

Logische Schaltungen

Bit = **binary digit** = Binärziffer, also 0 oder 1

Byte = Folge von 8 Bit = 8-stellige Binärzahl

z.B. 01001101

Wie viele Zahlen kann man mit einem Byte darstellen? $256 = 2^8 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2$
binär von 00000000 bis 11111111.

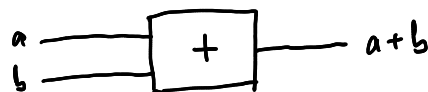
Ziel: Entwurf einer Schaltung, die zwei Byte addieren kann.

Logische Verknüpfungen

Eine Verknüpfung auf einer Menge X nimmt zwei Elemente von X (zweistellige) und liefert ein Element von X .

Bsp: (1) Addition, also $+$, ist eine Verknüpfung auf \mathbb{Z} :

Eingabe: $a, b \in \mathbb{Z}$ Ausgabe: $a+b \in \mathbb{Z}$.



(2) Subtraktion:



(3) Multiplikation: $a \cdot b$

Boolesche / logische Verknüpfungen: z.B. AND Konjunktion
OR Disjunktion

Beschreibungen von AND: (i)

falsch	AND	falsch	=	falsch
falsch	AND	wahr	=	falsch
wahr	AND	falsch	=	falsch
wahr	AND	wahr	=	wahr

(ii)

AND	falsch	wahr
falsch	falsch	falsch
wahr	falsch	wahr

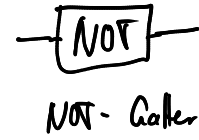
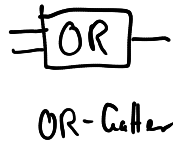
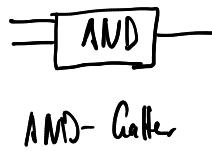
Es ist egal, ob wir falsch/wahr oder f/w oder False/True oder AUS/AN oder 0 / 1 schreiben

a	b	a AND b	a OR b
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Wahrheitstafel

a	NOT a	Vermutung / Negation
0	1	
1	0	

Youtube-Video:
Making logic gates from transistors



a AND b	mathematische Schreibweise:	a \wedge b	(\wedge stilisiertes A)
a OR b	— " — :	a \vee b	(\vee lateinisch vel: oder)
NOT a	— — :	\neg a	

Aufgabe: $\neg(a \wedge b)$ $(\neg a) \wedge (\neg b)$
 $\neg(a \vee b)$ $(\neg a) \vee (\neg b)$

a	b	$\neg(a \wedge b)$	$\neg(a \vee b)$	$(\neg a) \wedge (\neg b)$	$(\neg a) \vee (\neg b)$	$\neg a$	$\neg b$
0	0	1	1	1	1	1	1
0	1	1	0	0	1	1	0
1	0	1	0	0	1	0	1
1	1	0	0	0	0	0	0

Spalten sind gleich

Also: Es gelten die

De Morganschen Gesetze: $\neg(a \wedge b) = (\neg a) \vee (\neg b)$

$$\neg(a \vee b) = (\neg a) \wedge (\neg b)$$