

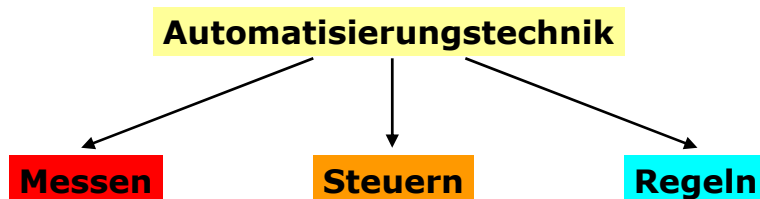
## Steuerungstechnik



Prof. Dr. Ing. Uwe Griebenow

## Steuerungstechnik

Die **Automatisierungstechnik** ist eine Einzelwissenschaft des Ingenieurwesens, die zum Ziel hat, Maschinen oder Anlagen automatisiert, also selbständig und ohne Mitwirkung von Menschen, zu betreiben.



Die **Steuerungstechnik** ist ein Teilgebiet der Automatisierungstechnik.



**Steuer-  
information**



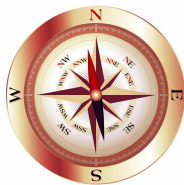
**Eingabe**



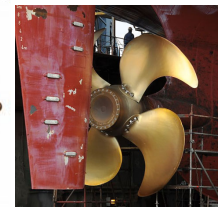
**Steuergerät**



**Stellglied**



**Kurs:  
Süd-West**

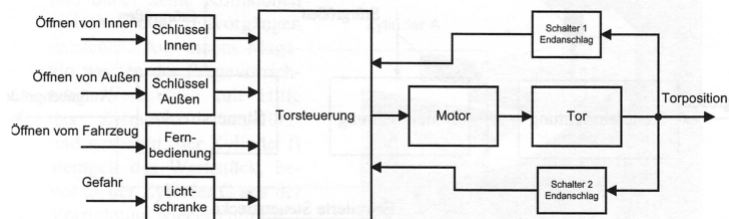


21/01/2021

3

## Definition: DIN 19226

Das **Steuern** – die **Steuerung** – ist der Vorgang in einem System, bei dem eine oder mehrere Größen als Eingangsgrößen, andere Größen als Ausgangsgrößen auf Grund der dem System eigentümlichen Gesetzmäßigkeit beeinflussen. → **offener Wirkungsablauf**



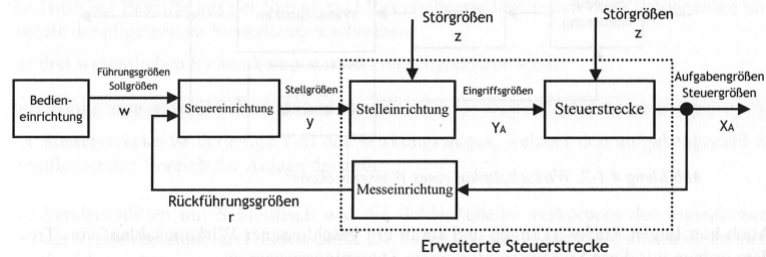
### Torsteuerung

- der Endschalter gibt ein Signal an das Steuerungssystem zurück
- Ist es dann eine Tor-Regelung?

21/01/2021

4

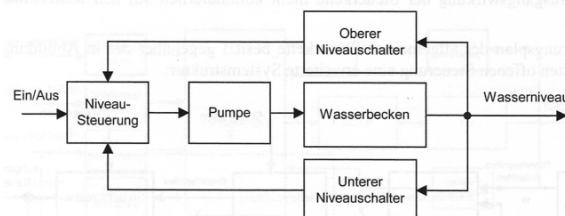
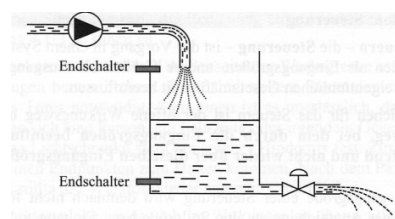
Kennzeichen für das Steuern ist der offene Wirkungsweg oder ein **geschlossener** Wirkungsweg, bei dem durch die Eingangsgrößen beeinflusste Ausgangsgrößen **nicht fortlaufend und nicht wieder** über dieselben Eingangsgrößen auf sich selbst wirken.



Die geschlossene Steuerkette

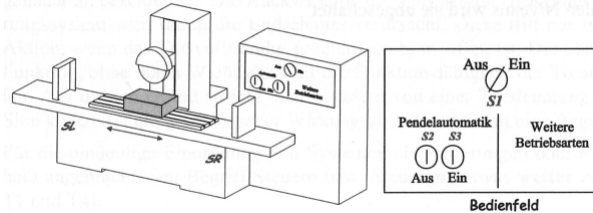
Definition: Die **Steuerstrecke** ist derjenige Teil des Wirkungsweges, welcher den aufgabengemäß zu beeinflussenden Bereich der Anlage darstellt.

Beispiel: Füllstandssteuerung eines Wasserbeckens



Wirkschluplan

Die **Stelleinrichtung** wirkt unmittelbar auf die Steuerstrecke ein. In aller Regel setzt sie sich aus Stellgliedern und Stellantrieb zusammen.



Pendelschleifmaschine

**Beispiel:** Pendelschleifmaschine

**Steuereinrichtung:** elektronische Schaltung,  
Eingangssignale → Verknüpfung → Ausgangssignale zur Ansteuerung

**Stelleinrichtung:** Ansteuerungselektrik und Motoren

**Soll- bzw. Führungsgröße w:**

- Vorgabegrößen für die gesamte Steuerung

**Stellgröße bzw. Eingriffsgröße y:**

- Ausgangsgrößen der Steuereinrichtung

**Aufgabengröße  $X_A$  bzw. Steuergröße x:**

- Ausgangsgrößen der Steuerstrecke

**Rückführungsgröße r:**

- Überwachung der Steuerstrecke **ohne ständigen** Vergleich mit der Führungsgröße

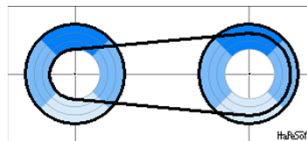
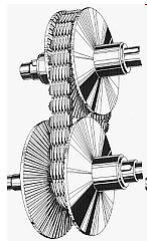
**Störgröße z:**

- von außen auf die Steuerung wirkende Größe

## Steuerungsarten:

### 1. Analoge Steuerung

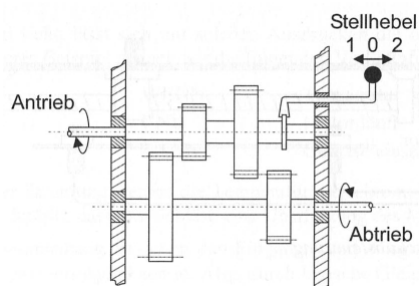
- verknüpft analoge Ein- und Ausgangssignale durch eine sich kontinuierlich ändernde Steuerungsvorschrift



Analoge mechanische Steuerung: stufenloses Getriebe

## 2. Binäre Steuerung

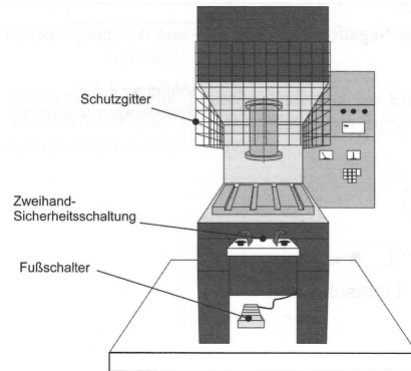
- verarbeiten vorwiegend zweiwertige logische Informationen durch Boole'sche Gesetzmäßigkeiten



