

# Robuster Selbsttest mit Diagnose

Cook, Alejandro; Hellebrand, Sybille; Indlekofer, Thomas;  
Wunderlich, Hans-Joachim

5. GMM/GI/ITG-Fachtagung Zuverlässigkeit und Entwurf (ZuE'11) Vol. 231,  
Hamburg-Harburg, Germany, 27-29 September 2011

url: <http://www.vde-verlag.de/proceedings-en/453357011.html>

**Abstract:** Robuste Schaltungen können bestimmte Fehler tolerieren, stellen aber auch besonders hohe Anforderungen an Test und Diagnose. Um Ausbeuteverluste zu vermeiden, muss der Test kritische Fehler von unkritischen Fehlern unterscheiden, die sich während des Systembetriebs nicht auswirken. Zur Verbesserung des Produktionsprozesses muss außerdem eine effiziente Diagnose für erkannte kritische Fehler unterstützt werden. Bisherige Arbeiten für Schaltungen mit Zeitredundanz haben gezeigt, dass ein Selbsttest mit Rücksetzpunkten kostengünstig kritische permanente Fehler von unkritischen transienten Fehlern unterscheiden kann. Hier wird der Selbsttest in N Sitzungen unterteilt, die bei einem Fehler sofort wiederholt werden. Tritt beim zweiten Durchlauf einer Sitzung kein Fehler mehr auf, geht man von einem transienten Fehler aus. Dabei genügt es, die Referenzantworten für die einzelnen Sitzungen in stark kompakter Form abzulegen. Statt einer vollständigen Signatur wird nur eine kurze Bitfolge gespeichert, welche die Signaturberechnung über mehrere Zeitpunkte hinweg charakterisiert. Die vorliegende Arbeit erweitert das Testen mit Rücksetzpunkten zu einem integrierten Test- und Diagnoseprozess. Es wird gezeigt, dass ein einziger Testdurchlauf mit stark kompaktierten Referenzwerten genügt, um eine vergleichbare diagnostische Auflösung zu erreichen wie bei einem Test ohne Antwortkompaktierung.

Preprint

## General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by *VDE Verlag*.

©2011 VDE Verlag

# Robuster Selbsttest mit Diagnose

Alejandro Cook<sup>1</sup>, Sybille Hellebrand<sup>2</sup>, Thomas Indlekofer<sup>2</sup>, Hans-Joachim Wunderlich<sup>1</sup>

<sup>1</sup>Institut für Technische Informatik, Universität Stuttgart

<sup>2</sup>Institut für Elektrotechnik und Informationstechnik, Universität Paderborn

## Kurzfassung

Robuste Schaltungen sind in der Lage, bestimmte Fehler zu tolerieren, und stellen besonders hohe Anforderungen an Test und Diagnose. Um Ausbeuteverluste zu vermeiden, muss der Test kritische Fehler von unkritischen Fehlern unterscheiden, die sich während des Systembetriebs nicht auswirken. Zur Verbesserung des Produktionsprozesses muss außerdem eine effiziente Diagnose für erkannte kritische Fehler unterstützt werden. Bisherige Arbeiten für Schaltungen mit Zeitredundanz haben gezeigt, dass ein Selbsttest mit Rücksetzpunkten kostengünstig kritische permanente Fehler von unkritischen transienten Fehlern unterscheiden kann. Hier wird der Selbsttest in  $N$  Sitzungen unterteilt, die bei einem Fehler sofort wiederholt werden. Tritt beim zweiten Durchlauf einer Sitzung kein Fehler mehr auf, geht man von einem transienten Fehler aus. Dabei genügt es, die Referenzantworten für die einzelnen Sitzungen in stark kompakter Form abzulegen. Statt einer vollständigen Signatur wird nur eine kurze Bitfolge gespeichert, welche die Signaturberechnung über mehrere Zeitpunkte hinweg charakterisiert.

Die vorliegende Arbeit erweitert das Testen mit Rücksetzpunkten zu einem integrierten Test- und Diagnoseprozess. Es wird gezeigt, dass ein einziger Testdurchlauf mit stark kompaktierten Referenzwerten genügt, um eine vergleichbare diagnostische Auflösung zu erreichen wie bei einem Test ohne Antwortkompaktierung.

## Abstract

Robust circuits can compensate certain faults, but they also pose new challenges for test and diagnosis. To avoid unnecessary yield loss, critical faults must be distinguished from faults that can be tolerated during system operation. Furthermore, efficient diagnosis procedures are needed to support yield ramp-up in the case of critical faults. Previous work on circuits with time redundancy has shown that “signature rollback” can distinguish between critical permanent faults and non-critical transient faults. The test is partitioned into shorter sessions, and a rollback is triggered immediately after each faulty session. If the repeated session shows the correct result, then a transient fault is assumed. The reference values for the sessions can be represented in a very compact format. It has been shown that storing only a few bits characterizing the MISR state over time can provide the same quality as storing the complete signatures.

