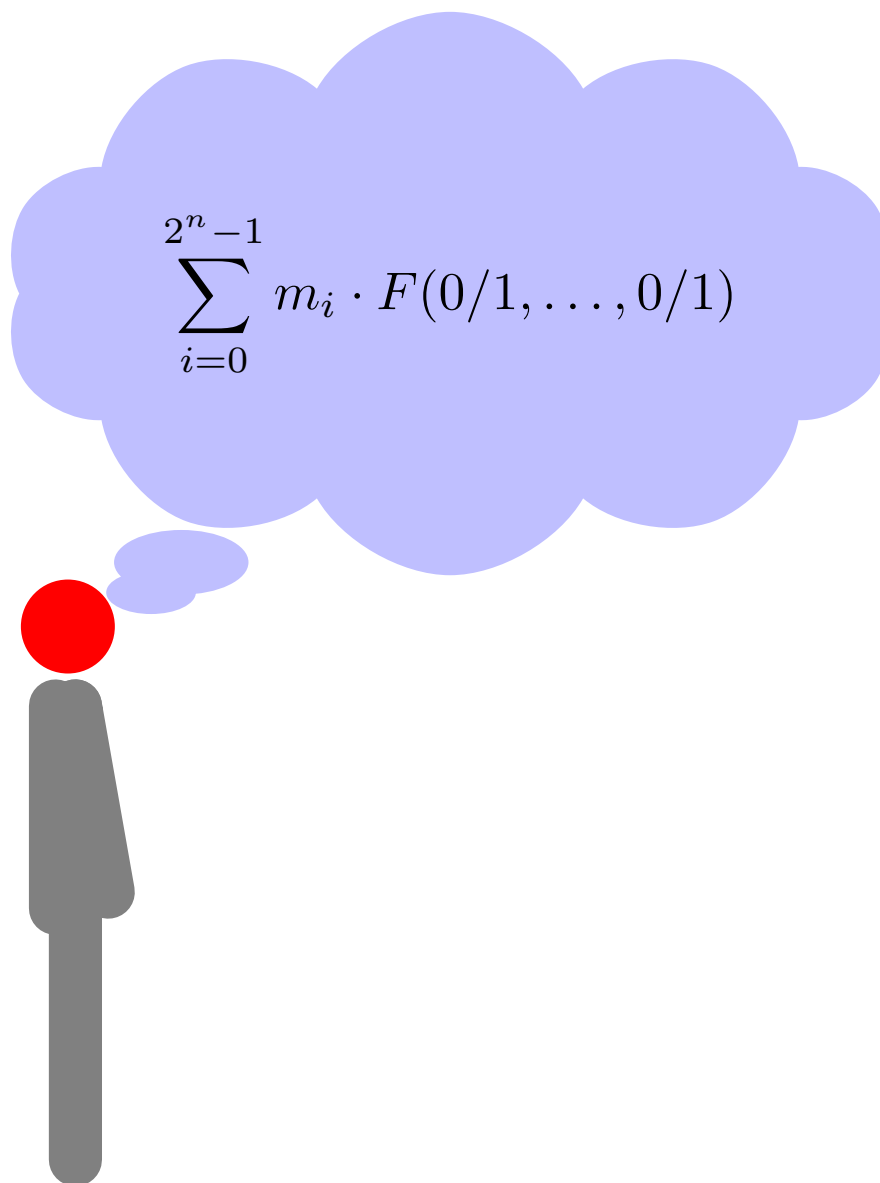


Aufgabensammlung - Teil 2



1 Theoretische Grundlagen

1.1 Boolesche Operation (Wiederholung)

Die Booleschen Formeln setzen sich aus Variablen und logischen Operatoren zusammen. Für die Aussage ‚falsch‘ und ‚wahr‘ setzen wir nun die binäre Darstellung ‚0_b‘ und ‚1_b‘ (der Index b macht deutlich, dass es sich um eine Binärzahl handelt). Zur Vereinfachung schreiben wir im Kontext dieses Skriptes ‚0_b = 0‘ und ‚1_b = 1‘. Variablen werden mit Kleinbuchstaben (*a, b, ..., x, y, z*) beschrieben. Aussagen denen die Aussage ‚falsch‘ oder ‚wahr‘ zugeordnet werden können, sind mit Großbuchstaben bezeichnet.

Tabelle 1: Einstellige und zweistellige Boolesche Funktionen

<i>x</i>	0	0	1	1	verbale Form	Boolesche Darstellung	Algebraische Darstellung	Bezeichnung
<i>y</i>	0	1	0	1				
<i>x o₁ y</i>	0	0	0	0	konstant 0	$\neg x \wedge x$	$\bar{x} \cdot x$	Nullfunktion
<i>x o₂ y</i>	0	0	0	1	<i>x</i> und <i>y</i>	$x \wedge y$	$x \cdot y$	Konjunktion
<i>x o₃ y</i>	0	0	1	0	nicht <i>y</i> , aber <i>x</i>	$x \wedge \neg y$	$x \cdot \bar{y}$	Inhibition
<i>x o₄ y</i>	0	0	1	1	identisch <i>x</i>	x	x	Identität
<i>x o₅ y</i>	0	1	0	0	nicht <i>x</i> , aber <i>y</i>	$\neg x \wedge y$	$\bar{x} \cdot y$	Inhibition
<i>x o₆ y</i>	0	1	0	1	identisch <i>y</i>	y	y	Identität
<i>x o₇ y</i>	0	1	1	0	<i>x</i> ungleich <i>y</i>	$x \not\leftrightarrow y$	$x \oplus y$	Antivalenz
<i>x o₈ y</i>	0	1	1	1	<i>x</i> oder <i>y</i>	$x \vee y$	$x + y$	Disjunktion
<i>x o₉ y</i>	1	0	0	0	nicht (<i>x</i> oder <i>y</i>)	$\neg(x \vee y)$	$\overline{(x + y)}$	Neg. Disjunk.
<i>x o₁₀ y</i>	1	0	0	1	<i>x</i> gleich <i>y</i>	$x \leftrightarrow y$	$\overline{x \oplus y}$	Äquivalenz
<i>x o₁₁ y</i>	1	0	1	0	nicht <i>y</i>	$\neg y$	\bar{y}	Negation
<i>x o₁₂ y</i>	1	1	0	1	wenn <i>x</i> dann <i>y</i>	$x \rightarrow y$	$x \leq y$	Implikation
<i>x o₁₃ y</i>	1	1	0	0	nicht <i>x</i>	$\neg x$	\bar{x}	Negation
<i>x o₁₄ y</i>	1	0	1	1	wenn <i>y</i> dann <i>x</i>	$y \rightarrow x$	$y \leq x$	Implikation
<i>x o₁₅ y</i>	1	1	1	0	nicht(<i>x</i> und <i>y</i>)	$\neg(x \wedge y)$	$\overline{(x \cdot y)}$	Neg. Konjuk.
<i>x o₁₆ y</i>	1	1	1	1	konstant 1	1	1	Tautologie

- Warum einstellige Binärfunktion? ⇒ Nur von einer Variablen abhängig!
- Warum zweistellige Binärfunktion? ⇒ Von mindestens zwei Variablen abhängig!

1.2 Vollständige Normalformen und die Entw. Boolescher Funktionen

Jede Boolesche Funktion kann durch verschiedene äquivalente Boolesche Terme dargestellt werden. Unter **Normalformen** versteht man die einheitliche normierte Darstellung, sowie das Verfahren zum Erreichen dieser Darstellung. Man unterscheidet **konjunktive** und **disjunktive** Normalformen. (KNF und DNF sind bis auf Vertauschungen eindeutig und lassen sich direkt aus den Funktionstabellen ablesen.) Normalformen werden zum Schaltkreisentwurf eingesetzt und sind dazu unerlässlich! (Boolesche Funktionen, die in KNF oder DNF vorliegen, können direkt in ein zweistufiges Schaltnetz überführt werden.)

Für die Programmierung von FPGAs oder PALs sowie in der Chip-Entwicklung (ASIC-Entwurf) ist die Darstellung einer logischen Verknüpfung dann vorteilhaft, wenn somit minimale Kosten (im Sinne von Fläche, Laufzeit, Leistungsaufnahme) erzielt werden.

