2.1 Grundgatter und Belegungstabellen

Leitidee

Ein Computer kennt in seinem Innersten nur zwei Zustände: Auf einer Leitung liegt Spannung an, oder es liegt keine Spannung an. Man schreibt dafür **1** und **0**. Trotzdem kann ein Computer rechnen, vergleichen und Entscheidungen treffen. Möglich machen das winzige elektronische Bausteine, die **Gatter**. Jedes Gatter bekommt einen oder zwei Werte 0 oder 1 und liefert daraus nach einer festen Regel wieder eine 0 oder eine 1. Der amerikanische Mathematiker und Ingenieur Claude Shannon hat gezeigt, dass man mit solchen Schaltungen logisch denken und sogar rechnen kann. In diesem Abschnitt lernst du die vier Grundgatter kennen. In den nächsten Abschnitten baust du daraus eine Alarmanlage und ein Rechenwerk.

Lernziele

Nach diesem Abschnitt kannst du

- die vier Grundgatter UND (AND), ODER (OR), XOR und NICHT (NOT) nennen und mit ihren Symbolen skizzieren,
- zu jedem Grundgatter die Belegungstabelle angeben,
- begründen, warum es bei n Eingängen 2^n Belegungen gibt,
- zu einer Schaltung aus mehreren Gattern die Belegungstabelle mit Hilfsspalten erstellen,
- Gatter im Simulator CircuitVerse aufbauen und testen,
- Belegungstabellen mit Python nachprüfen.

2.1.1 Nur zwei Zustände: 0 und 1

Denk an eine Taschenlampe. Der Schalter ist entweder aus oder an, die Lampe ist entweder dunkel oder hell. Einen dritten Zustand gibt es nicht. Genau so arbeitet die Elektronik in einem Computer. Jede Leitung hat zu jedem Zeitpunkt einen von zwei Werten: **0** (keine Spannung, „aus“, „falsch“) oder **1** (Spannung, „an“, „wahr“). Einen solchen Wert, der nur 0 oder 1 sein kann, nennt man ein **Bit**.

Eine **logische Schaltung** wandelt Eingangswerte in Ausgangswerte um. In den Zeichnungen gehen die Leitungen für die **Eingänge** links hinein, die Leitungen für die **Ausgänge** kommen rechts heraus. Die Eingänge bezeichnen wir mit Großbuchstaben wie A, B, C, den Ausgang oft mit Q.

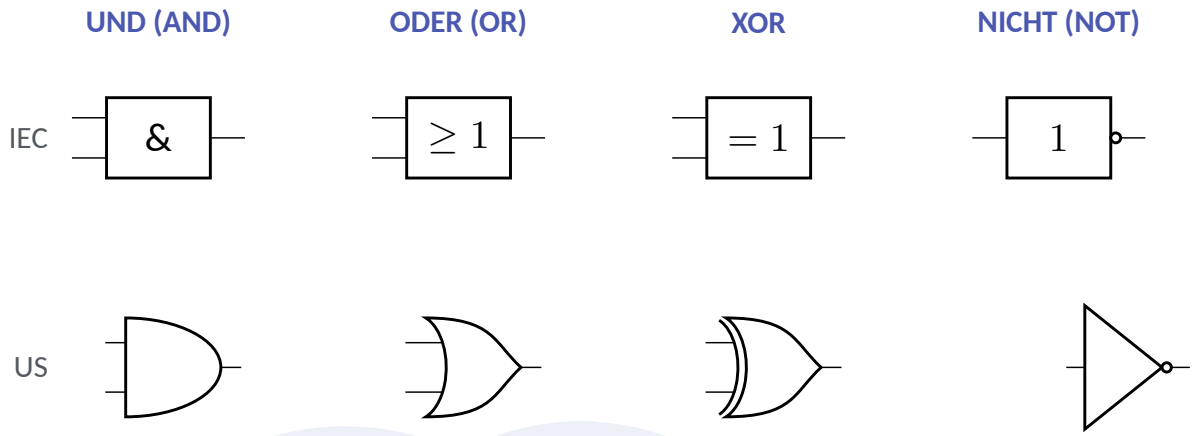
Die kleinsten Bausteine einer logischen Schaltung heißen **Gatter**. Ein Gatter ist ein elektronisches Bauteil mit einem oder mehreren Eingängen und einem Ausgang. Welcher Wert am Ausgang herauskommt, hängt nur davon ab, welche Werte an den Eingängen anliegen. Vier Gatter sind so wichtig, dass man sie **Grundgatter** nennt: UND, ODER, XOR und NICHT. Oft verwendet man die englischen Namen AND, OR, XOR und NOT.

Definition: Grundgatter

Ein **Gatter** ist ein Baustein einer logischen Schaltung, der die Werte 0 oder 1 an seinen Eingängen nach einer festen Regel in einen Ausgangswert 0 oder 1 umwandelt. Die **Grundgatter** sind UND (AND), ODER (OR), XOR und NICHT (NOT).