In this work the signature rollback scheme is extended to an integrated test and diagnosis procedure. It is shown, that a single test run is sufficient to reach a diagnostic resolution comparable to a diagnosis procedure without response compaction.

## 1 Einleitung

Zunehmende Parameterschwankungen bei Fertigungstechnologien im Nanometerbereich haben zur Entwicklung neuer robuster und selbstkalibrierender Architekturen geführt [6][9][12][26]. So lassen sich beispielsweise im RAZOR-Prozessor Verzögerungsfehler und transiente Störungen mit Hilfe zusätzlicher Schatten-Latches erkennen und korrigieren [9][12]. Durch einen Neustart der Pipeline mit den korrigierten Werten wird damit auch im Fehlerfall ein korrektes Rechenergebnis sichergestellt. Ebenfalls mit Zeitredundanz arbeitet die von Nikolaidis vorgestellte GRAAL-Architektur für pegelgesteuerte Entwürfe [22]. Weitere Arbeiten versuchen beispielsweise Alterungseffekte rechtzeitig zu erkennen und kompensieren [20][21].

Ein robuster Entwurf erlaubt es, die Vorteile neuer Technologien besser auszunutzen, gleichzeitig ergeben

sich jedoch eine ganze Reihe neuer Herausforderungen für Test und Diagnose. Einerseits können beim Test kritische Fehler durch den robusten Entwurf maskiert werden. Andererseits kann ein strukturorientierter Test kombiniert mit Maßnahmen zur Verbesserung der Testbarkeit zu unnötigen Ausbeuteverlusten führen, wenn im Test Fehler angezeigt werden, die sich im Systembetrieb kompensieren lassen. Testverfahren für robuste Schaltungen müssen also möglichst gut zwischen kritischen und unkritischen Fehlern unterscheiden können. Um eine optimale Interpretation der Testergebnisse und eine Verbesserung des Fertigungsprozesses zu unterstützen, ist außerdem ein effizientes Zusammenspiel von Test und Diagnose notwendig.

In [2] wurde ein Selbsttestverfahren auf der Grundlage der STUMPS-Architektur vorgestellt, das zwischen transienten und permanenten Fehlern unterscheiden und somit

Ausbeuteverluste bei Schaltungen mit Zeitredundanz reduzieren kann [4]. Der Test wird in  $N$  Sitzungen unterteilt und nach jeder Sitzung wird das Testergebnis mit einem abgespeicherten Referenzwert verglichen. Dabei genügt es, statt der vollständigen Signatur für die Sitzung eine kurze Bitfolge abzuspeichern, die den MISR-Inhalt über mehrere Zeitpunkte hinweg charakterisiert [16][17]. Bei einer Abweichung vom Referenzwert wird die betroffene Sitzung sofort wiederholt. Falls beim zweiten Durchlauf kein Fehler mehr auftritt, geht man von einem transienten Fehler aus. Andernfalls wird der Fehler als permanent eingestuft, und der Chip wird aussortiert.

Tatsächlich kann ein erneuter Fehler im zweiten Durchlauf einer Sitzung sowohl auf einen permanenten als auch auf einen weiteren transienten Fehler zurückzuführen sein. In beiden Fällen ist jedoch eine genaue Diagnose zur Verbesserung des Produktionsprozesses wichtig, denn auch eine Anhäufung von transienten Fehlern kann auf besonders anfällige Strukturen in der Schaltung hinweisen.

Die vorliegende Arbeit kombiniert deshalb das oben beschriebene Testen mit Rücksetzpunkten mit einem Verfahren zur direkten Diagnose, das sich nicht auf Haftfehler beschränkt, sondern die Analyse beliebiger Defekte und insbesondere auch transienter sowie intermittierender Fehler unterstützt [8]. Es wird gezeigt, dass damit in einem einzigen Testdurchlauf die gleiche diagnostische Auflösung erreicht wird wie bei einem Test ohne Ausgabekompaktierung.

Die Arbeit fasst in Kapitel 2 zunächst die notwendigen Grundlagen zu Test, Diagnose und Fehlermodellierung kurz zusammen. Anschließend wird in Kapitel 3 das neue integrierte Test- und Diagnoseverfahren vorgestellt. Zum Abschluss werden die Experimente zur Validierung des Verfahrens in Kapitel 4 ausführlich beschrieben.

## 2 Stand der Technik

### 2.1 Selbsttest mit Rücksetzpunkten

Der Selbsttest mit Rücksetzpunkten beruht auf der in Bild 1 skizzierten Standardarchitektur STUMPS [4]. Die vom Mustergenerator TMG erzeugte Testmenge  $T$  kann sowohl zufällige Muster als auch deterministische Muster umfassen, wobei für die deterministischen Muster geeignete Startwerte für TMG abgespeichert werden müssen. Die Testantworten werden mit einem MISR-Register kompaktiert.

