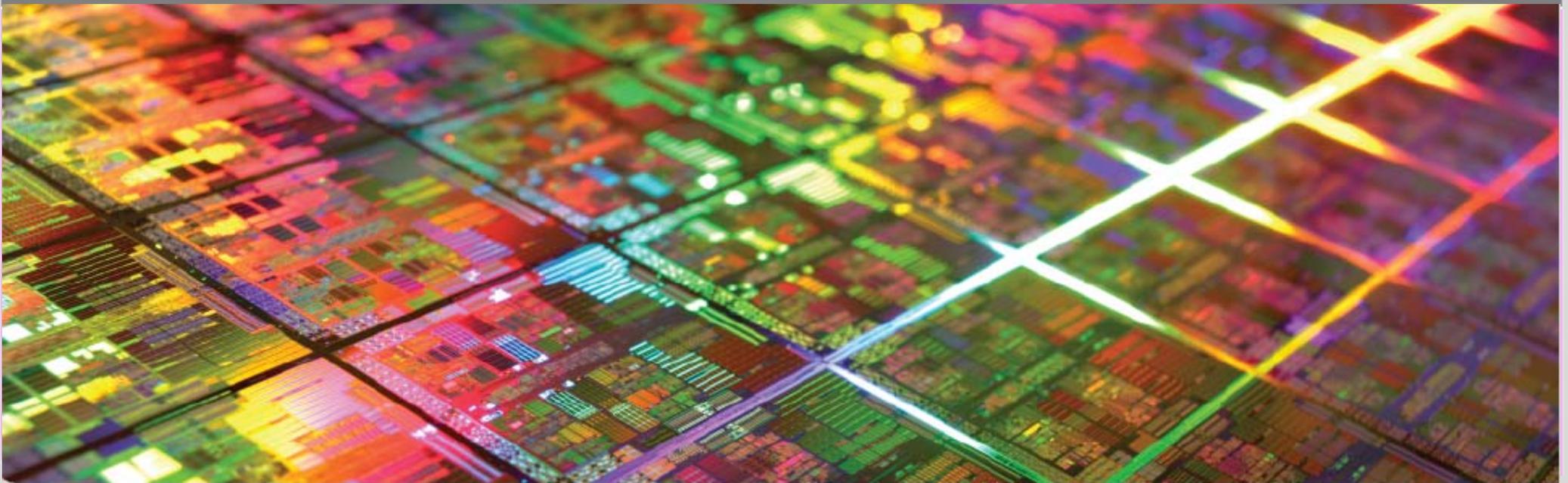


# Rechnerorganisation

Vorlesung im Wintersemester 2017/18

Prof. Dr. Wolfgang Karl

KIT-Fakultät für Informatik – Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung



# Dozenten

## ■ Vorlesung:

Prof. Dr. Wolfgang Karl

- Institut für Technische Informatik der KIT-Fakultät für Informatik, Karlsruher Institut für Technologie
- Büro: Geb. 7.21 Technologiefabrik, Haid-und-Neu-Str. 7, Raum B2 314.1
- Tel.: 0721 608 43771
- Email: karl@kit.edu



## ■ Übung

Dr.-Ing. Ömer Terlemez

- Institut für Anthropomatik der KIT-Fakultät für Informatik, Karlsruher Institut für Technologie
- Büro: Geb. 50.20, Raum 323
- Tel.: 0721 608 45339
- Email: terlemez@kit.edu



# Organisation

## ■ Vorlesung & Übung:

### ■ Montag:

- Ort: Audimax, Geb. 30.95
- Zeit: 14:00 – 15:30 Uhr

### ■ Mittwoch:

- Ort: Hörsaal am Fasanengarten, Geb. 50.35,
- Zeit: 14:00 – 15:30 Uhr

## ■ Aktuelle Termine (TI-Homepage):

<http://ti.ira.uka.de>

# Organisation

## ■ Tutorien

### ■ Ausgabe des 1. Übungsblattes:

- 06. November 2017. Übungsblatt kann von der TI-Homepage heruntergeladen werden (<http://ti.ira.uka.de>)

### ■ Abgabe der Übungsblätter:

- Immer montags bis **13:15 Uhr**, Briefkasten „Rechnerorganisation/Technische Informatik“ im Untergeschoß im Informatik-Hauptgebäude am Fasanengarten (Geb. 50.34)

### ■ Beginn der Tutorien:

- 06. November 2017, nähere Informationen auf der Homepage

# Organisation

## ■ Tutorien

- Einschreibung über das **WebInScribe**:
  - <http://webinscribe.ira.uka.de/>
- Anmeldung:
  - Notwendig: Studenten-Account des Rechenzentrums/SCC („rzstud“)
  - Server steht zur Verfügung von **Dienstag** 17.10.2017, 18.00 Uhr bis **Donnerstag** 29.10.2017, 18.00 Uhr
  - Ergebnisse der Einteilung ab **Freitag** 20.10.2017, 12:00 Uhr auf der WebInScribe Webseite

# Organisation

## Tutorien

### Termine

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
8:00 - 9:30	50.34, <b>SR -108</b> , Kevin Zhang	50.34, <b>SR -107</b> , Nicholas Kjær 50.34, <b>SR 301</b> , Felix Stengel	50.34, <b>SR -107</b> , Niklas Metz 50.34, <b>SR -118</b> , Iris Landerer		50.34, <b>SR -107</b> , Jannik Prinz
9:45 - 11:15				50.34, <b>SR -107</b> , Johannes Werner 50.34, <b>SR -109</b> , Leonard Otto	50.20, <b>SR 148</b> , Gloria Doci 50.34, <b>SR -109</b> , Tarek Wilkening 50.34, <b>SR -108</b> , Alexis Bernhard
11:30 - 13:00					50.34, <b>SR 131</b> , Alexis Bernhard 50.34, <b>SR 236</b> , Paul Kaiser 50.34, <b>SR 301</b> , Tarek Wilkening
14:00 - 15:30	<b>Rechnerorganisation</b>	50.20, <b>SR 148</b> , Adrian Schulte	<b>Rechnerorganisation</b>	50.20, <b>SR 148</b> , Adrian Schulte 50.34, <b>SR 131</b> , Johannes Werner	50.34, <b>SR -118</b> , Paul Kaiser 50.41, <b>SR -133</b> , Maximilian Wagner
15:45 - 17:15			50.34, <b>SR -118</b> , Daniel Schäble		
17:30 - 19:00	50.34, <b>SR -107</b> , Nathanael Funk 50.34, <b>SR -108</b> , Nicholas Kjær	50.34, <b>SR -119</b> , Nathanael Funk	50.34, <b>SR -118</b> , Daniel Meyer	50.34, <b>SR -107</b> , Daniel Meyer 50.34, <b>SR -109</b> , Kansei Hara	

# Organisation

## ■ Tutorien

### ■ Kriterien für den Schein

- Regelmäßige Teilnahme und Bereitschaft zur **aktiven** Mitarbeit in den Tutorien.
- Rechtzeitige Abgabe einer gültigen Ausarbeitung zu mindestens **acht** Übungsblättern.
- **Mindestens 50%** der insgesamt durch die Bearbeitung aller Übungsblätter erreichbaren Punktzahl
- Wer zweimal gegen folgende Bedingung verstößt, erhält keinen Schein:  
Eine Person, die eine korrekt gelöste (Teil-) Aufgabe abgegeben hat, muss auch in der Lage sein, diese im Tutorium vorzurechnen.

# Organisation

## ■ Tutorien

### ■ Bonussystem: Schein

- Sammelt schon während des Semesters Punkte für eure Klausur!
- Es gibt zwei Scheine, einen für DT und einen für RO
- Kriterien für einen Schein erfüllt → 2 Bonuspunkte auf die Klausur
- D. h. maximal 4 Bonuspunkte auf die TI-Klausur
- Bonus verhindert kein Durchfallen in der Klausur, d. h. Bonuspunkte werden nur auf bestandene Klausuren angerechnet

# Organisation

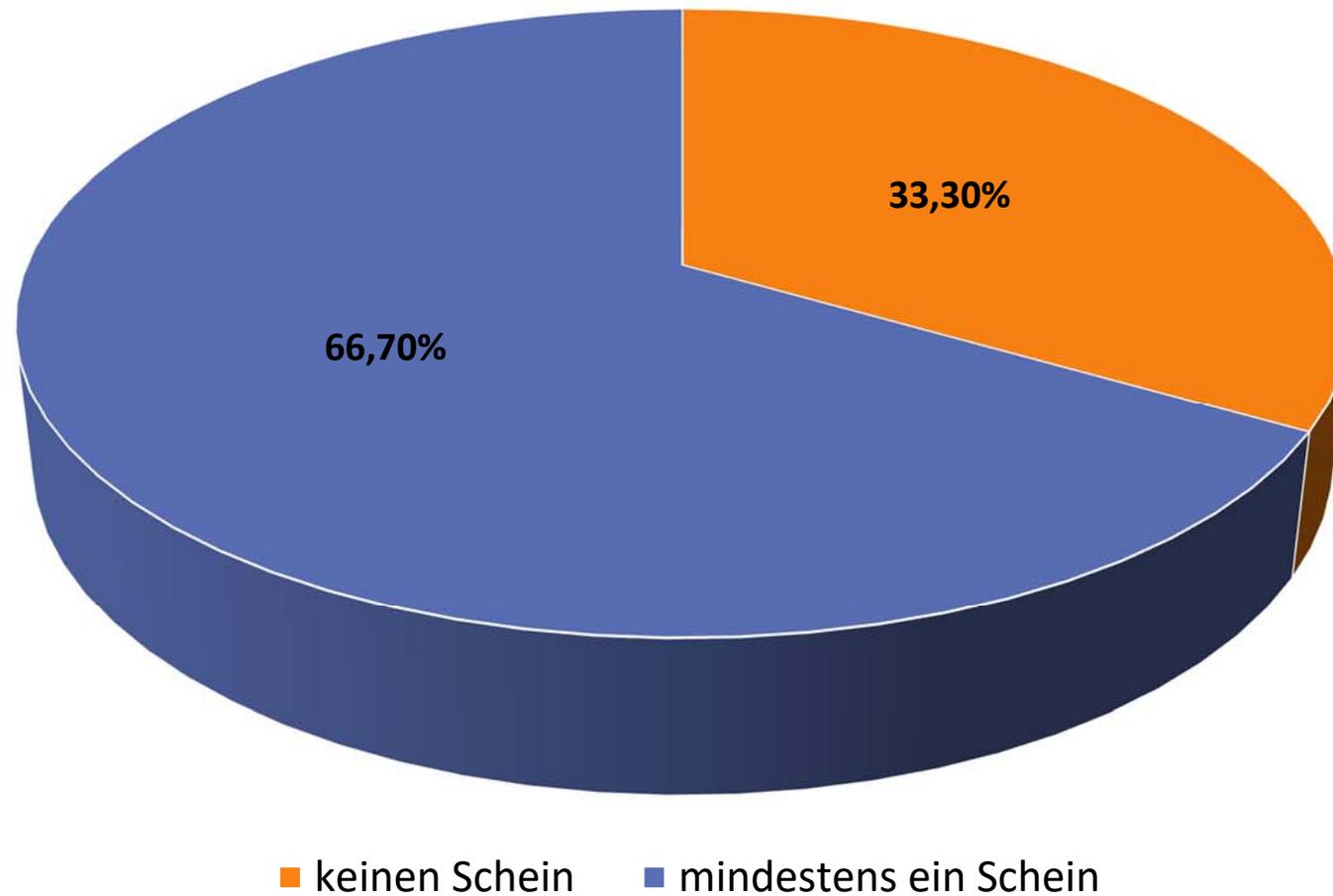
## ■ Klausur

- **Termin:** 23. Februar 2018
- **Nachklausur:** Ende Sommersemester (wschl. Juli 2018)
- Zusätzliche Informationen zu den Klausuren finden sich auf der TI-Homepage (<http://ti.ira.uka.de>)
- Es hat sich in der Vergangenheit auch gezeigt, dass Studierende, die regelmäßig an den Tutorien **teilgenommen** und **einen Übungsschein** erworben haben, erheblich bessere Prüfungsergebnisse erreichen.

# Organisation

## ■ Klausur 2017

Von allen bestandenen Klausuren hatten insgesamt ...



