Informatik

Tonja Pfeiffer / Stefan Eckrich

Über die multi-level Synthese von EXOR-Schaltkreisen

Die Studie ist aufgrund des Seitenumfangs nur digital erhältlich (CD oder Download)

Diplomarbeit

Diplom.de

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de/ abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlages. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 1996 Diplom.de
ISBN: 9783832406448

https://www.diplom.de/document/216574

Tonja Pfeiffer, Stefan Eckrich

Über die multi-level Synthese von EXOR-Schaltkreisen

Die Studie ist aufgrund des Seitenumfangs nur digital erhältlich (CD oder Download)

Diplom.de

Stefan Eckrich
Tonja Pfeiffer

Über die multi-level Synthese von EXOR-Schaltkreisen

Diplomarbeit
an der Johann Wolfgang Goethe-Universität Frankfurt am Main
April 1996 Abgabe

***Diplomarbeiten* Agentur**
Dipl. Kfm. Dipl. Hdl. Björn Bedey
Dipl. Wi.-Ing. Martin Haschke
und Guido Meyer GbR

Hermannstal 119 k
22119 Hamburg

agentur@diplom.de
www.diplom.de

ID 644

ID 644
Eckrich, Stefan / Pfeiffer, Tonja: Über die multi-level Synthese von EXOR-Schaltkreisen / Stefan Eckrich / Tonja Pfeiffer - Hamburg: Diplomarbeiten Agentur, 1998
Zugl.: Frankfurt am Main, Universität, Diplom, 1996

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden, und die Diplomarbeiten Agentur, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Dipl. Kfm. Dipl. Hdl. Björn Bedey, Dipl. Wi.-Ing. Martin Haschke & Guido Meyer GbR
Diplomarbeiten Agentur, http://www.diplom.de, Hamburg
Printed in Germany

Diplomarbeiten Agentur

Wissensquellen gewinnbringend nutzen

Qualität, Praxisrelevanz und Aktualität zeichnen unsere Studien aus. Wir bieten Ihnen im Auftrag unserer Autorinnen und Autoren Wirtschaftsstudien und wissenschaftliche Abschlussarbeiten – Dissertationen, Diplomarbeiten, Magisterarbeiten, Staatsexamensarbeiten und Studienarbeiten zum Kauf. Sie wurden an deutschen Universitäten, Fachhochschulen, Akademien oder vergleichbaren Institutionen der Europäischen Union geschrieben. Der Notendurchschnitt liegt bei 1,5.

Wettbewerbsvorteile verschaffen – Vergleichen Sie den Preis unserer Studien mit den Honoraren externer Berater. Um dieses Wissen selbst zusammenzutragen, müssten Sie viel Zeit und Geld aufbringen.

http://www.diplom.de bietet Ihnen unser vollständiges Lieferprogramm mit mehreren tausend Studien im Internet. Neben dem Online-Katalog und der Online-Suchmaschine für Ihre Recherche steht Ihnen auch eine Online-Bestellfunktion zur Verfügung. Inhaltliche Zusammenfassungen und Inhaltsverzeichnisse zu jeder Studie sind im Internet einsehbar.

Individueller Service – Gerne senden wir Ihnen auch unseren Papierkatalog zu. Bitte fordern Sie Ihr individuelles Exemplar bei uns an. Für Fragen, Anregungen und individuelle Anfragen stehen wir Ihnen gerne zur Verfügung. Wir freuen uns auf eine gute Zusammenarbeit

Ihr Team der *Diplomarbeiten* Agentur

Dipl. Kfm. Dipl. Hdl. Björn Bedey –
Dipl. Wi.-Ing. Martin Haschke –
und Guido Meyer GbR –

Hermannstal 119 k –
22119 Hamburg –

Fon: 040 / 655 99 20 –
Fax: 040 / 655 99 222 –

agentur@diplom.de –
www.diplom.de –

So eine Arbeit wird eigentlich nie fertig,
man muß sie für fertig erklären,
wenn man nach Zeit und Umständen
das möglichste getan hat.

(J.W.Goethe, Italienische Reise, 16. März 1787)

Erklärung

Hiermit bestätigen wir, daß die vorliegende Arbeit ausschließlich unter Zuhilfenahme der im Literaturverzeichnis aufgeführten Unterlagen durch unsere selbständige Arbeit entstanden ist.

Frankfurt am Main, den 11. April 1996

Stefan Eckrich
Fliederstraße 24
63456 Hanau

Tonja Pfeiffer
Rhönstraße 4
65597 Hünfelden

Danksagung

An dieser Stelle möchten wir Herrn Prof. Dr. Bernd Becker unseren besonderen Dank für die Möglichkeit aussprechen, ein solch aktuelles und interessantes Thema im Rahmen einer gemeinsamen Diplomarbeit bearbeiten zu können.

Herrn Dipl. Inform. Harry Hengster gilt unser Dank für die vielen Anregungen und Ideen, für die weiterführenden Gespräche und das Korrekturlesen des ersten Entwurfes dieser Arbeit. Weiter sei allen Korrekturlesern und den Mitarbeitern der Professur gedankt.

Benni, Olf, Rolf und Markus danken wir für die Lösung vieler Probleme der Programmierung. Herr Ulrich Plass bewies viel Geduld, wenn wir so oft mit Problemen mit der Rechneranlage zu ihm kamen.

Besonderer Dank gilt unseren Eltern, die uns ein Studium frei von finanziellen Sorgen ermöglichten.