So sehen Gatter aus

In deutschen Schulbüchern zeichnet man Gatter nach der Norm **IEC 60617**: Jedes Gatter ist ein Rechteck. Im Rechteck steht ein Kennzeichen, das die Regel des Gatters verrät, zum Beispiel & für UND. In diesem Skript werden alle Schaltungen so gezeichnet. Die Abbildung zeigt in der oberen Reihe die vier Grundgatter in dieser Form. Was die Kennzeichen bedeuten, erfährst du gleich bei den einzelnen Gattern.



Die vier Grundgatter: oben die IEC-Symbole (so zeichnest du sie im Heft und in Klassenarbeiten), unten die US-Symbole, die der Simulator CircuitVerse zeigt

Tipp: CircuitVerse zeigt andere Symbole

Zum Ausprobieren benutzt du den kostenlosen Online-Simulator **CircuitVerse**. Du findest ihn unter der Adresse `circuitverse.org/simulator`. CircuitVerse zeichnet die Gatter nicht nach IEC, sondern mit den in den USA üblichen Symbolen (Norm ANSI/IEEE): UND hat die Form eines „D“, ODER ist vorn spitz, XOR sieht aus wie ODER mit einem zusätzlichen Bogen, NICHT ist ein Dreieck mit einem kleinen Kreis an der Spitze (untere Reihe in der Abbildung).

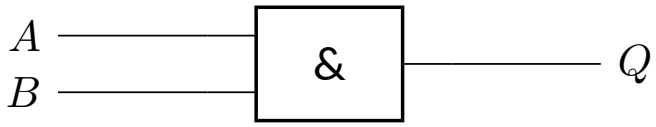
Beide Schreibweisen bedeuten **genau dasselbe**, nur die Zeichnung ist anders. In CircuitVerse geht es nur um das Verhalten der Schaltung. Im Heft und in Klassenarbeiten verwendest du die IEC-Symbole.

2.1.2 Das UND-Gatter (AND)

In einer Werkstatt steht eine Papierschneidemaschine. Damit niemand mit der Hand unter das Messer gerät, gibt es zwei Starttasten, eine links und eine rechts. Das Messer fährt nur herunter, wenn die linke Taste **und** die rechte Taste gedrückt sind. Ist auch nur eine Taste losgelassen, passiert nichts.

Genau diese Regel erfüllt das **UND-Gatter**: Sein Ausgang ist nur dann 1, wenn **alle** Eingänge 1 sind. Das Kennzeichen im Rechteck ist &, das Zeichen für „und“.

Mit zwei Eingängen gibt es genau vier Möglichkeiten, die Eingänge mit 0 und 1 zu **belegen**. Für jede dieser Belegungen schreibt man in einer Tabelle auf, welcher Wert am Ausgang herauskommt (rechts neben dem Symbol):



UND-Gatter: Die Eingänge A und B gehen links hinein, rechts kommt der Ausgang Q heraus

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Jede Zeile beschreibt einen Fall. Die dritte Zeile besagt zum Beispiel: Liegt an A eine 1 und an B eine 0 an, dann liefert das UND-Gatter eine 0. Nur in der letzten Zeile sind beide Eingänge 1, nur dort ist der Ausgang 1.

Über der Ausgangsspalte steht $A \wedge B$ (gelesen: „A und B“). Das Zeichen $\wedge$ ist die mathematische Schreibweise für UND. Ein solcher Ausdruck heißt **logischer Term**. Er beschreibt, was die Schaltung mit den Eingängen macht. Die UND-Verknüpfung heißt in der Logik auch **Konjunktion**.

Belegungstabellen

Eine Tabelle wie die oben heißt **Belegungstabelle** (manchmal auch Wahrheitstabelle). Sie ist das wichtigste Werkzeug in diesem Kapitel, denn sie beschreibt das Verhalten einer Schaltung vollständig.

Definition: Belegungstabelle

Eine **Belegungstabelle** (Wahrheitstabelle) listet für jede mögliche Kombination von Eingangswerten den zugehörigen Ausgangswert auf.

Wie viele Zeilen braucht man? Jeder Eingang kann 0 oder 1 sein, hat also 2 Möglichkeiten. Bei einem Eingang gibt es 2 Zeilen. Bei zwei Eingängen kann man jede der 2 Möglichkeiten von A mit jeder der 2 Möglichkeiten von B kombinieren: $2 \cdot 2 = 4$ Zeilen. Jeder weitere Eingang verdoppelt die Anzahl:

Anzahl der Eingänge n	**1**	**2**	**3**	**4**	**5**
Anzahl der Belegungen 2^n	2	4	8	16	32

Damit du keinen Fall vergisst, schreibst du die Zeilen immer in derselben Reihenfolge auf. Fange mit lauter Nullen an. In der **letzten** Eingangsspalte wechseln sich 0 und 1 in jeder Zeile ab. In der Spalte davor wechseln sie alle 2 Zeilen, davor alle 4 Zeilen und so weiter. Die erste Spalte besteht also aus einer Hälfte Nullen und einer Hälfte Einsen. Die letzte Zeile besteht aus lauter Einsen.

