

Bernd Becker, Christoph Meinel (Hrsg.)

Entwerfen, Prüfen, Testen

**Dagstuhl-Seminar-Report; 6
18.2.1991 - 22.2.1991 (9108)**

ISSN 0940-1121

Copyright © 1991 by IBFI GmbH, Schloß Dagstuhl, W-6648 Wadern, Germany
Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Verantwortlich für das Programm:

Prof. Dr.-Ing. José Encarnação,
Prof. Dr. Winfried Görke,
Prof. Dr. Theo Härder,
Dr. Michael Laska,
Prof. Dr. Thomas Lengauer,
Prof. Ph. D. Walter Tichy,
Prof. Dr. Reinhard Wilhelm (wissenschaftlicher Direktor).

Gesellschafter: Universität des Saarlandes,
Universität Kaiserslautern,
Universität Karlsruhe,
Gesellschaft für Informatik e.V., Bonn

Träger: Die Bundesländer Saarland und Rheinland Pfalz.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: dagstuhl@dag.uni-sb.de

Seminarbericht
Entwerfen, Prüfen, Testen
18. bis 22. Februar 1991

In der Woche vom 18.02. - 22.02.1991 trafen sich im IFBI Dagstuhl 31 Wissenschaftler zur Teilnahme am Seminar „**Entwerfen, Prüfen, Testen**“. Das Seminar fand unter Leitung von *B. Becker (Frankfurt/M)* und *Ch. Meinel (Berlin)* statt.

Im Mittelpunkt des Interesses standen zunächst einmal Fragen und Methoden, die auf Probleme im Bereich des **Prüfens** und **Testens**, sei es aus theoretischer oder aus praktischer Sicht, eingingen.

Resultate, die sich mit Fragen aus folgenden Themenkreisen befaßten, wurden vorgetragen und von den Teilnehmern eingehend diskutiert:

- Testen und Struktur
- Design for Testability
- Testerzeugung
- Fehlersimulation
- Probabilistische Methoden im Testen
- Fehlertoleranz

Die übrigen Vorträge verteilten sich auf die Gebiete der **Synthese**, insbesondere Logiksynthese, Layoutsynthese, Graphenbettungen, der **Verifikation** und der **Datenhaltung** im VLSI-Bereich.

Inhaltsverzeichnis

Pseudoduplication der Floating Point Addition – Eine softwareimplementierte Methode zur Erkennung von Permanenten Hardware Fehlern; Wolfgang Hahn, Michael Gössel	1
NBS - Netzbeschreibungssprache für den Schaltkreisentwurf; Reiner Witschurke	2
Zur Maskierung von Fehlfolgen mit ungeradem Gewicht durch serielle Signaturregister; Lutz Voelkel	3
Strukturorientierter Test von VLSI-Algorithmen; Hartmut Schmeck	4
Ein zellenbasierter Ansatz zur Dimensionierung kombinatorischer Schaltkreise; Reiner Kolla	5
Strukturbasierte Testmethoden für arithmetische Schaltkreise; Uwe Sparmann	6
Testmusterbeschaffung aus algorithmischen Beschreibungen; Reinhard Reisig	6
Testerzeugung - noch immer ein Problem?; Winfried Görke .	7
Entwurf & Testbarkeit eines optimalen systolischen Addierers; Lars Kühnel	8
On $l \times n$ Boolean Matrix with all $l \times k$ Submatrices Having 2^k Distinct Row Vectors; Hongzhong Wu	10
Branching Programme als Datenstrukturen für den Schaltkreisentwurf; Christoph Meinel	10
Einfache Modifikationen von Schieberegistern zur Verkürzung von Testfolgen; Steffen Tarnick	11

Die Bedeutung von Dominatoren für die Fehlersimulation in kombinatorischen Schaltkreisen; Rolf Krieger	12
Die Beschreibung von Rechteckzerlegungen und ihre Optimierung; Ernst-Günter Giessmann	13
Zur Anwendung eines Multi-Level- Fehlersimulators; B. Straube, G. Elst, M. Galetzka, W. Vermeiren	14
Verfahren zur Erhöhung der Zuverlässigkeit von Schaltkreisen; Rüdiger Reischuk	15
Einbettung baumartiger und würfelartiger Graphstrukturen in 2-dimensionale Gitter; Peter Zienicke	16
Verifikationsfreundlicher Entwurf (Design for Verifiability, DfV); Hans Evekling	17
Über einen zellenorientierten Fehlersimulator für kombinatorische Schaltkreise; Björn Schieffer	18
The random testability of the n-input AND gate; Joachim Hartmann	19
Ausnutzung funktionaler Eigenschaften zur Erhöhung der Verfügbarkeit von Computern; Günter Hotz	20
Probleme zur Lösung der globalen Phase des Layoutentwurfs bei Gate-Arrays und bei der Parallelisierung dieses Prozesses; Wolfgang Schade, B.Goetze, J.Liske, W.Nehrlich, B.Stube	22
Über ein Verfahren zur effektiven mehrstufigen Realisierung Boolescher Funktionen; Michael Weber	23
Über die Komplexität zuverlässiger kombinatorischer Netzwerke; Dietmar Uhlig	24

Pseudoduplication der Floating Point Addition – Eine softwareimplementierte Methode zur Erkennung von Permanenten Hardware Fehlern

Wolfgang Hahn, Michael Gössel
Zentralinstitut für Kybernetik und Informationsprozesse
ehemalige Akademie der Wissenschaften der DDR
Kurstraße 33, O-1086 Berlin

Am Beispiel der Gleitkommaaddition wurde gezeigt, wie ein und derselbe Prozessorbefehl diversitär, d.h. auf unterschiedlichen Datenpfaden ausgeführt werden kann. Dabei wird die Mantisse eines Operanden um eine Stelle nach rechts verschoben (RESO, recomputing with shifted operands). Durch Vergleich der unterschiedlich erzielten Ergebnisse können Fehler online erkannt werden, die zur Laufzeit entstehen. Zusätzlicher Hardwareaufwand ist dazu nicht notwendig. Fehlererkennungseigenschaften für Fehler im Ergebnis und die Beschränkung der Rundungsabweichungen bei der Wahl geeigneter Rundungsmodi wurden bewiesen.

Darüberhinaus wurden Wahrscheinlichkeiten angegeben, mit der Fehler erkannt werden, die sich nicht im Ergebnis auswirken, wobei funktionelle Fehlermodelle zur Anwendung kamen. Der Zeitaufwand für die Implementierung der sich selbst testenden Operation liegt prozessorabhängig zwischen fünf- und zehnfacher Ausführungszeit einer Instruktion. Daher wurde vorgeschlagen, die selbsttestenden Operationen nur an bestimmten kritischen Stellen in einem Programm einzubauen. Hierzu sind noch geeignete Strategien für den Compiler zu entwickeln.

NBS - Netzbeschreibungssprache für den Schaltkreisentwurf

Reiner Witschurke
Karl-Weierstraß-Institut für Mathematik, Berlin
Mohrenstraße 39, O-1086 Berlin

Die Hardwarebeschreibungssprache NBS wird zur textlichen Beschreibung von hierarchischen Schaltungen verwendet. Das zugehörige NBS-Programmsystem erfaßt und prüft diese Texte. Außerdem werden Netzlisten für spezielle Datenbasen erzeugt.

