Schaltalgebra

Inhalt

1. Schaltvariable und Schaltfunktion

Die Schaltalgebra der Digitaltechnik beruht auf der booleschen Algebra. Dabei spielen die Begriffe der Schaltvariablen und der Schaltfunktion eine wichtige Rolle.

Die *Schaltvariable* ist eine Variable, die nur die Werte 0 oder 1 annehmen kann. Bezeichnungen für Schaltvariablen sind etwa: $A, B\ldots; x_1, x_2 \ldots, y\ldots$ Oftmals (bei digitalen Bausteinen) werden auch sprechende Namen verwendet: $\mathrm{Enable}, \mathrm{Disable}\ldots$
Bsp. (Variablen mit Wertzuweisungen): $x_1 = 0$, $A = B = 1$, $\mathrm{On} = 0$, $\mathrm{EVEN} = 1$

Mit Schaltvariablen und entsprechenden Verknüpfungen können *Schaltfunktionen* (Boolesche Gleichungen, Funktionen) gebildet werden.

$y = f(x_1, x_2 \ldots x_n)$ oder
$z = f(x_0, x_1 \ldots x_{m-1})$

y, z: Ausgangsvariablen
x_i: Eingangsvariablen
i: Index: $1 \le i \le n, 0 \le i \le m-1$
f: Verknüpfung, Funktion

Obige Funktionen weisen n bzw. m Eingangsvariablen auf.
Auch y und z sind Schaltvariablen und können daher nur die Werte 0 oder 1 annehmen.
Bsp: $y = \neg x$, oder gleichbedeutend: $y = \bar{x}$, oder wiederum gleichbedeutend: $y = \mathrm{not}(x)$

Da Schaltfunktionen nur endlich viele Permutationen bei n oder m Eingangsvariablen x_i aufweisen, werden sie auch tabellarisch in sog. Wahrheitstabellen dargestellt.

Unäre Schaltfunktionen

Eine unäre Schaltfunktion stellt die not-Funktion (invertierende Funktion, EK-Funktion) dar:

x	y
0	1
1	0

Wahrheitstabelle der Inversion

$y = \neg x$, $y = !x$, $y = -x$, $y = /x$

Boolesche Schreibweise (not-Operator: "not x")

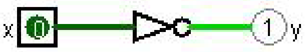
$y = \bar{x}$

alternative Schreibweise (quer-Operator: "x quer")

$y = \mathrm{not}(x)$, $y = \mathrm{EK}(x)$

Funktionsschreibweise

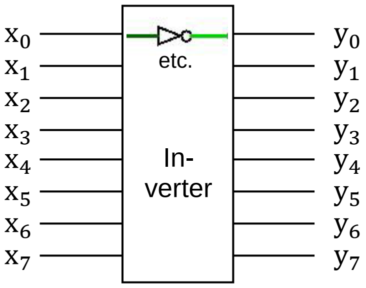
In Schaltungen wird die not-Funktion durch das Symbol des Inverters (not-Gatters) realisiert.



Beim not-Operatoren handelt es sich um einen *unären* Operatoren, da der Operator nur auf eine Schaltvariable allein wirkt.

Mit 8 Invertern (allgemeinere Bezeichnung: 8 Gattern, 8 Toren, 8 Gates) kann ein Baustein für ein Byte gebaut werden. Zugehörige Ein- und Ausgänge sind durch gleiche Indizes gekennzeichnet.

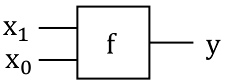
Aufgabe:
Charakterisieren Sie mit Wahrheitstabellen alle anderen unären Schaltfunktionen.



8-Bit Inverter ($y_i = \overline{x_i}$)
oder auch: EK-Baustein für ein Byte

Binäre Schaltfunktionen

$y = f(x0, x1)$



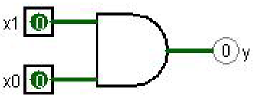
Die binäre Funktion f (Funktion mit 2 Schalt-variablen) verknüpft zwei Eingangsvariablen zu einer Ausgangsvariablen.

AND
Bsp. and-Verknüpfung, Und-Funktion, Konjunktion, logisches Produkt:

x_1	x_0	y
0	0	0
0	1	0
1	0	0
1	1	1

f: and

Die Eingangsvariablen werden in allen Werte-Permutationen aufgetragen. Wird x_1 als MSB und x_0 als LSB eines Wortes der Länge 2 aufgefasst, so werden die binären Werte 0 bis 3 untereinander dargestellt. Spalte y: Ausgangswerte, auch als Ausgangsvektor bezeichnet.



Schaltsymbol für die Und-Verknüpfung

$x_1 \cdot x_0 = y$

logisches Produkt (bevorzugte Schreibweise, auch ohne Punkt-Operator verwendet)

$x_1 \wedge x_0 = y, x_1 \,\&\, x_0 = y$

Weitere Und-Operatoren-Schreibweisen
Merke: der Operator wird infix notiert, d.h. zwischen den Operanden, auf die er wirkt.

$\mathrm{and}(x_0, x_1) = y$

Funktionsschreibweise

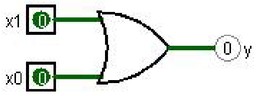
Die Und-Verknüpfung mit n Eingängen auf einen Ausgang kann allgemein formuliert werden: $\mathrm{and}(x_0, \ldots x_{n-1}) = y$. Die Verknüpfung lässt sich wie folgt eindeutig charakterisieren:

a) Der Ausgang y weist dann und nur dann den Wert 1 auf, wenn alle n Eingänge den Wert 1 aufweisen.
b) Der Ausgang y weist den Wert 0 auf, wenn mindestens 1 Eingang den Wert 0 aufweist.

OR
Bsp. or-Verknüpfung, Oder-Funktion, Disjunktion, logische Summe:

x_1	x_0	y
0	0	0
0	1	1
1	0	1
1	1	1

f: or



Schaltsymbol für die Oder-Verknüpfung

$x_1 + x_0 = y$ logische Summe (bevorzugte Schreibweise, der Oder-Operator muss stets geschrieben werden)

$x_1 \vee x_0 = y$, $x_1 \mid x_0 = y$ Weitere Oder-Operatoren-Schreibweisen

$\mathrm{or}(x_0, x_1) = y$ Funktionsschreibweise

Die Oder-Verknüpfung mit n Eingängen auf einen Ausgang kann allgemein formuliert werden: $\mathrm{or}(x_0, \ldots x_{n-1}) = y$. Die Verknüpfung lässt sich wie folgt eindeutig charakterisieren:

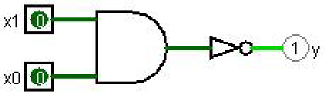
a) Der Ausgang y weist dann und nur dann den Wert 0 auf, wenn alle n Eingänge den Wert 0 aufweisen.
b) Der Ausgang y weist den Wert 1 auf, wenn mindestens 1 Eingang den Wert 1 aufweist.

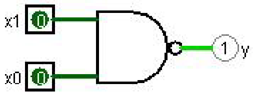
Fragen:
Wie viele mögliche binäre Verknüpfungen gibt es insgesamt? Oder anders gefragt: Mit 2 Eingangsvariablen können wie viele unterschiedliche Wahrheitstabellen produziert werden?

Kennen Sie die Namen einiger dieser binären Verknüpfungen nebst AND und OR?

Von praktischer Bedeutung und mit geläufigen Namen versehen sind die Verknüpfungen AND, OR, NAND, NOR, EXOR und EXNOR.

NAND
Ein NAND-Gatter entsteht durch ein AND-Gatter und das Nachschalten eines NOT-Gatters (NAND = NOT-AND).





x_1	x_0	y
0	0	1
0	1	1
1	0	1
1	1	0

$\mathrm{f:}\ \ \mathrm{nand}$

$\overline{x_1 \cdot x_0} = y$

logisches Produkt invertiert (bevorzugte Schreibweise). Mit dieser Schreibweise wird intuitiv die Operatorenpriorität zwischen AND und NOT ersichtlich. Zudem gruppiert der not-Operator, d.h. Klammern sind nicht nötig, auch in Verknüpfungen mit andern Operatoren nicht.

$\neg\,(x_1 \wedge x_0) = y,\ !\,(x_1 \,\&\, x_0) = y$

Weitere NAND-Operatoren-Schreibweisen (Klammern nötig!)

$\mathrm{nand}(x_0, x_1) = \mathrm{not}(\mathrm{and}(x_0, x_1) = y$

Funktionsschreibweise

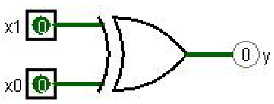
Aufgabe:
Erstellen Sie Wahrheitstabelle, Symbole etc. für das NOR-Gatter.

EXOR
Die Exclusive-OR-Funktion weist folgende Wahrheitstabelle auf:

x_1	x_0	y
0	0	0
0	1	1
1	0	1
1	1	0

f: exor

Der alternative Name Antivalenz wiedergibt die Werte des Ausgangsvektors:
Der Ausgangwert von y ist nur dann 1, wenn die Eingänge unterschiedliche Werte aufweisen.
(entweder der eine - oder der andere)



Schaltsymbol für die exor-Verknüpfung

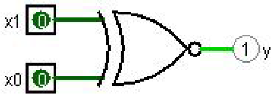
$x_1 \oplus x_0 = y$ EXOR (bevorzugte Schreibweise)

$x_1 \not\equiv x_0 = y, x_1 \nleftrightarrow x_0 = y$ Weitere EXOR-Operatoren-Schreibweisen

$\mathrm{xor}(x_0, x_1) = y$ Funktionsschreibweise

Frage:
Das EXOR-Gatter kann mit den sog. Grund-Gattern AND, OR und NOT aufgebaut werden. Wie?