Merke

- Eine **Belegungstabelle** (Wahrheitstabelle) listet für jede mögliche Kombination von Eingangswerten den zugehörigen Ausgangswert auf.
- Bei n Eingängen gibt es 2^n mögliche Belegungen.

Arbeitsauftrag 1: Zeilen zählen

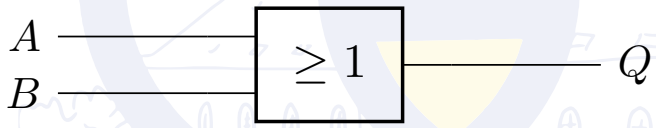
1. Wie viele Zeilen hat die Belegungstabelle einer Schaltung mit 3 Eingängen, wie viele bei 6 Eingängen?
2. Schreibe die Eingangsspalten A, B, C einer Belegungstabelle mit 3 Eingängen in der richtigen Reihenfolge auf.

Deine Antwort:

2.1.3 Das ODER-Gatter (OR)

Im Auto geht die Innenbeleuchtung an, wenn die Fahrertür geöffnet ist **oder** die Beifahrertür geöffnet ist. Sie leuchtet natürlich auch, wenn beide Türen offen sind. Nur wenn beide Türen geschlossen sind, bleibt es dunkel.

Das ist die Regel des **ODER-Gatters**: Sein Ausgang ist 1, wenn **mindestens ein** Eingang 1 ist. Daher kommt das Kennzeichen ≥1 („größer oder gleich 1“): Mindestens eine 1 muss an den Eingängen anliegen. Überlege zuerst selbst, wie die Belegungstabelle aussieht. Lies dann weiter.



ODER-Gatter: Das Kennzeichen ≥ 1 bedeutet „mindestens ein Eingang ist 1“

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Das Zeichen $\vee$ steht für ODER; $A \vee B$ liest man „A oder B“. Die ODER-Verknüpfung heißt in der Logik auch **Disjunktion**.

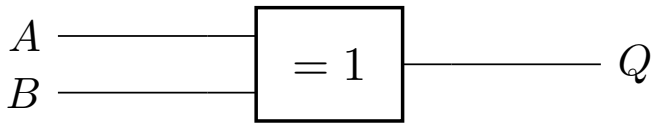
Stolperstein

Im Alltag bedeutet „oder“ oft „das eine oder das andere, aber nicht beides“: „Willst du Tee oder Kaffee?“ Das logische ODER ist anders. Es liefert auch dann eine 1, wenn **beide** Eingänge 1 sind (letzte Zeile der Tabelle). Wenn du in einer Aufgabe „oder“ liest, überlege genau, ob „beides“ erlaubt sein soll.

2.1.4 Das XOR-Gatter

Für das „entweder … oder“ gibt es ein eigenes Gatter: das **XOR-Gatter**. XOR ist die Abkürzung für *exclusive or*, auf Deutsch „ausschließendes Oder“ oder **Exklusiv-Oder**. Sein Ausgang ist 1, wenn die beiden Eingänge **unterschiedlich** sind, also wenn genau einer der beiden Eingänge 1 ist. Das Kennzeichen =1 bedeutet: Es muss **genau eine** 1 anliegen.

Ein Beispiel aus dem Alltag ist die Flurbeleuchtung mit zwei Schaltern, einem an jedem Ende des Flurs. Man stellt die Schaltung so ein, dass das Licht aus ist, wenn beide Schalter unten stehen. Kippt man einen der beiden Schalter um, geht das Licht an. Kippt man danach auch den anderen um, geht es wieder aus. Das Licht brennt also genau dann, wenn die beiden Schalter unterschiedlich stehen.



XOR-Gatter: Das Kennzeichen $= 1$ bedeutet „genau ein Eingang ist 1“

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Das Zeichen $\oplus$ steht für XOR. Vergleiche die Tabelle mit der Tabelle des ODER-Gatters: Nur die letzte Zeile ist anders. Wenn beide Eingänge 1 sind, liefert ODER eine 1, XOR aber eine 0.

2.1.5 Das NICHT-Gatter (NOT)

Das **NICHT-Gatter** hat nur **einen** Eingang. Es dreht den Wert einfach um: Aus 0 wird 1, aus 1 wird 0. Sein Ausgang ist also immer das **Gegenteil** des Eingangs. Man nennt das auch **Negation** (Verneinung). Ein Beispiel ist das Licht im Kühlschrank: Es ist an, wenn die Tür **nicht** geschlossen ist.