Inhaltsverzeichnis

1 Einleitung **1**

2 Boolesche Funktionen und Entscheidungsdiagramme **3**

2.1 Einleitung 3

2.2 Grundlegende Definitionen 3

2.3 Abhängigkeiten und quasireduzierte OKFDDs 11

3 Die Abhängigkeit zur terminalen Null **21**

3.1 Die allgemeine τ – Funktion 21

3.2 Die σ – Funktion 23

3.3 Nicht komplette OKFDDs 27

4 OKFDD basierte Schaltkreissynthese **31**

4.1 Die OKFDD – Darstellung über Abhängigkeiten 31

4.2 Beschreibung von Schaltkreisen 33

4.3 Verschiedene OKFDD – Schaltkreise 35

4.3.1 Von Abhängigkeiten zu Schaltkreisen 35

4.3.2 Die Knotenersetzung 37

4.3.3 Knotenersetzung und Abhängigkeit 38

4.3.4 Schaltkreise logarithmischer Tiefe 40

4.3.5 Oder und exklusiv–Oder Operation / Abhängikeit 42
4.4 Verbesserungen der Schaltkreissynthese 46
4.4.1 Levelübergreifende Kanten 46
4.4.2 Erkennung redundanter Gatter 50
4.5 Effizienz der Synthese 53
4.5.1 Komplementmarken entfernen 54
4.5.2 Vollständiges Quasireduzieren 55
4.5.3 Partielles Quasireduzieren 56
4.5.4 Die Erzeugung der Schaltkreisbeschreibungen . . 57
4.5.5 Die Gesamtlaufzeit 60

5 Die BMM – Reihenfolge 63
5.1 Speicherung einer BMM – Reihenfolge 64
5.2 Streng lineare BMM – Reihenfolge 65
5.3 Logarithmische Tiefe 66
5.4 Strategien zu besseren Freilosverteilungen 67
5.4.1 Die Matrixketten Multiplikation 67
5.4.2 Der Freilos Algorithmus 70
5.4.3 Zufällige Verteilung der Freilose 72

6 Größe der Schaltkreise 75
6.1 Vollständige Schaltkreise 75
6.2 Reduzierte Schaltkreise 77
6.3 Beeinflussung von OKFDDs 79
6.4 Beeinflussung der Schaltkreisgröße 80
6.4.1 Gatteranzahl als Kostenmaß 81
6.4.2 Matrixbelegung als Kostenmaß 82

6.4.3 Probleme bei der Implementierung 83

7 Datenstrukturen für OKFDDs und Schaltkreise 85

7.1 Eine Datenstruktur für OKFDDs 85

7.2 Eine Datenstruktur zur Schaltkreisbeschreibung 88

8 Implementierung der Schaltkreissynthese 91

8.1 Systemkompatiblität 91

8.2 Implementierung der Datenstruktur zur Synthese in C++ 92

8.3 Vorbereitungen 94

8.3.1 Entfernen der Komplementmarken 95

8.3.2 Quasireduzieren 97

8.4 Erzeugen von Schaltkreis – Beschreibungen 103

8.4.1 Parameter 104

8.4.2 Zellnamenskonventionen 108

8.4.3 Eine Klasse für Abhängigkeitsmatrizen 110

8.4.4 Primäre Abhängigkeitsmatrizen 111

8.4.5 Sekundäre Abhängigkeitsmatrizen 112

8.4.6 Terminale Abhängigkeitsmatrizen 114

8.4.7 Abschließende Optimierungen 115

8.5 Verifizieren der Korrektheit 116

8.5.1 Verifizieren der Vorbereitungen an den OKFDDs . 117

8.5.2 Verifizieren der erzeugten Schaltkreise 117

8.6 Öffentliche Variablen und Funktionen 119

8.6.1 Elementvariablen der Klasse OKFDD_to_TC . . . 119

8.6.2 Methoden der Klasse OKFDD_to_TC 123

8.6.3 Hilfsfunktionen und globale Konstrukte 129

8.6.4 Eine besondere Boolesche Matrixklasse 133
8.6.5 Eine Klasse ganzzahliger Matrizen 135
8.6.6 Die Klasse der Abhängigkeitsmatrizen 136

9 Ein Synthesebeispiel **139**

10 Meßergebnisse **147**

10.1 Größenveränderung an OKFDDs und OBDDs 149
10.2 Belegung der Abhängigkeitsmatrizen 155
10.3 Vergleich der Synthesemethoden 158
10.3.1 Auswirkungen der BMM–Reihenfolgen 158
10.3.2 Vergleich zwischen OKFDD und OBDD 163
10.3.3 Logiktyp und Schaltkreisgröße 166
10.3.4 Vergleich mit anderen Synthesemethoden 169
10.4 Synthesezeiten 180
10.5 Gatteranzahl in vollständigen Schaltkreisen 184
10.6 Sifting nach geringsten BMM – Kosten 186

11 Zusammenfassung **189**

A Listing **191**

A.1 Das Makefile 191
A.2 main.h 191
A.3 main.c 193
A.4 step.h 197
A.5 step.c 211

B Listing der Messprogramme **283**

B.1 Größe der DDs 283

B.1.1 Absolute Knotenzahlen 283

B.1.2 Größenwachstum 285

B.2 Zeilenbelegung der Abhängigkeitsmatrizen 287

B.3 Vergleich OKFDDs und OBDDs 289

Literaturverzeichnis **291**

Kapitel 1

Einleitung

Der Einzug Integrierter Schaltkreise in alle Bereiche unseres Lebens stellt immer höhere Anforderungen an den Entwurf dieser Schaltkreise. Toaster mit Hitzesensoren, Vorbackautomatik und vollautomatischen Krümmelschubladen ebenso wie immer leistungsfähigere Personal Computer erfordern immer mehr und immer komplexere Schaltkreise. Deshalb gewinnen Werkzeuge zum automatisierten Entwurf von Schaltkreisen immer mehr an Bedeutung.

Von verschiedenen Ausgangspunkten aus werden Methoden entwickelt, Schaltkreise nach vorgegebenen Kriterien zu synthetisieren. Die Anforderungen an einen Schaltkreis lassen sich durch Boolesche Funktionen beschreiben. Viele Boolesche Funktionen lassen sich mit Hilfe von Entscheidungsdiagrammen effizient repräsentieren und manipulieren.

