Technische Informatik
Boolesche Algebra

Thorsten Thormählen

02. November 2023

Teil 3, Kapitel 1

Dies ist die Druck-Ansicht.

Aktiviere Präsentationsansicht

Steuerungstasten

→ nächste Folie (auch Enter oder Spacebar).
← vorherige Folie
d schaltet das Zeichnen auf Folien ein/aus
p wechselt zwischen Druck- und Präsentationsansicht
CTRL + vergrößert die Folien
CTRL - verkleinert die Folien
CTRL 0 setzt die Größenänderung zurück

Notation

Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	a, b, x, y
Funktionen	aufrecht	$\mathrm{f}, \mathrm{g}(x), \max(x)$
Vektoren	fett, Elemente zeilenweise	$\mathbf{a}, \mathbf{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^\top$, $\mathbf{B} = (x, y, z)^\top$
Matrizen	Schreibmaschine	$\mathtt{A}, \mathtt{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, B = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}^2, \mathbb{R}^3$

Inhalt

Schaltkreise und Wahrheitstafeln

Boolesche Algebra

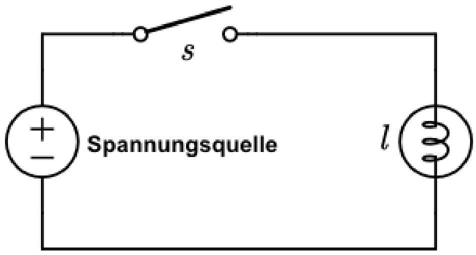
Schaltalgebra

Boolesche Funktionen

Boolesche Ausdrücke

Schaltkreise und Wahrheitstafeln

Betrachten wir einen einfachen Stromkreis bestehend aus einer Spannungsquelle (z.B. einer Batterie), einem Schalter s, und einer Lampe l



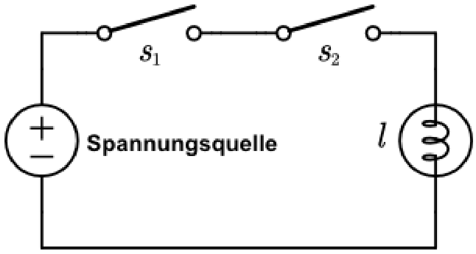
Ist der Schalter offen (s=0), brennt die Lampe nicht (l=0). Ist der Schalter geschlossen (s=1), brennt die Lampe (l=1).

Das Verhalten des Stromkreises kann in einer Wahrheitstafel dargestellt werden

s	l
0	0
1	1

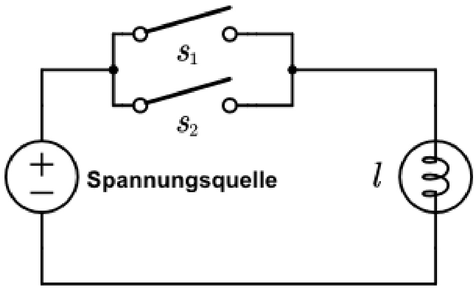
Schaltkreise und Wahrheitstafeln

Durch eine Reihenschaltung kann eine logische UND-Verknüpfung realisiert werden (Licht brennt, wenn s_1=1 UND s_2=1)



s_1	s_2	l
0	0	0
0	1	0
1	0	0
1	1	1

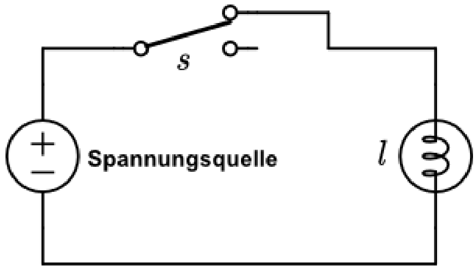
Durch eine Parallelschaltung kann eine logische ODER-Verknüpfung realisiert werden (Licht brennt, wenn s_1=1 ODER s_2=1)



s_1	s_2	l
0	0	0
0	1	1
1	0	1
1	1	1

Schaltkreise und Wahrheitstafeln

Durch umgekehrten Anschluss der Leitung kann eine logisches NICHT realisiert werden (Licht brennt, wenn s=0)



s	l
0	1
1	0

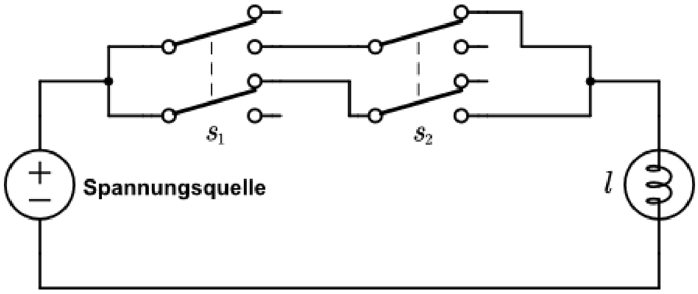
Damit haben wir schon die Grundelemente UND, ODER, NICHT (Englisch: AND, OR, NOT) kennengelernt, mit denen sich im Prinzip jeder mögliche Schaltkreis aufbauen lässt

