3 Boole'sche Algebra und Aussagenlogik

3-1 Boole'sche Algebra

Formale Grundlagen der Informatik I

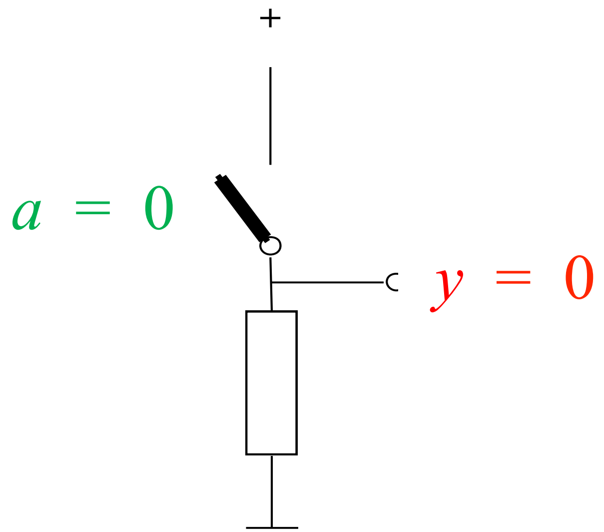
Herbstsemester 2012

Robert Marti

Vorlesung teilweise basierend auf Unterlagen

von Prof. emer. Helmut Schauer

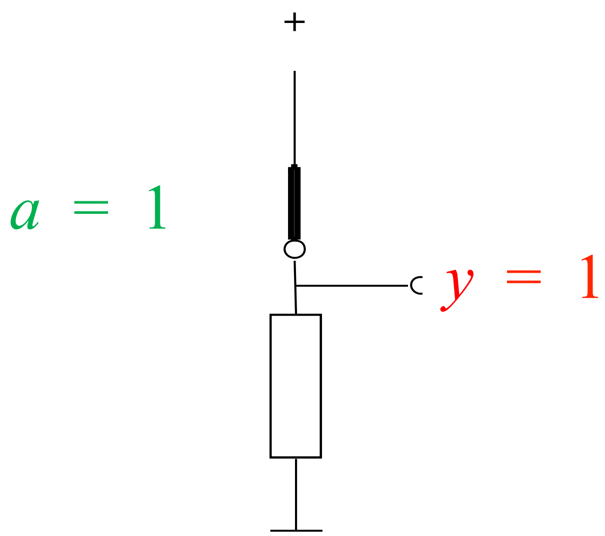
Einfache Schaltung [offen]



Schalter a offen

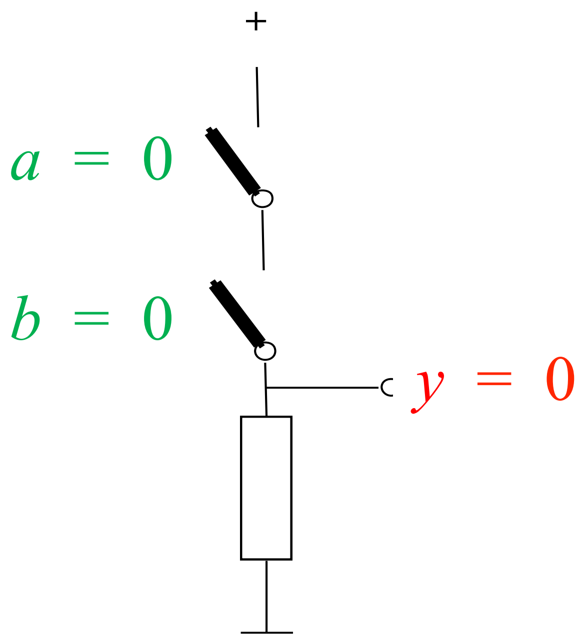
Ausgang y ohne Spannung

Einfache Schaltung [geschlossen]



Schalter a geschlossen Ausgang y unter Spannung

Serieschaltung [offen / offen]

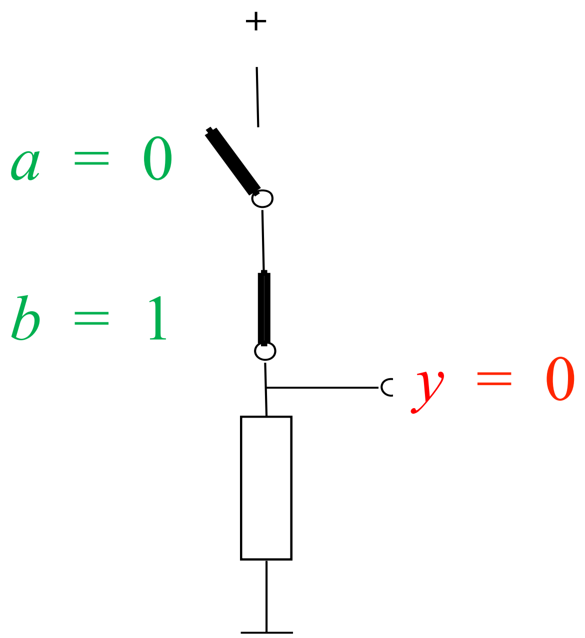


Schalter a offen

Schalter b offen

Ausgang y ohne Spannung

Serieschaltung [offen / geschlossen]

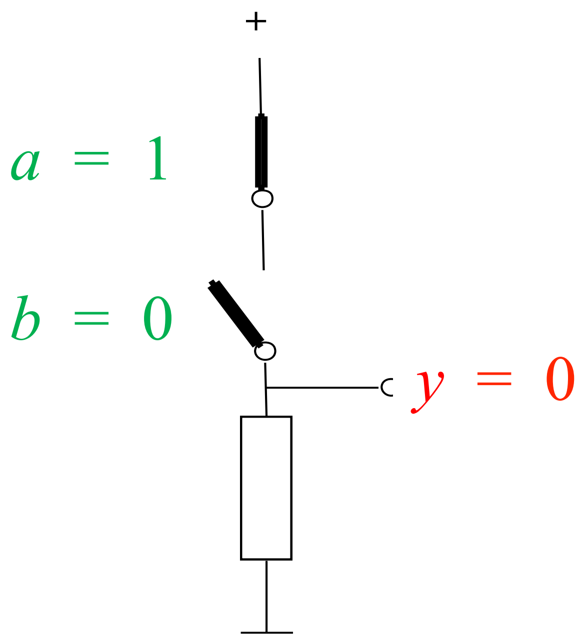


Schalter a offen

Schalter b geschlossen

Ausgang y ohne Spannung

Serieschaltung [geschlossen / offen]

