



Tag 1

Skript

Vorkurs Informatik

Sommersemester 2020

Ronja Düffel
Stand: 22. März 2020

Inhaltsverzeichnis

I. Digitaltechnik	5
1. Codes	9
1.1. Dualcode	9
1.2. Alphanumerische Codes	10
2. Schaltalgebra	11
2.1. Schaltvariablen und Schaltfunktionen	11
2.1.1. Grundverknüpfungen und ihre Darstellung	12
2.2. Darstellung von Schaltfunktionen	13
2.2.1. Wahrheitstabelle	13
2.2.2. Funktionsgleichung	14
Vorrang- und Klammerregeln	14
2.2.3. Darstellung mit Schaltsymbolen	15
Symbolische Darstellung logischer Gatter	15
2.3. Gesetze der Schaltalgebra	17

Teil I.

Digitaltechnik

Als eines der Hauptgebiete der Informatik beschäftigt sich die Technische Informatik mit Architektur, Entwurf, Realisierung, Bewertung und Betrieb von Rechnersystemen auf der Ebene der Hardware und der systemnahen Software.

Computer sind digitale Systeme. Das heißt, dass Informationen in wertdiskreten Zuständen codiert und verarbeitet werden. In der binären Digitaltechnik wie sie in Computern zur Anwendung kommt, gibt es nur zwei Zustände, nämlich 0 und 1 oder auch *false* und *true*. Technisch sind diese im Computer als unterschiedliche Spannungspegel realisiert. Liegt die Spannung im hohen Bereich, so liegt Zustand *H* vor, im unteren Bereich Zustand *L*. Ist der hohe Spannungspegel *H* dem 1-Zustand zugeordnet ($H \leftarrow 1$) und der niedrige 0 ($L \leftarrow 0$), so spricht man von *positiver Logik*. Der umgekehrte Fall, $H \leftarrow 0, L \leftarrow 1$ heißt *negative Logik*

positive und
negative Lo-
gik

Das heißt, für die Darstellung von Informationen im Computer stehen lediglich zwei Werte zur Verfügung. Um also nun Zahlen und Schriftzeichen darstellen zu können, müssen diese codiert werden.

1. Codes

Ein Code bildet Zeichen eines Zeichenvorrats auf die Zeichen eines anderen Zeichenvorrats ab. Ein bekanntes Beispiel ist der Morsecode, der Buchstaben auf Kombinationen von kurzen und langen Signalen (Licht, Ton, etc) abbildet. In digitalen Systemen kommen Dualcodes zum Einsatz.

1.1. Dualcode

Dualcodes beruhen auf nur zwei Zeichen, nämlich 0 und 1. Daher spricht man auch von *Binärcode*. Wie im Dezimalsystem auch, werden größere Zahlen durch Aneinanderreihung der zur Verfügung stehenden Zeichen gebildet, wobei die Position des Zeichens bestimmt, wieviel es zum Gesamtwert beiträgt. Man spricht auch von gewichteten Codes, da weiter links stehende Zeichen mehr Gewicht haben.

Binärcode

Eine Dual- oder Binärzahl Z_2 besteht aus einer Aneinanderreihung von Zeichen ($z_i \in \{0, 1\}$). Die Zeichen z_i eines Wortes werden in der Digitaltechnik *Bits* genannt. Ein Bit ist die kleinste Informationseinheit. Die nächst größere ist ein *Byte*. Ein Byte besteht aus 8 Bit.

Bit
Byte

Zurück zu den Binärzahlen. Eine Binärzahl mit n Stellen, sieht also wie folgt aus:

$$Z_2 = z_{n-1}z_{n-2}z_{n-3} \dots z_1z_0$$

Die Position zählt also von rechts nach links hoch. Da das erste (rechtste!) Zeichen die Position 0 hat, hat das n -te Zeichen die Position $n - 1$.

Den einzelnen Bits werden entsprechend ihrer Position i in der Zahl Gewichte 2^i zugeordnet. Damit kann die äquivalente Dezimalzahl Z_{10} berechnet werden:

$$Z_{10} = g(Z_2) = z_{n-1} \cdot 2^{n-1} + z_{n-2} \cdot 2^{n-2} + \dots + z_1 \cdot 2^1 + z_0 \cdot 2^0$$

Das sieht für manche jetzt vielleicht, wegen der mathematischen Notation, kompliziert aus. Es ist aber eigentlich nichts anderes, als das was wir von den Dezimalzahlen, mit denen wir alltäglich umgehen, kennen. Der Umgang mit Dezimalzahlen ist uns nur so geläufig, dass wir uns dieser Interpretation der Zahlen nicht mehr bewusst sind.

