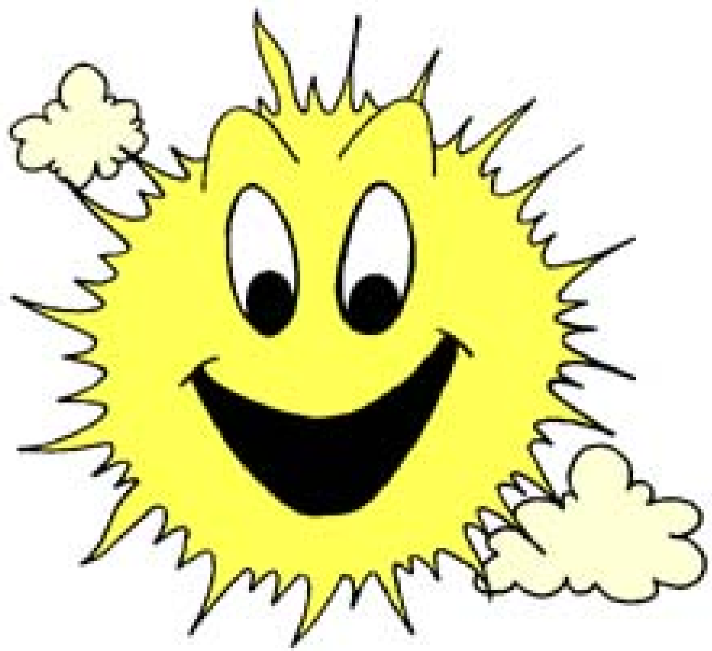
Literatur (DT und RG)

- **Computerarchitektur**
  Strukturen - Konzepte - Grundlagen
  Andrew S. Tanenbaum, James Goodman
  Verlag: Pearson Studium – „Bafög“-Ausgabe
- **Digitaltechnik**
  K. Beuth, O. Beuth
  Verlag. Vogel Fachbuch 2003

Wiederholung 2. Vorlesung



2 Boolesche Algebra

Informatik

Die Boolesche Algebra (Schaltalgebra) dient der Beschreibung digitaler Funktionen.

Ähnlich wie in der Algebra werden Variablen über Operatoren verknüpft. Es lassen sich Gleichungen aufstellen und umformen.

Bei der Booleschen Algebra nehmen die Variablen nur die Werte 0 und 1 an.

1854: Theorie durch Mathematiker George Boole

1940: Claude E. Shannon: Anwendung auf Schaltnetze

2.1 Boolesche Menge

- Eine Boolesche Menge besteht aus zwei unterscheidbaren Elementen.
- „0“ und „1“ werden in der Schaltalgebra verwendet.
- „F(alse)“ und „T(rue)“ werden zur Beschreibung logischer Verknüpfungen verwendet.
- „L(ow)“ und „H(igh)“ werden zur Beschreibung elektrischer Verknüpfungen verwendet.

3.1 Schließer und Öffner

$X = 0$: Schalter offen

Schließer

$X = 1$: Schalter geschlossen

$X = 0$: Schalter geschlossen

$X = 1$: Schalter offen

Öffner

3.1 Negation (NOT)

- „Wenn meine Schwiegermutter zu Besuch kommt, gehe ich heute Abend nicht ins Theater."
- Wenn die Aussage A „Schwiegermutter kommt zu Besuch" wahr ist, kann die Aussage X „Theaterbesuch" nicht wahr sein.
- Unäre Operation

3.2 UND Verknüpfung (AND)

- „Wenn morgen schönes Wetter ist und mein Bruder Zeit hat, gehen wir segeln."
- Aussage A „schönes Wetter" und Aussage B „mein Bruder Zeit hat" müssen zutreffen, damit die Aussage X „segeln gehen" wahr wird.
- Binäre Operation

