

## 4. Digitale Schaltungstechnik

### 4.1 Grundlagen digitaler Signale

#### Informationstechnische Grundlagen digitaler Signale

Ausgangspunkt zur Behandlung von Signalen sind die Begriffe **Information**, **Zeichen** und **Signal**, ( DIN 44300 ).

Umgangssprache → Sinn: Kenntniserwerb mit Neuigkeitsgehalt  
philosophischer Sinn: → an menschliches Denken gebunden  
→ **Informationen** in strengem Sinn können erst im Gehirn vorhanden sein.

**Shannon** (Begründer technische Informationstheorie): **Wert** der **Information** → durch sie „**beseitigte Unsicherheit**“ → wahrnehmungstheoretisch festgelegt  
Informationen müssen durch **Zeichen** repräsentiert werden

Träger der Zeichen sind **Signale**

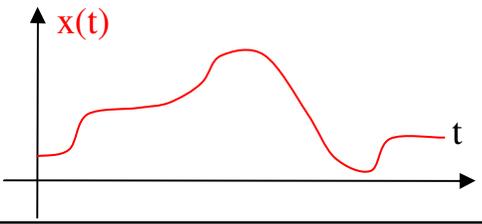
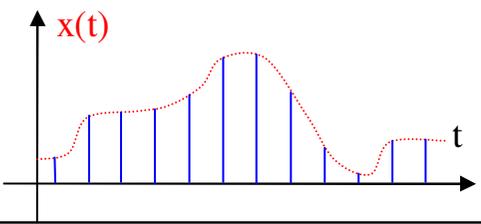
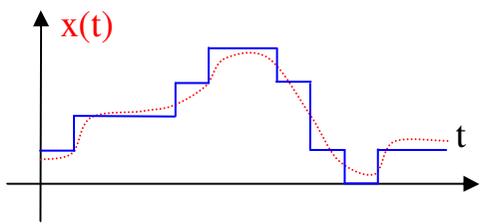
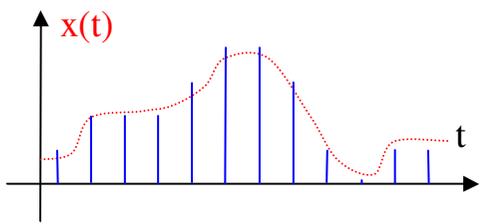
Technische Informationstheorie → von den drei Aspekten der Sprache

- Semantik – inhaltliche Bedeutung,
- Sigmatik – Zeichen sowie
- Pragmatik – Zweckbestimmung des Inhalts

**nur** den **sigmatischer** Aspekt.

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Einteilung von Signalen nach ihren Eigenschaften bezüglich Amplitude und Zeit

	kontinuierliches Signal	diskontinuierliches Signal
stetiges Signal	<p><b>analoges Signal</b> als zeitabhängige Größe (1)</p> 	<p><b>abgetastetes analoges Signal</b> zu diskreten Zeitpunkten (2)</p> 
diskretes Signal	<p><b>diskrete Signalstufen</b> als zeitkontinuierliches Signal (3)</p> 	<p><b>diskrete Signalstufen</b> zu diskreten Zeitpunkten (4)</p> 

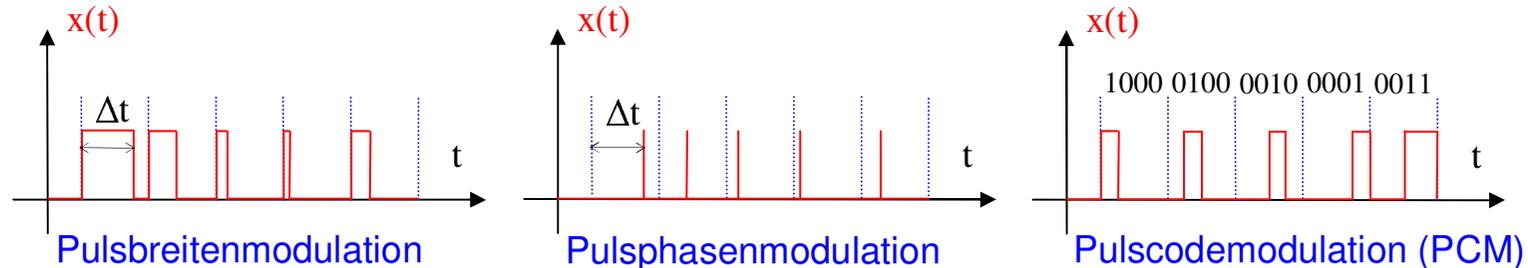
- (1) sehr viele Mess- und Sensorsignale
- (2) Abtast- oder Stichprobenmessungen
- (3) inkrementelle Geber, Codescheibe
- (4) zeit- und amplitudendiskrete Signale

- (1) frequenzanaloge Signale  $\hat{U} \cos\{2\pi f(t)t\}$  mit  $x(t) \sim f(t)$  und  $\hat{U} = \text{const}$
- (4) binäre Signale (Amplitudenstufe 0 oder 1, Pulscodemodulation PCM)

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

## Arten pulsmodulierter Signale

Pulsamplitudenmodulation entspricht abgetasteten analogen Signal



Pulsamplituden-, Pulsbreiten-, Pulsphasenmodulation sind **zeitdiskrete**, aber **amplitudenkontinuierliche** Signale

Pulsmodulation gehört zu **zeit- und amplitudendiskreten** und der **Sonderform** binären oder **digitalen** Signalen

(in Übertragungstechnik schon vor Entwicklung der Digitaltechnik genutzt)

binäre, binärcodierte oder digitale Signale → **eigenständige Schaltungstechnik**

Signalverarbeitung: mathematische Grundlage → Dualzahlen, Boole'sche Algebra (Schaltalgebra)

## Mathematische Grundlagen – Zahlensysteme

Dezimalzahlensystem als Stellen- oder **Positionssystem** (Ziffern mit Stellenwert im Zahlwort), basiert auf **Grundzahl „10“** → zehn **Ziffern (0, 1, ... 9)**. Im Zahlwort werden Ziffern mit ihrem Stellenwert multipliziert und addiert.

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Nach gleichen Verfahren → Zahlensysteme mit den Grundzahlen

- „2“, Ziffern 0 und 1 und Stellenwert  $2^n$  – **Dualzahlen** oder Binärzahlen  
(Kleinste Grundzahl, da nur eine Ziffer kein Positionssystem ist)
  - „8“ ( $= 2^3$ ), Ziffern 0, 1, ... 7 und Stellenwert  $8^n$  – **Oktalzahlen**
  - „16“ ( $= 2^4$ ), Ziffern 0, 1, ... 9, A ...F und Stellenwert  $16^n$  – **Hexadezimalzahlen**
- $n$  → ganze Zahl → mit jeder Stelle +1. Komma zwischen Stelle 0 und 1

Binärzahlen	Oktalzahlen	Dezimalzahlen	Hexadezimalzahlen	BCD-Code
0	0	0	0	0000 0000
1	1	1	1	0000 0001
10	2	2	2	0000 0010
11	3	3	3	0000 0011
100	4	4	4	0000 0100
101	5	5	5	0000 0101
110	6	6	6	0000 0110
111	7	7	7	0000 0111
1000	10	8	8	0000 1000
1001	11	9	9	0000 1001
1010	12	10	A	0001 0000
1011	13	11	B	0001 0001
1100	14	12	C	0001 0010
1101	15	13	D	0001 0011
1110	16	14	E	0001 0100
1111	17	15	F	0001 0101
10000	20	16	10	0001 0110
10001	21	17	11	0001 0111

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

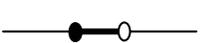
Dualzahlen → lange Zahlwörter → für Normalgebrauch nicht geeignet

Dualzahlen kompatibel → Hexadezimalzahlen → in Computertechnik

BCD-Code → heute nur bei speziellen Anwendungen z.B. für Zähler

technische Bedeutung der Dualzahlen (einfachstes Zahlensystem) ←

Möglichkeit, mit hoher Störfestigkeit zwei Ziffern (zwei Zustände) zu realisieren

Ziffer	logischer Wert	Schaltzustand	Beispiele für elektrische Signale
0	Nein oder Falsch		+3,6...+6V (V24) oder 0 ... 0,8 V (TTL-Eingang)
1	Ja oder Wahr		-3,6...-6V (V24) oder 2,0 ... 5 V (TTL-Eingang)