Schreibweise: Es ist x eine Variable und σ sei eine Konstante aus $\{0, 1\}$. Es gelte:

$$x^\sigma = \sigma \wedge x \vee \neg\sigma \wedge \neg x = \begin{cases} x, & \text{falls } \sigma = 1 \\ \neg x, & \text{falls } \sigma = 0 \end{cases}$$

Es gilt $x^\sigma = 1$ genau dann, wenn $x = \sigma$

Satz: Es sei $f(x_1, \dots, x_n)$ eine n -stellige Boolesche Funktion und m eine Zahl mit $1 \leq m \leq n$. Dann kann man f in der folgenden Form darstellen:

$$f(x_1, \dots, x_n) = \bigvee_{\sigma_1 \dots \sigma_m \in \{0,1\}} (x_1^{\sigma_1} \wedge \dots \wedge x_m^{\sigma_m}) f(\sigma_1, \dots, \sigma_m, x_{m+1}, \dots, x_n)$$

Konjunktive Normalform, KNF

Eine Formel der Aussagenlogik ist in „konjunktiver Normalform“, wenn sie eine **Konjunktion** (UND) von **Disjunktionstermen** (ODER) ist. Disjunktionsterme sind dabei Disjunktionen (ODER-Verknüpfung) von Literalen. Literale sind nicht-negierte oder negierte Variablen. Eine Formel in KNF hat also die Form:

$$f = \bigwedge_i \bigvee_j (\neg)x_{ij}$$

Vorgehensweise:

1. Aufstellen der Funktionstabelle.

$(i)_{10}$	x	y	z	$f(x, y, z)$	Klausel
0	0	0	0	0	$x \vee y \vee z$
1	0	0	1	1	–
2	0	1	0	0	$x \vee \neg y \vee z$
3	0	1	1	1	–
4	1	0	0	0	$\neg x \vee y \vee z$
5	1	0	1	0	$\neg x \vee y \vee \neg z$
6	1	1	0	1	–
7	1	1	1	1	–

2. Zu jeder Zeile mit dem Ergebnis 0: Aufstellung einer Disjunktion (OR-Verknüpfung) der beteiligten Variablen, d.h. Negation der Variablen, die mit einer 1 belegt sind; keine Negation der mit 0 belegten Variablen
3. AND-Verknüpfung der „0“-Zeilen-Ausdrücke: Die Klauseln aus den Zeilen 0, 2, 4 und 5 werden nun zur KNF über eine UND-Verknüpfung zusammengefasst. Die konjunktive Normalform ergibt sich somit als:

KNF: $f(x, y, z) = (x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z)$

Disjunktive Normalform, DNF

Eine Formel der Aussagenlogik ist in „disjunktiver Normalform“, wenn sie eine **Disjunktion** (ODER) von **Konjunktionstermen** (UND) ist. Ein Konjunktionsterm wird ausschließlich durch die konjunktive Verknüpfung von Literalen gebildet. Literale sind dabei nichtnegierte oder negierte Variablen. Eine Formel in DNF hat also die Form:

$$f = \bigvee_i \bigwedge_j (\neg)x_{ij}$$

Vorgehensweise:

1. Aufstellen der Wahrheitstabelle.

$(i)_{10}$	x	y	z	$f(x, y, z)$	Klausel
0	0	0	0	0	–
1	0	0	1	1	$\neg x \wedge \neg y \wedge z$
2	0	1	0	0	–
3	0	1	1	1	$\neg x \wedge y \wedge z$
4	1	0	0	0	–
5	1	0	1	0	–
6	1	1	0	1	$x \wedge y \wedge \neg z$
7	1	1	1	1	$x \wedge y \wedge z$

2. Zu jeder Zeile mit dem Ergebnis 1: Aufstellung einer Konjunktion (AND-Verknüpfung) der beteiligten Variablen, d.h. Negation der Variablen, die mit einer 0 belegt sind; keine Negation der mit 1 belegten Variablen
3. OR-Verknüpfung der „1“-Zeilen-Ausdrücke: Die Klauseln aus den Zeilen 1, 3, 6 und 7 werden nun zur KNF über eine UND-Verknüpfung zusammengefasst. Die konjunktive Normalform ergibt sich somit als:

$$\text{DNF: } f(x, y, z) = (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge y \wedge z)$$

Ein Boolescher Term befindet sich in **konjunktiver Minimalform (KMF)**, wenn er als Konjunktion von Disjunktionen vorliegt und sowohl die Anzahl der Disjunktionen als auch die Anzahl der auftretenden Variablen in jedem Disjunktionsterm minimal ist.

Ein Boolescher Term befindet sich in **disjunktiver Minimalform (DMF)**, wenn er als Disjunktion von Konjunktionen vorliegt und sowohl die Anzahl der Konjunktionen als auch die Anzahl der auftretenden Variablen in jedem Konjunktionsterm minimal ist.

Beispiele zur Minimalform:

$$f_4 = \neg x \wedge (\neg x \vee z) \wedge (y \vee z) \text{ befindet sich nicht in KNF und nicht in KMF}$$

$$\Rightarrow f_4 = \neg x \wedge (y \vee z) \text{ befindet sich nun in KMF}$$

1.3 Minterm

Ein Minterm $m_i(x_1, x_2, \dots, x_n)$ ist eine Funktion von n Variablen, deren Wahrheitstabelle eine 1 in Zeile i , $0 \leq i \leq 2^n - 1$, und eine 0 in allen anderen Zeilen hat.

Für ein gegebenes n existieren 2^n Minterme. Realisierung eines Minterms durch ein n -AND und optionale Inverter am Eingang, Beispiel für $n=2$:

$(i)_{10}$	x_1	x_2	m_0	m_1	m_2	m_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1
			$\neg x_1 \wedge \neg x_2$	$\neg x_1 \wedge x_2$	$x_1 \wedge \neg x_2$	$x_1 \wedge x_2$

Summe von Mintermen

$(i)_{10}$	x_1	x_2	x_3	f	$f =$
0	0	0	1	1	m_0
1	0	0	0	1	m_1
2	0	1	1	1	m_2
3	0	1	0	1	m_3
4	1	0	1	0	m_4
5	1	0	0	0	m_5
6	1	1	1	0	m_6
7	1	1	0	1	m_7

Es folgt die Summe der Minterme m_0, m_1, m_2, m_3 und m_7 :

$$f(x_0, x_1, x_2) = m_0 + m_1 + m_2 + m_3 + m_7 = \bar{x}_0\bar{x}_1\bar{x}_1 + \bar{x}_0\bar{x}_1x_2 + \bar{x}_0x_1\bar{x}_2 + \bar{x}_0x_1x_2 + x_0x_1x_2$$

Disjunktive Normalform und Minterme - Die disjunktive Normalform ist die ODER-Verknüpfung aller Minterme (UND-verknüpfte Teilterme) und wird daher auch Produktterm, bzw. Disjunktion von Produkttermen (Sum (+) Of Products (\cdot), SOP) genannt.

Anzuwenden bei Funktionen mit weniger Einsen als Nullen!

Ein Minterm ist eine Konjunktion, in der alle Eingangsvariablen vorkommen.

Jede Funktion $f(x)$ von n Variablen, deren Wahrheitstabelle Einsen in den Zeilen x, y, \dots, k hat und Nullen sonst, lässt sich als Summe von Mintermen darstellen (Mintermnormalform):

$$f(x) = m_a(x) + m_b(x) + \dots + m_k(x)$$

Dabei entspricht jeder Minterm genau einer 1 in der Ergebnisspalte der Wahrheitstabelle von $f(x)$.

1.4 Maxterm

Ein Maxterm $M_i(x_1, x_2, \dots, x_n)$ ist eine Funktion von n Variablen, deren Wahrheitstabelle eine 0 in Zeile i , $0 \leq i \leq 2^n - 1$, und eine 1 in allen anderen Zeilen hat.

Für ein gegebenes n existieren 2^n Maxterme. Realisierung eines Maxterms durch ein n -OR und optionale Inverter am Eingang, Beispiel für $n=2$:

$(i)_{10}$	x_1	x_2	M_0	M_1	M_2	M_3
0	0	0	0	1	1	1
1	0	1	1	0	1	1
2	1	0	1	1	0	1
3	1	1	1	1	1	0
			$x_1 \vee x_2$	$x_1 \vee \neg x_2$	$\neg x_1 \vee x_2$	$\neg x_1 \vee \neg x_2$

Produkt von Maxtermen

$(i)_{10}$	x_1	x_2	x_3	f	$f =$
0	0	0	0	1	M_0
1	0	0	1	1	M_1
2	0	1	0	1	M_2
3	0	1	1	1	M_3
4	1	0	0	0	M_4
5	1	0	1	0	M_5
6	1	1	0	0	M_6
7	1	1	1	1	M_7

Es folgt das Produkt der Maxterme M_4, M_5 und M_6 :

$$f(x_0, x_1, x_2) = M_4 \cdot M_5 \cdot M_6 = (\bar{x}_0 + x_1 + x_2) \cdot (\bar{x}_0 + x_1 + \bar{x}_2) \cdot (\bar{x}_0 + \bar{x}_1 + x_2)$$

Konjunktive Normalform und Maxterme - Die konjunktive Normalform ist die UND-Verknüpfung aller Maxterme (ODER-verknüpfte Teilterme) und wird daher auch Summenterm, bzw. Konjunktion von Summentermen (Product (·) Of Sums (+), POS) genannt.

Anzuwenden bei Funktionen mit weniger Nullen als Einsen!

Ein Maxterm ist eine Disjunktion, in der alle Eingangsvariablen vorkommen.

Jede Funktion $f(x)$ von n Variablen, deren Wahrheitstabelle Nullen in den Zeilen x, y, \dots, k hat und Einsen sonst, lässt sich als Produkt von Maxtermen darstellen (Maxtermnormalform):

$$f(x) = M_a(x) \cdot M_b(x) \cdot \dots \cdot M_k(x)$$

Dabei entspricht jeder Maxterm genau einer 0 in der Ergebnisspalte der Wahrheitstabelle von $f(x)$.

1.5 Minterme oder Maxterme

Zwischen **Maxtermen** und **Mintermen** besteht nach de Morgan die Beziehung:

$$M_i = \neg m_i$$

Bei der Darstellung von Funktionen ist es üblich, die ,1-Terme, niederzuschreiben, weshalb die disjunktive Normalform meist bevorzugt wird.