# Organisation

- Folien, Übungsblätter, Termine, ...
  - Abrufbar auf der TI-Webseite:

<http://ti.ira.uka.de>



Benutzername: **ti**

Passwort: **\*\*\*\*\***

*ti # ist # klasse*

# Kapitel 0

## Überblick über die Vorlesung

# Motivation für Modul „Technische Informatik“

Informatiker sollten verstehen, wie  
daten- bzw. informationsverarbeitende Systeme funktionieren



Problemlösungen erfordern sinnvolles  
Zusammenwirken von Hardware und Software



Informatiker sollen kompetent in Software und in Hardware sein

# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Grundlagen der Informationsdarstellung im Rechner
  - Zahlendarstellung im Rechner

Stellwertsystem

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

Euklidischer Algorithmus  
Horner Schema

Einerkomplement

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

Darstellung negativer  
Zahlen

Zweierkomplement

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

Festkommadarstellung

Gleitkommadarstellung

$$X = \pm \textit{Mantisse} \cdot b^{\textit{Exponent}}$$

Darstellung des Kommas

# Modul „Technische Informatik“

- Vorlesung 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

Hammingcodes

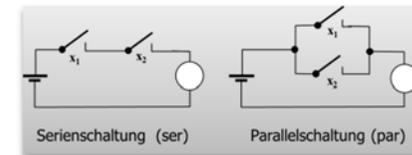
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltnetze
  - Formale Grundlagen

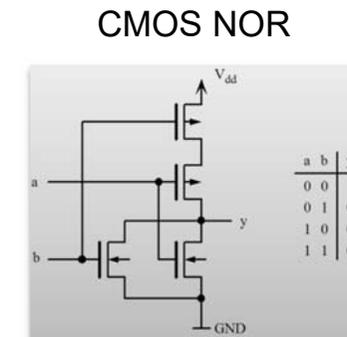
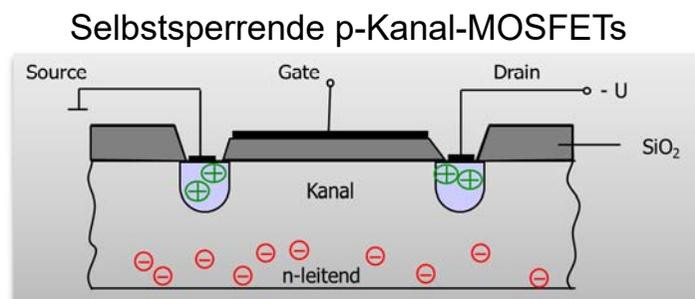
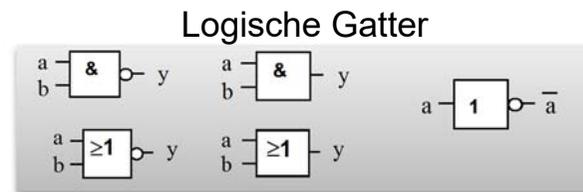
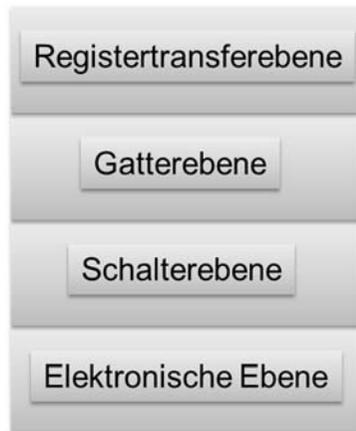
$$BA = [V, \oplus, \otimes]$$

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

Boole'sche Algebra  
Schaltalgebra

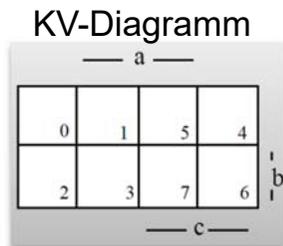


- Realisierung von Schaltnetzen



# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltnetze
  - Minimierungsverfahren



Quine-McCluskey - Verfahren

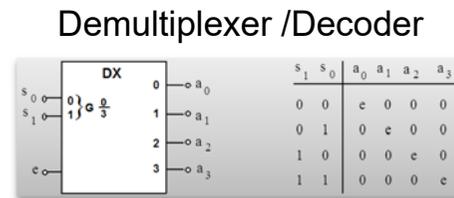
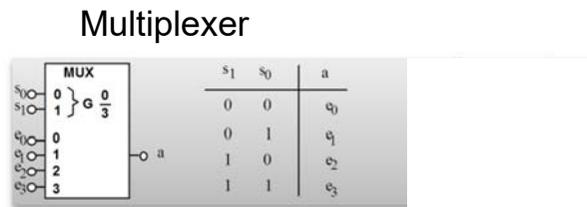
Nr.	0. Ordnung	Nr.	1. Ordnung	Nr.	2. Ordnung
2	00010	2, 6	00-10	2, 6, 10, 14	0--10
4	00100	2, 10	0-010	2, 6, 18, 22	-0-10
5	00101	2, 18	-0010	2, 10, 18, 26	--010
6	00110	4, 6	001-0	4, 5, 12, 13	0-10- A
10	01010	4, 12	0-100	4, 6, 12, 14	0-1-0 B
12	01100	5, 13	0-101	6, 14, 22, 30	--110
18	10010	6, 14	0-110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

Algebraische Verfahren  
 Graphische Verfahren  
 Tabellarische Verfahren

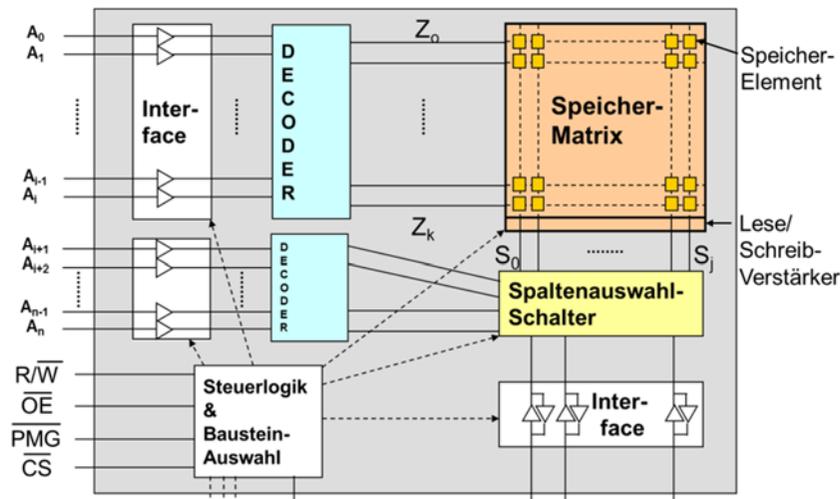
- Logikentwurf
  - Abbildung einer abstrakten logischen Form auf realisierbare Bauelemente.

# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltnetze
  - Komplexere Bausteine



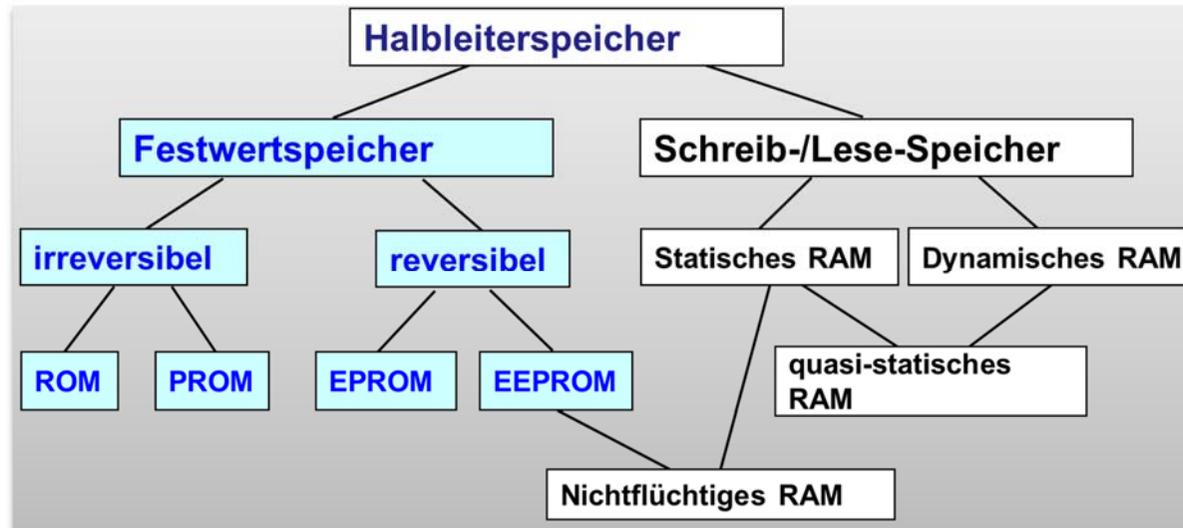
## ■ Speicherkomponenten



Organisation von Speicherbausteinen

# Modul „Technische Informatik“

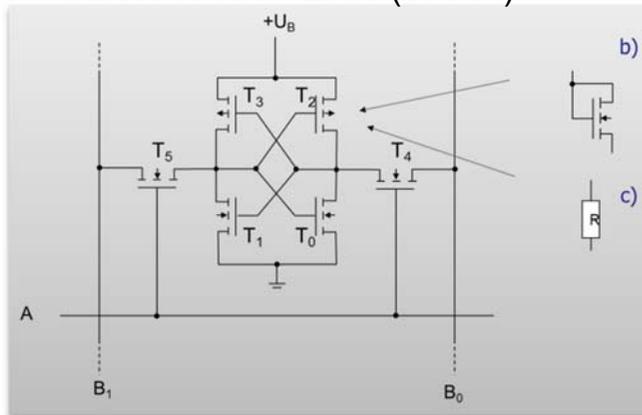
- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltnetze
  - Speichertypen



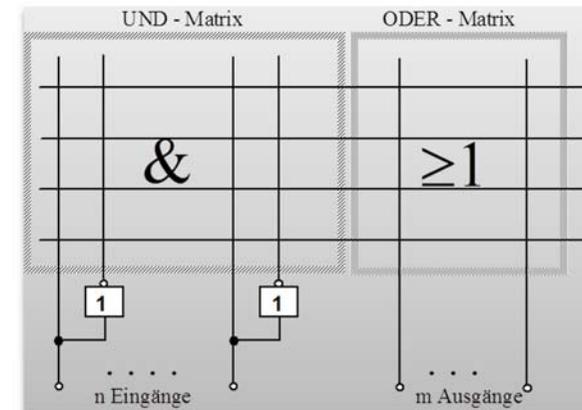
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltnetze
  - Speichertypen

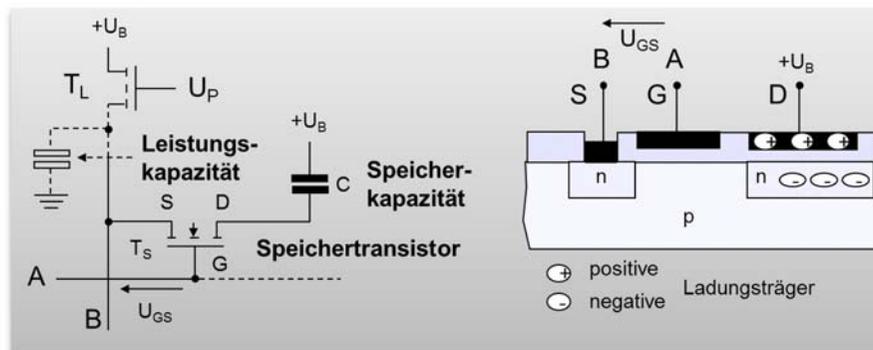
Statische RAM-Zelle (SRAM)



PLA (Programmable Logic Array)



Dynamische RAM-Speicherzelle (DRAM)



# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltwerk
  - Einführung in die Automatentheorie

Ein 6-Tupel  $A = (E, A, Z, \delta, \omega, z_0)$  heißt **Automat**, wenn

q  $E, A$  und  $Z$  nichtleere Mengen sind

- $E$  ist die Menge der **Eingangsbelegungen**  $e$ ,
- $A$  die Menge der **Ausgangsbelegungen**  $a$  und
- $Z$  die Menge der **Zustände**  $z$ .

q **Überföhrungsfunktion**  $\delta : Z \times E \rightarrow Z$

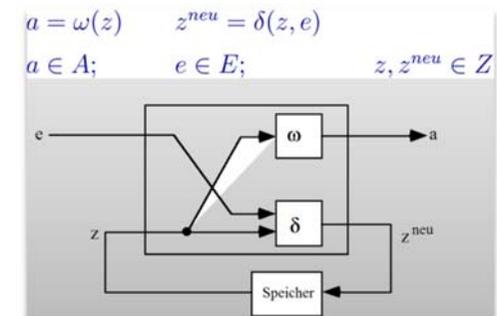
$\delta$  ist eine auf der Menge  $Z \times E$  definierte Funktion, deren Werte in  $Z$  liegen.

q **Ausgabefunktion**  $\omega : Z \times E \rightarrow A$

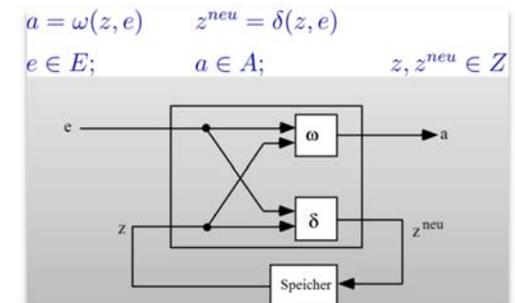
$\omega$  : eine auf der Menge  $Z \times E$  definierte Funktion, deren Werte in  $A$  liegen.

