

# Digitalelektronik - Inhalt

---

- **Grundlagen**
  - Boolesche Algebra
- **Schaltungsentwurf**
  - Kombinatorik
  - Sequentielle Schaltungen
  - Zeitverhalten
- **Entwurfswerkzeuge**
  - HW-Beschreibungssprachen (VHDL)

# Grundlagen – Eigenschaften

---

	<b>Analog</b>	<b>Digital</b>
<b>Signale</b>	Beliebige Werteverläufe	Zwei diskrete Werte
<b>Einheiten</b>	Spannung (Strom), Zeit	wahr, falsch, Zeit
<b>Bauelemente</b>	Widerstände, Kondensatoren, Spulen, Transistoren (Dutzende / Hunderte)	Transistoren, Gatter (Tausende / Millionen)
<b>Entwurf</b>	„manuell“	„automatisch“

# Grundlagen – Verknüpfungen

---

- **Mathematische Grundlage ist die Aussagenlogik**
- **Aussagen können wahr (w) oder falsch (sein)**
- **Aussagen können miteinander verknüpft werden**
- **Resultat kann wieder nur „wahr“ oder „falsch“ sein**
- **Beispiel: UND-Verknüpfung**
  - A1: Schalter S1 ist geschlossen (wahr/falsch)
  - A2: Schalter S2 ist geschlossen (wahr/falsch)
  - A3: Die Lampe brennt (wahr/falsch)

A3 ist **genau dann** wahr, wenn A1 wahr **und** A2 wahr sind.

# Grundlagen - Verknüpfungen

---

- Eine Wahrheitstabelle stellt eine **Abbildung** von Eingangsvariablen auf Ausgangsvariablen **dar**
- Daher ist auch eine **Darstellung als Funktion** möglich:  $A_3 = f(A_1, A_2)$
- Die **Rechenregeln** für diese zweiwertige Funktionen werden als **Boolesche Algebra** bezeichnet

# Grundlagen – Verknüpfungen

---

- **Darstellung als Wahrheitstabelle (UND)**

<b>A1</b>	<b>A2</b>	<b>A3</b>
<b>0</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>1</b>

# Grundlagen – Boolesche Algebra

---

- Die **UND-Verknüpfung** wird auch als **Konjunktion** bezeichnet
- Die **UND-Verknüpfung** entspricht der **Multiplikation** in der „üblichen“ Algebra (siehe Wahrheitstabelle mit „0“ und „1“ statt „w“ und „f“)
- **Übliche UND-Verknüpfungszeichen sind :**  
der „Multiplikationspunkt“  $y = x_1 \bullet x_2$   
gar kein Zeichen  $y = x_1 x_2$   
ein „^“  $y = x_1 \wedge x_2$

# Grundlagen – Boolesche Algebra

---

- **Erweiterung der UND-Verknüpfung auf n Stellen**

- Das Ergebnis ist nur wahr, wenn alle Teilaussagen wahr sind

- Funktionsdarstellung:  $y = x_1 x_2 x_3 x_4$

- Wahrheitstabelle:

- Symbol „-“ (don't care) bedeutet entweder 0 oder 1

x1	x2	x3	x4	y
0	-	-	-	0
-	0	-	-	0
-	-	0	-	0
-	-	-	0	0
1	1	1	1	1

# Grundlagen – Boolesche Algebra

---

- **Einstellige Verknüpfungen**

– Identität:  $y=x$

x	y
0	0
1	1

– Negation:  $y=\bar{x}$

x	y
0	1
1	0

– Die Negation (Inversion) entspricht in etwa einem Vorzeichenwechsel



# Grundlagen – Boolesche Algebra

---

- Die **ODER-Verknüpfung** wird auch als **Disjunktion** bezeichnet
- Die **ODER-Verknüpfung** entspricht der **Addition** in der „üblichen“ Algebra.
- **Übliche ODER-Verknüpfungszeichen sind :**  
das Additionszeichen  $y=x_1+x_2$   
ein „v“  $y=x_1 \vee x_2$