Im IEC-Symbol steht eine 1 im Rechteck, denn das Gatter gibt den einen Eingangswert weiter. Dass der Wert umgedreht wird, zeigt ein kleiner **Kreis** direkt am Ausgang. In manchen Büchern steht an seiner Stelle ein kleiner Schrägstrich. Beide Zeichen bedeuten dasselbe: Der Wert wird verneint.



NICHT-Gatter: Der kleine Kreis am Ausgang zeigt, dass der Eingangswert umgedreht wird

A	$\neg A$
0	1
1	0

Das Zeichen $\neg$ steht für NICHT; $\neg A$ liest man „nicht A“.

2.1.6 Die Grundgatter im Überblick

Die folgende Tabelle fasst die vier Grundgatter zusammen. Lerne sie auswendig. Du brauchst sie in allen weiteren Abschnitten dieses Kapitels.

Merke

Gatter	IEC-Symbol	Verhalten
UND (AND, Konjunktion)	&	Ausgang ist 1, wenn **alle** Eingänge 1 sind.
ODER (OR, Disjunktion)	≥1	Ausgang ist 1, wenn **mindestens ein** Eingang 1 ist.
XOR (Exklusiv-Oder)	=1	Ausgang ist 1, wenn die Eingänge **unterschiedlich** sind.
NICHT (NOT, Negation)	1 mit Kreis am Ausgang	Ausgang ist das **Gegenteil** des Eingangs.

So kannst du es sagen

So beschreibst du ein Gatter in Fachsprache – vom einfachen zum genauen Satz:

- *„Da kommt 1 raus, wenn beide 1 sind."* → *„Das UND-Gatter liefert genau dann eine 1, wenn an beiden Eingängen eine 1 anliegt."*
- *„Bei 1 und 0 gibt es eine 1."* → *„Zu der Belegung $A = 1$, $B = 0$ gehört beim ODER-Gatter der Ausgangswert 1."*
- *„Die Zeile 1, 1, 0 heißt …"* → *„Die letzte Zeile der Belegungstabelle besagt: Liegen an beiden Eingängen des XOR-Gatters Einsen an, ist der Ausgangswert 0."*

Arbeitsauftrag 2: Grundgatter skizzieren

Skizziere die vier Grundgatter UND, ODER, XOR und NICHT mit ihren **IEC-Symbolen**. Beschrifte jeweils die Eingänge mit A und B (beim NICHT-Gatter nur A) und den Ausgang mit Q. Schreibe unter jedes Gatter seinen Namen und in einem kurzen Satz, wann sein Ausgang 1 ist.

Tipp

Skizzieren heißt: einen Sachverhalt auf das Wesentliche reduzieren und grafisch oder sprachlich darstellen. Du brauchst also kein Lineal und keine schöne Zeichnung. Wichtig sind das Rechteck, das richtige Kennzeichen, der Kreis beim NICHT-Gatter und die beschrifteten Leitungen.

Deine Skizze:

2.1.7 Gatter in CircuitVerse ausprobieren

Im Simulator CircuitVerse kannst du Schaltungen am Bildschirm zusammenstecken und testen. Du brauchst nur einen Browser. Ein Login ist nicht nötig, aber praktisch: Dann kannst du deine Schaltungen speichern und später weiterbearbeiten. So baust du ein einzelnes Gatter auf:

1. Öffne die Seite `circuitverse.org/simulator`.
2. Im linken Bereich findest du unter **Input** den Baustein **Input**. Das ist ein Schalter, der 0 oder 1 liefert. Ziehe zwei davon ins Arbeitsfeld.
3. Unter **Output** findest du den Baustein **Output**. Er zeigt den Wert an, der bei ihm ankommt (bei einer Lampe: leuchtet = 1). Ziehe einen davon ins Arbeitsfeld.
4. Unter **Gates** findest du AND, OR, XOR und NOT. Ziehe ein Gatter zwischen die Schalter und die Anzeige.
5. Verbinde die Bausteine mit Leitungen: Klicke auf einen Anschluss, halte die Maustaste gedrückt und ziehe zum anderen Anschluss.
6. Klicke auf einen Input-Schalter. Er wechselt zwischen 0 und 1. Beobachte, was die Anzeige am Ausgang zeigt.

Arbeitsauftrag 3: Alle vier Grundgatter testen

Baue in CircuitVerse **nacheinander** je eine Schaltung mit einem AND-, einem OR-, einem XOR- und einem NOT-Gatter (beim NOT-Gatter reicht ein Eingang). Teste **alle** Eingangskombinationen und trage deine Beobachtungen in die Tabellen ein. Vergleiche danach mit den Tabellen weiter vorn in diesem Abschnitt.

