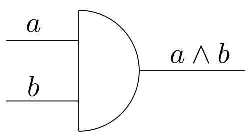
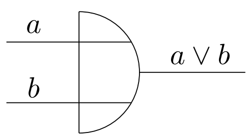
↑ Schaltelemente

Und-Gatter *Konjunktion* a und b



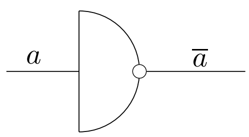
a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Oder-Gatter *Disjunktion* a oder b



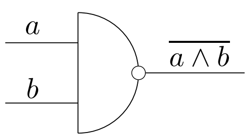
a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

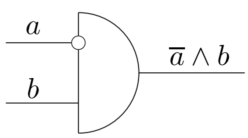
Negation nicht a

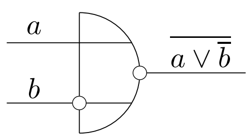


a	$\overline{a}$
0	1
1	0

Ein- und Ausgänge können beim Und- und Oder-Gatter negiert werden.

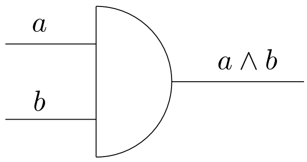


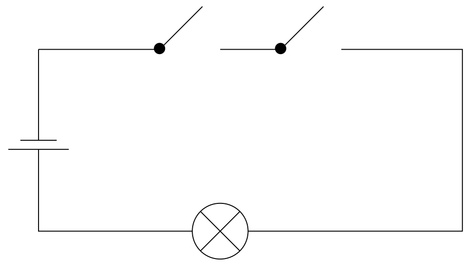




↑ Veranschaulichungen

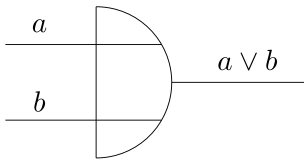
Und-Gatter

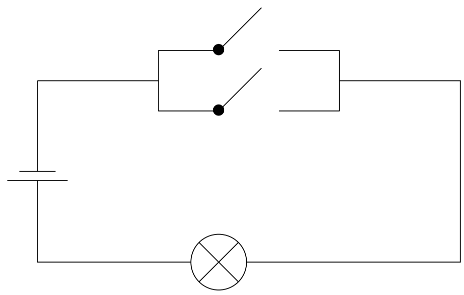




$a \wedge b$ ist genau dann 1, wenn alle Eingänge 1 sind.

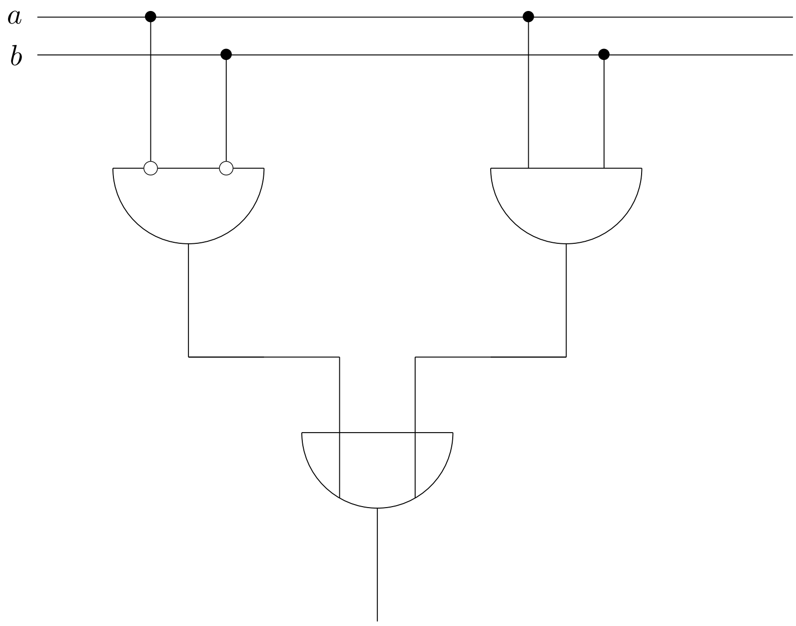
Oder-Gatter

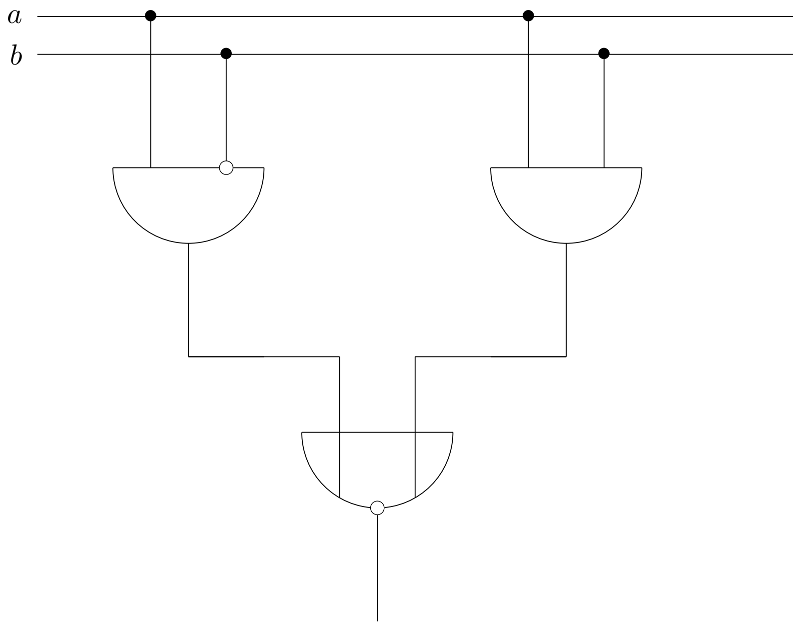




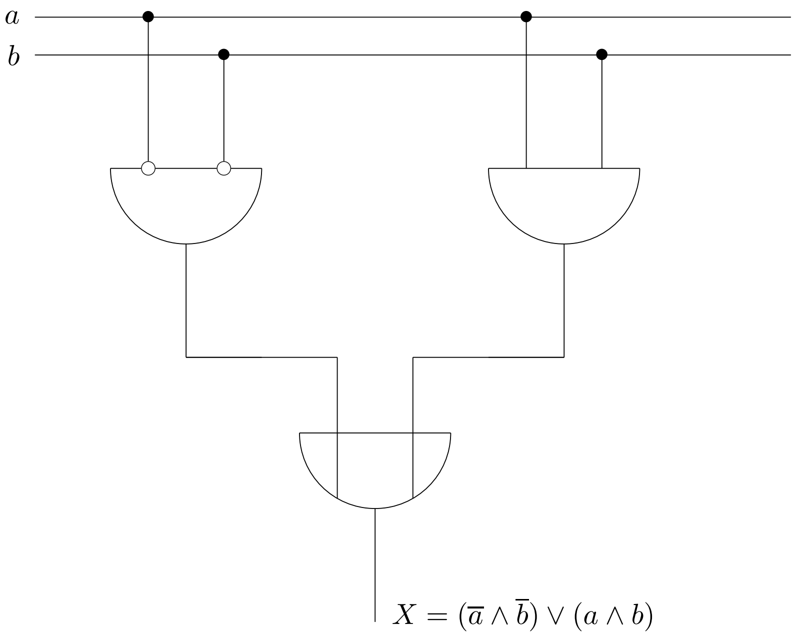
$a \vee b$ ist genau dann 1, wenn mindestens ein Eingang 1 ist.