$(i)_{10}$	x	y	z	Minterme	Maxterme
0	0	0	0	$\neg x \wedge \neg y \wedge \neg z = m_0$	$x \vee y \vee z = M_0$
1	0	0	1	$\neg x \wedge \neg y \wedge z = m_1$	$x \vee y \vee \neg z = M_1$
2	0	1	0	$\neg x \wedge y \wedge \neg z = m_2$	$x \vee \neg y \vee z = M_2$
3	0	1	1	$\neg x \wedge y \wedge z = m_3$	$x \vee \neg y \vee \neg z = M_3$
4	1	0	0	$x \wedge \neg y \wedge \neg z = m_4$	$\neg x \vee y \vee z = M_4$
5	1	0	1	$x \wedge \neg y \wedge z = m_5$	$\neg x \vee y \vee \neg z = M_5$
6	1	1	0	$x \wedge y \wedge \neg z = m_6$	$\neg x \vee \neg y \vee z = M_6$
7	1	1	1	$x \wedge y \wedge z = m_7$	$\neg x \vee \neg y \vee \neg z = M_7$

Will man jedoch eine Funktion in konjunktiver Normalform (Maxterme) darstellen, so muss man berücksichtigen, dass der Maxterm eine Eins-Fläche beschreibt und nur an einer Stelle den Wert ,0‘ annimmt. Maxterme werden konjunktiv verknüpft, so dass im **Karnaugh-Veitch-Diagramm** der gesamten Funktion nur an den Stellen eine „1“ verbleibt, die von allen beteiligten Maxtermen überdeckt werden.

1.6 Karnaugh-Diagramm

Das Karnaugh-Diagramme dient der graphischen Darstellung und der Vereinfachung von Normalformen. Die Wahrheitswerte der Wahrheitstabelle werden in die Matrix des Karnaugh-Diagramms eingetragen. Dabei ist jeder Zeile eine definierte Position innerhalb des Karnaugh-Diagramms zugeordnet.

Begriffsdefinitionen:

- **Implikant** ist ein Term der kanonischen Normalform der Funktionsgleichung $f(x)$ mit dem Funktionswert „1_b“ (ADN) bzw. dem Wert „0_b“ (AKN).
- **Primterm** oder **Primimplikant** ist der Implikant mit der geringsten möglichen Anzahl von Variablen, der den Funktionswert „1_b“ (DN) oder „0_b“ (KN) hervorruft und nicht durch einen anderen Implikanten impliziert wird.
- **Überdeckung** (Cover) bezeichnet alle Implikanten, die durch einen Primterm dargestellt werden können.
- **Minimale Überdeckung** : Teilmenge der Primterme, die ausreicht, um alle Implikanten der Funktion darzustellen.
- **Hauptterm**: Primterm, der einen Implikanten darstellt, welcher von keinem anderen Primterm überdeckt wird (notwendiger Primterm).

Die Vereinfachung mit Hilfe der Karnaugh-Diagramme nutzt aus, dass benachbarte Felder alle Variablen enthalten und der Unterschied nur darin besteht, dass genau eine Variable in einem Feld negiert und im anderen Feld nicht negiert auftritt. Hierdurch können zwei benachbarte Felder zu einer Fläche zusammengefasst werden. Sind zwei solche Flächen benachbart, so unterscheiden auch sie sich nur im Auftreten der gleichen negierten und nicht negierten Variablen, wodurch diese beiden Flächen wiederum zusammengefasst werden können. Hierdurch sind nur Flächengrößen von 2; 4; ... 2n (n = 1 ··· p) Feldern möglich. Bei Karnaugh-Diagrammen mit mehr als zwei Variablen sind Felder, die an den Außenkanten auf gleicher Höhe liegen, auch dann benachbart, wenn an der Außenseite des Karnaugh-Diagrammes zwei Variablen liegen.

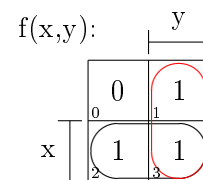
Ab fünf Variablen wird die Minimierung mit Karnaugh-Diagrammen schnell unübersichtlich, da Felder, die nach Variablenunterschied gesehen benachbart sind, nicht mehr flächig nebeneinander liegen.

Konstruktion eines Karnaugh-Diagramms - Es müssen mindestens zwei Variablen vorliegen. Somit sind vier Kombination möglich.

Wahrheitstabelle für 2 Variablen:

$(i)_{10}$	x	y	$f(x,y)$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Karnaugh-Diagramm für 2 Variablen:



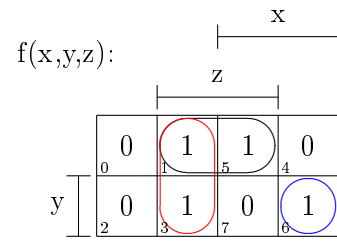
Den Wahrheitswerten wird jeweils eine Position innerhalb der 2 × 2-Matrix des Karnaugh-Diagramms zugeteilt. Die Nummern unten rechts innerhalb der Felder bezeichnen die Positionen, entsprechend den Zeilennummern in der Wahrheitstabelle. Der Strich oberhalb, unterhalb bzw. seitlich ordnet der Zeile bzw. Spalte den Wert „1_b“ der jeweiligen Variablen zu. So gilt für die Position 3 innerhalb des Karnaugh-Diagramms das für $x = y = 1$ und für $f = 1$. Für das obige Beispiel lassen sich zwei Überdeckungen bilden, wobei die Position 3 zweimal überdeckt wird. Die horizontale Überdeckung - Position 2 und 3 - liefert als Teilergebnis I (Minterm) $m_1 = x$. Die vertikale Überdeckung - Poition 1 und 3 - liefert $m_2 = y$. So folgt:

$$f(x,y)_{DMF} = \underbrace{x}_{(2,3)_{10}} \vee \underbrace{y}_{(1,3)_{10}}$$

Wahrheitstabelle für 3 Variablen:

$(i)_{10}$	x	y	z	$f(x, y, z)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Karnaugh-Diagramm für 3 Variablen:



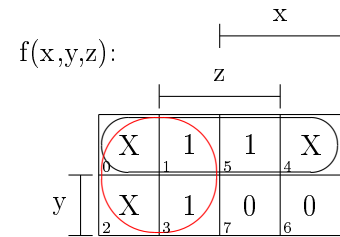
$$f(x, y, z)_{DMF} = (\neg x \wedge z) \vee (\neg y \wedge z) \vee (x \wedge y \wedge \neg z)$$

Bei manchen logischen Schaltnetzen kann es vorkommen, dass bestimmte Kombinationen der Variablen nicht auftreten oder der Ausgangszustand bei dessen Auftreten nicht von Interesse ist. Ein Beispiel sei ein BCD-Dekodierer mit vier Eingängen und zehn Ausgängen. Jeder Ausgang ist eigentlich mit nur einer Eingangskombination verknüpft. Da aber meistens davon ausgegangen wird, dass die Werte 10, ..., 15 als Dualzahl nicht auftreten, können diese als „don't care“-Bedingung genutzt werden. Die zugehörigen Bit-Kombinationen am Ausgang sind nicht von Interesse und somit können die Fälle 10, ..., 15 zur Vereinfachung herangezogen werden. In der Funktionstabelle und im Karnaugh-Diagramm werden die „don't care“-Bedingung mit einem (X) gekennzeichnet. Bei der Vereinfachung im Karnaugh-Diagramm durch Maximalflächenbildung, können diese „don't care“-Felder „0b“ oder „1b“ annehmen, je nachdem, wie es für die Vereinfachung praktischer ist.

Wahrheitstabelle für 3 Variablen:

$(i)_{10}$	x	y	z	f
0	0	0	0	X
1	0	0	1	1
2	0	1	0	X
3	0	1	1	1
4	1	0	0	X
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Karnaugh-Diagramm für 3 Variablen:

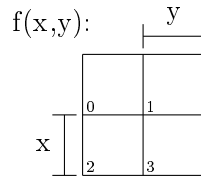


$$f(x, y, z)_{DMF} = \neg x \vee \neg y$$

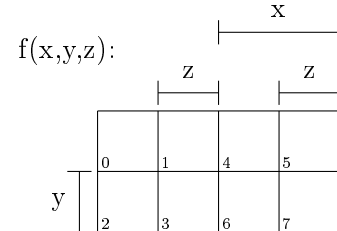
1.7 Veitch-Diagramm

Sowie das Karnaugh-Diagramm, so ist auch das Veitch-Diagramm zur Minimierung von Booleschen Ausdrücken entwickelt worden. Edward W. Veitch studierte Physik an der Harvard University, wo er 1947 und 1948 seinen Master Abschluss in Angewandter Physik und Physik machte. Er entwickelte 1952 ein grafisches Verfahren für die Optimierung von Digitalschaltungen, welches 1953 von Maurice Karnaugh weiterentwickelt wurde.

