



Tag 3

Skript
Vorkurs Informatik
Sommersemester 2020

Ronja Düffel
Stand: 24. 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
2.4. Weitere Verknüpfungen	18
2.4.1. Vollständige Operatorensysteme	20
2.5. Normalformen	20
2.5.1. Min- und Maxterme	21
2.6. KV-Diagramme	24
2.7. Minimierung von Schaltfunktionen	26
2.7.1. Minimierung mit den Gesetzen der Schaltalgebra	28
2.7.2. Minimierung mit KV-Diagrammen	28
Aufstellung einer minimalen DNF	32
2.8. Unvollständig spezifizierte Funktionen	33
2.9. Schaltungsentwurf	33

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 ($a_6a_5a_4a_3a_2a_1a_0$) 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}	$\overline{\overline{a} \vee \overline{b}}$	$g(a, b) = \overline{\overline{a} \vee \overline{b}}$
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

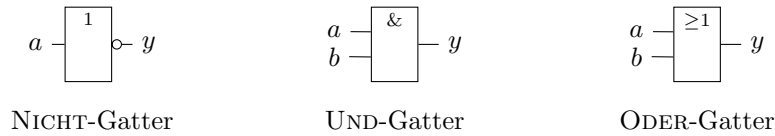


Abbildung 2.1.: Schaltsymbole der Grundverknüpfungen \neg , \wedge und \vee .

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).

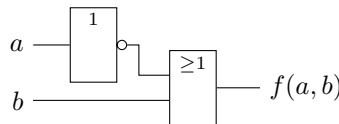


Abbildung 2.2.: Schaltung für $f(a, b) = (\bar{a} \vee b)$

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).

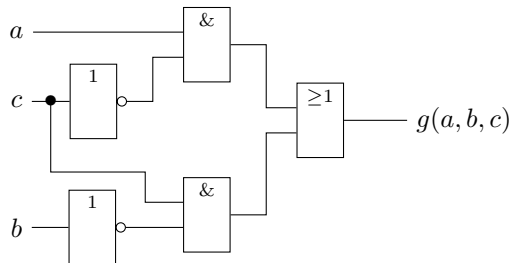


Abbildung 2.3.: Schaltung für $g(a, b, c) = ((a \wedge \bar{c}) \vee (\bar{b} \wedge c))$

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.

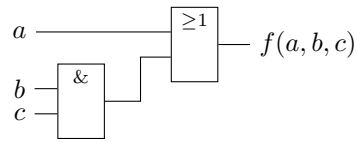
Betrachten wir die Schaltfunktionen g und h aus Beispiel 2.10.

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

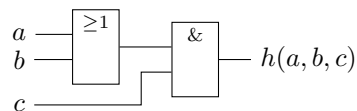
$$h(a, b, c) = (a \vee b) \wedge c$$

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:

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

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

Hier beginnt das Skript für den 2. Tag

2.4. Weitere Verknüpfungen

Außer der bereits genannten Grundverknüpfungen AND, OR und NOT, kommen in der Digitaltechnik auch die Verknüpfungen NAND, NOR, XOR und XNOR vor. Die Bedeutung der jeweiligen Verknüpfungen lässt sich an der Wahrheitstabelle sehr gut ablesen. Besinnt man sich darauf, dass 1 für *true* und 0 für *false* stehen könnte, dann ist die Belegung der Funktionswerte auch sehr intuitiv.

NICHT, NOT:	Die Ausgangsvariable y ist genau dann 1, wenn die Eingangsvariable 0 ist.
UND, AND:	Die Ausgangsvariable y ist genau dann 1, wenn alle Eingangsvariablen 0 sind.
ODER, OR:	Die Ausgangsvariable y ist genau dann 1, wenn mindestens eine der Eingangsvariablen 0 ist.
NAND:	Steht für Not And , und y ist genau dann 1, wenn $a \wedge b = 0$ ist. Also, wenn mindestens eine Eingangsvariable 0 ist.
NOR:	Steht für Not Or , und y ist genau dann 1, wenn $a \vee b = 0$ ist. Also, wenn alle Eingangsvariablen 0 sind.
XOR:	Steht für Exclusive Or , und y ist genau dann 1, wenn genau eine der Eingangsvariablen 1 ist.
XNOR:	Steht für Not Exclusive Or , und y ist genau dann 1, wenn keine oder mehr als eine Eingangsvariable 1 ist. Gibt es nur zwei Eingangsvariablen, kann man auch sagen, dass y genau dann 1 ist, wenn der Wert der beiden Eingangsvariablen identisch ist, also beide 1, oder beide 0.

2.4. Weitere Verknüpfungen

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																
NAND	<table border="1"> <tr><td>a</td><td>b</td><td>$\overline{a \wedge b}$</td></tr> <tr><td>0</td><td>0</td><td>1</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>0</td></tr> </table>	a	b	$\overline{a \wedge b}$	0	0	1	0	1	1	1	0	1	1	1	0		$y = f(a, b) = \overline{a \wedge b}$
a	b	$\overline{a \wedge b}$																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR	<table border="1"> <tr><td>a</td><td>b</td><td>$\overline{a \vee b}$</td></tr> <tr><td>0</td><td>0</td><td>1</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>0</td></tr> </table>	a	b	$\overline{a \vee b}$	0	0	1	0	1	0	1	0	0	1	1	0		$y = f(a, b) = \overline{a \vee b}$
a	b	$\overline{a \vee b}$																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR	<table border="1"> <tr><td>a</td><td>b</td><td>$a \oplus b$</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>0</td></tr> </table>	a	b	$a \oplus b$	0	0	0	0	1	1	1	0	1	1	1	0		$y = f(a, b) = a \oplus b$
a	b	$a \oplus b$																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR	<table border="1"> <tr><td>a</td><td>b</td><td>$\overline{a \oplus b}$</td></tr> <tr><td>0</td><td>0</td><td>1</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	$\overline{a \oplus b}$	0	0	1	0	1	0	1	0	0	1	1	1		$y = f(a, b) = \overline{a \oplus b}$
a	b	$\overline{a \oplus b}$																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Tabelle 2.5.: Wichtige Verknüpfungen in der Digitaltechnik

2. Schaltalgebra

2.4.1. Vollständige Operatorensysteme

Ein vollständiges Operatorensystem ist eine Menge von Operatoren, die es erlauben, jede beliebige Schaltfunktion darzustellen. Die Operatoren \wedge , \vee und $\bar{}$ bilden ein vollständiges Operatorensystem. Ferner bilden sowohl die Verknüpfung NAND, als auch die Verknüpfung NOR ein vollständiges Operatorensystem.

