

Anmerkungen zur Booleschen Normalform

Von LOTHAR DILCHER

Wenngleich Booles Mathematik und ihre Anwendung mittlerweile einer abgegrasten Wiese gleichen, seien hier einige Anmerkungen zur (disjunktiven) Normalform erlaubt.

Zumindest in der leichter zugänglichen Literatur erfährt die Normalform meiner Meinung nach eine zu geringe Beachtung, was ihre Anwendung im Hinblick auf Standardisierung und Systematisierung betrifft. Diese kann sich sowohl auf die Elektronik (Hardware) als auch auf die Programmierung (Software) beziehen.

Es finden sich in der Literatur umfangreiche Ausführungen zum Thema der Minimalisierung Boolescher Funktionen. Diese Theorie und ihre Anwendung bergen außerordentliche Schwierigkeiten und führen selten zu großen Erfolgen. Zudem wurden sie in einer Epoche entwickelt, als die Systeme klein und die Bauteile teuer waren. Im Zeitalter der ICs könnten sich die Maßstäbe verändert haben.

Die Normalform ist fast nie das Minimum einer Booleschen Funktion, könnte aber von ihrer Idee her bei einer übergreifenden Vereinfachung Einsatz finden,

Einige Attribute

Für Boolesche Funktionen wurden Normalformen definiert, die diese möglichst universell darstellen sollen. Jede Boolesche Funktion ist in die disjunktive und/oder die konjunktive Normalform überführbar. Die beiden Normalformen sind austauschbar und können ineinander umgerechnet werden (s.a. Boolesche Algebra). Die Literatur benutzt die Normalformen oft als Beweis, dass die Boolesche Algebra bereits mit den Operatoren logisch NICHT sowie wahlweise logisch UND oder logisch ODER funktioniert. (Logik-Gatter von Schaltungstechniken werden daher oft unter ausschließlicher Verwendung von NOR bzw. NAND konzipiert).

Die disjunktive Normalform (DNF) fasst die Variablen in UND Verknüpfungen zu sogenannten MINtermen zusammen, die wiederum über ODER verknüpft den Funktionswert ergeben (er ist entweder 0 oder 1). Die konjunktive Normalform (KNF) fasst die Variablen in ODER Verknüpfungen zu sogenannten MAXtermen zusammen, die über eine UND Verknüpfung den Funktionswert ergeben.

Ein MINterm bzw. MAXterm enthält immer genau alle Variablen der zugehörigen Booleschen Funktion. Die Variablen liegen als Literale vor, d.h. entweder in bejahter oder in verneinter Form.

$$DNF = MINterm1 ODER MINterm2 ODER ...etc.$$

$$KNF = (MAXterm1) UND (MAXterm2) UND ...etc.$$

Eine vollständige Normalform wird in der Literatur häufig so definiert, dass sie so viele MINterme bzw. MAXterme enthält wie es unterschiedliche 0/1-Kombinationen der Variablen gibt. Wenn alle möglichen MINterme bzw. MAXterme in der Normalform einer Booleschen Funktion vorhanden sind – wie in der vollständigen Normalform – dann hat sie den Wert 1.

Die Anzahl unterschiedlicher Kombinationen beträgt 2^n , wenn n die Anzahl der Variablen ist (das bedeutet 2^n Literale). Eine beliebige Boolesche Funktion ergibt in eine Normalform umgewandelt immer eine Teilmenge der vollständigen Normalform. Die disjunktive Normalform verwendet nur

Terme, deren Literalen-Kombinationen den Funktionswert 1 ergeben. Die Terme mit dem Wert 0 werden in der disjunktiven Normalform nicht aufgeführt.

Die durchschnittliche Anzahl der MINterme, aus denen eine Boolesche Funktion in disjunktiver Normalform gebildet wird, kann als die Hälfte der möglichen Terme – also der vollständigen Normalform - angenommen werden. Das wären dann $0.5 \cdot 2^n = 2^{n-1}$ MINterme.

Normalform aus Wahrheitstafel

Eine Wahrheitstafel kann aus jeder Booleschen Funktion erstellt werden, indem die Variablenwerte (0/1) eingesetzt werden und der Funktionswert y ausgerechnet wird.

Dezimal Äquivalent	x_1	x_2	x_3	y
0	0	0	0	0
1	0	0	1	0
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

Abb. 1 Wahrheitstafel mit 3 Variablen(x_1, x_2, x_3) und Funktionswert y

Wir verwenden folgende Operatoren:

NICHT $A = \bar{A}$ oder $\neg A$

A UND B $= A \cdot B$

A ODER B $= A + B$

Wenn obige Tabelle so interpretiert wird, dass die Zeilenvariablen mit logisch UND verknüpft sind, so ergibt sich eine Boolesche Funktion in disjunktiver Normalform (DNF):

$$y = \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3$$

Da die Funktion y bei den Dezimal-Äquivalenten 3 und 5 den Wert 1 annimmt, kann sie auch als DNF 3, 5 geschrieben werden (s.a. unten)

Die Verneinung ergibt eine einfache Möglichkeit der Umwandlung in eine identische KNF:

$$\bar{y}_{DNF} = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3$$

$$\neg \bar{y} = y = \neg(\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3)$$

$$y_{KNF} = (x_1 + x_2 + x_3) \cdot (x_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + \bar{x}_3) \cdot (\bar{x}_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$$

Zur Bildung der Verneinung werden die Zeilen der Abb. 1 mit $y=0$ als MINterme angeschrieben. Die Anwendung von de Morgans Regel wandelt die Darstellung in eine KNF um. Die Anzahl der benötigten MINterme addiert sich mit der Anzahl der MAXterme immer zu 2^n auf.

Man gewinnt so eine alternative Darstellung der Funktion, die gegebenenfalls einsetzbar ist. Wenn eine Boolesche Funktion in DNF sehr umfangreich ist und weit mehr als die Hälfte der möglichen Kombinationen beansprucht, empfiehlt es sich auf die KNF auszuweichen, da dann die Zahl der Terme $< 2^{n-1}$ ist.

Aus Wahrheitstafeln ist also immer eine Normalform ableitbar.

Normalform aus Entwicklungssatz

Zur Umwandlung einer beliebigen Booleschen Funktion in eine Normalform empfiehlt die Literatur immer die Anwendung des Entwicklungssatzes von Boole (oder Shannon). Da bei diesem Verfahren auch mit der „node explosion“ zu kämpfen ist, sollen im Beispiel nur bescheidene 3 Variable (6 Literale) auftreten. Das ergibt maximal $2^3 = 8$ Terme.

Hier das Entwicklungsrezept zur disjunktiven Normalform (DNF):

Die Boolesche Funktion $y = f(x_1, \bar{x}_1, \dots, x_i, \bar{x}_i, \dots, x_n, \bar{x}_n)$

nach x_i und $\bar{x}_i$ entwickelt ergibt

$$y = x_i \cdot f_{x_i} + \bar{x}_i \cdot f_{\bar{x}_i} \quad \text{wobei}$$

$$f_{x_i} = f(x_1, \bar{x}_1, \dots, x_i = 1, \bar{x}_i = 0, \dots, x_n, \bar{x}_n) \quad \text{und} \quad f_{\bar{x}_i} = f(x_1, \bar{x}_1, \dots, x_i = 0, \bar{x}_i = 1, \dots, x_n, \bar{x}_n)$$

($\cdot$ ist logisch UND; $+$ ist logisch ODER, f_{x_i} ist Kofaktor)

Wird die Entwicklung mit allen Variablen (Literalen) durchgeführt, ergibt sich folgendes Bild (3 Variable, f_{x_i} wird zu f_1 abgekürzt)

$$\text{nach } x_1: \quad y = x_1 \cdot f_1 + \bar{x}_1 \cdot f_{\bar{1}}$$

$$\text{nach } x_2: \quad y = x_1 \cdot x_2 \cdot f_{12} + x_1 \cdot \bar{x}_2 \cdot f_{1\bar{2}} + \bar{x}_1 \cdot x_2 \cdot f_{\bar{1}2} + \bar{x}_1 \cdot \bar{x}_2 \cdot f_{\bar{1}\bar{2}}$$

$$\text{nach } x_3: \quad y = x_1 \cdot x_2 \cdot x_3 \cdot f_{123} + x_1 \cdot x_2 \cdot \bar{x}_3 \cdot f_{12\bar{3}} + x_1 \cdot \bar{x}_2 \cdot x_3 \cdot f_{1\bar{2}3} + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot f_{1\bar{2}\bar{3}} + \\ \bar{x}_1 \cdot x_2 \cdot x_3 \cdot f_{\bar{1}23} + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot f_{\bar{1}2\bar{3}} + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot f_{\bar{1}\bar{2}3} + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot f_{\bar{1}\bar{2}\bar{3}}$$

Ein MINterm erscheint nur dann in der DNF, wenn der Kofaktor f den Wert 1 hat.

Die disjunktive Normalform einer Booleschen Funktion wird also von der Anzahl n der Variablen und dem Wert der Kofaktoren f_i (0 oder 1) bestimmt.

Der formale Aufbau der Normalform ist für alle möglichen Booleschen Funktionen gleich. Die 0/1-Kombinationen der n Variablen können so geordnet werden, dass sie die Dualzahlen von 0 bis $2^n - 1$ ergeben, was sozusagen eine Durchnummerierung der MIN/MAXterme ergibt.

Die Faktoren f (Kofaktoren) gehorchen dem gleichen Schema. Sie werden aus der Booleschen Originalfunktion gebildet, indem die Literalen nach denen sie entwickelt wird mit 0 bzw. 1 bewertet wer-

den. Die Kofaktoren f ergeben dann entweder den Wert 0 oder den Wert 1. Bei $f=1$ wird der zugehörige Term in die Normalform aufgenommen, bei $f=0$ entfällt er.

Daher wird in der Literatur eine bestimmte DNF häufig mit z.B. DNF 1,2,3,4... gekennzeichnet, wobei die Zahlen die dezimalen Äquivalente der als Dualzahlen interpretierten, 0/1-Kombinationen der Variablen sind. Bei dieser Kennzeichnung muss immer die Zuordnung der Variablen zur Stelle der „Dualzahl“ beachtet werden. In den folgenden Beispielen ist die Zuordnung wie in Abb.1 gewählt:

$$x_1=2^2, x_2=2^1, x_3=2^0.$$

Beispiel 1:

Die 2 von 3 Schaltung – die knappste Mehrheitsentscheidung:

$$\text{Originalfunktion: } y = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

Volständige, entwickelte DNF :

$$y = x_1 \cdot x_2 \cdot x_3 \cdot f_{123} + x_1 \cdot x_2 \cdot \bar{x}_3 \cdot f_{12\bar{3}} + x_1 \cdot \bar{x}_2 \cdot x_3 \cdot f_{1\bar{2}3} + x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot f_{1\bar{2}\bar{3}} + \\ \bar{x}_1 \cdot x_2 \cdot x_3 \cdot f_{\bar{1}23} + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot f_{\bar{1}2\bar{3}} + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot f_{\bar{1}\bar{2}3} + \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot f_{\bar{1}\bar{2}\bar{3}}$$

$$f_{123} = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 1$$

$$f_{12\bar{3}} = 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 = 1$$

$$f_{1\bar{2}3} = 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 = 1$$

$$f_{1\bar{2}\bar{3}} = 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 = 0$$

$$f_{\bar{1}23} = 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 = 1$$

$$f_{\bar{1}2\bar{3}} = 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 = 0$$

$$f_{\bar{1}\bar{2}3} = 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 = 0$$

$$f_{\bar{1}\bar{2}\bar{3}} = 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 0$$

Die DNF ist dann:

$$y = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3$$

Sie hat die Bezeichnung DNF 3,5,6,7

Beispiel 2:

Addition einer Stelle mit Übertrag (x_1 und x_2 sind die Summanden, x_3 ist der eingehende Übertrag):

$$y_{\text{SUMME}} = x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

$$y_{\text{ÜBERTRAG}} = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3$$

Die Booleschen Funktionen ergeben sich bereits aus der Aufgabenstellung als DNFs:

y_{SUMME} ist DNF 1,2,4,7

Y_{ÜBERTRAG} ist DNF 3,5,6,7 (wie 2v3)

Schaltung zur Ermittlung des Funktionswerts einer DNF

Beispiel: 8 Variable mit zeitgleicher Verarbeitung der DNF-Nummern

Im Folgenden wird ein Beispiel für eine Möglichkeit des Schaltungsaufbaus gezeigt. Es wird eine Parallelverarbeitung von 8 Bit angenommen, also MINterme mit 8 Eingängen. Die Boolesche Funktion soll bis zu 8 Variablen besitzen. Ist die Anzahl der Variablen < 8 , so wird mit Einsen aufgefüllt. Da bei 8 Bit insgesamt $2^8 = 256$ unterschiedliche 1/0-Kombinationen existieren, sollen auch 256 entsprechend unterschiedlich codierte MINterme als Einheit zur Verfügung stehen. Jeder MINterm zeigt am Ausgang eine 1, wenn sein Eingangs-Code erfüllt ist. Der Code ist immer die DNF-Nummer als Dualzahl.

Die Variablen dienen als Eingangsgrößen und werden in ein Register geladen. Die Ausgänge des Registers sind mit den Eingängen der 256 MINterme verbunden. Die MINterm-Ausgänge werden über je einen Schalter mit dem ODER-Glied der DNF verbunden. Nur die Schalter, die zu einem MINterm mit dem Kofaktor 1 gehören werden leitend geschaltet. Der Ausgangswert des ODER-Glieds ist gleich dem Funktionswert der Booleschen Funktion $y=f(x)$.

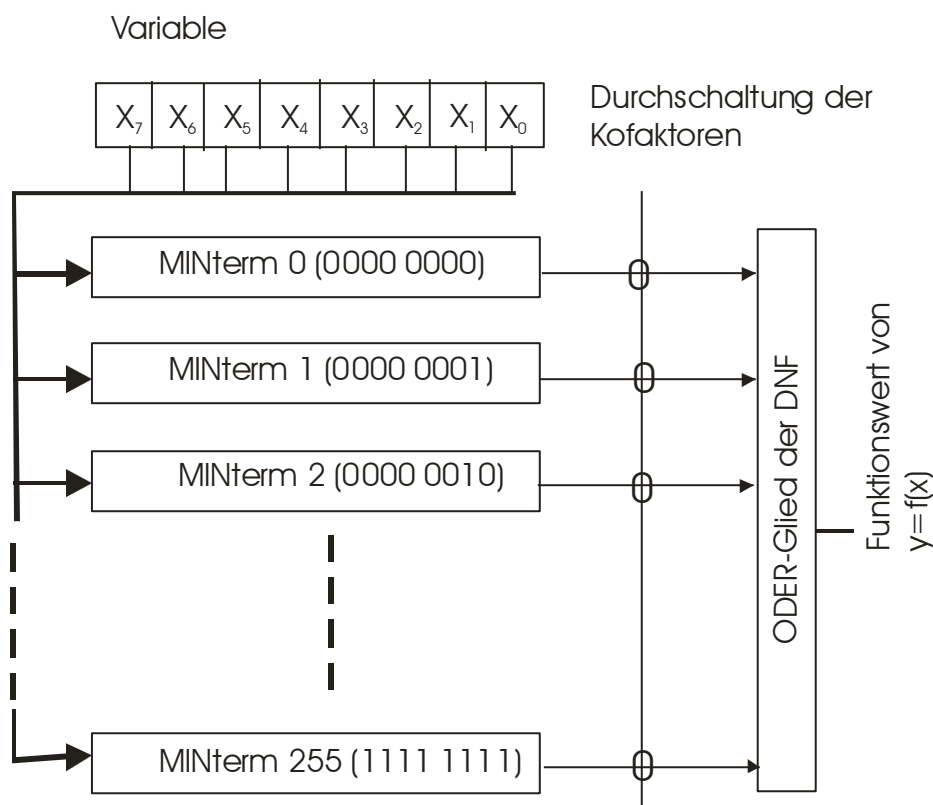


Abb. 2 Schema zur Parallel-Ermittlung des Funktionswerts mit 256er MINterm Einheit

Alternative Beschaltung der MINterm-Träger

Eine weitere Möglichkeit ist die Beschaltung ist der MINterme mit 2er Koinzidenzschaltungen am Eingang von UND-Gattern. Hierdurch wird erreicht, dass alle MINterme identisch sind.

Der eine Eingang der Koinzidenzschaltung wird von einer Variablen der Booleschen Funktion belegt, der andere von der zugehörigen Stelle der dualen DNF-Nummer aus dem Definitions-Programm der entsprechenden Normalform.

Damit wird festgestellt, ob das Variablen-Set mit der jeweiligen DNF-Nummer identisch ist. (Der Istwert einer Variablen muss geprüft werden, ob er mit dem vorgesehenen Wert der DNF-Nummer übereinstimmt. Dazu dient die Koinzidenzschaltung (auch Äquivalenz).

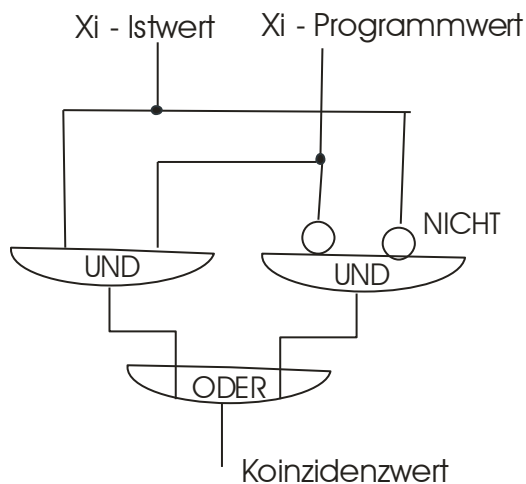


Abb 3 Koinzidenzschaltung am MINterm-Eingang

Der Istwert der Variablen wird mit dem zugehörigen programmierten Wert des aktuellen MINterms auf Koinzidenz verglichen. Sind die Koinzidenzwerte eines MINterms alle gleich 1, so ist die Boolesche Funktion gleich 1 und die Aufgabe ist gelöst. Ist eine Koinzidenz gleich 0, so ist der MINterm gleich 0. Sind alle relevanten MINterme der DNF gleich 0, so ist der Boolesche Funktionswert gleich 0 und die Aufgabe ist ebenfalls gelöst.

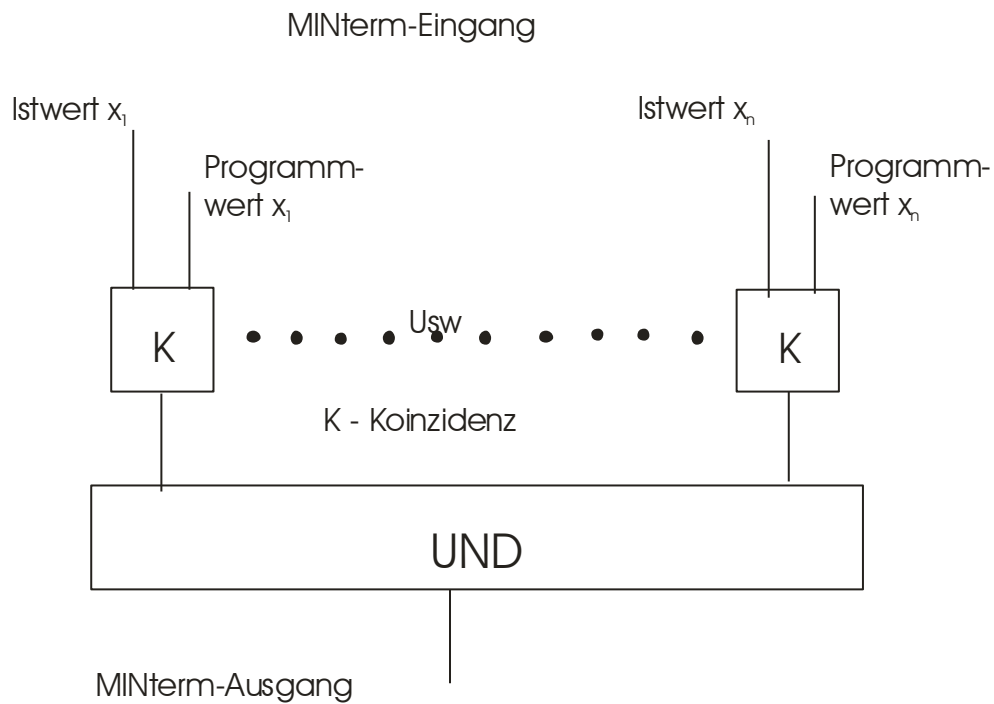


Abb. 4 MINterm mit Eingangs-Koinzidenzschaltung.

Ermittlung des Funktionswertes mit Koinzidenzschaltung

Beispiel: 8 Variable und serielle Beaufschlagung mit dualer DNF-Nummer

Die Ermittlung des Funktionswertes kann seriell und/oder parallel durchgeführt werden. Bei serieller Ermittlung wird das DNF-Register nacheinander mit den zugehörigen DNF-Zahlen geladen.

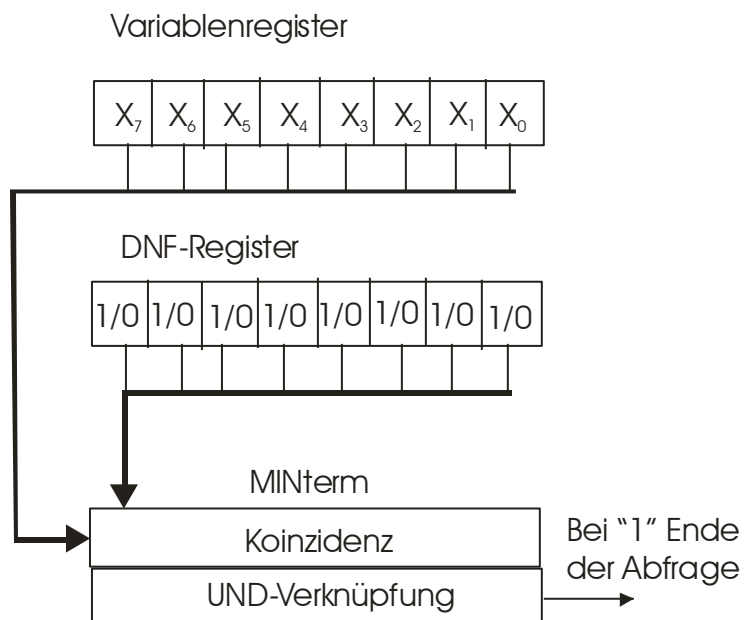


Abb. 5 Ermittlung des Funktionswertes mit serieller Abfrage der DNF-Zahlen.

Anwendungs-Aspekte in der Digitaltechnik

Die Idee der Normalform legt eine technische Anwendung für digitale Systeme nahe, die eine entsprechende Normierung der Elektronik sowie deren Programmierung zum Inhalt hat. Die Normalform ist zwar selten die Minimalform, verspricht jedoch von der Idee her eine übergreifende Vereinfachung der Schaltungstechnik, ihrer Vernetzung und der Programmierung.

Zum Aufbau einer Normalform Logik-Einheit in einem Automaten wären beispielsweise hauptsächlich MINterm-Einheiten vorzusehen, die mit Daten versorgt werden, welche wesentlich aus den Variablen und zugehörigen DNF-Nummern beständen.

Die Anzahl der MINterm-Einheiten könnte denkbarerweise im Bereich der Byte-Speicherzellen liegen. Der Aufwand wäre ungefähr gleich. Ein Automatenkonzept mit Normalformanwendung würde also eine große Anzahl MIN-/Maxterme beinhalten, die flexibel zur Ermittlung der jeweiligen Booleschen Funktionswerte verschaltet werden können. Parallel- und Serienbearbeitung sollte dem Umfang der Aufgabe angepaßt werden („node explosion“). Die Vielzahl der benötigten MINterme legt auch die Anwendung der zeitsparenden Staffelnung („pipelining“) nahe.

8-bit=1 Byte UND-Gatter als MINterm Träger mit der Option auf Erweiterung wären sicher auch für Normalform-Rechner geeignet (ähnlich der Zeichenverarbeitung im ANSI-Code mit der parallelen Grundeinheit des Byte). Es wäre auch immer Flexibilität zu weiterer Parallelisierung gegeben.

Die Umsetzung beliebiger Boolescher Funktionen in Normalformen könnte die Aufgabe des Interpreters/Compilers werden.

Als mögliches Anwendungsfeld für das Normalformkonzept erscheinen Prozssrechner mit ihrer Boole-bezogenen Aufgabenstellung geeignet, also im einfachsten Fall Steuerung von Schaltanlagen, sonstigen technischen Steuerungen oder „imbedded systems“ für Waschmaschinen o.ä.

Beispiel: : Verarbeitung einer 64-bit Variablen mit 8-bit MINtermen

Die Variable wird in 8 Teilabschnitte unterteilt, ebenso die dualen DNF-Nummern. Es werden 8 MINterm-Spalten zu je 256 Stück benötigt. Die Abschnitte der DNF-Nummern bestimmen die ko-Faktor-Durchschaltung. Es wird jeweils eine Gesamt-DNF-Nummer auf „1“ geprüft. Eine zu große Anzahl von DNF-Nummern würde zu großen Rechenzeiten führen. (z.B. 2^{30} Möglichkeiten würden bei einer Taktfrequenz von 1GHz zu 1 Sekunde Rechenzeit führen.)

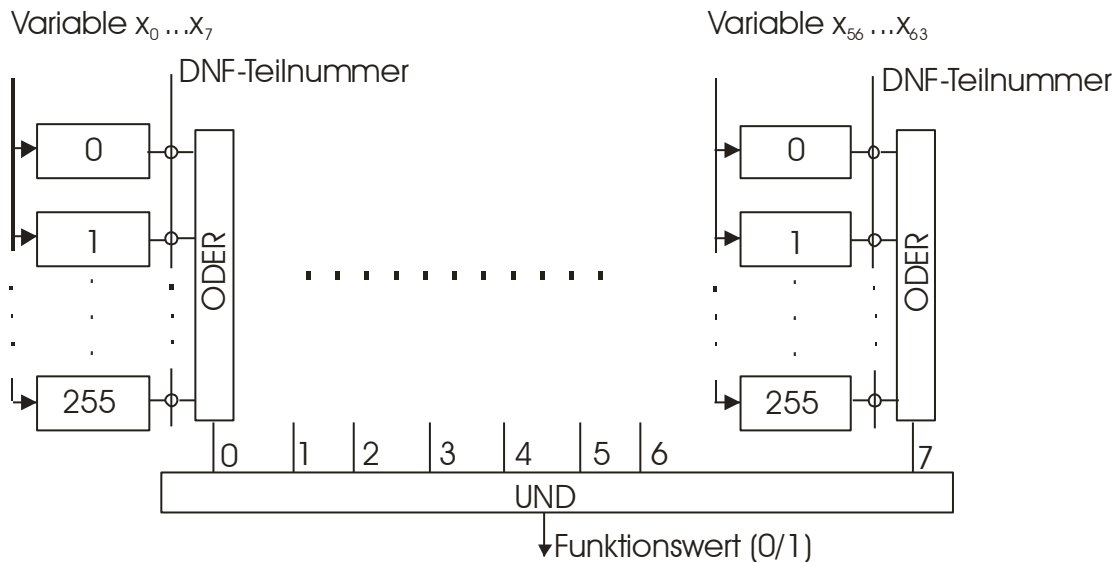


Abb. 6) Ermittlung des Funktionswerts

Hier wird jeder 64stelligen DNF-Nummer ein Ermittlungsschritt zugeordnet, indem die zugehörigen DNF-Teilnummern gleichzeitig durchgeschaltet werden..

Beispiel: Teiladdierer für 3 Stellen mit 8-bit MINterm

Die Summanden A und B mit je 3 Stellen sowie 1 Übertrag füllen 7 Plätze eines 8-bit MINterms. Die 8. Stelle wird mit 0 belegt oder mit 1, je nachdem welche Hälfte der Wahrheitstafel man nutzen will. Die Ausgangssignale (Boolesche Funktionen) sind 3 Summen plus 1 Übertrag. Die DFN-Nummern liegen fest. Die Anzahl der MINterme ist wieder 256.

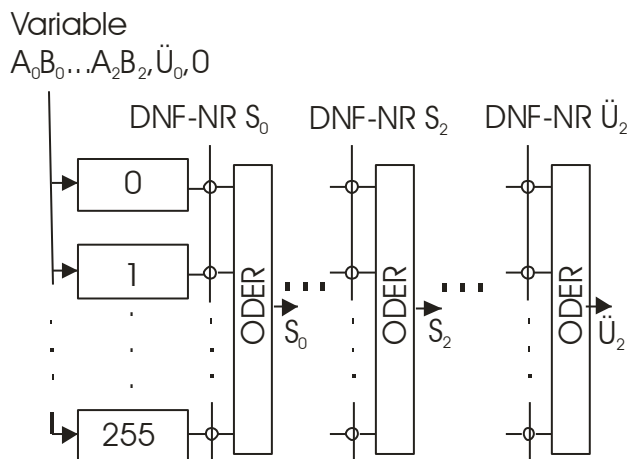


Abb7) Teiladdierer für 3 Stellen mit Übertrag

Die gesamte Addition bräuchte etwas mehr als 20 Schritte, um 2 64bit-Worte zu addieren.

Beispiel: Teiladdierer für 7 Stellen mit 16bit-MINterm

Die Anzahl der MINterme ist 65536

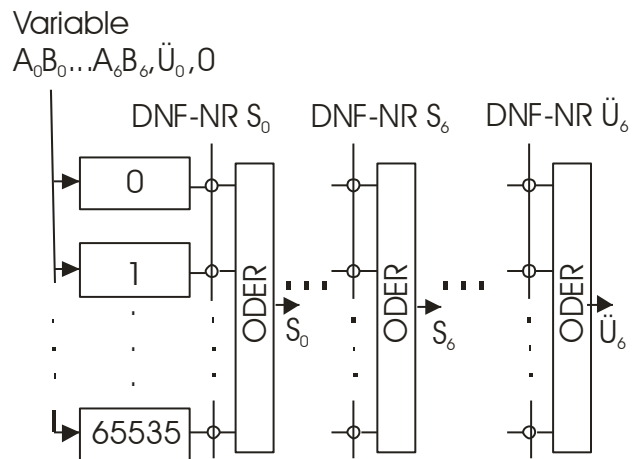


Abb8) Teiladdierer für 7 Stellen mit Übertrag

Mit diesem Aufwand wäre man bei <10 Schritten, um 2 64bit-Worte zu addieren. Kogge-Stone braucht hierfür auch mindestens 6 Schritte. Man ist also in der Spitzengruppe.