↑ Ermittle den booleschen Term und erstelle die Wertetabelle für die boolesche Funktion.



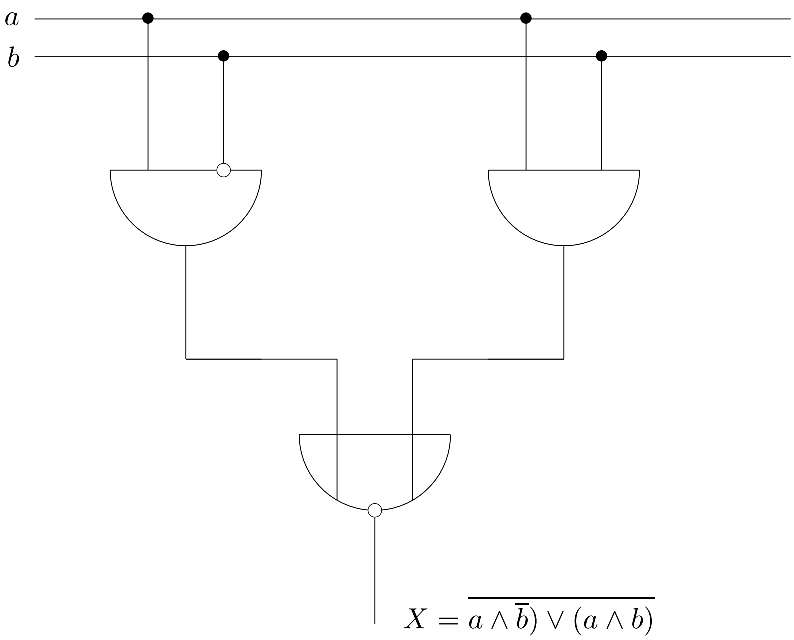


↑ Ermittle den booleschen Term und erstelle die Wertetabelle für die boolesche Funktion.