3.2 ODER-Verknüpfung (OR)

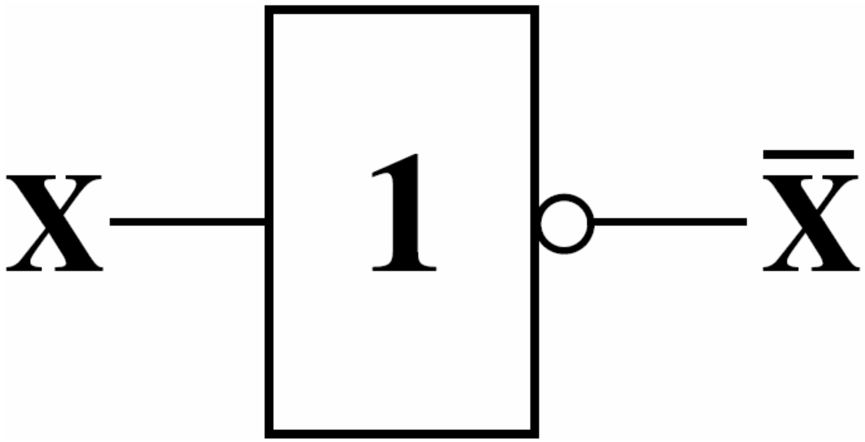
- „Wenn ich eine Erbschaft mache oder im Lotto gewinne, mache ich eine Weltreise."
- Wenn Aussage A „Erbschaft" oder Aussage B „Lottogewinn" zutrifft, oder beide Aussagen zutrefffen, wird Aussage X „Weltreise machen" wahr.
- Binäre Operation

3. Symbole und Zeichen logischer Funktionen

- Realisierung durch Schalter
- Schaltsymbole nach DIN 40900
- Formelzeichen der Booleschen Algebra
- Wahrheitstabellen (Funktion in Tabellenform)
- Weitere Schaltsymbole nach DIN und ANSI / ISO (teils veraltete Normen)

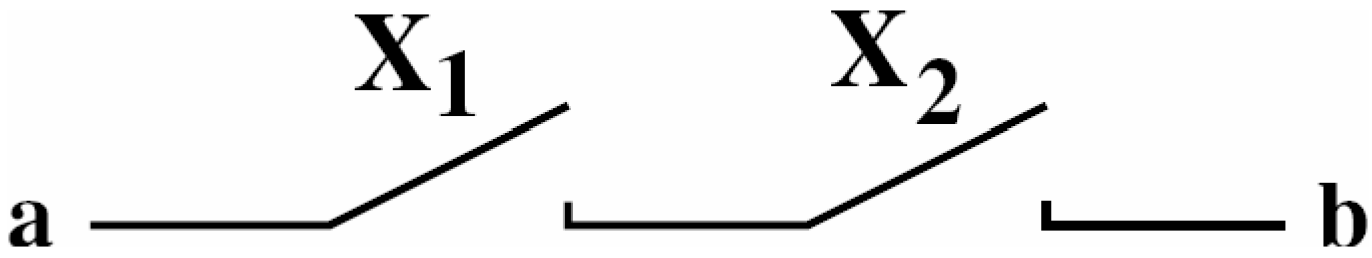
3.1 Negation (NOT)