Operatorensystem	Negation	Konjunktion	Disjunktion
$\wedge, \vee, \bar{}$	\bar{a}	$a \wedge b$	$a \vee b$
$\bar{}$	$a\bar{a}$	$(a\bar{b})\bar{(a\bar{b})}$	$(a\bar{a})\bar{(b\bar{b})}$
∇	$a\bar{\nabla}a$	$(a\bar{\nabla}a)\bar{(b\bar{\nabla}b)}$	$(a\bar{\nabla}b)\bar{(a\bar{\nabla}b)}$

Beweis 2.14.

Negation

a	\bar{a}	$a\bar{a}$	$a\bar{\nabla}a$
0	1	1	1
1	0	0	0

Konjunktion

a	b	$a \wedge b$	$x = a\bar{b}$	$x\bar{x}$	$y = a\bar{\nabla}a$	$z = b\bar{\nabla}b$	$y\bar{\nabla}z$
0	0	0	1	0	1	1	0
0	1	0	1	0	1	0	0
1	0	0	1	0	0	1	0
1	1	1	0	1	0	0	1

Disjunktion

a	b	$a \vee b$	$x = a\bar{a}$	$y = b\bar{b}$	$x\bar{y}$	$z = (a\bar{\nabla}b)$	$z\bar{\nabla}z$
0	0	1	1	1	0	1	0
0	1	1	1	0	0	0	0
1	0	1	0	1	0	0	0
1	1	0	0	0	1	0	1

2.5. Normalformen

Für die Darstellung von Schaltfunktionen als Funktionsgleichung, gibt es standardisierte Formen, sog. Normalformen.

Definition 2.15 (Disjunktive und Konjunktive Normalform).

- DNF Die *disjunktive Normalform* (DNF) einer Schaltfunktion ist eine VerODERUNG von UND-termen, bzw. eine Disjunktion von Konjunktionstermen (vergl. Bsp. 2.16, Gleichung 2.1). Enthalten die Konjunktionsterme *jede* Eingangsvariable in negierter oder nicht negierter Form, spricht man von einer *kanonischen disjunktiven Normalform* (KDNF) (siehe Bsp. 2.16, Gl. 2.2).
- KDNF
- KNF Die *konjunktive Normalform* (KNF) einer Schaltfunktion ist eine VerUNDUNG von ODER-termen, bzw. eine Konjunktion von Disjunktionstermen (Bsp. 2.16, Gl.2.3). Kommen in allen Disjunktionstermen alle Eingangsvariablen in negierter oder nicht negierter Form vor, so handelt es sich um eine *kanonische konjunktive Normalform* (KKNF) (Bsp. 2.16, Gl.2.4).
- KKNF

Beispiel 2.16.

Betrachten wir die Schaltfunktion aus Beispiel 2.9, $g(a, b, c) = (a \vee b) \wedge (\overline{a \wedge \overline{c}}) \vee (b \wedge c)$:

Eine DNF wäre:

$$g(a, b, c) = (a \wedge c) \vee (\overline{a} \wedge b) = (ac) \vee (\overline{a}b) \quad (2.1)$$

Die KDNF ist:

$$g(a, b, c) = (\overline{a} \wedge b \wedge \overline{c}) \vee (\overline{a} \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge c) \vee (a \wedge b \wedge c) = (\overline{a}b\overline{c}) \vee (\overline{a}bc) \vee (a\overline{b}c) \vee (abc) \quad (2.2)$$

Eine KNF der Funktion wäre:

$$g(a, b, c) = (\overline{a} \vee c) \wedge (a \vee b) \quad (2.3)$$

Die KKNF lautet:

$$g(a, b, c) = (a \vee b \vee c) \wedge (a \vee b \vee \overline{c}) \wedge (\overline{a} \vee b \vee c) \wedge (\overline{a} \vee \overline{b} \vee c) \quad (2.4)$$

2.5.1. Min- und Maxterme

Definition 2.17 (Min- und Maxterme).

Ein *Minterm* ist ein Konjunktionsterm, der alle Eingangsvariablen einer Schaltfunktion enthält. Bei n Eingangsvariablen gibt es 2^n verschiedene Minterme. Jeder Minterm hat nur bei *einer einzigen* Wertekombination der Eingangsvariablen den Wert 1. Den Minterm, der für eine bestimmte Wertekombination den Wert 1 hat, erhält man, indem man die Eingangsvariablen mit UND verknüpft und die Eingangsvariablen negiert, deren Wert 0 ist. So ist der zugehörige Minterm zur Wertekombination 1 0 der Minterm $a \wedge \overline{b}$ (siehe Tab 2.6).

a	b	$m_0 = \overline{a} \wedge \overline{b}$	$m_1 = \overline{a} \wedge b$	$m_2 = a \wedge \overline{b}$	$m_3 = a \wedge b$
0	0	1	0	0	0
0	0	0	1	0	0
0	1	0	0	1	0
0	1	0	0	0	1

Tabelle 2.6.: Wertekombinationen von zwei Schaltvariablen mit möglichen Mintermen und zugehörigen Werten.