$X = (\overline{a} \wedge \overline{b}) \vee (a \wedge b)$

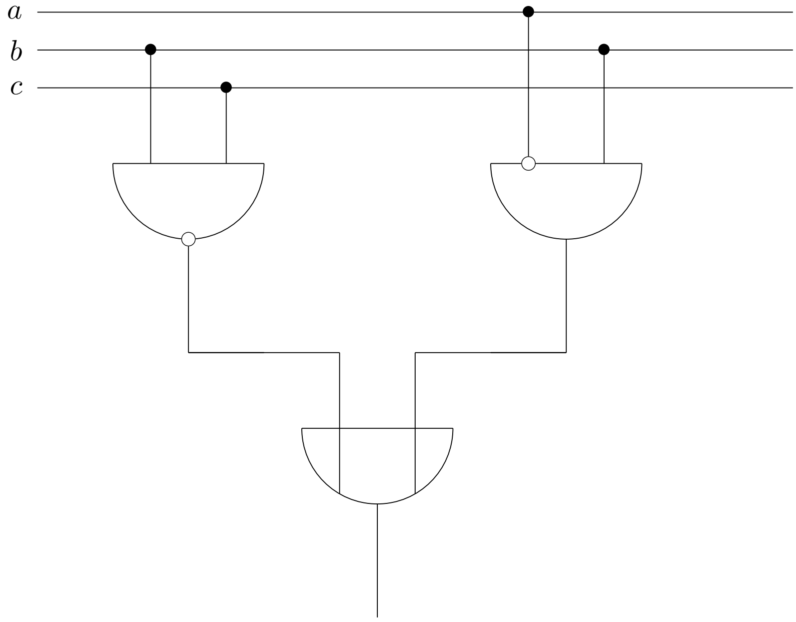
a	b	X
0	0	1
0	1	0
1	0	0
1	1	1

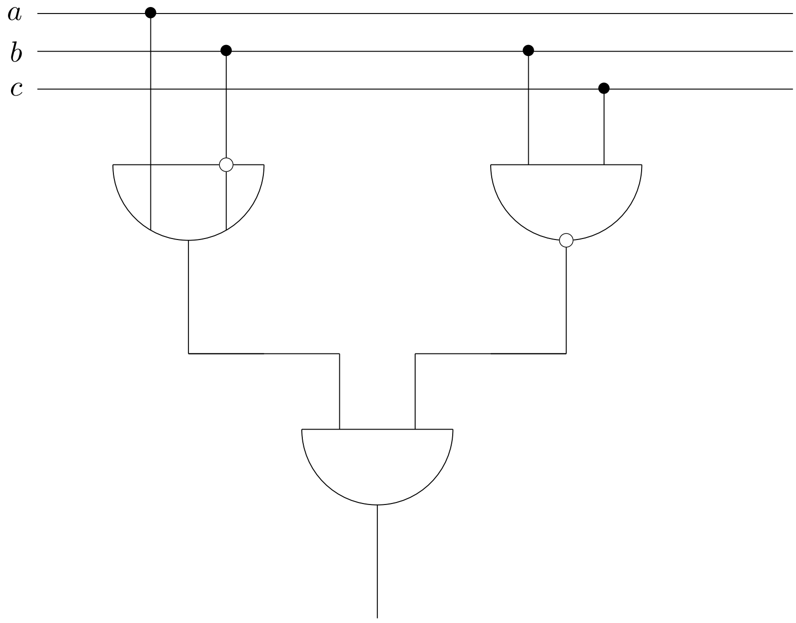


$X = \overline{a \wedge \overline{b}) \vee (a \wedge b)}$

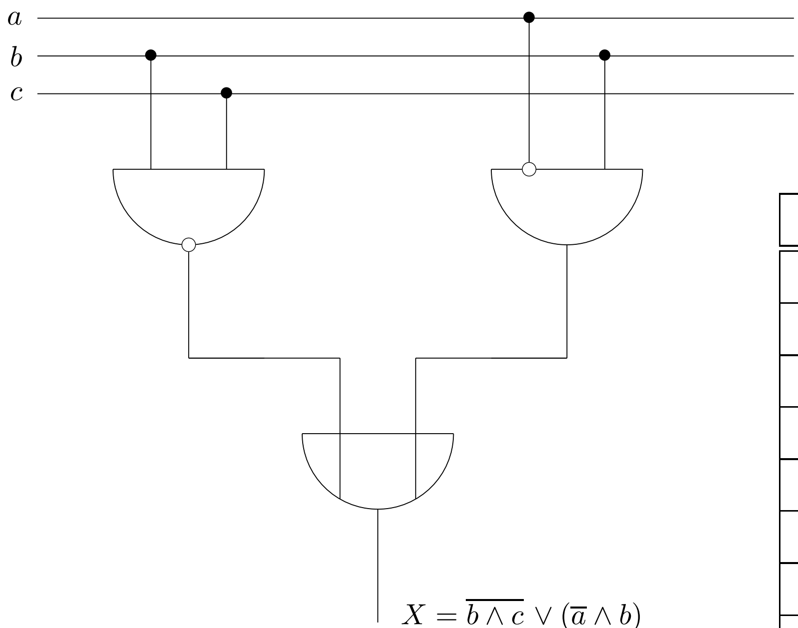
a	b	X
0	0	1
0	1	1
1	0	0
1	1	0

↑ Ermittle den booleschen Term und erstelle die Wertetabelle für die boolesche Funktion.



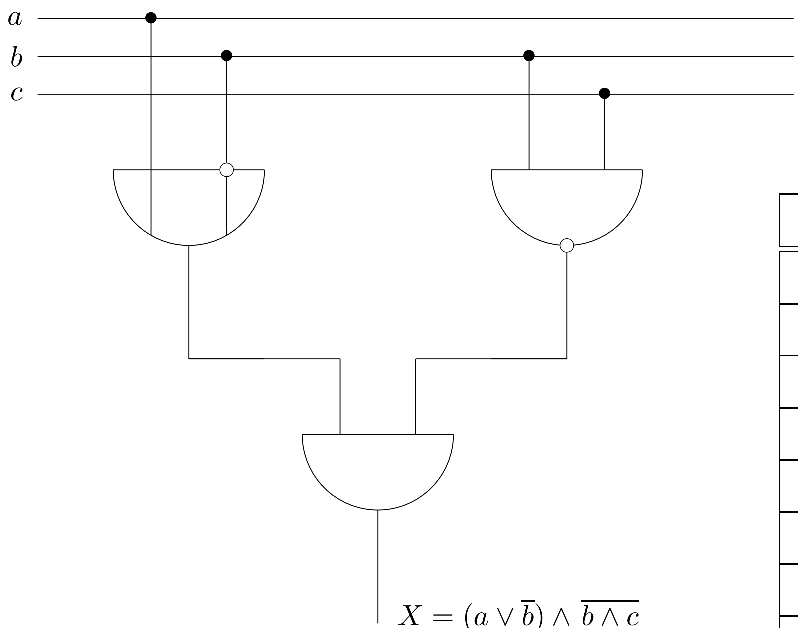


↑ Ermittle den booleschen Term und erstelle die Wertetabelle für die boolesche Funktion.



$$X = \overline{b \wedge c} \vee (\overline{a} \wedge b)$$

a	b	c	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$$X = (a \vee \overline{b}) \wedge \overline{b \wedge c}$$

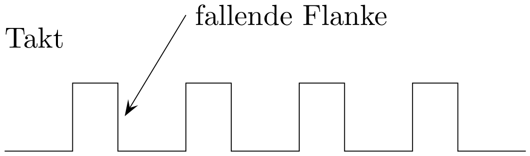
a	b	c	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

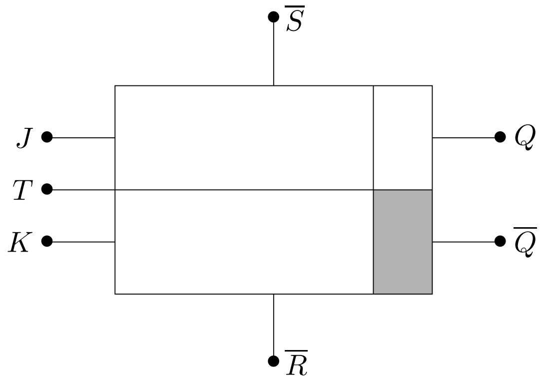
↑ Python

```
for a in [False,True]:
  for b in [False,True]:
      print(int(a), int(b), " ", int(a and not b))
```

```
for a in [False,True]:
  for b in [False,True]:
    for c in [False,True]:
      print(int(a), int(b), int(c), " ", int(a and b and not c))
```