X	$\bar{X}$
0	1
1	0

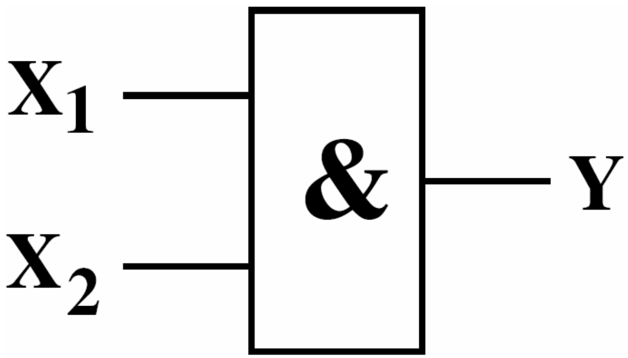


Negation: $\neg X$, $\bar{X}$, $!X$

3.2 UND-Verknüpfung (AND)

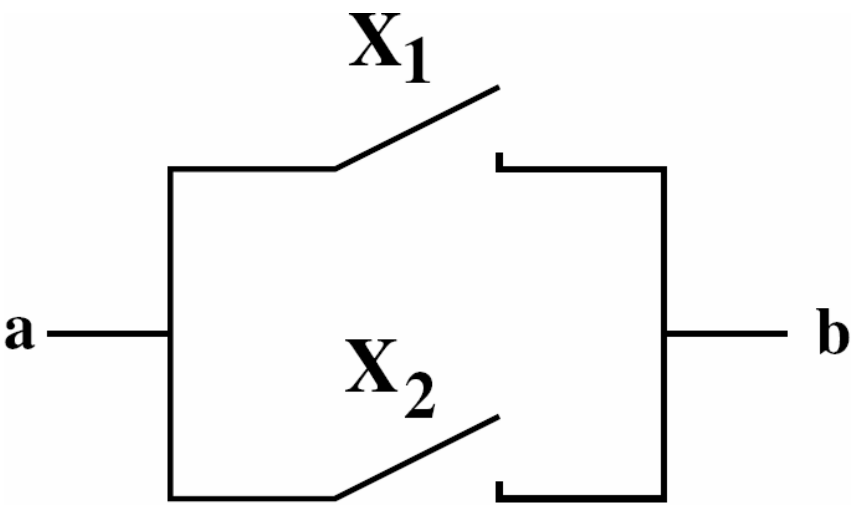


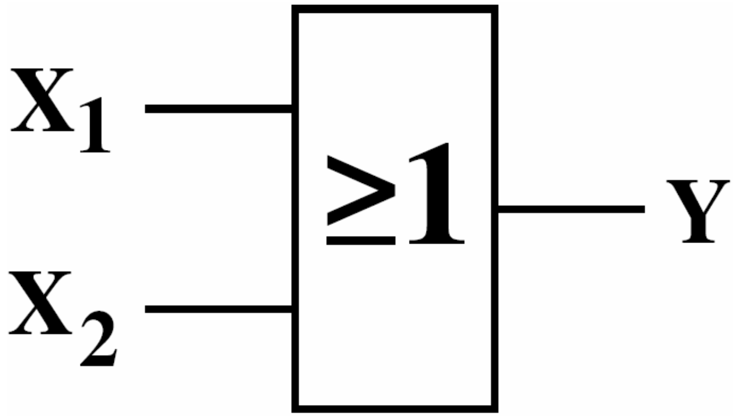
X_1	X_2	$X_1 \wedge X_2$
0	0	0
0	1	0
1	0	0
1	1	1



UND: $\wedge$, $\bullet$

3.2 ODER-Verknüpfung (OR)





X_1	X_2	$X_1 \vee X_2$
0	0	0
0	1	1
1	0	1
1	1	1

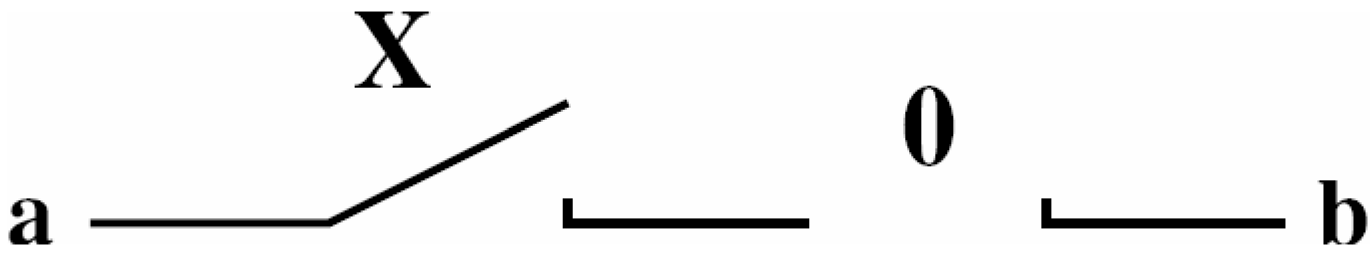
ODER: $\vee$, $+$

3. Boolesche Postulate

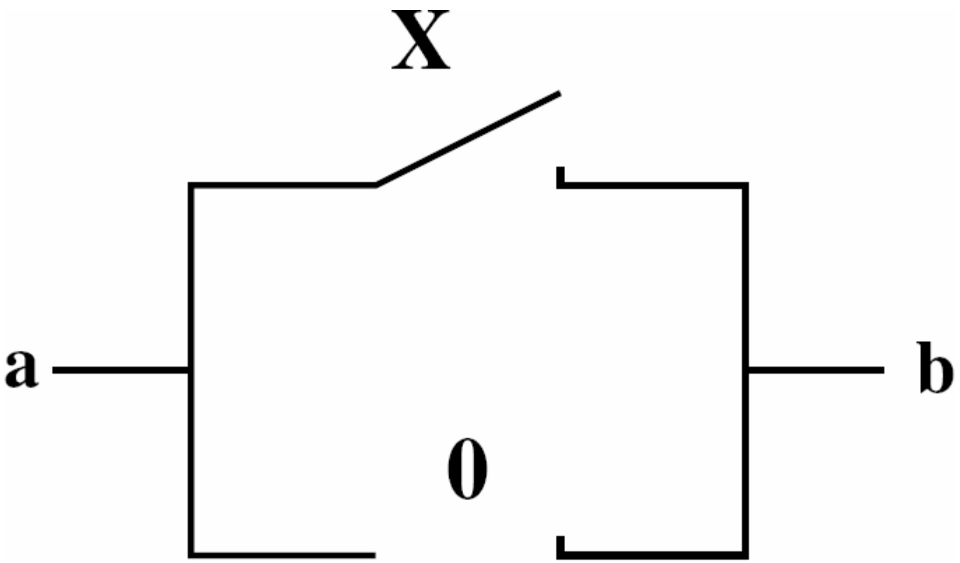
Menge A mit den Elementen 0 und 1 und den Operationen UND, ODER und Negation.