Schaltgetriebe als Binärsteuerung

## 3. Kombinatorische Steuerung

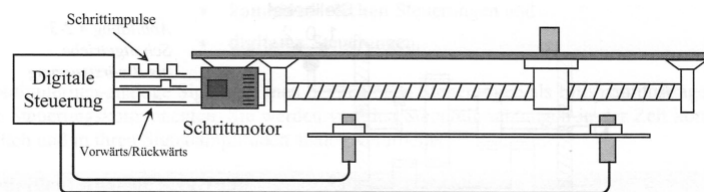
- zur Funktionsfähigkeit einer Anlage werden mehrere binäre Eingangssignale benötigt → z. B. bei Sicherheitsmaßnahmen



Sicherheitstechnik bei Werkzeugmaschinen

## 4. Digitale Steuerung

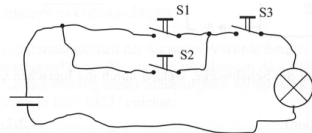
- verarbeitet numerische Werte,  
z.B. Zufahrtssteuerung zu einem Parkhaus, CNC-Maschine



Digitale Vorschubsteuerung

Unterteilung nach der Art der Signalverarbeitung

**Verknüpfungssteuerung**



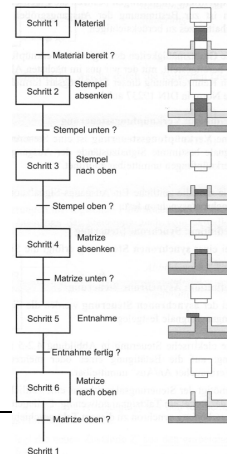
WENN Schalter 1 gedrückt ODER Schalter 2 gedrückt  
 UND Schalter 3 gedrückt  
 DANN Licht an.

**Eingangssignale**

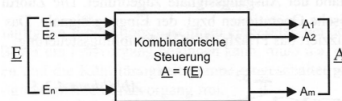
**Boolesche Verknüpfung**

**Ausgangssignale**

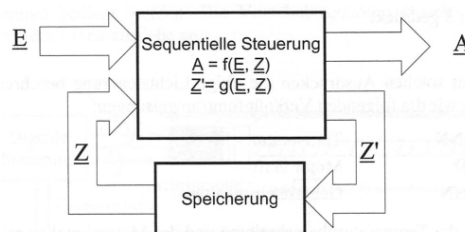
**Ablaufsteuerung**



Ein **Schaltnetz** verknüpft die Ein- & Ausgänge einer Steuerung durch logische Funktionen  $\underline{A}=f(\underline{E})$ , den sogenannten Schaltfunktionen.



**Schaltwerke (Automaten)** sind Realisierungen von Steuerungen mit logischen Funktionen und zusätzlichen Speicherfunktionen, derart, dass die Ausgänge der Steuerung auch vom aktuellen Schaltwerk-zustand abhängen.



### Automaten

#### Mealy-Automaten

- Zwischenspeicherung der Zustände  $\underline{Z}$
- besitzen ein „Gedächtnis“
- Ausgangswerte hängen von den **momentanen Zuständen** und den **aktuellen Eingangswerten** ab
- sequentielle Steuerungen

#### Moore-Automaten

- Ausgangsvariablen sind unabhängig von den momentanen Eingängen
- Ausgangswerte werden **allein** von **inneren Zuständen** bestimmt

Die **Ausgangsfunktion  $f(\underline{E}, \underline{Z})$**  beschreibt, wie sich die Ausgänge  $\underline{A}$  aus den momentan gespeicherten Zuständen  $\underline{Z}$  und den Eingangswerten  $\underline{E}$  berechnen.

Die **Zustandsübergangsfunktion  $g(\underline{E}, \underline{Z})$**  legt die neuen Zustände  $\underline{Z}'$  aus den gespeicherten Zuständen  $\underline{Z}$  und den Eingangswerten  $\underline{E}$  fest.

#### Sequentielle Steuerung:

- vorheriger Zustand des Schaltnetzes ist zu berücksichtigen

#### Synchrone Steuerung:

- Signalverarbeitung erfolgt synchron zu einem Taktsignal

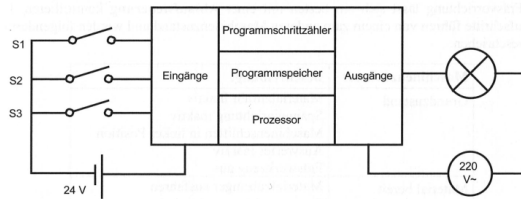
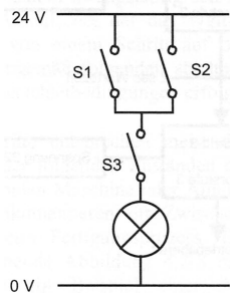
#### Asynchrone Steuerung:

- Signaländerungen am Ausgang werden nur durch die Eingangssignale festgelegt

**Steuerprogramm**

**Verbindungsprogrammierte Steuerung**

**Speicherprogrammierte Steuerung (SPS)**



**Verbindungsprogrammierte Steuerung:**

- feste Verdrahtung zwischen Schaltern und Stromverbraucher

**Nachteile:**

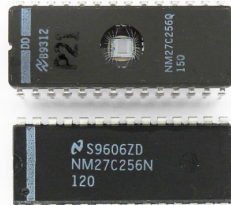
- Änderungsmöglichkeit von „Programmen“ ist sehr begrenzt
- hoher Aufwand an Dokumentation, Schaltpläne müssen bei Programmänderungen neu aufgelegt werden

**Vorteile:**

- schnelle Reaktionszeiten der „Steuerprogramme“ durch unmittelbare und parallele Signalverarbeitung
- es sind keine Programmiersprachen erforderlich
- klassische Methoden der Elektrotechnik, der Pneumatik und der Hydraulik reichen zur „Programmierung“ aus

### Speicherprogrammierbare Steuerung (SPS):

- Mikrocomputersteuerungen und deren Programme speziell für steuerungstechnische Anwendungen entwickelt
- frei programmierbare Steuerungen verwenden Schreib-Lese-Speicher (RAM oder EEPROM) → Inhalte können verändert werden  
*EEPROM = electrically erasable programmable read-only memory*
- austauschbare Steuerungen verwenden nur Lese-Speicher (ROM, PROM oder EPROM)



EPROM

### Speicherprogrammierbare Steuerung (SPS):

#### Nachteile:

- endliche Verarbeitungsgeschwindigkeit, hängt von der Größe des Steuerprogramms ab
- Steuersignale können nur bis zu einer maximalen Frequenz erfasst und ausgegeben werden

#### Vorteile:

- anschlussfertige universelle Geräte sind für viele Aufgaben verwendbar
- Programmänderungen sind mit Programmiergeräten einfach realisierbar
- kostengünstigere Lösung für komplexe Steuerungsaufgaben
- Dokumentation der Programme wird automatisch erstellt

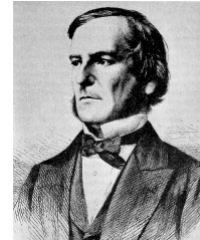
## Grundfunktionen für kombinatorische Steuerungen:

### Definition:

Die **Schaltalgebra** bzw. **Aussagelogik** beschreibt die Verknüpfung wahrer und unwahrer Aussagen.

### → Boole'sche Algebra

Eine **logische Variable** der Boole'schen Algebra stellt eine zweiwertige Größe mit den beiden Zuständen **0** und **1** (Ziffersymbole) dar.



George Boole 1860

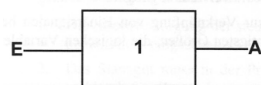
**Axiom 1:** Zustand  $x=0$  gilt genau dann, wenn  $x \neq 1$  und Zustand  $x=1$  gilt genau dann, wenn  $x \neq 0$ .