Schaltkreise und Wahrheitstafeln

Wie könnte ein Schaltkreis aussehen, der folgende Wahrheitstabelle realisiert?

s_1	s_2	l
0	0	0
0	1	1
1	0	1
1	1	0

Lösung: (s_1=0 AND s_2=1) OR (s_1=1 AND s_2=0)



Wir haben haben damit übrigens gerade das "EXKLUSIVE ODER" (Englisch: XOR) implementiert

Schaltkreise und Wahrheitstafeln

Wie viele mögliche Funktionen zweier Eingangsvariablen gibt es eigentlich?

Antwort:
Es gibt 16 mögliche Funktionen zweier Eingangsvariablen:

s_1	s_2	16 mögliche Funktionen l_0 bis l_{15}															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	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

Im Allgemeinen gibt es bei n Eingangsvariablen $2^{(2^n)}$ mögliche Funktionen

Boolesche Algebra

Definition mittels Huntington'scher Axiome:

Sei $\mathcal{V}$ eine nicht leere Menge von Elementen mit den binären Operatoren $\circ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ und $\bullet : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$.

Das Tripel $(\circ, \bullet, \mathcal{V})$ heißt *Boolesche Algebra*, wenn für beliebige $a, b, c \in \mathcal{V}$ folgende Axiome gelten:

	Operator $\bullet$	Operator $\circ$
kommutativ	$a \bullet b = b \bullet a$	$a \circ b = b \circ a$
distributiv	$a \bullet (b \circ c) = (a \bullet b) \circ (a \bullet c)$	$a \circ (b \bullet c) = (a \circ b) \bullet (a \circ c)$
Identität	$a \bullet 1 = a$	$a \circ 0 = a$
Inversion	$a \bullet a^{-1} = 0$	$a \circ a^{-1} = 1$

Dabei müssen die Elemente 0 und 1 jeweils ein bestimmtes ausgezeichnetes Element aus $\mathcal{V}$ sein

Boolesche Algebra

Ist der Körper der rellen Zahlen mit der Multiplikation und Addition $(\cdot, +, \mathbb{R})$ eine boolesche Algebra?

Antwort: Nein, einige Axiome gelten nicht

	Operator $\cdot$	Operator $+$
kommutativ	$a \cdot b = b \cdot a$	$a + b = b + a$
distributiv	$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	$a + (b \cdot c) = (a + b) \cdot (a + c)$
Identität	$a \cdot 1 = a$	$a + 0 = a$
Inversion	$a \cdot a^{-1} = 0$	$a + a^{-1} = 1$

Schaltalgebra

Die Schaltalgebra ist eine boolesche Algebra über dem Tripel $(\wedge, \vee, \mathcal{B})$ mit

- der Grundmenge $\mathcal{B} = \{0, 1\}$
- der AND-Verknüpfung (Konjunktionsoperator): "$\wedge$"
- der OR-Verknüpfung (Disjunktionsoperator): "$\vee$"
- und dem inversen Element: der NOT-Operation (Negation): "$\neg$"

Es gilt somit laut Definition:

	Operator $\wedge$	Operator $\vee$
kommutativ	$a \wedge b = b \wedge a$	$a \vee b = b \vee a$
distributiv	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Identität	$a \wedge 1 = a$	$a \vee 0 = a$
Inversion	$a \wedge \neg a = 0$	$a \vee \neg a = 1$

Schaltalgebra

Die Schaltalgebra besteht somit aus folgenden elementaren Grundoperationen

Konjunktion (AND): $y = a \wedge b$

a	b	$y = a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Disjunktion (OR): $y = a \vee b$

a	b	$y = a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Negation (NOT): $y = \neg a$

a	$y = \neg a$
0	1
1	0

Boolesche Funktionen (Schaltfunktion)