Der in dieser Arbeit vorgestellte Ansatz zum Entwurf von Schaltkreisen nutzt die Darstellung Boolescher Funktionen als Entscheidungsdiagramme. Um eine hohe Fexiblität zu erreichen, verwenden wir eine möglichst allgemeine Menge der Entscheidungsdiagramme, die Menge der *Orderd Kronecker Functional Desicion Diagrams, OKFDDs*.

Die Umsetzung von OKFDDs in Schaltkreisbeschreibungen kann über verschiedene Wege erfolgen. Diese werden durch die Anforderungen, die später an die entworfenen Schaltkreise gestellt werden, bestimmt. So ist es für ein OKFDD zum Beispiel möglich, jeden Knoten durch einen be-

stimmten Teilschaltkreis zu ersetzen. Ein Nachteil dieser Methode liegt in der so entstehenden Schaltkreistiefe, die proportional zur Anzahl der Eingangsvariablen ist. Dadurch entstehen sehr lange Signallaufzeiten, so daß die maximale Taktrate für diesen Schaltkreis gering sein wird.

Wir wollen nun einen variablen Ansatz zum automatischen Entwurf von Schaltkreisen vorstellen und zeigen eine mögliche Implementierung in C++. Dieser Ansatz führt zu einer geringeren Schaltkreistiefe und somit kürzeren Signallaufzeiten. Die meisten der auf diesem Wege erzeugten Schaltkreise zeigen eine nur leicht erhöhte Größe gegenüber den durch die im voherigen Absatz genannte Methode erzeugten Schaltkreisen.

Schon während der Ausarbeitung wurden Teile der vorliegenden Arbeit in [Hengster95] veröffentlicht.

Kapitel 2

Boolesche Funktionen und Entscheidungsdiagramme

2.1 Einleitung

Wir betrachten Entscheidungsdiagramme zur kompakten Repräsentation Boolescher Funktionen. In diesem Kapitel stellen wir die Definitionen verschiedener Entscheidungsdiagramme vor und zeigen deren Eigenschaften, auf die spätere Kapitel aufbauen. Entscheidungsdiagramme bilden die Grundlage unseres Verfahrens zur Schaltkreissynthese.

2.2 Grundlegende Definitionen

Sei $X_n := \{x_0, x_1, \ldots, x_{n-1}\}$ eine Menge Boolescher Variablen und sei ferner
$f(X_n) : \{0,1\}^n \rightarrow \{0,1\}$ eine Boolesche Funktion über X_n. Eine gerichtete Kante soll von ihrem Startpunkt zu ihrem Endpunkt gerichtet sein. Start– und Endpunkt sind somit Knoten.

Definition 2.1 *Ein* ***Entscheidungsdiagramm*** *(****Decision Diagram DD****) über* $X_n := \{x_0, x_1, \ldots, x_{n-1}\}$ *ist ein gerichteter, zusam-*

menhängender, azyklischer Graph $G = (V, E, W)$[1] *mit Knotenmenge* V, *Kantenmenge* $E \subseteq V \times V$ *und Wurzelmenge* $W \subseteq \{\emptyset\} \times V$, *der folgende Bedingungen einhält:*

- *Jeder Knoten* $v \in V$ *ist entweder ein terminaler Knoten oder ein nicht terminaler Knoten.*
- *Jeder terminale Knoten ist entweder mit 0 oder mit 1 markiert und besitzt keine ausgehenden Kanten. Einen mit 0 markierten terminalen Knoten bezeichnen wir als terminalen Null Knoten, kurz terminale Null,* t_0. *Einen mit 1 markierten terminalen Knoten bezeichnen wir als terminalen Eins Knoten, kurz terminale Eins,* t_1.
- *Jeder nicht terminale Knoten* $v \in V$ *ist mit einer Booleschen Variablen* $x_i \in X_n$ *markiert und hat genau zwei ausgehende Kanten, deren Endpunkte mit* $low(v), high(v) \in V$ *bezeichnet werden.*
- *Jeder Knoten ist Endpunkt von mindestens einer Wurzel oder von mindestens einer Kante.*

Die Wurzeln stellen in Definition 2.1 Kanten ohne Startpunkt dar. Die Menge der terminalen Knoten bezeichnen wir als T, die Menge der nicht terminalen Knoten als N. Es gilt $V = N \cup T$. $low(v)$ und $high(v)$ bezeichnen wir als **Söhne** des Knotens v. Insbesondere sei $low(v)$ der *low*–Sohn und $high(v)$ der *high*–Sohn von v. Die Markierung eines Knotens v bezeichnen wir als **Label** $lab(v)$ des Knotens. Die Markierung x_i eines nicht terminalen Knotens wird als seine **Entscheidungsvariable** bezeichnet.

Definition 2.2 *Sei* $G := (V, E, W)$ *ein DD. Dann ist ein* ***Pfad*** p *in* G *eine geordnete Menge von Kanten* $p = \{e_1, \ldots, e_k\}$, $e_i \in E$, *für die für alle* $i \in \{1, \ldots, k-1\}$ *gilt: Der Endpunkt der Kante* e_i *ist gleich dem Startpunkt der Kante* e_{i+1}. *Der Startpunkt der Kante* e_1 *heist Startpunkt des Pfades* p, *der Endpunkt der Kante* e_k *heist Endpunkt des Pfades* p.

[1] In der Literatur wird ein DD häufig als gerichteter, zusammenhängender, azyklischer Graph $G = (V, E)$ definiert. Die Wurzeln sind dann ausgesuchte Knoten $w \in V$. Insbesondere sind dann alle Knoten, die nicht Endpunkt einer Kante aus E sind, Wurzeln. Für uns ist die Definition der Wurzeln als Kante ohne Startpunkt aber wichtig.

Um DD's als Datenstruktur zur automatischen Logiksynthese benutzen zu können sind weitere Definitionen und Einschränkungen nötig.