Eine hierarchische Beschreibung einer Gesamtschaltung (Netzwerk) setzt sich aus Einzelschaltungen (Netze) zusammen. Diese Netze können entweder strukturell oder funktionell beschrieben sein. Auch Grundsaltungen (Elementare) beliebig komplexer Strukturen können als Netze auftreten. Außerdem sind in einem Netzwerk globale Verbindungsstrukturen zugelassen, die in der Beschreibung der Netzinterfaces nicht extra aufgeführt werden müssen (z.B. ERDE). Der Linkalgorithmus (Verbinden) prüft beim Abbau der Hierarchie die Paßfähigkeit der Interfaces der einzelnen Netze und ordnet die globalen Verbindungen im Netzwerk den jeweiligen Interfaces der Netze zu, für die sie relevant sind. Der Verbindungsprozeß kann auch über Teilnetzwerke ablaufen, so daß das Gesamtnetzwerk durch wiederholtes Verbinden erzeugt werden kann.

Ein Netz besteht aus Interface (Rand) und Körper (Rumpf). Im Rand werden neben dem Netznamen die Ports (Folgen von Verbindungen) und die Parameter angegeben. Den Ports können Attribute zugeordnet sein. Der Rumpf besteht aus Anweisungen. Es werden sprachliche und metasprachliche Anweisungen unterschieden. Sprachliche Anweisungen sind Aufrufe (Mehrpole) von Schaltungen, logische Beschreibungen (Verbindungsanweisungen), Deklarationen von Verbindungsfeldern und Verbindungsattributen, sowie funktionelle Beschreibungen von Schaltungsbewertungen. Zu den metasprachlichen Anweisungen, die spätestens beim Verbindungsprozeß abgearbeitet werden, gehören Indexanweisungen, Überlagerungen von Verbindungen, sowie Spezifikationen von Parametern.

Zur Maskierung von Fehlfolgen mit ungeradem Gewicht durch serielle Signaturregister

Lutz Voelkel

Hochschule für Seefahrt Warnemünde/Wustrow

Die Signaturanalyse ist ein Kompaktverfahren zur Diagnose digitaler Schaltungen, bei dem (i.a. lange) Signalfolgen durch linear rückgekoppelte Schieberegister komprimiert werden. Statt der Folgen, die beim Test einer möglicherweise fehlerhaften und der zugehörigen fehlerfreien Schaltung entstehen, werden nur noch die Kompressionsresultate - die sogenannten Signaturen - verglichen. Man spricht von Maskierung, wenn durch die Kompression einer fehlerhaften Folge gerade die zur fehlerfreien Folge gehörende Signatur entsteht. Das Maskierungsproblem spielt eine wesentliche Rolle in vielen Arbeiten zum Schaltungstest durch Signaturanalyse. Neben „quantitativen“ Untersuchungen, zu denen vor allem Abschätzungen bzw. Berechnungen von Maskierungswahrscheinlichkeiten (unter bestimmten Voraussetzungen über Fehlerverteilungen) zählen, sind auch „qualitative“ Betrachtungen von Interesse, bei denen es darum geht, unter welchen Bedingungen (an die Art der Rückkopplungen und an die betrachteten Folgen) eine Maskierung überhaupt möglich ist.

Schwerpunkt des Vortrages ist die Betrachtung von Fehlfolgen mit einer (kleinen) ungeraden Anzahl von Abweichungen.

Neben einigen Resultaten ergeben sich hier viele ungelöste Probleme. Simulationen sollen zu einer verbesserten Einsicht in diese Probleme führen.

Strukturorientierter Test von VLSI-Algorithmen

Hartmut Schmeck
Universität Kiel

Bei der Untersuchung der Testbarkeit von VLSI-Algorithmen bietet es sich an, Kenntnisse über die Struktur dieser durch hochintegrierte Schaltkreise realisierten Algorithmen zu verwenden. Ausgehend von der Beschreibung eines VLSI-Algorithmus durch Rechnergraph, Layout und I/O-Schema definiert man ein abstraktes Fehlermodell auf der Ebene der im Rechnergraphen vorkommenden logischen Knoten. Neben den auch im geläufigen Haftfehlermodell auf Gatterebene betrachteten Stuck-at-0 und Stuck-at-1 Fehlern an Ein- und Ausgängen berücksichtigt man dabei auch allgemeinere funktionelle Fehler. Darüber hinaus müssen wegen der sequentiellen Arbeitsweise der logischen Knoten auch Fehler in Verzögerungsgliedern betrachtet werden. Am Beispiel verschiedener Algorithmen fürs Sortieren, für die Addition und für die Multiplikation wird gezeigt, daß sich auf der Basis dieses Fehlermodells effiziente off-line Tests ergeben, d.h. kurze Testfolgen reichen aus, um den gemäß dem Layout als Chip realisierten Rechnergraphen periodisch auf permanente (Einfach-) Fehler zu testen. Auch bezüglich des Haftfehlermodells auf Gatterebene ergeben sich dabei für die untersuchten Algorithmen sehr hohe Fehlerüberdeckungsraten. Durch relativ geringe Erweiterungen des Rechnergraphen und Ausnutzung des I/O-Schemas ergeben sich ähnlich wie im RESO-Ansatz von Patel außerdem effizient Möglichkeiten für einen on-line Test, der während einer Berechnung auftretende Fehler sofort erkennt und ein entsprechendes Signal als Ausgabe erzeugt.

Ein zellenbasierter Ansatz zur Dimensionierung kombinatorischer Schaltkreise

Reiner Kolla
Universität des Saarlandes

Wir betrachten das Problem der Dimensionierung von Schaltkreisen, d.h. die physikalischen Parameter sind so abzustimmen, daß gewisse Kostenziele wie Fläche, Zeit minimiert werden. Im Gegensatz zu bisher vorgeschlagenen Verfahren auf Transistorebene, die die Kanalbreiten der Transistoren so einstellen, daß unter einer gewissen Schranke für die Fläche, die Laufzeit minimiert wird, schlagen wir ein abstraktes Verfahren auf Gatterebene vor. Man erhält dann ein Zuordnungsproblem folgender Art: Ordne jedem Gatter im Schaltkreis eine aus wenigen möglichen Realisierungen so zu, daß unter einer Flächenschranke A_{max} die Gesamtlaufzeit des Schaltkreises minimiert wird. Wir zeigen, daß dieses Problem durch dynamisches Programmieren effizient lösbar ist in Zeit $O(s^2 A_{max} n)$, wo s die Zahl der möglichen Realisierungen und n die Zahl der Gatter ist, falls die Schaltkreise einfache Bäume sind. Für allgemeine kombinatorische Schaltkreise ist dieses Problem NP-vollständig. Wir entwickeln daher Heuristiken, die durch iteratives Verbessern kritischer Pfade oder Bäume versuchen, die Gesamtlaufzeit zu minimieren und berichten über ermutigende experimentelle Erfahrungen an Beispielen. Zum Abschluß zeigen wir, wie man diese Strategien auch verknüpfen kann mit dem Problem der Abbildung auf komplexe Zellenbibliotheken, indem man aus dem Übersetzerbau bekannte Tree-matching Techniken einsetzt.

Strukturbasierte Testmethoden für arithmetische Schaltkreise

Uwe Sparmann
Universität des Saarlandes

Da das Problem der Testmustergenerierung NP-vollständig ist, liefern universelle, d.h. auf beliebige Schaltungen anwendbare Algorithmen für große Schaltkreise häufig nur unzureichende Resultate. Daher werden für spezielle wichtige Schaltkreisfamilien spezialisierte Testmethoden entwickelt. Auf dem Gebiet der Arithmetik gab es hierbei bisher nur Resultate für ganzzahlige Operationen. Im Vortrag wurde eine Erweiterung dieser Resultate auf Gleitkommaoperationen vorgestellt.