Umwandlung Dual- ↔ Dezimalzahlen

$$110101_B = \left\{ \begin{array}{l} 1 \cdot 2^5 \\ +1 \cdot 2^4 \\ +0 \cdot 2^3 \\ +1 \cdot 2^2 \\ +0 \cdot 2^1 \\ +1 \cdot 2^0 \end{array} \right\} = \left\{ \begin{array}{l} 1 \cdot 32 \\ +1 \cdot 16 \\ +0 \cdot 8 \\ +1 \cdot 4 \\ +0 \cdot 2 \\ +1 \cdot 1 \end{array} \right\} = 53_D$$

$$53_D = \left\{ \begin{array}{l} 5 \cdot 10^1 \\ +3 \cdot 10^0 \end{array} \right\} = \left\{ \begin{array}{l} 101 \cdot 1010 \\ + 11 \cdot 1 \end{array} \right\} = \left\{ \begin{array}{l} 101 \cdot 1000 \\ +101 \cdot 10 \\ + 11 \cdot 1 \end{array} \right\} = \left\{ \begin{array}{l} 101000 \\ + 1010 \\ + 111 \end{array} \right\} \begin{array}{l} \\ \\ \text{Übertrag} \end{array}$$

$$\underline{\hspace{1.5cm}} \\ 110101_B$$

Umwandlung **Dual- ↔ Hexadezimalzahlen**

$$\begin{array}{c} 0011|0101_B \\ 3 \quad | \quad 5 \quad H \end{array}$$

**Addition** von Dualzahlen

$$\begin{array}{r} 11101 \\ + 1110 \\ \hline 101011 \end{array} \quad \begin{array}{r} 29_D \\ 14_D \\ \hline 43_D \end{array}$$

$\leftarrow$  Überträge

**Multiplikation** von Dualzahlen

$$\begin{array}{r} 11101 \cdot 1110 \\ \hline 11101 \\ 11101 \\ 11101 \\ 10000 \\ \hline 110010110 \end{array} \quad \begin{array}{r} 29_D \cdot 14_D \\ \hline 29 \\ 116 \\ \hline 406_D \end{array}$$

$\leftarrow$  Überträge

**Subtraktion** und **Division** erfolgen in gleichartiger Weise

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Maßbezeichnung **SI System** für binäre Einheit (0 oder 1) wurde 1 **bit**

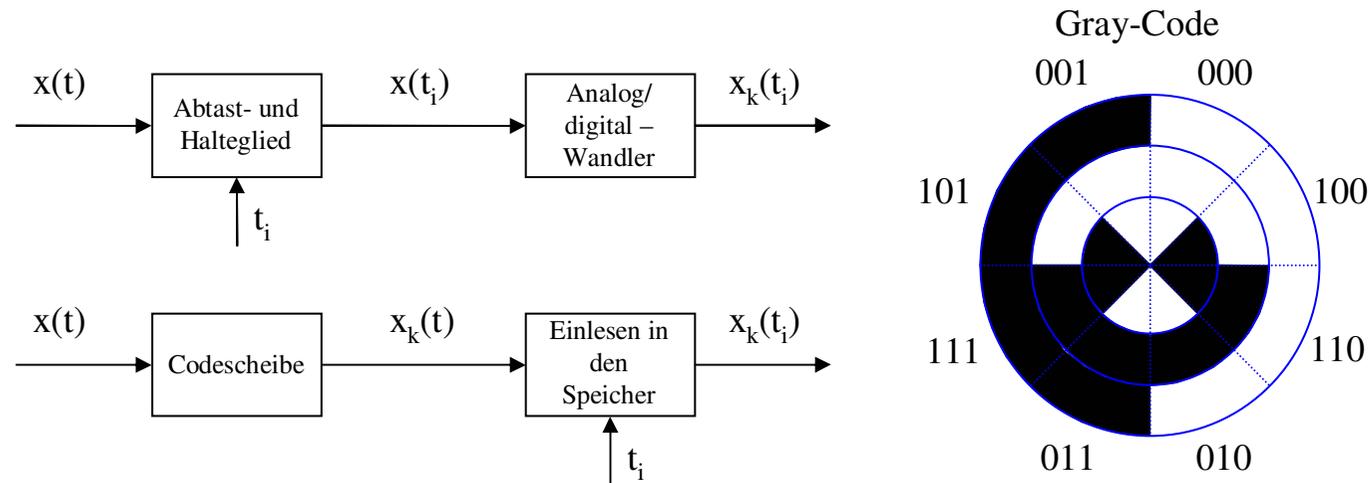
Da  $2^{10} = 1024 \approx 1000 \rightarrow$  Maßbezeichnung 1 Kbit, für  $2^{20}=1000 \cdot 1000 \rightarrow$  1 Mbit

Acht binäre Einheiten  $\rightarrow$  binäres Wort  $\rightarrow$  Einheit 1 **B** (Byte) und 1 KB, 1 MB

## Quantisierung von Signalen und Codierung

Zur Umwandlung **analoger Signale in digitale Signale**  $\rightarrow$   
**Zeitquantisierung** und Amplituden- oder **Wertquantisierung**

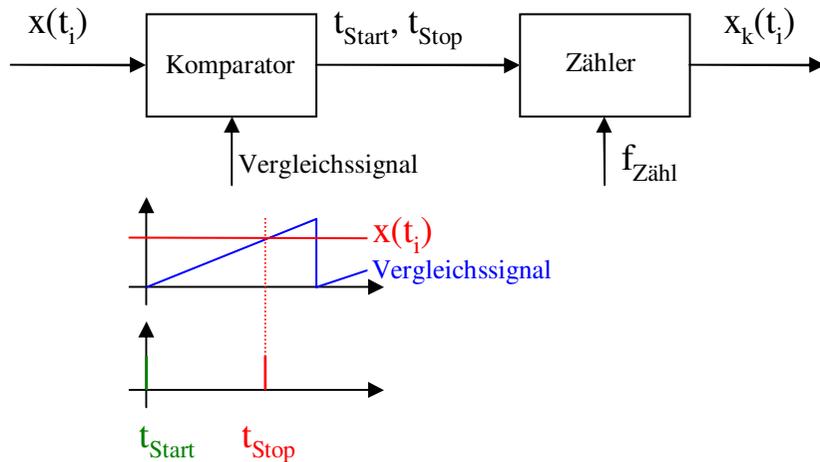
**Wandlung:** zwei Schritte  $\rightarrow$  zwei Varianten



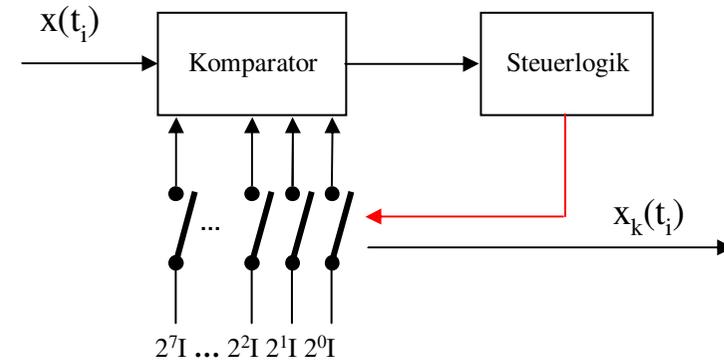
Analog-Digital-Umsetzer  $\rightarrow$  Wertquantisierung und Codierung in Dualzahlen

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

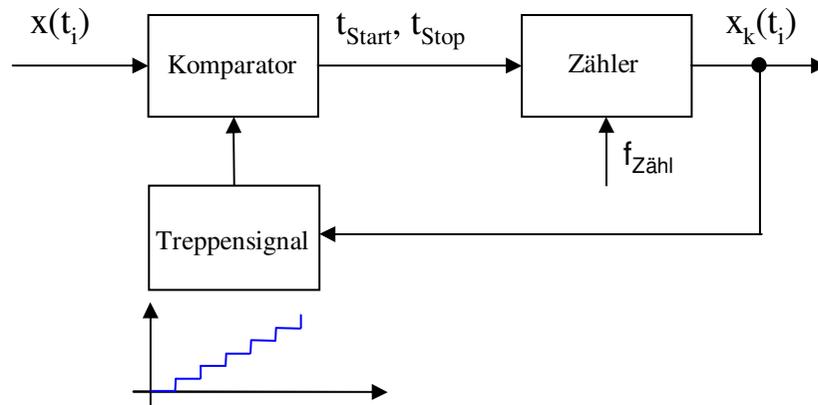
## • Dreiecksverfahren



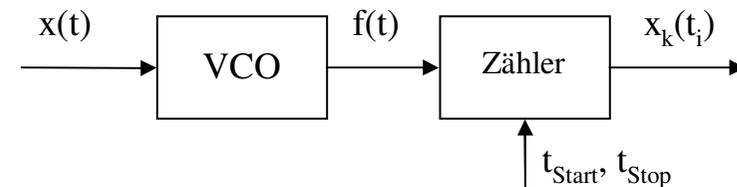
## • Balanceverfahren



## • Treppenverfahren



## • mit frequenzanalogem Signal



## Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

### • Inkrementelle Geber

Einzelimpulse für festgelegte Signaländerung (z.B. bewegtes Strichgitter)

→ Zählen von einer Nullstellung an → Dualzahl (Zählfehler summieren sich)

### Vielzahl Schaltkreise verfügbar Unterschiede:

- Eingangsspannungsbereich, Eingangswiderstand, Genauigkeit, Ausgangswertebereich (8, 12, 16 ...bit), Versorgungsspannung, -leistung
- evtl. Multiplexer, Abtast- und Halteglied oder ein Datenpuffer
- Fast immer Pegelanpassung notwendig
- manchmal Präzisionsgleichrichter erforderlich

## Zeitquantisierung → Abtastung des Signals (in Regel feste Abtastrate)

Damit original Signal aus abgetastetem Signal reproduziert werden kann → Abtastung genügend schnell > **zwei Abtastungen pro Periode von  $f_{\text{Max}}$**

$$\text{Abtasttheorem: } \Delta t_{\text{Ab}} = T_{\text{Min}}/2 = 1/2f_{\text{Max}}$$