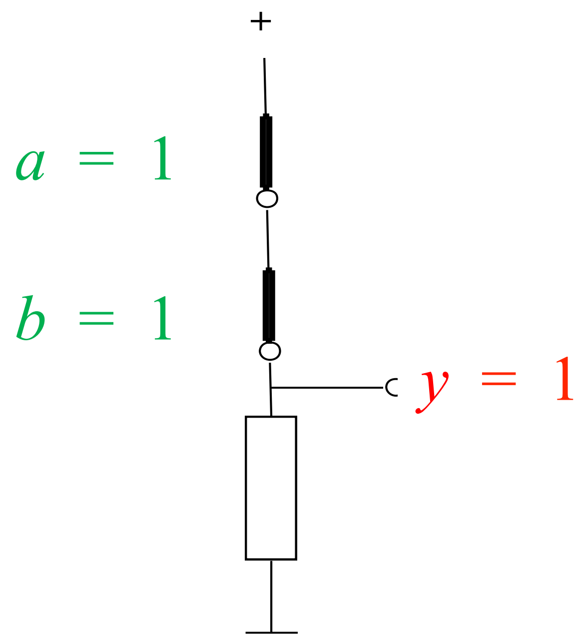


Schalter a geschlossen

Schalter b offen

Ausgang y ohne Spannung

Serieschaltung [geschlossen / geschlossen]



Schalter a geschlossen

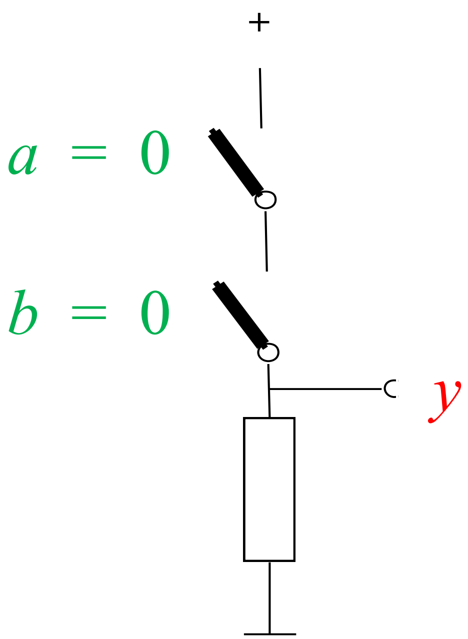
Schalter b geschlossen

Ausgang y unter Spannung

Konjunktion (and) [1]

$y = a \wedge b$ auch: a **and** b , `a & b` , `a * b` sowie a, b

Serieschaltung



Wertetabelle (truth table)

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

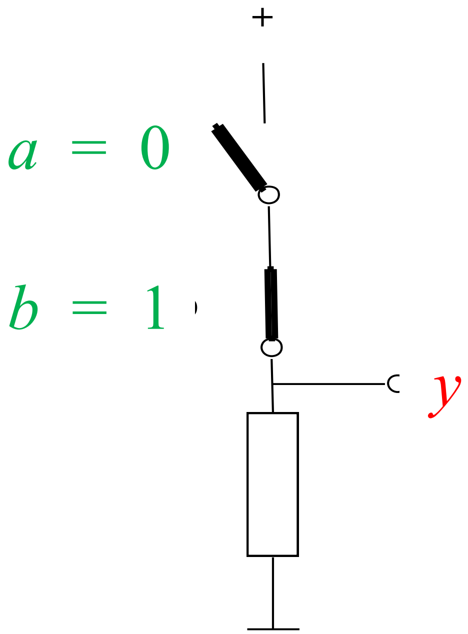
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [2]

$$y = a \wedge b$$

Serieschaltung



Wertetabelle
(truth table)

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

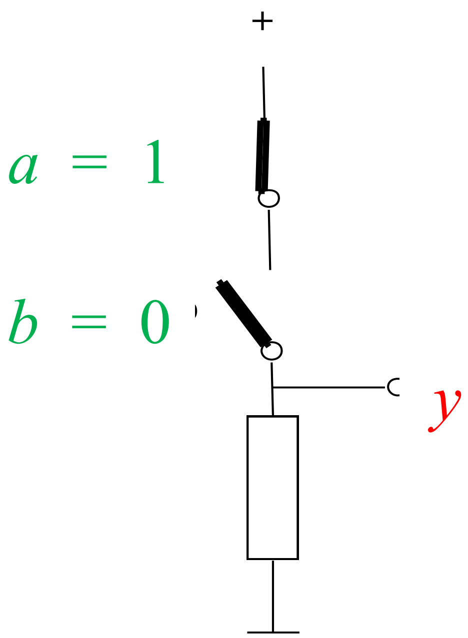
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [3]

$$y = a \wedge b$$

Serieschaltung



Wertetabelle
(truth table)

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

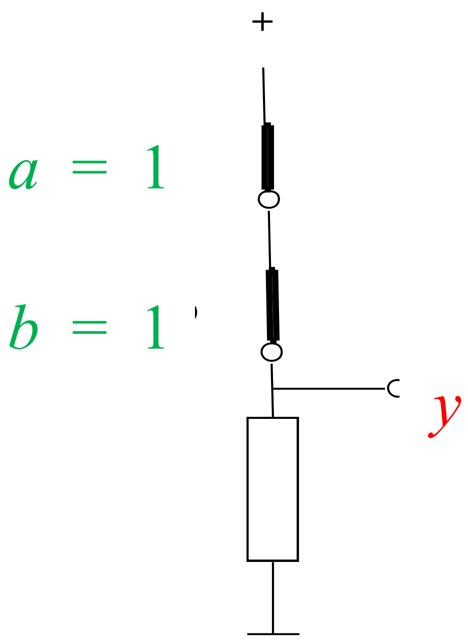
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [4]

$$y = a \wedge b$$

Serieschaltung



Wertetabelle (truth table)