- Eine Funktion $y = \mathrm{f}(x_1, x_2, \ldots, x_n)$ mit $x_1, x_2, \ldots, x_n \in \{0, 1\}$ heisst boolesche Funktion der Stelligkeit n
- Boolesche Funktionen werden auch Schaltfunktionen genannt
- Die Eingabevariablen $x_1, x_2, \ldots, x_n$ werden als *freie* Variablen bezeichnet
- Die Ausgabevariable y wird *abhängige* Variable genannt
- Beispiele:
  - Der Negationsoperator $y = \neg x_1$ ist eine boolesche Funktion der Stelligkeit 1
  - Konjunktionsoperator $y = x_1 \wedge x_2$ und Disjunktionsoperator $y = x_1 \vee x_2$ sind boolesche Funktionen der Stelligkeit 2

[**Quelle:** D. W. Hoffmann: *Grundlagen der Technischen Informatik*, 2. Auflage; Hanser 2009]

Boolesche Funktionen (Schaltfunktion)

Eine boolesche Funktion kann eindeutig durch eine vollständige Wahrheitstafel beschrieben werden

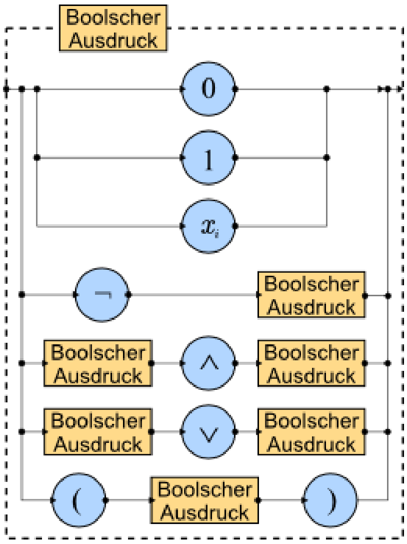
Beispiele:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

x_1	x_2	x_3	y
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	0
1	1	1	1

Boolesche Ausdrücke

- Die Darstellung durch Wahrheitstafeln wird schnell sehr groß und unübersichtlich
- Daher werden boolesche Funktionen häufig in Formeldarstellung angegeben
- Diese Formeldarstellungen werden *Boolesche Ausdrücke* genannt und sind durch das gezeigte Syntaxdiagramm definiert



Syntaxdiagramm

[**Quelle:** D. W. Hoffmann: *Grundlagen der Technischen Informatik*, 2. Auflage; Hanser 2009, Abbildung 4.5]

Boolesche Ausdrücke

Die Darstellung einer booleschen Funktion mittels boolescher Ausdrücke ist nicht eindeutig

Beispiel: Die Funktion

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

kann durch viele verschiedene boolesche Ausdrücke beschrieben werden:

$$y = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$$
$$y = (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)$$
$$y = \dots$$

Durch die Theoreme auf der nächsten Folie können verschiedene boolesche Ausdrücke in einander überführt werden

Liste der Axiome und Theoreme

Aus den Axiomen der booleschen Algebra lassen sich weitere Theoreme ableiten

#	Name	AND-Operator $\wedge$	OR-Operator $\vee$
1	Identität	$a \wedge 1 = a$	$a \vee 0 = a$
2	Elimination	$a \wedge 0 = 0$	$a \vee 1 = 1$
3	Idempotenz	$a \wedge a = a$	$a \vee a = a$
4	Involution	$\neg(\neg a) = a$	
5	Komplement	$a \wedge \neg a = 0$	$a \vee \neg a = 1$
6	Kommutativität	$a \wedge b = b \wedge a$	$a \vee b = b \vee a$
7	Assoziativität	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$	$(a \vee b) \vee c = a \vee (b \vee c)$
8	Distributivität	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
9	Vereinigung	$(a \vee b) \wedge (a \vee \neg b) = a$	$(a \wedge b) \vee (a \wedge \neg b) = a$
10	Absorption	$a \wedge (a \vee b) = a$	$a \vee (a \wedge b) = a$
11	Absorption (2)	$(a \wedge \neg b) \vee b = a \vee b$	$(a \vee \neg b) \wedge b = a \wedge b$
12	Faktorisierung	$(a \vee b) \wedge (\neg a \vee c) = (a \wedge c) \vee (\neg a \wedge b)$	$(a \wedge b) \vee (\neg a \wedge c) = (a \vee c) \wedge (\neg a \vee b)$
13	Konsens	$(a \vee b) \wedge (b \vee c) \wedge (\neg a \vee c) = (a \vee b) \wedge (\neg a \vee c)$	$(a \wedge b) \vee (b \wedge c) \vee (\neg a \wedge c) = (a \wedge b) \vee (\neg a \wedge c)$
14	de Morgans Gesetz	$\neg(a \wedge b \wedge \ldots) = \neg a \vee \neg b \vee \ldots$	$\neg(a \vee b \vee \ldots) = \neg a \wedge \neg b \wedge \ldots$