EXNOR
Das Symbol für das Exclusive-NOR-Gatter sieht wie folgt aus:



$\overline{x_1 \oplus x_0} = y$ EXNOR (bevorzugte Schreibweise)

Aufgabe:
Notieren Sie analog dem EXOR die Wahrheitstabelle des EXNOR (Aequivalenz) etc.

2. Rechenregeln der Booleschen Algebra

Bemerkung:
Der Vorteil der Verwendung der Operatoren "·" (logisches Produkt) und "+" (logische Summe) ist, dass meist auf gewohnte Art und Weise gerechnet werden kann:

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 1$ (Ausnahme)

Quer-Notation (NOT)
$\overline{0} = 1$ lies: Null quer
$\overline{1} = 0$ lies: Eins quer

Operatoren-Vorrang:
"Quer" gruppiert und ist der stärkste Operator.
Punkt vor Strich, wie gewohnt. Der Punkt (der logische Produkt-Operator) kann weggelassen werden.
Klammern können helfen, Klarheit zu schaffen. Anwendung ebenfalls wie gewohnt.

Beispiele:
$A \cdot B + CD$ im Normalfall wird der Punktoperator gar nicht geschrieben
$A(B + C)$
$(AB) + C$ ok, aber Klammerverwendung unüblich
$A\,\overline{B + C}$ quer macht Klammern unnötig
$A + BC$ Punkt vor Strich
$A\overline{B} + \overline{CDE}$

$\overline{A\overline{B} + \overline{CDE}}$ oberstes Quer ist äusserster Operator. Allenfalls Klammern schreiben

$\overline{A\overline{B} + \overline{CDE}}$ semantisch nicht möglich

Aufgabe:
Zeichnen Sie die Schaltungen mit den eingeführten Schaltsymbolen zu obigen Gleichungen.

Inversion

$A = \overline{\overline{A}}$

Kommutative Gesetze

$A \cdot B = B \cdot A$

$A + B = B + A$

Beispiele/Bezeichnungen

$A \cdot B = AB = (A \cdot B) = (AB)$

Assoziative Gesetze

$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$

$A + (B + C) = (A + B) + C = A + B + C$

Distributive Gesetze

$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

$A + (B \cdot C) = (A + B) \cdot (A + C)$

$A \cdot (B + C + D) = (A \cdot B) + (A \cdot C) + (A \cdot D)$

$A + (B \cdot C \cdot D) = (A + B) \cdot (A + C) \cdot (A + D)$

De Morgansche Theoreme

$A \cdot B = \overline{\overline{A} + \overline{B}}$

$A + B = \overline{\overline{A} \cdot \overline{B}}$

$A \cdot B \cdot C \cdot D = \overline{\overline{A} + \overline{B} + \overline{C} + \overline{D}}$

$A + B + C + D = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}}$

Logische Produkte

$A \cdot 0 = 0$

$A \cdot 1 = A$ — 1: neutrales Element

$A \cdot A = A$

$A \cdot \overline{A} = 0$ — komplementäre Elemente

Logische Summen

$A + 0 = A$ — 0: neutrales Element

$A + 1 = 1$

$A + A = A$

$A + \overline{A} = 1$ — komplementäre Elemente

Weitere Beziehungen

$A\,(A + B) = A$ — Absorption: $A\,(A + B) = A + AB = A(1 + B) = A$

$A + AB = A$ — Absorption: $A + AB = A(1 + B) = A$

$A\,(\overline{A} + B) = AB$

$A + \overline{A}B = A + B$

$AB + A\overline{B} = A$

$(A + B)(A + \overline{B}) = A$

Shannonscher Satz

$f(A, B, C, D \ldots, \cdot, +, 1, 0) = \overline{f(\overline{A}, \overline{B}, \overline{C}, \overline{D} \ldots, +, \cdot, 0, 1)}$

Klammern setzen, <u>falls Priorität</u> ändert:

$\overline{A} + (B \cdot C) + D = \overline{A \cdot \overline{(\overline{B} + \overline{C})} \cdot \overline{D}}$

In Worten:
Von der Ausgangsfunktion wird mit Shannon eine gleichwertige Funktion gefunden, indem alle einzelnen Schaltvariablen invertiert, die Oder- und Und-Operatoren vertauscht sowie 0 und 1 vertauscht werden. Zum Schluss muss die neue Funktion gesamthaft invertiert werden.

Bemerkungen:

- Schwierigkeiten bereitet oft das Distributivgesetz $\mathrm{A} + (\mathrm{B} \cdot \mathrm{C}) = (\mathrm{A} + \mathrm{B}) \cdot (\mathrm{A} + \mathrm{C})$.
- Die De Morganschen Theoreme lassen sich - wie alle andern Gesetze auch - via Wahrheitstabellen beweisen.
- Der Satz von Shannon verallgemeinert die De Morganschen Theoreme. Hier ist Vorsicht bezüglich der Ortserhaltung der Operatorenprioritäten geboten. Klammersetzung vor Anwendung löst das Problem:
  $\overline{\mathrm{A}} + (\mathrm{B} \cdot \mathrm{C}) + \mathrm{D} = \overline{\mathrm{A} \cdot (\overline{\mathrm{B}} + \overline{\mathrm{C}}) \cdot \overline{\mathrm{D}}}$
- Alle Gesetze sind symmetrisch oder dual. Dual heisst: Durch Vertauschen von AND und OR sowie 0 und 1 wird wieder ein gültiges Gesetz gefunden.
  Bsp.:
  $\mathrm{A} \cdot 0 = 0$ duales Gesetz: $\mathrm{A} + 1 = 1$
  $\mathrm{A} \cdot \mathrm{B} = \overline{\overline{\mathrm{A}} + \overline{\mathrm{B}}}$ duales Gesetz: $\mathrm{A} + \mathrm{B} = \overline{\overline{\mathrm{A}} \cdot \overline{\mathrm{B}}}$

3. Kanonische Normalformen

Die Kanonische Disjunktive Normalform (KDNF)

An einem Beispiel:
Gegeben: eine willkürliche Wahrheitstabelle

	x_2	x_1	x_0	y
X_0	0	0	0	0
$\mathbf{X_1}$	**0**	**0**	**1**	**1**
$\mathbf{X_2}$	**0**	**1**	**0**	**1**
X_3	0	1	1	0
X_4	1	0	0	0
X_5	1	0	1	0
X_6	1	1	0	0
$\mathbf{X_7}$	**1**	**1**	**1**	**1**

Vorgehen zum Bilden der KDNF obiger Tabelle
- Betrachten aller Eingangsvektoren (jeder Zeile) X_i, für die die Funktion $y = f(X_i)$ den Wert 1 annimmt.
  In unserem Beispiel sind dies X_1, X_2 und X_7.

- Für jeden oben gefundenen Eingangsvektor wird eine Konjunktion mit den Eingangsvariablen x_i (beachte: klein x) gebildet, die sog. Minterme m_i.
  In unserem Beispiel:
  Für X_1: $m_1 = \overline{x_2}\,\overline{x_1}\,x_0$,
  für X_2: $m_2 = \overline{x_2}\,x_1\,\overline{x_0}$,
  für X_7: $m_7 = x_2\,x_1\,x_0$

  Die x_i werden invertiert notiert, wenn sie den Wert 0 aufweisen, sonst werden sie nicht-invertiert notiert.

- Zur Bildung der KDNF werden die Minterme disjunktiv verknüpft und mit y gleich gesetzt.
  Hier:
  $y = m_1 + m_2 + m_7 = \overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_2}\,x_1\,\overline{x_0} + x_2\,x_1\,x_0$

Die Kanonische Konjunktive Normalform (KKNF)

Die Bildung der KKNF wird wie folgt gebildet:
- Betrachten der Eingangsvektoren, für die der Spaltenwert y den Wert 0 aufweist.
- Statt der Minterme m_i werden jetzt die Maxterme M_i formuliert: Die einzelnen x_i werden invertiert notiert, wenn sie den Wert 1 aufweisen, sonst werden sie nicht-invertiert notiert.
- Innerhalb der Maxterme werden die x_i disjunktiv verknüpft.
- Zur Bildung der KKNF werden die Maxterme konjunktiv verknüpft und mit y gleich gesetzt.

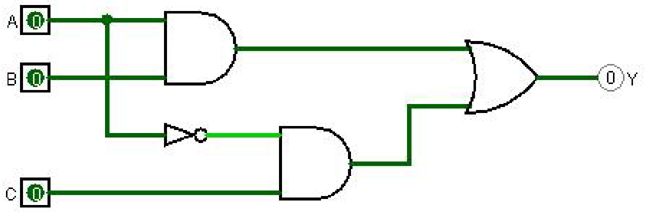
Aufgabe:
Bilden Sie die KKNF der obigen Tabelle. Formulieren Sie die Vorgehensweise zur Findung der KKNF wie dies für die KDNF formuliert wurde.

Hinweis:
Ihre Lösung sollte die Form $y = M_0 \cdot M_3 \cdot M_4 \cdot M_5 \cdot M_6$ aufweisen.

Die beiden Normalformen sind gleichwertig. Allerdings kann der schaltungsmässige Aufwand zur Realisierung unterschiedlich sein, wie obiges Beispiel zeigt.

Frage:
Wann ist welche Normalform für einen kleineren Realisierungsaufwand vorteilhafter?

Fragestellung
Gegeben ist untenstehende Schaltung. Die KDNF dieser Schaltung ist zu ermitteln.



Mögliches Vorgehen:
- Funktion ermitteln
- Wahrheitstabelle via Funktion erstellen
- KDNF aus Tabelle ermitteln

4. Verfahren zur Minimierung von Schaltnetzen

Repetition

$(x_0 \cdot x_1) + (x_0 \cdot \overline{x_1}) = x_0 \qquad = x_0 \cdot \underbrace{(x_1 + \overline{x_1})}_{1}$

$(x_0 + x_1) \cdot (x_0 + \overline{x_1}) = x_0 \qquad = x_0 + \underbrace{(x_1 \cdot \overline{x_1})}_{0}$

Mit diesen Regeln lassen sich oft minimierte Schaltungen finden.

An einem Beispiel:
Gegeben eine Boolesche Funktion in der KDNF:
$A\overline{B}CD + ABCD + AB\overline{C}D + \overline{A}BCD + \overline{A}B\overline{C}D = y$
$\quad r \quad + \quad s \quad + \quad t \quad + \quad u \quad + \quad v \quad = y$
Es soll minimiert werden!

Vorgehen:
- Ausklammern, so dass Formen $U(V + \overline{V})$ entstehen.
- Mehrstufig wiederholen.
- Allenfalls Terme verdoppeln, da $W + W = W$ gilt.

Für obiges Beispiel:

$$\underbrace{\begin{array}{ll} ACD(\overline{B} + B) & r + s \\ ABD(\overline{C} + C) & s + t \\ \overline{A}BD(\overline{C} + C) & u + v \end{array}}$$

$ACD + \underbrace{ABD + \overline{A}BD}_{BD(A + \overline{A})} = y$

$ACD + BD = y \quad [= D(AC + B)]$

Aufgaben:
Zeichnen Sie die minimierte Schaltung für $ACD + BD = y$.
Was bewirkt das Ausklammern bezüglich der Realisierung der Schaltung (Resultat in eckigen Klammern)?

Manchmal stehen nur 2-Input-Gatter zur Verfügung. Wie müsste dann die Schaltung realisiert werden?

Wie sieht die Schaltung aus, wenn entweder nur 2-Inpt-NOR oder nur 2-Input-NAND zur Verfügung stehen?

Graphisches Verfahren
Idee: Finden von Vereinfachungen erleichtern/automatisieren.

Wiederum an einem Beispiel erklärt:

	x_1	x_0	y
m_0	0	0	0
m_1	0	1	0
m_2	1	0	1
m_3	1	1	1

KDNF:

$$x_1\overline{x_0} + x_1x_0 = y = m_2+m_3$$
$$x_1(\overline{x_0} + x_0) = y$$
$$x_1 = y$$

Wie man aus der Tabelle erkennt, wäre dies sofort ersichtlich gewesen, sind doch die x_1- und die y-Spalte identisch.

Wir wollen uns nun Subterme der Form $(\overline{x_i} + x_i) = 1$ zunutze machen.

Karnaugh-Diagramme sind so aufgebaut, dass die Schaltvariablenwerte günstig nebeneinander liegen.
Für 2 Schaltvariablen in allgemeiner Form:

	$\overline{x_1}$	x_1
$\overline{x_0}$	$m_0 = \overline{x_1} \cdot \overline{x_0}$	$m_2 = x_1 \cdot \overline{x_0}$
x_0	$m_1 = \overline{x_1} \cdot x_0$	$m_3 = x_1 \cdot x_0$

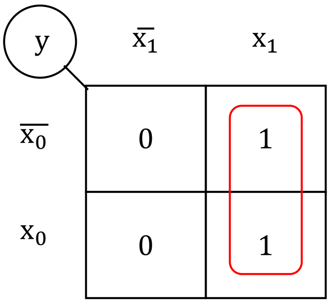
Für unser Beispiel:

	$\overline{x_1}$	x_1
$\overline{x_0}$	0	1
x_0	0	1

Feststellungen

- Jedes Feld repräsentiert einen Minterm.
- Die Felder sind so angeordnet, dass die benachbarten horizontalen oder vertikalen Felder nur in *einer* Schaltvariablen unterschiedlich sind.
- Dies gilt auch, wenn Felder als Nachbarn bezeichnet werden, die an horizontal oder vertikal entgegengesetzten Rändern liegen.

Verfahren zur Vereinfachung mittels Karnaugh-Diagrammen am obigen Beispiel illustriert:



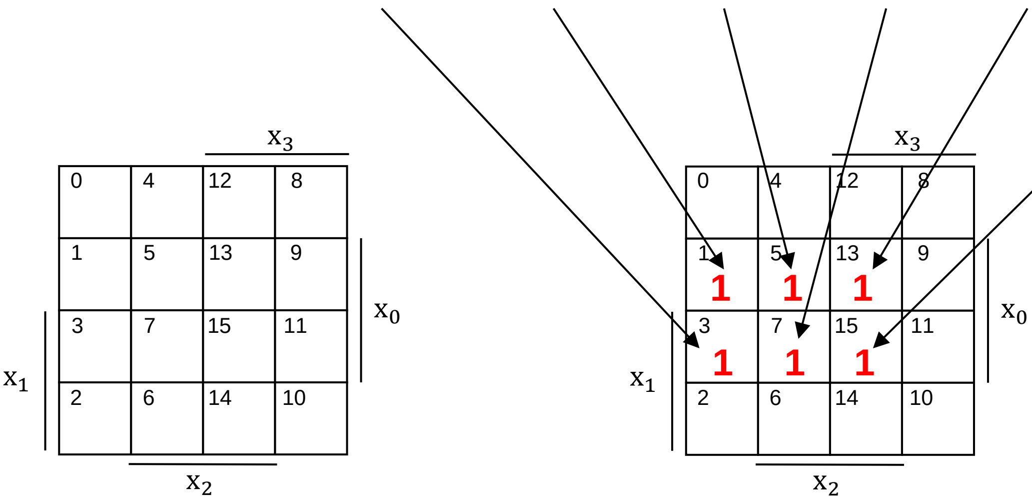
- Suchen von Superfeldern, bestehend aus benachbarten Feldern, die alle den Wert 1 aufweisen.
- Ein Superfeld muss die Dimension (Breite x Höhe) $2^{n-1} \times 2^{m-1}$ aufweisen ($n, m = 1, 2, 3$ ...).

Obiges Superfeld weist die Dimension 1×2 auf. $x_1(\overline{x_0} + x_0) = y$ lässt sich (via Minterme) gut ersehen. Die Nachbarschaft besagt nun, dass der Teil $(\overline{x_0} + x_0)$ weggelassen werden kann, somit $x_1 = y$ resultiert. Dieses Resultat lässt sich auch gleich an der Breite des Superfelds ablesen.

Weitere Beispiele sollen helfen, das Herauslesen vereinfachter Gleichungen zu festigen.

Beispiel 1 mit 4 Eingangsvariablen

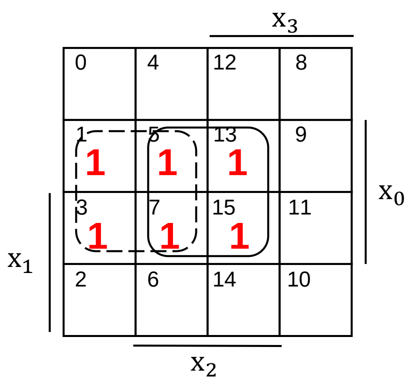
$$y = \overline{x_3}\,\overline{x_2}\,x_1\,x_0 + \overline{x_3}\,\overline{x_2}\,\overline{x_1}\,x_0 + \overline{x_3}\,x_2\,\overline{x_1}\,x_0 + \overline{x_3}\,x_2\,x_1\,x_0 + x_3\,x_2\,\overline{x_1}\,x_0 + x_3\,x_2\,x_1\,x_0$$



Die Feldzahlen entsprechen dem Index des Minterms, falls die Reihenfolge der Variablen $MSB = x_3, x_2, x_1, x_0 = LSB$ festgelegt wird.
Der Uebersichtlichkeit halber sind die Quer-Variablen am Diagrammrand nicht aufgetragen.

Da nur die "1" interessieren, werden die andern "0"-Felder der Uebersichtlichkeit halber nicht eingetragen.

Markieren der Superfelder des obigen Diagramms. Beachten Sie, dass Breite und Höhe nur 2er-Potenzen sein dürfen.

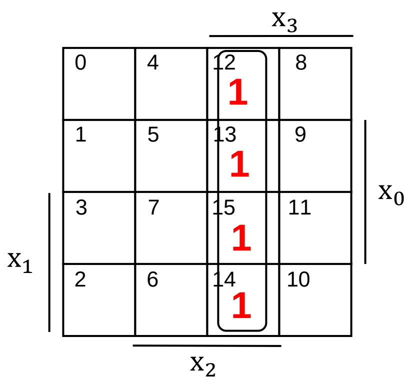


Bemerkungen:
- Möglichst grosse Superfelder bilden
- Möglichst wenige Superfelder bilden, aber "1" abgedeckt.
- Superfelder sollen sich überlagern, wenn dadurch grössere Felder gebildet werden können.

Herauslesen aus obigem Diagramm:
Gestricheltes Superfeld: $\overline{x_3}\, x_0$
Ausgezogenes Superfeld: $x_2\, x_0$

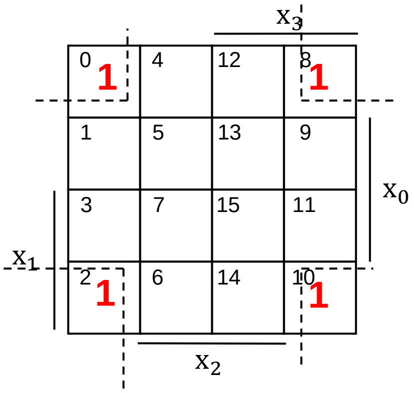
Die minimierte Funktion ergibt sich aus der Disjunktion herausgelesener Superfelder:
$y = \overline{x_3}\, x_0 + x_2\, x_0$

Beispiel 2 mit 4 Eingangsvariablen



Herauslesen der minimierten Funktion:
$y = x_3\, x_2$

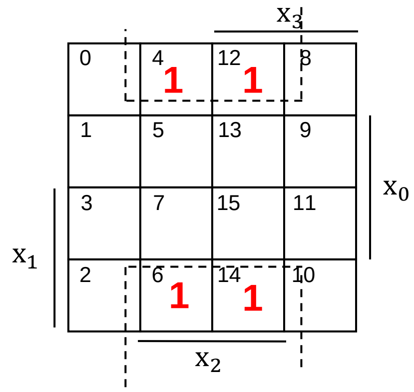
Weitere Fälle
Nachbarschaften via Ränder



Ein Superfeld der Dimension 2 x 2.

Wie lautet die vereinfachte Funktion?

y = !x2 * !x0



Ein weiteres Superfeld der Dimension 2 x 2.

Wie lautet die vereinfachte Funktion?

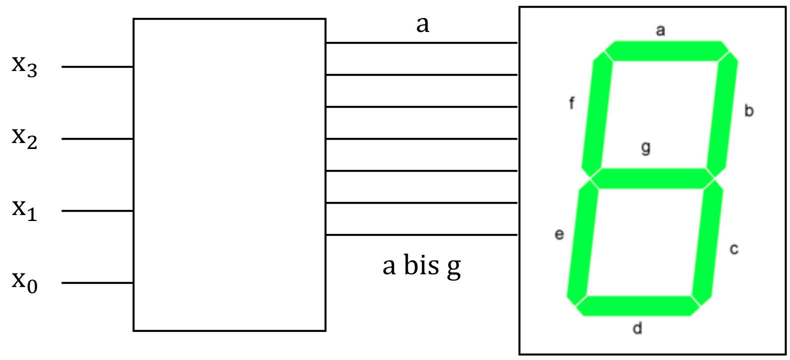
y = x2 * x0

Don't care-Werte

Es gibt Fragestellungen, wo es für einzelne Eingangspermutationen egal ist, ob eine "0" oder eine "1" angelegt wird. Die Eingangswerte solcher Permutationen heissen Don't cares und werden mit X gekennzeichnet. Bei Schaltungsvereinfachungen können sie in Karnaugh-Diagrammen vorteilhaft verwendet werden, um möglichst grosse Superfelder zu erhalten.

Beispiel dazu:
Es ist eine Schaltung zu entwerfen, die mit 4 Eingangsvariablen eine 7-Segmentanzeige zum Leuchten bringt. Dabei werde an den Eingängen nur BCD-Code verwendet. Die Schaltung soll minimiert werden.

Situation:

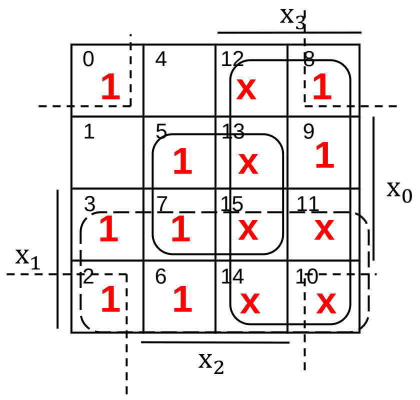


Aufgabe:
Für untenstehende Tabelle sind die Werte für "0" und alle Don’t cares eingetragen. Ergänzen Sie alle fehlenden Werte (vgl. Vereinbarung).

BCD	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g	Ziffer
0	0	0	0	0	1	1	1	1	1	1	0	"0"
1	0	0	0	1								"1"
2	0	0	1	0								"2"
3	0	0	1	1								"3"
4	0	1	0	0								"4"
5	0	1	0	1								"5"
6	0	1	1	0								"6"
7	0	1	1	1								"7"
8	1	0	0	0								"8"
9	1	0	0	1	1	1	1	1	0	1	1	"9"
-	1	0	1	0	x	x	x	x	x	x	x	
-	1	0	1	1	x	x	x	x	x	x	x	
-	1	1	0	0	x	x	x	x	x	x	x	
-	1	1	0	1	x	x	x	x	x	x	x	
-	1	1	1	0	x	x	x	x	x	x	x	
-	1	1	1	1	x	x	x	x	x	x	x	

Vereinbarung: 1 am Ausgang soll bedeuten: Segment der Anzeige leuchtet;
0 am Ausgang: Segment leuchtet nicht.

Die obigen x (Don't cares) können im Diagramm zu unseren Gunsten ausgenutzt werden.
Minimierung für das Segment a, KDNF.



$$a = x_3 + x_1 + x_2 x_0 + \overline{x_2}\,\overline{x_0}$$

Der besondere Nutzen der Karnaugh-Diagramme liegt im systematischen Einbeziehen der Don't cares.

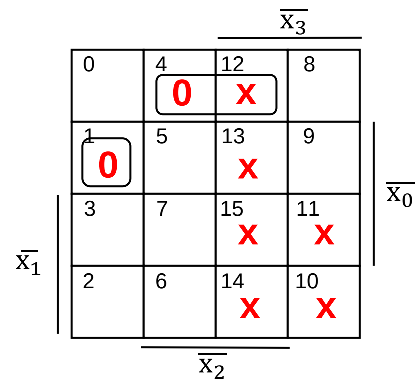
Obiger Fall lässt allenfalls vermuten, dass stets alle x = 1 gesetzt werden (mit Einbezug in die Superfelder). Dem ist jedoch nicht immer so.

Karnaugh-Diagramme für KKNF

Alle Betrachtungen wurden für die KDNF gemacht. Aus Dualitätsgründen kann mit entsprechend dualen Regeln auch eine KKNF-Funktion via Karnaugh-Diagramm minimiert werden.

Folgendes ist zu beachten:
- Die Beschriftung der Karnaugh-Diagramme an den Rändern muss invers geschehen.
- Werte werden wie üblich eingetragen.
- Beim Bilden der Superfelder sind jetzt 0 (nicht 1) einzukreisen.
- Werte werden wie gewohnt aus dem Diagramm gelesen, jedoch pro Superfeld als Disjunktion verknüpft, wenn mehr als eine Variable beteiligt ist.
- Resultate von mehreren Superfeldern sind konjunktiv zu verknüpfen.

Beispiel für das Segment a mit KKNF formuliert:



$$a = (\overline{x_2} + x_1 + x_0)(x_3 + x_2 + x_1 + \overline{x_0})$$

Vorlagen Karnaugh-Diagramme (KDNF)

KDNF, 2 Variablen: $\mathrm{MSB} = x_1,\ x_0 = \mathrm{LSB}$

x_1 (2. Spalte), x_0 (2. Zeile)

0	2
1	3

KDNF, 3 Variablen: $\mathrm{MSB} = x_2,\ x_1,\ x_0 = \mathrm{LSB}$

x_2 (Spalten 3–4), x_1 (Spalten 2–3), x_0 (2. Zeile)

0	2	6	4
1	3	7	5

KDNF, 4 Variablen: $\mathrm{MSB} = x_3,\ x_2,\ x_1,\ x_0 = \mathrm{LSB}$

x_3 (Spalten 3–4), x_2 (Spalten 2–3), x_1 (Zeilen 3–4), x_0 (Zeilen 2–3)

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

KDNF, 5 Variablen: $\mathrm{MSB} = x_4,\ x_3,\ x_2,\ x_1,\ x_0 = \mathrm{LSB}$

x_3 (Spalten 3–6), x_4 (Spalten 5–8), x_2 (Spalten 2–3 und 6–7), x_1 (Zeilen 3–4), x_0 (Zeilen 2–3)

0	4	12	8	24	28	20	16
1	5	13	9	25	29	21	17
3	7	15	11	27	31	23	19
2	6	14	10	26	30	22	18

Vorlagen Karnaugh-Diagramme (KKNF)

KKNF, 2 Variablen: $\mathrm{MSB} = x_1, x_0 = \mathrm{LSB}$

$\overline{x_1}$

	0	2
$\overline{x_0}$	1	3

KKNF, 3 Variablen: $\mathrm{MSB} = x_2, x_1, x_0 = \mathrm{LSB}$

$\overline{x_2}$

	0	2	6	4
$\overline{x_0}$	1	3	7	5

$\overline{x_1}$

KKNF, 4 Variablen: $\mathrm{MSB} = x_3, x_2, x_1, x_0 = \mathrm{LSB}$

$\overline{x_3}$

0	4	12	8	
1	5	13	9	$\overline{x_0}$
3	7	15	11	$\overline{x_0}$
2	6	14	10	

$\overline{x_1}$

$\overline{x_2}$

KKNF, 5 Variablen: $\mathrm{MSB} = x_4, x_3, x_2, x_1, x_0 = \mathrm{LSB}$

$\overline{x_3}$

$\overline{x_4}$

0	4	12	8	24	28	20	16	
1	5	13	9	25	29	21	17	$\overline{x_0}$
3	7	15	11	27	31	23	19	$\overline{x_0}$
2	6	14	10	26	30	22	18	

$\overline{x_1}$

$\overline{x_2}$ $\overline{x_2}$

Quellen

[1] Digitaltechnik, Klaus Fricke, Vieweg, 2005,
ISBN 3-528-33861-X