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

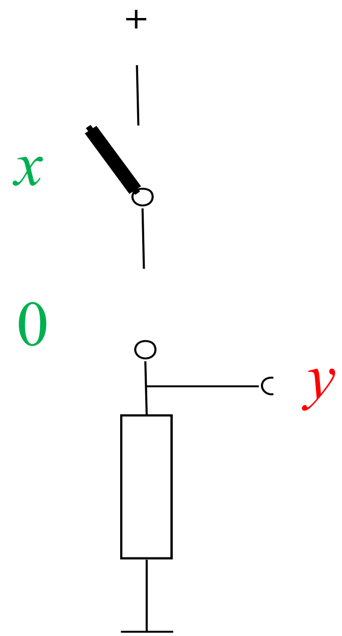
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [5]

$$y = a \wedge b$$

Serieschaltung



$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

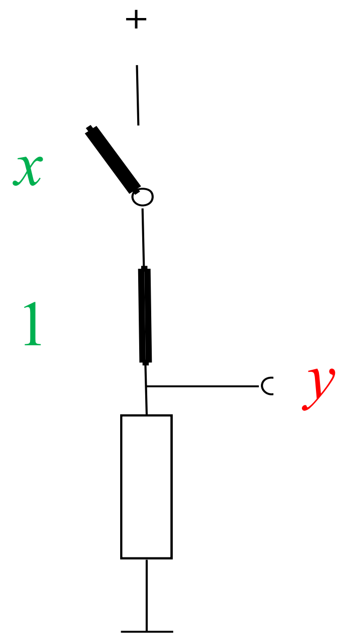
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [6]

$$y = a \wedge b$$

Serieschaltung



$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

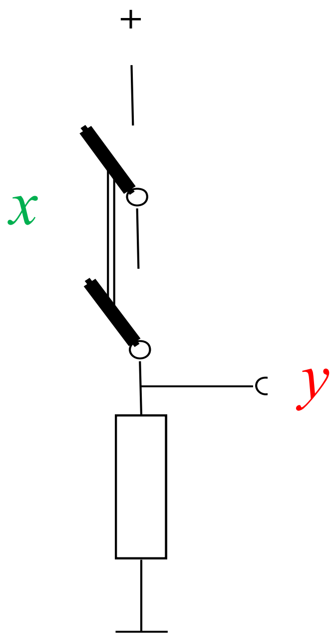
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Konjunktion (and) [7]

$$y = a \wedge b$$

Serieschaltung



$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

$x \wedge 0 = 0$

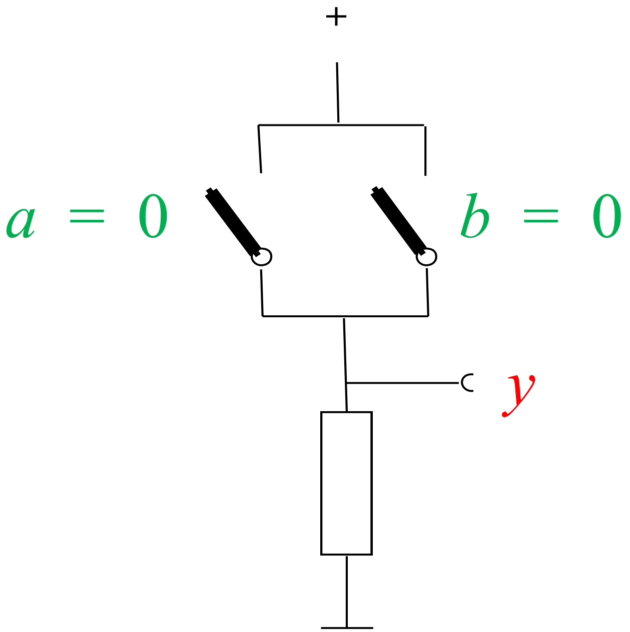
$x \wedge 1 = x$ (neutrales Element 1)

$x \wedge x = x$ (Idempotenz)

Disjunktion (or) [offen / offen]

$y = a \vee b$ auch: a **or** b , `a | b` , `a + b` sowie $a; b$

Parallelschaltung



a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)

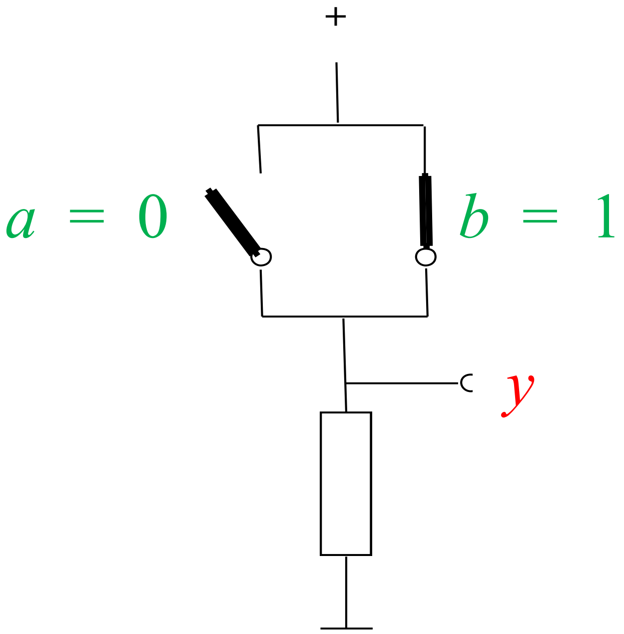
$x \vee 1 = 1$

$x \vee x = x$ (Idempotenz)

Disjunktion (or) [offen / geschlossen]

$$y = a \vee b$$

Parallelschaltung



a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)