P1	a=0 oder a=1	P5	$1 \vee 1 = 1$
P2	$0 \wedge 0 = 0$	P6	$1 \wedge 0 = 0,\ 0 \wedge 1 = 0$
P3	$1 \wedge 1 = 1$	P7	$1 \vee 0 = 1,\ 0 \vee 1 = 1$
P4	$0 \vee 0 = 0$	P8	$\neg 1 = 0,\ \neg 0 = 1$

3.1 Das NULL Gesetz

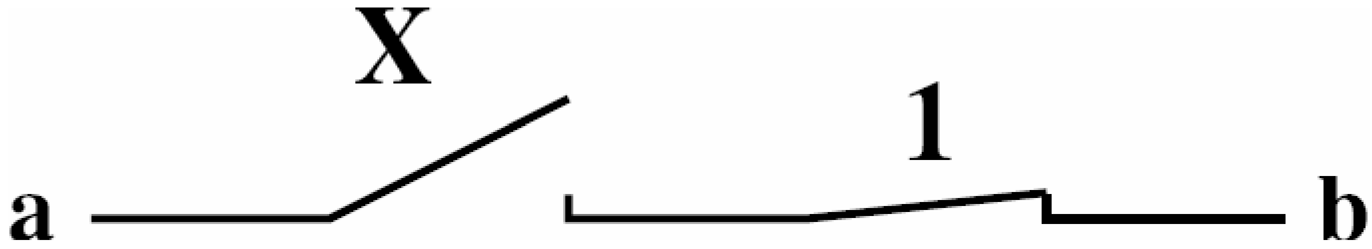


$$X \wedge 0 = 0$$

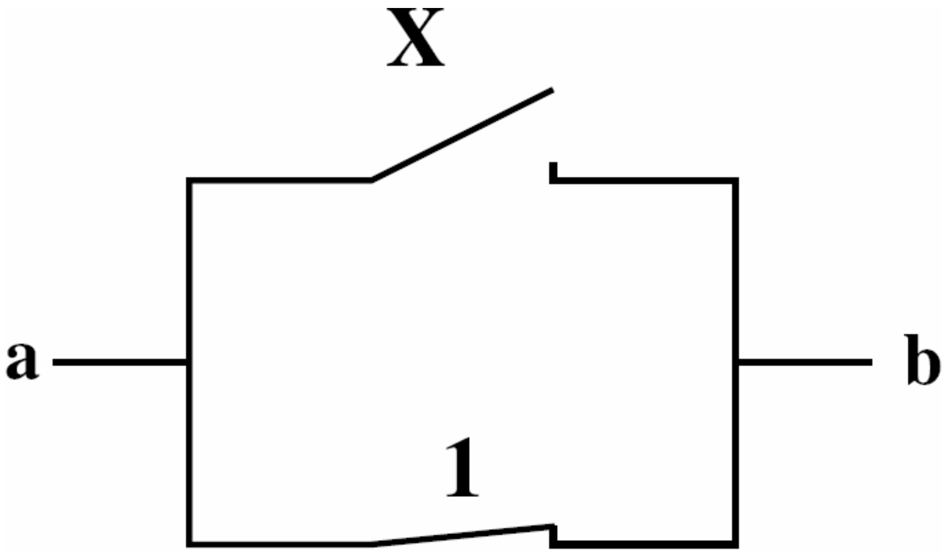


$$X \vee 0 = X$$

3.1 Das EINS Gesetz



$$X \wedge 1 = X$$



$$X \vee 1 = 1$$

4 Vollst. Systeme aus UND, ODER, NICHT

Bisher wurden die booleschen Funktionen durch die Grundverknüpfungen UND, ODER und NICHT dargestellt. Diese Verknüpfungen besitzen eine besondere Bedeutung, da man mathematisch nachweisen kann, daß sich mit diesen drei Operationen alle booleschen Funktionen darstellen lassen. Man bezeichnet daher diese Verknüpfungen auch als vollständiges System.

