Zur effizienten Testerzeugung für sequentielle Fehlermodelle*

Björn Schieffer Joachim Hartmann Uwe Sparmann
Lehrstuhl Prof. Hotz
Fachbereich 14.1 – Informatik
Universität des Saarlandes
W-6600 Saarbrücken

Zusammenfassung

Es wird ein allgemeiner Ansatz der Modellierung sequentieller Fehler auf seine Anwendbarkeit für Fehlersimulation und Testmustergenerierung untersucht. Zur Steigerung der Effizienz wird das Konzept der blind homing sequence vorgestellt.

1 Einleitung

Die bei der Fertigung höchstintegrierter Schaltkreise auftretenden physikalischen Defekte hängen von einer Vielzahl verschiedener Faktoren ab, wie z.B. der Technologie, dem Fertigungsprozeß oder dem Layout der Grundzellen der verwendeten Bibliothek. Um diese Vielfalt von Fehlermechanismen korrekt zu beschreiben, werden zur Zeit Methoden zur automatischen Extraktion von Fehlermodellen entwickelt [Mal87, HW88]. Auf der anderen Seite fehlen aber effiziente Algorithmen zur Fehlersimulation und Testmustergenerierung, die nicht, wie heute üblich [AS87, BHKS91, Köp86, MG91], auf ein spezielles Fehlermodell zugeschnitten sind, sondern beliebige kombinatorische oder sequentielle Fehlermechanismen bearbeiten können.

In [HSS91] wurde die Implementierung eines Fehlersimulators für beliebige kombinatorische Fehlermodelle vorgestellt. Hierbei hat es sich gezeigt, daß für die Effizienz von Testalgorithmen geeignete Vorberechnungen auf der Bibliothek der Fehlerbeschreibungen von besonderer Bedeutung sind. Es verbleibt die Frage, in wie weit solche Vorberechnungen auch für den Fall beliebiger sequentieller Fehlermodelle ausgenutzt werden können. In dieser Arbeit wird, der Begriff der *blind homing sequence* eines sequentiellen Fehlers definiert und gezeigt, daß er in diesem Zusammenhang von entscheidender Bedeutung ist. Verfahren zur Vorberechnung wichtiger

*Die Arbeit wurde unterstützt von der DFG im Rahmen des SFB 124, TP B1, VLSI Entwurfsmethoden und Parallelität.

Fehlereigenschaften wie der kürzesten blind homing sequence oder der erwarteten Länge einer blind homing sequence bei gewürfelten Eingaben werden vorgestellt.

2 Der Begriff der blind homing sequence und seine Anwendungen

Wir betrachten Fehlermechanismen, die einen kombinatorischen Baustein zu einer Zelle mit sequentiellem Verhalten modifizieren. Die Beschreibung der sequentiellen Fehler zu einem bestimmten Zelltyp erfolgt in der Fehlerbibliothek in Form von endlichen Automaten.

Nehmen wir an, daß der Startzustand des Fehlerautomaten bekannt ist. Um den sequentiellen Fehler zu finden, muß der Automat zunächst durch eine Folge von Eingaben vom Startzustand in einen „Entdeckungszustand“ gebracht werden, in dem sich sein Ein-/Ausgabe-Verhalten von dem der korrekten Zelle unterscheidet. Anschließend wird, wie auch bei der Behandlung kombinatorischer Fehler, eine Eingabekombination angelegt, die den Fehler an den Outputs der Zelle sichtbar macht und den Fehlereffekt zu den primären Outputs des Schaltkreises propagiert.

Ist der Startzustand des Fehlerautomaten nicht bekannt, so müssen die obigen Berechnungen für jeden möglichen Startzustand durchgeführt werden. Die Anzahl der möglichen Zustände geht somit als Faktor in die Laufzeit des entsprechenden Testalgorithmus ein. Dies läßt sich vermeiden, wenn die Testfolge zuerst dafür sorgt, daß man unabhängig vom Startzustand zu demselben Zustand gelangt. Bei stuck-open-Fehlern erreicht man das mit einem einzigen Eingabemuster. Man kann sie daher mit Folgen aus nur zwei Mustern testen [Köp86]. Im allgemeinen Fall eines sequentiellen Fehlers nennen wir eine solche Eingabefolge, die unabhängig vom Startzustand immer zum selben Zustand führt, eine *blind homing sequence*. Der Name soll an den Begriff der *homing sequence* [Gil62] erinnern. Im Gegensatz zur homing sequence wird jedoch davon ausgegangen, daß die Ausgaben des endlichen Automaten nicht beobachtet werden können, der eindeutige Zustand also „blind“ bestimmt werden muß. Diese Forderung ist notwendig, da die fehlerhafte Zelle in einen komplexen Schaltkreis eingebettet ist, der die Beobachtung ihrer Ausgaben nur bedingt erlaubt.