Ein *Maxterm* ist ein Disjunktionsterm, der alle Eingangsvariablen einer Schaltfunktion enthält. Bei n Eingangsvariablen gibt es 2^n verschiedene Maxterme. Jeder Maxterm hat nur bei *einer* Wertekombination der Eingangsvariablen den Wert 0, bei allen anderen ist der Wert 1. Den Maxterm, der für eine bestimmte Wertekombination den Wert 0 hat, erhält man, indem man die Eingangsvariablen mit ODER verknüpft und die Eingangsvariablen negiert, deren Wert 1 ist. So ist der zugehörige Maxterm zur Wertekombination 1 0 der Maxterm $\overline{a} \vee b$ (siehe Tab 2.7).

a	b	$M_0 = a \vee b$	$M_1 = a \vee \overline{b}$	$M_2 = \overline{a} \vee b$	$M_3 = \overline{a} \vee \overline{b}$
0	0	0	1	1	1
0	0	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

Tabelle 2.7.: Wertekombinationen von zwei Schaltvariablen mit möglichen Maxtermen und zugehörigen Werten.

Die Indizes der Min- und Maxterme ergeben sich direkt aus der Belegung der Schaltvariablen, wenn man diese als Dualzahl auffasst. $00 \rightarrow$ Index 0 und somit $m_0 = \overline{a} \wedge \overline{b}$ bzw. $M_0 = a \vee b$.

2. Schaltalgebra

Merke: Für den Minterm verknüpfe alle Eingangsvariablen mit UND (\wedge) und negiere alle, deren Wert 0 ist.

Für den Maxterm verknüpfe alle Eingangsvariablen mit ODER (\vee) und negiere alle, deren Wert 1 ist.

Man kann eine Schaltfunktion also auch durch Angabe der Min- bzw. Maxterme darstellen.

Beispiel 2.18.

Betrachten wir die Funktion $f_1(a, b) = m_0 \vee m_2$.

Minterm m_0 ist in diesem Fall $(\bar{a} \wedge \bar{b})$, Minterm m_2 ist $(a \wedge \bar{b})$ also gilt:

$$f(a, b) = m_0 \vee m_2 = (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b})$$

Achtung: Hier ist ganz wichtig, in welcher Reihenfolge die Eingangsvariablen in der Funktionsdefinition vorkommen. Die Funktion $f_2(b, a) = m_0 \vee m_2$ beschreibt nämlich eine ganz andere Funktion als $f_1(a, b)$. Der Minterm m_0 ist hier zwar auch $(\bar{b} \wedge \bar{a})$, allerdings nur, weil beide Eingangsvariablen negiert sind und in der bool'schen Algebra das Kommutativgesetz gilt. Der Minterm m_2 ist hier $(b \wedge \bar{a})$, denn b ist in dieser Funktion die höherwertige Eingangsvariable, steht weiter links und bekommt bei der Belegung $(1, 0)$ die 1 zugewiesen. Für $f_2(b, a)$ gilt also:

$$f_2(b, a) = m_0 \vee m_2 = (\bar{b} \wedge \bar{a}) \vee (b \wedge \bar{a})$$

Wie erhält man aus einer Wertetabelle oder Funktionsgleichung eine DNF bzw. KNF?

Um eine gegebene Funktion als DNF bzw. KNF angeben zu können, gibt es mehrere Möglichkeiten:

Umformen der Gleichung: Jede Funktionsgleichung kann durch Umformungen nach den Gesetzen der Schaltalgebra (Abschn. 2.3) in eine DNF, bzw. KNF gebracht werden.

Beispiel 2.19.

Die Funktion aus Beispiel 2.8 kann folgendermaßen in eine DNF umgeformt werden.

$$\begin{aligned} g(a, b, c) &= (a \vee b) \wedge (\overline{a \wedge \bar{c}}) \vee (b \wedge c) \\ &\stackrel{9}{=} (a \vee b) \wedge (\bar{a} \vee \bar{c}) \vee (b \wedge c) \\ &= (a \vee b) \wedge (\bar{a} \vee c) \vee (b \wedge c) \\ &\stackrel{4}{=} ((a \vee b) \wedge \bar{a}) \vee ((a \vee b) \wedge c) \vee (b \wedge c) \\ &\stackrel{4}{=} ((\bar{a} \wedge a) \vee (\bar{a} \wedge b)) \vee ((a \wedge c) \vee (b \wedge c)) \vee (b \wedge c) \end{aligned}$$

Dies ist bereits eine DNF. Jetzt kann sie noch ein wenig vereinfacht werden:

$$\begin{aligned} &((\bar{a} \wedge a) \vee (\bar{a} \wedge b)) \vee ((a \wedge c) \vee (b \wedge c)) \vee (b \wedge c) \\ &\stackrel{6}{=} 0 \vee (\bar{a} \wedge b) \vee (a \wedge c) \vee (b \wedge c) \vee (b \wedge c) \\ &\stackrel{7}{=} (\bar{a} \wedge b) \vee (a \wedge c) \vee (b \wedge c) \vee (b \wedge c) \\ &\stackrel{8}{=} (\bar{a} \wedge b) \vee (a \wedge c) \vee (b \wedge c) \end{aligned}$$

Diese DNF sieht zwar anders aus als die in Gleichung 2.1 angegebene, aber beide stellen die gleiche Schaltfunktion dar. Das wird anhand einer Wahrheitstabelle deutlich.

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

$$\begin{aligned}
g(a, b, c) &= (a \vee b) \wedge (\overline{a \wedge \bar{c}}) \vee (b \wedge c) \\
&\stackrel{9}{=} (a \vee b) \wedge (\bar{a} \vee \bar{\bar{c}}) \vee (b \wedge c) \\
&= (a \vee b) \wedge (\bar{a} \vee c) \vee (b \wedge c) \\
&\stackrel{4}{=} (a \vee b) \wedge (\bar{a} \vee b \vee c) \wedge (\bar{a} \vee c \vee c) \\
&= (a \vee b) \wedge (\bar{a} \vee b \vee c) \wedge (\bar{a} \vee c)
\end{aligned}$$

Dies ist bereits eine KNF. Sie sieht zwar anders aus, als die in Beispiel 2.16, Gleichung 2.3 angegebene, stellt aber die gleiche Schaltfunktion dar, wie mit einer Wahrheitstabelle schnell zu überprüfen ist.