A	B	Ausgang AND	Ausgang OR	Ausgang XOR
0	0			
0	1			
1	0			
1	1			

A	Ausgang NOT
0	
1	

2.1.8 Belegungstabellen mit Python nachprüfen

Belegungstabellen kannst du auch mit Python erzeugen und so deine Ergebnisse kontrollieren. Python kennt die logischen Operatoren `and`, `or` und `not` schon. Für XOR gibt es keinen eigenen Operator, aber der Vergleich `!=` („ungleich“) tut genau das Richtige: `a != b` ist wahr, wenn `a` und `b` unterschiedlich sind.

Werkzeug

Python behandelt die Zahl `0` wie `False` und die Zahl `1` wie `True`. Du kannst deshalb `and`, `or` und `not` direkt auf die Werte 0 und 1 anwenden. Das Ergebnis ist manchmal eine Zahl und manchmal ein Wahrheitswert. Damit am Ende immer 0 oder 1 herauskommt, wandeln wir es mit `int(...)` um: `int(True)` ergibt `1`, `int(False)` ergibt `0`.

Für jedes Gatter schreiben wir eine Funktion. Zwei ineinander geschachtelte `for`-Schleifen gehen alle vier Belegungen von `a` und `b` durch:

```
1  def und(a, b):
2      return int(a and b)
3
4  def oder(a, b):
5      return int(a or b)
6
7  def xor(a, b):
8      return int(a != b)
9
10 print("A B | UND ODER XOR")
11 for a in [0, 1]:
12     for b in [0, 1]:
13         print(f"{a} {b} |  {und(a, b)}   {oder(a, b)}   {xor(a, b)}")
```

AUSGABE

```
A B | UND ODER XOR
0 0 |  0   0   0
0 1 |  0   1   1
1 0 |  0   1   1
1 1 |  1   1   0
```

- **Zeilen 1–8:** Drei Funktionen, eine für jedes Gatter. Jede bekommt die beiden Eingangswerte als Parameter und gibt den Ausgangswert 0 oder 1 zurück.
- **Zeile 10:** Die Kopfzeile der Tabelle.
- **Zeilen 11–12:** Die äußere Schleife setzt `a` nacheinander auf 0 und 1. Für **jeden** dieser Werte läuft die innere Schleife einmal ganz durch und setzt `b` auf 0 und 1. So entstehen die vier Belegungen in der üblichen Reihenfolge: 0 0, 0 1, 1 0, 1 1.
- **Zeile 13:** Ein f-String gibt die Belegung und die drei Ausgangswerte in einer Zeile aus.

Das NICHT-Gatter hat nur einen Eingang. Hier genügt eine Schleife:

```
1 def nicht(a):
2     return int(not a)
3
4 for a in [0, 1]:
5     print(a, "->", nicht(a))
```

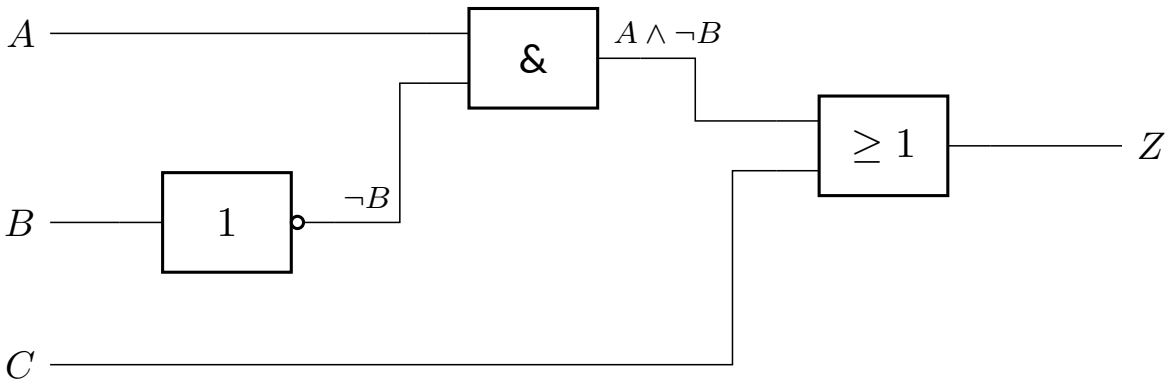
AUSGABE

```
0 -> 1
1 -> 0
```

- **Zeilen 1–2:** Die Funktion `nicht` bekommt nur einen Eingangswert. `not a` dreht ihn um, `int(...)` macht daraus wieder 0 oder 1.
- **Zeilen 4–5:** Die Schleife setzt `a` nacheinander auf 0 und 1 und gibt jeweils Eingang und Ausgang aus.

2.1.9 Schaltungen aus mehreren Gattern

Ein einzelnes Gatter kann nicht viel. Spannend wird es, wenn man Gatter hintereinanderschaltet: Der Ausgang eines Gatters wird zum Eingang des nächsten. Die folgende Schaltung hat drei Eingänge A, B, C und einen Ausgang Z.