→ Signalanteil von  $f_{\text{Max}}$  → mit Genauigkeit von gerade 50 % reproduzierbar

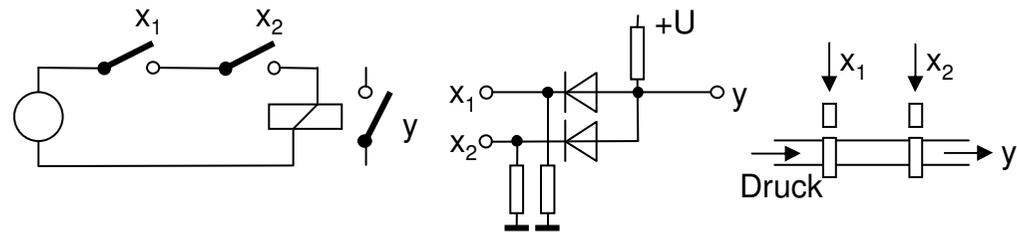
→ **vor der Abtastung** mit einem Filter die höheren Anteile abtrennen

## Grundlagen digitaler Signalverarbeitung – Schaltalgebra

Rechenoperationen mit Binärzahlen → drei logische Grundfunktionen.  
Diese stimmen mit Grundfunktionen der Aussagenlogik überein.

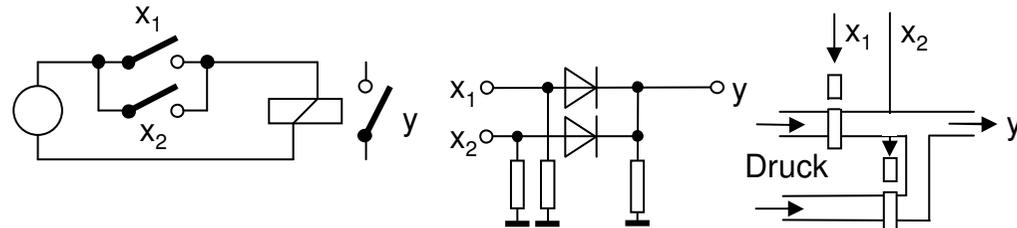
- **UND** – Funktion (oder AND bzw. Konjunktion)

$x_1 \wedge x_2 = y$	$x_1$	$x_2$	$y$	$x_1$ und $x_2$ erfüllt → $y$ wahr.
	0	0	0	
	1	0	0	
	0	1	0	
	1	1	1	



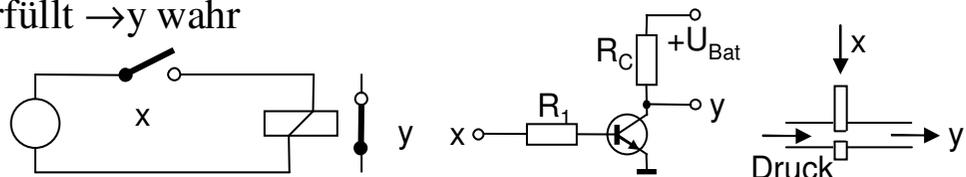
- **ODER** – Funktion (oder OR bzw. Disjunktion)

$x_1 \vee x_2 = y$	$x_1$	$x_2$	$y$	$x_1$ oder $x_2$ erfüllt → $y$ wahr.
	0	0	0	
	1	0	1	
	0	1	1	
	1	1	1	



- **NICHT** – Funktion (oder NOT bzw. Negation)

$\bar{x} = y$	$x$	$y$	$x$ nicht erfüllt → $y$ wahr
	0	1	
$\overline{(\bar{x})} = x$	1	0	



# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Durchgesetzt hat sich die **TTL- Schaltungstechnik** (Transistor-Transistor-Logik) und in hochintegrierten Schaltungen die **C-MOS Schaltungstechnik**

- N AND**

$\overline{x_1 \wedge x_2} = y$	$x_1$	$x_2$	$y$
	0	0	1
	1	0	1
	0	1	1
	1	1	0

( $x_1 \wedge x_2$ ) nicht erfüllt  $\rightarrow$  y wahr.

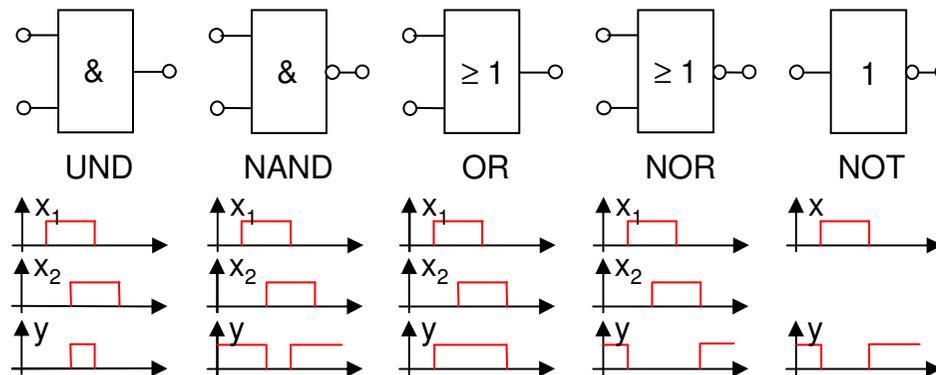
- NOR**

$\overline{x_1 \vee x_2} = y$	$x_1$	$x_2$	$y$
	0	0	1
	1	0	0
	0	1	0
	1	1	0

( $x_1 \vee x_2$ ) nicht erfüllt  $\rightarrow$  y wahr.

Nutzung dieser Bausteine  
 $\rightarrow$  **keine Einschränkung**

**Symbole** für  
 logischen Elemente  
 und ihre **Signalverläufe**



Zeitdarstellungen  $\rightarrow$  bei komplexen Schaltungen verschiedene Verzögerungen, Schaltfehler (Hasards)

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

## Rechenregeln:

- Rechenregeln mit Konstanten

$$\begin{array}{llll} \overline{0} = 1 & 0 \wedge 0 = 0 & & \\ \overline{1} = 0 & 0 \wedge x = 0 & 1 \wedge x = x & x_1 \wedge x_2 = x_1 x_2 \\ & x \wedge x = x & \overline{\overline{x}} \wedge x = 0 & \\ & 0 \vee 0 = 0 & & \\ & 0 \vee x = x & 1 \vee x = 1 & \\ & x \vee x = x & \overline{\overline{x}} \vee x = 1 & \end{array}$$

- Kommutatives Gesetz (Vertauschung)

$$\begin{array}{l} x_1 x_2 = x_2 x_1 \\ x_1 \vee x_2 = x_2 \vee x_1 \end{array}$$

- Assoziatives Gesetz (Verbindung)

$$\begin{array}{l} x_1 x_2 x_3 = x_1 (x_2 x_3) = (x_1 x_2) x_3 \dots \\ x_1 \vee x_2 \vee x_3 = x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3 \dots \end{array}$$

- Distributives Gesetz (Verteilung)

$$\begin{array}{l} x_1 (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3 \\ x_1 \vee (x_2 x_3) = (x_1 \vee x_2)(x_1 \vee x_3) \end{array}$$

- Absorptionsgesetze

$$\begin{array}{l} x_1 \vee x_1 x_2 = x_1 \\ x_1 (x_1 \vee x_2) = x_1 \\ x_1 (\overline{x_1} \vee x_2) = x_1 x_2 \\ x_1 \vee \overline{x_1} x_2 = x_1 \vee x_2 \\ \overline{x_1 x_2} = \overline{x_1} \vee \overline{x_2} \\ \overline{x_1 \vee x_2} = \overline{x_1} \overline{x_2} \end{array}$$

Rechenregeln mit Wahrheitstabellen  
oder durch Umformen beweisen

## Vorrangregeln:

Negation,

UND (AND, Multiplikationspunkt, kein Zeichen),

ODER (OR)

Regel Punkt- vor Strichrechnung

Redundante Formeln oder Ausdrücke → mit den Regeln vereinfachen

Minimieren bis zur Minimalform (ohne Redundanz) → auch graphische Methoden (z.B. [Karnaugh-Veitch-Diagramm](#))

Heute → Rechnerprogramme, die Minimieren und Optimieren auf vorhandene logische Elemente (z.B. eines ASIC) durch gezieltes Erweitern

**Aufgabe 4.1.1**

Wandeln Sie die Dezimalzahl 128, 192 und 240 (die in Subnetmasken vorkommen) in eine Hexadezimalzahl, Oktalzahl und Binärzahl um.