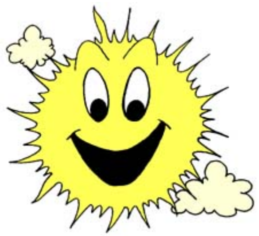
5 Vollst. Systeme mit NAND und NOR (2)

Ebenso wie die AND-, OR-Verknüpfung und NOT bilden NAND und NOR, jede für sich, ebenfalls vollständige Systeme.

Somit reicht es für die Entwicklung einer booleschen Hardware aus, entweder eine Grundschaltung für NAND oder für NOR zur Verfügung zu stellen.

Daraus kann durch Replizieren und Kombinieren jede boolesche Schaltung entwickelt werden.

Ende der Wiederholung

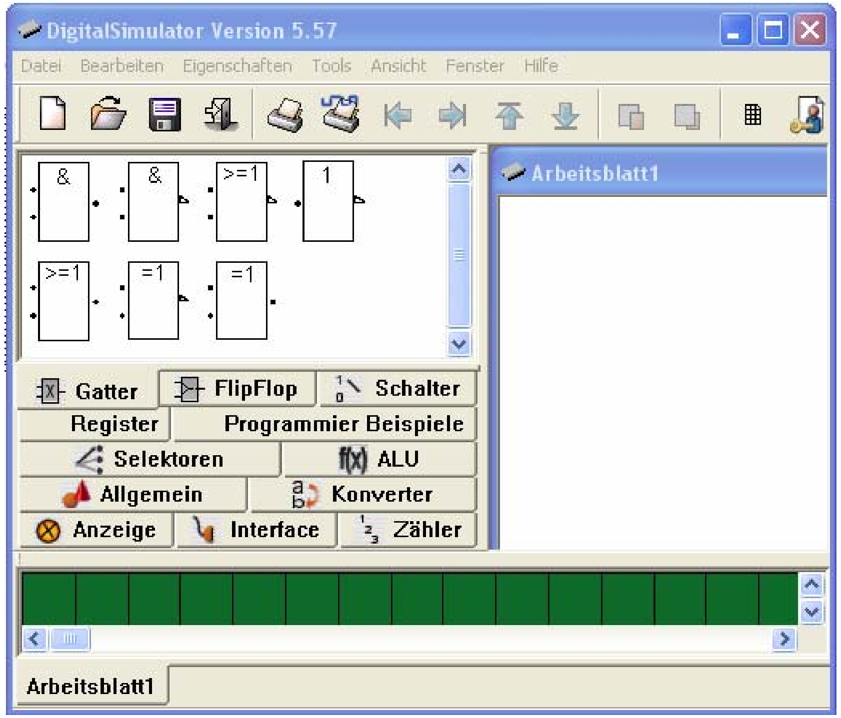


→ → → → → → → → → → → → → → → → →

Simulation digitaler Schaltungen

Simulationssoftware, beispielsweise

- Simulation digitaler Schaltungen: http:// www.digitalsimulator.de
- Hörsaal-Demo: DigitalSimulator.exe



2.4 Boolesche Gesetze

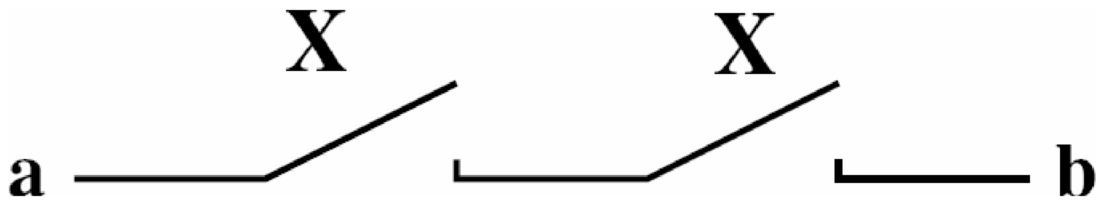
1. Doppelte Negierung
2. Idempotenzgesetz (Identitätsgesetz)
3. Komplementgesetz
4. Kommutativgesetze
5. Assoziativgesetze
6. Distributivgesetze
7. → Kürzungsregeln