# Grundlagen – Boolesche Algebra

---

## Absorptionen

$$x \cdot x = x$$

$$x \cdot \bar{x} = 0$$

$$x \cdot 1 = x$$

$$x \cdot 0 = 0$$

$$x + x = x$$

$$x + \bar{x} = 1$$

$$x + 1 = 1$$

$$x + 0 = x$$

## Doppelte Negation

$$\overline{\bar{x}} = x$$

## Distributionen

$$x_1 (x_2 + x_3) = x_1 x_2 + x_1 x_3$$

$$x_1 + (x_2 x_3) = (x_1 + x_2)(x_1 + x_3)$$

# Boolesche Algebra

---

- **Gesetz von Shannon zur Negation einer Funktion**
  - Ersetze alle bisherigen Konjunktionen durch Disjunktionen
  - Ersetze alle bisherigen Disjunktionen durch Konjunktionen
  - Negiere alle Variablen

$$\overline{f}(x_1, \dots, x_n, \bullet, +) = f(\overline{x}_1, \dots, \overline{x}_n, +, \bullet)$$

# Wahrheitstabelle - Funktion

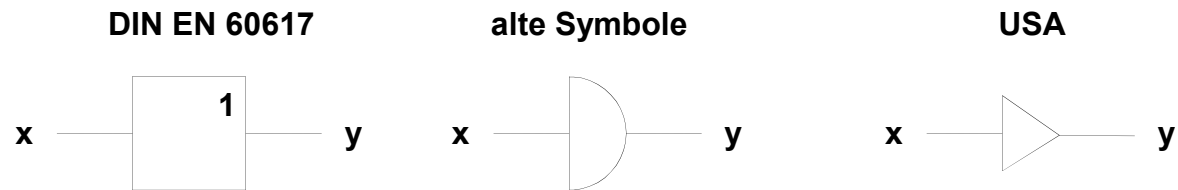
---

- **Die Darstellung einer Verknüpfung in einer Wahrheitstabelle kann leicht in eine Darstellung als Boolesche Funktion umgewandelt werden**
  - Jede Zeile, die eine ‚1‘ ergibt, entspricht einer Konjunktion (UND) aller Variablen
    - ‚0‘ wird durch die negierte Variable der Spalte ersetzt
    - ‚1‘ wird durch die ‚normale‘ Variable der Spalte ersetzt
  - Alle damit erhaltenen Konjunktionen werden danach durch eine Disjunktion (ODER) zusammengefasst

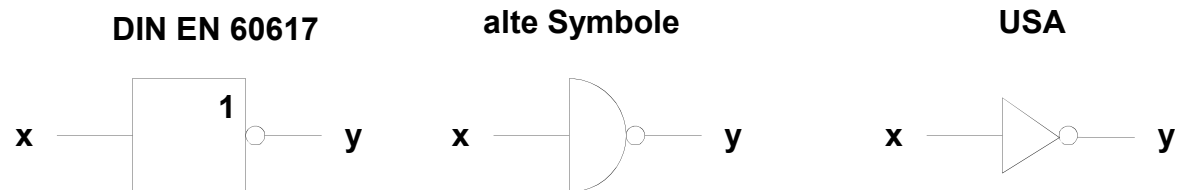
# Grundgatter

---

- **Buffer (realisiert die Identität)**



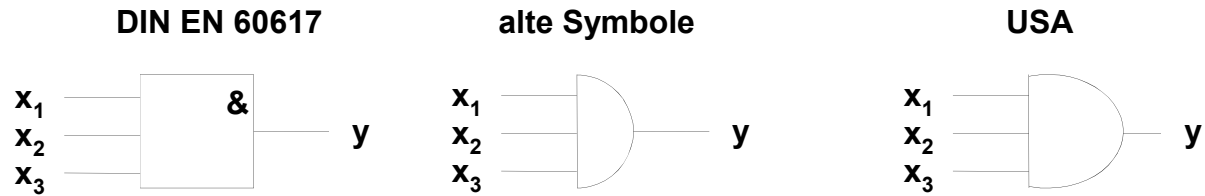
- **Inverter (realisiert die Negation)**



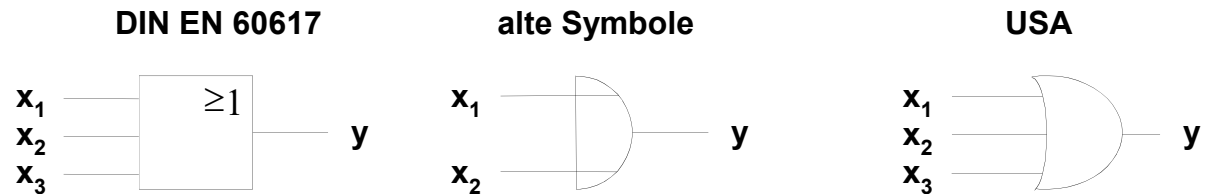
# Grundgatter

---

- **UND (AND)**



- **ODER (OR)**

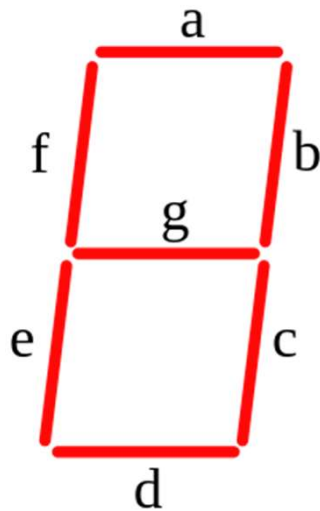


# Siebensegmentdekode (Tabelle)

Eingänge

$$Z = \sum_{i=0}^3 x_i \times 2^i$$

Ausgänge



Eingänge				Ausgänge						
$x_3$	$x_2$	$x_1$	$x_0$	a	b	c	d	e	f	g
0	0	0	0					1		
0	0	0	1					0		
0	0	1	0					1		
0	0	1	1							
0	1	0	0							
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

# Siebensegmentdecoder (Funktionen)

---

## Ableitung der Einzelfunktionen aus der Tabelle

$$a = \dots$$

$$b = \dots$$

$$c = \dots$$

$$d = \dots$$

$$e = \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 + \bar{x}_3 x_2 x_1 \bar{x}_0 + x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$$

$$f = \dots$$

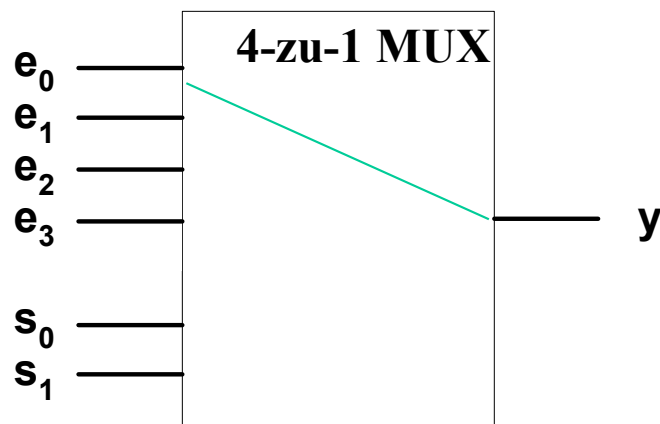
$$g = \dots$$



# Multiplexer

---

- Aufgabe: Durchschalten eines von n Eingängen auf einen einzigen Ausgang**

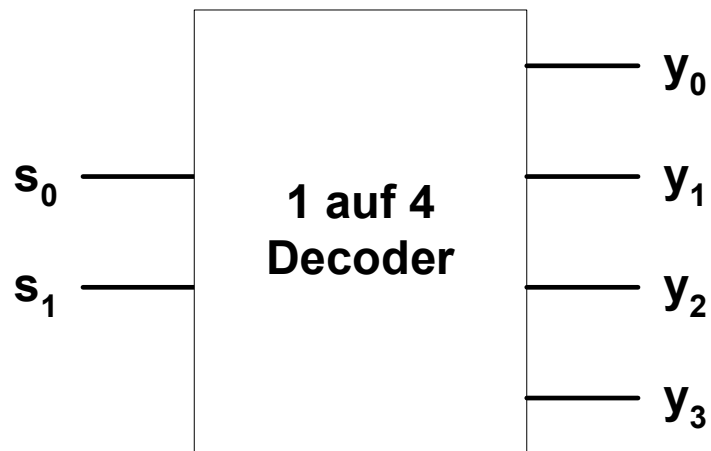


s1	s0	e3	e2	e1	e0	Y
0	0	-	-	-	1	1
0	1	-	-	1	-	1
1	0	-	1	-	-	1
1	1	1	-	-	-	1

# 1-aus-n Dekoder

---

- **Aufgabe: Aktivieren genau eines Ausgangs von n Ausgängen**



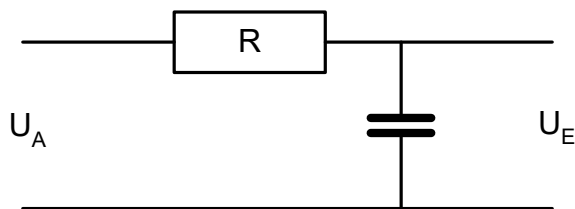
$s_1$	$s_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