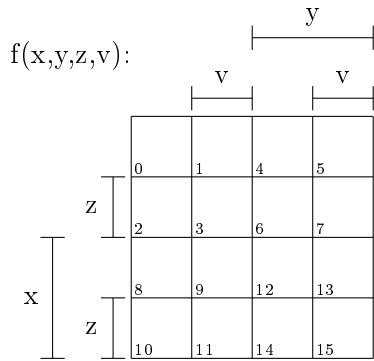
Veitch-Diagramm für
2 Variablen:



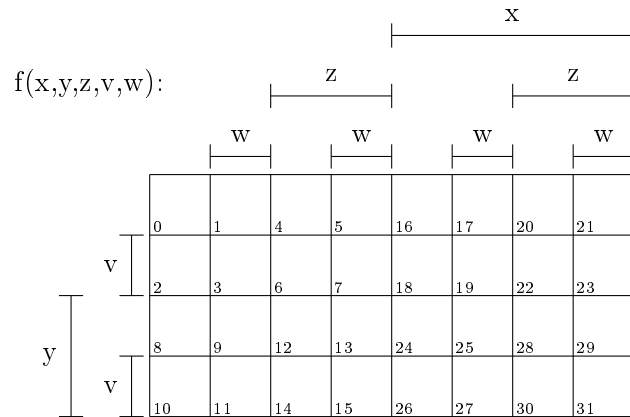
Veitch-Diagramm für 3 Variablen:



Veitch-Diagramm für
4 Variablen:



Veitch-Diagramm für 5 Variablen:



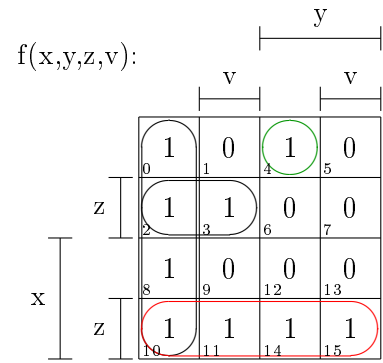
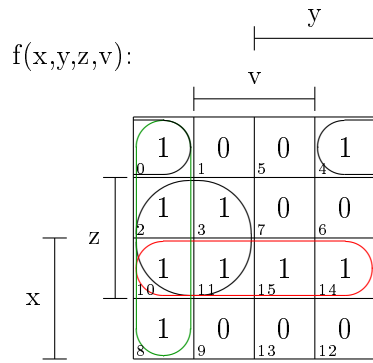
Für zwei Variablen ist die Anordnung innerhalb des Veitch-Diagramms gleich der des Karnaugh-Diagramms. Für mehr als zwei Variablen werden die Positionen der Wahrheitstabelle anders verteilt. Für drei Variablen spaltet sich die Variable z derart auf, dass für $z = 1$ die Spalte 2 und die Spalte 4 überdeckt werden. Für die Spalten 1 und 3 gilt somit $z = 0$. Zur Verdeutlichung der Anordnung wird ein Beispiel mit vier Variablen verwendet (siehe auch Beispiel 1 - Karnaugh-Diagramm).

$$\begin{aligned}
 f_1 = & \underbrace{(\neg x \wedge \neg y \wedge \neg z \wedge \neg v)}_{(0)_{10}} \vee \underbrace{(\neg x \wedge \neg y \wedge z \wedge \neg v)}_{(2)_{10}} \vee \underbrace{(\neg x \wedge \neg y \wedge z \wedge v)}_{(3)_{10}} \\
 & \vee \underbrace{(\neg x \wedge y \wedge \neg z \wedge \neg v)}_{(4)_{10}} \vee \underbrace{(x \wedge \neg y \wedge \neg z \wedge \neg v)}_{(8)_{10}} \vee \underbrace{(x \wedge \neg y \wedge z \wedge \neg v)}_{(10)_{10}} \\
 & \vee \underbrace{(x \wedge \neg y \wedge z \wedge v)}_{(11)_{10}} \vee \underbrace{(x \wedge y \wedge z \wedge \neg v)}_{(14)_{10}} \vee \underbrace{(x \wedge y \wedge z \wedge v)}_{(15)_{10}}
 \end{aligned}$$

Um den Unterschied zu verdeutlichen, wird das Ergebnis aus der Gleichung f_1 sowohl in einem Karnaugh-Diagramm als auch in einem Veith-Diagramm eingetragen.

Karnaugh-Diagramm für 4 Variablen:

Veitch-Diagramm für 4 Variablen:



Die Vereinfachung für das Karnaugh-Diagramm lautet:

$$f_{KD} = \neg y \wedge (z \vee \neg v) \vee (\neg x \wedge \neg z \wedge \neg v) \vee (x \wedge z)$$

$$= g_1 \vee g_2 \vee g_3$$

Die Gleichung f_{KD} besteht aus drei Termen, für die nacheinander die Gültigkeit ermittelt werden. Der Term $g_1 = \neg y \wedge (z \vee \neg v)$ ist erfüllt für die Zeilen (= Positionen im KD-Diagramm) $(i)_{10} = 0, 2, 3, 8, 10, 11$. Der Term $g_2 = (\neg x \wedge \neg z \wedge \neg v)$ ist nur für die Zeilen $(i)_{10} = 0, 2, 4$ erfüllt. Der Term $g_3 = (x \wedge z)$ ist nur für die Zeilen $(i)_{10} = 10, 11, 14, 15$ erfüllt.

Die Vereinfachung für das Veitch-Diagramm lautet:

$$f_{VD} = \underbrace{(\neg y \wedge \neg v)}_{(0,2,8,10)_{10}} \vee \underbrace{(\neg x \wedge y \wedge \neg z \wedge \neg v)}_{(4)_{10}} \vee \underbrace{(\neg x \wedge \neg y \wedge z)}_{(2,3)_{10}} \vee \underbrace{(x \wedge z)}_{(10,11,14,15)_{10}}$$

$$= \neg v \wedge (\neg y \vee \neg x \wedge y \wedge \neg z) \vee z \wedge (\neg x \wedge \neg y \vee x)$$

$$= k_1 \vee k_2$$

Die Gleichung f_{VD} besteht aus zwei Termen, für die nacheinander die Gültigkeit ermittelt werden. Der Term $k_1 = \neg v \wedge (\neg y \vee \neg x \wedge y \wedge \neg z)$ ist erfüllt für die Zeilen (= Positionen im VD-Diagramm) $(i)_{10} = 0, 2, 4, 8, 10$. Der Term $k_2 = z \wedge (\neg x \wedge \neg y \vee x)$ ist nur für die Zeilen $(i)_{10} = 2, 3, 10, 11, 14, 15$ erfüllt.

Die Äquivalenz der beiden Vereinfachungen wird mittels der Wahrheitstabelle nachgewiesen. Dabei werden die Ergebnisse schrittweise für jeden Term entwickelt. Nach der Wahrheitstabelle gilt somit:

$$f_1 \equiv f_{KV} \equiv f_{VD}$$

Vereinfachungsregeln:

- In Karnaugh-Veitch-Diagramme (K-V-Diagramme oder K-V-Tafeln) werden zur Vereinfachung von ODER- Normalformen alle Verknüpfungen der Wahrheitstabelle, für die $x = 1$ ist, in das entsprechende Feld der KV-Tabelle übertragen.
- Von Feld zu Feld ist jeweils nur eine Eingangsvariable zu ändern.
- Es sind 2^n Felder ($n = \text{Anzahl der Eingangsvariablen}$) erforderlich.
- Benachbarte Vollkonjunktionen werden zu 2-er-, 4-er- oder 8-er-Blöcken zusammengefasst.
- Benachbart sind Vollkonjunktionen der Felder, die mit einer Seite (nicht mit der Ecke) aneinander stoßen.
- Auch Felder, die am Rande der Tafel liegen, können zusammengefasst werden.

$(i)_{10}$	x	y	z	v	f_1	g_1	g_2	g_3	f_{KD}	k_1	k_2	f_{VD}
0	0	0	0	0	1	1	1	0	1	1	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	1	1	1	0	1	1	1	1
3	0	0	1	1	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	1	0	1	1	0	1
5	0	1	0	1	0	0	0	0	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	0
8	1	0	0	0	1	1	0	0	1	1	0	1
9	1	0	0	1	0	0	0	0	0	0	0	0
10	1	0	1	0	1	1	0	1	1	1	1	1
11	1	0	1	1	1	1	0	1	1	0	1	1
12	1	1	0	0	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0	0	0	0
14	1	1	1	0	1	0	0	1	1	0	1	1
15	1	1	1	1	1	0	0	1	1	0	1	1

- Variablen eines Blockes, die negiert und nicht negiert auftreten, entfallen.
- Die in einem Block verbleibenden Variablen werden UND-verknüpft.
- Werden diese UND-Verknüpfungen durch ODER-Verknüpfung miteinander verbunden, entsteht die vereinfachte Funktionsgleichung.