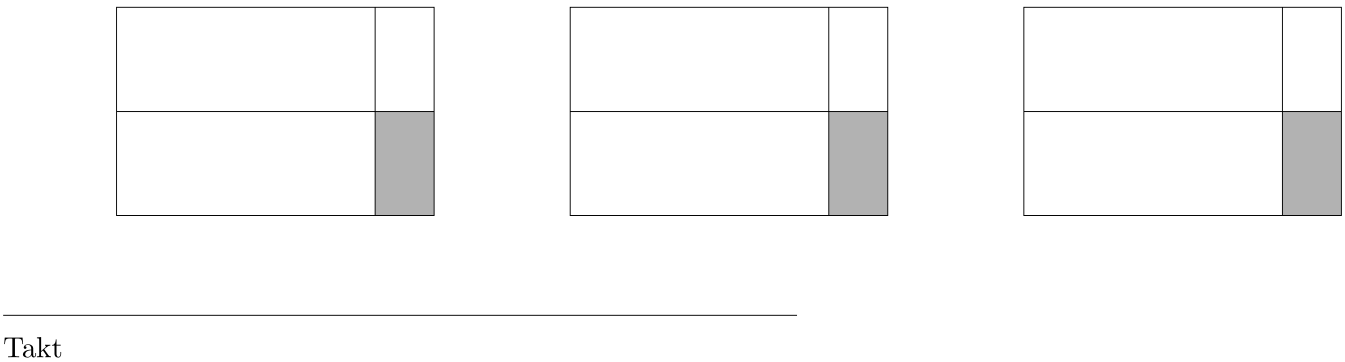
↑ Flip-Flop



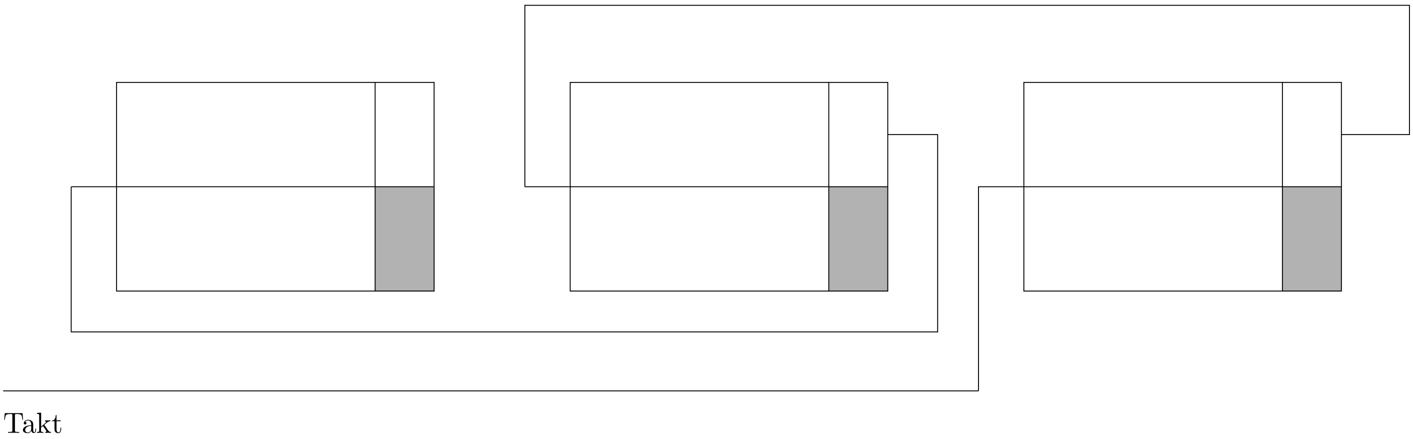


Das Flip-Flop wechselt bei jeder fallenden Taktflanke den Zustand,
wenn die Eingänge J, K offen (unbeschaltet) sind, es ist dann $J = K = 1$.
Für $J = K = 0$ wird der Zustand des Flip-Flops beibehalten.

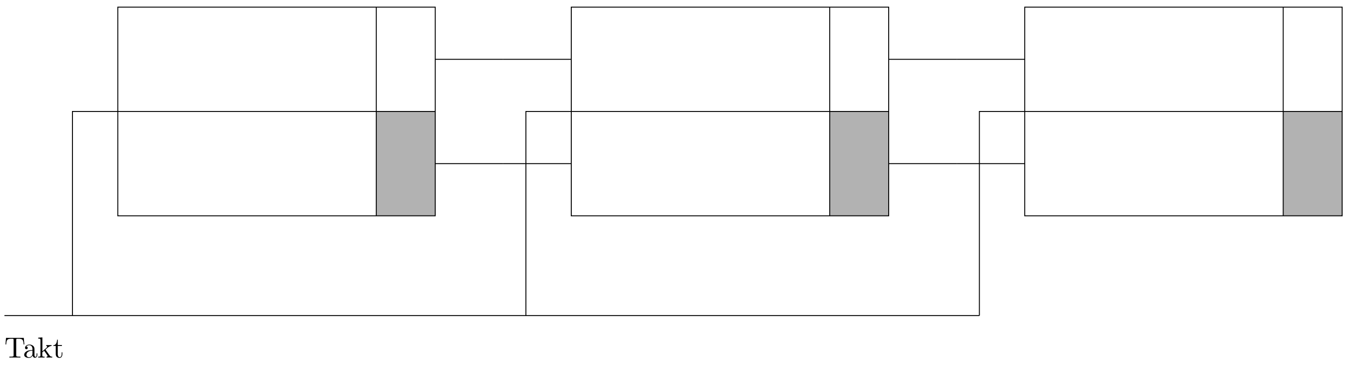
Dualzähler



↑ Dualzähler



↑ Schieberegister

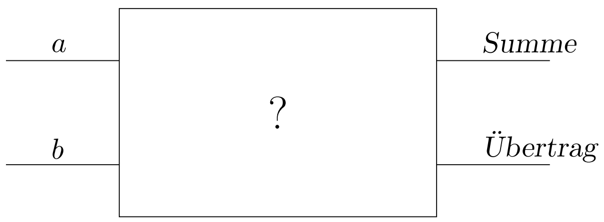


An offenen (unbeschalteten) Eingängen (siehe Dualzähler) liegt stets eine 1 an. Beim Schieberegister übernimmt bei jeder fallenden Taktflanke jedes Flip-Flop die gespeicherte Information (0 oder 1) seines linken Vorgängers.

Wie verhält sich das ganz links stehende Flip-Flop?

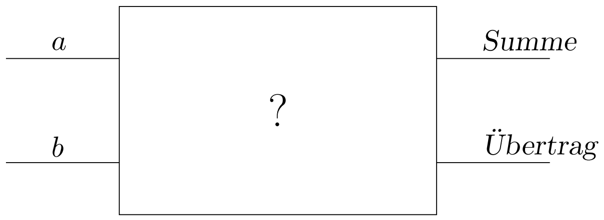
↑ Addierwerk

$$
\begin{array}{rcccccc}
 & & \mid & \mid & \bigcirc & \mid & \mid & = & 27 \\
+ & & \mid & \bigcirc & \bigcirc & \mid & \bigcirc & = & 18 \\
\hline
 & \mid & \bigcirc & \mid & \mid & \bigcirc & \mid & = & 45
\end{array}
$$



a	b	*Summe*	*Übertrag*
0	0		
0	1		
1	0		
1	1		

↑ Addierwerk



a	b	*Summe*	*Übertrag*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

↑ Disjunktive Normalform

Gib für die boolesche Funktion, deren Wertetabelle gegeben ist, einen Term an.

1.

a	b	X
0	0	0
0	1	1
1	0	0
1	1	0

a	b	Y
0	0	0
0	1	1
1	0	1
1	1	0

a	b	Z
0	0	1
0	1	1
1	0	0
1	1	1

2.

a	b	c	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

a	b	c	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

3.

a	b	c	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

a	b	c	W
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

↑ Disjunktive Normalform

Gib für die boolesche Funktion, deren Wertetabelle gegeben ist, einen Term an.

1.

a	b	X
0	0	0
0	1	1
1	0	0
1	1	0

$X = \overline{a} \wedge b$

a	b	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = (\overline{a} \wedge b) \vee (a \wedge \overline{b})$

a	b	Z
0	0	1
0	1	1
1	0	0
1	1	1

$Z = (\overline{a} \wedge \overline{b}) \vee (\overline{a} \wedge b) \vee (a \wedge b)$

2.

a	b	c	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$X = a \wedge \overline{b} \wedge c$

a	b	c	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$Y = (\overline{a} \wedge b \wedge \overline{c}) \vee (a \wedge \overline{b} \wedge c)$

3.

a	b	c	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$Z = (\overline{a} \wedge \overline{b} \wedge \overline{c}) \vee (\overline{a} \wedge \overline{b} \wedge c) \vee (a \wedge \overline{b} \wedge c)$

a	b	c	W
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$W = (\overline{a} \wedge \overline{b} \wedge c) \vee (\overline{a} \wedge b \wedge \overline{c}) \vee (\overline{a} \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge c)$

↑ Gesetze der booleschen Algebra

George Boole 1815-1864

Kommutativgesetze:	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$
Assoziativgesetze:	$(a \vee b) \vee c = a \vee (b \vee c)$	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$
Absorptionsgesetze:	$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$
Distributivgesetze:	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
Idempotenzgesetze:	$a \vee \overline{a} = 1$	$a \wedge \overline{a} = 0$
	$a \vee a = a$	$a \wedge a = a$
de Morgan-Gesetze:	$\overline{a \vee b} = \overline{a} \wedge \overline{b}$	$\overline{a \wedge b} = \overline{a} \vee \overline{b}$
neutrale Elemente:	$a \vee 0 = a$	$a \wedge 1 = a$
doppelte Negation:	$\overline{\overline{a}} = a$	

Vereinfache mit den Gesetzen der booleschen Algebra.

