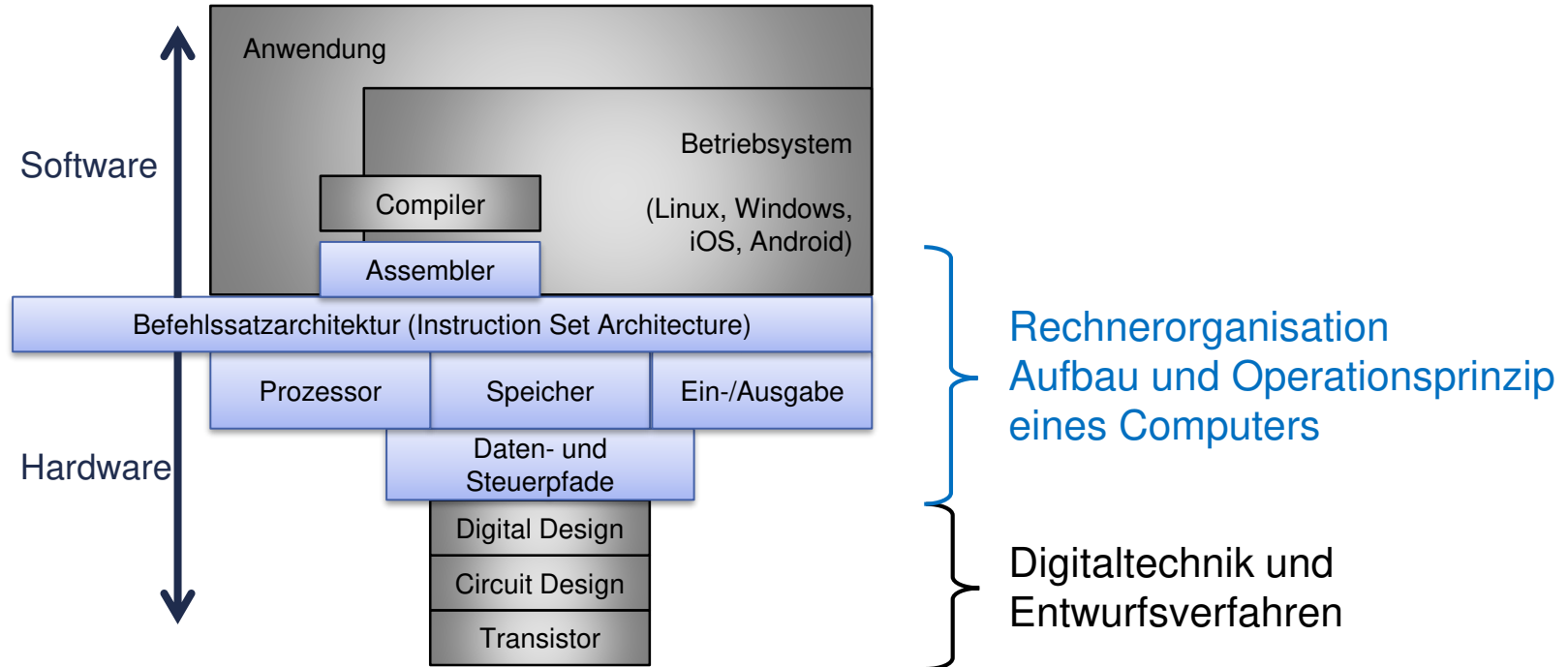
Einsatzgebiete / Rechnermärkte

Internet of Things (IoT)

- Eingebettete Systeme, die über bzw. mit dem Internet verbunden sind, typischerweise drahtlos
- Hohe Anforderungen an die Kommunikationstechnik
- IoT Geräte (Dinge) müssen eindeutig identifizierbar sein, Daten sammeln und mit der physikalischen Welt interagieren
- Intelligente Anwendungsszenarien („smart“ applications):
 - smart homes, smart grids, smart cities, smart cars ...
 - Industrie 4.0
 - Intelligente Vernetzung von Maschinen und Abläufen in der Industrie mit Hilfe von Informations- und Kommunikationstechnologie