Betrachtet wurden parametrisierte Entwürfe für die Gleitkommaaddition bzw. -multiplikation nach dem IEEE-Standard für binäre Gleitkommaarithmetik. Die Testbarkeit dieser Entwürfe kann mit Hilfe sehr geringer Hardwareerweiterungen garantiert werden.

Das Testkonzept kombiniert eine strukturbasierte Vorgehensweise für die großen regulären Module des Mantissentteils mit der Anwendung universeller Methoden zur Testerzeugung für die irreguläre Logik des Exponententeils. Die mit dieser Vorgehensweise abgeleiteten Tests erreichen eine vollständige Fehlerüberdeckung bei einer nahezu minimalen Testgröße.

Testmusterbeschaffung aus algorithmischen Beschreibungen

Reinhard Reisig
Universität -GH- Paderborn

Der hierarchische Schaltungsentwurf beginnt mit einem algorithmischen Modell. Dieses abstrakte, logisch z.B. Pascal-Programmen ähnelnde Modell

wird durch die Simulation ausgewählter Tests der modellierten Schaltung validiert. Das algorithmische Modell wird zum Register-Transfer-Modell transformiert und dieses mit den gleichen Tests wie das algorithmische Modell validiert. Aus diesem Ablauf lassen sich Fertigungstestmuster für die Steuerwerke integrierter Schaltungen folgendermaßen ableiten: Bei der Validierung des algorithmischen Modells werden zum Fertigungstest geeignete Teile der Tests ausgewählt. An Hand einer Beschreibung bestimmter Details der Transformation zum Register-Transfer-Modell werden die entsprechenden Teile der Register-Transfer-Tests identifiziert. Für jeden Taktimpuls dieser Teile der Register-Transfer-Tests werden Ausgangs- und Folgezustand sowie die Werte auf den Ein- und Ausgängen der Schaltung aufgezeichnet. Ist die Schaltung nun einstell- und beobachtbar wie z.B. von einem Prüfbus gewährleistet, können die Aufzeichnungen für jeden Taktimpuls genau als ein Parameter benutzt werden. In 3 Anwendungen dieses Verfahrens wurden Fehlererkennungsgrade von jeweils etwa 84% erreicht, bezogen auf die Einzelhaftfehler auf der Gatterebene. Das Verfahren unterstützt die Entwurfsvalidierung, beschafft nur zulässige Testmuster und verursacht nur sehr geringen Aufwand.

Testerzeugung - noch immer ein Problem?

Winfried Görke
Institut für Rechnerentwurf und Fehlertoleranz
Universität Karlsruhe

Die Fehlerdiagnose digitaler Schaltungen beruht zu einem wesentlichen Anteil auf der Erzeugung deterministischer Testmuster, für die grundlegende Algorithmen und Verfahren bereits vor 25 Jahren vorgeschlagen wurden. Wegen der NP-Vollständigkeit des Problems ist allerdings die heutige Situation, in der die Prüfung von kundenorientierten CMOS-Schaltungen mit über 100000 Schaltelementen und über 300 Anschlüssen vorzubereiten ist, keineswegs befriedigend. Dabei wurden wesentliche Verbesserungen der Testerzeugung durch heuristische Ansätze, Simulation oder Entwurfsmaßnahmen zur Verbesserung der Testbarkeit erreicht, die auch auf Selbsttests mit

gewichteten Zufallsmustern zurückgreifen. Andererseits wurden als Alternativen zur Reduktion des Aufwandes für die Testerzeugung immer wieder die Verwendung leicht prüfbarer Strukturen vorgeschlagen. Auf Quersummenschaltnetze, Ringsummenimplementierungen, PLA-Strukturen und iterierte Zellanordnungen wird näher eingegangen und dabei gezeigt, daß eine allgemeine Anwendbarkeit nach wie vor zu wünschen übrig läßt, so daß die Praxis auf weitere Verbesserungen längst bekannter Verfahren angewiesen bleibt. Selbsttests und gewichtete Zufallstestmuster bilden die derzeit aussichtsreichsten Alternativen.

Entwurf & Testbarkeit eines optimalen systolischen Addierers

Lars Kühnel
Universität Kiel

Das Thema dieses Vortrages läßt sich in den Bereich des Entwurfes von gut testbaren systolischen Arrays einordnen. Im ersten Teil wurde ein neuartiger systolischer Hardware-Algorithmus FASTA für die ganzzahlige Addition vorgestellt. Das FASTA-Verfahren verwendet einen Mesh-Connected-Array-artigen Rechnergraphen und nur 4 verschiedene Typen von sehr einfachen und global getakteten Prozessorelementen. Eine wesentliche Eigenschaft ist die ausschließliche Verwendung von sehr kurzen Leitungen für die Übertragung von Daten zwischen Prozessoren: jeder Prozessor kommuniziert nur mit seinen im geometrischen Sinne unmittelbaren Nachbarn. Es handelt sich hierbei um eine Eigenschaft, die von Kung mit dem Begriff „purely systolic“ belegt worden ist. Wesentliche Ideen:

- Carry Lookahead Technik
- \sqrt{n} Gruppen zu je \sqrt{n} Bitpositionen
- Ripple-Carry-Addierer-artige Berechnung innerhalb der Blöcke

- Berechnung der \sqrt{n} Block-generate & -propagate-Signale im Pipelining
- Akkumulation der Blockberechnungen in $O(\sqrt{n})$ Zeit
- 2-stufige Generierung der Summenbits.

Das Verfahren hat einen Flächen- bzw. Zeitbedarf $A(n) = O(n)$ bzw. $T(n) = O(\sqrt{n})$. Die Periode erfüllt die Beziehung $P(n) = O(\sqrt{n})$. Das Verfahren ist somit optimal bzgl. T , APT und AT^2 . Außerdem ist es optimal bzgl. aller hier üblichen Komplexitätsmaße in der Klasse der zeitoptimalen Algorithmen. Im zweiten Teil wurde gezeigt, daß das FASTA-Verfahren C-testbar bzgl. eines erweiterten Einzel-Haftfehlermodells ist. Da die Realisierungen der Zellen D-Flipflops (FFs) enthalten, müssen die Annahmen des klassischen Haftfehlermodells um entsprechende Annahmen über das Fehlverhalten von FFs erweitert werden. In diesem Zusammenhang wurde ein neues Gatter-Level-Fehlermodell GALEXSA für sequentielle Schaltungen vorgestellt. Es berücksichtigt außer den üblichen Haftfehlern auch sog. „Übergangs-“ und „Verzögerungsfehler“ der vorhandenen FFs. Auf der Basis des Gatter-Level-Modells wurde dann ein entsprechendes Modell FAXSA für das Fehlverhalten auf der Ebene von Prozessorelementen (d.h. logischen Knoten) eingeführt. Eine entsprechend gestufte Vorgehensweise bei der Testgenerierung führt zu einer Testfolge der Länge 5, mit der alle möglichen GALEXSA-Fehler entdeckt werden. Da die Anzahl der für einen vollständigen Test benötigten Testmuster also nicht von der Problemgröße n abhängt, handelt es sich hier um einen C-testbaren Hardware-Algorithmus. Weitergehende Ausführungen findet man in:

L.Kühnel: Optimal Purely Systolic Addition. Bericht 9002. Institut für Informatik, Universität Kiel, April 1990.