q Grundzustand  $z_0$

Moore-Automat



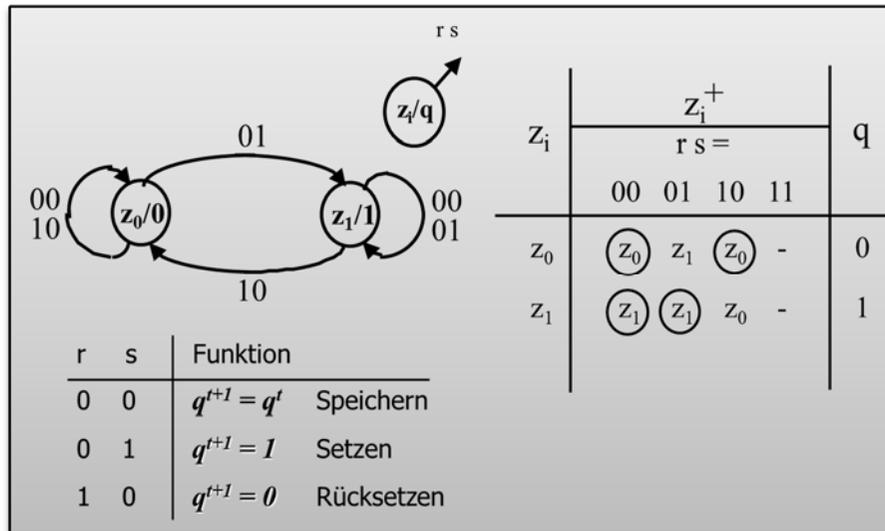
Mealy-Automat



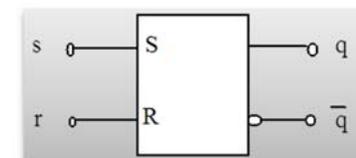
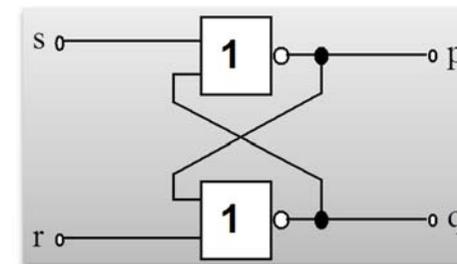
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltwerk
  - Entwurf von Schaltwerken

Entwurf eines einfachen Speichers



Asynchrones RS-Flipflop

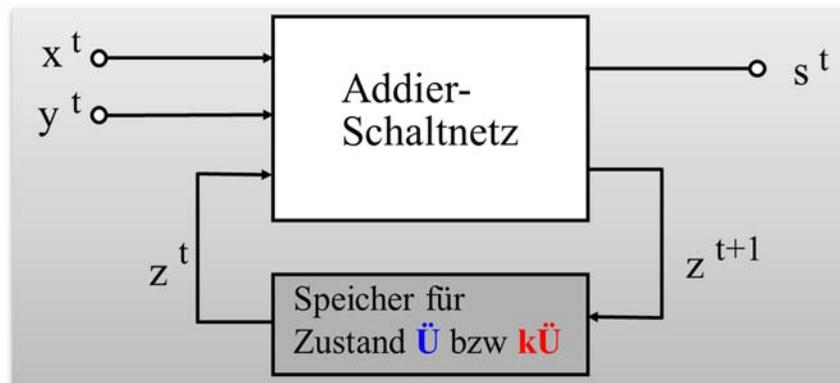


D-Flipflops  
JK-Flipflops  
T-Flipflops

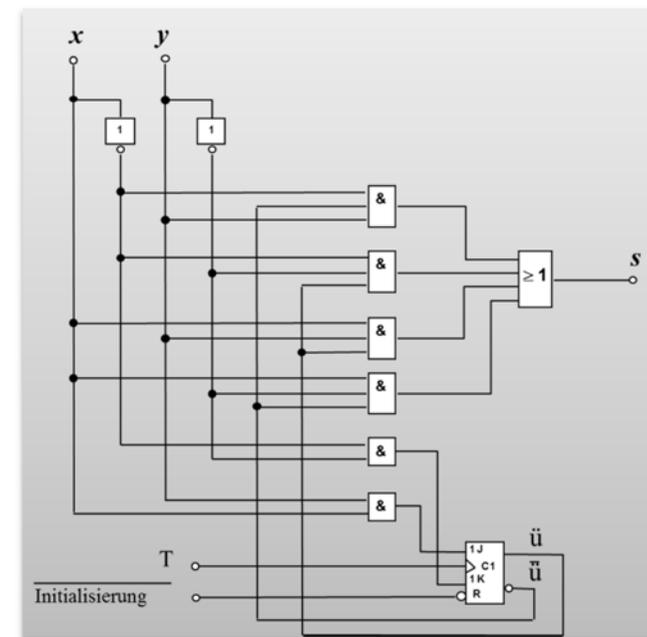
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltwerk
  - Entwurf von Schaltwerken
    - Serienaddierer

Blockschaltbild



Realisierung



$$j^t = (x \cdot y)^t$$

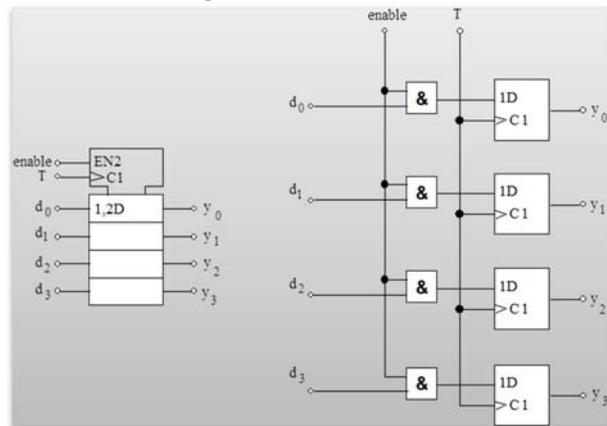
$$k^t = (\bar{x} \cdot \bar{y})^t$$

$$s^t = (x \bar{y} \bar{ü} \vee \bar{x} \bar{y} \bar{ü} \vee \bar{x} y \bar{ü} \vee x y \bar{ü})^t = (x \leftrightarrow y \leftrightarrow \bar{ü})^t$$

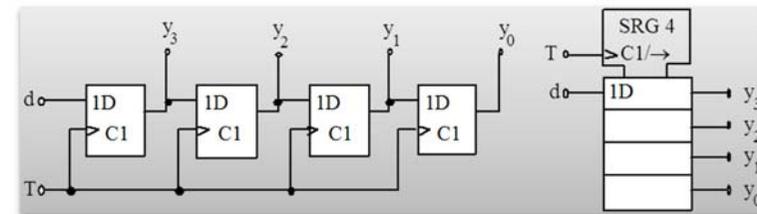
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Schaltwerk
  - Spezielle Schaltwerkbausteine
    - Register

4-Bit-Registers aus D-Flipflops



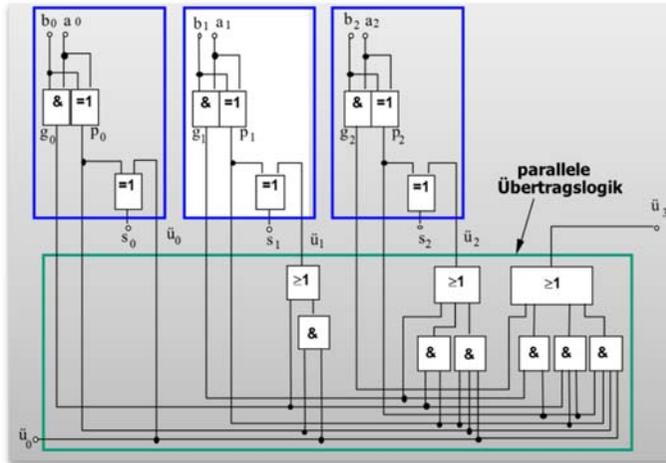
Schieberegister



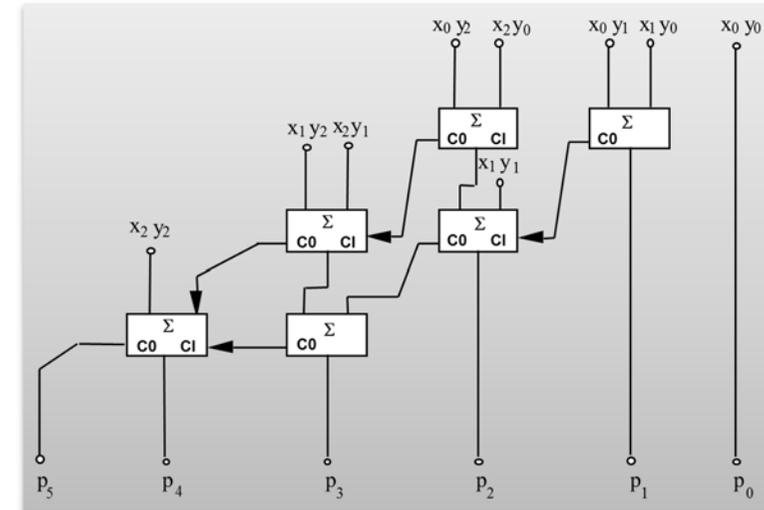
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Rechnerarithmetik
  - Beispiele

3-Bit-Carry-lookahead-Addierer



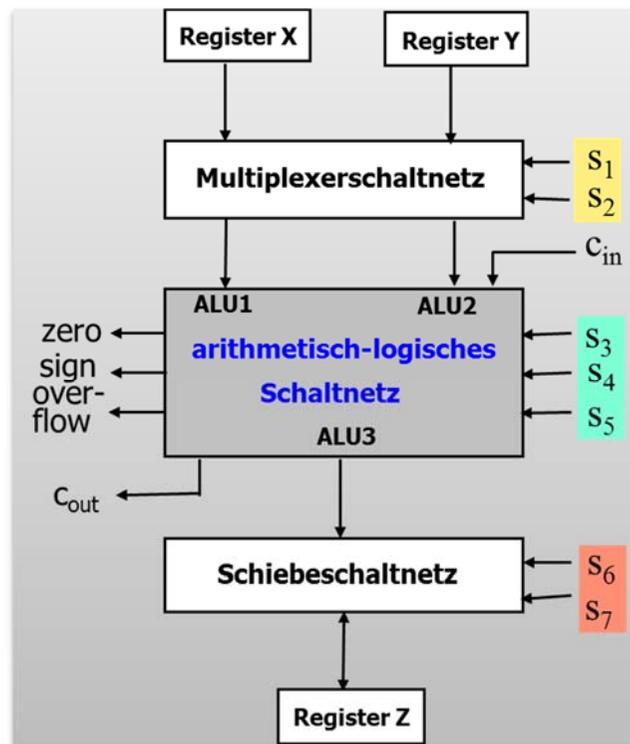
3-Bit-Parallelmultiplizierer



$$\begin{aligned}
 P &= (x_0 \cdot 2^0 + x_1 \cdot 2^1 + x_2 \cdot 2^2)(y_0 \cdot 2^0 + y_1 \cdot 2^1 + y_2 \cdot 2^2) \\
 &= 2^0 x_0 y_0 + 2^1 (x_0 y_1 + x_1 y_0) + 2^2 (x_0 y_2 + x_1 y_1 + x_2 y_0) + \\
 &\quad 2^3 (x_1 y_2 + x_2 y_1) + 2^4 x_2 y_2 \\
 &= 2^0 p_0 + 2^1 p_1 + 2^2 p_2 + 2^3 p_3 + 2^4 p_4 + 2^5 p_5
 \end{aligned}$$