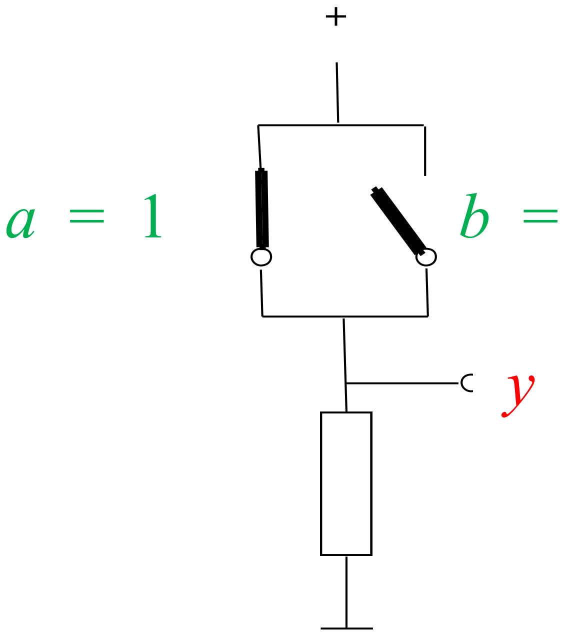
$x \vee 1 = 1$

$x \vee x = x$ (Idempotenz)

Disjunktion (or) [geschlossen / offen]

$$y = a \vee b$$

Parallelschaltung



a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

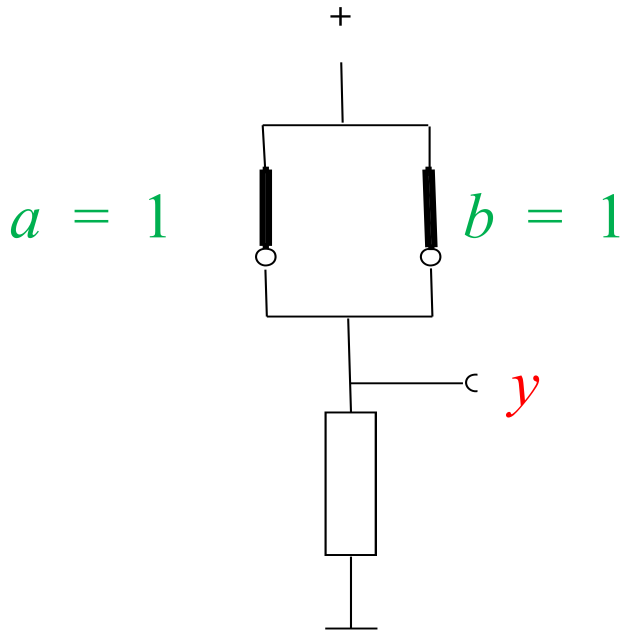
$0 \vee 0 = 0$
$0 \vee 1 = 1$
$1 \vee 0 = 1$
$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)
$x \vee 1 = 1$
$x \vee x = x$ (Idempotenz)

Disjunktion (or) [geschlossen / geschlossen]

$$y = a \vee b$$

Parallelschaltung



a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

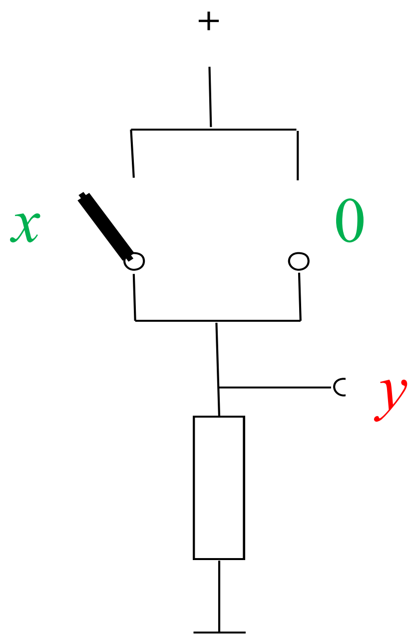
$0 \vee 0 = 0$
$0 \vee 1 = 1$
$1 \vee 0 = 1$
$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)
$x \vee 1 = 1$
$x \vee x = x$ (Idempotenz)

Disjunktion (or) [1]

$$y = a \vee b$$

Parallelschaltung



$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)

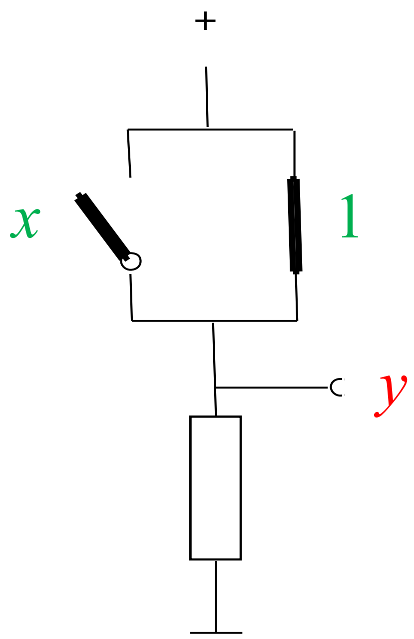
$x \vee 1 = 1$

$x \vee x = x$ (Idempotenz)

Disjunktion (or) [2]

$$y = a \vee b$$

Parallelschaltung



$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)

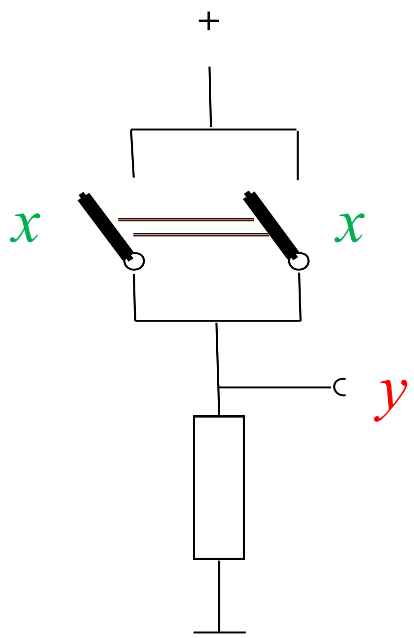
$x \vee 1 = 1$

$x \vee x = x$ (Idempotenz)