On $l \times n$ Boolean Matrix with all $l \times k$ Submatrices Having 2^k Distinct Row Vectors

Hongzhong Wu
Lehrstuhl Prof. G. Hotz
Universität des Saarlandes

By using our method presented in this seminar, one can almost get an acceptable pseudoexhaustive test set for multiple primary output circuits with n primary inputs, and every primary output depends on at most k primary inputs. These methods can also be used in designing fault tolerant computing systems.

Branching Programme als Datenstrukturen für den Schaltkreisentwurf

Christoph Meinel
Humboldt-Universität Berlin

Branching Programme sind ein wichtiges Berechnungsmodell für Boolesche Funktionen. Ihre Größe spiegelt die Komplexität der Funktion wider und steht in logarithmischem Zusammenhang zum bei Berechnung auf Turing Maschinen erforderlichen Speicheraufwand. Der Vortrag diskutiert über Möglichkeiten, Branching Programme (BP) in Schaltkreisentwurfswerkzeugen zur Repräsentation Boolescher Funktionen zu verwenden. Angeregt durch die aktuelle Bedeutung, die die BDD (binary decision diagrams) für den Schaltkreisentwurf, Logikverifikation, ... bereits erlangt haben und die Beobachtung, daß BDD nichts anderes als oblivious read-once-only BP sind, erscheint der Versuch sinnvoll, aussagekräftigere BP-Modelle (read-once-only BP, oblivious BP lin. Länge, BP, ...) für derartige Aufgaben einzusetzen.

Aus komplexitätstheoretischer Sicht scheint ein solcher Versuch dringend geboten, da Untere-Schranken-Untersuchungen für konkrete Funktionen zeigen, daß BP, BP1 bzw. linear lange oblivious BP Funktionen berechnen können, die nur von BDD bzw. Entscheidungsbäumen exponentieller Größe berechnet werden können.

Einfache Modifikationen von Schieberegistern zur Verkürzung von Testfolgen

Steffen Tarnick

Zentralinstitut für Kybernetik und Informationsprozesse Berlin

Beim Selbsttest von Schaltungen werden zur Erzeugung deterministischer Testsätze oft autonome Automaten verwendet. Diese Automaten erzeugen Testfolgen, in die der gewünschte Testsatz eingebettet ist. Zusätzlich werden aber auch übergangsbedingte Füllvektoren erzeugt. Um die Testzeit möglichst kurz zu halten, wird angestrebt, die Anzahl dieser Füllvektoren zu minimieren. Dazu sind verschiedene Verfahren bekannt. Am Beispiel des nichtlinear rückgekoppelten Schieberegisters (FSR) wird gezeigt, wie man durch einfache Modifikationen an der Hardware des Testmuster-generators die Testfolgen weiter verkürzen kann. Solche Modifikationen beinhalten, daß von ausgesuchten Flipflops des FSR der negierte Ausgang in den Eingang des darauffolgenden Flipflops geleitet wird, bzw. daß die Ausgänge des FSR permutiert werden. Beide Modifikationen erfordern keinen zusätzlichen Hardwareaufwand und führen zu einem weiteren Freiheitsgrad zur Minimierung der Testlänge. Eine weitere Modifikation, die mit einem Mehraufwand an Hardware verbunden ist, besteht im Einfügen zusätzlicher Flipflops in das FSR. Diese Modifikation erlaubt es, eine Menge von Testvektoren in einer vorgegebenen Reihenfolge ohne Füllvektoren zu erzeugen. Es läßt sich zeigen, daß sich alle drei Varianten der Modifikation des FSR auf eine entsprechende Modifikation des Testsatzes zurückführen lassen und somit die Minimierung der Testfolgenlänge für modifizierte FSR's auf ein Standardverfahren zurückgeführt werden kann.

Die Bedeutung von Dominatoren für die Fehlersimulation in kombinatorischen Schaltkreisen

Rolf Krieger
Johann Wolfgang Goethe Universität
Frankfurt a.M.

Die Fehlersimulation für kombinatorische Schaltkreise kann man in Gutsimulation, Fehlersimulation innerhalb der fanoutfreien Zonen und Einzelfehlersimulation der Fanoutstämme gliedern. Im Vortrag wurden Methoden beschrieben, die unter Verwendung von statischen und „dynamischen“ Dominatoren der Fanoutstämme (FOS) die laufzeitbestimmende Einzelfehlersimulation der FOS's beschleunigen.

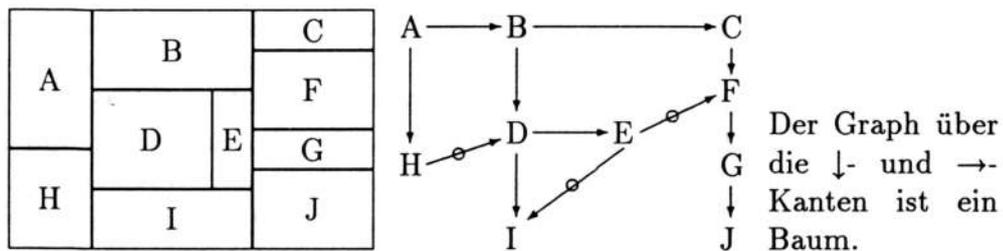
Eine dieser Methoden, der Propagationscheck, untersucht vor Start der Einzelfehlersimulation an jedem Dominator des gerade betrachteten FOS, ob lokal eine Propagation des Fehlers möglich ist. Schließt einer der Dominatoren eine Propagation aus, so ist die Einzelfehlersimulation für den betrachteten FOS überflüssig und muß folglich nicht ausgeführt werden. Desweiteren kann der Ablauf der Einzelfehlersimulation für einen bestimmten FOS durch seine „dynamischen“ Dominatoren beschleunigt werden. Dabei verstehen wir unter dynamischen Dominatoren Gatter, in denen sämtliche Propagationspfade rekonvergieren. Diese Eigenschaft ist offensichtlich belegungsabhängig.

Selbstverständlich erlaubt die Anwendung aller vorgestellten Methoden eine wortparallele Verarbeitung der Muster. Um die praktische Bedeutung der Methoden zu unterstreichen, wurde eine Implementierung durchgeführt und die Laufzeit für verschiedene Schaltungen bestimmt. Dabei konnte bei einigen Schaltungen eine erhebliche Beschleunigung des Gesamtverfahrens festgestellt werden.

Die Beschreibung von Rechteckzerlegungen und ihre Optimierung

Ernst-Günter Giessmann
Humboldt-Universität Berlin

Seit 1940 weiß man, wie man die Inzidenzen der Teilrechtecke einer Rechteckzerlegung durch einen Graphen der vertikalen und horizontalen Strecken beschreibt. Es existieren auch Algorithmen, die in linearer (bei gewissen Einschränkungen) oder quadratischer Zeit aus den Inzidenzgraphen die Rechteckzerlegungen erstellen. Diese sind so, daß bei vorgegebener Fläche jedes Teilrechteckes eine lückenlose Einbettung gefunden wird. Will man aber eine gute Einbettung suchen, die auch bezüglich der Abstände d_{ij} zwischen den Rechtecken R_i und R_j gut ist, so muß man nach einer besseren Beschreibung suchen. Und dazu bietet sich die folgende an:



Die restlichen \dashrightarrow -Kanten sind die minimaler Information, die die Inzidenzen eineindeutig beschreibt. Die oberen beiden Rechteckzerlegungen lassen sich so unterscheiden. Mit Transformationen auf den Graphen kann man dann nach einer günstigen Zerlegung suchen. Bewiesen ist, daß man alle Konfigurationen erreicht. Mit welcher Heuristik man aber schnell die beste findet, muß noch ausprobiert werden. Für das „floorplanning“ würde es ganz nützlich sein.