Definition 2.3 *Sei G ein DD über $X_n := \{x_0, \ldots, x_{n-1}\}$. Dann gilt:*

- *Die* **Größe** *von G, bezeichnet mit $|G|$, wird bestimmt durch die Anzahl der Knoten in V.*
- *G ist* **frei**, *genau dann wenn jede Variable auf jedem Pfad von der Wurzel zu einem terminalen Knoten maximal einmal vorkommt.*
- *G ist* **komplett**, *genau dann wenn jede Variable auf jedem Pfad von der Wurzel zu einem Terminalknoten genau einmal vorkommt.*
- *G ist* **geordnet** (**ordered**), *genau dann wenn es frei ist und auf jedem Pfad von einer Wurzel zu einem terminalen Knoten die Variablenreihenfolge gleich ist.*

Definition 2.4 *Sei G ein geordnetes DD über $X_n := \{x_0, \ldots, x_{n-1}\}$. Dann gilt:*

- *Die* **Variablenordnung** *in G ist definiert durch eine Permutation $\pi : \{1, \ldots, n\} \longmapsto \{1, \ldots, n\}$.*
- *Alle Knoten mit der Entscheidungsvariablen x_i bilden zusammen den* **Level** *$lev(x_i) = \pi(i)$. Die Terminalknoten bilden den Terminallevel $T := n$. Oberhalb des obersten Levels mit Knoten betrachten wir einen Level -1 ohne Knoten. Dieser Level stellt den Startlevel der Wurzeln dar.*

Wir benutzen den Buchstaben O um DD's als geordnet zu kennzeichnen. Der Variablenordnung können wir entnehmen, in welcher Reihenfolge die Entscheidungsvariablen der Knoten entlang eines Pfades in G auftreten.

Definition 2.5 *Sei $G = (V, E, W)$ ein geordnetes OKFDD. Eine Kante $e \in E$ heißt* **levelübergreifend** *genau dann, wenn ihr Startpunkt v_s und ihr Endpunkt v_e nicht in aufeinanderfolgenden Leveln liegen ($lev(v_s)+1 \neq lev(v_e)$). Eine Wurzel $w \in W$ heißt* **levelübergreifend** *genau dann, wenn ihr Endpunkt v_w nicht im obersten Level liegt ($lev(v_w) \neq 1$).*

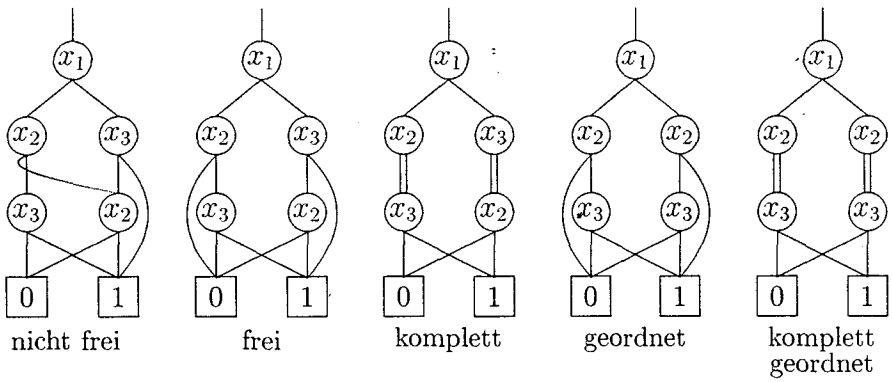


Abbildung 2.1: Entscheidungsdiagramme unterschiedlicher Eigenschaften

Um Boolesche Funktionen durch Entscheidungsdiagramme darzustellen, werden sie zunächst nach einer Variablen in zwei Teilfunktionen zerlegt. Diese Teilfunktionen werden dann rekursiv weiter zerlegt.

Definition 2.6 *Sei $f(X_n) : \{0,1\}^n \rightarrow \{0,1\}$ eine Boolesche Funktion über der Variablenmenge X_n.*

- *Dann bezeichnet f_i^0 den* ***Cofaktor*** *der Funktion f mit $\boldsymbol{x}_i = \mathbf{0}$, definiert durch $f_i^0(X_n) := f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$.*
- *Entsprechend bezeichnet f_i^1 den* ***Cofaktor*** *von f für $\boldsymbol{x}_i = \mathbf{1}$, definiert durch $f_i^1(X_n) := f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$.*
- *Schließlich ist f_i^2 definiert als $f_i^2(X_n) := f_i^0 \oplus f_i^1$, wobei $\oplus$ für die exklusiv-Oder Operation steht.*

Mit Hilfe dieser Faktoren definieren wir nun die drei Zerlegungstypen Shannon, positiv und negativ Davio. In [Drechsler95] wird gezeigt, daß diese drei Zerlegungstypen ausreichen, um eine allgemeinste Form der Entscheidungsdiagramme zu erreichen. Das heißt, durch Hinzunahme weiterer Zerlegungstypen können keine mächtigeren DDs erzeugt werden.

Definition 2.7 ***Zerlegungstypen*** *(Decomposition Types, DT):*

$$f = \overline{x_i} f_i^0 + x_i f_i^1 \quad \textit{Shannon} \quad (Shan) \tag{2.1}$$

$$f = f_i^0 \oplus x_i f_i^2 \quad \textit{positiv Davio} \quad (pDav) \tag{2.2}$$

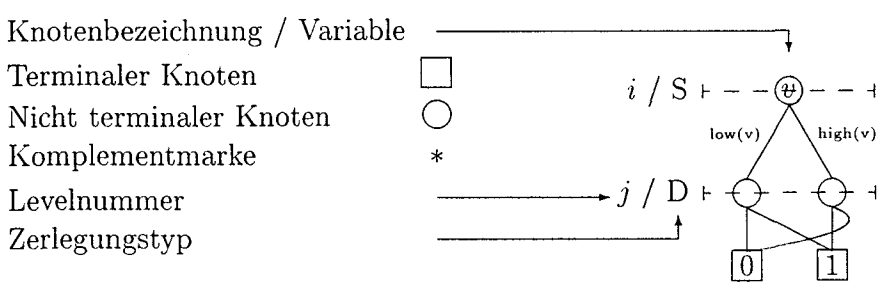
$$f = f_i^1 \oplus \overline{x_i} f_i^2 \quad \textit{negativ Davio} \quad (nDav) \tag{2.3}$$

Sei G ein DD über $X_n := \{x_1, \ldots, x_n\}$. Die den Variablen aus X_n zugeordneten Zerlegungstypen werden in einer Zerlegungstypenliste *(Decomposition Typ List, DTL)* $d = (d_1, \ldots, d_n)$ verwaltet. Der Zerlegungstyp der Variablen x_i ist d_i.