**Axiom 2:** Zustände 0 und 1 gehen durch Negation auseinander hervor:

$$\bar{0} = 1 \text{ bzw. } \bar{1} = 0$$

Das Ausgangssignal A der **Identität** besitzt genau dann den Wert 1, wenn das Eingangssignal E den Wert 1 hat.

Schaltfunktion:  $A=E$  mit  $E=0$  oder **1**



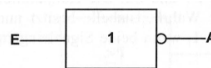
Symbol

E	A
0	0
1	1

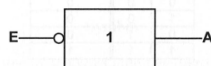
Wahrheitstabelle

Das Ausgangssignal A der **Negation** besitzt genau dann den Wert 1, wenn das Eingangssignal E den Wert 0 aufweist und umgekehrt.

Schaltfunktion:  $A = \bar{E}$



oder



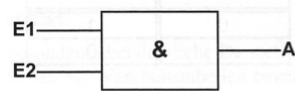
Symbol

E	A
0	1
1	0

Wahrheitstabelle

Das Ausgangssignal einer **Konjunktion** – UND-Verknüpfung – besitzt nur dann den Wert **1**, wenn alle Eingangssignale den Wert **1** besitzen.

Schaltfunktion:  $A = E1 \wedge E2$  bzw.  $A = E1 \cdot E2$



Symbol

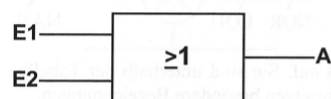
E1	E2	A
0	0	0
1	0	0
0	1	0
1	1	1

Wahrheitstabelle

Das Ausgangssignal einer **Disjunktion** – ODER-Verknüpfung – besitzt den Wert **1**, wenn mindestens ein Eingangssignal den Wert **1** hat.

Schaltfunktion:  $A = E1 \vee E2$  bzw.  $A = E1 + E2$

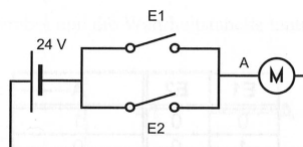
## Symbol der Disjunktion:



Symbol

E1	E2	A
0	0	0
1	0	1
0	1	1
1	1	1

Wahrheitstabelle



Elektrische ODER-Schaltung

## Erweiterte logische Grundfunktionen:

- für die 4 Eingangskombinationen von 2 Eingangswerten ergeben sich  $4^2=16$  Ausgangskombinationen

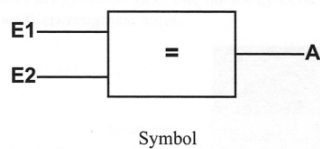
E1	E2	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Weitere wichtige Steuerfunktionen:

XOR    NOR    EQU    NAND

Das Ausgangssignal einer **Äquivalenz** - GLEICH-Verknüpfung - besitzt dann den Wert **1**, wenn beide Eingangssignale denselben Wert **0** oder **1** haben.

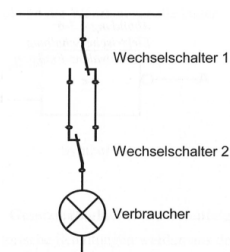
Schaltfunktion:  $A = (E1 \wedge E2) \vee (\overline{E1} \wedge \overline{E2}) \equiv E1 \leftrightarrow E2$



E1	E2	A
0	0	1
1	0	0
0	1	0
1	1	1

Wahrheitstabelle

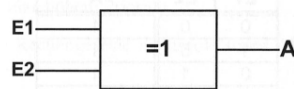
- aus der Elektroinstallation ist diese Schaltung als Wechselschaltung bekannt



Wechselschaltung

Das Ausgangssignal einer **Antivalenz** – Exklusiv-ODER-Verknüpfung – besitzt dann den Wert **1**, wenn beide Eingangssignale einen ungleichen Wert haben. → **XOR**

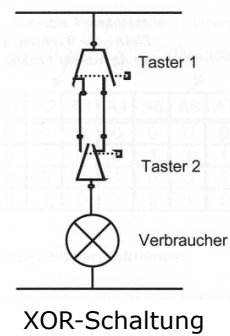
Schaltfunktion:  $A = (E1 \wedge \overline{E2}) \vee (\overline{E1} \wedge E2) \equiv E1$



Symbol

E1	E2	A
0	0	0
1	0	1
0	1	1
1	1	0

Wahrheitstabelle

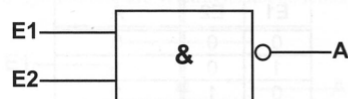


XOR-Schaltung

Nicht-UND-Verknüpfung: **NAND**

Das Ausgangssignal einer **Nicht-UND-Verknüpfung** besitzt dann den Wert **1**, wenn beide Eingangssignale nicht gleichzeitig den Wert **1** haben.

Schaltfunktion:  $A = \overline{(E1 \wedge E2)} \equiv E1 \overline{E2}$



Symbol

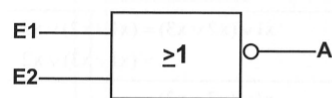
E1	E2	A
0	0	1
1	0	1
0	1	1
1	1	0

Wahrheitstabelle

Nicht-ODER-Verknüpfung: **NOR**

Das Ausgangssignal einer **Nicht-ODER-Verknüpfung** besitzt dann den Wert **1**, wenn beide Eingangssignale gleichzeitig den Wert **0** haben.

Schaltfunktion:  $A = \overline{(E1 \vee E2)} \equiv E1 \bar{\vee} E2$



Symbol

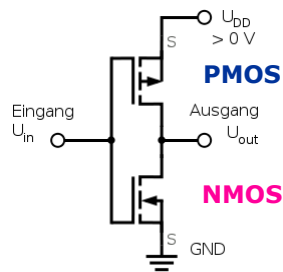
E1	E2	A
0	0	1
1	0	0
0	1	0
1	1	0

Wahrheitstabelle

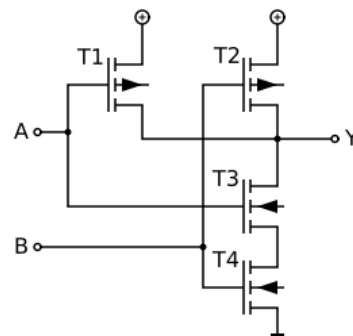
**Gesetzmäßigkeiten der Schaltalgebra**

- die Negation besitzt die höchste Priorität
- die Konjunktion, Disjunktion, NAND, NOR müssen **vor** der Äquivalenz und Antivalenz ausgeführt werden
- die Konjunktionen (AND, NAND) besitzen gegenüber den Disjunktionen (OR, NOR) eine höhere Priorität

## Gatter in CMOS Technologie



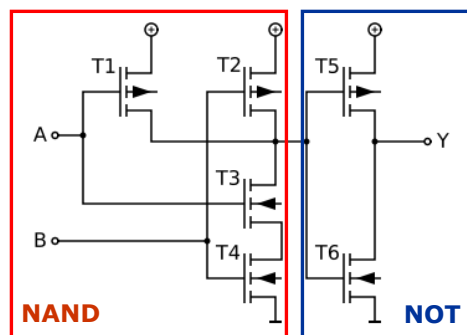
**Inverter  
NOT Gatter**



**NAND Gatter**

Liegt an Eingang A und B High-Potential, leiten T3 und T4, wobei T1 und T2 sperren. Somit liegt am Ausgang Y Low-Potential an.