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

Direktes Ablesen aus der Wahrheitstabelle: Liegt für die Schaltfunktion eine vollständige Wahrheitstabelle vor, dann können KDNF und KKNF direkt aus der Wahrheitstabelle abgelesen werden. In der KDNF treten genau die Minterme auf, bei denen die Ausgangsvariable y der Schaltfunktion den Wert 1 hat. In der KKNF treten genau die Maxterme auf, für die die Ausgangsvariable der Schaltfunktion den Wert 0 hat.

Beachte: Jede KDNF ist eine DNF. Die KDNF ist nur eine spezielle DNF, nämlich die, bei der die Konjunktionsterme alle Eingangsvariablen enthalten (siehe Def. 2.15). Gleiches gilt für die KKNF.

Beispiel 2.20.

Die Wahrheitstabelle für die Schaltfunktion aus Beispiel 2.8, $g(a, b, c) = (a \vee b) \wedge (\overline{a \wedge \bar{c}}) \vee (b \wedge c)$ mit den zugehörigen Min- und Maxtermen sieht wie folgt aus:

a	b	c	$y = g(a, b, c)$	Minterme	m_i	Maxterme	M_i
0	0	0	0	$\bar{a} \wedge \bar{b} \wedge \bar{c}$	m_0	$a \vee b \vee c$	M_0
0	0	1	0	$\bar{a} \wedge \bar{b} \wedge c$	m_1	$a \vee b \vee \bar{c}$	M_1
0	1	0	1	$\bar{a} \wedge b \wedge \bar{c}$	m_2	$a \vee \bar{b} \vee c$	M_2
0	1	1	1	$\bar{a} \wedge b \wedge c$	m_3	$a \vee \bar{b} \vee \bar{c}$	M_3
1	0	0	0	$a \wedge \bar{b} \wedge \bar{c}$	m_4	$\bar{a} \vee b \vee c$	M_4
1	0	1	1	$a \wedge \bar{b} \wedge c$	m_5	$\bar{a} \vee b \vee \bar{c}$	M_5
1	1	0	0	$a \wedge b \wedge \bar{c}$	m_6	$\bar{a} \vee \bar{b} \vee c$	M_6
1	1	1	1	$a \wedge b \wedge c$	m_7	$\bar{a} \vee \bar{b} \vee \bar{c}$	M_7

2. Schaltalgebra

Die aus der Tabelle abgelesene KDNF lautet:

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

Die abgelesene KKNF lautet:

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

Beachte: Die kanonische Normalform ist bis auf die Reihenfolge der Min- bzw Maxterm eindeutig.

Natürlich kann man für eine gegebene Funktionsgleichung auch zunächst die Wahrheitstabelle aufstellen und DNF, bzw. KNF dann daraus ablesen, wenn die Schaltfunktion nicht zu viele Eingangsvariablen hat, denn Wahrheitstabellen werden sehr schnell sehr groß.

2.6. KV-Diagramme

Das **Karnaugh-Veitch-Diagramm** ist eine graphische Darstellung einer Wahrheitstabelle oder Schaltfunktion. Es wurde Anfang der 50'er Jahre von Edward W. Veitch und Maurice Karnaugh entwickelt. Das Diagramm ist eine Matrix, in der jedem Feld eine Kombination der Eingangsvariablen zugeordnet ist. Dabei sind die Felder so zugeordnet, dass sich beim Übergang von einem Feld in ein benachbartes Feld, der Wert nur *einer* Eingangsvariablen ändert.

Mit Balken an Spalten und Zeilen der Matrix wird angezeigt, in welchen Feldern der Wert der Variablen 1 ist. So ist dem Feld mit Index 0 im KV-Diagramm für eine Eingangsvariable (a) (Abb.: 2.7) die Belegung 0, bzw. \bar{a} zugeordnet und dem Feld mit Index 1 die Belegung 1 bzw. a . Im KV-Diagramm für zwei Eingangsvariablen (a, b) ist Feld 0 die Belegung 00, bzw. \bar{a}, \bar{b} , Feld 1 die Belegung 01, bzw. $\bar{a}b$, Feld 2 die Belegung 10, bzw. $a\bar{b}$ und Feld 3 die Belegung 11, bzw. ab . Die Indizes der Felder im KV-Diagramm sind also identisch mit den Indizes der Min- und Maxterme, die ihnen zugeordnet sind (vergl. Tab. 2.8).

Bemerkung: Im Prinzip braucht man die Indizes nicht, wenn man verstanden hat, wie das mit den Balken an der Seite des KV-Diagramms geht. Und in den Übungen werden Sie keine Indizes in den KV-Diagrammen finden. Wir haben ein Video dazu gemacht, wie man KV-Diagramme ohne Indizes nutzt, da ich es sehr schwierig fand, das im Text zu erklären.

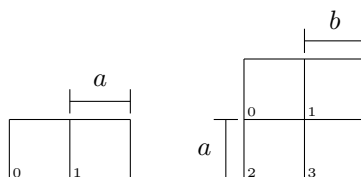


Abbildung 2.6.: KV-Diagramme für eine (a) und für zwei (a, b) Eingangsvariablen

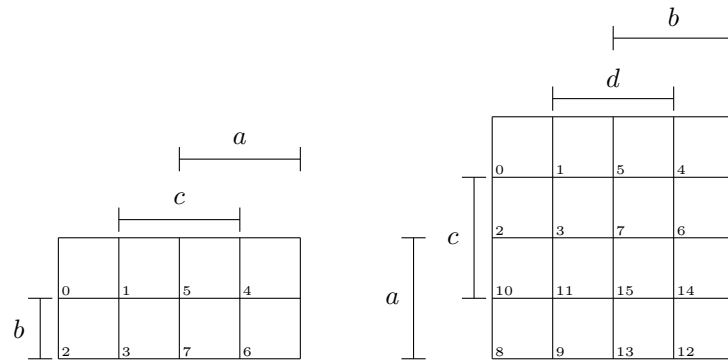


Abbildung 2.7.: KV-Diagramme für drei (a, b, c) und für vier (a, b, c, d) Eingangsvariablen