Beispiel 1.1.

Die Dezimalzahl 243_{10} , die durch den Index 10 als solche gekennzeichnet ist, hat den Wert $2 \cdot 10^2 + 4 \cdot 10^1 + 3 \cdot 10^0 = 2 \cdot 100 + 4 \cdot 10 + 3 \cdot 3 = 243$. Die Dezimalzahl 324_{10} besteht zwar aus den gleichen Zeichen, repräsentiert aber einen anderen Wert, nämlich $3 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 = 300 + 20 + 4 = 324$.

Im Dezimalsystem stehen zehn Zeichen zur Verfügung: 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9. Die Basis ist daher 10. Das garantiert, dass alle Werte dargestellt werden können und die Darstellung eindeutig ist. Eindeutigkeit heißt, dass eine Zahl nur auf eine Art und Weise dargestellt werden kann.

Im Dual- oder Binärsystem stehen zwei Zeichen zur Verfügung und die Basis ist dementsprechend 2. Um eine Dualzahl in eine Dezimalzahl umzurechnen, interpretieren wir sie genau so, wie zuvor die Dezimalzahl, allerdings zur Basis 2.

Beispiel 1.2.

Die Dualzahl 11110011_2 , gekennzeichnet durch den Index 2, steht also für den Dezimalwert $1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 243_{10}$

1. Codes

Um eine Dezimalzahl in eine Dualzahl umzuwandeln, teilt man durch 2 und notiert die Reste, von rechts nach links (**Achtung:** das ist entgegen der gewohnten Schreibrichtung!). Für die Dezimalzahl 324_{10} ergibt sich z.B.:

$$\begin{array}{rll} 324 : 2 = & 162 & \text{Rest 0} \\ & 162 : 2 = & 81 & \text{Rest 0} \\ & & 81 : 2 = & 40 & \text{Rest 1} \\ & & & 40 : 2 = & 20 & \text{Rest 0} \\ & & & & 20 : 2 = & 10 & \text{Rest 0} \\ & & & & & 10 : 2 = & 5 & \text{Rest 0} \\ & & & & & & 5 : 2 = & 2 & \text{Rest 1} \\ & & & & & & & 2 : 2 = & 1 & \text{Rest 0} \\ & & & & & & & & 1 : 2 = & 0 & \text{Rest 1} \end{array}$$

Die Dualzahl $101000100_2 = 324_{10}$

1.2. Alphanumerische Codes

Um nun Buchstaben, Ziffern und Sonderzeichen darstellen zu können, gibt es alphanumerische Codes. Ein sehr bekannter ist der ASCII-Code. Hierbei wird jedes Zeichen mit 7 Bit ($a_6 a_5 a_4 a_3 a_2 a_1 a_0$) codiert. Die ersten drei Bit a_6, a_5 und a_4 geben die Spalte, die letzten vier Bit a_3, a_2, a_1 und a_0 geben die Zeile der Codierungstabelle an. Die ASCII-Tabelle hat $2^7 = 128$ Zeichen. Im Folgenden interessieren uns aber erstmal nur Zahlen.

2. Schaltalgebra

Unter einer Algebra versteht man in der Mathematik eine Menge von Elementen und Verknüpfungen auf dieser Menge (z.B. die Menge der ganzen Zahlen und die Verknüpfungen der Multiplikation und Addition $(\mathbb{Z}, \cdot, +)$). Die Schaltalgebra als Spezialfall der *bool'schen Algebra*, liefert die theoretische Grundlage der Digitaltechnik. Mit ihr können digitale Schaltungen und Schaltnetze entworfen und beschrieben werden.

bool'sche Algebra

Die Schaltalgebra ist auf der zweielementigen Menge $\mathcal{B} = \{0, 1\}$ definiert. Es existieren die drei Grundverknüpfungen (*Operatoren*) \wedge , \vee und \neg , für die entsprechende Schaltzeichen (Logik-Gatter) eingeführt werden (siehe Tab. 2.1).