2.4.1 Doppelte Negierung

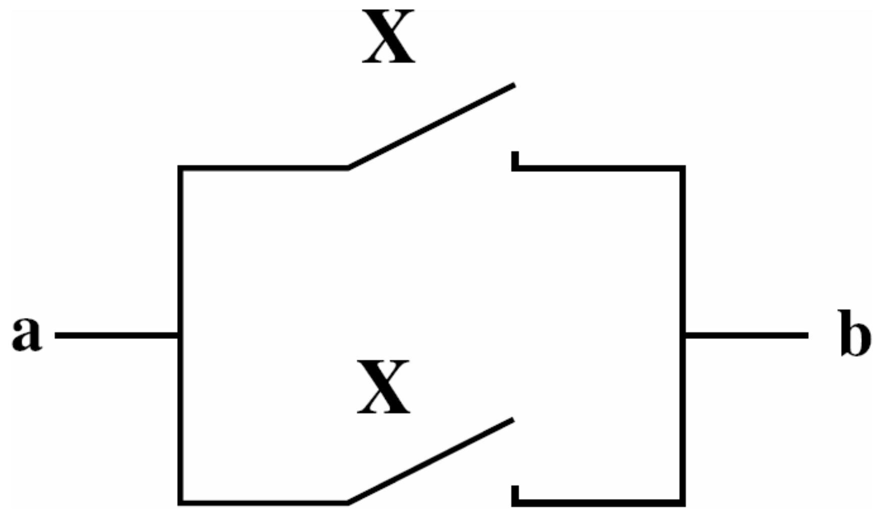
$$\overline{\overline{X}} = \overline{(\overline{X})} = X$$