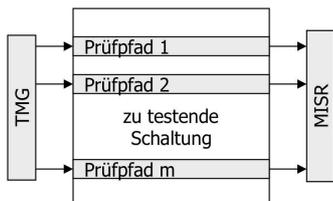


Bild 1: STUMPS-Architektur

Wie oben bereits beschrieben wird der Test  $T$  in  $N$  Sitzungen  $T_1, \dots, T_N$  unterteilt. Um transiente von permanenten Fehlern zu unterscheiden, werden Sitzungen mit fehlerhaftem Ergebnis wiederholt. In [2] wurde gezeigt, dass mit wachsendem  $N$  die zu erwartende Ausbeuteverbesserung steigt und die zu erwartende Testzeit abnimmt. Allerdings steigen mit wachsendem  $N$  auch die Kosten für das Abspeichern der Referenzwerte. Mit Hilfe des in [2] entwickelten analytischen Modells lässt sich  $N$  so bestimmen, dass für eine gegebene Fehlerrate  $\lambda$  der beste Kompromiss zwischen Testqualität und Kosten erreicht wird. Die Kosten können noch weiter reduziert werden, wenn als Referenzwerte keine vollständigen Signaturen abgespeichert werden, sondern kurze Paritätsfolgen, die den Zustand des MISR-Registers über mehrere Zeitpunkte hinweg charakterisieren [16][17]. Bild 2 zeigt die vollständige Architektur für einen „Mixed-Mode“ Test mit zufälligen und deterministischen Mustern.

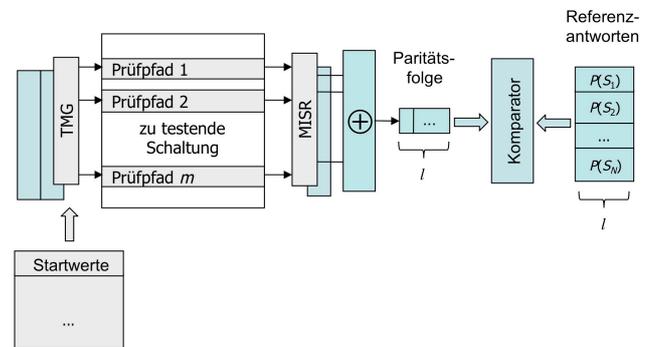


Bild 2: Selbsttest mit Rücksetzpunkten und kompaktierten Referenzantworten [16][17].

Zu Beginn einer Sitzung  $T_j$  wird der Inhalt des MISR-Registers in einem Backup-Register gesichert, um eine spätere Wiederholung der Sitzung zu ermöglichen. Bei deterministischen Mustern wird außerdem der Muster-generator mit dem entsprechenden Startwert initialisiert. Bei zufälligen Mustern läuft der Muster-generator einfach weiter, und der Startwert wird in einem Backup-Register abgespeichert. Wenn am Ende einer Sitzung die Signatur im MISR steht, wird das MISR noch  $l$ -mal getaktet und die erhaltene Folge von Paritätsbits mit der abgespeicherten Folge für den fehlerfreien Fall verglichen. In [16][17] wurde gezeigt, dass durch die Fehlerfortpflanzung im MISR bereits für kleine Werte von  $l$ , wie etwa für  $l = 8$ , in der Regel die gleiche Testqualität wie beim Auswerten der vollständigen Signaturen erreicht wird. Damit lässt sich der Hardware-Aufwand für das Abspeichern der Referenzwerte deutlich reduzieren. Pro Sitzung sind jedoch  $l$  zusätzliche Takte für das Berechnen der Paritätsfolge erforderlich.

### 2.2 Bedingte Haftfehler

Um beliebige Defekte und insbesondere auch transiente Fehler behandeln zu können, wird das bedingte Haftfehlermodell eingesetzt [15]. Ein bedingter Haftfehler  $cond\_0\_v$  bzw.  $cond\_1\_v$  setzt den Knoten  $v$  auf den Wert

0 bzw. 1, wenn die Bedingung *cond* erfüllt ist. Dabei beschreibt *cond* beliebige logische oder zeitliche Bedingungen. Zum Beispiel entspricht  $(v=1)_0_v$  einem permanenten Haftfehler und  $(v=1)_0_v \wedge (v=1)_1_v$  einem Verzögerungsfehler („slow-to-rise“).

Um transiente Fehler zu beschreiben, wird in *cond* ein spezielles Muster in einer Musterfolge  $P = (p_1, \dots, p_n)$  spezifiziert. Bei Eingängen  $x = (x_1, \dots, x_n)$  bedeutet  $(x=p_i | P)_0_v$ , dass bei Anlegen der Musterfolge  $P$  der Knoten  $v$  durch das Muster  $p_i$  auf Null gesetzt wird. Damit lassen sich auch transiente Fehler beschreiben, die bei bestimmten Musterkombinationen ausgelöst werden (z. B. wegen Spannungsabfall oder anderen dynamischen Parameterschwankungen).