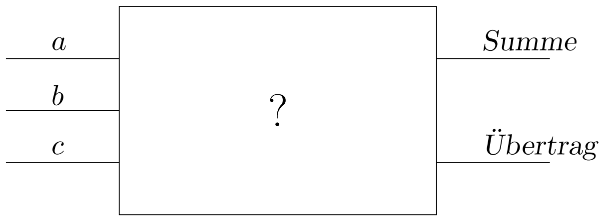
a) $(a \wedge b) \vee (a \wedge \overline{b})$

b) $\overline{a \vee b} \vee \overline{a}$

c) $\overline{\overline{a} \vee b} \vee (\overline{a} \wedge \overline{b})$

d) $\overline{\overline{a \vee b} \vee b}$

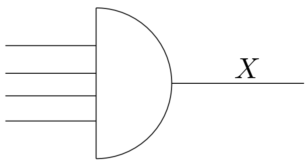
↑ Addierwerk

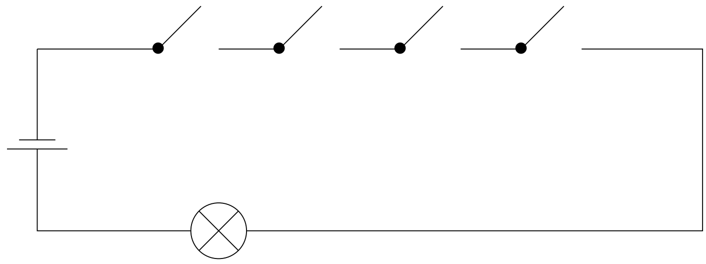


a	b	c	*Summe*	*Übertrag*
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

↑ Vereinfachungen

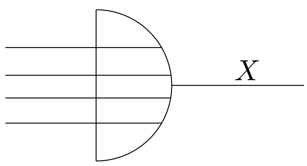
verallgemeinertes Und-Gatter

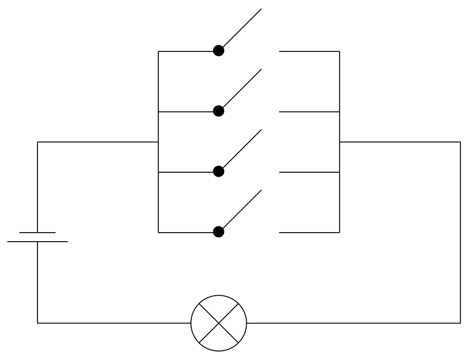




X ist genau dann 1, wenn alle Eingänge 1 sind.

verallgemeinertes Oder-Gatter



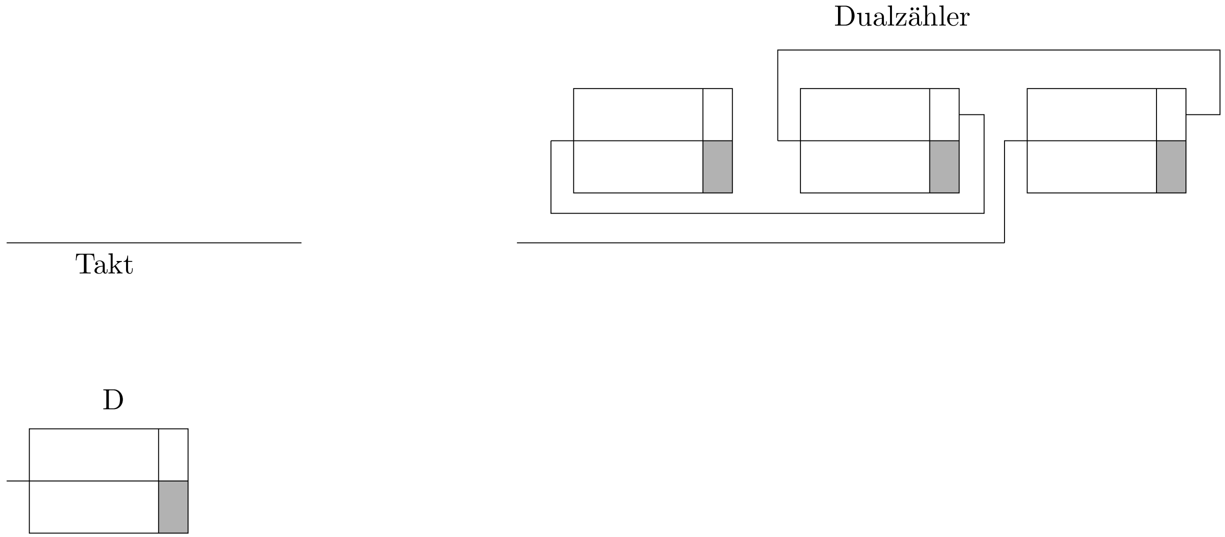


X ist genau dann 1, wenn mindestens ein Eingang 1 ist.

D steuert den Takt für den Dualzähler.

D = 1, Dualzähler erhält Takt.

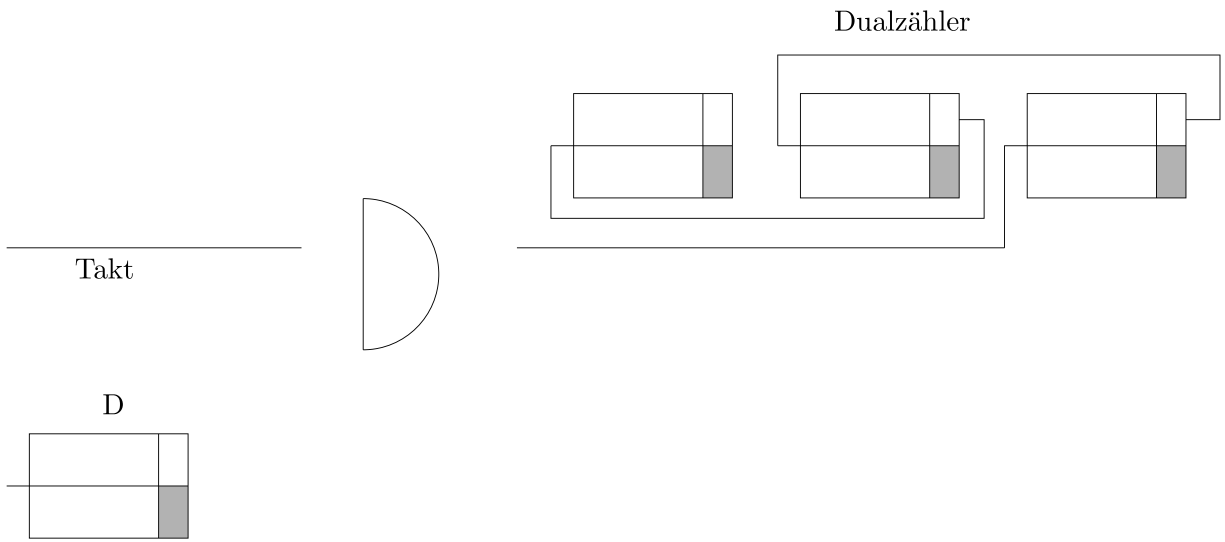
D = 0, Dualzähler erhält keinen Takt.