Eine Schaltung aus drei Gattern. Sie berechnet den Term $(A \wedge \neg B) \vee C$

Um die Belegungstabelle aufzustellen, gehst du die Schaltung **von links nach rechts** durch, Gatter für Gatter. Für jede Zwischenleitung legst du eine eigene Spalte an. Solche Spalten heißen **Hilfsspalten**.

1. Das NICHT-Gatter dreht B um. Die erste Hilfsspalte ist $\neg B$.
2. Das UND-Gatter verknüpft A mit dem Ausgang des NICHT-Gatters. Die zweite Hilfsspalte ist $A \wedge \neg B$. Sie ist nur dort 1, wo A **und** $\neg B$ beide 1 sind.
3. Das ODER-Gatter verknüpft diese Leitung mit C. Das ergibt den Ausgang $Z = (A \wedge \neg B) \vee C$. Er ist 1, wo die zweite Hilfsspalte **oder** C eine 1 enthält.

A	B	C	$\neg B$	$A \wedge \neg B$	$(A \wedge \neg B) \vee C$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

Lies zur Kontrolle eine Zeile in Worten: Die fünfte Zeile besagt, dass zu der Belegung $A = 1, B = 0, C = 0$ der Ausgangswert $Z = 1$ gehört. Das stimmt: $\neg B$ ist 1, also ist auch $A \wedge \neg B$ gleich 1, und damit ist das ODER-Gatter erfüllt.

Merke

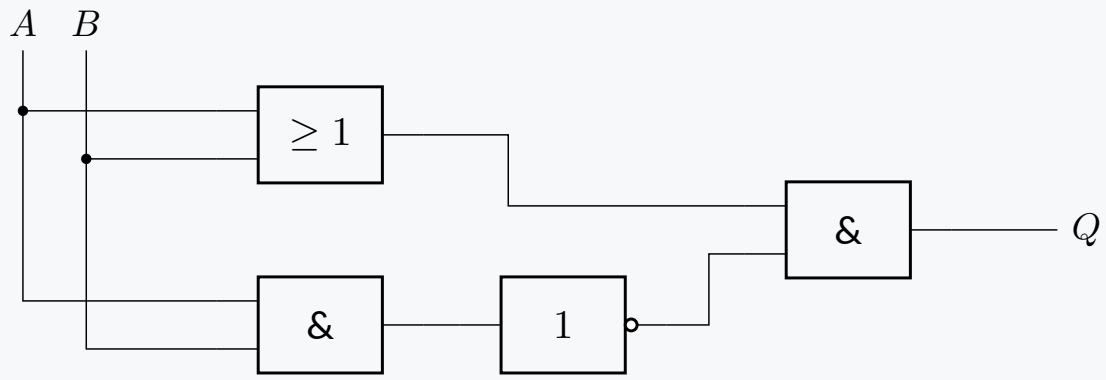
Zu einer Schaltung aus mehreren Gattern erstellt man die Belegungstabelle von links nach rechts: Für den Ausgang jedes Gatters legt man eine **Hilfsspalte** an und berechnet sie aus den Spalten, die schon da sind.

Stolperstein

Der häufigste Fehler bei Belegungstabellen ist, eine Zeile auszulassen oder doppelt zu schreiben. Halte dich deshalb immer an die feste Reihenfolge der Zeilen (letzte Eingangsspalte wechselt in jeder Zeile). Zähle am Ende nach: Bei 3 Eingängen müssen es genau 8 Zeilen sein.

Arbeitsauftrag 4: Belegungstabelle zu einer Schaltung

Erstelle die Belegungstabelle zur folgenden Schaltung. Fülle dazu die Hilfsspalten von links nach rechts aus. Mit welchem Grundgatter stimmt die letzte Spalte überein?



Schaltung mit zwei Eingängen und vier Gattern

A	B	$A \vee B$	$A \wedge B$	$\neg(A \wedge B)$	Q
0	0				
0	1				
1	0				
1	1				

Die letzte Spalte stimmt überein mit:

Arbeitsauftrag 5: Mehrheitsschaltung

Baue in CircuitVerse eine Schaltung mit **drei Eingängen** A, B, C, die genau dann 1 ausgibt, wenn **mindestens zwei** der drei Eingänge 1 sind (Mehrheitsentscheidung, englisch *majority*). Verwende dafür mehrere AND- und OR-Gatter.

1. Fülle die Belegungstabelle aus.

2. Skizziere die Schaltung mit IEC-Symbolen.
3. Teste deine Schaltung in CircuitVerse mit allen acht Belegungen.

Tipp: Die Schaltung lässt sich als $(A \wedge B) \vee (B \wedge C) \vee (A \wedge C)$ beschreiben: Entweder sind A und B beide 1, oder B und C, oder A und C. In CircuitVerse kannst du beim OR-Gatter die Anzahl der Eingänge auf 3 stellen oder zwei OR-Gatter hintereinanderschalten.