### 2.3 Diagnose

Zur eingebauten Diagnose gibt es umfangreiche Literatur, die hier nicht im Detail wiedergegeben werden soll. Grundsätzlich wird zwischen indirekter und direkter Diagnose unterschieden. Bei der indirekten Diagnose werden aus den kompaktierten Testergebnissen zunächst die Prüfpfadelemente mit fehlerhaften Werten identifiziert, wozu in der Regel bereits mehrere Testdurchläufe notwendig sind [5][13][18][19][23][24][25]. Anschließend werden die fehlerhaften Elemente in der Schaltung mit Algorithmen zur kombinatorischen Fehlerdiagnose lokalisiert [1][3][10][27]. Bei der direkten Diagnose werden die fehlerhaften Schaltungselemente direkt aus den kompaktierten Testergebnissen bestimmt [7]. Die meisten bisher bekannten Verfahren zur direkten Diagnose benötigen jedoch mehrere Testdurchläufe, können nur einfache Haftfehler behandeln oder sind nicht mit der Standardarchitektur STUMPS kompatibel [7][11][28].

In [8] wurde erstmals ein Ansatz zur direkten Diagnose vorgestellt, der auf STUMPS beruht und in einem Testdurchlauf eine hohe diagnostische Auflösung für beliebige Defekte erreicht. Ein Test  $T$  wird ähnlich wie beim Testen mit Rücksetzpunkten in  $N$  Teilsitzungen  $T_1, \dots, T_N$  unterteilt. Der Diagnosealgorithmus berechnet aus den Signaturen der Teilsitzungen eine sortierte Liste von möglichen Fehlerorten, die ein beobachtetes Fehlverhalten erklären können. Die Grundidee dazu wird im Folgenden kurz zusammengefasst.

Wenn die beobachtete Signatur  $S_{obs}(T_j)$  einer Sitzung  $T_j$  von der Referenzsignatur  $S_{ref}(T_j)$  abweicht, werden für jeden Schaltungsknoten  $v$  die bedingten Haftfehler  $(x=p_1 | T_j)_0_v, \dots, (x=p_n | T_j)_0_v$  sowie  $(x=p_1 | T_j)_1_v, \dots, (x=p_n | T_j)_1_v$  untersucht, wobei  $x$  wie oben die Eingänge der Schaltung und  $p_1, \dots, p_n$  die Testmuster in  $T_j$  bezeichnen. Jeder einzelne Fehler  $(x=p_i | T_j)_0_v$  bzw.  $(x=p_i | T_j)_1_v$  bewirkt eine Abweichung  $d(p_i, 0)$  bzw.  $d(p_i, 1)$  von der Referenzsignatur, die vorab berechnet werden kann. Die linearen Gleichungen

$$c_1 d(p_1, 0) \oplus \dots \oplus c_n d(p_n, 0) = S_{obs}(T_j) \oplus S_{ref}(T_j)$$

und

$$c_1 d(p_1, 1) \oplus \dots \oplus c_n d(p_n, 1) = S_{obs}(T_j) \oplus S_{ref}(T_j)$$

mit den Variablen  $c_1, \dots, c_n \in \{0,1\}$  beschreiben alle möglichen Bedingungen, welche das beobachtete Fehlverhalten mit einem Fehler am Knoten  $v$  erklären. Ein Fehler am Knoten  $v$  kommt also nur dann als Ursache des Fehlverhaltens in Frage, wenn mindestens eine der beiden Gleichungen eine Lösung hat. Die Anzahl der Teilsitzungen, in denen ein Knoten  $v$  als möglicher Fehlerort identifiziert wird, bestimmt die Position in der sortierten Kandidatenliste.

### 3 Integration von Test und Diagnose

Sowohl das in Abschnitt 2.1 beschriebene Testen mit Rücksetzpunkten als auch das Diagnoseverfahren aus Abschnitt 2.3 arbeitet mit einer Aufteilung des Tests in Teilsitzungen  $T_1, \dots, T_N$ . Beim Testen mit Rücksetzpunkten werden jedoch statt der vollständigen Signaturen nur kurze Paritätsfolgen ausgewertet. Allerdings ist dafür eine zusätzliche Paritätslogik notwendig und das Weiterarbeiten des MISRs nach jeder Sitzung bedingt eine Verlängerung der Testzeit um  $l$  Takte pro Sitzung. Um beide Verfahren in einen effizienten Test- und Diagnoseprozess zu integrieren werden deshalb die nachfolgend beschriebenen Anpassungen vorgenommen.

Damit der Diagnosealgorithmus weiterhin die einzelnen Sitzungen unabhängig voneinander behandeln kann, wird das MISR zu Beginn jeder Sitzung auf Null initialisiert. Das bisher zum Speichern des MISR-Startzustands genutzte Backup-Register wird jetzt zu einem Schatten-MISR erweitert und am Ende der Sitzung eingesetzt. Die berechnete Signatur wird in das Schatten-MISR geladen, das dann noch so lange weiterläuft bis das MISR das erste Muster der nächsten Sitzung kompaktiert hat. Zum einen läuft damit im fehlerfreien Fall der Test ohne jegliche Verzögerung ab. Zum anderen kann auch die Paritätslogik eingespart werden. Dies lässt sich leicht einsehen, wenn man wie in [16][17] die Fehlerauswirkung im MISR betrachtet.