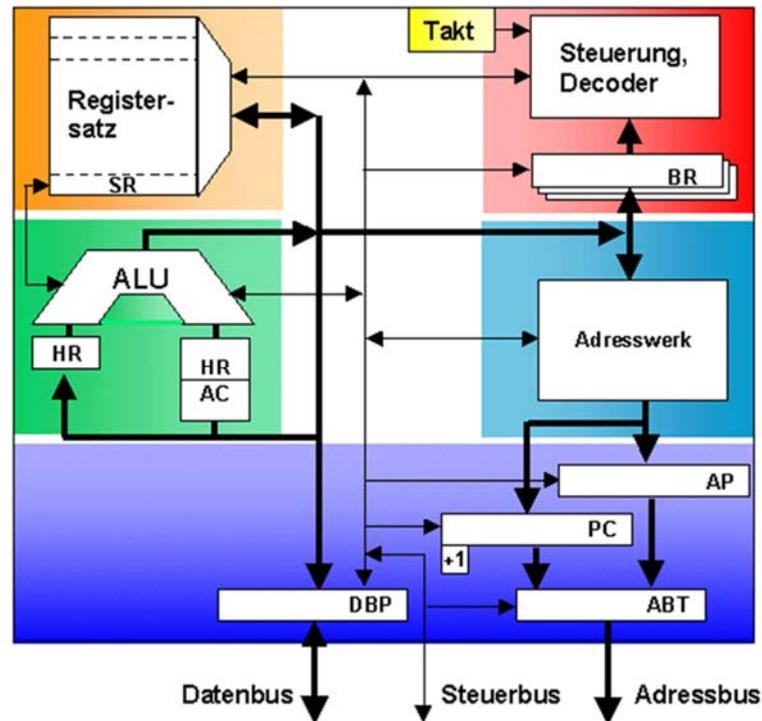
# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Rechnerarithmetik
  - Arithmetische-logische Einheit