Operatoren
Logik-
Gatter




Operator	Schaltzeichen	Benennung
\neg		NICHT - Gatter
\wedge		UND - Gatter
\vee		ODER - Gatter

Tabelle 2.1.: Operatoren und Schaltzeichen nach IEC 60617-12 : 1997

2.1. Schaltvariablen und Schaltfunktionen

Eine Schaltvariable ist eine binäre Variable, d.h. sie kann einen von zwei Werten annehmen, nämlich 0 oder 1. Mit den Verknüpfungen \wedge , \vee , \neg können Funktionen aufgestellt werden. Wie in der (allgemeinen) Algebra, die man aus der Schule kennt, beschreibt eine Funktion der Schaltalgebra die Abhängigkeit einer Ausgangsvariablen y von einer oder mehreren unabhängigen, binären Schaltvariablen a, b, c, \dots

Schaltvariable

Beispiel 2.1.

Funktion der allgemeinen Algebra:

$$f(x) = 5x + 3 \text{ oder } y = 5x + 3$$

Funktion der Schaltalgebra:

$$f(a, b) = a \wedge b \text{ oder } y = a \wedge b$$

Definition 2.2 (Schaltfunktion).

Eine Schaltfunktion ist eine Gleichung der Schaltalgebra, die die Abhängigkeit einer oder mehrerer, binärer Schaltvariablen y (Ausgangsvariable(n)) von einer oder mehreren, unabhängigen binären Schaltvariablen x beschreibt. Handelt es sich um mehrere Ein- und Ausgangsvariablen, so sind x und y Vektoren. Für mehrere Eingangsvariablen schreiben wir $x = (a, b, c, \dots)$, für mehrere Ausgangsvariablen $y = (y_1, y_2, y_3, \dots)$.

Schaltfunktion

2. Schaltalgebra

Beispiel 2.3.

Die Addition zweier 4 Bit Zahlen kann als Schaltfunktion mit einem Eingabevektor der Länge 8 und einem Ausgabevektor der Länge 5 realisiert werden. Hierbei entsprechen die ersten 4 Bit des Eingabevektors dem ersten Summanden, und die zweiten 4 Bit entsprechen dem zweiten Summanden.

Bemerkung: Im Vorsemesterkurs werden wir uns auf Schaltfunktionen mit nur einer Ausgangsvariablen beschränken.

Definition 2.4 (Stelligkeit einer Schaltfunktion).

Stelligkeit

Die *Stelligkeit* einer Schaltfunktion ist die Anzahl ihrer Eingangsvariablen.

Die Funktion $f(a) = \bar{a}$ beispielsweise ist eine einstellige Funktion. Bei der Funktion $g(a, b, c)$ handelt es sich um eine dreistellige Funktion, $h(a, b, c, d, e)$ ist eine 5-stellige Funktion.

Definition 2.5 (Belegung einer Schaltvariablen).

Belegung

Unter der *Belegung einer Schaltvariablen* versteht man die Zuweisung eines konkreten Wertes (0 oder 1) an eine Schaltvariable.

Die Belegung einer mehrstelligen Funktion kann als Vektor angegeben werden. Beispielsweise bedeutet die Belegung $(1, 1, 0)$ für die Schaltfunktion $g(a, b, c)$, dass a der Wert 1, b der Wert 1 und c der Wert 0 zugewiesen wird.

2.1.1. Grundverknüpfungen und ihre Darstellung

Für die Grundverknüpfungen der Schaltalgebra \wedge , \vee und $\bar{}$ gibt es drei gleichwertige Darstellungen.

- Wahrheitstabelle,
- Schaltzeichen und
- Funktion

Name	Wahrheitstabelle	Schaltzeichen	Funktion															
NICHT, NOT, Komplement, Negation	<table border="1"> <tr> <td>a</td> <td>\bar{a}</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	a	\bar{a}	0	1	1	0		$y = f(a) = \bar{a}$									
a	\bar{a}																	
0	1																	
1	0																	
UND, AND, Konjunktion	<table border="1"> <tr> <td>a</td> <td>b</td> <td>$a \wedge b$</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	a	b	$a \wedge b$	0	0	0	0	1	0	1	0	0	1	1	1		$y = f(a, b) = a \wedge b$
a	b	$a \wedge b$																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
ODER, OR, Disjunktion	<table border="1"> <tr> <td>a</td> <td>b</td> <td>$a \vee b$</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	a	b	$a \vee b$	0	0	0	0	1	1	1	0	1	1	1	1		$y = f(a, b) = a \vee b$
a	b	$a \vee b$																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Tabelle 2.2.: Grundverknüpfungen und ihre Darstellung

Definition 2.6 (Gleichheit von Schaltfunktionen).