Ein Zustandsübergang im MISR entspricht einer linearen Transformation, die sich mit einer Matrix  $H$  beschreiben lässt. Wenn wie in Bild 3 gezeigt im Zeitpunkt  $t = 0$  ein Fehler  $e$  auftritt, wird  $d^i + e$  statt  $d^i$  mit dem MISR verknüpft.

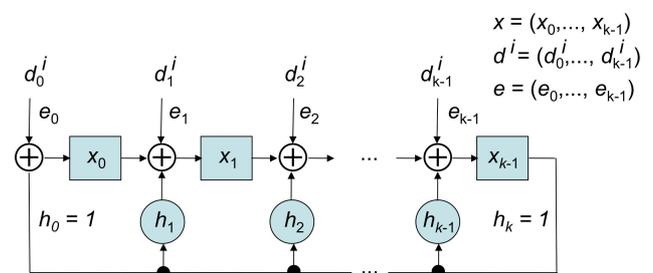


Bild 3: MISR

Die nachfolgenden Zustände im MISR unterscheiden sich um  $He, H^2e, H^3e, \dots$  von den jeweiligen fehlerfreien Zuständen. Die Fehlerauswirkung entwickelt sich also wie

die Zustandsfolge eines autonomen Schieberegisters mit Rückkopplungspolynom  $h(X)$  und Anfangszustand  $e$ . Bei primitivem  $h(X)$  durchläuft jedes einzelne Zustandsbit eine Pseudozufallsfolge, d. h. der Fehler  $e$  macht sich in jedem Zeitpunkt mit Wahrscheinlichkeit  $\frac{1}{2}$  an einem Zustandsbit bemerkbar. Werden  $l$  Zustandsbits beobachtet, macht sich der Fehler mit Wahrscheinlichkeit  $1 - 2^{-l}$  bemerkbar.

Obwohl diese Wahrscheinlichkeit bereits für kleines  $l$  nahe an 1 liegt, können in ungünstigen Fällen Fehler maskiert werden. Wenn beispielsweise die Zustandsbits  $x_{k-1}, \dots, x_{k-l}$  beobachtet werden und sich der Fehler  $e$  anfangs nur am Zustandsbit  $x_0$  auswirkt, werden mindestens  $k - l$  Takte benötigt, bis der Fehler beobachtet werden kann. Dieses Problem wurde in [16][17] durch die Paritätslogik über alle Zustandsbits abgefangen. Da hier aber wie oben beschrieben das Schatten-MISR so lange getaktet wird bis das erste Muster der nächsten Sitzung vollständig kompaktiert ist, bleibt ausreichend Zeit zur Fehlerfortpflanzung und es genügt  $l$  Zustandsbits zu beobachten. Die entsprechende Architektur ist in Bild 3 gezeigt.

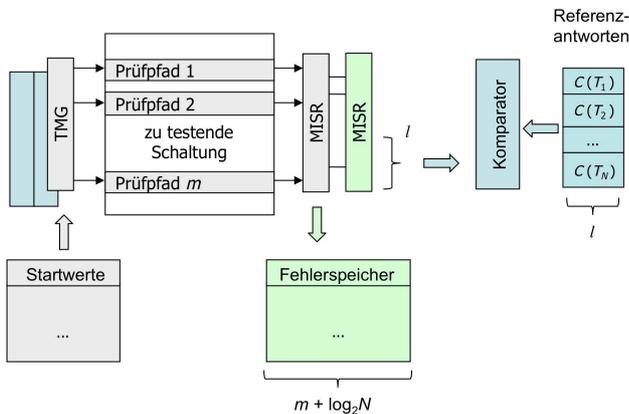


Bild 3: Architektur für den robusten Test mit Diagnose.

Die Hardware für den Test ist durch den Wegfall der Paritätslogik sogar noch reduziert worden. Für die Diagnose wird lediglich noch ein Fehlerspeicher hinzugefügt, der im Fehlerfall die vollständige Signatur sowie die Nummer der Teilsitzung aufnehmen kann.

## 4 Experimentelle Validierung

Um das in Abschnitt 3 beschriebene Test- und Diagnoseverfahren zu evaluieren, wurden mehrere Experimente für industrielle Schaltungen durchgeführt. Dabei wurden die Fehlerfassung, die diagnostische Auflösung sowie der Hardware-Aufwand untersucht. Tabelle 1 fasst die Eigenschaften der von NXP zur Verfügung gestellten Schaltungen zusammen. Für jede Schaltung sind die Anzahl der Gatter, die Anzahl der Prüfpfadelemente (# PPO), die Anzahl der Prüfpfade, die Länge des längsten Prüfpfades und Zahl der Haftfehler angegeben.

Wie in Abschnitt 3 beschrieben wurden als Referenzantworten nicht die vollständigen Signaturen sondern nur kurze Bitfolgen abgespeichert. Dafür wurde das Schatten-

MISR noch für ein ganzes Muster weiter simuliert bzw. die entsprechenden Linearkombinationen aus den Signaturbits am Ende der Sitzung berechnet. Tabelle 2 vergleicht die Fehlererfassung für Haftfehler, die sich damit erreichen lässt, mit der Fehlererfassung bei einem Test ohne Kompaktierung. In beiden Fällen wurden die gleichen mit einem kommerziellen Testmustergenerator erzeugten Muster verwendet.