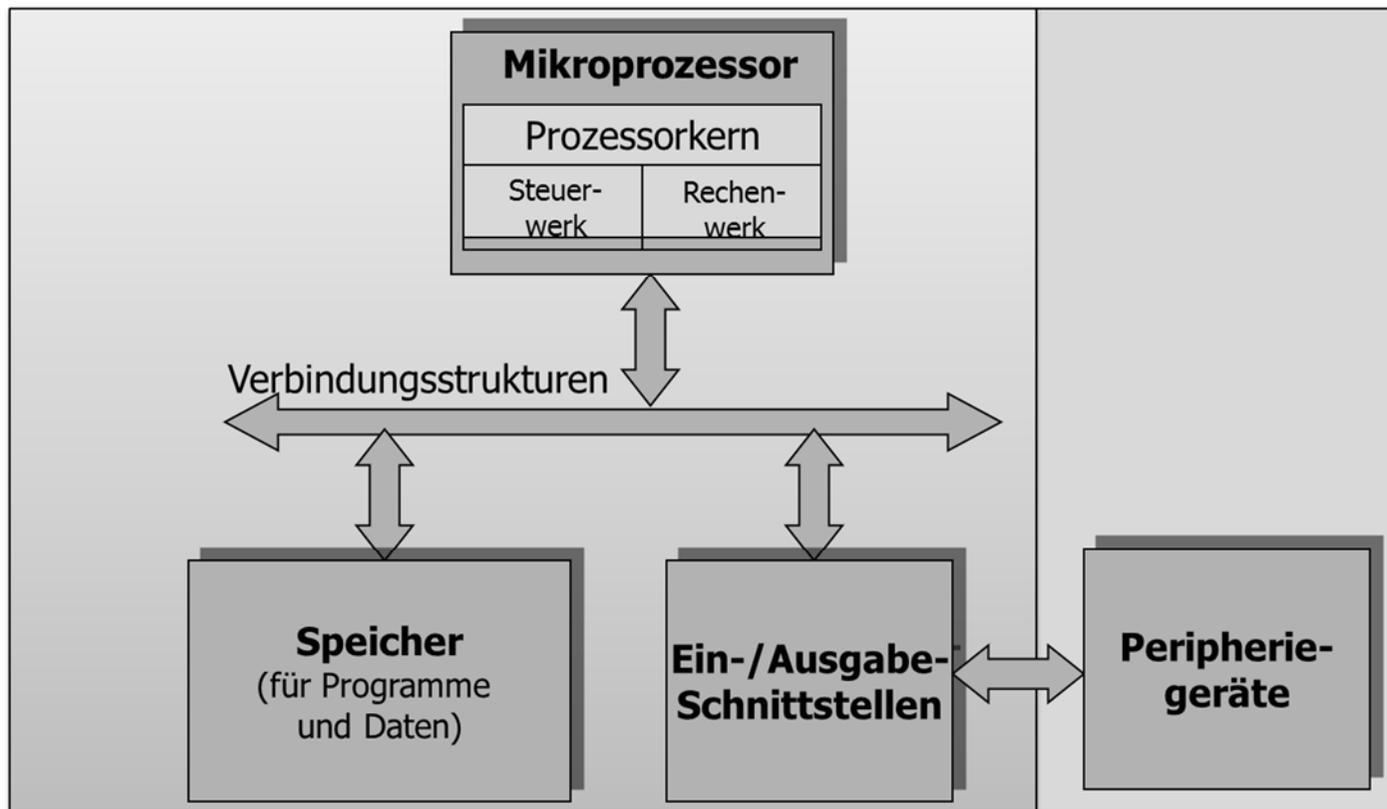


# Modul „Technische Informatik“

- Vorlesung Digitaltechnik und Entwurfsverfahren
- Einfacher Mikroprozessor



## ■ Mikroprozessorsystem



### Grundfunktionen:

- Ein- und Ausgabe von Daten
- Verarbeiten von Daten
- Speichern von Daten

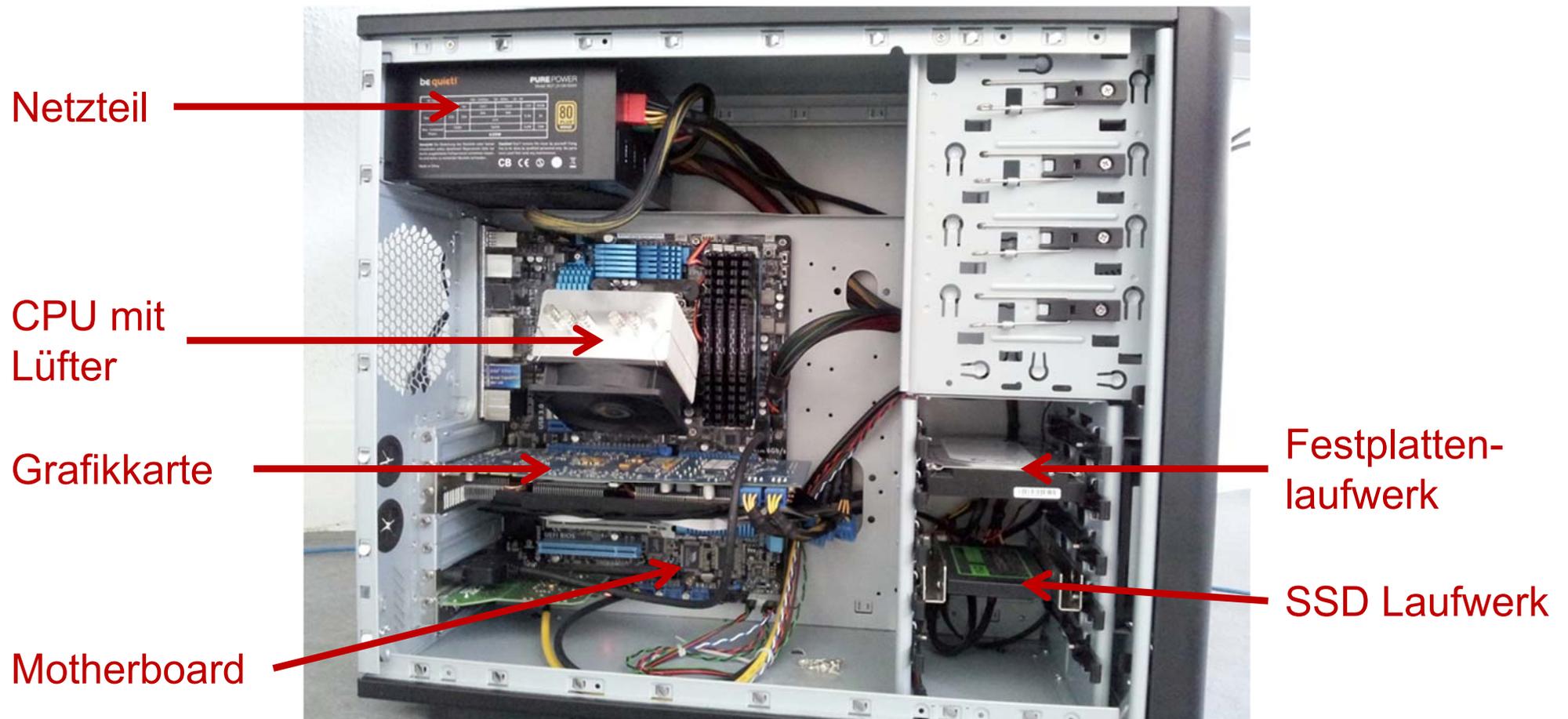
# Modul „Technische Informatik“

## ■ Arbeitsplatzrechner (Desktop)

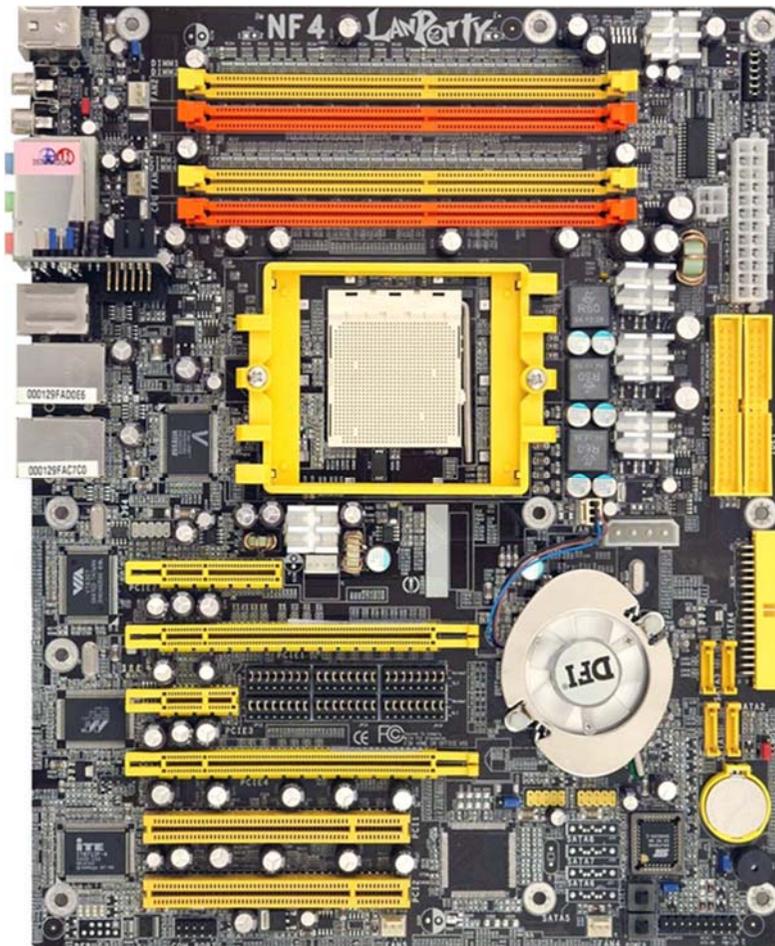


# Modul „Technische Informatik“

## ■ Arbeitsplatzrechner (Desktop): Öffnen des Gehäuses



## ■ Arbeitsplatzrechner (Desktop): PC Motherboard



Hauptspeicher

Prozessor

Grafikeinheit

Prozessor  
Schnitt-  
stelle

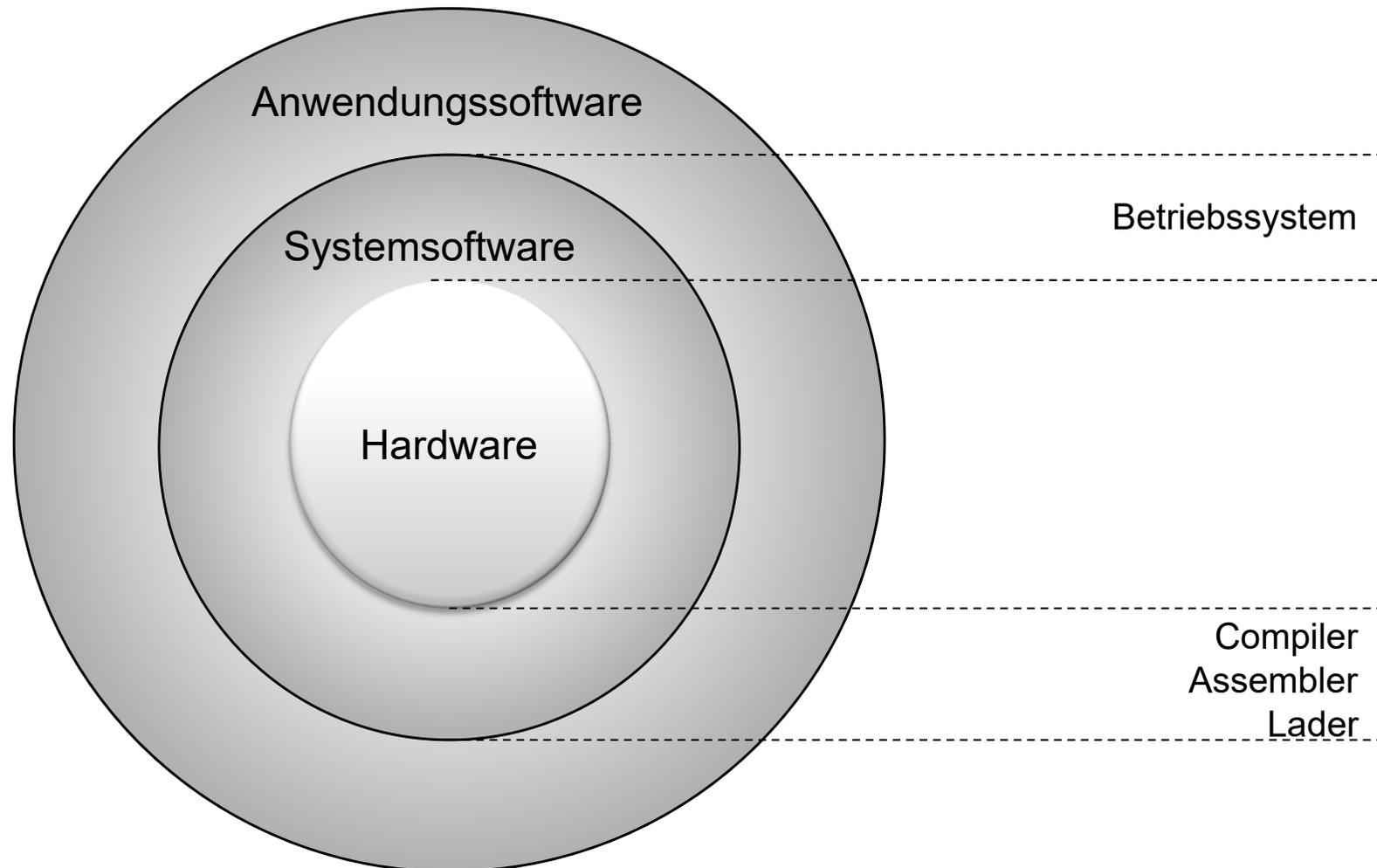
Steckplätze für  
Ein-/Ausgabe

Laufwerk  
und USB

## Thema der Technischen Informatik



- Vereinfachte Sicht von Hardware und Software als hierarchische Ebenen



# Modul „Technische Informatik“

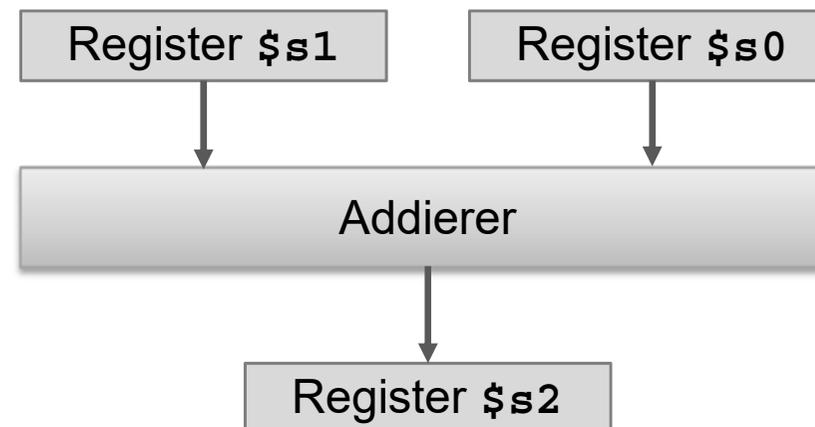
## ■ Abstraktionsebenen eines Rechnersystems



# Modul „Technische Informatik“

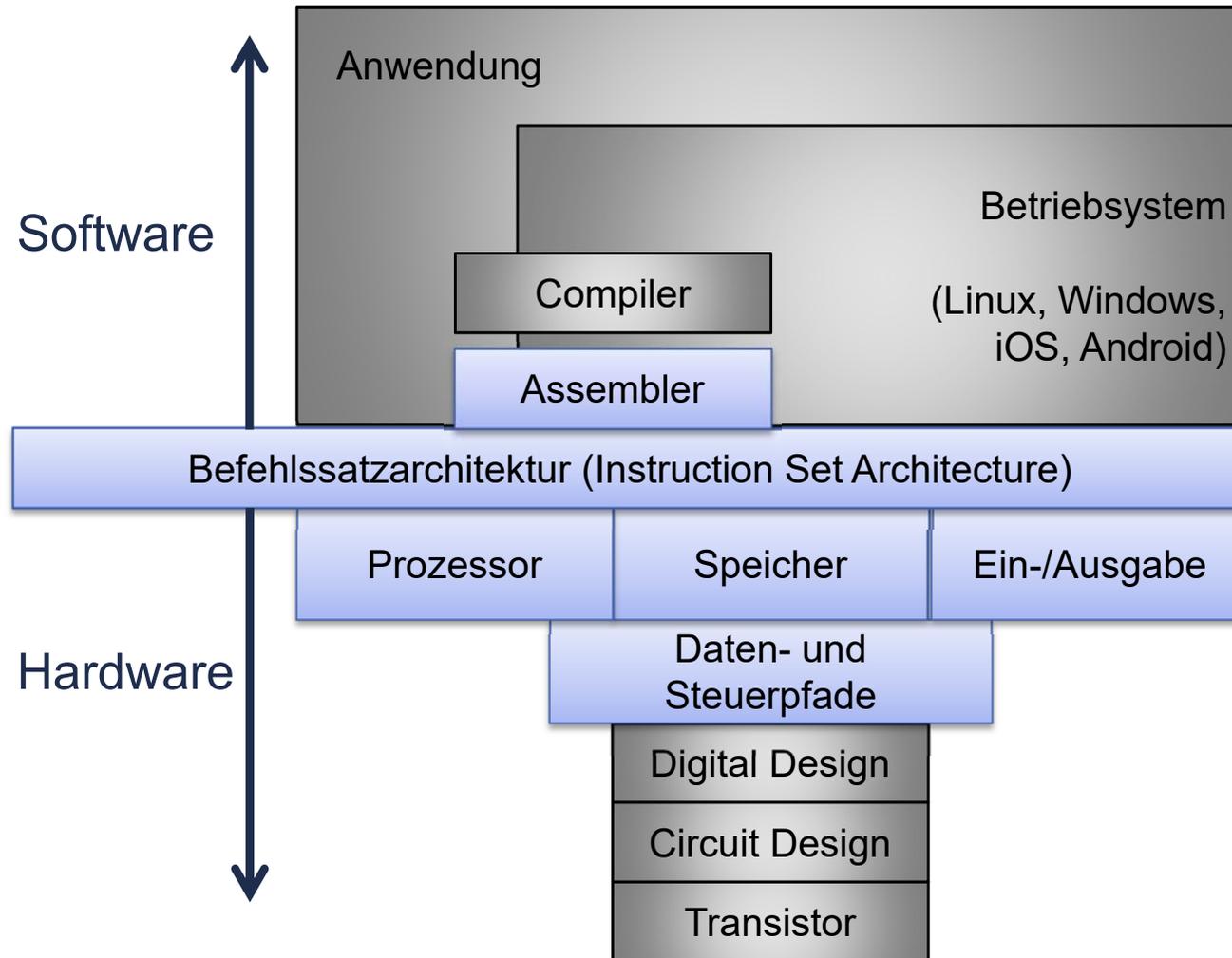
## ■ Von der höheren Programmiersprache zur Sprache der Hardware

- C – Anweisung: `int sum = a + b`
- Assemblerbefehl: `add $s2, $s1, $s0`
- Maschinenbefehl: `1000110011010000`
- Hardware



# Modul „Technische Informatik“

## ■ Vorlesung Rechnerorganisation



# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation

- **Einführung**
  - Motivation, Historische Anmerkungen
  
- **Anforderungen höherer Programmiersprachen**
  - Programmkonstrukte
  - Variable und Konstante
  
- **Ein grundlegendes Rechnermodell**
  - Steuerwerk, Rechenwerk
  - Speicherwerk
  - Ein-Ausgabewerk
  - Verbindungsstrukturen
  - Maschinenbefehlszyklus

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation

- **Instruction Set Architektur (ISA)**
  - Datentypen, Datenformate, Adressierungsarten
  - Befehlsformat, Befehlssatz
  - Diskussion: RISC & CISC; Fallstudien (MIPS)
  
- **Einführung in die Assemblerprogrammierung**
  - Programmiertechniken
  - MIPS-Assembler
  
- **Befehlsabarbeitung im grundlegenden Rechnermodell**
  - Logische Phasen des Maschinenbefehlszyklus
  - Pipelining

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation

- **Prozessorarchitektur**
  - Begriffe und Definitionen
  - Aufbau Steuerwerk
  - Aufbau Rechenwerk
  
- **Speicher**
  - Speicherkomponenten
  - Adressierung von Systemkomponenten
  - Speicherhierarchie
  - Cache-Speicher
    - Organisation
    - Cache-Kohärenz

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation

### ■ Betriebssystemunterstützung

- Speicherverwaltung
- Unterbrechungsbehandlung

### ■ Ein-Ausgabe

- Schnittstellenbausteine
- DMA
- Interrupt-Controller

### ■ Bussysteme

- Grundlegende Eigenschaften, Definitionen
- Zuteilung, Protokolle
- Fallstudien

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation

### ■ Betriebssystemunterstützung

- Speicherverwaltung
- Unterbrechungsbehandlung

### ■ Ein-Ausgabe

- Schnittstellenbausteine
- DMA
- Interrupt-Controller

### ■ Bussysteme

- Grundlegende Eigenschaften, Definitionen
- Zuteilung, Protokolle
- Fallstudien

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation: Literatur

- D. Patterson, J. Hennessy:  
**Computer Organisation & Design, The Hardware / Software Interface**  
Morgan Kaufmann Publ. 4. Auflage 2009
- D. Patterson, J. Hennessy:  
**Computer Organization and Design, Enhanced**  
Morgan Kaufmann Publ. 5. Auflage 2017
- D. Patterson, J. Hennessy:  
**Computer Organization and Design, MIPS Edition**  
Morgan Kaufmann Publ. 5. Auflage 2013
- D. Patterson, J. Hennessy:  
**Computer Organization and Design, ARM Edition**  
Morgan Kaufmann Publ. 1. Auflage 2016

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation: Literatur

- Uwe Brinkschulte und Theo Ungerer:  
**Mikrocontroller und Mikroprozessoren**  
Springer-Verlag, 2. überarbeitete Auflage 2007
- H. Bähring:  
**Mikrorechner-Systeme**, Springer-Lehrbuch,  
3. Auflage 2002 (Band I/II)
- Thomas Flik, H. Liebig:  
**Mikroprozessortechnik**; Springer-Lehrbuch,  
7. neu bearbeitete Auflage 2005
- W. Oberschelp, G. Vossen:  
**Rechneraufbau und Rechnerstrukturen**, 9. Auflage, Oldenbourg 2003

# Modul „Technische Informatik“

## Vorlesung Rechnerorganisation: Literatur

- Y.N. Patt & S.J. Patel:  
**Introduction to Computing Systems: From bits & gates to C & beyond,**  
McGrawHill, August 2003
  
- W. Stallings  
**Computer Organization and Design Designing for Performance**  
Pearson, 9. Auflage, 2013