1.8 Minimierung nach Quine-McCluskey

Die Methode geht von der ADN der zu vereinfachenden Funktion aus. Durch wiederholte Auflistung werden „benachbarte“ Terme gemäß $Ax_i \vee A(\neg x_i) = A$ vereinfacht. Nach Quine-McCluskey werden unnegierte Variablen durch „1“ und negierte Variable durch „0“ codiert. Dadurch können die Terme als Dualzahlen aufgefasst (und auch dezimal geschrieben) werden. Durch Anordnung der Terme in Gruppen mit gleichen Anzahlen von Einsen kann der Aufwand für das Vergleichen der Terme vermindert werden. Die Vergleichsergebnisse werden in eine neue Tabelle eingetragen, wobei die herausgefallene Variable durch ein „-“ gekennzeichnet wird. Das Verfahren wird solange wie möglich wiederholt. Durch sinngemäßes Substituieren erhält man die Primimplikanten. Lösungsweg:

- Alle Minterme, für die $f(x)=$ „1“ oder „-“ ist, werden nach Anzahl der „bejahten“ Variablen x_i in Gruppen geordnet und in die erste Spalte einer Vereinfachungstabelle eingetragen.
- Vergleich jeder Zeile einer Gruppe mit jeder Zeile der darauffolgenden Gruppe. Vereinfachungen der Form $x_i \vee \neg x_i$ werden beim Vergleich gesucht. Lässt sich aus zwei Konjunktionen eine Variable x_i eliminieren, werden diese Konjunktionen markiert (abgehakt) und das Vereinfachungsergebnis wird in eine neue Spalte geschrieben. Sie sind damit wieder automatisch nach der Anzahl „bejahter“ Variablen geordnet.

- Zeilenweiser Vergleich benachbarter Gruppen in der Spalte der Vereinfachungsergebnisse (2.Spalte) nach demselben Schema usw., bis keine Vereinfachung mehr möglich ist.
- Nicht markierte Terme sind Primimplikanten.
- Minimalform: disjunktive Verknüpfung derjenigen Primterme, durch die alle Minterme überdeckt werden. Hierzu wird die Auswahlfunktion nach Petrick benutzt.

Auswahlfunktion nach Petrick:

- In einer Primterm-Minterm Tabelle werden die Primterme den Mintermen zugeordnet. Minterme, die „don't cares“ beschreiben, werden nicht in die Tabelle aufgenommen.
- Minimale Lösung nach Petrick: Für jede Spalte der Primterm-Minterm Tabelle die eingetragenen Primterme disjunktiv verknüpfen und die so gefundenen Spaltenausdrücke konjunktiv verknüpfen, wodurch man die Primtermlösungsfunktion erhält.
- Der Primtermlösungsfunktion den kürzesten Ausdruck entnehmen, d.h. Primtermlösungsfunktion minimieren mit den Zielen:
 1. minimale Anzahl von Primtermen
 2. minimale Anzahl von Variablen bei gleicher Primtermlänge
- Die im ausgewählten Ausdruck enthaltenen Primterme für die Minimallösung disjunktiv verknüpfen.

Beispiel: Minimieren folgenden Ausdrucks:

$$f(x, y, z, v) = m_0 \vee m_1 \vee m_2 \vee m_3 \vee m_4 \vee m_6 \vee m_7 \vee m_8 \vee m_9 \vee m_{11} \vee m_{15}$$

Schritt 1: Wahrheitstabelle

$(i)_{10}$	x	y	z	v	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Schritt 2: Sortierung nach Anzahl

$(i)_{10}$	x	y	z	v	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
4	0	1	0	0	1
8	1	0	0	0	1
3	0	0	1	1	1
6	0	1	1	0	1
9	1	0	0	1	1
7	0	1	1	1	1
11	1	0	1	1	1
15	1	1	1	1	1

Schritt 4: Zweites (und hier endgültiges) Vereinfachungsergebnis mit Bezeichnung der unterschiedlichen Primimplikanten

Schritt 3: Erstes Vereinfachungsergebnis

$(i)_{10}$	x	y	z	v	f
0,1	0	0	0	-	1
0,2	0	0	-	0	1
0,4	0	-	0	0	1
0,8	-	0	0	0	1
1,3	0	0	-	1	1
1,9	-	0	0	1	1
2,3	0	0	1	-	1
2,6	0	-	1	0	1
4,6	0	1	-	0	1
8,9	1	0	0	-	1
3,7	0	-	1	1	1
3,11	-	0	1	1	1
6,7	0	1	1	-	1
9,11	1	0	-	1	1
7,15	-	1	1	1	1
11,15	1	-	1	1	1

$(i)_{10}$	x	y	z	v	f
0,1; 2,3	0	0	-	-	a
0,1; 8,9	-	0	0	-	b
0,2; 1,3	0	0	-	-	
0,2; 4,6	0	-	-	0	
0,4; 2,6	0	-	-	0	c
0,8; 1,9	-	0	0	-	
1,3; 9,11	-	0	-	1	d
1,9; 3,11	-	0	-	1	
2,3; 6,7	0	-	1	-	e
2,6; 3,7	0	-	1	-	
3,7; 11,15	-	-	1	1	
3,11; 7,15	-	-	1	1	g

Tabelle 2: Primterm-Minterm Tabelle

	$i \downarrow m \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	0 0 - -	1	1	1	1	1											
b	- 0 0 -									1	1	1					
c	0 - - 0			1	1	1		1									
d	- 0 - 1		1	1	1	1					1		1				
e	0 - 1 -			1	1	1		1	1	1							
g	- - 1 1												1				1

Primtermlösungsfunktion:

$$\begin{aligned}
 D &= (a \vee b \vee c) \wedge (a \vee b \vee d) \wedge (a \vee c \vee e) \wedge (a \vee d \vee e \vee g) \wedge (c \vee e) \wedge (e \vee g) \\
 &\quad \wedge b \wedge (b \vee d) \wedge (d \vee g) \wedge g \\
 &= b \wedge c \wedge g
 \end{aligned}$$

Als Minimallösung bleibt somit übrig:

$$f(x, y, z, v) = b \vee c \vee g = (\neg y \wedge \neg z) \vee (\neg x \wedge \neg v) \vee (z \wedge v)$$

Im Internet findet sich auf Youtube ein anschauliches Video zum Verfahren:

<https://www.youtube.com/watch?v=VnZLRrJYa2I>

2 Aufgaben

2.1 KNF und DNF

Gegeben ist folgende Funktionstabelle:

$(i)_{10}$	x	y	z	$f(x, y, z)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

- Bestimmen Sie die KNF.
- Bestimmen Sie die DNF.

2.2 Minimierung mit 3 Variablen

Minimierung des folgenden Ausdrucks unter Anwendung eines Karnaugh-Diagramms, in dem die Position mit „0“ zusammengefasst wird.

$(i)_{10}$	x	y	z	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

2.3 Minimierung mit 4 Variablen (1)

Minimierung folgender Funktion

$$f = (x \wedge \neg y \wedge z \wedge v) \vee (x \wedge y \wedge z \wedge v) \vee (x \wedge y \wedge \neg z \wedge v) \vee (\neg x \wedge y \wedge z \wedge v) \vee (\neg x \wedge y \wedge \neg z \wedge v)$$

mit Rechenregeln.

2.4 Minimierung mit 4 Variablen (2)

Gegeben ist folgende Funktion:

$$\begin{aligned} f = & (\neg x \wedge \neg y \wedge \neg z \wedge \neg v) \vee (\neg x \wedge \neg y \wedge z \wedge \neg v) \vee (\neg x \wedge \neg y \wedge z \wedge v) \\ & \vee (\neg x \wedge y \wedge \neg z \wedge \neg v) \vee (x \wedge \neg y \wedge \neg z \wedge \neg v) \vee (x \wedge \neg y \wedge z \wedge \neg v) \\ & \vee (x \wedge \neg y \wedge z \wedge v) \vee (x \wedge y \wedge z \wedge \neg v) \vee (x \wedge y \wedge z \wedge v) \end{aligned}$$

- Minimierung den gegebenen Ausdrucks unter Anwendung eines Karnaugh-Diagramms, in dem die Position mit „1“ zusammengefasst wird.
- Zeigen Sie Äquivalenz der DMF anhand der Wahrheitstabelle.

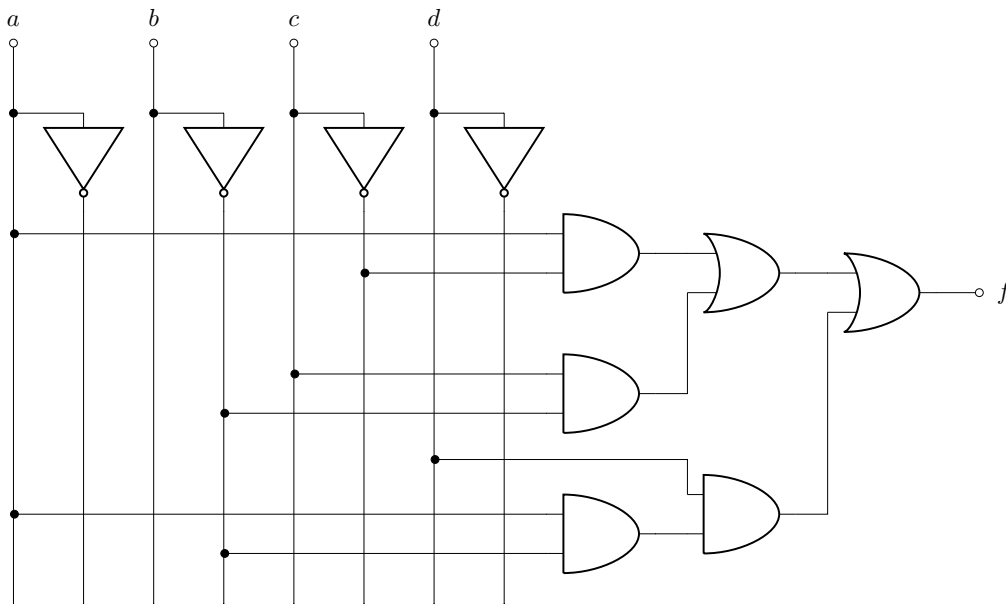
2.5 Minimierung mit 4 Variablen (3)