A	B	C	$A \wedge B$	$B \wedge C$	$A \wedge C$	Q
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Deine Skizze:

Gymnasium Ottweiler

2.1.10 Übungen

Arbeitsauftrag 6: Welches Gatter ist das?

Zu vier unbekannten Gattern mit den Eingängen A und B wurden die Ausgangswerte Q_1 bis Q_4 gemessen. Gib jeweils an, um welches Gatter es sich handelt. Eine der vier Spalten passt zu keinem Grundgatter. Begründe, woran du das erkennst.

A	B	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

Deine Antwort:

2.1.11 Weitere Übungen

Hier kannst du das Gelernte weiter üben. Die Punkte im Titel zeigen, wie schwer eine Aufgabe ist: ●○○ leicht, ●●○ mittel, ●●● knifflig. Fang mit den leichten an.

Arbeitsauftrag 7: Gatter erkennen ●○○

Die Abbildung zeigt vier Gatter in IEC-Darstellung. Nenne zu jedem Gatter den Namen und beschreibe in einem kurzen Satz, wann sein Ausgang 1 ist.



Vier Gatter, durchnummeriert von (1) bis (4)

Deine Antwort:

Gymnasium Ottweiler

Arbeitsauftrag 8: Ausgang bestimmen ●○○

Bestimme für jede Zeile den Ausgangswert des Gatters. Beim NICHT-Gatter gibt es nur den Eingang A.

Gatter	A	B	Ausgang
UND	1	0	
ODER	1	0	
XOR	1	0	
UND	1	1	
ODER	1	1	
XOR	1	1	
ODER	0	0	
NICHT	1	–	

Arbeitsauftrag 9: Gatter im Alltag ●○○

Gib zu jeder Situation das passende Grundgatter an. Begründe kurz.

a) Ein Handy lässt sich entsperren, wenn der Fingerabdruck erkannt wird oder die richtige PIN eingegeben wird.
b) Ein Online-Spiel startet nur, wenn du eingeloggt bist und das Internet verbunden ist.
c) Bei einem Quiz hat jedes der beiden Teams einen Buzzer. Die Lampe „Antwort“ leuchtet, wenn genau ein Team gedrückt hat. Drücken beide gleichzeitig, bleibt sie aus.
d) Die Warnleuchte am Wasserkocher leuchtet, wenn **kein** Wasser im Kocher ist. (Eingang: 1 = Wasser ist im Kocher.)

Deine Antwort:

Gymnasium Ottweiler

Arbeitsauftrag 10: Welche Zeile fehlt? ●○○

Bei dieser Belegungstabelle für drei Eingänge fehlt eine Zeile.

1. Gib an, wie viele Zeilen die Tabelle haben muss.
2. Bestimme die fehlende Belegung und gib an, an welcher Stelle sie eingefügt werden muss.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	1	0
1	1	1

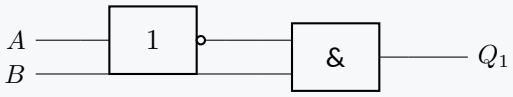
Deine Antwort:

Gymnasium Ottweiler

Arbeitsauftrag 11: Zwei kleine Schaltungen ●●○

Erstelle zu beiden Schaltungen die Belegungstabelle. Fülle die Hilfsspalten aus. Beschreibe danach in Worten, wann $Q_2 = 1$ ist.

a)



b)



Links Schaltung a) mit dem Ausgang Q_1, rechts Schaltung b) mit dem Ausgang Q_2

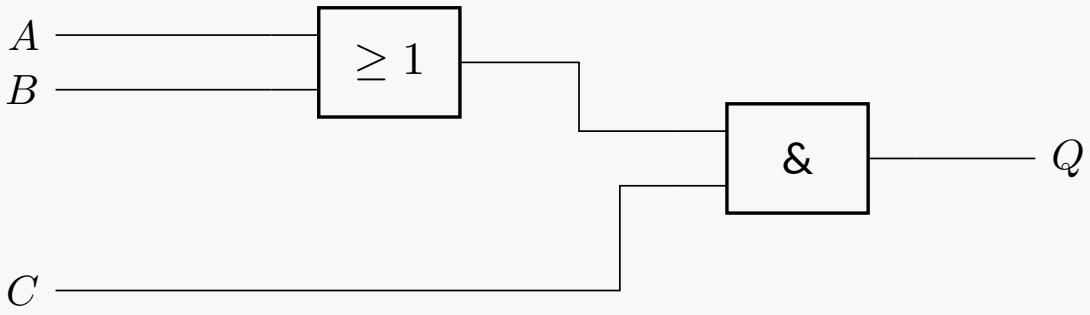
A	B	$\neg A$	Q_1	$A \oplus B$	Q_2
0	0				
0	1				
1	0				
1	1				

Deine Beschreibung:

Gymnasium Ottweiler

Arbeitsauftrag 12: Eine Schaltung mit drei Eingängen ●●○

Erstelle die Belegungstabelle zur folgenden Schaltung. Gib danach alle Belegungen an, für die $Q = 1$ ist.