Wir bezeichnen zwei Schaltfunktionen als gleich, wenn sie für dieselbe Belegung der Eingangsvariablen, dasselbe Ergebnis in Bezug auf den Wert der Ausgangsvariablen y haben.

Beispiel 2.7.

Die Funktionen $f(a, b) = a \wedge b$ und $g(a, b) = \overline{\overline{a} \vee \overline{b}}$ sind gleich. Das lässt sich leicht mit einer Wahrheitstabelle überprüfen.

a	b	$f(a, b) = a \wedge b$	\overline{a}	\overline{b}	$\overline{a} \vee \overline{b}$	$g(a, b) = \overline{\overline{a} \vee \overline{b}}$
0	0	0	1	1	1	0
0	1	0	1	0	1	0
1	0	0	0	1	1	0
1	1	1	0	0	0	1

Für alle Belegungen von a und b sind die Funktionswerte der Schaltfunktionen f und g gleich. $f(a, b) = 0$ genau dann, wenn $g(a, b) = 0$ und $f(a, b) = 1$ genau dann, wenn $g(a, b) = 1$.

Die Funktionen $f(a, b) = a \wedge b$ und $g(a, b) = a \vee b$ sind nicht gleich, denn für die Belegung $a = 0$ und $b = 1$ gilt: $f(a, b) = 0$ und $g(a, b) = 1$, wie man folgender Wahrheitstabelle entnehmen kann.

a	b	$f(a, b) = a \wedge b$	$g(a, b) = a \vee b$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

2.2. Darstellung von Schaltfunktionen

Es gibt vier gleichwertige Darstellungsformen von Schaltfunktionen:

- Wahrheitstabelle
- Funktionsgleichung
- Schaltzeichen
- KV-Diagramm

2.2.1. Wahrheitstabelle

Wahrheitstabellen haben wir oben bereits genutzt, um Schaltfunktionen darzustellen, hier führen wir sie ausführlicher ein. Die Wahrheitstabelle ist eine Tabelle, die alle möglichen Kombinationen der Eingangsvariablen und den, durch die Funktion zugeordneten Wert der Ausgangsvariablen y enthält. Eine Wahrheitstabelle gibt also für jede Wertekombination der Eingangsvariablen den zugehörigen Funktionswert der Ausgangsvariablen y an. Da es bei n binären Eingangsvariablen 2^n Kombinationsmöglichkeiten gibt, hat die Wahrheitstabelle einer Funktion mit n Eingangsvariablen 2^n Zeilen. So hat die Wahrheitstabelle der 3-stelligen Funktion $f(a, b, c)$ in Beispiel 2.8, $n = 3$ Eingangsvariablen (a, b und c) und somit $2^3 = 8$ Zeilen.

Ist die Schaltfunktion, für die man eine Wahrheitstabelle aufstellt, komplexer, kann es sinnvoll sein, die Wahrheitswerte einzelner Terme ebenfalls in der Wahrheitstabelle anzugeben (siehe Bsp. 2.9).

2. Schaltalgebra

Beispiel 2.8.

Die Wahrheitstabelle der 3-stelligen Schaltfunktion $f(a, b, c)$, die genau dann den Wert 1 hat, wenn eine ungerade Anzahl an Eingangsvariablen den Wert 1 hat, sieht folgendermaßen aus:

a	b	c	$y = f(a, b, c)$
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

Tabelle 2.3.: Wahrheitstabelle für 3-stellige Schaltfunktion $f(a, b, c)$.

2.2.2. Funktionsgleichung

Jede binäre Schaltfunktion kann allein durch die Grundverknüpfungen \wedge, \vee und $\bar{}$ dargestellt werden.

Beispiel 2.9.

$$y = g(a, b, c) = (a \vee b) \wedge (\overline{a \wedge \bar{c}}) \vee (b \wedge c)$$

a	b	c	\bar{c}	$a \vee b$	$a \wedge \bar{c}$	$\overline{a \wedge \bar{c}}$	$(a \vee b) \wedge (\overline{a \wedge \bar{c}})$	$b \wedge c$	$g(a, b, c)$
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0
0	1	0	1	1	0	1	1	0	1
0	1	1	0	1	0	1	1	1	1
1	0	0	1	1	1	0	0	0	0
1	0	1	0	1	0	1	1	0	1
1	1	0	1	1	1	0	0	0	0
1	1	1	0	1	0	1	1	1	1