D steuert den Takt für den Dualzähler.

D = 1, Dualzähler erhält Takt.

D = 0, Dualzähler erhält keinen Takt.



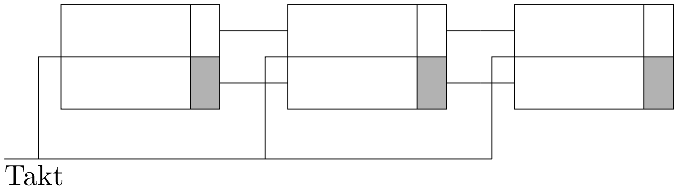
→ Steuerung Tipp

D steuert, welcher Inhalt (A oder B) nach C gelangt.

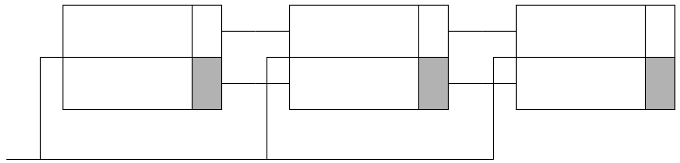
$D = 1, \;\; A \longrightarrow C$

$D = 0, \;\; B \longrightarrow C$

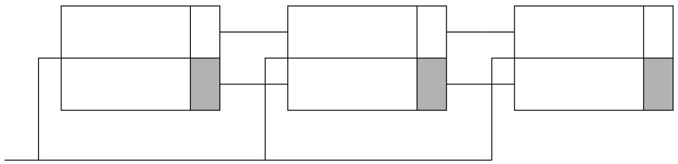
Register A



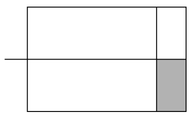
Register C



Register B



D

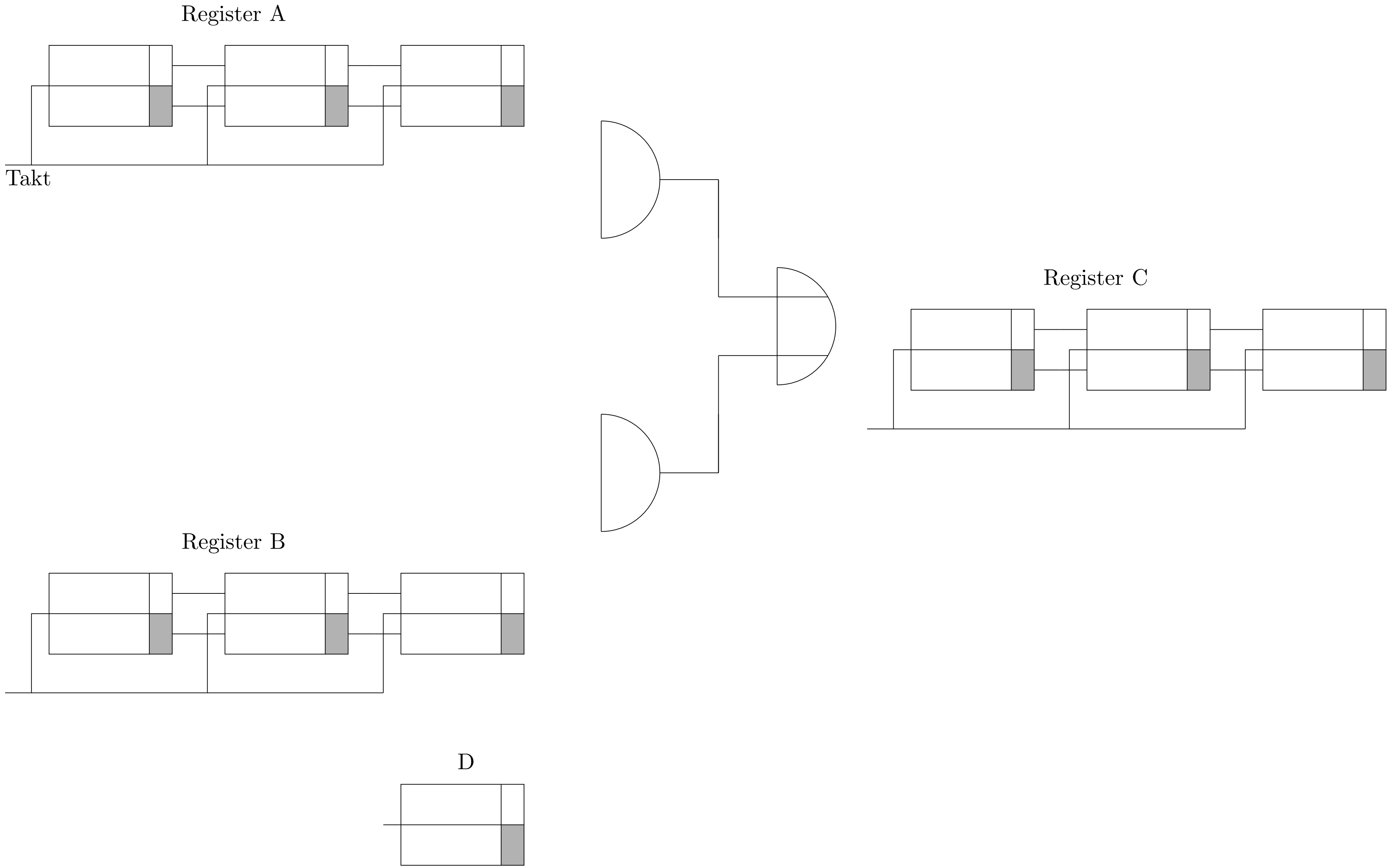


→ Steuerung

D steuert, welcher Inhalt (A oder B) nach C gelangt.