Minimierung des folgenden Ausdrucks unter Anwendung eines Karnaugh-Diagramms, in dem die Position mit „1“ zusammengefasst wird. Die nicht definierten Felder werden mit „undefiniert = X“ belegt.

$$\begin{aligned}
 f = & (\neg x \wedge \neg y \wedge z \wedge v) \vee (\neg x \wedge y \wedge z \wedge v) \vee (x \wedge \neg y \wedge \neg z \wedge v) \\
 & \vee (x \wedge \neg y \wedge z \wedge \neg v) \vee (x \wedge \neg y \wedge z \wedge v) \vee (x \wedge y \wedge \neg z \wedge v) \\
 & \vee (x \wedge y \wedge z \wedge \neg v)
 \end{aligned}$$

2.6 Schaltungs-Minimierung mit Karnaugh-Diagrammen

Vereinfachen Sie folgende Schaltung mithilfe eines Karnaugh-Diagramms!



i_{10}	a	b	b	d	f
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

2.7 Karnaugh-Diagramme mit „Don't Cares“

Wenn die Werte einer Funktion für gewisse Kombinationen unbekannt (oder nicht wichtig) sind, kann das entsprechende Feld im Karnaugh-Diagramm mit X markiert werden. Solche Felder können sowohl als 0 als auch als 1 betrachtet werden.

Benutzen Sie ein Karnaugh Diagramm mit „Don't Cares“ um die folgenden, in Form einer Wahrheitstabelle gegebenen, Funktionen für die Ausgangssignale y, y, z, v zu minimieren.

i_{10}	a	b	b	d	x	y	z	v
0	0	0	0	0	0	1	0	1
1	0	0	0	1	0	1	X	X
2	0	0	1	0	0	1	0	1
3	0	0	1	1	X	0	0	0
4	0	1	0	0	X	0	0	0
5	0	1	0	1	0	0	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	X	X	X	X
8	1	0	0	0	1	0	1	1
9	1	0	0	1	X	0	1	1
10	1	0	1	X	1	0	0	X
11	1	1	X	X	0	1	X	1

2.8 Quine-McCluskey Methode ohne „Don't Cares“ (1)

Im folgenden soll mit dem Verfahren von Quine-McCluskey ein Minimalpolynom für y abgeleitet werden, das in Abhängigkeit von x_3, x_2, x_1 und x_0 anhand folgender Tabelle definiert ist:

Minterm- Index i_{10}	x_3	x_2	x_1	x_0	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

- a) Bestimmen Sie die Primimplikanten mit dem Quine-McCluskey-Verfahren:
- b) Erstellen Sie eine Primimplikantentafel nach dem Quine-McCluskey-Verfahren:
- c) Geben Sie das Minimalpolynom an:

2.9 Quine-McCluskey Methode ohne „Don't Cares“ (2)

Benutzen Sie das Minimierungsverfahren von Quine-McCluskey, um folgende Ausdrücke zu vereinfachen (m_i bezeichnet einen Minterm):

a) $f(x, y, z) = m_1 + m_3 + m_6 + m_7$

b) $f(a, b, c, d) = m_1 + m_3 + m_7 + m_8 + m_9 + m_{10} + m_{11} + m_{14} + m_{15}$.

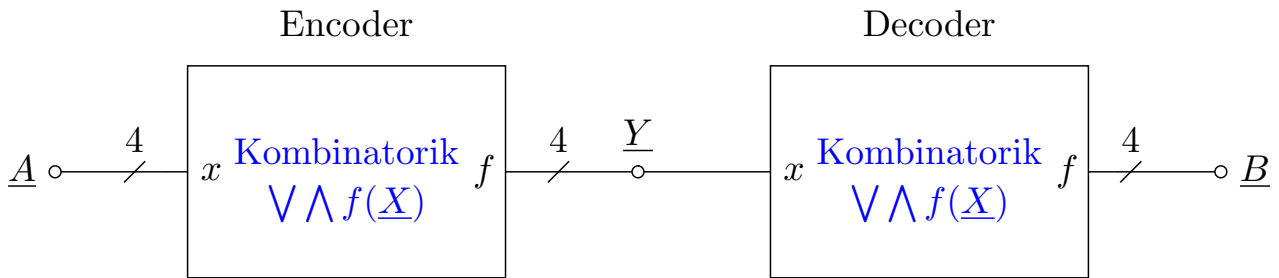
2.10 Quine-McCluskey Methode mit „Don't Cares“

Benutzen Sie die Quine-McCluskey Methode zur Minimierung der folgenden logischen Funktion, welche durch zwei Sets von dezimalen Indizes gegeben ist: $f(\{0, 1, 2, 3, 8, 10, 13\}) = 1$, $f(\{4, 5, 6, 7, 12, 14, 15\}) = 0$. Wieviele verschiedene minimale Lösungen existieren?

Tipp: Behandeln Sie im ersten Schritt der Methode die „Don't Cares“ als 1 und ignorieren Sie sie im zweiten Schritt.

2.11 Umkehrfunktion oder Decodierung

In diesem Laborversuch sollen zunächst Funktionstabellen für 4 unbekannte Logikfunktionen abgeleitet werden. Dazu wurde in Online die LogiSim-Schaltungssammlung **gds-labor-02.circ** zur Verfügung gestellt. Für die Laboraufgaben werden Ihnen 4 Schaltungen aus 14 Encoder-Varianten zugewiesen. Simulieren Sie die Encoder und stellen Sie die Funktionstabelle auf. Anschließend soll aus der Funktionstabellen heraus die Encoder-Vorschrift hergeleitet werden (z.B. als DMF). Sie entwickeln im nächsten Schritt eine Decoder-Schaltung, welcher die unbekannt Codes wieder zurück in die Ursprungscodierung wandelt. Es steht dafür ein Labortermin zur Verfügung.



Die Vektor-Zuordnung $(\underline{A}, \underline{Y}, \underline{B})$ ist in der obigen Abbildung dargestellt: A0 bis A3 sind die Eingänge, Y0 bis Y3 die Ausgänge. Jeder der Logik-Ausgänge Y0...3 realisiert eine Boolesche Funktion in Abhängigkeit von den 4 Eingängen. Leiten Sie als Erstes diese Funktionsabhängigkeit ab. Ordnen Sie dabei die Variablen rechtsbündig an, so dass die Darstellung derjenigen binärer Zahlen entspricht (also von links nach rechts: A3, A2, A1, A0; Y3...0 entsprechend).

Sie sollen nun als nächsten Schritt den erzeugten 4 Bit-Code \underline{Y} als Eingänge für eine Decoder-Schaltungen benutzen, die mit ihren 4 Ausgängen den Original-Code \underline{A} wieder herstellen. Anders gesagt: Legt man an das Ihnen zugeordneter Encoder z.B. die Bit-Kombination „0000“ an A3...0, erzeugt dies den von Ihnen zu bestimmenden Code an Y3...0. Diesen Code legen Sie nun an Ihre Decoder-Schaltung, die dann als Ergebnis wieder die „0000“ erzeugt. Dies soll für alle sechzehn möglichen Eingangs-Codes gelten.

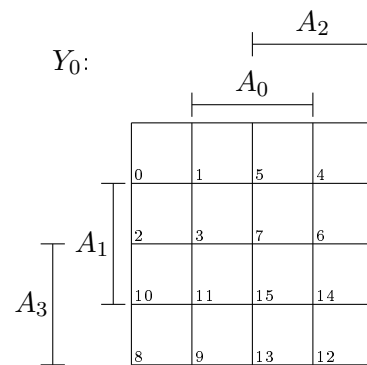
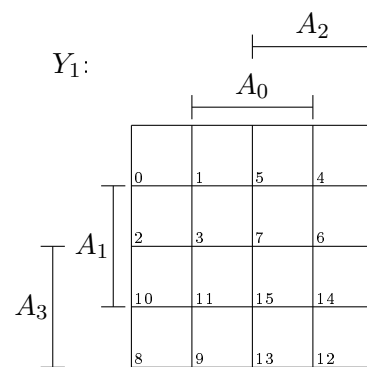
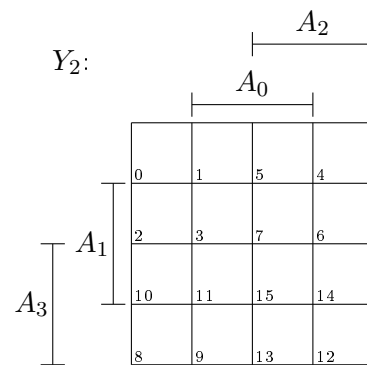
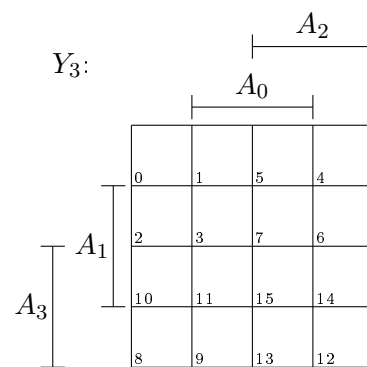
Hinweis: Für die Lösung dieser Aufgabe müssen Sie Karnaugh-Diagramme aufstellen, aus denen Sie die minimalen Schaltungen (DMF) ableiten.