Disjunktion (or) [3]

$$y = a \vee b$$

Parallelschaltung



$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

$x \vee 0 = x$ (neutrales Element 0)

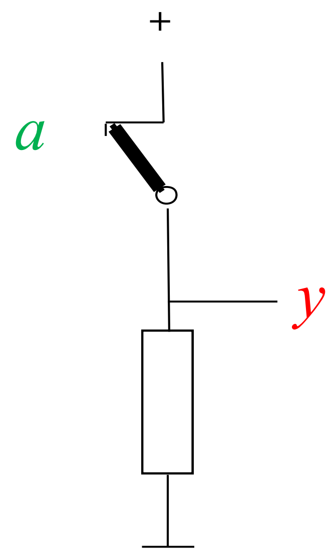
$x \vee 1 = 1$

$x \vee x = x$ (Idempotenz)

Negation (not)

$y = \neg a$ auch: **not** a , $\bar{a}$, `!a` , `~a` , `-a` und sogar `\+ a`

Ruhekontakt



a	y
0	1
1	0

$$\neg 0 = 1$$
$$\neg 1 = 0$$

Wertetabellen (Zusammenfassung)

oder
Disjunktion

∨ wie lat. **v**el (oder)

und
Konjunktion

∧ wie engl. **A**nd

nicht
Negation

∨	0	1
0	0	1
1	1	1

∧	0	1
0	0	0
1	0	1

¬	
0	1
1	0

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

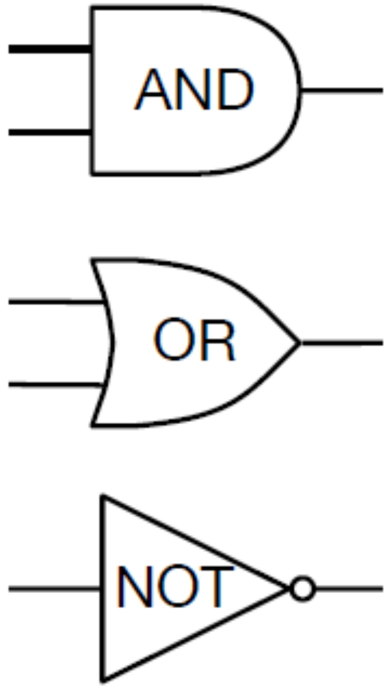
a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

a	$\neg a$
0	1
1	0

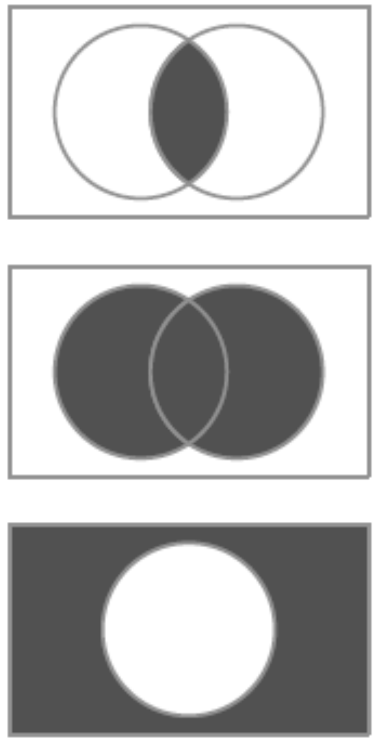
Notation – Variationen eines Themas ...

$a \wedge b$	$a * b$	$a \& b$
$a \vee b$	$a + b$	$a \mid b$
$\neg a$	$- a$	$! a$

Schaltungs-
symbole



Analogie in
Mengenlehre



Einige bekannte Gesetze (vgl. auch Gesetze für ganze Zahlen)

Kommutative Gesetze

$$a \wedge b = b \wedge a$$

$$a \vee b = b \vee a$$

Assoziative Gesetze

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c$$

$$a \vee (b \vee c) = (a \vee b) \vee c$$

Distributive Gesetze

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

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

Neutrale Elemente

$$a \wedge 1 = a$$

$$a \vee 0 = a$$

Komplement

$$a \wedge \neg a = 0$$

$$a \vee \neg a = 1$$

Weitere Gesetze

Gesetze der Verschmelzung (auch: Absorption)

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

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

Gesetz der doppelten Negation

$$\neg(\neg a) = a$$

DeMorgan'sche Gesetze

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

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

Beweis der Aequivalenz Boole'scher Formeln

Wie werden solche Gesetze bewiesen?

Grundprinzip:

Für jede vorkommende Variable werden alle möglichen Werte (jeweils 0 oder 1) eingesetzt ("Input").
Danach werden linke und rechte Seiten verglichen ("Output").
Wenn jeder Input das gleiche Resultat erzeugt, ist das Gesetz bewiesen.

Beispiel: DeMorgan'sches Gesetz $\neg(a \wedge b) = \neg a \vee \neg b$

	Output (links)	Input				Output (rechts)
$a \wedge b$	$\neg(a \wedge b)$	a	b	$\neg a$	$\neg b$	$\neg a \vee \neg b$
0	1	0	0	1	1	1
0	1	0	1	1	0	1
0	1	1	0	0	1	1
1	0	1	1	0	0	0

Alle Boole'schen Funktionen mit (max.) zwei Parametern

Total 16 mögliche "Outputs"
(gewisse dieser Funktionen tauchen später wieder auf)

Inputs →

Outputs →

a 0011 b 0101	
0000	$y_0 = 0$
0001	$y_1 = a \wedge b$
0010	$y_2 = a \wedge \neg b$
0011	$y_3 = a$
0100	$y_4 = \neg a \wedge b$
0101	$y_5 = b$
0110	$y_6 = a \neq b$
0111	$y_7 = a \vee b$

a 0011 b 0101	
1111	$y_{15} = 1$
1110	$y_{14} = \neg a \vee \neg b$
1101	$y_{13} = \neg a \vee b$
1100	$y_{12} = \neg a$
1011	$y_{11} = a \vee \neg b$
1010	$y_{10} = \neg b$
1001	$y_9 = a \equiv b$
1000	$y_8 = \neg a \wedge \neg b$

Bem.: y_0 und y_{15} haben kein Argument, y_3, y_5, y_{10} und y_{12} nur je ein Argument.