D = 1, A $\longrightarrow$ C

D = 0, B $\longrightarrow$ C



→ Steuerung Tipp

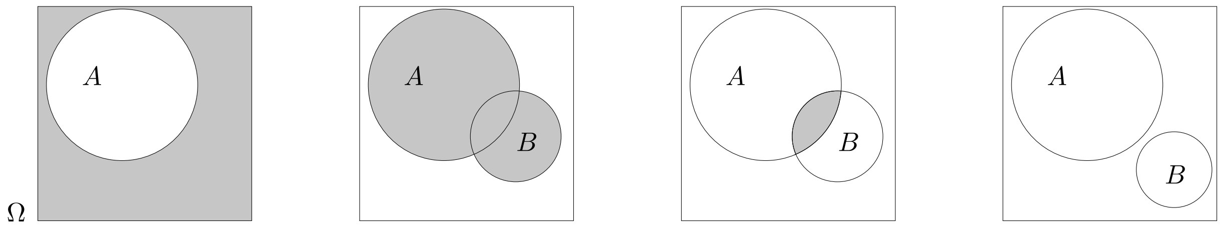
↑ Gesetze der Mengenalgebra

$\subset$ enthalten (ist Teilmenge)

$\cup$ vereinigt
$\cap$ geschnitten

$\emptyset$ leere Menge

1)	$A \cap B = B \cap A$ $A \cup B = B \cup A$	Kommutativgesetze
2)	$(A \cap B) \cap C = A \cap (B \cap C)$ $(A \cup B) \cup C = A \cup (B \cup C)$	Assoziativgesetze
3)	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributivgesetze
4)	$A \cap (A \cup B) = A$ $A \cup (A \cap B) = A$	Absorptionsgesetze
5)	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan-Gesetze
6)	$A = \overline{\overline{A}}$ $A \cap \overline{A} = \emptyset$ $A \cup \overline{A} = \Omega$	Komplementgesetze



$\overline{A}$ $\quad$ $A \cup B$ $\quad$ $A \cap B$ $\quad$ $A \cap B = \emptyset$

Ω, ω letzter Buchstabe des griechischen Alphabets O,o
„Das A und O“
Alpha-Tier, Omega-Tier (auf dem letzten Platz der Rangordnung)

↑ Aussagenlogik

Eine Aussage ist entweder wahr (1) oder falsch (0).
Wir bezeichnen Aussagen mit großen Buchstaben.
Aussagen können verknüpft werden mit

und	$A \wedge B$
oder	$A \vee B$
wenn ..., dann ...	$A \rightarrow B$
... genau dann, wenn ...	$A \leftrightarrow B$

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

$A \rightarrow B$ mache man sich am Beispiel klar:
Wenn Frau L. Musik hört, benutzt sie einen Kopfhörer.
Diese Aussage ist nur falsch, wenn ....

Für Umformungen kann $A \rightarrow B = \overline{A} \vee B$ nützlich sein.
$\overline{A} \vee B$ ist nur 0, wenn $A = 1$ und $B = 0$ vorliegt.

Die Gesetze der Booleschen Algebra können neu formuliert werden.
Hierzu wird das Gleichheitszeichen durch das Äquivalenzzeichen ersetzt, z. B. $\overline{A \vee B} \leftrightarrow \overline{A} \wedge \overline{B}$.

Eine Aussage, die stets wahr ist, heißt Tautologie, in Zeichen: $\Rightarrow$ (Implikation), $\Leftrightarrow$ (Äquivalenz).
Wir schreiben also genauer:

$\overline{A \vee B} \Leftrightarrow \overline{A} \wedge \overline{B}$ gemeint: $\overline{A \vee B} \Leftrightarrow (\overline{A} \wedge \overline{B})$, also stets $(\ \dots\ ) \Leftrightarrow (\ \dots\ )$
$(\ \dots\ ) \Rightarrow (\ \dots\ )$

Weiteres

$(A \rightarrow B) \wedge (B \rightarrow C) \Rightarrow (A \rightarrow C)$

$A \wedge (A \rightarrow B) \Rightarrow B$

$A \rightarrow B \Leftrightarrow \overline{B} \rightarrow \overline{A}$ Kontraposition

$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$

$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (\overline{A} \rightarrow \overline{B})$

Überprüfe

$\overline{B} \wedge (A \rightarrow B) \Rightarrow \overline{A}$

$(A \rightarrow B) \wedge (\overline{A} \rightarrow B) \Leftrightarrow B$

$\overline{A \leftrightarrow B} \Leftrightarrow A \leftrightarrow \overline{B}$

↑ Logisches Schließen

Für welche Variablenbelegung ist alles wahr?   Sei hier $\overline{A} = \neg A$.

Wenn A redet, schweigt B.
Wenn C redet, dann auch B.
C redet.

---

a)
$$\begin{array}{l} A \rightarrow \neg B \\ C \rightarrow B \\ C \\ \hline \end{array}$$

b)
$$\begin{array}{l} \neg A \leftrightarrow B \\ B \rightarrow C \\ \neg C \\ \hline \end{array}$$

c)
$$\begin{array}{l} \neg C \leftrightarrow D \\ B \rightarrow C \wedge A \\ B \\ \hline \end{array}$$

d)
$$\begin{array}{l} \neg C \leftrightarrow A \\ B \rightarrow C \vee A \\ A \wedge B \\ \hline \end{array}$$

↑ Logisches Schließen

Für welche Variablenbelegung ist alles wahr? Sei hier $\overline{A} = \neg A$.

Wenn A redet, schweigt B.
Wenn C redet, dann auch B.
C redet.

---

a)
$$\begin{array}{l} A \;\rightarrow\; \neg B \\ C \;\rightarrow\; B \\ \underline{C \qquad\qquad} \end{array}$$

$A = 0,\;\; B = 1,\;\; C = 1$

b)
$$\begin{array}{l} \neg A \;\leftrightarrow\; B \\ \;\; B \;\rightarrow\; C \\ \underline{\neg C \qquad\qquad} \end{array}$$

$A = 1,\;\; B = 0,\;\; C = 0$

c)
$$\begin{array}{l} \neg C \;\leftrightarrow\; D \\ \;\; B \;\rightarrow\; C \wedge A \\ \underline{\;\; B \qquad\qquad\quad} \end{array}$$

$A = 1,\;\; B = 1,\;\; C = 1,\;\; D = 0$

d)
$$\begin{array}{l} \quad \neg C \;\leftrightarrow\; A \\ \qquad B \;\rightarrow\; C \vee A \\ \underline{A \wedge B \qquad\qquad\quad} \end{array}$$

$A = 1,\;\; B = 1,\;\; C = 0$

↑ Quantoren

„Es gibt (mindestens) ein n_0 aus $\mathbb{N}$, für das die Aussage $A(n)$ wahr ist.“
kann mit dem Existenzquantor kurz formuliert werden: $\underset{n_0 \in \mathbb{N}}{\exists} A(n)$