Bemerkung 2.1 *Für die Zerlegung nach Shannon kann anstelle der Oder Operation auch die exklusiv-Oder Operation verwendet werden, da sich die beiden Summanden gegenseitig ausschließen.*

$$f = \overline{x_i} f_i^0 \oplus x_i f_i^1$$

Zur Verdeutlichung verschiedener Eigenschaften zeigen wir einige Abbildungen von DDs. In diesen führt für jeden Knoten $v \in V$ die rechte ausgehende Kante zu $high(v)$ und die linke zu $low(v)$. Die im folgenden Bild erwähnte Komplementmarke führen wir auf Seite 10 ein.



Mit Hilfe der eingeführten Zerlegungstypen können wir nun verschiedene Entscheidungsdiagramm Typen definieren.

Definition 2.8 *Sei $X_n := \{x_1, \ldots, x_n\}$ eine Menge Boolescher Variablen. Dann seien BDDs, FDDs, pFDDs, nFDDs und KFDDs wie folgt definiert:*

- *Ein* ***Binary DD*** *(BDD) über X_n ist ein Entscheidungsdiagramm, das ausschließlich den Zerlegungstypen Shan enthält.*

- *Ein* ***PositiveFunctional DD*** *(pFDD) über X_n ist ein Entscheidungsdiagramm, das ausschließlich den Zerlegungstypen pDav enthält.*

- *Ein* ***NegativeFunctional DD*** *(nFDD) über X_n ist ein Entscheidungsdiagramm, das ausschließlich den Zerlegungstypen nDav enthält.*

- *Ein* ***Functional DD*** *(FDD) über X_n ist ein Entscheidungsdiagramm mit fester DTL d, in dem jeder Variablen $x_i \in X_n$ ein eindeutig bestimmter Zerlegungstyp $d_i \in \{pDav, nDav\}$, $\forall d_i \in d$ zugeordnet ist.*

- *Ein* ***Kronecker Functional DD*** *(KFDD) über X_n ist ein Entscheidungsdiagramm mit fester DTL d, in dem jeder Variablen $x_i \in X_n$ ein eindeutig bestimmter Zerlegungstyp $d_i \in \{Shan, pDav, nDav\}$, $\forall d_i \in d$ zugeordnet ist.*

Die Menge der KFDDs enthält somit die Menge der BDDs, pFDDs, nFDDs und FDDs. Ist das zugrundeliegende DD geordnet, so ist auch das BDD, das pFDD, das nFDD, das FDD beziehungsweise das KFDD geordnet.

Einem OKFDD G mit DTL $d = (d_1, \ldots, d_n)$ ist die Boolesche Funktion $f_G^d : B^n \to B$ folgendermaßen induktiv zugeordnet:
Für terminale Knoten $v \in T$ gilt:

$$f(v) = 0 =: f_{const_0} \qquad v = t_0 \tag{2.4}$$

$$f(v) = 1 =: f_{const_1} \qquad v = t_1 \tag{2.5}$$

und für nicht-terminale Knoten $v \in N$ mit $lab(v) = x_i$ und $DT(x_i) = d_i$ gilt:

$$f(v) = \overline{x_i} f(low(v)) + x_i f(high(v)) \qquad d_i = Shan \tag{2.6}$$

$$f(v) = f(low(v)) \oplus x_i f(high(v)) \qquad d_i = pDav \tag{2.7}$$

$$f(v) = f(low(v)) \oplus \overline{x_i} f(high(v)) \qquad d_i = nDav \tag{2.8}$$

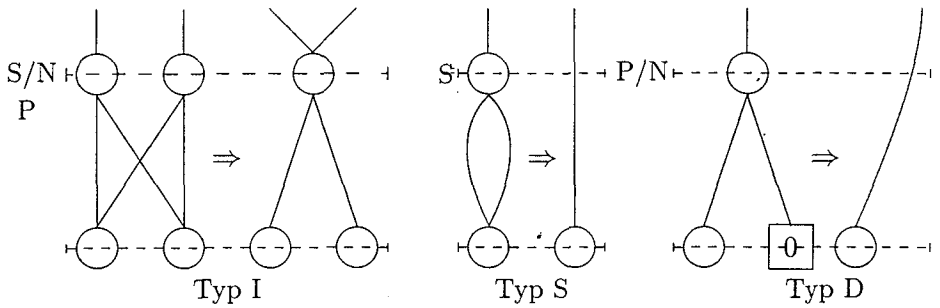


Abbildung 2.2: Reduktionen auf einem OKFDD

Sei der Knoten $w_i \in V$ der Endpunkt der Wurzel $i \in W$ in OKFDD $G = (V, E, W)$ mit DTL d. Dann ist die Funktion $f^d_{iG} = f(w_i)$ die Funktion, die diese Wurzel repräsentiert und $F^d_G = \{f(w_1), \ldots, f(w_{|W|})\}$ das Funktionenbündel über alle Wurzeln. Die Funktionen eines OBDDs G' bezeichnen wir durch $f^{BDD}_{iG'}$.

Notation: Zur Vereinfachung bezeichnet der Term f^d_G (f^{BDD}_G) im folgenden die Funktion f^d_{iG} (f^{BDD}_{iG}) der Wurzel i des OKFDDs (OBDDs) G für alle Wurzeln $i \in \{1, \ldots, |W|\}$.

Auf einem DD können verschiedene Reduktionen definiert werden:

Typ I: Entferne einen Knoten v', dessen Söhne identisch sind mit den Söhnen eines Knotens v und hänge alle Kanten, die auf v' zeigen, auf v um.

Typ S: Entferne einen Knoten v, dessen auslaufende Kanten zum gleichen Knoten v' führen und verbinde seine einlaufenden Kanten mit dem entsprechenden Sohn v'.

Typ D: Entferne alle Knoten v deren *high*–Sohn zur terminalen 0 führen und verbinde seine einlaufenden Kanten mit dem *low*–Sohn.

Die durch die Wurzeln eines OKFDDs dargestellten Funktionen ändern sich nicht, wenn Reduktionen von Typ S nur für Shannon Knoten und Reduktionen vom Typ D nur für Davio Knoten durchgeführt werden

und wenn Reduktionen vom Typ I für beliebige nicht terminale Knoten durchgeführt werden. Diese Behauptung wird in [Hett95] bewiesen.

Die Abhängigkeit von einer Wurzel zum terminalen Knoten t_0 (siehe Definition 2.11 auf Seite 11) ändert sich dagegen bei Reduktionen vom Typ D. Dies diskutieren wir genauer in Abschnitt 3.3, Seite 27.

Definition 2.9 *Ein OKFDD ist* ***reduziert****, genau dann wenn keine Reduktionen mehr auf dieses OKFDD angewendet werden können.*

Definition 2.10 *Zwei OKFDDs G_1 und G_2 mit der gleichen DTL d sind* ***equivalent****, genau dann wenn G_1 in G_2 durch mehrfache Anwendung von Reduktionen und inversen Reduktionen überführt werden kann.*

Ein OKFDD G_2 wird als ***Reduktion*** *eines OKFDDs G_1 bezeichnet, genau dann wenn G_1 und G_2 equivalent sind und G_2 reduziert ist.*

In [Drechsler95] wird gezeigt, daß die Reduktion eines OKFDDs G in Zeit $O(|G|)$ berechnet werden kann. Ebenso läßt sich aus einem reduzierten OKFDD das dazu äquivalente komplette OKFDD in linearer Zeit berechnen. Auf Details geht [Drechsler95] näher ein. Wir betrachten inverse Reduktionen in Abschnitt 4.5, Seite 53. Basis dieser Arbeit stellen reduzierte OKFDDs dar.

Abbildung 2.2 verdeutlicht die Reduktionstypen. In Abbildung 2.4 zeigen wir ein Beispiel für ein reduziertes OKFDD und darauf ausgeführte mögliche inverse Reduktionen.

Bemerkung 2.2 *In [Drechsler95] wurde gezeigt, daß ein freies reduziertes DD G kanonisch bezüglich einer DTL d und einer Variablenordnung π ist.*

Vorteile im Zusammenhang mit OKFDDs ergeben sich dadurch, daß sie auch kompakte Darstellungen von Funktionen ermöglichen, für die eine Darstellung als OBDD bzw. als OFDD exponentiell groß wäre.

Weiterhin können OKFDDs in ihrer Größe noch reduziert werden, indem man komplementierte Kanten benutzt. Bedingung für die Kanonizität von OBDDs mit komplementierten Kanten finden sich in [Brace90].

Analog können komplementierte Kanten auch im Falle von Davio Knoten benutzt werden [Becker93]. Sei G ein OKFDD und v und w Knoten in G. Repräsentieren v und w zueinander inverse Funktionen, d.h. $f(v) = \overline{f}(w)$, so können alle Kanten, die in w enden, auf v umgehängt werden, falls dann zu jeder dieser Kanten vermerkt wird, daß sie an ihren Startpunkt nicht die durch den Knoten v repräsentierte Funktion $f(v)$, sondern die dazu inverse Funktion $\overline{f}(v)$ weiterreicht.

2.3 Abhängigkeiten und quasireduzierte OKFDDs

Für unsere Methode der Logiksynthese sind noch zusätzliche Definitionen und Eigenschaften einzuführen. Zunächst definieren wir die Booleschen Bedingungen für die Abhängigkeit eines Knotens u von einem Knoten v in einem nachfolgenden Level in dem OKFDD. Im Anschluß daran erweitern wir die Abhängigkeit auf ganze Pfade.

Definition 2.11 *Die **Abhängigkeit** eines Knotens u von einem Knoten v, formal als $f_{u \to v} : B^n \to B$ notiert, sei definiert durch:*

- *$u = v$:*
  $f_{u \to u} = 1$
  Ein Knoten ist per Definition von sich selbst immer abhängig.

- *v ist Sohn von u:*
  $f_{u \to v}$ ergibt sich aus Tabelle 2.1.
  Die rechte Spalte der Tabelle folgt aus Anwendung des folgenden Punktes auf die beiden linken Spalten. Der Grund: ist v sowohl high- als auch low- Sohn von u, so entspricht dies zwei einzelnen Pfaden.

- *$\exists$ mindestens einen Pfad von u nach v:*
  $f_{u \to v} = \bigoplus_{\text{alle Pfade von u nach v}} \prod_{\text{alle Teilpfade } w_i, w_j} f_{w_i \to w_j}$
  Die Pfade werden durch And realisiert und durch Xor miteinander verknüpft.

$f_{u\to v}$	$v = low(u)$	$v = high(u)$	$v = low(v) = high(v)$
$d_i = Shan$	$\overline{x_i}$	x_i	1
$d_i = pDav$	1	x_i	$\overline{x_i}$
$d_i = nDav$	1	$\overline{x_i}$	x_i

Tabelle 2.1: direkte Abhängigkeiten, $lab(u) = x_i$

- *sonst:*
  $f_{u\to v} = 0$
  Gibt es keinen Pfad von u nach v, so ist u von v unabhängig.

Satz 2.1 *Die Abhängigkeit von u nach v ist gleich der Booleschen Funktion, die u repräsentiert, wenn v als terminale Eins und die übrigen terminalen Knoten als terminale Null interpretiert werden.*

Beweis: Durch Induktion über die Anzahl der Variablen des OKFDDs.

$\underline{n = 0, n = 1}$

Ergibt sich direkt aus den Punkten 1 und 2 in Definition 2.11.

$\underline{n \to n+1}$

Die Funktionen für die Abhängigkeit $f_{low(u)\to v}$ und $f_{high(u)\to v}$ sind nach Voraussetzung gleich der Funktionalität der Knoten $low(u)$ und $high(u)$, wenn v als terminaler Eins Knoten und alle anderen terminalen Knoten als terminale Null Knoten betrachtet werden. Die entsprechende Funktion für den Knoten u, zerlegt nach der Variablen x_i, ist dann:

$$f(u) = \begin{cases} \overline{x_i} \cdot f_{low(u)\to v} \oplus x_i \cdot f_{high(u)\to v} & \text{für } d_{lab(u)} = Shan \\ f_{low(u)\to v} \oplus x_i \cdot f_{high(u)\to v} & \text{für } d_{lab(u)} = pDav \\ f_{low(u)\to v} \oplus \overline{x_i} \cdot f_{high(u)\to v} & \text{für } d_{lab(u)} = nDav \end{cases}$$

□

Korrollar 2.1 *Aus Satz 2.1 folgt insbesondere für Knoten v in OKFDD G mit terminalem Eins Knoten t_1:*

$$f_{v \to t_1} = f(v)$$

Das heißt , die Abhängigkeit eines Knotens zur terminalen Eins ist gleich der Booleschen Funktion, die dieser Knoten repräsentiert.

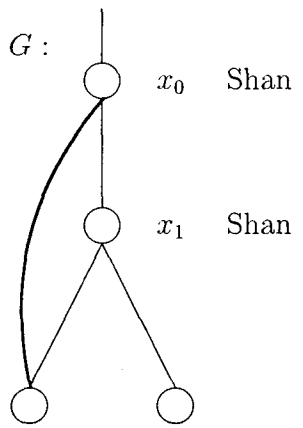
Die Bedeutung von $f_{v \to t_0} = f(v)$ für G mit terminalem Null Knoten t_0 klären wir in Abschnitt 3.2.

Die Definition der Abhängigkeiten ist zu komplementierten Kanten nicht verträglich. Die Abhängigkeitsbeziehung zwischen zwei Knoten ist lokal auf die Level dieser beiden Knoten beschränkt. Eine Komplementmarke einer Kante invertiert aber die gesammte Funktion des Sohnes und beeinflußt somit die Abhängigkeiten aller nachfolgenden Level. Die Lokalität der Abhängigkeiten ist aber Vorraussetzung für unser Verfahren. Nur so können wir die Abhängigkeitsmatrizen verschiedener Level parallel als Schaltkreis erzeugen (siehe Kapitel 4).

Definition 2.12 *Sei G ein OKFDD, i und j zwei Level in G. Sei m_i und m_j die Anzahl der Knoten in diesen Leveln. Die Knoten in Level $n \in \{i, j\}$ seien $\{v_n^0, \ldots, v_n^{m_n-1}\}$. Dann ist die* ***Abhängigkeitsmatrix*** *R_{ij} eine $m_i \times m_j$ Matrix mit den Elementen $r_{k,l}^{ij} = f_{v_i^k \to v_j^l}$ mit $k \in \{0, \ldots, m_i - 1\}$, $l \in \{0, \ldots, m_j - 1\}$, d.h. die Matrixelemente stellen die Abhängigkeiten von den Knoten in Level i zu den Knoten in Level j dar.*

Zwei Abhängigkeitsmatrizen lassen sich durch eine Boolesche Matrix Multiplikation $BMM : B^{i \cdot j} \times B^{j \cdot k} \to B^{i \cdot k}$ zu einer neuen Matrix verknüpfen. Dies definieren wir im Anschluß. Wir unterscheiden zwischen einer Exor – basierten BMM und einer Or – basierten BMM. Erstere verknüpft die einzelnen Produktterme mit der exklusiven Oder Operation, letztere mit der einfachen Oder Operation.

Da in einem nicht kompletten OKFDD G manche Kanten einen oder mehrere Level überspringen, führen wir eine *erweiterte Boolesche Matrix Multiplikation* $eBMM : B^{i \cdot j} \times B^{j \cdot k} \to B^{i \cdot k}$ ein. Diese bezieht die



$$R_{0\,1} \times R_{1\,2} = R_{0\,2}$$

$$(\boxed{x_0}) \times (\boxed{\overline{x}_1},\ x_1) = (\boxed{\overline{x}_0 \oplus x_0\overline{x}_1},\ x_0x_1)$$

Abbildung 2.3: Erweiterte BMM für einen Graphen G. Die Abhängigkeit der hervorgehobenen Kante fließt nicht in die Matrizen $R_{0\,1}$ und $R_{1\,2}$ ein. Sie wird erst an der durch den Start – und Zielknoten ihrer Kante bestimmten Stelle (0,0) in der Matrix $R_{0\,2}$ durch die Oder bzw. exklusiv Oder Operation mit den entsprechenden Produktermen verknüpft.

Abhängigkeiten von überspringenden Kanten in die Produktmatrix mit ein (Abbildung 2.3). Auch die erweiterte BMM ist je nach Anwendung Exor – oder Or – basiert. Die Abhängigkeitsbedingungen der überspringenden Kanten werden mit den übrigen Produkttermen ebenfalls durch die Oder bzw. exklusive Oder Operation verknüpft.

Definition 2.13 *Sei G ein OKFDD mit Level $i < j < k$. Sei m_i, m_j, m_k die Anzahl der Knoten in diesen Leveln und $R_{i\,j} \in B^{m_i \cdot m_j}$ mit Elementen $r^{ij}_{z,s}$, $0 \leq z < m_i$, $0 \leq s < m_j$ und $R_{j\,k} \in B^{m_j \cdot m_k}$ mit Elementen $r^{jk}_{z,s}$, $0 \leq z < m_j$, $0 \leq s < m_k$ Abhängigkeitsmatrizen. Dann sei die* ***Boolesche Matrix Multiplikation*** *definiert durch*

$$BMM: \qquad r^{ik}_{z,s} := \sum_{e=0}^{m_j} r^{ij}_{z,e} \cdot r^{jk}_{e,s} \qquad 0 \leq z < m_i,\ \ 0 \leq s < m_k$$

und die erweiterte BMM sei definiert durch

$$eBMM: \qquad r_{z,s}^{ik} := u_{z,s}^{ik} + \sum_{e=0}^{m_j} r_{z,e}^{ij} \cdot r_{e,s}^{jk} \qquad 0 \leq z < m_i, \quad 0 \leq s < m_k$$

wobei die Summierung entweder exklusiv oder inklusiv ist, je nachdem ob die BMM Exor - oder Or - basiert ist und $u_{z,s}^{ik}$ stelle die Abhängigkeit über eine Kante vom z-ten Knoten in Level i zum s-ten Knoten in Level k dar, wenn es eine solche gibt, andernfalls sei es 0.

In welchen Fällen die aus einer BMM beziehungsweise aus einer eBMM der Abhängigkeitsmatrizen R und R' resultierende Matrix eine Abhängigkeitsmatrix zu dem R und R' zugrundeliegenden OKFDD ist, diskutieren wir in Abschnitt 4.1 ab Seite 31.

Notation: Als ***BMM – Reihenfolge*** $\Re$ bezeichnen wir die Klammmerung, durch die die Reihenfolge bestimmt wird, in der die Abhängigkeitsmatrizen eines OKFDDs miteinander multipliziert werden bzw. die Menge der Abhängigkeitsmatrizen im Ablauf der Synthese, die sich aus obiger Klammerung ergibt, selbst.

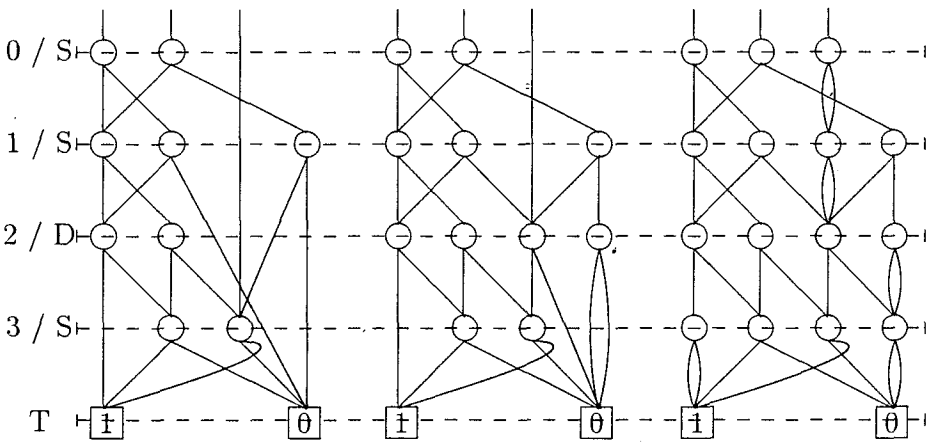
Definition 2.14 *Sei G ein OKFDD mit den Leveln $i \in \{0, \ldots, n = T\}$.*

- *Eine Matrix, die genau die Abhängigkeiten der Knoten eines Levels i zu den Knoten des Levels $i+1$ enthält, heißt* ***primäre Matrix****.*
- *Alle anderen Abhängigkeitsmatrizen zu G heißen* ***sekundäre Matrizen****.*
- *Eine Matrix, die genau die Abhängigkeiten der Knoten eines Levels i zu den Knoten des Levels T enthält, heißt* ***terminale Matrix****.*

Durch $\Psi = \{R_{0\,1}, R_{1\,2}, \ldots, R_{n-1\,T}\}$. bezeichnen wir die Menge der primären Matrizen.

Definition 2.15 *Ein OKFDD heißt* ***quasireduziert****, wenn es komplett ist und keine Reduktionen vom Typ I mehr durchgeführt werden können.*

Ein OKFDD heißt ***partiell quasireduziert****, wenn es frei ist und keine Reduktionen vom Typ I mehr durchgeführt werden können.*



(a) reduziert (b) bzgl. $\Re$ partiell quasireduziert (c) quasireduziert

$$\Re = \{R_{0\,1}, R_{1\,2}, R_{2\,3}, R_{3\,T}, R_{0\,2}, R_{2\,T}, R_{0\,T}\}$$

Abbildung 2.4: Reduziertes, pratiell quasireduziertes und quasireduziertes OKFDD der gleichen Funktionalität

Ein OKFDD heißt ***partiell quasireduziert bezüglich einer BMM-Reihenfolge*** $\Re$*, wenn es partiell quasireduziert ist und folgende Bedingungen gelten:*

- *Es gibt keine Kante e_{ij} von Level i nach Level j, zu der es in $\Re$ keine Abhängigkeitsmatrix $R_{i\,j}$ gibt.*
- *Es gibt keine Wurzel w zu einem Knoten in Level i, zu der es in $\Re$ keine Abhängigkeitsmatrix $R_{i\,T}$ gibt.*
- *Es können keine Reduktionen mehr durchgeführt werden, ohne daß eine der beiden ersten Bedingungen verletzt wird.*

Für quasireduzierte OKFDDs lassen sich folgende Eigenschaften nachweisen:

Satz 2.2 (a) *Quasireduzierte OKFDDs sind kanonische Repräsentationen für Boolesche Funktionen, falls die Dekompositionstypliste und die Variablenordnung fest gewählt sind.*