Modul „Technische Informatik“

Abstraktionsebenen eines Rechnersystems



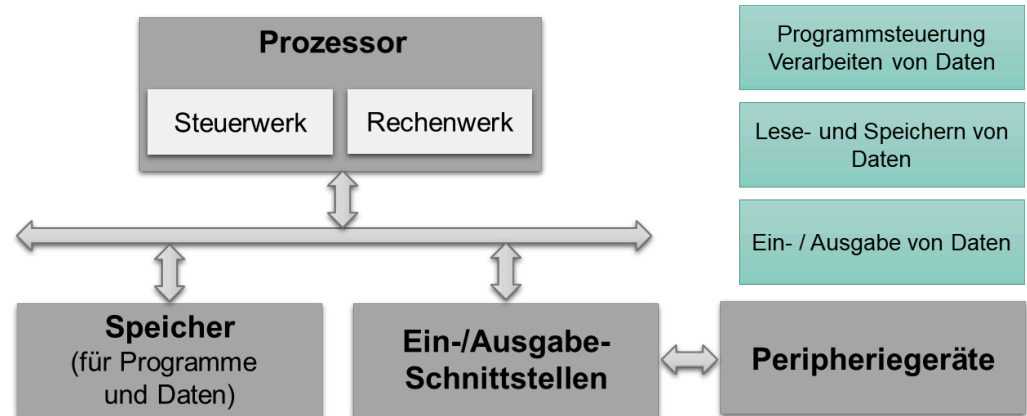
Vorlesung Rechnerorganisation: Inhaltsübersicht

■ Einführung

- Motivation, Überblick über die Vorlesung

■ Grundlegendes Modell des programmierbaren Universalrechners

- Steuerwerk, Rechenwerk
- Speicherwerk
- Ein-Ausgabewerk
- Verbindungsstrukturen
- Maschinenbefehlszyklus



Vorlesung Rechnerorganisation: Inhaltsübersicht

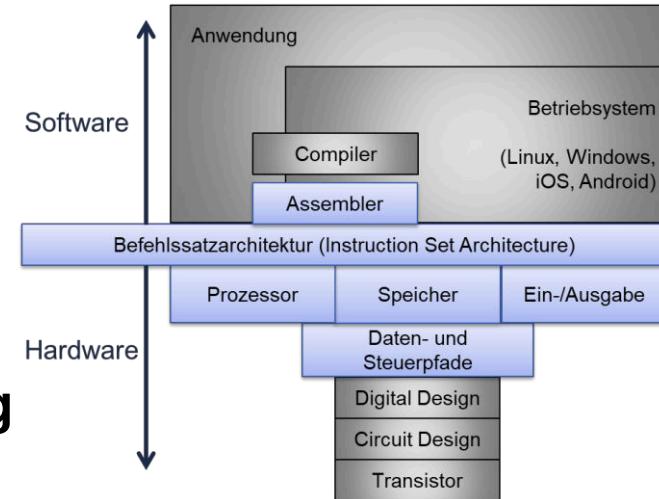
■ Aufbau eines einfachen Mikroprozessors

■ Instruction Set Architektur (ISA)

- Datentypen, Datenformate, Adressierungsarten
- Befehlsformat, Befehlssatz
- Diskussion: RISC & CISC; Fallstudien

■ Einführung in die Assemblerprogrammierung

- Programmieretechniken
- RISC-V-Assembler



Vorlesung Rechnerorganisation: Inhaltsübersicht

■ Prozessororganisation

- Aufbau RISC-V Prozessor
- Pipeline-Organisation des RISC-V-Prozessors

■ Speicher

- Halbleiterspeicher
- Aufbau und Organisation des Hauptspeichers
- Adressierung der Speicherkomponenten

Vorlesung Rechnerorganisation: Inhaltsübersicht

■ Cache-Speicher

- Speicherhierarchie
- Systemaufbau mit Cache-Speicher
- Grundlegendes Operationsprinzip
- Organisationsformen
- Fragen beim Entwurf
- Gültigkeitsproblem

Vorlesung Rechnerorganisation: Inhaltsübersicht

■ Ein-Ausgabe-Werk

- Schnittstellenbausteine
- Periphere Komponenten
- Interrupt-Controller
- Ein-/Ausgabe-Verfahren

■ Verbindungsstrukturen

- Einführung
- Busse
- Punkt-zu-Punkt-Verbindungen

Vorlesung Rechnerorganisation: Inhaltsübersicht

■ Betriebssystemunterstützung

- Unterbrechungsbehandlung
- Speicherverwaltung
- Schutzmechanismen

■ Ausblick

Vorlesung Rechnerorganisation: Inhaltsübersicht

Literatur

- D. Patterson, J. Hennessy:
Computer Organization and Design, RISC-V Edition, The Hardware Software Interface.
Morgan Kaufmann Publ. 2. Auflage 2021
- W. Stallings:
Computer Organization and Architecture - Designing for Performance.
Pearson, 10. Auflage, 2015

Vorlesung Rechnerorganisation: Inhaltsübersicht

Literatur

- Y.N. Patt & S.J. Patel:
Introduction to Computing Systems: From bits & gates to C & beyond, McGrawHill, August 2003
- Uwe Brinkschulte und Theo Ungerer:
Mikrocontroller und Mikroprozessoren
Springer-Verlag, 2. überarbeitete Auflage 2007
- Thomas Flik, H. Liebig:
Mikroprozessortechnik; Springer-Lehrbuch,
7. neu bearbeitete Auflage 2005