„Für alle n aus $\mathbb{N}$ ist die Aussage $A(n)$ wahr.“ wird abgekürzt mit (Allquantor): $\underset{n \in \mathbb{N}}{\forall} A(n)$

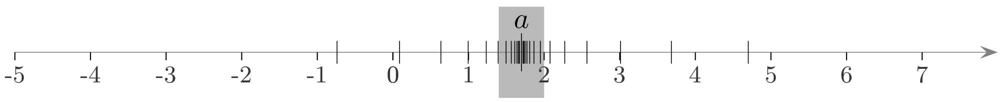
Die Negation kann leicht gebildet werden:

$$\neg \Big[\underset{n_0 \in \mathbb{N}}{\exists} A(n) \Big] \iff \underset{n \in \mathbb{N}}{\forall} \neg A(n) \qquad \text{und} \qquad \neg \Big[\underset{n \in \mathbb{N}}{\forall} A(n) \Big] \iff \underset{n_0 \in \mathbb{N}}{\exists} \neg A(n)$$

Die Folge $(a_n)_{n \in \mathbb{N}}$ konvergiert (strebt gegen) den Grenzwert a, $\lim\limits_{n \to \infty} a_n = a$, falls gilt:

$$\underset{\varepsilon > 0}{\forall} \; \underset{n_0 \in \mathbb{N}}{\exists} \; \underset{n \geq n_0}{\forall} \; |\, a_n - a \,| < \varepsilon$$

In Worten:
Zu jedem $\varepsilon > 0$ existiert eine natürliche Zahl n_0 (ein Index), so dass für alle nachfolgenden Indizes gilt $|\, a_n - a \,| < \varepsilon$, d. h. die Glieder a_n unterscheiden sich für $n > n_0$ von a um weniger als ε.
Anschaulich: Bei der Berechnung der Folgenglieder einer konvergenten Folge (mit einem Computer) bleiben immer mehr Dezimalstellen stabil.



Die Negation, d. h. die Folge konvergiert nicht gegen den Wert a, lautet:

$$\neg \underset{\varepsilon > 0}{\forall} \; \underset{n_0 \in \mathbb{N}}{\exists} \; \underset{n \geq n_0}{\forall} \; |\, a_n - a \,| < \varepsilon \iff$$

$$\underset{\varepsilon_0 > 0}{\exists} \; \underset{n \in \mathbb{N}}{\forall} \; \underset{m \geq n}{\exists} \; |\, a_m - a \,| \geq \varepsilon_0$$

Das Negationszeichen wandert von links nach rechts und kehrt die Quantoren um.

Formuliere die negierte Aussage.

a) Auf jeden Topf passt ein Deckel.

b) Heute morgen erschienen alle Lehrer pünktlich zum Unterricht.

c) Alle Schüler der 8a kennen alle englischen Vokabeln der 5. Lektion.

d) Ein Schüler der 10c hat alle gestellten Mathematik-Aufgaben gelöst.

Die konträre Aussage zu a) lautet:
Heute morgen erschien kein Lehrer pünktlich zum Unterricht.
Beides, a) und konträre Aussage, kann falsch sein.
Daher ist die konträre Aussage keine Negation.
Die Negation einer wahren Aussage ist falsch, und umgekehrt.

Bei Aussagen wie:

Alle Deutschen ... .

Alle Engländer ... .

Alle Türken ... .

handelt es sich meistens um statistische Aussagen (oder Vorurteile).

Die Negation ist dann nicht:

Es gibt einen Deutschen ... .

Es gibt einen Engländer ... .

Es gibt einen Türken ... .

Statistische Aussagen sollten einen (prozentualen) Anteilswert enthalten,
damit eine Negation möglich ist.

Höchstens 15 % der Deutschen ... .

Mindestens 90 % der Engländer ... .

Mindestens 60 % der Türken ... .

↑ Klausurthemen Informatik

a) boolescher Term / Schaltung / Wertetabelle

b) Darstellung dual / dezimal

c) boolesche Terme mit den Rechenregeln vereinfachen

d) disjunktive Normalform aufstellen (Tabelle gegeben), möglichst einfachen Term ermitteln

e) Aufgaben zur Logik, Umkehrung, Kontraposition bilden, Untersuchung wahr/falsch, Variablenbelegung für wahre Aussage ermitteln, Tautologie nachweisen, logisch Schließen,
siehe auch Logik
...

1. Zeige mit den Gesetzen der Booleschen Algebra die Gleichheit der Terme.

   a) $(a \wedge \overline{b}) \vee (a \wedge \overline{b} \wedge c) = a \wedge \overline{b}$

   b) $(a \vee b) \wedge (\overline{a} \vee b) \wedge (a \vee \overline{b}) \wedge (\overline{a} \vee \overline{b}) = 0$

   c) $(\overline{a \wedge b} \vee \overline{c}) \wedge (\overline{a} \vee b \vee \overline{c}) = \overline{a} \vee \overline{c}$

   d) $\overline{\overline{a \wedge b} \vee c} \vee (a \wedge c) = a \wedge (b \vee c)$

2. Gib für die booleschen Funktionen möglichst einfache Terme an.

			a)	b)	c)	d)
a	b	c	X	Y	Z	U
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	1	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	1
1	1	1	1	0	1	1

   a) $X = (\overline{a} \wedge b) \vee (a \wedge c)$

   b) $Y = \overline{b} \vee (a \wedge \overline{c})$

   c) $Z = (\overline{a} \wedge \overline{b}) \vee (a \wedge c)$

   d) $U = a \wedge b$

↑ Einladungsproblem

Wegen bestehender Freund- und Feindschaften muss Birgit eingeladen werden, wenn man Anne einlädt. Diese muss genau dann eingeladen werden, wenn man Claus einlädt. Es darf aber höchstens Birgit oder Claus eingeladen werden.

↑ Einladungsproblem

Wegen bestehender Freund- und Feindschaften muss Birgit eingeladen werden, wenn man Anne einlädt. Diese muss genau dann eingeladen werden, wenn man Claus einlädt. Es darf aber höchstens Birgit oder Claus eingeladen werden.

Wir formulieren diese Aussagen.

$$\begin{array}{c} A \to B \\ A \leftrightarrow C \\ \neg B \lor \neg C \\ \hline \end{array}$$

$(A \to B) \land (A \leftrightarrow C) \land (\neg B \lor \neg C)$

```
def folgt(X,Y):
    return  not X or Y


def äqui(X,Y):
    return  X == Y          # alternativ  folgt(X,Y) and folgt(Y,X)



for A in [False,True]:
    for B in [False,True]:
        for C in [False,True]:
            D1 = ...
            D2 = ...
            D3 = ...
            print(int(A), int(B), int(C), "  ", int(D1 and D2 and D3))
```