Ein Fehlersimulator oder Testmustergenerator liest die Beschreibung eines sequentiellen Fehlers in der Bibliothek. Er muß entscheiden, ob für diesen Fehler ein „einfacher“ Test mithilfe einer blind homing sequence sinnvoll ist. Hierzu benötigt er Wissen über deren Existenz und Struktur. Gibt es zu einem Fehler eine sehr kurze blind homing sequence , so ist dies vor allem für die Testmustergenerierung von Vorteil. Wie lang also kann eine kürzeste sein, und wie findet man diese? Auch bei der Fehlersimulation erlauben blind homing sequences eine Beschleunigung der Zustandsberechnung für die Fehlerautomaten unter Ausnutzung einer wortparallelen Verarbeitung [Sch91]. Von zentraler Bedeutung ist hierbei die mittlere Länge einer blind homing sequence bei gewürfelten Eingaben. Ist die mittlere Länge gering, so

kann z.B. die zu simulierende Eingabefolge in Wortbreite viele Teilfolgen aufgespalten werden, die muster-parallel simuliert werden. Der Nachteil, daß hierbei für jede Teilfolge der Startzustand des Fehlerautomaten nicht bekannt ist, wird schnell durch eine blind homing sequence bereinigt[1].

3 Berechnung wichtiger Eigenschaften von blind homing sequences

Die vorhergehenden Ausführungen haben gezeigt, daß Eigenschaften von blind homing sequences sequentieller Fehler essentiell für die Effizienz von Testalgorithmen sind. Wir haben Algorithmen entwickelt, die diese Eigenschaften für die Fehlerautomaten einer Fehlerbibliothek entscheiden, um sie den entsprechenden Tools zur Verfügung zu stellen.

Die Frage nach der Existenz einer blind homing sequence kann in quadratischer Laufzeit in der Größe des Fehlerautomaten beantwortet werden. Zur Lösung der Probleme „Auffinden einer kürzesten blind homing sequence “ und „Bestimmung der mittleren Länge bei gewürfelter Eingabe“ wird exponentielle Laufzeit benötigt. Hierfür muß also die Größe der zu untersuchenden Automaten beschränkt werden.[2]

Um einen Eindruck zu erhalten, wie lang die kürzeste blind homing sequence – und damit wie groß der Aufwand, mit ihr einen Test zu generieren – werden kann, haben wir sie nach oben abgeschätzt. Besitzt ein Automat eine blind homing sequence, so gibt es auch eine mit einer Länge von höchstens $1 + \frac{n(n-1)(n-2)}{2}$, wobei n die Anzahl der Zustände des Automaten bezeichnet. Daß diese Abschätzung nicht sehr pessimistisch ist, folgt aus der Existenz eines Automaten dessen kürzeste blind homing sequence die Länge $n(n-2)+1$ hat.

4 Schlußbemerkung und Ausblick

In dieser Arbeit wurde untersucht, wie durch geeignete Vorberechnungen auf der Fehlerbibliothek wichtige Informationen zur Beschleunigung der Testerzeugung bzgl. sequentieller Fehlermodelle berechnet werden können. Leider konnten durch die angestrebte Kürze die Probleme und ihre Lösungsansätze nur angedeutet werden. Detailliertere Informationen finden sich in [Sch91].

In zukünftigen Arbeiten möchten wir die gewonnenen Erkenntnisse in eine Erweiterung des Fehlersimulators COFS [HSS91] einbringen.

[1] Hierbei wird an den Anfangsstücken der Teilfolgen zu pessimistisch gerechnet, der dabei gemachte Fehler kann aber (falls notwendig) leicht korrigiert werden [Sch91].

[2] Für die Praxis dürfte dies keine große Einschränkung darstellen, da die Anzahl der Zustände eines Fehlerautomaten in der Regel gering sein wird.

Danksagungen

Der Dank der Autoren gilt Prof. Dr. Günter Hotz für seine Unterstützung und Ratschläge.

Literatur

[AS87] Kurt J. Antreich and Michael H. Schulz. Accelerated fault simulation and fault grading in combinational circuits. *IEEE Transactions on Computer-Aided Design*, CAD-6(5):704–712, September 1987.

[BHKS91] B. Becker, R. Hahn, R. Krieger, and U. Sparmann. Structure based methods for parallel pattern fault simulation in combinational circuits. In *Proceedings of the 2nd European Design Automation Conference (EDAC91)*, 1991.

[Gil62] A. Gill. *Introduction to the theory of finite-state machines.* McGraw - Hill, 1962.

[HSS91] J. Hartmann, B. Schieffer, and U. Sparmann. COFS – a cell oriented fault simulator. In *Kurzfassungen der Beiträge zum 3. ITG/GI-Workshop: Testmethoden und Zuverlässigkeit von Schaltungen und Systemen*, 1991.

[HW88] J. Holzinger and H.J. Wunderlich. Automatische Fehlermodellierung und Testerzeugung aufgrund der Layoutbeschreibung von MOS-Schaltungen. In *Kurzfassungen der Beiträge zum Workshop der Siemens AG: Neue Forschungsaspekte zum Testen von VLSI-Schaltungen*, 1988.

[Köp86] S. Köppe. Modeling and simulation of delay faults in CMOS logic circuits. In *Proceedings 1986 International Test Conference*, pages 530 – 536, 1986.

[Mal87] W. Maly. Realistic fault modeling for VLSI testing. In *Proceedings 24th ACM/IEEE Design Automation Conference*, pages 173–180, 1987.

[MG91] Steven D. Millman and James P. Garvey. An accurate bridging fault test pattern generator. In *Proceedings 1991 International Test Conference*, pages 411 – 418, 1991.

[Sch91] B. Schieffer. Hierarchische Fehlersimulation mit Nicht-Standard-Fehlermodellen. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, W–6600 Saarbrücken 11, FRG, 1991.