Dualität

Auf der vorangehenden Folie sind auf jeder Zeile die Outputs der linken Tabelle jeweils die Negation der Outputs der rechten Spalte.

Es fällt auf, dass in den entsprechenden Formeln

- die Argumente a und b (bzw. $\neg a$ und $\neg b$) negiert sind: $\neg a$ und $\neg b$ (bzw. a und b)
- die Operatoren Konjunktionen ($\wedge$) statt Disjunktionen ($\vee$) sind (bzw. umgekehrt)

Zu jeder Funktion f von max. 2 Variablen gibt es eine **duale Funktion** g, so dass gilt (bzw.: f heisst dual zu g falls gilt):

$\neg f(x, y) = g(\neg x, \neg y)$ bzw. $f(x, y) = \neg g(\neg x, \neg y)$

Beispiele für duale Funktionen:

0 (y_0) und 1 (y_{15})

$\wedge$ und $\vee$

nand (y_{14}) und **nor** (y_8) – siehe auch später

Beispiele von Funktionen, die zu sich selbst dual sind

f(x,y)	!f(x,y)	f(!x,!y)	
y0(a,b) = 0	= 1	y0(!a,!b) = 0	
y1(a,b) = a & b	= !a | !b	y1(!a,!b) = !a & !b	
y2(a,b) = a & !b	= !a | b	y2(!a,!b) = !a & b	
y3(a,b) = a	= !a	y3(!a,!b) = !a	D
y4(a,b) = !a & b	= a | !b	y4(!a,!b) = a & !b	

...
etc. (exercise left to the reader)
...

Beispiele von Funktionen, die kommutativ sind

f(x,y)	f(y,x)	
y0(a,b) = 0	y0(b,a) = 0	K
y1(a,b) = a & b	y1(b,a) = b & a	K
y2(a,b) = a & !b	y2(b,a) = b & !a	
y3(a,b) = a	y3(b,a) = b	
y4(a,b) = !a & b	y4(b,a) = !b & a	

...

etc. (exercise left to the reader)

...

Boole'sche Algebra

Eine Boole'sche Algebra ist ein abgeschlossenes System, in welchem 2 Operationen definiert sind, für die

- Kommutativgesetz
- Assoziativgesetz
- Distributivgesetz
- Verschmelzungsgesetz (=Absorptionsgesetz)

gelten und in dem es

- ein Nullelement
- ein Einselement
- sowie zu jedem Element ein komplementäres Element

gibt.

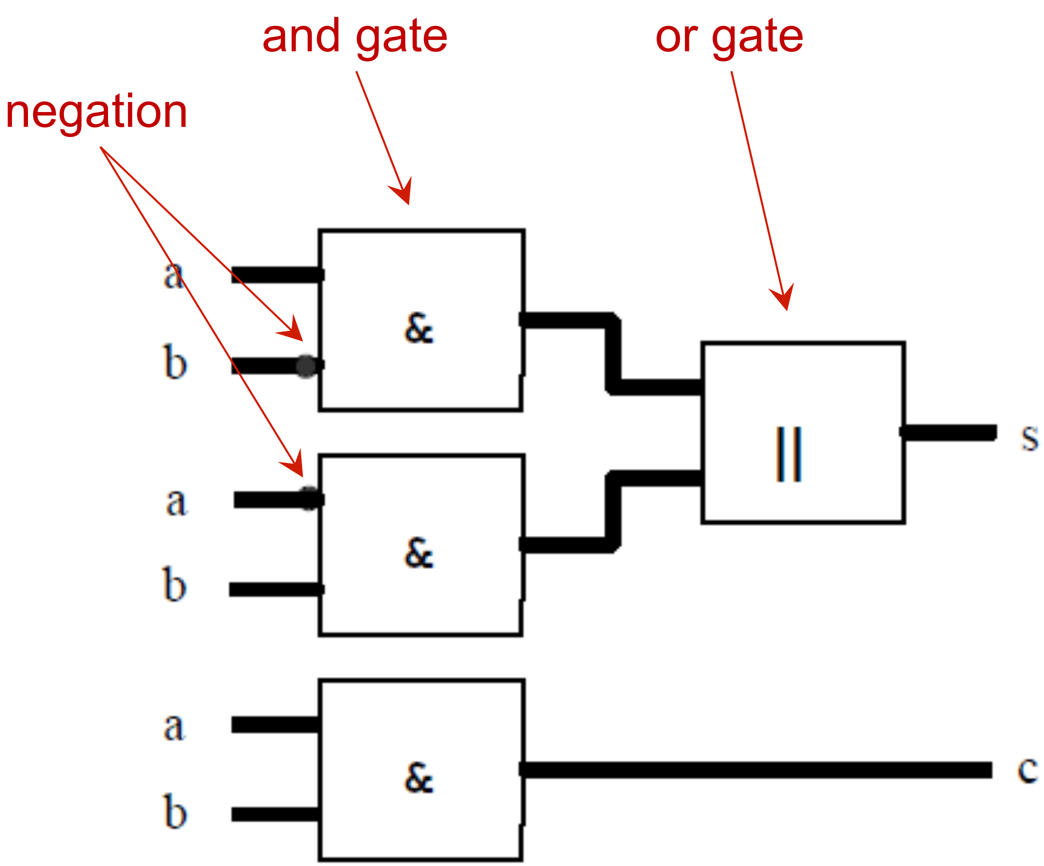
Beispiele für Boole'sche Algebren

- Schaltalgebra (wie gehabt + demnächst mehr dazu)

- Aussagenlogik (später mehr dazu)

- Mengenlehre
  - Vereinigung zweier Mengen ($\cup$) entspricht dem logischen **or**
  - Durchschnitt zweier Mengen ($\cap$) entspricht dem logischen **and**
  - Komplement einer Menge entspricht dem logischen **not**
  - Die leere Menge ($\emptyset$) entspricht 0
  - Die Gesamtmenge (Ω) entspricht 1