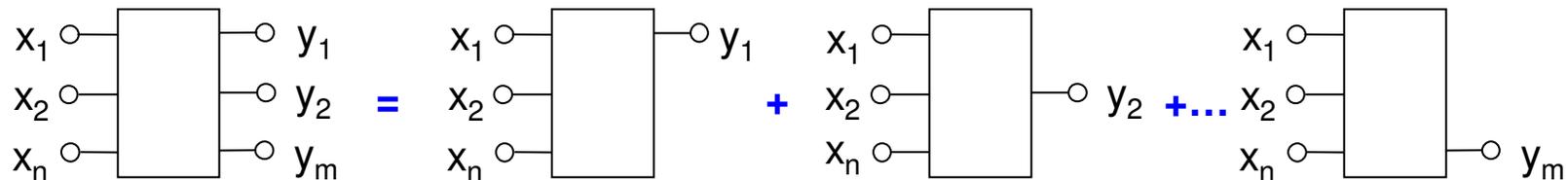
**Aufgabe 4.1.2**

Führen Sie eine bitweise UND-Verknüpfung der Binärzahlen der IP-Adresse 134.28.125.31 mit der Subnetmaske 255.255.255.128 (Dezimalzahlangabe) durch.

## 4.2 Kombinatorische Schaltungen

Kombinatorische Schaltungen – **Schaltnetze** - bestehen aus OR, (NOR,) AND, (NAND), NOT Funktionen (Bausteinen), sind **speicherfrei**.

Allgemein getrennte Betrachtung aller Eingangs- mit jeweiligen Ausgangsgrößen



Aufgabenstellung: → für jede Kombination der Eingänge → Werte der Ausgänge

**Belegungstabelle:** Beispiel Treppenlichtsteuerung

- $x_1$  Taster Erdgeschoss (ein oder aus)
- $x_2$  Taster Obergeschoss (ein oder aus)
- $x_3$  Zeitschalter (läuft oder abgelaufen)
- $y_1$  Licht (ein oder aus)
- $y_2$  Kontrolllicht der Taster (ein oder aus)
- $y_3$  Zeitschalter (starten oder nicht starten)

## Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Belegungstabelle → je Ausgang alle Kombinationen möglicher Zustände aller Eingänge festlegen → bei drei Eingängen 8 Kombinationen

Nr.	TE	TO	Z	L	KL	Z	
	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	
1	0	0	0	0	1	0	nur Kontrollleuchten sollen ein sein
2	1	0	0	1	1	1	wenn Taster 1 betätigt, Licht und Zeitschalter ein
3	0	1	0	1	1	1	wenn Taster 2 betätigt, Licht und Zeitschalter ein
4	1	1	0	1	1	1	gleichfalls, wenn zufällig beide Taster betätigt
5	0	0	1	1	0	0	wenn Licht ein, alles ignorieren
6	1	0	1	1	0	0	wenn Licht ein, alles ignorieren
7	0	1	1	1	0	0	wenn Licht ein, alles ignorieren
8	1	1	1	1	0	0	wenn Licht ein, alles ignorieren

(Kombination 2, 3, 4 könnten die Kontrollleuchten der Taster auch aus sein, da Licht)

Belegungstabelle → **disjunktive kanonische Normalformen**

(eine ODER-Verknüpfung aller Eingangsbelegungen für jedes  $y = 1$ )

$y_1$  → die Belegungen 2 bis 8,  $y_2$  → die Belegungen 1 bis 4,  $y_3$  → die Belegungen 2 bis 4)

$$y_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3$$

$$y_2 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3$$

$$y_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3$$

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

**Minimierung** → geringster Bauelementebedarf

$$y_3 = (x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2) \bar{x}_3 = [(x_1 \vee \bar{x}_1) x_2 \vee (x_2 \vee \bar{x}_2) x_1] \bar{x}_3 = (x_1 \vee x_2) \bar{x}_3$$

zweimal verwendet ◀┘                      └▶                      └▶ immer erfüllt

$$y_2 = (\bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2) \bar{x}_3 = \bar{x}_3$$

└▶ Klammer immer erfüllt

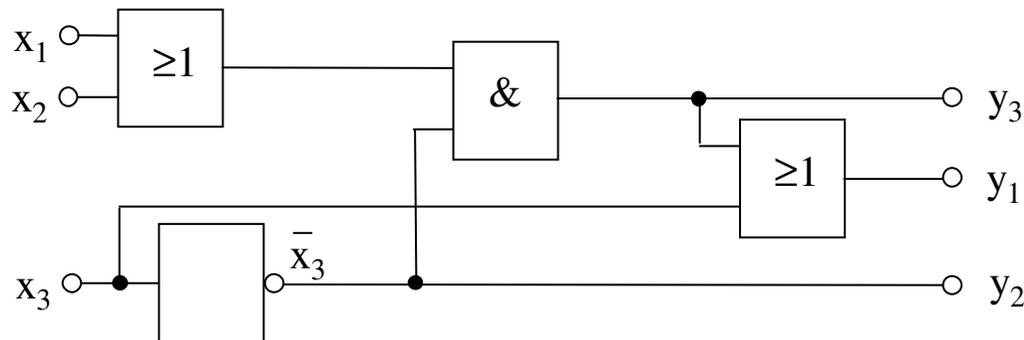
$$y_1 = (x_1 \vee x_2) \bar{x}_3 \vee (\bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2) x_3 = \underline{(x_1 \vee x_2) \bar{x}_3 \vee x_3}$$

└▶ 2...4 wie oben                      └▶ Klammer immer erfüllt



(Mit  $y_1 = x_3$  → richtige Funktion → Licht erst nach Loslassen des Tasters)

**logische Schaltung** (das Schaltnetz)



Drei verschiedene Schaltkreise (1x OR, 1x AND und 1x NOT) → es bleiben in Mehrfachgattern viele Gatter ungenutzt.

Oft günstiger, die Schaltung nur aus einer Sorte Gatter zu realisieren.



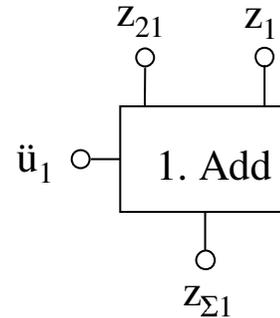
# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

## Additionsschaltung

$$\begin{array}{r}
 111 \\
 + 11 \\
 \hline
 1010
 \end{array}
 \begin{array}{l}
 7_D \\
 3_D \\
 \hline
 10_D
 \end{array}$$

Überträge

niedrigste Stelle beider Zahlen



Belegungstabelle

Eingänge		Ausgänge	
z <sub>11</sub>	z <sub>21</sub>	z <sub>Σ1</sub>	ü <sub>1</sub>
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

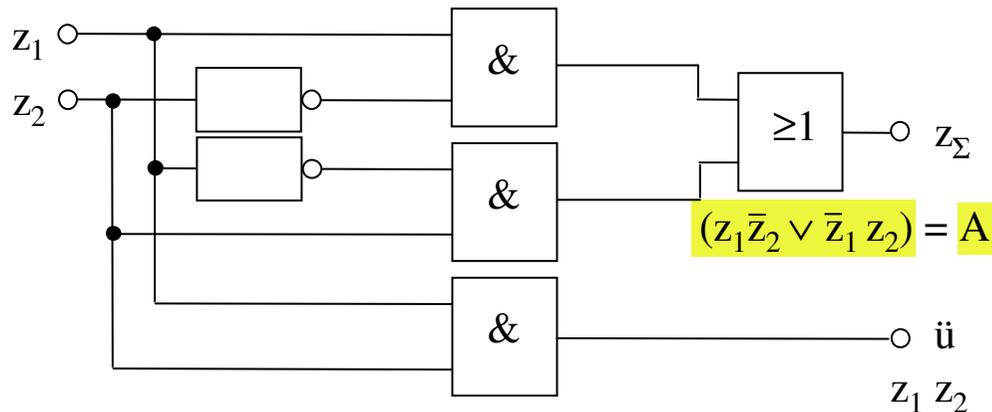
Die logischen Funktionen aus der Belegungstabelle werden:

- für Summe (disjunktive Normalform → zwei Ausdrücke für  $z_{\Sigma 1} = 1$ )

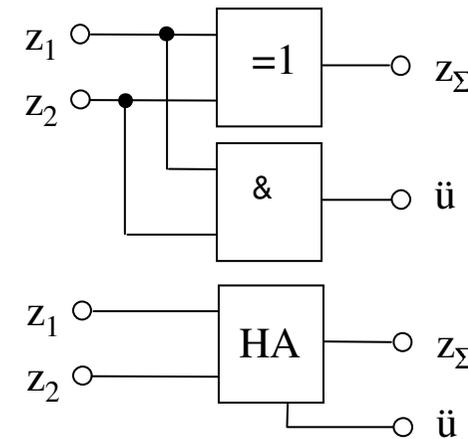
$$z_{\Sigma 1} = z_{11} \bar{z}_{21} \vee \bar{z}_{11} z_{21} \rightarrow \text{EXOR Symbol „=1“}$$

- und für den Übertrag (mit nur einem Ausdruck)

$$\ddot{u}_1 = z_{11} z_{21}$$

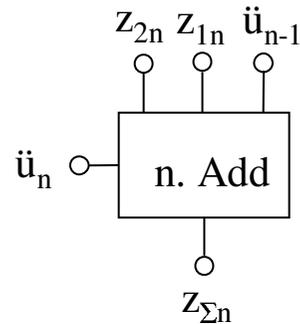


Halbaddierer



# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

**n-te Stelle** → Übertrag der (n-1)-ten Stelle zusätzlich verarbeiten



Belegungstabelle

Eingänge			Ausgänge	
$z_{1n}$	$z_{2n}$	$\ddot{u}_{n-1}$	$z_{\Sigma n}$	$\ddot{U}_n$
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Die **logischen Funktionen** für die n-te Stelle (mit je vier Ausdrücken) werden:

$$z_{\Sigma n} = z_{1n} \bar{z}_{2n} \bar{\ddot{u}}_{n-1} \vee \bar{z}_{1n} z_{2n} \bar{\ddot{u}}_{n-1} \vee \bar{z}_{1n} \bar{z}_{2n} \ddot{u}_{n-1} \vee z_{1n} z_{2n} \ddot{u}_{n-1}$$

$$\ddot{u}_n = z_{1n} z_{2n} \bar{\ddot{u}}_{n-1} \vee z_{1n} \bar{z}_{2n} \ddot{u}_{n-1} \vee \bar{z}_{1n} z_{2n} \ddot{u}_{n-1} \vee z_{1n} z_{2n} \ddot{u}_{n-1}$$

Nach dem **Minimieren** wird daraus:

$$z_{\Sigma n} = (z_{1n} \bar{z}_{2n} \vee \bar{z}_{1n} z_{2n}) \bar{\ddot{u}}_{n-1} \vee (\bar{z}_{1n} \bar{z}_{2n} \vee z_{1n} z_{2n}) \ddot{u}_{n-1}$$

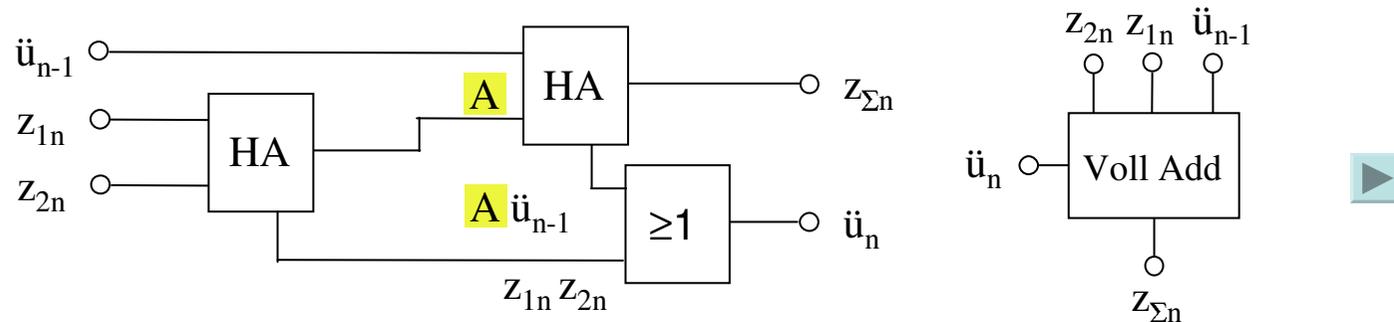
$$z_{\Sigma n} = (z_{1n} \bar{z}_{2n} \vee \bar{z}_{1n} z_{2n}) \bar{\ddot{u}}_{n-1} \vee (z_{1n} z_{2n} \vee \bar{z}_{1n} \bar{z}_{2n}) \ddot{u}_{n-1}$$

$$\ddot{u}_n = z_{1n} z_{2n} \bar{\ddot{u}}_{n-1} \vee z_{1n} \bar{z}_{2n} \ddot{u}_{n-1} \vee \bar{z}_{1n} z_{2n} \ddot{u}_{n-1} \vee z_{1n} z_{2n} \ddot{u}_{n-1}$$

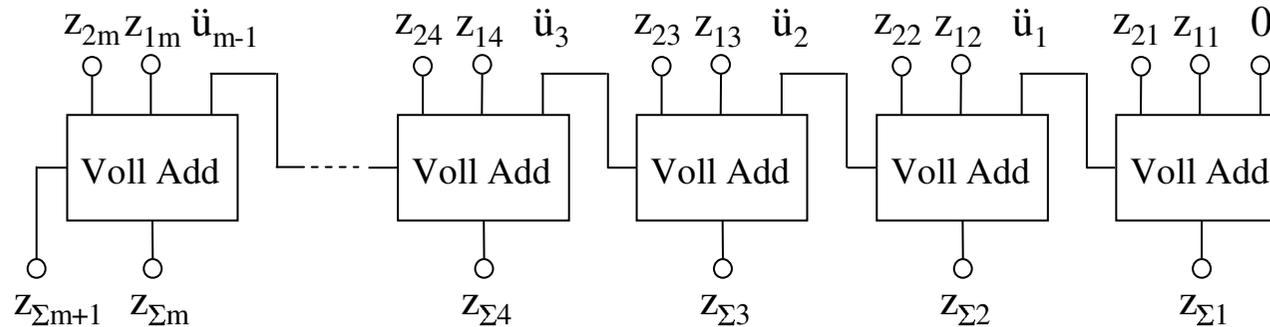
$$\ddot{u}_n = z_{1n} z_{2n} (\bar{\ddot{u}}_{n-1} \vee \ddot{u}_{n-1}) \vee (z_{1n} z_{2n} \vee \bar{z}_{1n} \bar{z}_{2n}) \ddot{u}_{n-1} = z_{1n} z_{2n} \vee (z_{1n} \bar{z}_{2n} \vee \bar{z}_{1n} z_{2n}) \ddot{u}_{n-1}$$

$z_{\Sigma n}$  entspricht  $\mathbf{A} \bar{\ddot{u}}_{n-1} \vee \bar{\mathbf{A}} \ddot{u}_{n-1} \rightarrow$  nutzen von **EXOR**

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen



Für eine Addition von  $m$  Ziffern werden  $m$  Volladdierer verbunden.



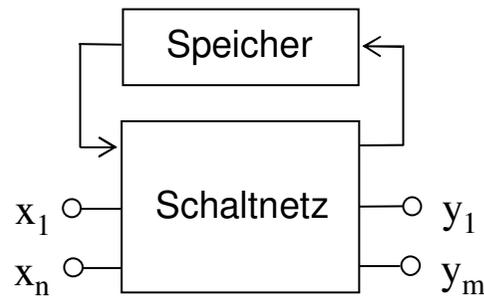
Rechentechnik: weitere kombinatorische Schaltungen (ohne Speicher) z. B. Vergleich ( $>$ ,  $=$ ,  $<$ ), Multiplikation  $\rightarrow$  Linksverschiebung, Division  $\rightarrow$  Rechtsverschiebung oder Komplementbildung.

## Aufgabe 4.2.1

Entwickeln Sie die Belegungstabelle, die Funktion und die Schaltung für einen Vergleich zweier logischer Werte entsprechend  $y = x_1 > x_2$  (Werte jeweils True oder False).

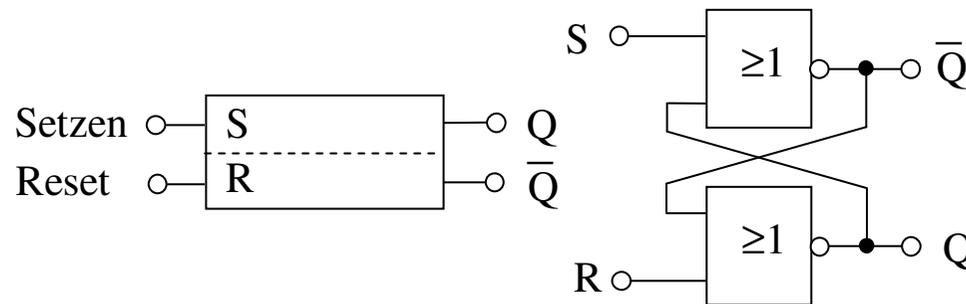
### 4.3 Sequentielle Schaltungen

Sequentielle Schaltungen (**Schaltwerke**) → Schaltnetzen mit **Speichern** (mindestens einem) **zur Rückführung** von Aus- zu den Eingangswerten.



Speicher → in der Elektronik **bistabile Kippstufen** (asynchron oder synchron)

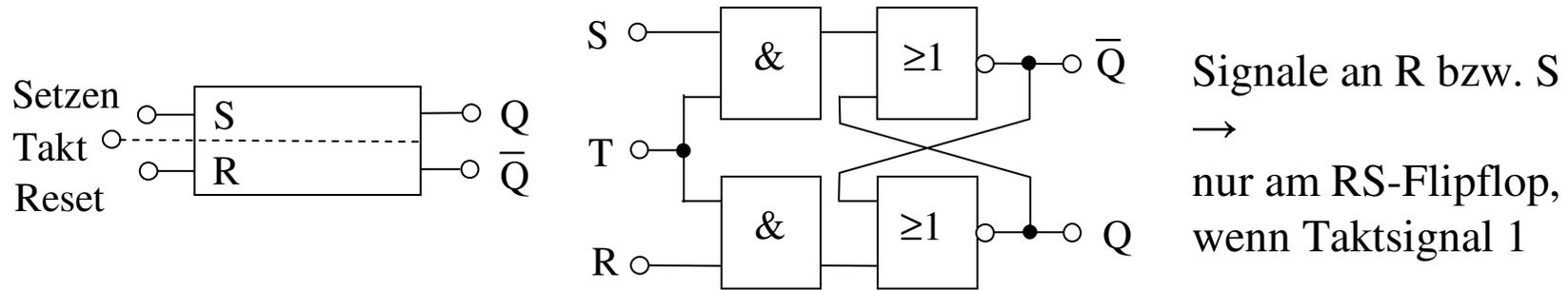
**asynchronen Kippstufen** → RS-Flipflop



Belegungstabelle				
R	S	Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	gespeicherter Wert bleibt
1	0	0	1	Zurücksetzen
0	1	1	0	Setzen
1	1	verboten		← zufällig, undefiniert

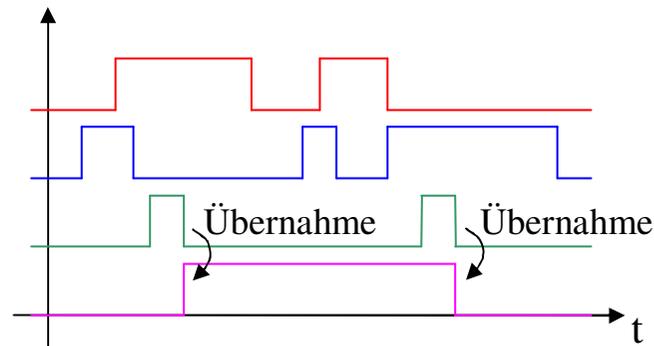
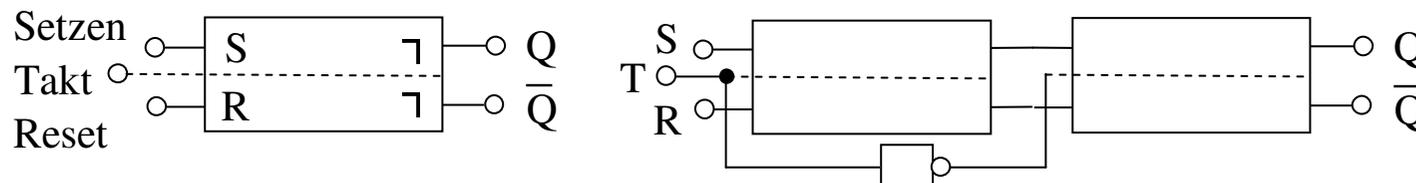
RS-Flipflops werden in der Praxis aus **zwei NOR-Gattern** realisiert und stehen in der Regel nicht als Spezialbausteine zur Verfügung.

**synchrone Kippstufe** → zusätzlich zum RS-Flipflop Taktsteuerung (Auffangflipflop)



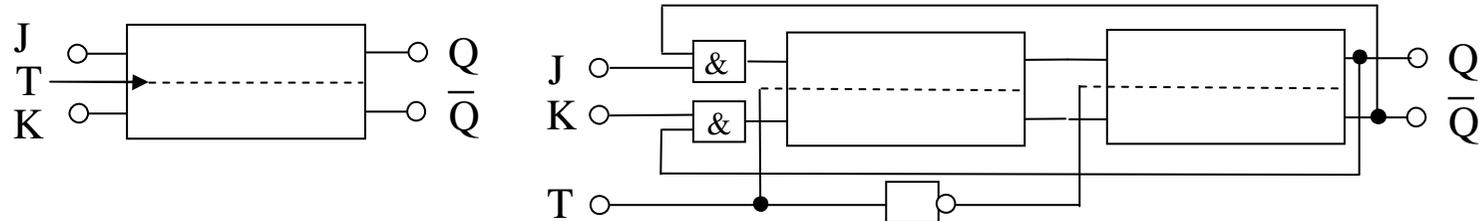
während Taktzeit → der Wert läuft durch mehrere Speicher

**RS-Master-Slave-Flipflop** neben Hauptspeicher ein Zwischenspeicher



- S** Taktsignal „1“ → &-Gatter erstes Flipflop → öffnen
- R** Taktsignal „0“ → &-Gatter zweites Flipflop → öffnen
- T** → Übernahme
- Q** nach wie vor **gleichzeitiges** Auftreten von R = S = 1 (während des Taktes) **verboten**

## JK-Master-Slave-Flipflop

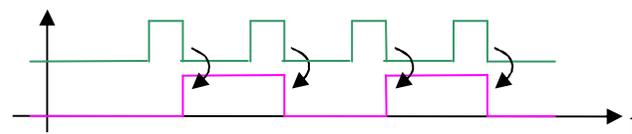
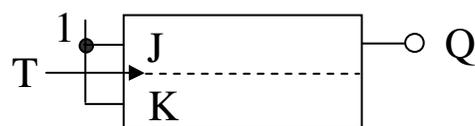


J und K gleichzeitig „1“ → je nach Wert von Q oberes oder unteres &-Gatter geöffnet.  
 Durch die gekreuzte Rückführung erfolgt nach Taktende eine Umkehrung

K	J	Q	
0	0	Q	Speichern Wert wird gehalten
1	0	0	Löschen (Reset)
0	1	1	Setzen
1	1	$\bar{Q}$	Umkehr $\bar{Q}$ gegenüber vor dem Takt

R- und S-Eingänge können als allgemeines Preset und Reset herausgeführt sein.  
 JK-Master-Slave-Flipflop → universell einsetzbares Speicherelement.

### Anwendung für Teilerschaltungen

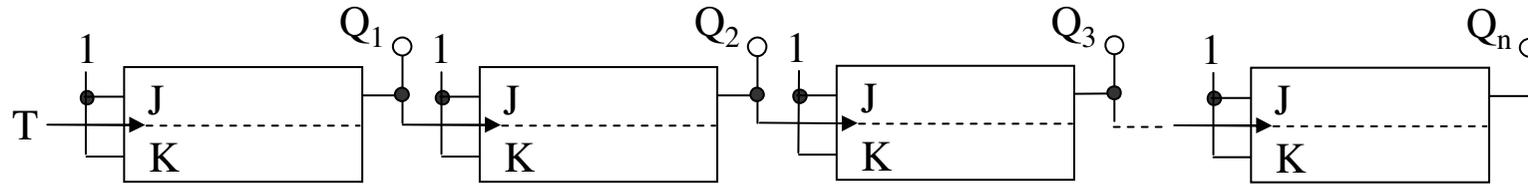


J und K ständig „1“  
 T Ausgangssignal  
 Q → halbe Frequenz

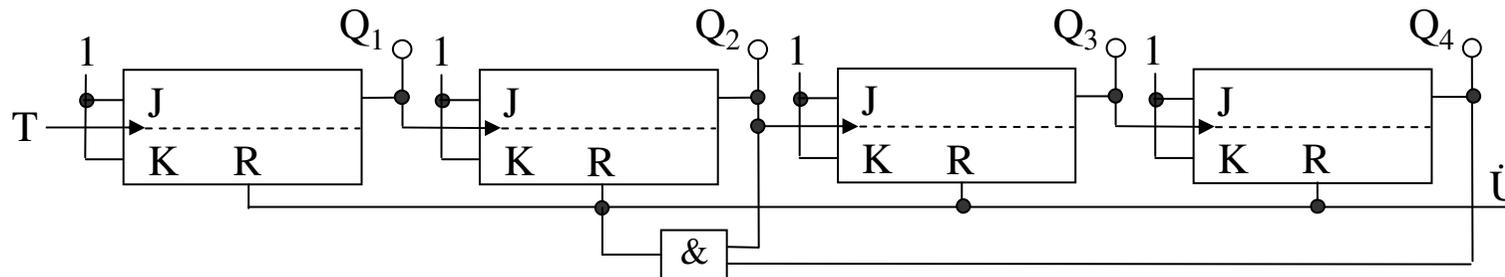
Hintereinander geschaltet → **Teiler für  $f/2^n$**  (auch andere Verhältnisse möglich)

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

## Anwendung für **Zählerschaltungen**

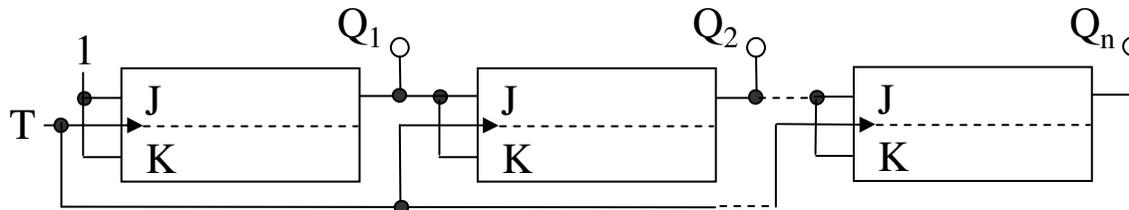


jeder Baustein  $\rightarrow$  Takt halbiert  $\rightarrow$   $Q_n$  bis  $Q_1$   $\rightarrow$  Anzahl Impulse  $\rightarrow$  **Binärzähler**



Ziffer  $10_D$  ( $1010_B$ )  $\rightarrow$  alle Bausteine zurückgesetzt  $\rightarrow$  **Dezimalzähler**

Ein- und Ausschaltzeiten der Bausteine  $\rightarrow$  Laufzeitverzögerungen  
 $\rightarrow$  **asynchrone Zählung**  $\rightarrow$  kurzzeitig falsche Zwischenzustände

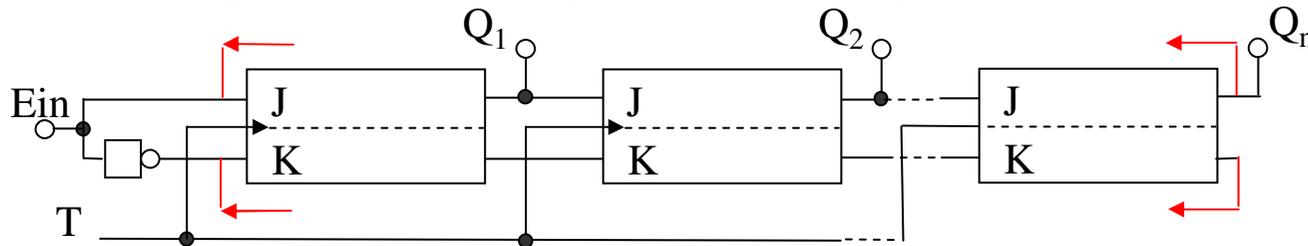


**synchrone Zählung:** nach Taktende  $\rightarrow$  alle Stufen haben aktuellen Wert  
(Laufzeiten sind vor Taktende und das Taktende ist an allen Stufen gleichzeitig)

## Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

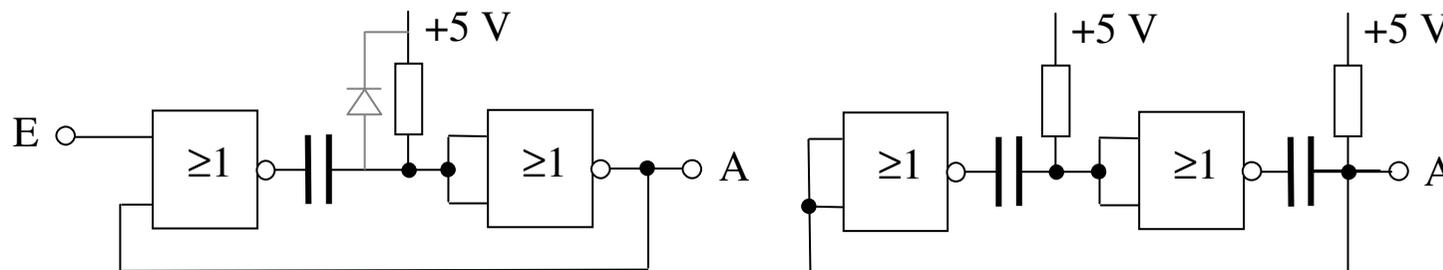
→ **universelle Zähler** mit **Voreinstellung** (zusätzlich → **Vorwärts und Rückwärts**)  
Zähler sind als **Schaltkreise** in vielen Konfigurationen und Varianten verfügbar.

Anwendung für **Schieberegister-** und **Ringzählerschaltungen**



Bit-Muster am Eingang läuft mit Taktsignal Stufe für Stufe durch Schieberegister  
Rückführung → Ringzähler (voreingestelltes Bit-Muster läuft mit Takt im Kreis)

**monostabile** und **astabile** Kippstufen



stabiler und instabiler Zustand

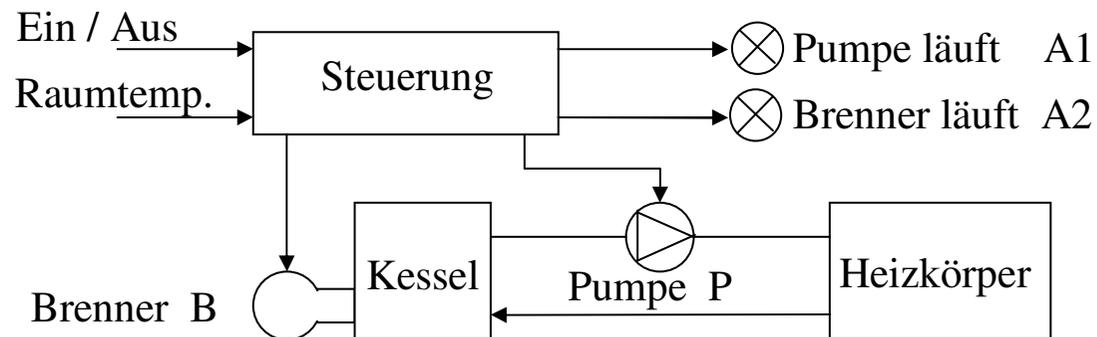
nur instabile Zustände

**Beispiel: Entwurf einer Ablaufsteuerung** (Regelung einer Heizungsanlage)

**Ausgangspunkt** ist **Analyse der Funktion**. Die Hauptfunktionen sind:

- Verarbeiten des manuellen **Ein- und Ausschaltens** der Anlage,
- Steuerung der **Umwälzpumpe** (Einschalten, wenn die Heizung läuft),
- **temperaturabhängige Steuerung des Brenners** und
- Steuerung von **Kontrolllampen** für Pumpe und Brenner.

Im Beispiel Temperaturregelung durch Ein- und Ausschalten des Brenners

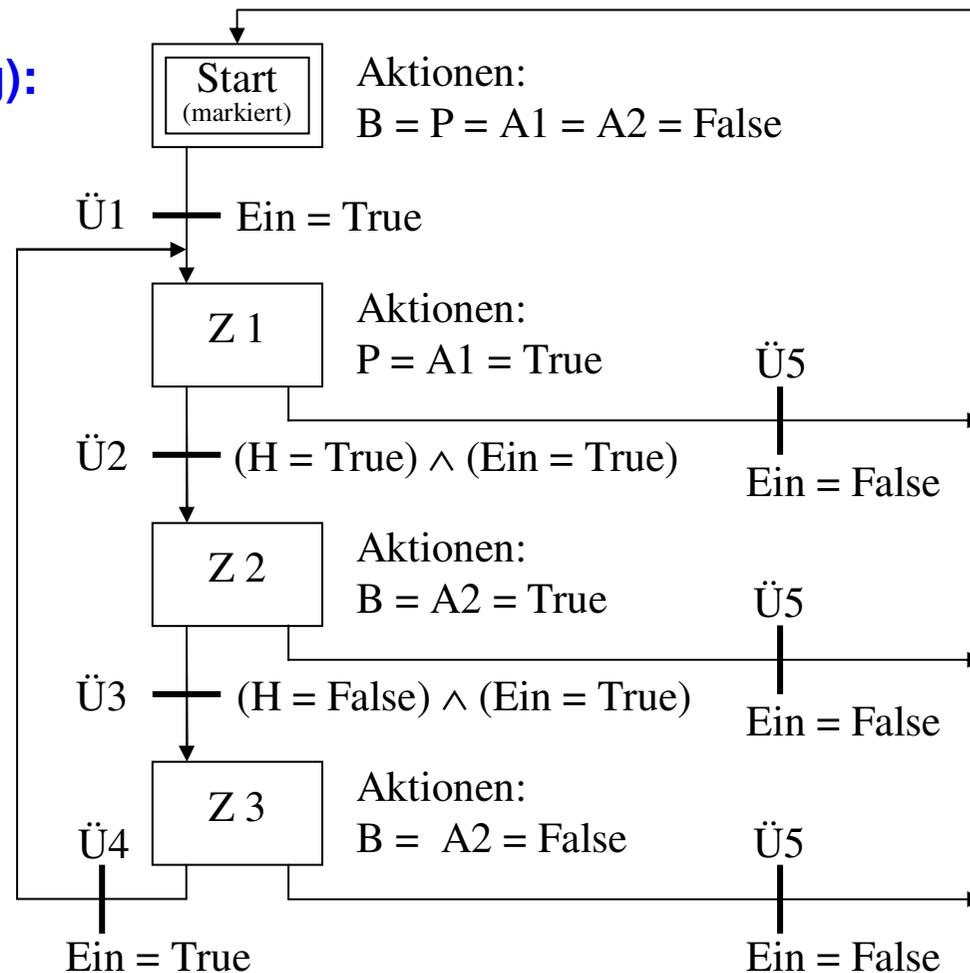


**Funktionsplan der Steuerung** (IEC 61131-3):

- Start: **Startzustand**, Beginn aktiv (markiert) → Aktionen → **Ausgangszustand**
- Z1...Z3 = **Zustände** mit **Aktionen** (mindestens genau einmal durchgeführt)
- Ü1...Ü5 = Übergangsbedingungen (**Transitionen**) → **nächster Zustand**
- Ein = True, wenn die Heizung manuell eingeschaltet wurde, sonst False.
- H = Wärmeanforderung → True  $t_{\text{ist}} < t_{\text{soll untere}}$  und False  $t_{\text{ist}} > t_{\text{soll obere}}$

Transitionen → bei mehreren möglichen nur eine richtig (auch **Prioritäten**)

**Funktionsplan  
(Ablaufsteuerung):**



Auf der Basis des Funktionsplanes kann eine Hardwaresteuerung als **Schrittfolge** (eine Grundform des **Schaltwerkes**) entworfen werden.