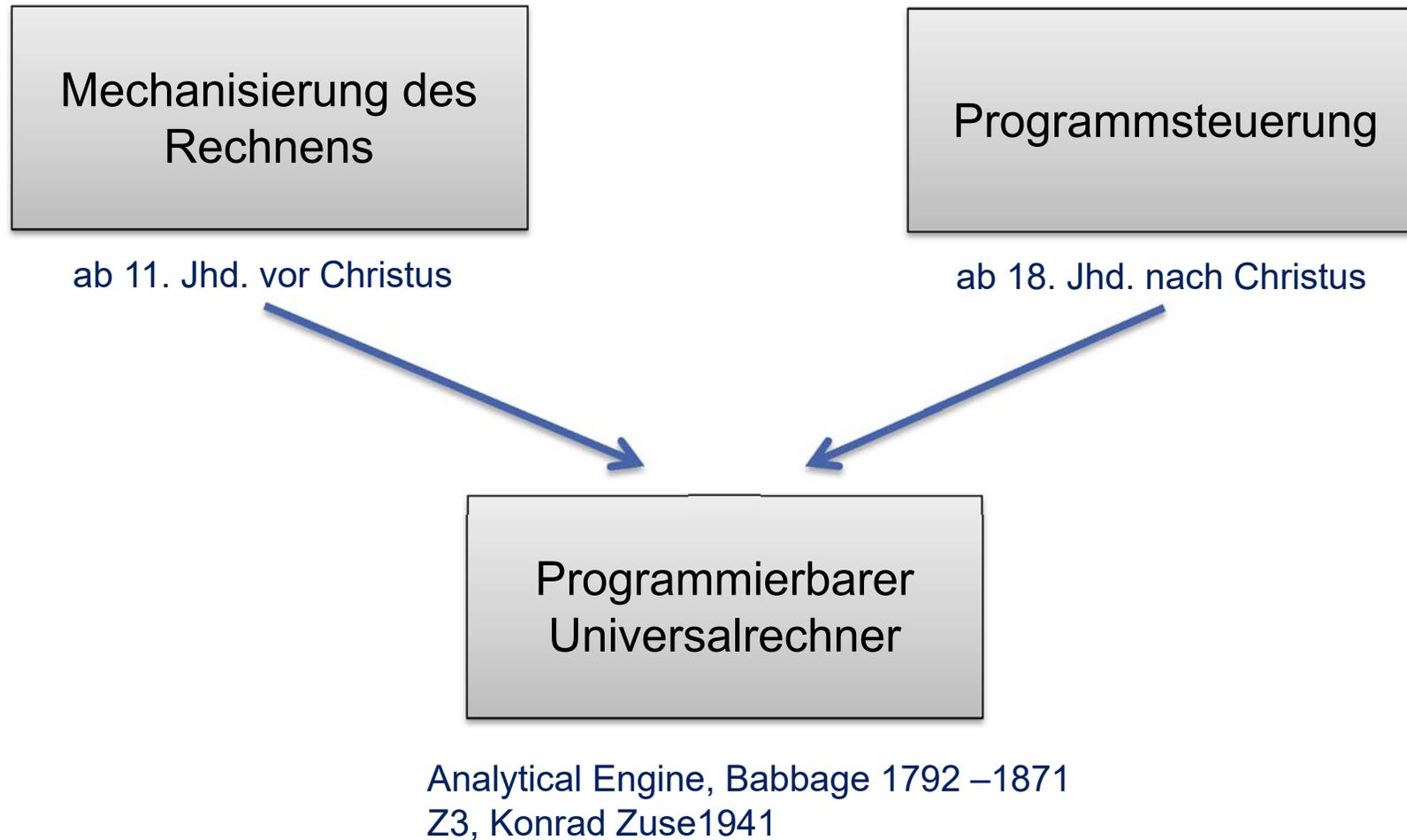
# Kapitel 1

## 1.1 Historische Anmerkungen

## 1.2 Trends in der Rechnerarchitektur

# Ein Blick zurück ...

## Genealogie des programmierbaren Universalrechners

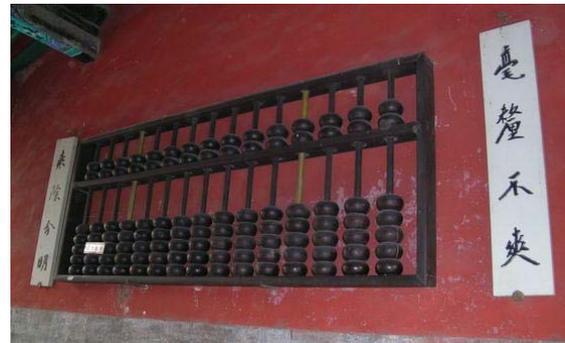


# Ein Blick zurück ...

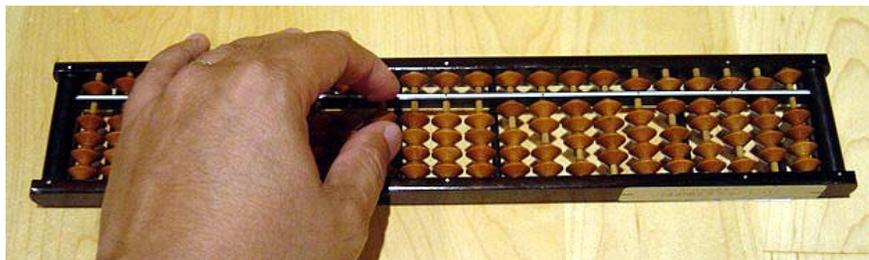
## Genealogie des programmierbaren Universalrechners

### Mechanisierung des Rechnens

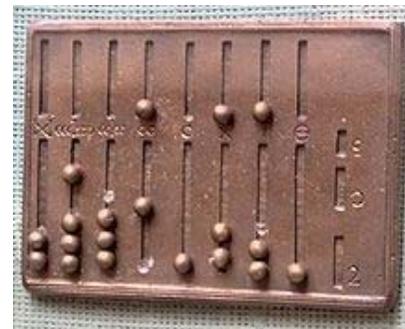
ab 11. Jhd. vor Christus



Chinesischer Suan Pan



Japanischer Soroban



Rekonstruktion eines römischen Abakus



Russische Stschoty

Quelle: Wikipedia

# Ein Blick zurück ...

## Genealogie des programmierbaren Universalrechners

### Mechanisierung des Rechnens

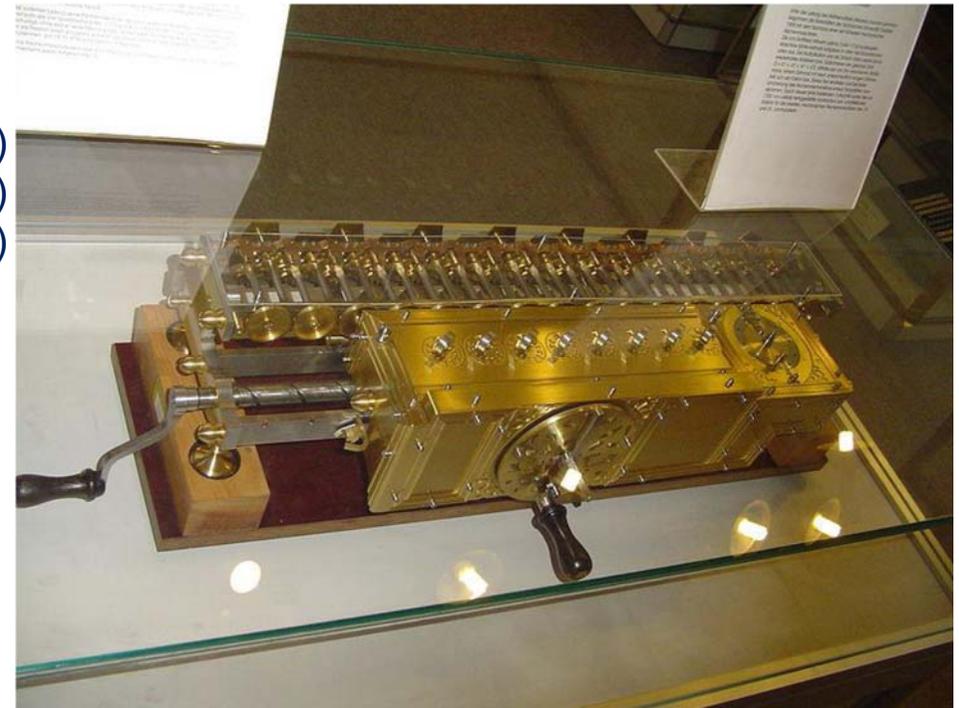
ab 17. Jhd. n. Chr.: Vierspezies Rechenmaschine

Schickard(1592 -1635)

Pascal (1629 -1662)

Leibniz (1646 -1716)

Leibniz: Idee einer auf dem dualen Zahlensystem basierenden „Machina Arithmeticae Dyadicae“



Quelle: Wikipedia

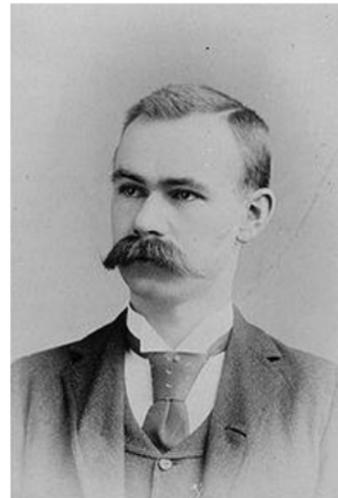
# Ein Blick zurück ...

## Genealogie des programmierbaren Universalrechners

Programmsteuerung



Ab 18. Jhd. Lochkarten-  
gesteuerter Webstuhl  
Jacquard (1752 –1834)



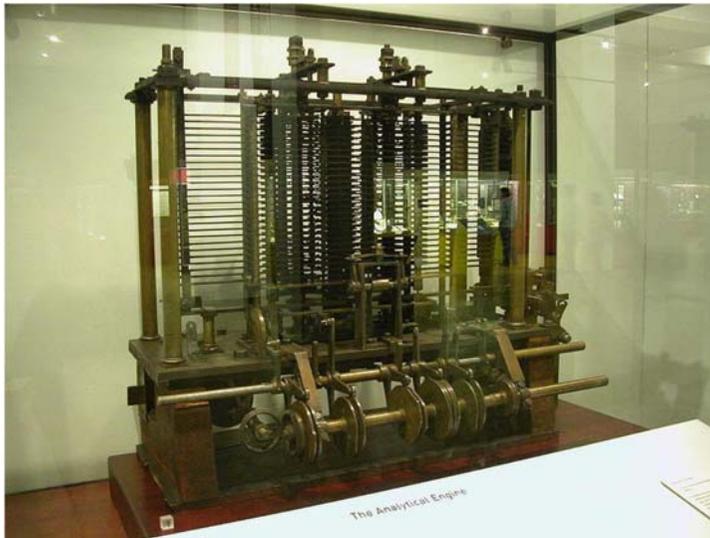
Hollerith (1860 –1929)  
1889: Patent auf Tabelliermaschine:  
Lochkartenstanzer, -leser, -sortierer



Quelle: Wikipedia

# Ein Blick zurück ...

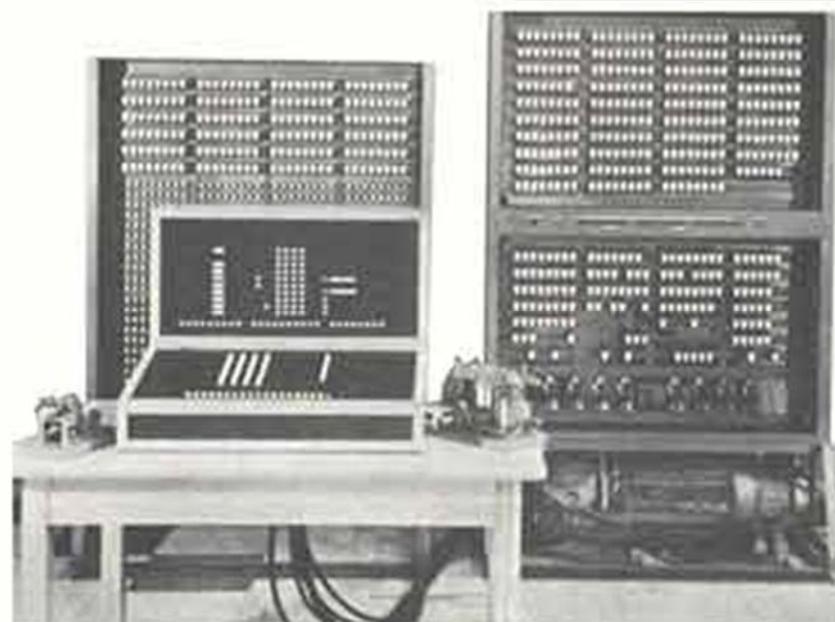
## Genealogie des programmierbaren Universalrechners



Quelle: Wikipedia

Versuchsmodell der Analytical Engine  
Babbage (1791 -1871)

Programmierbarer  
Universalrechner



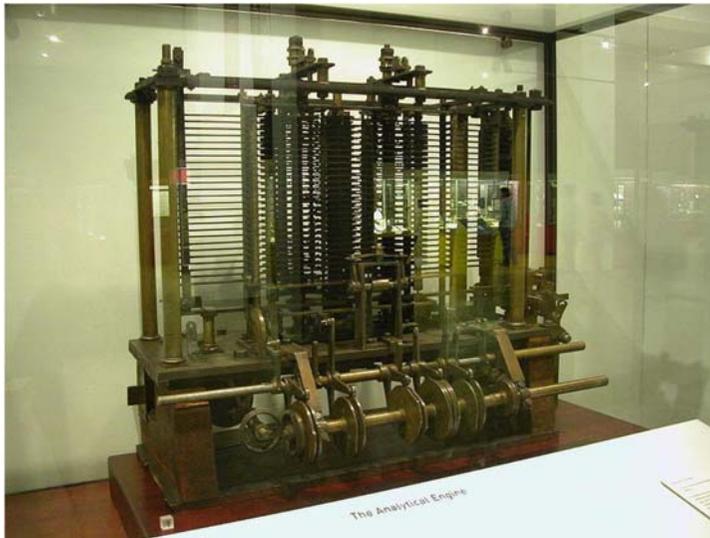
Z3: Konrad Zuse (1910 – 1995): programmierbarer  
Rechner (voll funktionsfähig)