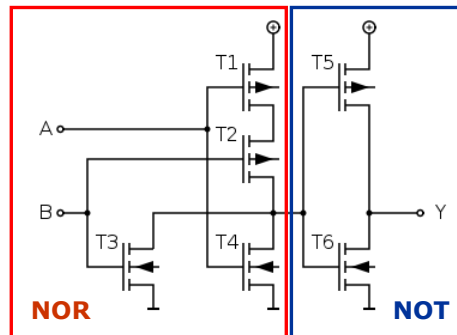
## Gatter in CMOS Technologie



**UND-Gatter**

Liegt an den Eingängen A und B High-Potential an, dann leiten T3 und T4, wobei T1 und T2 sperren. Dadurch liegt an T5 und T6 Low-Potential an und T5 leitet und T6 sperrt, weswegen am Ausgang Y High-Potential anliegt.

Bei allen anderen Eingangszuständen liegt Low-Potential am Ausgang, weil T6 leitet.



## ODER-Gatter

Liegt z.B. an Eingang A oder B High-Potential an, dann leitet T3 bzw. T4, wobei T1 bzw. T2 sperrt.

## Gesetzmäßigkeiten der Schaltalgebra

Neutralität	$x1 \wedge 1 = x1$	$x1 \vee 1 = 1$
0-Regel	$x1 \wedge 0 = 0$	$x1 \vee 0 = x1$
Idempotenz-Gesetz	$x1 \wedge x1 = x1$	$x1 \vee x1 = x1$
Komplement-Gesetz	$x1 \wedge \overline{x1} = 0$	$x1 \vee \overline{x1} = 1$
Doppeltes Komplement	$\overline{\overline{x}} = x$	

## Gesetzmäßigkeiten der Schaltalgebra

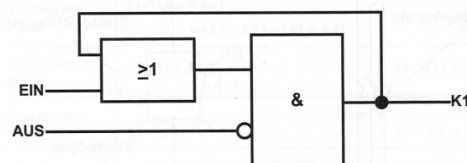
Kommutativ-Gesetz	$x1 \wedge x2 = x2 \wedge x1$	$x1 \vee x2 = x2 \vee x1$
Assoziativ-Gesetz	$x1 \wedge (x2 \wedge x3) = (x1 \wedge x2) \wedge x3$ $= (x1 \wedge x3) \wedge x2$	$x1 \vee (x2 \vee x3) = (x1 \vee x2) \vee x3$ $= (x1 \vee x3) \vee x2$
Distributives Gesetz	$x1 \wedge (x2 \vee x3) = (x1 \wedge x2) \vee (x1 \wedge x3)$	$x1 \vee (x2 \wedge x3) = (x1 \vee x2) \wedge (x1 \vee x3)$
Absorptions-Gesetz	$x1 \wedge (x1 \vee x2) = x1$	$x1 \vee (x1 \wedge x2) = x1$
Gesetze von De Morgan	$\overline{x1 \wedge x2} = \overline{x1} \vee \overline{x2}$	$\overline{x1 \vee x2} = \overline{x1} \wedge \overline{x2}$

Die Gesetze von De Morgan besagen, dass eine Schaltfunktion insgesamt negiert werden kann, indem man die logischen Variablen einzeln negiert und gleichzeitig die Verknüpfung der Konjunktion durch die Disjunktion austauscht.

- für sequentielle Steuerungen werden Selbsthaltefunktionen benötigt
- während der einzelnen Prozessschritte müssen die gerade vorliegenden Prozesszustände festgeschrieben werden
- Gedächtniselemente müssen zur Verfügung stehen
- Selbsthaltefunktion

EIN	AUS	K1
0	0	Vorheriges K1
1	0	1
0	1	0
1	1	0

Wahrheitstabelle der Selbsthaltefunktion



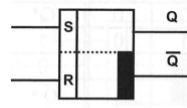
Logikschaltung für die Selbsthaltefunktion

**Definition: Flipflop-Schaltung**

Speicherschaltungen mit zwei festen Ausgangszuständen, die nicht nur vom Zustand der Eingangssignale abhängen, sondern auch vom Ausgang der Schaltung, bezeichnet man als **Flipflop-Schaltungen** (bistabile Kippstufen).

**RS-Flipflop (Reset-Set-Flipflop):**

Bei einem RS-Speicherelement (RS-Flipflop) nimmt der Ausgang Q den Signalzustand **1** an, wenn der Setzeingang S kurzzeitig ein **1**-Signal führt. Der **1**-Zustand am Ausgang Q wird solange gespeichert, bis am Eingang R ein **1**-Signal angelegt wird.



RS-Flipflop

- wird am S-Eingang und am R-Eingang ein Signal mit dem Wert **1** angelegt, ist das Ausgangssignal unbestimmt

→ **dominierend setzendes** RS-Flipflop → **Q=1**

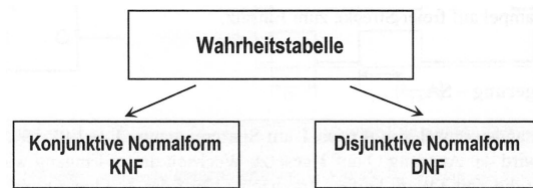
→ **dominierend rücksetzendes** RS-Flipflop → **Q=0**

S	R	Q
0	0	Q*
1	0	1
0	1	0
1	1	0

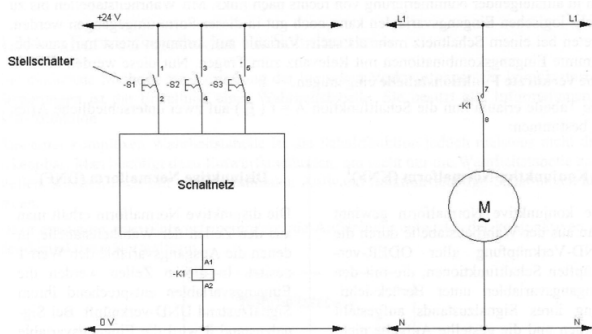
Symbol und Schalttable des RS-Flipflops

**Entwurfsmethoden für Schaltnetze**

- einfachste Methode zur Ermittlung der logischen Verknüpfungen für kombinatorische Steuerungen ist die Erstellung einer Wahrheitstabelle
  - aus einer komplexen Wahrheitstabelle läßt sich die Schaltfunktion meist nicht direkt erkennen
- 2 Entwurfsmethoden: **Konjunktive & Disjunktive Normalform**



**Beispiel: Schaltnetz für eine Motorsteuerung**



S3	S2	S1	K1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Schalttabelle

**Disjunktive Normalform:**

Für jede Zeile mit K1=1 wird eine Konjunktion gebildet. → **Minterme**

Hier z.B. für Zeile 2:  $S1 \wedge \overline{S2} \wedge \overline{S3}$

## Konjunktive Normalform:

Für jede Zeile mit  $K1=0$  wird eine Disjunktion gebildet. → **Maxterme**

Hier z.B. für Zeile 3:  $\overline{S1} \vee \overline{S2} \vee S3$

S3	S2	S1	K1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Bildung der KNF

Bildung der DNF

### Konjunktive Normalform (KNF)

Maxterme:

- 1)  $S1 \vee S2 \vee S3$
- 2)  $S1 \vee \overline{S2} \vee S3$
- 3)  $S1 \vee S2 \vee \overline{S3}$
- 4)  $\overline{S1} \vee S2 \vee \overline{S3}$

21/01/2021

### Disjunktive Normalform (DNF)

Minterme:

- 1)  $S1 \wedge \overline{S2} \wedge \overline{S3}$
- 2)  $S1 \wedge S2 \wedge \overline{S3}$
- 3)  $\overline{S1} \wedge S2 \wedge S3$
- 4)  $S1 \wedge S2 \wedge S3$

41

Die **Schaltfunktion** ergibt sich durch:

- 1: Disjunktive (ODER) Verknüpfung aller Minterme
- 2: Konjunktive (UND) Verknüpfung aller Maxterme

### Konjunktive Normalform (KNF)

Schaltfunktion:

$$K1 = (S1 \vee S2 \vee S3) \wedge (S1 \vee \overline{S2} \vee S3) \wedge (S1 \vee S2 \vee \overline{S3}) \wedge (\overline{S1} \vee S2 \vee \overline{S3})$$

### Disjunktive Normalform DNF

Schaltfunktion:

$$K1 = (S1 \wedge \overline{S2} \wedge \overline{S3}) \vee (S1 \wedge S2 \wedge \overline{S3}) \vee (\overline{S1} \wedge S2 \wedge S3) \vee (S1 \wedge S2 \wedge S3)$$

Die **konjunktive Normalform** ist dann vorzuziehen, wenn die Ausgangsvariable A in der vollständigen Funktionstabelle mehr den Signalzustand **1** als den Wert **0** aufweist.

21/01/2021

42

## 2. Vereinfachung mit Hilfe der Schaltalgebra

Kommutativgesetz:

$$K1 = (S1 \vee S3 \vee S2) \wedge (S1 \vee S3 \vee \overline{S2}) \wedge (S1 \vee S2 \vee \overline{S3}) \wedge (\overline{S1} \vee S2 \vee \overline{S3})$$

Distributivgesetz:

$$K1 = (S1 \vee S3) \wedge (S2 \vee \overline{S2}) \wedge (S1 \vee \overline{S1}) \wedge (S2 \vee \overline{S3})$$

Komplementgesetz:

$$K1 = [(S1 \vee S3) \wedge 1] \wedge [1 \wedge (S2 \vee \overline{S3})]$$

**Endgültige Schaltfunktion:**

$$K1 = (S1 \vee S3) \wedge (S2 \vee \overline{S3})$$

Kommutativgesetz:

$$K1 = (S1 \wedge \overline{S3} \wedge \overline{S2}) \vee (S1 \wedge \overline{S3} \wedge S2) \vee (\overline{S1} \wedge S2 \wedge S3) \vee (S1 \wedge S2 \wedge S3)$$

Distributivgesetz:

$$K1 = (S1 \wedge \overline{S3}) \wedge (\overline{S2} \vee S2) \vee (\overline{S1} \vee S1) \wedge (S2 \wedge S3)$$

Komplementgesetz:

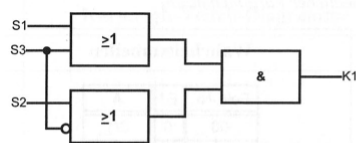
$$K1 = [(S1 \wedge \overline{S3}) \wedge 1] \vee [1 \wedge (S2 \wedge S3)]$$

**Endgültige Schaltfunktion:**

$$K1 = (S1 \wedge \overline{S3}) \vee (S2 \wedge S3)$$

**Endgültige Schaltfunktion:**

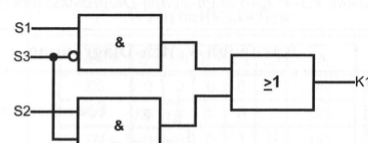
$$K1 = (S1 \vee S3) \wedge (S2 \vee \overline{S3})$$



Konjunktive Verknüpfung

**Endgültige Schaltfunktion:**

$$K1 = (S1 \wedge \overline{S3}) \vee (S2 \wedge S3)$$



Disjunktive Verknüpfung

*Schaltnetze für die Motorsteuerung*

- für komplexere Steuerungen ist die Boole'sche Algebra sehr aufwendig
- Graphische Methode zur Vereinfachung logischer Funktionen

→ **Karnaugh-Veitch-Diagramm (KV-Diagramm)**

## Steuerungstechnik – Karnaugh Diagramm

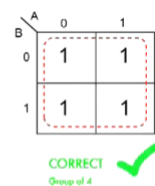
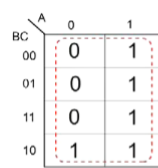
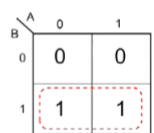
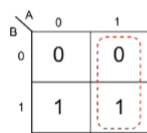
- in den KV-Diagrammen werden die Ausgangswerte der Wahrheitstabellen (**0** oder **1**) auf die entsprechenden Felder geschrieben
- anschließend werden Felder, die eine **1** enthalten und nebeneinander oder untereinander stehen (aber **nicht** diagonal) markiert

Dabei gelten folgende Regeln:

1. Felder müssen eine **1** enthalten (Rechtecke sind möglich)

### Karnaugh Maps - Rules of Simplification A2 Unit 1 - Electronic & Microelectroni

1. Groups may not include any cell containing a zero

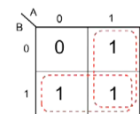
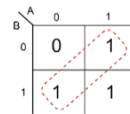


## Steuerungstechnik – Karnaugh Diagramm

2. Felder dürfen **nicht diagonal** zueinander stehen

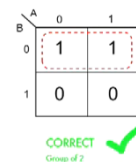
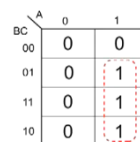
### Karnaugh Maps - Rules of Simplification A2 Unit 1 - Electronic & Microelectroni

2. Groups may be horizontal or vertical, but **not diagonal**.



3. Die Gruppen müssen  $2^n$  Felder mit einer **1** enthalten, also 2, 4, 8, 16 ...

3. Groups must contain  $2^n$  cells. ie. 1,2,4,8 etc cells

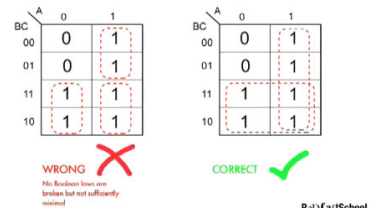


## Steuerungstechnik - Karnaugh Diagramm

4. Die Gruppen sollten so groß wie möglich sein

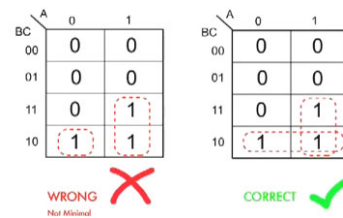
→ kleinere Gruppen sind nicht falsch, es wird aber nicht die maximale Vereinfachung erreicht

4. Each group should be as large as possible resulting in as few groups as possible



5. Jede Zelle mit einer 1 sollte Teil einer Gruppe sein (wenn es möglich ist)

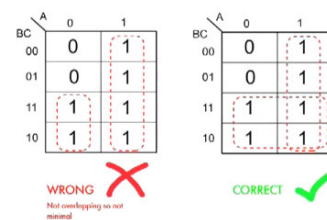
5. Each cell containing a one must be part of a group if possible



## Steuerungstechnik - Karnaugh Diagramm

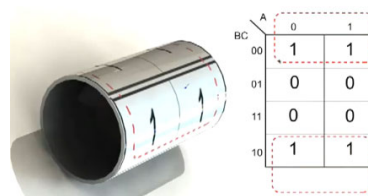
6. Die Gruppen können überlappen

6. Groups may overlap



7. Die Gruppen können über den Rand hinaus gehen  
- wrap around the table

7. Groups may wrap around the table

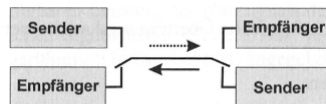


Bei der **Vollduplex-Betriebsart** (Gegenbetrieb) erfolgt die Signalübertragung gleichzeitig in beide Richtungen.  
 Beispiel: RS232-Schnittstelle

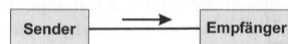


RS232-Schnittstelle

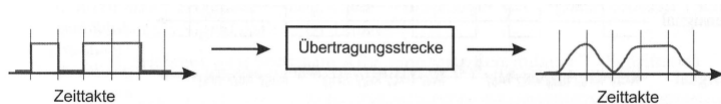
Der Nachrichtenaustausch beim **Halbduplex-Betrieb** (Wechselbetrieb) erfolgt bidirektional alternierend, d.h. zeitlich nacheinander.  
 Beispiel: Internet, Sprechfunk bei Noteinsätzen,  
 (Jede Unterhaltung unter Menschen sollte von dieser Art sein.)



Die Nachrichten werden beim **Simplex-Betrieb** (Richtungsbetrieb) nur in eine Richtung gesendet.  
 Beispiel: Rundfunkübertragung, Messwertaufnehmer, Mikrofon



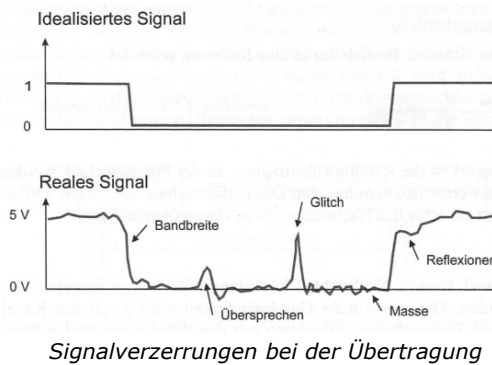
Die maximale Datenübertragungsgeschwindigkeit wird durch die physikalischen Eigenschaften des Kanals mit den dabei auftretenden Störmöglichkeiten bestimmt.  
 z.B. Kabelkapazität & -induktivität



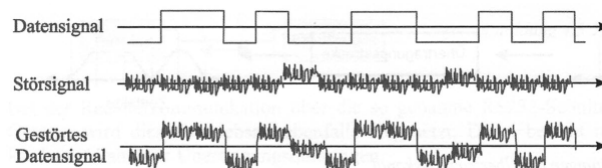
Die **Bandbreite B** [Einheit: Hertz] eines Übertragungskanal ist die Differenz zwischen der maximal und der minimal möglichen Frequenz eines Nachrichtensignals.

Weitere geschwindigkeitsbegrenzende Faktoren:

- Reflexionen an Kabelenden und -übergängen
- Dispersion (frequenzabhängige Wellenausbreitung)
- Übersprechen von Signalen benachbarter Leitungen
- elektromagnetische Einkopplung z.B. „Glitch“



Das **Signal-Rausch-Verhältnis (SNR)** ist ein Maß für die Qualität eines Signals. Es ist das Verhältnis der verfügbaren Leistung des Nutzsignals der Signalquelle zur Rauschleistung der Quelle. Es wird als absolute Zahl oder logarithmisch in Dezibel (dB) angeben.



Beispiel für das logarithmische Maß der Intensität:

$$L_I = 10 \cdot \lg \frac{I}{I_0}$$

Die **Kanalkapazität C** ist der maximal mögliche noch störungsfrei zu übertragende Informationsfluß, angegeben in Bit/s.

- C ist abhängig von der Bandbreite B und dem SNR

Bei einer Gaußverteilung der Amplitudendichte gilt nach Shannon:

$$C = B \cdot \log_2(1 + S/N) = B \cdot \frac{\lg(1 + S/N)}{\lg 2}$$

Beispiel: Bandbreite einer Telefonleitung: 3000Hz  
Störabstand: 30dB

→ Theoretische Kanalkapazität: 30kBit/s

H. Nyquist: Maximale Datenrate **D<sub>max</sub>** in einem störungsfreien Kanal:

$$D_{\max} = 2 \cdot C \cdot \log_2(n)$$

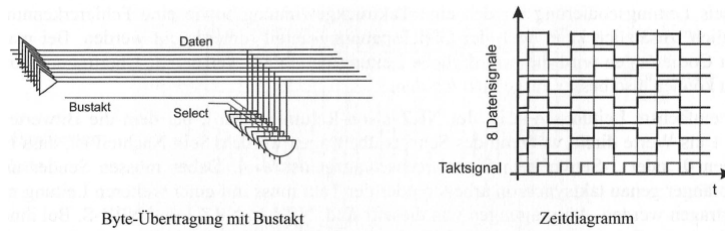
Beispiel: Telefonnetz mit C=3kHz mit 2 Zuständen

Theoretisch: D<sub>max</sub>=6000bit/s

Praktisch: Schon ab 1200bit/s werden Codierungen mit mehreren Zuständen verwendet, da nie ein störungsfreier Kanal vorliegt.

**Leitungscodierung**

1. **Parallele** Datenübertragung → großer Verdrahtungsaufwand
  - pro Schritt wird ein Byte übertragen (8bit = 8 Leitungen+Taktsignal)
  - bei bidirektionaler Kommunikation verdoppelt sich die Anzahl

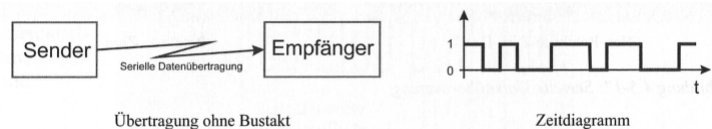


Byte-Übertragung mit Bustakt

Zeitdiagramm

*Parallele Datenübertragung*

2. **Serielle** Datenübertragung → eine Signalleitung zzgl. Taktsignal
  - Codierung des Signals ist erforderlich



Übertragung ohne Bustakt

Zeitdiagramm

*Serielle Datenübertragung*

- bei synchronisierenden Codes kann das Taktsignal aus dem Nutzsinal zurückgewonnen werden
- es müssen dafür genügend 0/1-Übergänge im Signal vorhanden sein

Der **Leitungscode** legt bei der digitalen Kommunikation fest, wie ein Signal auf der physikalischen Ebene übertragen wird. Bestimmte Pegelfolgen (z.B. Lichtintensitäten auf Glasfasern oder Spannungen in elektr. Leitungen) werden binären Bitsequenzen zugeordnet.

Einfachster Leitungscode: **NRZ** (Non-Return-to-Zero)

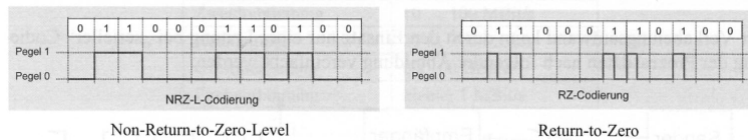
- Bitwerte 0 und 1 werden direkt während des Schrittes übertragen

**Nachteil:** Bei langen 0- oder 1-Serien wird kein Signalwechsel ausgelöst, Problem mit der Taktsynchronisation zwischen Sender und Empfänger.

**Weiterentwicklung: RZ-Codierung** (Return-to-Zero)

- bei langen 0-Serien wird nach n Wiederholungen eine 1 eingestreut  
→ „**Bit-Stuffing**“

**Nachteil:** Pegelwechsel bei der 1 erfordert doppelte Bandbreite  
Beide Codes sind **nicht** gleichstromfrei und ungeeignet für Taktrückgewinnung.



Serielle Datenübertragung

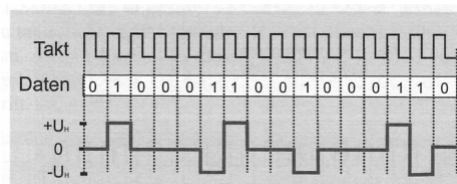
Der **AMI-Code** (**A**lternate **M**ark **I**nversion) ist ein ternärer Code (3 Zustände). Eine logische **0** wird als physikalische **0** übertragen, eine **1** abwechselnd durch  $+U_H$  und  $-U_H$ .