Vorrang- und Klammerregeln

Wie in der allgemeinen Algebra, gibt es auch in der Schaltalgebra bei der Auswertung der Ausdrücke Vorrangregeln zu beachten. Es gilt Negation, vor Konjunktion, vor Disjunktion, also $\bar{}$, vor \wedge , vor \vee , analog zu "Punkt- vor Strichrechnung" in der allgemeinen Algebra. Wenn Formeln durch die Verknüpfung anderer Formeln entstehen, sollten Klammern gesetzt werden. Manchmal ist das setzen von Klammern unerlässlich (siehe Beispiel 2.10)

Bei direkten UND-Verknüpfungen einzelner Variablen, wird häufig das Verknüpfungszeichen \wedge der besseren Lesbarkeit wegen weggelassen. Statt $a \wedge \bar{b} \wedge c$ schreibt man $a\bar{b}c$.

Beispiel 2.10.

Betrachten wir die drei Schaltfunktionen f, g und h und die zugehörige Wahrheitstabelle (Tab. 2.4):

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

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

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

a	b	c	$b \wedge c$	$f(a, b, c) = a \vee b \wedge c$	$g(a, b, c) = a \vee (b \wedge c)$	$h(a, b, c) = a \vee b$	$(a \vee b) \wedge c$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	0
1	1	1	1	1	1	1	1

Tabelle 2.4.: Wahrheitstabelle für die Schaltfunktionen $f(a, b, c)$, $g(a, b, c)$ und $h(a, b, c)$.

Die Schaltfunktionen f und g sind gleich, da die Auswertung von f aufgrund der Vorrangregeln sowieso so erfolgt, wie in g durch die Klammern vorgegeben. Insofern sind die Klammern in g überflüssig, sollten aber trotzdem gesetzt werden, um Fehler zu vermeiden. Funktion h jedoch ist nicht gleich zu den Funktion f und g , da hier durch die Klammerung die ODER-Verknüpfung vor der UND-Verknüpfung ausgewertet werden muss.

Ein analoges Beispiel aus der allgemeinen Algebra wären die Gleichungen

$$x = 3 + (5 \cdot 2) = 3 + 10 = 13 \text{ und}$$

$$y = (3 + 5) \cdot 2 = 8 \cdot 2 = 16.$$

2.2.3. Darstellung mit Schaltsymbolen

Alle Verknüpfungen können auch durch Schaltsymbole dargestellt werden (vergl. Tab. 2.1). Also lassen sich auch alle Schaltfunktion mit Hilfe von Schaltsymbolen als Schaltung darstellen. Die technischen Bauteile, die die jeweiligen Verknüpfungen darstellen, nennt man *Gatter*. Sie können technisch auf unterschiedliche Arten realisiert werden. Wie, erfahren Sie in der Lehrveranstaltung "Rechnertechnologie und kombinatorische Schaltungen" (RTKS).

Gatter

Symbolische Darstellung logischer Gatter

Die Darstellung logischer Gatter ist in IEC 60617-12 geregelt. Logische Gatter werden durch ein hochkant stehendes Rechteck dargestellt. Die Funktion des Gatters wird durch ein Symbol innerhalb des Rechtecks angeben. $\&$ steht für UND (Konjunktion, \wedge), ≥ 1 für ODER (Disjunktion, \vee). Das Zeichen für ODER behält man besser, wenn man sich klar macht, dass eine Disjunktion immer dann zu 1 auswertet, wenn mindestens eine der Eingangsvariablen den Wert 1 hat.

Das Gatter für die Negation (\neg) hat als Symbol im Rechteck eine 1 und am Ausgang einen kleinen Kreis.

Die Eingänge sind links, die Ausgänge rechts der Symbole angeordnet (siehe Abb.:2.1).

2. Schaltalgebra

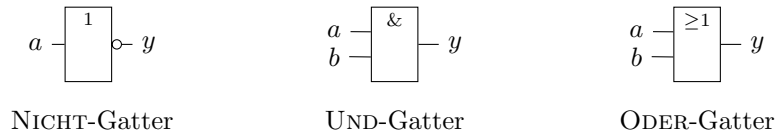


Abbildung 2.1.: Schaltsymbole der Grundverknüpfungen \neg , \wedge und \vee .

Bemerkung: Dies sind die Symbole, wie sie im deutschen Sprachraum üblich sind. In englischsprachigen Texten finden sich häufig die amerikanischen Symbole, wie sie in ANSI/IEEE Std 91/91a-1991 geregelt sind.¹

Durch Kombination der Symbole lassen sich nun Schaltfunktionen darstellen

Beispiel 2.11.