Zur Anwendung eines Multi-Level- Fehlersimulators

B. Straube, G. Elst, M. Galetzka, W. Vermeiren
ZKI Dresden

Ein Fehlersimulator wird für solche Netzwerke eingesetzt, für die es keine konstruktiven Verfahren zur Testpatterngenerierung gibt. Das sind i.a. Netzwerke, die (auch im Testmode) sequentiell sind. Außerdem ist es nicht ohne weiteres möglich, für beliebige Teilnetzwerke auf dem Switch-Level äquivalente Gatterersatznetzwerke mit einer adäquaten Fehlermodellierung anzugeben. Soll außerdem noch die Systemumgebung durch abstraktere Darstellungen, wie Logik-Wert-Modelle, RT-Modelle oder Funktionsprozeduren in einer üblichen Programmiersprache, in das Modell der Schaltung mit einbezogen werden, ist ein Multi-Level-Fehlersimulator erforderlich. Die Effizienz eines Multi-Level-Fehlersimulators ist wegen seines Organisationsoverheads im Vergleich zu einem Einebenen-Fehlersimulator geringer. Deshalb sollten neben dem Vorteil einer geeigneten Fehlermodellierung auf den entsprechenden Beschreibungsniveaus die Kenntnisse des Entwerfers über die strukturellen und funktionellen Zusammenhänge der Module seiner Schaltung und auch die Kenntnisse über die Zuordnung von Abschnitten des Testsatzes zu den einzelnen Moduln beim Fehlersimulationsprozeß gezielt ausgenutzt werden.

Der Aufwand für einen Fehlersimulationslauf steigt mit der Größe des Netzwerkes und mit der Anzahl der zu simulierenden Fehler (-klassen). Mit dem Multi-Level-Fehlersimulator FSIM wurden einige Experimente durchgeführt, die zeigen, daß mit einem auf der Kenntnis über die Struktur des Netzwerkes und der Testpattern basierenden Teile-und-Herrsche-Vorgehen der Fehlersimulationsprozeß handhabbar gemacht werden kann. Es sind folgende Fälle möglich:

- a) Ein hierarchisch beschriebenes Netzwerk bestehe aus n Unternetzwerken. Es erfolgen mit dem vollständigen Testsatz n Fehlersimulationsläufe, wobei in jeweils nur einem Unternetzwerk die Fehler injiziert werden, während alle anderen Unternetzwerke fehlerfrei sind. Weil die

Anzahl der in einem Lauf zu simulierenden Fehler geringer ist, wird eine geringere Rechenzeit benötigt. Die gesamte Fehlerüberdeckungsrate ergibt sich aus der gewichteten Summe der einzelnen Überdeckungsraten.

- b) Für jedes der n Unternetzwerke gibt es außerdem ein (kompakteres) Simulationsmodell. Mit dem vollständigen Testsatz erfolgen n Fehlersimulationsläufe derart, daß für die fehlerfreien Netzwerke das kompakte Modell eingesetzt wird. Neben der verringerten Anzahl von injizierten Fehlern hat das jeweilige Netzwerk eine geringere Anzahl von auszuwertenden Funktionen. Es muß aber gesichert werden, daß ein Unternetzwerk und sein Simulationsmodell auch bezüglich der Fehlerübertragung gleiches Verhalten haben.
- c) Es sei weiterhin bekannt, welche Abschnitte des Testsatzes sich auf welche Module der Schaltung beziehen. Das Netzwerk wird nur mit dem Testsatzabschnitt simuliert, der sich auf ein Unternetzwerk mit den injizierten Fehlern bezieht. Damit wird die Menge der Fehler durch Fault-dropping wesentlich verringert, ehe der Gesamtestsatz auf das Netzwerk mit einer wesentlich geringeren Anzahl noch nicht erkannter Fehler angewendet wird.

Die vorgeschlagene Vorgehensweise wirkt sich außerdem vorteilhaft aus, weil die durch den Nutzer auszuwertenden Listen der nichtentdeckten Fehler für einen Lauf wesentlich kürzer sind und falls der vorhandene Testsatz noch ergänzt werden muß, weil sein Überdeckungsgrad nicht ausreichend ist.

Verfahren zur Erhöhung der Zuverlässigkeit von Schaltkreisen

Rüdiger Reischuk
TH Darmstadt

Ein gestörter Schaltkreis besteht aus Elementen (Gattern und Drähten), von denen jedes unabhängig von allen anderen mit WS höchstens ϵ inkorrekt

arbeitet ($0 < \epsilon < \frac{1}{2}$). Es stellt sich die Aufgabe, einen fehlertoleranten Schaltkreis C aus diesen Elementen für eine gegebene Boolesche Funktion f zu konstruieren, d.h. C berechnet f korrekt mit WS mindestens $1 - \delta$ für ein δ mit $\epsilon < \delta < \frac{1}{2}$.

Ergebnisse von Neumann, Dobrushin/Ortyukov und Pippenger zeigen, daß Fehlertoleranz durch Redundanz und Fehlerkorrektur erreicht werden kann. Diese Verfahren erhöhen die Schaltkreisgröße um einen logarithmischen Faktor. Für die Funktion $x_1 \oplus x_2 \oplus \dots \oplus x_n$ ist gezeigt worden, daß ein logarithmischer Mehraufwand für Fehlertoleranz notwendig ist.

Wir diskutieren Verallgemeinerungen dieser unteren Schranke.

Betrachtet man andererseits Flächen von 2-dimensionalen Layouts gewöhnlicher bzw. fehlertoleranter Schaltkreise, so zeigen wir: Durch Wahl spezieller Codes für die Informationsübertragung in langen Drähten und anschließende Kompaktifizierung kann man für viele Funktionen das Anwachsen der Fläche begrenzen. Anstelle einer quadratisch-logarithmischen Flächenredundanz, die sich durch eine Einbettung des fehlertoleranten Schaltkreises in einfacher Weise erzielen läßt, genügt oftmals bereits konstanter Mehraufwand.

Diese Ergebnisse wurden gemeinsam mit Bernd Schmeltz, Darmstadt, erzielt.

Einbettung baumartiger und würfelartiger Graphstrukturen in 2-dimensionale Gitter

Peter Zienicke
Humboldt-Universität zu Berlin

Einbettungen von Graphen in 2-dimensionale Gitter werden als Modell für VLSI-Schaltkreisentwurf betrachtet. Dabei wird ein Modell benutzt, bei dem sowohl die Leitungen über Funktionselemente „hinweglaufen“ dürfen, als auch mehrere Leitungen durch dieselbe Gitterkante verlaufen können.

In diesem Zusammenhang interessiert sich die Arbeit insbesondere für die erhaltene Leitungslänge d („Verzögerung“) als auch für die Mehrfachnutzung c („Spuranzahl“) von Gitterkanten von Einbettungen im Vergleich zum

Verhältnis e der Knotenanzahlen des Gitters und des eingebetteten Graphen („Expansion“).

Es werden für verschiedene Graphfamilien Einbettungen angegeben und es erfolgt in vielen Fällen ein Optimalitätsnachweis solcher Einbettungen. Für binäre Hyperwürfel werden untere Schranken für Verzögerung ($ed^2 = \Omega(\frac{2^n}{n})$) und Spuranzahl ($ec^2 = \Omega(2^n)$) bewiesen, sowie je eine Einbettung angegeben, die bei Expansion 1 eine dieser Schranken erreicht.

