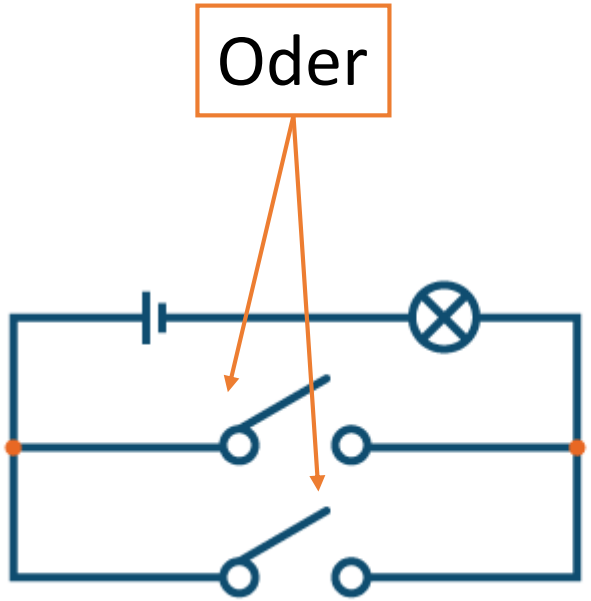
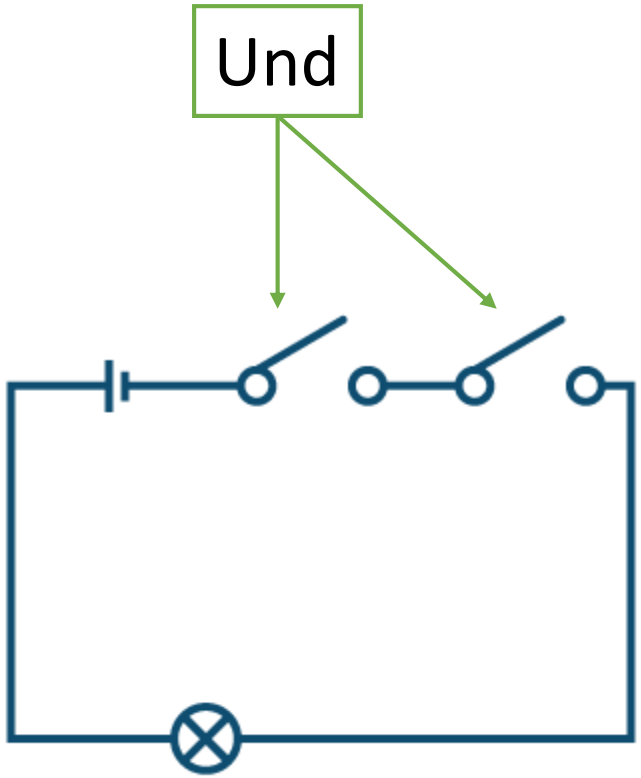
Boolesche Algebra

u.a. zur Umsetzung von Addition mit logischen Gattern

Wann leuchtet das Licht?





Um diese und auch andere „Rechnungen“ ausdrücken zu können, benutzen wir die Boolesche Algebra

Mathematische Definition: Boolesche Algebra

- Sei $B = \{0,1\}$ ein Alphabet für eine Repräsentation
- Seinen auf B **Operatoren** einer Algebra wie folgend definiert:

Für $x, y \in B$:

1.	$x \wedge y := Min(x, y)$	Und-Verknüpfung
2.	$x \vee y := Max(x, y)$	Oder-Verknüpfung
3.	$\neg x :=$Komplement von x	Negation

- $(B,\ \wedge,\ \vee,\ \neg)$ heißt **Boolesche Algebra**

Funktionstabellen

Boolsche Funktionen werden oft durch Funktionstabellen definiert:

Für jede Input-Kombination wird der Output explizit angegeben

a	b	$a \wedge b$	$a \vee b$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

a	¬a
0	1
1	0

Funktionstabelle

Schaltfunktionen

Seien $n, m \in \mathbb{N}$ mit $n, m \geq 1$

Dann heißt die Funktion $F: B^n \rightarrow B^m$ *(boolesche) Schaltfunktion in* n *Variablen* mit m *Ausgängen*.