Die Schaltfunktion $f(a, b) = (\bar{a} \vee b)$ kann durch Kombination eines NICHT-Gatters und eines ODER-Gatters dargestellt werden (Abb.: 2.2).

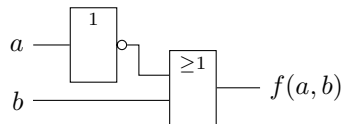


Abbildung 2.2.: Schaltung für $f(a, b) = (\bar{a} \vee b)$

Beispiel 2.12.

Die Schaltfunktion $g(a, b, c) = (a \wedge \bar{c}) \vee (\bar{b} \wedge c)$ kann durch Kombination zweier NICHT-Gatter, zweier UND-Gatter und eines ODER-Gatters dargestellt werden (siehe Abb.: 2.3).

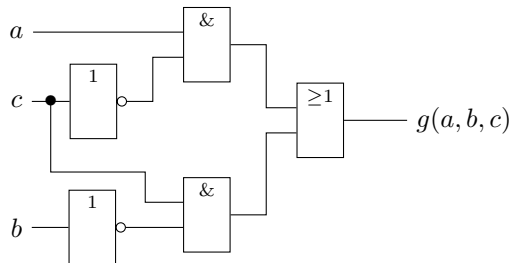


Abbildung 2.3.: Schaltung für $g(a, b, c) = ((a \wedge \bar{c}) \vee (\bar{b} \wedge c))$

Bei der Darstellung der Schaltfunktion als Schaltung, muss natürlich ebenfalls auf die Auswertungsreihenfolge der Verknüpfungen geachtet werden. Die Auswertung der Signale in einer logischen Schaltung erfolgt von links nach rechts. Also müssen Terme, die zuerst ausgewertet werden müssen, weiter links stehen, als solche, die später ausgewertet werden.

Beispiel 2.13.

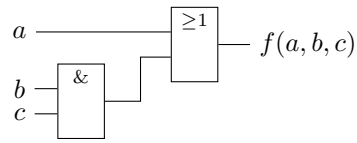
Betrachten wir die Schaltfunktionen g und h aus Beispiel 2.10.

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

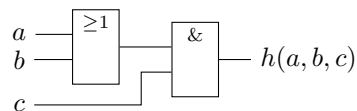
$$h(a, b, c) = (a \vee b) \wedge c$$

In Funktion g muss der Term $b \wedge c$ vor der ODER-Verknüpfung mit a ausgewertet werden. Die Schaltung dazu zeigt Abbildung 2.4).

¹Einen guten Überblick liefert die Wikipediaseite: <https://de.wikipedia.org/wiki/Logikgatter>

Abbildung 2.4.: Schaltung für $g(a, b, c) = a \vee (b \wedge c)$

In Funktion h muss der Term $a \vee c$ vor der UND-Verknüpfung mit a ausgewertet werden. Abbildung 2.5 zeigt die Schaltung zu Funktion $h(a, b, c)$.

Abbildung 2.5.: Schaltung für $h(a, b, c) = (a \vee b) \wedge c$

2.3. Gesetze der Schaltalgebra

Wie für die allgemeine Algebra gibt es auch für die Schaltalgebra einige Umformungsregeln und Gesetze.

Tipp: Wenn man sich \wedge als Multiplikation und \vee als Addition vorstellt, kommen einem viele der Gesetze gar nicht neu vor.

Für Schaltvariablen a, b und c , gelten folgende Gesetze:

1. Kommutativgesetze:

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

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

2. Assoziativgesetze:

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

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

3. Absorptionsgesetze:

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

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

4. Distributivgesetze:

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

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

5. Neutrale Elemente:

$$a \wedge 1 = a$$

$$a \vee 0 = a$$

2. Schaltalgebra

6. Inverse Elemente:

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

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

7. Reduktionsgesetze:

$$a \wedge 0 = 0$$

$$a \vee 0 = a$$

$$a \vee 1 = 1$$

$$a \wedge 1 = a$$

8. Idempotenzgesetze:

$$a \wedge a = a$$

$$a \vee a = a$$

9. Resolutionsregeln:

$$(a \wedge b) \vee (a \wedge \bar{b}) = a$$

$$(a \vee b) \wedge (a \vee \bar{b}) = a$$

10. De Morgansche Regeln:

$$\overline{a \wedge b} = \bar{a} \vee \bar{b}$$

$$\overline{a \vee b} = \bar{a} \wedge \bar{b}$$