Feldindex	Dualzahl	a	b	c	d	Minterm	Maxterm
0	0000	0	0	0	0	$\bar{a} \wedge \bar{b} \wedge \bar{c} \wedge \bar{d}$	$a \vee b \vee c \vee d$
1	0001	0	0	0	1	$\bar{a} \wedge \bar{b} \wedge \bar{c} \wedge d$	$a \vee b \vee c \vee \bar{d}$
2	0010	0	0	1	0	$\bar{a} \wedge \bar{b} \wedge c \wedge \bar{d}$	$a \vee b \vee \bar{c} \vee d$
3	0011	0	0	1	1	$\bar{a} \wedge \bar{b} \wedge c \wedge d$	$a \vee b \vee \bar{c} \vee \bar{d}$
4	0100	0	1	0	0	$\bar{a} \wedge b \wedge \bar{c} \wedge \bar{d}$	$a \vee \bar{b} \vee c \vee d$
5	0101	0	1	0	1	$\bar{a} \wedge b \wedge \bar{c} \wedge d$	$a \vee \bar{b} \vee c \vee \bar{d}$
6	0110	0	1	1	0	$\bar{a} \wedge b \wedge c \wedge \bar{d}$	$a \vee \bar{b} \vee \bar{c} \vee d$
7	0111	0	1	1	1	$\bar{a} \wedge b \wedge c \wedge d$	$a \vee \bar{b} \vee \bar{c} \vee \bar{d}$
8	1000	1	0	0	0	$a \wedge \bar{b} \wedge \bar{c} \wedge \bar{d}$	$\bar{a} \vee b \vee c \vee d$
9	1001	1	0	0	1	$a \wedge \bar{b} \wedge \bar{c} \wedge d$	$\bar{a} \vee b \vee c \vee \bar{d}$
10	1010	1	0	1	0	$a \wedge \bar{b} \wedge c \wedge \bar{d}$	$\bar{a} \vee b \vee \bar{c} \vee d$
11	1011	1	0	1	1	$a \wedge \bar{b} \wedge c \wedge d$	$\bar{a} \vee b \vee \bar{c} \vee \bar{d}$
12	1100	1	1	0	0	$a \wedge b \wedge \bar{c} \wedge \bar{d}$	$\bar{a} \vee \bar{b} \vee c \vee d$
13	1101	1	1	0	1	$a \wedge b \wedge \bar{c} \wedge d$	$\bar{a} \vee \bar{b} \vee c \vee \bar{d}$
14	1110	1	1	1	0	$a \wedge b \wedge c \wedge \bar{d}$	$\bar{a} \vee \bar{b} \vee \bar{c} \vee d$
15	1111	1	1	1	1	$a \wedge b \wedge c \wedge d$	$\bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d}$

Tabelle 2.8.: Feldindizes und zugeordnete Belegung der Eingangsvariablen sowie Min- und Maxterme bei KV-Diagrammen für vier Eingangsvariablen.

Soll eine gegebene Schaltfunktion, die in KDNF vorliegt, in einem KV-Diagramm dargestellt werden, so wird in die Felder mit den Indizes, deren zugeordnete Minterme in der Schaltfunktion auftauchen, eine 1 geschrieben. In alle Felder, deren zugeordnete Minterme nicht in der gegebenen KDNF auftauchen, wird eine 0 eingetragen.

Beispiel 2.21.

Betrachten wir wieder die Funktion aus Beispiel 2.8 in KDNF (vergl. Bsp. 2.20),

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

Die Funktion hat drei Eingangsvariablen und die aufgeführten Minterme entsprechen den Belegungen $(0, 1, 0)$, $(0, 1, 1)$, $(1, 0, 1)$ und $(1, 1, 1)$. Demnach erhalten die Felder mit Indizes 2, 3, 5 und 7 eine 1, in alle anderen Felder muss eine 0 eingetragen werden.

2. Schaltalgebra

$g(a, b, c)$

	----- a			
	----- c			
	0	0	1	0
----- b	1	1	1	0
	0	1	5	4
	2	3	7	6

Liegt eine gegebene Schaltfunktion in KKNF vor, so muss in alle Felder, deren zugeordnete Maxterme in der KKNF auftauchen, eine 0 eingetragen werden. In alle anderen Felder muss eine 1 eingetragen werden.

Beispiel 2.22.

Betrachten wir die vierstellige Funktion $f(a, b, c, d)$ in KKNF:

$$f(a, b, c, d) = (a \vee b \vee \bar{c} \vee \bar{d}) \wedge (a \vee \bar{b} \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee b \vee \bar{c} \vee d) \wedge (\bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d})$$

Die Maxterme entsprechen den Belegungen $(0, 0, 1, 1)$, $(0, 1, 1, 1)$, $(1, 0, 1, 0)$ und $(1, 1, 1, 1)$. Dementsprechend müssen in die Felder mit Indizes 3, 7, 10 und 15 eine 0 und in alle anderen Felder eine 1 eingetragen werden.

$f(a, b, c, d)$

		----- b			
		----- d			
		1	1	1	1
	----- c	1	0	0	1
		0	1	0	1
----- a		1	1	1	1
		0	1	5	4
		2	3	7	6
		8	9	13	12

Hier beginnt das Skript für den 3. Tag

2.7. Minimierung von Schaltfunktionen

Schaltfunktionen, die in kanonischer Normalform oder als Wahrheitstabelle gegeben sind, enthalten oft redundante Terme und können vereinfacht werden.

Beispiel 2.23.