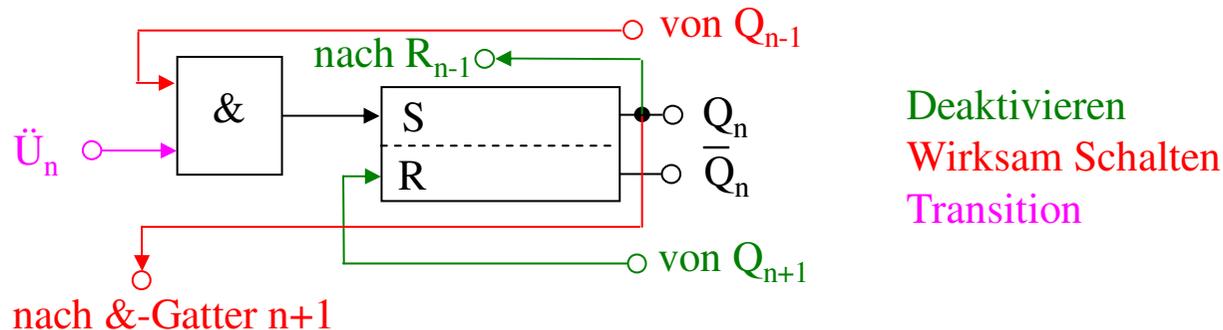
## Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen

Hardwareschritt → **Speicher** zur Aktivierung gesetzt, zur Deaktivierung gelöscht

→ Speicherausgang steuert Aktionen

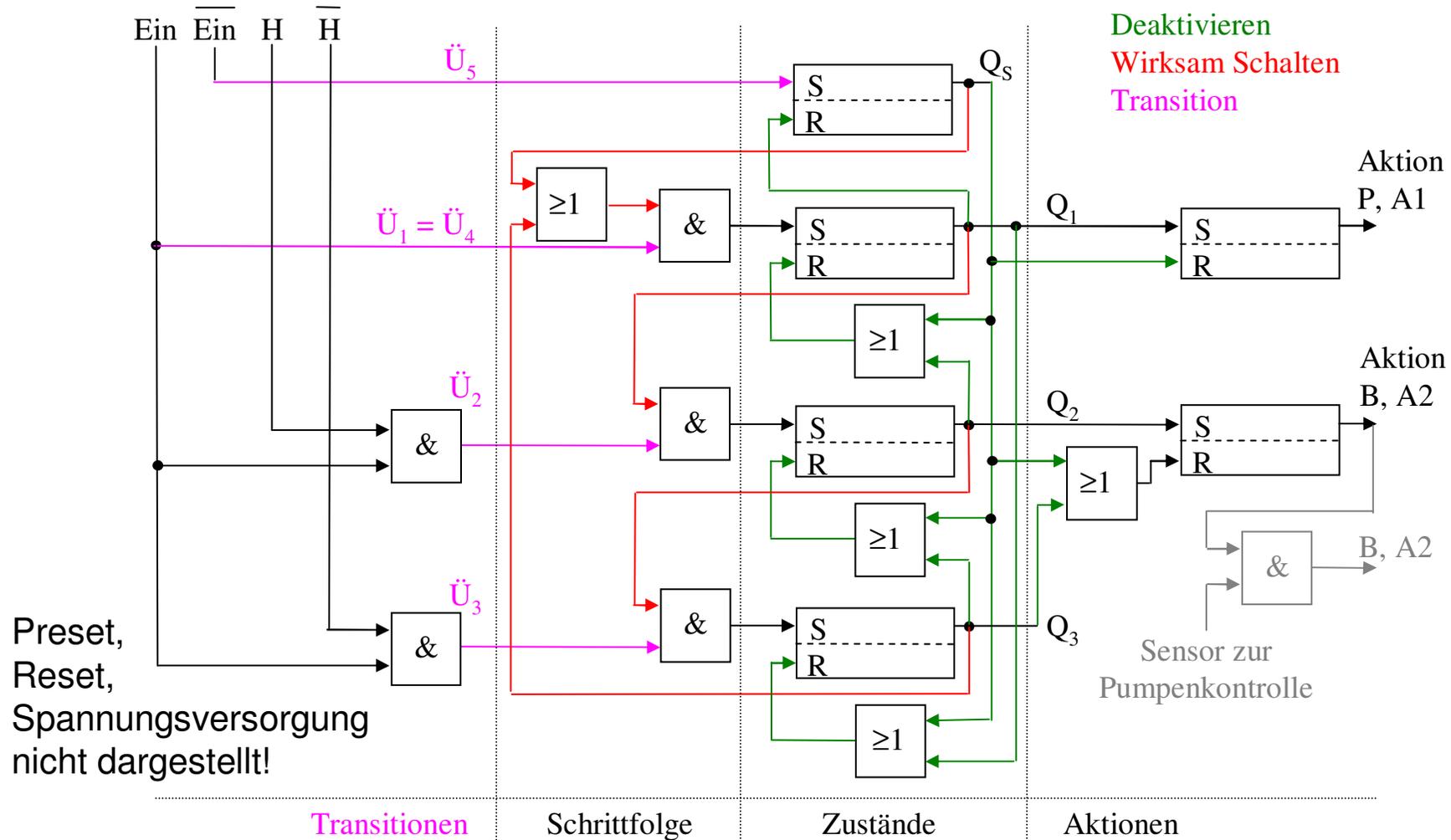
Beim Start (Power-On-Reset):

→ Speicher des ersten Schritts Preset- alle anderen Reset-Impuls



- **vorheriger Schritt aktiv** ( $Q_{n-1} = 1$ ) → **Transition** ( $\ddot{U}_n$ ) über das & kann wirken
- $\ddot{U}_n = 1 \rightarrow S = 1 \rightarrow Q_n = 1 \rightarrow R_{n-1}$  **löscht** den Speicher des **vorherigen Schritts**
- $Q_n = 1 \rightarrow$  Transition des nächsten Schritts kann wirken
- nächster Schritt aktiv ( $Q_{n+1} = 1$ ) → **jetziger Schritt** wird über R **gelöscht**
- $Q_n$  und  $\bar{Q}_n$  **schalten die Aktionen**
- Übergangsbedingungen → kombinatorische Schaltungen
- Aktionen → weiterer Speicher, wenn über aktuellen Schritt hinaus wirksam
- durch kombinatorische Schaltungen zusätzliche Verriegelungen
- mehreren Transitionen können aktivieren → mehrere Vorgänger

# Grundlagen der Halbleiterbauelemente, Analyse analoger und digitaler Schaltungen



mehrere Nachfolger → ODER-Gatter → jeder Zustand kann aktivieren  
 mehrere Vorgänger → ODER-Gatter → jeder Zustand kann löschen  
 Verriegelung für den Fall, dass die Pumpe ausfällt **Grau**



## Richtungen der weiteren Entwicklung

- Dargestellte Bausteine und deren Möglichkeiten → für kleinere Aufgaben sehr vorteilhaft einsetzbar.
- Weitere Entwicklung →
  - **SPLD** (Simple Programmable Logic Device),
  - **PLA** (Programmable Logic Array),
  - **PAL** (Programmable Array Logic),
  - **GAL** (Gate Array Logic),
  - **CPLD** (Complex Programmable Logic Device),
  - **FPGA** (Field Programmable Gate Array) bis zum
  - **ASIC** (Application Specific Integrated Circuit)
- **Hochintegrierten Logikanordnungen**, in der Regel mit Konfigurationsprogramm einmalig frei konfigurierbar
- In Bausteinen je nach Typ relativ große Anzahl Gatter, Speicher ... Ein- und Ausgänge, evtl. A/D- und D/A-Wandler
- **Programmierbare bis prozessorgesteuerte Lösungen** → am Anfang SPS, heute insbesondere für größere Aufgaben → vorteilhaft Industrie-PC
- Aus aufgezeigten Entwicklungen → genormten Programmiersprachen (IEC 61131-3) für SPS
- Viele Lösungen wurden direkt übernommen (z.B. die Schrittkette).

## 5. Beispielaufgabe Prozessinterface für einen PC

Sammeln erster eigener praktischer Erfahrungen → **Projektaufgabe**.  
Immer wieder wird es notwendig sein, **Sensoren** und **Aktoren** an die **Ein- und Ausgänge** von SPS, Feldbusknoten oder PC anzupassen. (Ob neue, günstig verfügbare oder als Ersatz beschaffte Sensoren oder Aktoren → zusätzliches **Interface** erforderlich).

### Gegebene Sensoren:

- Temperatursensor PT 100
- Helligkeitssensor
- Shunt zur Stromüberwachung
- Durchflussmesswertgeber (125 Impulse/Liter und  $R_a = 15 \text{ k}\Omega$  oder  $\infty$ )

### Gegebener Aktor:

- Es muss ein Relais (12 V Steuerspannung, 230 V / 2 A Schaltkontakte) angesteuert werden.

Der an das Interface anzuschließende Prozesseingang wird über **USB** an einen PC angeschlossen und hat **Low-Power-Schotky TTL Ein- und Ausgänge** sowie **A/D-Wandlereingänge** für 0...10 V. Für dieses Gerät stehen ein **OPC-Server** und **Prozessvisualisierungssoftware** zur Verfügung.