Quelle: Horst Zuse

# Ein Blick zurück ...

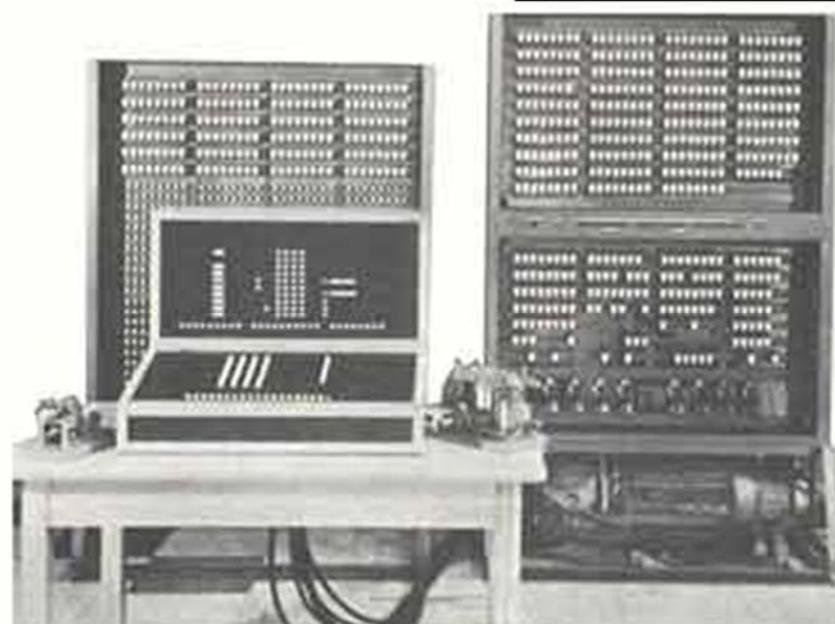
## Genealogie des programmierbaren Universalrechners



Quelle: Wikipedia

Versuchsmodell der Analytical Engine  
Babbage (1791 -1871)

Programmierbarer  
Universalrechner



Z3: Konrad Zuse (1910 – 1995): programmierbarer  
Rechner (voll funktionsfähig)

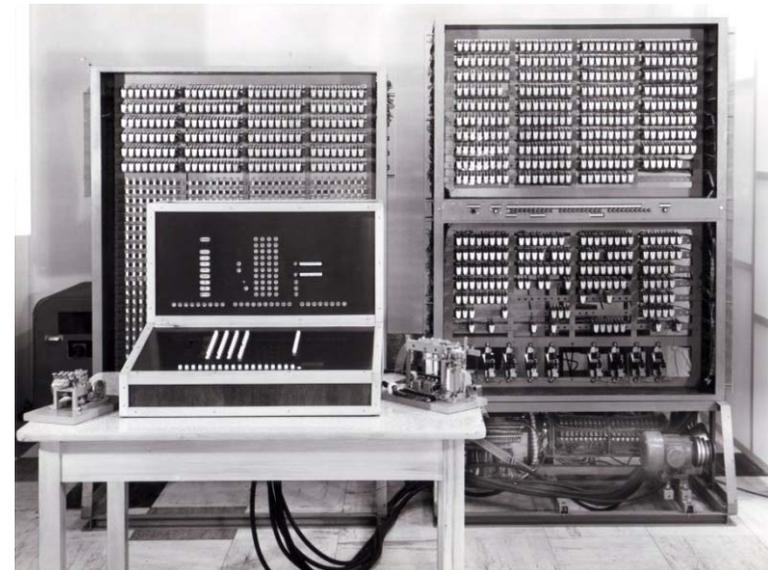


Quelle: Horst Zuse

# Ein Blick zurück ...

## Zuse Z3 (1941):

- „der erste funktionsfähige, frei programmierbare, auf dem binären Zahlensystem (Gleitkommazahlen) und der binären Schaltungstechnik basierende Rechner der Welt.“
- Speicherkapazität:
  - 64 Worte zu je 22 Bit
- 4 Grundrechenarten, Quadratwurzel
  - 1 Addition benötigte 3 Takte
  - 1 Multiplikation: ~ 3s
- Taktfrequenz: 5 Hz
- Rechenleistung: ~ 1 Gleitkommaoperation pro Sekunde

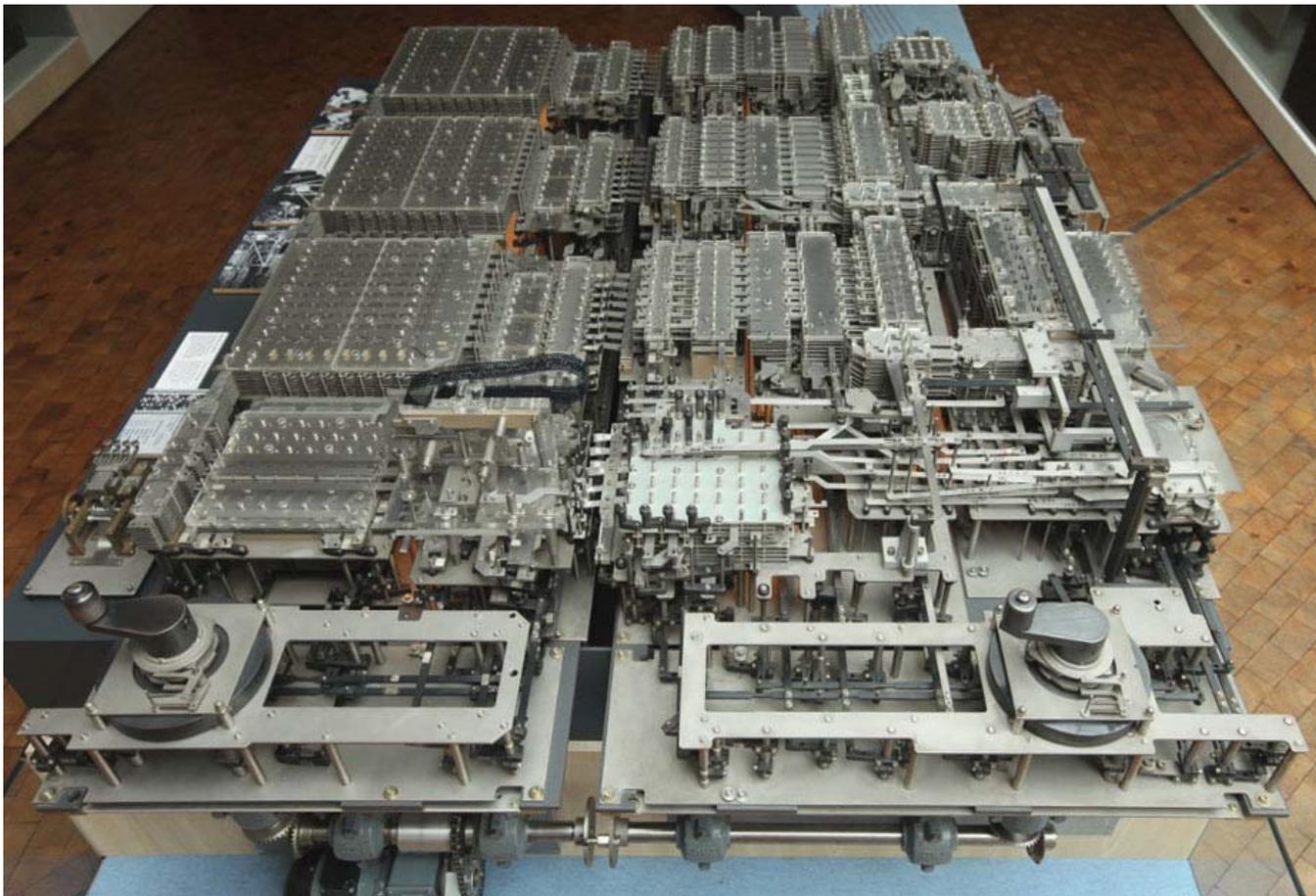


Quelle: H.Zuse, [www.zuse.de](http://www.zuse.de)

# Ein Blick zurück ...

## Zuse Z1 (1936):

- Die Z1 im Deutschen Technikmuseum Berlin

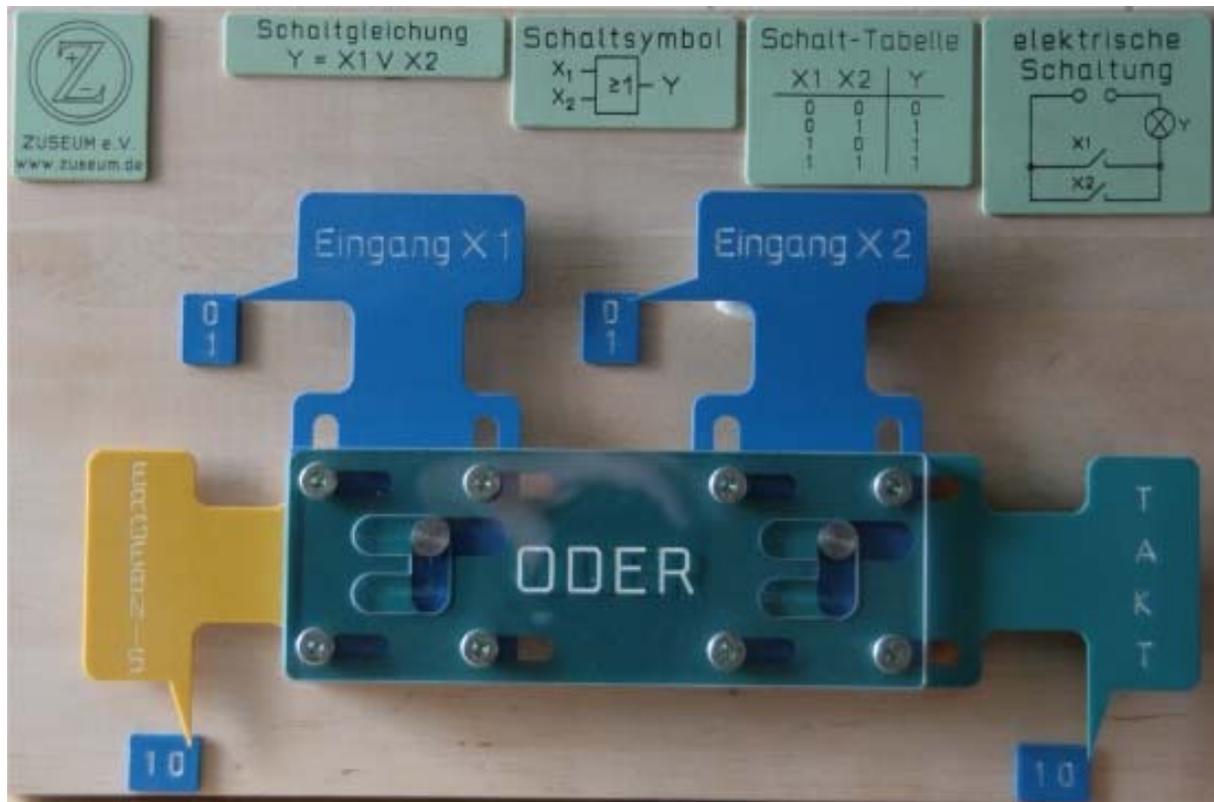


Quelle: Freie Universität Berlin

# Ein Blick zurück ...

## Zuse Z1 (1936):

- Mechanischer Rechner
  - Logische ODER-Funktion



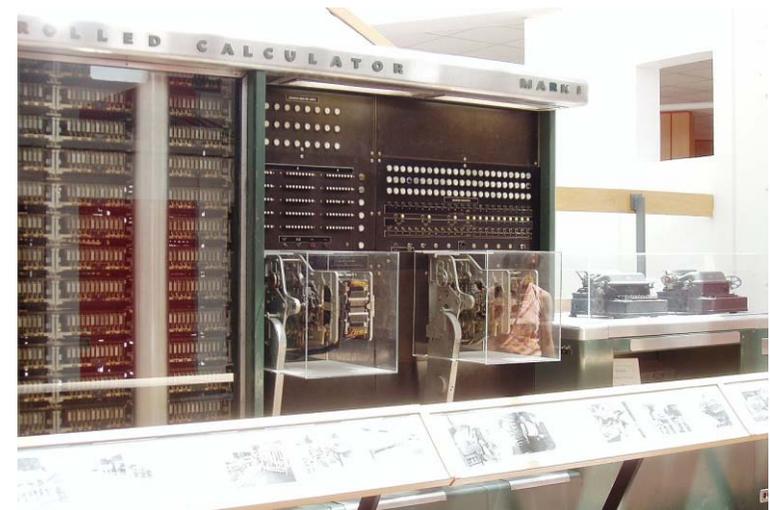
Quelle: H.Zuse, [www.zuse.de](http://www.zuse.de)

# Ein Blick zurück ...

## MARK I (1944)

### ■ Howard Aiken

- Erster programmgesteuerter Rechenautomat der USA
- Dezimales Zählrad-Prinzip
  - Rechenelemente und Speicher sind als dekadische Zählräder realisiert
- Sehr große Maschine
  - 17 m lang, 2,5 m breit
  - 760.000 Einzelteile (u.a. 3000 Kugellager, 80 km Leitungsdraht)
- Ein- und Ausgabe:
  - gelochte Papierbänder benutzt
- Programm:
  - 24-spuriger Lochstreifen

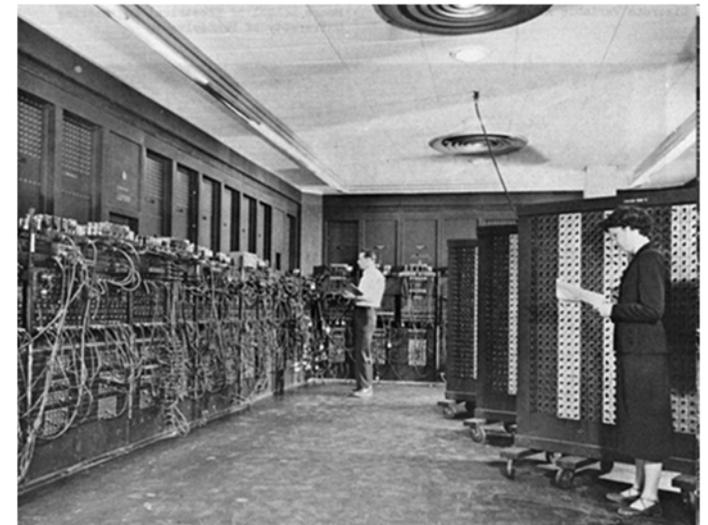


Quelle: Wikipedia

# Ein Blick zurück ...

## ENIAC 1946

- J. Presper Eckert und John W. Mauchly
  - 40 parallel arbeitende Komponenten
    - Jeweils 60 cm breit, 270 cm hoch, 70 cm tief
    - Aufgebaut in U-Form (Fläche 10m x 17m)
    - Gewicht: 70 Tonnen
  - Erster elektronischer Universalrechner
    - Röhrentechnologie
      - 17.468 Elektronenröhren
      - ....
  - Operationen:
    - Addition ( $\sim 0,2$  ms), Subtraktion, Multiplikation ( $\sim 2,8$  ms), Division, Quadratwurzel
    - 10-stellige Dezimalzahlen
  - Programmierung:
    - Verbinden der Komponenten mit Kabeln



Quelle: Wikipedia

# Ein Blick zurück ...

## Vergleich der ersten Computer

*Eigenschaften der ersten Computer*

Computer	Land	Inbetriebnahme	Gleitkomma-arithmetik	Binär	Elektronisch	Programmierbar	Turingmächtig
Zuse Z3	Deutschland	Mai 1941	Ja	Ja	Nein	Ja, durch Lochstreifen	Ja, ohne Praxisnutzen
Atanasoff-Berry-Computer	USA	Sommer 1941	Nein	Ja	Ja	Nein	Nein
Colossus	UK	1943	Nein	Ja	Ja	Teilweise, durch Neuverkabelung	Nein
Mark I	USA	1944	Nein	Nein	Nein	Ja, durch Lochstreifen	Ja
Zuse Z4	Deutschland	März 1945	Ja	Ja	Nein	Ja, durch Lochstreifen	Ja
ENIAC	USA	1946	Nein	Nein	Ja	Teilweise, durch Neuverkabelung	Ja
		1948	Nein	Nein	Ja	Ja, durch eine Matrix aus Widerständen	Ja

Quelle: Wikipedia

# Ein Blick zurück ...

## Entwicklung der Rechnertechnologie



Quelle: H.Zuse, www.zuse.de

Konrad Zuse: Z1  
Mechanische  
„Schaltungstechnik“

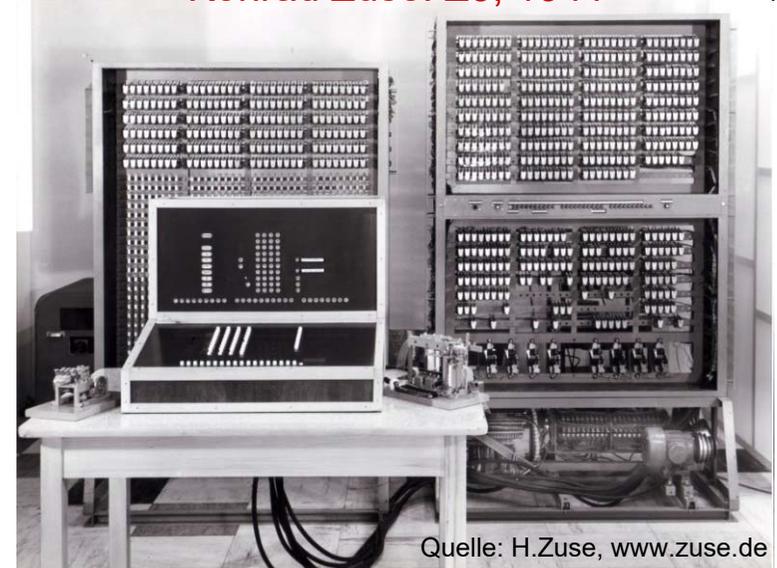
Relais: ab ~1940  
Elektromechanisch  
Schaltzeit: ms-Bereich



Elektronenröhre: ab ~1945  
Elektronisch  
Schaltzeit: ms-Bereich

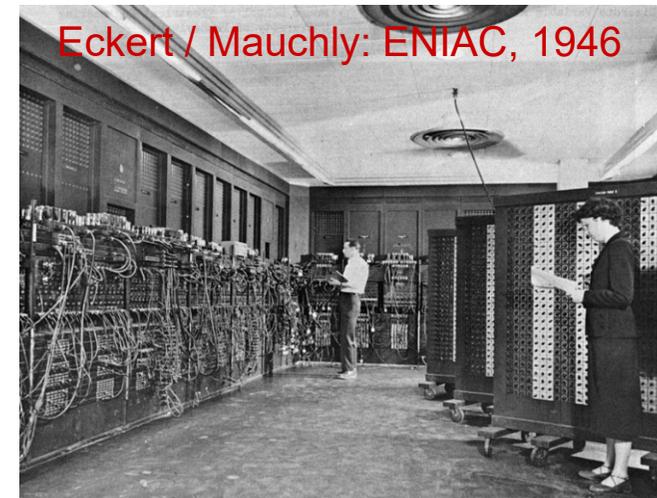


Konrad Zuse: Z3, 1941



Quelle: H.Zuse, www.zuse.de

Eckert / Mauchly: ENIAC, 1946

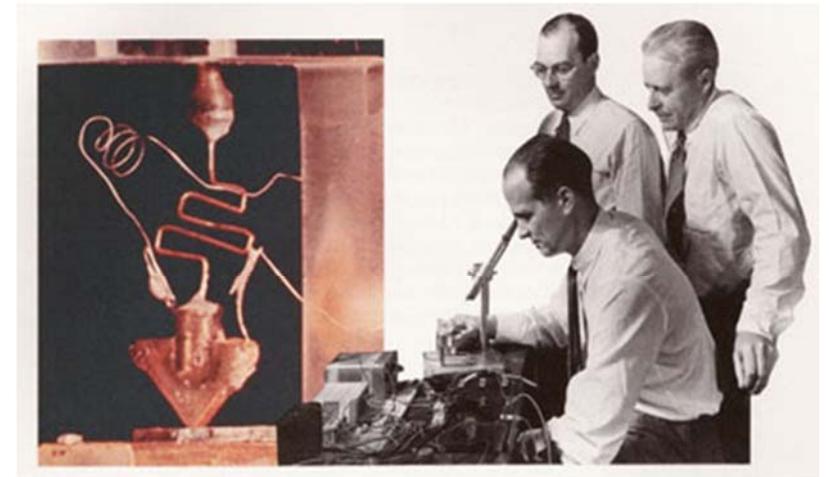


# Ein Blick zurück ...

## ■ Entwicklung der Rechnertechnologie

### ■ Transistor (1947)

- Ersetzt die Vakuum-Röhre und eröffnet die Möglichkeit zur Integration und ist damit die Voraussetzung für das rapide Wachstum integrierter Schaltkreise
- Erfinder: Shockley, Bardeen, Brattain in den AT&T Bell Labs, Murray Hill, NJ
- Nobel Preis in Physik 1956
- US Patent #02569347
- Basierend auf Halbleitern; pn-Übergängen



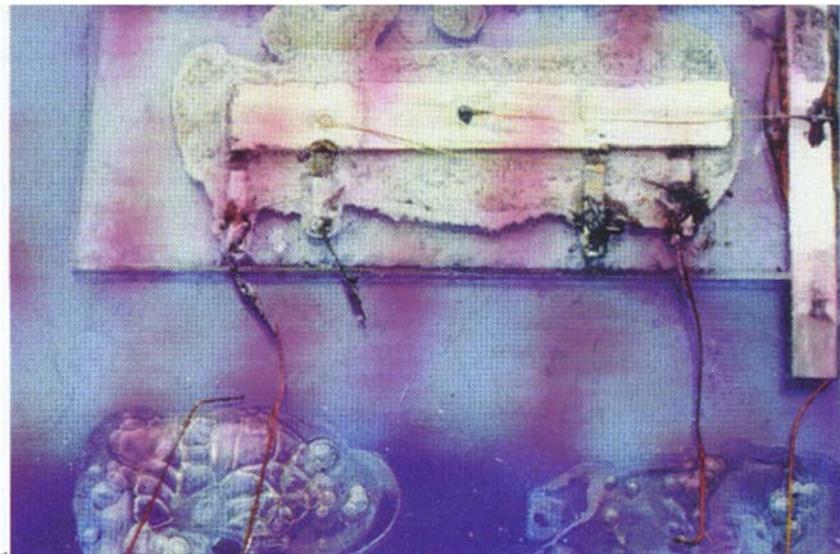
[Quelle: Bell Labs]

# Ein Blick zurück ...

## ■ Entwicklung der Rechnertechnologie

### ■ Integrierte Schaltkreise (etwa ab 1958)

- 1958 bauen Jack S. Kilby und Robert N. Noyce unabhängig voneinander die erste Integrierte Schaltung
- Jack S. Kilby wechselte im Mai 1958 zu Texas Instruments nach Dallas
  - September 1958 lötet er den ersten IC mit Golddrähten als Verbindungen auf einem einzigen Stück Germanium



Erster integrierter Schaltkreis von Kilby

Nobelpreis für Physik  
geht an Kilby (2000)

# Ein Blick zurück ...

## ■ Entwicklung der Rechnertechnologie

### ■ Integrierte Schaltkreise (etwa ab 1958)

- 1958 bauen Jack S. Kilby und Robert N. Noyce unabhängig voneinander die erste Integrierte Schaltung
- Robert N. Noyce erfindet in der gleichen Zeit bei Fairchild Semiconductor den IC mit Silizium als Träger- und Aluminium als Leiterbahnmaterial
- Noyce gründete 1968 mit Moore die Firma Intel, der er bis 1979 vorstand
- Intel: **I**ntegrated **e**lectronics



Andy Grove, Bob Noyce, Gordon Moore  
Quelle: Intel

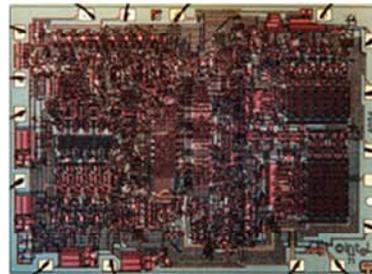
# Ein Blick zurück ...

## Entwicklung der Rechnertechnologie

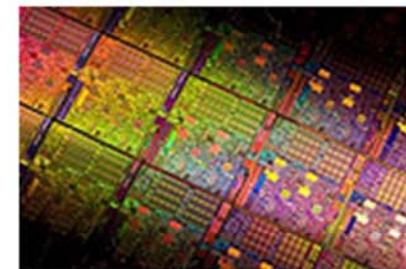
Transistoren: ab ~1947  
Halbleitertechnologie  
Schaltzeit: ns-Bereich



Intel 4004, 1971  
~ 2300 Transistoren  
Strukturbreite: 10µm



Intel „Nehalem“, 2010  
~ 2 Mrd. Transistoren  
Strukturbreite: 32nm



Planartechnik: ab ~1958  
Integrierte Schaltkreise  
Schaltzeit: ms-Bereich