2.4.2 Idempotenz- / Identitäts-Gesetze

$$X \wedge X = X$$

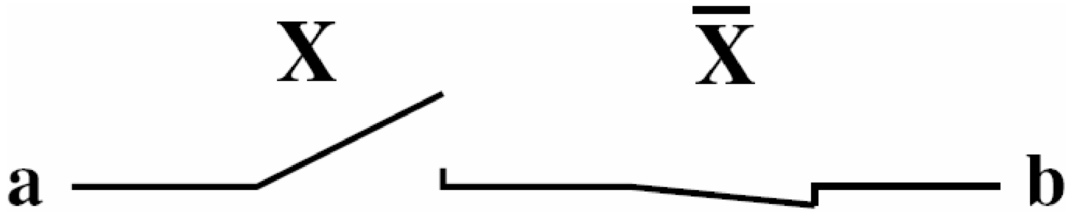


$$X \vee X = X$$

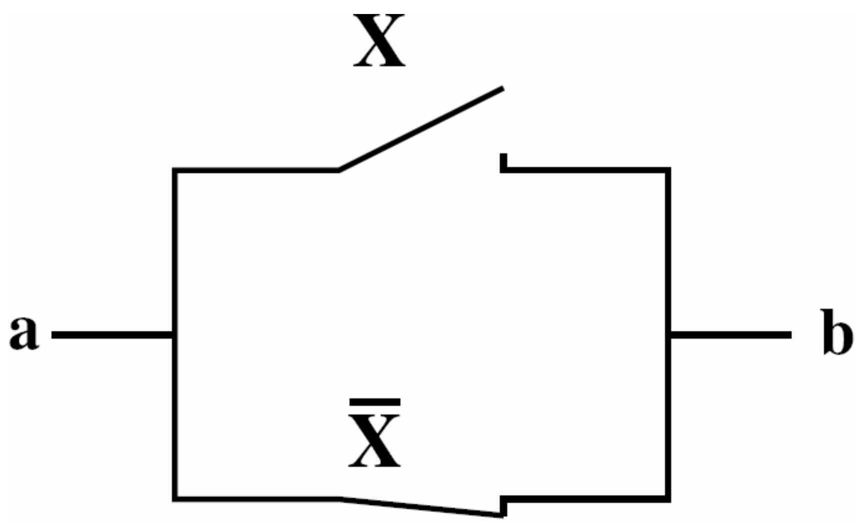


2.4.3 Komplementgesetze

$$X \wedge \bar{X} = 0$$



$$X \vee \bar{X} = 1$$



2.4.4 Kommutativgesetze

$$X_1 \wedge X_2 = X_2 \wedge X_1$$

$$X_1 \vee X_2 = X_2 \vee X_1$$

2.4.5 Assoziativgesetze

$$(X_1 \wedge X_2) \wedge X_3 = \quad X_1 \wedge (X_2 \wedge X_3)$$

$$(X_1 \vee X_2) \vee X_3 = \quad X_1 \vee (X_2 \vee X_3)$$

2.4.6 Distributivgesetze

$$X_1 \wedge (X_2 \vee X_3) = (X_1 \wedge X_2) \vee (X_1 \wedge X_3)$$

1. Distributivgesetz (konjunktiv)

$$X_1 \vee (X_2 \wedge X_3) = (X_1 \vee X_2) \wedge (X_1 \vee X_3)$$

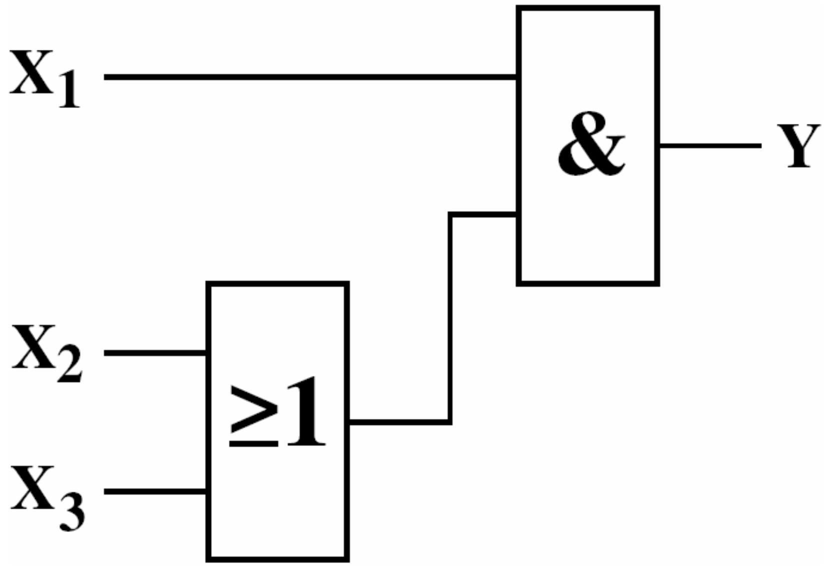
2. Distributivgesetz (disjunktiv)

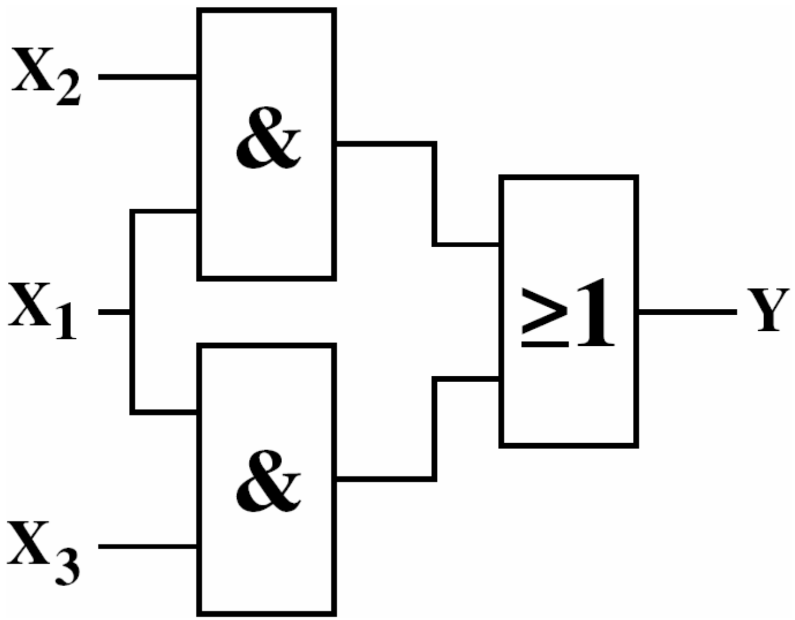
2.4.6 Das 1. Distributivgesetz

Der Ausdruck $X_1 \wedge (X_2 \vee X_3)$ ist genau dann wahr, wenn X_1 wahr ist und zugleich X_2 oder X_3 wahr ist.

Der Ausdruck $(X_1 \wedge X_2) \vee (X_1 \wedge X_3)$ ist genau dann wahr, wenn einer der beiden Klammerausdrücke wahr ist. Entweder muss X_1 und X_2 wahr sein oder es muss X_1 und X_3 wahr sein. X_1 muss also auf jeden Fall wahr sein und zugleich muss X_2 oder X_3 wahr sein.