Schaltung mit drei Eingängen: ein ODER-Gatter und ein UND-Gatter

A	B	C	$A \vee B$	Q
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Gymnasium Ottweiler

Deine Antwort:

Arbeitsauftrag 13: Kontrolle mit Python ●●○

Überprüfe deine Tabellen aus den Aufträgen *Zwei kleine Schaltungen* und *Eine Schaltung mit drei Eingängen* mit Python. Schreibe für jede der drei Schaltungen eine Funktion (`q1(a, b)`, `q2(a, b)` und `q3(a, b, c)`), die 0 oder 1 zurückgibt. Gib die Belegungstabellen mit geschachtelten `for`-Schleifen aus.

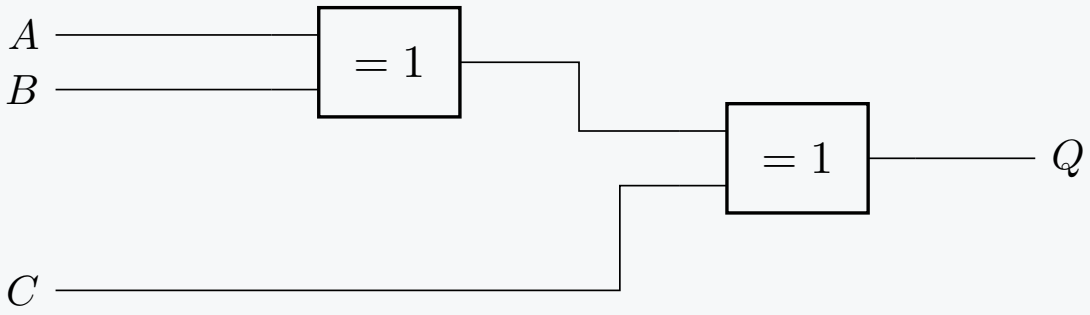
Tipp

XOR schreibst du in Python als `a != b`. Ein NICHT-Gatter hinter dem XOR-Gatter wird dann zu `not (a != b)`. Die Klammern sind wichtig, damit `not` sich auf das ganze Ergebnis bezieht.

Arbeitsauftrag 14: Zwei XOR-Gatter hintereinander ●●●

In der folgenden Schaltung ist der Ausgang eines XOR-Gatters mit einem zweiten XOR-Gatter verbunden.

1. Erstelle die Belegungstabelle mit der Hilfsspalte $A \oplus B$.
2. Zähle in jeder Zeile, wie viele der drei Eingänge 1 sind, und trage die Anzahl in die Spalte „Einsen“ ein. Beschreibe damit in einem Satz, wann $Q = 1$ ist.



Zwei XOR-Gatter hintereinander

A	B	C	$A \oplus B$	Q	**Einsen**
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Deine Beschreibung:

2.1.12 Das Wichtigste in Kürze

- In logischen Schaltungen hat jede Leitung den Wert **0** oder **1** (ein **Bit**).
- Die vier **Grundgatter**: UND (&, $A \wedge B$, alle Eingänge 1), ODER (≥1, $A \vee B$, mindestens ein Eingang 1), XOR (=1, $A \oplus B$, Eingänge unterschiedlich), NICHT (1 mit Kreis am Ausgang, $\neg A$, Gegenteil des Eingangs).
- Eine **Belegungstabelle** listet zu jeder Kombination der Eingangswerte den Ausgangswert auf. Bei n Eingängen hat sie 2^n Zeilen.
- Bei Schaltungen aus mehreren Gattern rechnet man von links nach rechts und legt für jede Zwischenleitung eine **Hilfsspalte** an.
- In Zeichnungen verwendet man die IEC-Symbole. CircuitVerse zeigt die US-Symbole; sie bedeuten dasselbe.

Selbstcheck

- ☐ Ich kann die vier Grundgatter nennen und mit IEC-Symbolen skizzieren.
- ☐ Ich kann zu jedem Grundgatter die Belegungstabelle aus dem Kopf aufschreiben.
- ☐ Ich kann den Unterschied zwischen ODER und XOR erklären.
- ☐ Ich weiß, warum eine Belegungstabelle mit 3 Eingängen 8 Zeilen hat, und kann die Zeilen in der richtigen Reihenfolge aufschreiben.
- ☐ Ich kann zu einer Schaltung aus mehreren Gattern die Belegungstabelle mit Hilfsspalten erstellen.
- ☐ Ich kann ein Gatter in CircuitVerse aufbauen, testen und die US-Symbole lesen.

Gymnasium Ottweiler