Dualität

Von den aufgelisteten Theoremen muss sich nur eine Spalte der Tabelle gemerkt werden. Die andere Spalte ergibt sich direkt durch die *Dualität*

Der duale boolesche Ausdruck f_d kann aus f ermittelt werden, indem $\vee$ mit $\wedge$ und 0 mit 1 vertauscht wird

$$\mathrm{f}(x_1, x_2, \ldots, x_n, 0, 1, \wedge, \vee) \leftrightarrows \mathrm{f}_d = f(x_1, x_2, \ldots, x_n, 1, 0, \vee, \wedge)$$

De Morgansche Gesetze

Die Gesetze von August De Morgan (*1806; †1871) lauten

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

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

Die Gültigkeit der Gesetze kann leicht anhand von Wahrheitstafeln gezeigt werden

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

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

Augustus De Morgan

Negationstheorem

Theorem #14 aus unserer Liste ist eine Verallgemeinerung der De Morganschen Gesetze:

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

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

Noch allgemeiner formuliert es das Negationstheorem:

$$\neg(\mathrm{f}(x_1, x_2, \ldots, x_n, 0, 1, \wedge, \vee)) = \mathrm{f}(\neg x_1, \neg x_2, \ldots, \neg x_n, 1, 0, \vee, \wedge)$$

Beispiel:

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

Weitere Operatoren

- Neben den elementaren logischen Operationen (AND, OR, NOT) gibt es noch weitere wichtige Operatoren
  - Implikation und Inverse Implikation
  - Äquivalenz und Antivalenz
  - Sheffer-Funktion (NAND) und Peirce-Funktion (NOR)
- Diese Operationen könnten jedoch alle auch mit den elementaren Operationen realisiert werden
- Sie verkürzen lediglich die Darstellung

Implikation und Inverse Implikation

Implikation: $y = a \rightarrow b = \neg a \vee b$

Gültigkeit der Aussage: "a ist eine hinreichende Bedingung für b".

Beispiel: "Dass die Sonne scheint, resultiert zwangsläufig (hinreichende Bedingung) in Sonnenbrand"

Diese Aussage ist nur falsch, wenn die Sonne scheint und trotzdem kein Sonnenbrand auftritt

a	b	$y = a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Inverse Implikation: $y = a \leftarrow b = \neg b \vee a$

a	b	$y = a \leftarrow b$
0	0	1
0	1	0
1	0	1
1	1	1

Äquivalenz und Antivalenz

Äquivalenz: $y = a \leftrightarrow b = (\neg a \wedge \neg b) \vee (a \wedge b)$

Wahr, wenn beide Operanden den gleichen Wahrheitswert besitzen

a	b	$y = a \leftrightarrow b$
0	0	1
0	1	0
1	0	0
1	1	1

Antivalenz: $y = a \nleftrightarrow b = (\neg a \wedge b) \vee (a \wedge \neg b)$

Wahr, wenn beide Operanden verschiedene Wahrheitswerte besitzen

Exklusives Oder (XOR)

a	b	$y = a \nleftrightarrow b$
0	0	0
0	1	1
1	0	1
1	1	0

Sheffer-Funktion und Peirce-Funktion

Sheffer-Funktion: $y = a \mathbin{\bar{\wedge}} b = \neg(a \wedge b)$

Konjunktion mit anschießender Negation (NAND)

a	b	$y = a \mathbin{\bar{\wedge}} b$
0	0	1
0	1	1
1	0	1
1	1	0

Peirce-Funktion: $y = a \mathbin{\bar{\vee}} b = \neg(a \vee b)$

Disjunktion mit anschließender Negation (NOR)

a	b	$y = a \mathbin{\bar{\vee}} b$
0	0	1
0	1	0
1	0	0
1	1	0

Operatorsymbole der Schaltalgebra nach DIN 66000

Operator	Benennung	Beispiel	Sprechweise
$\neg$ $\bar{}$	Negation, NOT	$y = \neg a = \bar{a}$	nicht a
$\wedge$	Konjunktion, AND	$y = a \wedge b$	a und b
$\vee$	Disjunktion, OR	$y = a \vee b$	a oder b
$\rightarrow$	Implikation	$y = a \rightarrow b$	a impliziert b
$\leftrightarrow$	Äquivalenz	$y = a \leftrightarrow b$	a äquivalent b
$\nleftrightarrow$	Antivalenz, XOR	$y = a \nleftrightarrow b$	a xor b
$\bar{\wedge}$	Sheffer-Funktion, NAND	$y = a \mathbin{\bar{\wedge}} b$	a nand b
$\bar{\vee}$	Peirce-Funktion, NOR	$y = a \mathbin{\bar{\vee}} b$	a nor b