→ Gleichstromkomponente wird vermieden

**Coderegelverletzung**, wenn eine **1** durch denselben Zustand wie die vorherige **1** übertragen wird → wird zur Trennung verschiedener Rahmen angewandt

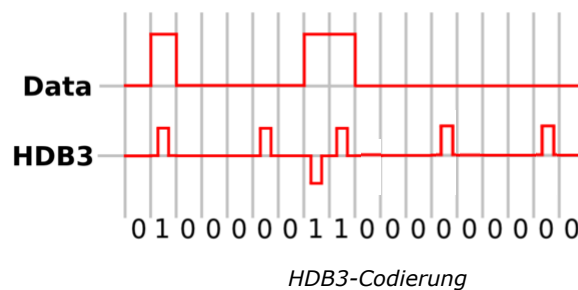
- bei langen 0-Folgen ist keine Taktrückgewinnung möglich

→ Deshalb werden 2 aufeinanderfolgende Nullen durch eine **0** und eine **umgekehrte 1** codiert.



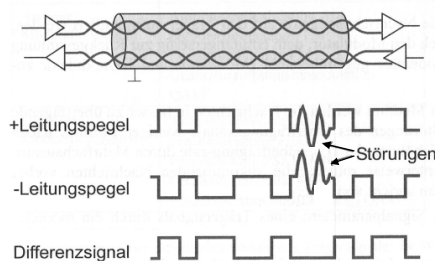
AMI-Codierung

Der **HDB-Code (High-Density-Bipolar Code)** der n-ten Ordnung) ist eine Weiterentwicklung des AMI-Codes. Bei der Übertragung von mehr als n aufeinanderfolgenden Nullen wird eine besondere Bit-Kombination eingefügt. Z.B. HDB3-Verfahren → n=3 → Nach 3 Nullen wird eine 1 in der „falschen Richtung“ hinzugefügt.



**Symmetrische Datenübertragung (differentielle Übertragung):**

- zwei Leiter pro Datenleitung
- Störungen auf der Masseleitung wirken sich nicht auf das Signal aus (wirken symmetrisch auf beide Leitungen)
- elektromagnetische Felder können eliminiert werden
- Beispiel: symm. XLR-Verbindung, USB-Schnittstelle



XLR-Stecker

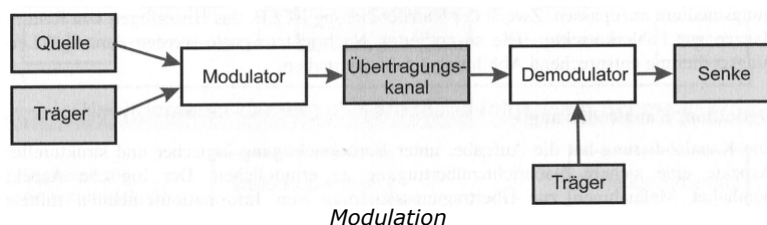
Symmetrische Übertragung mit Signaleigenschaften

## Basisbandübertragung:

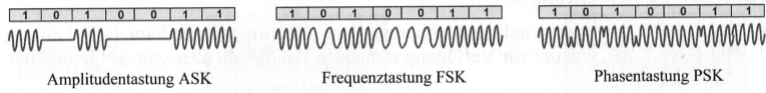
- einfachste Form der Datenübertragung
- Sender und Empfänger sind direkt galvanisch verbunden
- Leitungssignal beansprucht die gesamte Bandbreite
- mehrere Kanäle lassen sich nur zeitverschachtelt hintereinander übertragen (Zeitmultiplex - Time Division Multiplexing - TDM)

## Breitbandübertragung:

- mehrere unabhängige Trägersignale können gleichzeitig auf einem Kabel übertragen werden
- Signale werden auf unterschiedlichen Frequenzen übertragen (Frequenz-Multiplex-Verfahren - **F**requency **D**ivision **M**ultiplexing)
- Nachrichtensignale müssen auf den Träger aufmoduliert werden
- **Modulator** + **Demodulator** = Modem
- Beispiel: Kabelfernsehen



**Modulationsarten:**

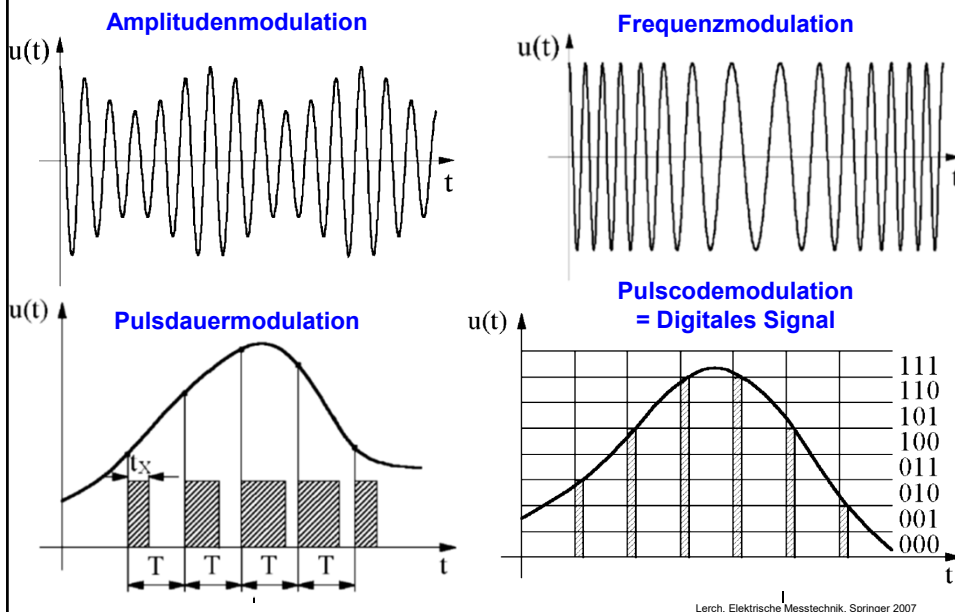


	Sinussignal	Pulssignal
Analoges Modulationssignal	Amplitudenmodulation AM Frequenzmodulation FM Phasenmodulation PM Quadraturamplitudenmodulation QAM <sup>1</sup>	Pulsamplitudenmodulation PAM Pulsfrequenzmodulation PFM Pulsphasenmodulation PPM Pulsdauermodulation PDM
Digitales Modulationssignal	Amplitudentastung ASK <sup>2</sup> Frequenzastung FSK <sup>1</sup> Phasentastung PSK <sup>1</sup> Quadraturphasentastung (Quadraturamplitudentastung) QAM <sup>3</sup>	Pulscode modulation PCM

<sup>1</sup> Überlagerung zweier amplitudenmodulierter Trägersignale, die 90° phasenverschoben sind.

<sup>2</sup> ASK steht für „amplitude shift keying“. Entsprechendes gilt für die weiteren Abkürzungen.

<sup>3</sup> Mischung aus Phasen- und Amplitudentastung bzw. -modulation mit evtl. mehreren Phasendifferenzen und Amplitudenwerten.



Die **Bitfehlerrate (BER – Bit Error Rate)** ist der Quotient aus der Anzahl fehlerhafter Bit zu der Anzahl insgesamt übertragener (oder gespeicherter) Bits.

z.B. BER von  $5 \cdot 10^{-6}$  bedeutet: Von 1 Million Bits sind 5 falsch

Einfachste Art der Fehlererkennung: **Paritätsprüfung**

Jeder Datenblock erhält ein Prüfbit:

→ Anzahl aller Einsen → gerade oder ungerade

→ 2 Bit-Fehler sind nicht erkennbar

Verbesserung durch fehlerkorrigierende Codes:

**Hamming Codes**

Der **Hamming-Abstand** zweier Codewörter ist die Anzahl an Stellen, an denen die Codewörter unterschiedliche Bits haben.