Tabelle 1: Schaltungseigenschaften

Schaltung	# Gatter	#PPO	# Prüfpfade	max. Länge	# Haftfehler
p100k	84356	5829	270	53	162129
p141k	152808	10502	264	45	283548
p239k	224597	18495	260	61	455992
p259k	298796	18495	360	61	607536
p267k	239687	16621	260	62	366871
p269k	239771	16621	360	62	371209
p279k	257736	17835	385	59	493844
p286k	332726	17835	385	60	648044
p295k	249747	18521	330	62	472124
p330k	312666	17468	320	64	540758
p378k	341315	17420	325	64	816534

Tabelle 2: Fehlererfassung

Schaltung	# Muster	Fehlerfassung ohne Kompaktierung	Fehlererfassung mit Kompaktierung ( $l = 8$ )
p100k	5397	99.56%	99.55%
p141k	5642	98.86%	98.86%
p239k	4778	98.84%	98.83%
p259k	4919	99.10%	99.10%
p267k	5191	99.60%	99.58%
p269k	5164	99.60%	99.59%
p279k	5360	97.89%	97.87%
p286k	6224	98.34%	98.33%
p295k	7916	99.15%	99.14%
p330k	9165	98.95%	98.93%
p378k	664	100.00%	100.00%

Ein Vergleich der beiden letzten Spalten zeigt, dass mit dem vorgeschlagenen Verfahren bereits für  $l = 8$  annähernd die gleiche Fehlererfassung wie ohne Antwortkompaktierung erreicht wird.

Im Hinblick auf die diagnostische Auflösung wurden neben Haftfehlern auch Crosstalk- und Verzögerungsfehler sowie Brückenfehler („Wired And“) untersucht. Von jeder Fehlerart wurden jeweils 100 Fehler zufällig in die Schaltung injiziert. Die Testmuster wurden wie im ersten Experiment mit einem kommerziellen Testmuster-generator erzeugt. Außerdem wurde ein Fehlerspeicher fester Größe für 50 Einträge angenommen. Ein Fehler wird als korrekt diagnostiziert bezeichnet, wenn er vom Diagnosealgorithmus als einziger Kandidat an die erste Stelle der Kandidatenliste gesetzt wird. Entsprechend beschreibt die diagnostische Auflösung den Prozentsatz aller Fehler, die korrekt diagnostiziert wurden. Tabelle 3 zeigt zunächst die diagnostische Auflösung, die bei einem Test ohne jegliche Antwortkompaktierung erreicht wird. D. h. es werden für jedes Muster die vollständigen Testantworten ausgewertet („Bypass“).

Tabelle 3: Diagnostische Auflösung ohne Kompaktierung („Bypass“)

Schaltung	Haftfehler	Crosstalk	Verzögerungsfehler	Wired-And
p100k	70%	68%	76%	75%
p141k	83%	61%	79%	67%
p239k	80%	76%	85%	82%
p259k	78%	70%	82%	77%
p267k	79%	63%	70%	69%
p269k	72%	66%	74%	75%
p279k	67%	60%	73%	67%
p286k	76%	56%	67%	68%
p295k	66%	45%	47%	54%
p330k	71%	65%	72%	71%
p378k	87%	91%	95%	93%

Die diagnostische Auflösung für das in Abschnitt 3 vorgeschlagene Verfahren ist in Tabelle 4 aufgelistet. Hier wurden jeweils 32 Testmuster zu einer Sitzung zusammengefasst und  $l = 8$  Bits pro Sitzung ausgewertet. Ein Vergleich der beiden Tabellen zeigt, dass mit dem vorgeschlagenen Verfahren in fast allen Fällen eine etwas höhere diagnostische Auflösung erreicht wird. Dies ist auf die feste Größe des Fehlerspeichers zurückzuführen. Während im „Bypass“-Modus bei jedem Muster mit einer fehlerhaften Antwort ein Eintrag in den Fehlerspeicher erfolgt, gibt es beim vorgeschlagenen Diagnoseverfahren nur für jede Teilsitzung einen Eintrag. So kann insgesamt mehr Fehlerinformation im Fehlerspeicher abgelegt und eine höhere Auflösung erzielt werden.