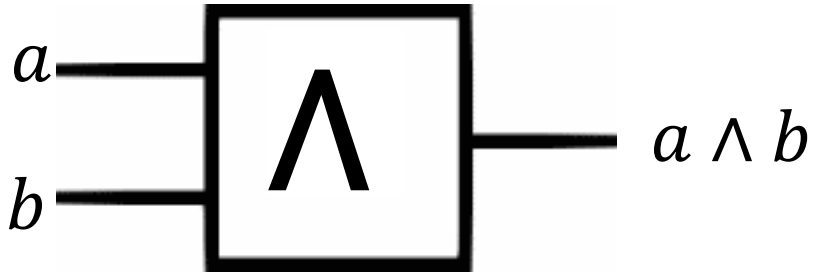
Ist m$= 1$, heißt $F: B^n \rightarrow B$ (n-stellige) boolesche Funktion

Bsp.: $\wedge, \vee, sind\ zweistellige\ Funktionen;\ \neg\ eine\ einstellige\ Funktion$

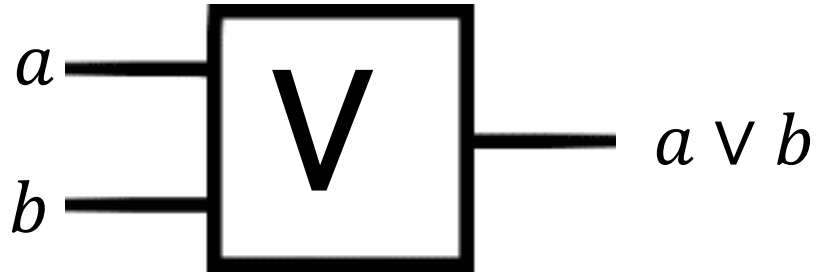
Schaltfunktionen können durch Gatter realisiert werden. Diese haben jeweils eine Menge an Eingängen und eine Menge an Ausgängen.

Gatter-Notation:

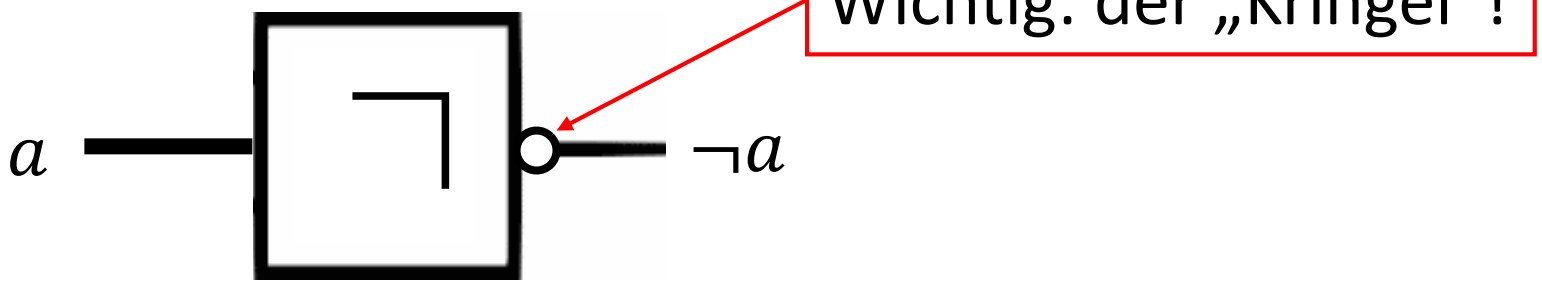
- Und-Gatter



- Oder-Gatter



- Negation-Gatter



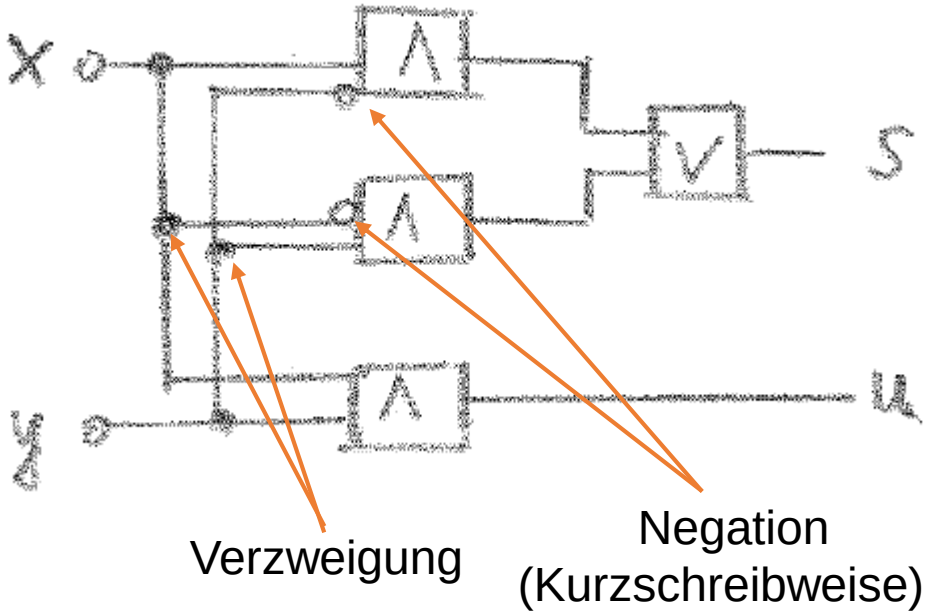
Halbaddierer: Addition zweier Bits (x, y) mit Übertrag

- Ein Halbaddierer lässt sich durch die folgenden Formeln beschreiben:

$$s = (x \wedge \neg y) \vee (\neg x \wedge y)$$

$$u = (x \wedge y)$$

x	Y	Summe (s)	Übertrag (u)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Gatter-Notation zu umständlich!