Beispiel: 10110011 und 11010010 → Hamming-Abstand=3



Richard Hamming

Eine **Schnittstelle** ist eine Vorrichtung zum Zweck des Informationsaustauschs mit anderen informationsverarbeitenden Systemen.

- Definition von normierten Übertragungsprotokollen

→ **Software-Protokoll**

Mechanische Eigenschaften:

- Steckerart, Steckerbelegung

Elektrische Eigenschaften:

- Signalpegel, Frequenzen

Funktionelle Eigenschaften:

- Leitungscode



Schnittstellen am PC



SCSI Schnittstelle



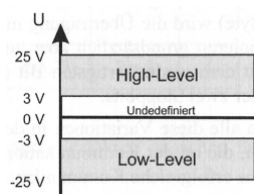
MIDI Schnittstelle

Beispiel: RS232-Schnittstelle (EIA-232)

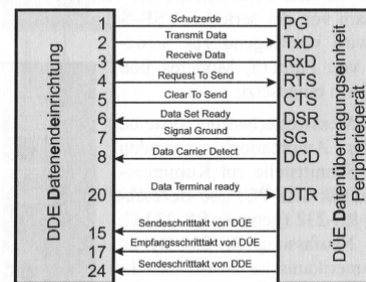
- 1969 als Standard eingeführt
- ehemals 25-polig
- später als 9-poliger Sub-D-Stecker
- logische **1** wird durch Spannungen zwischen -3V und -15V dargestellt
- logische **0** wird durch Spannungen zwischen +3V und +15V dargestellt



RS232-Schnittstelle



RS232-Signalpegel



V.24 Verbindungsschema

## Handshake-Verfahren:

- Sender signalisiert, wenn er neue Daten senden möchte
- Empfänger signalisiert, wenn er neue Daten verarbeiten kann

**Hardware-Handshake:** Signalübertragung über eine Leitung

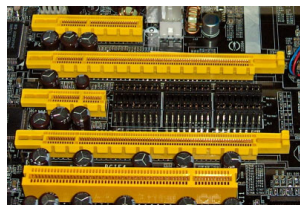
**Software-Handshake:** Signalübertragung über ein Zeichen

- XON-XOFF-Handshake, ETX-ACK-Handshake (End of Text, Acknowledge)

Ein **Bus** ist ein Übertragungssystem, bei dem mehrere Teilnehmer auf eine oder mehrere gemeinsame Leitungen zugreifen.

## Parallele Bussysteme

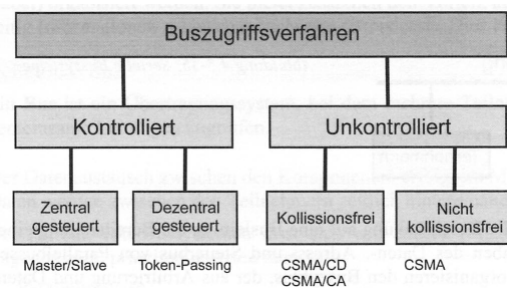
- Datenbus übermittelt die Daten
  - Adressbus wählt den Teilnehmer aus
  - Steuerbus organisiert Datenaustausch
- z.B. AT-Bus, PCI-Bus, AGP-Bus



PCI-Bus

## Serielle Bussysteme

- Kommunikationsteilnehmer sind über **eine** Busleitung verbunden
  - Busprotokoll organisiert die Datenübertragung
- z.B. CAN-Bus, Bit-Bus



## Master/Slave-Verfahren (deterministisch, zentral gesteuert)

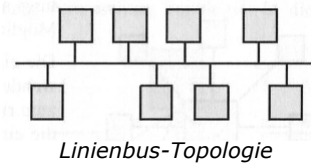
- Master stellt Verbindung zum ausgewählten Slave her
- Slave antwortet unmittelbar
- Master scannt alle Slave-Teilnehmer nacheinander durch (polling)
- Ausfall des Masters blockiert das ganze System
- Nachrichtenübertragung zw. den Teilnehmern ist mind. doppelt so hoch wie die Buszykluszeit

## Bustopologie:

- Verbindungsstruktur zwischen den Teilnehmern eines Kommunikationsnetzes

### 1. Linienbus-Topologie

- wird in der Automatisierung am häufigsten verwendet
- keine zentrale Netzwerkkomponente
- meist mit Abschlußwiderstand
- Nachrichten werden mit Adressen verschickt
  - nur die adressierte Station liest die Nachricht und schickt eine Bestätigung



## 2. Ring-Topologie

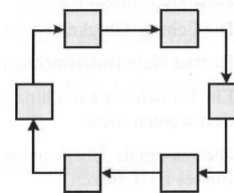
- geschlossene Kette
- keine aktive Netzwerkkomponente im Ring
- ein Protokoll regelt die Steuerung

### Vorteil:

- ausgedehnte Netzwerke sind realisierbar

### Nachteile:

- Betriebsunterbrechung bei Neueingliederung
- Ausfall eines Teilnehmers
  - Ausfall des Gesamtsystems



*Ring-Topologie*

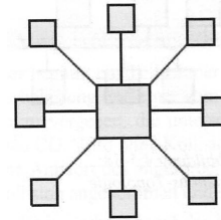
### 3. Stern-Topologie

- a) Zentralstation verschickt Nachrichtenpakete an den Adressaten und einzelne Stationen verständigen sich über ein Protokoll
- b) Zentralstation steuert den Zugriff

**Vorteil:** - einfach erweiterbar

**Nachteile:**

- hohe Anzahl an Schnittstellen
- keine Kommunikation bei Ausfall der Zentrale mehr möglich



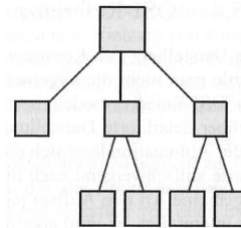
*Stern-Topologie*

### 4. Baum-Topologie

- erweiterte Stern-Topologie
- geeignet für größere Netzwerke

**Vorteile:**

- minimaler Verkabelungsaufwand
- leicht erweiterbar
- Kommunikation zwischen den Wurzeln ist durch Brücken möglich



*Baum-Topologie*

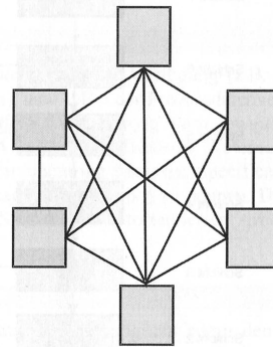
## 5. Vermaschte Topologie

- jedes Gerät wird mit jedem verbunden
- nur selten realisierbar
- Komplexität des Netzes steigt mit dem Quadrat der Anzahl der Teilnehmer

Beispiel: Internet

### Vorteile:

- ausfallsicherste Netzwerktopologie
- dezentrales Netzwerk
- redundante Wege ermöglichen Datentransport auch bei Unterbrechung eines Weges



Vermaschte Topologie

## Bussysteme und ihre Anwendungen

Bussysteme	Anwendungsgebiete	Hersteller
ASi	Aktor-Sensor-Bus auf unterster Ebene für die Fertigungsautomatisierung;	Siemens Peperl & Fuchs u.a.
INTERBUS-S	Fertigungsautomatisierung;	PHONIX-CONTACT
CAN	Fahrzeugtechnik, immer mehr in der Messtechnik;	BOSCH
BITBUS	Produktion und Fertigung;	Intel
PROFIBUS-DP	Fertigungsautomatisierung;	Siemens
PROFIBUS-PA	Prozessautomatisierung;	Siemens
EIB	Heim- und Gebäudeautomatisierung;	Siemens
LON	Gebäudetechnik, Energiemanagement;	Echolon
PROFIBUS-FMS	Fertigungsautomatisierung;	Siemens
Industrial Ethernet	Produktion und Büro;	Siemens