Die Funktionsgleichung

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

Kann vereinfacht werden zu

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

Mit Vereinfachung ist hier gemeint, dass die Funktion mit möglichst wenig Verknüpfungen und möglichst wenig Variablen dargestellt wird. Eine vereinfachte, oder *minimierte* Funktion lässt sich mit weniger Bauteilen realisieren.

Beispiel 2.24.

Betrachten wir die Funktion $f(a, b, c)$ aus Beispiel 2.23. Abbildung 2.8 zeigt die KDNF der Funktion als Schaltung realisiert.

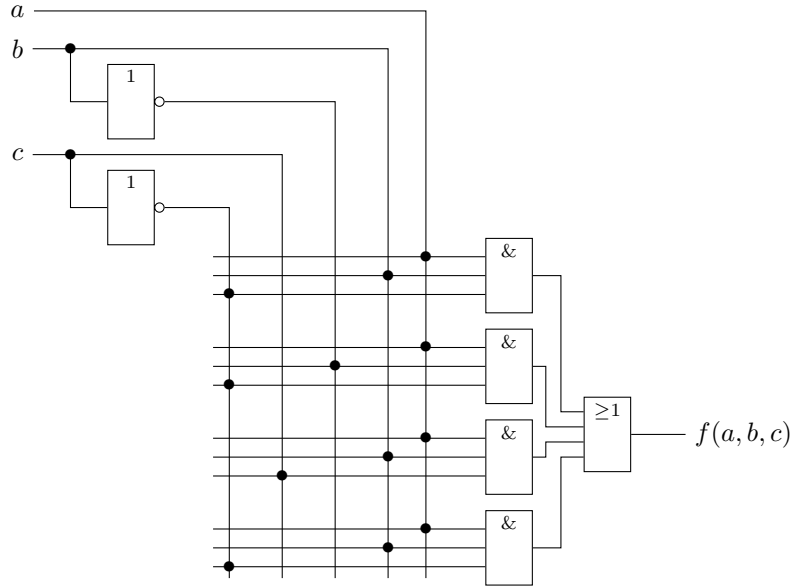


Abbildung 2.8.: Schaltung für $f(a, b, c) = (a \wedge b \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge \bar{c}) \vee (a \wedge b \wedge c) \vee (a \wedge \bar{b} \wedge c)$.

Abbildung 2.9 zeigt die Schaltung der minimierten Funktion.

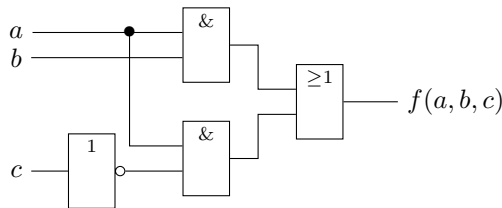


Abbildung 2.9.: Schaltung für $f(a, b, c) = (a \wedge \bar{c}) \vee (a \wedge b)$

Es gibt verschiedene Verfahren um Schaltungen zu minimieren. Die Grundlage aller Verfahren sind die Resolutionsregeln (siehe Kap. 2.3).

$$\begin{aligned}
 (a \wedge b) \vee (a \wedge \bar{b}) &= a \wedge (b \vee \bar{b}) && \text{Distributivgesetz} \\
 &= a \wedge 1 && \text{Inverse Elemente} \\
 &= a && \text{Reduktionsgesetze}
 \end{aligned}$$

$$\begin{aligned}
 (a \vee b) \wedge (a \vee \bar{b}) &= a \vee (b \wedge \bar{b}) && \text{Distributivgesetz} \\
 &= a \vee 0 && \text{Inverse Elemente} \\
 &= a && \text{Reduktionsgesetze}
 \end{aligned}$$

Das heißt, unterscheiden sich UND- bzw. ODER-Terme nur in der Negation einer einzigen Variablen, so können sie zu einem Term verschmolzen werden, bei dem diese Variable wegelassen werden kann.

2. Schaltalgebra

2.7.1. Minimierung mit den Gesetzen der Schaltalgebra

Beispiel 2.25.

Bei der KDNF der Funktion

$$f(a, b, c, d) = (\bar{a}bcd) \vee (abcd) \vee (ab\bar{c}d) \vee (\bar{a}bcd) \vee (\bar{a}\bar{b}cd)$$

können z.B. die Terme 1 und 2, die Terme 2 und 3, sowie 4 und 5 zusammengefasst werden zu

$$f(a, b, c, d) = (acd) \vee (abd) \vee (\bar{a}bd)$$

Nun können noch die letzten beiden Terme zusammen gefasst werden, und man erhält

$$f(a, b, c) = (acd) \vee (bd)$$

2.7.2. Minimierung mit KV-Diagrammen

Das Verfahren mit KV-Diagrammen ist ein graphisches Verfahren. Im KV-Diagramm ist jedem Feld ein Min- bzw. Maxterm zugeordnet, und zwar so, dass sich Min- bzw. Maxterme benachbarter Felder nur in der Belegung einer Variablen unterscheiden (vergl. 2.6). Daher können Felder mit benachbarten 1-en bzw. 0-en zusammengefasst werden.

Beispiel 2.26.

Betrachten wir die Funktion aus Beispiel 2.25. In ein KV-Diagramm eingetragen, sieht die Funktion wie folgt aus:

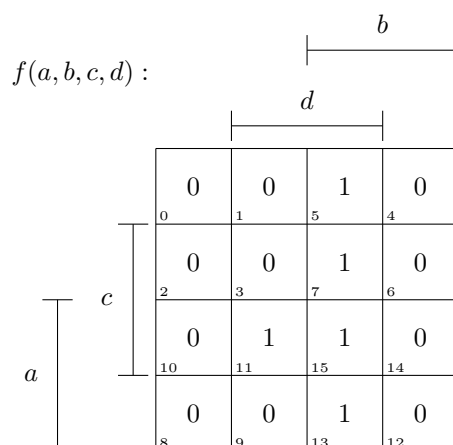
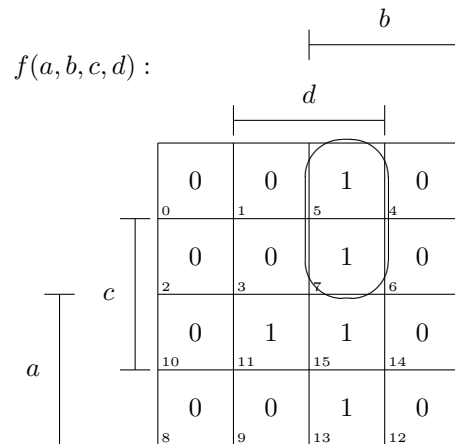