Für diesen Laborversuch müssen Sie keinen eigenen Laborbericht schreiben. Auf den nachfolgenden Seiten sind die notwendigen Wahrheitstabellen, Karnaugh-Diagramme und Tabellen zum Erfassen der Funktionen dargestellt. Füllen Sie diese bitte vollständig aus.

Encoder N_1 :

$(i)_{10}$	A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

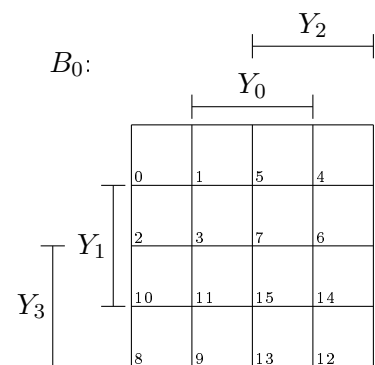
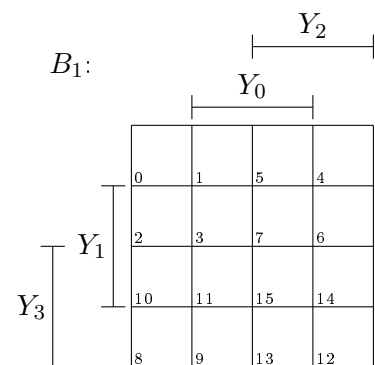
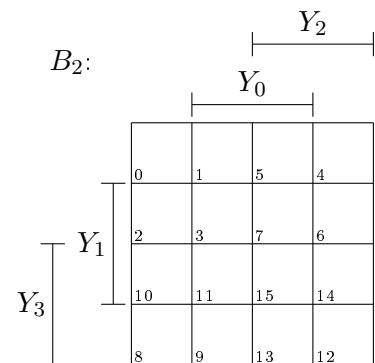
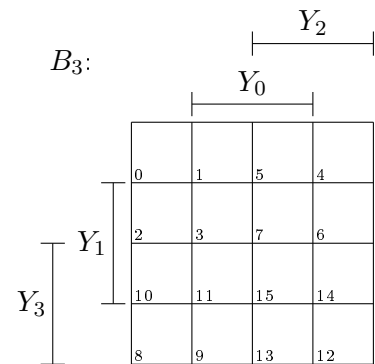
Ausgang	Boolesche Funktion
$Y_3 =$	
$Y_2 =$	
$Y_1 =$	
$Y_0 =$	



Decoder N_1 :

$(i)_{10}$	Y_3	Y_2	Y_1	Y_0	B_3	B_2	B_1	B_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Ausgang	Boolesche Funktion
$B_3 =$	
$B_2 =$	
$B_1 =$	
$B_0 =$	

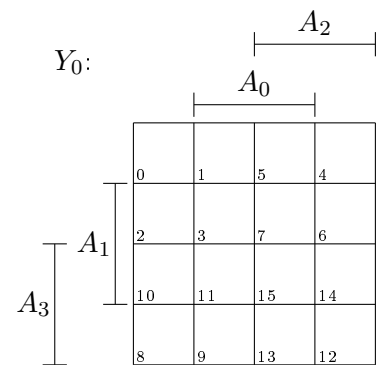
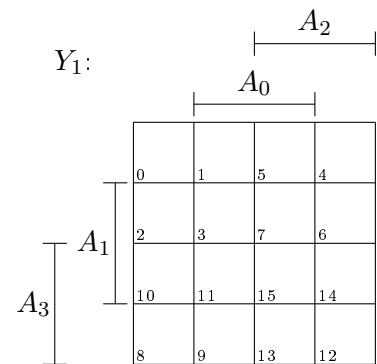
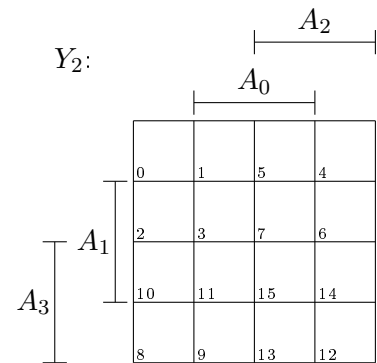
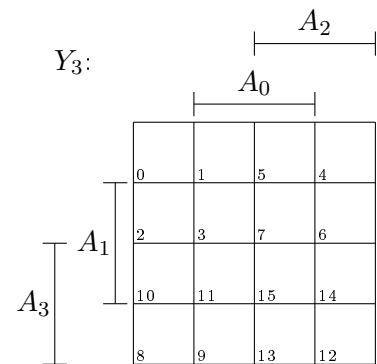


Hinweis: Unter dem Link <http://wahrheitstabelle.daug.de/#/> findet sich ein Tool zur Verifikation einer Booleschen Funktion versus die dazugehörige Wahrheitstabelle.

Encoder N_2 :

$(i)_{10}$	A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

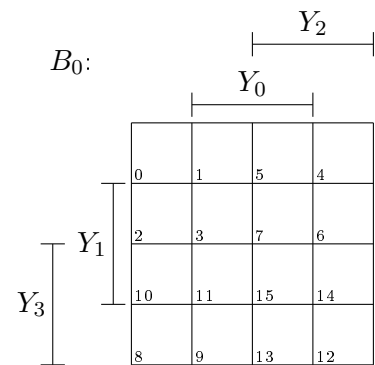
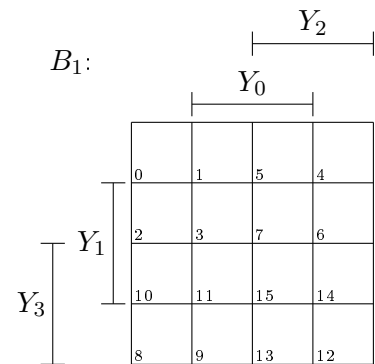
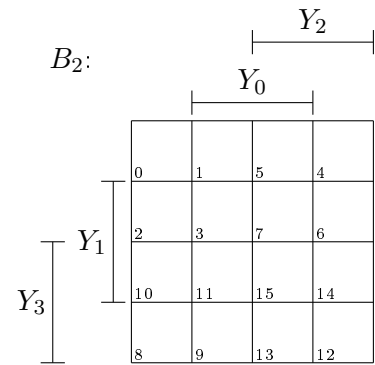
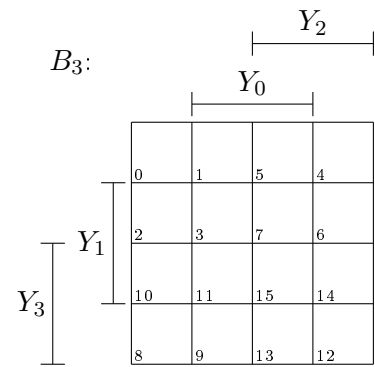
Ausgang	Boolesche Funktion
$Y_3 =$	
$Y_2 =$	
$Y_1 =$	
$Y_0 =$	



Decoder N_2 :

$(i)_{10}$	Y_3	Y_2	Y_1	Y_0	B_3	B_2	B_1	B_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

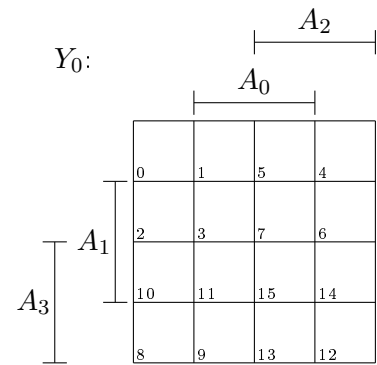
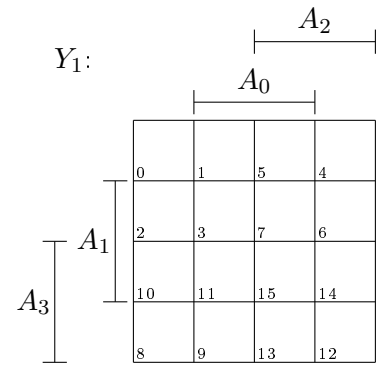
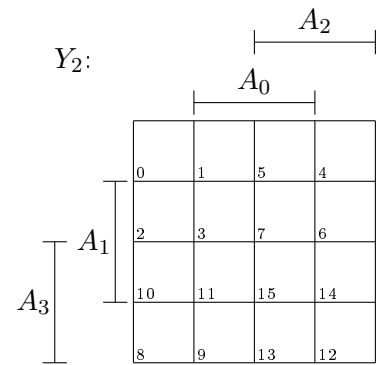
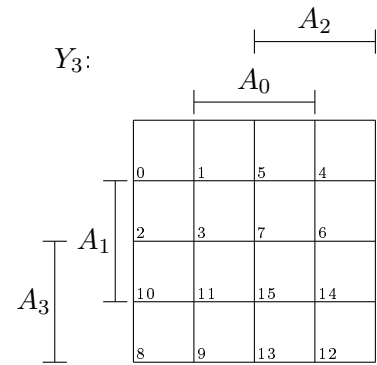
Ausgang	Boolesche Funktion
$B_3 =$	
$B_2 =$	
$B_1 =$	
$B_0 =$	



Encoder N_3 :