Operatorsymbole der Schaltalgebra nach DIN 66000

Gemäß DIN 66000 besitzen die Symbole folgende Vorrangregeln:

- Stärkste Bindung: $\neg$
- Mittlere Bindung: $\wedge$, $\vee$, $\bar{\wedge}$, $\bar{\vee}$
- Niedrige Bindung: $\rightarrow$, $\leftrightarrow$, $\nleftrightarrow$

Symbole mit gleicher Bindungsstärke werden linksassoziativ ausgewertet

Beispiele:

$$y = a \wedge \neg b = a \wedge (\neg b)$$
$$y = a \wedge b \wedge c = (a \wedge b) \wedge c$$
$$y = a \vee b \vee c = (a \vee b) \vee c$$
$$y = a \vee b \wedge c = (a \vee b) \wedge c$$
$$y = a \leftrightarrow \neg b \vee \neg c \wedge d = a \leftrightarrow (((\neg b) \vee (\neg c)) \wedge d)$$

Operatorsymbole der Schaltalgebra, US-Schreibweise

Operator	Benennung	Beispiel	Sprechweise
$-$	negation, NOT	$y = \bar{a}$	not a
$\cdot$	conjunction, AND	$y = a \cdot b = ab$	a and b
$+$	disjunction, OR	$y = a + b$	a or b
$\implies$	implication	$y = a \implies b$	a implies b
$\equiv$ $\overline{\oplus}$	equivalence, (E)XNOR	$y = a \equiv b = \overline{a \oplus b}$	a exnor b
$\oplus$	exclusive disjunction, (E)XOR	$y = a \oplus b$	a exor b
$\overline{\cdot}$	NAND	$y = \overline{a \cdot b} = \overline{ab}$	a nand b
$\overline{+}$	NOR	$y = \overline{a + b}$	a nor b

Thorsten Thormählen 28 / 31

Abgekürzte Schreibweise boolescher Ausdrücke

- In dieser Vorlesung wird eigentlich durchgängig die DIN-Schreibweise verwendet
- Ein entscheidender Unterschied zwischen DIN und US-Schreibweise ist die Bindung der UND-Verknüpfung
- Während bei der DIN-Norm UND- sowie ODER-Verknüpfung gleichrangig sind, gilt in der US-Schreibweise analog zur Algebra "Punkt- vor Strichrechnung"
  - US-Schreibweise: $a + b \cdot c = a + (b \cdot c) = a + bc$
  - DIN-Schreibweise: $a \lor b \land c = (a \lor b) \land c$
- Abweichend von der DIN-Norm wird in dieser Vorlesung (und den Übungen) analog zur US-Version teilweise die verkürzte Schreibweise $ab = (a \land b)$ angewendet, um die Ausdrücke lesbarer zu machen
- Ausschließlich diese verkürzte Schreibweise der UND-Verknüpfung soll eine höhere Bindung haben, als die anderen Operatoren. Die stärkste Bindung hat weiterhin die Negation.
  - DIN-Schreibweise: $(a \land b \land \neg(c \land d)) \lor (\neg e \land f) = (a \land b \land \overline{c \land d}) \lor (\bar{e} \land f)$
  - Abgekürzte Schreibweise: $ab\overline{cd} \lor \bar{e}f$

Zusammenfassung: Alle zweistelligen Schaltfunktionen

a	0011	
b	0101	
$y = 0$	0000	Nullfunktion
$y = a \wedge b$	0001	Konjunktion
$y = a \wedge \neg b$	0010	
$y = a$	0011	
$y = \neg a \wedge b$	0100	
$y = b$	0101	
$y = a \nleftrightarrow b$	0110	Antivalenz
$y = a \vee b$	0111	Disjunktion
$y = a \mathbin{\bar{\vee}} b$	1000	Peirce-Funktion, NOR
$y = a \leftrightarrow b$	1001	Äquivalenz
$y = \neg b$	1010	
$y = a \leftarrow b$	1011	Inverse Implikation
$y = \neg a$	1100	
$y = a \rightarrow b$	1101	Implikation
$y = a \mathbin{\bar{\wedge}} b$	1110	Sheffer-Funktion, NAND
$y = 1$	1111	Einsfunktion

Gibt es Fragen?

Anregungen oder Verbesserungsvorschläge können auch gerne per E-mail an mich gesendet werden: [Kontakt]

[Weitere Vorlesungsfolien]

[Impressum, Datenschutz]