Tabelle 4: Diagnostische Auflösung mit Kompaktierung (32 Testmuster pro Teilsitzung,  $l = 8$ )

Schaltung	Haftfehler	Crosstalk	Verzögerungsfehler	Wired-And
p100k	83%	67%	78%	76%
p141k	85%	62%	76%	78%
p239k	86%	77%	85%	83%
p259k	79%	69%	83%	77%
p267k	87%	64%	75%	78%
p269k	83%	64%	78%	85%
p279k	79%	56%	68%	73%
p286k	79%	56%	69%	70%
p295k	73%	51%	53%	58%
p330k	71%	66%	73%	73%
p378k	87%	91%	95%	93%

Dies wird in Tabelle 5 noch etwas genauer quantifiziert. Sowohl für die Diagnose ohne jegliche Kompaktierung („Bypass“) als auch für das Verfahren aus Abschnitt 3 sind die Durchschnittswerte über alle Fehlerarten aufgeführt. Die Spalte „Verbesserung“ zeigt noch einmal deutlich, dass mit dem vorgeschlagenen Verfahren die diagnostische Auflösung in keinem Fall schlechter und in fast allen Fällen besser wird. In einigen Fällen wird sogar eine Verbesserung um bis zu 5,8 % erzielt.

Zur Bewertung der Hardware-Kosten wurde der Speicherumfang von Antwort- und Fehlerspeicher mit dem Speicherplatzbedarf für die Startwerte von TMG bei deterministischen Testmustern verglichen. Als Vergleichswerte dienen dabei die Werte aus [14], da dieses Verfahren eine sehr effiziente Kodierung der deterministi-

schen Testmuster erlaubt. Tabelle 6 zeigt den Speicherplatzbedarf für die deterministischen Muster in KB und den Speicherumfang von Antwort- und Fehlerspeicher relativ dazu. Für die meisten Schaltungen sind die Kosten für Startwerte sowie Antwort- und Fehlerspeicher vergleichbar, d. h. der Mehraufwand für Antwort und Fehlerspeicher liegt unter 2 %. Nur bei der Schaltung p100k liegt der Mehraufwand über 5 %, was aber auf die besonders gute Kodierbarkeit der Eingabemuster zurückzuführen ist.

Tabelle 5: Diagnostische Auflösung - Vergleich

Schaltung	kompaktiert (32 Muster pro Teilsitzung)		„Bypass“ Durchschnitt
	Durchschnitt	Verbesserung	
p100k	76.0%	+3.8	72.2%
p141k	75.2%	+2.8	72.5%
p239k	82.8%	+2.0	80.8%
p259k	77.0%	+0.2	76.8%
p267k	76.0%	+5.8	70.2%
p269k	77.5%	+5.8	71.8%
p279k	69.0%	+2.2	66.8%
p286k	68.5%	+1.8	66.8%
p295k	58.8%	+5.8	53.0%
p330k	70.8%	+1.0	69.8%
p378k	91.5%	0.0	91.5%

Tabelle 6: Hardware-Kosten bei 32 Mustern pro Teilsitzung ( $l = 8$ )

Schaltung	Startwerte für TMG ([14]) [KB]	Mehraufwand (%)
p100k	7.25	5.18%
p141k	36.18	1.06%
p239k	17.97	1.98%
p259k	23.54	1.53%
p267k	47.95	0.77%
p269k	47.44	0.78%
p279k	48.37	0.77%
p286k	63.69	0.63%
p295k	–	–
p330k	76.56	0.64%
p378k	–	–

## 5 Zusammenfassung und Ausblick

Das Testen mit Rücksetzpunkten kann bei Schaltungen mit Zeitredundanz unkritische transiente Fehler identifizieren und somit Ausbeuteverluste reduzieren. Es wurde gezeigt, dass sich das Verfahren sehr einfach mit einer hochauflösenden Diagnose kombinieren lässt. Insbesondere genügt es auch für Diagnose, mit extrem kompaktierten Referenzantworten zu arbeiten. Dadurch bleibt der Speicheraufwand für Antwort- und Fehlerspeicher im gleichen Bereich wie der Speicheraufwand für die Startwerte zur deterministischen Testerzeugung.

## 6 Literatur

- [1] M. Abramovici and M. A. Breuer, “Fault diagnosis based on effect cause analysis: An introduction,” Proc. 17th Design Automation Conf. (DAC’80), Minneapolis, MN, USA, 1980, pp. 69-76.

- [2] U. Amgalan, C. Hachmann, S. Hellebrand, and H.-J. Wunderlich, "Signature Rollback - A Technique for Testing Robust Circuits," Proc. IEEE VLSI Test Symposium, San Diego, CA, USA, April 2008, pp. 125-130.
- [3] M. E. Amyeen, D. Nayak, and S. Venkataraman, "Improving precision using mixed-level fault diagnosis," Proc. IEEE Int. Test Conference (ITC'06), Santa Clara, CA, USA, 2006, pp. 1-10.
- [4] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," Proc. IEEE Int. Test Conference (ITC'82), Philadelphia, PA, USA, Nov. 1982, pp. 200-204.
- [5] I. Bayraktaroglu and A. Orailoglu, "Gate Level Fault Diagnosis in Scan-Based BIST," Proc. Design, Automation and Test in Europe (DATE'02), Paris, France, 2002, pp. 376-381.
- [6] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," IEEE Micro, Nov.-Dec. 2005, pp. 10-16.
- [7] W.-T. Cheng, M. Sharma, T. Rinderknecht, L. Lai, and C. Hill, "Signature based diagnosis for logic BIST," in Proc. IEEE Int. Test Conference (ITC'06), Santa Clara, CA, USA, 2006, pp. 1-9.
- [8] A. Cook, M. Elm, H.-J. Wunderlich, U. Abelein, "Structural In-Field Diagnosis for Random Logic Circuits," to appear in Proc. European Test Symposium (ETS'11), Trondheim, May 2011.
- [9] S. Das et al., "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", IEEE Journal of Solid-State Circuits (JSSC), April 2006, pp. 792-804.
- [10] R. Desineni, O. Poku, and R. D. S. Blanton, "A logic diagnosis methodology for improved localization and extraction of accurate defect behavior," Proc. IEEE Int. Test Conference (ITC'06), Santa Clara, CA, USA, 2006, pp. 1-10.
- [11] M. Elm and H.-J. Wunderlich, "BISD: Scan-Based Built-In Self-Diagnosis," Proc. Design Automation and Test in Europe (DATE'10), Dresden, Germany, March 8-12, 2010.
- [12] D. Ernst, et al., "Razor: Circuit-Level Correction of Timing Errors for Low Power Operation," IEEE Micro, Vol. 24, No. 6, Nov.-Dec. 2004, pp. 10-20.
- [13] J. Ghosh-Dastidar and N. A. Touba, "A rapid and scalable diagnosis scheme for BIST environments with a large number of scan chains," Proc. 18th IEEE VLSI Test Symposium (VTS'00), Montreal, Canada, 2000, pp. 79-85.
- [14] A.-W. Hakmi, S. Holst, H.-J. Wunderlich, J. Schlöffel, F. Hapke, A. Glowatz, "Restrict encoding for mixed-mode BIST," Proc. 27th IEEE VLSI Test Symposium (VTS'09), Santa Cruz, CA, USA, 2009, pp. 179-184.
- [15] S. Holst and H.-J. Wunderlich, "Adaptive debug and diagnosis without fault dictionaries," Proc. 12th European Test Symposium (ETS'07), Freiburg, Germany, 2007, pp. 7-12.
- [16] T. Indlekofer, M. Schnittger, S. Hellebrand, "Efficient Test Response Compaction for Robust BIST Using Parity Sequences," Proc. 28th IEEE Int. Conference on Computer Design (ICCD'10), Amsterdam, The Netherlands, October 2010.
- [17] T. Indlekofer, M. Schnittger, S. Hellebrand, "Robuster Selbsttest mit extremer Kompaktierung," 4. GMM/GI/ITG-Fachtagung "Zuverlässigkeit und Entwurf", Wildbad Kreuth, September 2010.
- [18] A. Leininger, M. Goessel, and P. Muhmenthaler, "Diagnosis of scan chains by use of a configurable signature register and error-correcting codes," Proc. of the Design, Automation and Test in Europe (DATE'04), Paris, France, 2004, pp. 1302-1307.
- [19] C. Liu, K. Chakrabarty, and M. Goessel, "An interval-based diagnosis scheme for identifying failing vectors in a scan-BIST environment," Proc. Design, Automation and Test in Europe (DATE'02), Paris, France, 2002, pp. 382-386.
- [20] S. Mitra, "Globally Optimized Robust Systems to Overcome Scaled CMOS Reliability Challenges," Proc. Design Automation and Test in Europe (DATE'08), Munich, Germany, March 2008, pp. 941-946.
- [21] S. Nassif, "Delay Variability: Sources, impact and trends," Proc. IEEE Int. Solid-State Circuits Conference (ISSCC'00), 2000, pp. 368-369.
- [22] M. Nicolaidis, "GRAAL: A New Fault Tolerant Design Paradigm for Mitigating the Flaws of Deep Nanometric Designs," Proc. IEEE Int. Test Conference (ITC'07), San Jose, USA, Oct. 2007.
- [23] J. Rajski and J. Tyszer, "Diagnosis of scan cells in BIST environment," IEEE Transactions on Computers, Vol. 48, No. 7, July 1999, pp. 724-731.
- [24] J. Rajski, J. Tyszer, C. Wang, and S. Reddy, "Finite memory test response compactors for embedded test applications," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 4, April 2005, pp. 62-634.
- [25] K. K. Saluja and M. Karpovsky, "Testing computer hardware through data compression in space and time," Proc. IEEE Int. Test Conference (ITC'83), Philadelphia, PA, USA, 1983, pp. 83-89.
- [26] D. Sylvester, D. Blaauw, and E. Karl, "ElastIC: An Adaptive Self-Healing Architecture for Unpredictable Silicon," IEEE Design & Test, Vol. 23, No. 6, Nov.-Dec. 2006, pp. 484-490.
- [27] J. Waicukauski and E. Lindbloom, "Failure diagnosis of structured VLSI," IEEE Design & Test of Computers, Vol. 6, No. 4, 1989, pp. 49-60.
- [28] P. Wohl, J. A. Waicukauski, S. Patel, and G. Maston, "Effective diagnostics through interval unloads in a BIST environment," Proc. 39th Design Automation Conference (DAC'02), New Orleans, LA, USA, 2002, pp. 249-254.