# Dynamisches Verhalten

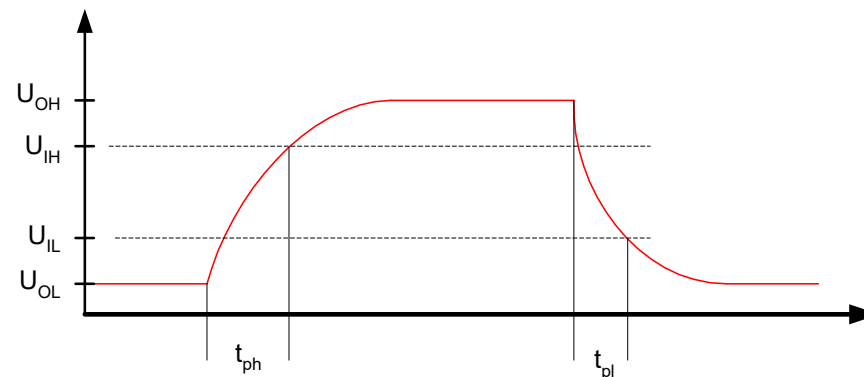
---

- **Reale Schaltung schaltet mit Verzögerung**
  - Gatterlaufzeiten innerhalb der Logikgatter
  - Leitungslaufzeiten zwischen den Gattern

## Modell einer Leitung



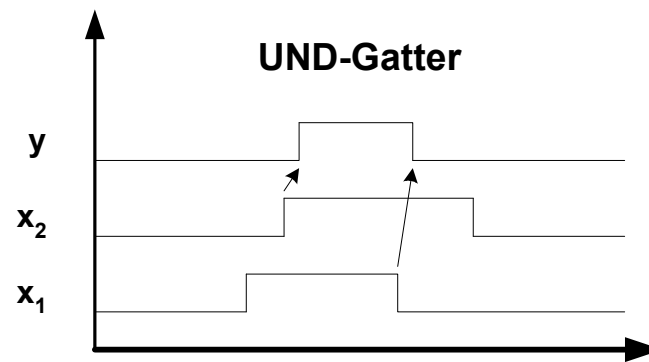
## Spannungsverlauf



# Dynamisches Verhalten

---

- **Signaländerungen pflanzen sich von den Eingängen zu den Ausgängen fort**



# Dynamisches Verhalten

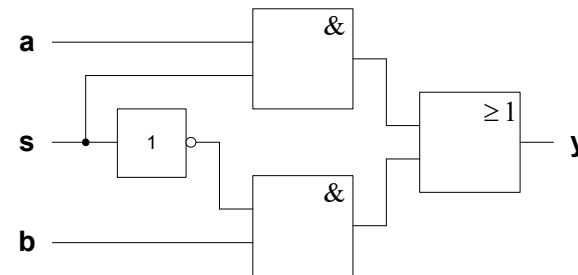
---

- In einer mehrstufigen Schaltung laufen die Signaländerungen auf allen Pfaden parallel „um die Wette“

Wertetabelle Mux

s	a	b	y
0	1	-	1
1	-	1	1

Reale Schaltung



Annahme:  $a=1$ ,  $b=1$ ,  $s$  wechselt von 1 auf 0

$y$  bleibt immer 1

$y$  geht von 1 auf 0 und dann wieder nach 1 !

# Dynamisches Verhalten

---

- **Bei unterschiedlich langen Pfaden zum selben Gatter trifft ein Signal später ein (**race condition**)**
  - kurzfristig können Pegel abweichend vom Endwert am Ausgang des Gatters entstehen (**Spikes**)
  - Spikeentstehung hängt ab von
    - Belegung der anderen Signaleingänge
    - Art der Signaländerung
    - Topologie der Schaltung (Struktur)
    - Physikalischen Bedingungen (Technologie, Verdrahtung etc.)
- **Stabiler Endwert spätestens nach Verzögerungszeit des längsten Pfades**