Für die Familie der Pyramiden wird eine Einbettung angegeben, die mit Spuranzahl 3 und Expansion < 3 auskommt, in dieser Beziehung also nicht wesentlich verbessert werden kann.

Es wird für X-Bäume gezeigt, daß sich diese in ihr optimales quadratisches Gitter (minimale Expansion) mit konstanter Spuranzahl 6 einbetten lassen, für vollständige binäre Bäume wird Spuranzahl 2 als ausreichend nachgewiesen.

Außerdem wird für die Verzögerung bei vollständigen binären Bäumen eine untere Schranke von $O(\frac{2^{n/2}}{n})$ bewiesen, sowie eine Einbettung gezeigt, die diese Schranke erreicht. Letzteres basiert auf einem Beweis, der das Bandweitenproblem mit Bandweite $O(\frac{2^n}{n})$ für vollständige binäre Bäume der Höhe n löst.

Verifikationsfreundlicher Entwurf (Design for Verifiability, DfV)

Hans Evekling
Institut für Datentechnik
Technische Hochschule Darmstadt

„Design for Verifiability“ meint in Analogie zu „Design for Testability“ die Unterstützung der vollständigen und mechanisierten Verifikation eines Hardwaresystems durch Maßnahmen während des Entwurfs.

Diese Maßnahmen zielen auf eine Unterteilung des Verifikationsproblems in einfacher zu lösende Teilprobleme. Dazu gehört insbesondere

- die Dokumentation eines Entwurfs auf mehreren Abstraktionsebenen (vertikale Dekomposition). Dadurch wird beispielsweise die

Überprüfung der korrekten Implementierung eines Maschinenbefehlsatzes durch ein Transistornetz aufgespalten in die Verifikation einer Reihe von Beschreibungen auf jeweils benachbarten Abstraktionsebenen.

- die Zerlegung eines Entwurfs in Teile (horizontale Dekomposition), die unabhängig voneinander verifiziert werden. Bei der Zusammensetzung korrekt implementierter Teile muß geprüft werden, ob die Annahmen erfüllt sind, die für eine korrekte Implementierung der Teile erfüllt sein müssen.

Vertikale und horizontale Dekomposition verlangen deutlich mehr an Dokumentation von Entwurfsentscheidungen, als gegenwärtig in der Entwurfspraxis üblich ist. Zur Dokumentation werden Hardwarebeschreibungssprachen mit definierter Semantik benutzt. DfV hängt ebenso wie DfT stark von den jeweils bekannten Methoden und Techniken der Verifikation sowie den verfügbaren Hilfsmitteln ab. Gegenwärtig gibt es viele Anwendungen, die BDD's (binary decision diagrams) erfolgreich einsetzen.

Über einen zellenorientierten Fehlersimulator für kombinatorische Schaltkreise

Björn Schieffer
Universität des Saarlandes

Die Mängel eines festen Fehlermodells wurden schon vielfach aufgezeigt. Ferguson und Shen haben infolgedessen einen Fehlergenerator vorgestellt. Leider können die meisten Testalgorithmen aber keine externen Fehlermodelle übernehmen, sondern ziehen sich auf ein fest eingebautes – i.A. das stuck-at Modell – zurück. Hier wird nun eine Verlagerung der Fehlerbeschreibung in eine externe Zellenbibliothek vorgeschlagen. Sie sollte dabei beliebige kombinatorische Fehlverhalten modellieren können. Am Beispiel der Fehlersimulation durch Compiler single fault propagation habe ich gezeigt, daß eine Verallgemeinerung des Fehlermodells und das Herausziehen

aus dem Algorithmus nicht zu einer Laufzeitsteigerung führen muß. Durch vorberechnetes Wissen, abgelegt in der Zellenbibliothek, läßt sich im Gegenteil ein deutlicher Zeitgewinn feststellen.

The random testability of the n -input AND gate

Joachim Hartmann
Universität des Saarlandes

Probabilistic testing is very attractive due to the low test generation costs. Unfortunately, not all circuits are well random testable. To eliminate this deficiency, biased random testing has been adopted. We distinguish input signal biased (ISB) random testing, where a distribution different from the uniform one is achieved by giving each primary input an individual signal probability from pattern biased (PB) random testing, which allows an arbitrary distribution for the input patterns.

An extreme example, illustrating the poor conventional random testability is the n -input AND gate. It is shown that its expected test length is $2^n \cdot H_{n+1}$, where $H_k = 1 + \frac{1}{2} + \dots + \frac{1}{k}$. For PB random testing the optimal strategy is presented. It yields $(n+1) \cdot H_{n+1}$ as expected test length. It is shown how the optimal signal probabilities for ISB random testing can be calculated on condition that the signal probabilities at all inputs are equal. Furthermore, $e \cdot n \cdot H_{n+1}$ ($e=2.71\dots$) is proved to be an upper bound for the best achievable expected test size by ISB random testing. Hence it follows that PB random testing is not much superior to ISB random testing for AND gates.

Ausnutzung funktionaler Eigenschaften zur Erhöhung der Verfügbarkeit von Computern

Günter Hotz
Universität des Saarlandes

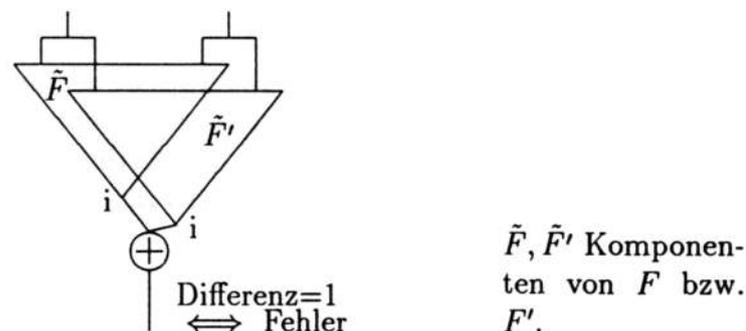
Das Problem: Ein Computer an einer schwer zugänglichen Stelle, z.B. im Weltraum, ist ein wenig defekt. Wie kann man den Computer sinnvoll weiter verwenden. Wir betrachten zwei verschiedene Situationen. In der einfachen Situation können wir in einem Sonderfall eine gute Lösung angeben.

Die Idee: Durch einen Test, der in gewisser Regelmäßigkeit wiederholt wird (Situation 1) oder durch eine permanente Kontrolle der Berechnungen (Situation 2) wird festgestellt, daß z.B. die Multiplikation nicht in jedem Fall korrekt ausgeführt wird. Ist $a \cdot b$ eine solche Multiplikationsaufgabe, dann beschaffe man sich eine geeignete Zahl x und berechne $(a - x) \cdot b + x \cdot b = a \cdot b$.

Dies tut man in der Hoffnung, daß $(a - x) \cdot b$ und $x \cdot b$ noch korrekt ausgeführt werden können.

Diese Idee läßt sich so aufgrund der Unsymmetrie bzgl. der Behandlung der beiden Faktoren nicht realisieren. Es gelingt das aber durch eine leichte Variation des Ansatzes.