Abbildung 2.10.: KV-Diagramm für Funktion $f(a, b, c, d) = (\bar{a}bcd) \vee (abcd) \vee (ab\bar{c}d) \vee (\bar{a}bcd) \vee (\bar{a}\bar{b}cd)$

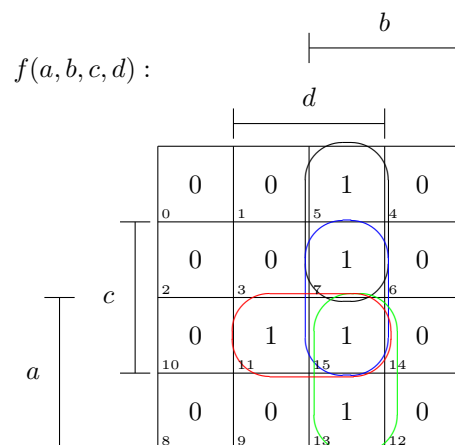
Nun können z.B. die 1-en der Felder mit Indizes 5 und 7 zusammengefasst werden.

2.7. Minimierung von Schaltfunktionen



Dem zusammengefassten Block ist der Term $(\bar{a} \wedge b \wedge d)$ oder $(\bar{a}bd)$ zugeordnet. Wenn wir uns zurück erinnern an Beispiel 2.25, entsprechen die Felder mit Indizes 5 und 7 den Mintermen $(\bar{a}b\bar{c}d)$ und $(\bar{a}bcd)$, welche zu $(\bar{a}bd)$ zusammengefasst wurden, da sie sich lediglich in der Belegung der Variablen c unterscheiden.

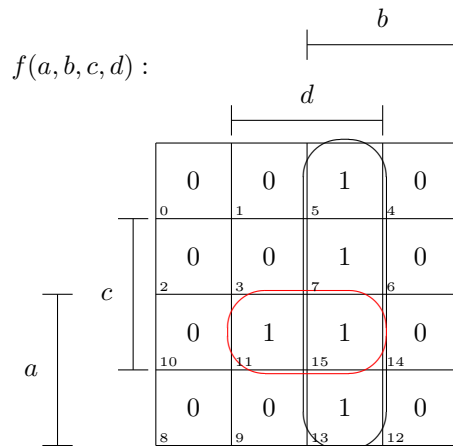
Es können weiterhin noch die Felder mit den Indizes 7 und 15 (blau), 15 und 13 (grün), sowie 11 und 15 (rot) zusammen gefasst werden



Dies entspräche den Termen (cbd) , (abd) und (acd) .

Allerdings können wir die Felder mit Indizes 5, 7, 15 und 13 auch in einem Block zusammenfassen, da sie alle benachbart sind.

2. Schaltalgebra



Der neue Block wird durch den Term (bd) abgedeckt. Zusammen mit dem anderen Block (acd) , ergibt sich nun für die minimierte Funktionsgleichung

$$f(a, b, c) = (bd) \vee (acd)$$

Das entspricht dem Ergebnis, welches wir bei der Minimierung durch Anwendung der Reduktionsregeln erhalten haben.

Bemerkung: Das muss nicht immer so sein. Die minimierte DNF muss nicht eindeutig sein. Bei manchen Funktionen gibt es mehrere Möglichkeiten der Minimierung, wie wir sehen werden.

Definition 2.27 (Implikant, Primimplikant).

Implikant

Ein UND-Term m einer Bool'schen Funktion f , heißt *Implikant*, wenn er Teil der Funktion f ist. D.h. wenn gilt: falls $m(x) = 1$ dann $f(x) = 1$ für alle möglichen Belegungen der Eingangsvariablen $x = (a, b, c, \dots)$. Der kleinste Implikant einer Funktion f ist ein Minterm, der in der KDNF vorkommt. Für das KV-Diagramm heißt das, dass ein kleinster Implikant aus genau einem Feld besteht in dem eine 1 steht.

Primimplikant

Ein *Primimplikant* von f ist ein Implikant m von f , der nicht weiter verkürzt werden kann. D.h. es ist ein Implikant minimaler Länger. Im KV-Diagramm bedeutet das, dass keinen größeren Block von 1-en gibt, der m enthält.

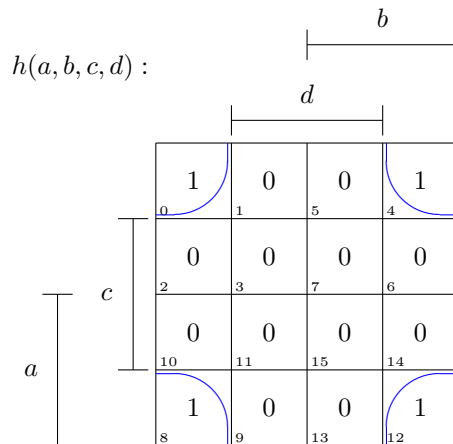
In Beispiel 2.26 sind $(\bar{a}bd)$, (cbd) , (abd) , (acd) und (bd) Implikanten, (acd) und (bd) sind Primimplikanten. $(\bar{a}bd)$, (cbd) und (abd) sind keine Primimplikanten, da (bd) größer ist und die anderen Implikanten enthält.

Je größer die Fläche wird, die ein Implikant im KV-Diagramm abdeckt, desto weniger Variablen tauchen im Implikanten auf. Die Blockgröße wächst in Zweierpotenzen. Es dürfen immer nur Blöcke von 2, 4, oder 8 usw. Feldern zusammengefasst werden.