Weitere Schaltfunktionen

- Aufgrund unserer Definition der Booleschen Algebra aus 3 Operatoren reichen uns prinzipiell die 3 Gatter für Und, Oder und Nicht, um alle Funktionen darzustellen
- Es gibt aber weitere Funktionen (mit entsprechenden Gattern), um oft benötigte Funktionen darzustellen:

  - XOR: $a \oplus b = (a \wedge \neg b) \vee (\neg a \wedge b)$
  - NOR: $a \overline{\vee} \mathrm{b} = \neg(a \vee b) = \neg a \wedge \neg b$
  - NAND: $a \overline{\wedge} b = \neg(a \wedge b) = \neg a \vee \neg b$
  - (Implikation): $a \Rightarrow b = \neg a \vee b$
  - (Äquivalenz): $a \Leftrightarrow b = (a \wedge b) \vee (\neg a \wedge \neg b)$

Halbaddierer mit XOR-Schaltfunktion

$$s = (x \oplus y)$$

$$u = (x \wedge y)$$

XOR-Gatter lässt sich sehr leicht in Hardware realisieren!

Volladdierer: mit Übertrag als Input c_{in} und Output c_{out}

3 Inputs: x, y, c_{in}

2 Outputs: s, c_{out}

$$s = (x \oplus y \oplus c_{in})$$

$$c_{out} = \big(c_{in} \wedge (x \oplus y)\big) \vee (x \wedge y)$$

x	y	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$c_{out} = (\neg x \wedge y \wedge c_{in}) \vee (x \wedge \neg y \wedge c_{in}) \vee (x \wedge y \wedge \neg c_{in}) \vee (x \wedge y \wedge c_{in})$$

$$s = (\neg x \wedge \neg y \wedge c_{in}) \vee (\neg x \wedge y \wedge \neg c_{in}) \vee (x \wedge \neg y \wedge \neg c_{in}) \vee (x \wedge y \wedge c_{in})$$

Addition mehrstelliger Zahlen durch Verbindung von c_{out} mit c_{in} der nächsten Stelle

Gesetze der Booleschen Algebra

Name	Form für Oder	Form für Und
Kommutativgesetz	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
Assoziativgesetz	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Distributivgesetz	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
Komplementgesetz	$x \vee \neg x = 1$	$x \wedge \neg x = 0$
Idempotenzgesetz	$x \vee x = x$	$x \wedge x = x$
Gesetz vom kleinsten und größten Element	$x \vee 0 = x$ $x \vee 1 = 1$	$x \wedge 0 = 0$ $x \wedge 1 = x$
De` Morgansche Regeln	$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$

Klammerung und Operatorreihenfolge (links nach rechts) gilt wie in der „normalen“ Algebra!

Aufgabe: Weitere Schaltfunktionen

a	B	$a \oplus b$	$a \barwedge b$	$a \barvee b$	$a \Rightarrow b$	$a \Leftrightarrow b$
0	0					
0	1					
1	0					
1	1					

Lösung: Weitere Schaltfunktionen

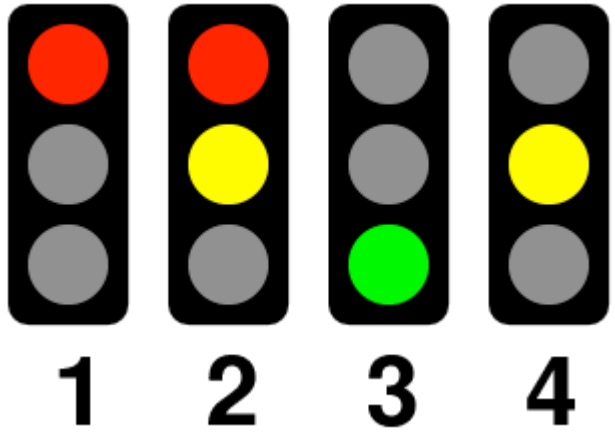
a	B	$a \oplus b$	$a \barwedge b$	$a \barvee b$	$a \Rightarrow b$	$a \Leftrightarrow b$
0	0	0	1	1	1	1
0	1	1	0	1	1	0
1	0	1	0	1	0	0
1	1	0	0	0	1	1

Beispiel: Ampelschaltung

- Gesucht ist eine Funktionstabelle, die die Schaltung einer Ampel „validiert“
- Die Funktion hat 3 Eingänge: r (rot), y (gelb), g (grün) und einen Ausgang z („zulässig“)
- Ausgang z soll genau dann den Wert „1“ („wahr“) liefern, wenn die Ampel eine zulässige Kombination von Farben anzeigt, ansonsten eine „0“ („falsch“)

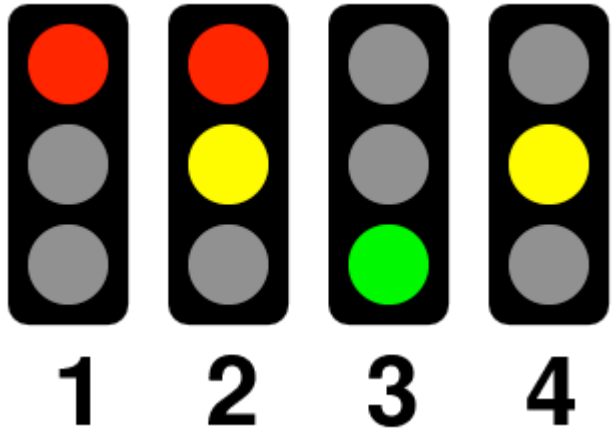
Aufgabe: Ampelschaltung

r	y	g	z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Lösung: Ampelschaltung

r	y	g	z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



Funktionstabelle ➔ Funktion

- Nun, da wir eine Funktionstabelle haben, wollen wir auch eine Funktion erstellen. Kann uns dabei die Funktionstabelle helfen?

- Antwort: Ja, mit der „Disjunktiven Normalform“

- Algorithmus:
  1. Nimm alle „Wahr“-Zeilen der Funktionstabelle
  2. Tausche die Eingangswerte mit der (negierten) Variable, falls Eingang 1 (0)
  3. „Ver-und-e“ die Variablen als Und-Terme
  4. „Ver-oder-e“ die Und-Terme (Klammerung beachten!)

Beispiel: Disjunktive Normalform

r	y	g	z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$(\neg r \wedge \neg y \wedge g)$$
$$\vee$$
$$(\neg r \wedge y \wedge \neg g)$$
$$\vee$$
$$(r \wedge \neg y \wedge \neg g)$$
$$\vee$$
$$(r \wedge y \wedge \neg g)$$

Anmerkung: Schaltfunktion

- Die Disjunktive Normalform ist nicht die einzige korrekte Form einer Funktion (aber am einfachsten zu erstellen)
- Es gibt meist mehrere gleichwertige Formen einer Funktion

Beispiel Ampelschaltung:

- DNF:

$$z = (\neg r \wedge \neg y \wedge g) \vee (\neg r \wedge y \wedge \neg g) \vee (r \wedge \neg y \wedge \neg g) \wedge (r \wedge y \wedge \neg g)$$

- KNF („Konjunktive Normalform“; ähnlicher Algorithmus wie bei DNF):

$$z = (r \vee y \vee g) \wedge (r \vee \neg y \vee \neg g) \wedge (\neg r \vee \neg y \vee g) \wedge (\neg r \vee \neg y \vee \neg g)$$

- Andere Form (Bestimmung durch „Nachdenken“):

$$z = g \oplus (r \vee y)$$

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

Berechnung von „Zwischenergebnissen“

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0		
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0		
0	0	1	0		
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0		
0	0	1	0		
0	1	0	1		
0	1	1	1		
1	0	0	1		
1	0	1	1		
1	1	0	0		
1	1	1	0		

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0	0	
0	0	1	0	1	
0	1	0	1	0	
0	1	1	1	1	
1	0	0	1	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	0	1	

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0	0	0
0	0	1	0	1	
0	1	0	1	0	
0	1	1	1	1	
1	0	0	1	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	0	1	

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	
0	1	1	1	1	
1	0	0	1	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	0	1	

Funktion ➔ Funktionstabelle (am Beispiel)

- Beispiel: $z = (a \oplus b) \wedge (a \vee c)$

a	b	c	$a \oplus b$	$a \vee c$	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	1	0	1	0

Aufgabe: Gilt: $z = g \oplus (r \vee y)$?

r	y	g	$r \vee y$	$g \oplus (r \vee y)$	z
0	0	0			0
0	0	1			1
0	1	0			1
0	1	1			0
1	0	0			1
1	0	1			0
1	1	0			1
1	1	1			0

Lösung: $z = g \oplus (r \vee y)$

r	y	g	$r \vee y$	$g \oplus (r \vee y)$	z
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

Beweise die De` Morganschen Regeln:

$$\neg(x \vee y) = \neg x \wedge \neg y$$
$$\neg(x \wedge y) = \neg x \vee \neg y$$

x	y
0	0
1	0
0	1
1	1