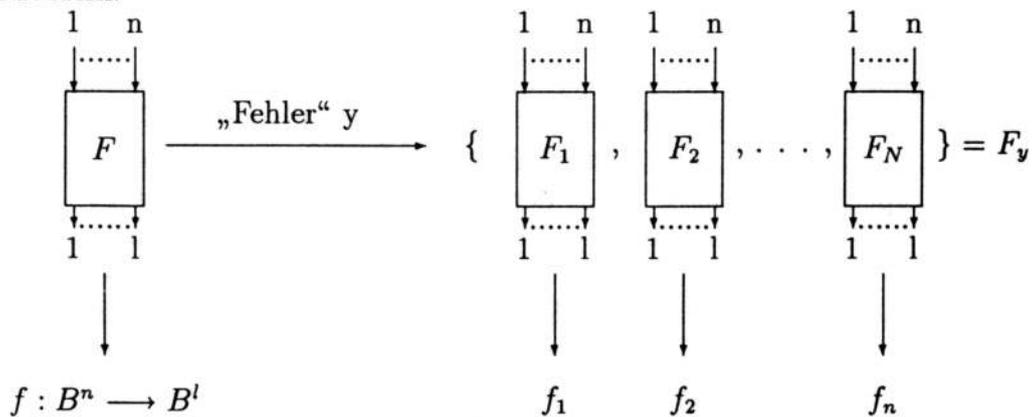
Wir fassen das Problem zunächst etwas allgemeiner.



Haben wir in einem Schaltkreis F die Situation, daß an der Stelle i ein Fehler vorliegt, dann ist eine Eingabe von F ein Test bezüglich dieses Fehlers, wenn F und der durch den Fehler aus F hervorgehende Schaltkreis F' an der

Stelle i verschiedene Werte erzeugen und wenn die Verschiedenheit der Werte in F und F' auf mindestens einem Berechnungspfad in den Netzen erhalten bleibt.

Wir nennen y einen OK-Wert von F bzgl. i , falls y kein Test von F bzgl. i ist. Führt der Test y zu einem Fehler, dann ist ein y' zu berechnen, das ein OK-Wert ist, um y' und andere vorhandene Operationen zu nutzen, um den Fehler i von F zu umgehen. Die Schwierigkeit besteht darin, daß sehr viele verschiedene Fehler i existieren können, für die y ein Test ist. So lautet das Problem:



Gegeben F und wenig verschiedene Netze F_1, \dots, F_n , die durch y unterscheidbar sind. Verwende irgendein f_i , von dem nichts als $F_i \in F_y$ bekannt ist, y und weiter verfügbare Operationen g_1, \dots, g_l um f zu realisieren.

Im Falle der Multiplikation mittels eines einfachen Multiplikationsnetzes kann man eine einfache Lösung angeben. Diese wird möglich bei Verwendung des double-line-Tricks, der es unter Verwendung der 1-Fehlerhypothese erlaubt, unter höchstens 8 Versuchen eine korrekte Umgehung des Fehlers zu finden.

Literatur: Diplomarbeit von B.Müller am Fachbereich Informatik der Universität des Saarlandes.

Probleme zur Lösung der globalen Phase des Layoutentwurfs bei Gate-Arrays und bei der Parallelisierung dieses Prozesses

Wolfgang Schade, B.Goetze, J.Liske, W.Nehrlich, B.Stube
Karl-Weierstraß-Institut für Mathematik
Mohrenstr. 39
D O-1086 Berlin

Ausgehend von einem Master, der nach dem Transistorsee-Konzept erstellt ist, jedoch in Flächen für Kombinatorikgatter, Flip-Flops sowie spezifische, z.T. vorplazierte Teilschaltungen (RAM, ROM, ...) unterteilt ist, wird ein Konzept für die globale Phase der Layoutgenerierung vorgestellt.

Es besteht aus der Auflösung und Partitionierung der Schaltung zu Clustern, der Einordnung der Schaltungscluster in gleichgroße Boxgebiete, in die der Master zerlegt wird, und dem globalen Verdrahten der Cluster über alle Boxen. Bei der Auflösung der Schaltungshierarchie (die Beschreibung erfolgt in NBS-84) ist darauf zu achten, daß fest vorplazierte und Schaltkreise mit spezifischem Untergrund (RAM, ROM, ...) von der restlichen Schaltung separiert werden und daß die gebildeten Cluster gewisse „Wunschclustergrößen“ (z.B. hinsichtlich ihres Kombinatorik- bzw. FF-Bedarfs) erfüllen, so daß für das Einpacken in die Boxen Spielraum besteht. Die relative Platzierung der Cluster erfolgt mittels des bekannten Kräfte Modells von Quinn/Breuer. Das Einpacken der Cluster in die Boxen unter Berücksichtigung der dort vorgegebenen Höchstgrenzen für FF bzw. Kombinatorikgatter und der höchstmöglichen Anzahl von zugelassenen externen Potentialen ist bei Minimierung der Kosten (Entfernung relative Lage des Clusters zum Boxmittelpunkt) NP-vollständig. Eine wirksame Heuristik wird angegeben. Die globale Verdrahtung erfolgt sukzessive über die Berechnung minimaler Steinerbäume. Netze, die über zu stark beanspruchte Kanten verlaufen, werden nach Erhöhung der Kantenübertrittskosten neu verlegt.

Vorgestellt wird eine Idee, wie global verlaufende Netze auf den Boxkanten vorsortiert werden können, um möglichst wenig globale Überkreuzungen zu erhalten. Diese Methode kann auf ein Skalierungsproblem zufälliger Variabler zurückgeführt werden und resultiert in der Ermittlung von Eigenschaften

einer quadratischen Form zur Maximierung eines (linearen) Korrelationskoeffizienten. Diese Methode liefert auch erste Anhaltspunkte für die relative Platzierung der Cluster in den einzelnen Boxen.

Damit ist es möglich, bei einer Parallelisierung des Layoutprozesses, bei dem die Probleme innerhalb der Boxen separat – z.B. durch Transputer – gelöst werden, den etwaigen Übertritt der externen Potentiale über die Boxkanten vorzugeben. Bei einer durch Transputer vorzunehmenden Boxverdrahtung ist dann von Zeit zu Zeit der angestrebte exakte Kantenübertritt mit dem benachbarten Prozessor abzustimmen.

Über ein Verfahren zur effektiven mehrstufigen Realisierung Boolescher Funktionen

Michael Weber
Humboldt-Universität zu Berlin

Das vorgestellte Verfahren wurde an der WMK der Moskauer Staatlichen Universität von Loshkin und anderen entwickelt und implementiert. Es steht die Aufgabe, zu einer i.a. unvollständig durch Tabellen gegebenen Funktion $f : B^m \rightarrow B^n$ eine effektive mehrstufige Realisierung durch Gatter einer vorgegebenen Basis zu synthetisieren. Grundlage ist die Zerlegung von Funktionen nach Variablen entsprechend dem Shannonschen Verfahren. Die entstehende Zerlegung hängt ab von der Reihenfolge, in der nach den Variablen entwickelt wird. In dem Verfahren wird versucht, durch Reduktion der Anzahl sowie der Komplexität der benötigten Unterfunktionen eine günstige Realisierung der Gesamtfunktion zu erzielen. In einem ersten Schritt wird eine günstige Reihenfolge der Variablen ermittelt, nachfolgend erfolgt die Synthese, bei der gemäß der gefundenen Reihenfolge jeweils nach der nächsten Variablen entwickelt wird und versucht wird, die entstehenden Unterfunktionen durch Gatter der Basis auf andere Unterfunktionen oder direkt auf Variablen zurückzuführen.

Über die Komplexität zuverlässiger kombinatorischer Netzwerke

Dietmar Uhlig
Ingenieurhochschule Mittweida

Es wurden Komplexitätsvergleiche minimaler unzuverlässiger und minimaler zuverlässiger Realisierungen durch kombinatorische Netzwerke von „fast allen“ Booleschen Funktionen betrachtet. Man sagt, daß „fast alle“ Booleschen Funktionen eine gewisse Eigenschaft haben, wenn die Anzahl der Booleschen Funktionen $f(x_1, \dots, x_n)$, die diese Eigenschaft haben, asymptotisch gleich der Anzahl aller Booleschen Funktionen von den Variablen x_1, \dots, x_n ist.

Es sei $C(F)$ die Komplexität von F , definiert als minimale Anzahl an binären Elementen, die zur Realisierung von F benötigt werden, und es sei analog dazu $C_r(F)$ die minimale Anzahl an binären Elementen, die zur Realisierung von F durch r -Fehler-korrigierende kombinatorische Netzwerke benötigt werden, welche mehrere Ausgänge haben und auch dann an den meisten Ausgängen den richtigen Wert ausgeben, wenn maximal r beliebige Elemente ausgefallen sind. Es läßt sich zeigen, daß für $\epsilon > 0$ und $r(n) = 2^{O(n)}$ für „fast alle“ Booleschen Funktionen $f(x_1, \dots, x_n)$

$$C_{r(n)}(f(x_1, \dots, x_n)) \leq (1 + \epsilon) \cdot C(f(x_1, \dots, x_n)) \quad (*)$$

gilt (Kirienko (1970), Uhlig (1974, 1975)).

Analoges gilt für die simultane Berechnung von f auf $r(n) = 2^{O(\frac{n}{\log_2 n})}$ Booleschen Vektoren (Uhlig (1974)). Der dazu entwickelte Code bietet eine Möglichkeit für die Konstruktion von Speichern mit uneingeschränktem Simultanzugriff.

Es wurden auch Realisierungen von Booleschen Funktionen durch zuverlässige kombinatorische Netzwerke betrachtet, die man bei geeigneter Einbeziehung von (in der Regel sehr vielen) Majoritätsschaltungen mit einer Ausfallwahrscheinlichkeit $\delta > 0$ erhält und deren Ausfallwahrscheinlichkeiten für $\gamma > 0$ den Wert $(1 + \gamma) \cdot \delta$ nicht übersteigen. In diesem Falle ergibt sich

Korrekturblätter zum Dagstuhl-Seminar-Report; 6 18.-22.2.91 (9108)

S. 24: in den beiden Gleichungen $r(n)=\dots$ ist im Exponenten das große O durch ein kleines o zu ersetzen.

S. 25 und die Teilnehmerliste fehlen:

– falls δ hinreichend klein ist – eine zu (*) ähnliche Beziehung für „fast alle“ Booleschen Funktionen (Uhlig (1987)). Die Anregung zu diesem Ergebnis stammt aus einer originellen Pippenger-Arbeit (1985), der nachwies, daß für „fast alle“ Booleschen Funktionen für die zuverlässige Realisierung ein (relativ großes) Vielfaches an Elementen ausreichend ist. Die erste fundamentale Arbeit auf diesem Gebiet stammt von v. Neumann (1952).

Teilnehmer:

Frank Bauernöppel
Fachbereich 16 - Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Bernd Becker
Fachbereich 20 - Informatik
Johann Wolfgang Goethe-Universität Frankfurt
Postfach 11 19 36
W-6000 Frankfurt

Hans Eveking
Institut für Datentechnik
TH Darmstadt
Merkstr. 25
W-6100 Darmstadt

Ernst-Günter Giessmann
Fachbereich Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Winfried Görke
Inst. für Rechnerentwurf & Fehlertoleranz
Universität Karlsruhe
Postfach 6980
W-7500 Karlsruhe 1

Michael Gössel
Zentralinstitut für Kybernetik und Information-
prozesse
Kurstraße 33
Postfach 1298
O-1086 Berlin

Ralf Hahn
Fachbereich 20 - Informatik
J.W.Goethe - Universität
W-6000 Frankfurt

Wolfgang Hahn
Zentralinstitut für Kybernetik und

Informationsprozesse
Kurstraße 33
O-1086 Berlin

Joachim Hartmann
Fachbereich 14
Universität des Saarlandes
W-6600 Saarbrücken 11

Günter Hotz
Fachbereich 14
Universität des Saarlandes
W-6600 Saarbrücken 11

Rainer Kolla
Fachbereich 14
Universität des Saarlandes
W-6600 Saarbrücken 11

Matthias Krause
Fachbereich Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Rolf Krieger
Fachbereich 20 - Informatik
J.W. Goethe - Universität
W-6000 Frankfurt

Lars Kühnel
Inst. für Informatik und Praktische Mathematik
Universität Kiel
Olshausenstraße 40-60
W-2300 Kiel 1

Christoph Meinel
Fachbereich Informatik
Humboldt-Universität Berlin
O-1086 Berlin

Ralf Oelschlägel
Humboldt-Universität Berlin

Fachbereich Informatik
O-1086 Berlin

Rüdiger Reischuk
Institut für Theoretische Informatik
TH Darmstadt
Alexanderstr. 10
W-6100 Darmstadt

Reinhard Reisig
Fachbereich Mathematik - Informatik 17
Universität - Gesamthochschule Paderborn
W-4790 Paderborn

Wolfgang Schade
Karl-Weierstraß-Institut für Mathematik
Mohrenstraße 39
O-1086 Berlin

Björn Schieffer
Lehrstuhl Prof. Hotz
Informatik
Universität des Saarlandes
W-6600 Saarbrücken 11

Harald Schmeck
Inst. für Informatik und Praktische Mathematik
Universität Kiel
Olshausenstraße 40
W-2300 Kiel 1

Uwe Sparmann
Fachbereich 14
Universität des Saarlandes
W-6600 Saarbrücken 11

Bernd Straube
Zentralinstitut für Kybernetik und
Informationsprozesse
Institutsteil Dresden
Haeckelstraße 20
O-8027 Dresden

Steffen Tarnick
Zentralinstitut für Kybernetik und
Informationsprozesse
Kurstraße 33

O-1086 Berlin

Dietmar Uhlig
Ingenieurhochschule Mittweida
Platz der DSF 17
O-9250 Mittweida

Lutz. Voelkel
Hochschule für Seefahrt
Warnemünde-Wustrow
FB Elektrotechnik/Informationstechnik
LD Technische Informatik
Richard-Wagner-Straße 31
O-2530 Warnemünde

Stephan Waack
Karl-Weierstraß-Institut für Mathematik
Mohrenstr. 39
O-1086 Berlin

Michael Weber
Fachbereich Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Reiner Witschurke
Karl-Weierstraß-Institut für Mathematik
Mohrenstraße 39
O-1086 Berlin

Steffen Zahn
Fachbereich 16 - Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Peter Zienicke
Fachbereich Informatik
Humboldt-Universität Berlin
Postfach 1297
O-1086 Berlin

Christoph Zimmermann
TU Chemnitz
Sektion Informationstechnik
PSF 964
O-9010 Chemnitz

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):

Architecture and Performance, Dagstuhl-Seminar-Report; 1,
18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):

Tree Adjoining Grammars, 1st. International Workshop on TAGs: Formal Theory
and Application, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):

Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3,
3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):

Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

J. Berstel, J.E. Pin, W. Thomas (editors):

Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-
Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):

Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):

Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-
1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):

Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991
(9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):

Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-
Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):

Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10,
18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):

Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report;
11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):

Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):

Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, ,
20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):

Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):

Query Processing in Object-Oriented, Complex Object, and Nested Relation Data-
bases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):

Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Re-
port; 16, 24.-28.6.1991 (9126)