2.4.6 Das 1. Distributivgesetz





2.4.7 Kürzungsregeln

1. $X_1 \vee (X_1 \wedge X_2) = X_1$
2. $X_1 \wedge (X_1 \vee X_2) = X_1$
3. $X_1 \vee (\overline{X_1} \wedge X_2) = X_1 \vee X_2$
4. $X_1 \wedge (\overline{X_1} \vee X_2) = X_1 \wedge X_2$
5. $(X_1 \wedge X_2) \vee (X_1 \wedge \overline{X_2}) = X_1$
6. $(X_1 \vee X_2) \wedge (X_1 \vee \overline{X_2}) = X_1$

2.4.7 Beispiel 1. Kürzungsregel

1. $X_1 \vee (X_1 \wedge X_2) = X_1$

$$
\begin{aligned}
X_1 \vee (X_1 \wedge X_2) &= (X_1 \vee X_1) \wedge (X_1 \vee X_2) \\
&= X_1 \wedge (X_1 \vee X_2) \\
&= X_1
\end{aligned}
$$

2.4.7 Beispiel 4. Kürzungsregel

X_1	X_2	$\overline{X_1} \vee X_2$	$X_1 \wedge (\overline{X_1} \vee X_2)$
0	0	1	0
0	1	1	0
1	0	0	0
1	1	1	1

$$X_1 \wedge (\overline{X_1} \vee X_2) =$$

$$X_1 \wedge X_2$$

2.4.7 Bindungsregel

$$Y = X_1 \vee X_2 \wedge X_3$$

Nicht eindeutig!

$$Y = (X_1 \vee X_2) \wedge X_3$$

$$Y = X_1 \vee (X_2 \wedge X_3)$$

Besser ist, Klammern zu setzen.

Allerdings gilt die Regel:

Eine UND-Verknüpfung bindet stärker als eine ODER-Verknüpfung

2.5 De Morgan'sche Gesetze

1. $\overline{(X_1 \wedge X_2)} = (\overline{X_1} \vee \overline{X_2})$

2. $\overline{(X_1 \vee X_2)} = (\overline{X_1} \wedge \overline{X_2})$

2.6 Dualitätsprinzip, Shannon'sche Gesetz

$$\overline{f(X_1, X_2, \ldots, X_n, \wedge, \vee)} = f(\overline{X_1}, \overline{X_2}, \ldots, \overline{X_n}, \vee, \wedge)$$

Der invertierte Wert einer Booleschen Funktion ist gleich dem Wert, die diese Funktion liefert, wenn alle Operanden und alle Operatoren **invertiert** werden.

2.2 Unäre Boolesche Funktionen

Gleichung	Bezeichnung
$Y = 0$	Konstante 0
$Y = \overline{X}$	Negation
$Y = X$	Identität
$Y = 1$	Konstante 1

2.2 Binäre Boolesche Funktionen (1)

Gleichung	Bezeichnung
$Y = 0$	Konstante 0
$Y = \overline{(X_1 \vee X_2)}$	NOR
$Y = (X_1 \wedge \overline{X_2})$	Inhibition
$Y = \overline{X_2}$	Negation (X_2)

2.2 Binäre Boolesche Funktionen (2)

Gleichung	Bezeichnung
$Y = (\overline{X_1} \wedge X_2)$	Inhibition
$Y = \overline{X_1}$	Negation (X_1)
$Y = (X_1 \wedge \overline{X_2}) \vee (\overline{X_1} \wedge X_2)$	XOR
$Y = \overline{(X_1 \wedge X_2)}$	Nand

2.2 Binäre Boolesche Funktionen (3)

Gleichung	Bezeichnung
$Y = (X_1 \wedge X_2)$	UND
$Y = (X_1 \wedge X_2) \vee (\overline{X_1} \wedge \overline{X_2})$	XNOR
$Y = X_1$	Identität (X_1)
$Y = X_1 \vee \overline{X_2}$	Implikation

2.2 Binäre Boolesche Funktionen (4)

Gleichung	Bezeichnung
$Y = X_2$	Identität (X_2)
$Y = \overline{X_1} \vee X_2$	Implikation
$Y = (X_1 \vee X_2)$	ODER
$Y = 1$	Konstante 1

Nächste Vorlesung

- Vollständige Systeme
- Normalformen