Anwendung Halbaddierer (Half Adder)



s = Summe (sum,actually XOR)

c = Uebertrag (carry)

a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

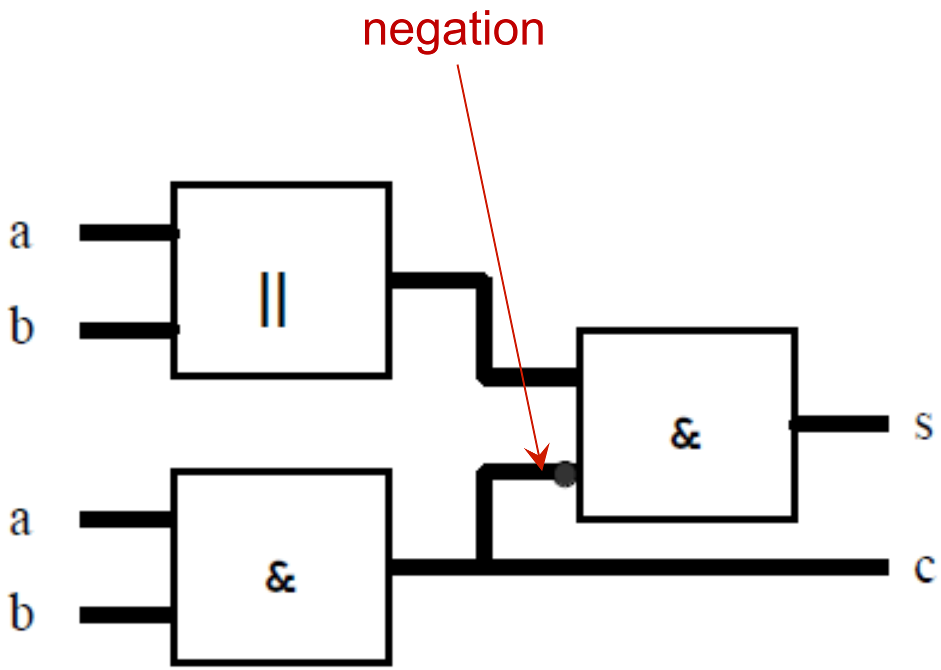
$$s = (\neg a \land b) \lor (a \land \neg b)$$
$$c = (a \land b)$$

Anwendung Halbaddierer: Vereinfachungen

$$
\begin{aligned}
s &= (\neg a \wedge b) \vee (a \wedge \neg b) = \\
&= ((\neg a \wedge b) \vee a) \wedge ((\neg a \wedge b) \vee \neg b) = \\
&= (\neg a \vee a) \wedge (b \vee a) \wedge (\neg a \vee \neg b) \wedge (b \vee \neg b) = \\
&= (b \vee a) \wedge (\neg a \vee \neg b) = \\
&= (a \vee b) \wedge \neg (a \wedge b)
\end{aligned}
$$

$$c = (a \wedge b)$$

Anwendung Halbaddierer: Optimiert

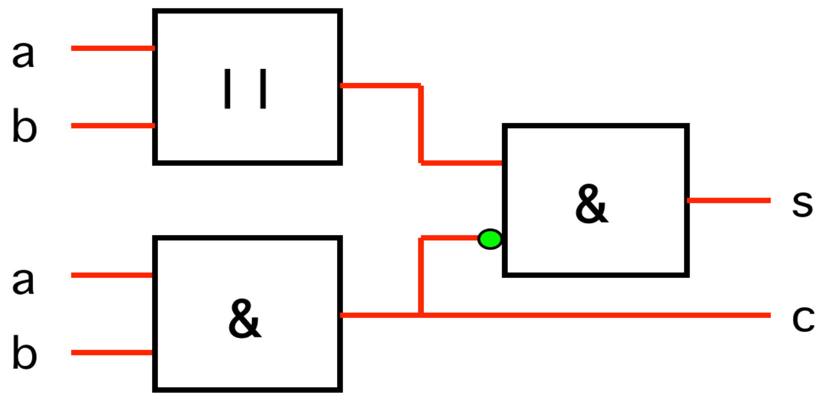


a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

Anwendung Halbaddierer

- Halbaddierwerk

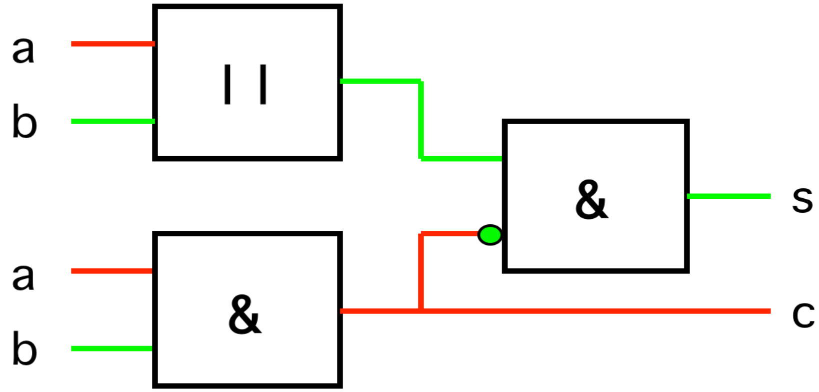


a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

Anwendung Halbaddierer

- Halbaddierwerk

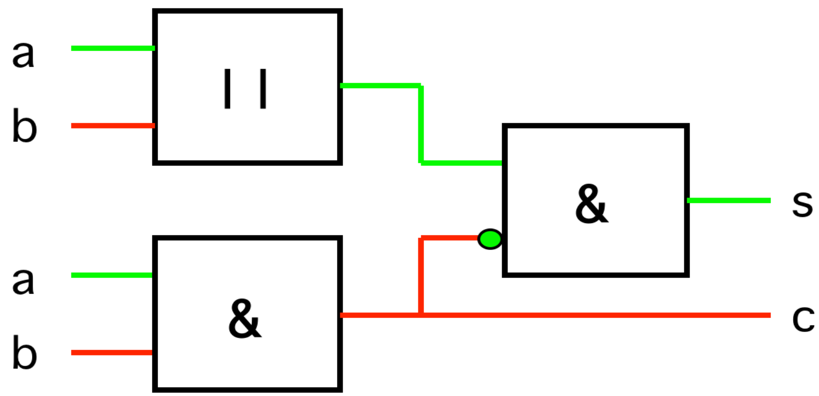


a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

Anwendung Halbaddierer

- Halbaddierwerk

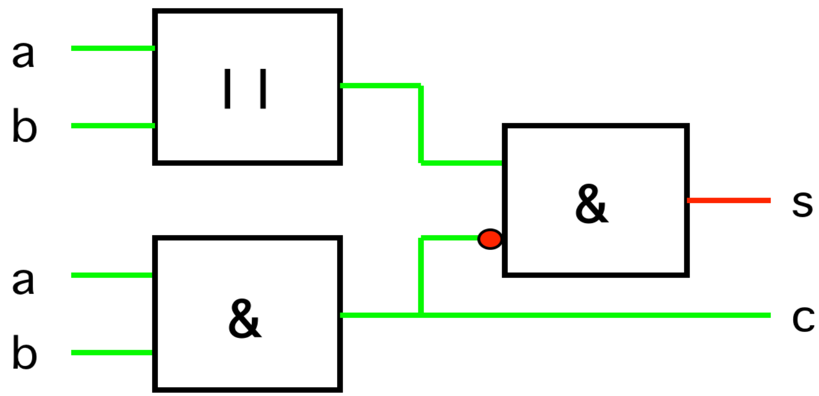


a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$s = (a \lor b) \land \neg(a \land b)$
$c = (a \land b)$

Anwendung Halbaddierer

- Halbaddierwerk



a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

Spezielle Boole'sche Funktionen mit 2 Parametern

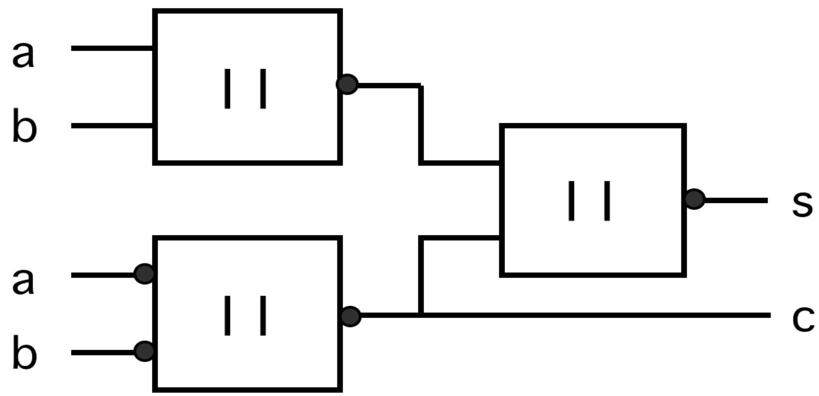
Peirce-Funktion **nor** (sog. nor-gate, nor = **n**ot **or**)

$$y_8 = \neg a \wedge \neg b = \neg(a \vee b)$$

Sheffer-Funktion **nand** (sog. nand-gate, nand = **n**ot **and**)

$$y_{14} = \neg a \vee \neg b = \neg(a \wedge b)$$

Halbaddierer: Verwendung von NOR Gattern

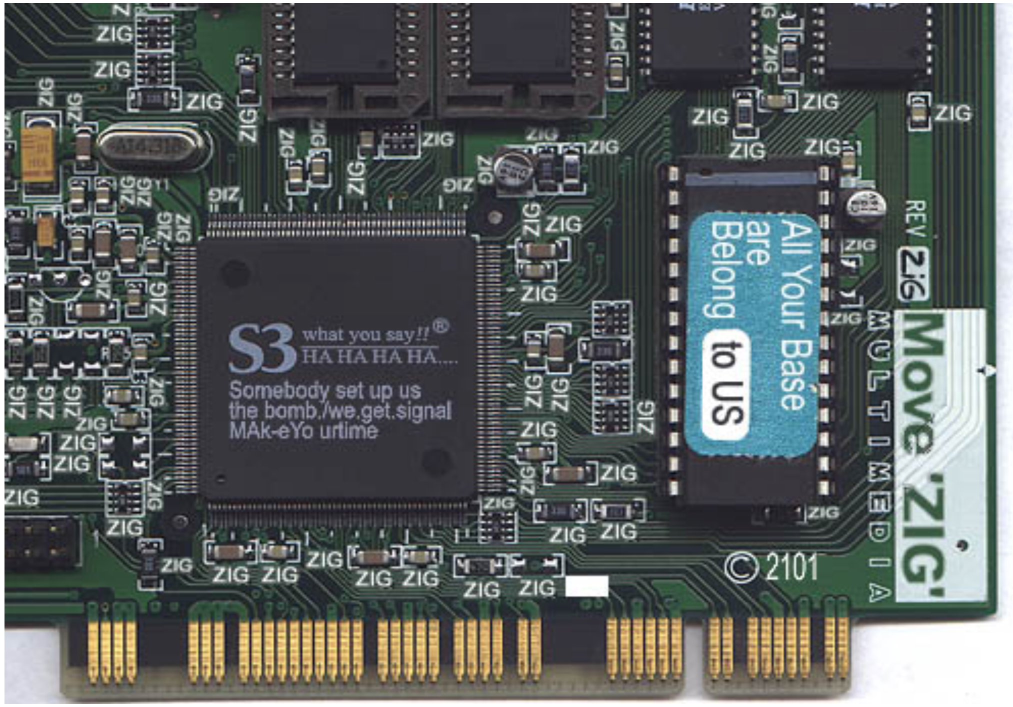


a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

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

Prozessor


S3
what you say!!®
HA HA HA HA....
Somebody set up us
the bomb./we.get.signal
MAk-eYo urtime
All Your Base
are
Belong to US
Move 'ZIG'
MULTIMEDIA
REV
© 2101