$(i)_{10}$	A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

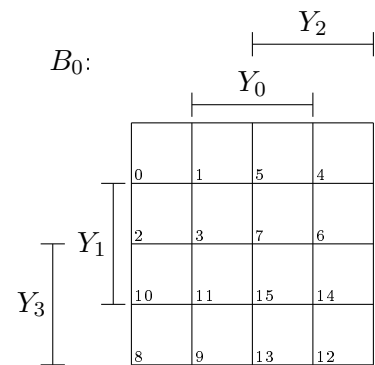
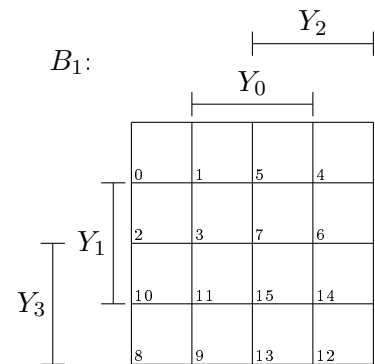
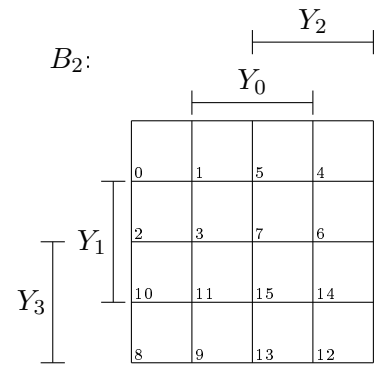
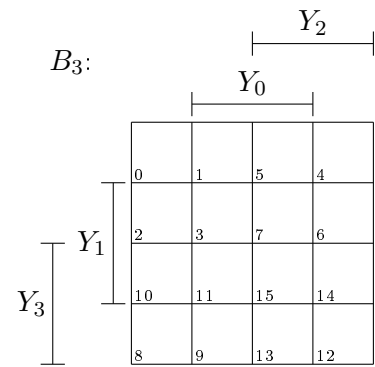
Ausgang	Boolesche Funktion
$Y_3 =$	
$Y_2 =$	
$Y_1 =$	
$Y_0 =$	



Decoder N_3 :

$(i)_{10}$	Y_3	Y_2	Y_1	Y_0	B_3	B_2	B_1	B_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

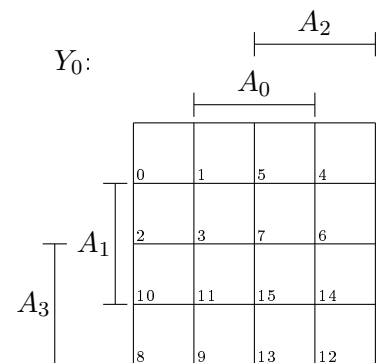
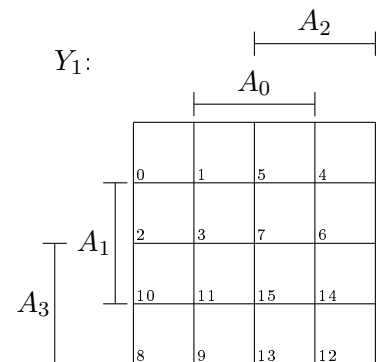
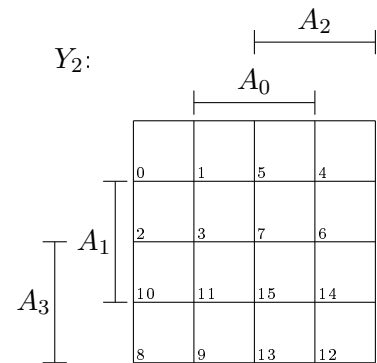
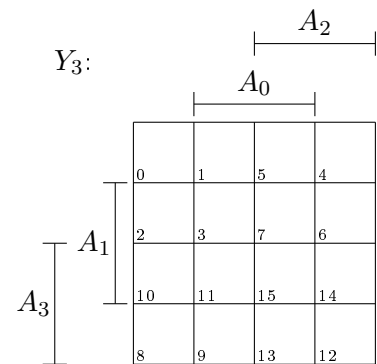
Ausgang	Boolesche Funktion
$B_3 =$	
$B_2 =$	
$B_1 =$	
$B_0 =$	



Encoder N_4 :

$(i)_{10}$	A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

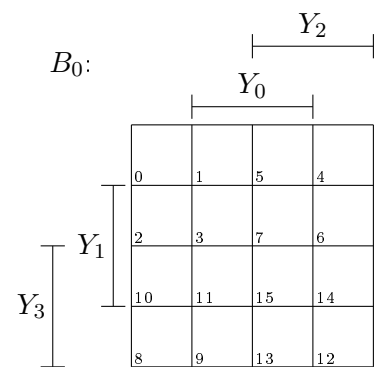
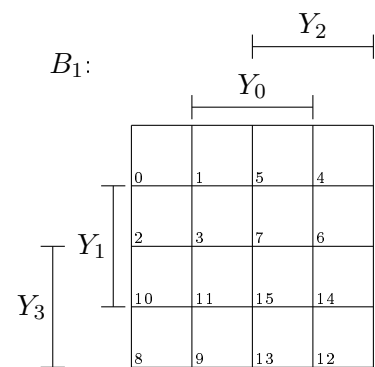
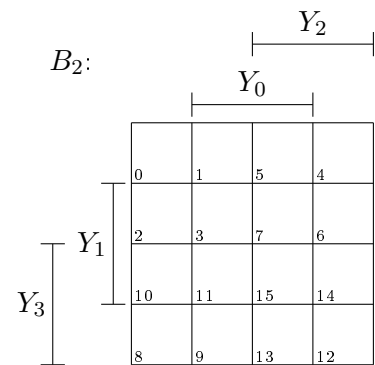
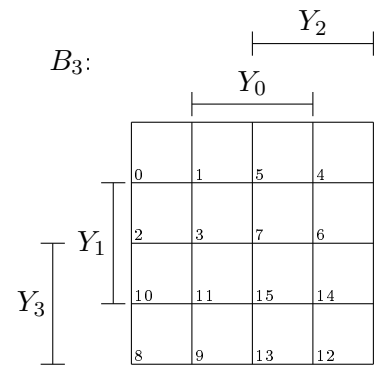
Ausgang	Boolesche Funktion
$Y_3 =$	
$Y_2 =$	
$Y_1 =$	
$Y_0 =$	



Decoder N_4 :

$(i)_{10}$	Y_3	Y_2	Y_1	Y_0	B_3	B_2	B_1	B_0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Ausgang	Boolesche Funktion
$B_3 =$	
$B_2 =$	
$B_1 =$	
$B_0 =$	

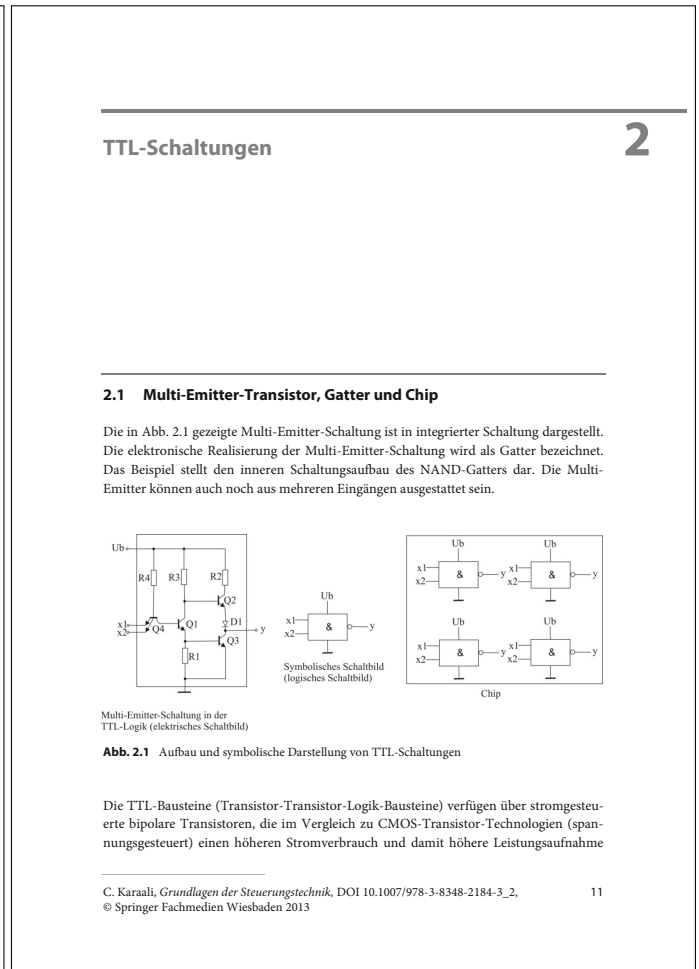


3 Literaturstudium und Protokollerstellung

Zur Vor- und Nachbereitung des Laborversuchs sei folgendes Fachbuch empfohlen:



(a) Titelseite



(b) Kapitel 2

Abbildung 1: Literaturhinweis <https://link.springer.com/>

Laborprotokoll: Es steht Ihnen frei, welches Textprogramm Sie zur Erstellung des Laborprotokolls verwenden. Für das wissenschaftliche Schreiben hat sich die LATEX-Umgebung etabliert (<https://www.latex-project.org/>). Ein Online-Tool zur Dokumentenerstellung findet sich unter <https://de.overleaf.com/>.

Notizen: