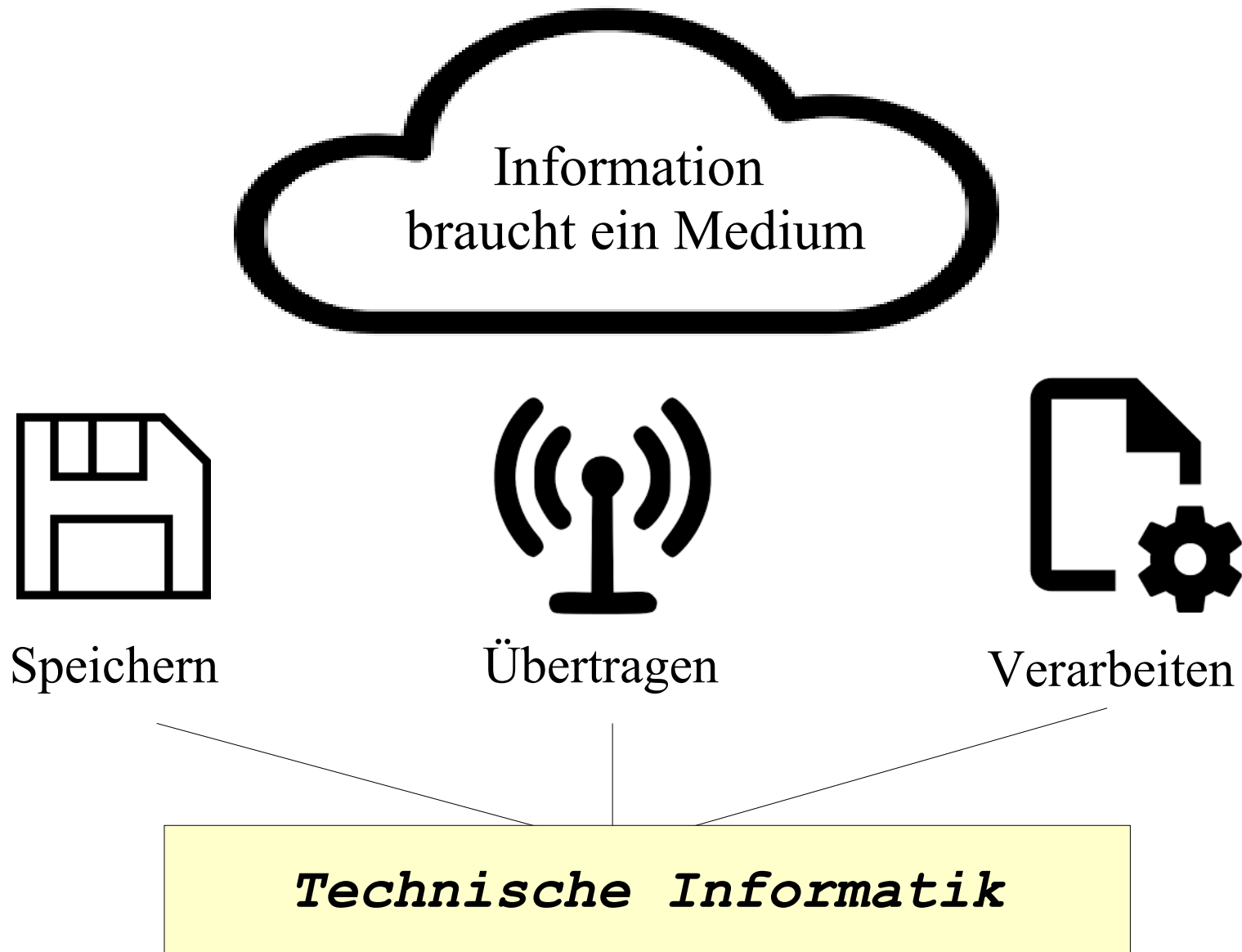




# Technische Informatik

## Einführung



„Wenn Sie Informatik studieren, lernen Sie nicht nur programmieren sondern auch die Grundlagen von Software und Hardware kennen.“

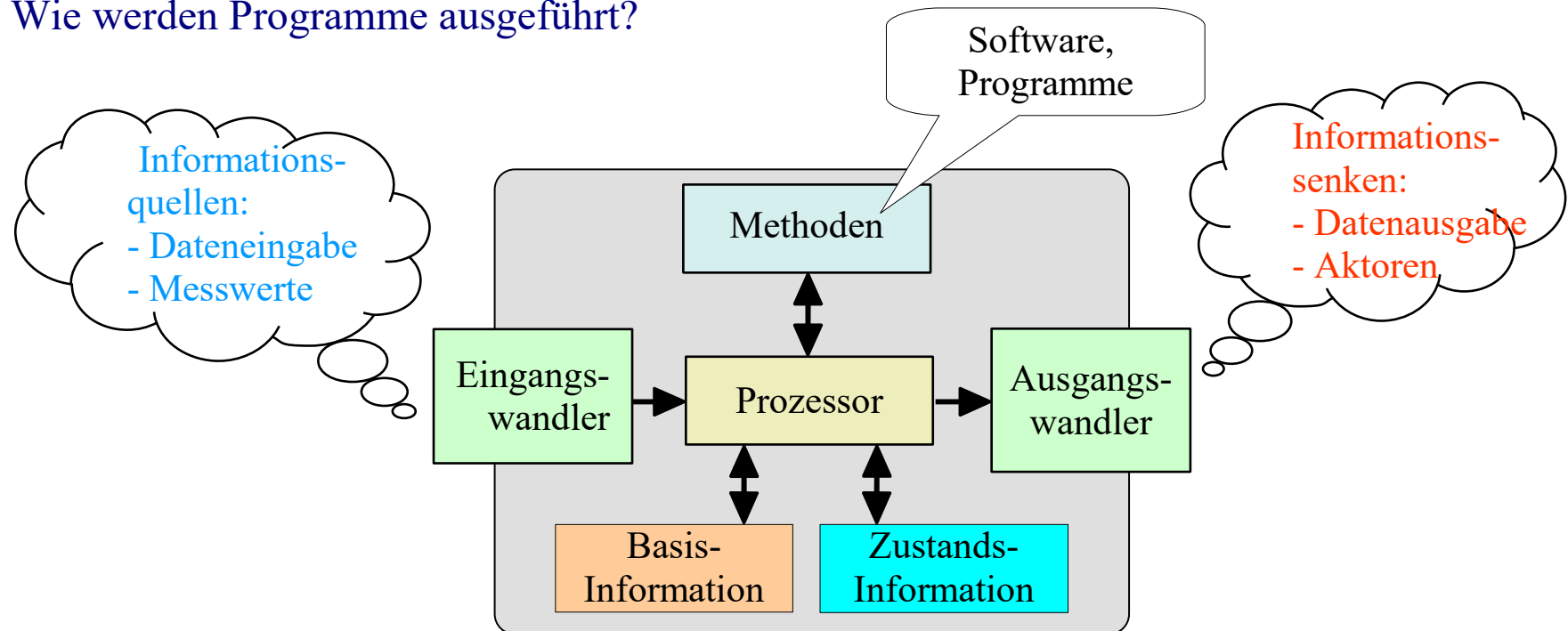
([www.hochschulkompass.de/studienbereiche-kennenlernen/mathematik-naturwissenschaften/informatik.html](http://www.hochschulkompass.de/studienbereiche-kennenlernen/mathematik-naturwissenschaften/informatik.html), 22.09.25)

„Technische Informatik ist ein Hauptgebiet der Informatik, das sich mit Architektur, Entwurf, Realisierung, Bewertung und Betrieb von Rechner-, Kommunikations- und eingebetteten Systemen auf der Ebene der Hardware als auch der systemnahen Software beschäftigt.“

([de.wikipedia.org/wiki/Technische\\_Informatik](http://de.wikipedia.org/wiki/Technische_Informatik))

# Technische Informatik im Studium

- „Informatik“
  - ist nicht nur „Programmieren“, sondern auch:
  - Konzepte, Methoden und Algorithmen der Datenverarbeitung ...
  - ... sowie (Ingenieur-)Wissenschaft und Technik
- Computer sind technische Systeme
  - Wie funktionieren diese?
  - Wie werden Programme ausgeführt?



# Ziele der Vorlesung

- Wissenschaftliche Denkweise und Methodik:
  - Darstellung von Wissen (Tabellen, Grafiken, ...)
  - Mathematische Modellierung (Formeln, Algorithmen,...)
  - Abstraktion (Verallgemeinerung)
  - Strukturiertes und systematisches Vorgehen
- Verständnis der funktionalen und strukturellen Grundlagen von Rechnerplattformen
  - Faktenwissen & Methodenwissen, aufbauend auf den Grundlagen der Digitaltechnik
  - Basiswissen zu den physikalischen und elektronischen Grundlagen der Computertechnik und ihrer Anwendungen

- Information = Wissen, Nachricht
  - Darstellung in Texten, Tabellen, Bildern, Audio-Signalen, ....
- Vereinheitlichte Basis der Informationsdarstellung im Rechner:
  - Bit als Elementarinformation („0“ oder „1“)
  - Komplexere Informationen durch Bit-Folgen, z.B. Buchstaben 'A' = 01000001
- Aber: Codierung muss bekannt sein bzw. vereinbart werden
  - Ist mit der Bitfolge 01000001 das Zeichen 'A' oder die Zahl 65 gemeint?



# Digital vs. Analog

- „digital“
  - lateinisch: digitus, der Finger, also hier: „mit dem Finger abzählbar“
  - Diskrete Signale mit typ. zwei („binären“) Zuständen, z.B. „ein“/„aus“
- „analog“
  - griechisch: verhältnismäßig, entsprechend
  - Information wird durch eine proportionale, physikalische Größe dargestellt und in **Analogtechnik** verarbeitet



Funktionsbeschreibung:

„Wenn der Schalter betätigt wird, leuchtet die Lampe!“

Schaltplan:





## Schalter S0

mit den möglichen Zuständen:

„0“ – Schalterposition wie in Zeichnung

„1“ – Alternativposition

Schaltplan:



## Beleuchtung L

mit den möglichen Zuständen:

„0“ – dunkel

„1“ – hell

Funktion:

Eingang			Ausgang
		S0	L
		0	0
		1	1

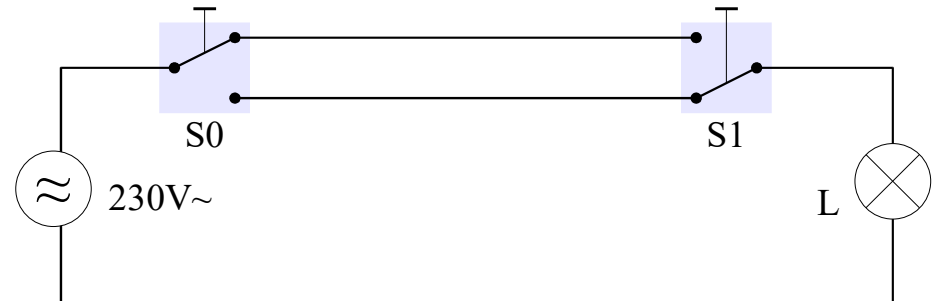
## Schalter S0 , S1

mit den möglichen Zuständen:

„0“ – Schalterposition wie in Zeichnung

„1“ – Alternativposition

## Schaltplan:



## Beleuchtung L

mit den möglichen Zuständen:

„0“ – dunkel

„1“ – hell

## Funktion:

Eingang			Ausgang
	S1	S0	L
	0	0	0
	0	1	1
	1	1	0
	1	0	1

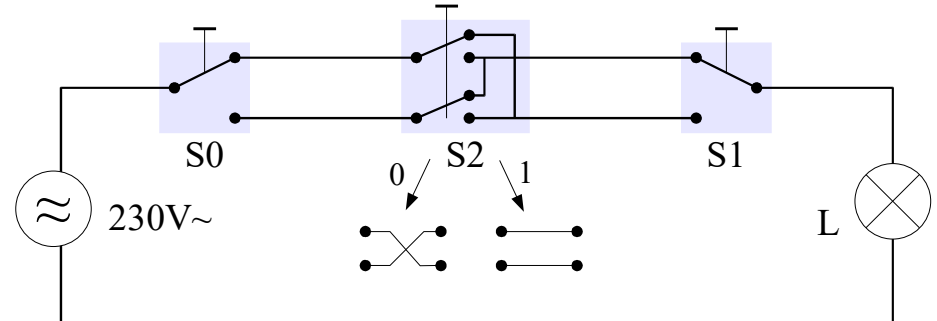
## Schalter S0, S1, S2

mit den möglichen Zuständen:

„0“ – Schalterposition wie in Zeichnung

„1“ – Alternativposition

Schaltplan:



## Beleuchtung L

mit den möglichen Zuständen:

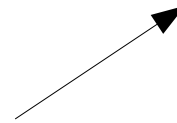
„0“ – dunkel

„1“ – hell

Funktion:

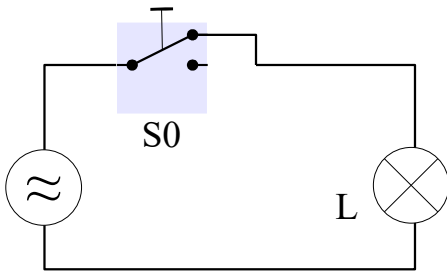
Eingang			Ausgang
S2	S1	S0	L
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	1	0	0
1	1	1	1
1	0	1	0
1	0	0	1

„Formale Spezifikation“



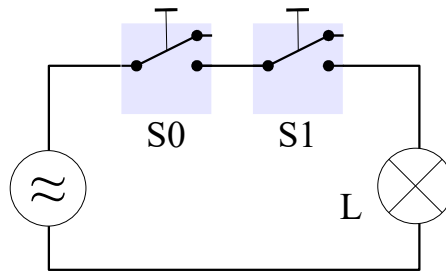
# Boole'sche Algebra

- Boole'sche Algebra:
  - Definiert elementare Verknüpfungen binärer Zustände



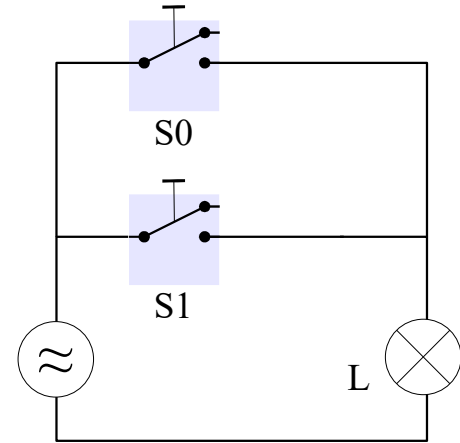
S0	L
0	1
1	0

NOT



S1	S0	L
0	0	0
0	1	0
1	0	0
1	1	1

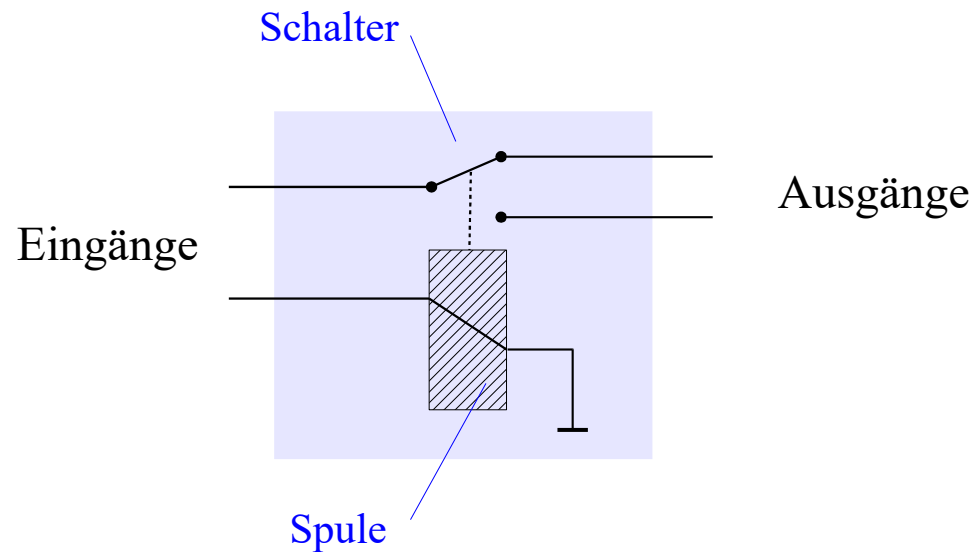
AND



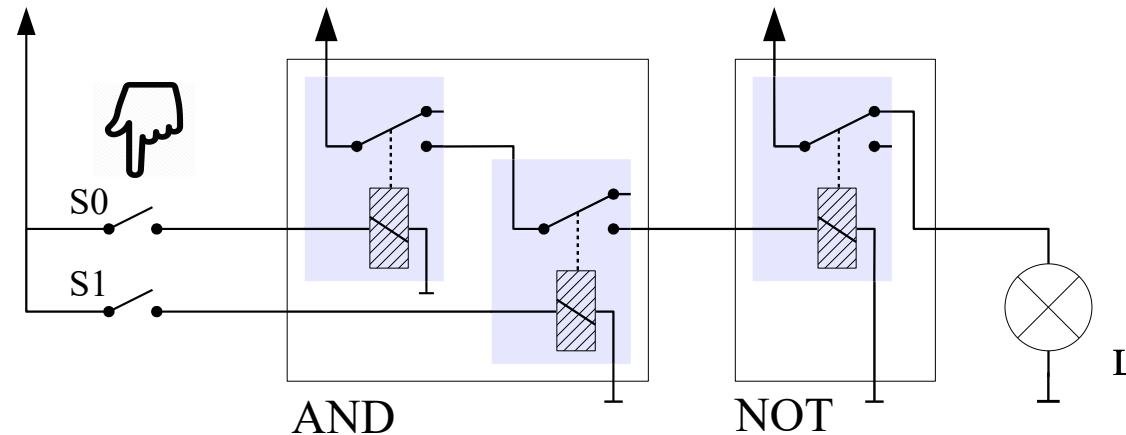
S1	S0	L
0	0	0
0	1	1
1	0	1
1	1	1

OR

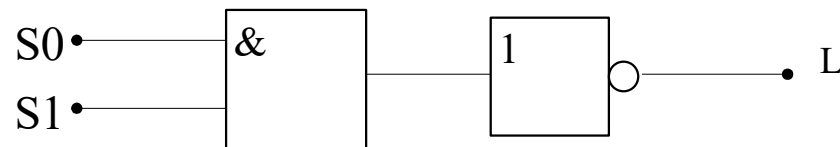
- Verkettung logischer Funktionen wird durch „elektrische Schalter“ ermöglicht
  - Realisierung heute durch Transistor / hochintegrierte Schaltkreise (IC)
  - Ausgänge können mit den Eingängen weiterer Schaltelemente verknüpft werden
- Relais:
  - Funktionsweise: Sobald eine Spannung an der Spule angelegt, wird der Schalter betätigt. Der Schalter kehrt in die Ausgangslage zurück, sobald die Spannung an der Spule abgeschaltet wird.



- Verkettung logischer Funktionen wird durch „elektrische Schalter“ ermöglicht



Abstrakte Darstellung:



Mathematische Darstellung:

$$L = \overline{S0 \cdot S1}$$

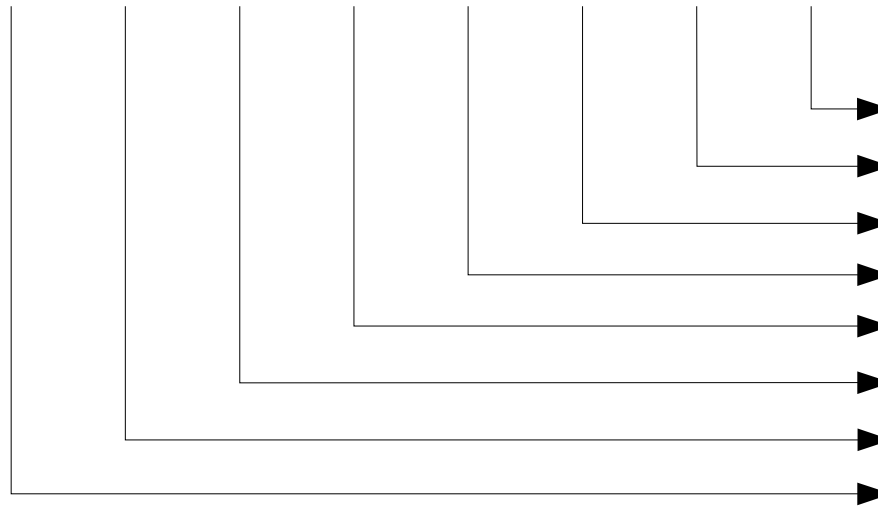
# Zahlendarstellung

- Zahlen im Dualsystem

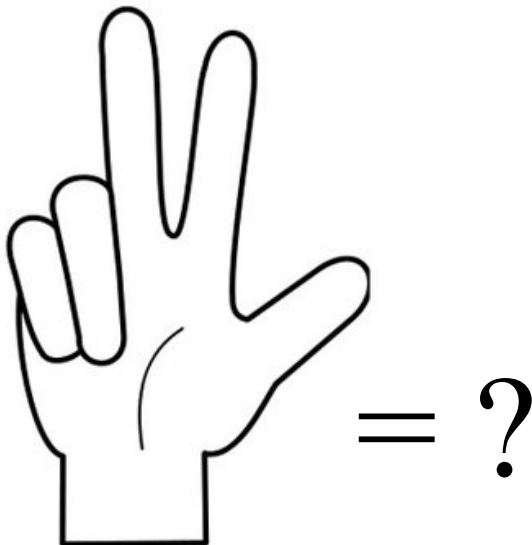
Wertigkeit:

Dualzahl:

128	64	32	16	8	4	2	1
0	1	1	0	0	0	1	1



$$\begin{array}{r} 1 \cdot 1 \\ + 1 \cdot 2 \\ + 0 \cdot 4 \\ + 0 \cdot 8 \\ + 0 \cdot 16 \\ + 1 \cdot 32 \\ + 1 \cdot 64 \\ + 0 \cdot 128 \\ \hline = 99 \end{array}$$



# Rechnerarithmetik: Addierer

- Addition einer mehrstelligen Dualzahl

- Beispiel:

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1 \\ +\ 0\ 1\ 1\ 1\ 0 \\ \hline \text{1 1 1 0} \leftarrow \text{Übertrag} \\ \hline =\ 1\ 0\ 1\ 0\ 1 \end{array}$$

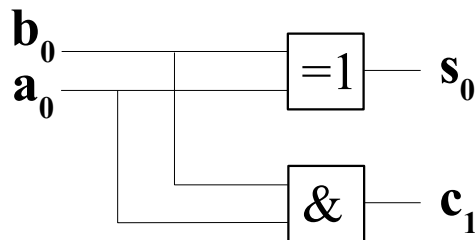
dezimal:

$$\begin{array}{r} 07 \\ +\ 14 \\ \hline \text{1} \\ \hline =\ 21 \end{array}$$

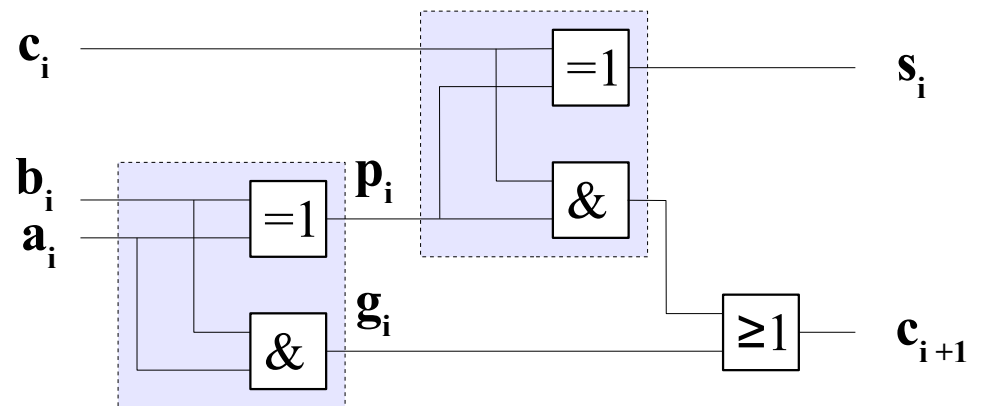
- Basiselemente:

- Halbaddierer (HA) zur Addition der niederwertigsten Bits (ohne Übertragseingang)
- Volladdierer (VA) zur Addition aller höherwertigen Bits

Halbaddierer:



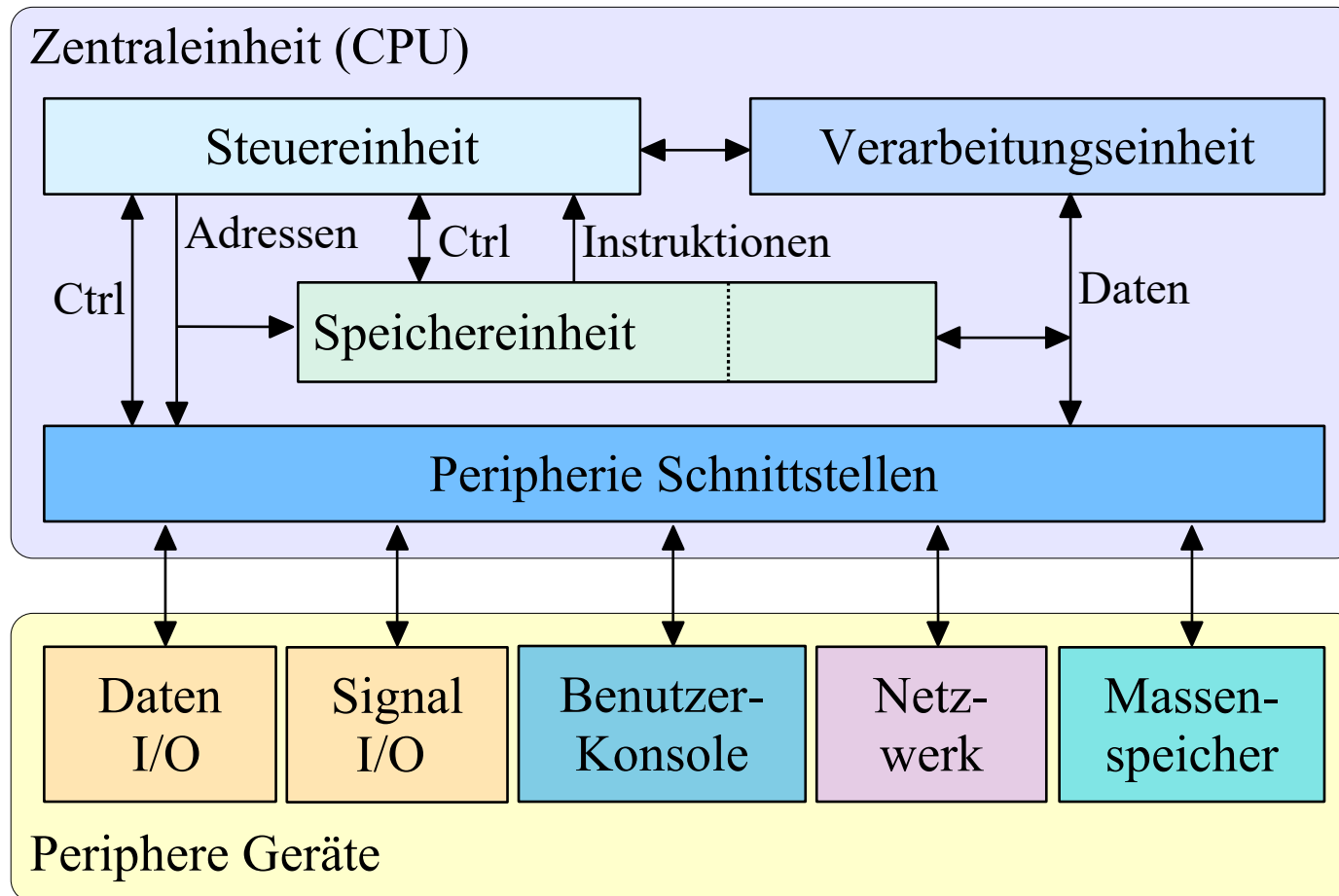
Volladdierer:





# Grundlegende Strukturen

- Architektur eines Rechnersystems



... und auf eine gute Zusammenarbeit!