Definition 2.28 (Kernprimimplikant).

Ein *Kernprimimplikant* ist ein Primimplikant, der eine 1 enthält, die von keinem andern Primimplikanten abgedeckt wird.

2. Schaltalgebra

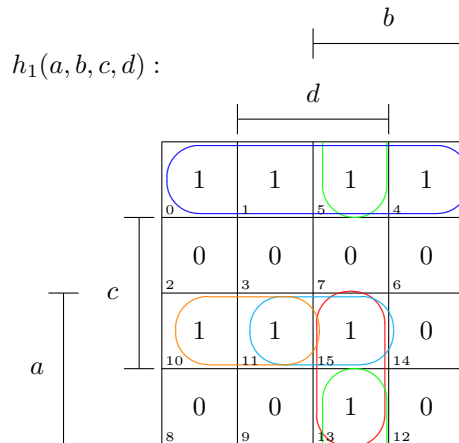


Aufstellung einer minimalen DNF

Um aus dem KV-Diagramm die minimale DNF aufzustellen, werden zunächst graphisch alle Primimplikanten eingezeichnet. Die minimale DNF muss alle 1-Terme der Schaltfunktion überdecken. Also sind alle Kernprimimplikanten in der minimalen DNF vertreten. Die absolut eliminierbaren Primimplikanten benötigt man nicht, denn ihre 1-en sind bereits von den Kernprimimplikanten überdeckt. Aus den relativ eliminierbaren Primimplikanten wählt man dann die aus, die die restlichen 1-en überdecken.

Beispiel 2.32.

Im KV-Diagramm der Funktion $h_1(a, b, c, d)$ sind bereits alle Primimplikanten eingezeichnet:



Kernprimimplikanten sind $(\bar{a}\bar{c})$, und $(\bar{a}\bar{b}c)$. Alle anderen sind relativ eliminierbare Primimplikanten. Um nun die fehlenden 1-en in den Feldern 13 und 15 abzudecken, wählen wir Primimplikant (abd) , denn das ist kleiner, als die Alternative, die beiden Felder mit den beiden anderen Primimplikanten $(b\bar{c}d)$ und (acd) abzudecken.

Nun werden die gewählten Primimplikanten zu einer DNF zusammengefügt. Die minimale DNF lautet also

$$h_1(a, b, c, d) = (\bar{a}\bar{c}) \vee (\bar{a}\bar{b}c) \vee (abd)$$

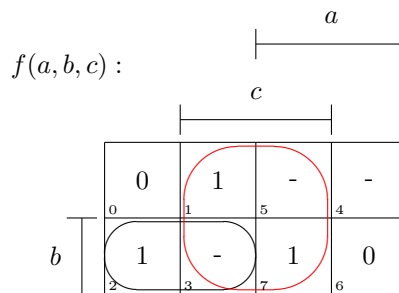
Bemerkung: Das Aufstellen der minimalen KNF funktioniert nach demselben Prinzip. Statt mit Primimplikaten Mintermen wird mit Maxtermen gearbeitet und die Blöcke heißen nicht Implikanten sondern *Primimplikate*.

2.8. Unvollständig spezifizierte Funktionen

Es kommt vor, dass Schaltfunktionen nicht vollständig spezifiziert sind. In dem Fall können manche Funktionswerte beliebig gewählt werden. Im KV-Diagramm wird das mit $-$ oder d (don't care) markiert. Die don't care-Terme können zur Minimierung der Funktion genutzt werden, müssen aber nicht überdeckt werden, wenn sie die Funktion "vergrößern" würden. Man kann sich also aussuchen, ob die Funktion zu an der Stelle zu 1 oder zu 0 auswerten soll.

Beispiel 2.33.

Funktion $f(a, b, c)$ ist durch folgendes KV-Diagramm gegeben.



Die Primimplikanten können unter Einbeziehung der don't care-Felder so eingezeichnet werden, dass sie möglichst groß werden.

So kommt man auf die minimierte Form

$$f(a, b, c) = (\bar{a}b) \vee c$$

Ohne die Verwendung der don't care-Terme, hätte man folgende minimierte Form gefunden:

$$f(a, b, c) = (\bar{a}b\bar{c}) \vee (\bar{a}bc) \vee (abc)$$

2.9. Schaltungsentwurf

Um nun eine Schaltung für ein konkretes Problem zu entwerfen, müssen verschiedene Schritte durchlaufen werden.

1. Problemanalyse und Aufstellung einer Wahrheitstabelle für die zu implementierende Funktion
2. Minimierung der Funktion, z.B. mit Hilfe von KV-Diagramm
3. Implementation der Schaltung

Beispiel 2.34.

Wenn wir z.B. eine Schaltung $P(a, b, c)$ entwerfen wollen, die immer dann eine 1 ausgibt wenn die Eingabebelegung ein Palindrom² ist.

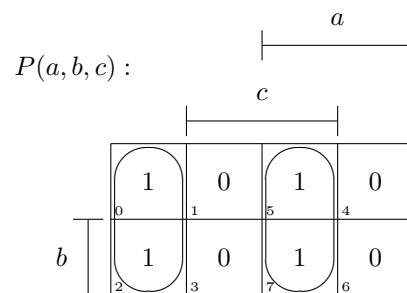
Wie sieht die Wahrheitstabelle für solch eine Funktion aus?

²Ein Palindrom ist ein Wort, das von rechts und von links gelesen, gleich ist. z.B. die Wörter Rentner oder Ebbe

2. Schaltalgebra

a	b	c	$P(a, b, c)$
0	0	0	1
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	1

Das zugehörige KV-Diagramm sieht so aus:



Die minimierte DNF lautet dann:

$$P(a, b, c) = (\bar{a}\bar{c}) \vee (ac)$$

Diese Funktion könnten wir nun mit zwei NICHT-Gattern, zwei UND-Gattern und einem ODER-Gatter realisieren.

